Subspace Selection for Resolution Synthesis

Atsunori KANEMURA†, Shin-ichi MAEDA†, and Shin ISHII††

†, †† Graduate School of Informatics, Kyoto University
Gokasho, Uji, Kyoto 611–0011, Japan
E-mail: †{atsu-kan,ichi}@sys.i.kyoto-u.ac.jp, ††ishii@i.kyoto-u.ac.jp

Abstract Resolution synthesis (RS) is a framework for expanding a given image using an interpolator *trained in advance* with a training dataset. We address how to determine the optimal size of the support for RS using a sparse Bayesian formulation. Experiments show that compact supports can be automatically learned by our Bayesian RS.
Key words Image expansion, resolution synthesis, sparse Bayesian estimation, subspace selection

1. Introduction

Resolution synthesis (RS) [1,2] is a framework for expanding a given image using an interpolator *trained in advance* using a training dataset. Prior training is the characterizing feature of RS and it differentiates RS from classical image expansion methods such as bilinear interpolation and splines. When determining the value of a pixel in a high-resolution image, the bilinear interpolation filter uses at most four low-resolution pixels around the pixel of interest. In contrast, RS in principle can use a support of arbitrary size. Atkins' original RS [1,2] used a 5×5 window without providing logical justification to this choice. The supports should be simple for efficient processing of images and also for preventing overfitting, whereas those that are too simple will deteriorate the expansion performance. We address the problem of determining an optimal support by formulating RS from a viewpoint of sparse Bayesian estimation.

Let r be an integer magnification factor. The purpose here is to estimate an $rM \times rN$ expanded image $\hat{\boldsymbol{\xi}}$ from a given $M \times N$ image $\boldsymbol{\xi}$. In RS, an interpolator called a resolution synthesizer (RSer) expands the image by replacing each one pixel in the given image by an $r \times r$ high-resolution patch. To estimate the high-resolution patch, RS uses the low-resolution pixel patch surrounding the low-resolution pixel to be replaced (Fig. 1). This local interpolation is repeated for every pixel in the given image and the expanded image is constructed by tessellating the high-resolution patches.

In Section 2, we describe the classical maximum likelihood RS (MLRS). Section 3 presents a Bayesian modeling of RS (BayesRS), and we derive an iterative algorithm to find the optimal BayesRSer in Section 4. Experimental results are given in Section 5. Section 6 summarizes this article.

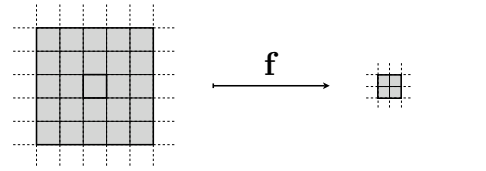

Fig. 1 Resolution synthesis uses $m \times m = Q$ low-resolution pixels to estimate $r \times r = D$ high-resolution pixels.

2. Resolution Synthesis

In advance of real image expansion jobs, we train a RSer using a training dataset. The dataset consists of a large number of low- and high-resolution patches, and the RSer learns the relationship between the low- and high-resolution patches. Let $\mathbf{z}_n$ be the $m^2 = Q$-dimensional vectors of low-resolution patches, $\mathbf{x}_n$ be the $r^2 = D$-dimensional vectors of high-resolution patches, and $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{z}_n)\}_{n=1}^N$ be the dataset consisting of N pairs of the patches. We stack the vectors column-wise and obtain matrices $\mathsf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ and $\mathsf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$.

We assume a linear relationship between $\mathbf{x}_n$ and $\mathbf{z}_n$:

$$\mathbf{x}_n = \mathsf{W}\mathbf{z}_n + \boldsymbol{\mu} + \boldsymbol{\varepsilon}_n, \tag{1}$$

where W is a $D \times Q$ filtering matrix, $\boldsymbol{\mu}$ is a D-dimensional bias vector, and $\boldsymbol{\varepsilon}_n$ is isotropic Gaussian noise with precision (inverse variance) β. Let $\mathbf{w}_d$ be the dth row of W. Then $\mathbf{w}_d$ is the filtering kernel to estimate the dth pixel of the high-resolution patch. Therefore we regard W as a matrix built by stacking D filters. This model leads to the probability distribution of $\mathbf{x}_n$, or the likelihood,

$$p(\mathbf{x}_n|\mathbf{z}_n, \mathsf{W}, \boldsymbol{\mu}, \beta) = \mathcal{N}(\mathbf{x}_n|\mathsf{W}\mathbf{z}_n + \boldsymbol{\mu}, \beta^{-1}\mathsf{I}_D), \tag{2}$$

where $\mathcal{N}(\cdot)$ denotes the Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = |2\pi\Sigma|^{-1/2}e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$ and I_D is the D-dimensional identity matrix.

MLRS estimates the parameters via the maximum likelihood rule

$$(\mathsf{W}^*, \boldsymbol{\mu}^*) = \arg\max_{(\mathsf{W},\boldsymbol{\mu})} \Big(\sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathbf{z}_n, \mathsf{W}, \boldsymbol{\mu}) \Big), \tag{3}$$

whose solution can be easily found as

$$\widetilde{\mathsf{W}}^* = (\widetilde{\mathsf{Z}}\widetilde{\mathsf{Z}}^{\mathrm{T}})^{-1}\widetilde{\mathsf{Z}}\mathsf{X}^{\mathrm{T}}, \tag{4}$$

where $\widetilde{\mathsf{W}}$ and $\widetilde{\mathsf{Z}}$ are extended matrix and vector, respectively, to include $\boldsymbol{\mu}$ and defined as

$$\widetilde{\mathsf{W}} = \begin{bmatrix} \mathsf{W} & \boldsymbol{\mu} \end{bmatrix}, \quad \widetilde{\mathsf{Z}} = \begin{bmatrix} \mathsf{Z} \\ \mathbf{1}^{\mathrm{T}} \end{bmatrix}. \tag{5}$$

MLRS then estimates a high-resolution patch $\mathbf{x}$ from a given $\mathbf{z}$ by the following filtering equation:

$$\mathbf{x} = \widetilde{\mathsf{W}}^* \begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix} = \mathsf{W}^*\mathbf{z} + \boldsymbol{\mu}^*. \tag{6}$$

Note that maximum likelihood estimation inherently suffers from overfitting, that is, increasing the size of the filters beyond certain complexity results in increased generalization errors, although the training errors always decrease [3].

3. Bayesian Modeling of RS

According to the Bayesian framework, all parameters are treated as *random variables*, and prior distributions are put on them as follows:

$$p(\mathsf{W}|\mathsf{A}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} \mathcal{N}(w_{dq}|0, \alpha_{dq}^{-1}), \tag{7}$$

$$p(\boldsymbol{\mu}|\rho) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{0}, \rho^{-1}\mathsf{I}_D), \tag{8}$$

$$p(\beta) = \mathcal{G}(\beta|a_{\beta 0}, b_{\beta 0}), \tag{9}$$

where the gamma distribution is denoted by $\mathcal{G}(\tau|a,b) = \frac{1}{\Gamma(a)}b^a\tau^{a-1}e^{-b\tau}$. We further put hierarchical priors

$$p(\mathsf{A}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} \mathcal{G}(\alpha_{dq}|a_{\alpha 0}, b_{\alpha 0}), \tag{10}$$

$$p(\rho) = \mathcal{G}(\rho|a_{\rho 0}, b_{\rho}). \tag{11}$$

The joint density is decomposed according to the model as

$$\begin{aligned} &p(\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta, \mathsf{X}|\mathsf{Z}) \\ &\quad = p(\mathsf{A})p(\mathsf{W}|\mathsf{A})p(\rho)p(\boldsymbol{\mu}|\rho)\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n, \mathsf{W}, \boldsymbol{\mu}, \beta). \end{aligned} \tag{12}$$

The prior for the filtering matrix W, (7), is similar to that in a sparse Bayesian treatment called *automatic relevance determination* (ARD), which was first introduced for neural networks [4]. The parameters α_{dq} work as regularizers that pull w_{dq} toward the prior mean 0. Therefore, if α_{dq} are large, estimated values of w_{dq} become small. It is known that in this "sparse Bayesian" type of estimation [5], the elements of W irrelevant to the filtering subspace are automatically pruned because the corresponding elements of A diverge to infinity.

The filtering equation that maps a low-resolution patch $\mathbf{z}$ to a corresponding high-resolution patch $\mathbf{x}$ is given by the mean value of the predictive distribution:

$$\mathbb{E}(\mathbf{x}) = \int d\mathbf{x}\, \mathbf{x} p(\mathbf{x}|\mathbf{z}, \mathcal{D}). \tag{13}$$

The predictive distribution $p(\mathbf{x}|\mathbf{z}, \mathcal{D})$ is given by

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}, \mathcal{D}) = \int d\mathsf{A} d\mathsf{W} d\boldsymbol{\mu} d\rho d\beta\; &p(\mathbf{x}|\mathbf{z}, \mathsf{W}, \boldsymbol{\mu}, \beta) \\ &\times p(\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta|\mathcal{D}), \end{aligned} \tag{14}$$

where the posterior is given by the Bayes theorem as

$$\begin{aligned} &p(\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta|\mathcal{D}) \\ &\quad = \frac{p(\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta, \mathsf{X}|\mathsf{Z})}{\int d\mathsf{A} d\mathsf{W} d\boldsymbol{\mu} d\rho d\beta\, p(\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta, \mathsf{X}|\mathsf{Z})}. \end{aligned} \tag{15}$$

However, analytical evaluation of the true predictive distribution is intractable because it is a complex of Gaussian and gamma variables. Therefore, we adopt an efficient computation procedure based on variational estimation.

4. Variational Estimation

The posterior distribution $p(\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta|\mathcal{D})$ is approximated by a trial distribution q, which is a distribution restricted to have a factorization property:

$$q(\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta) = q(\mathsf{A})q(\mathsf{W})q(\rho)q(\boldsymbol{\mu})q(\beta). \tag{16}$$

We denote the latent variables by $\boldsymbol{\tau} = \{\mathsf{A}, \mathsf{W}, \rho, \boldsymbol{\mu}, \beta\}$ for simplicity. Within the restricted distribution space, we search for the optimal trial distribution that minimizes the Kullback-Leiber (KL) divergence to the true posterior distribution:

$$q^\star(\boldsymbol{\tau}) = \operatorname*{argmin}_{q} D_{\mathrm{KL}}(q(\boldsymbol{\tau})\|p(\boldsymbol{\tau}|\mathcal{D})), \tag{17}$$

where the KL divergence is defined by

$$D_{\mathrm{KL}}(q(\boldsymbol{\tau})\|p(\boldsymbol{\tau}|\mathcal{D})) = -\int d\boldsymbol{\tau}\, q(\boldsymbol{\tau}) \ln \frac{p(\boldsymbol{\tau}|\mathcal{D})}{q(\boldsymbol{\tau})} \tag{18}$$

$$= -\Big\langle \ln \frac{p(\boldsymbol{\tau}|\mathcal{D})}{q(\boldsymbol{\tau})} \Big\rangle. \tag{19}$$

Here, $\langle\cdot\rangle$ is the expectation operator with respect to $q(\boldsymbol{\tau})$. The KL divergence is always nonnegative, $D_{\mathrm{KL}}(q\|p) \geqq 0$, for any q and p, and $D_{\mathrm{KL}}(q\|p) = 0$ if and only if q and p

are equivalent distributions. This variational optimization problem can be analytically solved if we optimize only one factor, fixing the other factors. We then iterate computing optimal factors $q^*(\mathsf{A})$, $q^*(\mathsf{W})$, $q^*(\rho)$, $q^*(\boldsymbol{\mu})$, and $q^*(\beta)$ in a sequential manner until convergence to find a minimum $q^\star$.

The optimal trial factors are found as follows:

$$q^*(\mathsf{A}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} \mathcal{G}(\alpha_{dq}|a_{\alpha dq}, b_{\alpha dq}), \tag{20}$$

$$q^*(\mathsf{W}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_d|\mathbf{m}_{\mathrm{w}}^{(d)}, \boldsymbol{\Sigma}_{\mathrm{w}}^{(d)}), \tag{21}$$

$$q^*(\rho) = \mathcal{G}(\rho|a_\rho, b_\rho), \tag{22}$$

$$q^*(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_\mu, \boldsymbol{\Sigma}_\mu), \tag{23}$$

$$q^*(\beta) = \mathcal{G}(\beta|a_\beta, b_\beta), \tag{24}$$

where the parameters are

$$a_{\alpha dq} = a_{\alpha 0} + \frac{1}{2}, \quad b_{\alpha dq} = b_{\alpha 0} + \frac{1}{2}\langle w_{dq}^2\rangle, \tag{25}$$

$$\boldsymbol{\Sigma}_{\mathrm{w}}^{(d)} = \Big(\langle \mathrm{diag}(\alpha_{d1}, \ldots, \alpha_{dQ})\rangle + \langle\beta\rangle \sum_{n=1}^{N} \mathbf{z}_t \mathbf{z}_t^{\mathrm{T}}\Big)^{-1}, \tag{26}$$

$$\mathbf{m}_{\mathrm{w}}^{(d)} = \langle\beta\rangle \boldsymbol{\Sigma}_{\mathrm{w}}^{(d)} \sum_{n=1}^{N} (x_{dn} - \langle\mu_d\rangle)\mathbf{z}_n, \tag{27}$$

$$a_\rho = a_{\rho 0} + \frac{D}{2}, \quad b_\rho = b_{\rho 0} + \frac{1}{2}\langle \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}\rangle, \tag{28}$$

$$\boldsymbol{\Sigma}_\mu = \frac{1}{\langle\rho\rangle + N\langle\beta\rangle}\mathsf{I}_D, \tag{29}$$

$$\mathbf{m}_\mu = \langle\beta\rangle \boldsymbol{\Sigma}_\mu \sum_{n=1}^{N} (\mathbf{x}_n - \langle\mathsf{W}\rangle\mathbf{z}_n), \tag{30}$$

$$a_\beta = a_{\beta 0} + \frac{ND}{2}, \tag{31}$$

$$\begin{aligned} b_\beta = b_{\beta 0} + \frac{1}{2}\sum_{n=1}^{N} \big\{ &\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_n - 2\mathbf{x}_n\langle\mathsf{W}\rangle\mathbf{z} - 2\mathbf{x}_n^{\mathrm{T}}\langle\boldsymbol{\mu}\rangle \\ &+ \mathbf{z}_n^{\mathrm{T}}\langle\mathsf{W}^{\mathrm{T}}\mathsf{W}\rangle\mathbf{z}_n^{\mathrm{T}} + 2\mathbf{z}_n^{\mathrm{T}}\langle\mathsf{W}\rangle^{\mathrm{T}}\langle\boldsymbol{\mu}\rangle + \langle\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}\rangle\big\}. \end{aligned} \tag{32}$$

The expectations remaining in the above equations can be evaluated easily using the well-known results in statistics.

We denote the mean of the joint trial distribution $q(\mathsf{W})$ by M_{W}, that is, we put $\mathsf{M}_{\mathrm{W}} = [\mathbf{m}_{\mathrm{w}}^{(1)}, \ldots, \mathbf{m}_{\mathrm{w}}^{(D)}]^{\mathrm{T}}$. The filtering equation for the variational BayesRS is obtained by substituting the true posterior distribution with the trial distribution, which results in

$$\mathbb{E}(\mathbf{x}) \approx \langle\mathbf{x}\rangle = \langle\mathsf{W}\rangle\mathbf{z} + \langle\boldsymbol{\mu}\rangle = \mathsf{M}_{\mathrm{W}}\mathbf{z} + \mathbf{m}_\mu. \tag{33}$$

As a criterion to check convergence and stop iterating (20)–(24), we monitor the relative change of the Frobenius norm of M_{W}

$$\Delta = \|\mathsf{M}'_{\mathrm{W}} - \mathsf{M}_{\mathrm{W}}\|_{\mathrm{F}} / \|\mathsf{M}'_{\mathrm{W}}\|_{\mathrm{F}}, \tag{34}$$

where M'_{W} is the matrix at the previous iteration step, and terminate the algorithm when $\Delta < 10^{-6}$. To accelerate the convergence, the expected values of α_{dq} are thresholded to infinity when they are greater than e^{20}.

Fig. 2 Images used as for training RSers (4.1.[01–08] in the USC-SIPI image database [6]).

We shall use a hyperparameter setting of the noninformative limit, $a_{\beta 0} = b_{\beta 0} = 0, a_{\rho 0} = b_{\rho 0} = 0$, for β and ρ but we use $a_{\alpha 0} = 20, b_{\alpha 0} = 0$ to facilitate the divergence of α_{dq}. Having zero hyperparameters makes the priors improper, but it is not a problem since the posteriors are well defined.

5. Experiments

We conducted experiments to see which subspace would be selected by BayesRS and to compare the performance of BayesRS with that of MLRS. The expanding factor was chosen to be $r = 2$. The training dataset was prepared by the following procedure. High-resolution patches were prepared by cutting the eight images of size 256×256 shown in Fig. 2 into non-overlapping pieces, resulting in $N = 8 \cdot 256^2/r^2 = 131,072$ patches in total. To make low-resolution patches, first the high-resolution images were shrunk by a factor of 2, and overlapping patches of size $m \times m$ were extracted to produce $131,072$ low-resolution patches. To extract patches near the boundaries, the low-resolution images were extended by replication.

To evaluate the generalization performance in expanding images, we used the Lena image shown in Fig. 7(a) (4.2.04 in the USC-SIPI image database) as the original image $\boldsymbol{\xi}$, which was not included in the training image dataset. This

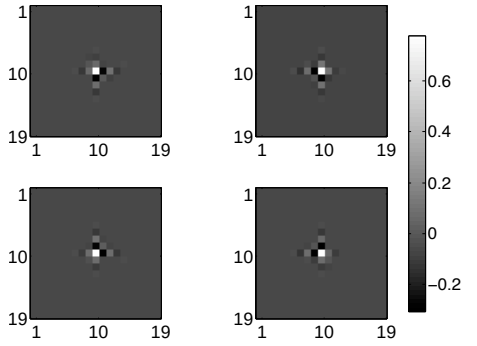

Fig. 3 Learned BayesRS filters (in log scale).

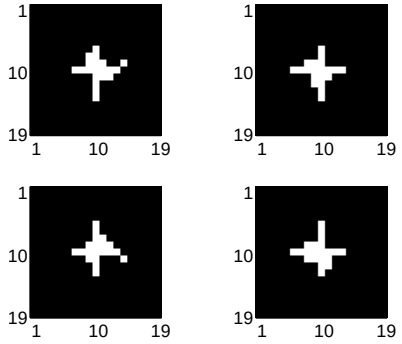

Fig. 4 Supports of BayesRS filters.

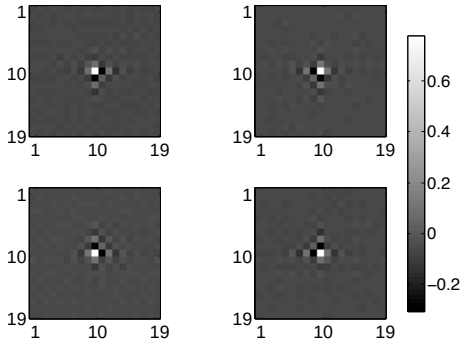

Fig. 5 Learned MLRS filters (in log scale).

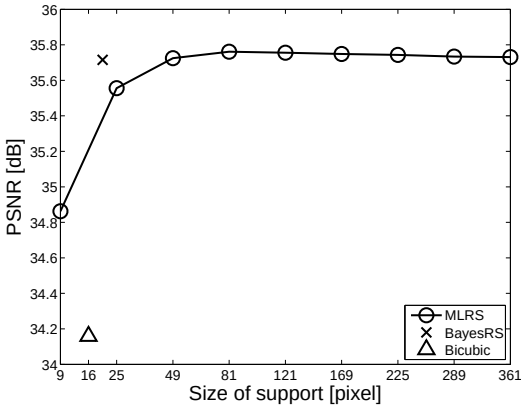

Fig. 6 Performance of RSers with effective sizes of support. The circles connected by line show the performance of the MLRSers and the cross is the one of the BayesRSer. For comparison, the performance of the bicubic interpolation method is shown by the triangle.

image was shrunk by a factor of 2 (Fig. 7(b)) and given to the trained RSers. To quantitatively assess the performance of the RSers, the peak signal-to-noise ratio (PSNR) of the expanded image $\hat{\boldsymbol{\xi}}$ was measured. PSNR is defined by

$$PSNR(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}}) = 10 \log_{10} \frac{\kappa^2}{\|\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}\|^2 / MN} \quad [\text{dB}], \tag{35}$$

where κ is the maximum pixel value and MN is the number of pixels. When displaying filters, we use a log conversion $\text{sign}(w_{dq}) \ln(1 + |w_{dq}|)$, where $\text{sign}(x) = +1/0/-1$ if x is positive, zero, or negative, respectively.

The BayesRS algorithm was executed with the size of the filters being $m \times m = 19 \times 19$. The shapes of the learned filters are shown in Fig. 3, and the supports (regions where the filters had nonzero values) are shown in Fig. 4. The sizes of the learned supports were 20, 20, 20, and 21. An interesting point is that the learned supports had asymmetric shapes. From the shapes of the learned supports, we can say that the direct horizontal and vertical pixels are highly relevant for estimating high-resolution pixels, but the diagonal pixels are of less importance. The expanded Lenna image using the learned filters is shown in Fig. 7(d) and its PSNR was 35.72 dB, which was significantly (about 1.6 dB) better than the image expanded by the bicubic method (Fig. 7(c)). The cross in Fig. 6 indicates the PSNR and its horizontal coordinate is the mean support size (20.25).

Next, we measured the performances of the MLRSers with sizes of the supports varying from $3 \times 3 = 9$ pixels to $19 \times 19 = 361$ pixels. Fig. 5 shows the shapes of the filters trained by the MLRSer when the size of the support was 19×19. There were no nonzero element in the filters. The PSNRs of the MLRSers are shown in Fig. 6 as the circles connected by the line. The maximum PSNR of 35.76 dB was attained when the support size was $9 \times 9 = 81$, and the use of larger supports only degraded the performance, showing a typical overfitting.

6. Conclusion

We showed that automatic selection of the subspace relevant to RS image expansion was successfully achieved using a sparse Bayesian methodology that incorporated the prior setting called ARD. The PSNR of BayesRS's estimation was 0.04 dB worse than that of the best MLRS, which indicates an essentially ignorable loss of performance. The mean size of BayesRS's support, 20.25, was 1/4 of that of the best MLRS, which was significantly smaller. These facts suggest BayesRS should be advantageous for future practical applications.

Acknowledgment

We thank Dr. S. Oba at Kyoto University for his insightful comments on ARD and an early version of this article.

References

[1] C. B. Atkins, Classification-Based Methods in Optimal Im-

(a) 512 × 512 original image.

(b) 256 × 256 low-resolution image.

(c) Bicubic interpolation. PSNR: 34.16 dB.

(d) Bayesian RS. PSNR: 35.72 dB.

Fig. 7 Images.

(a) Original image (close-up).

(b) Low-resolution image (close-up).

(c) Bicubic interpolation (close-up).

(d) BayesRS (close-up).

Fig. 8 Close-up images.

age Interpolation, Ph.D thesis, Purdue University, 1998.

[2] C. B. Atkins, C. A. Bouman, and J. P. Allebash, "Tree-based resolution synthesis," Proc. of PICS Conference, pp. 405–410, Cavannah, GA, Apr. 1999.

[3] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, New York, 2001.

[4] D. J. C. MacKay, "Probable networks and plausible predictions," Network: Compt. Neural. Syst., vol. 6, no. 3, pp. 469–505, 1995.

[5] A. C. Faul, and M. E. Tipping, "Analysis of sparse Bayesian learning," Advances in NIPS 14, pp. 383–389, MIT Press, 2002.

[6] The USC-SIPI Image Database, University of Southern California, `http://sipi.usc.edu/database/`.