Example-Based Face-Image Restoration for Block-Noise Reduction

Suhail Hamdan, Yohei Fukumizu, Tomonori Izumi, and Hironori Yamauchi
Ritsumeikan University, Shiga, Japan
Email: {fukumizu, t-izumi}@se.ritsumei.ac.jp

***Abstract*—Highly compressed images from surveillance-camera systems suffer from block noise. We propose two methods to restore such degraded face images based on an example-based Super-Resolution (SR) method. The base method preliminarily generates a database of patches from a training set of face-image examples and reconstructs a High-Resolution (HR) image from a given Low-Resolution (LR) image by using the patches and taking the positions of facial parts into consideration. The proposed methods aim to restore highly-compressed degraded images instead of LR images. One of the proposed methods named as the Direct method, synthesizes a restored texture directly with a given degraded image. The other method, named as the Smooth method, synthesizes a restored texture with a filtered image generated from a given degraded image. The Direct method results in a 3.2 dB improvement in terms of Peak Signal-to-Noise Ratio (PSNR) on average for the lowest quality rate i.e. 1% (around 120:1 compression rate), while the conventional Gaussian-filtering method results in a 2.5 dB improvement. Although the Direct method results in better quality for highly-compressed images compared to the conventional Gaussian-filtering method and Smooth method, unnatural block noise is still observed in the restored images. The Smooth method yields more natural images and better PSNRs for images when the quality rate is around 5%.**

***Index Terms*—example-based, compressed image, image restoration, Gaussian-filtering, Direct method, Smooth method**

I. Introduction

Several methods have been developed to fight crime such as using Closed-Circuit Television (CCTV) installations. Such installations are often used for surveillance in areas, such as banks, airports, and convenience stores that may require monitoring. These CCTVs record digital images of crime footage that are useful as evidence in investigations. Information from images, especially those of criminals' faces, is helpful in solving crimes.

Unfortunately, most security cameras involve long-term recording, resulting in low-resolution and highly compressed images with low frame decimation. The quality of the footage is frequently too poor to be used in investigations since less information can be obtained from such images. It is useless if the obtained images of criminal faces are unidentifiable due to poor image quality. Therefore, it is important to improve the quality of footage as well as increase their quantity to prevent crimes.

Poor-quality images from CCTV's footage are often caused by block noise due to image compression and worsened by dark environments since crimes often occur at night. It is necessary to make images clearer to obtain more information from the surrounding shapes, textures, or even small details as well as reduce noise.

There are several noise reduction methods, such as Point-Spread Function (PSF) [1]-[3]. These methods smooth an image and eventually reduce block noise. The stronger the PSF is, the lesser the noise. However, the high-frequency component such as sharp edges or detailed texture is inevitably lost due to the spread process, which leads to a corrupted and blurred image. Therefore, it is important to restore the missing high-frequency texture component in an image [4]-[7].

Super-Resolution (SR) methods [8]-[16] have been proposed to predict detailed information of missing high-frequency component as well as enhancing their resolution during the image-enlargement process. Amongst them, Freeman et al. proposed example-based method [17]-[19]. Since then, variations of the example-based methods have been proposed [20]-[31].

Example-based methods are significant SR methods that generally enable the production of SR image with high magnification ratios [20]. Compared to conventional interpolation-based methods that only use information from input images, example-based methods can estimate the missing high-frequency components by exploiting external information available in the database [32].

Various image processes benefit from example-based methods such as in panoramic imaging [21], image compression [22], pattern recognition, and 3D modeling [23], [24]. Several methods have been proposed through different approaches in terms of feature extraction and type of learning model. Lui et al. proposed a Markov random field (MRF)-based SR method using wavelet synthesis as feature maps [25]. Yang et al. implemented a dictionary-learning strategy using sparse representation [26], [27]. Jiji et al. used a best-matching model on wavelets [28] and contourlet feature maps [29]. Several years later, Wu et al. proposed combining a contourlet transform for feature extraction and MRF for a learning model [30].

Manuscript received December 20, 2018, revised March 14, 2019.

doi: 10.18178/joig.7.1.9-17

Example-based methods are most suitable for applications that handle a specific type of images such as faces. Recently, an extension of example-based methods [31] is proposed from the observation that each facial part is in the closer position even in different faces. The method in [31] takes the position of each patch in the normalized face image into consideration to select the suitable HR patch, while to Freeman et al.'s method only considers the similarity of appearance and matching of the borders.

We incorporate the idea from [31] into image restoration for noise reduction and propose two methods, both of which construct the database of patch pairs from training images and their degraded images, combining with their positions in the normalized face images. The two methods are named as Direct method and Smooth method. The Direct method reconstructs a restored image by adding restored texture directly to a given degraded image. While the Smooth method generates a restored image by adding restored texture to a filtered degraded image.

We evaluated our methods and compared them with a conventional PSF method, i.e. Gaussian-filter on several sets of degraded face samples that had different JPEG quality rates. The PSF method produces higher quality images than proposed ones for JPEG compression images at a quality rate of 10% and above. On the other hand, the Direct method produces higher quality images for highly degraded images at a quality rate of 3% and below. However, unnatural block noise is still observed in the restored images with the Direct method. The Smooth method yields more natural images and results in better quality for images at a quality rate of around 5%.

The rest of the paper is organized as follows. Section II describes the algorithm of the example-based methods. Section III presents our image-restoration methods. Section IV shows experimental results and discussions and Section V summarizes the paper.

II. Example-Based Method

Freeman et al. proposed an example-based method implementing an MRF network as the learning model for SR [17]. They use external information from a large volume of training LR and HR image pairs. The MRF network statistically models the spatial relationship among patches and the reference from each patch of the input image to a patch in the database.

Given an input LR image, the method first applies upsampling to generate an HR image, where high-frequency component was lost or corrupted due to the enlargement process. The generated image was divided into a low-frequency and middle-frequency component. The middle-frequency component is split into patches, which are the LR patches of the input image.

Fig. 1 illustrates the structure of the MRF network where each circle represents a network node and each line indicates the spatial relationship and statistical dependencies between nodes. The nodes Y(i,j) represents the LR patches at the position (i,j) in the input (observed) image. The nodes X(i,j) represents the LR patches in the database whose corresponding HR patches are used to estimate the high-frequency components at the position (i,j) for the restored (hidden) image.

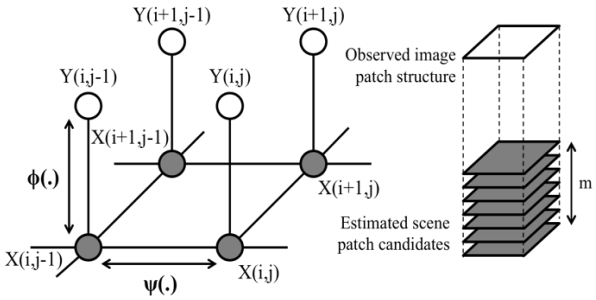


Figure 1. MRF-network-based learning model.

The dependency between nodes are represented as two compatibility functions $\phi(.)$ and $\psi(.)$ in [17]. For a position (i,j) in the MRF network, the function $\phi[X(i,j), Y(i,j)]$ called as image compatibility function, represents the compatibility between the observed patch $Y(i,j)$ and the estimated patch $X(i,j)$. For a position (i,j) and its adjacent position (u,v), the function $\psi[X(i,j), X(u,v)]$ called as border compatibility function, represents the compatibility of the common border between the estimated patches $X(i,j)$ and $X(u,v)$.

The joint probability of $X(.)$ under the condition $Y(.)$ is defined as

$$P(X|Y) = \prod_{ij} \phi[X(i,j), Y(i,j)] \times \prod_{ij,(u,v)\in NB(i,j)} \psi[X(i,j), X(u,v)] \quad (1)$$

where $NB(i,j)$ is a set of right, left, top, and bottom neighbors of (i,j) in the MRF network. In order to reduce the computational load, a number of LR patch candidates for $X(i,j)$ are previously selected based on the image compatibility function. The number of patch candidate is given as a constant parameter m.

The image compatibility function $\phi(.)$ is defined as

$$\phi[X(i,j), Y(i,j)] = \exp\left\{-\frac{d[X(i,j), Y(i,j)]}{2\sigma_1}\right\} \quad (2)$$

where $d(.)$ is the distance of the two matrices (or vectors) and σ_1 is a constant parameter. We use the root sum of the squared differences of the pixels in the two patches.

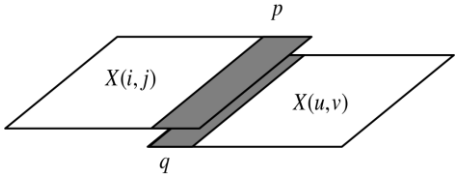


Figure 2. Region of overlap.

Fig. 2 outlines the overlap region between two adjacent patches. The border compatibility, $\psi(.)$ is defined as

$$\psi[X(i,j), X(u,v)] = \exp\left[-\frac{d(p,q)}{2\sigma_2}\right] \quad (3)$$

where $p(q)$ is the vector of pixels of the overlap region in patch $X(i,j)$ ($X(u,v)$, respectively) and σ_2 is a constant parameter.

Given an initial set of candidates for $X(.)$, the method iteratively changes each candidate for $X(i,j)$ in turn to improve $P(X|Y)$ greedily until no significant improvement is observed. The HR patches corresponding to the final candidates for $X(i,j)$ forms the estimated high-frequency component. The method combines the upsampled input image and the estimated high-frequency component to obtain the restored image.

Recently, the Freeman's method was modified in [31] for face-image SR taking the correspondence of facial parts into account, which uses facial-parts compatibility function, $\lambda(.)$. The underlying idea is to select patches in the database according to their facial parts. In order to increase the probability for the patches of the corresponding facial part to be selected, the face-images were normalized and the distance from the original position of the selected patch was utilized in the $\lambda(.)$. Fig. 3 illustrates a learning model where patches' positions are taken into consideration. The facial-parts compatibility function, $\lambda(.)$ is defined as

$$\lambda[X(i,j)] = \exp\left[-\frac{L}{2\sigma_3}\right] \tag{4}$$

where L is the Euclidean distance from the original position of the patch $X(i,j)$ in the training face image to the position (i,j) in the restored face image and σ_3 is a constant parameter.

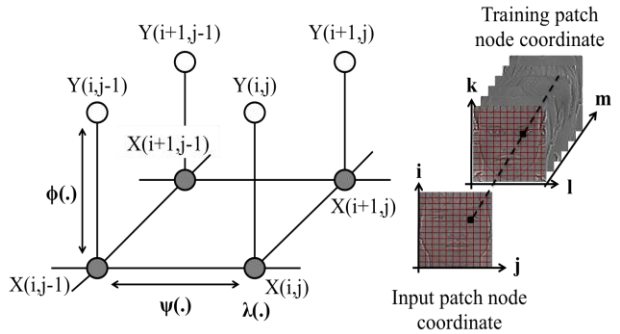


Figure 3. Learning-model including facial-parts compatibility function $\lambda(.)$

The joint probability of $X(.)$ under the condition $Y(.)$ is an extension of (1), which is defined as

$$P(X|Y) = \prod_{ij} \phi[X(i,j), Y(i,j)] \times \prod_{ij,(u,v)\in NB(i,j)} \psi[X(i,j), X(u,v)] \times \prod_{ij} \lambda[X(i,j)] \tag{5}$$

Patch candidates for each $Y(i,j)$ are selected based on $\phi[X(i,j), Y(i,j)]$ and $\lambda[X(i,j)]$ with a given weighting factor to adjust preference. The remaining part of the method is similar to the Freeman's method.

III. Proposed Method

In example-based tradition, the training set should only include images that are similar in the type of input image for better probability of similarity among textures. The key idea is to categorize training patches based on facial parts, i.e., as a database of eye patches and a database of nose patches. Since the MRF model already includes $\lambda(.)$, it is unnecessary to distribute the patches manually.

We start from a collection of normalized high-resolution face images to construct a database. The normalized training images have the same size and ratio of facial features, where facial-feature points (e.g., eye, nose, mouth, chin, and face boundary lines) in each image are at approximately closer positions. We degrade each image by applying JPEG compression to them with different quality rates, creating several sets of degraded images. The main characteristic of the database is each patch's original position in the training images, as shown in Fig. 4, where (k,l) is training-patch-position coordinate, to be used later in the proposed MRF model function during the super-resolution phase.

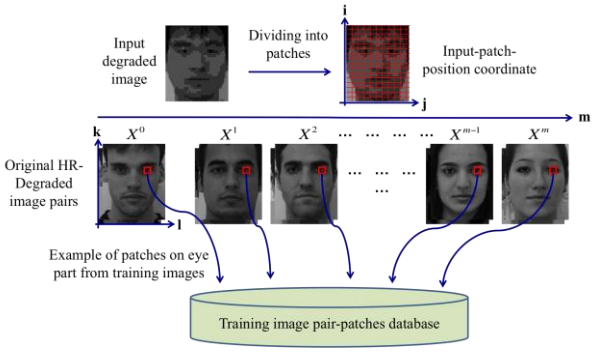


Figure 4. Patch-coordinate-based database construction

A. Database Construction

The training images in the database need to be pre-processed to extract high-frequency information so that only the texture are observed. There are several methods of texture extraction in image processing that have led to the proposal of several example-based SR methods, as mentioned in the introduction.

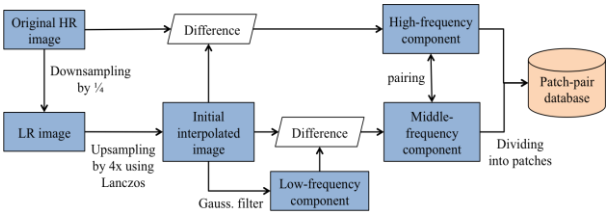


Figure 5. Previous SR method's database construction

Fig. 5 shows an SR method's database-construction process that applies a Gaussian filter as the PSF onto the initial interpolated LR image to extract texture components, while texture components in the original HR image are extracted from the difference in the interpolated image. Both original HR and LR images' texture components are then divided into patch pairs, where patches are typically 5x5 or 7x7 pixels in size. A hundred images are pre-processed through the same

procedure to create a large number of patch pairs in the database.

In this study, our targets were different from those of previous studies. We proposed a straight-forward method, named as Direct method, as outlined in Fig. 6, which directly takes the difference between the original HR images and degraded images as high-frequency components for original HR image. While for degraded-images' texture extraction, we apply a Gaussian filter on the degraded images to obtain their low-frequency components. Then we subtract low-frequency components from the degraded image to obtain the middle-frequency components. Both high-frequency component and middle-frequency component are stored in pairs in the database.

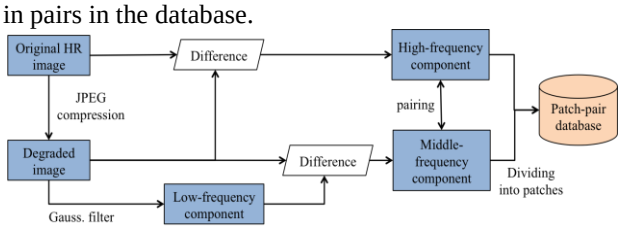


Figure 6. Direct method's database construction

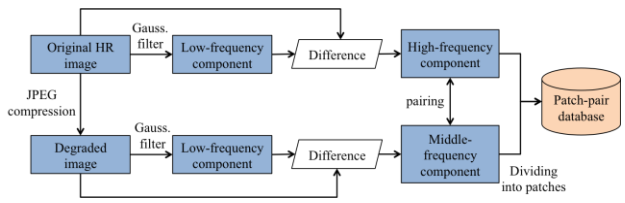


Figure 7. Smooth method's database construction

Another proposed method named as Smooth Method, uses a different extraction process for original HR images for independently extracting high-frequency components from original HR images by applying a Gaussian filter into itselves. We smooth out the original HR images in advance using the Gaussian filter to obtain low-frequency components, then we subtract it from the original HR images to obtain their textures (high-frequency component), as outlined in Fig. 7. For the degraded images, we use the same process of extraction as for the Direct method. Both texture components are then divided into patches in which both patches correspond to each other and store them in the database.

B. Patch-candidate Selection

To reduce computational cost, a fixed amount of patch candidates for each node is selected in advance so that we do not have to consider thousands of patches available in the database iteratively during the iteration process. Several patch candidates that had the most similar texture according to their $\phi(.)$, and closest facial parts according to $\lambda(.)$ with the input patches are chosen beforehand.

To adjust the preference between $\phi(.)$ and $\lambda(.)$, we apply a weighting factor α as follows.

$$P_C(X \mid Y) = \prod_{ij} \phi\left[X(i,j), Y(i,j)\right]^{\alpha} \times \prod_{ij} \lambda\left[X(i,j)\right]^{(1-\alpha)} \quad (6)$$

where α is between zero and one.

When α is one, the function does not take into account $\lambda(.)$, which means this is the same as that with Freeman et al.'s method. If α is zero, it only takes into account $\lambda(.)$ and ignores $\phi(.)$. A set of patch candidates that have maximum likelihood are selected when α is 0.0, 0.1, 0.2, ..., and 1.0 to find the best output result.

The number of patch candidates proportional with the number of training images in the database, where there is 1 patch candidate per training image, e.g., if we used 100 training images, 100 patches among the total number of patches would be chosen as candidates.

C. Image Restoration Phase

The main objective for this study was to restore a degraded (block noise) image into a clear HR image. We adopt the same process as with example-based methods, but we use original and degraded image pairs as training images and a degraded input sample as the target.

The MRF network in Fig. 3 probabilistically models the relation between the degraded input image patches and degraded HR training image-pair patches in terms of $\phi(.)$ and $\lambda(.)$, and between the four neighboring estimated HR patches in terms of $\psi(.)$. We have to find a set of patch candidates for each input node according to $\phi(.)$ and $\lambda(.)$ with α previously. Then, the only element left to be considered is which patch amongst the candidates is the best neighbor.

The iteration process involves a stitching algorithm that iteratively infers a set of best neighboring patches that have the most compatible $\psi(.)$ values on the overlap region between nodes, as illustrated in Fig. 2.

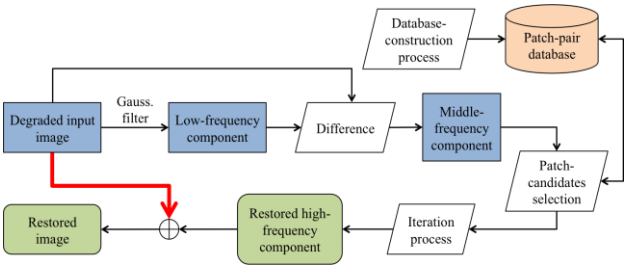


Figure 8. Restoration algorithms for Direct method

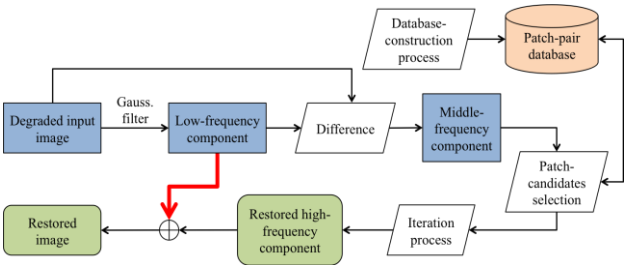


Figure 9. Restoration algorithms for Smooth method

Fig. 8 and Fig. 9 outline the overall image-restoration algorithms for our Direct and Smooth methods, respectively. The texture components from the degraded input image are extracted through the same process as in the database, which is using a Gaussian filter. We then find a set of patch candidates for each input patch node for the iteration process.

Before iteration, we create an initial estimated image as the base image by merging the most similar patch candidates (patches with the lowest $\phi(.)$ and $\lambda(.)$) of all nodes. This image would looks uneven and unnatural since the chosen patches would not yet be compatible with their neighboring patches. Local patches alone are not sufficient to estimate plausible looking HR images [17]. Hence, we applied the iteration process to reduce the incompatible effects in the image.

Theoretically, the aim with the iteration process is to find a set of $X(.)$, that will improve the joint probability, $P(X|Y)$ in (5) as best as possible. By only using a limited number of selected candidates to determine the $P(X|Y)$, the processing time can be reduced. We find the best $P(X|Y)$ to get the best neighboring patches by replacing $X(.)$ with the selected patch-candidates alternately. The $P_C(X|Y)$ in (7) is the known probability value from (6), which is different for each patch candidate.

$$P(X|Y) = P_C(X|Y) \times \prod_{ij,(u,v)\in NB(i,j)} \psi[X(i,j), X(u,v)] \quad (7)$$

We replace the initial chosen patches on every node with the best patches among patch candidates that have highest compatibility with their neighboring nodes. The first iteration is done when all nodes have been processed. The image resulting from the first iteration is better than that from the initial image model. We carry out the same procedure iteratively until no improvement of $P(X|Y)$ occurred.

Since the extraction process during database construction is different for both methods, the combined components for the restored output image are also different. For the Direct method, we combine the estimated restored high-frequency component image directly with the degraded input image, while with the Smooth method, we combine them with the low-frequency component from the degraded input image to obtain a final restored HR image.

IV. Results and Discussion

We used a set of normalized face images taken from 100 subjects for the database. All the original HR face images used in the experiment were 288x240 in size. We applied block noise to the original HR images by using a JPEG compression tool with different quality rates between 1 to 13%, resulting in compression ratios from 134:1 to 111:1, to generate six sets of degraded samples. Our main target was the lowest quality, i.e., 1% quality-rated degraded samples. The JPEG block sized in 8x8 pixels yielded similar noise to simple 8x magnification from 1/8 low-resolution images but further degraded images because of mosquito and quantization noise.

We evaluated our methods on a set of 30 lowest-quality face images to determine their effectiveness. We ran the iteration process 10 times under different α (0.0, 0.1, …, and 1.0) and Gaussian parameter σ values (4 to 12) to find the best resulting images.

We used PSNRs to evaluate the resulting images and compared them to the original ones (ideal result) to assess the performance of our methods. The mean value of all pixels in the images was used as a quality index. Thus, higher PSNRs indicate better quality. Fig. 10 presents three examples of the lowest quality rate (1%) of JPEG-degraded face samples.

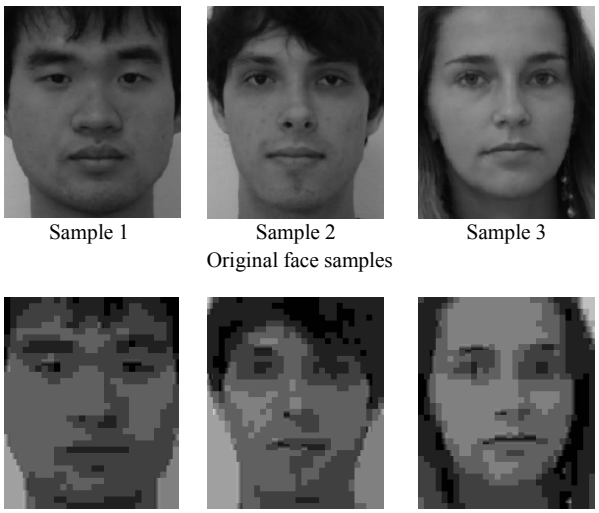


Figure 10. Original and degraded test images

A. Fix σ and α Value

It is necesssary to fix the σ and α. To find the best σ for the Direct and Smooth methods, we ran the restoration process on 30 input images of lowest quality samples under different σ to find their average PSNRs.

Fig. 11 shows the average PSNRs for 30 samples: input images, Gaussian-filtered images, and best output images from our methods (when best α was applied) for the Direct and Smooth methods for different σ. Gaussian-filtered blurred images and our methods' output images effectively exhibited higher average PSNRs than the input images. However, the higher the σ, the lower the PSNRs value of the Gaussian-filtered images. We can observe that PSNRs fell drastically when σ was more than 5. Conversely, those for the output images from both of our methods increased gradually.

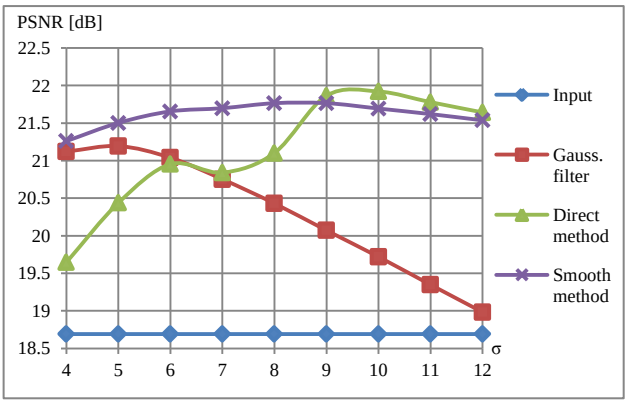


Figure 11. Average PSNRs of 30 samples: input image, Gaussian-filtered image, and Direct and Smooth methods' output images.

Even the Gaussian filter has significantly reduced noise, consequently the filtered image losses the important high-frequency texture component. The image becomes too blurry when σ was 12. Conversely, the output images with our methods' exhibited higher PSNRs since they also included predicted high-frequency components.

Table I lists those average PSNRs for each σ. According to Table I, the best σ for the Gaussian-filtering method was 5, for the Direct method was 10, and for the Smooth method was 8. So, we fixed the σ for each method onwards.

TABLE I. AVERAGE PSNRS OF 30 FACE IMAGES: INPUT IMAGE, GAUSSIAN-FILTERED IMAGE, AND PROPOSED METHODS' OUTPUT IMAGES

Gaussian parameter, σ	PSNR [dB]			
	Input	Gaussian-filtered	Direct method	Smooth method
4	18.690	21.121	19.646	21.260
5	18.690	**21.193**	20.439	21.503
6	18.690	21.039	20.954	21.656
7	18.690	20.752	20.840	21.698
8	18.690	20.430	21.102	**21.765**
9	18.690	20.071	21.864	21.765
10	18.690	19.719	**21.921**	21.694
11	18.690	19.348	21.779	21.620
12	18.690	18.983	21.642	21.537

We then found the average α from a set of best α values when σ was 10 for the Direct method and 8 for the Smooth method, for 30 output samples to have a fixed α. From the set of the 30 best output images from assessment in Fig. 11, where α is variable from 0.0 to 1.0, the average α was 0.4 for both the Direct and Smooth methods.

According to (6), when α was set to 0.4, the MRF function would be weighted towards $\lambda(.)$ rather than $\phi(.)$, i.e., 60% $\lambda(.)$ and 40% $\phi(.)$ during patch-candidate selection. Hence, the learning model including $\lambda(.)$, as shown in Fig. 3 [31], performed well in terms of the proposed image-restoration process.

Fig. 12 presents resulting images for the lowest quality of 1% for Sample 1. Fig. 12(a) is the original HR image and Fig. 12(b) is its degraded input image. The results obtained from the Gaussian-filtering, Direct, and Smooth methods are shown in Fig. 12(c), 12(d), and 12(e), respectively. The boxed images on the right side of each image are the zoomed versions of the marked regions in the images, where the details of the results can be observed. The Gaussian-filtering method effectively reduced block noise in the image; however, most of the face details are blurred, as shown in Fig. 12(c), especially along the edges. The restored images with our methods in Fig. 12(d) and 12(e) are significantly sharper than the Gaussian-filtered output image. Our methods effectively restored the texture in images.

B. Comparison between Direct and Smooth Methods

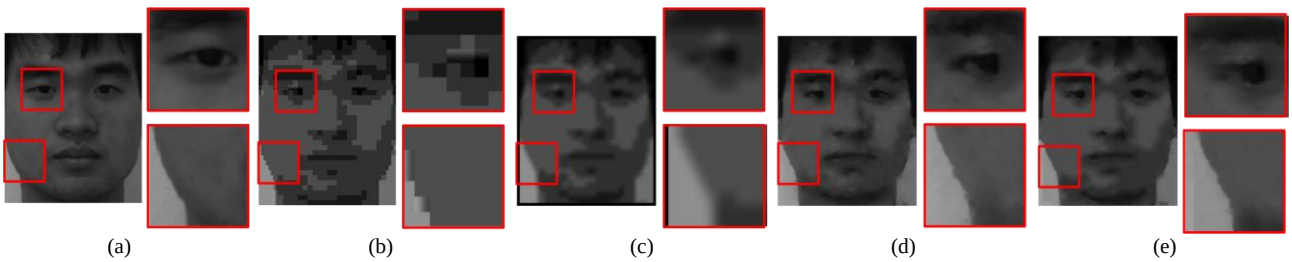
We compared the resulting PSNRs amongst the output images from our Direct and Smooth methods for the 30 samples. We found that outputs for Sample 2 and 3 had the largest difference in PSNRs when the Direct method outperformed the Smooth method and vice versa. Table II shows the different PSNRs for the two samples.

TABLE II. PSNRS FOR OUTPUT SAMPLE 2 AND 3 FROM DIRECT AND SMOOTH METHODS

Method	Samples	
	2	3
Direct	22.850	22.561
Smooth	22.153	22.844

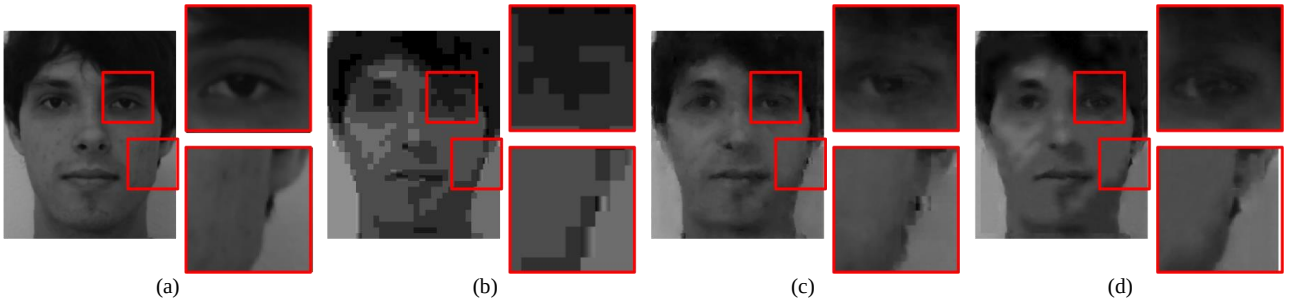
For Sample 2, the Direct method outperformed the Smooth method with a difference of 0.697 dB. For Sample 3, the Smooth method outperformed the Direct method with a difference of 0.283 dB. Fig. 13 and Fig. 14 show the details of the processed images for Samples 2 and 3, respectively, showing the entire face and parts of the (a) original, (b) degraded input, (c) output images with the Direct method, and (d) output images with the Smooth method.

Because the patches and input images were normalized from the positions of the eyes, the algorithms of our methods, which use the positions of facial parts, tend to use patches of eyes for the target eyes. Therefore, the eye parts were accurately restored. Regarding the PSNR evaluation, the Direct method outperformed the Smooth method due to the fact that the Direct method numerically reduced the block noise directly since patches for the Direct method are generated from the image including JPEG noise, while patches for the Smooth method are generated by eliminating the effect of JPEG noise.



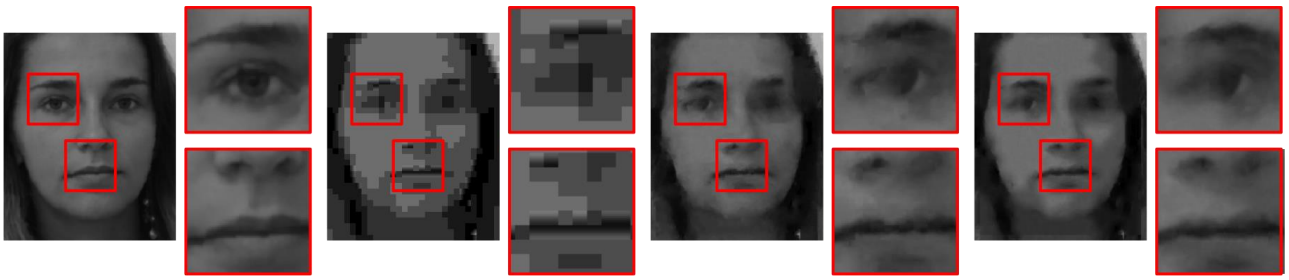
(a) (b) (c) (d) (e)

Figure 12. Resulting images for Sample 1 where σ and α were fixed: (a) original face image; (b) degraded input image; (c) Gaussian-filtered image (σ=5); (d) Direct method's output image (σ=10, α=0.4); (e) Smooth method's output image (σ=8, α=0.4).



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

Figure 13. Resulting images for best sample from Direct method compared to smoothed method: (a) original face image; (b) degraded input image; (c) Direct method's output image; (d) Smooth method's output image.



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

Figure 14. Resulting images for best sample from smoothed method compared to Direct method: (a) original face image; (b) degraded input image; (c) Direct method's output image; (d) Smooth method's output image.

However, the images generated with the Smooth method seemed to be more natural, and the quality was more stable for different σ. Therefore, the Smooth method can be used when natural appearing images are required.

C. Evaluation for Different Quality-Rate Samples

We demonstrated conventional Gaussian-filtering and our two methods on six sets of 30 samples, where the JPEG quality rate was 1, 3, 5, 8, 10, and 13% for each set for the sake of comparison. For the Gaussian-filtering method, we applied σ from 1 to 6 to find the highest average of output PSNR for each set. Table III shows average PSNRs of Gaussian-filtering method on different σ for the six sets of 30 samples.

TABLE III. AVERAGE PSNRS OF 30 SAMPLES FOR GAUSSIAN-FILTERED OUTPUT IMAGES UNDER DIFFERENT σ

Quality rate for set of 30 samples [%]	Gaussian filtering, σ					
	1	2	3	4	5	6
1	19.64	20.38	20.87	21.12	**21.19**	21.04
3	19.89	20.62	21.08	21.32	**21.38**	21.20
5	23.23	24.03	**24.31**	24.14	23.72	23.12
8	26.36	**26.95**	26.34	26.03	25.08	24.12
10	27.74	**28.15**	27.60	26.52	25.38	24.29
13	29.27	**29.39**	28.35	26.99	25.65	24.46

We fixed σ at 10 and 8 for the Direct and Smooth methods, respectively, regardless of the sample quality rate to standardize the results for the lowest-quality-rate samples. Table IV shows the average PSNRs for input images, best Gaussian-filtering output images (from Table III), and our methods' output images for different quality rates. Table V shows the improvement of PSNR, while Fig. 15 illustrates their graph.

TABLE IV. AVERAGE PSNRS OF 30 SAMPLES FOR INPUT, BEST GAUSSIAN-FILTERED, AND PROPOSED METHODS' OUTPUT IMAGES FOR DIFFERENT QUALITY RATES

Sample	JPEG quality rate [%]					
	1	3	5	8	10	13
Input	18.690	18.961	22.157	25.130	26.516	28.044
Gaussian-filtering method	21.193	21.376	24.308	26.948	28.151	29.392
Direct method	21.921	22.255	24.618	26.715	27.607	28.676
Smooth method	21.765	22.059	24.930	27.098	27.987	28.985

TABLE V. AVERAGE PSNRS' IMPROVEMENT FROM GAUSSIAN-FILTER AND PROPOSED METHODS FOR DIFFERENT QUALITY OF SAMPLES

PSNR improvement [dB]	JPEG quality rate [%]					
	1	3	5	8	10	13
Gaussian-filtering method	2.503	2.415	2.151	1.818	1.635	1.347
Direct method	3.231	3.294	2.461	1.585	1.091	0.632
Smooth method	3.075	3.098	2.773	1.968	1.471	0.941

From Fig. 15, our methods provided better improvement for images of lower quality rates. Direct method outperformed Smooth method and Gaussian-filtering method at a quality rate of 3% and below, while Smooth method gave best improvement at a quality rate between 5 to 8%. However, as the quality rate increased, i.e., higher than 8%, the conventional Gaussian-filtering method performed better for noise reduction. In other

words, our proposed methods perform well for low-quality images.

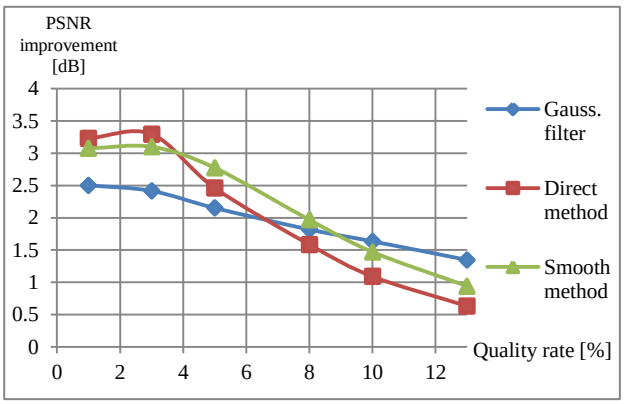


Figure 15. Average PSNR improvement of Gaussian-filtered and proposed methods' output images against quality rate of input images

V. Summary

We proposed an image-restoration process that involves the example-based method by constructing a patch database using a set of image pairs between original and degraded images. We used normalized training face images and models of the Markov random field that includes the facial-parts compatibility function. Generally, we divided training images (texture components) into patches and stored them into a database with information on their original positions on an image. Our process involves the Direct and Smooth methods that are appropriate to the block noise image. Then, we used the MRF to model patch structures for a restored image by considering facial-parts compatibility together with pixel-value similarity between patches. Finally, the missing high-frequency components due to the Gaussian-filtering process were inferred using a stitching algorithm iteratively. The experimental results obtained from face images demonstrated excellent performance of our methods in terms of PSNR in comparison with the conventional Gaussian-filtering method.

References

[1] M. Petrou and C. Petrou, *Image Processing: The Fundamentals*, Wiley, 2010.

[2] P. J. Shaw and D. J. Rawlins, "The point-spread functions of a confocal microscope: Its measurement and use in deconvolution of 3-D data," *Journal of Microscopy*, vol. 163, pp. 151–165, 1991.

[3] S. Hartung, "Image subtraction noise reduction using point spread function cross-correlation," arXiv e-prints 1301.1413, Jan. 2013.

[4] S. Al-ameri, N. Kalyankar, and S. Khamitkar, "Deblured Gaussian blurred images," *J. Comput.*, vol. 2, no. 4, pp. 33–35, 2010.

[5] G. M. P. V. Kempen and L. J. V. Vliet, "Improving the restoration of textured objects with prefiltering," in *Proc. 3rd Annual Conference of the Advanced School for Computing and Imaging*, 1997, pp. 174–179.

[6] G. Dougherty and Z. Kawaf, "The point spread function revisited: Image restoration using 2-D deconvolution," *Radiography*, vol. 7, no. 4, pp. 255–262, 2001.

[7] C. Dalitz, R. Pohle-Fröhlich, and T. Michalk, "Point spread functions and deconvolution of ultrasonic images," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 62, no. 3, March 2015.

[8] S. C. Park, M. K. Park, and M. G. Kang, "Super-Resolution image reconstruction: A technical overview," *IEEE Signal Processing Magazine*, vol. 20, pp. 21–36, May 2003.

[9] D. Capel and A. Zisserman, "Computer vision applied to super resolution," *IEEE Signal Processing Magazine*, vol. 20, pp. 75–86, May 2003.

[10] H. Ozdemir and B. Sankur, "Subjective evaluation of single-frame super-resolution algorithms," in *Proc. EUSIPCO*, Aug. 2009, pp. 1102–1106.

[11] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. on Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[12] H. He and W. C. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE Int. Conf. Comp. Vision and Pattern Recog.*, Jun. 2011, pp. 449–456.

[13] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, "SoftCuts: A soft edge smoothness prior for color image super-resolution," *IEEE Trans. on Im. Process.*, vol. 18, no. 5, pp. 969–981, May 2009.

[14] J. Sun, J. Sun, Z. Xu, and H. Y. Shum, "Gradient profile prior and its applications in image super-resolution and enhancement," *IEEE Trans. on Image Processing*, vol. 20, no. 6, pp. 1529–1542, Jun. 2011.

[15] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.

[16] Z. Lin and H. Y. Shum, "Fundamental limits of reconstruction-based super-resolution algorithms under local translation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, Jan. 2004.

[17] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comp. Graphics Appl.*, vol. 22, no. 2, pp. 56–65, 2002.

[18] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, Oct. 2000.

[19] W. T. Freeman and E. C. Pasztor, "Markov networks for super-resolution," in *Proc. 34th Ann. Conf. Info. Sciences and Systems*, Princeton Univ., 2000.

[20] S. Shuji, S. Takashi, and I. Akihiko, "Example-based super-resolution to achieve fine magnification of low-resolution images," *NEC Technical Journal*, vol. 7, no. 2, pp. 81–85, 2012.

[21] T. Nir and N. Karpel, "Example based learning of image stitching for an omni-directional camera using a variational optical flow methodology," in *Proc. SPIE 7000, Optical and Digital Image Processing*, April 2008.

[22] J. Y. Cui, *et al.*, "Example-Based image compression," presented at the 17th IEEE International Conference on Image Processing, 2010.

[23] K. Grauman, G. Shakhnarovich, and T. Darrell, "Virtual visual hulls: Example-Based 3D shape inference from silhouettes," in *Proc. 2nd Workshop on Statistical Methods in Video Processing*, Prague, Czech Republic, May 2004.

[24] B. Y. Koo, *et al.*, "Example-based statistical framework for parametric modeling of human body shapes," *Computers in Industry*, vol. 73, pp. 23–38, 2015.

[25] S. F. Lui, J. Y. Wu, and H. S. Mao, "Learning-based super-resolution system using single facial image and multi-resolution wavelet synthesis," *Lect. Notes Comp. Sci.*, vol. 4844, pp. 96–105, 2007.

[26] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. on Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[27] J. Yang, Z. Wang, Z. Lin, and S. Cohen, "Coupled dictionary training for image super-resolution," *IEEE Trans. on Image Process.*, vol. 21, no. 8, pp. 3467–3478, 2012.

[28] C. V. Jiji and S. Chaudhuri, "Single frame super-resolution using learned wavelet coefficients," *Int. Journal of Imaging Systems and Tech.*, vol. 14, no. 3, pp. 105–112, 2004.

[29] C. V. Jiji and S. Chaudhuri, "Single-frame images super-resolution through contourlet learning," *EURASIP Journal on Advances in Signal Processing*, pp. 1–11, 2006.

[30] W. Wu, Z. Liu, W. Gueaieb, and X. He, "Single-image super-resolution based on Markov Random Field and contourlet transform," *SPIE Journal of Electronic Imaging*, vol. 20, no. 2, p. 023005, 2011.

[31] S. Hamdan, Y. Fukumizu, T. Izumi, and H. Yamauchi, "Example-based face image super-resolution taking into consideration

correspondence of facial parts," *IEEJ Trans. on Electronics, Information and Systems*, vol. 12, no. 6, 2017.
[32] W. C. Siu and K. W. Hung, "Review of image interpolation and super-resolution," in *Proc. Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, 2012, pp. 1–10.

Suhail Hamdan is a doctoral student in science and engineering at the Ritsumeikan University, Japan. He received his B.E. from the Department of VLSI System Design at the Ritsumeikan University in 2011 and his M.E. in advanced electrical, electronic and computer systems from the same university in 2013. His current research interests include signal processing and image processing.

Yohei Fukumizu is presently an associate professor in Department of Electrical and Electronic Engineering since April 2013. He received his B.E. and M.E. degrees in computer and systems engineering from Kobe University, Kobe, Japan, in 2001 and 2003, respectively, and received the Ph.D. degree in computer engineering from Kobe University, Japan, in 2007. He joined in the Solutions Research Organization (SRO), the Integrated Research Institute (IRI), Tokyo Institute of Technology, Japan, as post-doctoral researcher in 2007, in medical and biotechnology project. From April 2008, he had been an assistant professor in Department of VLSI System Design, Ritsumeikan University, Japan. His research interests currently focus on intellectual signal processing systems that contribute to a Safe and Secure Society. He is also interested in design methodologies of communication systems. He is a member of IEEE, IEICE, IEE, IIEEJ, ITE, RISP, and JAFST.

Tomonori Izumi is presently a professor in Department of Electronic and Computer Engineering since 2016. Concurrently, he has also been a senior research scientist at Synthesis Corporation since 1998. He received his B.E. degree in computer engineering and M.E. and Ph.D. degrees in electrical and electronic engineering all from Tokyo Institute of Technology, Japan, in 1992, 1994, and 1998, respectively. From 1998 to 2005, he was a research associate at Department of Communications and Computer Engineering, Kyoto University, Japan. He joined Ritsumeikan University, Japan as an associate professor in 2005. His research interests include system, architecture, design and design methodologies of digital, especially reconfigurable hardware. He is a member of IEICE, IPSJ, ITE, RISP, IIEEJ, ISCIE and IEEE.

Hironori Yamauchi has been a Professor in the Faculty of Science and Engineering, Ritsumeikan University, Japan since 1996. He received his M.E. and Ph.D. degrees from the University of Tokyo in 1975 and 1994, respectively. In 1975, he joined the Electrical Communications Laboratories of Nippon Telephone and Telegraph Public Corporation. His research interests include pattern recognition, image signal processing, low power embedded systems architecture and related VLSI design. He is a fellow member of IEICE, and a member of IEEE, IPSJ, RISP and IIEEJ.