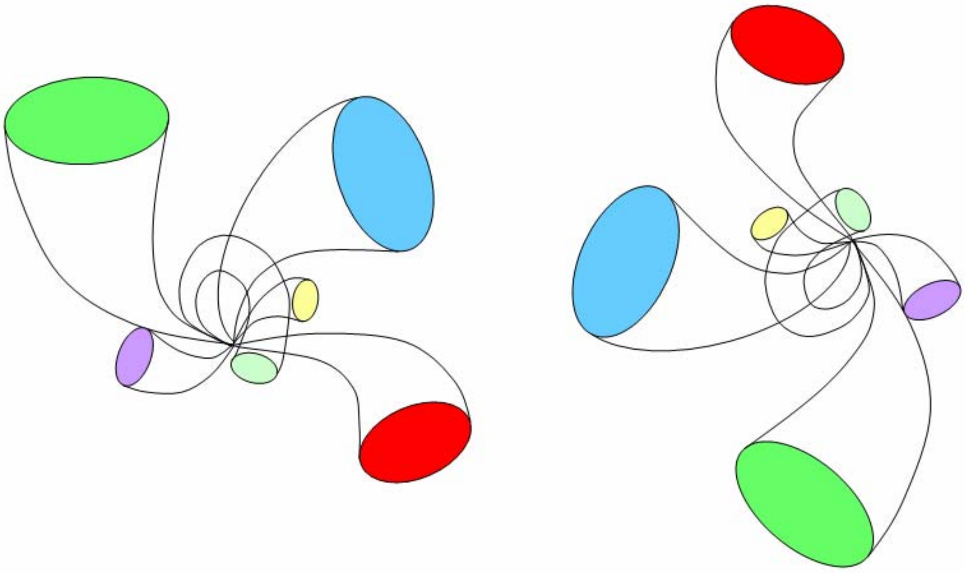
Subspace 2007

ACCV 2007 Workshop



Proceedings of the ACCV 2007 Workshop
Subspace 2007

November 19, 2007

Tokyo, Japan

Subspace 2007 committee

Organizers:

Roberto Cipolla (University of Cambridge),

Kazuhiro Fukui (University of Tsukuba),

Hitoshi Sakano (NTT Data),

David Suter (Monash University)

Program committee:

Sang-Woon Kim (Myongi University),

Xi Li (University of Tsukuba),

Ken-ichi Maeda (Toshiba),

Atsuto Maki (Kyoto University),

Hiroshi Murase (Nagoya University),

Shree K. Nayar (Columbia University),

Erkki Oja (Helsinki University of Technology),

Shinichiro Omachi (Tohoku University),

Seiichi Uchida (Kyushu University),

Shinichi Sato (National Institute of Informatics),

Atsushi Sato (NEC),

Yoichi Sato (University of Tokyo)

Liwei Wang(Peking University)

Preface

Welcome to the Subspace 2007. It gives us great pleasure to bring forth its proceedings.

Subspace methods such as CLAFIC* are not only important theory for solving many pattern recognition problems in computer vision, but also they have been widely used as a practical methodology for a wide variety of real applications. In particular, subspace methods have been studied intensively in the field of character recognition, contributing to a number of commercial optical character recognition (OCR) systems. Although the firstly proposed subspace method is nearly three decades old, a number of new techniques based on the same approach are still proposed every year in emerging fields between computer vision and other related technologies. For example, it is interesting to see that the Mutual Subspace Method, which is one of the most successful variants of subspace methods, is indeed almost identical to the Canonical Correlation Method in multivariate statistical analysis. The concept of subspace methods is also strongly related to the factorization method. Unfortunately, however, the significance of subspace methods is not yet fully recognized in the community of computer vision, despite the notable success of commercial products based on the subspace method.

The goal of this workshop is to share the prominent potential of subspace methods with researchers working on various problems in computer vision, and encourage interactions leading to further developments of subspace methods. The fundamental theories of subspace methods and their applications in computer vision will be discussed at the workshop. We believe that Subspace2007 will stimulate fruitful discussions among participants and provide them novel ideas for future research in computer vision.

* S. Watanabe, N. Pakvasa, "Subspace method in pattern recognition", in Proc. 1st Int. J. Conf on Pattern Recognition, 1973.

Subspace2007 Organizers
Roberto Cipolla (University of Cambridge)
Kazuhiro Fukui (University of Tsukuba)
Hitoshi Sakano (NTT Data)
David Suter (Monash University)

Subspace 2007 Program

9:00-9:30 Registration and poster preperation

9:30-9:40 Opening

9:40-10:40 Oral session 1 *Tracking and pose estimation* Chair D. Suter and S. Uchida

- **SS01** An Adaptive Shape Subspace Model for Level Set-Based Object Tracking, Xue Zhou, Weiming Hu, Xi Li (National Laboratory of Pattern Recognitionm Institute of Automation)
- **SS02** Upgrading Eigenspace-based Prediction using Null Space and its Application to Path Prediction, Yuji Shinomura (Hiroshima Univ.), Toru Tamaki (Hiroshima Univ.), Toshiyuki Amano (NAIST), Kazufumi Kaneda (Hiroshima Univ.)
- **SS03** The Secret of Rotating Object Images – Using Cyclic Permutation for View-based Pose Estimation –, Toru Tamaki (Hiroshima Univ.), Toshiyuki Amano (NAIST), Kazufumi Kaneda (Hiroshima Univ.)

10:40-10:50 Cofee break

10:50-11:50 Oral session 2 *Face and object recognition* Chair S. Sato and H. Sakano

- **SS06** Nonlinear K-subspaces based Appearances Clustering of Objects under Varying Illumination Conditions, Xi Li, Kazuhiro Fukui (Univ. of Tsukuba)
- **SS13** Face Recognition based on Whitening Transformation of Distribution of Subspaces, Tomokazu Kawahara, Masashi Nishiyama, Tatsuo Kozakaya, Osamu Yamaguchi (Toshiba)
- **SS16** Face Recognition Using Mutual Projection of Feature Distributions, Akira Inoue, Atsushi Sato (NEC)

11:50-12:00 Cofee break

12:00-13:00 Oral session 3 *Theory* Chair S. Omachi and A. Sato

- **SS04** Scale-based Principal Component Analysis of Point Cloud, Tomoya Sakai, Atsushi Imiya (Chiba Univ.)
- **SS09** Kernel and Learning Local Manifold Matching for Image Classification, Seiji Hotta (Tokyo Univ. of Agri. and Tech.)
- **SS15** Regularization vs. Rank Reduction in Quadratic Classifiers, Yoshikazu Washizawa (RIKEN)

13:00-15:30 Poster session and lunch (Conjunction with Workshop on Multi-dimensional and Multi-view Image Processing)

SS01 An Adaptive Shape Subspace Model for Level Set-Based Object Tracking, Xue Zhou, Weiming Hu, Xi Li (National Laboratory of Pattern Recognitionm Institute of Automation)

SS02 Upgrading Eigenspace-based Prediction using Null Space and its Application to Path Prediction, Yuji Shinomura (Hiroshima Univ.), Toru Tamaki (Hiroshima Univ.), Toshiyuki Amano (NAIST), Kazufumi Kaneda (Hiroshima Univ.)

SS03 The Secret of Rotating Object Images – Using Cyclic Permutation for View-based Pose Estimation –, Toru Tamaki (Hiroshima Univ.), Toshiyuki Amano (NAIST), Kazufumi Kaneda (Hiroshima Univ.)

SS04 Scale-based Principal Component Analysis of Point Cloud, Tomoya Sakai, Atsushi Imiya (Chiba Univ.)

SS05 Kernel Eigenspace for Detecting and Separating Multi-class Nonlinear Objects, Masud Rahman (TokyoTech Australia), Anand Santhanam (Australian National Unive.), Seiji Ishikawa (Kyushu Inst. of Tech.)

SS06 Nonlinear K-subspaces based Appearances Clustering of Objects under Varying Illumination Conditions, Xi Li, Kazuhiro Fukui (Univ.of Tsukuba)

SS07 Face Recognition based on Holistic Information and Minimum Mahalanobis Classifier, I Gede Pasek Suta Wijaya, Keiichi Uchimura, Zhencheng Hu (Kumamoto Univ.)

SS08 Spherical PCA with Euclideanization, Jun Fujiki, Shotaro Akaho (AIST)

SS09 Kernel and Learning Local Manifold Matching for Image Classification, Seiji HOTTA (Tokyo Univ. of Agri. and Tech.)

SS10 A Study on PCA-based Fourier Descriptor in Complete and Incomplete Contour Representations, Li Tian, Sei-ichiro Kamata (Waseda Univ.)

SS11 ICA-Based Analysis of Temporal Image Sequence, Naoya Ohnishi, Atsushi Imiya (Chiba Univ.)

SS12 A Supervised Method to Chart Multiple Manifolds, Dan Zhang (Tsinghua Univ.), Yangqiu Song (Tsinghua Univ.), Qifeng Qiao (Tsinghua Univ.), Zhenwei Shi (Beihang Univ.), Changshui Zhang (Tsinghua Univ.)

SS13 Face Recognition based on Whitening Transformation of Distribution of Subspaces, Tomokazu Kawahara, Masashi Nishiyama, Tatsuo Kozakaya, Osamu Yamaguchi (Toshiba)

SS14 Personal Favorite Scene Selection from Broadcast Soccer Video using Eigenspace Method, Masao Izumi (Osaka Prefecture Univ.)

SS15 Regularization vs. Rank Reduction in Quadratic Classifiers, Yoshikazu Washizawa(RIKEN)

SS16 Face Recognition Using Mutual Projection of Feature Distributions, Akira Inoue, Atsushi Sato (NEC)

15:30-16:30 **SS00** Tutrial "The Engineer's Guide to The Subspace Method", *Yoshiaki Kurosawa* (Toshiba Solutions) Chair A. Maki and K. Fukui

16:30-16:35 Closing

Table of Contents

The Engineer's Guide to The Subspace Method
- English Version -

Yoshiaki Kurosawa (Toshiba Solutions Corporation)

***Abstract*—The Subspace Method was developed around 1970, and is now used as important basic technology in the pattern recognition field. This report is written as a guide to this field. The Subspace Method itself along with the Weighted Subspace Method, Principal Component Analysis (PCA) as a tool for fixing a subspace, it's relationship to PCA used in Bayesian Discrimination, the relationship between Subspace Method and Bayesian Discrimination, and an incremental learning method as another fixing method of subspace are all explained in this report. Finally, "How to fix a subspace", "How to construct a similarity" and "New targets and applications" are discussed as a future work.**

***Index Terms*—subspace, Principle Component Analysis, Multiple Similarity, CLAFIC, weight**

I. Introduction

THIS is a guide for any beginners and students starting out in the field of pattern recognition and the Subspace Method world. This is the English version of literature [1] written in Japanese and previously presented at the workshop. Therefore, it may be of little interesting for experts in this field and those who attended the former workshop. This report consists of an overview and basic facts about the Subspace Method. For more detail about this field, book [2] is well known as a good textbook. Unfortunately, however it is now out of print.

The definition of the Subspace Method and it's history are first described in this report. The Weighted Subspace Method and Principal Component Analysis as a method for determining a subspace are then explained. These methods have a relationship with Bayesian Discrimination. Finally, an incremental learning method for creating references is also introduced and three typical directions in future research of this field are explained.

The Subspace Method was introduced around 1970 and has been applied to many applications with various modifications.

The concept of the Subspace Method is to consider a pattern cluster as a subspace in a linear space in which a pattern is expressed as a vector. For more precise consideration, it must be treated as a manifold, but it might be natural to treat it a subspace for simple and basic design of pattern recognition systems.

For pattern recognition, it is basic and essential to discriminate clusters which consist of many vectors. Even in NN recognition systems, it is a fact that a linear space is separated by piecewise hyper curved surface. In this sense, the Subspace Method approach is reasonable for discriminating patterns.

II. the Subspace Method

THE definition of the Subspace Method is as follows: Let N be the number of pattern space dimension, that is the number of pattern vector's elements; let φ_i be reference vectors which are normal and orthogonal; let r be the number of reference vectors; and let x be an input vector. The similarity S is defined as

$$S = \sum_{i=0}^{r-1} (x, \varphi_i)^2. \tag{1}$$

The reference vectors are defined for each category and the similarity S is also calculated for each category. A category which should be determined as an answer is a category which has the maximum similarity. Here, the number r is the dimension of the spanned space by φ_i, while the number of N is the total space's dimension. Both are "dimension" but the meanings are different. Care should be taken to ensure they are not confused.

If the vectors φ_i and x are not normalized, the similarity is described as the following equation which is more general:

$$S = \sum_{i=0}^{r-1} (x, \varphi_i)^2 / \|x\|^2 \|\varphi_i\|^2. \tag{2}$$

As for the above mentioned two equations, sometimes we make a mistake in mathematical analysis and computer programming when we confuse (1) with (2). If possible, it is better to use (2).

In this method, a subspace spanned by reference vectors is considered as a reference subspace for a certain category. The input pattern's projection on this subspace is considered as a similarity between the input and the reference. A method which has (2) with $r = 1$ is called Simple Similarity.

If we define a projection P as

$$P = \sum_{i=0}^{r-1} \varphi_i \varphi_i^T, \tag{3}$$

and represent an inner product as $(a, b) = a^T b$, (1) are converted to

$$S = x^T P x. \tag{4}$$

This is a simple and convenient expression.

The meaning of the Subspace Method is that the patterns of a category which exist in some area of the total space is considered as a subspace, and this subspace represents the category. As for Simple Similarity, its reference which consists of one vector represents a one dimensional subspace.

Principal Component Analysis (PCA) is a typical method to obtain this reference subspace. As for the input patterns of a category, their average projection on the reference subspace is maximized when this subspace is obtained by PCA. The method of PCA in this field was introduced as feature reduction or feature selection method before the Subspace Method was proposed.

In 1963, a feature selection method in which the PCA was applied to all patterns in all categories was proposed by Taizo Iijima [3] in Japan at Electrotechnical Laboratory (ETL). ETL was a research laboratory which had been recently united with other laboratories and is now a division of Advanced Industrial

Science and Technology (AIST). The same idea was also proposed by Japanese researcher Satosi Watanabe, the professor at the University of Hawaii [11] who proposed SELFIC. Then the Subspace Method was proposed by Watanabe [7][8][9][10], and this was called CLAFIC.

Independently, Iijima had been developing a similar idea after the propose of PCA for feature reduction, and had been developing a new printed English letter character reader as a member of a national project's OCR development team. This project's target was pattern recognition and had several themes. A high spec printed character reader had been developed by ETL and Toshiba as one of the achievements of this project. This had been started in 1966 and finished in 1970 and the OCR was called ASPET70 (Analog Spatial Processor developed by Electrotechnical laboratory and Toshiba). ASPET71 was then developed as an enhanced version. Multiple Similarity Method proposed by Iijima [4] was used as core technology in these machines.

As for Multiple Similarity Method, it has been widely used in the pattern recognition field, such as printed and handprinted character recognition, kanji (Chinese characters used in Japan) character recognition, speech recognition and image recognition. This method was introduced by the fundamental theory of visual pattern. This theory consists of integral equation, describes image characteristics, and includes the idea of "Gaussian scale space", which was the theory of "blurring." In practical use, this method is equal to the Subspace Method if the weight parameters are all one. Therefore, Multiple Similarity Method can be considered as one of the first research results of the Subspace Method.

Those two pieces of independent research by Watanabe and Iijima reached the Subspace Method world and both research results had a huge affect on later researchers in this field.

The blurred character image pattern was used as a feature pattern in Multiple Similarity Method. But the importance of "blurring" was not known at the time. It was common sense that a pattern must be observed precisely and sharply for better recognition accuracy. The blurring technique was out of question. Nowadays, the "blurring" is recognized as an important technique of a pre-processing method for input image patterns. This is the concept known as Gaussian scale space [6] now. But, in fact, this was introduced by Iijima in 1959.

III. THE WEIGHTED SUBSPACE METHOD

AN important extension of the Subspace Method is the Weighted Subspace Method, which is described by the following equation with the introduction of weight values μ_i.

$$S = \sum_{i=0}^{r-1} \mu_i (x, \varphi_i)^2. \qquad (5)$$

In this equation, a set of weight values determines the characteristics of a recognition system.

The horizontal axis of Fig.1 shows the number of eigen vectors which are ordered by the correspondent eigen value. The eigen values which have a larger value correspond to smaller numbers. The origin is zero here. The weight is one until a certain number and after that it is zero in the Subspace Method. In case of the method which is relevant to Modified Quadratic Discriminant Function (MQDF) [12][13], the weight value gradually falls down to a certain value and there it is zero afterwards. In case of the concept of Compound Similarity [5], the weight values of certain eigen vectors are set to minus. In this case of minus values, the order of eigen vectors is not ruled by the value itself.

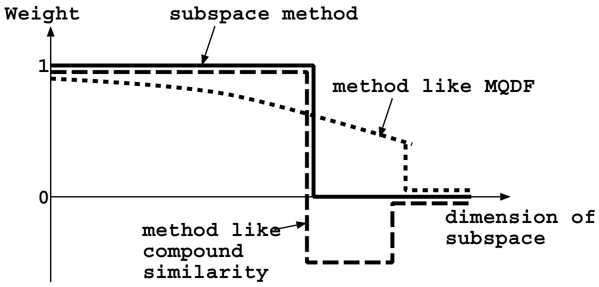


Fig. 1. Example of weight in the Weighted Subspace Method.

The method in which the weight goes down while the number increases indicates a concept that smaller contribution to a similarity for smaller eigen values.

The concept of Compound Similarity is described as follows: First, the difference between similar patterns are represented by a subspace which is made to be orthogonal to the subspace of a correct category. The normalized and orthogonal vectors of this subspace representing the difference are added to the reference vectors. The similarity is obtained by subtracting the projection length of an input pattern on this subspace from the original similarity.

As for MQDF, it is different from the Subspace Method, but they are related to each other. Their relationship is described later.

IV. PRINCIPAL COMPONENT ANALYSIS

IN many cases, the term "Subspace Method" implies that the reference of the Subspace Method is made by PCA. The normalized orthogonal system of reference vectors is defined by eigen vectors obtained by PCA.

Generally, PCA means that principal components are determined as vectors which start from the center point which is the mean vector of pattern distribution. In comparison, the mean vector is not used in the Subspace Method in which the mean vector is treated as zero. The Subspace Method's PCA is different from usual PCA at this point.

Here, we begin by explaining usual PCA used in Quadratic Discriminant Function (QDF).

Let α be the name of a learning target vector and this vector be x_α. Let m be the mean vector of these vectors. Then, let φ_0 be an unit vector which represents the best direction of vectors $x_\alpha - m$. Here, the meaning of "representing the best direction of vectors" is that maximizing the square of inner product of $x_\alpha - m$ and φ_0 . In this case, a condition that $\|\varphi_0\| = 1$ are required, then an unknown multiplier λ_0 is introduced for this condition in the following equation. The vector φ_0 is obtained by PCA as a vector which maximize the evaluation value J in this equation:

$$J = \sum_\alpha (x_\alpha - m, \varphi_0)^2 - \lambda(\|\varphi_0\|^2 - 1)$$

$$= \sum_\alpha \{\sum_j (x_{\alpha j} - m_j)\varphi_{0j}\}^2 - \lambda_0(\sum_j \varphi_{0j}^2 - 1). \qquad (6)$$

The vector φ_0 is obtained by setting the partial derivatives of J to zero. The evaluation value J is differentiated by φ_{0j} which is the element of φ_0.

$$\frac{\partial J}{\partial \varphi_{0j}} = 2\sum_{\alpha}(x_\alpha - m, \varphi_0)(x_{\alpha j} - m_j) - \lambda_0(2\varphi_{0j}) = 0. \quad (7)$$

The following expression is obtained by φ_0, m and x_α which are made by vertically arranging φ_{0j}, m_j and $x_{\alpha j}$.

$$2\sum_{\alpha}(x_\alpha - m, \varphi_0)(x_\alpha - m) - \lambda_0(2\varphi_0) = 0. \quad (8)$$

This is rewritten further as

$$\sum_{\alpha}(x_\alpha - m)(x_\alpha - m)^T \varphi_0 = \lambda_0 \varphi_0. \quad (9)$$

Here, let K be as

$$K = \sum_{\alpha}(x_\alpha - m)(x_\alpha - m)^T. \quad (10)$$

So (9) is expressed by

$$K\varphi_0 = \lambda_0\varphi_0. \quad (11)$$

In conclusion, φ_0 becomes the eigen vector of covariance matrix K. As for $\varphi_i (i > 0)$, the story is the same, and φ_i are obtained as the eigen vectors of (11). The number of eigen vectors are equal to the dimension of total space and are orthogonal to each other.

V. PCA on Hyper Tangent Plane

The method of PCA in the Subspace Method is described as a method to find a projection P which maximize the evaluation value J in the following equation:

$$J = \sum_{\alpha} {x_\alpha}^T P x_\alpha. \quad (12)$$

Here, P is defined as

$$P = \sum_{i=0}^{r-1} \varphi_i {\varphi_i}^T, \quad \|\varphi_i\|^2 = 1. \quad (13)$$

An unknown multiplier λ_i is introduced to the above equation and the following equation is obtained:

$$J = \sum_{\alpha}\sum_{i=0}^{r-1}\{(x_\alpha, \varphi_i)^2 - \lambda_i(\|\varphi_i\|^2 - 1)\}. \quad (14)$$

The vectors φ_i which maximize J above are used as reference vectors. This equation is the same as (6) with $m = 0$ except the summation of i. The eigen vectors φ_i are independent of each other and therefore the eigen equation becomes the same equation (11) by a process which is the same as the previously described calculation.

The result is a covariance matrix K such as,

$$K = \sum_{\alpha} x_\alpha x_\alpha^T, \quad (15)$$

and an eigen equation such as,

$$K\varphi = \lambda\varphi. \quad (16)$$

An input pattern's norm is fixed to one in the Subspace Method. Consequently, patterns are distributed on the surface of a hyper sphere.

Instead of the surface of the hyper sphere, a hyper plane which is tangent to the hyper sphere is used. In Figure 2, as approximation for making it simpler, the patterns are assumed to be distributed on this hyper plane which is orthogonal to an unit vector e, which is the vector $\vec{PQ}$ where P is the origin of the hyper sphere and Q is the tangent point of this hyper plane to the hyper sphere. The point Q is also the center of the distribution.

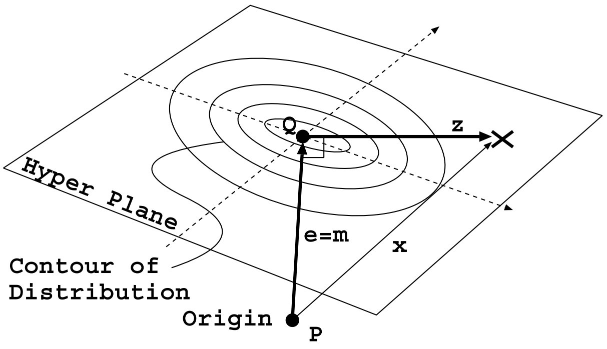


Fig. 2. Example of pattern distribution on hyper plane which is tangential to spherical surface.

In this figure, the mean vector m becomes to equal to the vector e if the distribution is a Gaussian whose center is the end of vector e. The reason is probably almost clear but it is explained briefly below. Let the vector x_α be expressed by $x_\alpha = e + z_\alpha$. The distribution of the vectors z_α which starts from the point Q to the end of x_α is Gaussian. It lies on the hyper plane. Let m be the mean vector of x_α. Then $m = \overline{x_\alpha} = e$ because $\overline{z_\alpha} = 0$. Here, the norm of x_α is not one but must be near one. Consequently, z_α must be small.

The vector m is used instead of e and the norm of x_α is considered as approximately one, hereafter.

If we use usual PCA in this case, the vector m is used as the mean vector and the covariance matrix becomes as follows:

$$K = \sum_{\alpha}(x_\alpha - m)(x_\alpha - m)^T. \quad (17)$$

Consequently, it is converted to,

$$K = \sum_{\alpha} z_\alpha z_\alpha^T, \quad (18)$$

because $x_\alpha = m + z_\alpha$. The equation (18) is similar to (15), but they are different. Here the problem is "Which is better?" or "Which is correct?". In the viewpoint of usual PCA, that is, the viewpoint of Gaussian distribution as statistical assumption, the covariance matrix (18) which is obtained from (17) is correct. Is this true?

The conclusion is simple. Both are the same. The following consideration proves this fact:

The equation (15) is obtained formally by making the mean vector be zero in the covariance matrix in (17) which is equivalent to (18), in spite of the fact that the mean vector is m. It is not possible to make m be zero for this conversion. We need a conversion with m which is not zero. Here, we see below the fact that the mean vector m and the eigen vectors of (18) are equal to the eigen vectors of (15).

It is probably more sophisticated to use the equation,

$$K = \sum_{\alpha} x_\alpha x_\alpha^T - mm^T, \tag{19}$$

but here, it is proved by low-tech style:
(Characteristic 1)
The vector m is the eigen vectors of (15).
(Proof)
The inner product $(z_\alpha, m) = 0$ because the hyper plane and m is perpendicular. Here, $x_\alpha = m + z_\alpha,\ \|m\| = \|e\| = 1$.

$$\sum_{\alpha} x_\alpha x_\alpha^T m = \sum_{\alpha} x_\alpha (x_\alpha, m) = \sum_{\alpha} x_\alpha (z_\alpha + m, m)$$

$$= \sum_{\alpha} x_\alpha (m, m) = \sum_{\alpha} x_\alpha = m, \tag{20}$$

Consequently, m becomes the eigen vector of (15). (End of proof)

(Characteristic 2)
Eigen vectors ψ_i obtained from (18) is the eigen vectors of (15).
(Proof)
Since the vectors ψ_i is the eigen vectors of (18),

$$\sum_{\alpha} z_\alpha z_\alpha^T \psi_i = \lambda_i \psi_i. \tag{21}$$

The vectors ψ_i and m are orthogonal because the hyper plane and m are orthogonal.

In addition to that, $\sum_\alpha z_\alpha = 0$ because the mean vector of Gaussian is the origin of the hyper plane.

$$\sum_{\alpha} x_\alpha x_\alpha^T \psi_i = \sum_{\alpha} x_\alpha (x_\alpha, \psi_i)$$

$$= \sum_{\alpha} (m + z_\alpha)(m + z_\alpha, \psi_i)$$

$$= \sum_{\alpha} m(z_\alpha, \psi_i) + \sum_{\alpha} z_\alpha (z_\alpha, \psi_i)$$

$$= \sum_{\alpha} z_\alpha z_\alpha^T \psi_i = \lambda_i \psi_i. \tag{22}$$

Consequently, ψ_i becomes the eigen vectors of (15) and it's eigen values become λ_i. (End of proof)

It becomes clear from previous consideration that "to adopt the eigen vectors φ_i of (15) as a reference of Subspace Method" and "to adopt the vector m and ψ_i in (18)" are the same. More precisely, previous model is different from the Subspace Method because $m \neq \varphi_0$ and $\|x_\alpha\| \neq 1$. But this consideration is useful for appropriate understanding of PCA in the Subspace Method.

The matrix K in (15) is called correlation matrix or moment matrix. Confusingly, it is also often referred to as covariance matrix.

VI. Multiple Similarity

IN Multiple Similarity Method, it is resembles the Subspace Method in that eigen vectors are obtained by PCA as reference vectors. Let the number of vectors be r. Let this eigen vectors be φ_i, eigen values be λ_i. Multiple Similarity S is given by the following equation:

$$S = \sum_{i=0}^{r-1} \frac{\lambda_i}{\lambda_0} \cdot \frac{(x, \varphi_i)^2}{\|x\|^2 \cdot \|\varphi_i\|^2}. \tag{23}$$

Sometimes, $\sqrt{S}$ is also called as Multiple Similarity.

If r is equal to the dimension of total space N, (23) becomes as,

$$S = \frac{1}{\lambda_0} x^T K x = \frac{1}{\lambda_0} \sum_{\alpha} (x_\alpha, x)^2, \tag{24}$$

where,

$$K = \sum_{\alpha} x_\alpha x_\alpha^T = \sum_{i=0}^{N-1} \lambda_i \varphi_i {\varphi_i}^T. \tag{25}$$

The equation (24) corresponds to (4).

The meaning of (24) is that similarity is defined as the summation of squared inner product of a input pattern vector and a reference pattern vector. It is a reasonable measure to scale the similarity.

Generally, (23) is called Multiple Similarity, but the term "Multiple Similarity" often means the total concept of "blurring", and the fundamental theory of visual pattern.

VII. Canonicalization

A technique called canonicalization is used in Multiple Similarity Method. This method is for deducting the constant component of an input pattern from itself. It is effective if the number of bits is small for expressing the elements of an input pattern vector. It is expected that this method contributes recognition accuracy, but it has not yet been clarified. Let $\vec{1}$ denote the vector whose elements are all 1. The symbol N express the dimension of input vectors.

The canonicalization in which input vector x is converted to x' is shown below:

$$x' = x - \frac{1}{N}(x, \vec{1})\vec{1}, \tag{26}$$

There is a way of using the mean vector of all category's patterns instead of the vector $\vec{1}$. Eigen vectors which are made from canonicalized patterns has been also canonicalized; that is, canonicalization for reference vectors is not required if they are made of canonicalized vectors and by PCA.

VIII. Bayesian Discrimination

BAYESIAN Discrimination (BD) is also called Quadratic Discriminant Function, and it is abbreviated to QDF in this report. QDF is not the Subspace Method, but it has an important relationship. This relationship is described briefly later. Before this explanation, QDF and MQDF are introduced in this section and the next section.

Let A and B denote two categories, and consider the case of discriminating these two categories. It is possible to obtain a frequency distribution by collecting pattern vectors x which belong to A. The probability $P(x|A)$ is calculated by dividing this frequency distribution by the number of patterns which belong to A. This probability is called likelihood. Bayes' Theory is as follows:

$$P(A|x) = \frac{P(x|A)P(A)}{P(x)}. \tag{27}$$

The probability $P(A|x)$ is called posterior probability and $P(A)$ is prior probability. The probability $P(A|x)$ for category A and $P(B|x)$ for category B are compared to decide which category the input vector x belongs to. This is the basic idea of QDF.

Here, the probability $P(A)$ is assumed to have the same value for any category. In addition, the probability $P(x)$ has the same value when the comparison is executed. Therefore, $P(A)$ and $P(x)$ are negligible in the comparison and the likelihood $P(x|A)$ can be used for discrimination.

It is assumed that the pattern distribution is Gaussian in QDF; that is, the appearance probability of pattern x is defined by the following equation as the likelihood. Here, N is the dimension of an input vector.

$$P(x) = \frac{1}{\sqrt{(2\pi)^N|K|}} exp\{-\frac{1}{2}(x-m)^T K^{-1}(x-m)\}. \tag{28}$$

In this equation, $(x-m)^T K^{-1}(x-m)$ is a quadratic form. It is easy to understand if the case is two dimensional. For a simple example, let m be 0. And let x be $x = (x_0, x_1)^T$ and define K^{-1} as,

$$K^{-1} = \begin{bmatrix} a & c \\ c & b \end{bmatrix}. \tag{29}$$

Then the following equation is obtained:

$$x^T K^{-1} x = ax_0^2 + 2cx_0x_1 + bx_1^2. \tag{30}$$

This shows that a contour obtained by (28) is an ellipse.

The vector m is a mean vector, K is the covariance matrix of the pattern distribution in (28). This m and K are defined for each category and the probability $P(x)$ is calculated for each input pattern x and for each category. In QDF, the category which has the maximum $P(x)$ is an answer category for the input pattern x.

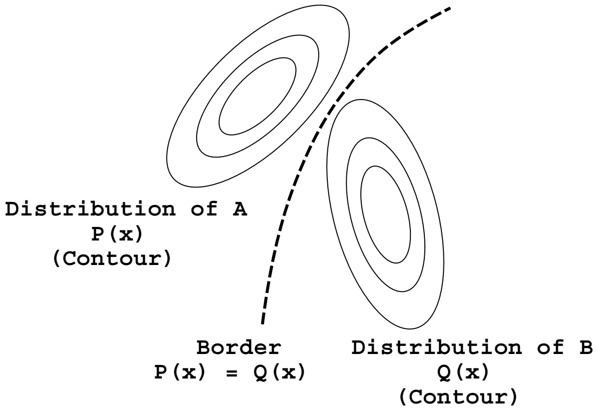


Fig. 3. Two category discrimination by Bayes Discriminant Function.

The example of two category case is shown in figure 3 which displays two pattern distributions on a two dimensional plane.

In this figure, $P(x)$ denotes the appearance probability of the input pattern x which belongs to the category A. Also $Q(x)$ for B. The area defined by $P(x) > Q(x)$ is the area where an input pattern is decided to be A. In addition, the area $P(x) < Q(x)$ means the category B. The line $P(x) = Q(x)$ is the border and is a superquadratic surface.

Let's apply log operation to the both sides of the equal symbol in (28) and modify it. So when,

$$-2logP(x) - Nlog(2\pi) = \\ (x-m)^T K^{-1}(x-m) + log|K|, \tag{31}$$

is obtained. The recognition process is changed to see the minimum value of the right side term in this equation. The equation is called Mahalanobis distance if the last term $log|K|$ is omitted. Next, let φ_i be the eigen vectors of K, λ_i be the eigen values, Φ be the matrix which consists of eigen vectors. The following equation is obtained by expressing K by the eigen vectors.

$$K = \Phi \begin{bmatrix} \lambda_0 & 0 & ... & 0 \\ 0 & \lambda_1 & ... & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & ... & \lambda_{N-1} \end{bmatrix} \Phi^T, \tag{32}$$

Let the right term of (31) be the distance D as follows:

$$D = \sum_{i=0}^{N-1} \frac{1}{\lambda_i}\{(x-m, \varphi_i)^2\} + log(\prod_{i=0}^{N-1} \lambda_i). \tag{33}$$

The actual calculation of QDF is done by this equation. In Mahalanobis distance, the term of log is omitted.

IX. MQDF

As for QDF, the inverse of eigen value λ_i is used in the right side term of (33). If we consider the above mentioned $1/\lambda_i$ a weight, the weight is rising up from the value $1/\lambda_0$.

This is the way to see the total pattern space, whether a subspace includes category's patterns essential components or not. The large value of $1/\lambda_i$ means that less essential components are included and this cause the large distance value. This concept is no good for a case where the eigen values of high dimension are extremely low where the components are mostly noise. This is because the distance calculated in the high dimension where λ_i is extremely small becomes unstable.

To avoid this problem, there is a method in which the weight is considered to be constant where the dimension number i is greater than or equal to certain r in (33) [12][13]. This is MQDF.

Here, let's see how the equation in MQDF is obtained from (33). Firstly, (33) is modified by considering a constant value δ. The eigen values whose number is greater than or equal to r is substituted by δ .

$$D = \sum_{i=0}^{r-1} \frac{1}{\lambda_i}\{(x-m, \varphi_i)^2\} + \sum_{i=r}^{N-1} \frac{1}{\delta}\{(x-m, \varphi_i)^2\} \\ + log(\prod_{i=0}^{r-1} \lambda_i) + log(\prod_{i=r}^{N-1} \delta) \\ = \sum_{i=0}^{r-1} \frac{1}{\lambda_i}\{(x-m, \varphi_i)^2\}$$

$$+\sum_{i=0}^{N-1}\frac{1}{\delta}\{(x-m,\varphi_i)^2\}-\sum_{i=0}^{r-1}\frac{1}{\delta}\{(x-m,\varphi_i)^2\}$$

$$+log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta)$$

$$=\frac{1}{\delta}\|x-m\|^2-\sum_{i=0}^{r-1}(\frac{1}{\delta}-\frac{1}{\lambda_i})\{(x-m,\varphi_i)^2\}$$

$$+log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta)$$

$$=\frac{1}{\delta}\{\|x-m\|^2-\sum_{i=0}^{r-1}(1-\frac{\delta}{\lambda_i})\{(x-m,\varphi_i)^2\}\}$$

$$+log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta), \qquad (34)$$

The term $1/\delta$ at the first part is common for any category. Therefore, it becomes possible to multiply δ by all the terms in the equation, and the final distance is obtained as follows:

$$D=\|x-m\|^2-\sum_{i=0}^{r-1}(1-\frac{\delta}{\lambda_i})(x-m,\varphi_i)^2$$

$$+\delta\left\{log(\prod_{i=0}^{r-1}\lambda_i)+log(\prod_{i=r}^{N-1}\delta)\right\}. \qquad (35)$$

This distance of MQDF has been proposed by the group from Mie and Nagoya universities in the literature [12] with the method Weighted Direction Index Histogram.

In that literature, it was introduced as Mahalanobis style and was called Quasi-Mahalanobis Distance (Q-MD), and the Bayesian style method was called Quasi-Bayes Distance (Q-BD). The name MQDF used in the paper [13] is basically the same as Q-BD. In this connection, the above mentioned feature extraction method is the same as the method called Weighted Direction Code Histogram (WDCH) and it is also effective and useful for character recognition.

X. Projection Distance

We have another method similar to the Subspace Method. Let a mean vector be m, and it is defined as follows:

$$D=\|x-m\|^2-\sum_{i=0}^{r-1}(x-m,\varphi_i)^2. \qquad (36)$$

In this equation, there is no restriction for norm of vectors. This is the difference to the Subspace Method. In this method, the distance of x and m is measured in the complement space of the subspace which is spanned by the reference vectors φ_i . PCA is often used in this case to obtain the reference vectors φ_i .

XI. Inner Product Type and Distance Type

TH e Weghted Subspace Method corresponding to MQDF is shown here [14]. It is called Spherical MQDF (S-MQDF). This method is obtained by a hypothesis that patterns are distributed on a spherical surface and the same process by which MQDF is introduced. The equation has the same structure as MQDF except for some terms.

The following equation shows the spherical version of Q-MD, that is S-Q-MD. This is a simpler version of S-MQDF.

$$S=2\delta\ln(x,\varphi_0)+(x,\varphi_0)^2+\sum_{i=1}^{r-1}(1-\frac{\delta}{\lambda_i})(x,\varphi_i)^2. \qquad (37)$$

For the comparison of this to Q-MD, the equation of Q-MD is shown as follows:

$$D=\|x-m\|^2-\sum_{i=0}^{r-1}(1-\frac{\delta}{\lambda_i})(x-m,\varphi_i)^2, \qquad (38)$$

We can see the correspondence between above two equations. The vector φ_0 in (37) corresponds to the mean vector m in (38).

As for the other correspondence, Q-BD (= MQDF) corresponds to S-Q-BD, Projection Method to the Subspace Method and Euclidean Distance to Simple Similarity.

Table I shows relationship between these methods. The vertical group of Q-BD in this table is based on Euclidean Distance and uses distance for discrimination. In contrast to that, the group of S-Q-BD is based on inner product and uses similarity. It is said that the former is distance type and the later is inner product type if we see the calculation process.

TABLE I
DISTANCE TYPE AND INNER PRODUCT TYPE.

distance	inner product
Q-BD(MQDF)	S-Q-BD
Q-MD	S-Q-MD
Projection Distance	Subspace Method
Euclidean Distance	Simple Similarity

As for the characteristics of recognition accuracy, the methods shown horizontally in table I are similar while the characteristics of the methods shown vertically are different from each other. From the experimental result [14], the performance is almost the same if the weight parameter is the same. In conclusion, the importance is in the weight parameters but not in the difference of calculation method.

XII. Incremental Learning

HERE, we start form Learning Vector Quantization (LVQ), and see the learning method for the Subspace Method. The framework of LVQ [15] proposed by Teuvo Kohonen is shown below.

Let x denote an input pattern, let m denote a reference vector, let D denote distance.

$$D=||x-m||^2, \qquad (39)$$

is the distance used in LVQ. The learning process is as follows:

$$m = m \pm \alpha(x - m), \quad (+: correct, \; -: error). \tag{40}$$

Here the vector m is changed by the above equation as learning process. This method can be also introduced by the learning method of Probabilistic Descent (PD) [16][17]. This LVQ is one of the basic techniques in the pattern recognition field and often recognized as one of the Neural Network systems. On the other hand, PD, an incremental learning method, is a powerful technique applicable to many pattern recognition systems. It is important to understand LVQ with PD concept. At the same time, it is simpler to use PD to introduce the incremental learning method for the Subspace Method.

Firstly, let's apply PD to Simple Similarity.

$$S = (x, \varphi)^2 / \|x\|^2 \|\varphi\|^2, \tag{41}$$

is the definition of Simple Similarity and an recurrence formula for φ is shown below:

$$\varphi = \varphi \pm \alpha(x, \varphi)\{x - (x, \varphi)\varphi\}. \tag{42}$$

This formula is rather complicated. Let's look at a simpler one where the normalization term is ignored.

$$S = (x, \varphi)^2, \tag{43}$$

is the similarity and the recurrence formula is,

$$\varphi = \varphi \pm \alpha(x, \varphi)x. \tag{44}$$

It is a precondition that the vectors x, φ are normalized in this equation. Consequently, (44) obtained by the process without normalization can be said to be incorrect. Therefore, it is necessary to introduce some operation of normalization. The most simple way of doing this is to normalize φ each time the recurrence formula is used. It is possible to adopt this approach instead of (42).

Next, the case of the Subspace Method is shown.

The similarity is shown below:

$$S = \sum_{i=0}^{r-1} (x, \varphi_i)^2. \tag{45}$$

The recurrence formula for φ_i is the same as (44). But, the orthogonality and normality of the vectors φ_i are not preserved in this case. The orthogonality and normality are required when we adopt PD to (45). It is possible, but the calculation is complex and there is a method like the one adopted in Simple Similarity, i.e. a method in which the orthogonalization and normalization are executed for the vectors φ_i at the time the recurrence formula is used. An incremental learning method for the Subspace Method with PD and Gram-Schmidt orthogonalization is the same as LSM [18] which was proposed by Kohonen.

There is a similar method ALSM[19][20] in which the covariance matrix K in (15) is changed by the following equation:

$$K = K \pm \alpha x x^T, \tag{46}$$

This method is the same as the method called the Learning Multiple Similarity Method, [21][22] which was proposed by Ken-ichi Maeda in 1980.

There is another incremental learning technology for the Subspace Method, that is, DMD[23]. In the method LSM, the recurrence formula is obtained from evaluation function without consideration of othogonalization and normalization if PD is assumed to be a basic concept. In contrast to that, the orthogonalization and normalization is included naturally in iterative process in DMD. Rotation parameters in the projection matrix is changed in this method.

There is more research [24] for applying PD to methods relevant to the Subspace Method.

XIII. How to Fix a Subspace

ONE of important research targets in the Subspace Method is how to fix a subspace.

A. Independent Component Analysis

In the recent research results, Independent Component Analysis (ICA) is one of the interesting methods as a substitute for PCA. Orthogonalization is not required in ICA and reference vectors are extracted as a vector representing direction where a pattern distribution has high density.

B. Feature Selection

The method of feature selection [3][8] is for selecting important features from all features obtained from an input pattern. It has been a major item for investigation since Iijima's theory and Watanabe's SELFIC were introduced and is a research target at the present time.

XIV. Similarity

HOW to organize the similarity is a core factor if the Subspace Method is used in pattern recognition.

A. The Mutual Subspace Method

The Mutual Subspace Method was created as an expansion of the Subspace Method. It was proposed by K. Maeda [25] in 1985. The similarity is defined as an angle between a subspace and a subspace where it is an angle between a vector and a subspace in the Subspace Method. This method is especially effective for face recognition when faces are captured as image sequence such as video [26] which includes many stills with many variations. It is an effective approach to treat an input pattern as a subspace which is made of an input image sequence. This has been proved to be good for face recognition and it is expected to be applicable in other areas.

B. Kernel Trick

The technique "kernel trick" is a noticeable technique as a key factor in SVM. The idea is that recognition accuracy is expected better if feature space has high dimension and is sparse space. For this purpose, a conversion from original low dimensional feature space to high, often infinitive, feature space is used in SVM. This is called a kernel trick. This concept is the reverse of the feature selection concept in which the feature space's dimension is decreased by a conversion. Kernel trick might be applicable to the Subspace Method, and it has been already tried in the pattern recognition community.

C. Compound Similarity Like Methods

The details of Compound Similarity [5] are not described in this report, but it's concept can be explained simply as follows:

In this method, a similar pattern pair is considered, one is denoted as A and the other is B. For the subspace of category A, a subspace which represents B is introduced as S'_B in addition to an ordinary subspace S_A for A. For an input pattern which belongs to category A, projection on S_A is treated as positive, while projection on S'_B is treated as negative. This idea is useful to improve accuracy for similar pattern discrimination. It is possible to construct various recognition systems by using this concept.

XV. New Targets and New Applications

THE typical application of the Subspace Method is character recognition and it has been extended it's capability to the area of speech recognition and image recognition. It is expected that it's applications will spread to other fields. Even in the field of character, speech and image recognition fields, it might be possible to increase its use. Two areas are described as examples of the recent applications of the Subspace Method.

A. Eigen Face

Eigen face [27] is the eigen vector of a covariance matrix made of face images and it is used in a face recognition system as a reference vector. This is one of the Subspace Method applications. It was the typical approach to extract some parts from a face image and analyze their relative positions and their figure for recognition. It might be a breakthrough to use the Subspace Method because it becomes easy to generate a reference which is constructed by human operation in ordinary concept. This application was astonishing to researchers who only applied the Subspace Method to character recognition. It was felt that a face was too complex to recognize with the Subspace Method.

B. Parametric Eigenspace Method

An image in the sequence of images is considered as a vector in a linear space in Parametric Eigenspace Method [28] . Consequently, the image sequence is treated as a vector sequence in the space. Tracking or posture analysis is achieved by this method. As for an ordinary concept, tracking object is achieved by tracking feature points, portions or their figure which are extracted from input images. This method could be a breakthrough because it automatically constructs a system for object tracking or posture analysis.

XVI. Conclusion

TH e capabilities of the Subspace Method are still not high enough at the present time, some 40 years after the birth of the Subspace Method. But new ideas are still being applied to many fields of pattern recognition with modification of the Subspace Method.

Euclidean Distance, inner product, angle between vectors, Gaussian distribution and other important key factors are all necessary for pattern recognition and the concept of subspace is also necessary in this field. In comparison with Euclidean Distance, inner product, and other simple items, a subspace is rather complex. Consequently, there are many themes to investigate in the concept of subspace. The subspace concept and the Subspace Method will continue to play important roles in the pattern recognition field.

References

[1] Y. Kurosawa, “The Engineer’s Guide to the Subspace Method,” *Subspace2006: satellite workshop of MIRU2006,* pp.136-143, 2006 (in Japanese).

[2] E. Oja, *Subspace Method of Pattern Recognition,* New York: J. Wiley, 1983.

[3] T. Iijima, “Basic Theory on Feature Extraction for Visual Pattern,” *Trans. IECE Japan,* vol.46. no. 11, pp.1714-1721, 1963 (in Japanese).

[4] T. Iijima, H. Genchi and K. Mori, “A Theory of Character Recognition by Pattern Matching Method,” *Proc. 1st. IJCPR,* pp.50-56, 1973.

[5] T. Iijima, ”A Theory of Pattern Recognition by Compound Similarity Method,” *IEICE Technical Report,* PRL74-24, 1974 (in Japanese).

[6] J. Weickert, S. Ishikawa and A. Imiya, ”On the History of Gaussian Scale Space Axiomatics,” *Gaussian Scale-space Theory*, J. Sporring et. al., ed., Netherlands: Kluwer Academic Publishers, pp.45-59, 1997.

[7] S. Watanabe, *Knowing and Guessing,* New York: J. Wiley, 1969.

[8] S. Watanabe, P. F. Lambert, C. A. Kulikowski, J. L. Buxton and R. Walker, “Evaluation and Selection of Variables in Pattern Recognition,” *Computer and Information Sciences,* vol.2, J. Tou, ed., New York: Academic Press, pp.91-122, 1967.

[9] C. A. Kulikowski, “Pattern Recognition Approach to Medical Diagnosis,” *IEEE Trans. Systems Science and Cybernetics,* vol. SSC-6, no.3, pp.173-178, 1970.

[10] S. Watanabe and N. Pakvasa, “Subspace Method in Pattern Recognition,” *Proc. 1st. IJCPR,* pp.25-32, 1973.

[11] S. Watanabe, “Karhunen-Loeve Expansion and Factor Analysis,” *Trans. 4th Prague Conf. on Information Theory, Statistical Decision Functions, Random Processes,* Prague: Publishing House of the Czechoslovak Academy of Sciences, pp.635-660, 1967.

[12] M. Kurita, S. Tsuruoka, S. Yokoi and Y. Miyake, ”Handprinted “KANJI” and “HIRAGANA” Character Recognition Using Weighted Direction Index Histogram and Quasi-Mahalanobis Distance,” *IEICE Technical Report,* PRL82-79, 1983 (in Japanese).

[13] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake, “Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition,” *IEEE Trans. PAMI,* Vol.9, No.1, pp.149-153, 1987.

[14] Y. Kurosawa, “Subspace Method Obtained from Gaussian Distribution on a Hyper Spherical Surface,” *IEICE Trans. Inf. & Syst.,* J81-D-II, No.6, pp.1205-1212, 1998 (in Japanese).

[15] T. Kohonen, “Learning Vector Quantization,” *Helsinki University of Technology, Laboratory of Computer and Information Science,* Report TKK-F-A601, 1986.

[16] S. Amari, “A Theory of Adaptive Pattern Classifiers,” *IEEE Trans. EC,* Vol.16, No.3, pp.299-307, 1967.

[17] S. Katagiri, C.-H. Lee and B.-H. Juang, “A Generalized Probabilistic Descent Method,” *ASJ, Fall Conf.,* 2-p-6, pp.141-142, Nagoya, Japan, 1990.

[18] T. Kohonen, G. Nemeth, K.-J. Bry, M. Jalanko and H. Riittinen, “Classification of Phonemes by Learning Subspaces,” *Helsinki University of Technology, Dept. of Technical Physics,* Report TKK-F-A348, 1978.

[19] M. Kuusela and E. Oja, “The Averaged Learning Subspace Method for Spectral Pattern Recognition,” *Proc. 6th. IJCPR,* pp.134-137, 1982.

[20] E. Oja and M. Kuusela, “The ALSM Algorithm - an Improved Subspace Method of Classification,” *Pattern Recognition,* Vol.16, No.4, pp.421-427, 1983.

[21] K. Maeda, “Pattern Recognition Apparatus,” *Japanese Patent Public Disclosure,* 137483/81, 1980 (in Japanese).

[22] K. Maeda, “Dimension Selection by Learning for Class Discrimination and Information Representation” *AIAI Technical Reports* AIAI-TR-75, 1990.

[23] H. Watanabe, T. Yamaguchi and S. Katagiri, “Discriminative Metric Design for Pattern Recognition,” *ICASSP-95,* Vol.5, pp.3439-3442, 1995.

[24] Y. Kurosawa, ”Probabilistic Descent Method Applied to Similarity and Distance Measure of Quadratic Form for Pattern Recognition,” *IEICE Technical Report,* PRMU97-181, 1997 (in Japanese).

[25] K. Maeda and S. Watanabe, “A Pattern Matching Method with Local Structure,” *IEICE Trans. Inf. & Syst.,* J68-D, No.3, pp.345-352 1984 (in Japanese).

[26] O. Yamaguchi, K. Fukui and K. Maeda, “Face Recognition System using Temporal Image Sequence,” *IEICE Technical Report,* PRMU97-50, 1997 (in Japanese).

[27] M. A. Turk and A. P. Pentland, “Face Recognition Using Eigenface,” *Proc. of Computer Vision and Pattern Recognition,* pp.586-591, 1991.

[28] H. Murase and S. K. Nayar, “Visual Learning and Recognition of 3-D Objects from Appearance,” *International Journal of Computer Vision,* Vol.14, pp.5-24, 1995.

An Adaptive Shape Subspace Model for Level Set-Based Object Tracking

Xue Zhou, Weiming Hu, and Xi Li
National Laboratory of Pattern Recognition, Institute of Automation, Beijing, China
{xzhou, wmhu, lixi}@nlpr.ia.ac.cn

Abstract

Shape priors have been widely used for level set-based tracking to solve some difficult problems, such as noisy data, partial occlusions and weak contrast at the boundaries. In this paper, we propose a two-layer hierarchical level set-based tracking framework in which color and shape information are fused sequentially. In the first layer, the initial contour is evolved only with the color feature, then the Mahalanobis distance-based discriminant criterion is adopted to determine whether the shape model is needed. If the shape model is needed, in the second layer the contour is evolved with the shape constraint continuously. For the second layer, a weighted shape distance term (WSDT) is introduced into the pixel-wise contour evolution equation to fuse the global shape information and the local color information. Principal Component Analysis (PCA) subspace of shape samples is trained off-line and updated using an on-line algorithm. The experimental results on several real video sequences demonstrate the robustness and the effectiveness of our method.

1. Introduction

Level set-based methods are very popular in dealing with many computer vision tasks such as object detection, image segmentation and tracking [6] [7][15]. In the case of noise or partial occlusions, low-level features (color, texture, etc.) are inadequate to perform the above tasks. Thus, some prior knowledge about the shape of objects is necessary to be integrated into the level set evolution framework.

There are lots of work on modeling statistical shape priors in the literature [2] [11] [17]. Leventon et al. [2] construct a Gaussian distribution in the low-dimensional shape subspace. A set of aligned training shape samples represented by the signed distance maps are projected into an orthogonal subspace by Principal Component Analysis (PCA). Paragios and Rousson [11] construct a pixel-wise shape model that accounts for local variabilities. Each grid location is described in the shape model using a Gaussian density function. Unfortunately, the above two stable shape models couldn't be updated on-line, resulting in they are not adaptation to new shapes.

Active Shape Models (ASMs) [9] are imperative for tracking objects with continuous and large shape changes. The partially learned ASMs (only finite samples are considered in the training step) are used to conduct tracking and the new tracking results are returned to update the ASMs [9]. In [3], Cremers proposes a dynamical statistical shape prior model which combines the concepts of Markov chains and autogressive models. The probability of observing a particular shape at a given time instance depends on the shapes observed at previous time instances. Fussenegger et al. [9] create an on-line active shape model. The training samples are embedded into an orthogonal subspace and an incremental PCA (IPCA) algorithm [14] is used to update the active shape model. However, the above IPCA algorithm only handles one new sample per update. A robust extended R-SVD algorithm [10] is widely used for appearance-based tracking recently. This algorithm not only handles multiple samples at the same time, but also computes the eigenbasis with mean update.

Generally, researchers [11] [13] introduce the shape priors into level set-based framework through construction of a shape difference term on the variational level. In our method, we also incorporate a level set-based shape difference term into the variational model. However, our method is different from current methods in several aspects. Firstly, an exponential term weighting the shape difference term (named WSDT) is proposed to balance the initial shape and the target shape. Secondly, the initial shape corresponds to the one obtained only with the color information, the target shape is the one inverse-transformed to the image after being evolved in the shape subspace. Thirdly, WSDT not only captures the whole shape information learned from the shape subspace model, also regards local color variabilities.

Our method has the following characteristics:

- We propose a two-layer level set-based framework for combing color feature and shape priors hierarchically. At first, the initial contour is evolved only with the color information. And then, the Mahalanobis distance-based discriminant criterion is proposed to determine whether the shape model is integrated into the whole tracking framework or not. If the shape model is needed, sequentially, the contour obtained from the first layer is evolved with the shape constraint.

- The shape subspace-based evolution and the color space-based evolution are well fused by a weight term in the pixel-wise contour evolution equation. With this weight term, both the global shape information and the local color information are considered.
- The shape model is constructed in an orthogonal subspace. A robust incremental learning algorithm is adopted to update the shape model. The shape model is updated every few frames to reflect shape changes of the object.

The remainder of this paper is organized as follows: Section 2 gives an overview of our method. Section 3 describes the off-line training process of the shape prior model. The on-line tracking process which comprises the color-based contour evolution and the shape priors-based contour evolution is introduced in Section 4. Section 5 presents the incremental learning algorithm for shape subspace model. Experimental results are given in Section 6. The last section concludes the paper.

2 Overview of our method

The proposed method contains two phases: the off-line training stage and the on-line tracking stage. The goal of the off-line training stage is to train a shape subspace model using a set of aligned samples. The shape subspace are obtained through the Singular Value Decomposition (SVD). At the on-line tracking stage, for each frame initial contours are evolved only with the color information in the first layer, after being aligned to the mean shape of the shape model, the Mahalanobis distance-based criterion is adopted to determine whether the shape model is added or not. If the shape model is needed, in the second layer the contours are evolved with the shape constraint continuously, otherwise, the tracking results obtained only with the color feature are considered as the final tracking results. After the final tracking results are obtained, they are returned to update the ASMs using an incremental learning algorithm. Obtained contours at time t are the initial contours for the time $t + 1$. Figure 1 shows the flowchart of our framework.

3 Off-line training

The off-line training stage consists of two steps: (1) shape registration; (2) computation of the eigenspace of the training samples.

3.1 Shape registration

Contours are represented by the level set method [8]. The level set function chosen in our method is the commonly used signed distance function. Zero value of this function corresponds to a contour. The shape information of the object is also embedded in the signed distance map represented by Φ:

$$\Phi(x,y) = \begin{cases} 0 & (x,y) \in C \\ d(x,y,C) & (x,y) \in R_{out} \\ -d(x,y,C) & (x,y) \in R_{in} \end{cases} \tag{1}$$

where R_{in} and R_{out} denote respectively the regions inside and outside the contour C and $d(x,y,C)$ is the smallest Euclidean distance from point (x,y) to the contour C:

$$d(x,y,C) = \min_{x_c,y_c \in C} \sqrt{(x-x_c)^2 + (y-y_c)^2} \tag{2}$$

Shape registration is implemented using the Paragios's variational method [12]. An objective function is constructed to find a global optimal transformation A which minimizes the sum of squared differences between the current shape D and the target shape S (randomly chosen from the training samples). The function is defined as:

$$E(s,\theta,T) = \iint_{\Omega} (s\Phi_D(x,y) - \Phi_S(A(x,y)))^2 dxdy \tag{3}$$

where A includes three parameters: a scale factor s, a rotation angle θ and a translation vector $\mathbf{T} = (\mathbf{T}_x, \mathbf{T}_y)$. Φ_S and Φ_D are the signed distance maps for the current shape D and the target shape S respectively. Three iteration equations about $(s,\theta,\mathbf{T})$ are obtained using a gradient descent method:

$$\begin{cases} \frac{\partial s}{\partial t} &= 2\iint_{\Omega}(-\Phi_D + \nabla\Phi_S \cdot \nabla_s A)(s\Phi_D - \Phi_S(A))dxdy \\ \frac{\partial \theta}{\partial t} &= 2\iint_{\Omega}(\nabla\Phi_S \cdot \nabla_\theta A)(s\Phi_D - \Phi_S(A))dxdy \\ \frac{\partial \mathbf{T}}{\partial t} &= 2\iint_{\Omega}(\nabla\Phi_S \cdot \nabla_{\mathbf{T}} A)(s\Phi_D - \Phi_S(A))dxdy \end{cases} \tag{4}$$

3.2 Computation of the eigenspace of the training samples

After obtaining the optimal transformation parameters $(s,\theta,\mathbf{T})$, the training samples are transformed to be aligned with the target shape. Before the SVD is calculated, each signed distance map is flattened into a column vector. The mean vector μ is computed by taking the mean of these column vectors. Construct an $M \times N$-dimensional data matrix X whose column is a sample vector subtracted from the mean vector μ. M is the length of each sample vector and N is the number of the training samples. Compute the SVD of X:

$$X = U\Sigma D^T \tag{5}$$

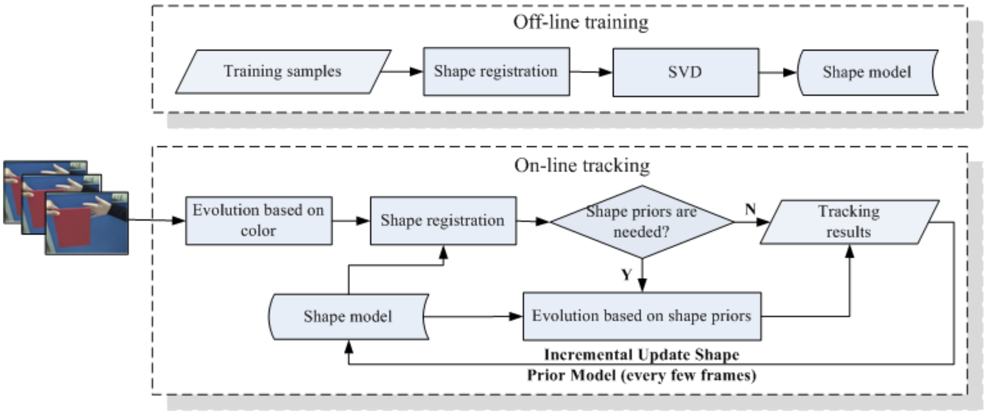


Figure 1. Flowchart of our method.

where U is an $M \times N$-dimensional matrix whose column vectors are the eigenvectors of the shape subspace and Σ is an $N \times N$-dimensional diagonal matrix of the corresponding singular values.

In our shape model, we choose the first k ($k \leq N$) columns of U represented by U_k as the eigenbasis and Σ_k is the diagonal matrix composed of the corresponding first k singular values. Thus, our shape model $\{\mu, U_k, \Sigma_k\}$ is constructed through the above procedure.

4 On-line tracking

The on-line tracking stage hierarchically consists of three steps: contour evolution based on the color feature (the first layer), the Mahalanobis distance-based discriminant criterion and contour evolution with the shape constraint (the second layer).

4.1 Contour evolution with the color feature

The method we adopt in this step is a region-based active contours method, modeling the features of both object and background regions in the level set speed model. In our method, we train a color Gaussian Mixture Model (GMM) in object and background regions respectively. The HSV color space is chosen in this model. The estimated probability density function (pdf) at pixel x_i in the color space can be formulated as:

$$p(\mathbf{x}_i|\mu, \sigma) = \sum_{j=1}^{k} \omega_j \eta(\mathbf{x}_i^c, \mu_j^c, \Sigma_j^c) \quad (6)$$

where x_i^c is the color feature at pixel x_i, ω is the weight parameter of the GMM model, k is the number of the Gaussian modes and η is a Gaussian pdf:

$$\eta(\mathbf{x}^c, \mu_j^c, \Sigma_j^c) \propto \frac{1}{|\Sigma_j^c|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{x}^c - \mu_j^c)^T (\Sigma_j^c)^{-1} (\mathbf{x}^c - \mu_j^c)\} \quad (7)$$

After the color GMM model is constructed, the data energy function is formulated based on the segmentation idea similar to [4] [5]. The key factor in this process is to find the optimal partition operator represented by a contour between the object region and the background region. The data energy function E_{data} is defined as follows:

$$E_{data} \approx -\iint_{x_i \in R_{in}} \log P(x_i|\theta_{in}) d\mathbf{x}_i - \iint_{x_j \in R_{out}} \log P(x_j|\theta_{out}) d\mathbf{x}_j \quad (8)$$

where the definitions of R_{in} and R_{out} are same as those in (1), θ_{in} is the parameters of the object GMM model $\{\omega_{in}, \mu_{in}^c, \Sigma_{in}^c\}$ and $P(x_i|\theta_{in})$ is the object likelihood function which is computed using (6). $P(x_i|\theta_{out})$ and θ_{out} are defined by analogy.

Minimizing the above energy function by solving the correlated Euler-Lagrange equations [4], we obtain the level sets advection speed model in which a $(2l + 1) \times (2l + 1)$ square neighboring subregion around the center pixel is defined. The object and the background posterior probabilities which we denote by $P_{R_{in}}(I_{\tilde{x}})$ and $P_{R_{out}}(I_{\tilde{x}})$ are also calculated in the speed model with the assumption that they have the same prior probabilities:

$$P_{R_{in}}(I_{\tilde{x}}) = P(\tilde{x}|\theta_{in})/[P(\tilde{x}|\theta_{in}) + P(\tilde{x}|\theta_{out})] \quad (9)$$

$$P_{R_{out}}(I_{\tilde{x}}) = P(\tilde{x}|\theta_{out})/[P(\tilde{x}|\theta_{in}) + P(\tilde{x}|\theta_{out})] \quad (10)$$

The level sets advection speed model of each pixel (x, y) is obtained by:

$$\begin{aligned} F_{x,y} = & -\sum_{i=-l}^{l} \sum_{j=-l}^{l} \log P_{R_{in}}(I_{\tilde{x}}) H_a(\Phi(\tilde{x}, t)) \quad (11) \\ & + \sum_{i=-l}^{l} \sum_{j=-l}^{l} \log P_{R_{out}}(I_{\tilde{x}})(1 - H_a(\Phi(\tilde{x}, t))) \end{aligned}$$

where $\tilde{x}$ is the neighboring pixels of center pixel (x, y): $\tilde{x} = (x + i, y + j)$ and $H_a(\Phi(\tilde{x}, t))$ is a Heaviside function:

$$H_a(\Phi(\tilde{x}, t)) = \begin{cases} 0 & \Phi(\tilde{x}, t) \geq 0 \\ 1 & \Phi(\tilde{x}, t) < 0 \end{cases} \tag{12}$$

The contour is evolved to the desired boundary by modifying Φ iteratively with the overall speed F in the normal direction:

$$\frac{\partial \Phi}{\partial t} + (F_{x,y} + F_{curv})|\nabla \Phi| = 0 \tag{13}$$

where $F_{x,y}$ is the external force reflecting the data attachment and $F_{curv} = -\varepsilon\kappa(x, y)$ is the internal force proportional to the curvature $\kappa(x, y)$ of the contour, the term F_{curv} has a smoothness effect on the contour. The detailed stable numerical approximation scheme of the above equation is given in [1]. The evolution result obtained in this step is denote by Φ_c.

4.2 Mahalanobis distance-based discriminant criterion

After the contour evolution result based on the color feature Φ_c is obtained, it's aligned with the target shape by shape registration which has been described in Subsection 3.1. The current shape D and the target shape S in this step correspond to the obtained shape Φ_c and the mean shape of the shape model respectively.

Based on the optimal registration parameters, the result obtained in the first layer Φ_c is affinely transformed into a normalized vector represented by x and then projected into the subspace. Eventually, Φ_c is represented by a k-dimensional vector α:

$$\alpha = U_k^T (x - \mu) \tag{14}$$

where U_k and μ are parameters of the trained shape subspace model. The Mahalanobis distance between x and the mean shape μ is formulated as follows:

$$\gamma^2 = (x - \mu)^T C^{-1} (x - \mu) \tag{15}$$

where C is the covariance matrix $C \approx U_k \Sigma_k^2 U_k^T$, substitution of this formula into (15), we can obtain:

$$\gamma^2 \approx (x - \mu)^T U_k \Sigma_k^{-2} U_k^T (x - \mu) = \alpha^T \Sigma_k^{-2} \alpha \tag{16}$$

If γ^2 is bigger than the predefined threshold T, the shape subspace model is integrated into the whole tracking framework and the result Φ_c obtained in the first layer is evolved continuously. Otherwise Φ_c is considered as the final tracking result and is returned to update the shape model directly.

4.3 Contour evolution with the shape constraint

In the second layer, a weighted shape difference term (WSDT) is proposed to evolve contour (obtained in the first layer) with the shape constraint. With this term, both the global shape information and the local color information are considered.

Firstly, the obtained contour Φ_c is evolved in the shape subspace. We construct the global shape energy function E_{Gshape} based on the Mahalanobis distance described in (16):

$$E_{Gshape} = \alpha^T \Sigma_k^{-2} \alpha \tag{17}$$

where α is the k-dimensional vector representing a contour in the shape subspace. The iteration function about α is obtained using gradient descent method:

$$\frac{\partial \alpha}{\partial t} = -\frac{\partial E_{Gshape}}{\partial \alpha} = -2\Sigma_k^{-2} \alpha \tag{18}$$

Let the initial value of α represented by $\alpha_{initial}$ be Φ_c. Through finite iteration steps the final α_{final} is obtained.

Secondly, an estimate of the evolved shape $\tilde{x}$ is reconstructed from U_k and μ based on the obtained α_{final}:

$$\tilde{x} = U_k \alpha_{final} + \mu \tag{19}$$

The reconstructed shape $\tilde{x}$ is inverse-transformed into the image plane based on the transformation parameters obtained in the shape registration step. The signed distance map containing the inverse-transformed shape is recomputed and we denote it by Φ_0.

Thirdly, current methods usually model the pixel-wise contour evolution equation with shape constraint by constructing a shape distance term which has the following format:

$$\frac{\partial \Phi}{\partial t} = -2(\Phi - \Phi_m) \tag{20}$$

where Φ_m is the level set function of the given training shape or the mean of a set of training shapes. In our method, we replace Φ_m with Φ_0 and a weighted shape distance term (WSDT) is proposed into the pixel-wise contour evolution equation to balance the result evolved only with the color feature (denoted by Φ_c) and the result evolved in the shape subspace (denoted by Φ_0):

$$\frac{\partial \Phi}{\partial t} = -2(\Phi - \Phi_0)[1 - e^{-(\frac{\Phi - \Phi_0}{\sigma})^2}] \tag{21}$$

where σ is the parameter controlling how fast the exponent function converges to zero. In this pixel-wise evolution equation, the initial value of Φ is Φ_c obtained in the first layer. The item in the square bracket is the weight item which is a function of the distance between Φ and Φ_0 (denoted by $d(\Phi, \Phi_0)$). This weight term has the following characteristics:

1. Within the evolution process, if $d(\Phi, \Phi_0)$ is small in a pixel location, the weight item is close to zero. Thus, the value of Φ doesn't change much, i.e. it approximates to the initial value Φ_c. This means the result based on the color feature Φ_c is more credible compared with the result obtained with the shape constraint Φ_0 in this pixel location.

2. Within the evolution process, if $d(\Phi, \Phi_0)$ is large in a pixel location, the weight item is close to one. Thus, the evolution process for Φ is going on till its value approximates to the value Φ_0. This means in this pixel location Φ_c is trustless due to the background noise or occlusions and the result with the shape constraint Φ_0 is more credible.

After several iteration steps, $d(\Phi, \Phi_0)$ is becoming smaller and smaller. Due to the effect of the weight term, Φ converges for all pixels. However, for some pixel locations where $d(\Phi, \Phi_0)$ are large at the beginning, Φ converges too soon, resulting in an inaccurate result. To solve the above problem, we adjust the parameter σ to control the evolution speed at different pixel locations. σ is determined by the distance between Φ_c and Φ_0 (denoted by $d(\Phi_c, \Phi_0)$):

$$\sigma = \beta e^{-(\Phi_c - \Phi_0)^2} \tag{22}$$

where β is a positive constant. Thus, at pixel locations where $d(\Phi_c, \Phi_0)$ are large, the evolution of Φ is still going on even though $d(\Phi, \Phi_0)$ are getting smaller and smaller. At pixel locations where $d(\Phi_c, \Phi_0)$ are small Φ stops evolution soon. Consequently, the result obtained with the color feature (represented by Φ_c) and the result evolved in the shape subspace (denoted by Φ_0) are combined well.

5 Incremental learning algorithm

After the final evolution results are obtained, they are transformed into the normalized format and returned to update the shape model. In this step, we adopt an efficient and effective online algorithm [10] which has been widely used for appearance-based tracking. This algorithm is superior to other current subspace updating algorithms [14] [16]. Firstly, it correctly updates the sample mean and the eigenbasis. Secondly, it can handle blocks of data rather than a single data. Through this algorithm, the shape changes of the tracking target are incrementally learned.

For our method, the shape model is updated every few frames. Assuming that the previous shape model $\{\mu_{i-1}, U_{i-1}, \Sigma_{i-1}\}$ and new data X are given , at stage i the new shape model $\{\mu_i, U_i, \Sigma_i\}$ is incrementally updated as shown in Algorithm 1.

Algorithm 1 Incremental Learning Algorithm for subspace

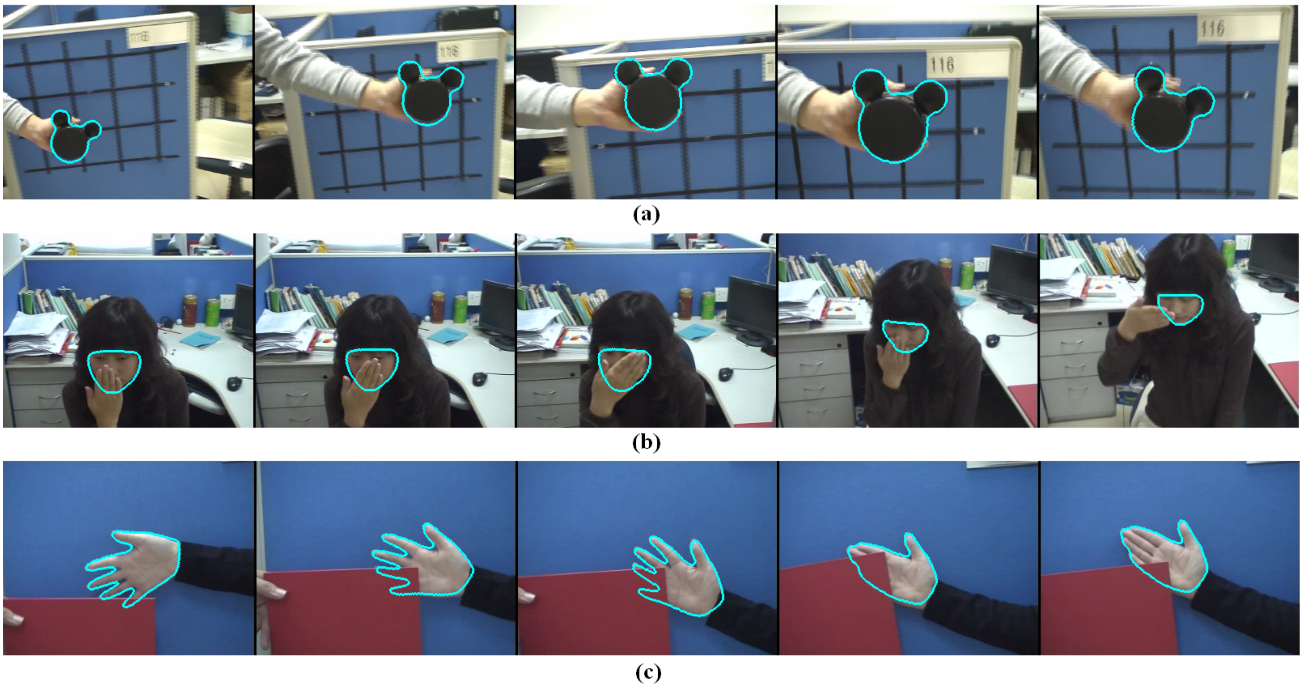
```
Input :
old mean vector $\mu_{i-1}$, old basis $U_{i-1}$, old singular values $\Sigma_{i-1}$, new data $X$ with mean $\mu_{new}$, the number of the previous data $n_0$, the number of the new data $n_{new}$, “forgetting factor” coefficient $ff$, the maximum number of columns for the $U_i$ matrix $k$
Output :
new mean vector $\mu_i$, new basis $U_i$, new singular values $\Sigma_i$
ith stage :
1. Update the mean vector $\mu_i = \frac{ff \cdot n_0}{(n_{new} + ff \cdot n_0)} \mu_{i-1} + \frac{n_{new}}{(n_{new} + ff \cdot n_0)} \mu_{new}$
2. Let $X$ be zero mean $X = X - \mu_{new}$
3. Construct the combined matrix
        $X' = (ff \cdot U_{i-1} \Sigma_{i-1} \mid X \mid \sqrt{\frac{n_0 n_{new}}{n_0 + n_{new}}} (\mu_{i-1} - \mu_{new}))$
4. Compute the QR decomposition of the combined matrix
        $X' = QR$
5. Compute the SVD of matrix $R$
        $R = U \Sigma D^T$
6. Compute the final eigenvectors and singular values
        $U'_i = QU, \ \Sigma'_i = \Sigma \cdot \sqrt{n_0 / (n_{new} + ff \cdot n_0)}$
7. Let $\Sigma_i$ be the diagonal matrix whose elements are the $k$ largest singular values of $\Sigma'_i$ and $U_i$ be the final eigenbasis matrix whose first $k$ columns are chosen from $U'_i$
```

6 Experiments

To verify our method, we have performed several experiments on various real sequences.

In our experiments, all the videos are captured with a moving camera, a tracked object is represented with a grayed contour (colored in color image). Some difficult cases occur in our experiments, such as background disturbance (similar color between object and background), motion blur and partial occlusions. At the shape model offline training stage, good training samples without missing data (obtained before difficult cases happen) are used to construct the eigenspace of shapes. The sample is a 3000-dimensional vector. The first 20 eigenvectors are chosen as the eigenbasis. l in (11) is independent of sequences and is fixed to 2. The threshold T for the Mahalanobis distance-based discriminant criterion is set to be 16. At the on-line updating stage, tracking results are returned every three frames. The ”forgetting factor” coefficient ff in Algorithm (1) is set to be 0.95.

In the first experiment, we track a Mickey head with a moving camera from Frame 1 to Frame 155. In this sequence, there are some disturbances around the tracked object. The color of the scripts in background is the same as the color of the Mickey head. The camera zooms and tilts as the object moves. The parameter β in (22) is set to be 100 for this sequence. As shown in Figure 2.(a), although with the background disturbance, we still track the contour of the Mickey head accurately. In the 151^{th} frame, there exists severe motion blur. The boundary contrast between object and


(a)

(b)

(c)

Figure 2. Tracking results for several real sequences: (a) Tracking results for a moving Mickey head sequence. The frame numbers are, respectively, 1, 49, 111, 139 and 151; (b) Tracking results for a moving face sequence. The frame numbers are, respectively, 50, 59, 65, 107 and 118; (c) Tracking results for a moving hand sequence. The frame numbers are, respectively, 25, 38, 45, 110 and 117.

background is weak. But the object is still tracked robustly.

In the second experiment, we demonstrate the performance of our method on the sequence of a moving face. The camera zooms and moves as the person changes her face's pose continuously. During the process, the face is partially occluded by a moving hand. A shape model is used to recover the part of the contour occluded by the hand and is incrementally updated as the tracking proceeds. The parameter β in (22) is set to be 200 for this sequence. The tracking results are shown in Figure 2.(b). We are still able to track the contour of the face even though it is occluded by a hand which has the similar color with it. From Frame 50 to Frame 118, the face looking ahead becomes looking downwards, which leads to shape changes. With the incremental shape model, continuous shape changes caused by pose changes are well handled. In the 107^{th} and 118^{th} frames, the contours still enclose the object tightly and accurately.

To stress the learning ability of the active shape model, obvious shape changes occur in the third experiment. We track a moving hand partially occluded by an object from Frame 1 to Frame 120. Firstly, the shape samples of hand with separate fingers are trained off-line. Secondly the trained shape model is used to conduct contour evolution when occlusion occurs in the tracking process. Thirdly, the shape of the hand is gradually changed to the one with combined fingers. The shape model is updated on-line and it is used to recover the occluded part when the new shape is confronted with occlusion. The parameter β in (22) is set to be 150 for this sequence. As we can see from the tracking results shown in Figure 2.(c), we keep good track of the contour of the occluded hand even though the new shape appears in the 110^{th} and 117^{th} frames. Our method is good adaptation to the new shapes.

In Figure 3, we have shown the tracking results for three different cases. The adopted sequence is the moving hand sequence in Figure 2.(c). The first case is we only consider the color information for the contour evolution. The tracking results are illustrated in Figure 3.(a). We can find that only considering the color feature is not enough to perform tracking when confronted with partial occlusions. Shape priors are needed to be incorporated into the whole framework. Figure 3.(b) and Figure 3.(c) are the latter two cases which add the shape priors into the whole tracking frame-

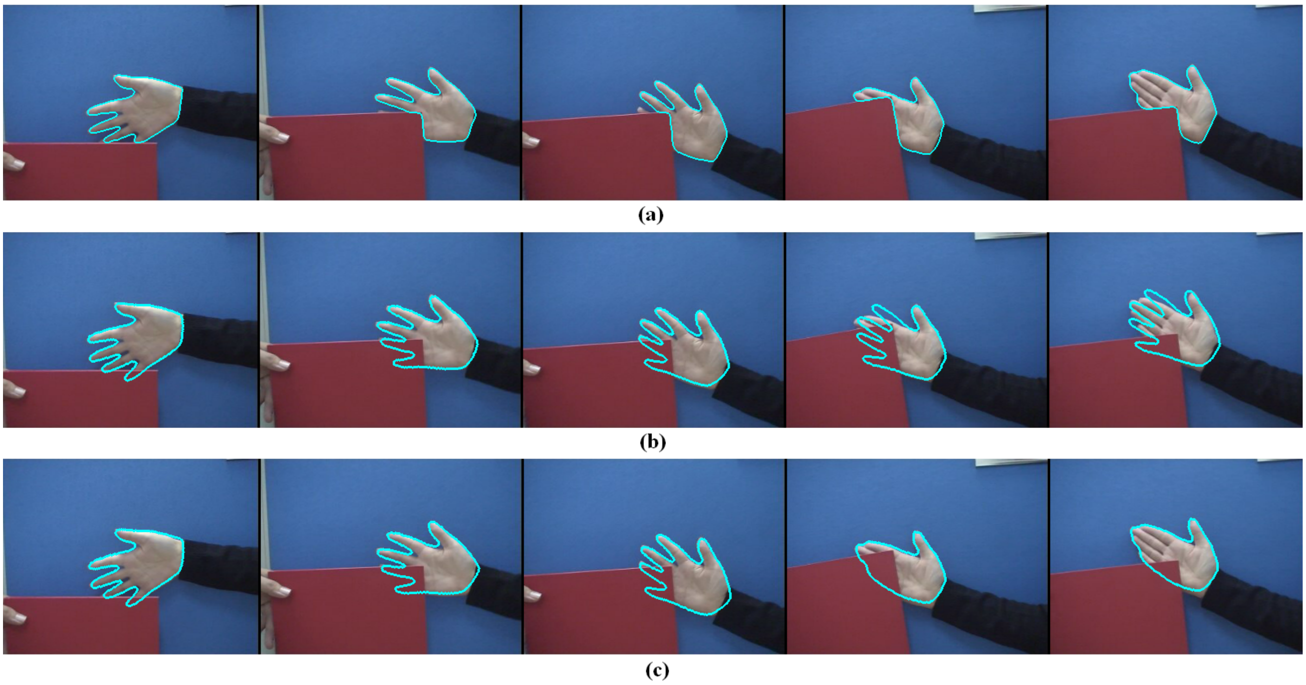


Figure 3. Tracking results for three different cases: (a) Tracking results of using color feature only; (b) Tracking results without the shape subspace update; (c) Tracking results of our proposed method which updates the shape subspace model on-line. The same column corresponds to the same frame. From left to right, the frame numbers are, respectively, 25, 38, 45, 110 and 117.

work. The difference between these two cases is the shape subspace model isn't updated during the tracking process in Figure 3.(b) and the shape model is incrementally updated in Figure 3.(c). As shown in Figure 3.(b), in the beginning, the shapes of the object don't change a lot, the tracking results are satisfied with the help of the shape priors. However, when new shapes appear, the stable shape priors couldn't provide the correct prior information. Thus, from this comparison, we can find our method with adaptive shape subspace model can keep good track of the objects with continuous shape changes.

7 Conclusions

In this paper, we have proposed a robust level-set based object tracking framework in which color information and shape priors are fused sequentially. Firstly, the initial contour is evolved using the color feature , providing the initial value for pixel-wise contour evolution with the shape constraint. Secondly, the obtained result is evolved in the shape subspace, providing the final value for pixel-wise contour evolution with the shape constraint. Thirdly, a weighted shape distance term (WSDT) is proposed into the pixel-wise contour evolution equation to balance the above two values. Finally, the obtained final result is returned to update the adaptive shape model incrementally. Our method has been tested on several real video sequences. Objects are accurately tracked under partial occlusions, background disturbance and motion blur. Experimental results have demonstrated the effectiveness of our approach.

8 Acknowledgments

This work is partly supported by NSFC (Grant No. 60520120099 and 60672040) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453).

References

[1] J.A. Sethian. "Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science", *Cambridge University Press,* 1999.

[2] M. Leventon, E. Grimson and O. Faugeras. “Statistical shape influence in geodesic active contours”, *in Computer Vision and Pattern Recognition,* vol.1, pp.316-323, 2000.

[3] D. Cremers. “Dynamical statistical shape priors for level set based tracking”, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.28, pp.1262-1273, Aug. 2006.

[4] A. Yilmaz, X. Li and M. Shah. “Object contour tracking using level sets”, *in Asian Conference on Computer Vision,* 2004.

[5] S.C. Zhu and A. Yuille. “Region competition: unifying snakes, region growing and bayes/mdl for multiband image segmentation”, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.18, pp.884-900, Sep. 1996.

[6] N. Paragios and R. Deriche. “Geodesic active contours and level sets for the detection and tracking of moving objects”, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.22, pp.266-280, Mar. 2000.

[7] T.F. Chan and L.A. Vese. “Active contours without edges”, *IEEE Trans.on Image Processing.,* vol.10, pp.266-277, Feb. 2001.

[8] S. Osher and J. Sethian. “Fronts propagation with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations ”, *J. Comput. Phys.,* vol.79, pp.12-49, 1988.

[9] M. Fussenegger, P.M. Roth, H.Bischof and A. Pinz. “On-line, incremental learning of a robust active shape model ”, *DAGM-Symposium.,* pp.122-131, 2006.

[10] J. Lim, D. Ross, R.S. Lin and M.H. Yang. “Incremental learning for visual tracking ”, *Neural Information Processing Systems Conference.,* MIT press, pp.793-800, 2004.

[11] N. Paragios and M. Rousson. “Shape priors for level set representations ”, *in European Conference on Computer Vision.,* pp.78-92, 2002.

[12] N. Paragios, M. Rousson and V. Ramesh. “Matching distance functions: a shape-to-area variational approach for global-to-local registration ”, *in European Conference on Computer Vision.,* pp.775-789, 2002.

[13] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K.S. Gopinath, R.W. Briggs and E. Geiser. “Using shape priors in geometric active contours in a variational framework ”, *International Journal of Computer Vision.,* 50(3), pp.315-328, 2002.

[14] D. Skočaj and A. Leonardis. “Weighted and robust incremental method for subspace learning ”, *Internation Conference on Computer Vision.,* vol.2, pp.1494-1501, 2003.

[15] T Chan and W. Zhu. “Level set based shape prior segmentation ”, *Technical report.,* UCLA, 2003.

[16] Y. Li, L.Q. Xu, J. Morphett and R. Jacobs. “On incremental and robust subspace learning ”, *Pattern Recognition.,* pp.1509-1518, 2004.

[17] D. Cremers, F. Tischh*ä*user, J. Weickert and C. Schn*ö*rr. “Diffusion snakes: introduction statistical shape knowledge into the Mumford-Shah functional ”, *International Journal of Computer Vision.,* 50(3) pp.295-313, 2002.

Upgrading Eigenspace-based Prediction using Null Space and its Application to Path Prediction

Yuji Shinomura
Hiroshima University, Japan
shino@eml.hiroshima-u.ac.jp

Toru Tamaki
Hiroshima University, Japan
tamaki@ieee.org

Toshiyuki Amano
NAIST, Japan
amano@is.naist.jp

Kazufumi Kaneda
Hiroshima University, Japan
kin@hiroshima-u.ac.jp

Abstract

This paper proposes a method for an Eigenspace-based prediction of a vector with missing components by modifying a projection of conventional Eigenspace method, and demonstrates the application to the prediction of the path of a walking person. This modification is based on domain-specific knowledge of data, and a linear combination of vectors in the null space *of Eigenspace is added so that a cost function of smoothness of path is minimized. Some experimental results on actual paths are shown to demonstrate how the proposed method works.*

1 Introduction

It is useful to estimate or predict unknown future data from previously observed data in past or present not only for meteorology and economics, but also for computer vision. Generally, the AR model or Kalman filter are used to estimate time series of data. Although predicting gestures and tracking people also use similar methods, a prediction for such sequences is not so simple because usually their behavior cannot be captured by the Gaussian signal model. On the other hand, patterns of behavior and motion of people in daily life have few variations: same gesture has similar motion and a same person walks in similar paths in a same scene. Thus, scene-dependent information of time series in such applications can be learned as prior knowledge in advance.

Eigenspace approach has been widely used to learn such domain-specific information from samples. Fod et al.[4] and Yacoob et al.[10] used Eigenspace to recognize motion of a person, and Nakajima et al.[7] predicted spatially and temporally to recognize gestures by Eigenspace made from sample gestures. These methods use learning of Eigenspace E with samples, and recognition and prediction are performed based on projection of a vector $\boldsymbol{x}$ onto Eigenspace spanned by several eigenvectors $\boldsymbol{e}_j$:

$$\boldsymbol{a} = E^T \boldsymbol{x}, \quad E = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots]. \tag{1}$$

A problem of these Eigenspace-based methods is that they merely use a projection of a vector with all components: i.e., for a vector to be recognized or predicted, it should have the same dimensionality as the samples that were used to construct the Eigenspace. However, we have no components corresponding to future data for prediction, and occluded data for recognition. A simple solution is to just put 0 for such missing components in the vector:

$$\hat{\boldsymbol{a}} = E^T \hat{\boldsymbol{x}}, \quad \hat{\boldsymbol{x}} = [x_1, x_2, \ldots, x_p, 0, 0, \ldots]^T, \tag{2}$$

but its result is awful[6] and a reconstructed vector $E\hat{\boldsymbol{a}}$ is not similar to the original vector $\boldsymbol{x}$.

One way to reconstruct a vector without using all pixels has been proposed by Leonardis et al.[6] to achieve a robust recognition when an object is occluded. Fidler et al.[3] utilized it to make LDA robust. Nakajima et al.[7] used a similar method for reconstruction and prediction, and Amano [2, 1] proposed methods to fill-in occluded regions. These methods are good for discrimination or recognition, but seem to fail to reconstruct or predict a vector with missing components[1] because characteristics of domain-specific data, such as smoothness or continuity, are ignored. These are summarized in section 2.

In this paper, we propose a new Eigenspace-based prediction method by modifying the conventional projection-based prediction with domain-specific knowledge of data,

[1] These missing components can be regarded as outliers, but robust subspace techniques such as a robust PCA proposed by De la Torre et al.[5] are not applicable because there are outliers not in learning samples but in a new test sample and also our case more than 50% components in the test sample are missing.

and demonstrate an application to predict the walking path of people. The modification uses *Null space*, the orthocomplement of Eigenspace, and a linear combination of vectors in the null space (*null vectors*) is added to the prediction so that a reconstructed vector with missing components (in our case, a person's walking path) satisfies some characteristic of data such as smoothness. Coefficients of the linear combinations are computed by the decent gradient method.

The organization of this paper is as follows. Eigenspace-based prediction is explained in Sec.2, then in Sec.3 we describe modification of the prediction with null vectors and estimation of linear combination of null vectors with the gradient decent method. Experimental results on actual paths are shown in Sec.4.

2 Eigenspace-based Prediction of a Path

This section introduces a prediction of a person's path based on projection onto Eigenspace.

2.1 Construction of Eigenspace with Sample Paths

In this paper, a path of a person is defined as a sequence of successive coordinates of the person over frames. Here we describe how to obtain a sequence of a path for learning.

First, regions in a frame where changes in intensity occur are extracted by using background subtraction. Then, the size of each region is used to reject regions other than people. The center of gravity of a region is used as a position of a person in the frame.

N paths are acquired for learning, then the paths are normalized in length that is defined as a sum of Euclidean distance between two successive coordinates. First the shortest path in the N paths is chosen. All paths are cut to the shortest length, then resampled so that all paths have the same length, M number of coordinates. Each i-th normalized path is represented by a vector $\boldsymbol{y}_i$ with $2M$ elements $\boldsymbol{p}_t$ as follows:

$$\begin{aligned} \boldsymbol{y}_i &= (\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \ldots, \boldsymbol{p}_M^T)^T \in \mathbb{R}^{2M}, && (3) \\ \boldsymbol{p}_t &= (p_{x_t}, p_{y_t})^T \in \mathbb{R}^2, && (4) \end{aligned}$$

where $\boldsymbol{p}_t$ is a 2D vector representing t-th coordinates in a path.

Eigenspace E is constructed with the normalized N sample paths $\boldsymbol{y}_i$ that are centered by subtracting an average vector $\boldsymbol{m}$ $(= \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_i)$ in advance, then eigenvectors $\boldsymbol{e}_i$ are computed:

$$\begin{aligned} E &= [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_N], && (5) \\ \boldsymbol{e}_i &= (\boldsymbol{e}_{i1}^T, \boldsymbol{e}_{i2}^T, \ldots, \boldsymbol{e}_{iM}^T)^T \in \mathbb{R}^{2M}, && (6) \end{aligned}$$

where E represents a matrix of Eigenspace spanned by the eigenvectors (or *Eigenpaths*) $\{\boldsymbol{e}_i\}$, and $\boldsymbol{e}_{it} \in \mathbb{R}^2$ corresponds to t-th 2D coordinates in $\boldsymbol{e}_i$.

2.2 Eigenspace-based Path Prediction

In prediction, a path of a new person is not fully traced, and there is no coordinates of the person in future. Suppose that a new person is tracked and the path is normalized to have s coordinates $\boldsymbol{p}'_1, \ldots, \boldsymbol{p}'_s$ as the same way for the learned paths.

$$\boldsymbol{y}' = (\boldsymbol{p}_1'^T, \ldots, \boldsymbol{p}_s'^T)^T \in \mathbb{R}^{2s}, \quad \text{where } s \le M. \qquad (7)$$

For unknown coordinates $\boldsymbol{p}'_{s+1}, \ldots, \boldsymbol{p}'_M$, we set them to zero $\boldsymbol{p}'_t = \boldsymbol{0} = (0,0)^T$, then an extended vector $\boldsymbol{y}''$ is obtained:

$$\begin{aligned} \boldsymbol{y}'' &= (\boldsymbol{p}_1'^T, \ldots, \boldsymbol{p}_s'^T, \underbrace{\boldsymbol{0}^T, \ldots, \boldsymbol{0}^T}_{(M-s)})^T && (8) \\ &= (\boldsymbol{y}'^T, \underbrace{0, \ldots, 0}_{2(M-s)})^T \in \mathbb{R}^{2M}. && (9) \end{aligned}$$

In the framework of conventional Eigenspace methods, the observed vector $\boldsymbol{y}''$ is represented by a linear combination of the eigenvectors so that the following L-2 error norm is minimized [6] with respect to $\boldsymbol{a}$:

$$\begin{aligned} ||\boldsymbol{y}'' - E\boldsymbol{a}||^2 &= \left|\left| \boldsymbol{y}'' - \sum_j^N a_j \boldsymbol{e}_j \right|\right|^2 && (10) \\ &= \sum_t^M \left|\left| \boldsymbol{p}'_t - \sum_j^N a_j \boldsymbol{e}_{jt} \right|\right|^2, && (11) \end{aligned}$$

where $\boldsymbol{a} = (a_1, a_2, \ldots, a_N)^T$ is the coefficient of the linear combination. In our case, unknown coordinates are set to zero, so the norm is rewritten as:

$$\begin{aligned} ||\boldsymbol{y}'' - E\boldsymbol{a}||^2 = &\sum_{t=1}^{s} \left|\left| \boldsymbol{p}'_t - \sum_j^N a_j \boldsymbol{e}_{jt} \right|\right|^2 \\ &+ \sum_{t=s+1}^{M} \left|\left| \sum_j^N a_j \boldsymbol{e}_{jt} \right|\right|^2. \qquad (12) \end{aligned}$$

The second term in the above equation affects greatly the estimates of the coefficient $\boldsymbol{a}$. Instead, using only the first term and omitting the second term lead to a more appropriate estimate of the coefficient. This estimation is done by solving the following linear system [6, 3]:

$$\begin{aligned} E'^T E' \boldsymbol{a} &= E'^T \boldsymbol{y}'', && (13) \\ E' &= \mathrm{diag}(\overbrace{1, \cdots, 1}^{2s}, \overbrace{0, \cdots, 0}^{2(M-s)})\, E && (14) \end{aligned}$$

where E' is a subspace of E spanned by truncated eigenvectors, but their basis are no longer orthogonal to each other. Note that $\mathrm{rank}(E'^T E') = N$ or $\det(E'^T E') \neq 0$ should be held so that the linear system doesn't become underdetermined. This means $2s > N$, hence the estimation can be done after several positions of a person are observed.

The reconstruction with the estimated coefficients $\boldsymbol{a}$ is as follows [2, 1, 7]:

$$\boldsymbol{y}^* = E\boldsymbol{a} = E(E'^T E')^{-1} E'^T \boldsymbol{y}''. \qquad (15)$$

2.3 Modifying a Projection Outside of Eigenspace

The predicted path $\boldsymbol{y}^*$ is represented by a linear combination of eigenvectors $\boldsymbol{e}_i$,

$$\boldsymbol{y}^* = E\boldsymbol{a} = a_1\boldsymbol{e}_1 + a_2\boldsymbol{e}_2 + \cdots + a_N\boldsymbol{e}_N = \sum_{i=1}^{N} a_i\boldsymbol{e}_i. \quad (16)$$

However, Eq.(15) shows us that $\boldsymbol{y}^*$ is a projection of $\boldsymbol{y}''$ onto a subspace spanned by non-orthonormal vectors[8, 3], in this case not E but the truncated subspace E'. Therefore, there is no reason to believe that the projection represents the original data well because the truncation of the Eigenspace depends not on principal components corresponding to small eigenvalues (usually referred as dimensionality reduction) but just the length of observation. Also, this projection does not take into account the characteristics of a person's walking path, and the estimated path $\boldsymbol{y}^*$ results in something different from a real path.

In this paper, we propose the use of the orthocomplement of the Eigenspace, denoted as $E^\perp$, where $\mathbb{R}^{2M} = E + E^\perp$. All vectors in $E^\perp$ are orthogonal to any vectors in E, and vice versa: e.g., $\forall \boldsymbol{\ell} \in E^\perp \Rightarrow E\boldsymbol{\ell} = 0$. Therefore, we call $E^\perp$ as the *null space* of E, and a vector in the null space is called a *null vector*. By using null vectors in the null space, a path is represented as follows:

$$\widetilde{\boldsymbol{y}} = \boldsymbol{y}^* + \sum_k b_k\boldsymbol{\ell}_k = \sum_{i=1}^{N} a_i\boldsymbol{e}_i + \sum_k b_k\boldsymbol{\ell}_k \qquad (17)$$

Estimated path $\boldsymbol{y}^*$ in Eq.(15) is identical to the equation above when coefficients b_k for null vectors in the second term are zero.

The concept of the proposed method is that domain-specific knowledge discarded by the conventional projection can be found in the null space if we can find the appropriate coefficients b_k for the null vectors $\boldsymbol{\ell}_k$. This topic is described in the next section. It should be noted that the projection of $\widetilde{\boldsymbol{y}}$ onto E is still $\boldsymbol{y}^*$.

3 Null Vector Modifications

The proposed method shown in this section adds null vectors to the projected path $\boldsymbol{y}^*$ so that the modified path $\widetilde{\boldsymbol{y}}$ looks like a person's walking path. In this paper, we make an assumption that *a person walks toward a destination, and does not turn suddenly, and the path is smooth and does not have a sharp curve*. Here we introduce a cost function of smoothness of a path that has never been used by conventional Eigenspace-based estimations.

First, we assume that K null vectors $\boldsymbol{\ell}_k = (\boldsymbol{\ell}_{k1}^T, \boldsymbol{\ell}_{k2}^T, \ldots, \boldsymbol{\ell}_{kM}^T)^T \in E^\perp$ are given. Then the linear representation of the modified path $\widetilde{\boldsymbol{y}}$ is:

$$\begin{aligned} \widetilde{\boldsymbol{y}} &= \sum_{i=1}^{N} a_i\boldsymbol{e}_i + \sum_{k=1}^{K} b_k\boldsymbol{\ell}_k \\ &= (\widetilde{\boldsymbol{p}}_1^T, \widetilde{\boldsymbol{p}}_2^T, \ldots, \widetilde{\boldsymbol{p}}_M^T)^T, & (18) \\ \widetilde{\boldsymbol{p}}_t &= \sum_{i=1}^{N} a_i\boldsymbol{e}_{it} + \sum_{k=1}^{K} b_k\boldsymbol{\ell}_{kt}. & (19) \end{aligned}$$

Let $\boldsymbol{u}_t$ be a vector defined by two successive[2]coordinates $\widetilde{\boldsymbol{p}}_t, \widetilde{\boldsymbol{p}}_{t+1}$, and θ_t be an angle subtended by $\boldsymbol{u}_t$ and $\boldsymbol{u}_{t+1}$:

$$\boldsymbol{u}_t = \widetilde{\boldsymbol{p}}_{t+1} - \widetilde{\boldsymbol{p}}_t, \qquad (20)$$

$$\cos\theta_t = \frac{\boldsymbol{u}_t^T \boldsymbol{u}_{t+1}}{||\boldsymbol{u}_t||\,||\boldsymbol{u}_{t+1}||}, \quad 1 \leq t \leq M-2. \qquad (21)$$

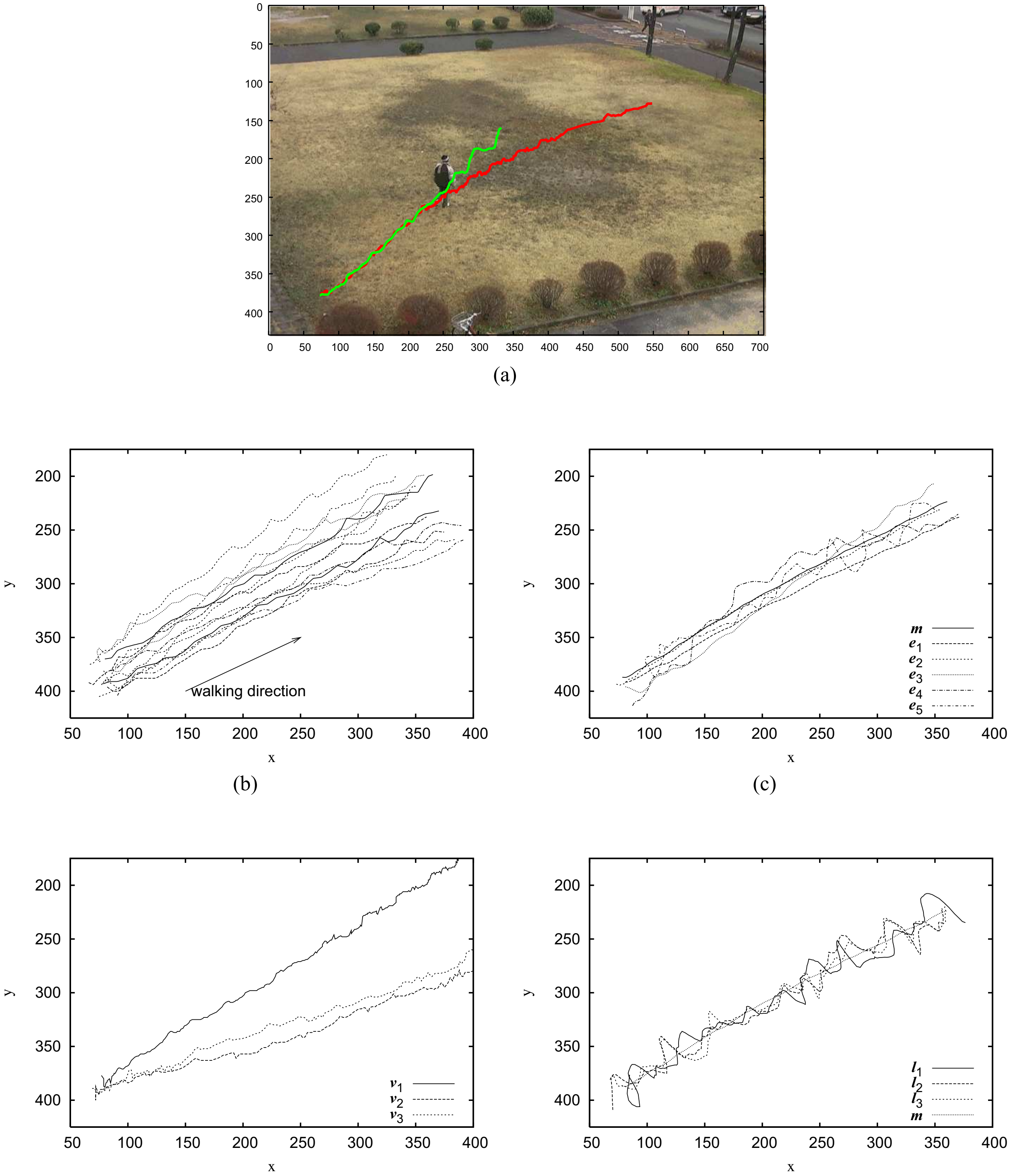
Next, we define a cost function J so that the smaller the angle θ_t is the smoother the path is:

$$J = \sum_{t=1}^{M-2} \cos^\alpha \theta_t, \qquad \alpha = 1, 3, 5, \ldots \qquad (22)$$

The steepest gradient method is used to maximize the cost function for coefficients of the null vectors b_k $(k = 1, \ldots, K)$ as $b_k \leftarrow b_k + \frac{\partial J}{\partial b_k}$, and all b_k are initialized to 0. A stopping condition is $\max_k \left|\frac{\partial J}{\partial b_k}\right| < 10^{-5}$. The Jacobian of J comprises $\boldsymbol{u}_t$ and $\boldsymbol{\ell}_t$ (omit detail).

In the discussion above, we assume that the null vectors are given. However, there are no established methods to get null vectors. Also there are a lot of variations to choose null vectors from the null space. For example, assume that there are 13 paths comprised of 250 coordinates given as samples. The dimensionality of the Eigenspace is up to 13, however, the null space has $500 - 13 = 487$ dimensions. Usually the number of samples is much fewer than the number of coordinates in a path. Therefore it is difficult to find the most appropriate null vector to modify the predicted path.

[2] Of course, we can two coordinates distant from each other $\widetilde{\boldsymbol{p}}_t$ and $\widetilde{\boldsymbol{p}}_{t+k}$. In this case, the sum of k-curvature (see, for example, [9]) over the path is minimized.



(a)

(b)

(c)

(d)

(e)

Figure 1. A frame of video and paths used in the experiments. (a) Predicted path y^* (green) and actual path y (red). (b) 13 sample paths $y_1, \ldots, y_{13}$, (c) 5 eigenvectors $e_1, \ldots, e_5$, (d) 3 samples used for null vectors $v_1, \ldots, v_3$ and (e) 3 null vectors $\ell_1, \ldots, \ell_3$. Note that e_j and ℓ_k are scaled properly for visualization.

Table 1. Initial and converged values of the cost function J with a null vector $\boldsymbol{\ell}_1$ for different α.

α	1	3	5	7	9
J (init.)	234.56	215.23	201.26	190.71	181.87
J (conv.)	235.20	217.95	206.06	196.84	189.18
b_1	-15.14	-27.36	-34.18	-36.96	-38.33

In this paper, null vectors are obtained from paths other than sample paths. In general, the dimension is so high that the new paths probably do not lay on the Eigenspace spanned by the sample paths. The null vectors $\boldsymbol{\ell}_k$ in the null space are made from the new vectors $\boldsymbol{v}_k$ by using the Gram-Schmidt orthonormalization:

$$\boldsymbol{\ell}'_k = \boldsymbol{v}_k - \sum_{i=1}^{N}(\boldsymbol{v}_k^T\boldsymbol{e}_i)\boldsymbol{e}_i - \sum_{j=1}^{k-1}(\boldsymbol{v}_k^T\boldsymbol{\ell}_j)\boldsymbol{\ell}_j, \quad (23)$$

$$\boldsymbol{\ell}_k = \frac{\boldsymbol{\ell}'_k}{|\boldsymbol{\ell}'_k|}. \quad (24)$$

We may add the new paths for null vectors to the learning sample paths to construct the Eigenspace instead to make the null space. This way seems to make use of the information of the new paths for better prediction, however, all information of learning samples are truncated by eq.(14), then no way to retrieve information corresponding to the missing components in the new paths.

4 Experimental Results

We implemented the proposed method, and evaluated using real image sequences of 714×480 in size. In the experiment, a video camera was fixed to a tripod, and movies were recorded as MPEG files, then 17 paths were obtained by off-line processing. People walked from the bottom left to the top right of the frame (Fig.1(a)). 13 paths were used as samples to make an Eigenspace (Fig.1(b)), and another three paths were used for null vectors (Fig.1(d)(e)). The remaining path is used for prediction (Fig.2). When learning, each path was cut so that it consisted of 350 coordinates, then 50 points are sparsely downsampled with linear interpolation for noise reduction. Finally $M = 250$ coordinates are resampled for a path. When predicting, a person is tracked and the path was normalized at each frame.

Predicted paths $\boldsymbol{y}^*$ with $N = 13$ are shown in Fig.2(a) for several different positions $\boldsymbol{p}'_s$ represented by ◯. The prediction near to the start position (for small s) deviated largely from the actual path. As s increases, the prediction becomes similar to the actual path.

Next, Fig.2(b) shows the modification by a null vector $\boldsymbol{\ell}_1$. The estimated path $\boldsymbol{y}^*$ was predicted at $s = 100$. Actually the modification is slight, but $\widetilde{\boldsymbol{y}}$ is indeed more smoother than $\boldsymbol{y}^*$. Table.1 shows values of the cost function and estimated coefficient b_1 when α changes. Although b_1 differs for different α, this variation is so small and does not affect the shape of the path because the null vector $\boldsymbol{\ell}_1$ has 500 elements but its norm is normalized to 1. Therefore, the choise of α is trivial and we set $\alpha = 1$ for all experiments.

Table 2. Comparison of the number of the null vector.

	$\boldsymbol{\ell}_1$	$\boldsymbol{\ell}_2$	$\boldsymbol{\ell}_3$	$\sum b_k\boldsymbol{\ell}_k$
K	1	1	1	3
J	235.20	234.67	239.88	240.727
b_1	-39.73	0	0	-4.80
b_2	0	-5.90	0	14.07
b_3	0	0	-34.54	-42.62

Fig.3(a) illustrates results of modification by each null vector. Fig.3(b) shows the result by using 3 null vectors at the same time, and Table.2 shows the estimated parameters. Although the modified path depends on which path is used, the difference is small.

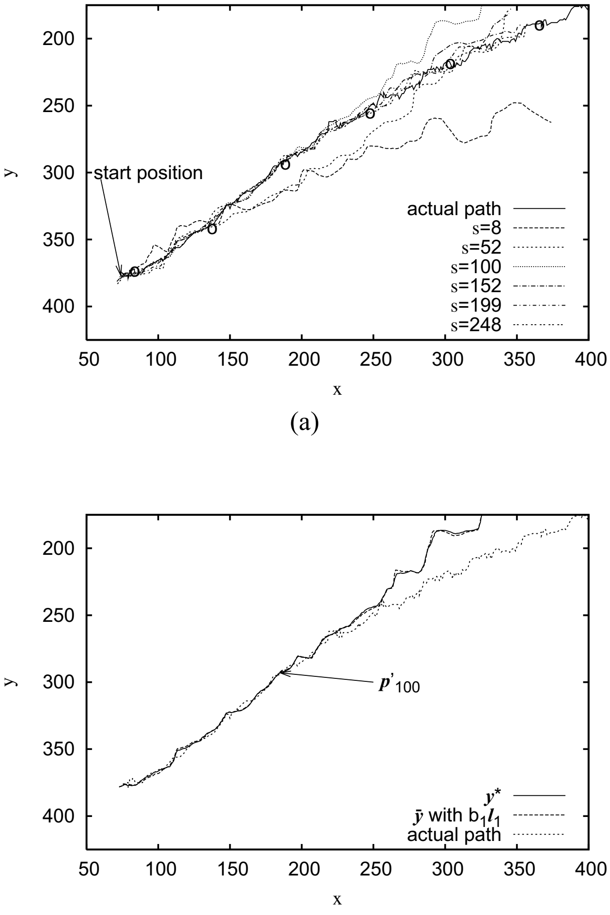
Another experiment is shown in Fig.4. Fig.4(a) shows 30 sample paths used to construct Eigenspace. Unlike the previous experiment, the walking path curves twice and looks like the S letter. Predicted and modified paths of a new path are shown in Fig.4(b) for different positions. This result shows that the proposed method is applicable to curved complex path in which prior knowledge is effectively used.

5 Conclusion

In this paper, we proposed a method for predicting a vector with missing components based on Eigenspace with null space modifications. We applied the method to paths of walking people in a real sequence, and demonstrated in the limited experiments how the proposed method works. There are many things to be considered, such as the number of the null vectors, the way to obtain the null vectors, the choice of other cost functions that represent domain-specific knowledge. Also futher experiments should be done. Nevertheless, the concept of the proposed method — to explore *out of the subspace* spanned by samples based on a prior knowledge — can be applicable to any other subspace recognition methods. We will investigate the possibility in other pattern recognition problems.

Acknowledgement

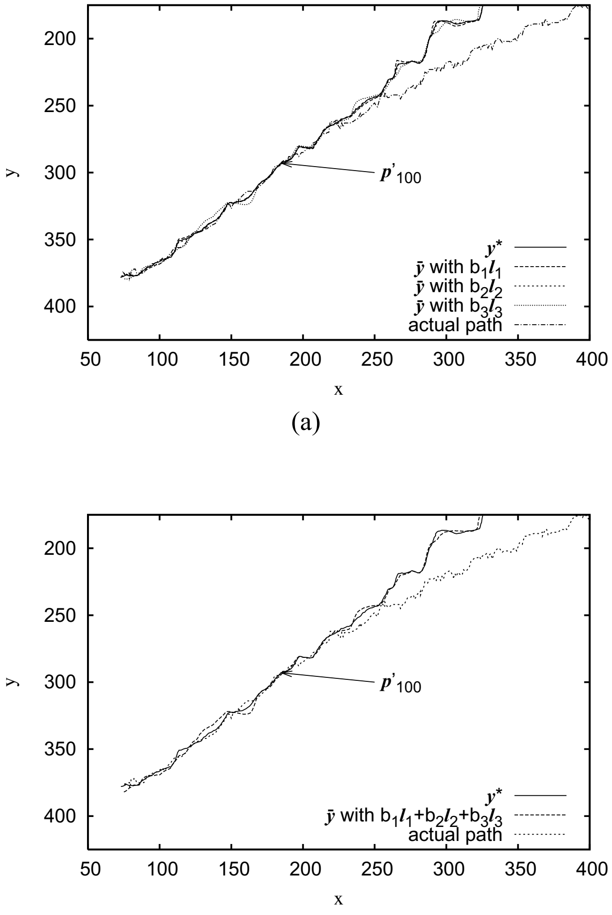
We thank Tomotaka Harano, an undergraduated student at Hiroshima University, for taking movies for the research.



(a)

(b)

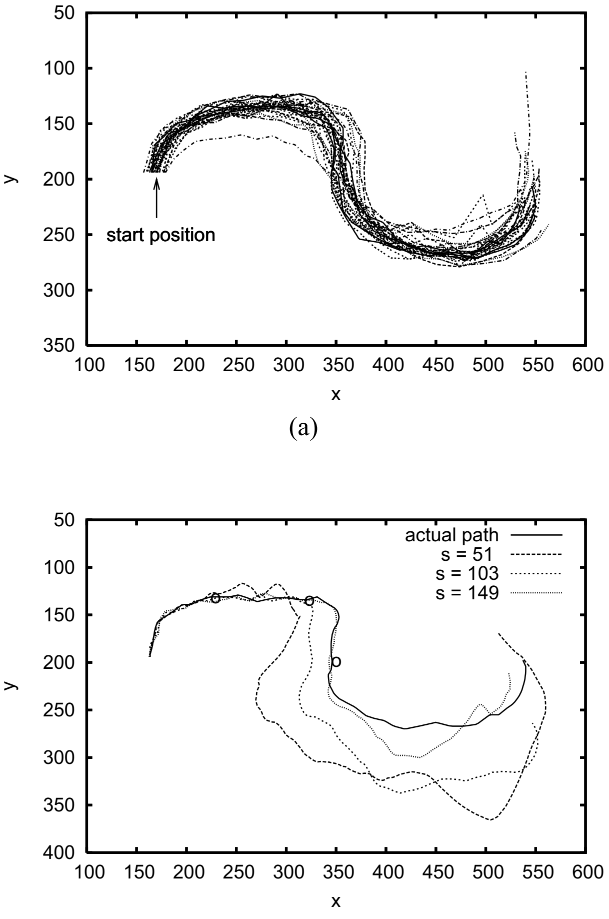
Figure 2. (a) Predicted paths y^* for different position s. (b) Predicted path y^* (solid) when $s = 100$, and modified path $\widetilde{y}$ (dashed).



(a)

(b)

Figure 3. (a) Modified path $\widetilde{y}$ using each null vector. (b) Modified path $\widetilde{y}$ using 3 null vectors.



(a)

(b)

Figure 4. (a) Learned 30 paths. $N = 30, M = 300$**. (b) Predicted and modified path.**

References

[1] T. Amano. Image interpolation by high dimensional projection based on subspace method. *Proc. of ICPR2004*, 4:665–668, 2004.

[2] T. Amano and Y. Sato. Image interpolation using BPLP method on the eigenspace. *Systems and Computers in Japan*, 38(1):457–465, 2007.

[3] S. Fidler, D. Skocaj, and A. Leonardis. Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(3):337–350, 2006.

[4] A. Fod, M. J. Matarić, and O. C. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, 2002.

[5] F. D. la Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54:117–142, 2003.

[6] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118, 2000.

[7] M. Nakajima, S. Uchida, A. Mori, R. Kurazume, R. Taniguchi, T. Hasegawa, and H. Sakoe. Motion prediction based on eigen-gestures. *Proc. of the 1st First Korea-Japan Joint Workshop on Pattern Recognition (KJPR2006)*, pages 61–66, 2006.

[8] E. Oja. *Subspace Methods of Pattern Recognition*. Research Studies Press, 1983.

[9] Y. Shirai. *Three-Dimensional Computer Vision*. Springer-Verlag, 1987.

[10] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73(2):232–247, 1999.

The secret of rotating object images — Using cyclic permutation for view-based pose estimation —

Toru Tamaki
Hiroshima University, Japan
tamaki@ieee.org

Toshiyuki Amano
NAIST, Japan
amano@is.naist.jp

Kazufumi Kaneda
Hiroshima University, Japan
kin@hiroshima-u.ac.jp

Abstract

In this paper, we propose a novel pose estimation method for a cyclic image squence of a rotating object with subspace by block diagonalization of a matrix representing transformation from an image to another. The transformation by the matrix is formulated as the action of cyclic group, and the power of a block diagonal matrix represents pose and appearance change in the sequence. Distance-based and angle-based methods are proposed to estimate pose. Experimental results with real image sequences of COIL-20 demonstrate that the subspace proposed in this paper is useful for pose estimation.

1 Introduction

When a three dimensional object rotates about an axis (as shown in Fig.1), the sequence of images of the object is cyclic: the last image is followed by the first image. When we have such a sequence of n images $\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_{n \quad 1}$, the cyclic property is represented by cyclic group:

$$\boldsymbol{x}_{j+1 \bmod n} = G\boldsymbol{x}_j .$$

G is an element of a cyclic group, however, we can think it is a matrix. This relationship is essential for images of one parameter rotation, but no attentions have been paid. We propose to use the cyclic property for view-based pose estimation by linear subspace approach.

1.1 Related works

Estimation of pose of an object in an image is an important task in computer vision and pattern recognition, and methods are categorized into model-based and view-based. Model-based methods, such as [8], assume a model is given: such as known object shape, rigid motion, and projections. This approach estimates 3DOF (degrees of freedom) rotation of objects, however, requires precise geometry and restricted scene where the model can be applied. On the other hand, the advantage of view-based or appearance-based methods is to use just images of the object and make no assumptions about shape of objects and projections from 3-D to 2-D. Although it is difficult to deal with 3DOF rotation, many studies have been done even if the rotation is 1DOF (one parameter rotation).

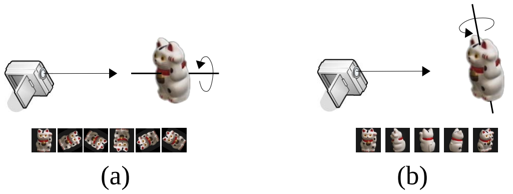

(a)
(b)

Figure 1. Images of an object by (a) in-plane and (b) off-the-plane rotation.

A major view-based method is Parametric Eigenspace method proposed by Murase et al.[10]. It learns Eigenspace of images of an object with continuously changing pose parameters. This method has been applied in a variety of areas and demonstrated its usefulness. However, there are practical problems including that it is not easy to extend the expression of spline to many (more than 2) DOF, and the search over a spline curve/surface is not closed-form but an iterative search involving expensive computation. And a theoretical question arises: *what is the Eigenspace or subspace of images of a 3D object rotating about an axis?*

For some special cases, analyses has been developed. Uenohara et al. [19, 5] proposed an efficient computation of Eigenspace for images rotating about the optical axis (so just two dimensional image rotation, or *in-plane rotation* as shown in Fig.1(a)) by using DCT or DFT[13]. Chang et al.[3] showed the same result for translational shift. Jorgan et al.[7, 5] extended for images of multiple objects rotating in-plane. Sengel et al.[17] considered in the limit when the number of images is infinite for Jorgan's method[7], then estimated a pose parameter directly with arctan.

It is not easy for *off-the-plane rotation*: when an object is rotated about an arbitrary axis in three dimensional space (see Fig.1(b)). In the case of in-plane rotation, appearance of an object in images basically does not change. But in three dimensional rotation, even for 1DOF, it is impossible to find eigenvectors analytically because the appearance change depends on many properties of an object, such as shape, reflectance, shadow and etc. Therefore, many researches have been done with kernel methods or nonlinear manifold learning such as [21, 18]. Gabriele [14, 15] proposed feature-based pose estimation and view generation with elaborated grid graph representation with Gabor jets. Zhao et al.[21] used kernel PCA instead of linear PCA[10], and recently Vik et al.[20] proposed non-Gaussian modeling of appearance subspace with a method similar with [10].

However, there are few linear subspace approaches while it is still important[4]. Chang et al.[3] demonstrated to compute eigenvectors for synthetic images of an 3D cylinder just painted in black and white and rotated about an axis, then observed that eigenvectors of the images are similar with cosines. Sengel et al.[17] handled appearance changes in images of a rotating object as different image templates, but continuous pose parameters are not estimated.

A subspace approach for off-the-plane rotation was proposed by Okatani et al.[12]. They applied linear regression to the problem: first relates images with parameters by a linear map (matrix), and estimates the matrix by using pseudoinverse, then parameters are estimated by applying the matrix to an image of novel view. Amano et al.[1] used a variation of pseudoinverse with dimensionality reduction of Eigenspace of images, then estimated pose parameter linearly. Some authors use kernel methods: Ando et al.[2] used support vector regression instead of linear regression for 3DOF rotation, and Melzer et al.[9] employed kernel canonical correlation analysis (kernel CCA) for 2DOF.

These regression-like methods have shown their ability of pose estimation. However, they do not explain how the images are represented in a subspace. The answer has been shown for in-plane rotation by analytically obtained eigenvectors, but still not for off-the-plane rotation.

1.2 Our approach

In this paper, we propose a novel approach for off-the-plane rotation with a cyclic group acting on an image sequence. As mentioned above, analytical methods derived eigenvectors of images of in-plane rotating object, while regression methods used a matrix between images and parameters for off-the-plane rotation. In contrast, the proposed method focuses on the transformation from an image to another in an image sequence of off-the-plane 1DOF rotation in three dimensional space. The transformation can be seen as cyclic group, and we represent it as a matrix decomposed by block diagonalization. The main contribution of this paper is to show that the appearance change in an off-the-plane sequence can be realized by the power of the block diagonal matrix discussed from the view point of subspace. This have never been done by regression/CCA subspace methods or analytical Eigenspace methods.

2 Formulation of appearance change in image sequence with cyclic permutation

2.1 Matrix representation of relationship between images

We represent a relationship of n images in a given image sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n\ \ 1}$. The images are taken by rotating an object about an axis in three dimensional space (i.e., off-the-plane rotation), and each image $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jN})^T \in \mathbb{R}^N$ is a N dimensional vector taken at angle[1] $\theta_j = 2j\pi/n$. Throughout the paper, we assume $N > n$, the number of pixels in the images is larger than the number of images.

First we consider the following matrix G that transforms an image vector $\boldsymbol{x}_j$ into $\boldsymbol{x}_{j+1}$:

$$\boldsymbol{x}_{j+1 \bmod n} = G\boldsymbol{x}_j, \quad \boldsymbol{x}_j = G^j\boldsymbol{x}_0, \quad \boldsymbol{x}_j = G^n\boldsymbol{x}_j. \tag{1}$$

This transformation is the result of the action by a cyclic group $G_n = \{G, G^2, \ldots, G^n\}$ of degree n acting from left side on the image sequence. G is called a generator (or primitive element) of G_n, and G^n is an identity element. The group theory is an abstract concept, however, we focus only on linear transformation: that is, throughout the paper, $G \in \mathbb{R}^{N \times N}$ is a matrix and $\boldsymbol{x} \in \mathbb{R}^N$ is a vector.

However, one can ask the question: *Why can you obtain the jth image $\boldsymbol{x}_j$ from the first image $\boldsymbol{x}_0$ by just multiplying a matrix j times? When $\boldsymbol{x}_0$ is the frontal pose and $\boldsymbol{x}_j$ is the back, due to occlusions and so, does not $\boldsymbol{x}_j$ have any common information with $\boldsymbol{x}_0$?* The answer is below.

The transform can be written in a matrix form as follows:

$$[\boldsymbol{x}_1\, \boldsymbol{x}_2\, \cdots\, \boldsymbol{x}_{n\ \ 1}\, \boldsymbol{x}_0] = G[\boldsymbol{x}_0\, \boldsymbol{x}_1\, \cdots\, \boldsymbol{x}_{n\ \ 2}\, \boldsymbol{x}_{n\ \ 1}], \tag{2}$$

or

$$X_1 = GX_0, \tag{3}$$

where

$$X_1 = [\boldsymbol{x}_1\, \boldsymbol{x}_2\, \cdots\, \boldsymbol{x}_{n\ \ 1}\, \boldsymbol{x}_0], \tag{4}$$

$$X_0 = [\boldsymbol{x}_0\, \boldsymbol{x}_1\, \cdots\, \boldsymbol{x}_{n\ \ 2}\, \boldsymbol{x}_{n\ \ 1}]. \tag{5}$$

[1] For simplicity, the angles are evenly spaced. If the angles are irregularly sampled, the linear function $\theta(j) = 2j\pi/n$ is replaced with an appropriate nonlinear function such as piecewise linear functions or a spline curve.

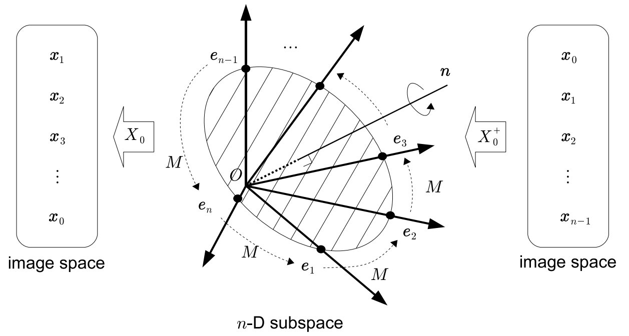


Figure 2. Two projections X_0^+, X_0 and a rotation M composing the transformation G.

Here we obtain G with X_0^+, a Moore-Penrose generalized (pseudo) inverse of X_0 with the singular value decomposition (SVD) $X_0 = E\Sigma V^T$, as follows:

$$G = X_1 X_0^+, \quad X_0^+ = (X_0^T X_0)\ ^1 X_0^T = V\Sigma\ ^1 E^T. \quad (6)$$

Therefore, the answer of the question above is that the matrix G indeed transforms $\boldsymbol{x}_0$ to $\boldsymbol{x}_j$ whatever the geometry of an object in the images is[2]. The reason is that Eq.(3) is an under-determined system because $N > n$. Of course the pseudoinverse in Eq.(6) is not a unique[3] and many pseudoinverses hold Eq.(3), however, this is not a problem but a necessary condition that Eq.(1) and Eq.(3) exactly hold.

When we consider the transformation from X_0 to X_1, it can be represented with a $n \times n$ column permutation matrix M multiplied from right side of X_0:

$$X_1 = X_0 \begin{pmatrix} 0 & & & & & 1 \\ 1 & 0 & & & & \\ & 1 & 0 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 0 & \\ & & & & 1 & 0 \end{pmatrix} = X_0 M, \quad (7)$$

then Eq.(6) is rewritten as follows:

$$G = X_0 M X_0^+. \quad (8)$$

2.2 Two projections and a rotation

With Eq.(8), it is interesting that we can interpret G as an combination of projections to an subspace and a rotation in the subspace. See Fig.2.

First G transforms the sequence X_0 into I_n ($n \times n$ identity matrix) because of $X_0^+ X_0 = I_n$. This means $\boldsymbol{x}_{j\ 1} \mapsto \boldsymbol{e}_j$, i.e., each image $\boldsymbol{x}_{j\ 1}$ is mapped to a canonical unit vector $\boldsymbol{e}_j$ in which all components are 0 except j th component is 1. Next, M moves the unit vector $\boldsymbol{e}_j$ to $\boldsymbol{e}_{j+1}$. This can be done by just shifting components in $\boldsymbol{e}_j$, but M is indeed rotation about the axis $\boldsymbol{n} = (1, 1, \ldots, 1) \in \mathbb{R}^n$ and makes the unit vector form a locus of a hypercircle on a hyperplane[4] in $\mathbb{R}^n$. Finally X_0 projects vectors back to the image space from the subspace.

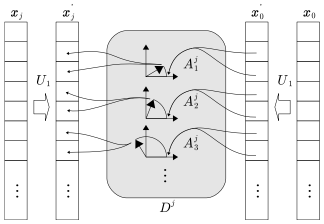


Figure 3. Rotations by A_k^j in 2-D subspaces.

Therefore, the images in the sequence are projected onto the circle in the subspace, and well separated with distance $\sqrt{2}$ from each other[5], and transfered from one to the next by M.

For recognizing unknown pose between learned poses, the concept of the proposed method is to extend this *discrete rotation* M into *continuous rotation* by interpolating M with block diagonalization discussed below.

2.3 Decomposition of G

M is decomposed with a real block diagonal matrix D and a real orthogonal matrix W as $M = WDW^T$. Then, the decomposition of G is

$$G = U_2 D U_1, \quad U_1 = W^T X_0^+, \quad U_2 = X_0 W, \quad (9)$$

where

$$D = \begin{pmatrix} 1 & & & \\ & A_1 & & \\ & & A_2 & \\ & & & \ddots \end{pmatrix}, \quad A_k \in \mathbb{R}^{2\times 2}, \quad (10)$$

D has 2×2 blocks A_k at its diagonal part. See appendix for the detail of the block diagonalization.

With $U_1 U_2 = I_n$, the transformation from $\boldsymbol{x}_0$ to $\boldsymbol{x}_j$ can be represented as

$$\boldsymbol{x}_j = U_2 D^j U_1 \boldsymbol{x}_0, \quad (11)$$

instead of $\boldsymbol{x}_j = G^j \boldsymbol{x}_0$.

Here, the matrix U_1 can be regarded as a projection from the image space onto a n-dimensional subspace representing the pose of an object in the images. See Fig.3 in which we call $\boldsymbol{x}\ = U_1 \boldsymbol{x}$ *an image in the subspace*. Each pair of

[2] Imagine how large G is — $N \times N$! Even when G is decomposed, U_1 and U_2 are the same size with X_0. Therefore, G has so enough elements that represent information between images even if $\boldsymbol{x}_0$ and $\boldsymbol{x}_j$ do not have.

[3] It is unique in the sense that a minimum norm solution is given.

[4] It is perpendicular to $\boldsymbol{n}$, and the distance to the origin is $\frac{1}{\sqrt{n}}$.

[5] $\forall j, k, j \neq k \Rightarrow ||\boldsymbol{e}_j - \boldsymbol{e}_k|| = \sqrt{1+1} = \sqrt{2}$.

row vectors of U_1 corresponding a 2×2 block A_k of D is a linear projection from the image space onto two dimensional (2-D) subspace spanned by the row vectors. These 2-D subspaces are independent and orthogonal to each other because all blocks do not overlap. Therefore, the projection of an original image is a set of projections onto different 2-D subspaces, and multiplying D in the subspace means 2-D rotations (with A_k by θ_k) of 2-D vectors comprised of two pixels of the image in the subspace.

2.4 Demonstrating the subspace

As the derivation above, the matrix G transform an image to another in the image sequence X_0 by the power of G:

$$\boldsymbol{x}_j = G^j \boldsymbol{x}_0, \quad \text{or} \quad \boldsymbol{x}_j = U_2 D^j U_1 \boldsymbol{x}_0. \tag{12}$$

Therefore, j (the power of D^j) decides how much the image $\boldsymbol{x}_0$ is transformed in the image sequence.

Now we are interested in not only observing the transformation from $\boldsymbol{x}_0$ to $\boldsymbol{x}_j$ but also extending the range of the power j from several integer numbers $(0, 1, \ldots, n-1)$ to a real interval $[0, n[$.

For an off-the-plane rotation sequence, Fig.4(a) demonstrates an example using object 4 in COIL-20 [11]. 36 images including $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2$ (0, 10, 20[deg]) are used for learning. Two images (5,15 [deg]) corresponding to $\boldsymbol{x}_{0.5}, \boldsymbol{x}_{1.5}$ are shown for comparison. The lower row shows images $\boldsymbol{x}_{0.1j}$ created by

$$\boldsymbol{x}_{0.1j} = G^{0.1j} \boldsymbol{x}_0 = U_2 D^{0.1j} U_1 \boldsymbol{x}_0, \tag{13}$$

for $j = 0, 1, 2, \ldots, 20$. The created images $\boldsymbol{x}_{1.0}$ and $\boldsymbol{x}_{2.0}$ are exactly same with the learned images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. For the other images between learned images, especially $\boldsymbol{x}_{0.5}$ and $\boldsymbol{x}_{1.5}$ for comparison, the appearance are very similar with actual intermediate images. Actually they look like one made by blending two learned images, but our objective is not to make created images close to the real ones, but to utilize them for pose estimation as shown in the next section.

Although the proposed method is formulated for a single axis rotation, Eq.(3) can be applicable to any revolving image sequence such as a light turns around in front of a face. Fig.4(b) illustrates such an example for different light directions. 20 face images of P00 in the Yale Face Database B [6] including $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5$ (cropped) are used for learning. The lower row shows images are created by $\boldsymbol{x}_{0.25j} = G^{0.25j} \boldsymbol{x}_0$. Discussion on the estimation of light direction is out of scope of this paper, but this example implies that the proposed method can be used for estimating illumination change.

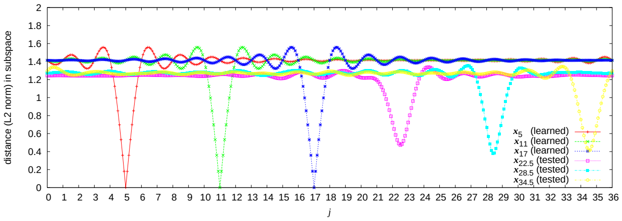


Figure 5. Euclidean distance in the subspace between $D^j \boldsymbol{x}_0$ and $\boldsymbol{x}_j$ (learned $j = 5, 11, 17$, and not learned $j = 22.5, 28.5, 34.5$). Horizontal axis is the power j of D^j.

3 Estimation of pose of an object in novel view

In this section, we propose two methods for estimation of pose of a new image by using the subspace described in the previous section.

3.1 Estimation by distance in the subspace D

As shown at the end of the last section, we have shown that extending real numbers of the power j of D^j gives images between learned samples.

Here we make an assumption that *a novel image $\boldsymbol{x}$ is matched with $G^j \boldsymbol{x}_0$ for some j and this also holds for images in the subspace: $\boldsymbol{x}$ is matched for some j with $D^j \boldsymbol{x}_0$ in the subspace*, where $\boldsymbol{x} \ = U_1 \boldsymbol{x}$ and denotes an image in the subspace. For matching, we use the Euclidean distance in the subspace:

$$j = \operatorname*{argmin}_{j \in [0,n[} ||\boldsymbol{x} \ - D^j \boldsymbol{x}_0||^2, \tag{14}$$

$$\theta = j\theta_1 = \frac{2\pi}{n} j. \tag{15}$$

See appendix for θ_1 and constructing D^j.

The estimation performs exhaustive search for j and it seems to be computationally expensive. However, we can use an effective algorithm for the search by using coarse-to-fine strategy. Fig.5 shows distances in the subspace by Eq.(15) for some real image sequence (see the later section for details). We can observe that the distances have sharp minima at corresponding j for learned images. Even for images not used for the learning, the distances have smooth minima around correct j. Based on this observation, first we search a minimum of j with a large step, then find around the minimum again with more smaller step, and gradually the interval of search shrinks. This strategy decreases computational cost and achieves estimation at any precision.

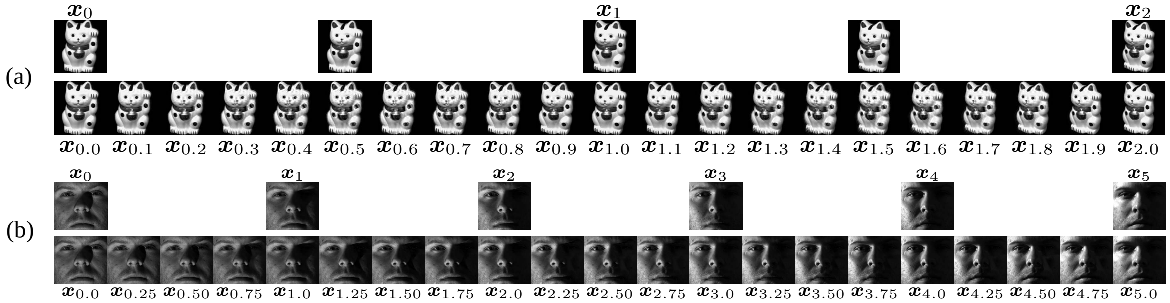

(a)
x_0
x_1
x_2
$x_{0.0}$ $x_{0.1}$ $x_{0.2}$ $x_{0.3}$ $x_{0.4}$ $x_{0.5}$ $x_{0.6}$ $x_{0.7}$ $x_{0.8}$ $x_{0.9}$ $x_{1.0}$ $x_{1.1}$ $x_{1.2}$ $x_{1.3}$ $x_{1.4}$ $x_{1.5}$ $x_{1.6}$ $x_{1.7}$ $x_{1.8}$ $x_{1.9}$ $x_{2.0}$
(b)
x_0
x_1
x_2
x_3
x_4
x_5
$x_{0.0}$ $x_{0.25}$ $x_{0.50}$ $x_{0.75}$ $x_{1.0}$ $x_{1.25}$ $x_{1.50}$ $x_{1.75}$ $x_{2.0}$ $x_{2.25}$ $x_{2.50}$ $x_{2.75}$ $x_{3.0}$ $x_{3.25}$ $x_{3.50}$ $x_{3.75}$ $x_{4.0}$ $x_{4.25}$ $x_{4.50}$ $x_{4.75}$ $x_{5.0}$

Figure 4. Images created by repeatedly multiplying a matrix G^j to the first image x_0. (a) images of off-the-plane rotation from COIL-20. $G^j = G^{0.1}$. (b) images of changing light direction from Yale Face Database B. $G^j = G^{0.25}$. Upper row shows learned images, and lower row shows created images between each learned images. *Note that supplemental full-length movies are attached/embedded in this PDF file (use Adobe Reader to see it).*

3.2 Estimation by angle of vectors in a 2D subspace A_1

The estimation method described above involves iterative search for minimum even if there is the efficient algorithm. Here we propose a direct estimation method without any searching. As mentioned before, an image is projected by U_1 onto many different 2-D subspaces in which a 2-D vector of two pixels is rotated by A_k. Now we focus on two pixels corresponding A_1 where the pair of pixels in two learned images next to each other, $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j+1}$, have the angle θ_1, the incremental step of the rotation.

So we propose to estimate a pose parameter for a novel image in the subspace $\boldsymbol{x}\ $ with $\boldsymbol{x}_0$ by using the angle subtended by two 2-D vectors, $\boldsymbol{x}\ , \boldsymbol{x}_0 \in \mathbb{R}^2$, corresponding to the 2-D subspace of A_1. To extract a 2-D vector $\boldsymbol{x}\ \in \mathbb{R}^2$ from $\boldsymbol{x}\ \in \mathbb{R}^n$ corresponding to A_1, just multiply the following $2 \times n$ matrix:

$$\boldsymbol{x}\ = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix} \boldsymbol{x}\ . \tag{16}$$

But $\boldsymbol{x}\ = U_1 \boldsymbol{x}$ is substituted above, the 2-D vector $\boldsymbol{x}\ \in \mathbb{R}^2$ is directly extracted from $\boldsymbol{x} \in \mathbb{R}^N$ by combining U_1 and the $2 \times n$ matrix:

$$\boldsymbol{x}\ = U_1 \boldsymbol{x}, \tag{17}$$

$$U_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix} U_1, \tag{18}$$

and $\boldsymbol{x}_0 \in \mathbb{R}^2$ is extracted:

$$\boldsymbol{x}_0 = A_1 U_1 \boldsymbol{x}_0. \tag{19}$$

Here U_1 is the 2-D subspace proposed in this paper for estimating the pose angle.

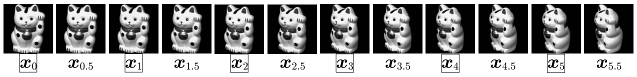


Figure 6. A part of images used for the experiments. $x_0, x_1, \ldots$ are learned (with box marks), $x_{0.5}, x_{1.5}, \ldots$ are tested images.

The angle θ subtented by the two 2-D vectors is calculated with $\cos\theta$ and $\sin\theta$. The innter product between $\boldsymbol{x}$ and $\boldsymbol{x}_0$ computes $\cos\theta$:

$$\cos\theta = \frac{\boldsymbol{x}_0{}^T \boldsymbol{x}}{||\boldsymbol{x}_0||\ ||\boldsymbol{x}\ ||}. \tag{20}$$

$\sin\theta$ is computed by cross product with two 3-D vectors extented with 0:

$$\boldsymbol{x}_0\ = (\boldsymbol{x}_0{}^T, 0)^T \in \mathbb{R}^3, \tag{21}$$

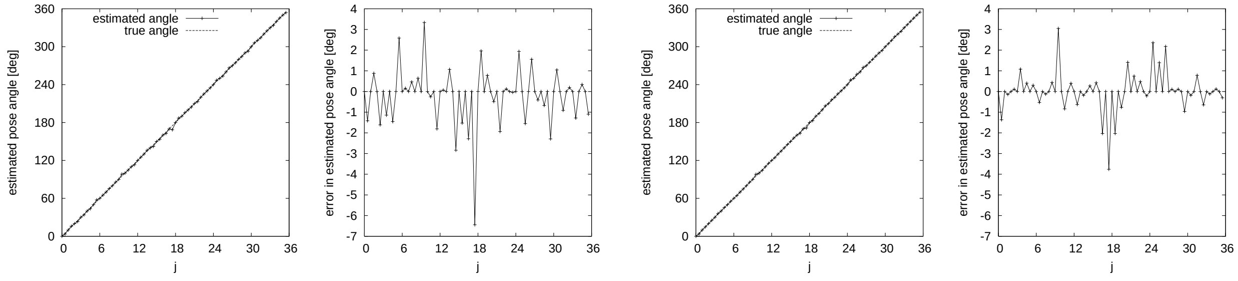
$$\boldsymbol{x}\ \ = (\boldsymbol{x}\ {}^T, 0)^T \in \mathbb{R}^3, \tag{22}$$

$$(0, 0, \sin\theta)^T = \frac{\boldsymbol{x}\ \ \times \boldsymbol{x}_0}{||\boldsymbol{x}_0||\ ||\boldsymbol{x}\ ||}. \tag{23}$$

Then, $\theta = \tan\ ^1 \left(\frac{\sin\theta}{\cos\theta}\right)$ is the angle between $\boldsymbol{x}\ $ and $\boldsymbol{x}_0$, then the estimate of the pose of the image $\boldsymbol{x}$.

4 Experimental results

We implemented the proposed method with Scilab-4.1 and evaluated with a real image sequence of the object 4 (the cat) from COIL-20[11]. The 72 images are $N = 128 \times 128$ in size, taken by rotating the object by 5 degrees each (see Fig.6). The rotation of the images is 1DOF (single axis rotation), but it is off-the-plane rotation because the axis is not the optical axis of the camera.



(a) (b) (c) (d)

Figure 7. Estimation results with (a)(b) distance-based and (c)(d) angle-based method for images x_j $(j = 0, 0.5, 1, 1.5, 2, \ldots, 35.5)$. (a)(c) Estimated pose.

Table 1. RMSEs with two methods for 20 objects in COIL-20 (in [deg]).

object No.	1	2	3	4	5	6	7	8	9	10
with distance	1.21	1.39	1.56	1.80	1.23	29.71	1.44	1.60	1.05	1.53
with angle	0.73	1.69	3.84	1.23	2.55	7.33	2.97	2.69	6.74	1.66

object No.	11	12	13	14	15	16	17	18	19	20	average
with distance	1.58	23.05	1.03	1.90	8.33	8.83	7.71	13.10	1.64	3.32	5.65
with angle	5.78	4.18	3.11	7.38	2.01	1.53	2.35	3.64	6.99	2.21	3.53

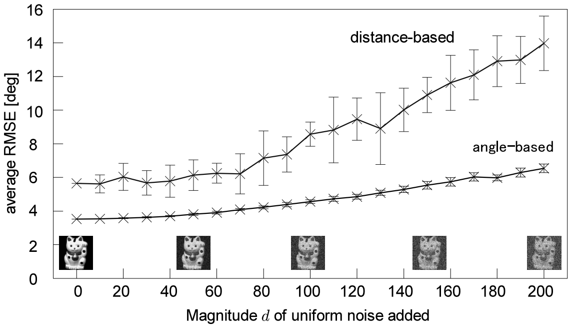


Figure 8. RMSEs of estimation with std. for 10 trials for noisy images. Horizontal axis is the magnitude $[-d, d]$ of uniform noise added. Vertical axis is average RMSE for 20 objects for only images not learned, but with noise.

For learning eigenspace and computing U_1, we used 36 images corresponding to 0, 10, 20, ... degrees as images $x_0, x_1, \ldots, x_{35}$ in the experiment. Therefore, $\theta_1 =$ 10[deg] in this experiment. Another 36 images corresponding to $5, 15, 25, \ldots$ degrees were used not for learning but for evaluation as images $x_{0.5}, x_{1.5}, \ldots, x_{35.5}$.

To illustrate properties of the subspace, we computed Euclidean distances between learned images x and the image x_0 rotated by the power of D in the subspace. Fig.5 shows the distances, and the horizontal axis is the power j of D^j, and the vertical axis is the Euclidean distance. For example, the distance with x_5 is $||x_5 - D^j x_0||^2$ and has a sharp minimum at $j = 5$ which means that the subspace is well learned. The distances with x_5 and the other learned images x_j, or equivalently $D^j x_0$, are all the same distance, $\sqrt{2}$. When the power j is a real number, the distance deviates from $\sqrt{2}$ and seems to be an interpolated curve comprised of sinusoids with different frequencies. The deviation from $\sqrt{2}$ (or ripple width) is so small that the search for minimum is not affected.

Fig.5 also shows distances with images not used for learning. Even if the images are not learned, the distance have smooth minimum around correct power. This means that the distance in the subspace is useful for the pose estimation.

Next, in Fig.7(a)(b) we show result of pose estimation with the method described in section 3.1, the search for minimum of j with the distance. Correct poses for the learned images x_j $(j = 0, 1, 2, \ldots)$ are estimated with no error. Poses for the images not learned x_j $(j = 0.5, 1.5, 2.5, \ldots)$ are also estimated well. The maximum error is about 7[deg], and almost less than ±2[deg], and RMSE (root mean squared error) for tested images only (not including learned images) is 1.80[deg]. Fig.7(c)(d) shows estimation result with the method described in section 3.2, the use of angle of two vectors in 2-D subspace. The maximum error is about 4[deg], and RMSE is 1.23[deg]. This means that the angle-based method is better than the distance-based method, and the angle of the two vectors in the 2-D subspace well represents the pose of the object in an image.

This is supported by estimation results shown in Tab.1 for all 20 objects in COIL-20 with both distance-based and angle-based methods. The result of Fig.7 is shown at object No. 4 in Tab.1. In average, RMSE of the angle-based method (3.53[deg]) is smaller than that of the distance-based (5.65[deg]).

Fig.8 shows the robustness of the angle-based method for noisy images shown. These images are contaminated by uniform noise up to ±200 without any intensity normalization (negative pixel values and large values are just used) where the range of pixel value in original images is between 0 and 255. Even when ±200 uniform noise is added, the average RMSE of angle-based method is less than 7[deg], while error of the distance-based method increases larger than 14 [deg]. This result demonstrates how robust the angle-based method is as well as the subspace proposed in this paper is useful for pose estimation. Note that for clutterd images (e.g., objects are occluded by a black rectangle), the proposed angle-based method has shown a good performance (not shown in this paper).

5 Conclusions

We have proposed a novel framework with cyclic group for appearance change in an image sequence of a rotating (1DOF but off-the-plane) object in 3-D. The proposed method constructs a subspace by block diagonalization of a matrix that represents cyclic group acting on the image seqeunce and transforms an image to another in the sequence. We have shown how the power of the block diagonal matrix produces the transform between images in and not in the sequence, then proposed two methods to estimate pose of a novel image; distance-based and angle-based. Experimental results with real image sequences demonstrated that the angle-based method is robust against noise and better than the distance-based method. The experiments are still limited, and comparisons with conventional methods are planned for the future.

Some limitations of the proposed method should be noticed. First, the method is applicable to sequences in which an object in images are revolutionary rotated: for example, a face sequence taken from left side to right side with frontal face has no images of the back of the head, so it is not applicable. Second, it seems to be difficult to extend the proposed method to handle with 3DOF rotation of an object. These are caused by the use of the matrix G as a cyclic group. Therefore, future works include to find an appropriate representation of relationship between such image sequences with group theory for extending application area of the proposed method. And also we have to investigate the pseudoinverse used in the derivation that is theoretically not determined uniquely because we assume $N > n$. It is clear that the linear map defined by the pseudoinverse is crucial to improve generalization and decrease estimation error for unknown pose.

Acknowledgments We thank Hiroshi Tamaru and Yumiko Ichihara at Hiroshima University for discussing the distance between images in the subspace. Also we would like to thank Hitoshi Sakano at NTT Data for his comments to improve the quality of the paper.

A Complex diagonalization of M

A $n \times n$ permutation matrix M to be diagonalized and its characteristic equation are[16, 3, 13, 7]:

$$M = \begin{pmatrix} 0 & & & & & 1 \\ 1 & 0 & & & & \\ & 1 & 0 & & & \\ & & & \ddots & \ddots & \\ & & & 1 & 0 & \\ & & & & 1 & 0 \end{pmatrix},$$

$$|M - \lambda I| = \begin{vmatrix} \lambda & & & & & -1 \\ -1 & \lambda & & & & \\ & -1 & \lambda & & & \\ & & & \ddots & \ddots & \\ & & & -1 & \lambda & \\ & & & & -1 & \lambda \end{vmatrix} = \lambda^n - 1,$$

so the eigenvalues λ are n different primitive n-th roots of unity ζ_n:

$$\lambda_k = \sqrt[n]{1} = \zeta_n^k = e^{\frac{2k\pi}{n}i}, \quad k = 0, 1, 2, \ldots, n-1,$$

where $i = \sqrt{-1}$. Let $\boldsymbol{w}_k = (w_1, w_2, \ldots, w_n)^T$ be the eigenvector corresponding ζ_n^k, then

$$M\boldsymbol{w}_k = \zeta_n^k \boldsymbol{w}_k$$
$$(w_n, w_1, w_2, \ldots, w_{n \quad 1})^T = (\zeta_n^k w_1, \zeta_n^k w_2, \ldots, \zeta_n^k w_n)^T.$$

Therefore, the eigenvector is

$$\boldsymbol{w}_k = (\zeta_n^{(n \quad 1)k}, \ldots, \zeta_n^{2k}, \zeta_n^k, 1)^T,$$

and M is diagonalized as $M = W\ D\ W^{\ H}$ with:

$$\begin{aligned} D\ &= \mathrm{diag}(1, \zeta_n, \zeta_n^2, \ldots, \zeta_n^{n \quad 1}), \\ W\ &= (\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{n \quad 1}), \end{aligned} \quad (24)$$

where ^{H} denotes complex conjugate and W is the basis of complex DFT (Discrete Fourier Transform) [13].

B Real block diagonalization of M

Next, block diagonalization of M is shown[16, 3, 13]. ζ_n^k and $\zeta_n^{n \quad k}$, eigenvalues of M, are complex conjugate to each other. To make corresponding complex conjugate eigenvectors $\boldsymbol{w}_k, \boldsymbol{w}_{n \quad k}$ real vectors, dividing them into real and imaginary parts:

$$\boldsymbol{w}_k = \frac{1}{\sqrt{2}}(\boldsymbol{c}_k + i\boldsymbol{s}_k), \quad \boldsymbol{w}_{n \quad k} = \frac{1}{\sqrt{2}}(\boldsymbol{c}_k - i\boldsymbol{s}_k).$$

Then, the multiplication of M with the vectors

$$M(\boldsymbol{w}_k, \boldsymbol{w}_{n\ \ k}) = (\boldsymbol{w}_k, \boldsymbol{w}_{n\ \ k}) \begin{pmatrix} \zeta_n^k & 0 \\ 0 & \zeta_n^{n\ \ k} \end{pmatrix},$$

is rewritten with $\zeta_n^k = \cos\theta_k + i\sin\theta_k$ as follows:

$$\begin{aligned} M(\boldsymbol{c}_k, \boldsymbol{s}_k) &= (\boldsymbol{c}_k, \boldsymbol{s}_k) \begin{pmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{pmatrix} \\ &= (\boldsymbol{c}_k, \boldsymbol{s}_k) A_k. \end{aligned}$$

Now M is diagonalized with block diagonal matrix D as $M = WDW^T$, where

$$D = \begin{array}{ll} \mathrm{diag}(1, A_1, A_2, \ldots, A_s), & n \text{ is odd,} \\ \mathrm{diag}(1, A_1, A_2, \ldots, A_s, -1), & n \text{ is even,} \end{array}$$

$$W = \begin{array}{ll} (\boldsymbol{w}_0, \boldsymbol{c}_1, \boldsymbol{s}_1, \boldsymbol{c}_2, \boldsymbol{s}_2, \ldots, \boldsymbol{c}_s, \boldsymbol{s}_s), & n \text{ is odd,} \\ (\boldsymbol{w}_0, \boldsymbol{c}_1, \boldsymbol{s}_1, \boldsymbol{c}_2, \boldsymbol{s}_2, \ldots, \boldsymbol{c}_s, \boldsymbol{s}_s, \boldsymbol{w}_{n/2}), & n \text{ is even,} \end{array}$$

$$s = \begin{array}{ll} \frac{n\ \ 1}{2}, & n \text{ is odd,} \\ \frac{n\ \ 2}{2}, & n \text{ is even,} \end{array}$$

where W is the basis of DFT[13]. Note that W and W are normalized so that norm of each column vector is 1.

C The power of D

If we need D^j, the angle in the 2×2 blocks A_k are multiplied:

$$\begin{aligned} D^j &= \begin{array}{ll} \mathrm{diag}(1, A_1^j, A_2^j, \ldots, A_s^j), & n \text{ is odd,} \\ \mathrm{diag}(1, A_1^j, A_2^j, \ldots, A_s^j, (-1)^j), & n \text{ is even,} \end{array} \\ A_k^j &= \begin{pmatrix} \cos j\theta_k & \sin j\theta_k \\ -\sin j\theta_k & \cos j\theta_k \end{pmatrix}. \end{aligned}$$

Note that D^j becomes a complex matrix when n is even.

This property is the most usefull one for the proposed formulation because G^j can be calculated by just multiplying the angle θ_k with j. If you use the Jordan (normal or canonical) form as block diagonalization of M, D^j is not easy to compute. And actually all eigenvectors of M are different to each other, the Jordan form of M is equivalent to the eigendecomposition Eq.(24); no Jordan form exists for M.

References

[1] T. Amano and T. Tamaki. A fast linear pose estimation method of 3D object using EbC image pair. *IEICE Trans.*, J90-D(8):2060–2069, 2007. (in Japanese).

[2] S. Ando, Y. Kusachi, A. Suzuki, and K. Arakawa. Appearance based pose estimation of 3D object using support vector regression. *ICIP2005*, 1:I–341–344, 2005. online.

[3] C.-Y. Chang, A. Maciejewski, and V. Balakrishnan. Fast eigenspace decomposition of correlated images. *IEEE Trans. IP*, 9(9):1937–1949, 2000. online.

[4] P. Chen and D. Suter. An analysis of linear subspace approaches for computer vision and pattern recognition. *Intl. J. of Computer Vision*, 68(1):83–106, 2006. online.

[5] F. De la Torre. Component analysis for computer vision. *ECCV2006 Tutorial*, 2006. online.

[6] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. PAMI*, 23(6):643–660, 2001. online.

[7] M. Jorgan, E. Žagar, and A. Leonardis. Karhunen-Loéve expansion of a set of rotated templates. *IEEE Trans. IP*, 12(7):817–825, 2003. online.

[8] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. PAMI*, 13(5):441–450, 1991. online.

[9] T. Melzer, M. Reiter, and H. Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 36:1961–1971, 2003. online.

[10] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Intl. J. of Computer Vision*, 14(1):5–24, 1995. online.

[11] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-20). Technical Report CUCS-005-96, Columbia University, 1996. online.

[12] T. Okatani and K. Deguchi. Yet another appearance-based method for pose estimation based on a linear model. *IAPR Workshop on Machine Vision Applications 2000*, pages 258–261, 2000.

[13] R.-H. Park. Comments on ”Optimal approximation of uniformly rotated images: Relationship between Karhunen-Loève expansion and discrete cosine transform”. *IEEE Trans. IP*, 11(3):332–334, 2002. online.

[14] G. Peters. Efficient pose estimation using view-based object representations. *Machine Vision and Applications*, 16(1):59–63, 2004. online.

[15] G. Peters and C. von der Malsburg. View reconstruction by linear combination of sample views. *British Machine Vision Conference*, 1:223–232, 2001. online.

[16] I. Satake. *Linear Algebra*. Marcel Dekker Inc., 1975.

[17] M. Sengel and H. Bischof. Efficient representation of in-plane rotation within a PCA framework. *Image and Vision Computing*, 23:1051–1059, 2005. online.

[18] T. Tangkuampien and D. Suter. 3D object pose inference via kernel principal component analysis with image euclidian distance (IMED). *British Machine Vision Conference*, 1:137–146, 2006. online.

[19] M. Uenohara and T. Kanade. Optimal approximation of uniformly rotated images: Relationship between Karhunen-Loeve expansion and discrete cosine transform. *IEEE Trans. IP*, 7(1):116–119, 1998. online.

[20] T. Vik, F. Heitz, and P. Charbonnier. Robust pose estimation and recognition using non-gaussian modeling of appearance subspaces. *IEEE Trans. PAMI*, 29(5):901–905, 2007. online.

[21] L.-W. Zhao, S.-W. Luo, and L.-Z. Liao. 3D object recognition and pose estimation using kernel PCA. *Proc. of Intl. Conf. Machine Learning and Cybernetics*, 5:3258–3262, 2004. online.

Scale-Based Principal Component Analysis of Point Cloud

Tomoya Sakai
Institute of Media and Information Technology
Chiba University
1-33, Yayoi, Inage, Chiba, Japan
tsakai@faculty.chiba-u.jp

Atsushi Imiya
Institute of Media and Information Technology
Chiba University
1-33, Yayoi, Inage, Chiba, Japan
imiya@faculty.chiba-u.jp

Abstract

Principal component analysis (PCA) evaluates geometric features of a point cloud for dimension reduction, pattern-recognition and classification. We develop the PCA of a point cloud on the basis of scale-space representation of its probability density function (PDF). First, we explain the geometric features of point cloud in scale space, and observe reduction of dimensionality with respect to the loss of information. Second, we introduce a hierarchical clustering of point cloud, and analyse the statistical significance of the clusters and their subspaces. Finally, we present a mathematical framework of scale-based PCA, which derives a statistically reasonable criterion for how to choose the number of components to retain, or how to reduce the dimensionality of point cloud.

1. Introduction

In this paper, we investigate the principal components in the scale space of point cloud[1]. In the book “Pattern Recognition” [1], Iijima introduced a framework of principal component analysis of generalised figures, which are images of an image in the linear scale space. He showed that the Hermite functions are base function system in the scale space of two-dimensional images. This analytical property of the base functions in the linear scale space is dimension-independent. In the book the global properties of the base functions in the scale space treat modal expression of images in the scale space. In the first one-thirds of book, linear scale space theory is dealt with from the view point of observation of two-dimensional images, since “Patten Recognition” of Iijima was established to introduce a mathematical framework of character recognition [2].

As a sequel of the clustering method using scale-scale analysis, we develop a local principal component analysis in Gaussian scale space. This analysis evaluates local dimensionalities and directionalities of clusters in a point cloud in a Euclidean space of arbitrary dimension [9, 10]. In [9], dimensionalities and directionalities of the clusters of a point cloud are extracted using the voting-based learning algorithm. In [2, 3, 4, 5, 6, 7, 8], they clarified that scale-space analyses clarify hierarchy among the clusters, and derived scale-based analyses and algorithms for the determination of the number of clusters in a point cloud. In this paper, we develop a framework to extract the clusters in point cloud and to evaluate their statistical significance or cluster validity. This treatment clarifies the dimensionalities, principal directions, and hierarchical relations of valid clusters in point cloud under the uncertainty of spatial resolution of observation. According to the scale-space theory, such uncertainty is axiomatically approximated by the Gaussian kernel.

In Section 2, we review and apply classical and modern scale-space analyses to the point cloud density. In Section 3, we introduce a hierarchical clustering of the point cloud. We also present a cluster validation scheme based on statistics. In Section 4, we derive stochastic moments and their estimators for the point cloud density in scale space. Then, we present the scale-based PCA for the hierarchical clusters. We discuss statistical significance of the principal components and structure of the point cloud.

2. Scale-Space Analysis of Point Cloud

2.1. Scale-Space Representation of Point Cloud Density

Let $P = \{\mathbf{p}|\ \mathbf{p} \sim f, \mathbf{p} \in \mathbb{R}^d\}$ be a point cloud in d-dimensional Euclidean space. Our problem is to extract *informative features* from the point cloud P. The spatial information is defined by the underlying probability density function (PDF) $f(\mathbf{x})$. The PDF describes the distribution of relative frequency with which the sample point would be

[1] In this paper, we call a set of points distributed in a space of arbitrary dimension a point cloud. A point cloud is a set of points in the three dimensional Euclidean space for shape modelling, and is a set of feature values in a feature space of arbitrary dimension for manifold learning.

[2] IIjima called the general theory of pattern recognition based on the theory of pattern.

obtained as an element of the point cloud by a finite number of repetition of observation. Unless the PDF is uniform, it provides meaningful difference of the frequency with respect to spatial position. The point cloud P with a finite cardinality illustrates the PDF at some level of geometric detail. Therefore, extraction of the informative features from P is essentially the estimation of $f(\mathbf{x})$ and its geometric features.

If we do not have prior belief on parametric form of $f(\mathbf{x})$, nonparametric approach is applicable to the PDF estimation. The nonparametric kernel estimate of $f(\mathbf{x})$ [11, 12] is

$$\tilde{f}(\mathbf{x},\sigma) = \frac{1}{\text{card}(P)} \mathop{}_{\mathbf{p}\in P} K(\mathbf{x}-\mathbf{p},\sigma) \tag{1}$$

where K is the normalised kernel function, and σ is called the bandwidth of the kernel. This estimated PDF $\tilde{f}(\mathbf{x},\sigma)$ in (1) is normalised so that

$$\int_{\mathbf{x}\in\mathbb{R}^d} \tilde{f}(\mathbf{x},\sigma) dV = 1 \tag{2}$$

The estimated PDF $\tilde{f}(\mathbf{x},\sigma)$ with a suitable kernel function converges to the true PDF $f(\mathbf{x})$ if $\sigma \to 0$ when the cardinality of P approaches to infinity.

The Gaussian kernel

$$K(\mathbf{x},\sigma) = G(\mathbf{x},\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}^d} e^{\frac{|\mathbf{x}|^2}{2\sigma^2}} \tag{3}$$

has been widely used for the PDF estimation [13]. In the PDF estimation using the Gaussian kernel, the bandwidth σ determines how much the point cloud is smoothed to produce the density estimate. Although there is a body of literature dealing with the problem on bandwidth selection in the kernel density estimation [13], analyses of $\tilde{f}(\mathbf{x},\sigma)$ in $(\mathbf{x},\sigma)$-space are of great help to clarify how the structure of the point cloud is simplified with increasing σ. The $(\mathbf{x},\sigma)$-space is called the Gaussian scale space, whose theory [14, 15, 16, 17] can be applicable to the PDF estimation as well as image analyses. In this paper, we regard $\tilde{f}(\mathbf{x},\sigma)$ with the Gaussian kernel as a scale-space representation of PDF estimated from the point cloud P. The scale parameter σ controls the level of estimated geometric detail. We enumerate some remarkable properties of the PDF estimation with an isotropic Gaussian kernel.

- $\tilde{f}(\mathbf{x},\sigma)$ satisfies the scale-space axioms [14, 16, 18], which include invariance under basic geometric transforms.

- Setting $\sigma = \sqrt{2\tau}$, $\tilde{f}(\mathbf{x},\sqrt{2\tau})$ satisfies the linear diffusion equation

$$\frac{\partial \tilde{f}}{\partial \tau} = \Delta \tilde{f}. \tag{4}$$

The initial function at $\tau = 0$ is the Delta mixture in Eq. (5), and a superposition of the Gaussian functions represents uncertainty of the location of the points after the time τ.

- $f(\mathbf{x},\sigma)$ converges to mixture of the Dirac delta function as $\sigma \to 0$.

$$\lim_{\sigma\to 0} \tilde{f}(\mathbf{x},\sigma) = \frac{1}{\text{card}(P)} \mathop{}_{\mathbf{p}\in P} \delta(\mathbf{x}-\mathbf{p}) \tag{5}$$

In other words, $f(\mathbf{x},0_+)$ acts as a lookup table because it returns ∞ only at the data points $\mathbf{x} = \mathbf{p}$ and otherwise 0.

- In the limit as $\sigma \to \infty$, the function $\tilde{f}(\mathbf{x},\sigma)$ converges to zero with the volume of one. Such PDF is said to be featureless providing null information.

- The number of modes of the homoscedastic Gaussian mixture seldom increases as the scale τ increases [22]. That is, mode creation is less expected if the Gaussian functions are unequally weighted. It is known that anisotropic Gaussian mixtures with different covariances yield spurious modes outside the convex hull of P.

- The scale parameter σ controls the information quantity measured by Shannon entropy [21], that is, the measure of the uncertainty monotonically increases with σ.

In pattern recognition, we regard the point cloud P as a set of feature vectors. Local geometric features of $f(\mathbf{x})$ correspond to specific features of data. For example, geometric moments characterise the distribution shape by a centroid, variance, distortion (asymmetry by skewness, peakedness by kurtosis, ..), and so on. In PCA, the PDF $f(\mathbf{x})$ is assumed to be a d-dimensional Gaussian function whose mean vector and covariance matrix define its ellipsoidal equiprobable level set. We will deal with the stochastic moments in Section 4. The derivatives of $f(\mathbf{x})$ also describe the differential geometric features. Local maximisers of $f(\mathbf{x})$, or the modes $\{\boldsymbol{\xi} | \nabla f(\boldsymbol{\xi}) = \mathbf{0}\}$, are the most expected and typical features within a group or a class of the features. We will discuss the behaviour of the modes in scale space, which give us an insight into dimension reduction with respect to the loss of information. The mode behaviour also leads to a natural method of data clustering known as the scale-based clustering or scale-space clustering [2, 3, 4, 5, 7]. Hereafter, we call $\tilde{f}(\mathbf{x},\sigma)$ the *generalised PDF* after the fashion of scale-space theory [20].

2.2. Behaviour of Modes in Scale Space

One of the primitive geometric features of the generalised PDF is the stationary point (a. k. a. critical point) where the spatial gradient vanishes.

$$\{(\boldsymbol{\xi}, \sigma) | \, \nabla \tilde{f}(\boldsymbol{\xi}, \sigma) = \mathbf{0}\} \tag{6}$$

The stationary point can be classified into $d + 1$ types based on the combination of signs of the eigenvalues λ_l of the Hessian matrix $\mathbf{H} = \nabla\nabla^\top f(\mathbf{x}, \sigma)$. We denote the signs of the eigenvalues as $(\pm, \pm, \ldots, \pm)$. For example, if $d = 2$ we have 3 types of stationary point: local maximum $(-, -)$, saddle $(+, -)$, and local minimum $(+, +)$. A local maximum $(-, -, \ldots, -)$ of a PDF is called the mode in probability theory and statistics.

Trajectory of Mode The position of the stationary point changes with respect to scale. The trajectory of the stationary point in the scale space is called the stationary curve (a. k. a. critical curve) in the scale-space theory. A stationary curve can be denoted by one-dimensional manifold $\boldsymbol{\xi}(\sigma)$ in scale space. Zhao and Iijima [19] have firstly showed that the stationary curve is a solution to the following system of differential equations.

$$\mathbf{H}\frac{d\boldsymbol{\xi}(\sqrt{2\tau})}{d\tau} = -\nabla\Delta\tilde{f}(\boldsymbol{\xi}, \sqrt{2\tau}) \tag{7}$$

The trajectory of a mode of $\tilde{f}(\mathbf{x}, \sigma)$ also satisfies Eq. (7). Every point $(\mathbf{x}, \sigma) = (\mathbf{p}, 0)$ $(\mathbf{p} \in P)$ is a starting point of the trajectory of mode. The trajectory of mode has a endpoint in scale space. In this paper, we denote the scale of the endpoint by σ^{t}.

Equiprobable Level Set The probability density at the mode, i.e., $\tilde{f}(\boldsymbol{\xi}, \sigma)$, must decrease with increasing σ since $\tilde{f}(\mathbf{x}, \sigma)$ obeys the diffusion equation (4) and Laplacian $\Delta\tilde{f} = \mathrm{trace}\,\mathbf{H} = \quad_l \lambda_l < 0$ at the mode. This indicates that equiprobable level sets are nested in the scale space. In image analysis, the nested level set associated with a local extremum is called the extremum stack [24, 25]. In the same manner, one can associate a mode $\boldsymbol{\xi}(\sigma_0)$ with an equiprobable level set in scale space whose probability density is equal to $\tilde{f}(\boldsymbol{\xi}, \sigma_0)$.

Flow of Probability Density As the scale σ increases, the probability density disperses in $\mathbb{R}^d$ space maintaining the normalised condition in Eq. (2). Since the diffusion equation (4) governs this process, the dispersing flow $\mathbf{F}$ of probability density can be defined as

$$\mathbf{F} = -\nabla\tilde{f}(\mathbf{x}, \sigma). \tag{8}$$

The local maxima, minima and saddles are sources, drains, and confluent points of the density flow with respect to scale [20, 26, 28].

In the scale-space analysis [20, 27, 28], topological structure among the stationary points can be analysed by the flow curves of the density flow. In two-dimensional case, a saddle $(+, -)$ has a pair of each inward and outward flow curves called *separatrices*. A separatrix of the inward flow curves connect between modes of $\tilde{f}(\mathbf{x}, \sigma)$, i.e., the sources of the flow.

In higher dimensional spaces, the separatrices are hypersurfaces which separate regions of different flow behaviour. We have $d - 1$ types of saddles in a d-dimensional space, e.g., $(+, -, -)$ and $(+, +, -)$ for $d = 3$. Let us denote the number of positive and negative eigenvalues of $\mathbf{H}$ at a stationary point by s_+ and $s\ \ $, respectively. Then, the space in the vicinity of the saddle can be decomposed into s_+-dimensional and $s\ \ $-dimensional subspaces S_+ and $S\ \ $ each of which is spanned by the corresponding eigenvectors. Since the saddle is a local minimum in the subspace S_+, the density flow in S_+ is in inward directions to the saddle. A similar statement holds in $S\ \ $. Therefore, S_+ and $S\ \ $ can be called the subspaces of attracting separatrix and repelling separatrix of a saddle, respectively.

We remark that the stationary points of $\tilde{f}(\mathbf{x}, \sigma)$ are representative points of geometric components, which can be symbolised as a graph of the flow-curve connections. In three-dimensional space, for example, local maxima $(-, -, -)$ correspond to vertices of the graph. Saddles $(+, -, -)$ and $(+, +, -)$ represent edges and faces of the graph, respectively. Local minima reside in volumes. See Fig. 1(a).

Structural Simplification If the scale σ is sufficiently small, the generalised PDF $\tilde{f}(\mathbf{x}, \sigma)$ consists of $\mathrm{card}(P)$ small blobs in an isotropic Gaussian shape. As σ increases, the blobs merge with each other into large ones, and the modes at their peaks disappear one after another. The topological structure formed by the flow-curve connections is simplified according to this degeneration of $\tilde{f}(\mathbf{x}, \sigma)$.

It is known from scale-space theory and catastrophe theory that the Fold catastrophe generically describes annihilation and creation events of two stationary points, which differ with respect to the signs of one eigenvalue of $\mathbf{H}$ that becomes zero at the point of events [29, 30]. Therefore, a mode of $\tilde{f}(\mathbf{x}, \sigma)$ with signs $(-, -, \ldots, -)$ is generically annihilated with a saddle with $(s_+, s\ \ ) = (1, d - 1)$. Similarly, two saddles with $(s_+, s\ \ )$ and $(s_+ + 1, s\ \ - 1)$ (or $(s_+ - 1, s\ \ + 1)$) meet and disappear at a point in the scale space. Saddles with $(s_+, s\ \ ) = (d - 1, 1)$ can be annihilated with local minima with $(s_+, s\ \ ) = (d, 0)$. We have introduced a point at infinity in a scale space as one of the local minima for topological consistency [31]. We have also shown from Eq. (7) that motion of the two points just before the annihilation is in the direction of the zero principal curvature [32].

As a consequence, every type of stationary points are involved in a sequence of the simplification of topological structure. Figure 1 illustrates an example of the structural simplification in a three dimensional space. Observe how the dimensionality is reduced from three to zero especially when the modes disappear. Even in higher dimensional spaces, one can find a subspace spanned only by the stationary points involved in the structural simplification, and observe a similar process of dimension reduction.

Convergence to Centroid If σ is sufficiently larger than the spatial size of the point cloud, the whole point cloud is regarded as a universal cluster represented by one remaining mode of the generalised PDF. This mode converges to the centroid of the point cloud according to the following proposition in [20, 23].

Proposition 1 *One remaining local maximum* $\boldsymbol{\xi}(\sigma)$ *of* $\tilde{u}(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) * u(\mathbf{x})$*, i.e., a convolution function with Gaussian, converges to the centroid of* $u(\mathbf{x})$ *if* $\sigma \to \infty$.

Proof

$$\begin{aligned} \nabla\tilde{u}(\mathbf{x}, \sigma) &= \nabla(G * u) = (\nabla G) * u \\ &= (\mathbf{x}G) * u - \mathbf{x}(G * u). \end{aligned}$$

Since $\nabla\tilde{u} = \mathbf{0}$ at the local maximum $\boldsymbol{\xi}(\sigma)$, we have

$$\boldsymbol{\xi}(\sigma) = \frac{(\mathbf{x}G) * u}{G * u} \to \frac{\mathbf{x} * u}{1 * u} \quad (\sigma \to \infty)$$

□

The generalised PDF $\tilde{f}(\mathbf{x}, \sigma)$ can be described as a Gaussian convolution of the delta mixture in Eq. (5). The centroid of a PDF is nothing more than the mean vector, i.e., the first moment.

Mode Hierarchy The generalised PDF $\tilde{f}(\mathbf{x}, \sigma)$ starts with card(P) Gaussian blobs. The merging process of the blobs with respect to scale hierarchically associates the modes of blobs with each other. Thus, the card(P) points are classified into hierarchical clusters. The hierarchy among modes is described as a tree, which is called the mode tree [33]. The mode tree also represents the hierarchy of the points or clusters in the point cloud. Note that not only the modes but also all of the other types of stationary points have hierarchical relationships among them. The mode tree is a subgraph of the scale-space tree for local maxima [32].

3. Hierarchical Clustering and Validation

3.1. Scale-Based Hierarchical Clustering

Clustering methods of data points using scale space have been proposed by many authors [2, 3, 4, 5, 6, 7, 8]. Most of them can be considered to be based on the mode hierarchy described in the previous section. We presented an algorithm of the construction of mode tree for hierarchical clustering [8].

The detected clusters, however, are invalid at small scales. The smaller the scale is, the more the modes of $\tilde{f}(\mathbf{x}, \sigma)$ are dependent on the positions of sample points. If the point cloud P does not have enough cardinality, the generalised PDF $\tilde{f}(\mathbf{x}, \sigma)$ cannot approximate at the small scales the true PDF $f(\mathbf{x})$ in detail. As the result, any estimate using the generalised PDF $\tilde{f}(\mathbf{x}, \sigma)$ with small scales is so random and experimentally less reproducible. We require a validation scheme to identify the clusters by modes with the statistically significant reproducibility.

In [3, 4, 5, 7], the detected clusters in scale space have been validated by several properties of the clusters: the number of clusters vs. scale, compactness, isolation, lifetime and birthtime. It is suggested that the decrease in the total number of clusters pauses at the number of valid clusters for a relatively long period of scale. Their methods of finding such pause, however, are heuristic.

3.2. Cluster Validation by Critical Scale

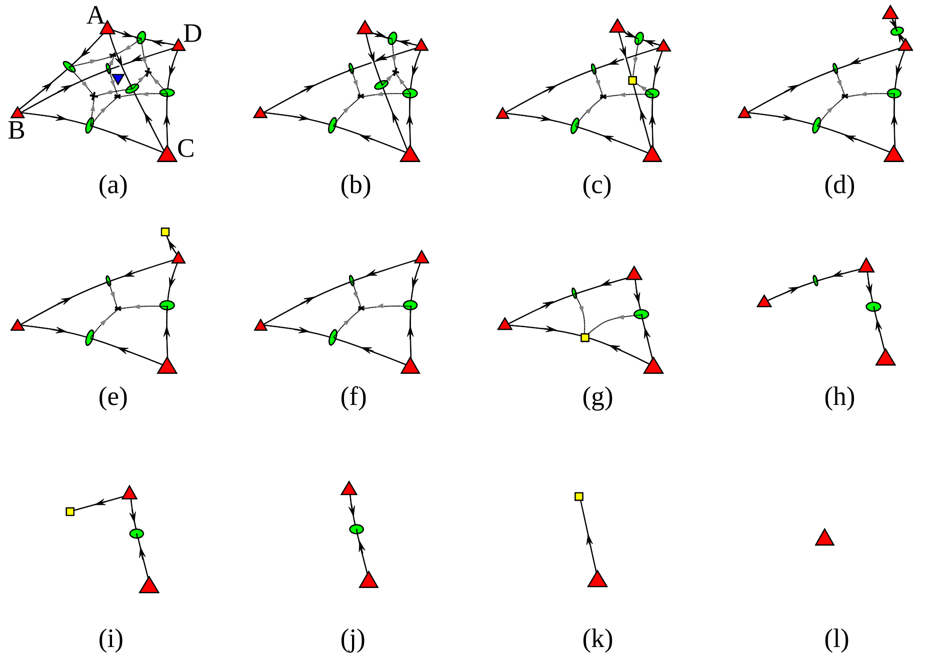
We have proposed a statistical criterion to identify the valid clusters by the *life* of mode. The life is defined as the terminating scale of the trajectory of mode in scale space, i.e., σ^{t}. We focused on the axiomatic fact that a set of uniformly distributed points does not contain the reproducible clusters. We showed that the uniformly distributed points present a Weibull-like unimodal distribution of the life. The valid cluster can be defined as a cluster with a statistically significant life out of this unimodal distribution. Consequently, the cluster validation can be established by the statistical rejection method using the unimodal life distribution. See [8] for more detail.

We call the critical value of scale used for the rejection the *critical scale*. Since any estimate using $\tilde{f}(\mathbf{x}, \sigma)$ with scales smaller than the critical scale is judged to be invalid, the critical scale is a threshold of spatial measure above which the given data is informative and under which any result of pattern analyses based on the PDF estimation looses statistical significance. In the scale-based clustering, the critical scale is a significance level of cluster validity. The critical scale also provides the statistical significance in the determination of dimensionalities of subspaces of clusters, which will be discussed in the next section.

We present an algorithm of recursive discovery of valid clusters using the mode tree.

ClusterDiscovery(mode tree T, cluster list L, critical value α)

```
1 let $\Sigma$ be the set of life values stored in $T$;
2 let $s$ be the subroot of $T$ with the second largest life-
    time $\sigma_2$;
3 if IsRejected($\sigma_2$, $\Sigma$, $\alpha$)
```



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 1. An example of the simplification of topological structure of generalised PDF. The up and down triangles indicate the local maximum (mode) and local minimum. The disc and cross indicate the saddles with signs $(+,-,-)$ and $(+,+,-)$, respectively. The solid lines are the flow-curve connections associated with the saddles with $(+,-,-)$. The dotted lines are the connections between the saddles with $(+,-,-)$ and $(+,+,-)$. The structure is simplified from (a) to (l). At a scale of annihilation, the annihilation point is indicated by the square.

```
    ClusterDiscovery(Subtree($T$, $s$), $L$);
    ClusterDiscovery($T\backslash$Subtree($T$, $s$), $L$);
else push $C$:=Leaves(Subtree($T$, $s$)) into $L$;
endif.
```

Here, **IsRejected** is the function that performs the rejection method and returns true if σ_2 is significantly large for the given critical value α. The function **Subtree** extracts the subtree with subroot s from the tree T. **Leaves** returns the leaves of the given tree.

4. Mathematical Framework of Scale-Based PCA

4.1. Stochastic Moment

In probability theory and statistics, structural features of data distribution are typically described by the *stochastic moments*. For a given PDF $f(\mathbf{x})$, the nth-order stochastic moment is given by

$$I_n : f \to \frac{1}{i^n}\,\nabla^n_{\mathbf{k}}\phi(\mathbf{k})|_{\mathbf{k}=\mathbf{0}}\,. \tag{9}$$

Here, $i=\sqrt{-1}$, and ϕ denotes the characteristic function defined as

$$\phi(\mathbf{k}) = \mathcal{F}[f(\mathbf{x})](\mathbf{k}) = \int_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})e^{i\mathbf{k}^\top\mathbf{x}}dV \tag{10}$$

which is similar to the well-known moment-generating function $M(\mathbf{t}) = \phi(-i\mathbf{t})$. For example, I_1 and I_2 are respectively mappings from a PDF $f(\mathbf{x})$ to the mean vector and the moment matrix with respect to the origin.

$$I_1 : f \to \int_{\mathbf{x}\in\mathbb{R}^d} \mathbf{x}f(\mathbf{x})dV = E[\mathbf{x}] = \boldsymbol{\mu} \tag{11}$$

$$I_2 : f \to \int_{\mathbf{x}\in\mathbb{R}^d} \mathbf{x}\mathbf{x}^\top f(\mathbf{x})dV = E[\mathbf{x}\mathbf{x}^\top] = \mathbf{M} \tag{12}$$

4.2. Central Moment

The stochastic moment can be converted to a moment about a specific point. We modify I_n to define the stochastic moment with respect to $\mathbf{a}\in\mathbb{R}^d$ as follows.

$$I_n(\mathbf{a}) : f \to \frac{1}{i^n}\,\nabla^n_{\mathbf{k}}\phi(\mathbf{k})e^{\;i\mathbf{k}^\top\mathbf{a}}\Big|_{\mathbf{k}=\mathbf{0}} \tag{13}$$

Clearly, $I_n(\mathbf{0})$ is equivalent to I_n. $I_n(\mathbf{a})$ defines the translation of the stochastic moment. The characteristic function can be locally expanded in power series as

$$\begin{aligned}\phi(\mathbf{k})e^{\quad i\mathbf{k}^\top\mathbf{a}} &= \int_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})e^{i\mathbf{k}^\top(\mathbf{x}\quad\mathbf{a})}dV \\ &= \int_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \quad {}_{m=0}^{\infty} \frac{1}{m!}\{i\mathbf{k}^\top(\mathbf{x}-\mathbf{a})\}^m dV \\ &= \int_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})dV + i\int_{\mathbf{x}\in\mathbb{R}^d} \mathbf{k}^\top(\mathbf{x}-\mathbf{a})f(\mathbf{x})dV \\ &\quad + \frac{1}{2}i^2\int_{\mathbf{x}\in\mathbb{R}^d}(\mathbf{x}-\mathbf{a})^\top\mathbf{k}\mathbf{k}^\top(\mathbf{x}-\mathbf{a})f(\mathbf{x})dV \\ &\quad + \cdots. \end{aligned} \tag{14}$$

As a special case, letting $\mathbf{a} = \boldsymbol{\mu}$, we obtain the so-called *central moments*. For instance, $I_1(\boldsymbol{\mu})$ always maps f to zero vector, and $I_2(\boldsymbol{\mu})$ yields the covariance matrix.

$$\begin{aligned} I_2(\boldsymbol{\mu}) : f \to &\int_{\mathbf{x}\in\mathbb{R}^d}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top f(\mathbf{x})dV \\ &= E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top] = \boldsymbol{\Sigma} \end{aligned}$$

Therefore, it is possible to regard $I_n(\boldsymbol{\mu})$ as the *central moment generator*. The central moments correspond to the Taylor expansion coefficients of Fourier transform of characteristic function about the mean $\boldsymbol{\mu}$. The PDF $f(\mathbf{x})$ can be reconstructed from inverse Fourier transform of its characteristic function if all of the stochastic moments are finite and the series in Eq. (14) converges absolutely near $\mathbf{x} = \mathbf{a}$.

4.3. Moment Estimation

The central moments can be derived from any PDF model. If a single d-dimensional Gaussian distribution is assumed as the PDF model for a point cloud, i.e., $\mathbf{p} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then one can obtain the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ by $I_1(\mathbf{0})$ and $I_2(\boldsymbol{\mu})$, respectively. One should note that odd-order central moment generators for the Gaussian distribution provide zeros due to the symmetry of Gaussian function, and any even-order central moment can be expressed as the second-order central moment $\boldsymbol{\Sigma}$. The central moment generator $I_n(\boldsymbol{\mu})$ for $n \geq 3$ therefore provides no additional information about the distribution. In fact, PCA can only deal with a linear subspace under the assumption of multidimensional Gaussian distribution. PCA estimates the major axes of ellipsoidal equiprobable level set, which is determined by $\boldsymbol{\Sigma}$. Usually, the maximum-likelihood estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ written as

$$\boldsymbol{\mu}^{\mathrm{ML}} = \frac{1}{\mathrm{card}(P)} \quad {}_{\mathbf{p}\in P} \quad \mathbf{p} \tag{15}$$

$$\boldsymbol{\Sigma}^{\mathrm{ML}} = \frac{1}{\mathrm{card}(P)} \quad {}_{\mathbf{p}\in P} \quad (\mathbf{p}-\boldsymbol{\mu}^{\mathrm{ML}})(\mathbf{p}-\boldsymbol{\mu}^{\mathrm{ML}})^\top \tag{16}$$

are used in PCA. They are called sample moments.

Henceforth, we derive the central moments from the generalised PDF $\tilde{f}(\mathbf{x},\sigma)$. We define the characteristic function with scale σ as

$$\begin{aligned}\tilde{\phi}(\mathbf{k},\sigma) &= \mathcal{F}[\tilde{f}(\mathbf{x},\sigma)] \\ &= \frac{1}{\mathrm{card}(P)} \quad {}_{\mathbf{p}\in P} \quad e^{\quad \frac{|\mathbf{k}|^2}{2\frac{1}{\sigma^2}}} e^{i\mathbf{k}^\top\mathbf{p}}. \end{aligned}\tag{17}$$

The mean vector and central moment are then

$$I_1(\mathbf{0}) : \tilde{f} \to \frac{1}{\mathrm{card}(P)} \quad {}_{\mathbf{p}\in P} \quad \mathbf{p} = \tilde{\boldsymbol{\mu}} \tag{18}$$

$$I_n(\tilde{\boldsymbol{\mu}}) : \tilde{f} \to \frac{-i^n}{\mathrm{card}(P)}\nabla^n_{\mathbf{k}} e^{\quad\frac{|\mathbf{k}|^2}{2\frac{1}{\sigma^2}}} \quad {}_{\mathbf{p}\in P} \quad e^{i\mathbf{k}^\top(\mathbf{p}\quad\tilde{\boldsymbol{\mu}})}. \tag{19}$$

The mean vector $\tilde{\boldsymbol{\mu}}$ of generalised PDF coincides with $\boldsymbol{\mu}^{\mathrm{ML}}$. While $\tilde{\boldsymbol{\mu}}$ is independent of the scale σ, $I_n(\tilde{\boldsymbol{\mu}})$ generates the central moment as a function of σ. The covariance matrix $\tilde{\boldsymbol{\Sigma}}(\sigma)$ of generalised PDF is given by $I_2(\tilde{\boldsymbol{\mu}})$ as

$$\begin{aligned} I_2(\tilde{\boldsymbol{\mu}}) : \tilde{f} \to &\frac{1}{\mathrm{card}(P)} \quad {}_{\mathbf{p}\in P} \quad (\mathbf{p}-\tilde{\boldsymbol{\mu}})(\mathbf{p}-\tilde{\boldsymbol{\mu}})^\top + \sigma^2\mathbf{I} \\ &= \boldsymbol{\Sigma}^{\mathrm{ML}} + \sigma^2\mathbf{I} = \tilde{\boldsymbol{\Sigma}}(\sigma) \end{aligned}\tag{20}$$

where $\mathbf{I}$ denotes the identity matrix. The scale σ increases diagonal dominance by σ^2. Accordingly, every eigenvalue of the matrix $\tilde{\boldsymbol{\Sigma}}(\sigma)$ is incremented by σ^2 while the eigenvectors of $\tilde{\boldsymbol{\Sigma}}(\sigma)$ are equal to those of $\boldsymbol{\Sigma}^{\mathrm{ML}}$.

The matrix $\tilde{\boldsymbol{\Sigma}}(0)$ coincides with $\boldsymbol{\Sigma}^{\mathrm{ML}}$. This property might seem confusing because the generalised PDF with $\sigma = 0$ is not Gaussian but delta mixture. This is caused by the difference of PDF model. Unlike the single Gaussian distribution assumed in PCA, the generalised PDF $\tilde{f}(\mathbf{x},\sigma)$ approximates any distribution at a scale above a critical scale. If we can select a suitable scale σ, the scale-based central moments we have derived can quantify the structural feature of distribution such as the principal directions, asymmetry, peakedness, and so on.

4.4. Scale-Based PCA

We apply the moment estimation for each valid cluster discovered in the point cloud P by the algorithm **Cluster-Discovery**. According to the mode tree, the generalised PDF $\tilde{f}(\mathbf{x},\sigma)$ can be hierarchically decomposed into the PDFs for the valid clusters.

$$\tilde{f}(\mathbf{x},\sigma) = \quad {}_{c=1}^{C} \quad \tilde{f}_c(\mathbf{x},\sigma) = \quad {}_{c=1}^{C} \quad {}_{\mathbf{p}\in P_c} \quad G(\mathbf{x}-\mathbf{p},\sigma) \tag{21}$$

Here, P_c ($c = 1,\ldots,C$) are the valid clusters corresponding to the subtrees in mode tree, and $P = \cup_c P_c$. We remark that the scale σ in Eq. (21) is just a common parameter controlling the whole scale of the PDF. There exists a suitable scale σ_c for each $\tilde{f}_c$ to describe the distribution of the c-th cluster P_c. At least, such σ_c should be greater than the critical scale so as for the cluster P_c to be valid, and less than the life σ^{t} so as for P_c to be separate from the others.

Let $\tilde{\boldsymbol{\mu}}_c$ and $\tilde{\boldsymbol{\Sigma}}_c(\sigma_c)$ denote the mean vector and covariance matrix of P_c calculated by $I_1(\mathbf{0})$ in Eq. (18) and $I_2(\tilde{\boldsymbol{\mu}}_c)$ in Eq. (20) using $\tilde{f}_c(\mathbf{x}, \sigma_c)$, respectively. Then, we can employ PCA for the cluster P_c by eigendecomposition of $\tilde{\boldsymbol{\Sigma}}_c(\sigma_c)$. Since the scale-based covariance matrix is written as

$$\tilde{\boldsymbol{\Sigma}}_c(\sigma_c) = \boldsymbol{\Sigma}_c^{\mathrm{ML}} + \sigma_c^2 \mathbf{I}, \tag{22}$$

the eigenvectors of $\tilde{\boldsymbol{\Sigma}}_c(\sigma_c)$ is same as those of $\boldsymbol{\Sigma}_c^{\mathrm{ML}}$. The eigenvalues of $\tilde{\boldsymbol{\Sigma}}_c(\sigma_c)$ are greater than or equal to σ_c^2. This contribution of the scale parameter to the eigenvalues is quite simple, but it suggests a very remarkable consequence that the eigenvalues of $\boldsymbol{\Sigma}_c^{\mathrm{ML}}$ less than the square of critical scale are buried under the scale contribution σ_c^2. Such small eigenvalues are neither *principal* nor statistically significant.

A major problem in PCA is how to choose the number of principal components to retain. This problem is essentially the same as what the dimensionality of the subspace of the data is. The suggestion by Eq. (22) provides us with a statistically reasonable criterion: if the cluster is discovered by the rejection method with a critical scale σ , choose the eigenvalues of $\boldsymbol{\Sigma}_c^{\mathrm{ML}}$ which are greater than σ^2.

5. Summary and Discussion

We have attempted to make a first step towards a mathematical framework of PCA on the basis of the scale-space theory on point cloud density and statistical principles of cluster discovery. The main novelties of this work are:

- Observation of topological structure of point cloud in scale space and the reduction of dimensionality with respect to scale.
- Explanation for scale-based clustering and cluster validity in terms of statistical significance.
- Derivation of stochastic moment generator from a PDF estimated in scale space.
- Scale-based PCA and a criterion for choosing principal components.

We extended a scale-space theory to the kernel density estimation of the point cloud in a Euclidean space of arbitrary dimension. The topological structure of the point cloud density is naturally determined by the flow of probability density with respect to scale. Since the bandwidth of the Gaussian kernel, i.e., scale, controls the information quantity, we can observe structural simplification of the estimated point cloud density with respect to the loss of information.

We explained the validity of clusters discovered by the scale-based hierarchical clustering. The statistical significance of reproducibility of the cluster discovery is guaranteed by the rejection method using the life of mode in scale space. The critical value of scale in the rejection method is the so-called critical scale, which discriminates between valid and invalid clusters.

The scale-based PCA is mathematically derived from the first- and second-order stochastic moments calculated by the Gaussian kernel density estimate of point cloud. The scale-based PCA can be applied to individual valid clusters. Although the calculated second moment in Eq. (22) is very simple, this result suggests an important fact. If one switches the PDF model from parametric to nonparametric, the bandwidth of the kernel contributes to the significance of eigenvalues. Consequently, the eigenvalues of principal components must be greater than the square of critical scale. Otherwise, the cluster is not guaranteed its validity in the subspace spanned by eigenvectors with the small eigenvalues.

Determining the number of principal components is an essential problem in PCA, and it has been treated extensively in the literature prior to subspace methods. We refer to [34] for a comprehensive overview. For this problem, we would like to note that the term $\sigma_c^2 \mathbf{I}$ in Eq. (22) induces the so-called sphericity of the cluster. The sphericity is the degree of how spherical the distribution is. Generically, the covariance matrix of a point cloud or its cluster does not have a purely zero eigenvalue. That is, the estimated density distribution is not confined in a subspace of $\mathbb{R}^d$ but it is d-dimensional elliptic in a strict sense. Our scale-based PCA implies that the cluster requires some sphericity to be identified as a valid cluster, and the sphericity need to be greater than the critical scale in radius. From this point of view, only the principal components with the eigenvalue greater than the critical scale might reflect a valid contribution to the similarity measure with subspace in classification methods.

References

[1] T. Iijima: *Pattern Recognition*. Corona: Tokyo (in Japanese) 1976.

[2] S. V. Chakravarthy and J. Ghosh: Scale-Based Clustering Using the Radial Basis Function Network. *IEEE Trans. on Neural Networks*, vol. 7, no. 5, pp. 1250–1261, 1996.

[3] S. J. Roberts: Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, vol. 30, no. 2, pp. 261–272, 1997.

[4] E. Nakamura and N. Kehtarnavaz: Determining number of clusters and prototype locations via multiscale clustering. *Pattern Recognition Letters*, vol. 19, no. 14, pp. 1265–1283, 1998.

[5] A. Hinneburg and D.A. Keim: An efficient approach to clustering in large multimedia databases with noise. *Proc. 4th International Conference on Knowledge Discovery and Data Mining*, pp. 58–65, 1998.

[6] R. Kothari and D. Pitts: On finding the number of clusters. *Pattern Recognition Letters*, vol. 20, pp. 405–416, 1999.

[7] Y. Leung, J.-S. Zhang, and Z.-B. Xu: Clustering by scale-space filtering. *IEEE Trans. PAMI*, vol. 22, no. 12, pp. 1396–1410, 2000.

[8] T. Sakai, A. Imiya, T. Komazaki, and S. Hama: Critical scale for unsupervised cluster discovery. *Lecture Notes in Artificial Intelligence*, vol. 4571, pp. 218–232, 2007.

[9] A. Imiya, H. Ootani, and K. Kawamoto: Linear manifolds analysis: theory and algorithm. *Neurocomputing*, vol. 57, pp. 171–187, 2004.

[10] A. Imiya and K. Kawamoto: Learning dimensionality and orientations of 3D objects. *Pattern Recognition Letters*, vol. 22, pp. 75–83, 2001.

[11] M. Rosenblatt: Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[12] E. Parzen: On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[13] A. J. Izenman: Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, vol. 86, no. 413, pp. 205–224, 1991.

[14] A.P. Witkin: Scale space filtering. *Proc. of 8th IJCAI*, pp. 1019–1022, 1983.

[15] J.J. Koenderink: The structure of images. *Biological Cybernetics*, vol. 50, pp. 363–370, 1984.

[16] J. Weickert, S. Ishikawa, and A. Imiya: Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*, vol. 10, pp. 237–252, 1999.

[17] T. Lindeberg: *Scale-Space Theory in Computer Vision*, Kluwer, Boston 1994.

[18] R. Duits, L. Florack, J. Graaf, and B. ter Haar Romeny: On the axioms of scale space theory. *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 267–298, 2004.

[19] N.-Y. Zhao and T. Iijima: Theory on the method of determination of view-point and field of vision during observation and measurement of figure. *IEICE Japan, Trans. D.*, vol. J68-D, pp. 508–514, 1985 (in Japanese).

[20] N.-Y. Zhao and T. Iijima: A theory of feature extraction by the tree of stable view-points. *IEICE Japan, Trans. D.*, vol. J68-D, pp. 1125–1135, 1985 (in Japanese).

[21] J. Sporring and J. Weickert: Information measures in scale-spaces. *IEEE Trans. Information Theory*, vol. 45, pp. 1051–1058, 1999.

[22] M.A. Carreira-Perpinan and Christopher K.I. Williams: On the number of modes of a Gaussian mixture. *Lecture Notes in Computer Science*, vol. 2695, pp. 625–640, 2003.

[23] M. Loog, J. J. Duistermaat, and L. M. J. Florack: On the behavior of spatial critical points under Gaussian blurring. *Lecture Notes in Computer Science*, vol. 2106, pp. 183–192, 2001.

[24] L. M. Lifshitz and S. M. Pizer: A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Trans. PAMI*, vol. 12, no. 6, pp. 529–540, 1990.

[25] A. Simmons, S. R. Arridge, P. S. Tofts, and G. J. Barker: Application of the extremum stack to neurological MRI. *IEEE Trans. MI*, vol. 17, no. 3, pp. 371–382, 1998.

[26] M. Lindenbaum, M. Fischer, and A. Bruckstein: On Gabor's contribution to image enhancement. *Pattern Recognition*, vol. 27, no. 1, pp. 1–8, 1994.

[27] L.D. Griffin and A. Colchester: Superficial and deep structure in linear diffusion scale space: Isophotes, critical points and separatrices. *Image and Vision Computing*, vol. 13, no. 7, pp. 543–557, 1995.

[28] T. Sakai and A. Imiya: Figure field analysis of linear scale-space image. *Lecture Notes in Computer Science*, vol. 3459, pp. 374–385, 2005.

[29] A. Kuijper, L.M.J. Florack, and M.A. Viergever: Scale space hierarchy. *Journal of Mathematical Imaging and Vision*, vol. 18, no. 2, pp. 169–189, 2003.

[30] A. Kuijper and L.M.J. Florack: Using catastrophe theory to derive trees from images. *Journal of Mathematical Imaging and Vision*, vol. 23, no. 3, pp. 219–238, 2005.

[31] T. Sakai and A. Imiya: Qualitative descriptions of image in the Gaussian scale-space. *Interdisciplinary Information Sciences*, vol. 11, no. 2, pp. 157–166, 2005.

[32] T. Sakai and A. Imiya: Scale-space hierarchy of singularities. *Lecture Notes in Computer Science*, vol. 3753, pp. 181–192, 2005.

[33] M. C. Minnotte and D. W. Scott: The mode tree: A tool for visualization of nonparametric density features. *Journal of Computational and Graphical Statistics*, vol. 2, no. 1, pp. 51–68, 1993.

[34] D. A. Jackson: Stopping rules in principal components analysis: A comparison of heuristical and statistical approaches. *Ecology*, vol. 74, no. 8, pp. 2204–2214, 1993.

Kernel Eigenspace for Detecting and Separating Multi-class Non-linear Objects

Masud Rahman, Anand Santhanam
Department of Information Enginering
The Australian National University, Ausrtalia
masudbitk@gmail.com,anand@rsise.anu.edu.au

Seiji Ishikawa
Department of Control Engineering
Kyushu Institute of Technology
Sensuicho, Tobata, Kiatakyushu, Japan

Abstract

Kernel eigenspace method is introduced in this paper for representing, classifying and detecting multi-class objects with non-linear characteristics. The main focus of this paper is to detect and classify moving vehicle from its viewpoint image using kernel eigenspace. The kernel eigenspace extracts non-linear features of multi-class moving vehicles by mapping input space to a higher dimensional feature space choosing an appropriate radial basis function. The obtained results provide us a clustered feature space of the car and non-car for classifying them by separating the dimensional space or comparing the eigenvectors called eigendimension matching. Eigendimension matching conforms whether an image feature is in-space or out-space comparing the dimensional ranges. The experimental results show the robustness of the feature separation using kernel eigenspace in our car database that lead to the cars' detection and classification from learning only the viewpoints. Extended experiments with various generic databases shows the remarkable performance of clustering, detecting and separating vehicle image using the the proposed method.

1 Introduction and Motivation

In practice, a vision researcher has no choice to detect or not to detect any particular objects in a real time environment. It is considered as a failure of a machine (ultimately to a researcher) when you can not detect any observation. Therefore, detection and classification in an unconstrained environment is always a challenging problem. In the past years, many fruitful methods have been developed for the object detection [6] and classification. In general, object detection can mainly be done by two ways: part based [1, 2, 14] and shape based object recognition [11]. In the part based approaches, an object structure is encoded by using a set of patches covering important parts of the object. These patches themselves are detected using interest point operators. In affine invariant approaches for object recognition small patches are extracted from the image which are characterized by view point invariant descriptors [2]. These descriptors are used to match the object. Shape or appearance based methods [3, 8–10] use a global approach for capturing the object structure. Eigenspace (also well-known as PCA [5]) is one of the powerful technique for extracting global structure from a high dimensional data set. It has become well-known to the vision communities after its successful application for extracting facial features in the Eigenfaces method [15, 16]. However, eigenspace is only powerful for linear feature extraction and, therefore, it is not suitable for non-linear feature selection. Kernel eigenspace (well-known as kernel PCA), on the other hand, was introduced as a nonlinear extension of eigenspace in [14], which computes the principal components [5] in a high dimensional feature space which is nonlinearly related to the input space. Our interest is on non-linear data and, therefore, we focus on using kernel eigenspace for classifying our non-linear datasets. It has also previously been introduced in some literatures as a KPCA. Yang [18] and Moghaddam [8] compared the face recognition performance and Eigenfaces method by using Kernel PCA with the cubic polynomial kernel and Gaussian kernel, respectively. In addition, Kernel eigenspace is also used to model the variability in classes of 3D-shapes [13, 17]. Liu [7] has recently employed it for recognition of facial expression using Gabor filters. Features derived by Gabor filters were nonlinearly projected onto higher dimensional feature space by employing fractional power polynomial as a kernel function. For the classification, traditional classifiers employed for classifying the object features include Bayes classifier, SVM, Discriminant functions, etc. The traditional classifiers only work with linearly separable datasets. In order to avoid this limitation, similarity measures using L1 or L2 norm are employed for classifying the features and/or objects. However, our focus is to deal with highly nonlinear datasets that produce non-convex feature space.

This study will focus to separate and cluster the feature spaces with appropriate designing the radial basis function. Once the features are separated with respect to the datasets, we then define the maximum and minimum ranges of eigendimensions or we group the feature space to classify the object classes such as car and non-car images. If there are more classes in the space, it clusters appropriately and then group them using the eigendimension ranges. It is worthwhile to mention that the present study employs various car's viewpoint images for training the system. We develop the feature space only using viewpoint images (negative and positive samples) and then classify the respective feature spaces matching the eigendimension. Therefore, the proposed method does not require to employ any traditional classifiers. The classifications do not depend only on two-class problems as proposed in [2] but it can successfully classify the multi-class problems as well.

2 Kernel eigenspace for extracting non-linear feature space

Let assume that our data mapped into a higher dimensional feature space, $\phi(x_1), \ldots, \phi(x_l)$, is centered, i.e., $\sum_{k=1}^{l} \phi(\mathbf{x}_k) = 0$. In this circumstance, the covariance matrix is

$$\overline{C} = \frac{1}{l} \sum_{j=1}^{l} \phi(\mathbf{x}_j)\phi(\mathbf{x}_j)^T. \quad (1)$$

The eigenvalue equation now becomes $\lambda\mathbf{V} = \overline{C}\mathbf{V}$. The eigenvalues ($\lambda \geq 0$) and eigenvectors ($\mathbf{V}\epsilon\mathbf{F}$) need to be computed satisfying the eigen equation. Instead of explicitly computing the nonlinear map ϕ, we achieve the same goal by employing the kernel function $k(x_i, x_j) = (\phi(x_i), \phi(x_j))$, which computes the dot product of vectors x_i and x_j in the higher dimensional space. As all solutions $\mathbf{V}$ lie in the span of $\phi(x_1), \ldots, \phi(x_l)$, we consider the equivalent system substituting the Eq. 1 onto the basic eigen equation of λ**V**=**CV**

$$\lambda(\phi(\mathbf{x}_k).\mathbf{V}) = (\phi(\mathbf{x}_k).\overline{C}\mathbf{V}) \quad (2)$$

for all $k = 1, \ldots, l$, and their existence co-efficient $\alpha_1, \ldots, \alpha_l$ such that

$$\mathbf{V} = \sum_{i=1}^{l} \alpha_i \phi(\mathbf{x}_i). \quad (3)$$

Substituting Eq. 1 and Eq. 3 into Eq. 2 and defining a $l \times l$ gram matrix $\mathbf{K} := (\phi(x_i).\phi(x_j))$, we arrive at this eigenvalue problem for solving non-zero eigenvalues. A detail calculation can be found in [14].

$$l\mathbf{K}\alpha = \mathbf{K}^2\alpha \quad (4)$$

For principal component extraction, we compute projection of the image of a test sample $\phi(\mathbf{x})$ onto the eigenvectors $\mathbf{V}^k$ in **F** according to

$$(\mathbf{V}^k.\phi(\mathbf{x})) = \sum_{i=1}^{l} \alpha_i^k (\phi(\mathbf{x}_i).\phi(\mathbf{x})). \quad (5)$$

The kernel functions can also be considered of as functions measuring likelihood between instances. If the two samples are similar, the kernel will be greater. If the samples, on the other hand, are dissimilar the kernel value falls to zero [2]. We use the following gaussian kernel for this measurement:

$$k(x, y) = exp(-||x - y||^2)/(2\sigma^2) \quad (6)$$

where the radial basis function $2\sigma^2$ or $2c$. Some other kernels are also widely used such as polynomials ($k(x, y) = (x.y)^d$) and sigmoid ($k(x, y) = tanh(\kappa(x.y) + \Theta)$).

Finally, for extracting the features of kernel eigenspace, let **x** be a test example whose map in the higher dimensional feature space is $\phi(\mathbf{x})$. The features of the kernel eigenspace for this test image can be derived by

$$f = \mathbf{V}^T\phi(\mathbf{x}) = A^T B \quad (7)$$

where $A = \alpha_1, \ldots, \alpha_l$ and $B = [\phi(x_1)\phi(\mathbf{x}), \ldots, \phi(x_l)(\mathbf{x})]$.

The ultimate feature vectors become $F = f_{1k}, \ldots, f_{nk}$ where k is the total number of eigenvectors that we retained for each class.

3 Eigendimension as a classifier

In the visual classification, many popular classifiers are employed for separating the features including nearest neighborhood (such as L_1 and L_2 norm), SVM, Bayes classifier, fisher linear discriminent, etc. Among them, nearest neighborhood is mostly used for such classification. In fact, the above mentioned classifiers work as hyperplane classifiers where the datasets are considered to be linearly separable. However, our employed datasets produced highly non-linear and non-convex feature space and, therefore, the traditional classifiers are, most of the time, not suitable for such application. An alternate solution has been proposed to learn the classifiers using adaboost in [2] that need manual computation, and it is computationally expensive. To avoid these, we propose eigendimension matching algorithm [12] that work as a classifier itself.

In the eigenspace, classes of objects or different objects conform to different spaces, and therefore, they can easily be classified by comparing their vectorial dimensions. The basis of the proposed method lies on this fact and we need only a few eigendimensions for classifying the unknown manifold. In this method, we need to calculate the minimum and maximum range of each eigendimension of the training datasets. The classification decisions are:

- Every selected eigendimension of the testing dataset should be greater than or equal to the minimum range of the corresponding eigendimension of the training dataset.

- Every selected eigendimension of the testing dataset should be lesser than or equal to the maximum range of the corresponding eigendimension of the training dataset.

This, in fact, groups the eigenspace based on their classes and they are classified by comparing their dimensional space.

4 Experimental Details

To observe the effectiveness of kernel eigenspace's ability to separate the feature space, we have designed two different experiments, (1) First experiment concentrates on being able to classify the various orientations of a car. (2) the second experiment focuses on making an accurate and fast decision on whether an image observed by our system is of a car or not. The orientations of the cars are limited in three viewpoints: *Car-rear*, *Car-front* and *Car-side*. The other road side images are considered as non-cars.

The platform used in this study is a 1999 Toyota Landcruiser 4WD as shown in Fig. 1(a). It is equipped with the appropriate hardware and software to provide the environment for a developing drivers' assistance system that includes traffic system monitoring, monitoring of the drivers' state, vehicle state monitoring and vehicle control. A single camera is used, shown in Fig. 1(b), for tracking the outside scenes including car and non-car images. Since camera mounted in the car, we can obtain only the view points of the other vehicle around the smart cars. These viewpoints are classified as three different views: *Car-rear*, *Car-front* and *Car-side*. The other road side images are considered as non-cars.

For classifying a car's presence in a particular image, we employed the eigendimension matching technique. By considering only the maximum and minimum ranges of the cars' feature space, we are able to classify images that lie within this range as cars and the rest as Non-cars. We then compare the success of the eigendimension matching classifier against the conventional distance-based classifiers including Euclidean and Mahalanobis distance. Some of comparison results with other methods are also given in Table 1. For detecting a car, a sliding window along with Bayes voting method is employed.

(a) Toyota Landcruiser (b) Camera mounted in the car

Figure 1. Smart Car Platform: Toyota 1999

4.1 Data Sets

As a simulation for various traffic scenarios, we used the standard and non-standard data sets, shown in Table 1, some of them are available in the public domain. It should be noted that all images are captured from real-world scenes with natural lighting change, background and occlusion. Some of the images used in the experiments are shown in Fig. 2

Each of the images that has been used in our experiments are initially resized into a 32 × 32 image from its original resolution. The images obtained from the Smart Cars and Caltech are colour images. These are converted to greyscale before the eigenspace is developed. The other images from RTA and UIUC are already grey images, hence we omit this pre-processing conversion step.

Table 1. Datasets used in the experiment

Data Set	Train Image	Test Image	Our Perfm.	Sad & Ali [2]	Fergus [4]
Smart Cars -Car-back	1800	1800	97%	Nil	Nil
Smart Cars - Car-front	90	90	97%	Nil	Nil
Smart Cars -Car-side	34	34	88%	Nil	Nil
Smart Cars -Non-car	100	100	99%	Nil	Nil
Caltech - Car-back	170	480	97%	96%	90.03%
RTA - Car-front	260	260	96%	92%	87.2%
UIUC - Non-car	260	260	97%	94%	82%

The training part of the data was used for computing the

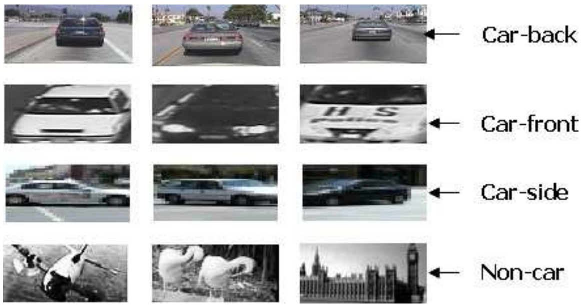


Figure 2. Some of the images used in the experiments

non-linear feature spaces and base learners, while the other part was employed for testing. The detection was based on whether an object of interest, in our case car, was present in the scene or not. Since the classification is our main focus, details of object detection are out of scope of this study. The results from experiments are then optimized by varying the scale factor of the Gaussian kernel function.

5 Experimental Results

For both experiments, we first develop the feature space for the different data sets according to Eq. 7. Fig. 3 and Fig. 4 show the feature space of the various orientations of the Smart Cars' and the caltech datasets. Fig. 4 and Fig. 6 shows the clustering results between caltech Car-rear, RTA's Car-front images and non-car images. Fig. 3 and Fig. 4 represent the feature spaces for the first Experiment where results of clustered spaces are well separated to implement eigen dimensional matching for the classification. As for Experiment 2, Fig. 5 and Fig. 6 show the clustering results between caltech Car-rear and UIUC's Non-car images and RTA's (NSW Road Transport Authority, Australia) Car-front and UIUC's Non-car respectively.

As discussed earlier, we tuned the Gaussain Kernel with the scaling factor to achieve higher classification rates. This was a trial and error process. A scaling factor of 1 can be noted as the optimum. We can easily observe the clear separation involved in the feature space. As the next step, we conducted performance evaluation of the three classifiers by associating a confidence statistic to each classifier. In this case we used a T-statistic, from which we computed the bounds for the confidence intervals. For both experiments we computed the bounds at 95% confidence. The results can be interpreted as follows - "One can say with 95% confidence that classifier X will produce classification rates between Y% and Z% with false positive rates between M% and N%", where X is the classifier, Y and Z the classification rates and M and N the false positive rates. Performance of the various classifiers for both experiments are shown below in Table 2 and Table 3.

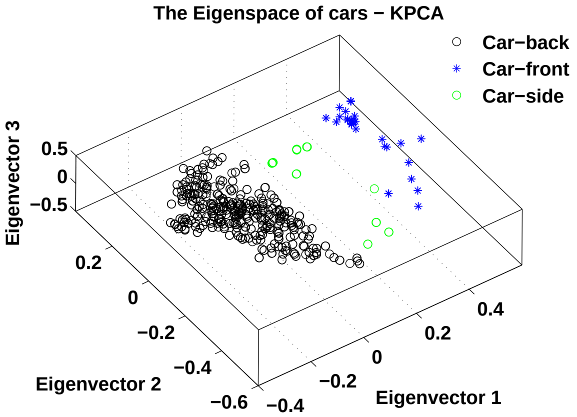


Figure 3. Clustering the spaces of different viewpoints (rbf 1.0)

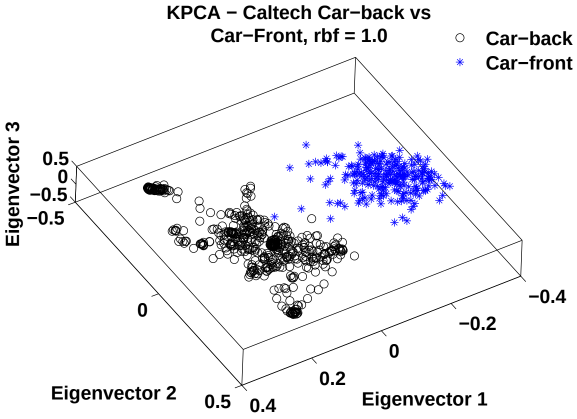


Figure 4. Clustering the caltech and car-front space (rbf 1.0)

The results show the excellent performance of the eigendimension matching classifier as compared to the conventional distance based classifiers after performing feature extraction using kernel eigenspace. All car orientations, car and non-car images are well separated in their respective feature spaces. The reason for the success behind eigendimension matching algorithm lies in the fact that the feature spaces were well separated for classifying car and non-car from their respective viewpoints. Detection performances have also listed in Table 1 and the obtained results are also compared with two well known methods.

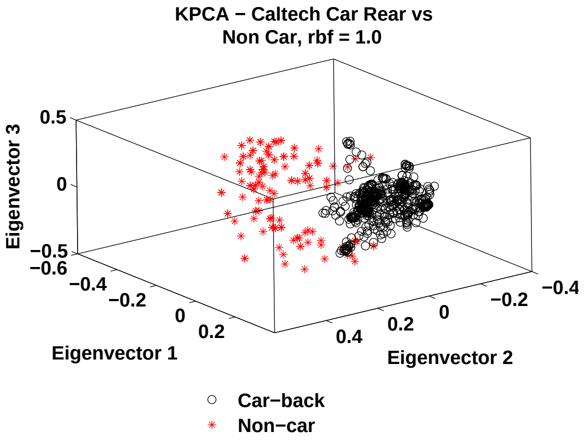


Figure 5. Clustering the caltech and non-car space (rbf 1.0)

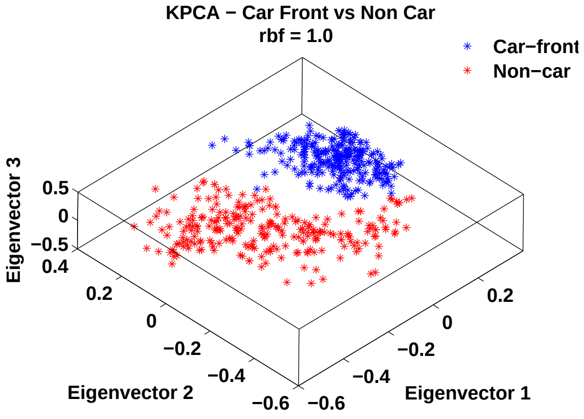


Figure 6. Clustering RTA's car-front and non-car spaces (rbf 1.0) of UIUC data

Table 2. Experiment 1 - Evaluation of the classifiers

Classifier	Confidence Intervals	
	Classification Rate	False Positives
Eigendim. Matching	87.1%-100%	0%-10%
Euclid. Distance	61.2%-81.1%	18.5%-38.1%
Mahalan. Distance	31.9%-100%	0%-64.4%

Table 3. Experiment 2 - Evaluation of the classifiers

Classifier	Confidence Intervals	
	Classification Rate	False Positives
Eigendim. Matching	84.0%-100%	0%-14.4%
Euclid. Distance	66.4%-98.7%	2.4%-28.1%
Mahalan. Distance	55.2%-97.4%	0%-23.4%

6 Conclusion

This study highlights the achievements of kernel eigenspace in vehicle classification and detection. The success of multi-class feature separations for the car classification and detection of multi-class objects can be seen in this paper. The ability of the classifier to identify the presence of a car from an occluded image is also a highlight of this study. The simplicity of kernel eigenspace for feature extraction, which only involves optimizing the scaling factor of the kernel, makes it more of an attractive prospect than the method proposed by [2]. Feature classification using eigenspace grouping is also a simple steps compare to other methods such as adaboost. However, the employed data variations which claimed to be non-linear and multi-class in this study was not clearly described due to its limited scope. Moving to the future, we will strive to improve the detection rate by integrating the other methods such as Adaptive Boosting and SVM classifier.

References

[1] S. Agarwal. Learning to detect objects in images via a sparse part-based representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] S. Ali and M. Shah. A supervised learning framework for generic object detection in images. In *International Conference on Computer Vision*, pages 1347–1354, 2005.

[3] Y. Bogomolov and et al. Classification of moving targets based on motion and appearance. In *British Machine Vision Conference*, 2003.

[4] R. Fergus and et al. Object class recognition by unsupervised scale invariant learning. In *Computer Vision and Pattern Recognition*, 2003.

[5] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing Company Limited, 1986.

[6] S. Gopte. Detection and classification of vehicles. *IEEE Transaction on Intelligent Transportation System*, 3(1):37–47, 2002.

[7] C. Liu. Gabor-based kernel pca with fractional power polynomial models for face recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2004.

[8] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(6):780–788, 2002.

[9] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 5(14):39–50, 1995.

[10] M. M. Rahman and S. Ishikawa. Recognizing human behaviors employing global eigenspace. In *International Conference on Pattern Recognition*, 2002(CDROM Version).

[11] M. M. Rahman and S. Ishikawa. Human motion recognition using an eigenspace. *Pattern Recognition Letters*, 6(26):687–697, 2005.

[12] M. M. Rahman and A. Santhanam. Moving vehicle classification using eigenspace. In *IEEE/RSJ International Conference on Robotic Systems*, pages 3849–3854, 2006.

[13] S. Romdhani and S. Gong. A multi-view nonlinear active shape model using kernel pca. In *British Machine Vision Conference*, pages 483–492, 1999.

[14] B. Schölkopf, A. J. Smola, and K.-R. Müller. Non-linear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[15] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991.

[16] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *International Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[17] C. J. Twining and C. J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. In *British Machine Vision Conference*, 2001.

[18] M. H. Yang and N. Ahuja. Face recognition using kernel eigenfaces. In *International Conference on Image Processing*, pages 37–40, 2000.

Nonlinear k-subspaces based appearances clustering of objects under varying illumination conditions

Xi LI and Kazuhiro Fukui
Graduate School of Systems and Information Engineering
University of Tsukuba, JAPAN
{xili@viplab.is,kfukui@cs}.tsukuba.ac.jp

Abstract

Unsupervised clustering of image sets of 3D objects has been an active research field within vision community. It is a challenging task since the appearance variation of the same object under different illumination condition is often larger than the appearance variation of different object under the same illumination condition. Some previous methods perform the appearance clustering using k-subspaces algorithm by assuming that the set of images of a Lambertian object approximately reside in a low dimensional linear subspace. This paper further extends the original k-subspaces clustering algorithm to the nonlinear case. The sum of the squares of distance to corresponding feature points of each nonlinear subspace cluster centers is minimized using Expectation-Maximization like iteration procedure. Those distances can be novelly defined via inner product by kernel trick. Experiments on different datasets show that the proposed kernel-based nonlinear k-subspaces clustering algorithm achieves much higher clustering rate than its linear counterpart.

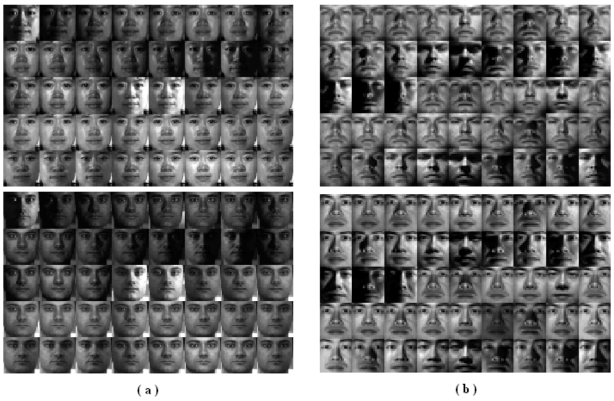


Figure 1. Sample images of two persons under varying illumination conditions in the a) PIE database[8] b) YaleB database[9]. The appearance variation of the same object under different illumination condition is larger than the appearance variation of different object under the same illumination condition which renders the clustering according to the underlying identity a difficult task.

1 Introduction

Unsupervised clustering of image sets of 3D objects under varying viewing conditions has been an active research field within computer vision community[1, 2, 3, 4, 5, 6, 7]. Typically, there are several viewing aspects that could affect the 2D image appearances during the projection procedure: the relative orientation between the viewing camera and the target object, the illumination conditions under which the images are acquired, and the reflective properties of the surface. As in [7], this paper studies the problem of unsupervised clustering of images sets of objects with Lambertian surface taken under varying illumination conditions while the target objects are in fixed poses. The objective is to partition the given image sets into disjoint subsets corresponding to underlying identities. It is a challenging task since the appearance variation of the same object under different illumination condition is often larger than the appearance variation of different object under the same illumination condition. Figure 1 shows an example of such case using CMU PIE face dataset[8] and YaleB face dataset[9]. It can be clearly seen that a direct standard clustering using Euclidean distance metric such as k-means algorithm will yield poor result.

Previously, several algorithms have been proposed for the problem of appearances clustering of objects under varying illumination conditions[1, 2, 3, 4, 5, 6, 7]. The most related to our work is[7]. In their work, J. Ho et al.

presented an appearance-based methods for clustering image sets of 3-D objects, acquired under varying illumination conditions, into disjoint subsets corresponding to each subject.They iteratively performed appearances clustering using K-subspaces algorithm by assuming that images for the same object approximately reside in a low dimension linear subspace[2, 5, 6]. The K-subspaces clustering algorithm can fully exploit the linear geometric structure hidden among the image sets. They proposed two method for the initialization: One is based on the concept of illumination cones and the other is based on spectral clustering, where the affinity matrix is computed by image gradient comparisons.

Recent studies show that non-linear subspace approximation via kernel trick is superior compared to their linear counterpart because the real life high dimensional data, such as the vectorized image data, is often inherently nonlinear rather than simple normally distribution[10]. One of the most representative innovation has been the kernel principal component analysis(KPCA), which makes use of the kernel trick to non-linearize PCA and extract nonlinear subspaces. This kind of kernel based algorithms can model complex real life data structures more faithfully and have achieved much success within machine learning and pattern recognition communities[10]. Motivated by those successes, this paper proposes a novel algorithm that performs nonlinear subspace clustering in the mapped high dimensional feature space. Firstly, the input patterns are mapped into a high-dimensional feature space via a nonlinear mapping function. Then the nonlinear subspaces are extracted in the feature space and distances between the mapped feature points and extracted nonlinear subspaces are defined via inner products by kernel trick. The objective function, which is the sum of the squares of distance to corresponding feature points of each nonlinear subspace cluster centers, is minimized using Expectation-Maximization like iteration procedure. Experiments on two different face datasets show that the proposed nonlinear Kernel K-subspaces Clustering(Kernel-KsC) algorithm converges quickly and achieves much higher clustering rate than that of the original Linear K-subspaces Clustering (Linear-KsC) algorithm.

The rest of this paper is organized as follows: Firstly, we describe the Linear-KsC algorithm for unsupervised appearances clustering of objects under varying illumination conditions in Section 2. Section 3 describes the proposed Kernel-KsC algorithm in detail. Experimental results of the clustering performance comparison between the Linear-KsC algorithm and the proposed Kernel-KsC algorithm using CMU PIE face dataset and Yale face dataset are presented in Section 4. Section 5 draws the conclusion.

2 Unsupervised appearances clustering

J. Ho et al.[7] showed that both the illumination cones based method and the gradient metric based method give reasonable results for the initialization of the iteration procedure of the linear K-subspaces clustering. They claimed that the computation of gradient metric was reliable in low-resolution images and could give promising clustering results. Here we adopt the similar framework as in [7]. That is to say, firstly we also use the gradient metric based method for the initialization and then the initial rough clustering results are further refined using subspace clustering. We put the emphasis on showing the superiority of the proposed kernel K-subspaces clustering algorithm over its linear counterpart. For the sake of completeness, we describe the main idea of gradient metric based clustering initialization and the iterative procedure of the original linear K-subspaces clustering algorithm briefly in the next subsections.

2.1 Spectral clustering based on gradient affinity

Suppose there are N input images $\{I_1, ..., I_N\}$ where each image has s pixels. The idea of gradient affinity is simple to directly compare between image gradient pairs. The differences in the magnitude of the image gradient and the relative orientation over the whole image plane are summed. Once we get the affinity matrix, standard spectral based algorithms[11] can be used to perform unsupervised clustering. For more details, refer to literatures[7]. Also, Some variants of the spectral clustering algorithm have been developed for the problem of automatic determination of the number of cluster centers[12]. This paper focuses on the superior performance of the proposed nonlinear K-subspaces clustering algorithm over its linear counterpart and we simply assume that the number of the cluster centers in known in advance.

Previous studies on illumination invariants show that the set of monochrome images of a convex object with a Lambertian reflectance forms a convex polyhedral cone when illuminated by an arbitrary number of point light sources at infinity[4, 5]. This implies that the collection of appearances of objects can be approximated by some low dimensional linear subspaces. So the initial rough clustering results using spectral method based on gradient affinity can be further refined via subspace clustering algorithm.

2.2 Linear K-subspaces Clustering

Linear K-subspaces Clustering(Linear-KsC) algorithm is an extension of the traditional K-means clustering algorithm. While the K-means clustering algorithm tries to find

K cluster centers using Euclidean distance metric between point pairs, the objective of K-subspaces clustering algorithm is to find K linear subspace base clusters using distance metric between points and subspace bases. The K-subspaces clustering algorithm shares the similar idea with the k-means algorithm and the flowcharts of both iteration procedure are almost the same. Firstly, each point is assigned to the nearest subspace cluster base. The distance is computed as the length of the difference vector between the original point and its reconstruction using the corresponding subspace base center (In the next section, we will show that the computation of the distance can be written in the form of inner product, which renders the extension to the nonlinear case possible) . Then the subspace bases are updated by principal component analysis. Usually the iteration procedure converges quickly in just several number of loops. It should be noted that the original 2D image matrix representation is firstly transformed into 1D vector form.

Specifically, the linear K-subspaces clustering algorithm can be described as follows:

Algorithm 1: Linear K-subspaces Clustering(Linear-KsC)

1. *Initialization*: Suppose there are N input images $\{I_1, ..., I_N\}$ where each image has s pixels. Starting with a collection $\{S_1, ..., S_K\}$ of K subspaces of dimension d, where $S_i \in R^s$. The corresponding orthnormal bases for each subspace is denoted as U_i with size $s \times d$;

2 *Points assignment*: Denotes $\rho(x_i) \in \{1, ..., K\}$ as the cluster label for point x_i. Then each point is assigned a new label as follows:

$$\rho(x_i) = argmin_k \|(I_{s\times s} - U_k U_k^T)x_i\| \tag{1}$$

where $k \in 1, ..., K$;

3 *Subspace update*: Update each subspace bases $U_i, i \in \{1, ..., K\}$ using the new label information. U_i can be formed by retaining the eigenvectors corresponding to the top d eigenvalues of the scatter matrix constructed using those sample points with label i. This can be easily computed via principal component analysis[10];

4 *Repeat step 2 and 3 until convergence*: The iteration procedure will stop if the label information does not change in two successive iteration steps.

3 Kernel K-subspaces Clustering(Kernel-KsC)

Although the method in the previous section can give the clustering result reasonable to some extent. It still has the limitation that the refining procedure using Linear-KsC is based on the assumption that the appearances of a target object can be approximated well using linear subspace. Recent studies show that often the distributions of the high dimensional image data are inherently nonlinear. Many successful algorithm for extracting those complex nonlinear structures in real life data have been proposed and one of the most representative ones is the Kernel Principal Component Analysis(KPCA)[10]. KPCA has achieved great success in the areas of pattern recognition and image processing, such as face recognition and image de-noising . We will show that combining the K-type clustering framework with the kernel based nonlinear feature extraction would yield much better result for the problem of appearances clustering of objects under varying illumination conditions.

In the next of this section, we first review the nonlinear subspace extraction using KPCA for completeness. Then we define the distance between point and nonlinear subspace in the transformed feature space. Intuitively, the distance can be defined as the "length" of the difference vector between points and nonlinear subspace in the transformed feature space. But a direct computation of the distance is infeasible due to the high, or even infinite, dimensional space. Fortunately, those difference vectors can be written in the form of linear combination of transformed high dimensional feature points, which makes it possible to compute the distance via "kernel trick" without explicitly implement the inner product in the high dimensional feature space. Next we will describe the proposed nonlinear Kernel K-subspaces Clustering(Kernel-KsC) algorithm in detail. Promising experimental results will be presented in section 4.

3.1 Nonlinear subspace extraction via KPCA

Recent studies in pattern recognition community show that often the target distributions, such as those of multi-view patterns of a 3D object or image sets of a single objects under varying illumination conditions, is highly nonlinear. The simple linear subspace representation is not suitable for representing highly nonlinear structures. Several non-linear dimension reduction methods have been proposed. One representative is the kernel principal component algorithm which is an unsupervised non-linear feature extractor[10]. Kernel principal component analysis allows estimation of non-linear subspace for the data distribution such as face images.

First, the input pattern $x_i \in R^s, i \in \{1, ..., m\}$ is transformed from s dimensional input space I onto an higher dimensional feature space F via a nonlinear mapping $\phi : R^s \to R^{s_\phi}, x \to \phi(x)$. To perform the standard PCA on the mapped patterns, we need to calculate the inner product $(\phi(x_i) \bullet \phi(x_j))$ between the function values. Direct calculation of those inner products is difficult since the dimension of the feature space F could be very high, possibly infinite. Kernel learning theory shows that if the nonlinear map ϕ is defined through a kernel function $k(x, y)$ which satisfies Mercers conditions, the inner products $(\phi(x_i) \bullet \phi(x_j))$ can be calculated from the inner products $k(x \bullet y)$. A common choice is to use the Gaussian kernel function: $k(x, y) = exp(-\frac{\|x_i - x_j\|^2}{\sigma^2})$ where σ is the scale parameter. The N orthonormal basis vectors $e_i, i = \{1, ..., N\}$, which span the nonlinear subspace, can be represented by the linear combination of all the m transformed patterns in the feature space $\phi(x_j), j = \{1, ..., m\}$, i.e. $e_i = \sum_{j=1}^m a_{ij}\phi(x_j)$ where the coefficient a_{ij} is the j-th component of the eigenvector a_i corresponding to the i-th largest eigenvalue λ_i of the $m \times m$ matrix K defined by $Ka = \lambda a$ where $k_{ij} = (\phi(x_i) \bullet \phi(x_j)) = k(x_i, x_j)$. Each a_i is normalized to satisfy $\lambda_i(a_i, a_i) = 1$ The projection of the mapped $\phi(x)$ onto the i-th orthonormal basis vector e_i of the nonlinear subspace base can be computed by the following equation via the kernel trick: $(\phi(x), e_i) = \sum_{j=1}^m a_{ij}k(x, x_j)$

3.2 Kernel K-subspaces Clustering

The purposed Kernel K-subspaces Clustering(Kernel-KsC) algorithm assigns each input pattern to its nearest nonlinear subspace base. Denote $D(x, S)$ as the difference between an input pattern in the transformed space $\phi(x)$ and its reconstruction using a nonlinear subspace S with the corresponding orthonormal basis vector defined in Section 3.1. Here we only keep the first d basis vectors the nonlinear subspace S with higher eigenvalues. Then

$$D(x, S) = \phi(x) - \Theta_S(\Xi_S(\phi(x))) \tag{2}$$

where $\Xi_S(\phi(x)) \in R^d$ is the projection of $\phi(x)$ onto the nonlinear subspace S and $\Theta_S(\Xi_S(\phi(x)))$ is its reconstruction. Denote the d dimensional nonlinear subspace S as

$$\begin{aligned} S &= [e_1, ..., e_d] \\ &= [\sum_{j=1}^m a_{1j}\phi(x_j), ..., \sum_{j=1}^m a_{dj}\phi(x_j)] \end{aligned} \tag{3}$$

then

$$\begin{aligned} &\Theta_S(\Xi_S(\phi(x))) \\ &= [\sum_{t=1}^m a_{1t}\phi(x_t), ..., \sum_{t=1}^m a_{dt}\phi(x_t)] \times \\ &[\sum_{s=1}^m a_{1s}\phi(x_s), ..., \sum_{s=1}^m a_{ds}\phi(x_s)]^T\phi(x) \\ &= [\sum_{t=1}^m a_{1t}\phi(x_t), ..., \sum_{t=1}^m a_{dt}\phi(x_t)] \times \\ &[\sum_{s=1}^m a_{1s}\phi(x_s) \bullet \phi(x), ..., \sum_{s=1}^m a_{ds}\phi(x_s) \bullet \phi(x)]^T \\ &= [\sum_{t=1}^m a_{1t}\phi(x_t), ..., \sum_{t=1}^m a_{dt}\phi(x_t)] \times \\ &[\sum_{s=1}^m a_{1s}k(x_s, x), ..., \sum_{s=1}^m a_{ds}k(x_s, x)]^T \end{aligned} \tag{4}$$

Here the inner product in the transformed space , which is difficult to compute due to the inherently high or infinite dimension of the transformed space, is implemented via the kernel trick. After some algebraic derivations, we obtain:

$$\begin{aligned} &\Theta_S(\Xi_S(\phi(x))) \\ &= \sum_{t=1}^m \{\sum_{r=1}^d a_{rt} \sum_{s=1}^m a_{rs}k(x_s, x)\}\phi(x_t) \\ &= \sum_{t=1}^m B_t\phi(x_t) \end{aligned} \tag{5}$$

where

$$B_t = \sum_{r=1}^d a_{rt} \sum_{s=1}^m a_{rs}k(x_s, x), t = 1, ..., m \tag{6}$$

So from the definition of equation2

$$D(x, S) = \phi(x) - \sum_{t=1}^m B_t\phi(x_t) \tag{7}$$

For the sake of clearness, we represent x as x_0 and let B_0 equals to the value of -1, then

$$D(x, S) = -\sum_{t=0}^m B_t\phi(x_t) \tag{8}$$

That is to say, the difference vector $D(x, S)$ can be written in the form of linear combination of nonlinear transformed input patterns. The square of the length of the difference vector $D(x, S)$ can be computed using the inner product as follows:

$$\begin{aligned}\|D(x,S)\|^2 &= \sum_{i=0}^{m}\sum_{j=0}^{m} B_iB_j\phi(x_i)\bullet\phi(x_j) \\ &= \sum_{i=0}^{m}\sum_{j=0}^{m} B_iB_jk(x_i,x_j)\end{aligned} \tag{9}$$

Based on the definition of the distance between input patterns in the transformed space and nonlinear subspace bases, we can define the objective function to be minimized as follows:

$$\sum_{i=1}^{K}\sum_{\rho(x_j)\in i}\|D(x_j,S_i)\|^2 \tag{10}$$

The iteration procedure of the proposed Kernel K-subspaces Clustering (Kernel-KsC) method can be described as follows:

Algorithm 2: Kernel K-subspaces Clustering(Kernel-KsC)

1. *Initialization*: Starting with an initial labeling of the input image pattern $\{x_1,...,x_n\}$ into K clusters. Compute the nonlinear subspaces $S_i, i = \{1,...,K\}$ of the corresponding data with label i via kernel principal component analysis defined in Section 3.1;

2 *Points assignment*: Denotes $\rho(x_i)$ as the cluster label for point x_i. Then each point is assigned a new label as follows:

$$\rho(x_i) = argmin_k\|D(x_i,S_k)\|^2 \tag{11}$$

where $k \in \{1,...,K\}$. The $\|D(x_i,S_k)\|^2$ can be computed using equation 9;

3 *Non-linear subspaces update*: Update each nonlinear subspace bases $S_i, i \in \{1,...,K\}$ using the new label information via kernel principal component analysis;

4 *Repeat step 2 and 3 until convergence*. The iteration procedure will stop if the change of the value of the objection function is small enough, or equivalently if the label information does not change anymore in two successive iteration steps.

The iterative procedure of the proposed Kernel-KsC algorithm implement appearances clustering by assigning each input pattern(the vector form of the original 2D image matrix) according to the underlying nonlinear subspace distribution structure, which is a more accurate description of the real life data than its simple linear counterpart. Thus a higher correct clustering rate can be achieved, which will be demonstrated in the next section. The correct clustering rate can be defined as:

$$\sum_{i=1}^{K}\tau_i/n \tag{12}$$

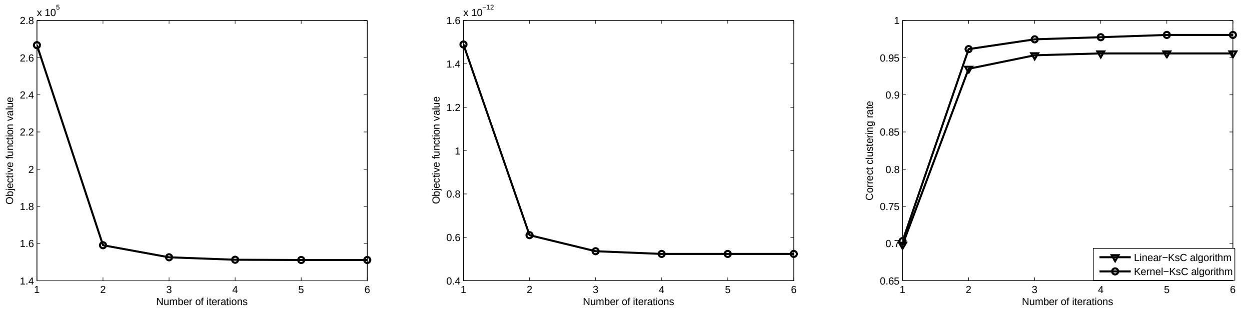
where n denotes the total number of images and τ_i denotes the maximum number of images with the same true identity clustered into the class i.

4 Experimental results

We used the CMU PIE database[8] and the Yale Face Database[9] as the test sets for performing unsupervised appearance clustering under varying illumination conditions. We compared the clustering performance of the proposed nonlinear kernel K-subspaces method with that of its linear counterpart. The iteration procedures of both algorithms were initialized using the gradient affinity based spectral clustering method proposed in [7]. For both of the data sets, the proposed nonlinear K-subspaces clustering method achieves satisfactory clustering results and outperforms its linear counterpart. The detail of the experiments will be given below. We use the Gaussian kernel function[10] in all the following experiments.

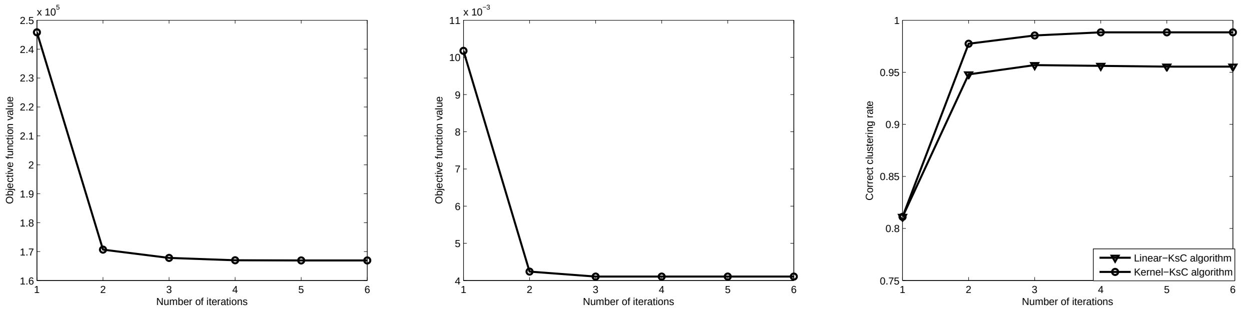
For the CMU PIE database, we used a subset of 40 frontal or near-frontal images of 68 individuals which were taken under different illumination conditions. Figure 1(a) shows the sample image sets for two specific subjects. First, the resolution of the original images are resized from original 32×32 pixels to 16×16 pixels and the range of image intensities are normalized to $\{0, 1\}$. Then the gradient fields are computed and the spectral clustering was implemented using the similarity measurement matrix . After initialization, the proposed nonlinear Kernel K-subspaces Clustering (Kernel-KsC) and Linear K-subspaces Clustering (Linear-KsC) algorithms are implemented respectively. Although there are several studies show that the number of cluster centers can be selected automatically by analyzing the distributions of the corresponding eigenvalues such as in [12], in this paper we assume the number of clusters, i.e. K, is known in advance since the emphasis of this work is to show the superiority of the proposed Kernel-KsC algorithm, which is a nonlinear extension to the original Linear-KsC algorithm by exploiting the inherent nonlinear distribution property in the input patterns, over its linear counterpart. Figure 2(a) and (b) show the objective function value during iteration procedure for the proposed nonlinear K-subspaces clustering algorithm and the original K-subspaces clustering algorithm, respectively. Figure 2(c) shows the correct clustering rate comparison of the two methods during iteration. Although the appearance variation of the same person is fairly large due to the varying illumination conditions,the proposed nonlinear K-subspaces clustering method achieved satisfactory clustering results and outperforms its linear counterpart greatly.

The Yale B database used in our experiment consists of 450 images with 45 frontal images of each person captured under varying light directions. Figure 1(b) shows sample



(a) (b) (c)

Figure 2. Experimental results for PIE database:(a) and (b)are the objective function values as a function of the number of iterations for Linear-KsC and Kernel-KsC,respectively. (c) shows correct clustering rate comparison of the two methods during iteration.



(a) (b) (c)

Figure 3. Experimental results for YaleB database:(a) and (b)are the objective function values as a function of the number of iterations for Linear-KsC and Kernel-KsC,respectively. (c) shows correct clustering rate comparison of the two methods during iteration.

images of two persons from these subsets. Each image is resize to resolution of 16×14 pixels and initialized using the same method as for the PIE database. Figure 3 show the experiment results.

Both of the above experiments clearly show that nonlinear K-subspaces and linear K-subspaces method converge quickly in just several iteration steps. And the proposed nonlinear kernel-based K-subspaces clustering algorithm achieves much lower error rate than the linear K-subspaces clustering algorithm.

5 Conclusions and future work

This paper studies the problem of appearance clustering under varying illumination conditions and a novel nonlinear kernel K-subspaces clustering algorithm is presented. The proposed Kernel-KsC algorithm further extends the original K-subspaces clustering algorithm to the nonlinear case since inherently the distribution of the real life image data has a complex nonlinear structure rather than simple linear case. Firstly, the input space is mapped into a high-dimensional feature space using nonlinear mapping function. Then the nonlinear subspaces are extracted in the feature space and distances between the mapped feature points and those nonlinear subspaces are computed via inner products by kernel trick. Experiments on several real life data sets show that the proposed nonlinear kernel K-subspaces clustering algorithms converges quickly and achieves higher clustering rate that that of the original linear K-subspaces clustering algorithm.

Besides successful applications in computer vision community, recently subspace clustering algorithm also achieves successes in other areas such as data mining and bio-informatics, where the data may also have inherent nonlinear properties. We believe that the proposed kernel-based nonlinear subspace clustering algorithm can outperform its linear counterpart for those problems. The application of the proposed nonlinear K-subspaces clustering in other fields might be future research directions.

References

[1] D. W. Jacobs, P. N. Belhumeur, and R. Basri. Comparing images under varying illumination. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 610-617, 1998.

[2] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, volume 1, pp. 254-261, 2000.

[3] R. Basri, D. Roth, and D. Jacobs. Clustering appearances of 3D objects. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 414-420, 1998.

[4] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. In Proc. Intl Conf. on Computer Vision, volume 2, pp 383-390, 2001.

[5] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions. In Intl Journal of Computer Vision, volume 28, pp 245-260, 1998.

[6] A. Georghiades, D. Kriegman, and P. Belhumeur. From few to many: Generative models for recognition under variable pose and illumination. IEEE Int. Conf. on Automatic Face and Gesture Recognition, pp. 277-284, 2000.

[7] J. Ho, M. H. Yang, J. Lim, K. C. Lee, D. Kriegman. Clustering Appearances of Objects Under Varying Illumination Conditions. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 11-18, 2003.

[8] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression Database. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(12), pp. 1615-1618, 2003

[9] A. S. Georghiades, P. N. Belhumeur and D. J. Kriegman, From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. IEEE Transactions on Pattern Analysis and Machine Intelligence. 23(6), pp. 643-660, 2001

[10] B. Schlkopf and A.J. Smola: Learning with Kernels. MIT Press, Cambridge, MA, USA, 2002

[11] A. Ng, M. Jordan, and Y.Weiss. On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems 15, pp. 849-856, 2002.

[12] G. Sanguinetti, J. Laidler and N. Lawrence. Automatic determination of the number of clusters using spectral algorithms. IEEE Machine Learning for Signal Processing, Mystic, Connecticut, USA., pp. 28-30, Sep, 2005

Face Recognition Based on Holistic Information and Minimum Mahalanobis Classifier

I Gede Pasek Suta Wijaya, Keiichi Uchimura, and Zhencheng Hu
Computer Science and Electrical Engineering Graduate School of Science and Technology,
Kumamoto University Kumamoto-Shi, 860-8555 JAPAN
gdepasek@yahoo.com, uchimura@cs.kumamoto-u.ac.jp, hu@cs.kumamoto-u.ac.jp

Abstract

Human face image recognition is an active research area in image processing applications because there are many potential applications which cover human computer interactions, forensics, and surveillance. The proposed scheme based on holistic information face image obtained by multi-resolution wavelet analysis and minimum Mahalanobis classifier to classify the facial features to the person's class. Facial feature is built by keeping small part of frequency domain coefficients which have big magnitude value. Next, from facial features we calculate the mean of each face class and global covariance. By assuming that each class has multivariate normal distribution and all classes have the same covariance matrix, Mahalanobis distance can be used to classify the facial features to person's class. The aim of proposed system is to solve the high space requirement and retraining problems of classical LDA and PCA. The system is tested using several face databases and the experiments result is compared to well-known classical LDA and PCA method.

Keywords : *facial feature, matching, wavelet, LDA, eigenface*

1. Introduction

The idea of face recognition was inspired by the ability of human being to recognize object or pattern based on training process. As we have known, human has good recognizer system for some objects or patterns because they have continuously performed the learning/training process since childhood. That process was adopted by some researchers to build any kinds of recognition systems, such as recognition system based on geometrical analysis, statistical analysis, neural network, etc [1].

Simply, face recognition is a matching process between a query facial feature and target facial features using certain technique. It is difficult to do because face variations in a single face can be very large, while the variations between different faces are quite small. Furthermore, face image information depends on ethnicity and registration method (i.e., capture method, lighting condition, and device). For example, frontal face images contain better information than lateral images as well as the face acquired by digital than analog camera.

Face recognition has been examined since more than 20 years ago and up to now it is still one of the most active research areas because there are some potential application areas which range from human computer interaction to authentication, security, and surveillance. Researchers have proposed some approaches of face recognition systems, such as face recognition based on statistical analysis, texture analysis, frequency analysis, artificial intelligent (neural network, genetics algorithm, fuzzy logic), and combination of them[1]. PCA-based and LDA-based face recognition systems have been successfully implemented [2,3,4,5] and encouraging results has been achieved. However, both of them have their limitation: large computational costs and high memory space requirement. The main limitation is retraining problems which means that we have to retrain all face image class to get optimal projection when the new classless are added to the PCA-based and LDA-based face recognition systems. Moreover, PCA has poor discriminatory power and LDA has singularity problems.

This paper proposes an alternative face recognition system which is based on holistic or global information of face image and minimum Mahalanobis classifier. The holistic information of face image is obtained by multiresolution wavelet analysis of entire image without geometrical normalization. The minimum Mahalanobis classifier is implemented to classify the facial feature to person's class. The main aims of this method are to decrease the memory space requirement and to improve the performance recognition of the PCA and LDA-based face recognition. The proposed method consists of three main processes: pre-processing, facial feature extraction, and face classification.

2. Previous Work

The previous works related to our approach are face recognition based on holistic or global approach as described in Ref. [2,3,4,5,7,8]. Ref. [7] describes face recognition based on wavelet packet tree analysis for frontal view of human faces under roughly constant illumination. The facial feature was built by implementing wavelet packet tree analysis of bounding box face and then calculating the mean and variance of sixteen matrixes wavelet coefficient. That approach does not work for non-frontal faces view and needs constant illumination to make the face bounding box. Ref. [8] describes face recognition based on combination of DCT analysis and face localization technique for finding the global information of face image, but it requires eyes coordinate, which have to input manually, to perform geometrical normalization. The global face information was created by keeping small part of big magnitude value of DCT coefficients.

PCA [2, 4] and LDA [3, 4] is a well-known scheme for feature extraction and dimensional reduction. Those scheme and their variations have been successfully applied in face recognition. It was reported that LDA was superior in face recognition to PCA, but both of them have their limitations: large computational cost, high memory space requirement, and retraining problems. Moreover, PCA has poor discriminatory power while LDA has singularity problems. The mostly related approach to our system are face recognition based on LDA and its variations as described in Ref. [3,4,5,6]. Ref. [3] proposed the combination D-LDA and F-LDA to cover the weakness of classical LDA. It only solves the poor discriminatory and singularity problem. However, it still needs high memory space and should be retrained when new face class is added. Ref. [4] implemented DCT to reduce data dimensional and only small part of DCT coefficients is analyzed by LDA. Ref. [5] implemented the wavelet transforms to reduce the dimension of face image and employ a regulation scheme for the within-scatter matrix and use optimization procedure. It was reported that the Daubechies (Db-6) was implemented to filter image to resolution 29 x 23. However, this resolution is still coarse and lack of frequency-resolution.

In our method, we implement multiresolution wavelets analysis for reducing the original data dimension and minimum Mahalanobis classifier for classifying the face class without geometrical normalization and bounding box processing. It is difficult to compare our results with previous works because time consuming rarely reported and the test was carried out with different databases. Therefore, our approach results will be compared to classical PCA and LDA which has been tested with four face databases.

3. Subspace Based Face Recognition

3.1 Classical PCA

The aim of PCA is to find a transformation data such that feature clusters are most separable after the transformation. The most popular PCA analysis that is implemented for face recognition is eigen-face algorithm developed in Ref. [2] as described below.

Suppose a set of face image then we can define a set of vector $X=[x_1, x_2, x_3, \dots , x_N]$, where N is number of member set, by converting each image to a vector x_i of length m (*m= image width x image height*) and then placed x_i into X. Next, from matrix X, we calculate the mean of X using $\overline{X} = (1/n)\sum_{k=1}^{n} x_k$ and the covariance using $C_X = (X-\overline{X})(X-\overline{X})^T$. If the size of matrix X is large the calculation of covariance matrix will need high memory space and large computational cost. Turk et. all [2] proposed an strategy to overcome that limitations such as $C_X = (X-\overline{X})^T(X-\overline{X})$. The PCA projections matrix U can be obtained by eigen analysis of the covariance matrix C_X using the following equation:

$$C_X u_i = \lambda_i u_i, \quad i = 1, 2, 3, \dots , m \tag{1}$$

Where u_i is the i-th largest eigen-vector of C_X. We select a small number of eigen-vectors (m) corresponding to the largest eigen-value (i.e. $m<n$) and then the selected eigen vectors are placed into $U=[u_1, u_2, u_3, \dots , u_m]^T$. The projection of face image features to eigenface is obtained by equations below:

$$\mathrm{T} = U^T\left(X - \overline{X}\right) \tag{2}$$

Finally, the matching process is performed by calculating the score between the query (probe) facial features projection and the training facial feature projection set using L_2 metric. The minimum score is selected as the best likeness. This system requires high memory space for training and matching process.

In this research we modify the above PCA algorithm for large samples and each sample has some member.

As note that, the facial matrix is obtained by DWT analysis. The detail modified algorithm is described below:

1. Let define a big matrix $Q \in \Re^{m \times n \times c}$ which is collection of facial matrix class as denoted by $Q=[C_1, C_2, C_3,\ldots, C_c]$, where $C_i=[x_1, x_2, x_3,\ldots, x_n]$ is vector collection of member class, x_i is a column vector each image's holistic information, c is number of classes, and n is number of member class C_i.
2. The mean of each class is calculated by $\overline{\mu}_i = E(C_i) = (1/n_i)\sum_{k=1}^{n_i} x_k$ and then placed its into the mean vector set i.e. $\mathrm{M} = [\mu_1, \mu_2, \mu_3, \ldots, \mu_C]$
3. The global covariance is calculated using $C_Q = \frac{1}{c}\sum_{k=1}^{C}(C_k - \mu_k)(C_k - \mu_k)^T$.
4. The PCA projections matrix U can be obtained by eigen analysis of the covariance matrix C_Q using the equation (2)
5. The projection of each class matrix is performed using equation $\mathrm{T}_i = U^T(C_i - \mu_i), i = 1,2,3,\ldots,c$
6. Finally the similarity is determined by Euclidean distance (d) with the match criteria is the minimum of d and d is less then the threshold q.

The main problem of the PCA method is lack of power discriminant and requires retraining of all samples to obtain the optimum projection matrix.

3.2 Classical LDA

Like PCA, the main purpose of LDA is to find a linear transformation such that feature clusters are most separable after the transformation. It can be achieved by scatter matrix analysis. Let a big matrix $Q \in \Re^{m \times n \times c}$ as defined previously. The between-class scatter matrix S_b and the within-class scatter matrix S_w are respectively defined as

$$S_b = \sum_{i=1}^{c} n_i (\overline{m}_i - \overline{m})(\overline{m}_i - \overline{m})^T \qquad (3)$$

$$S_w = \sum_{i=1}^{c} \sum_{x_j \in C_i} (x - \overline{m}_i)(x - \overline{m}_i)^T \qquad (4)$$

Where $\overline{m}_i = (1/n_i)\sum_{k=1}^{n_i} x_k$ is the mean of class C_i (mean of i-th class) and $\overline{m} = (1/c)\sum_{k=1}^{c} m_k$ is the mean of all samples (global mean).

The class separation can be measured by the ratio of determinant of between-class scatter matrix of projected samples to the within-class scatter matrix of the projected samples, as equation below.

$$E = \arg\max_{E} \frac{|E^T S_b E|}{|E^T S_w E|} \qquad (5)$$

Where $E = [e_1, e_2, e_3,\ldots, e_m]$ is set of eigen-vectors corresponding to m largest eigen-values λ_i, which satisfy the equation (6). The eigen-vector and eigen-values can be obtained by computing the inverse of S_b and then solving the eigen problem of $S_W^{-1} S_b$ matrix.

$$S_b e_i = \lambda_i S_W e_i,\ i = 1, 2, 3, \ldots, m \qquad (6)$$

The projection of the linear discriminant functions is:

$$Y_i(C) = E^T(C_i - \overline{m}_i) \qquad (7)$$

The intrinsic problem of above algorithm is singularity problem in scatter matrix due to the high dimensional data and small number of training samples. Some methods have been proposed to solve the singularity problem as described in Ref. [3,4,5]. However, those methods can only solve the large computational load and can reduce memory requirement, but the retraining problem has not been covered yet.

4. Proposed Method

Face recognition algorithm discussed in this paper is shown in Fig. 1.

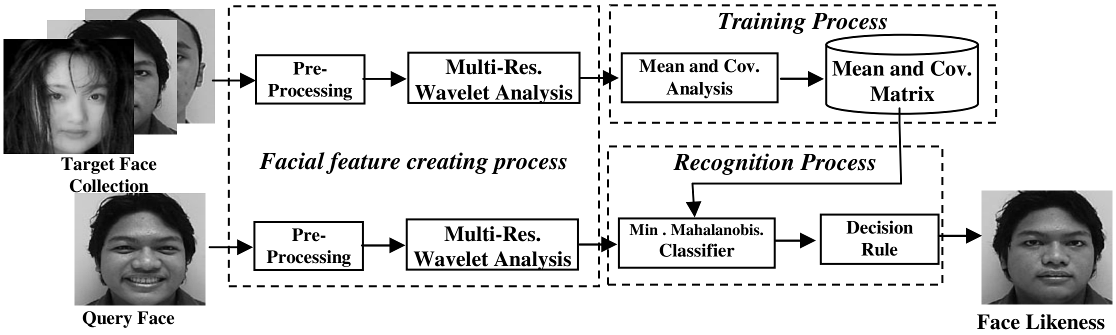


Figure 1. Proposed face recognition diagram bock.

It involves creating the holistic information, training and recognition process. This system works on gray scale image. If the system receives the color image, it will automatically be converted to grayscale using the luminance model which is used by NTSC. In the pre-processing block, each face image is normalized and equalized to remove non uniform lighting effect on face capturing. Next, the normalized and equalized face image is decomposed by multiresolution DWT algorithm as explained below. The multiresolution wavelet analysis can be performed by implementing repeatedly classical DWT which is called as filter bank decomposition, as shown in Fig. 2.

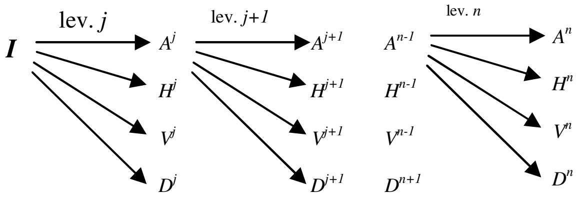


Figure. 2. Filter-bank wavelet decomposition.

Where *A, H, V,* and *D* is calculated by the classical DWT algorithm below:

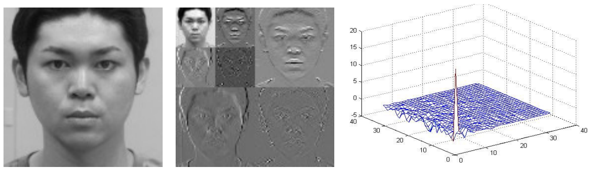
$$\begin{aligned} A &= \left[h * [h * f\,]_x \downarrow_2\right]_y \downarrow_2 \\ H &= \left[g * [h * f\,]_x \downarrow_2\right]_y \downarrow_2 \\ V &= \left[h * [\,g * f\,]_x \downarrow_2\right]_y \downarrow_2 \\ D &= \left[g * [d * f\,]_x \downarrow_2\right]_y \downarrow_2 \end{aligned} \tag{8}$$

where, * denote convolution , $\downarrow_2$ represent down sampling for *x* and *y* direction, *g* and *h* are high and low fast filter.

In order to make simple and fast the decomposition process, we apply two different Daubachies wavelets basis, namely Db4 and Db1. First, Db4 basis decomposes face images until level 2 and it just return the approximation coefficients. Second, the db1 basis decomposes the approximation coefficient, which is outcome of the first step, until maximum level. The pseudo code of multiresolution wavelet analysis is written below.

```
func dwtMultiDecompose(I:array[0 … r-1,
0 .. c-1])
        //First step for DB 4 basis
        for res = 1 to 2  do
              [A, H, V, D]_res=dwt (I,h,g)
              I = A_res;
          end for
         // Second step for DB1 basis
          j = size(A,1)
          c ← A/2^j
          g ← 2^j
          while g ≥ 2 do
               for row ←1 to g do
                    decStep(c[row, 1..g])
               end for
               for col ← 1 to g do
                    decStep(c[1..g, col])
               end for
               g ← g/2
          end while
          return (c)
     end func
```

The multiresolution of wavelet coefficients is denoted by ***c***. The pseudo code above will return the wavelet decomposition coefficient as shown below.



(a) (b) (c)

Figure 3. The output of multiresolution wavelet analysis psudocode: (a) original image, (b) the First step decomposition coefficients, (c) the second step decomposition coefficients.

Each face images is represented using small part of the second step decomposition coefficients which is called as holistic information. The holistic information is a compact and meaningful facial feature created by three steps: firstly, convert the frequency domain matrix coefficients to vector using row ordering technique; secondly, sort the vector descending using quick sort algorithm, finally truncate a small part of vector (i.e., less then 100 elements). Those processes are performed on both training and query (probe) face images but in the training process those are done one times.

In training process, the system calculates the mean of each facial feature class and global covariance using the mean and the covariance analysis and then save them as meaningful data for face classification as describe below.

Suppose Q is collection of wavelet domain facial matrix class, which is defined as $Q=[C_1, C_2, C_3,\ldots, C_c]$, where $C_i=[x_1, x_2, x_3,\ldots, x_n]$, x_j is holistic facial feature vector of j-th member of class C_i, c denote number of classes, and n denote number of member class C_i. It is easy to calculate the mean of each class and then placed it to matrix as $M=[\mu_1, \mu_2, \mu_3,\ldots, \mu_c]$, where $\mu_i = (1/N)\sum_{k=1}^{N} x_k$. Next, we can determine the global covariant using equation below:

$$\Lambda = \frac{1}{c}\sum_{i=1}^{c}\sum_{x_j \in C_i}\left(x_j - \mu_i\right)\left(x_j - \mu_i\right)^T ;\ j = 1, 2, 3, \ldots, n. \tag{9}$$

If assumed that the global covariance is multivariate normal distribution and the covariance matrix is not diagonal, the Bayesian classifies become minimum (Mahalanobis) distance classifier as written below.

$$F_c = \min\left[f_i, f_2, f_{3,} ..., f_c\right] \qquad (10)$$

Where f_i is

$$f_i = -\frac{1}{2}(x-\mu_i)^T \Lambda^{-1}(x-\mu_i) \qquad (11)$$

The minimum score is decided as the best likeness. This algorithm has some advantages in classifying the face image class, such as:

1. It is simple to classify face because it does not need to determine the eigen-values and vectors of covariance matrix as performed in the PCA and LDA.
2. When the new class member is added to the set, the proposed system just calculate its mean and covariance only, then placed the new class's mean into the matrix *M* and update the previous covariance matrix by adding with the new class's covariance.
3. The computation complexity is less than the computation complexity of PCA and LDA

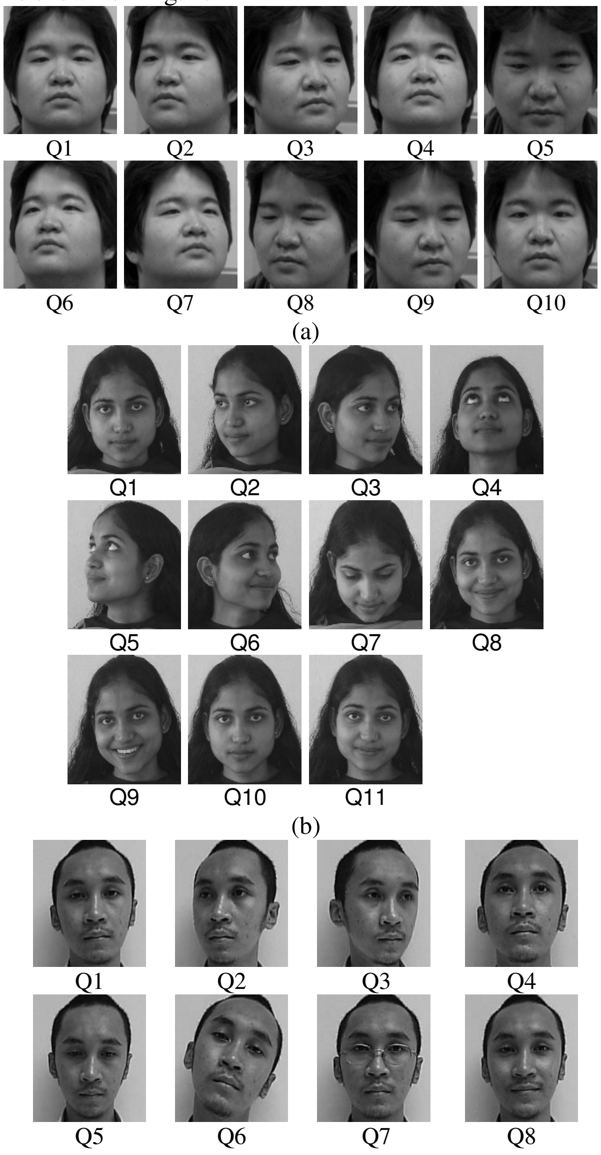
The weakness of the proposed method is the covariance matrix will be singular due to the high dimensional data and small number of training samples. To avoid the singular problem, we reduce the data dimensional of face image using multiresolution wavelets analysis. By keeping small part of transformed coefficients, the data dimensional can be reduced about 99.39% when the image size is 128 *x*128 pixels (i.e. we just keep less then 100 of 16384 coefficients).

The parameters used to know the effectiveness of the proposed method are success rate, training time, querying time, and receiver operating characteristics (ROC) curve. The training and querying times should be considered when the real-time system recognition is built

5. Experiments and Results

The experiments were carried out in four face databases: ITS-Lab. Kumamoto University database, EE-UNRAM database, Indian database [11], and ORL database [9,12], Each database has special characteristics.

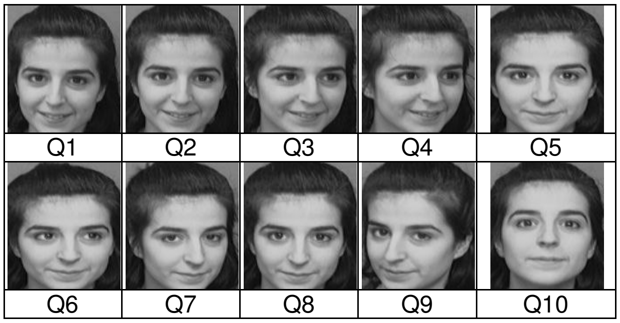
ITS-Lab database consists of 48 persons and each person has 10 pose orientations as shown on Fig. 4(a). The face was taken by Konica Minolta camera series VIVID 900 under varying light condition. EE-UNRAM database consists of 40 persons and each person has 8 pose orientations: looking front, looking left about 30^0, looking right about 30^0, looking up, looking down, and wearing accessory such as glasses. Indian database consists 61 persons (22 women and 39 men) and each person has eleven pose orientations: looking front, looking left, looking right, looking up, looking up towards left, looking up towards right, and looking down. Indian data base also included the emotions: neutral, smile, laughter, sad/disgust. ORL database is grayscale face database which was taken at different times, under varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All of the images were taken against a dark homogeneous background. Faces are subjects in an upright, frontal position (with tolerance for some side movement). ORL database is grayscale face database that consists of 40 person most of them are men. The face orientation of all databases is shown on Fig. 4.



(a)

(b)

(c)



(d)

Figure 4. Example of face poses: (a) ITS-Lab., (b) INDIA, (c) UNRAM, and (d) ORL. face database

The first experiment was performed to investigate the effect of facial vector size to the success rate. This was performed on ITS-Lab. database and 4 training face per person. The experiment was performed as follow. The first 4 images per person from the ITS-Lab database were selected to form a training face image set (total 192 face images). The rest of the face images (288 images) were used as query image (probe image). The test result is plotted on Fig. 4. Generally, the success rate slightly decreases when the size of facial feature goes to high size. The facial feature size, which is good enough to achieve high success rate (Fig. 5) and need short enough training and querying time (Fig 7.a and Fig. 7.b), ranges from 45 until 80 elements. This result can be achieved because the face image of ITS-Lab was taken by good camera and the face pose variations are not very large as shown on Fig. 4(a). Also note that the proposed method also shows higher success rate and less training and time query than PCA and LDA method.

The objective of the second experiment is to investigate the effect of number of face image training per person's class to the recognition rate. It was performed on ITS-Lab database, and 64 elements of facial features size. The result shows that the more face training per person's class was given the better success rate was achieved as plotted in Fig. 6. It means that the more face image is trained the better mean and covariance projection is gotten. The proposed methods show the best result among the other methods when the training is more than 2. We did not perform the test from 1 training face per person's class because the LDA and proposed-based face recognition will give the singular projection matrix when 1 face image per class is.

The time consumption of the proposed method is shown on Fig. 7. This result was gotten when the test was performed on ITS-Lab database, 4 training image per person's class, and the personal computer with Intel Pentium Celeron D 3.06 GHz, 1 GB RAM, and 160 GB hard disk. It shows that the proposed system needs less time in both training and querying process than other methods because it does not need eigen analysis.

The fourth experiment was performed to know the robust of the proposed method on several face databases. The test was carried out on 64 elements facial feature size and 4 faces training for each class. The difference of success rate among four databases is not more than 3 %, as shown in Fig. 8, it means that the proposed method is robust for tested face databases. Furthermore, the proposed method is superior to classical PCA and LDA.

There are two aspects which can be used to justify a good recognition system: first, how well the system can match image from the same people; second, how well the system distinguish images from different people the False Acceptance Rate (FAR). FAR is the success probability of unauthorized user to be falsely as accepted or recognized as legally user. FRR is the success probability legally registered user to be falsely

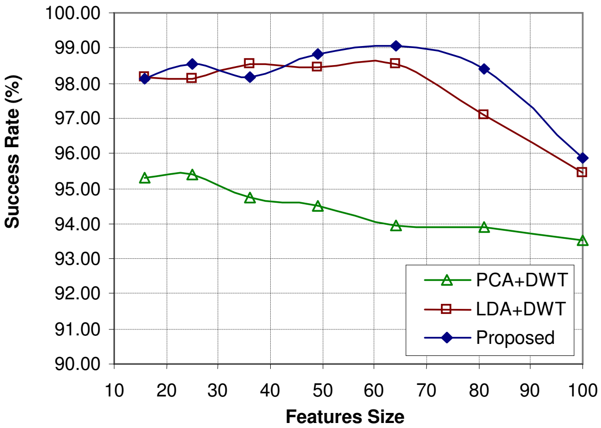


Figure 5. The success rate as function of features size.

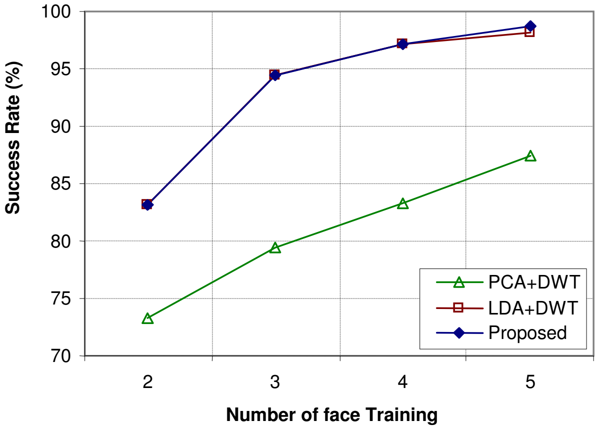
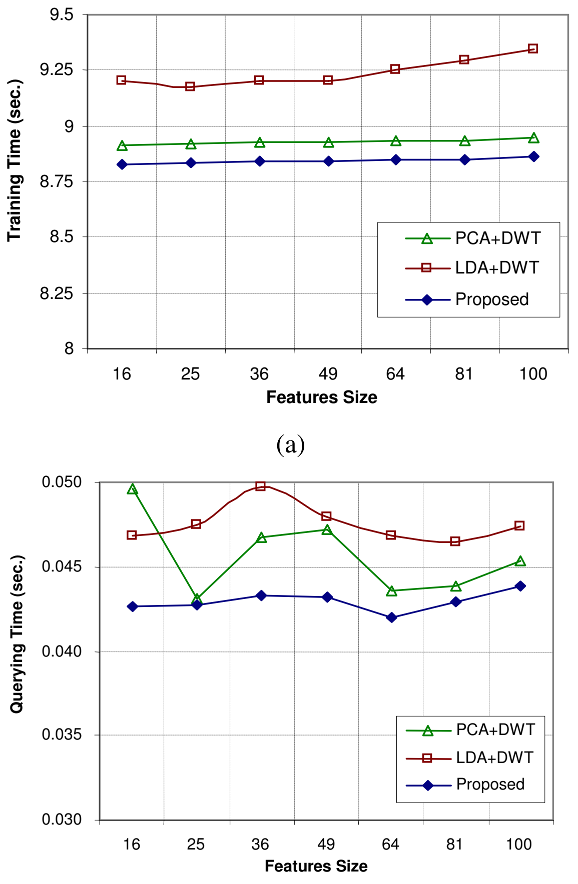


Figure 6. The effect of number of face training per class to success rate in the ITS-Lab face database.



(a)

(b)

Figure 7. The time consumptions comparison: (a) the training time and (b) the querying time as function of features size in the ITS-Lab face database.

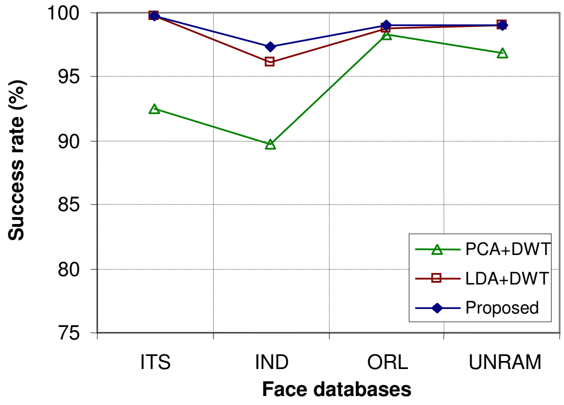
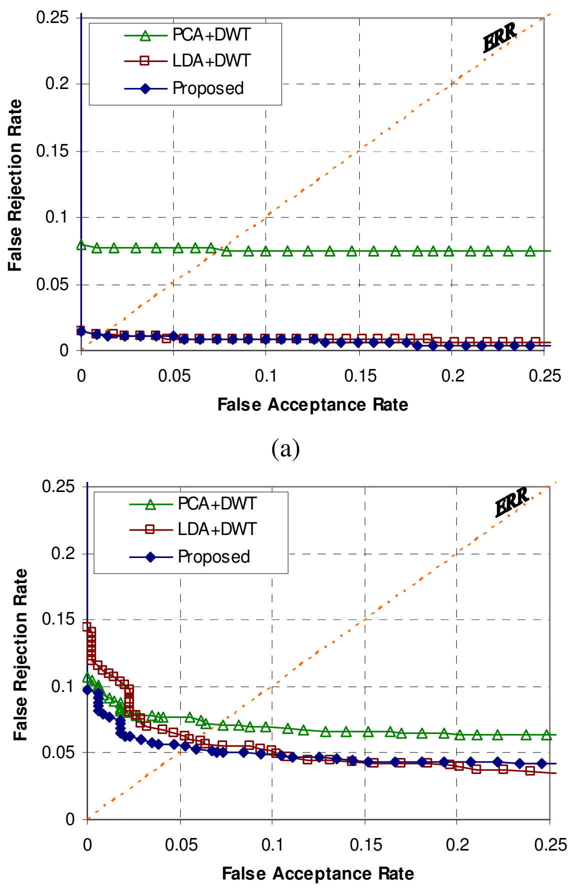


Figure 8. The robustness of success rate in the four face- databases.

[10].Based on this information we performed the last test for knowing the effectiveness of the proposed system. In this test, we will investigate two important parameters, namely the False Rejection Rate (FRR) and rejected by the system. If the value of FAR and EAR is equal then this point is called as Equal Error Rate (EER). The test was performed on four databases (i.e., ITS-Lab, INDIA, UNRAM, and ORL). Those of databases are subjected as predicted positive (known face) and the frontal face image of CVL database is subjected as predicted negative (unknown face). In this case, we add a threshold for the distance measure between features permit rejection of unknown face and verifications of those that are known. In other word, we sent an unknown face and a “claimed” identity to the system for verifications. If the distance between the face’s features to those of database image which it is being verified less than the given threshold, the claimed is accepted, otherwise, it is rejected. The system which performs perfect classification is denoted by 100% true positive rate and 0% false positive rate or the value of ERR is small or close to zero. The last experiment was performed on 36 elements facial feature size and 4 faces training per person’s class.

All of the experimental results show that the proposed method has good performance, robust to tested face databases, and need short training and querying times. This performance can be achieved because the holistic information of face image have good low-frequency resolution representation. In this case, the low frequency component is good enough for face image representation because most information of signal can be found in low frequency component. It can be described that if an image is transformed to frequency domain and removed the high frequency component, the reconstruction image will loss a little significant information. This phenomenon was successfully used for signal compression. Moreover, wavelet decomposition property gives advantage for features extraction, such as it has good capability to separate information signal to low frequency component and its coefficients is uncorrelated with other frequency indices.

The multiresolution wavelet decomposition is an efficient way of reducing the original data dimensional. In this paper we shows that the original data size can be reduced about 99.61% of original size (i.e., 64 elements of 16384 elements), while the success rate is high enough. Also note that holistic information of face image was obtained by fast wavelet transforms and the classifier just need mean of each face class and global covariance to classify an image to person’s class. Computational complexity of wavelet decomposition is linear with the number (*N*) of computed coefficients *(O(N)),* where *N* number of data. Therefore, our method needs short training and querying times.



(a)

(b)

Figure 9. ROC curve of proposed method compared to the other systems on: (a) ITS-Lab, (b) combination face database.

6. Conclusion

The alternative approach of human face image recognition has been successfully implemented which based on holistic/global information obtained by multiresolutioon wavelet analysis and minimum Mahalanobis classifier. It is an efficient way of reducing space requirement and computational load of LDA and PCA. The proposed method gives good performance, robust to face databases, need little training and querying times, and performs better classification which is shown by the smallest EER of other methods. Moreover, the proposed methods could overcome the retraining problem. Based on these results, we think that our method can be adopted for real-time face recognition. This process needs some improvements such as applying cluster face analysis to make group of face based on skin color detection and implementing the moment to detect the angel of face capture.

7. References

[1] Chellappa, R., Wilson, C., and Shirohey, S., “Human and Machine recognition of faces: A survey”, *In Proc. IEEE,* vol. 83, no. 5, pp. 705-740, 1995.

[2] Turk, M., and Pentland, A., "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol.3, no. 1, pp. 71-86, 1991.

[3] Lu, J., Plataniotis K.N., and Venetsanopoulus A.N., “Face Recognition Using LDA-Based Algorithmn”, *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 195-200, 2003.

[4] Chen, W., Er, Meng J., and Wu S., “PCA and LDA in DCT Domain”, *ELSEVIER Pattern Recognition Letter*, 26, pp. 2474-2482, 2005.

[5] Dai, Dao-Qing and Yuen P.C., “Wavelet Based Discriminant Analysis for Face Recognition”, *ELSEVIER Applied Mathematics and Computation*, 175, pp. 307-318, 2006.

[6] Huang, F. J., Zhou, Z., Zhang, H-J., and Chen, T., “Pose Invariant Recognition”, *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble France, pp. 245-250, 2000.

[7] Garcia C., Zikos G., and Tziritas G., “Wavelet Packet Analysis for Face Recognition”, *Image and Vision Computing*, 18, pp. 289-297, 2000.

[8] Hafed, Ziad M., and Levine, Martin D., “Face Recognition Using the Discrete Cosine Trasnforms”, International Journal of Computer Vision, 43(3), pp. 167-188, 2001.

[9] Samaria F., and Harter A., "Parameterization of a stochastic model for human face identification" *2nd IEEE Workshop on Applications of Computer Vision,* Sarasota (Florida), pp. 138-142, 1994.

[10] Feng, G C., Yuen, P C., and Dai D Q., “ Human Face Recognition Using PCA on Wavelet Subband”, *Jounal of Electronic Imaging,* 9 (2), pp. 226-233, 2000.

[11] Franc Solina, Peter Peer, Borut Batagelj, Samo Juvan, Jure Kovac, “Color-Based Face Detection in the 15 Seconds of Fame Art Installation In: Mirage 2003”, Conference on Computer Vision/Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects, INRIA Rocquencourt, France, Wilfried Philips, Rocquencourt, INRIA, 2003, pp. 38-47.

[12] http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase

[13] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Spherical PCA with Euclideanization

Jun Fujiki
National Institute of Advanced Industrial
Science and Technology
Tsukuba-Central 2, 1–1–1 Umezono,
Tsukuba-shi, Ibaraki 305–8568, JAPAN
Email: jun-fujiki@aist.go.jp

Shotaro Akaho
National Institute of Advanced Industrial
Science and Technology
Tsukuba-Central 2, 1–1–1 Umezono,
Tsukuba-shi, Ibaraki 305–8568, JAPAN
Email: s.akaho@aist.go.jp

***Abstract*—To measure the similarity between two high dimensional vector data, correlation coefficient is often used instead of Euclidean distance. For this purpose, the high dimensional vectors are mapped into hyperspherical points by normalization and the distance is measured as the length along geodesic on the hypersphere. On the hypersphere, the Pythagorean theorem does not hold and it makes difficult to apply data analysis methods defined in Euclidean space, such as principal component analysis, to hyperspherical data. In this paper, we propose spherical principal component analysis to analyze hyperspherical data on the criterion of spherical least squares. Spherical principal component analysis is defined as lower dimensional great hypersphere fitting to higher dimensional hyperspherical data. We also propose the approximation of spherical principal component analysis because it is a kind of non-linear minimization.**

I. Introduction

Recently, data analysis on hypersphere is getting of great significance.

In computer vision, omnidirectional camera such as catadioptric camera and fish-eye camera is widely used for robot navigation, surveillance system and so on. To understand and to unify omnidirectional cameras, spherical camera is defined [2], [4], [8], [10], [13], [15], [17], [18]. On the spherical camera, points and lines in 3-dimensional space are projected to points and great circle on 2-dimensional sphere (S^2), respectively. Then analysis of S^2 data is needed in computer vision. Camera motions are also identified as a sequence of spherical points which are corresponding to the front directions of cameras. Since we do not need to consider the horizontal directions of cameras and the distances from cameras to an object by rotating and scaling of the size of images appropriately. Therefore, smoothing the camera motion is realized by fitting a small or great circle to an sequence of spherical points[6], [7], [14].

In finance engineering, the correlation matrix, consists of inner products of each of two hyperspherical data, plays an important role in interest model[3]. To reduce the computational cost of simulating the model, dimension reduction of the correlation matrix, which is the fitting of a great hypersphere to the data on the unit hypersphere, is an important process[12]. In the fields of bioinformatics and data mining, when analyzing gene expression profile data and processing text data, we can regard the data as hyperspherical points[16]. These examples show that data analysis on hypersphere is getting of great significance recently.

In this paper, we propose **principal component analysis (PCA)** for spherical data, called **spherical PCA (SPCA)**, to analyze hyperspherical data on the criterion of **least squares (LS)**. SPCA is defined as low dimensional great hypersphere fitting to the high dimensional hyperspherical data. We also propose the approximation of SPCA, called Euclideanization, because SPCA is a kind of non-linear minimization. We also refer two methods to reduce the dimension of data space, that is, extracting a low-dimensional small hyperspherical structure embedded in high-dimensional hypersphere. The former method is called **SLS-ESG**, abbreviation of **spherical LS (SLS)** by Euclideanization for **stereographic projection (SGP)** [9], and the latter method is called **sequential dimension reduction (SDR)**[9].

II. Spherical PCA (SPCA)

One purpose of PCA is compressing data dimension to remove the noise on the data. Therefore SPCA should be defined as the data compression method for the hyperspherical data. From this point of view, SPCA is the low dimensional fitting to the high dimensional hyperspherical data.

A. Spherical Least Squares (SLS)

For the hyperspherical data, the fitting error should be based on not the Euclidean distance but the distance along geodesic (angle) between the data and fitting space.

The distance between the hyperspherical data $\boldsymbol{x}_i$ and some specific subset of hypersphere S is defined as $\mathrm{dist}(\boldsymbol{x}_i, S) = \min_{\boldsymbol{y}\in S} \cos^{-1}(\boldsymbol{x}_i^\top \boldsymbol{y})$. When the best fitting is defined as the minimizer of the sum of square distance, the fitting error is given by $\sum\{\mathrm{dist}(\boldsymbol{x}_i, S)\}^2$. We call the minimization of sum of square of the distance along geodesic of hypersphere, SLS.

B. Spherical PCA (SPCA)

In spherical geometry, a line on the sphere is equivalent to a great circle of the sphere. Then SPCA is defined as *low dimensional great hypersphere fitting* to the hyperspherical data. In the low dimensional great hypersphere fitting, we should pay attention to the following fact.

Fact: *Let S^i and S^j $(i < j)$ be the best i-dimensional great hypersphere fitting and j-dimensional great hypersphere fitting, respectively. Generally, $S^i \not\subset S^j$ holds.*

Figure 1 is the example of SPCA, that is great hypersphere fitting of the data and the blue circle is the result of S^1 fitting. Red point is the result of S^0 fitting, that is, the center of the data which minimizes the square distance along geodesic between data. It is easy to find that the center is not on the best S^1 fitting, that is $S^0 \not\subset S^1$.

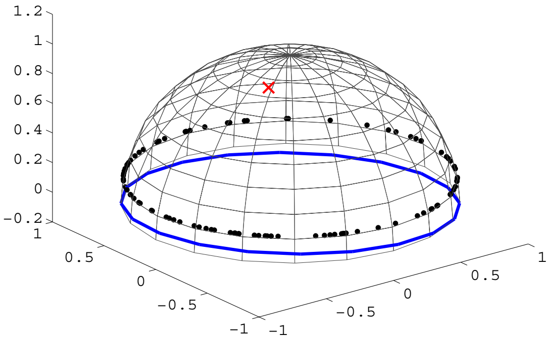


Fig. 1. S_0 fitting (red) does not belongs to S_1 fitting (blue).

On the ordinal PCA in the Euclidean space, the Pythagorean theorem ensures the best i-dimensional hyperplane fitting is always belongs to the best j-dimensional hyperplane fitting when $i < j$ by using the criterion of LS. However, the Pythagorean theorem does not hold on the sphere by using the distance along geodesic, then the fact holds.

III. Hypersphere fitting

For SPCA, it is enough to consider the great hypersphere fitting, but also the small hypersphere fitting is also considered in the paper. Of course the great hypersphere fitting is the special case of the small hypersphere fitting.

Let $\boldsymbol{R}^{n+1}$ be an $n+1$-dimensional Euclidean space, and O be its origin. The n-dimensional unit hypersphere is defined as

$$S^n = \{\mathrm{X} \,|\, \mathrm{OX} = ||\boldsymbol{x}|| = 1, \boldsymbol{x} \in \boldsymbol{R}^{n+1}\}. \quad (1)$$

We consider a d-dimensional small hypersphere on the unit hypersphere S^n. Generally, hypersphere is represented as the intersection between S^n and a $d+1$-dimensional Euclidean space $\boldsymbol{E}^{d+1}$.

Let C be the center of the small hypersphere, $\overrightarrow{\mathrm{OC}} = \boldsymbol{c}$ is perpendicular to $\boldsymbol{E}^{d+1}$ (when $\mathrm{O} \in \boldsymbol{E}^{d+1}$, $\mathrm{C} = \mathrm{O}$).

Let one of the orthonormal basis of associated linear space $\boldsymbol{E}^{d+1}$ be $\langle \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{d+1} \rangle$, and $R = \begin{pmatrix} \boldsymbol{r}_1 & \cdots & \boldsymbol{r}_{d+1} \end{pmatrix}$ be the $n+1 \times d+1$ matrix arraying the orthonormal basis (Generally, orthonormal coordinate matrix R is not unique to represent the same Euclidean space), the d-dimensional small hypersphere is parameterized as

$$\left\{ \mathrm{X} \,\middle|\, \mathrm{X} = \mathrm{C} + R\boldsymbol{t}, \boldsymbol{t} \in \boldsymbol{R}^{d+1}, ||\boldsymbol{t}|| = \alpha \right\} \quad (2)$$

where $\alpha = \sqrt{1 - ||\boldsymbol{c}||^2}$, which is the radius of the small hypersphere. The small hypersphere is described only by the C and R, but radius α is also used to represent the small hypersphere as $\alpha S^d(\mathrm{C}, R)$ for convenience. When we consider only its radius α, we abbreviate the representation to αS^d. Note that a great hypersphere $1S^d = S^d$ is described only by the R.

The aim of the paper is dimension reduction of the data distributed on the unit hypersphere S^n on $\boldsymbol{R}^{n+1}$, and compress the data to the data on the great and/or small hypersphere $\alpha S^d(\mathrm{C}, R)$. On small hypersphere fitting, C (,α) and R are chosen so as to minimize the square distance along geodesic. And on great hypersphere fitting, only R is chosen so as to minimize the square distance along geodesic.

Let $\boldsymbol{z}_f$ ($||\boldsymbol{z}_f|| = 1; f = 1, \ldots, F$) be data on S^n, and $r_f = \mathrm{dist}(\boldsymbol{z}_f, \alpha S^d(\mathrm{C}, R))$, there holds

$$r_f = \cos^{-1}(\boldsymbol{z}_f^\top \boldsymbol{c} + \alpha ||R^\top \boldsymbol{z}_f||). \quad (3)$$

To estimate the best small and/or great hypersphere under LS along geodesic is to minimize

$$J = \frac{1}{2} \sum_{f=1}^{F} r_f^2, \quad (4)$$

which is a function of C (,α) and R. This is the hypersphere fitting under SLS. However, it is not easy to solve the hypersphere fitting under SLS because the differentiation of $||R^\top \boldsymbol{z}_f||$ with respect to R is needed to apply the Newton's method for minimizing the equation (4) or some other minimizing technique computing gradient of the cost function. Therefore, we propose the approximations of SLS.

IV. Euclideanization

An approximation of SLS, called **Euclideanization** of hypersphere, is presented. Euclideanization is the adjustment of the metric of projected space to keep the metric of the original space as well as possible.

A. Example to understand Euclideanization

To understand the general case of Euclideanization, fitting small circle to data on the sphere ($n = 2$, $d = 1$) is discussed for example. SGP is used to map S^2 to $\boldsymbol{E}^2$.

The characteristics of the data are as follows: we regard the true small circle as latitude of the globe. We determine the arc of the small circle by setting some longitude range (the angle unit is radian) on the latitude. We generate 100-data from uniform distribution on the arc of the small circle and adding the Gauss noise of 0.01-rad. standard deviation along meridional direction. We determine $\boldsymbol{z}_f$ as rotating the 100-data by 3-dimensional rotation matrix. We estimate the best small circle from the 100-data under some criterion.

The results are shown as the true small circle as an latitude for convenience. The green line represents the true small circle and the red line represents the estimated small circle.

B. Hyperplane fitting

The easiest way to estimate small hypersphere is as follows: Fitting $d+1$-dimensional hyperplane to $\boldsymbol{z}_f$ by LS criterion, and the estimated small hypersphere is the intersection between the hyperplane and the unit hypersphere (Gray et al.[11] uses this estimation as the initial value of their algorithm).

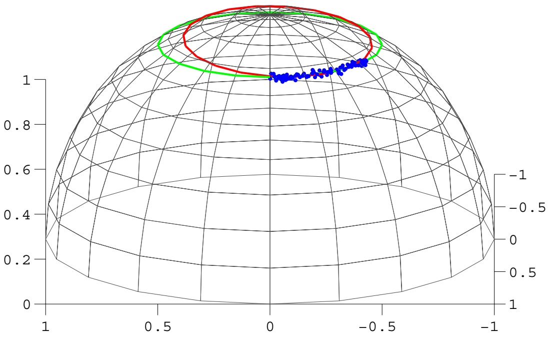


Fig. 2. Hyperplane fitting: latitude $\pi/3$, longitude range $\pi/3$.

Figure 2 shows the estimation of the small circle by plane fitting. When a longitude range is small, the estimation tends toward the tangent plane of the unit sphere because the sphere itself is approximated by the tangent plane.

C. Estimation by SGP

To overcome the problem that an estimation tends toward a tangent hyperplane, we try to estimate a small circle by SGP. SGP has a property that the small circle on the unit sphere is projected to the circle on the projective plane. Therefore, we estimate the small circle by fitting the circle on the projective plane under LS criterion and project the circle back to the unit sphere by SGP. Figure 3 shows the estimation by SGP for the same data as Fig.2. By SGP, the tendency toward a tangent hyperplane is reduced.

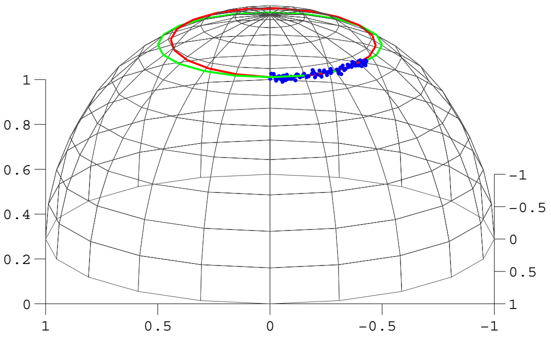


Fig. 3. Estimation by SGP: latitude $\pi/3$, longitude range $\pi/3$

The estimation method referred in this subsection is as follows: first, fitting d-dimensional small hypersphere or d-dimensional hyperplane on the n-dimensional projective hyperplane under LS criterion. Next, mapping the fitted d-dimensional small hypersphere or d-dimensional hyperplane onto the n-dimensional hypersphere S^n by the inversion.

D. Euclideanization for SGP (ESG)

The estimation method referred in the previous subsection sometimes gives a poor result. Then, we propose the **weighted least squares (wLS)** where the weights are determined by the changing of metric under SGP. Simply speaking, the enlargement rate of some point on the hypersphere by SGP is k, the weights are determined as k^{-2}. Like this, the geodesic distances are approximately replaced with weighted Euclidean distance. And the weights are determined by changing of metric[1]. We call the weighting **Euclideanization**.

In this subsection, we propose a method to estimate the approximation of SLS by **Euclideanization for SGP** (**ESG**). We call the method the **SLS-ESG**.

First, Fig.4 shows the estimation by ESG for the same data as Fig.2. For this dataset, the estimation by ESG gives almost the same result as the LS without weighting (Fig.3).

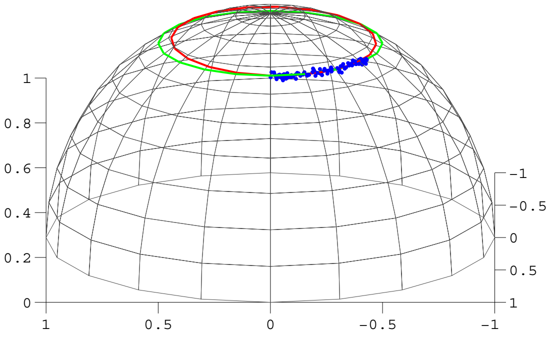


Fig. 4. Estimation by ESG: latitude $\pi/3$, longitude range $\pi/3$

However, Fig.5 (up) shows a very poor estimation by LS without weights. The reason of such a poor estimation is that the existence of data around the north pole. The enlargement rate of the area by SGP is getting higher when the area is getting closer to the north pole. When we apply ordinary LS to the data on the projective plane (Fig5 (down)), the estimation tends to fit the data far from the origin of the projective plane (the image of the south pole).

To overcome this problem, we set weights to revise the metric (enlargement by SGP). As shown in the next section, the weight of the datum of the distance r from the origin on the projective plane is calculated as $(1+r^2/4)^{-1}$.

Figure 6 shows that the estimation by SLS-ESG is improved. This is why we propose the Euclideanization.

V. Equi-directional projection (EDP)

Many of the projection from hypersphere surface to its Euclidean tangent space are unified by **equi-directional projection (EDP)** of hypersphere.

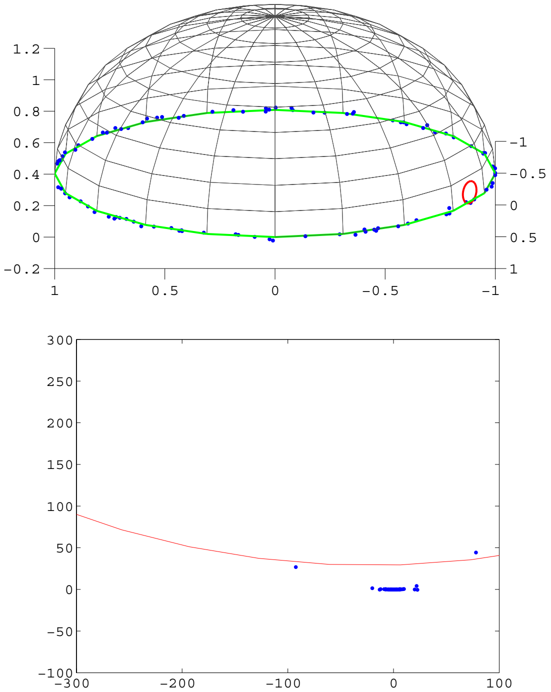


Fig. 5. Estimation by SGP and usula LS on sphere (up) and projective plane (down), latitude 0, longitude range 2π

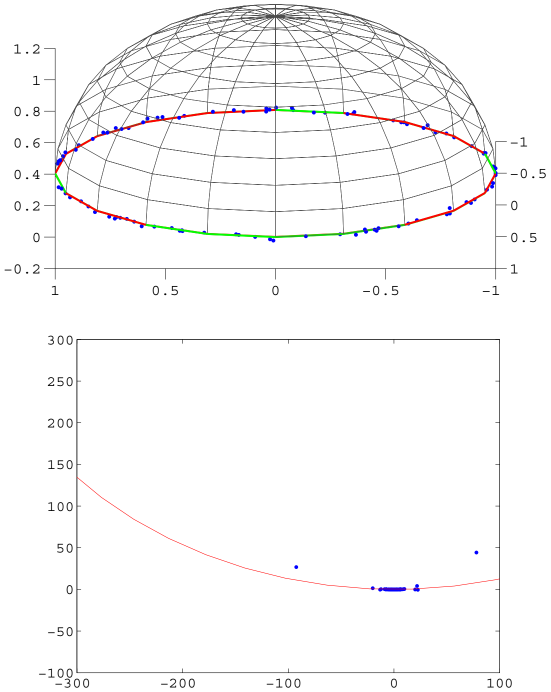


Fig. 6. Estimation by SLS-ESG on sphere (up) and projective plane (down), latitude 0, longitude range 2π

Let $\widetilde{\boldsymbol{o}}_{\pm}$ be defined as $\widetilde{\boldsymbol{o}}_{\pm} = (\boldsymbol{0}_n^{\top}, \pm 1)^{\top}$, that is north and/or south pole of S^n, and let $\widetilde{\boldsymbol{x}} = (\boldsymbol{x}^{\top}, x_{n+1})^{\top}$.

Let $\boldsymbol{E}_{-}^{n}$ be the n-dimensional Euclidean tangent space at $\widetilde{\boldsymbol{o}}_{-}$.

The map from $S^n - \{\widetilde{\boldsymbol{o}}_{+}\}$ to $\boldsymbol{E}_{-}^{n}$ is called EDP of hypersphere iff the map is represented as

$$\widetilde{\boldsymbol{x}} \mapsto f(x_{n+1})\boldsymbol{x} = r(x_{n+1}) \cdot \frac{\boldsymbol{x}}{||\boldsymbol{x}||} \tag{5}$$

where $f(x)$ satisfies $f(-1) = 1$ and $f(x) > 0$, and the distance between south pole and the projection point $r(x) = f(x)\sqrt{1-x^2}$ satisfies $\frac{dr}{dx} > 0$, which is the monotonous increasing function of r.

To compute the weights of Euclideanization, we should pay attention to the changing of volume element by the EDP of the hypersphere. The ratio of volume element is equivalent to the Jaccobian of the mapping as

$$J = \{f(x_{n+1})\}^n \left\{ \frac{f'(x_{n+1})(1-x_{n+1}^2)}{f(x_{n+1})} - x_{n+1} \right\}. \tag{6}$$

Then the length is expected to enlarged $J^{\frac{1}{n}}$ times by the EDP. This means the weight of Euclideanization for EDP is $J^{-\frac{1}{n}}$.

For **orthographic projection (OGP)**,

$$f(x) = 1, \quad J^{-\frac{1}{n}} = (1-r^2)^{-\frac{1}{2n}}.$$

For **stereographic projection (SGP)**,

$$f(x) = \frac{2}{1-x}, \quad J^{-\frac{1}{n}} = \left(\frac{r^2}{4} + 1\right)^{-1}.$$

For **azimuthal equidistant projection (AEP)**,

$$f(x) = \frac{\pi - \cos^{-1} x}{\sqrt{1-x^2}}, \quad J^{-\frac{1}{n}} = \left(\frac{\sin r}{r}\right)^{\frac{n-1}{n}}.$$

For **gnomic projection (GMP)**,

$$f(x) = -x^{-1}, \quad J^{-\frac{1}{n}} = (r^2+1)^{-\frac{n+1}{2n}}.$$

For **equisolidangle projection (ESP)**, the weights are constant 1 ($J = 1$) because ESP does not change the volume element. The representation of ESP is

$$f(x) = \frac{\{-n \int_{\pi}^{\cos^{-1} x} \sin^n \phi \, d\phi\}^{\frac{1}{n}}}{\sqrt{1-x^2}}. \tag{7}$$

The ESP is equivalent to the AEP for $n = 1$. The concrete representation of the ESP for $n \geq 3$ is so complicated that x_{n+1} is difficult to represent by r, then the inverse mapping from projected space to S^3 is difficult to compute.

VI. Euclideanization for EDP

In this subsection, we compare the estimation of SPCA with Euclideanization to without Euclideanization. we discuss the case $n = 2$ and $d = 1$, that is, fitting great circle to the data on the sphere.

The characteristics of the data are as follows: we regard the true great circle as equator of the globe. We generate 100-data from the great circle as their longitudes are distributed from Gauss distribution of its mean and standard deviation are 0 and ω-rad., respectively. We add the Gauss noise of σ-rad. standard deviation along x, y and z-axes, and normalize the data as their norm are equal to 1. We determine z_f as rotating the 100-data by 3-dimensional rotation matrix. We estimate the best great circle fitting (S^1 fitting) and the best point fitting (S^0 fitting) from the 100-data under some criterion.

Figure 7∼10 show the S^0 and S^1 fitting error for OGP, OGP cut off at $\pi/3$ (OGPc), SGP, AEP, GMP and ESP. Fitting error is the average of 100-trial. The OGP cut off at $\pi/3$, which means using only each datum of its colatitude is less than $\pi/3$, that is, using only each datum of its latitude is greater than $\pi/6$. Blue bar in the graphs represents error of estimation with Euclideanization and red bar in the graphs represents error of estimation without Euclideanization.

When the data are distributed short range of equator (Fig. 7 and 8), Euclideanization yields worse S^0 fitting than without Euclideanization (Fig. 7). However, the S^0 fitting errors are sufficiently small and we can say the estimation with Euclideanization and without Euclideanization are almost the same. Figure 8 shows that there is little difference between the estimation with Euclideanization and without Euclideanization in S^1 fitting.

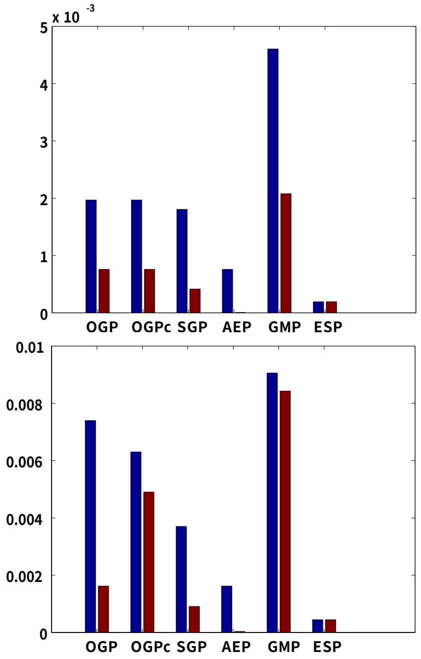


Fig. 7. S^0 fitting error; $\omega = 0.25$, $\sigma = 0.05$(up) and $\sigma = 0.2$(down).

When the data are distributed wide range of equator, (Fig. 9 and 10), Euclideanization yields better S^0 fitting than without Euclideanization except OGP (Fig. 9).

Especially, Euclideanization for SGP and GMP yield good performance. The feature of these two projection is the enlargement rate $J^{\frac{1}{n}}$ is getting to infinity when the point is getting far from the south pole. This means a slight noise of the points far from the south pole is enlarged to large noise and it yields worse estimation without Euclideanization. In such a case, Euclideanization is very important for estimation.

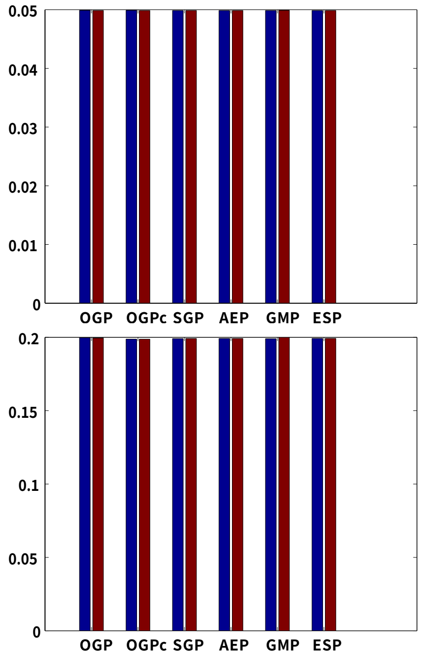


Fig. 8. S^1 fitting error; $\omega = 0.25$, $\sigma = 0.05$(up) and $\sigma = 0.2$(down).

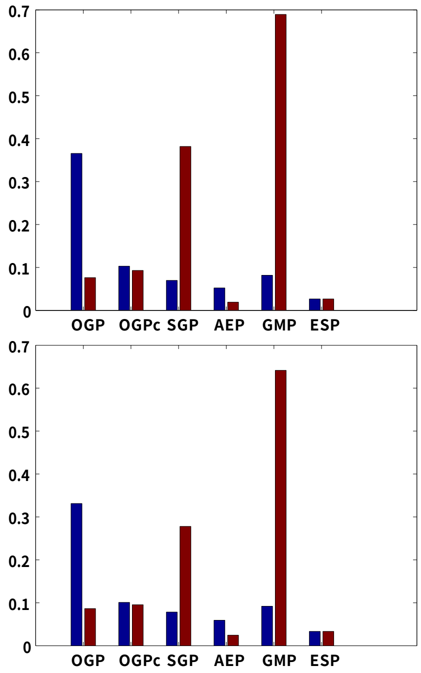


Fig. 9. S^0 fitting error; $\omega = 1$, $\sigma = 0.05$(up) and $\sigma = 0.2$(down).

For OGP, Euclideanization yields worse S^0 fitting than without Euclideanization. The feature of the orthogonal projection is the enlargement rate $J^{\frac{1}{n}}$ is getting close to 0 when the point is getting far from the south pole. This means the large noise of the points far from the south pole is neglected and it yields worse estimation with Euclideanization. In such a case, Euclideanization is not suitable for estimation.

Figure 10 shows that there is little difference between the estimation with Euclideanization and without Euclideanization in S^1 fitting except without Euclideanization for SGP. The reason of the worse estimation is not sure, but Euclideanization gives better performance for SGP.

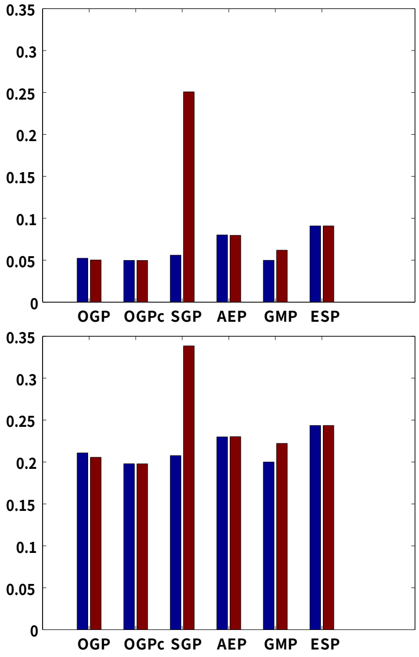


Fig. 10. S^1 fitting error; $\omega = 1$, $\sigma = 0.05$(up) and $\sigma = 0.2$(down).

Figure 11∼16 show the S^0 estimation and S^1 estimation of the data when the data are distributed wide range of equator. In these figures, red lines denote estimation of SLS by Newton's method, green lines denote estimation of EDP without Euclideanization, and blue lines denote estimation of EDP with Euclideanization. From these figures, Euclideanization derives stable estimation for all EDP.

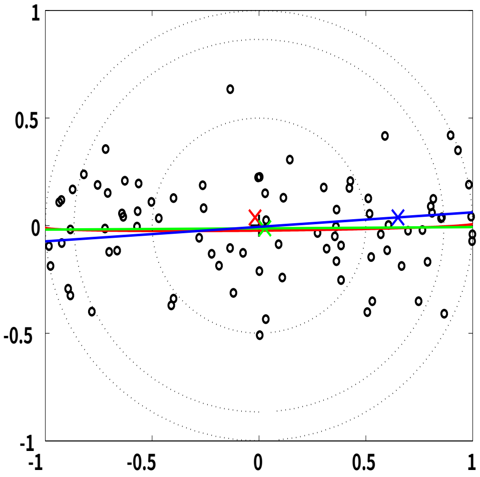


Fig. 11. Orthographic; $\omega = 1$, $\sigma = 0.2$.

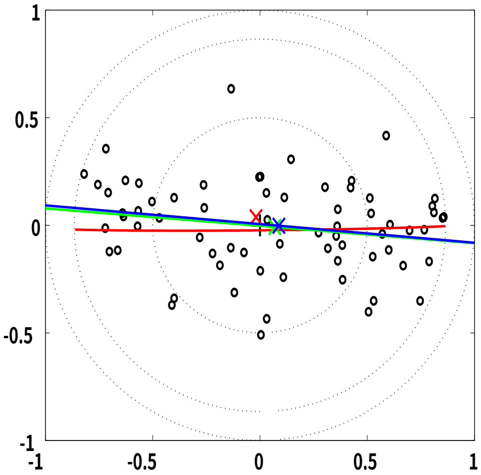


Fig. 12. Orthographic with cutoff at $\pi/3$; $\omega = 1$, $\sigma = 0.2$.

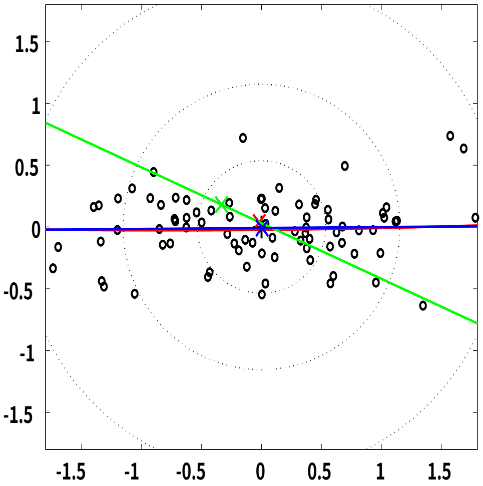


Fig. 13. Stereographic; $\omega = 1$, $\sigma = 0.2$.

VII. Euclideanization for SGP (ESG)

In this section, we highlight to **Euclideanization for SGP (ESG)** because SGP has a property that the small hypersphere on the unit hypersphere is projected to the hypersphere on the projective hyperplane. Euclidean space is regarded as hypersphere of its radius is infinity, then many data analysis method for Euclidean space is applicable for hypersphere by applying these data analysis method to the projective hyperplane. Then, the approximation of SPCA is realized by applying Euclidean PCA to the projective hyperplane. we call the method **SPCA by ESG (SPCA-ESG)**.

A. The algorithm of SPCA-ESG

The algorithm of SPCA-ESG under the criterion of SLS (SPCA-ESG-SLS) is as follows:

(1) Define the fitting dimension d.

(2) Map the data on the unit hypersphere S^n onto the data on the projective plane Π^n by SGP.

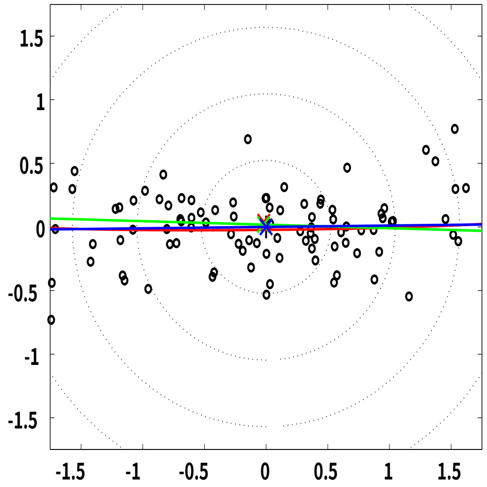


Fig. 14. Azimuthal equidistant; $\omega = 1$, $\sigma = 0.2$.

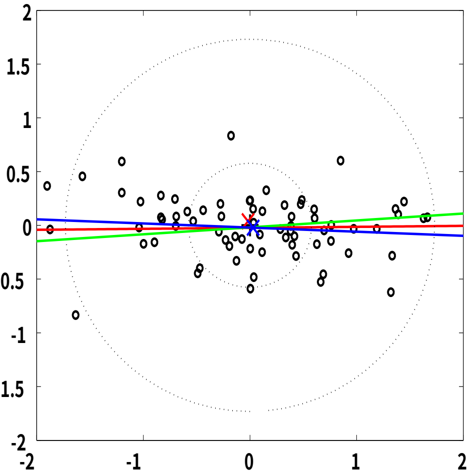


Fig. 15. Gnomic; $\omega = 1$, $\sigma = 0.2$.

(3) Fit $d+1$-dimensional Euclidean space $\boldsymbol{E}^{d+1}$ to the data on the projective plane Π^n by wLS.

(4) Project the data on Π^n onto $\boldsymbol{E}^{d+1}$ by OGP.

(5) Fit d-dimensional hypersphere for the data on $\boldsymbol{E}^{d+1}$ by wLS.

(6) Map the d-dimensional hypersphere on Π^n onto S^n by SGP, which is the estimated small hypersphere αS^d.

(7) Compute the estimation of $\boldsymbol{z}_f$ by projecting onto αS^d.

Note that the projection of $\boldsymbol{z}_f$ onto αS^d is the mapping from $\boldsymbol{z}_f$ to $\boldsymbol{c} + \frac{\alpha}{||R^\top \boldsymbol{z}_f||} R R^\mathrm{T} \boldsymbol{z}_f \in \alpha S^d$ which minimizes the distance along geodesic between $\boldsymbol{z}_f$ and the point on αS^d.

VIII. Sequential Dimension Reduction (SDR)

A. One-dimension reduction

The dimension of the orthogonal complement of R of $n-1$-dimensional hypersphere $\alpha S^{n-1}(\mathrm{C}, R)$ on S^n is one.

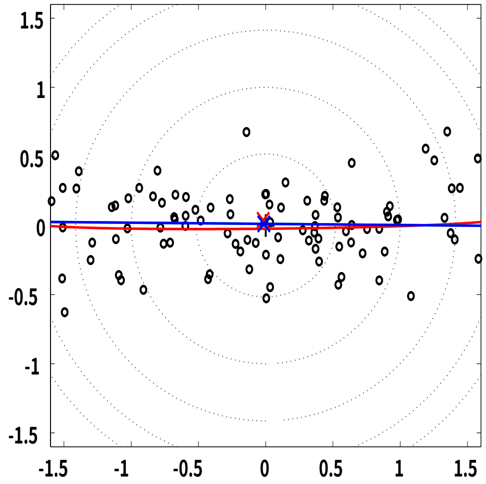


Fig. 16. Equisolidangle; $\omega = 1$, $\sigma = 0.2$.

Let the normal basis of the orthogonal complement be $\boldsymbol{\lambda}$ and define two angles $\psi_f = \cos^{-1}(\boldsymbol{z}_f^\top \boldsymbol{\lambda})$ and $\mu = \cos^{-1} ||\boldsymbol{c}||$, there holds $r_f = \psi_f - \mu$. Then the cost function J can be minimized by Newton's method.

We consider the minimization of non-negative function J parameterized by $\boldsymbol{\alpha}$ under the constraint $g = 0$.

Let the values of J, g and $\boldsymbol{\alpha}$ at the n-th iteration be J_n, g_n and $\boldsymbol{\alpha}_n$ respectively.

By using the second order Taylor expansion of J around J_n, and the first order Taylor expansion of g around g_n, the parameter $\boldsymbol{\alpha}$ is updated by

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n - H_n^{-1}\left\{\boldsymbol{q}_n + \frac{g_n - \boldsymbol{b}_n^\top H_n^{-1}\boldsymbol{q}_n}{\boldsymbol{b}_n^\top H_n^{-1}\boldsymbol{b}_n}\boldsymbol{b}_n\right\} \qquad (8)$$

where $\boldsymbol{q}_n$, H_n and $\boldsymbol{b}_n$ are the quantities of $\frac{\partial J}{\partial \boldsymbol{\alpha}}$, $\frac{\partial^2 J}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^\top}$ and $\frac{\partial g}{\partial \boldsymbol{\alpha}}$ at $\boldsymbol{\alpha} = \boldsymbol{\alpha}_n$, respectively.

We minimize eq.(4) by Newton's method with constraint $g = \boldsymbol{\lambda}^\top \boldsymbol{\lambda} - 1$ in the parameter space $\widetilde{\boldsymbol{\lambda}} = (\boldsymbol{\lambda}^\top, \mu)^\top$.

The updates of parameters are computed by the following values: (Gray et al.[11] gave the case $n = 2$ and $d = 1$)

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{\lambda}} &= -\sum_f \frac{r_f}{\sin \psi_f} \boldsymbol{z}_f, \quad \frac{\partial J}{\partial \mu} = -\sum_f r_f, \\
\frac{\partial^2 J}{\partial \boldsymbol{\lambda} \partial \boldsymbol{\lambda}^\top} &= \sum_f \frac{\sin \psi_f - r_f \cos \psi_f}{\sin^3 \psi_f} \boldsymbol{z}_f \boldsymbol{z}_f^\top, \\
\frac{\partial^2 J}{\partial \boldsymbol{\lambda} \partial \mu} &= \left(\frac{\partial^2 J}{\partial \mu \partial \boldsymbol{\lambda}^\top}\right)^\top = \sum_f \frac{1}{\sin \psi_f} \boldsymbol{z}_f, \\
\frac{\partial^2 J}{\partial \mu \partial \mu} &= \sum_f 1 = F, \quad \frac{\partial g}{\partial \widetilde{\boldsymbol{\lambda}}} = 2\begin{pmatrix} \boldsymbol{\lambda} \\ 0 \end{pmatrix}.
\end{aligned} \qquad (9)$$

B. Sequential Dimension Reduction (SDR)

We consider the sequence of small hyperspheres as

$$S^n \supset \alpha_1 S^{n-1} \supset \alpha_2 S^{n-2} \supset \cdots \supset \alpha_{n-d} S^d, \qquad (10)$$

and SDR estimates from $\alpha_1 S^{n-1}$ to $\alpha_{n-d} S^d$ sequentially. In each estimation, initial value of Newton's method is given by SLS-ESG[7]. To avoid the cumulative errors by a sequential procedure, z_f's are directly projected onto $\alpha_k S^{n-k}$ to estimate $\alpha_{k+1} S^{n-k-1}$.

IX. Experiment for small hypersphere fitting

We generated 100 data from 5-dimensional hypersphere on S^{20}. The data are sampled from uniform distribution on the 5-dimensional hypersphere. We add the 0-mean Gauss noise of its variance 0.01 for each axis and normalize data by divided by its norm. The hyperspherical data $\boldsymbol{z}_f$ which we use in experiments are the rotation of the normalized data by some 20-dimensional rotation matrix.

Figure 17 shows the dimension of estimated small hypersphere against the mean square error along geodesic. The solid line denotes the mean square error for SDR, and dotted line denotes the mean square error for SLS-ESG.

From 20-dimension to 5-dimension which is the true dimension, both methods give good performance. However, below the true dimension, which is over-compression of the data, the error raised rapidly. In this case, SLS-ESG gives a better estimation than SDR. We can compute the mean square error only by the observed data, we can estimate the "true" dimension of data only by the observed data.

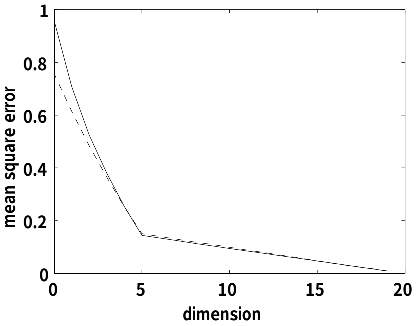


Fig. 17. Training error against dimension reduction: solid line is for SDR and dotted line is for SLS-ESG

Figure 18 shows the mean square error of 10^4-test data for 100 trials. Both methods give good performance for test data.

X. Future

From the proposed methods, following points are suggested:

(1) LS along geodesic for n-dimensional metric manifold is resolved to the wLS in Euclidean space when we can define the continuous and differentiable bijection from the manifold to Euclidean space. That is, Euclideanization is applicable.

(2) The clustering on S^n is resolved to the clustering on $\boldsymbol{E}^n$ by Euclideanization.

(3) Euclideanization is putting weights for data points. However, we should consider the weights not only for data points but also the geodesic from the data points to the low dimensional structure for fitting. From this point of view, the proposed Euclideanization should be called **0-th order**, and we should investigate it to the **1-st order**.

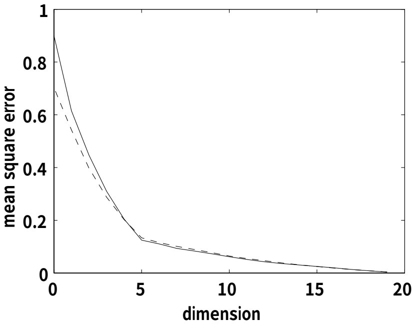


Fig. 18. Test error against dimension reduction: solid line is for SDR and dotted line is for SLS-ESG

References

[1] S. Akaho, "Curve fitting that minimizes the mean square of perpendicular distances from sample points," In Proc. of SPIE93, Vision Geometry II, Vol.2060, (1993).

[2] S. Baker and S. K. Nayar, "A theory of catadioptric image formation," In Proc. of IEEE International Conference on Computer Vision (ICCV98), pp.35–42, Jan. 1998.

[3] A. Brace, D. Gątarek, M. Musiela, "The market model of interest rate dynamics," Mathematical Finance, Vol.7, No.2, pp.127-155, (1997).

[4] K. Daniilidis, A. Makadia and T. Bulow, "Image processing in catadioptric planes: spatiotemporal derivatives and optical flow computation," *OMNIVIS02*, pp.3-10, 2002.

[5] N. I. Fisher, T. Lewis and B. J. J. Embleton, "Statistical anaylsis of spherical data," Cambridge Univ. Press, 1987.

[6] J. Fujiki, S. Akaho and N. Murata, "Nonlinear PCA/ICA for the structure from motion problem," In Proc. of ICA04(LNCS 3195), pp. 750-757, Granada, Sep. 2004.

[7] J. Fujiki and S. Akaho, "Small ciecle fitting to the sequence of spherical points -Towards smoothing of camera motions-," Technical Report of IEICE, PRMU2004-149, pp.91-96, 2004 , (In Japanese).

[8] J. Fujiki and S. Akaho, "Epipolar geometry for spherical camera and its calculations," Technical Report of IEICE, PRMU2005-41, pp.41-46, 2005 , (In Japanese).

[9] J. Fujiki and S. Akaho, "Small hypersphere fitting by Spherical Least Square," In Proc. of ICONIP05, pp.439-444, 2005.

[10] C. Geyer and K. Daniilidis, "Catadioptric Projective Geometry," IJCV, vol.45, no.3, pp.223-243, Dec. 2001.

[11] N. H. Gray, P. A. Geiser and J. R. Geiser, "On the least-squares fit of small and great circles to spherically projected orientation data," Mathematical Geology, vol.12, no.3, pp.173–184, 1980.

[12] I. Grubišić and R. Pietersz, "Efficient rank reduction of correlation matrices," Utrecht Univ., preprint, 2005.

[13] A. Makadia and K. Daniilidis, "Direct 3D-rotation estimation from spherical images via a generalized shift theorem," In proc. of CVPR03, pp.II: 217-224, 2003.

[14] J. Mimura, N. Murata and J. Fujiki, "Smoothing of camera motions via small circle fitting for the sequence of spherical points," Technical Report of IEICE, PRMU, 2005, (In Japanese).

[15] K. Miyamoto, "Fish eye lens," Journal of Optical Society of America, vol.54, no.8, pp.1060-1061, Aug. 1964.

[16] S. Oba and S. Ishii, "Kernel density estimation on hypersphere," 2004 Workshop on Information-Based Induction Sciences(IBIS2004), pp.197-202, (2004), (In Japanese).

[17] T. Svoboda and T. Pajdla, "Epipolar Geometry for Central Catadioptric Cameras," IJCV, vol.49, no.1, pp.23-37, 2002.

[18] A. Torii and A. Imiya, "Analysis of Central Camera Systems for Computer Vision," CVIM 154-30, 2006.

Kernel and Learning Local Manifold Matching for Image Classification

Seiji Hotta
Tokyo University of Agriculture and Technology
2–24–16 Naka-cho, Koganei, Tokyo, 184–8588 Japan
s-hotta@cc.tuat.ac.jp

Abstract

In this paper, two extensions of the memory-based subspace classifier called local manifold matching (LMM) are proposed for image classification. One is a kernel LMM method for incorporating transform-invariance of images such as shifts into LMM via kernel mappings. Other is a learning LMM method for reducing memory and computational costs of the original LMM algorithm. It is verified with experiments on a handwritten digit dataset that my methods can outperform other classifiers such as a support vector machine.

1. Introduction

For improving the performance of the nearest neighbor (NN) rule, Liu *et al.* proposed a classifier called *local manifold matching* (LMM) [1]. In LMM, a training sample is represented with a manifold (alias affine subspace) that is spanned by its k-closest training samples from the same class. In test phase, the projection distance between an input sample and each manifold is measured, and an input sample is classified into the class to which the nearest manifold belongs. By this representation, LMM can expand the representation capacity of available prototypes, thus LMM tends to outperform the NN rule.

The LMM classifier can be regarded as a combination method of subspace and nearest neighbor classifiers. In fact, the classification rule of LMM is same as that of *projection distance method* (PDM) [2] which is a kind of subspace methods [3, 4, 5]. The different point between LMM and PDM is the way to construct a manifold: In LMM, a manifold is formed per training sample. In contrast, a manifold is formed per class in PDM. On the other hand, when the Euclidean distance between an input sample and the original point of a manifold is used for classification instead of a projection distance, LMM will be equivalent to the NN rule.

As might be expected from the above relation, LMM would share the difficulties of both subspace and NN classifier. In fact, since the distance between a sample and a manifold is defined by a projection distance, it is hard to incorporate transform-invariance of images into LMM even if they are available. In contrast, such invariance can be incorporated easily into the NN rule by using some adequate distance measures such as a tangent distance [6, 7]. Moreover, memory a nd computational costs of LMM would be large because LMM is a kind of memory-based classifiers.

To overcome these difficulties, a kernel LMM (KLMM) classifier and a learning LMM (LLMM) classifier are proposed in this paper. First, KLMM is formulated with the same manner of kernel nonlinear subspace methods [8, 9]. In KLMM, first each training sample is represented as a manifold with its k-nearest neighbors after nonlinear kernel mapping. Next, the projection distance between an input sample and each manifold is computed by using a kernel trick so that a mapping function is never computed explicitly. In a test phase, an input sample is classified into the class to which the nearest manifold belongs. Furthermore, a combination of LMM and *learning subspace method* (LSM) is proposed for reducing the number of manifolds. In practice, LSM based on generalized learning vector quantization (henceforth denoted as GLSM) [10] is applied to LMM. GLSM is designed on the basis of generalized learning vector quantization (GLVQ) [11], so stable convergence of algorithms and improvement of the accuracy would be achieved by this method. In the proposed method, a learning rule is the same as that of the original GLSM algorithm, but learning is run not on individual classes but on each training manifold. The performance of my methods are verified with the experiments on handwritten digit image dataset USPS [12].

2. Local Manifold Matching

Let us begin with a brief review of *local manifold matching* (LMM) [1] before presenting the proposed methods.

2.1 Local Manifold Matching (LMM)

In [1], the LMM technique was presented for improving the performance of the NN classifier by expanding the

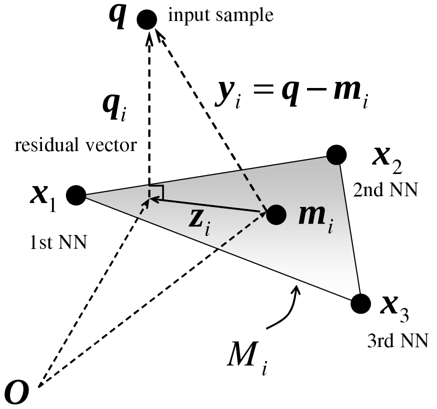


Figure 1. The concept of LMM in the ith datum with $k = 3$. The input sample q is classified according to the residual length from the input sample q to the linear manifold $\mathcal{M}_i$.

representation capacity of available prototypes. In LMM, a training sample is selected from dataset and its k-closest training samples are searched from the same class of it. Next, the selected training sample is replaced with a linear manifold spanned by its k-nearest training samples. This process is computed on each training sample. The classification is then based on the shortest distance from an input sample to each manifold. The simple concept of LMM in the ith datum with $k = 3$ is shown in Fig. 1.

2.2 Algorithm of LMM

Now a procedural formulation of the LMM algorithm is shown. Let $\boldsymbol{x}_i = (x_{i1} \cdots x_{id})^\top \in \mathbb{R}^d$ $(i = 1, ..., N)$ be the d-dimensional ith training sample, where N is the number of training samples. In addition, let y_i and C be the class to which $\boldsymbol{x}_i$ belongs and the number of classes, respectively.

Step1 Select a training sample $\boldsymbol{x}_i$, and find its k-closest training samples from the same class of $\boldsymbol{x}_i$ with some metrics $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (e.g., the Euclidean distance $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$). Here, the selected k-nearest neighbors are denoted as $\boldsymbol{x}_1$ $(= \boldsymbol{x}_i)$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_k$. Note again that these k-nearest neighbors should be selected from the class of $\boldsymbol{x}_i$.

Step2 Compute the mean of the selected neighbors by

$$\boldsymbol{m}_i = \frac{1}{k}\sum_{l=1}^{k} \boldsymbol{x}_l, \qquad (1)$$

where $\boldsymbol{x}_l$ is the lth nearest neighbor of $\boldsymbol{x}_i$. Next, form a $d \times k$ matrix $((\boldsymbol{x}_1 - \boldsymbol{m}_i) \cdots (\boldsymbol{x}_k - \boldsymbol{m}_i))$ and orthonormalize it by using the Gram-Schmidt process or eigenvalue decomposition. Resulting orthonormal basis $\mathbf{U}_i$ and the mean vector $\boldsymbol{m}_i$ span the local linear manifold of the training sample $\boldsymbol{x}_i$. Henceforth, we denote the local manifold constructed with $\mathbf{U}_i$ and $\boldsymbol{m}_i$ as $\mathcal{M}_i$. By this means, local manifolds for each training sample are constructed before a test phase.

Step3 When an input sample $\boldsymbol{q} = (q_1 \cdots q_d)^\top$ is given, it is classified into the class to which the nearest manifold belongs. Hence, we have to measure the distance between $\boldsymbol{q}$ and each manifold $\mathcal{M}_i$. In LMM, the distance between $\boldsymbol{q}$ and $\mathcal{M}_i$ is measured by projecting the difference vector $\boldsymbol{q} - \boldsymbol{m}_i$ into $\mathcal{M}_i$. For that purpose, calculate the residual of $\boldsymbol{q}$ relative to $\mathcal{M}_i$ (cf. Fig. 1) by the following equation:

$$\boldsymbol{q}_i = \boldsymbol{y}_i - \boldsymbol{z}_i = (\mathbf{I} - \mathbf{U}_i\mathbf{U}_i^\top)(\boldsymbol{q} - \boldsymbol{m}_i), \qquad (2)$$

where the projection of $\boldsymbol{q} - \boldsymbol{m}_i$ on the manifold is given as

$$\boldsymbol{z}_i = \mathbf{U}_i\mathbf{U}_i^\top(\boldsymbol{q} - \boldsymbol{m}_i). \qquad (3)$$

According to the Pythagorean theorem, the norm of $\boldsymbol{q}_i$ (i.e., projection distance) can be computed as follows:

$$d_i = \|\boldsymbol{q}_i\|^2 = \|\boldsymbol{q} - \boldsymbol{m}_i\|^2 - \|\mathbf{U}_i^\top(\boldsymbol{q} - \boldsymbol{m}_i)\|^2. \qquad (4)$$

By this distance, the class of $\boldsymbol{q}$ (denoted as ω) is determined as that of the nearest manifold: $\omega = y_{i^*}$, where i^* is

$$i^* = \arg\min_{i=1,...,N} d_i. \qquad (5)$$

The LMM classifier is a very general one because it includes various classifiers. In fact, when $k = 1$, LMM is equivalent to the NN rule. On the other hand, if each manifold is constructed with not neighbor samples but all samples in individual classes, LMM is equivalent to PDM [2]. In addition, when the distance between $\boldsymbol{q}$ and $\boldsymbol{m}_i$ is used instead of Eq. (4) for classification, LMM is equivalent to the bootstrap sample method [13]. Furthermore, when the k-closed training samples are selected for an input sample, the LMM algorithm is equivalent to *local subspace classifier* (LSC) [14, 15]. Nobody has pointed out the relation of these algorithms in the past.

2.3 Kernel LMM (proposal)

In LMM, the distance between an input sample and a manifold is defined by a projection distance, so it is hard to incorporate transform-invariance into LMM even if it is available. Hence, we derive a kernel LMM (KLMM) classifier for incorporating such invariance into LMM via kernel mappings [16]. Extension from LMM to KLMM can be achieved by a kernel trick $\Phi(\boldsymbol{x})^\top\Phi(\boldsymbol{y}) = K(\boldsymbol{x}, \boldsymbol{y})$ for mapping samples from an input space to a feature space $\mathbb{R}^d \mapsto \mathcal{F}$. In addition, we make use of the results of kernel PCA [17] for computing the residual length from an input sample to linear manifolds in $\mathcal{F}$. In brief, the algorithm of KLMM is given as follows:

Step1 Find the k-closest training samples to a selected training sample $\Phi(\boldsymbol{x}_i)$ in $\mathcal{F}$ and denote them as $\Phi(\boldsymbol{x}_1)$

$(= \Phi(\boldsymbol{x}_i))$,...,$\Phi(\boldsymbol{x}_k)$. Note that they are selected from the same class of $\Phi(\boldsymbol{x}_i)$, and the Euclidean distance between $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x}_j)$ in $\mathcal{F}$ is given by the following equation:

$$\begin{aligned} &D^2(\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j)) = \|\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)\|^2 \\ &= K(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2K(\boldsymbol{x}_i, \boldsymbol{x}_j) + K(\boldsymbol{x}_j, \boldsymbol{x}_j). \end{aligned} \tag{6}$$

Step 2 Let $\mathbf{X}_i$ be the basis of vectors in $\mathcal{F}$, i.e., $\mathbf{X}_i = [(\Phi(\boldsymbol{x}_1) - \Phi(\boldsymbol{m}_i)) \cdots (\Phi(\boldsymbol{x}_k) - \Phi(\boldsymbol{m}_i))]$, where

$$\Phi(\boldsymbol{m}_i) = \frac{1}{k} \sum_{l=1}^{k} \Phi(\boldsymbol{x}_l). \tag{7}$$

Next, form a $k \times k$ matrix $\mathbf{N}_i$ by

$$\mathbf{N}_i = \mathbf{X}_i^\top \mathbf{X}_i = \mathbf{K}_i - \frac{1}{k}(\mathbf{K}_i^{(m)} + \mathbf{K}_i^{(n)}) + \frac{1}{k^2}\mathbf{K}_i^{(1)}, \tag{8}$$

where $\mathbf{K}_i$ is a Gram (kernel) matrix of which the (m, n) element is defined as $\mathbf{K}_i(m, n) = K(\boldsymbol{x}_m, \boldsymbol{x}_n)$ $(m, n = 1, ..., k)$. In addition, the matrices $\mathbf{K}_i^{(m)}$, $\mathbf{K}_i^{(n)}$ and $\mathbf{K}_i^{(1)}$ are $k \times k$ matrices, where the mth row of $\mathbf{K}_i^{(m)}$ is defined as $\sum_{l=1}^{k} K(\boldsymbol{x}_m, \boldsymbol{x}_l)$, the nth column of $\mathbf{K}_i^{(n)}$ is defined as $\sum_{l=1}^{k} K(\boldsymbol{x}_n, \boldsymbol{x}_l)$ and all elements of $\mathbf{K}_i^{(1)}$ are defined as $\sum_{l=1}^{k} \sum_{m=1}^{k} K(\boldsymbol{x}_l, \boldsymbol{x}_m)$, respectively. Finally, decompose $\mathbf{N}_i$ by using eigenvalue decomposition and form a $k \times (k-1)$ orthonormal matrix using the eigenvectors $\boldsymbol{u}_1, ..., \boldsymbol{u}_{k-1}$ normalized with the corresponding eigenvalues $(\lambda_1, ..., \lambda_{k-1})$ such as

$$\mathbf{U}_i = \left[(\boldsymbol{u}_1/\sqrt{\lambda_1}) \cdots (\boldsymbol{u}_{k-1}/\sqrt{\lambda_{k-1}})\right]. \tag{9}$$

By this orthonormal matrix, we can represent the orthonormal basis in feature space $\mathcal{F}$ as $\mathbf{V}_i = \mathbf{X}_i\mathbf{U}_i$. However, we cannot represent $\mathbf{V}_i$ as a matrix explicitly due to its high-dimensionality.

Step 3 The input sample $\Phi(\boldsymbol{q})$ is classified according to the minimal residual length to the linear manifolds in $\mathcal{F}$, i.e., the class of the input sample ω is determined as $\omega = y_{i^*}$, where i^* is given by

$$\begin{aligned} &i^* = \\ &\arg\min_{i=1,\ldots,N}\{\|\Phi(\boldsymbol{q}) - \Phi(\boldsymbol{m}_i)\|^2 - \|\mathbf{V}_i^\top(\Phi(\boldsymbol{q}) - \Phi(\boldsymbol{m}_i))\|^2\} \\ &= \arg\min_{i=1,\ldots,N}\{D^2(\Phi(\boldsymbol{q}), \Phi(\boldsymbol{m}_i)) - \|\mathbf{U}_i^\top \boldsymbol{Q}_i\|^2\}, \end{aligned} \tag{10}$$

where $\boldsymbol{Q}_i$ is a $k \times 1$ vector of which the jth element is given by the following equation:

$$\begin{aligned} Q_{ij} &= (\Phi(\boldsymbol{x}_j) - \Phi(\boldsymbol{m}_i))^\top(\Phi(\boldsymbol{q}) - \Phi(\boldsymbol{m}_i)) \\ &= K(\boldsymbol{x}_j, \boldsymbol{q}) - \frac{1}{k}\sum_{l=1}^{k}(K(\boldsymbol{x}_j, \boldsymbol{x}_l) + K(\boldsymbol{x}_l, \boldsymbol{q})) \qquad (11) \\ &+ \frac{1}{k^2}\sum_{l=1}^{k}\sum_{m=1}^{k} K(\boldsymbol{x}_l, \boldsymbol{x}_m). \end{aligned}$$

If necessary, it is possible to reduce the dimensionality of a linear manifold on the basis of a criterion such as a cumulative proportion, however, the accuracy of KLMM is sensitive against not the dimensionality of a linear manifold but the number of k-nearest neighbors.

2.4 Learning LMM (proposal)

Memory and computational costs of memory-based classifiers such as the NN rule tend to be large. For reducing them without deterioration of accuracy, several methods such as learning vector quantization [18, 11] were proposed in the past. However, they were designed for vector samples, so it is hard to apply them to LMM directly. In this paper, we apply *learning subspace method* (LSM) [5, 10] to LMM for reducing the number of manifolds without deterioration of accuracy. In practice, we adopt GLSM proposed by Sato and Yamada [10] to LMM. The experimental results reported in [10] show that recognition performance of PDM was improved by this learning method. The learning rule shown in this paper is the same as that of the original algorithm proposed in [10], but learning is run not on individual classes but on each manifold constructing with codebook vectors. For convenience, we call the proposed method *learning local manifold matching* (LLMM). Note that we can not apply the same learning rule to LSC [14, 15] directly because local manifolds in LSC are computed with neighbor training samples of an input sample.

Now the algorithm of LLMM is summarized as follows: Let $\boldsymbol{m}_1$ and $\mathbf{U}_1$ be the mean vector and the orthogonal basis matrix of the nearest manifold $\mathcal{M}_1$ that belongs to the same class of an input sample $\boldsymbol{x}$. In contrast, let $\boldsymbol{m}_2$ and $\mathbf{U}_2$ be the mean vector and the orthogonal basis matrix of the nearest manifold $\mathcal{M}_2$ that belongs to a different class from $\boldsymbol{x}$. Let us consider the relative distance difference $\mu(\boldsymbol{x})$ defined as follows:

$$\mu(\boldsymbol{x}) = \frac{d_1 - d_2}{d_1 + d_2}, \tag{12}$$

where d_1 and d_2 are the distance values of $\boldsymbol{x}$ from $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively (cf. Eq (4)). The above $\mu(\boldsymbol{x})$ satisfies $-1 < \mu(\boldsymbol{x}) < 1$. If $\mu(\boldsymbol{x})$ is negative, $\boldsymbol{x}$ is classified correctly; otherwise, $\boldsymbol{x}$ is misclassified. For improving accuracy, we should minimize the following cost function:

$$S = \sum_{i=1}^{N} f(\mu(\boldsymbol{x}_i)), \tag{13}$$

where N is the number of samples for training, and $f(\mu)$ is a monotonically increasing function. To minimize S, the steepest descent method with a small positive constant α

$(0 < \alpha < 1)$ is adopted to Eq. (13) [10, 11]:

$$\boldsymbol{m}_i \leftarrow \boldsymbol{m}_i - \alpha \frac{\partial S}{\partial \boldsymbol{m}_i}, \mathbf{U}_i \leftarrow \mathbf{U}_i - \alpha \frac{\partial S}{\partial \mathbf{U}_i}, (i = 1, 2). \tag{14}$$

Now $\partial S / \partial \boldsymbol{m}_i$ and $\partial S / \partial \mathbf{U}_i$ can be derived as

$$\frac{\partial S}{\partial \boldsymbol{m}_i} = \frac{\partial S}{\partial \mu} \frac{\partial \mu}{\partial d_i} \frac{\partial d_i}{\partial \boldsymbol{m}_i} = \mp \frac{\partial f}{\partial \mu} \frac{4 d_{3-i}}{(d_1 + d_2)^2} (\boldsymbol{y}_i - \mathbf{U}_i \mathbf{U}_i^\top \boldsymbol{y}_i), \tag{15}$$

$$\frac{\partial S}{\partial \mathbf{U}_i} = \frac{\partial S}{\partial \mu} \frac{\partial \mu}{\partial d_i} \frac{\partial d_i}{\partial \mathbf{U}_i} = \mp \frac{\partial f}{\partial \mu} \frac{4 d_{3-i}}{(d_1 + d_2)^2} (\boldsymbol{y}_i \boldsymbol{y}_i^\top \mathbf{U}_i), \tag{16}$$

where $\boldsymbol{y}_i = \boldsymbol{x} - \boldsymbol{m}_i$. Consequently, the update rule for LLMM can be written as follows:

$$\boldsymbol{m}_i \leftarrow \boldsymbol{m}_i + \alpha_i \frac{\partial f}{\partial \mu} \frac{d_{3-i}}{(d_1 + d_2)} (\boldsymbol{y}_i - \mathbf{U}_i \mathbf{U}_i^\top \boldsymbol{y}_i), \tag{17}$$

$$\mathbf{U}_i \leftarrow \mathbf{U}_i + \alpha_i \frac{\partial f}{\partial \mu} \frac{d_{3-i}}{(d_1 + d_2)} (\boldsymbol{y}_i \boldsymbol{y}_i^\top \mathbf{U}_i), \tag{18}$$

where $\alpha_2 = -\alpha_1$. In addition, correction $d_i/(d_1 + d_2)$ dose not affect the convergence condition [11]. In practice, the updated $\mathbf{U}_i$ is not an orthogonal matrix, so the Gram-Schmidt process is applied to $\mathbf{U}_i$ for orthogonalization in each step. In [10] and [11], $f(\mu, t)\{1 - f(\mu, t)\}$ is used for $\partial f / \partial \mu$, where t is learning time and $f(\mu, t)$ is a sigmoid function $1/(1 - e^{-\mu t})$. In this case, $\partial f / \partial \mu$ has a single peak at $\mu = 0$, and the peak width becomes narrower as t increases. After the above training, an input sample is classified by the same rule of LMM with each trained manifold.

3. Experiments

The experimental results on the handwritten digit image dataset USPS [12] are shown. The USPS dataset consists of 7,291 training and 2,007 test images. The size of images is 16 × 16 pixels. In experiments, the intensities of images were directly used as feature vectors. First, the effectiveness of KLMM was verified with comparing to *other classifiers*: *k-nearest neighbor rule* (kNN), PDM, kernel PDM (KPDM) [9] and *support vector machine* (SVM). Next, the effectiveness of LLMM was verified with comparing to *other classifiers*: kNN, LMM, GLVQ and *averaged learning subspace method* (ALSM) [5]. Note that all the parameters of these classifiers were determined at the minimum validation errors. However, the dimensionalities of linear manifolds in my method and LMM were always set as a full rank (i.e., less than or equal to $k - 1$) in every k for convenience. All methods were implemented with MATLAB on a standard PC that has Pentium 1.86GHz CPU and 2GB RAM.

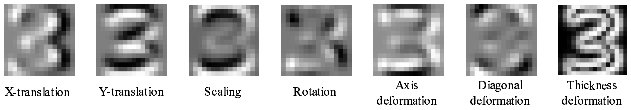


Figure 2. Seven tangent vectors: x and y translation, scaling, rotation, axis, diagonal and thickness deformation, respectively.

Table 1. Test error rates on USPS.

method	error rate [%]
1-NN (Euclidean Distance)	5.5
1-NN (one sided TD)	4.1
PDM	5.2
KPDM (one sided TDK)	3.9
SVM (RBF kernel)	4.6 [19]
SVM (one sided TDK)	4.1 [19]
LMM ($k = 16$)	4.1
KLMM (RBF, $k = 10$)	4.0
KLMM (TDK, $k = 11$) **this work**	**3.1**

3.1 Experimental results of KLMM

First, we investigated the error rates of LMM and KLMM with respect to the number of k-nearest neighbors. In kernel machines, we used a tangent distance kernel (TDK) proposed by Haasdonk *et al.* [19] because of the reasonable performance that can be obtained in spite of their extreme simplicity. TDK is defined by replacing the Euclidean distance with a *tangent distance* (TD) [6] in arbitrary distance-based kernels. For example, if we modify the following radial basis function (RBF) kernel

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\beta \|\boldsymbol{x} - \boldsymbol{y}\|^2) \tag{19}$$

by replacing the Euclidean distance with a one sided TD, we then obtain the kernel called *one sided TD kernel*:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\beta \times d_{1S}^2(\boldsymbol{x}, \boldsymbol{y})), \tag{20}$$

where $d_{1S}^2(\boldsymbol{x}, \boldsymbol{y})$ is

$$d_{1S}^2(\boldsymbol{x}, \boldsymbol{y}) = \min_{\boldsymbol{\alpha}} \|\boldsymbol{x} - (\boldsymbol{y} + \boldsymbol{\alpha}\mathbf{T})\|^2. \tag{21}$$

In the above equation, $\mathbf{T}$ is a $d \times r$ matrix of which columns are tangent vectors corresponding to r image transformation (cf. [7] for details), and $\boldsymbol{\alpha}$ is a $r \times 1$ parameter vector for each tangent vector. Minimization of Eq. (21) is simple since the squared distance is a quadratic function of $\boldsymbol{\alpha}$. Thus, the solution of $\boldsymbol{\alpha}$ can be derived as follows:

$$\boldsymbol{\alpha} = (\mathbf{T}^\top \mathbf{T})^{-1} \mathbf{T}^\top (\boldsymbol{x} - \boldsymbol{y}). \tag{22}$$

In this paper, seven tangent vectors as shown in Fig. 2 were used for experiments. We can achieve higher accuracy by

Table 2. Error rates [%] and standard deviations on USPS.

n_j	1-NN	GLVQ	LMM	LLMM (**this work**)	ALSM	SVM
10	22.7 ± 1.5	7.1 ± 0.4	17.0 ± 0.9	$\mathbf{5.8 \pm 0.4}$	7.7 ± 0.5	22.1 ± 1.3
50	12 ± 0.8	7.4 ± 0.3	8.2 ± 0.5	$\mathbf{5.4 \pm 0.2}$	6.3 ± 0.4	11.6 ± 0.9
100	9.4 ± 0.6	6.8 ± 0.3	6.5 ± 0.4	$\mathbf{5.2 \pm 0.2}$	6.0 ± 0.3	8.6 ± 0.5
150	8.1 ± 0.5	6.5 ± 0.4	5.7 ± 0.2	$\mathbf{4.8 \pm 0.3}$	5.8 ± 0.3	8.1 ± 0.4
200	7.5 ± 0.4	6.1 ± 0.3	5.2 ± 0.2	$\mathbf{4.6 \pm 0.3}$	5.7 ± 0.2	7 ± 0.4
250	6.9 ± 0.3	5.9 ± 0.2	4.9 ± 0.2	$\mathbf{4.5 \pm 0.2}$	5.6 ± 0.2	6.6 ± 0.3
complete set	5.5		4.1	Learning PDM [10] 5.0		4.6 [19]

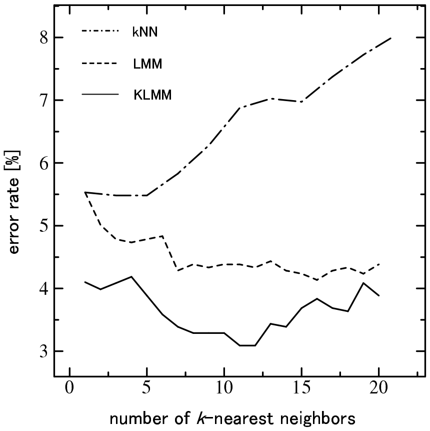


Figure 3. Error rates WRT k on USPS.

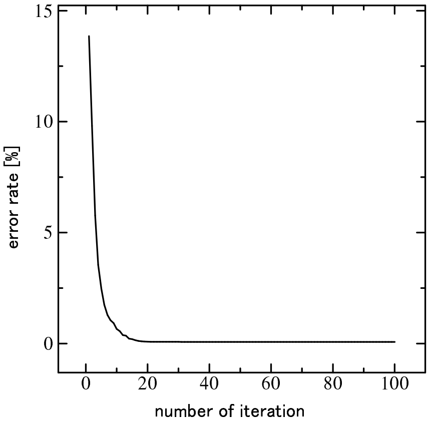


Figure 4. Training error WRT the number of iteration.

this simple modification than the use of the original RBF kernel [19]. In addition, the above modification is adequate for kernel setting because of its natural definition. Hence, we used the one sided TDK on our experiments.

Figure 3 shows the relationship between the number of neighbors k and the error rates of *each classifier*: kNN, LMM, and KLMM. As shown in this figure, the error rates of LMM and KLMM against test samples decreased as the k increased. Note that the error rate on KLMM is lower than those on kNN and LMM in every k.

Table 1 lists the lowest error rates with the parameter values of each classifier. The number of neighbors k in LMM and KLMM were tuned on a separate validation set (6291 training samples, 1000 validation samples). The results on SVM is referred to [19]. Table 1 shows that KLMM with TDK outperformed the other classifiers. However, the error rate of KLMM with RBF was the almost same as that of LMM because LMM is already effective for nonlinearity of data. Thus, nonlinear mappings via RBF kernel were not effective for improving accuracy of LMM.

3.2 Experimental results of LLMM

Next, we evaluated LLMM on the USPS digit dataset. For experiments, the set of training images was randomly split into two sets of equal size (i.e., about 3600 images in each set). One of them was used in initializing manifolds and the other was used as training samples. This random splitting was performed independently for ten repetitations of the training and testing. In addition, the number of the codebook vectors (manifolds in LMM and KLMM) of the class j (denoted as n_j) was varied from 10 to 250 (the total number of the images of '8' is 542, so the maximum size of initial manifolds per class is almost 270). For GLVQ, the fixed learning rate $\alpha = 0.03$ and 100 learning time were introduced. On the other hand, the fixed learning rate $\alpha = 10^{-6}$ and 100 learning time were introduced for LLMM. For SVM, 1-NN and PDM, the initial training vectors were directly used for a test phase.

Figure 4 shows the relationship between the number of iteration and mean training error rate with $n_j = 10$ and $k = 4$. As shown in this figure, the error rate was almost equal to 0 at around 20 iterations. The error rates with different manifold sizes are shown in Table 2. The mean values and standard deviations were estimated from ten independent outcomes. For SVM, we used the RBF kernel for a nonlinear mapping. As shown in Table 2, the proposed learning rule LLMM outperformed the other methods such as ALSM in all cases. Furthermore, the error rate of LLMM with $n_j = 250$ was lower than those of the original learning PDM (5.0%) and SVM with a complete set (4.6%).

4. Conclusions

In this paper, two extensions of the memory-based subspace classifier called local manifold matching (LMM) were proposed. One is a kernel LMM method (KLMM) for incorporating transform-invariance into LMM via kernel mappings. Other is a learning LMM method (LLMM) for reducing memory and computational costs of the original LMM classifier. It was verified with experiments on handwritten digit image dataset USPS that my methods achieved lower error rates than other classifiers such as SVM with low memory costs.

In this paper, two extensions of LMM called KLMM and LLMM were discussed separately. However, it can be expected that accuracy would be improved by combining them. In fact, kernel GLVQ [20] was proposed, and the experimental results reported in [20] showed that kernelized GLVQ can improve accuracy of the original GLVQ. Hence, future work will be dedicated to combine KLMM and LLMM for improving accuracy with low memory requirements.

Acknowledgements

This research was supported by The Ministry of Education, Culture, Sports, Science and Technology in Japan under a Grant-in-Aid for Division of Young Researchers.

References

[1] W. Liu, W. Fan, Y. Wang, and T. Tan, "Local manifold matching for face recognition," Proc. of ICIP, vol. 2, pp. 926–929, 2005.

[2] K. Ikeda, H. Tanaka, and T. Motooka, "Projection distance method for recognition of hand-written characters," J. IPS. Japan, vol. 24, no. 1, pp. 106–112, 1983.

[3] S. Watanabe and N. Pakvasa, "Subspace method in pattern recognition," Proc. of 1st Int. J. Conf on Patt. Recog., WashingtonDC, 1973.

[4] T. Iijima, H. Genchi, and K. Mori, "A theory of character recognition by pattern matching method," Proc. of 1st Int'l J. Conf. on Patt. Recog., pp. 50–56, 1973.

[5] E. Oja, "Subspace methods of pattern recognition," Research Studies Press, 1983.

[6] P.Y. Simard, Y. LeCun, and J.S. Denker, "Efficient pattern recognition using a new transformation distance," Proc. of NIPS, vol. 5, pp. 50–58, 1993.

[7] P.Y. Simard, Y. LeCun, J.S. Denker, and B. Victorri, "Transformation invariance in pattern recognition – Tangent distance and tangent propagation," Int'l J. of Imag. Sys. and Tech., vol. 11, no. 3, 2001.

[8] K. Tsuda, "Subspace classifier in the Hilbert space," Patt. Recog. Lett., vol. 20, no. 5, pp. 513–519, 1999.

[9] E. Maeda and H. Murase, "Multi-category classification by kernel based nonlinear subspace method," Proc. of ICASSP, vol. 2, pp. 1025–1028, 1999.

[10] A. Sato and K. Yamada, "A learning subspace method based on generalized learning vector quantization," Proc. of the Society Conference of IEICE (Japanese Edition), vol. 1997, p. 236, 1997.

[11] A. Sato and K. Yamada, "Generalized learning vector quantization," Prop. of NIPS, vol. 7, pp. 423–429, 1995.

[12] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neur. Comp., vol. 1, no. 4, pp. 541–551, 1989.

[13] Y. Hamamoto, S. Uchimura and S. Tomita: "A bootstrap technique for nearest neighbor classifier design", PAMI, vol. 19, no. 1, pp. 73–79, 1997.

[14] J. Laaksonen, "Local subspace classifier," Proc. of ICANN'97, pp. 637–642, 1997.

[15] J. Laaksonen, "Subspace classifiers in recognition of handwritten digits," PhD thesis, Helsinki University of Technology, 1997.

[16] B. Schölkopf and A.J. Smola, "Learning with kernels," MIT press, 2002.

[17] B. Schölkopf, A.J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neur. Comp., vol. 10, pp. 1299–1319, 1998.

[18] T. Kohonen, "Self-organizing maps," 2nd Ed, Springer-Verlag, Heidelberg, 1995.

[19] B. Haasdonk and D. Keysers, "Tangent distance kernels for support vector machines," Proc. of ICPR, vol. 2, pp. 864–868, 2002.

[20] A.K. Qin and P.N. Suganthan, "A novel kernel prototype-based learning algorithm," Proc. of ICPR, vol. 4, pp. 621–624, 2004.

A Study on PCA-based Fourier Descriptor in Complete and Incomplete Contour Representations

Li Tian, Sei-ichiro Kamata
Waseda University
Graduate School of Info., Pro.&Sys.
2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Japan
tianli@ruri.waseda.jp, kam@waseda.jp

Abstract

Shape descriptors are very important in many image classification and retrieval applications. Among various shape descriptors, Fourier Descriptor (FD) has been proved to be the most efficient and stable one with low computational cost and few parameter tuning. In this study, a new Principal Component Analysis (PCA)-based FD for shape representation and retrieval is presented. Instead of using a few of Fourier coefficients directly in the standard FD, we describe a shape by applying PCA to FD with a large number of Fourier coefficients. A comparative study on its performance in complete and incomplete contour representations using large shape database is also given. The experimental results show that PCA-FD is better in complete contour representation, however, is worse in incomplete contour representation than the classical standard FD.

1 Introduction

Efficient and effective methods for retrieving images from large image databases is an interesting research topic in recent years. Shape is considered to be a much more effective feature than other image features such as color and texture [21, 5, 3] for characterizing the content of an image. How to represent the shape by using shape descriptors so as to do efficient and effective retrieval is an essential problem in this field. A number of approaches and solutions have been developed to characterize the shapes.

We usually divide shape descriptors into two categories: region-based and contour-based techniques [32]. The former group usually uses moment descriptors [13, 25, 20] to describe shapes while the latter group uses only shape contour information. In this study, we concentrate on the contour-based ones. Conventional contour-based shape descriptors include Curvature Scale Space Descriptor (CSSD) [22] which has been selected for MPEG-7 standardization, Wavelet Descriptor (WD) [26], Fourier Descriptor (FD) [9] and its variations [30, 3, 17], and some other descriptors [8, 15, 7, 18, 24, 4, 12]. Three basic requirements [3]:

- *compactness* and *simplicity* for saving storage and computational cost,
- *robustness to noise* and *invariance to transformations* for effective retrieving,
- *indexability* for searching in a large image database

are necessary for effective and efficient retrieval of images based on their shapes. FD and CSSD are thought to be the available candidates for obtaining good effectiveness along with efficiency retrieval for large image databases according to the above requirements. Moreover, as reported in [28, 31, 17], FD descriptors outperformed CSSD in terms of retrieval accuracy and efficiency, it is considered as the best choice for shape representation and retrieval.

Comparative studies on FD has been given in [31, 33] and the authors indicated that the retrieval performance does not improve significantly nor degrade with the number changes of Fourier coefficients used in FD in a small range from 10 to 90. We find this phenomenon is true in small ranges, however, is not tenable when the number increases to thousands. We will discuss this issue in more detail later. FD with large number of Fourier coefficients can improve the retrieval performance much, but it will be computational expensive for retrieving in a large database. Accordingly, a natural idea is for solving this problem is to reduce the dimension of FD with information lost as few as possible.

Principal Component Analysis (PCA) [14] is such a method enabling us to linearly-project high-dimensional samples onto a low-dimensional feature space. It is not only important theory for solving many pattern recognition problems in computer vision, but also it has been widely used as a practical methodology for a wide variety of real applications such as feature selection [10], face recognition [27], object recognition [23] and image matching [16]. In this study, PCA is used to reduce the dimension of the standard FD.

We give a study on PCA-FD in shape retrieval in the experiment. Both complete and incomplete contour representations [11] are examined in the experiment using large shape data set. The experiments demonstrate that the proposed PCA-FD is a more compact and effective shape descriptor than the standard FD in complete contour representation but is not as good as FD in incomplete contour representations.

The rest of paper are organized as follows. In Section 2, we review the proposed PCA-FD in detail. Section 3 is about different experimental designs and achieved results. Finally, we conclude this study in the last section.

2 PCA-based FD descriptors

2.1 Fourier Descriptor

FD for shapes can be obtained by applying a Fourier transform on a shape signature such as complex coordinates, the curvature, the cumulative angle or the centroid distance. Since it has been shown that FD based on centroid distance is more effective than FDs based on other signatures [29], we will present our work on the centroid distance-based FD through this study.

If the contour coordinates of a shape in 2D space are $(x(t), y(t)), t = 0, 1, 2, ..., N-1$ where N is the number of points on the contour. First we calculate their centroid point (x_c, y_c) as

$$x_c = \frac{1}{N}\sum_{t=0}^{N-1} x(t), \quad y_c = \frac{1}{N}\sum_{t=0}^{N-1} y(t). \tag{1}$$

The centroid distance function is expressed by the distances of contour points to the centroid point defined as

$$r(t) = \sqrt{(x(t)-x_c)^2 + (y(t)-y_c)^2}. \tag{2}$$

It is easy to understand this shift makes the shape representation invariant to translation. The discrete Fourier transform of the centroid distance series is given by

$$F(n) = \frac{1}{N}\sum_{t=0}^{N-1} r(t)e^{-j2\pi nt/N}, \tag{3}$$

where $F(n)$ are the transform coefficients of $r(t)$. Furthermore, the coefficients have to be normalized to achieve invariance to rotation and scaling. The descriptors can be made invariant to rotation by using only the magnitudes of the transformation coefficients and invariant to scaling by dividing the magnitudes of the coefficients by the magnitude of the DC components (the first component). Because all distances are real valued for centroid distance signature, only half of the coefficient feature vector is needed to index the shape. A simple outline of calculating FD is presented in Fig. 1.

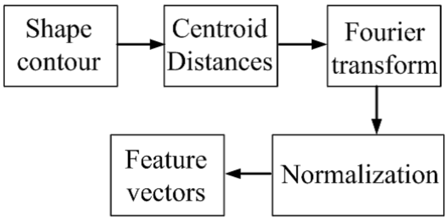


Figure 1. The outline of calculating FD descriptor.

As mentioned previously, although authors indicated that the retrieval performance does not improve significantly nor degrade with the number changes of FD features in a small range from 10 to 90 in [31, 33], we found this phenomenon is not tenable when the number increases to hundreds or thousands. Fig. 2 shows an illustration of the overall retrieval rate changes with the increasing dimension of FD using centroid distance as the shape signature on MPEG-7 data set [2]. We can observe that the overall retrieval rate (bulls-eye test) increases from 21.15% to 38.56% when the dimension of FD increases from 10 to 1200. Considering the number of shape signature is always much less than thousands, using high-dimensional FD can be viewed as a dimension-increase process in this study. The overall retrieval rate is comparatively low here, however, we only care about the improvement by increasing the dimension of FD in this case, not the overall retrieval rate itself.

2.2 PCA-FD

Using PCA to reduce the dimension of FD is not novel. The major contribution of this study is vividly demonstrating that the dimension increase-reduction process is useful and PCA is well-suited to representing FD. The retrieving performance of standard FD is significantly improved by this presentation.

Our PCA-FD for shape description has the same input as the standard FD: the coefficients obtained by Fourier transform on the signature of a shape. In this study, we compute 1200 normalized coefficients for each shape for training eigenspace. PCA-FD can be obtained in the following steps:

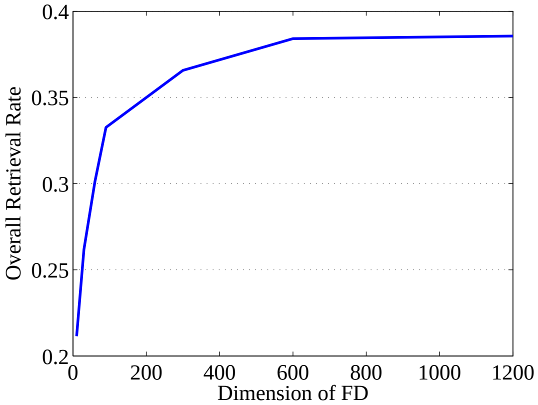


Figure 2. Overall retrieval precision changes with increasing dimension of FD on MPEG-7 data set.

1. pre-compute eigenspace to express the high-dimensional FD;
2. given a shape, compute its standard high-dimensional FD;
3. project the standard high-dimensional FD onto a compact feature vector using the pre-computed eigenspace.

As mentioned above, the input vector is created by computing a 1200–element FD for each shape and all elements have been normalized to obtain the invariance to scaling.

We collect a large number of different shapes and calculate their FDs to build our eigenspace. Each was processed as the steps described previously to create a 1200-element vectors. Pre-compute eigenspace process is described as follows. Suppose we have a large FD vector population $\mathbf{x}$ where

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)^T \tag{4}$$

from n training shapes. The mean of that population is denoted by

$$\mu_{\mathbf{x}} = E\{\mathbf{x}\} \tag{5}$$

and the covariance matrix of the same data set is

$$C_x = E\{(\mathbf{X} - \mu_{\mathbf{x}})(\mathbf{X} - \mu_{\mathbf{x}})^T\}. \tag{6}$$

First an eigen-decomposition of C_x is performed to produce a matrix

$$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n)^T \tag{7}$$

where $\mathbf{w}_i$ are the eigenvectors arranged by their eigenvalues in decreasing order. This space can be reduced to $\mathbf{W}_k$, where k indicates only k ($k << n$) eigenvectors with the largest eigenvalues are needed, and a linear combination of them

$$\tilde{\mathbf{x}}_i = \sum_{j=1}^{k} a_j(\mathbf{x})\mathbf{w}_j = \mathbf{W}_k\mathbf{W}_k^T\mathbf{x}_i \tag{8}$$

can represent $\mathbf{x}_i$ to a sufficient degree of accuracy where $\tilde{\mathbf{x}}_i$ is the approximation to $\mathbf{x}_i$ and a_j denotes the coefficients obtained by projecting original data onto the PCA eigenspace. The matrix of the coefficients is referred to as the feature matrix. The matrix consisting of the top n eigenvectors was stored on disk and used as the projection matrix for PCA-FD.

To construct the feature vector for a given shape in testing data, we first simply create its 1200-element FD descriptor and then simply project it into the PCA feature space using the stored eigenspace. The most top $n = 30$ values are selected empirically in this study which results in significant space benefits and time saving.

3 Experimental Results

We set up two retrieval experiments in this section: an experiment for complete contour representation and an other one for incomplete one compared with the standard FD.

3.1 Complete Contour Representation

This experiment is setup for comparing the retrieval performance of the proposed PCA-FD and standard FD in complete contour representation. The experimental setup are as follows:

Shape database The original shape database for testing is MPEG-7 data set B [2] including 1400 shapes of 70 classes. All these shapes are normalized to 128×128 pixels and the contours of them are extracted by using Canny Edge Detector [6]. Some contours of the testing shapes are shown in Fig. 3. For comparing the performances of different descriptors fairly, the shapes used for constructing eigenspace should not be used in the retrieving experiment. Thus, we divide the MPEG-7 data set B into two equivalent groups, each group contains 700 shapes from 10 shapes in each class. One group is used for training eigenspace and the other is for testing.

Evaluation criterion Recall-Precision is used for evaluating the retrieval results in this experiment:

$$recall = \frac{\sharp correct\ positives}{\sharp positives} \tag{9}$$

and

$$precision = \frac{\sharp correct\ positives}{\sharp matches(correct\ and\ false)}. \tag{10}$$

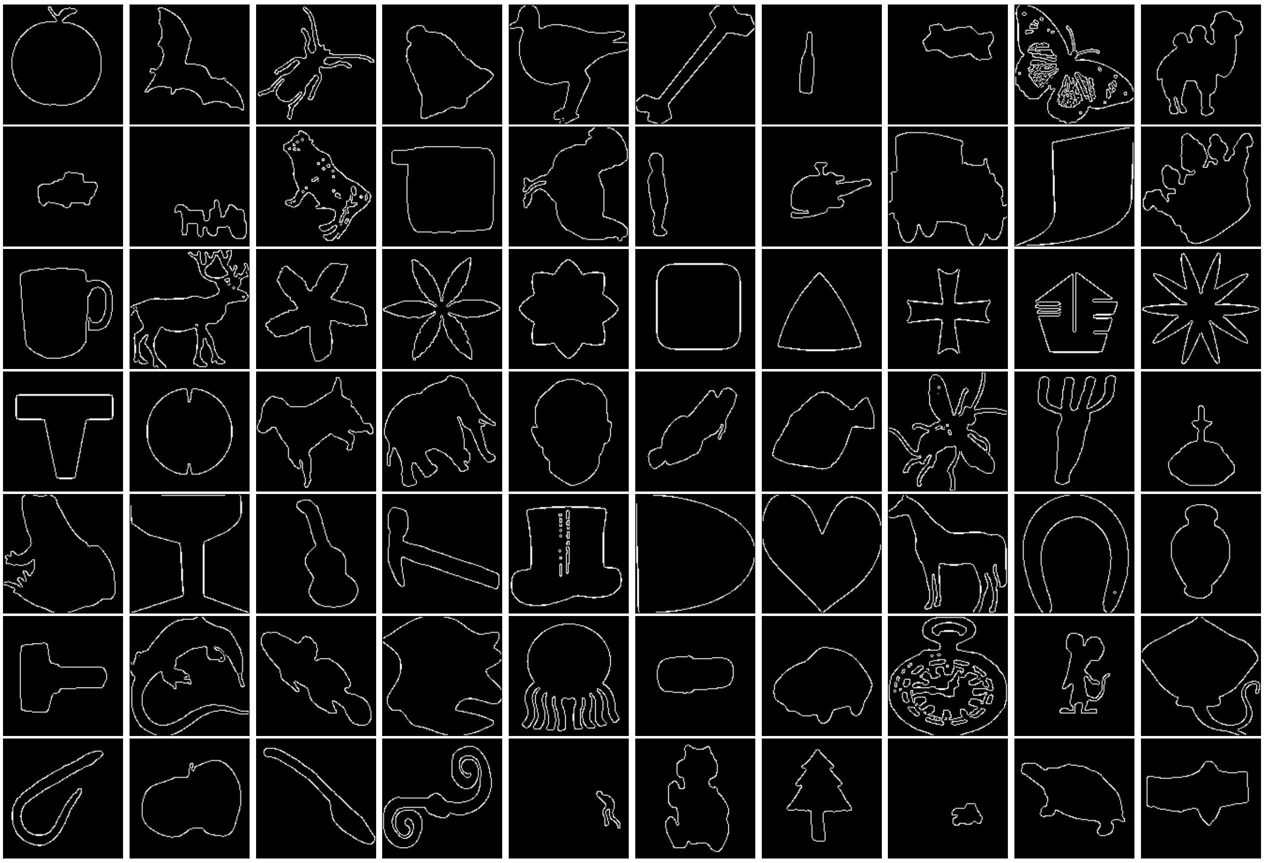


Figure 3. Contours of typical shapes in MPEG-7 data set B, one shape from each class.

A correct positive is a correct match between a query shape and one of its retrieved shape in the shape database.

We test all 700 shapes as query shapes and each has 10 correct matches (including itself). The Recall-Precision graph is computed on the above test by varying different threshold for retrieved shape number. The retrieved results of the standard FD with 30 Fourier coefficients and 30-element PCA-FD are shown in Fig. 4. From the figure, it is easy to see that the proposed PCA-FD outperforms the standard FD with the same number of dimension significantly. The retrieval precision improves from 0.2 to 0.4 near 0.2 recall level.

Fig. 5 also gives two examples of the retrieved shapes by standard FD (Row 1 and 3) and our PCA-FD (Row 2 and 4). Ten most similar retrieved shapes including the query themselves (the shapes in first column in the figure) are given. From them, we also can observe that the retrieved shapes by PCA-FD are more similar to the query than those by the standard FD.

3.2 Incomplete Contour Representation

This experiment is setup for comparing the retrieval performance of the proposed PCA-FD and standard FD in incomplete contour representation. Incomplete contours often occur in shape extraction from images because of noise, complex background or other factors. Fig. 6 show some typical incomplete contour representation examples including depletion, occlusion and segment-wise deletion [11]. The experimental setup for this experiment are as follows:

Shape database Shapes used in this experiment includes two parts: one is for training and the other is for testing. The original incomplete shape database for testing and training is from ICR shape database [1] including two shape data sets: COIL data set containing 20 object classes and modified MPEG-7 data set containing 70 object classes. All these shapes are normalized to 128 $\times$ 128 pixels. Some contours of the COIL data set are shown in Fig. 7. The percentage of remained pixels in contours are chosen from 15 to 90 in steps of 5 percentage, which results in 16 incomplete contours for each object class. Four types of incomplete

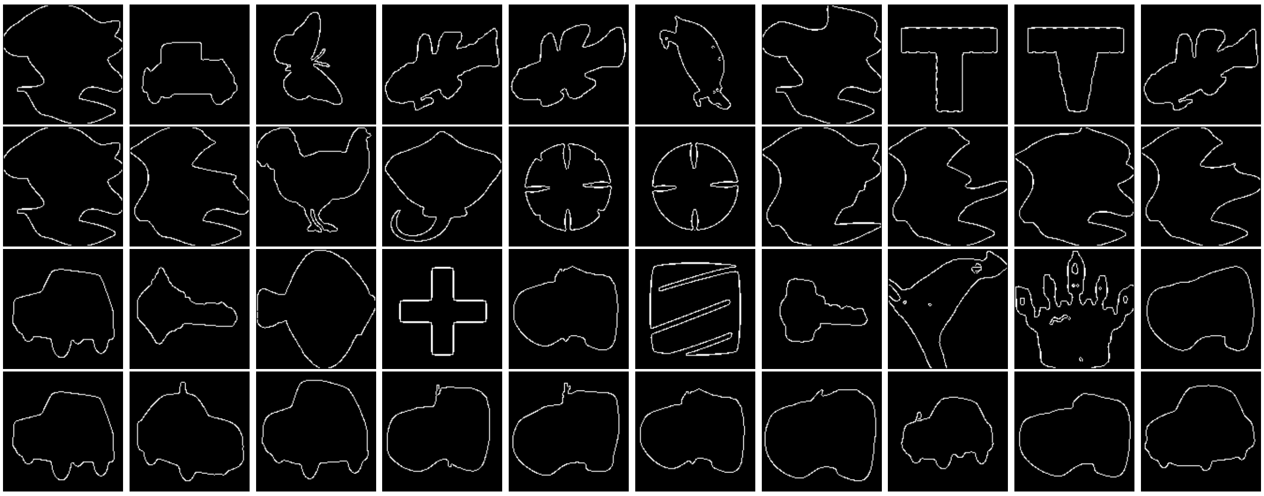


Figure 5. Examples of retrieved results by standard FD (Row 1 and 3) and our PCA-FD (Row 2 and 4).

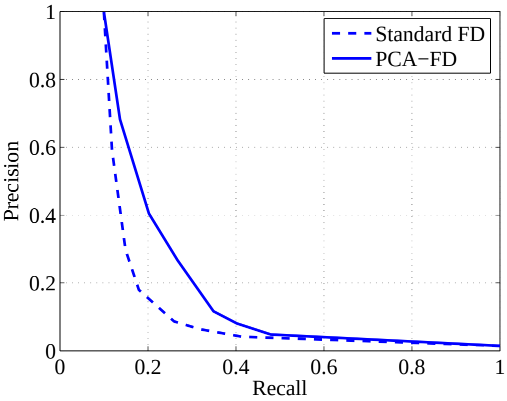


Figure 4. Recall-Precision results of experiment in complete contour representation.

contour including depletion, occlusion (left and right) and segment-wise deletion are used here. We also use two training data sets for training eigenspace in this experiment: one is as the same in complete contour representation including 700 shapes and the other one is the modified incomplete MPEG-7 data set from ICR shape database [1] including 4480 shapes. For more details about the incomplete shape data sets, please refer to [11].

Evaluation criterion Overall Retrieval Rate (ORR) and Recall-Precision are both popular metrics for evaluating the retrieval results. Since ORR can be computed without threshold tuning while Recall-Precision needs to vary thresholds for obtaining the recall-precision graphs, we choose ORR as the evaluation criterion for convenience in this experiment. The retrieval rate is measured by counting the number of shapes from the same class which are found in the first 32 most similar matches (bullseye test). The maximum number of objects from the same class is 16 including itself. The total number of possible correct matches when all 320 shapes are used in turn as queries is thus 5120. ORR is computed as the ratio of the total number of actual correct matches and the total number of possible correct matches

$$r = \frac{\sharp correct\ matches}{\sharp possilbe\ correct\ matches}. \qquad (11)$$

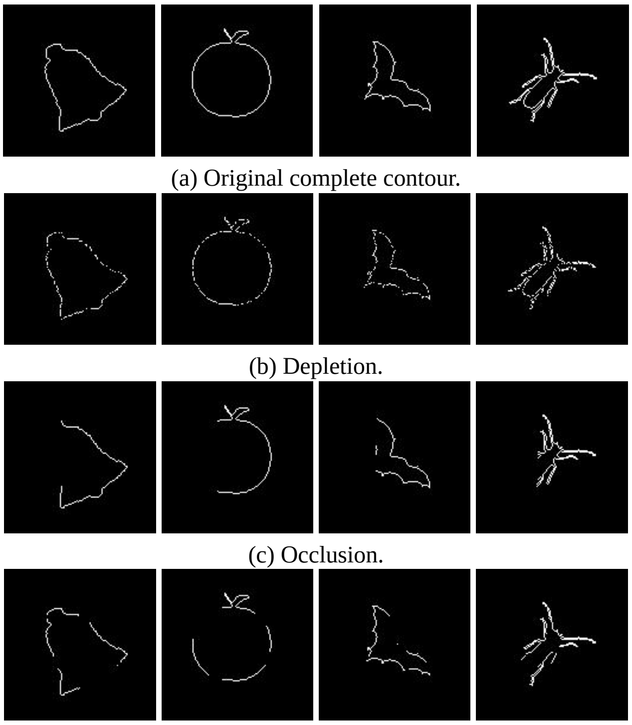
The results of this experiment are shown in Table 1. From the table, we can find that PCA-FD is worse than FD for retrieving incomplete contours and there is no significant differences between the results by using eigenspace trained by complete or incomplete shapes for PCA-FD. The possible reason for this phenomenon is that: the incomplete contour representation changes the Fourier coefficients of low frequencies much, which is used in PCA, that the final descriptors may become very different.

4 Conclusions

In this study, we have presented a new shape descriptor for shape representation and retrieval, called PCA-FD. Two experiments for evaluating its performance in both complete and incomplete contour representations are also given. The experiments show that the propose PCA-FD was significantly more compact and effective than the standard FD

	Depletion	Left Occlusion	Right Occlusion	Segment-wise Deletion
PCA-FD(trained by complete shapes)	0.2337	0.3521	0.3663	0.2498
PCA-FD(trained by incomplete shapes)	0.2427	0.3439	0.3457	0.2623
FD	0.2775	0.3657	0.5462	0.3234

Table 1. Results in incomplete contour retrieval.



(a) Original complete contour.

(b) Depletion.

(c) Occlusion.

(d) Segment-wise deletion.

Figure 6. Examples of incomplete contour representation.

shape descriptor in complete contour representation but is worse than FD in incomplete contour representation.

More suitable shape signatures for incomplete contour representation should be considered in future work and other subspace methods such as Non-negative Matrix Factorization (NMF) [19] which is proved to be valid for partial representation for face recognition should also be tested.

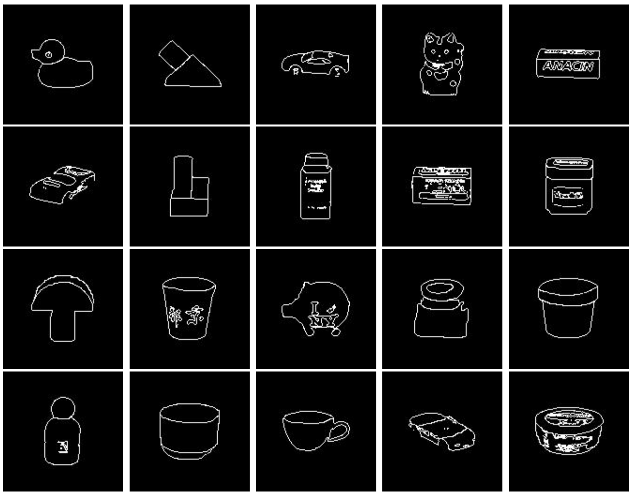


Figure 7. Classes in COIL data set.

References

[1] Icr shape database. *http://www.cs.rug.nl/∼anarta/ Gollin_test_database/*.

[2] Shape data for the mpeg-7 core experiment ce-shape-1. *http://www.cis.temple.edu/∼latecki/TestData/ mpeg7shapeB.tar.gz*.

[3] I. Bartolini, P. Ciaccia, and M. Patella. Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Trans. PAMI*, 27(1):142–147, January 2005.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24(4):509–522, April 2002.

[5] S. Berretti, A. Del Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Trans. Multimedia*, 2(4):225–239, 2000.

[6] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8(6):679–698, November 1986.

[7] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. PAMI*, 19(2):121–132, February 1997.

[8] H. Freeman and L. Davis. A corner finding algorithm for chain coded curves. *TC*, 26:297–303, 1977.

[9] K. Fu and E. Persoon. Shape discrimination using fourier descriptors. *IEEE Trans. PAMI*, 8(3):388–397, May 1986.

[10] K. Fukunaga and W. Koontz. Application of the karhunen-loeve expansion to feature selection and ordering. *TC*, 19(4):311, April 1970.

[11] A. Ghosh and N. Petkov. Robustness of shape descriptors to incomplete contour representations. *IEEE Trans. PAMI*, 27(11):1793–1804, November 2005.

[12] C. Grigorescu and N. Petkov. Distance sets for shape filters and shape recognition. *IEEE Trans. IP*, 12(10):1274–1286, October 2003.

[13] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, February 1962.

[14] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[15] H. Kauppinen, T. Seppanen, and M. Pietikainen. An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification. *IEEE Trans. PAMI*, 17:201–207, 1995.

[16] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *Proc. of the CVPR*, 2:506–513, 2004.

[17] I. Kunttu, L. Lepisto, J. Rauhamaa, and A. Visa. Multiscale fourier descriptors for defect image retrieval. *PRL*, 27(2):123–132, January 2006.

[18] L. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. PAMI*, 22(10):1185–1190, October 2000.

[19] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[20] S. Liao and M. Pawlak. On image analysis by moments. *IEEE Trans. PAMI*, 18:254–266, 1996.

[21] W. Y. Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[22] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. *Proc. of BMVC*, pages 53–62, 1996.

[23] H. Murase and S. Nayar. Detection of 3d objects in cluttered scenes using hierarchical eigenspace. *PRL*, 18(4):375–384, April 1997.

[24] F. Sanchez-Marin. Automatic recognition of biological shapes using the hotelling transform. *Computers in Biology and Medicine*, 31:85–99, 2001.

[25] M. Teague. Image analysis via the general theory of moments. *JOSA*, 70(8):920–930, August 1980.

[26] Q. Tieng and W. Boles. Recognition of 2d object contours using the wavelet transform zero-crossing representation. *IEEE Trans. PAMI*, 19(8):910–916, August 1997.

[27] M. Turk and A. Pentland. Eigenfaces for recognition. *CogNeuro*, 3(1):71–96, 1991.

[28] D. Zhang and G. Lu. A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. *JVCIR*, 14(1):39–57, March 2002.

[29] D. Zhang and G. Lu. A comparative study of fourier descriptors for shape representation and retrieval. *Proc. of ACCV2002*, pages 646–651, 2002.

[30] D. Zhang and G. Lu. Shape-based image retrieval using generic fourier descriptor. *SP:IC*, 17(10):825–848, November 2002.

[31] D. Zhang and G. Lu. Evaluation of mpeg-7 shape descriptors against other shape descriptors. *Multimedia Systems*, 9(1):15–30, 2003.

[32] D. Zhang and G. Lu. Review of shape representation and description techniques. *PR*, 37(1):1–19, January 2004.

[33] D. Zhang and G. Lu. Study and evaluation of different fourier methods for image retrieval. *IVC*, 23(1):33–49, January 2004.

ICA-Based Analysis of Temporal Image Sequence

Naoya Ohnishi
Graduate School of Science and Technology
Chiba University
Yayoicho 1-33, Inage-ku, Chiba, Japan
ohnishi@graduate.chiba-u.jp

Atsushi Imiya
Institute of Media and Information Technology
Chiba University
Yayoicho 1-33, Inage-ku, Chiba, Japan
imiya@faculty.chiba-u.jp

Abstract

In this paper, we present an ICA-based analysis for the temporal image sequence captured by a moving camera. We apply ICA to the optical flow field computed from the sequence of images. ICA-based separation of optical flow derives a obstacle region and a ground plane region in a space. For these applications, we also introduce an ordering criterion of independent components using its variances. The algorithm can be extended to the method for separating multiple planes in the image. We show that the segmented planes have a hierarchical structure in the image. Furthermore, the pyramid transform of an image sequence is used for computing the optical flow field. The pyramidal optical flow fields drive a global and local motion in the image sequence.

1 Introduction

Independent component analysis(ICA) [8] extracts statistically independent features from signals and still images. In this paper, we apply ICA to a temporal image sequence. The temporal image sequence drive the optical flow field [1, 6, 10]. The optical flow is the apparent motion of successive images and is independent of the features in images, unlike edges or corner points in images. Furthermore, optical flow is considered to be fundamental information for navigation and obstacle avoidance in the context of biological data processing [18]. We attempt to apply ICA to the optical flow fields.

Statistical approaches to the optical-flow analysis have also been examined [5, 16, 12]. Fermüller et al. analyzed noise parameters of optical flow using the maximum likelihood [5]. Roth and Black developed a method for learning the spatial statistics of optical flow fields using a Markov random field model [16]. Therefore, it is appropriate to use the statistical properties of optical flow for mobile robot navigation in a real environment.

In neuroscience, it is known that the medial superior temporal (MST) area performs visual motion processing. For motion cognition at the MST area in the brain [19, 9, 15], it is shown that independent components of optical flow are used. Furthermore, since the optical flow field on an image can be represented as a linear combination of independent components of optical flow, we can use ICA for the detection of the dominant plane by separating obstacles and the dominant part in an image. Our application of ICA separates the planes from image sequences.

The planar-area detection and segmentation methods using optical flow are also proposed [4, 17, 20]. Enkelmann [4] proposed the plane-detection method using the model vectors from motion parameters. Santos-Victor and Sandini [17] also proposed a plane-detection algorithm for a mobile robot using the inverse projection of optical flow to a ground floor, assuming that the motion of the camera system mounted on a robot is pure translation with a uniform velocity. However, even if a camera is mounted on a wheel-driven robot, the vision system does not move with a uniform velocity due to mechanical errors of the robot and unevenness of the floor. Therefore, we use ICA for separating the optical flow fields.

For the concurrent detection of local and global motion, we use independent components of optical flow fields on pyramidal layers. It is known that animals, insects, and human beings use the independent component of optical flow fields for visual behavior [9, 15, 18]. In human object recognition, the hierarchical model is proposed [3]. Furthermore, for the computation of optical flow, the pyramid transform of an image sequence is used for the analysis of global motion and local motion [2, 11]. The pyramid transform generates multiple-resolution images as layered images. These layered images are used for computation of optical flow in its original images from the image in the lowest layer. This idea based on the assertion that global motion is described as the collection of local motion. We introduce the application of hierarchical image expression for motion analysis. That is, we develop an algorithm for the detection layered optical flows from a multi resolution image sequence.

2 ICA of optical flow field

In this section, we introduce an algorithm for applying ICA to optical flow fields.

The optical flow is apparent motion of each points computed from successive two images [1]. Setting $I(x, y, t)$ to be time-varying image, the optical flow is computed by solving the equation

$$I_x\dot{x} + I_y\dot{y} + I_t = 0, \tag{1}$$

where $(\dot{x}, \dot{y})^\top$ is the optical flow vector. To solve this singular equation, we adopt the Lucas and Kanade method with the pyramid transform [1, 2, 13].

Setting $I^0(x, y, t) = I(x, y, t)$ as the original image and $I^l(x, y, t)$ as the pyramid transformation of image $I(x, y, t)$ at the layer l, the pyramid representation is expressed as

$$I^{l+1}(x, y, t) = \underset{,\beta\in N_l}{\qquad} a_{\ \beta} I^l(2x - \alpha, 2y - \beta, t), \tag{2}$$

where N_l is the neighborhood of point $(x, y)^\top$ at the layer l and $a_{\ \beta}$ is the weight parameter of the neighborhood pixel. We set N_l as a 3×3 neighborhood and

$$a_{\ \beta} = \begin{cases} \frac{1}{4}, & (\alpha = 0, \beta = 0) \\ \frac{1}{8}, & (\alpha = \pm 1, \beta = 0), (\alpha = 0, \beta = \pm 1) \\ \frac{1}{16}, & (\alpha = \pm 1, \beta = \pm 1) \end{cases} . \tag{3}$$

We use the Lucas-Kanade method with pyramids [2]. Therefore, Eq. (1) can be solved by assuming that the optical flow vectors of pixels are constant in the neighborhood of each pixel. We set the window size to be 5×5. Equation (1) is expressed as a system of linear equations,

$$I_{\ x}\dot{x} + I_{\beta y}\dot{y} + I_t = 0, \quad |\alpha| \leq 2, |\beta| \leq 2 \tag{4}$$

$$I_{\ \beta}(x, y, t) = I(x + \alpha, y + \beta, t + 1), \tag{5}$$

where $I_{\ \beta}(x, y)$ is the spatial neighborhood of a pixel. Optical flow $(\dot{x}, \dot{y})^\top$ is solved by the Lucas-Kanade method [10]. Setting this phase as the estimation of optical flow at the layer 0 of the pyramid representation of the image, we estimate optical flow at layers from 0 to L.

The optical flow is obtained by warping the optical flows of each layer of the pyramid representation. The procedure is illustrated in Fig. 1, which is taken from to Bouguet [2]. We call $\boldsymbol{u}(x, y, t)$, which is a set of optical flow $(\dot{x}, \dot{y})$ computed for all pixels in an image, the optical flow field at time t. Furthermore, we set $\boldsymbol{u}^l(x, y, t)$ to be the optical flow field at the l-th layer in the pyramid transform, where $\boldsymbol{u}^0(x, y, t) = \boldsymbol{u}(x, y, t)$. The traditional optical flow analysis computes $\boldsymbol{u}^0(x, y, t)$. We, however in this paper, use optical flow vectors in all layers in multi-resolution images. This method allow us to extract hierarchical information from optical flows.

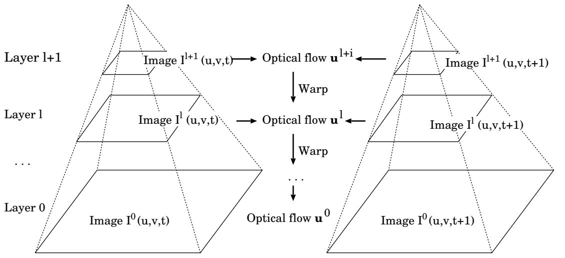


Figure 1. Procedure for computing optical flow in Lucas-Kanade method with pyramids. Optical flow is computed by warping of optical flows of each pyramid layer.

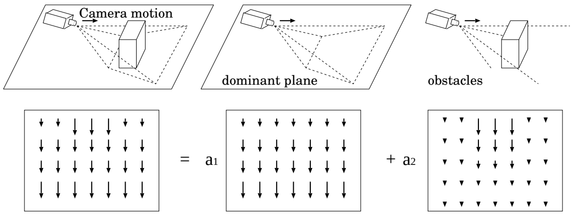
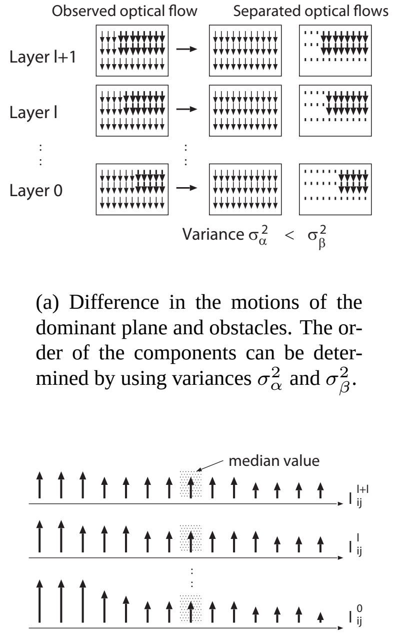


Figure 2. Linear combination of optical flow field in the scene. a_1 and a_2 are mixture coefficients.

Similar to ICA separating mixture signals into independent components, the MST area in the brain separates the motion fields from visual perception into independent components [9, 15]. As previously introduced, we accept the assumption that optical flow fields observed by the moving camera are linear combinations of optical flow fields of the dominant plane and the obstacles. That is, setting $\dot{\boldsymbol{u}}_{\text{dominant}}$ and $\dot{\boldsymbol{u}}_{\text{obstacle}}$ to be optical flow fields of the dominant plane and the obstacles, respectively, the observed optical flow field $\dot{\boldsymbol{u}}$ is approximately expressed by a linear combination of $\dot{\boldsymbol{u}}_{\text{dominant}}$ and $\dot{\boldsymbol{u}}_{\text{obstacle}}$ as

$$\dot{\boldsymbol{u}} = a_1\dot{\boldsymbol{u}}_{\text{dominant}} + a_2\dot{\boldsymbol{u}}_{\text{obstacle}}, \tag{6}$$

where a_1 and a_2 are the mixture coefficients, as shown in Fig. 2. This assumption is numerically and geometrically acceptable if motion displacement is small compared with the size of obstacles, as shown in a numerical experiment. Therefore, ICA is suitable for the separation of optical flow into the independent flow components. For each image in a sequence, we consider that optical flow vectors in the dominant plane correspond to independent components.



(a) Difference in the motions of the dominant plane and obstacles. The order of the components can be determined by using variances σ_α^2 and σ_β^2.

(b) Sorting norm $\boldsymbol{l}$ for determination of output order

Figure 3. Ordering of independent components of optical flow fields.

For ICA of optical flow fields, we align the matrix of two-dimensional vectors to a one-dimensional array as

$$\begin{aligned}\dot{\boldsymbol{u}} &\rightarrow ((\dot{u},\dot{v})_1\ \cdots(\dot{u},\dot{v})_k\cdots(\dot{u},\dot{v})_n)^\top\\ &\rightarrow (\dot{u}_1\cdots\dot{u}_n\ \dot{v}_1\cdots\dot{v}_n)^\top = \mathrm{vec}\dot{\boldsymbol{u}}.\end{aligned} \quad (7)$$

Since the relation $\dot{\boldsymbol{w}} = \alpha\dot{\boldsymbol{u}} + \beta\dot{\boldsymbol{v}}$ leads to the relation $\mathrm{vec}\dot{\boldsymbol{w}} = \alpha\mathrm{vec}\dot{\boldsymbol{u}} + \beta\mathrm{vec}\dot{\boldsymbol{v}}$. These steps are invertible. Therefore, it is possible to extract regions corresponding to $\dot{\boldsymbol{u}}$ and $\dot{\boldsymbol{v}}$ if the observation $\dot{\boldsymbol{w}}$ is decomposed into two independent components $\dot{\boldsymbol{u}}$ and $\dot{\boldsymbol{v}}$. We use this vector, derived from a vector-valued image, as input to ICA.

3 Experimental Results

3.1 Dominant-plane detection by ICA

For the detection of the dominant plane, ICA requires at least two input signals for separation into two independent components. Then, we use optical flow field $\dot{\boldsymbol{u}} = \{(\dot{u},\dot{v})_{ij}^\top\}_{i=1,j=1}^{h,w}$ and planar flow field $\hat{\boldsymbol{u}} = \{(\hat{u},\hat{v})_{ij}^\top\}_{i=1,j=1}^{h,w}$ as the input vectors of ICA, where w and h are the width and the height of an image. Since planar flow is the motion of the dominant plane relative to the robot motion, the use of planar flow is suitable for separation into the dominant plane and obstacles.

Setting $\boldsymbol{v}\ $ and $\boldsymbol{v}_\beta$ to be the output vectors, $\boldsymbol{v}\ $ and $\boldsymbol{v}_\beta$ have ambiguities in those order and length of each component. We are required to determine whether components have optical flow of the dominant plane or of obstacle areas. We solve this problem using the difference between the variances of the norms of $\boldsymbol{v}\ $ and $\boldsymbol{v}_\beta$.

Setting $\boldsymbol{l}\ _{,\beta} = \{l_{ij}\}_{i=1,j=1}^{h,w}$ to be the norm of $\boldsymbol{v}\ _{,\beta} = \{(\dot{u},\dot{v})_{ij}\}_{i=1,j=1}^{h,w}$, that is, $l_{ij} = |(\dot{u},\dot{v})_{ij}|$ and the variance σ^2 is computed as

$$\sigma^2 = \frac{1}{hw}\sum_{i=1,j=1}^{h,w}(l_{ij}-\bar{l})^2, \ \text{where } \bar{l} = \frac{1}{hw}\sum_{i=1,j=1}^{h,w} l_{ij}. \quad (8)$$

The motions of the dominant plane and obstacles in the images are different, and the dominant-plane motion is smooth on the images compared with obstacle motion, as shown in Fig. 3. Consequently, the output signal of obstacle motion has larger variance than the output signal of dominant-plane motion. Therefore, if $\sigma^2\ > \sigma_\beta^2$, we use the norm $\boldsymbol{l}\ $ of output flow field $\boldsymbol{v}\ $ for dominant-plane detection; else we use the norm $\boldsymbol{l}_\beta$ of output flow field $\boldsymbol{v}_\beta$.

Since the planar flow field is subtracted from the optical flow field including obstacle motion, $\boldsymbol{l}$ is constant on the dominant plane. However, the length of $\boldsymbol{l}$ is ambiguous. Then, we use the median value of $\boldsymbol{l}$ for the detection of the dominant plane. Since the dominant plane occupies the largest domain in the image, we compute the distance between $\boldsymbol{l}$ and the median of $\boldsymbol{l}$, as shown in Fig. 3(b). The area which has the median value of the component is detected as the dominant plane. Setting m to be the median value of the elements in $\boldsymbol{l}$, the distance $\boldsymbol{d} = \{d_{ij}\}_{i=1,j=1}^{h,w}$ is

$$d_{ij} = |l_{ij} - m|. \quad (9)$$

We detect the area in which $d_{ij} \approx 0$ as the dominant plane.

Figure 4 shows the procedure of dominant-plane detection from the image sequence using ICA. The procedure for dominant-plane detection by ICA is summarized as follows.
1. Input optical flow field $\dot{\boldsymbol{u}}$ and planar flow field $\hat{\boldsymbol{u}}$ to ICA, and output the optical flow fields $\boldsymbol{v}\ $ and $\boldsymbol{v}_\beta$.
2. Compute the norms $\boldsymbol{l}\ $ and $\boldsymbol{l}_\beta$ from $\boldsymbol{v}\ $ and $\boldsymbol{v}_\beta$, respectively.
3. Compute the variances σ^2 and σ_β^2 from $\boldsymbol{l}\ $ and $\boldsymbol{l}_\beta$, respectively.
4. If $\sigma^2\ > \sigma_\beta^2$, then $\boldsymbol{l} = \boldsymbol{l}\ $, else $\boldsymbol{l} = \boldsymbol{l}_\beta$.
5. Compute the distance $\boldsymbol{d}$ between $\boldsymbol{l}$ and the median of $\boldsymbol{l}$.
6. Detect the area in which $d_{ij} \approx 0$ as the dominant plane.

For the processing of ICA, we use the Fast ICA package for MATLAB [7]. Figures 5 - 9 are experimental results

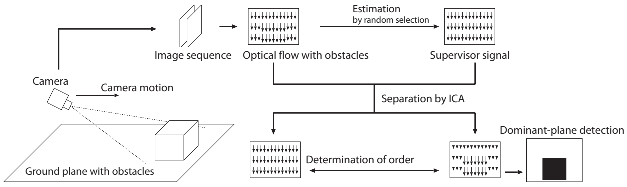
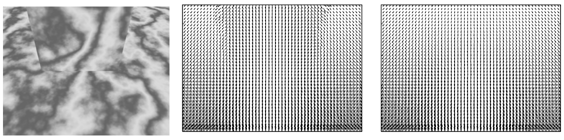


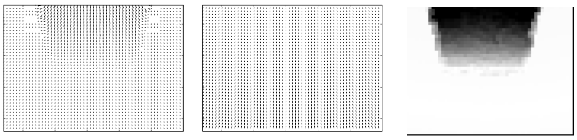
Figure 4. Procedure for dominant-plane detection.



(a) $I(x, y, t)$ (b) $\dot{\boldsymbol{u}}$ (c) $\hat{\boldsymbol{u}}$

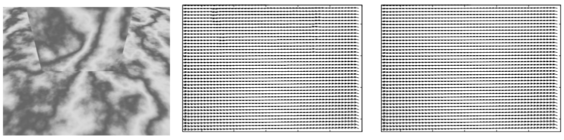
Figure 5. Optical flow fields input to our ICA-based algorithm for translational motion in an environment with one obstacle. (a) Synthetic image. (b) Computed optical flow field $\dot{u}$ for the first input signal. (c) Estimated planar flow field $\hat{u}$ for the second input signal.

for translational and the rotational motions. Figures 10 - 14 are experimental results in the environment with two obstacles. Figures 15 - 17 are experimental results for optical flow fields with noises. Figures 18 and 19 and Figs. 20 and 21 are experimental results for the marbled block sequence and the flower garden sequence, respectively.



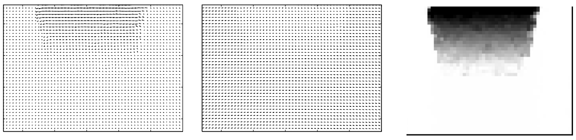
(a) $\boldsymbol{v}_\alpha$ (b) $\boldsymbol{v}_\beta$ (c) d_{ij}

Figure 6. Output optical flow fields and detected dominant plane for Fig. 5. (a) Variance $\sigma^2 = 1.60$. (b) Variance $\sigma_\beta^2 = 0.51$. (c) Detected dominant plane.



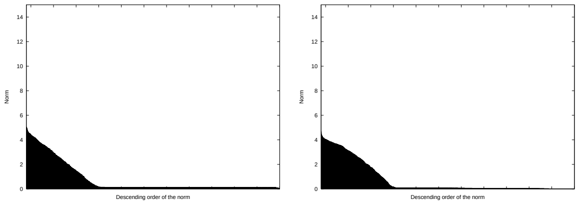
(a) $I(x, y, t)$ (b) $\dot{\boldsymbol{u}}$ (c) $\hat{\boldsymbol{u}}$

Figure 7. Optical flow fields input to our ICA-based algorithm for rotational motion in an environment with one obstacle. Labels of (a), (b), and (c) are same as Fig. 5.



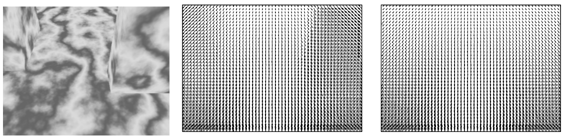
(a) $\boldsymbol{v}_\alpha$ (b) $\boldsymbol{v}_\beta$ (c) d_{ij}

Figure 8. Output optical flow fields and detected dominant plane for Fig. 7. (a) Variance $\sigma^2 = 1.54$. (b) Variance $\sigma_\beta^2 = 0.17$. (c) Detected dominant plane.



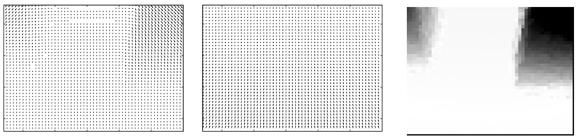
(a) (b)

Figure 9. Sorted norm l of output v . The area where norm l is large corresponds to an obstacle, and the area where $l_{ij} \approx m$ corresponds to the dominant plane. (a)For translational motion. The median value is $m = 0.09$. (b)For rotational motion. The median value is $m = 0.15$.



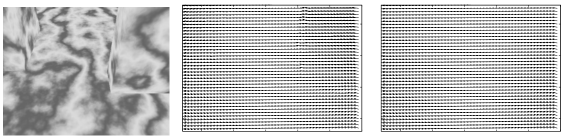
(a) $I(x, y, t)$ (b) $\dot{\boldsymbol{u}}$ (c) $\hat{\boldsymbol{u}}$

Figure 10. Optical flow fields input to our ICA-based algorithm for translational motion in an environment with two obstacles. Labels of (a), (b), and (c) are same as Fig. 5.



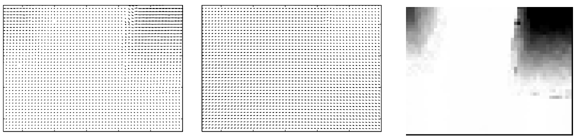
(a) $\boldsymbol{v}_\alpha$ (b) $\boldsymbol{v}_\beta$ (c) d_{ij}

Figure 11. Output optical flow fields and detected dominant plane for Fig. 12. (a) Variance $\sigma^2 = 1.60$. (b) Variance $\sigma_\beta^2 = 0.55$. (c) Detected dominant plane.



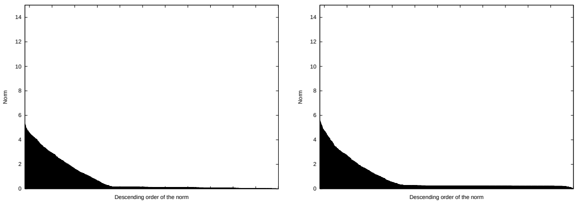
(a) $I(x, y, t)$ (b) $\dot{\boldsymbol{u}}$ (c) $\hat{\boldsymbol{u}}$

Figure 12. Optical flow fields input to our ICA-based algorithm for rotational motion in an environment with two obstacles. Labels of (a), (b), and (c) are same as Fig. 5.



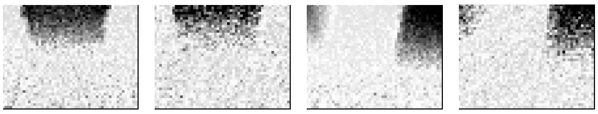
(a) $\boldsymbol{v}_\alpha$ (b) $\boldsymbol{v}_\beta$ (c) d_{ij}

Figure 13. Output optical flow fields and detected dominant plane for Fig. 12. (a) variance $\sigma^2 = 1.40$. (b) variance $\sigma_\beta^2 = 0.12$. (c) Detected dominant plane.



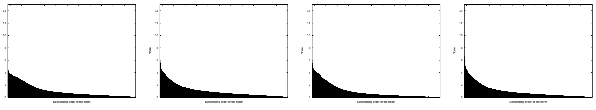
(a) (b)

Figure 14. Sorted norm l of output v . The area where norm l is large corresponds to an obstacle, and the area where $l_{ij} \approx m$ corresponds to the dominant plane. (a)For translational motion. The median value is $m = 0.14$. (b)For rotational motion. The median value is $m = 0.25$.



(a) (b) (c) (d)

Figure 15. Results obtained using optical flows with error. (a) Translational motion in an environment with one obstacle. (b) Rotational motion in an environment with one obstacle. (c) Translational motion in an environment with two obstacles. (d) Rotational motion in an environment with two obstacles.



(a) (b) (c) (d)

Figure 16. Sorted norm l of output v . (a) - (d) correspond to Figs. 15(a) - (d), respectively. (a) Median value $m = 0.60$. (b) Median value $m = 0.68$. (c) Median value $m = 0.45$. (d) Median value $m = 0.60$.

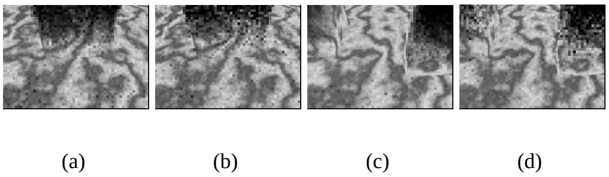


Figure 17. Obstacle area of the experimental results overlapped with the images. (a) Translational motion in an environment with one obstacle. (b) Rotational motion in an environment with one obstacle. (c) Translational motion in an environment with two obstacles. (d) Rotational motion in an environment with two obstacles.

3.2 Iterative multiple plane segmentation

Using the dominant-plane-detection algorithm iteratively, we develop an algorithm for multiple-plane segmentation in an image. After removing the region corresponding to the dominant plane from an image, we can extract the second dominant planar region from the image. Then, it is possible to extract the the third dominant plane by removing the second dominant planar area. This process is expressed as

$$D_k = \begin{cases} \mathbf{A}(R \setminus D_{k\ \ 1}), & k \geq 2, \\ \mathbf{A}(R), & k = 1, \end{cases} \quad (10)$$

where $\mathbf{A}$, R, D_k stand for the dominant-plane-extraction algorithm, the region of interest observed by a camera, and the k-th dominant planar area, respectively. The algorithm is stopped after iterated to a pre-determined iteration time or the size of k-th dominant plane is smaller th pre-determined size.

Setting R to be the root of the tree, this process derives a binary tree such that

$$R\langle D_1, R \setminus D_1 \langle D_2, R_2 \setminus D_2 \langle \cdots, \rangle \rangle \quad (11)$$

Assuming that D_1 is the ground plane on which the robot moves, D_k for $k \geq 2$ is the planar areas on the obstacles. Therefore, this tree expresses the hierarchical structure of planar areas on the obstacles. We call this tree the binary tree of planes. Using this tree constructed by the dominant-plane detection algorithm, we obtain geometrical properties of planes in a scene. For example, even if an object exists in a scene and it lies on D_k $k \geq 2$, the robot can navigate ignoring this object, using the tree of planes. Figure 22 and 23 are experimental results for the simulated image sequence. Figures 24 and 25 are experimental results on detecting multiple planes for the marbled-block sequence and the flower garden sequence, respectively.

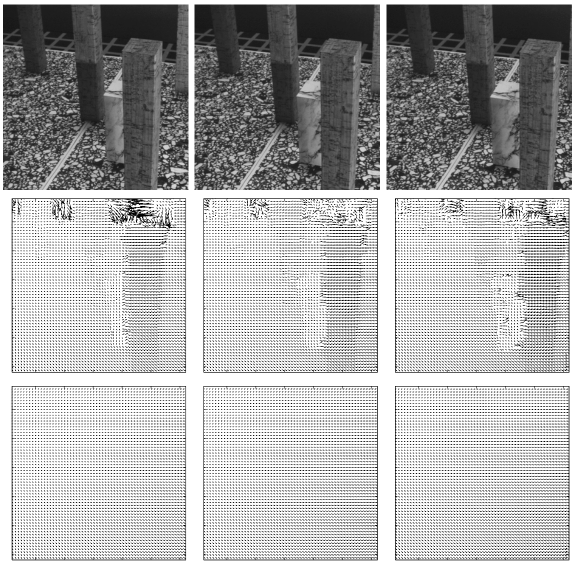


Figure 18. Input optical flow fields to our ICA-based algorithm for the Marbled-Block sequence. The first, second, and third rows show the image sequence for translational motion, computed optical flow $\dot{u}$ for the first input signal, and estimated planar flow $\hat{u}$ for the second input signal, respectively.

3.3 Obstacle detection using ICA on pyramid layers

Our algorithm is processed at layers $l = 0, \cdots, L$ in the pyramid transform. Using the optical flow field $\boldsymbol{u}^l(x, y, t)$ at layer l, we detect obstacles in a image sequence.

Figure 26 shows that, setting O_l to be the obstacle region on the l-th layer, the hierarchical expression of obstacles satisfies the relations

$$O_0 \subset O_1 \subset \cdots \subset O_L \text{ and } D^L \subset D^{L\ \ 1} \subset \cdots \subset D^0 \quad (12)$$

for the dominant plane $D^K = R^0_k$. These relations imply that a pair $C_l = (D^l, O_l)$ shows global and local configuration in the work space for a larger and a smaller l, respectively. This hierarchical relation is automatically detected from a pyramid-based hierarchical expression of images for optical flow computation. The system uses selectively C_l for navigation and spatial perception.

We show experimental results on the detection of obstacles in an image sequence at each layer. For the computation of optical flow, we use the Lucas-Kanade method with pyramids [2]. We set the maximum layer $L = 3$. For the visual representation of the results of obstacle detection, the value of d_{ij} in Eq. (9) is normalized in the range from 0 to

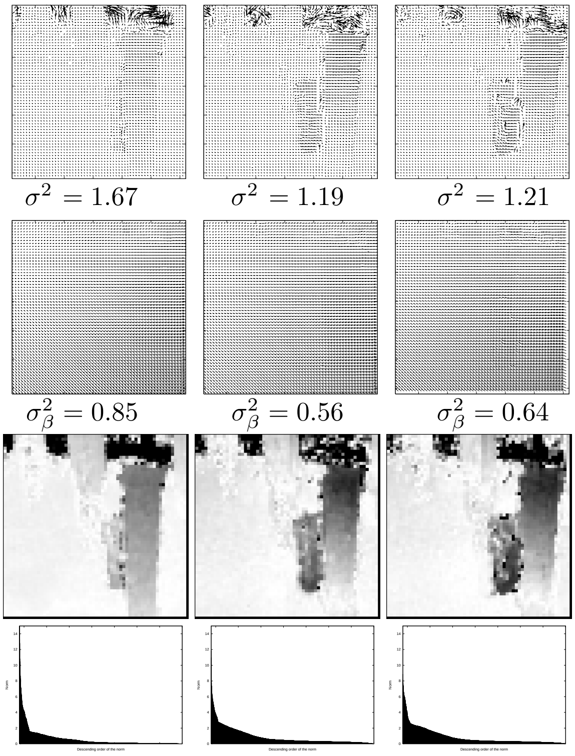


Figure 19. Output optical flow fields and detected dominant plane for Fig. 18. The first, second, third, and fourth rows show output signal v , output signal v_β, images of the dominant plane, and sorted norm l of output v , respectively.

255. The image of the detected obstacle $D^l(u, v)$ at the l-th layer is defined as

$$D^l(i, j) = \frac{d_{ij} \times 255}{\max(d^l_{ij})}, \tag{13}$$

where d^l_{ij} is d_{ij} at the l-th layer.

The Marbled-Block image sequence and captured images in a real environment are used for the experiment. Fig. 26 shows the Marbled-Block images at each layer, the computed optical flow fields at each layer from each image, and the detected obstacle at each layer. In this figure, the black and white region indicate the obstacle and dominant plane, respectively.

Figure 27 shows the captured images in a real environment using a mobile robot which moves toward the obstacle in front of the robot, the computed optical flow fields at each layer from each image, and detected obstacles at each layer.

Another experimental results are shown in Fig. 28.

These examples show that in each layer the obstacle-regions are detected. Therefore, the algorithm detects the global configuration of obstacles from higher layer images, though the lower layer images allows us to detect the detailed configuration of obstacles. The hierarchical description of the layered obstacle-region [14] and the extraction of the navigation-direction from this hierarchical expression are future problems.

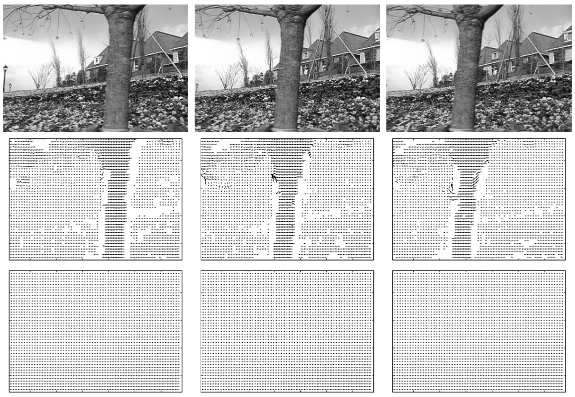


Figure 20. Input optical flow fields to our ICA-based algorithm for the Flower Garden sequence. The first, second, and third rows show image sequence for translational motion, computed optical flow $\dot{u}$ for the first input signal, and estimated planar flow $\hat{u}$ for the second input signal, respectively.

4 Conclusions

We present an algorithm for detecting the hierarchy of planar areas in an image sequence using independent components of optical flow fields. The optical flow fields are observed through a moving camera. The use of the ICA for the optical flow enables the robot to detect a feasible region in which robot can move without any preknowledge. The presented experimental results support the application of our method to the navigation and path planning of a mobile robot with a vision system.

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.

[2] J. Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. In *Intel Corporation, Microprocessor Research Labs, OpenCV Documents*, 1999.

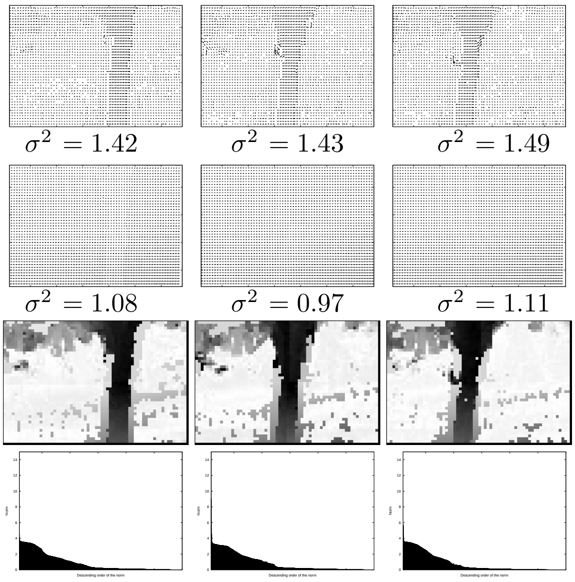


Figure 21. Output optical flow fields and detected dominant plane for Fig. 20. The first, second, third, fourth, and fifth rows show output signal v , output signal v_β, images of the dominant plane, and sorted norm l of output v , respectively.

[3] R. G. Domenella and A. Plebe. A neural model of human object recognition development. In *1st International Symposium on Brain, Vision and Artificial Intelligence*, pages 116–125, 2005.

[4] W. Enkelmann. Obstacle detection by evaluation of optical flow fields from image sequences. *Image and Vision Computing*, 9:160–168, 1991.

[5] C. Fermüller, D. Shulman, and Y. Aloimonos. The statistics of optical flow. *Computer Vision and Image Understanding*, 82:1–32, 2001.

[6] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] J. Hurri, H. Gavert, J. Sarela, and A. Hyvarinen. The fastica package for matlab. website: http://www.cis.hut.fi/projects/ica/fastica/.

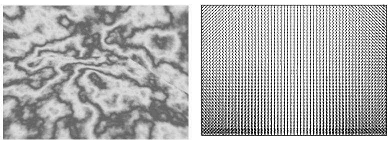


Figure 22. Simulated image and its optical flow field. There are three orthogonal planes in front of the camera.

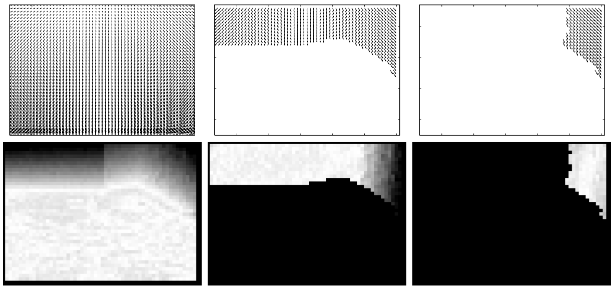


Figure 23. Experimental results at first, second, and third steps. Top row: planar flow fields. bottom row: dominant plane.

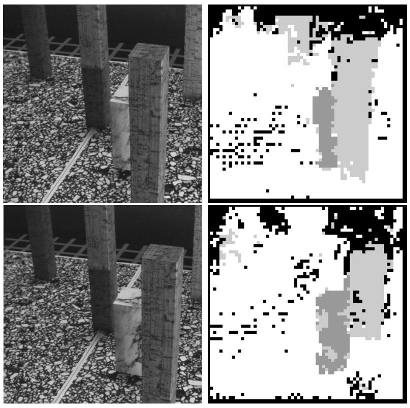


Figure 24. Image and results estimated from the marbled-block sequence. The white area is the first dominant plane. The light-gray and dark-gray areas are the second and third dominant plane.

[8] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and application. *Neural Networks*, 13:411–430, 2000.

[9] M. A. Jabri, K.-Y. Park, S.-Y. Lee, and T. J. Sejnowski. Properties of independent components of self-motion optical flow. In *IEEE International Symposium on Multiple-Valued Logic*, pages 355–362, 2000.

[10] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th IJCAI*, pages 674–679, 1981.

[11] M. R. Mahzoun, J. Kim, S. Sawazaki, K. Okazaki, and S. Tamura. A scaled multigrid optical flow algorithm based on the least rms error between real and estimated second images. *Pattern Recognition*, 32:657–670, 1999.

[12] N. Ohnishi and A. Imiya. Dominant plane detection using optical flow and independent component analysis. In *1st International Symposium on Brain, Vision and Artificial Intelligence*, pages 478–496, 2005.

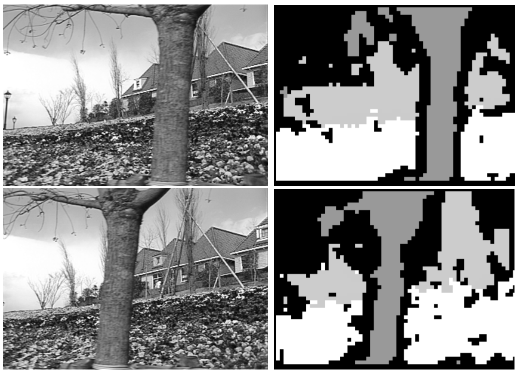


Figure 25. Image and results estimated from the flower garden sequence. The white area is the first dominant plane. The light-gray and dark-gray areas are the second and third dominant plane.

[13] N. Ohnishi and A. Imiya. Dominant plane detection from optical flow for robot navigation. *Pattern Recognition Letters*, 27:1009–1021, 2006.

[14] N. Ohnishi and A. Imiya. Model-based plane-segmentation using optical flow and dominant plane. In *3rd International Conference on MIRAGE*, LNCS 4418, pages 295–306, 2007.

[15] K.-Y. Park, M. Jabri, S.-Y. Lee, and T. J. Sejnowski. Independent components of optical flows have mstd-like receptive fields. In *2nd International Workshop on ICA and Blind Signal Separation*, pages 597–601, 2000.

[16] S. Roth and M. J. Black. On the spatial statistics of optical flow. In *IEEE International Conference on Computer Vision*, pages 42–49, 2005.

[17] J. Santos-Victor and G. Sandini. Uncalibrated obstacle detection using normal flow. *Machine Vision and Applications*, 9:130–137, 1996.

[18] L. M. Vaina, S. A. Beardsley, and S. K. Rushton. *Optic flow and beyond*. Kluwer Academic Publishers, 2004.

[19] J. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. In *Royal Society of London, Series B*, volume 265, pages 359–366, 1998.

[20] M. Zucchelli, J. Santos-Victor, and H. Christensen. Multiple plane segmentation using optical flow. In *British Machine Vision Conference*, 2002.

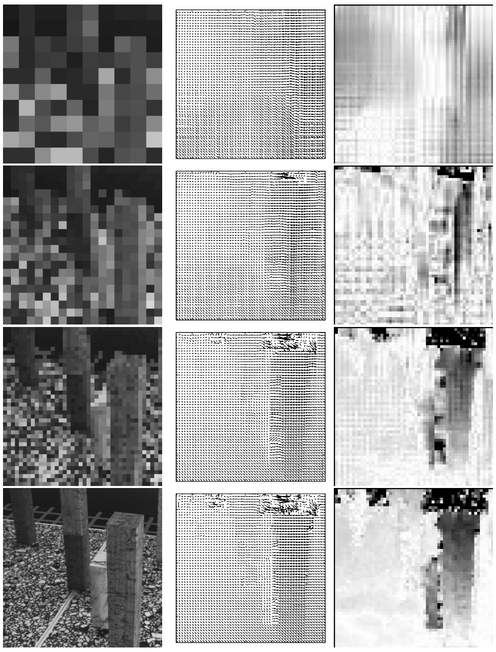


Figure 26. Experimental results for the marbled-block sequence. Left column: pyramidal representations of the image I^i. Middle column: optical flow fields u^i at each layer. Right column: detected obstacle D^i at each layer.

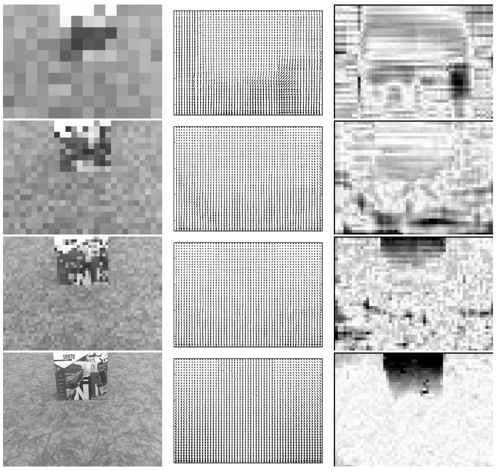
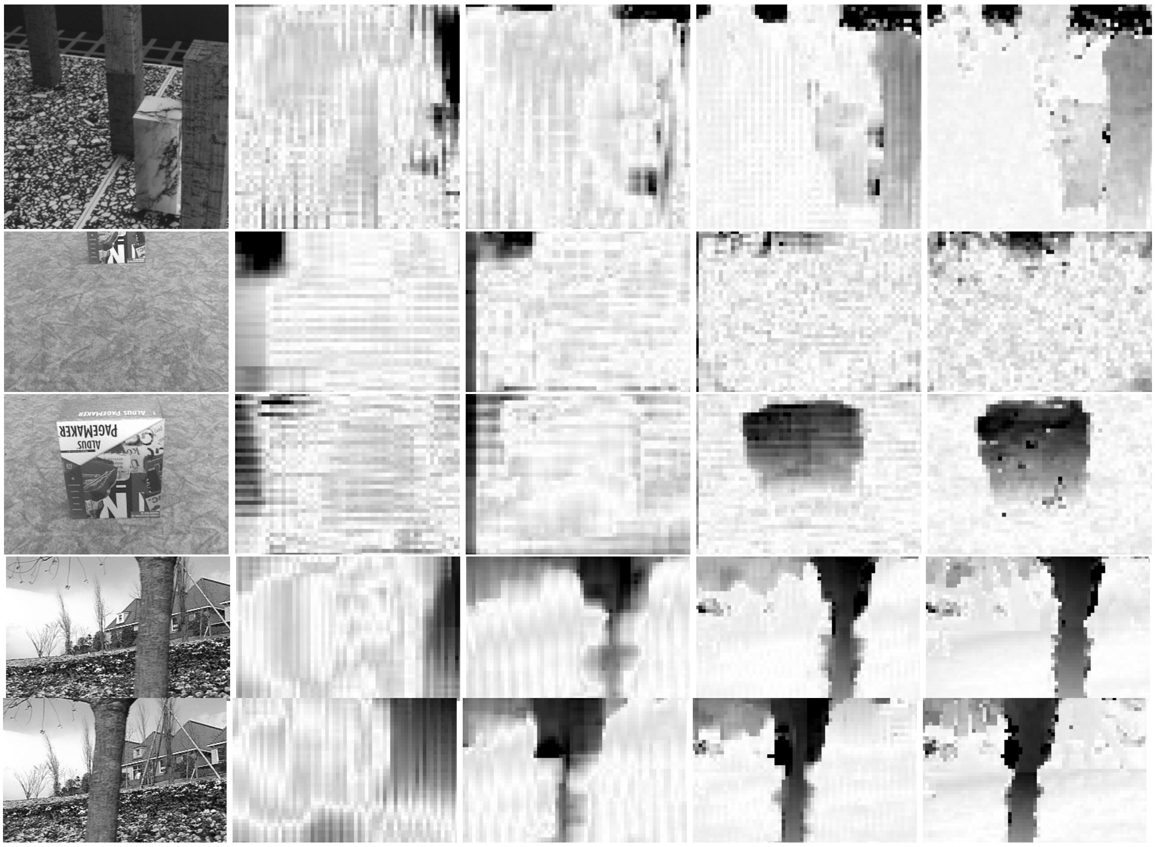


Figure 27. Experimental results for the real image sequence. Left column: pyramidal representations of the image I^i. Middle column: optical flow fields u^i at each layer. Right column: detected obstacle D^i at each layer.



(a) I^0 (b) D^3 (c) D^2 (d) D^1 (e) D^0

Figure 28. Experimental results for the marbled block sequence, the real image sequence, and the flower garden sequence. (a) Original image. (b) (c) (d) and (e) are detected obstacles at the layers 3, 2, 1, and 0, respectively.

A Supervised Method to Chart Multiple Manifolds

Dan Zhang, Yangqiu Song, Qifeng Qiao, Zhenwei Shi, Changshui Zhang
State Key Laboratory on Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Automation,Tsinghua University, Beijing 100084, China
Email: {dan-zhang05, songyq99, qqf05}@mails.tsinghua.edu.cn, {shizhenwei, zcs}@mail.tsinghua.edu.cn

***Abstract*—The discovery of the manifolds has long been a hot topic in computer vision. In many practical problems, high-dimensional data poses a great obstacle to the researchers. But these data points are often sampled from several low-dimensional sub-manifolds. Therefore, charting the sub-manifolds in one coordinate system will help visualize them simultaneously. However, algorithms developed so far all have their own limitations in solving this problem. In this paper, we propose a new supervised method to capture multiple sub-manifolds.**

I. Introduction

Dimensionality Reduction is an interesting topic in the computer vision community. One of the main reasons is that real-world problems are often confronted with high-dimensional data points. In most cases, there exist low-dimensional structures, i.e. sub-manifolds, underlying these high dimensional data points. It is beneficial to design efficient dimensionality reduction algorithms to find these intrinsic manifolds. The main goal of our paper is to design such an algorithm. Unlike the classification task, whose main goal is to classify unlabeled data points accurately, we focus on how to arrange the labeled data in a space, whose dimension does not exceed three, to help the user visualize these sub-manifolds conveniently.

Most of the dimensionality reduction algorithms can be categorized into two groups: unsupervised and supervised. As for the unsupervised group, some of the previous works, such as Principle Component Analysis (PCA) [1], Locally Linear Embedding (LLE) [9], Isomap [10], Locality Preserving Projections(LPP) [7] and Laplacian Eigenmaps (LE) [6] have been developed to discover the intrinsic data structures, regardless of the labels of these data points. For the supervised ones, Linear Discriminant Analysis (LDA) [1], and its variants, such as Kernel LDA [3], null space LDA [4] and uncorrelated LDA [5], utilize the available discriminant information of the data points, and have shown a great success.

From another point of view, these dimensionality reduction algorithms can also be divided into linear and nonlinear ones. PCA, LPP and LDA, seeking linear maps from high-dimensional feature space to a lower one, are among the linear ones, while Kernel PCA, Kernel LDA, Laplacian Eigenmaps do not restrict this map to be linear, and therefore are categorized into the nonlinear group.

The authors in [14] present a nonlinear algorithm. However, they presume that all the data points lie on the same manifold, and do not take into account the discriminant information. Authors in [15] propose Supervised LLE, and can handle the situation when multiple-manifolds exist. But their main focus is on classification, rather than charting the sub-manifolds. As illustrated by the authors, their algorithm tends to lose the within-class structure, and is not suitable to treat the charting task. The same problem also exists in Kernel LDA. In [11], the authors propose a Supervised Isomap to perform this task. In their paper, the discriminant information is used to redefine the distances between data points. They aim to reduce the distances between data points sharing the same labels, as well as enlarge the distances between data points with different labels. Their motivation is quite reasonable. However, the redefinition of the distance function is too empirical.

In most cases, supervised algorithms can utilize more information than unsupervised methods and nonlinear algorithms have fewer mapping restrictions than the linear ones. Therefore, charting the manifolds through supervised nonlinear dimensionality reduction is a good choice. In this paper, we propose a new Supervised Nonlinear Dimensionality Reduction (SNDR) algorithm to discover the sub-manifolds for each category. SNDR is a nonlinear method, and does not restrict the mapping be linear. Unlike the unsupervised algorithms, SNDR utilizes the class labels of the input data points to guide the dimensionality reduction work.

The rest of the paper is organized as follows: In Section II, we will formulate the multiple mainfolds charting problem. A related method will be given in Section III. We will elaborate our proposed algorithm in Section IV. In Section V, the experimental results are presented. In the end, conclusions will be drawn in Section VI.

II. Problem Statement

We are given a set of n data points $\{\mathbf{x_i}, i = 1, 2, ...n\}$, which are sampled from k categories, as well as their corresponding labels $l(\mathbf{x_i}) \in \{1, 2, ..., k\}$. In most cases, for each category, a sub-manifold exists. The goal is to help chart, in a coordinate system, these sub-manifolds as faithfully as possible. In the context, we will use the term 'observed set' to denote the set of these data points and hence the 'observed data points' refers to the data points in the 'observed set'. The feature space after dimensionality reduction will be referred to as 'reduced feature space'.

The work was supported by the National Science Foundation of China (60475001, 60605002)

III. Linear Transformation

Exploring data structures from a global way, such as PCA, often gives undesirable results. So, some recent algorithms, such as [13], [12] and [8], have considered solving it from a local way.

For each data point $\mathbf{x_i}$, we first find its K nearest neighbors. Then these K nearest neighbors are split into two sets: the data points with the same label as $l(\mathbf{x_i})$ ($l(\mathbf{x_i})$ denotes the label of $\mathbf{x_i}$), i.e. $N_w(\mathbf{x_i})$ and the data points with labels different from $l(\mathbf{x_i})$, i.e. $N_b(\mathbf{x_i})$. Specifically,

$$\begin{array}{l} N_w(\mathbf{x_i}) = \{\mathbf{x_i^m} | l(\mathbf{x_i^m}) = l(\mathbf{x_i}), 1 \le m \le K\} \\ N_b(\mathbf{x_i}) = \{\mathbf{x_i^m} | l(\mathbf{x_i^m}) \neq l(\mathbf{x_i}), 1 \le m \le K\} \end{array}$$

Then, two graphs, the within-class graph G_w and the between-class graph G_b, are constructed, with each node representing a data point and the adjacency relationship between two data points representing an edge. The corresponding adjacency matrices, $\mathbf{W_w}$ and $\mathbf{W_b}$, are determined as follows:

$$W_{w,mn} = \begin{cases} 1, & if\, \mathbf{x_m} \in N_w(\mathbf{x_n})\, or\, \mathbf{x_n} \in N_w(\mathbf{x_m}) \\ 0, & otherwise \end{cases}$$

$$W_{b,mn} = \begin{cases} 1, & if\, \mathbf{x_m} \in N_b(\mathbf{x_n})\, or\, \mathbf{x_n} \in N_b(\mathbf{x_m}) \\ 0, & otherwise \end{cases}$$

Let $\mathbf{y} = (y_1, y_2, ..., y_n)^T$ be a one-dimensional representation for the data points in the observed set. Suppose $\mathbf{a}$ is a projection vector. i.e. $\mathbf{y}^T = \mathbf{a}^T\mathbf{X}$, where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$. Then, the objective is to seek a projection vector $\mathbf{a}$ to minimize $\sum_{ij}(y_i - y_j)^2 W_{w,ij}$ and maximize $\sum_{ij}(y_i - y_j)^2 W_{b,ij}$, simultaneously. In this way, the local margins between different categories can be maximized. We find that,

$$\begin{array}{l} \frac{1}{2}\sum\limits_{ij}(y_i - y_j)^2 W_{w,ij} \\ = \frac{1}{2}\sum\limits_{ij}(\mathbf{a}^T\mathbf{x_i} - \mathbf{a}^T\mathbf{x_j})^2 W_{w,ij} \\ = \mathbf{a}^T\mathbf{X}\mathbf{D}_w\mathbf{X}^T\mathbf{a} - \mathbf{a}^T\mathbf{X}\mathbf{W}_w\mathbf{X}^T\mathbf{a} \end{array} \qquad (1)$$

$$\begin{array}{l} \frac{1}{2}\sum\limits_{ij}(y_i - y_j)^2 W_{b,ij} \\ = \frac{1}{2}\sum\limits_{ij}(\mathbf{a}^T\mathbf{x_i} - \mathbf{a}^T\mathbf{x_j})^2 W_{b,ij} \\ = \mathbf{a}^T\mathbf{X}(\mathbf{D}_b - \mathbf{W}_b)\mathbf{X}^T\mathbf{a} \\ = \mathbf{a}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{a} \end{array} \qquad (2)$$

$\mathbf{D_w}$ and $\mathbf{D_b}$ are diagonal matrices, with entries $D_{w,ii} = \sum_j W_{w,ij}$ and $D_{b,ii} = \sum_j W_{b,ij}$. The Laplacian matrix for the between-class graph G_b is $\mathbf{L_b} = \mathbf{D_b} - \mathbf{W_b}$. By restricting $\mathbf{a^TXD_wX^Ta} = 1$, Eq. (1) turns to $1 - \mathbf{a^TXW_wX^Ta}$. Also taking Eq. (2) into account, maximizing the local margins is equivalent to solving the following optimization problem:

$$\mathop{\arg\max}_{\substack{a \\ \mathbf{a^TXD_wX^Ta}=1}} \quad \mathbf{a^TX}(\eta\mathbf{L_b} + (1-\eta)\mathbf{W_w})\mathbf{X^Ta} \qquad (3)$$

η is a trade-off parameter ($0 \le \eta \le 1$). This optimization problem can be solved efficiently by the generalized eigenvalue problem.

The objective function Eq. (3) explores the geometry of the data manifolds, characterizes both the geometrical and discriminant structures by utilizing the within-class and between-class graph. Experiments on face recognition give us an impressive result [8]. However, this algorithm restricts that the map should be linear and may sacrifice some manifold information when the dimension of the reduced feature space is very low. Therefore, this is not suitable for the visualization.

In this paper, we seek for a nonlinear map to visualize the sub-manifolds underlying the high dimensionality data points.

IV. The Proposed Algorithm

A. The Construction of The Adjacency Matrix

In SNDR, two graphs are constructed, i.e. the within-class graph $G_w^{'}$ and between-class graph $G_b^{'}$. However, the method how we construct these two graphs is different. For each data point $\mathbf{x_i}$ ($1 \le i \le n$), k_w nearest neighbors with labels $l(\mathbf{x_i})$, as well as k_b nearest neighbors with labels different from $l(\mathbf{x_i})$ are selected. k_w and k_b are two parameters that are determined beforehand. Then, the adjacency matrices $\mathbf{W_w^{'}}$ and $\mathbf{W_b^{'}}$ for $G_w^{'}$ and $G_b^{'}$ can be formulated as follows:

$$W_{w,mn}^{'} = \begin{cases} 1, & if\, \mathbf{x_m} \in N_w^{'}(\mathbf{x_n})\, or\, \mathbf{x_n} \in N_w^{'}(\mathbf{x_m}) \\ 0, & otherwise \end{cases} \qquad (4)$$

$$W_{b,mn}^{'} = \begin{cases} 1, & if\, \mathbf{x_m} \in N_b^{'}(\mathbf{x_n})\, or\, \mathbf{x_n} \in N_b^{'}(\mathbf{x_m}) \\ 0, & otherwise \end{cases} \qquad (5)$$

Here, $N_w^{'}(\mathbf{x_i})$ denotes the k_w nearest neighbors with the same label as $l(\mathbf{x_i})$, while $N_b^{'}(\mathbf{x_i})$ refers to the k_b nearest neighbors with labels different from $l(\mathbf{x_i})$. The between-class Laplacian matrix for the graph $G_b^{'}$ can be defined as $\mathbf{L_b^{'}} = \mathbf{D_b^{'}} - \mathbf{W_b^{'}}$. $\mathbf{D_b^{'}}$ and $\mathbf{D_w^{'}}$ are diagonal matrices, with diagonal entries $D_{b,ii}^{'} = \sum_j W_{b,ij}^{'}$, $D_{w,ii}^{'} = \sum_j W_{w,ij}^{'}$. Remind that in Section III, for $\mathbf{x_i}$, $N_w(\mathbf{x_i})$ and $N_b(\mathbf{x_i})$ are partitioned within its K nearest neighbors. By finding for each $\mathbf{x_i}$ the k_w nearest neighbors with labels $l(\mathbf{x_i})$, $D_{w,ii}^{'}$ will always be nonzero, and $\mathbf{D_w^{'}}$ can be nonsingular. This modification is reasonable, since it still reflects the local margin information between different categories, but from a different perspective. The reason why we require $\mathbf{D_w^{'}}$ to be nonsingular will be elaborated in the section IV-B.

B. Charting The Sub-manifolds on The Observed Data Set

Still assume that $\mathbf{y} = (y_1, y_2, ..., y_n)^T$ is a one-dimensional representation for the observed data points. Our aim is to find a nonlinear projection that can faithfully preserve the pairwise relationship in the low dimensional space by minimizing the within-class scatter $\sum\limits_{ij}(y_i - y_j)^2 W_{w,ij}^{'}$ and maximizing the between-class scatter $\sum\limits_{ij}(y_i - y_j)^2 W_{b,ij}^{'}$, simultaneously. Inspired by Eq. (3), the optimization problem can be formulated as:

$$\mathop{\arg\max}_{\substack{\mathbf{y} \\ \mathbf{yD_w^{'}y^T}=1}} \ \mathbf{y}(\eta\mathbf{L_b^{'}} + (1-\eta)\mathbf{W_w^{'}})\mathbf{y^T} \qquad (6)$$

Here, we do not restrict $\mathbf{y}$ be a linear transformation $\mathbf{y} = \mathbf{a}^T\mathbf{X}$. Conversely, we directly estimate it from the optimization problem, which results in a non-linear feature space. This amounts to solving the generalized eigenvalue problem:

$$(\eta\mathbf{L}_{\mathbf{b}}^{'} + (1-\eta)\mathbf{W}_{\mathbf{w}}^{'})\mathbf{y}^{\mathbf{T}} = \lambda\mathbf{D}_{\mathbf{w}}^{'}\mathbf{y}^{\mathbf{T}} \tag{7}$$

η is the trade-off parameter, $0 \leq \eta \leq 1$. Still note that why we require $\mathbf{D}_{\mathbf{w}}^{'}$ to be nonsingular in Section 4.1. That's because if $\mathbf{D}_{\mathbf{w}}^{'}$ is singular, it would deteriorate the solution of Eq. (7). Let the column vector $\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_d}$ denote the solutions of equation (7), ordered according to eigenvalues $\lambda_1 > ... > \lambda_d$. the embedding will be given by:

$$\mathbf{Y} = (\mathbf{y_1}, \mathbf{y_2}, ...\mathbf{y_d})^T \tag{8}$$

C. The Whole Algorithm

For the observed set, the visualization procedure can be shown in Table I.

V. Experiments

In this section, we present experimental results on the MNIST handwritten digit test set [1]. This data set contains 10,000 28 × 28 pixels images, with 1000 for each category and 10 categories in total.

In our experiment, for each category, 100 images are randomly selected as the observed set to chart the manifolds. k_b and k_w are both set to 10, and the trade-off η is set to 0.1. The result of SNDR is shown in Fig. 1. As for comparison, the charting results of some representative algorithms such as LDA, Kernel LDA, Local Sensitive Discriminative Analysis(LSDA) [8] and Laplacian Eigenmaps are shown in Fig. 2. It can be seen that SNDR can separate different manifolds very well. From the several magnified manifolds, we can see that the multiple manifolds for these categories are well retained. For example, the lean degrees of the digit 1 and 7 increase with the x-coordinate.

LDA (Fig. 2(a))doesn't provide a good charting result because its underlying assumption is that the data distribution of each category is gaussian and LDA is itself a linear method. Although Kernel LDA provides a nonlinear supervised map, it maps the data points with the same labels onto the same point in the reduced feature space, as can be shown in Fig. 2(b), and therefore loses the inner sub-manifold struture for each category. LSDA(Fig. 2(c)) restricts the map to be linear and may lose some manifold information in such low dimensions. Therefore, it can not give a good charting result. Laplacian Eigenmaps (Fig. 2(d))is an unsupervised method and can not utilize the discriminative information, so it can not separate the manifold of each category well.

We provide a comparison result with Supervised Isomap [11] in Fig. 2(e). The observed set in this figure is the same as that in Fig. 1. But it is hard to distinguish the inner structure for each category. This is because, in Supervised Iso-map, the between-class distances for some categorie pairs are much longer than the within-class distances and therefore the within-class structures are concealed. However, the long distance between several categories doesn't mean a good separation for each category pairs, either. In fact, in Fig. 1, digit 2 overlaps digit 5 a little. But in Fig. 2(e), digit 1 and 3, digit 5 and 6 are strongly overlapped. This is because, in Supervised Isomap, the redefinition of the distance between two data points is too empirical, and may contradict the real distribution.

VI. Conclusions

In this paper, by utilizing the local information, we propose a new algorithm-SNDR to chart the sub-manifolds of different categories under one coordinate system. Experimental results on Minist have shown its superior performance over several state-of-art algorithms. In the future, we will consider how to use this supervised nonlinear map to help improve the accuracy of classification tasks and how to depict the low-dimensional coordinates of the out-of-sample data points.

References

[1] Richard O. Duda, Peter E. Hart, and David G. Stork: Pattern Analysis. John Wiley and Sons, Inc.(2001)

[2] G. Baudat and F. Anouar: Generalized discriminant analysis using a kernel approach. Neural Computation,12(10): 2385-2404,(2000)

[3] S. Mika, G.R$\ddot{a}$tsch, B. Sch$\ddot{o}$lkopf, A. Smola, J. Weston, and K. R. M$\ddot{u}$ller: Invariant feature extraction and classification in kernel spaces. NIPS, 12, (1999)

[4] Li-Fen Chen, Hong Yuan, Mark Liao etc: A new LDA-based face recognition system which can solve the small sample size problem. PR 33 (2000) 1713–1726

[5] Jin, Zhong; Yang, Jingyu; Hu, Zhongshan; Lou, Zhen: Face recognition based on the uncorrelated discriminant transformation. PR 34, No.7, 1405-1416 (2001).

[6] Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 15 (2003), 1373–1396.

[7] Xiaofei He; Partha Niyogi: Locality Preserving Projections.NIPS (2005)

[8] Cai, D., He, X., Zhou, K., Han, J., Bao, H.: Locality sensitive discriminant analysis. IJCAI (2007) 1713–1726

[9] Roweis, S. T., Lawrance, K. S.: Nonlinear dimensionality reduction by locally linear embedding. Science, 290 (2000), 2323-2326.

[10] Tenenbaum, J., de Silva, V., and Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science, 290(2000):2319-2323.

[11] Xin Geng, De-Chuan Zhan, and Zhi-Hua Zhou: Supervised Nonlinear Dimensionality Reduction for Visualization and Classification. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, (2005), 35(6): 1098-1107.

[12] Mingrui Wu, Bernhard Sch$\ddot{o}$lkopf: Transductive Classification via Local Learning Regularization. AIStatics(2007)

[13] Masashi Sugiyama: Local Fisher discriminant analysis for supervised dimensionality reduction. ICML(2006)

[14] RMatthew Brand: Charting a manifold. NIPS(2003)

[15] D de Ridder, O Kouropteva, etc: Supervised locally linear embedding.ICANN/ICONIP(2003)

[1] http://yann.lecun.com/exdb/mnist/

Input: data points $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$, as well as their labels, where n denotes the number of all the observed data points. The desired dimension of the reduced feature space is d ($d \leq 3$); k_b, k_w, and the trade-off parameter η.

1. Construct the within-class matrix $\mathbf{W}^{'}_{\mathbf{w}}$ and the between-class matrix $\mathbf{W}^{'}_{\mathbf{b}}$, using (4) and (5), respectively.

2. Calculate the Laplacian matrix $\mathbf{L}^{'}_{\mathbf{b}} = \mathbf{D}^{'}_{\mathbf{b}} - \mathbf{W}^{'}_{\mathbf{b}}$

3. Solving the optimization problem (6). The optimal embedding is given by Eq. (8)

Ouput: Eq. (8) gives the optimal embedding for the observed set.

TABLE I

THE PROCEDURE TO FIND THE OPTIMAL EMBEDDING FOR THE OBSERVED DATA POINTS

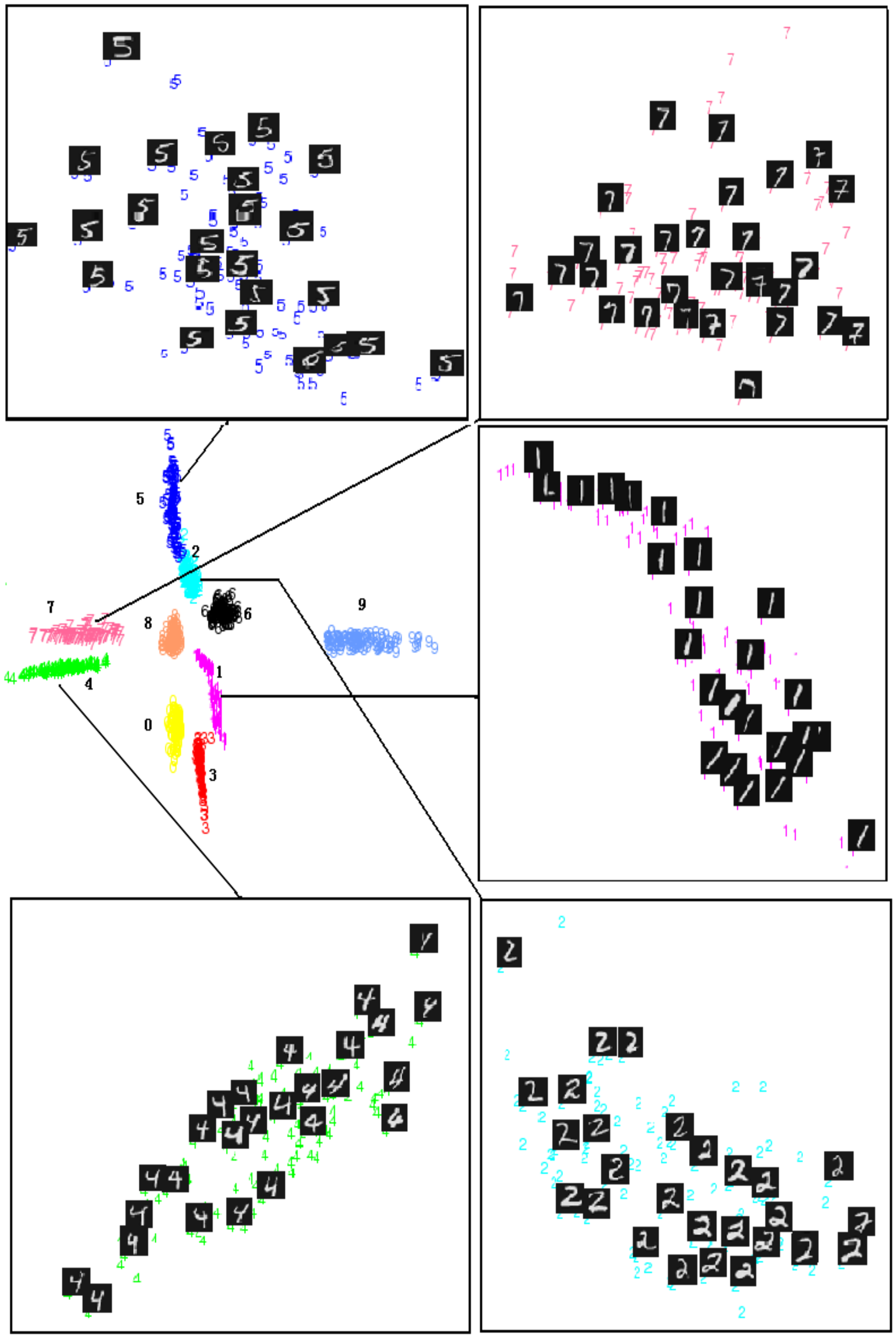
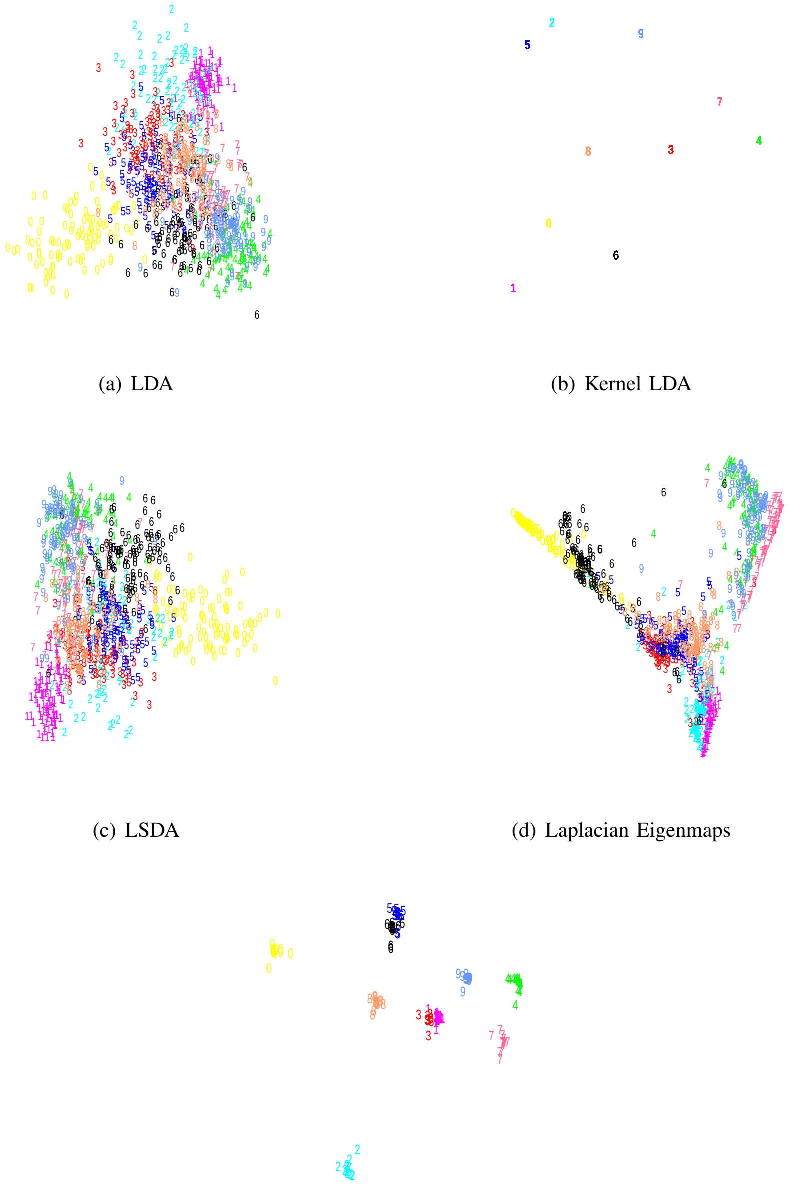


Fig. 1. The charting for the manifolds of all the categories. Each color represents a different category, and the sub-manifolds of several digits are magnified.



(a) LDA

(b) Kernel LDA

(c) LSDA

(d) Laplacian Eigenmaps

(e) Supervised Isomap

Fig. 2. The comparison charting result of LDA, Kernel LDA, LSDA, Laplacian Eigenmaps and Supervised Isomap

Face Recognition based on Whitening Transformation of Distribution of Subspaces

Tomokazu Kawahara, Masashi Nishiyama, Tatsuo Kozakaya, Osamu Yamaguchi
Corporate Research & Development Center Toshiba Corporation
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki-shi, Kanagawa 212-8582, Japan
{tomokazu.kawahara, masashi.nishiyama, tatsuo.kozakaya, osamu1.yamaguchi}@toshiba.co.jp

Abstract

We propose a new face recognition method that separates subspaces representing individuals based on the mathematical analysis of angles between multiple subspaces. A low-dimensional subspace representation by principal component analysis is known to be an effective approach for describing variation of facial patterns. A similarity between individuals is defined by an angle between their subspaces. Since all facial patterns have the same structure of facial parts, it is significant to extract individual characteristics from each subspace by considering the cross-relationship between categories. Our method applies "whitening transformation of distribution of subspaces", which can uniformize the distribution according to eigenvalues of the autocorrelation matrix of the subspaces. We derive the equation relating angles between subspaces to uniformity of the distribution of these subspaces. From this equation, the whitening transformation is effective for separation of the subspaces. Under the ideal condition, the whitening transformation orthogonalizes all subspaces. In other words, all similarities between each other are equal to 0. *We show the proposed method works well even in a practical case through evaluation experiments on the FRGC 1.0 and the FERET databases and outperforms other methods.*

1 Introduction

Many face identification methods have been proposed, which represent variation of patterns for an individual as a low-dimensional subspace generated from a set of patterns by principal component analysis (PCA) [2, 7, 11]. Since these methods are able to cope with variation in appearance, a robust face identification application can be built.

Yamaguchi et al. have proposed face recognition using the *Mutual Subspace Method* (MSM)[11]. They represent not only reference patterns as a reference subspace but also input patterns as an input subspace. To compare an input subspace with the reference subspace representing an individual, a similarity of MSM is defined by an angle between the input subspace and the reference subspace. MSM has a problem in that reference subspaces crowd since all facial patterns have the same structure of facial parts and MSM does not have a function that separates the subspaces of individuals.

To improve the recognition accuracy by separating subspaces, Fukui et al. have extended MSM to the *Constrained Mutual Subspace Method* (CMSM)[2]. In CMSM, reference subspaces are projected onto a constraint subspace, which is designed to emphasize the difference between individuals. Fukui et al. confirmed empirically that the projection to the constraint subspace creates a larger angle between multiple reference subspaces and explained that subspaces are separated because the common components of subspaces are removed.

We propose a new method, the *Whitened Mutual Subspace Method* (WMSM), based on a more mathematical analysis of angles between subspaces, which uses the whitening transformation of the distribution of subspaces for separation of subspaces. Whitening is a process to make a distribution uniform. First, we derive the equation that relates angles between multiple subspaces to a standard deviation of eigenvalues of an autocorrelation matrix of these subspaces. This equation describes that uniformizing a distribution of multiple subspaces makes angles between these subspaces larger. In other words, the whitening transformation emphasize the difference between individuals. In particular, the whitening transformation of a distribution of subspaces orthogonalizes reference subspaces when the number of reference subspaces is small. We show the proposed method works well even in a practical case through evaluation experiments on the FRGC 1.0 and the FERET databases and outperforms other methods.

The remainder of this paper is organized as follows. First, to explain the reason for using the whitening transformation of a distribution of subspaces mathematically, we analyze angles between multiple subspaces in section 2. Next, we describe the proposed method of face recognition in section 3. We demonstrate the effectiveness of our method by face recognition experiments in section 4.

2 Mathematical analysis of angles between subspaces

In this section, we explain the mathematical reason for using the whitening transformation of a distribution of subspaces for separation of these subspaces. For the purpose of the explanation, two mathematical objects that are calculated from multiple subspaces are prepared and the equation describing the relationship between these mathematical objects are derived. One of the mathematical objects is a measure of separability of multiple subspaces, that consists of canonical angles between these subspaces [1]. The other is an autocorrelation matrix of multiple subspaces[2]. We derived the equation that consists of a measure of separability of multiple subspaces and a standard deviation of eigenvalues of an autocorrelation matrix of these subspaces. This equation describes that a measure of separability of subspaces becomes large when a standard deviation of eigenvalues of this matrix becomes small. In other word, uniformizing distribution of subspaces separates these subspaces. Based on this mathematical analysis of angles between subspaces, we propose the method using whitening transformation for separation of multiple subspaces.

In MSM, a similarity between two subspaces is defined by an angle between these subspace. In this paper, therefore, we represent that subspaces separate when angles between these subspaces are large.

2.1 A measure of separability of subspaces

In this section, we define a measure of separability of two subspaces based on canonical angles between these subspaces and extend it to multiple subspaces.

To prepare the definition of a measure of separability of subspaces, we explain canonical angles between two subspaces, which are described in [1]. d canonical angles $\theta^{(1)}, \ldots \theta^{(d)}$ between the d-dimensional subspaces $\mathbf{V}_1$ and $\mathbf{V}_2$ in a vector space are defined as follows;

- $\mathbf{V}_1^{(1)} = \mathbf{V}_1$ and $\mathbf{V}_2^{(1)} = \mathbf{V}_2$.
- $\theta^{(i)}$ is the angle between $v_1^{(i)}$ and $v_2^{(i)}$, where $v_1^{(i)} \in \mathbf{V}_1^{(i)}$ and $v_2^{(i)} \in \mathbf{V}_2^{(i)}$ are the nearest vectors under the condition $|v_1^{(i)}| = |v_2^{(i)}| = 1$.
- $\mathbf{V}_1^{(i+1)} = \{v \in \mathbf{V}_1^{(i)} | v \perp v_1^{(i)}\}$ and $\mathbf{V}_2^{(i+1)} = \{v \in \mathbf{V}_2^{(i)} | v \perp v_2^{(i)}\}$.

where $i = 1, \ldots d$ and $|\cdot|$ denotes the norm. The subspaces $\mathbf{V}_j^{(1)}, \ldots, \mathbf{V}_j^{(d)}$ have the following relation;

$$\mathbf{V}_j^{(1)} \supset \mathbf{V}_j^{(2)} \supset \ldots \supset \mathbf{V}_j^{(d)} \tag{1}$$

where $j = 1, 2$. In particular, $\theta^{(1)}$ is equal to the angle between $\mathbf{V}_1$ and $\mathbf{V}_2$. Therefore, we use canonical angles instead of a single angle because more detailed analysis of separability is possible. When two subspaces are identical and orthogonal, all canonical angles are equal to 0 and $\pi/2$, respectively. From the definition of canonical angles, we obtain the inequation $\theta^{(1)} \leq \ldots \leq \theta^{(d)}$.

The canonical angles between these subspaces become large when two subspaces separate. Therefore, we define a measure of separability of two subspaces $\mathbf{V}_1$ and $\mathbf{V}_2$ as follows:

$$\mathrm{Sep}(\mathbf{V}_1, \mathbf{V}_2) = 1 - \frac{1}{d} \sum_{i=1}^{d} \cos^2 \theta^{(i)}, \tag{2}$$

where $\theta^{(1)}, \ldots \theta^{(d)}$ are canonical angles between $\mathbf{V}_1$ and $\mathbf{V}_2$. If two subspaces are identical and orthogonal, measures of separability of these subspaces are equal to 0 and 1, respectively. When this measure of two subspaces is large, these two subspaces separate.

For calculation of the measure of two subspaces (2) using orthonormal bases of these subspaces, we derive the equation between a measure of separability of two subspaces and projection matrices of these subspaces. The projection matrix $\mathbf{P}$ of subspace $\mathbf{V}$ is defined by equation (3) [8].

$$\mathbf{P} = \sum_{i=1}^{d} \psi_i \psi_i^T, \tag{3}$$

where $\{\psi_1, \ldots \psi_d\}$ is an orthonormal basis of $\mathbf{V}$. Generally, a projection matrix is defined by $d \times D$ matrix $(\psi_1, \ldots \psi_d)^T$ where D is dimension of the vector space. However, we use the former definition since the latter definition does not have the information of position of the subspace on the vector subspace. Let $\mathbf{P}_j$ be the projection matrix of $\mathbf{V}_j$, where $j = 1, 2$. By calculation of the trace of $\mathbf{P}_1\mathbf{P}_2$, the equation (4) is obtained.

$$\mathrm{Sep}(\mathbf{V}_1, \mathbf{V}_2) = 1 - \frac{1}{d} \mathrm{tr}(\mathbf{P}_1 \mathbf{P}_2), \tag{4}$$

where $\mathrm{tr}(\cdot)$ is a trace of a matrix, which is a sum of diagonal components of the matrix. (See Appendix for a detailed calculation of (4)).

We extend a measure of two subspaces (2) to a measure of separability of multiple subspaces. Let $\mathbf{V}_1, \ldots, \mathbf{V}_N$ be d-dimensional subspaces in a D-dimensional vector space. A measure of separability of subspaces $\mathbf{V}_1, \ldots, \mathbf{V}_N$ is defined as an average of measures of $\mathbf{V}_k$ and $\mathbf{V}_l$ $(1 \leq k <$

$l \leq N$)

$$\text{Sep}(\mathbf{V}_1, \ldots, \mathbf{V}_N) = \frac{2}{N(N-1)} \sum_{1 \leq k < l \leq N} \text{Sep}(\mathbf{V}_k, \mathbf{V}_l). \tag{5}$$

When all subspaces are identical and orthogonal, measures of separability of these subspaces are equal to 0 and 1, respectively. The more this measure of multiple subspaces is, the more these subspaces separate. We obtain the equation (6) from (5) and (4).

$$\text{Sep}(\mathbf{V}_1, \ldots, \mathbf{V}_N) = 1 - \frac{2}{N(N-1)} \sum_{1 \leq k < l \leq N} \frac{1}{d} \text{tr}(\mathbf{P}_k \mathbf{P}_l), \tag{6}$$

where $\mathbf{P_k}$ is the projection matrix of $\mathbf{V}_k$ defined by (3). Therefore, we calculate a measure of separability of multiple subspaces using orthonormal bases of these subspaces.

2.2 An autocorrelation matrix of subspaces

To prepare calculation of a measure of multiple subspaces (5) we explain an autocorrelation matrix of distribution of subspaces, which is described in [2], and calculate an average and a standard deviation of its eigenvalues. An autocorrelation matrix of distribution of subspaces $\mathbf{A}$ is defined as an average of all projection matrices, like an autocorrelation matrix of distribution of vectors [8], and its eigenvalue problem is solved as follows,

$$\mathbf{A} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{P}_k = \mathbf{B} \mathbf{\Lambda} \mathbf{B}^T, \tag{7}$$

where $\mathbf{B}$ is the matrix whose columns are the orthonormal eigenvectors of $\mathbf{A}$ and $\mathbf{\Lambda}$ is the diagonal matrix of the corresponding eigenvalues $\lambda_1 \geq \ldots \geq \lambda_D$.

We calculate an average and a standard deviation of eigenvalues of an autocorrelation matrix. Let m_λ and σ_λ be an average and a standard deviation of eigenvalues $\lambda_1, \ldots, \lambda_D$, respectively. In the first step, we calculate an average of eigenvalues of an autocorrelation matrix. An average of eigenvalues of the autocorrelation matrix m_λ is equal to the constant value d/D regardless of arrangement of subspaces $\mathbf{V}_1, \ldots, \mathbf{V}_N$ from the following calculation,

$$\begin{aligned} m_\lambda &= \frac{1}{D} \sum_{l=1}^{D} \lambda_l = \frac{1}{D} \text{tr}\mathbf{A} = \frac{1}{D} \text{tr}(\frac{1}{N} \sum_{k=1}^{N} \mathbf{P}_k), \\ &= \frac{1}{DN} \sum_{k=1}^{N} \text{tr}(\mathbf{P}_k) = \frac{1}{DN} \sum_{k=1}^{N} d = \frac{d}{D}. \end{aligned} \tag{8}$$

Next, we calculate a standard deviation of eigenvalues of an autocorrelation matrix using (8) as follows,

$$\begin{aligned} \sigma_\lambda &= \frac{1}{D} \sum_{l=1}^{D} (\lambda_l - m_\lambda)^2, \\ &= \frac{1}{D} \sum_{l=1}^{D} \lambda_l^2 - m_\lambda^2 = \frac{1}{D} \text{tr}\mathbf{A}^2 - (\frac{d}{D})^2. \end{aligned} \tag{9}$$

2.3 Equation between a measure of separability and an autocorrelation matrix

In this section, we show that a separability of multiple subspaces is decided from only a standard deviation of eigenvalues of an autocorrelation matrix from the equation that consists of a measure of separability of multiple subspaces, a standard deviation of eigenvalues of an autocorrelation matrix and a constant term.

Using (6), (8) and (9), a measure of separability of multiple subspaces $\mathbf{S} = \text{Sep}\ (\mathbf{V}_1, \ldots \mathbf{V}_N)$ is calculated as follows,

$$\begin{aligned} \mathbf{S} &= 1 - \frac{2}{N(N-1)} \sum_{1 \leq k < l \leq N} \frac{1}{d} \text{tr}(\mathbf{P}_k \mathbf{P}_l), \\ &= 1 - \frac{1}{dN(N-1)} \sum_{1 \leq k \neq l \leq N} \text{tr}(\mathbf{P}_k \mathbf{P}_l), \\ &= 1 - \frac{1}{dN(N-1)} \text{tr}(\sum_{k,l=1}^{N} \mathbf{P}_k \mathbf{P}_l - \sum_{k=1}^{N} \mathbf{P}_k), \\ &= 1 - \frac{1}{dN(N-1)} \text{tr}(N^2 \mathbf{A}^2 - N\mathbf{A}), \\ &= -\frac{DN}{d(N-1)} \sigma_\lambda^2 + \frac{N(D-d)}{(N-1)D}. \end{aligned} \tag{10}$$

From the equation (10), a transformation that decreases the standard deviation of eigenvalues σ_λ separates the subspaces $\mathbf{V}_1, \ldots \mathbf{V}_N$. In particular, all subspaces are separated most when all eigenvalues $\lambda_1, \ldots \lambda_D$ are the same values.

2.4 Whitening transformation of distribution of subspaces

We propose whitening transformation of distribution of subspaces for separation of multiple subspaces based on the analysis in the previous section. From the analysis in section 2.3, a transformation that decreases standard deviation of eigenvalues of autocorrelation matrix of subspaces separates these subspaces. In other words, whitening transformation of distribution of subspaces is effective to separate these subspaces (Fig. 1). “Whitening” is a process to make all eigenvalues of an autocorrelation matrix the same. The

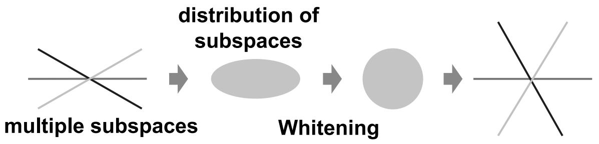


Figure 1. The ellipse and the circle in the center of the figure represent the distribution of subspaces. "Whitening" makes the distribution uniform.

matrix $\mathbf{W}$ that represents whitening transformation of distribution of subspaces $\mathbf{V}_1, \ldots, \mathbf{V}_N$ is defined and makes an autocorrelation matrix the identity matrix $\mathbf{I}$ as follows:

$$\mathbf{W} = \mathbf{\Lambda}^{-1/2}\mathbf{B}^T, \quad (11)$$

$$\mathbf{W}\mathbf{A}\mathbf{W}^T = (\mathbf{\Lambda}^{-1/2}\mathbf{B}^T)\mathbf{B}\mathbf{\Lambda}\mathbf{B}^T(\mathbf{B}\mathbf{\Lambda}^{-1/2}) = \mathbf{I},(12)$$

where $\mathbf{B}$ and $\mathbf{\Lambda}$ are defined in (7).

Another method, named the *Orthogonal Subspace Method* (OSM), in which whitening orthogonalizes subspaces, has been proposed by Fukunaga et al.[3] and Kittler[4]. In OSM, an autocorrelation matrix of each class is transformed by the whitening of the autocorrelation matrix generated from all samples in all classes before a subspace of each class is generated from the eigenvectors of the autocorrelation matrix of this class, the eigenvalues of which are large. In other words, a set of samples in each class is represented as a low-dimensional subspace after the distribution of all samples in all classes is made uniform. In this method, the eigenvector of an autocorrelation matrix of a class, the eigenvalue of which is 1, is orthogonal to all samples in other classes since all eigenvalues of the autocorrelation matrix generated from all samples in all classes are equal to 1.

In our method and OSM, subspaces are orthogonalized using whitening. The difference between our method and OSM is the order of the linearization and the transformation. In other words, an input subspace and a reference subspace are generated from a set of patterns before whitening in our method, but after whitening in OSM. Therefore, our method does not use eigenvectors of the autocorrelation matrix whose eigenvalues are small. Furthermore, when the number of subspaces is small, these subspaces are always orthogonalized in our method but not always orthogonalized in OSM (section 2.5).

2.5 Transformation under the ideal condition

We show that subspaces can be orthogonalized by the whitening transformation of distribution of these subspaces in the ideal case that the number of these subspaces is small.

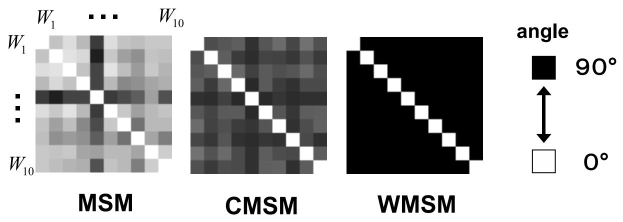


Figure 2. Similarity matrix: angles between ten reference subspaces in MSM, CMSM and WMSM in the case that the condition (14) is satisfied. The darker a pixel is, the larger the angle between subspaces is.

In the first step, we prove the following proposition. Let $u_1, \ldots, u_N$ be bases of 1-dimensional subspaces in D-dimensional vector space. A matrix $\mathbf{U}$ denotes $(u_1, \ldots, u_N)$ and $\mathbf{\Lambda}, \mathbf{\Lambda}^{-1/2}, \mathbf{B}, \mathbf{W}$ are defined as in (7) and (11). Let u'_i be $\mathbf{W}u_i$ for all i.

Proposition 1 *if* $u_1, \ldots, u_N$ *is linearly independent,* $u'_1, \ldots, u'_N$ *is orthonormal.*

Proof *Let* $\mathbf{U}'$ *be* $\mathbf{WU}$. *Since* $\mathbf{U}\mathbf{U}^T = \mathbf{A} = \mathbf{B}\Lambda\mathbf{B}^T$ *in the equation (7) and* $u_1, \ldots, u_N$ *is linearly independent,*

$$\mathbf{U}'\mathbf{U}'^T = \mathbf{W}\mathbf{U}\mathbf{U}^T\mathbf{W}^T = \mathbf{\Lambda}^{-1/2}\mathbf{B}^T\mathbf{B}\mathbf{\Lambda}\mathbf{B}^T\mathbf{B}\mathbf{\Lambda}^{-1/2} = \tilde{\mathbf{I}}_N, \quad (13)$$

where $\tilde{\mathbf{I}}_N$ *is a diagonal matrix in which the number of* 1 *on the diagonal is* N *and others are* 0. *The symmetric matrix* $\mathbf{U}'^T\mathbf{U}'$ *is an identity matrix because all eigenvalues of* $\mathbf{U}'^T\mathbf{U}'$ *are the same as those of* $\mathbf{U}'\mathbf{U}'^T$ *without* 0 *and the rank of* $\mathbf{U}'^T\mathbf{U}'$ *is* N. *Therefore,* $u'_1, \ldots, u'_N$ *is orthonormal because a component of* $\mathbf{U}'^T\mathbf{U}'$ *is an inner product of* u'_k *and* u'_l. □

Generally, we can orthogonalize all subspaces using the whitening transformation of the distribution of these subspaces if the following inequation is satisfied;

$$dN \leq D, \quad (14)$$

where D is the dimension of the vector space including these subspaces, because we apply Proposition 1 to bases of these subspaces. The equation (14) requires the dimension or the number of these subspaces to be small. In the case that the condition (14) is satisfied, the measure of separability of subspaces transformed by the whitening transformation is equal to 1 since the autocorrelation matrix of these subspaces is $\tilde{\mathbf{I}}_{dN}$ from the same calculation (13).

Fig. 2 shows similarity matrix images whose pixel values represent an angle between pairs of subspaces in MSM,

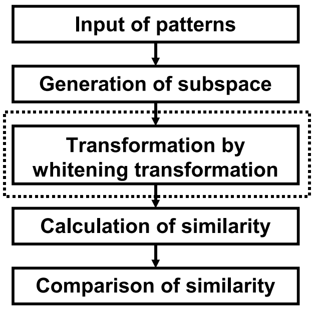

Input of patterns
Generation of subspace
Transformation by whitening transformation
Calculation of similarity
Comparison of similarity

Figure 3. The flow chart of WMSM.

CMSM, and our method in the case that the condition (14) is satisfied. This figure shows that our method orthogonalizes all these subspaces.

3 Face Recognition using the whitening transformation of the distribution of subspaces

In this section, we describe the procedure of WMSM (Fig. 3).

3.1 Algorithm for face recognition

First, we located the face pattern from the positions of the feature points and cropped to 32×32 pixels using 3D normalization[5] and preprocessing[6]. In order to adapt localization error of feature points, we represent variation of face patterns due to the localization error as a subspace in the feature space by perturbation of the feature points and obtaining multiple face patterns from a single face image. We apply PCA to the vectors to generate an input subspace. Let $\{x_i\}_{i=1,\ldots n}$ be a set of vectors. The basis of the input subspace is the eigenvectors of the autocorrelation matrix $\mathbf{Z} = 1/n \sum_{i=1}^{n} x_i x_i^T$ [8].

The whitening transformation (11) is generated from an autocorrelation matrix of reference subspaces. To allow for the variation in appearance for each individual, it is effective to increase the dimension of the reference subspace by addition of other bases that are generated from reference patterns and not used for comparison with an input subspace.

To compare the input subspace with the reference subspace registered in a database for each individual, we calculate their similarities after transforming the input subspace and the reference subspaces by the whitening transformation of a distribution of reference subspaces. The person in the image is identified as the person who corresponds to the reference subspace with the highest similarity.

3.2 Transformation of a subspace and calculation of a similarity

In our proposed method, to transform the input subspace $\mathbf{V}_{\text{input}}$ and the reference subspace $\mathbf{V}_{\text{ref}}$ by whitening of distribution of reference subspaces, we carry out the following steps:

1. Transform a basis of a subspace by the whitening transformation $\mathbf{W}$.

2. Apply Gram-Schmidt orthogonalization to them.

The orthonormal basis is a basis of the transformed subspace.

We define a similarity s between the d-dimensional subspaces $\mathbf{V}_{\text{input}}$ and $\mathbf{V}_{\text{ref}}$ as $s = \cos^2 \theta$, where θ is the angle between $\mathbf{V}_{\text{input}}$ and $\mathbf{V}_{\text{ref}}$. The angle θ is equal to the 1-th canonical angle $\theta^{(1)}$ between $\mathbf{V}_{\text{input}}$ and $\mathbf{V}_{\text{ref}}$. If $\mathbf{V}_{\text{input}}$ and $\mathbf{V}_{\text{ref}}$ are identical, the angle θ is equal to 0. The angle is calculated using the MSM[11]. The similarity s equals the largest eigenvalue λ_{max} of $\mathbf{X} = (x_{mn})$ using

$$x_{mn} = \sum_{l=1}^{d} (\psi_m, \phi_l)(\phi_l, \psi_n) \ \ (m, n = 1 \ldots d) \ , \qquad (15)$$

where $\{\psi_i\}_{i=1,\ldots,d}$ and $\{\phi_j\}_{j=1,\ldots,d}$ are the orthonormal bases of $\mathbf{V}_{\text{input}}$ and $\mathbf{V}_{\text{ref}}$, respectively; (ψ_m, ϕ_l) is the inner product of ψ_m and ϕ_l.

4 Evaluation with the FRGC 1.0 and FERET databases

We show the proposed method works well even in a practical case. We performed experiments using the controlled still images (exp1) in the FRGC 1.0 database [9] and the *fa* and the *fb* data sets in the FERET database[10]. The controlled still images in FRGC 1.0 consisted of 152 gallery images and 608 probe images. The *fa* and the *fb* in FERET consisted of images of 1196 people with one image per person and 1195 people with one image per person, respectively.

We compare five methods, namely, MSM, CMSM, *Multiple* CMSM (MCMSM) [7], WMSM and *Multiple* WMSM (MWMSM). MCMSM and MWMSM apply ensemble learning with bagging to CMSM and WMSM, respectively. In MCMSM, multiple constraint subspaces are generated from reference subspaces selected randomly in the same way of bagging. The input subspace and the reference subspaces are projected onto each constraint subspace and a similarity is determined with the similarities

Table 1. The methods and their parameters. d is the dimension of input and reference subspaces. L is the number of constraint subspaces and whitening transformations. d' is the dimension of reference subspaces that generate constraint subspaces and whitening transformations. C is the dimension of constraint subspaces.

	d	L	d'	C
MSM	7	–	–	–
CMSM	7	1	15	210
MCMSM	7	10	15	210
WMSM	7	1	15	–
MWMSM	7	10	15	–

Table 2. Experimental results using FRGC 1.0 in terms of Correct Match Rate (CMR) and Equal Error Rate (EER).

	CMR (%)	EER (%)
MSM	96.4	3.45
CMSM	96.5	2.47
MCMSM	97.2	2.28
WMSM	97.0	1.81
MWMSM	97.2	1.81

calculated on each constraint subspace. In MWMSM, multiple whitening transformations are generated from reference subspaces selected randomly and the similarity is determined with an average of the similarities calculated after transformation by each whitening transformation. Their parameters in the experiments are listed in Table 1.

Table 2 shows the evaluation results in FRGC 1.0 for each method in terms of Correct Match Rate (CMR) and Equal Error Rate (EER). Correct Match Rate is the probability that an input of the right person is correctly accepted. Equal Error Rate is the probability that false acceptance rate (FAR) equals the false rejection rate (FRR). It can be seen that the proposed method and the proposed method with ensemble learning are equivalent to MCMSM with regard to Correct Match Rate and superior to the other methods with regard to Equal Error Rate on FRGC 1.0.

To evaluate the generalization ability of our method, we performed experiments using another database. Fig. 4 shows the evaluation results for each method and the best result (UMD97) of the partially automatic algorithms reported in FERET'97 [10] in terms of Cumulative Match Rate. It can be seen that the proposed method and the proposed method with ensemble learning are superior to the other methods.

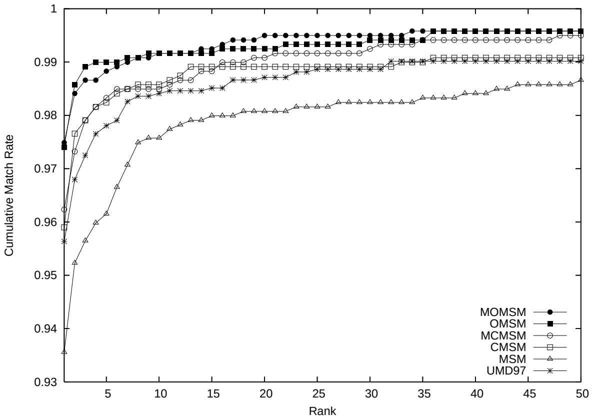


Figure 4. Experimental results using FERET database in terms of Cumulative Match Rate.

5 Conclusions

This paper presented a face recognition method based on mathematical analysis of angles between subspaces in which we apply whitening of a distribution of subspaces to emphasize the difference between individuals. We derived the equation (10) that relates angles between subspaces to a distribution of these subspace. This equation describes that the whitening transformation is effective for separation of these subspaces. In the experiment, we obtained high performance compared with other methods on the FRGC 1.0 and the FERET database .

References

[1] F. Chaitin-Chatelin. *Eigenvalues of Matrices*. Johe Wiley & Sons Ltd, 1993. (Enlarged Translation of the French Publication with Masson).

[2] K. Fukui and O. Yamaguchi. Face recognition using multiviewpoint patterns for robot vision. In *11th International Symposium of Robotics Research*, pages 192–201, 2003.

[3] K. Fukunaga and W. Koontz. Application of the Karhunen-Loeve expansion to feature extraction and ordering. *IEEE Transactions on Computers*, 19(5):311–318, 1970.

[4] J. Kittler. The subspace approach to pattern recognition. In R. Trappl, G. J. Klir, and L. Ricciardi, editors, *Progress in cybernetics and systems research*, Washington, 1978. Hemisphere Publishing Corporation.

[5] T. Kozakaya and O. Yamaguchi. Face recognition by projection-based 3d normalization and shading subspace orthogonalization. In *Proceedings IEEE 7th International Conference on Automatic Face and Gesture Recognition*, pages 163–168, 2006.

[6] M. Nishiyama and O. Yamaguchi. Face recognition using the classified appearance-based quotient image. In *Proceed-*

ings IEEE 7th International Conference on Automatic Face and Gesture Recognition, pages 49–54, 2006.

[7] M. Nishiyama, O. Yamaguchi, and K. Fukui. Face recognition with the multiple constrained mutual subspace method. In *Audio- and Video-based Biometric Person Authentication*, pages 71–80, 2005.

[8] E. Oja. *Subspace Methods of Pattern Recognition*. Research Studies Press, 1983.

[9] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[10] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[11] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *Proceedings of IEEE 3rd International Conference on Automatic Face and Gesture Recognition*, pages 318–323, 1998.

A Calculation of a measure of separability of subspaces

To obtain equation (4), we prove the following equation,

$$\frac{1}{d}\mathrm{tr}(\mathbf{P}_1\mathbf{P}_2) = \mathrm{Sep}(\mathbf{V}_1, \mathbf{V}_2). \tag{16}$$

To calculate the trace of the product of projection matrices, we describe several facts about vectors $v_1^{(1)}, \dots, v_1^{(d)}$ and $v_2^{(1)}, \dots, v_2^{(d)}$ in section 2.1. A set of vectors $\{v_j^{(1)}, \dots, v_j^{(d)}\}$ is an orthonormal basis of $\mathbf{V}_j$ since $v_j^{(i)}$ is orthogonal to $\mathbf{V}_j^{(i+1)}$, where $j = 1, 2$. Furthermore, $v_1^{(k)}$ is orthogonal to $v_2^{(l)}$ $(k \neq l)$ since the equations (17) and (18) are derived from the definition of $v_1^{(i)}$ and $v_2^{(i)}$,

$$\begin{aligned} \mathbf{P}_2^{(i)} v_1^{(i)} &= \cos\theta^{(i)} v_2^{(i)}, &(17)\\ \mathbf{P}_1^{(i)} v_2^{(i)} &= \cos\theta^{(i)} v_1^{(i)}, &(18)\end{aligned}$$

where $\mathbf{P}_j^{(i)}$ is projection matrix of $\mathbf{V}_j^{(i)}$ $(j = 1, 2)$. From these facts, the following equations (19) and (20) is acquired.

$$(v_1^{(k)}, v_1^{(l)}) = (v_2^{(k)}, v_2^{(l)}) = \begin{cases} 1 & (k = l), \\ 0 & (k \neq l), \end{cases} \tag{19}$$

$$(v_1^{(k)}, v_2^{(l)}) = \begin{cases} \cos\theta^{(k)} & (k = l), \\ 0 & (k \neq l), \end{cases} \tag{20}$$

where $(\cdot, \cdot)$ is the inner product of vectors.

We calculate the trace of the product of projection matrices in the equation (16). The projection matrix $\mathbf{P}_j$ of $\mathbf{V}_j$ is defined by $\sum_{i=1}^{d} v_j^{(i)} v_j^{(i)T}$ since $\{v_j^{(1)}, \dots v_j^{(d)}\}$ is an orthonormal basis, where $j = 1, 2$. By calculation of trace of $\mathbf{P}_1\mathbf{P}_2$ using the equations (19) and (20) as follows, the equation (16) is obtained.

$$\begin{aligned} \frac{1}{d}\mathrm{tr}(\mathbf{P}_1\mathbf{P}_2) &= \frac{1}{d}\mathrm{tr}((\sum_{k=1}^{d} v_1^{(k)} v_1^{(k)T})(\sum_{l=1}^{d} v_2^{(l)} v_2^{(l)T})), \\ &= \frac{1}{d}\mathrm{tr}(\sum_{k,l=1}^{d} v_2^{(l)T} v_1^{(k)} v_1^{(k)T} v_2^{(l)}), \\ &= \frac{1}{d}\sum_{k=1}^{d} \cos^2\theta^{(k)} = \mathrm{Sep}(\mathbf{V}_1, \mathbf{V}_2). \end{aligned}$$

Personal Favorite Scene Selection from Broadcast Soccer Video using Eigenspace Method

Masao Izumi
Graduate School of Engineering, Osaka Prefecture University
1–1, Gakuen-cho, Naka-ku, Sakai, Osaka 599–8531 JAPAN
izumi@cs.osakafu-u.ac.jp

Abstract

In this paper, we propose a novel method for selecting personal favorite scenes using eigenspace method based approach from broadcast soccer videos. In our method, we use physical parameters extracted from videos, such as image space frequency and time space frequency. We use these physical parameters as feature vectors, make an eigenspace of these vectors extracted from image slices of broadcasted soccer videos, and select the image slices that the distance from prior selected image slices as a personal favorite scene in the eigenspace is small. Personal favorite scenes are characterized as the feature vectors which are selected from physical parameters extracted from videos. The experimental results show the ability of our proposed method.

1. Introduction

In recent years, the spread of cable TV, DVD recorders, etc, enabled individuals to record a lot of TV programs easily. But, it needs immense time and efforts to search scenes wanting to watch in large amount of videos. Then, the technology which gives an effective index automatically will be more indispensable from now on.

However, it is difficult to create a general technique of indexing for all kinds of video. So there are many challenging researches about scene estimation, event estimation, and indexing. Leonardi at al. [1] estimate major soccer scenes using cameraworks peculiar to soccer videos, for example it pans and zooms rapidly on shoot scenes and corner kick scenes. Xinghua *et al.* [2] estimate goal events using textures and score boards peculiar to soccer videos. Moreover, it takes into consideration that a video is generally multiple streams of media information, such as audios, texts, and images, it will be thought that the performance of indexing can be raised more by unifying multiple media information. Uegaki *et al.* [3] proposed multimodal indexing from broadcast soccer video using Dynamic Bayesian networks which input cameraworks, players and ball trajectories, and audio power spectrum.

These advanced researches show good performance on automatic indexing of broadcast soccer video, but these indexes are quite general explanations of scene features, such as shoot scene, free-kick scene, throw-in scene, etc. On the other hand, when these methods are implemented on the personal used equipments, such as DVD recorders, users' demands are more personal. For example, one person likes aggressive passing scenes of offensive side, another likes placement kicks, etc. But sometimes the general explanation of index extracted from above methods isn't appropriate to these personal favorite demands, because the personal demands are wide-ranged various.

From these considerations, we try to establish the method of personal favorite scene selection from broadcast soccer video in this paper. We think personal demands are so various that physical parameters of each frame in video sequence, such as image space frequency and time space frequency, tend to be unique, then we propose the method of selecting personal favorite scenes using image space frequency and time space frequency. The second section explains our approach, and the third section discusses feature vectors and these eigenspace, the forth section performs our proposed method of personal favorite scene selection, and the fifth section gives the conclusions.

2. Our approach overview

In the case of sports video retrieval, it is an important issue how to represent the trend of personal favorite scenes. For example, some persons are prefer shooting scenes and placement kick scenes in broadcast soccer video, the others like skillful passing scenes or powerful defense scenes. Recent researches of automatic retrieval for sports videos are mainly focused on the search of scenes which are generally demanded by many people. But there are few researches which treat the way of searching personal favorite scenes. In this paper, we introduce the method of automatic retrieval for searching personal favorite scenes from broadcast soccer video using principal component analysis (PCA). Firstly, we extract physical parameters such as image space frequency and time space frequency as a feature vector from each frame of video sequence. Secondly, we construct the eigenspace of these feature vectors as the feature subspace. Then we calculate the distance from points that are projected by the feature vectors of previously selected as the personal favorite scenes. After extracting frames which distance is below the threshold as the personal favorite candidate frames, next we extract shots which include certain percentage of frames extracted as the personal favorite candidate frames. Here, a 'shot' means time-sequential frames which are shot by the same camera. Finally we select these shots as the results for searching personal favorite scenes.

3. Feature vector

Feature vectors should include not only the feature of each frame, such as color histograms and image space frequency which represent the feature of instant at the time of each frame. But also the feature vectors should include the feature of time sequential change between the time of each frame and the time of several frames later. In this research, we use space frequency of pixel intensities for the representation of the features at the time of each frame, and time space frequency of pixel intensities on x-t and y-t planes of time sequential images for the representation of the features between the time of each frame and several frames later. Feature vector $\boldsymbol{v}(t)$ is defined as follows,

$$\boldsymbol{v}(t) = (\boldsymbol{v}_{xy}^T, \boldsymbol{v}_{xt}^T, \boldsymbol{v}_{yt}^T)^T \tag{1}$$

$$\begin{aligned}\boldsymbol{v}_{xy} = (&\boldsymbol{v}_{00}^T, \boldsymbol{v}_{01}^T, \boldsymbol{v}_{02}^T, \boldsymbol{v}_{03}^T, \boldsymbol{v}_{10}^T, \boldsymbol{v}_{11}^T, \boldsymbol{v}_{12}^T, \boldsymbol{v}_{13}^T, \boldsymbol{v}_{20}^T, \boldsymbol{v}_{21}^T,\\ &\boldsymbol{v}_{22}^T, \boldsymbol{v}_{23}^T, \boldsymbol{v}_{30}^T, \boldsymbol{v}_{31}^T, \boldsymbol{v}_{32}^T, \boldsymbol{v}_{33}^T, \boldsymbol{v}_{40}^T, \boldsymbol{v}_{41}^T, \boldsymbol{v}_{42}^T, \boldsymbol{v}_{43}^T)^T\end{aligned} \tag{2}$$

$$\begin{aligned}\boldsymbol{v}_{xt} = (&\boldsymbol{v}_{00xt}^T, \boldsymbol{v}_{01xt}^T, \boldsymbol{v}_{02xt}^T, \boldsymbol{v}_{03xt}^T, \boldsymbol{v}_{04xt}^T, \boldsymbol{v}_{10xt}^T, \boldsymbol{v}_{11xt}^T, \boldsymbol{v}_{12xt}^T,\\ &\boldsymbol{v}_{13xt}^T, \boldsymbol{v}_{14xt}^T, \boldsymbol{v}_{20xt}^T, \boldsymbol{v}_{21xt}^T, \boldsymbol{v}_{22xt}^T, \boldsymbol{v}_{23xt}^T, \boldsymbol{v}_{24xt}^T)^T\end{aligned} \tag{3}$$

$$\begin{aligned}\boldsymbol{v}_{yt} = (&\boldsymbol{v}_{00yt}^T, \boldsymbol{v}_{01yt}^T, \boldsymbol{v}_{02yt}^T, \boldsymbol{v}_{03yt}^T, \boldsymbol{v}_{10yt}^T, \boldsymbol{v}_{11yt}^T, \boldsymbol{v}_{12yt}^T, \boldsymbol{v}_{13yt}^T,\\ &\boldsymbol{v}_{20yt}^T, \boldsymbol{v}_{21yt}^T, \boldsymbol{v}_{22yt}^T, \boldsymbol{v}_{23yt}^T)^T\end{aligned} \tag{4}$$

T means *transposition*, and x-y space frequency $\boldsymbol{v}_{ij}$ is 2-dimensional DCT values of 64 × 64 pixel rectangle which is positioned at $i \times 64$ of x-coordinate of each frame and $j \times 64$ of y-coordinate of each frame (which means the left-top coordinates of the rectangle is $(i \times 64, j \times 64)$.) The number of 2-dimensional DCT values of 64×64 pixel rectangle are $64 \times 64 = 256$. 256 is quite large, so we use three means of low-frequency, middle-frequency, and high-frequency. The mean value of low-frequency is the mean of the values at $(0, 1)$, $(1, 0)$, and $(1, 1)$. The mean value of middle-frequency is the mean of the values at $(32, i)$ and $(j, 32)$ $(i = 0, 1, 2, ..., 63; j = 0, 1, 2, ..., 63.)$ The mean value of high-frequency is the mean of the values at $(63, i)$ and $(j, 63)$ $(i = 0, 1, 2, ..., 63; j = 0, 1, 2, ..., 63.)$ The x-y space frequency is calculated from each 64×64 pixels region of 5×4 regions pictured on Fig.1. The x-t space frequency $\boldsymbol{v}_{ijxt}$ is three means of low-frequency, middle-frequency, and high-frequency as the same manner mentioned above of 2-dimensional DCT values of 64 × 64 rectangle at the position of $((i+1) \times 80, j \times 64) - ((i+1) \times 80, (j+1) \times 64 - 1)$. The x-t space frequency is calculated from three x-t planes pictured on Fig.2.

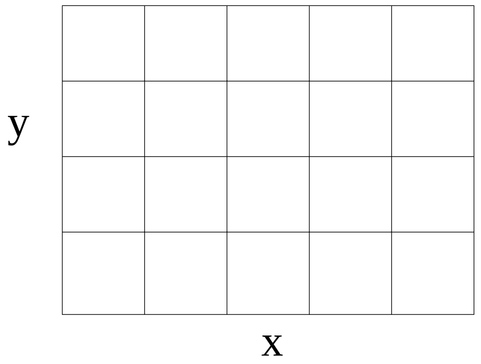


Figure 1. x-y planes.

In this figure, x_0, x_1, x_2 are the positions on the x coordinates, and we use three combinations of x_0, x_1, x_2. One is $x_0 = \frac{1}{4} \times x_{width}$, $x_1 = \frac{1}{2} \times x_{width}$, $x_2 = \frac{3}{4} \times x_{width}$, where x_{width} is the horizontal length of the image plane. Second is $x_0 = \frac{1}{6} \times x_{width}$, $x_1 = \frac{5}{12} \times x_{width}$, $x_2 = \frac{2}{3} \times x_{width}$. Third is $x_0 = \frac{1}{3} \times x_{width}$, $x_1 = \frac{7}{12} \times x_{width}$, $x_2 = \frac{5}{6} \times x_{width}$. These combinations are selected, because the positions of planes of which time space frequency is calculated are the key values to examine the similarities of time sequential frames. For example, time frequency of the first combination of three planes from the typical shooting scene is very

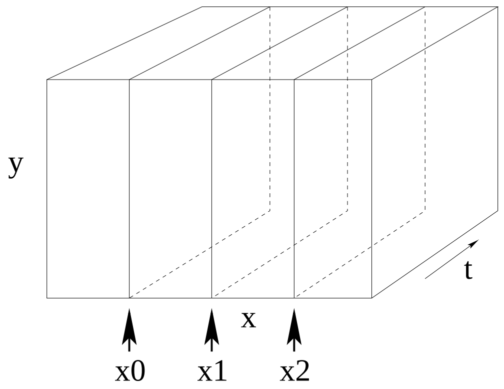


Figure 2. x-t planes.

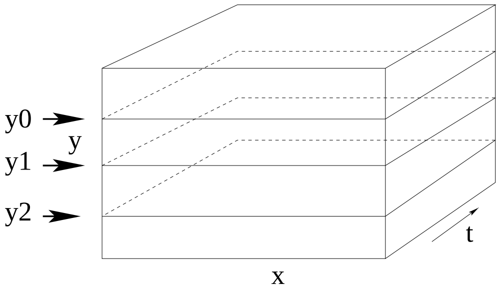


Figure 3. y-t planes.

similar to time frequency of the second combination of three planes from the other similar scene. So we use all combination of these positions to examine the similarity of scenes. In Eq.3, $\boldsymbol{v}_{nmxt}$, $(n = 0, 1, 2)$, $(m = 0, 1, 2, 3, 4)$ means m-th DCT values of x-t plane x_n.

The y-t space frequency $\boldsymbol{v}_{ijyt}$ is three means of low-frequency, middle-frequency, and high-frequency of 2-dimensional DCT values of 64 × 64 rectangle at the position of $(j \times 64, (i + 1) \times 60) - ((j + 1) \times 64 - 1, (i + 1) \times 60)$. Also the y-t space frequency is calculated from three combinations of y-t planes in the same manner of x-t planes. The y-t space frequency is calculated from three y-t planes pictured on Fig.3.

We use time-sequential frames of 320 × 240 pixels captured from broadcast soccer videos. After all, the number of elements of each feature vector is 141, and we extract nine set (three combinations of x-t planes × three combinations of y-t planes) of feature vectors from each frame.

4. Eigenspace of feature vectors

For selecting personal favorite scenes, we apply the eigenspace of collected feature vectors from learning samples of frames captured from broadcast soccer video. All feature vectors are regularized before following calculation. $\boldsymbol{m}$ is the mean of all feature vectors. You can make the eigenspace of all learning feature vectors as following manner.

$$\boldsymbol{\Sigma} = E\{(\boldsymbol{v} - \boldsymbol{m})(\boldsymbol{v} - \boldsymbol{m})^T\} \tag{5}$$

If $1 \leq j \leq d = 141$, then the solution of the following eigen problem (6) can make the subspace of κ eigenvectors corresponded by κ largest eigenvalues ($\kappa < d$.)

$$\boldsymbol{\Sigma}\boldsymbol{u}_j = \lambda_j \boldsymbol{u}_j \tag{6}$$

Each feature vector can be projected to the position in the subspace above as the vector $\boldsymbol{y}$.

$$\boldsymbol{y} = (\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{\kappa-1})^T \boldsymbol{v} \tag{7}$$

Each user can select the favorite scenes as the group of frames that the user points out by hand. For example, user A has selected aggressive shooting scene as the frame number $N1$ to $N2$, Corresponding feature vectors of the frame number $N1$ to $N2$ are projected in the subspace calculated above as the vectors from $\boldsymbol{y}_{N1}$ to $\boldsymbol{y}_{N2}$. These vectors $\boldsymbol{y}_{N1}$ to $\boldsymbol{y}_{N2}$ are the representation of user A's favorite scenes in the subspace.

After these learning process, you can select the user A's favorite candidate frames as the collection of frames which subspace vectors $\boldsymbol{y}_{A'sfavorite}$ satisfied by the following equation.

$$distance(\boldsymbol{y}, \boldsymbol{y}_A) \leq dis_{thres} \tag{8}$$

for any $A = N1$ to $N2$, where $distance(\boldsymbol{y1}, \boldsymbol{y2})$ is the distance between $\boldsymbol{y1}$ and $\boldsymbol{y2}$. In this paper, we use the square root of inner product of subtractions of two vectors. dis_{thres} is the threshold value of the distance.

Each frame has nine set of feature vectors mentioned in the previous section, so there are a lot of chances that many not similar frames can be selected as the A's favorite scenes. Then we apply the following selection rule to select candidate frames.

1. Select the frame if the number of satisfied Equation 8 from nine feature vectors of each frame is above the threshold number $num1_{thres}$.

2. Select the shot (means time series of frames which are captured by the same camera) if the ratio of the number of selected frames in the shot is above the threshold $num2_{thres}$.

We assume that shot boundary detection is done previously.

5. Experiments and discussions

We applied the proposal method to a broadcast soccer video which resolution is 320×240, the frame rate is $30/sec$. The subspace has been made by the feature vectors of learning video which includes five shooting scene and one goal scene (the length of video is about 20 minutes = 36000 frames. And there are 163 shots.) User A has selected 108 frames (one shot) as the example of his favorite scene (these frames are the shooing and goal scene.) The distance threshold $dis_{threshold}$ is 0.3, $num1_{thres}$ is 4, and $num2_{thres}$ is 50From the learning video, the number of selected shots as user A's favorite scene is 7 shots which include similar shooting shots (3 shots), similar but non-shooting shots (3 shots), and not-similar scene (1 shot). The learning video includes five shooting scene (one of them is a goal scene which is selected by user A as the favorite scene), and two of five (not include one goal scene) are quite aggressive shooting scene similar to the goal scene, but the rest two shooting scenes are not similar to the goal scene (one is the scene of the long kick shoot, and the other is the scene that the length is very short.) Then ground truth of user A's favorite scene is these three shooting scene, one is the goal scene, 108 frames, the second is the shooting scene, 139 frames which include 90 frames collected by our proposed method, and the third is the shooting scene, 109 frames which include 60 frames selected by our method. Other selected 3 shots are not shooting scenes but very similar scene to A's selected shooting scene. The rest 1 shot is not shooting scene and not similar scene to A's favorite scene. The summary of this result is represented by the two indices, *recall* and *precision* as follows.

$recall = 6/6 = 100\%$.

$precision = 6/7 = 85.7\%$.

We use 2-dimensional DCT of x-y plane and x-t, y-t planes of frames, so the difference of the angle of the camera seems a little sensitive. But we use three combinations of x-t and y-t planes, then our method can be robust the little difference of the angle of the camera.

6. Conclusions

In this paper, we present a novel framework to selecting personal favorite scenes using *principle component analysis*. This time, we implemented this framework only with image space DCT and time space DCT, and estimated moderately for the training data only from these two of information. If color histograms or audio power spectrum are extracted, our framework can be easily extended by adding them to the feature vectors, and we expect more effective results of scene selection of personal favorite scenes. Our future works are to introduce color histograms and audio information, and to prepare more training and testing data.

References

[1] R. Leonardi, P. Migliorati, "Semantic Indexing of Multimedia Documents, " *IEEE Multimedia*, pp.44–51, April-June 2002.

[2] S. Xinghua, J. Guoying, H. Mei, and X. Guangyou, "Bayesian network based soccer video event detection and retrieval, " *Proceedings of SPIE - The International Society for Optical Engineering 5286*, pp. 71-76, 2003.

[3] N. Uegaki, M. Izumi, and K. Fukunaga, "Multimodal Automatic Indexing for Broadcast Soccer Video," *Proceedings of 14th Scandinavian Conference on Image Analysis*, pp.802–809, June 2005.

[4] Kevin P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning, " *PhD thesis*, University of California, 2002.

[5] J. Assfalg, M. Bertini, C. Colombo, and A. Del Bimbo, "Semantic Annotation of Sports Videos, " *IEEE Multimedia*, pp.52–60, April-June 2002.

[6] N. Uegaki, K. Nakatsuji, M. Izumi, K. Fukunaga, "Automatic indexing for broadcast soccer video using multiple information, " *MIRU2004*, pp.II-329–334, July 2004. (in Japanese)

[7] K. Nakatsuji, N. Uegaki, M. Izumi, K. Fukunaga, "Estimation of Players' Position from Image Sequences of Soccer Game TV Program, " *IEICE Technical Report*, PRMU2003-214, pp.95-100, Jan. 2004. (in Japanese)

[8] N. Uegaki, K. Nakatsuji, M. Izumi, K. Fukunaga, "Tracking of Multiple Players from Soccer Game TV Programs, " *IEICE General Conference*, no.D-12-112, p.273, Mar. 2003. (in Japanese)

Regularization vs. Rank Reduction in Quadratic Classifiers

Yoshikazu Washizawa
Brain Science Institute, RIKEN, washizawa@brain.riken.jp

Abstract

Subspace methods such as CLAFIC use rank reduction, and rank selection is a sensitive problem because useful features can be lost as a result of truncation. This problem can be avoided by using Tikhonov regularization instead of rank reduction. This paper therefore describes a quadratic classifier using Tikhonov regularization, compares it with other quadratic classifiers, and shows experimental results demonstrating its advantages.

1. Introduction

Subspace methods such as CLAFIC (Class feature information compression) [18] have long been used in many kinds of pattern classification problems [11]. These methods use similarity functions to measure the similarity between classes and input patterns. The independence of these functions makes it easy to increase or reduce the number of classes and reject of the classification. This is an important advantage in problems where the number of classes is large. Subspace methods can also be applied to multi-template problems or problems such that samples belong plural classes.

Subspace methods are quadratic classifiers of the form

$$f(\boldsymbol{x}) = \langle \boldsymbol{x}, A\boldsymbol{x}\rangle + \langle \boldsymbol{b}, \boldsymbol{x}\rangle + c, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The rank of A is constrained below a fixed value, but the reason for this constraint is not clear. Even though, the features of a class may be concentrated in a lower dimensional subspace, choosing rank of the classifier is a sensitive problem of in subspace methods. Too much truncation erases valuable features, and too little truncation can make it impossible to separate different classes. Furthermore, since a rank is a natural number, not complete and not differentiable, it is difficult to find the optimal rank, especially, when the input patterns are low dimensional ones.

From the viewpoint of inverse problems, rank reduction is only one way to make ill-conditioned inverse problems more amenable to computation, however, and another is Tikhonov regularization. This paper shows how Tikhonov regularization can be applied to a quadratic classifier to produce a regularized quadratic classifier (RQC). Since the parameter of Tikhonov regularization is real, complete, and differentiable, the optimal parameter can be obtained easily.

Section 4 of this paper presents experimental results demonstrating that in the classification of handwritten digits, in face recognition and in open benchmark classification problems, a RQC is more accurate than CLAFIC.

The notations used in the paper are listed in Table 1.

Table 1. Notations

$\mathbb{R}^n$	n-dimensional Euclidean space.
$\mathbb{R}^{n\times m}$	set of all of $n \times m$ real matrices.
d	dimension of input vector
$\boldsymbol{x} \in \mathbb{R}^d$	input pattern vector
R_j	correlation matrix of samples of jth class
$\langle \cdot, \cdot \rangle$	inner product
$\Vert \cdot \Vert$	l_2 norm
$\cdot^\top$	transpose of vector and matrix
$\Vert \cdot \Vert_F$	Frobenius norm
$A^\dagger$	Moore-Penrose pseudo inverse of A
I_n	identity matrix of size n
$f_i(\boldsymbol{x})$	similarity function of class i
$E_{\boldsymbol{y}}[\cdot]$	ensemble mean with respect to $\boldsymbol{y}$
$\mathcal{R}(A)$	range of A
$\mathcal{N}(A)$	null space of A
$\mathrm{diag}(\boldsymbol{y})$	diagonal matrix whose diagonal elements are $\boldsymbol{y}$
$\mathrm{Tr}[A]$	trace of A

2. CLAFIC

This section reviews CLAFIC and redefines it as an optimization problem. In subspace methods, the similarity between class j and input pattern vector $\boldsymbol{x}$ is measured by a function given as

$$f_j(\boldsymbol{x}) = \Vert P_j \boldsymbol{x}\Vert^2 = \langle \boldsymbol{x}, P_j\boldsymbol{x}\rangle \tag{2}$$

$$= \sum_{i=1}^{r} \langle \boldsymbol{u}_i^j, \boldsymbol{x}\rangle^2, \tag{3}$$

where P_j is an orthogonal projection matrix, $\{\boldsymbol{u}_i^j\}_{i=1}^r$ is a set of orthonormal bases of the subspace, and r is the rank of P_j as well as the dimension of the subspace. In CLAFIC, $P_j = \sum_{i=1}^r \boldsymbol{u}_i^j \boldsymbol{u}_i^{j\top}$ is an orthogonal projection matrix onto a Kerhunen-Loève (KL) subspace of the class j [10]. The value of $\boldsymbol{u}_i^j$ is usually not specified uniquely but is determined by eigen vectors of the correlation matrix of class j.

CLAFIC has been used in learning subspace methods [12], parametric eigen space method [9], relative KL transforms [19], and kernel subspace methods [16, 7, 17].

Definition 1 (KL subspace [10])**.** *The Karhunen-Loève (KL) subspace with rank r is the subspace spanned by vectors $\{\tilde{\boldsymbol{u}}_i\}_{i=1}^r$ that are the solution of the following optimization problem;*

$$\begin{array}{ll} \max\limits_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_r} & \mathop{E}\limits_{\boldsymbol{x}}\left[\sum_{i=1}^r \langle \boldsymbol{u}_i, \boldsymbol{x}\rangle^2\right] \\ \text{subject to} & \|\boldsymbol{u}_i\| = 1 \quad {}^\forall i. \end{array} \tag{4}$$

Proposition 1 (KL subspace)**.** *The solution of the following optimization problem is a projector onto the KL subspace of $\boldsymbol{x}$;*

$$\begin{array}{ll} \min\limits_{X}: & \mathop{E}\limits_{\boldsymbol{x}}\|\boldsymbol{x} - X\boldsymbol{x}\|^2 \\ \text{subject to:} & \operatorname{rank}(X) \le r. \end{array} \tag{5}$$

Proofs are in [10, 11].

The optimization problem (4) finds a set of r vectors that extract features of a random vector $\boldsymbol{x}$. The set of vectors extracts more features of $\boldsymbol{x}$ when r is larger, but if r is too large the subspace overlap too much and it is difficult to determine the class. For accurate classification, we therefore have to choose the optimal rank r from a set of integers in $1 \le r < d$.

The optimization problem (5) finds the matrix that minimizes the Euclidean distance between $\boldsymbol{x}$ and $X\boldsymbol{x}$ under the constraint $\operatorname{rank}(X) \le r$. As in the optimization problem (4), a larger r extracts more features of $\boldsymbol{x}$, however too large an r extracts too many features to determine a class. The optimization problem (5) is essentiality not one of rank reduction but one of approximating of $\boldsymbol{x}$ under certain constraints for the matrix X. The rank reduction is one of the implementation of the constraint but not a necessary one. The weakness of CLAFIC that is described in the Introduction can be avoided by using a different constraint.

3. Regularized Quadratic classifiers

3.1. Linear inverse problem and regularization

Inverse problems are described here only briefly. See [4] for details about them. Although linear inverse problems are usually discussed in an infinite dimensional space, here they are described in a real finite dimensional space.

Let us consider following linear equation, for example, a case that observation signal $\boldsymbol{y}$ is transformed from original signal $\boldsymbol{x}$ by a matrix A;

$$\boldsymbol{y} = A\boldsymbol{x}, \tag{6}$$

where $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{x} \in \mathbb{R}^n$. If A is square and non-singular, the solution is $\boldsymbol{x} = A^{-1}\boldsymbol{y}$. If A is not square or A is singular, generalized inverses (g-inverse) such as the minimum norm g-inverse, least squares g-inverse, or Moore Penrose (MP) g-inverse are often used [5]. The MP g-inverse is both minimum norm and least squares, and it is determined uniquely.

Let the singular value decomposition (SVD) of A be

$$A = \sum_{i=1}^r \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top = U\Lambda V^\top, \tag{7}$$

where r is the rank of A, $U = [\boldsymbol{u}_1 \ldots \boldsymbol{u}_r] \in \mathbb{R}^{m\times r}$, $V = [\boldsymbol{v}_1 \ldots \boldsymbol{v}_r] \in \mathbb{R}^{n\times r}$ and $\Lambda = \operatorname{diag}([\lambda_1, \ldots, \lambda_r])$. Suppose that the singular values are sorted in descending order. The MP g-inverse of A is

$$A^\dagger = \sum_{i=1}^r \frac{1}{\lambda_i} \boldsymbol{v}_i \boldsymbol{u}_i^\top = V\Lambda^{-1}U^\top, \tag{8}$$

and the MP solution of the linear equation (6) is

$$A^\dagger \boldsymbol{y} = \sum_{i=1}^r \frac{1}{\lambda_i} \langle \boldsymbol{u}_i, \boldsymbol{y}\rangle \boldsymbol{v}_i. \tag{9}$$

Consider the case that $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n}$ is additive noise. Then the MP g-inverse is

$$A^\dagger \boldsymbol{y} = \sum_{i=1}^r \frac{1}{\lambda_i} \langle \boldsymbol{u}_i, \boldsymbol{y}\rangle \boldsymbol{v}_i + \sum_{i=1}^r \frac{1}{\lambda_i} \langle \boldsymbol{u}_i, \boldsymbol{n}\rangle \boldsymbol{v}_i. \tag{10}$$

The singular values of a matrix usually decrease exponentially. If A has very small singular values λ_j, then the inverses λ_j are very large. Thus even if $\boldsymbol{n}$ is negligible, the second term of eq. (10) becomes large and the solution is very far from $\boldsymbol{x}$. In this case, the problem is ill-conditioned and the matrix is ill-posed. There are several ways make ill-conditioned problems relaxed, and two that are often used are truncated singular value decomposition (TSVD) and Tikhonov regularization.

3.1.1 TSVD

TSVD uses instead of A a matrix $A'_{r'}$ whose rank is truncated to r';

$$A'_{r'} = \sum_{i=1}^{r'} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top. \tag{11}$$

The MP g-inverse of $A'_{r'}$ is

$$(A'_{r'})^{\dagger} = \sum_{i=1}^{r'} \frac{1}{\lambda_i} \boldsymbol{v}_i \boldsymbol{u}_i^{\top}. \tag{12}$$

We call $(A')^{\dagger}$ a truncated MP g-inverse of A. When TSVD is used, very small singular values are neglected and the effect of noise becomes smaller.

Proposition 2 (TSVD)**.** *Let A be a given $m \times n$ matrix, $\boldsymbol{w} \in \mathbb{R}^n$ be a white random vector whose correlation matrix is I_n ($E_{\boldsymbol{w}}[\boldsymbol{w}\boldsymbol{w}^{\top}] = I_n$), and $\boldsymbol{x} = A\boldsymbol{w}$. Then the truncated MP g-inverse of A is one of the solutions of the following optimization problem:*

$$\begin{aligned} \min_{X} : \quad & E_{\boldsymbol{x}} \|\boldsymbol{x} - AX\boldsymbol{x}\|^2 \\ \text{subject to:} \quad & \operatorname{rank}(X) \le r'. \end{aligned} \tag{13}$$

The proof is shown in the Appendix.

In classification problems , $\boldsymbol{x}$ is considered an input pattern and its correlation matrix R can be interpreted as a transformed correlation matrix of a white random vector $\boldsymbol{w}$:

$$\begin{aligned} \boldsymbol{x} &= R^{1/2}\boldsymbol{w} && (14) \\ \underset{\boldsymbol{x}}{E}[\boldsymbol{x}\boldsymbol{x}^{\top}] &= \underset{\boldsymbol{w}}{E}[(R^{1/2}\boldsymbol{w})(R^{1/2}\boldsymbol{w})^{\top}] \\ &= R^{1/2} E_{\boldsymbol{w}}[\boldsymbol{w}\boldsymbol{w}^{\top}] R^{1/2} = R. && (15) \end{aligned}$$

Hence, by letting $A = R^{1/2}$, $P = A(A'_{r'})^{\dagger}$ is a projector onto KL subspace of the class that is the matrix of CLAFIC. TSVD can thus be interpreted as CLAFIC in classification problems.

3.1.2 Tikhonov regularization

Tikhonov regularization (or Tikhonov-Phillips regularization) [15, 13] uses instead of $A^{\dagger}$ following regularized MP g-inverse matrix:

$$\begin{aligned} A_{\mu}^{\dagger} &= (A^{\top}A + \mu^2 I_n)^{-1} A^{\top} && (16) \\ &= A^{\top}(AA^{\top} + \mu^2 I_m)^{-1}, && (17) \end{aligned}$$

where μ is a regularization parameter. If $\mu \neq 0$, the inverse in eqs. (16) and (17) is not singular. If A is symmetric,

$$A_{\mu}^{\dagger} = (A + \mu^2 I_n)^{-1}. \tag{18}$$

The SVD of $A_{\mu}^{\dagger}$ is

$$A_{\mu}^{\dagger} = \sum_{i=1}^{r} \frac{\lambda_i}{\lambda_i^2 + \mu^2} \boldsymbol{v}_i \boldsymbol{u}_i^{\top}. \tag{19}$$

If $\lambda_i \gg \mu$, $\frac{\lambda_i}{\lambda_i^2+\mu^2} \simeq 1/\lambda_i$; and if $\lambda_i \ll \mu$, $\frac{\lambda_i}{\lambda_i^2+\mu^2} \simeq \lambda_i/\mu^2$. Figure 2 shows singular values of $A^{\dagger}$ and $A_{\mu}^{\dagger}$ when $\mu = 0.5$ and elements of A are random values from the uniform distribution in $[0, 1]$. The horizontal axis is the order of sorted singular values. If singular values of A are sufficiently large (i.e., if its inverse is small), the singular values $A^{\dagger}$ and $A_{\mu}^{\dagger}$ are almost the same. The singular values of $A^{\dagger}$ get very large at the end while those of $A_{\mu}^{\dagger}$ do not.

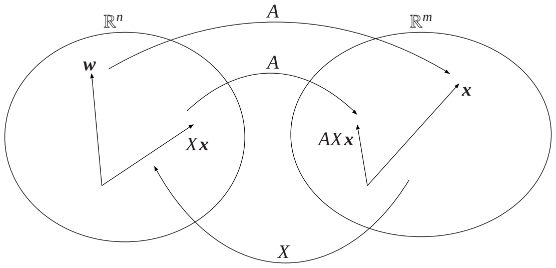


Figure 1. Illustration of TSVD: w is white random vector. The objective is to obtain X that minimizes the distance between x and AXx.

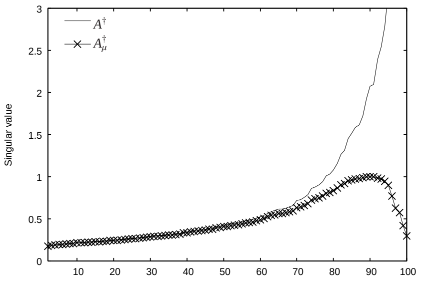


Figure 2. Singular values of $A^{\dagger}$ and $A_{\mu}^{\dagger}$; A is 100×100 matrix of which elements are uniform distribution in $[0, 1]$.

Proposition 3 (Tikhonov regularization)**.** *Let A be a given $m \times n$ matrix, $\boldsymbol{w} \in \mathbb{R}^n$ be a white random vector whose correlation matrix is I_n ($E_{\boldsymbol{w}}[\boldsymbol{w}\boldsymbol{w}^{\top}] = I_n$), and $\boldsymbol{x} = A\boldsymbol{w}$. Then the regularized MP g-inverse of A is one of the solutions of the following optimization problem:*

$$\min_{X} : \quad \underset{\boldsymbol{x}}{E} \|\boldsymbol{x} - AX\boldsymbol{x}\|^2 + \mu^2 \|AX\|_F^2. \tag{20}$$

The proof is shown in the Appendix.

3.2. Regularized quadratic classifier

In the previous section showed that CLAFIC is equivalent to TSVD when $A = R^{1/2}$. This section derives the regularized quadratic classifier (RQC) from the Tikhonov regularization optimization problem.

Definition 2 (Regularized quadratic classifier)**.** *Let $\boldsymbol{x}^j$ be a labeled input pattern vector of the class j, and let B_j be a* *solution of the following optimization problem:*

$$\min_{X} : \underset{\boldsymbol{x}^j}{E} \|\boldsymbol{x}^j - X\boldsymbol{x}^j\|^2 + \mu^2 \|X\|_F^2. \tag{21}$$

The regularized quadratic classifier is the classifier whose similarity function is $f_j(\boldsymbol{x}) = -\|\boldsymbol{x} - B_j\boldsymbol{x}\|^2$, and μ is the regularization parameter.

Theorem 1 (Solution of RQC)**.** *If $\mu \neq 0$, the optimization problem (21) is minimized by*

$$X = R_j(R_j + \mu^2 I_d)^{-1}, \tag{22}$$

where R_j is the correlation matrix of class j.

This is easily proved from the proof of Proposition 3.

In CLAFIC, the similarity functions $f_j(\boldsymbol{x}) = \|P_j\boldsymbol{x}\|$ and $f_i(\boldsymbol{x}) = -\|\boldsymbol{x} - P_j\boldsymbol{x}\|$ are equivalent because P_j is an orthogonal projector. In RQC, however, they are not equivalent. Since the transform by the matrix means approximation of input patterns, the latter function is suitable and natural. In our simulations, it yielded results better than those yielded by the former function.

3.3. Hybrid quadratic classifier

The hybrid quadratic classifier (HQC) is obtained by combining TSVD and Tikhonov regularization.

Definition 3 (Hybrid quadratic classifier)**.** *Let $\boldsymbol{x}^j$ be a labeled input pattern vector of the class j, and let C_j be a solution of the following optimization problem;*

$$\begin{array}{rl} \min\limits_{X} : & \underset{\boldsymbol{x}^j}{E} \|\boldsymbol{x}^j - X\boldsymbol{x}^j\|^2 + \mu^2 \|X\|_F^2 \\ \text{subject to:} & \operatorname{rank}(X) \leq r. \end{array} \tag{23}$$

The hybrid quadratic classifier is the classifier whose similarity function is $f_j(\boldsymbol{x}) = -\|\boldsymbol{x} - C_j\boldsymbol{x}\|^2$.

Theorem 2 (Solution of HQC)**.** *Let the eigenvalue decomposition (EVD) of the correlation matrix of class j be*

$$R_j = \sum_{i=1}^{d} \lambda_i^j \boldsymbol{u}_i^j \boldsymbol{u}_i^{j\top}, \tag{24}$$

where eigenvalues are sorted in descending order. Then the solution of the optimization problem (23) is

$$X = \sum_{i=1}^{r} \frac{\lambda_i^j}{\lambda_i^j + \mu^2} \boldsymbol{u}_i^j \boldsymbol{u}_i^{j\top}. \tag{25}$$

This is easily proved from the proofs of propositions 2 and 3.

4. Experiments

The advantages of the proposed method were evaluated in three kinds of classification experiments.

4.1. Classification of Handwritten digits

The performances of the RQC and CLAFIC were compared by using the MNIST database of handwritten digits, which has 70,000 28x28-pixel samples (60,000 for training and 10,000 for testing). Each of the vectors in this experiment was normalized to the unit norm.

Optimal parameters (rank r of CLAFIC, and regularization parameter μ of RQC) were obtained in the following validation procedure.

- randomly extract from the training set 1,000 validation samples for each class
- from the remaining samples in the training set, construct classifiers with several different parameters
- obtain error rate of validation samples

This procedure was followed 100 times, each with a different selection of validation samples. The results are shown in Fig. 3, where mean values and standard deviations for 100 times trials are plotted.

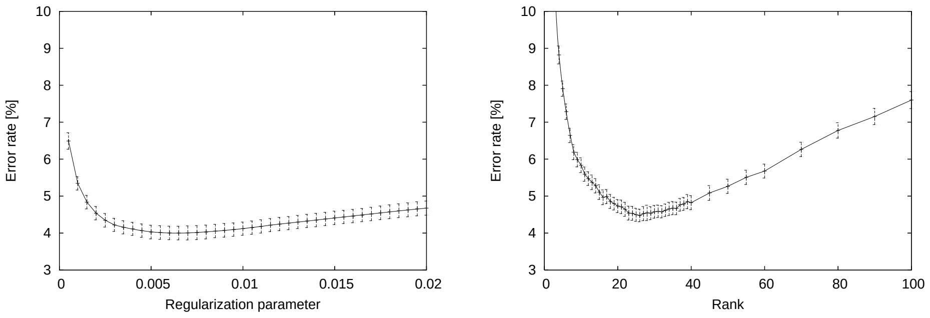
The minimum error rates of validation were $4.48 \pm 0.17\%$ for CLAFIC, and $4.00 \pm 0.18\%$ for RQC. The difference between these rates is significant at the 1% label (one side student t-test). One sees in Fig. (3) that RQC is not sensitive to its regularization parameter but CLAFIC is sensitive to its rank. Furthermore, since rank is an integer smaller than the number of input dimensions, finding optimal rank is not easy. If the plot of error rates against regularization parameter is a convex-downward curve, the optimal parameter can be found in the following way:

1. select any three points for initial points.
2. obtain the validation error rates from those three points.
3. find a quadratic function goes through there three points, and obtain its minimum point
4. remove the worst of point from three points and add the minimum point of the quadratic function
5. go (2)

The test set error rates were 4.08% for CLAFIC and 3.76% for RQC. Thus the proposed method was more accurate than CLAFIC.

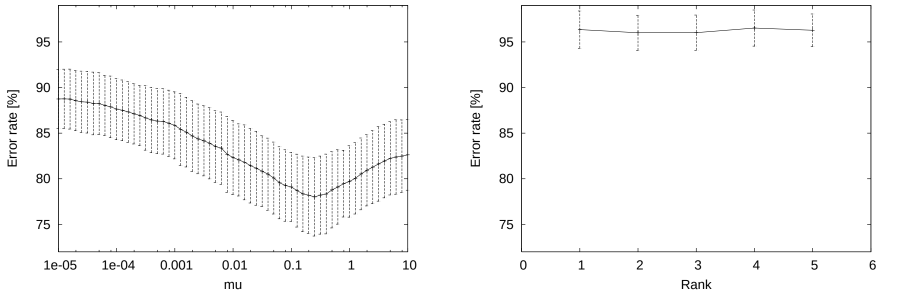
4.2. Face recognition

Experimental results were obtained using the Yale face database [1]. This database consists of 165 320x243-pixel face images, 11 images for each of 15 people. Since each image is very large, for this experiment they were reduced to



RQC CLAFIC

Figure 3. Validation results (mean error rates and standard deviation)



RQC CLAFIC

Figure 4. Results of face recognition; error rates and standard deviation

six dimensional vectors by KL transformation using whole dataset and then normalized. Six dimensional space might be too small for most face recognition problems, because of the small number of samples, we used it for fair comparison.

The procedure of the experiment was as follows

- randomly extract five samples from each individual set for testing randomly
- construct classifiers using the remaining samples
- obtain error rate using test samples

This procedure was repeated 100 times, and the results are shown in Fig. (4). Since this problem was a 15-class classification problem, random classification would have yielded an error rate of 93.3%.

The best results were 96.00 ± 1.92% for CLAFIC and 78.01 ± 4.92% for RQC. RQC thus performed better than CLAFIC.

4.3. Open benchmark test

The datasets for the open benchmark used in [14, 8] were downloaded from (`http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm`). They consisted of 13 binary classification problems, each having 100 or 20 sets of paired training and testing set.

After each vector was normalized to the unit norm, parameters of classifiers were obtained in the following validation procedure:

- extract 10% of the samples in the training set for validation
- construct classifiers using the remaining training samples
- obtain error rates of validation set

This procedure was repeated 100 times with different selections of test set samples. Parameter sets we used were $[1, 2, \ldots,$(input dimension -1)] for CLAFIC and $[10^{-8}, 10^{-7.5}, 10^{-7}, \ldots, 10^{5}]$ for RQC, and in each set the one giving the lowest error rate was designated the optimal parameter.

The error rates, standard deviations and t-test p-values are listed in Table (2). The RQC error rates were lower ones in all classification problems, and 11 of 13 p-values were less than 1%.

Table 2. Comparison of error rates, standard deviations and p-values of t-test between RQC and CLAFIC

Dataset	d	RQC	CLAFIC	p
banana	2	34.73 ± 2.40	35.47 ± 2.40	1.4
b-cancer	9	27.87 ± 2.26	30.80 ± 3.17	0.0
diabetis	8	33.22 ± 2.00	35.82 ± 2.02	0.0
f-solar	9	32.78 ± 1.09	32.98 ± 1.13	10.0
german	20	23.20 ± 1.64	29.07 ± 2.06	0.0
heart	13	16.38 ± 2.32	19.54 ± 2.70	0.0
image	18	11.41 ± 0.57	16.18 ± 0.64	0.0
ringnorm	20	23.01 ± 1.93	33.16 ± 6.22	0.0
splice	60	20.00 ± 6.37	29.20 ± 10.00	0.0
thyroid	5	12.97 ± 1.74	21.59 ± 2.78	0.0
titanic	3	22.14 ± 4.11	24.05 ± 6.70	0.8
twonorm	20	24.99 ± 2.90	33.07 ± 6.87	0.0
waveform	21	23.05 ± 5.22	37.92 ± 4.71	0.0

5. Discussion

5.1. Theoretical comparison with quadratic classifiers

The RQC, HQC, CLAFIC, and pseudo Bayes classifier [6]. are compared here without taking into account the class index.

Let R be the correlation matrix of a class and let its EVD be

$$R = \sum_{i=1}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top = U\Lambda U^\top, \qquad (26)$$

where $U = [\boldsymbol{u}_1 \dots \boldsymbol{u}_d]$ and $\Lambda = \mathrm{diag}([\lambda_1, \dots, \lambda_d])$.

Suppose the following similarity function: $f_1(\boldsymbol{x}) = \|P\boldsymbol{x}\|^2$ for CLAFIC, $f_2(\boldsymbol{x}) = -\|\boldsymbol{x} - B\boldsymbol{x}\|^2$ for RQC, $f_3(\boldsymbol{x}) = -\|\boldsymbol{x} - C\boldsymbol{x}\|^2$ for HQC, and $f_4(\boldsymbol{x})$ for the pseudo Bayes classifier. Then we have

$$\begin{aligned} P &= U\Lambda_1 U^\top & (27)\\ B &= U\Lambda_2 U^\top & (28)\\ C &= U\Lambda_3 U^\top & (29)\\ \mathrm{Hess}(f_3) &= U\Lambda_4 U^\top & (30)\end{aligned}$$

where $\mathrm{Hess}(\cdot)$ denotes the Hessian matrix and

$$\Lambda_1 = \mathrm{diag}([\underbrace{1, \dots, 1}_{r}, \underbrace{0, \dots, 0}_{d-r}]) \qquad (31)$$

$$\Lambda_2 = \mathrm{diag}([\frac{\lambda_1}{\lambda_1 + \mu^2}, \dots, \frac{\lambda_d}{\lambda_d + \mu^2}]) \qquad (32)$$

$$\Lambda_3 = \mathrm{diag}([\frac{\lambda_1}{\lambda_1 + \mu^2}, \dots, \frac{\lambda_r}{\lambda_r + \mu^2}, \underbrace{0, \dots, 0}_{d-r}) \qquad (33)$$

$$\Lambda_4 = \mathrm{diag}(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_r}, \underbrace{\frac{1}{\delta}, \dots, \frac{1}{\delta}}_{d-r}). \qquad (34)$$

Since in CLAFIC a rank of P is truncated to r, features included in the complementary space of P cannot be extracted. CLAFIC performance is therefore sensitive to the selection of rank and is bad if inappropriate rank is used. When the input dimension d is very low, the optimal rank might not be found.

In RQC, on the other hand, the matrix B does not truncate rank. Thus all features in the input vector can therefore be extracted according to their eigenvalues. Thus even if the input dimension is very low, we can find the optimal parameter, and RQC is expected to be less sensitive to the parameter than CLAFIC is.

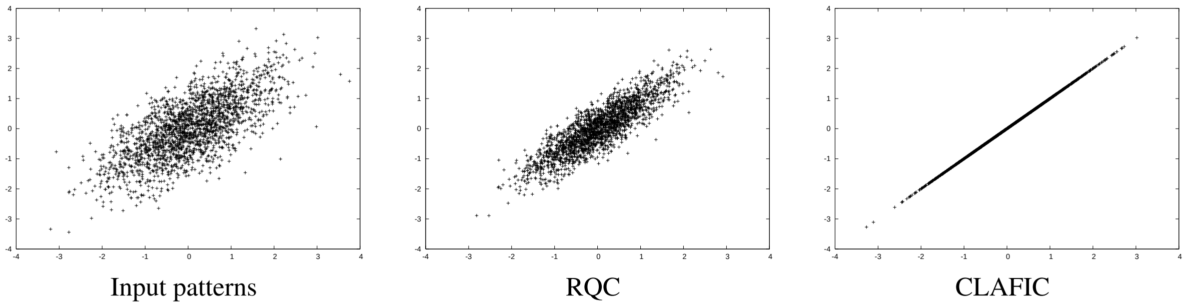
Figure (5) is an illustration of RQC and CLAFIC in two dimensional space. The input vectors are random vectors from a Gaussian distribution whose covariance matrix is $\begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$. In CLAFIC, vectors are projected onto one dimensional space and features in the complementary space vanished. In RQC, on the other hand, vectors are shrunk to the origin, and the degree of shrinking of the first principal component is less than that of the second principal component. Thus, features are extracted according to their eigenvalues.

If the correlation matrix R has very small eigenvalues, they might not be features. Then rank reduction helps reduce calculation cost. In this case, HQC works well. Calculation cost is discussed in the next section.

The pseudo Bayes classifier is also a quadratic classifier. It too uses rank reduction technique, and the reason of replacing eigenvalues to δ is not clear.

5.2. Calculation cost

In the construction stage, most of the CLAFIC, Pseudo Bayes and HQC calculations are EVD and most of the RQC calculations are for an inverse operation of a matrix. The calculation cost of an inverse operation is generally lower than that of EVD. The calculation times for an inverse operation and EVD for a 1000x1000 real symmetric matrix are compared in Table (3), where the values listed are the medians for five trials using the TSUBAME super computer (which has AMD Opteron 880 2.4GHz processors), and GNU Octave software [2] with the Goto BLAS library [3]. One sees from these values that in this system the calculation cost for RQC construction is ten times less than the calculation costs for CLAFIC, Pseudo Bayes, and HQC



Input patterns RQC CLAFIC

Figure 5. Illustration of CLAFIC and RQC in two dimensional space; input vectors are Gaussian random vectors of which variance are $\sigma_{11}^2 = \sigma_{22}^2 = 1$, and covariance $\sigma_{12}^2 = \sigma_{21}^2 = 0.7$.

construction. Moreover, Gaussian elimination can be used for RQC.

# of CPUs	1	2	4
Inverse	0.70	0.45	0.43
EVD	8.55	7.82	11.36
EVD for first 50 eigenvalues	3.74	–	–

Table 3. Comparison of calculation time

In the recognition stage, the only calculation required is multiplication and addition. The numbers of multiplications for one class are as follows;

CLAFIC: $(d + 1) \times r$
RQC: $(d + 1) \times d$
HQC: $(d + 1) \times r$.

Since $r < d$, the calculation costs of CLAFIC and HQC are lower than the calculation cost of RQC.

5.3. Further extensions

CLAFIC has been extended in various ways (e.g., kernel method, learning subspace method, relative KLT or mutual subspace methods). RQC can also be extended in these ways.

6. Conclusion

A quadratic classifier with regularization has been developed, and experimental results show that it is more accurate than CLAFIC and has a lower construction cost.

This work was supported by No. 19700153, Grant-in-Aid for Young Scientists (B).

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[2] J. W. Eaton. GNU octave, `http://www.octave.org/`.

[3] K. Goto. High-performance BLAS, `http://www.tacc.utexas.edu/~kgoto/`.

[4] C. W. Groetsch. *Inverse problems in the mathematical sciences*. Vieweg, 1993.

[5] A. B. Israel and T. N. E. Greville. *Generalized Inverse: Theory and Applications*. John Wiley and Sons, 1974.

[6] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake. Modified quadratic discriminant functions and the application to chinese character recognitio n. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):149–153, 1987.

[7] E. Maeda and H. Murase. Multi-category classification by kernel based nonlinear subspace method. In *IEEE International Conference On Acoustics, speech, and signal processing (ICASSP)*, volume 2, pages 1025–1028. IEEE press., 1999.

[8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[9] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[10] H. Ogawa. Karhunen-Loève subspace. In *Proc. 11th IAPR Int. Conf. Patt. Recogn.*, volume 2, pages 75–78. The Hague, The Netherlands, Aug.-Sep. 1992.

[11] E. Oja. *Subspace methods of pattern recognition*. Wiley, New-York, 1983.

[12] E. Oja and M. Kuusela. The ALSM algorithm - an improved subspace method of classification. *Pattern Recognition*, 16(4):421–427, 1983.

[13] D. L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *Journal of the Association for Computing Machinery*, 9:84–97, 1962.

[14] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, Mar. 2001. also NeuroCOLT Technical Report NC-TR-1998-021.

[15] A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-posed problems*. V. H. Winston and Sons, 1977.

[16] K. Tsuda. Subspace classifier in reproducing kernel Hilbert space. In *International Joint Conference on Neuran Networks (IJCNN)*, pages 3054–3057. IEEE press., 1999.

[17] Y. Washizawa, K. Hikida, T. Tanaka, and Y. Yamashita. Kernel relative principal component analysis for pattern recognition. In *Proc. of Joint IAPR Iinternational Workshops on Syntactical and Structural Pattern Recogn ition and Statistical Pattern Recognition (SSPR/SPR 2004)*, pages 1105–1113, 2004.

[18] S. Watanabe and N. Pakvasa. Subspace method in pattern recognition. *Proc. 1st Int. J. Conf on Pattern Recognition, Washington DC*, pages 25–32, Feb. 1973.

[19] Y. Yamashita, Y. Ikeno, and H. Ogawa. Relative karhunen-loeve transform method for pattern recognition. In *Proc. of International Conference on Pattern Recognition (ICPR 1998)*, pages 1007–1010, 1998.

Appendix

Proof of Proposition 2

The objective function of the problem (13) yields

$$\begin{aligned}
&\underset{\boldsymbol{x}}{E}\|\boldsymbol{x} - AX\boldsymbol{x}\|^2 \\
&= \underset{\boldsymbol{x}}{E}\,\mathrm{Tr}[\boldsymbol{x}\boldsymbol{x}^\top - AX\boldsymbol{x}\boldsymbol{x}^\top - \boldsymbol{x}\boldsymbol{x}^\top X^\top A^\top + AX\boldsymbol{x}\boldsymbol{x}^\top X^\top A^\top].
\end{aligned}$$

Since $E_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^\top] = E_{\boldsymbol{w}}[A\boldsymbol{w}\boldsymbol{w}^\top A^\top] = AA^\top$,

$$\begin{aligned}
&\underset{\boldsymbol{x}}{E}\|\boldsymbol{x} - AX\boldsymbol{x}\|^2 \\
&= \mathrm{Tr}[AA^\top - AXAA^\top - AA^\top X^\top A^\top + AXAA^\top X^\top A^\top] \\
&= \mathrm{Tr}[(AXA - A)(AXA - A)^\top] \\
&= \|AXA - A\|_F^2. \qquad (35)
\end{aligned}$$

Let SVD of A be

$$A = \sum_{i=1}^{r} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top.$$

Since $\mathrm{rank}(AXA) \leq r'$, $\mathcal{R}(AXA) \subset \mathcal{R}(A)$ and $\mathcal{N}(AXA) \supset \mathcal{N}(A)$ clearly, eq. (35) is minimized if and only if

$$AXA = \sum_{i=1}^{r'} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top.$$

Form the operator equation theorem [5], we have

$$\begin{aligned}
X &= A^\dagger \sum_{i=1}^{r'} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top A^\dagger + Y - A^\dagger AYAA^\dagger \qquad (36)\\
&= \sum_{i=1} \frac{1}{\lambda_i} \boldsymbol{v}_i \boldsymbol{u}_i^\top + Y - A^\dagger AYAA^\dagger, \qquad (37)
\end{aligned}$$

where Y is an arbitrary matrix in $\mathbb{R}^{n \times m}$. If $Y = 0$, the solution equals to the truncated MP g-inverse.

Lemma 1. *Let A be any matrix, then*

$$A^\top (AA^\top + \mu^2 I)^{-1} = (AA^\top A + \mu^2 I)^{-1} A^\top. \qquad (38)$$

Proof Let us consider an equation;

$$A^\top (AA^\top + \mu^2 I) = (A^\top A + \mu^2 I) A^\top.$$

By multiplying $(AA^\top + \mu^2 I)^{-1}$ from right hand, and $(A^\top A + \mu^2 I)^{-1}$ from left hand, we have eq. (38).

Proof of Proposition 3

The objective function of the problem (20) yields

$$\begin{aligned}
&\underset{\boldsymbol{x}}{E}\|\boldsymbol{x} - AX\boldsymbol{x}\|^2 + \mu^2\|X\|^2 \\
&= \underset{\boldsymbol{x}}{E}\mathrm{Tr}[\boldsymbol{x}\boldsymbol{x}^\top - AX\boldsymbol{x}\boldsymbol{x}^\top - \boldsymbol{x}\boldsymbol{x}^\top X^\top A^\top + AX\boldsymbol{x}\boldsymbol{x}^\top X^\top A^\top \\
&+ \mu^2 AXX^\top A]
\end{aligned}$$

Since $E_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^\top] = E_{\boldsymbol{w}}[A\boldsymbol{w}\boldsymbol{w}^\top A^\top] = AA^\top$,

$$\begin{aligned}
&\underset{\boldsymbol{x}}{E}\|\boldsymbol{x} - AX\boldsymbol{x}\|^2 + \mu^2\|X\|^2 = \mathrm{Tr}[AA^\top - AXAA^\top \\
&- AA^\top X^\top A^\top + AX(AA^\top + \mu^2 I)X^\top A^\top].
\end{aligned}$$

Let $J(X)$ be the objective function. Then Gateaux differential of J at X with increment Y is

$$\begin{aligned}
&\delta J(X; Y) \\
&= \lim_{\delta \to 0} \frac{1}{\delta} \mathrm{Tr}[J(X + \delta Y) - J(X)] \\
&= 2 \lim_{\delta \to 0} \frac{1}{\delta} \mathrm{Tr}[(\delta Y)^\top (A^\top AX(AA^\top + \mu^2 I) - AA^\top A)].
\end{aligned}$$

For arbitrary matrix Y, δJ is zero if and only if

$$\begin{aligned}
A^\top AX(AA^\top + \mu^2 I) - AA^\top A = 0 \\
A^\top AX = A^\top AA^\top (AA^\top + \mu^2 I)^{-1}. \qquad (39)
\end{aligned}$$

Form the operator equation theorem [5], X that satisfies eq. (39) is

$$\begin{aligned}
X &= (A^\top A)^\dagger A^\top AA^\top (AA^\top + \mu^2 I)^{-1} \\
&\quad + (I - (A^\top A)^\dagger (A^\top A))Y,
\end{aligned}$$

where Y is an arbitrary matrix.

Since $\mathcal{R}(A^\top A) = \mathcal{R}(A^\top)$, $((A^\top A)^\dagger A^\top A)$ is a projector onto $\mathcal{R}(A^\top)$. Hence, we have

$$X = A^\top (AA^\top + \mu^2 I)^{-1} + (I - (A^\top A)^\dagger (A^\top A))Y.$$

If $Y = 0$, from Lemma 1, X is a regularized MP g-inverse matrix of A.

Face Recognition Using Mutual Projection of Feature Distributions

Akira Inoue and Atsushi Sato
NEC Corporation
a-inoue@cp.jp.nec.com, asato@ay.jp.nec.com

Abstract

This paper proposes a new face recognition method using mutual projection of feature distributions. The proposed method introduces a new robust measurement between two feature distributions. This measurement is computed by a harmonic mean of two distance values obtained by projection of each mean value into the opposite feature distribution. The proposed method does not require eigenvalue analysis of the two subspaces. This method was applied to face recognition task of temporal image sequence. Experimental results demonstrate that the computational cost was improved by about 50% without degradation of identification performance in comparison with the conventional method.

1. Introduction

Person identification by face recognition has an advantage of lower psychological stress for uses than other biometrics technologies because it does not use a contact sensor. Therefore, face recognition technologies[1] have gained attention in several applications such as an entrance control system, human machine interfaces and personal robots. However, face recognition technologies in general have a problem of robustness under environment with illumination and pose variations. In recent years, recognition methods by using a temporal image sequence instead of a single image have been suggested to cope with the problem and improve identification performance [2][3][4][5].

In the field of face recognition by using temporal image sequence, Mutual Subspace Method (MSM)[2][3] has been proposed, and it is reported a better recognition performance in illuminant varying environment in comparison with a single image recognition[3]. In MSM, the minimum angle (square of cosine) between two subspaces uses as a similarity measurement between query and enrollment feature distributions. In [4], it has proposed that Kernel function was applied to MSM, and expected to improve in the case of non linear distributions. MSM and the expansion method have better characteristics for robust identification because they only use a few eigen vectors and decrease noise influences. However, they require eigenvalue analysis to compute the minimum angle between the two subspaces. Namely, MSM requires maximizing of the following matrix **X** for each test.

$$\mathbf{X} = \mathbf{U}^{\mathbf{T}}\mathbf{V} \tag{1}$$

where, **U** is formed by the eigen vectors for query feature subspace, and **V** is for enrollment feature subspace.

Recently, Inter-subspace distance (ISD) was proposed for face recognition [5]. This method uses the minimum distance between two subspaces. The method has reported a similar identification performance to MSM, and it also needs an eigenvalue analysis to find the distance. In aspect of practical application, a processing time is an important issue and some applications like robot systems need lower computational cost. However, these conventional face recognition methods[2][3][4][5] for image sequences require eigenvalue analysis of the two subspaces, and this causes an increase of computational cost.

This paper presents a new face recognition method using mutual projection of feature distributions. This method is referred to as, Mutual Projection Method (MPM) in this paper. The proposed method introduces a new robust measurement between two feature distributions. This measurement is computed by a harmonic mean of two distance values gotten by using projection each mean vector into the opposite feature distribution.

In section 2, we describe the algorithm of our proposed method. Experimental results to evaluate performances are demonstrated in section 3.

2. Mutual Projection Method

A new face recognition method, Mutual Projection Method (MPM) is described in this section. Processing flow of MPM is represented by following steps.

(1) Query facial image sequences are entered.

(2) Distance between a query and an enrollment feature distribution is obtained for each person. The inter-distribution distance is referred to as Mutual Projection Distance (MPD). MPD is calculated using two distance values gotten by projecting each mean vector into the opposite feature distribution.

(3) The face recognition is done by choosing the person obtaining smallest MPD.

2.1 Definition of Mutual Projection Distance

Recognition of image sequence is considered to evaluate distance between a query and an enrollment feature distribution formed by the image sequence.

We consider an input feature distribution C_1 and an enrolment feature distribution C_2 .It is assumed that the distances between a feature vector x and the distribution C_1 , C_2 are defined as $d_1(x)$, $d_2(x)$ respectively. If m_1 and m_2 represent the centers of each distribution, the distance between m_1 and C_2 is shown as $d_2(m_1)$. The distance between m_2 and C_1 is also shown as $d_1(m_2)$.

Here, we define a new measurement for inter distribution which is referred to as Mutual Projection Distance (MPD). Desired distance value D between distributions is considered to be represented as the function $d_1(x)+d_2(x)$ for a certain vector x , as shown in Fig.1. The vector x is defined on the line between m_1 and m_2 . $d_1(x)$ and $d_2(x)$ have the minimum value 0 when $x = m_1, m_2$, and these are monotonic increasing functions according to $\|x-m_1\|$ and $\|x-m_2\|$ respectively. In this condition, there exists a feature vector a $\{a \in x\}$ satisfying $d_1(x)=d_2(x)$. The vector a is considered the equal distance point from both distributions' means. Therefore, the distance between the two distributions can be defined the 2 times of the distance value at the point a . In other words, the distance value D is obtained by sum of $d_1(a)$ and $d_2(a)$ (as shown in Eq.(2)). The formula of D is the definition of the inter distribution distance, MPD. Figure 1 shows the distance D for examples of $d_1(x)$ and $d_2(x)$.

$$D = d_1(a) + d_2(a) \tag{2}$$

By combining both distance values $d_1(a)$ and $d_2(a)$, we can take both feature distributions into account for the inter distribution measurement.

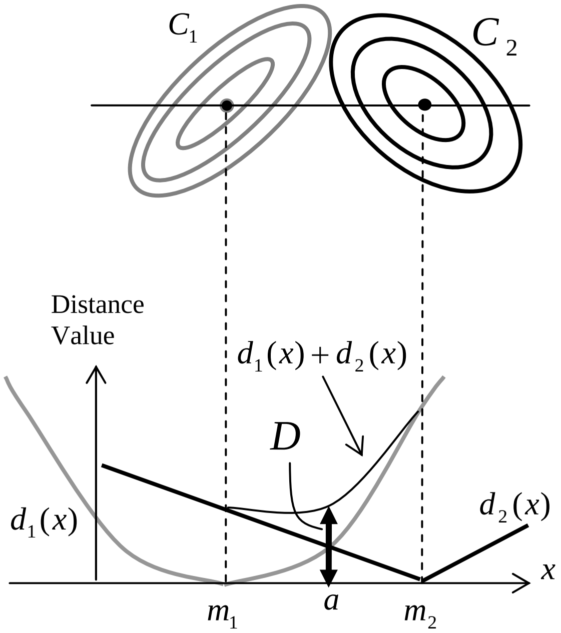


Figure 1: Distance between two distributions

2.2 Computation of Mutual Projection Distance

In order to obtain MPD, it is required the following two steps. At first it needs to compute the distance value between a vector and a distribution, $d_1(x)$ or $d_2(x)$. Then, we combine the two distances by considering the equal distance point from both distributions. These computations are presented in the following subsections.

2.2.1 Distance between a vector and a distribution

The distance a vector and a distribution, $d_1(x)$ or $d_2(x)$ can be calculated by using simple subspace projection. However, we have defined a formula for $d_1(x)$ and $d_2(x)$ based on Mahalanobis distance, in order to consider variances for axes of the subspace.

The Mahalanobis distance is widely used for a normalized distance in the field of pattern recognition. The distance d_m is defined as Eq.(3):

$$d_m^2 = (x-m)^T \Sigma^{-1} (x-m) \qquad (3)$$

where x is a n-dimensional feature vector, m is a mean vector of a distribution and Σ shows a covariance matrix. However, Mahalanobis distance becomes unstable when the Σ is singular. This situation often occurs when the number of training samples of recognition target is small. To avoid this instability, several distance measurements are proposed in this field[6][7]. In this paper, the covariance matrix is estimated as following equation.

$$\Sigma = \hat{\Sigma} + \sigma^2 \mathrm{I} \qquad (4)$$

In Eq.(5), $\hat{\Sigma}$ is a covariance matrix calculated using training samples, and the second term shows an initial estimation (σ^2 is a constant and I is an identity matrix.). Eigenvalues and engenvectors obtained are represented as λ_i and Φ_i. Then, Mahalanobis distance formula is transformed into Eq.(5).

$$d_M^2 = \sum_{i=1}^{n} \frac{1}{\lambda_i + \sigma^2} \{\Phi_i^T (x-m)\}^2 \qquad (5)$$

Components of subspace spanned by eigenvectors with small envenvalues are dominated by noise. Therefore, we assumed $\lambda_i << \sigma^2$ when $i > k$, and obtained the formula Eq.(6).

$$d_p^2(x) = \|x-m\|^2 - \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i + \sigma^2} \{\Phi_i^T (x-m)\}^2 \quad (6)$$

In this paper, $d_P(x)$ which is the root of Eq.(6), are called Pseudo Mahalanobis Distance (PMD). In our method, PMD is used for the distance function $d_1(x)$ and $d_2(x)$. It is obvious that PMD coincide with the projective distance to the subspace in the case of $\sigma^2 = 0$. The transformations of the equation from Eq.(4) to Eq.(6) are based on the literature [6].

2.2.2 Combining two projective distances

Figure 2 shows a distance value when PMD is applied. It is assumed that the statistical properties (mean, eigenvalues and eigenvectors) of one distribution are represented as $\{m_1, \lambda, \Phi\}$ and the other distribution is represented as $\{m_2, \mu, \Psi\}$. $d_1^2(x)$ and $d_1^2(x)$ using PMD can be described in Eq.(7).

$$\begin{aligned} d_1^2(x) &= \|x-m_1\|^2 - \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i + \sigma^2} \{\Phi_i^T (x-m_1)\}^2 \\ d_2^2(x) &= \|x-m_2\|^2 - \sum_{i=1}^{k} \frac{\mu_i}{\mu_i + \sigma^2} \{\Psi_i^T (x-m_2)\}^2 \end{aligned} \qquad (7)$$

Because $d_p^2(x)$ is proportional to $\|x-m\|^2$ shown in Eq.(6), $d_P(x)$ must be proportional to $\|x-m\|$. The $d_1(x)$, $d_2(x)$ using PMD are defined on the line between m_1 and m_2 in Eq.(8) .

$$\begin{aligned} d_1(x) &= \frac{d_1(m_2)}{L}(x-m_1), \\ d_2(x) &= \frac{d_2(m_1)}{L}(m_2-x) \end{aligned} \qquad (8)$$

where $L = |m_2 - m_1|$. Equal distance point a is obtained by solving $d_1(x) = d_2(x)$.

$$a = \frac{d_1(m_2)m_1 + d_2(m_1)m_2}{d_1(m_2) + d_2(m_1)} \qquad (9)$$

Therefore, Mutual Projection Distance (MPD) is represented as D of Eq.(10) in this condition (derived from Eq.(3) and Eq.(9)).

$$D = 2 \cdot \frac{d_1(m_2) \cdot d_2(m_1)}{d_1(m_2) + d_2(m_1)} \qquad (10)$$

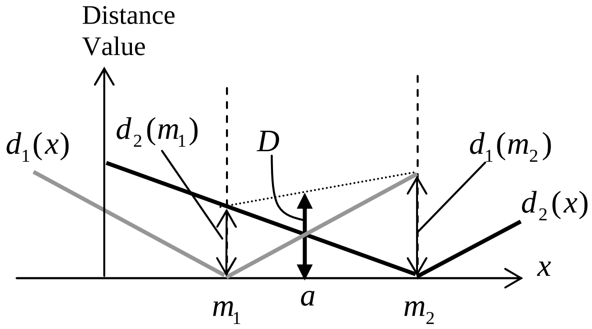


Figure 2: Distance between subspaces using PMD

The D of Eq.(10) is the MPD formula based on PMD. The MPD is computed as a harmonic mean of two PMDs, which obtained by projecting each mean vector into the opposite subspace.

Mutual Projection Method (MPM) is a recognition method by using MPD for measurement of inter-distribution.

3. Experiments for Image sequences

Several applications such as gate control system and human machine interface are considered for the proposed method. These applications are used under various illuminations, and target face images include various poses or expressions. They require robustness in these conditions.

We have applied MPM to several face recognition tasks for image sequences. Two experiments were performed to investigate face recognition performance of MPM. The first experiment used image sequences taken under various illuminations, and the second experiment applied pose variations.

The face recognition experiments have used temporal image sequences for both query and enrollment samples. The following steps were used for the two experiments.

1) Beforehand, statistical properties of enrollment samples $\{m_1, \lambda, \Phi\}$ are calculated for each person.
2) Sequential N frame images are obtained from a query image sequence. (Frame No.: f = 1 to N)
3) Statistical properties of query samples $\{m_2, \mu, \Psi\}$ are calculated for N frame images.
4) MPD is computed for each person and the identification is performed by using the distance.
5) The next N frame images (Frame No.: f=2 to N+1) are obtained. The identification process continues until the sequence ends.

3.1 Pseudo Divergence

Distance between two distributions has been studied for years in statistic research field, such as Bhattacharyya distance and Divergence[8]. In assumption that two distributions are normal, the divergence is represent as Eq.(11).

$$div = \frac{(M_1 - M_2)^t \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)(M_1 - M_2)}{2} + tr\left[\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 + 2 \cdot I\right] \quad (11)$$

Divergence value is thought to be useless for our applications because it needs the inverse of covariance matrices, which is empirically unstable to noise. However, we found the divergence formula of Eq.(11) becomes arithmetic mean of Maharanobis distances by eliminating the second term. Therefore, we applied PMD to the divergence formula, and defined Pseudo Divergence (PD) shown in Eq.(12). This measurement is used for comparison with MPM in following experiments.

$$D_D = \frac{1}{2}(d_1(m_2) + d_2(m_1)) \quad (12)$$

3.2 Experiment I (Under Various Illuminations)

In order to investigate performance under various illuminations, following experiment was performed Facial images used in the experiments were captured by Digital Video Camera in home environment, and they have taken under 12 kinds of illuminations. These facial images were geometrically corrected by both eyes location, and normalized by mean and variance of luminance histogram. Eyes locations were given by hand. Then the face images were resized to 12x18. Figure 3 shows examples of face images used. Table 1 represents the image dataset details.

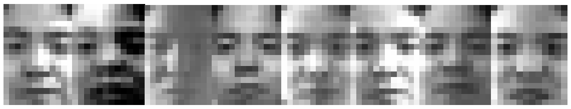


Figure 3: Face image examples

Table 1: Dataset used in experiment I

Environment	Enrollment	Query
12 illuminations At home indoor	200 images for each person (8 persons)	4420 images (16 persons)

In the experiment, a person identification performance was evaluated. The evaluation was done by ROC curve (plotting FRR and FAR). Figure 4 shows the experimental result. The number of the projective dimension k was 20 for query and 20 for enrollment, the number of input frames N was 30, and σ^2 was 0.001. The performance of the proposed method was compared with several other methods: Subspace Method (SM), Mutual Subspace Method (MSM) and Pseudo Divergence (PD). Subspace Method represents the identification method by using a distance of subspace projection. Note that average score of N frames was used in identification with SM.

Experimental results in Fig.4 show that the identification performance (ROC curve) of MPM was better than SM and PD. The results also represent the error rate of MPM was similar to MSM.

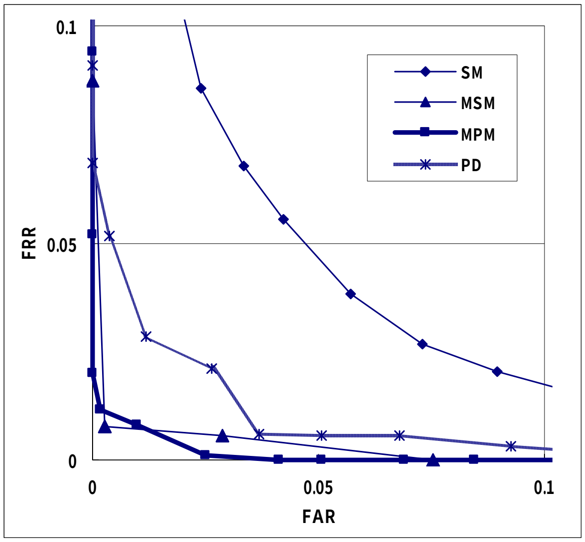


Figure 4: ROC curve of experiment I

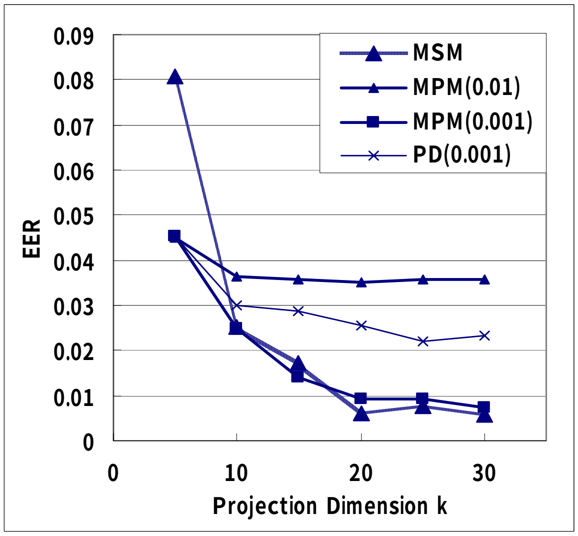


Figure 5: Relationship EER and k (experiment I)

Figure 5 presents EER (the value when FAR=FRR) transition when the projective dimension k varies. In the condition of k=20, EER values of SM, MSM, MPM (σ^2=0.001) and PD are 0.047, 0.0059, 0.0094 and 0.026 respectively.

When k < 20, MPM (σ^2=0.001) performance was better than MSM. However, MSM results were better when k was large. In practice, it is thought that the difference of identification performance between MPM and MSM is quite small. The results indicate that k and σ^2 were important factors for identification performance of MPM.

3.3 ExperimentII (Under Pose Variation)

We investigated performance of MPM under pose variation. Face image dataset we used was consisted of images with pose variation of around 20 degrees and small expression changes. Query image sequences were captured on a few days after we got enrollment image sequences. Subjects have slightly changed their expressions when the query images captured. Table 2 shows the image dataset details. Preprocessing and normalization was same as experiment I.

Table 2: Dataset used in experiment II

Environment	Enrollment	Query
Fixed illuminant Pose variation	800 images for each person (20 persons)	800 images for each person (24 persons)

Figure 6 shows the experimental result. The number of the projective dimension k was 20 for query and 20 for enrollment, the number of input frames N was 30, and σ^2 was 0.001. In Fig.6, MPM showed the best performance among the compared method. SM represents the worst result as we expected. However, MSM performance was worse than PD in experiment II (different from experiment I). This is because that distribution form of the image sequence did not adequate to MSM.

Figure 7 presents EER (the value when FAR=FRR) transition when the projection dimension k varies. In the condition of k=20, EER values of SM, MSM, MPM (σ^2=0.001) and PD are 0.18, 0.054, 0.041 and 0.044 respectively. The results also presented EER values did not depend on the k value in experiment II.

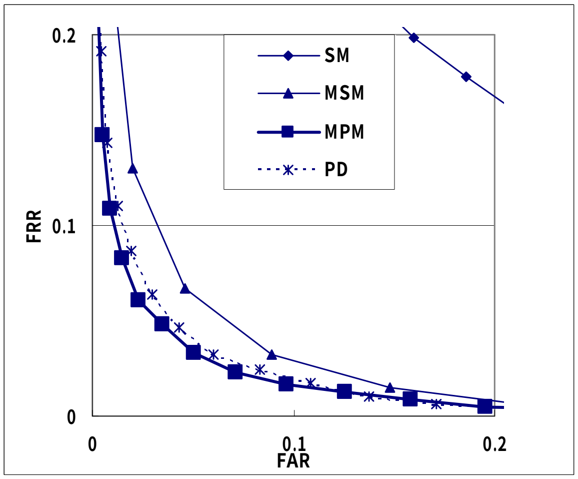


Figure 6: ROC curve of experiment II

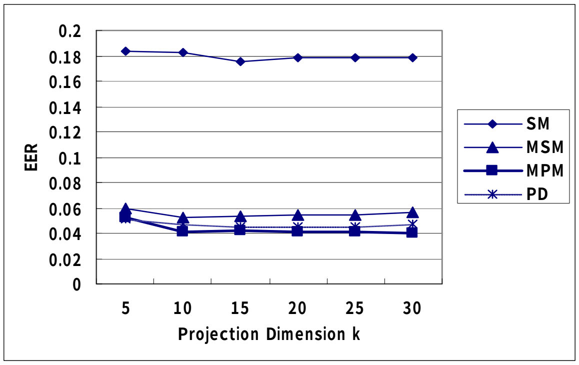


Figure 7: Relationship EER and k (experiment II)

3.4 Comparison of Computational Performance

Table 3 represents an average processing time for each method. The processing time was calculated by taking an average for each person and each query in experiment I. The results have shown that the processing time of MPM was about 50% of MSM.

Table 3: Comparison of processing time

Method	Time (msec)
SM (N frames mean)	32
MSM	23
PD	12
MPM	12

(Frames for each query N=30, on PIII 866MHz)

From the result, MPM has an advantage of computational cost in comparison with MSM. The reason is thought that while MSM needs to compute the maximum eigenvalue of the matrix $\mathbf{X}=\mathbf{U}^{\mathrm{T}}\mathbf{V}$, MPM does not require such an eigenvalue analysis. (PD has also shown a good computational performance by the same reason.)

4. Conclusion

This paper has proposed a new face recognition method using mutual projection of feature distributions. The proposed method, referred to as Mutual Projection Method (MPM), introduced a new measurement between two feature distributions. This measurement was computed by a harmonic mean of two distance values obtained by projection of each mean value into the opposite feature distribution. The MPM does not require eigenvalue analysis of the two subspaces. The MPM was applied to several face recognition tasks of temporal image sequence. The experimental results have demonstrated that the computational cost was improved by about 50% compared with the Mutual Subspace Method. The identification performance was much better than Subspace Method (a single image based method), and represents similar results to the Mutual Subspace Method for various conditions of illumination and facial pose. Therefore, MPM is promising for various applications using image sequence recognition.

Acknowledgements

The authors would like to thank Dr. Akihiko Sugiyama and Dr. Masahiro Serizawa of Common Platform Software Research Laboratories, NEC Corporation for their useful comments on this paper.

References

[1] W. Zhao, et. al., ”Face Recognition: A literature survey,” ACM Comput. Surv., vol.35, no.4, pp.399-458, 2003.

[2] K. Maeda and S. Watanabe, ”A pattern matching method with local structure,” IEICE Trans.,Inf.&Syst.(Japanese Edition), vol.J68-D, no.3, pp.345-352, 1985

[3] O. Yamaguchi, K. Fukui and K. Maeda, “Face recognition using temporal image sequence,” Proc. of IEEE Int. Conf. on AFG, pp.318-323, 1998

[4] H. Sakano and N. Mukawa, “Kernel Mutual Subspace Method for Robust Facial Image Recognition,” Proc. of the 4th Int. Conf. Knowledge based Engineering System, Vol.1, pp.245-248, 2000

[5] J. Chen, S. Yeh and C. Chen, ”Inter-subspace distance: a new method for face recognition with multiple samples,” Proc. of the ICPR, Vol.3, pp.140-143, 2004

[6] F. Kimura et.al, ”Handwritten Numerical Recognition based on Multiple Algorithms,” Pattern Recognition, vol.24,no.10, 1991

[7] B. Moghaddam, A. Pentland, “Probablistic Visual Learning for Object Detection,” Proceedings of the ICCV, 1995

[8] K. Fukunaga, “Statistical Pattern Recognition 2nd ed.,” Academic Press, 1990