FAST IMAGE INPAINTING USING SIMILARITY OF SUBSPACE METHOD

Tomoki Hosoi[1], *Koji Kobayashi*[1], *Koichi Ito*[2] and *Takafumi Aoki*[2]

1 Yamatake Corporation, Japan. E-mail: {t.hosoi.sh, k.kobayashi.ck}@azbil.com

2 Tohoku University, Japan. E-mail: ito@aoki.ecei.tohoku.ac.jp, aoki@ecei.tohoku.ac.jp

ABSTRACT

Image inpainting is a technique for estimating missing pixel values in an image by using the pixel value information obtained from neighbor pixels of a missing pixel or the prior knowledge derived from learning the object class. In this paper, we propose a fast and accurate image inpainting method using similarity of the subspace. The proposed method generates the subspace from many images related to the object class in the learning step and estimates the missing pixel values of the input image belonging to the same object class so as to maximize the similarity between the input image and the subspace in the inpainting step. Through a set of experiments, we demonstrate that the proposed method exhibits excellent performance in terms of both inpainting accuracy and computation time compared with conventional algorithms.

Index Terms— image inpainting, subspace method, eigenspace method, BPLP

1. INTRODUCTION

Image inpainting is a technique for estimating missing pixel values in an image by using the pixel value information obtained from neighbor pixels of a missing pixel or the prior knowledge derived from learning the object class. So far, various image inpainting methods have been proposed in the field of image processing and computer vision. Recently, some famous image inpainting methods such as Navier-Stokes method [1], Criminisi and Perez's method [2], etc. are available through the use of OpenCV [3]. Image inpainting serves a wide range of applications such as removing objects from photographs and retouching the scanned documents [4], and also can be used as a preprocessing task for pattern recognition.

For example, when using pattern recognition methods based on Principle Component Analysis (PCA), it is important to interpolate missing pixels of the input image, since the missing pixels result in low recognition accuracy. In the case of face recognition, it is expected that the missing information occluded by glasses or a mask can be interpolated by using the prior knowledge which the object class is a face.

The general-purpose image inpainting methods such as Navier-Stokes method and Telea's method implemented in OpenCV do not employ the prior knowledge of the object class. So, these methods cannot reconstruct the appearance of the object. Therefore, the image inpainting method using the learning-based approach is suitable for pattern recognition applications. One of the image inpainting methods using the learning-based approach is BPLP (Back Propagation with Lost Pixels) [5, 6]. BPLP interpolates missing pixel information of the input image by using the subspace generated from local regions without missing pixels of the input image. If we use the subspace generated from many images related to the object class instead of the subspace generated from only the input image, BPLP can reconstruct eyes or a mouth occluded by glasses or a mask. However, BPLP is not suitable for real-time applications, since the computational cost of BPLP is relatively high.

In this paper, we propose a fast and accurate image inpainting method using the similarity of subspace method. The proposed method generates the subspace from many images related to the object class and estimates the missing pixel values of the input image belonging to the same object class so as to maximize the similarity between the input image and the subspace. The computational cost for image inpainting can be reduced by estimating the pixel value for each missing pixel. Through a set of experiments using the frontal face images in the Multiple Biometrics Grand Challenge (MBGC) [7] and the images in the Colombia Object Image Library (COIL-20) [8], we demonstrate that the proposed method exhibits excellent performance in terms of both inpainting accuracy and computation time compared with conventional algorithms such as Navier-Stokes method and BPLP.

2. RELATED WORKS

2.1. Pattern recognition using subspace

The subspace method [9] obtains the subspace of the target object class generated from many training samples by applying PCA and recognizes patterns using the similarity between the input image and the subspace.

Let $\mathbf{x}_l$ be an N-dimensional vector consisting of pixel values $x_{l1}, x_{l2}, \cdots, x_{lN}$ of the image $\mathbf{I}_l$ as

$$\mathbf{x}_l = [x_{l1} \quad x_{l2} \quad \cdots \quad x_{lN}]^{\mathrm{T}}. \tag{1}$$

Let $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_M]$ be the learning sample, where M is the number of images.

We calculate the covariance matrix between pixels from $\mathbf{X}$ and obtain the eigenvectors $\mathbf{E}$ as

$$\begin{aligned}\mathbf{E} &= \begin{bmatrix} e_{11} & e_{21} & \cdots & e_{D1} \\ e_{12} & e_{22} & \cdots & e_{D2} \\ \vdots & \vdots & \ddots & \vdots \\ e_{1N} & e_{2N} & \cdots & e_{DN} \end{bmatrix} \\ &= [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \cdots \quad \mathbf{e}_D],\end{aligned} \tag{2}$$

where D $(D \leq M)$ is the number of eigenvectors (it also means the dimension of subspace). The eigenvectors $\mathbf{E}$ are treated as the orthogonal base vectors of the subspace.

The similarity $S(\mathbf{x})$ between the input image $\mathbf{x}$ and the base vector of each dimension $\mathbf{e}_i$ is defined by

$$S(\mathbf{x}) = \sum_{i=1}^{D} \frac{(\mathbf{x}^{\mathrm{T}}\mathbf{e}_i)^2}{\|\mathbf{x}\|^2}. \tag{3}$$

According to the similarity $S(\mathbf{x})$, we can recognize the class of the input image.

2.2. Image inpainting using subspace

In this section, we briefly introduce BPLP [5, 6], which is one of the image inpainting methods using subspace.

The degradation model can be represented by

$$\mathbf{x} = \mathbf{W}\hat{\mathbf{x}}, \tag{4}$$

where $\hat{\mathbf{x}}$ and $\mathbf{x}$ are the original image and the image with missing pixels, respectively. $\mathbf{W}$ is a missing pattern matrix (an $N \times N$ diagonal matrix) defined by

$$\mathbf{W} = \mathrm{diag}(1,1,\cdots,0,\cdots,1), \tag{5}$$

where the values 1 and 0 represent the valid and missing pixels, respectively. Image inpainting can be understood as a problem to estimate the original image $\hat{\mathbf{x}}$ from the image $\mathbf{x}$ after degrading by $\mathbf{W}$. The original image $\hat{\mathbf{x}}$ can be estimated by using the subspace $\mathbf{E}$ and the missing pattern matrix $\mathbf{W}$ as

$$\hat{\mathbf{x}} \cong \mathbf{E}(\mathbf{E}^{\mathrm{T}}\mathbf{W}\mathbf{E})^{-1}\mathbf{E}^{\mathrm{T}}\mathbf{x}. \tag{6}$$

BPLP is one of the image inpainting methods based on the learning-based approach using a lot of sample images belonging to certain class and can be expected to result in accurate interpolation for the input image with the same class. On the other hand, the computational cost of BPLP is relatively high, since BPLP employs the approach to optimize the results using the entire image. Hence, BPLP is not suitable for real-time applications.

3. PROPOSED METHOD

In this section, we describe a fast and accurate image inpainting method proposed in this paper (Fig. 1).

3.1. Learning step

This step is to generate the subspace of the target object class from training images $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_M]$ in the same way in Sect. 2.1.

3.2. Inpainting step

We assume that the input image $\mathbf{x}$ having missing pixels belongs to the same object class as the subspace generated in the learning step. Since it assumes that the base vectors in Eq. (3) are normalized, i.e., $\|\mathbf{e}_i\|^2 = 1$, the similarity $S(\mathbf{x})$ can be rewritten as

$$\begin{aligned} S(\mathbf{x}) &= \sum_{i=1}^{D} \frac{(\mathbf{x}^{\mathrm{T}}\mathbf{e}_i)^2}{\|\mathbf{x}\|^2\|\mathbf{e}_i\|^2} \\ &= \sum_{i=1}^{D} \frac{(\sum_{j=1}^{N} x_j e_{ij})^2}{(\sum_{j=1}^{N} x_j{}^2)(\sum_{j=1}^{N} e_{ij}{}^2)}. \end{aligned} \tag{7}$$

Let the k-th pixel x_k in the input image $\mathbf{x}$ be a missing pixel. In order to interpolate the missing pixel x_k, we introduce $\mathrm{T}(\mathbf{x}, k)$ representing the similarity, which is calculated from x_k and all the valid pixels $x_j (j \in \mathbf{R})$.

$$\mathrm{T}(\mathbf{x}, k) = \sum_{i=1}^{D} \frac{(x_k e_{ik} + \sum_{j\in\mathrm{R}} x_j e_{ij})^2}{(x_k{}^2 + \sum_{j\in\mathrm{R}} x_j{}^2)(e_{ik}{}^2 + \sum_{j\in\mathrm{R}} e_{ij}{}^2)}. \tag{8}$$

The interpolated pixel value $\hat{x}_k$ can be obtained as x_k maximizing $\mathrm{T}(\mathbf{x}, k)$.

$\mathrm{T}(\mathbf{x}, k)$ losts the orthogonality of base vectors due to the lack of the information of the other missing pixels. Note that if we consider that $S(\mathbf{x})$ evaluates the difference of the angle between the input image and the subspace, $\mathrm{T}(\mathbf{x}, k)$ is also appropriate for evaluating the similarity between the input image and the subspace.

$\mathrm{T}(\mathbf{x}, k)$ can be rewritten as

$$\mathrm{T}(\mathbf{x}, k) = \frac{f x_k{}^2 + g x_k + h}{x_k{}^2 + \alpha}, \tag{9}$$

where $\alpha = \sum_{j\in\mathrm{R}} x_j{}^2$, $f = \sum_{i=1}^{D} \frac{e_{ik}{}^2}{e_{ik}{}^2 + \sum_{j\in\mathrm{R}} e_{ij}{}^2}$, $g = \sum_{i=1}^{D} \frac{2e_{ik}\sum_{j\in\mathrm{R}} x_j e_{ij}}{e_{ik}{}^2 + \sum_{j\in\mathrm{R}} e_{ij}{}^2}$, $h = \sum_{i=1}^{D} \frac{(\sum_{j\in\mathrm{R}} x_j e_{ij})^2}{e_{ik}{}^2 + \sum_{j\in\mathrm{R}} e_{ij}{}^2}$.

In order to maximize $\mathrm{T}(\mathbf{x}, k)$ at x_k, x_k has to be a local maximal value of Eq. (9). So, we use the partial derivative of $\mathrm{T}(\mathbf{x}, k)$ as

$$\begin{aligned} \frac{\partial \mathrm{T}(\mathbf{x}, k)}{\partial x_k} &= \frac{\partial}{\partial x_k}\left\{\frac{f x_k{}^2 + g x_k + h}{x_k{}^2 + \alpha}\right\} \\ &= \frac{x_k{}^2 - 2\beta x_k + \alpha}{-(x_k{}^2 + \alpha)^2/g}, \end{aligned} \tag{10}$$

where $\beta = (f\alpha - h)/g$.

In order that x_k is a local maximal value of Eq. (9), the following condition should be satisfied:

$$x_k{}^2 - 2\beta x_k + \alpha = 0. \tag{11}$$

By solving Eq. (11), the estimated pixel value $\hat{x}_k$ is calculated as

$$\hat{x}_k \cong \frac{-\beta \pm \sqrt{\beta^2 - \alpha}}{\beta}. \tag{12}$$

If Eq. (11) has two solutions, the solution having a local maximal value of Eq. (9) should be selected as $\hat{x}_k$. The coefficients $\alpha, \beta_i, \gamma_i$ in Eq. (9) need to be calculated only once and the other coefficients can be calculated very easily. The most of processing time for estimating $\hat{x}_k$ is spent for solving Eq. (11).

By executing these processes for each missing pixel, the interpolated image $\hat{\mathbf{x}}$ can be generated.

Learning step

1. Generate the subspace of the target object class from training images $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_M]$.

Inpainting step

1. Calculate coefficients $\alpha, \sum_{j\in\mathrm{R}} x_j e_{ij}$ and $\sum_{j\in\mathrm{R}} {e_{ij}}^2$ in Eq. (9) within $1 \leq i \leq D$ using all the valid pixels in $\mathbf{x}$.

2. Estimate the interpolated image $\hat{\mathbf{x}}$ according to the following processes for each missing pixel x_k:

   (i) Calculate coefficients f, g and h in Eq. (9).

   (ii) Calculate a coefficient β in Eq. (10).

   (iii) Calculate the estimated value $\hat{x}_k$ by Eq. (12).

Fig.1: Procedure of the proposed method.

4. EXPERIMENTS

We evaluate the inpainting accuracy and computation time of the proposed method and two conventional methods by using MBGC and COIL databases.

The MBGC database [7] contains about 30,000 frontal face images of about 200 individuals. These images were taken under different conditions and the size of these images is not fixed. The COIL-20 database [8] contains 1,440 images of 20 objects (72 images per object) with the image size 128x128 pixels. Each object was placed in a stable configuration at approximately the center of the turntable. The turntable was then rotated through 360 degrees and 72 images were taken per object, that is, the images were taken at every 5 degrees of rotation.

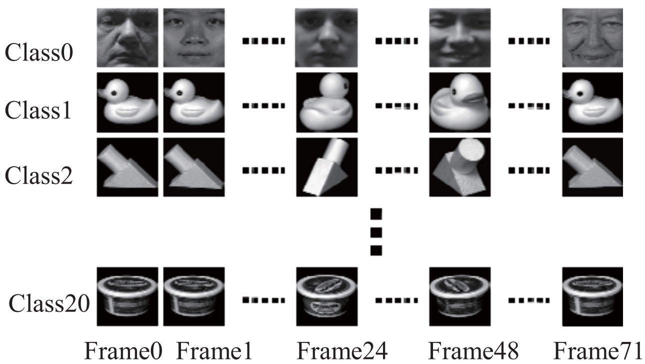

Fig.2: Sample set for the experiments.

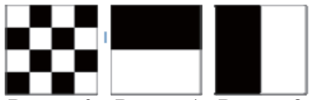

Fig.3: Missing pixels patterns used in the experiments.

In the experiments, we use the sample set consisting of 1,512 images of 21 classes (72 images per class) with 128x128 pixels. Fig. 2 shows example images in the sample set. The images in the Class0 are randomly selected from all the images in the MBGC database. We extract only a face region from the selected images using the face detector [10], and resize the extracted regions to 128x128 pixels. The images in Class1--Class20 are the same as images in the COIL-20 database. The sample images are classified into training images and test images. The sample images labeled even frame number are classified into training images, which are used for generating the subspace of each class. The sample images labeled odd frame number are classified into test images, which are used for evaluating the inpainting performance. The input images are generated from the test images by using missing pixel pattern as shown in Fig. 3. In the experiments, we estimate all the missing pixel values in the input image.

We compare the performance of three image inpainting methods: Naveir-Stokes [1], BPLP [5, 6] and the proposed method. The Navier-Stokes method is implemented by using the OpenCV function “cvInpaint.” BPLP and the proposed algorithm are implemented by C++, where the dimension of subspace is *D*=32 for both methods.

Figs. 4, 5, and 6 show experimental results for Class0, Class1 and Class4, respectively. The experimental results of the Navier-Stokes method are not good, since this method does not use the knowledge of its class. Although the results of BPLP are better than those of the Naivier-Stokes method, the proposed method exhibits the most accurate inpainting results.

We evaluate the inpainting accuracy of each method by calculating RMS (Root Mean Square) error between the original and interpolated images. Table 1 shows the average RMS error of all the test images for each missing pixel pattern. These results indicate that the proposed method is the most accurate.

We evaluate the computation time for each method by using Core2 Duo (2.2GHz) processor. Table 2 shows the summary of the computation time for each method. The computation time of the proposed method is below 13 msec, which is much faster than conventional methods.

As is observed in the above experiments, we demonstrated that the use of the proposed method makes it possible to interpolate missing pixels quickly and accurately.

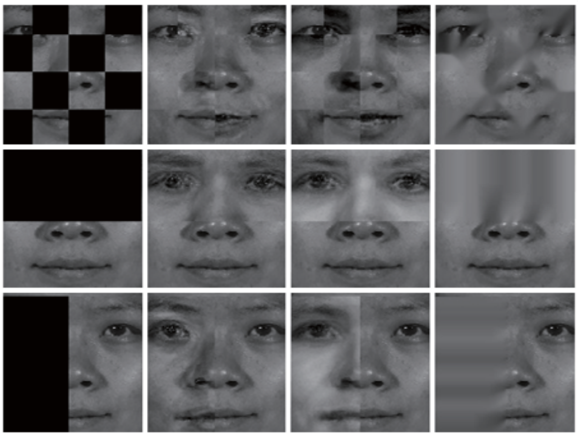

Fig.4: Experimental results for the frame 1 of Class0.

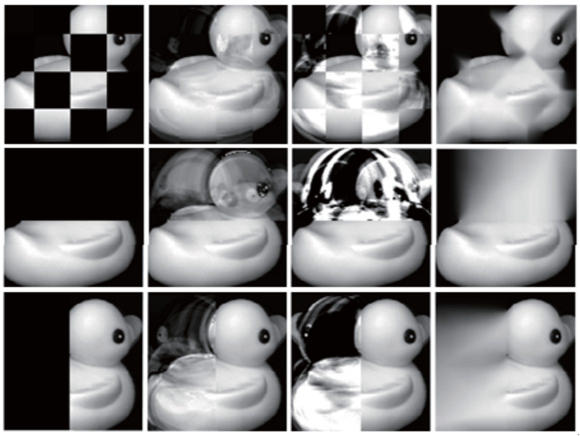

Fig.5: Experimental results for the frame 29 of Class1.

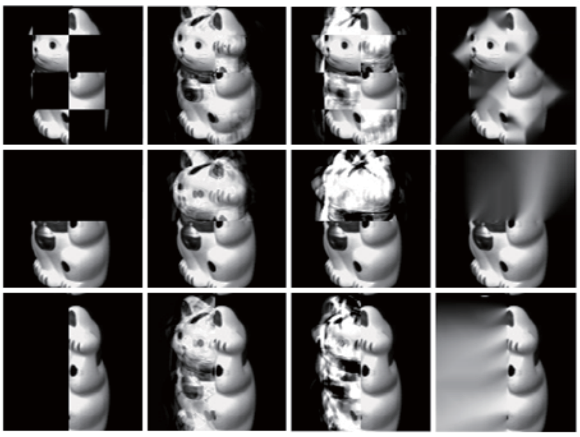

Fig.6: Experimental results for the frame 9 of the Class4.

5. CONCLUSION

In this paper, we have proposed a fast and accurate image inpainting method using the similarity between an input image and the subspace. Through a set of experiments, we have demonstrated that the proposed method can interpolate an image with high accuracy in about 13-msec computation. From these experimental results, the proposed method is expected to be used as a useful preprocessing technique for pattern recognition tasks. As future work, we will develop an automatic missing pixels detection method. Also, we have a plan to improve the inpainting accuracy of the proposed method by extending the linear subspace to the non-liner subspace.

Table 1: Average RMS errors for each method.

Missing pattern	Average RMS error [pixel]		
	Proposed	BPLP	Navier-Stokes
0	30.02	45.88	40.56
1	32.45	52.63	86.54
2	34.79	48.35	58.95

Table 2: Computation time for each method.

Missing pattern	Processing time [msec]		
	Proposed	BPLP	Navier-Stokes
0	10.9	54.7	64.0
1	12.5	56.3	54.6
2	11.0	54.6	54.7

6. REFERENCES

[1] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier-Stokes, fluid dynamics, and image and video inpainting,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 355--362, Dec. 2001.
[2] A. Criminisi, P. Perez, and K. Toyama “Region filling and object removal by exemplar-based image inpainting,” IEEE Trans. Image Processing, vol.13, no.9, pp.1200--1212, 2004.
[3] OpenCV: http://sourceforge.net/projects/opencvlibrary/
[4] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, “Image inpainting,” Proc. 27th Annual Conf. Computer Graphics and Interactive Techniques, pp. 417--424, 2000.
[5] T. Amano and Y. Sato, “Image interpolation using BPLP method on the eigenspace,” IEICE Trans. Information and Systems, vol. J85-D-II, no. 3, pp. 457--465, March 2002 (in Japanese).
[6] T. Amano, “Image interpolation by high dimensional projection based on subspace method,” Proc. 17th Int’l Conf. Pattern Recognition, vo. 4, pp. 665--668, August 2004.
[7] MBGC: http://face.nist.gov/mbgc/
[8] COIL-20: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[9] S. Watanabe and N. Pakvasa, “Subspace methods of pattern recognition,” Proc. 1st Int’l Joint Conf. Pattern Recognition, pp. 283--328, 1973.
[10] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.511--518, 2001.