Both-hand Gesture Recognition Based on KOMSM with Volume Subspaces for Robot Teleoperation

Martin Peris
CYBERDYNE Inc.
D25-1, Gakuen-Minami, Tsukuba
Ibaraki, Japan 305-0818
Email: martin@cyberdyne.jp

Kazuhiro Fukui
Graduate School of
Systems and Information Engineering
University of Tsukuba, 1-1-1 Tennodai, Tsukuba
Ibaraki, Japan 305-8573
Email: kfukui@cs.tsukuba.ac.jp

***Abstract*—This paper implements a real-time hand gesture recognition system capable of reliably discriminate a potentially large number of hand gestures using both hands to teleoperate a robot. For that we make use of Kernel Orthogonal Mutual Subspace Method (KOMSM, an extension of MSM) and introduce the definition of *Volume Subspace*, which is the type of subspace generated by using depth images from sensors like Microsoft Kinect. We propose to take advantage of the properties of Volume Subspace to accurately classify hand gestures for robot teleoperation.**

I. Introduction

Every day new achievements in the field of robotics are announced and it is very likely that in the future many service robots with different sets of skills will walk and interact among humans [1]. If we are to reach the point when robots and humans coexist and interact on a daily basis, it will be very important that the communication between humans and robots is performed in a way as natural as possible. As it has been proved [2], a big portion of human communication is carried out by hand gestures.

Hand detection and segmentation on complex images has remained a challenging problem forcing researchers to use colored gloves [3], color and depth information [4], motion and color cues [5] or sophisticated camera setups [6] to deal with it.

Recently, the appearance on the market of the Kinect sensor has drastically improved the environment of capturing a depth image of a target object. Thanks to the effectiveness of Kinect and the availability of Open-Source software to interact with it [7], the application range of computer vision techniques has spread rapidly [8], [9].

However, the essential difficulty of 3D object recognition still remains despite tasks such as segmentation and 3D model fitting are partially solved [9], [10]. In dealing with our problem, hand shape recognition, even if a depth image is available as input, it can be difficult to achieve high performance by using only one depth image due to the complexity of hand shapes (two different hand shapes might look very similar depending on their points of view). A shape is just a part of a 3D volume as seen from a certain point of view and to reconstruct the complete 3D volume of an object using only one depth image is impossible. In a similar way to Volume Intersection [11] (which is a method to reconstruct the 3D shape of an object based on multiple silhouettes), to reconstruct the 3D volume of a hand shape multiple view depth images are required. The use of such 3D volume information would completely remove the ambiguity between two hand shapes introduced by different points of view and improve the stability of hand gesture recognition systems.

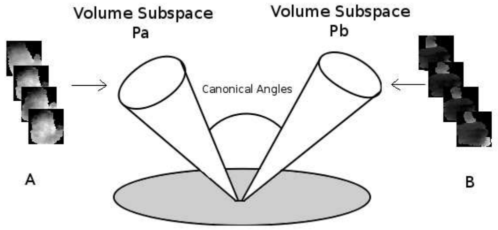


Fig. 1. Proposed framework of hand gesture recognition based on Canonical angles between *volume subspaces*

Therefore, the use of multiple view depth images is required. Inside the category of methods that can effectively classify multiple sets of images, the methods based on MSM (Mutual Subspace Method) achieve high performance on various applications [12]. In this methods, a set of color images is represented by a linear subspace generated by applying PCA (Principal Component Analysis) to that set of images. The similarity between two different image sets is calculated from the Canonical angles between the two subspaces that represents them. However, when the set of images correspond to multiple viewpoints the performance of this methods decreases due to the strong nonlinearity of such image set.

To deal with this, MSM has been extended to several nonlinear methods, for instance: *Kernel MSM* (KMSM) [13] or *Kernel Orthogonal MSM* (KOMSM) [14] .

We propose to use the Kinect sensor to get a sequence of multiple-view depth images of a hand gesture as input set of images for KOMSM. A broader set of gestures can be established using both hands instead of a single-hand, therefore our system will handle both-hand gestures. We chose KOMSM given its excellent performance in the task of hand shape

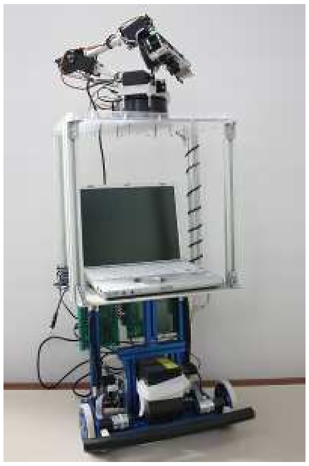


Fig. 2. Prototype robot used in our experiments.

recognition [6], low computational cost and small number of parameters to set. In this framework, a reference subspace of each class is generated from a set of depth images, we will refer to this new kind of subspace as *Volume Subspace*. In the same way as using color images, the similarity between an input volume subspace and each reference volume subspace is measured using the Canonical angles between each subspace as shown in Fig. 1.

Here, we should note that volume subspace characteristics differ to conventional subspace, generated from a set of color images, in the following points:

1) Implicitly contains the information of the object's 3D Volume
2) It is invariant to illumination conditions and object color.
3) It is invariant to translation and scale of the object by simple normalization of size and depth.

The above advantages of our volume subspace against the conventional subspace enable us to achieve high performance hand gesture recognition in challenging environments.

We use then KOMSM to implement a real-time both-hand gesture recognition system using Kinect with the ability to discriminate among a large number of different gestures. The gestures recognized by our system will be transformed into commands to control the prototype robot showed in Fig. 2.

We will perform a series of tests to validate our approach. The test phase will be composed of two parts, a first part where we perform off-line tests to verify the accuracy of our approach and an on-line part to test the behavior of the system in our experimental environment. Regarding the on-line tests we will also propose an heuristic method to improve the stability of our method when transitioning from one gesture to another. The key contributions of this paper are:

- We introduce the concept of *Volume subspace*, which is invariant to illumination and color changes.
- A real-time both-hand gesture recognition system based on KOMSM that is capable of perform well in challenging environments.
- Implementation of the system into a prototype mobile robot for more natural human-robot interaction.

The remainder of the paper is structured as follows: section II describes the proposed system for hand gesture recognition, section III presents the flow of the proposed method, section IV introduces the experiments performed to test and validate our proposed system. We conclude on section V.

II. Proposed Hand Gesture Recognition System

In this section we firstly describe the concept of *Volume Subspace*, we outline the recognition method based on KOMSM, then explain the process of hand detection and finally we define the hand gestures to recognize in our problem.

A. Volume Subspace

As stated before, we will use a sequence of multiple-view depth images obtained using Kinect to recognize a gesture. This sequence of depth images span what we call a *Volume Subspace* (see Fig. 1) which has the following advantages against conventional subspace:

1) Contains the information of the object's 3D Volume. If the number of canonical angles to consider is greater than 3 then the 3D volume information of the object will be implicitly contained in the volume subspace.
2) It is invariant to illumination conditions and object color. In a depth image, the value of each pixel represents the distance from the camera to the object viewed on that pixel. Logically, this distance is not affected by changes in illumination condition or color of the object, therefore the depth image remains invariant to illumination changes and object color.
3) It is invariant to translation and scale of the object. After the object of interest is detected and cropped from a depth image, if we normalize the size of the cropped depth images and we also normalize the values of each pixel by the minimum and maximum values on the cropped depth images, then the volume subspace becomes invariant to translation and scale of the object.

One of the main advantages of using this kind of subspace is that while in color images the subspace highly depends on the illumination condition, with volume subspace recognition can be performed in a very broad set of illuminations (even in no-illumination conditions). Also, in the case of hand shape recognition, volume subspace allows to perform recognition independently of the skin tone of subject in front of the sensor.

B. Recognition Based on KOMSM

1) Measurement between two subspaces: We will use KOMSM, an extension of KMSM, in our work so we will briefly outline the algorithm of KMSM. In KMSM, the distributions of reference patterns and input patterns are represented by nonlinear subspaces, which are generated by Kernel PCA [15]. Then the canonical angles between two nonlinear subspaces are used to quantify the similarity between those subspaces [13], [14].

The canonical angles can be calculated as follows. Given an m_a-dimensional nonlinear input subspace P_a and an m_b-dimensional nonlinear reference subspace P_b in high dimensional feature space, the m_a canonical angles $\{0 \leq \theta_1, ..., \theta_{m_a} \leq \frac{\pi}{2}\}$ between P_a and P_b (for convenience $m_a \leq m_b$) are uniquely defined [13].

If we assume Φ_i and Ψ_i to be the i-th n-dimensional orthonormal basis vectors of the nonlinear subspaces P_a and P_b, then a practical method to find the canonical angles is computing the matrix $X = A^\top B$, where $A = [\Phi_1, ..., \Phi_{m_a}]$ and $B = [\Psi_1, ..., \Psi_{m_b}]$. These orthonormal basis vectors can be obtained from r learning patterns $\{x\}$ of each class by Kernel PCA.

Let $\{k_1, ..., k_{m_a}\}$ be the singular values of the matrix X, then the canonical angles can be obtained as $\{cos^{-1}(k_1), .., cos^{-1}(k_{m_a})\}$

2) Orthogonalization of nonlinear subspaces: In KOMSM, the nonlinear subspaces are orthogonalized in the framework of Fukunaga and Koontz's method [16] before measuring the canonical angles between them. If we define the matrix $G = \sum_{i=1}^{r} P_i$ as the sum of the projection matrix corresponding to the projection onto the class i nonlinear subspace P_i, then the orthogonalization can be achieved by using the whitening matrix O defined as $O = \Lambda^{-1/2} H^\top$, where Λ is the diagonal matrix with i-th highest eigenvalue of the matrix G as the i-th diagonal component, and H is the matrix whose i-th column vector is the eigenvector of the matrix G corresponding to the i-th highest eigenvalue.

3) Definition of similarity: In practice, the similarity between two nonlinear subspaces P_a and P_b is defined as:

$$S = \frac{1}{m_a} \sum_{i=1}^{m_a} \cos^2 \theta_i \tag{1}$$

The value S reflects the structural similarity between two nonlinear subspaces.

4) Reduction of computing time: A drawback of KOMSM is that the computation time increases in proportion to the number of learning patterns, as the kernel trick is used for nonlinear mapping. To reduce the computational complexity the learning patterns are clustered into k clusters using k-means and then the kernel orthogonalization is calculated from the centroids of the k clusters obtained. If k is set to a value smaller than the number of learning patterns, the computing time is remarkably reduced. For details on KOMSM, please refer to [6], [14].

C. Hand Detection

In our approach we will use the functionalities provided by ROS and OpenNI to reliably track the skeleton of the person in front of the Kinect and extract the position of the hands in 3D space as a point cloud [17]. This is performed by the ROS package MIT Kinect Demos [18], this package provides only two point clouds representing the detected hands so we modified it at our convenience to also get the cropped RGB and Depth images of both hands. The depth images are converted to a size of 16x16 pixels, the depth values are normalized and used for the recognition system. The RGB images are used for visualization purposes. Fig. 3 shows the hand detection system used in our work.

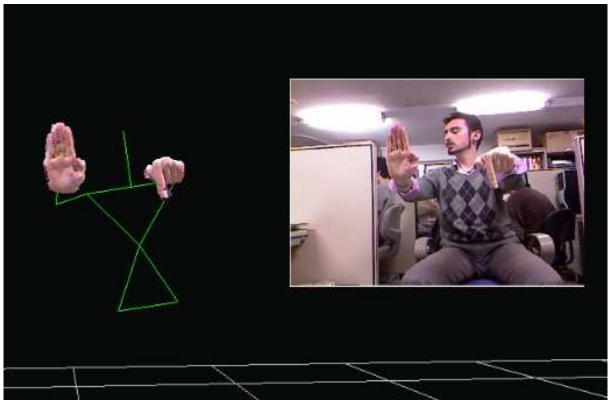


Fig. 3. User skeleton being tracked and detected hand point cloud.

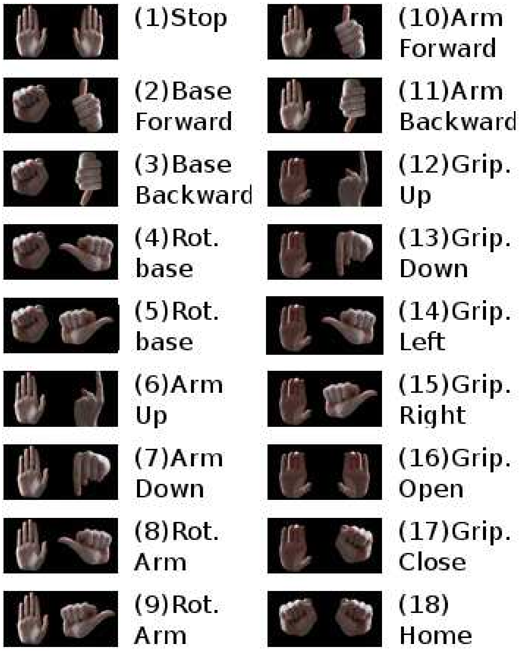


Fig. 4. Both-hand gestures to operate our prototype robot.

D. Hand Gestures Definition

In order to control the motion of the different parts of the robot using hand gestures we defined the set of gestures that can be seen on Fig. 4. Each hand gesture has an action number associated ((1): stop, (2): Move base forward, etc.), when the hand gesture recognition system identifies a gesture it sends the corresponding command to the robot.

In total there is 18 different gestures to control the robot, so our hand gesture recognition system should be able to decide between 18 different classes. Let's consider, for example, gesture (2) in Fig. 4, in this gesture the left hand should have a *fist* shape and the right hand a *thumb-up* shape. If our

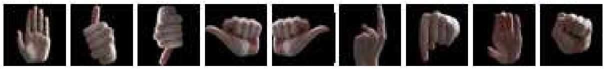
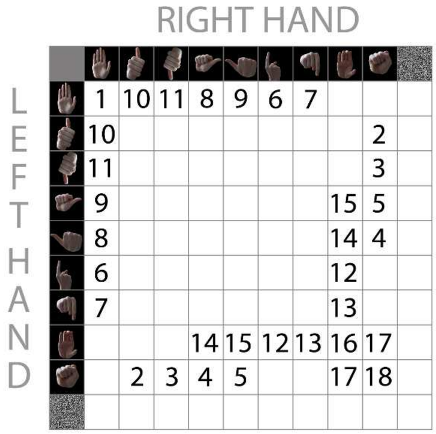


Fig. 5. Single-hand gestures to learn.


RIGHT HAND
LEFT HAND

	1	10	11	8	9	6	7			
	10								2	
	11								3	
	9							15	5	
	8							14	4	
	6							12		
	7							13		
				14	15	12	13	16	17	
		2	3	4	5			17	18	

Fig. 6. Relation matrix between left/right hand gestures and robot action.

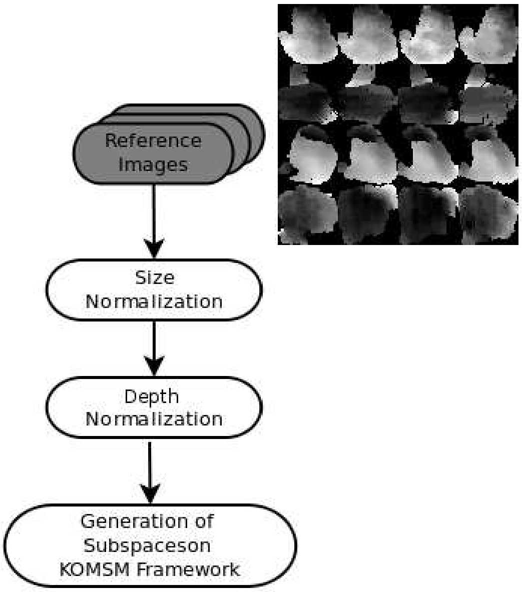


Fig. 7. Flow of the learning phase.

system is only able to classify 18 different gestures then if the user controlling the robot gets confused and shows the *thumb-up* shape with the left hand and the *fist* shape with the right hand, the recognition system would fail. To be independent of the hand order of the gestures our system should be able to recognize 33 different gestures, which is a relatively big number of classes and starts to be computationally expensive and error prone.

To avoid this, we propose to learn only a small number n of gestures with the right hand. We decided to learn 9 different gestures for the right hand plus an extra class to handle spurious effects from the hand segmentation system, so in total $n = 10$. Fig. 5 depicts the single-hand gestures considered in this work. As the gestures that we defined in Fig. 4 are the combination of both hands, we will exploit the fact that the human body is symmetric to transform the images from the left hand to images similar to the ones for the right hand by flipping horizontally the images of the left hand. This will allow us to discriminate as much as n^2 (in our case 100) different gestures. Not all the combinations will have meaning for us so Fig. 6 shows the matrix with the relation of right and left hands and the associated command for the robot. In section IV we will prove that this approach is viable in practice.

E. Robot control

The gestures defined previously are designed to allow the teleoperation of our prototype robot. The robot is formed by a mobile base with a robotic arm mounted on top, the control is carried out by an on-board laptop computer using *Robot Operative System* (ROS) [1]. Under ROS framework the different tasks to process are implemented in *nodes* that can be distributed over a network, each node is in charge of a small part of the whole system and they are all interconnected to achieve more complex tasks. For instance, there is a node to capture depth and color images from Kinect, another node to track a human skeleton on a depth image, etc...

Following this philosophy, the core of the hand gesture recognition (described in section III-B) is implemented into a ROS node. This node receives the images of the detected left and right hands from the process described in section II-C (which is also implemented into a ROS node) performs the hand gesture recognition and outputs the recognized hand gesture class (see Fig. 4). This hand gesture class is the input for another ROS node that sends the pertinent commands to the robot to move according to the detected gesture.

Given ROS's ability to physically distribute the nodes over a network, the nodes related to hand gesture recognition are located in a desktop PC and the nodes related to robot control are located in the on-board PC of the prototype robot. The robot and the desktop PC are connected trough a wireless network allowing us to teleoperate the robot using the hand gestures recognized in the desktop PC.

III. Flow of the proposed method

A. Learning phase

During the learning phase we perform the following steps:

1) Reference images for n different gestures performed by the right hand are captured using the process described in section II-C.
2) The size and the depth values of the reference images are normalized.
3) We generate and store a *Volume Subspace* for each one of the n different gestures using the KOMSM framework.

B. Recognition phase

Then, during the recognition phase, the following steps are performed:

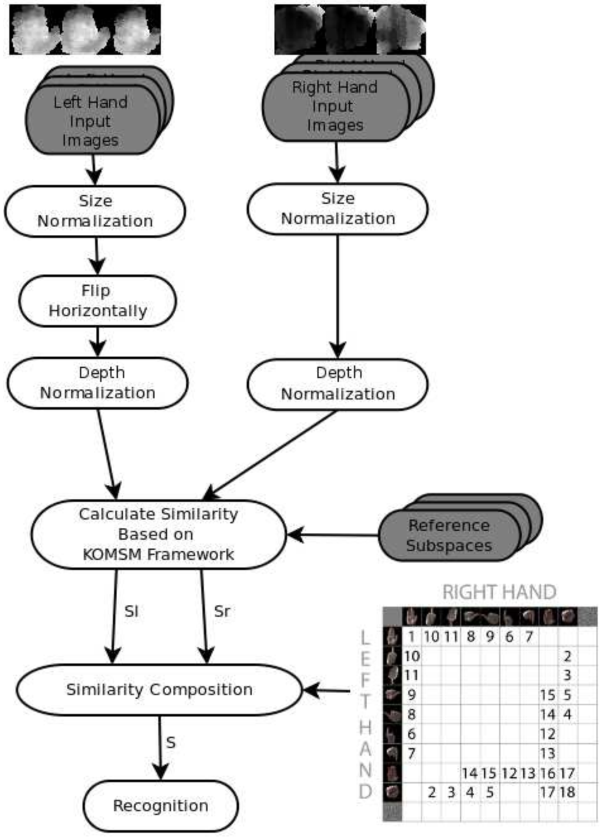


Fig. 8. Flow of the recognition phase.

1) Input images for the left and right hand are captured using the process described in section II-C.
2) The input images for left and right hand are normalized in size.
3) The input images for the left hand are flipped horizontally so that they can be evaluated against the references learned from the right hand.
4) The depth values for the left and right hand input images are normalized.
5) The left and right hand input *volume subspaces* are calculated on the framework of KOMSM. The canonical angles between the left hand input subspace and all the reference subspaces are calculated. This process is also performed for the right hand input subspace. Using equation (1) we calculate S_l and S_r. S_l is a vector of size n that contains the similarity between the left hand input subspace and the n different reference subspaces. Analogously, S_r is a vector of size n that contains the similarity between the right hand input subspace and the n different reference subspaces.
6) We use S_l and S_r, which correspond to the similarities of the left and right single-hand gestures, to compose the matrix S of size $n \times n$ that will contain the similarity to the both-hand gestures with:

$$S_{i,j} = S_{l_i} + S_{r_j} \tag{2}$$

TABLE I
SINGLE-HAND OFF-LINE TESTS VARIATING THE SIZE OF THE INPUT SUBSPACE

Input Size	1	5	10	15	20
Right Hand	96.2%	97.7%	99.1%	99.6%	99.7%
Left Hand	94.8%	95.3%	96.2%	98.1%	98.6%

where $1 \leq i \leq n$ and $1 \leq j \leq n$.

7) Finally we find the recognized both-hand gesture as:

$$\underset{i,j}{\operatorname{argmax}} S_{i,j} \tag{3}$$

where i represents the gesture recognized for the left hand and j represents the gesture recognized for the right hand. As not all the possible combinations of both-hand gestures have meaning for our task, we will use the matrix in Fig. 6 to find out the command to send to the robot.

IV. EXPERIMENTS

To evaluate the performance of our system we have conducted a series of off-line and on-line experiments. For the off-line tests we captured 300 images of each hand using the process defined in II-C for each single-hand gesture and for 3 different subjects. In total we acquired 1.800 images. Among all those images, 100 images per single-hand gesture class of the right hand were used to create the reference *volume subspace* of each single-hand gesture. The same reference *volume subspaces* were used in both, off-line and on-line, tests. The rest of the images were used for testing.

In our experiments the number k of clusters used to reduce KOMSM's computational cost was set empirically to 20. This number of clusters is small enough as to achieve real-time performance while keeping high accuracy in recognition. Another important parameter is the number of canonical angles to calculate. We also set empirically the number of canonical angles to 5 and the coefficient of Gaussian kernel to 0.5 in the experiments.

Once the reference *volume subspace* of each gesture was calculated, we evaluated the performance of the classification of single-hand gesture for each hand variating the size of the input subspace (the number of images in the input sequence). Table I shows the results.

As seen in Table I, despite the fact that the reference *volume subspaces* were generated using only reference images of the right hand, the drop of accuracy for the left hand is barely noticeable, confirming our hypothesis that using horizontally flipped images of the left hand against right hand references would produce good results. Table II shows the accuracy of our method on both-hand gestures.

During the on-line tests the input subspace is updated with every new frame captured by Kinect. As seen on Fig. 10, this may cause instability on the recognition result during transition between gestures because for a time t there will be information of two different classes in the input subspace. To solve this

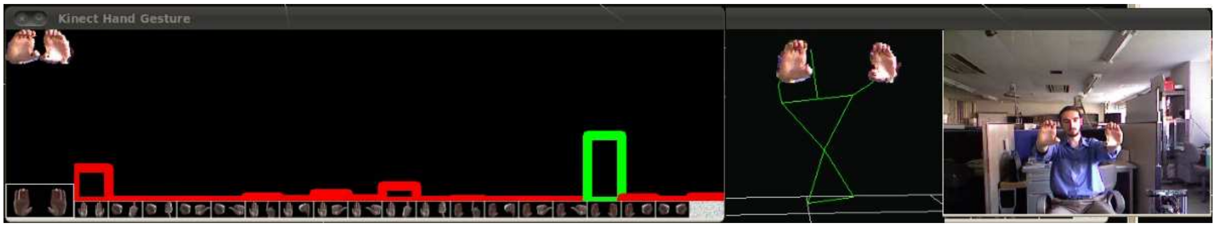


Fig. 9. Screen shot of the software implemented to recognize hand gestures for robot interaction. Left: Recognized hand gesture and similarity plot of all gestures. Center: Skeleton tracking and hand segmentation. Right: Original color image.

TABLE II
BOTH-HAND OFF-LINE TESTS VARIATING THE SIZE OF THE INPUT SUBSPACE

Input Size	1	5	10	15	20
Both Hand	91.1%	93.1%	95.3%	97.7%	98.3%

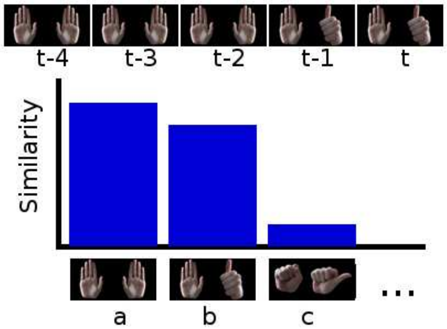


Fig. 10. When transitioning from gesture a to gesture b, the input subspace contains information of both gestures for a short period of time. The correct gesture class at time t is b but the gesture with higher similarity is still a.

issue we heuristically propose to flush the input subspace when the value of the two higher scored gestures is too close. For the on-line tests our method has been implemented into a prototype robot and verified its satisfactory behavior. Fig. 9 shows a screen shot of the system implemented for on-line both-hand gesture recognition.

V. CONCLUSION

In this paper we have proposed a real-time method for both-hand gesture recognition for robot teleoperation using KOMSM and Kinect. We introduced the concept of *volume subspace* which is invariant to illumination and color changes. Our method is capable of achieving high recognition rate over a potentially large number of different hand gestures by only learning a small set of single-hand gestures. The experiments and implementation on a prototype robot validated our proposed method. In future works we will consider how to actively decide which point of view to take during recognition phase to maximize the separability between similar gestures.

VI. ACKNOWLEDGEMENT

This work was supported by KAKENHI (23650081).

REFERENCES

[1] W. Garage, “Robot operative system,” http://www.ros.org.

[2] A. Kendon, *Gesture: Visible Action as Utterance*. Cambridge University Press, Oct. 2004.

[3] J. Alon, V.Athitsos, Q.Yuan, and S.Sclaroff, “Simultaneous localization and recognition of dynamic hand gestures,” *IEEE Motion Workshop*, pp. 254–260, 2005.

[4] M. V. den Berg and L. V. Gool, “Combining rgb and tof cameras for real-time 3d hand gesture interaction,” *Proc. of the IEEE Workshop on Applications of Computer Vision (WACV 2011)*, January 2011.

[5] Y. Fang, K. Wang, J. Cheng, and H. Lu, “A real-time hand gesture recognition method,” *IEEE International Conference on Multimedia and Expo*, pp. 995–998, July 2007.

[6] Y. Ohkawa and K. Fukui, “Hand shape recognition based on kernel orthogonal mutual subspace method,” *IAPR Conference on Machine Vision Applications*, pp. 222–225, 2009.

[7] O. Organization, “Openni,” http://www.openni.org.

[8] P. Doliotis, A. Stefan, C. Mcmurrough, D. Eckhard, and V. Athitsos, “Comparing gesture recognition accuracy using color and depth information,” *Conference on Pervasive Techonologies Related to Assistive Environments (PETRA)*, May 2011.

[9] M. V. den Berg, D. Carton, R. D. Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. V. Gool, and M. Buss, “Real-time 3d hand gesture interaction with a robot for understanding directions from humans,” *20th IEEE International Symposium on Robot and Human Interactive Communication*, August 2011.

[10] Y. Cui and J. Weng, “Appearance-based hand sign recognition from intensity image sequences,” *Computer Vision and Image Understanding*, vol. 2, pp. 157–176, 2000.

[11] A. Laurentini, “The visual hull concept for silhouettebased image understanding.” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 150–162, 1994.

[12] K. Fukui and O. Yamaguchi, “Face recognition using multi-viewpoint patterns for robot vision,” *International Symposium of Robotics Research*, pp. 192–201, 2003.

[13] H. Sakano, N. Mukawa, and T. Nakamura, “Kernel mutual subspace method and its application for object recognition,” *Electronics and Communications in Japan*, 2005.

[14] K. Fukui and O. Yamaguchi, “The kernel orthogonal mutual subspace method and its application to 3d object recognition,” *Asian Conference on Computer Vision (ACCV)*, pp. 467–476, 2007.

[15] B. Scholkopf, A. Smola, and K. Muller, “Nonlinear principal component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[16] F. Fukunaga and W.Koontz, “Applications of the karhunen-loeve expansion to feature selection and ordering.” *IEEE Trans. Computers*, vol. 19, pp. 311–318, 1970.

[17] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[18] MIT, “Mit kinect demos,” http://www.ros.org/wiki/mit-ros-pkg/KinectDemos.