

A VQ-BASED DEMOSAICING BY SELF-SIMILARITY

Yoshikuni Nomura and Shree K. Nayar

Sony Corporation

Yoshikuni.Nomura@jp.sony.com

Department of Computer Science, Columbia University

nayar@cs.columbia.edu

ABSTRACT

In this paper, we propose a learning-based demosaicing and a restoration error detection. A Vector Quantization (VQ)-based method is utilized for learning. We take advantage of a self-similarity in an image for a codebook generation in VQ. The mosaic image is interpolated via a traditional method, and applied scaling, blurring, phase-shifting and resampling are used to create a training data for the codebook. The characteristics of the training data are similar to those of an ideal image. Using such training data and approximation of an ideal codevector by a locally linear embedding (LLE)- based method increases the probability of finding a suitable codevector from the codebook. Even if we cannot find a good codevector in an ill-conditioned case, the error detection finds poorly estimated pixel values and replaces them with better restoration results by another demosaicing method.

Index Terms— Vector quantization, Image reconstruction, Image resolution,

1. INTRODUCTION

Many methods have been proposed for Super-Resolution (SR) from a single low-resolution image. Recently, learning-based SR has become popular in this area [1, 2, 3, 4]. These methods assume a model (Markov network, polynomial function, LLE, etc...). These models are trained by training data. In general, learning-based methods do not have a reasonable criterion for the selection of training data. No one knows what kind of images and how many images are sufficient for learning. To address this problem, we propose to utilize self-similarity of images. Image self-similarity is well known for fractal image coding [5]. A natural image contains similar textures at different positions and scales. Such similarity is useful for image compression and inspires our method. In this paper, we describe Bayer demosaicing as an example of learning-based SR by self-similarity. Fig. 1 is a model of our idea. Whatever demosaicing method we use, it cannot restore high frequency components perfectly from a mosaic image (Fig.1

(a), (c)). In contrast, a low frequency component is restored well (Fig.1 (b), (d)). Therefore we can arrive at a similar texture for (a) from a scaled (d) (Fig. (a), (f)). Of course, an aliasing remains after scaling (Fig. 1 (e)). We use such a scaled image for a training data. If part of our training data contains the aliasing, a learning process removes it automatically, because the data is not ‘natural’ as in a natural image.

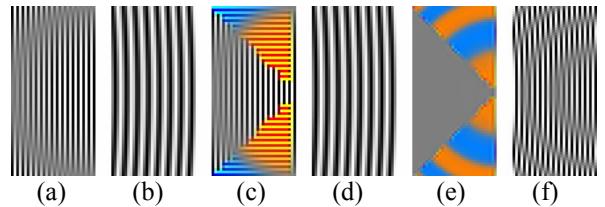


Fig. 1. Part of a CZP chart. (a) and (b) are an original image at Nyquist frequency (N_f) and $1/2 N_f$. (c) and (d) are restored images from a mosaic at N_f and $1/2 N_f$ by [6]. (e) and (f) are scaled-down (1/2) images of (c) and (d).

Our method uses a VQ-based learning method similar to [3, 4]. The training data is divided into small RGB patches and corresponding degraded mosaic patches are generated. To restore a given mosaic image, we divide it into small patches and compare them to those of the training data. We select similar mosaic patches from the training data and merge corresponding RGB patches to estimate RGB patches of the given mosaic image. Many learning-based SR methods apply a frequency band separation to their training data to compress a feature space. In the Bayer demosaicing case, such band separation is not practical. An optical low-pass filter used for a digital camera cannot sufficiently cut off a high frequency component, thus the captured mosaic image contains the aliasing and it makes the band separation difficult. Instead of band separation, the proposed method uses self-similarity to compress the feature space.

2. PROPOSED METHOD

Fig. 2 shows the framework of our algorithm, codebook generation, reconstruction, and error detection.

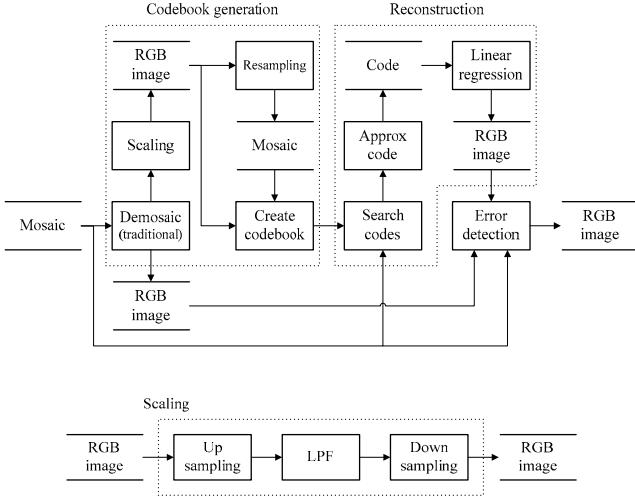


Fig. 2. The framework description of the proposed algorithm and detail of the scaling block.

2.1. Codebook generation

The codebook is generated from an interpolated image of a mosaic. For interpolation, any method can be used and we choose an adaptive color plane interpolation proposed by Hamilton and Adams [6]. The interpolated image is converted to various scaled, blurred and phase-shifted images at the scaling block in Fig. 2. These images are used for training data. The scaling block is a typical decimator. We control cutoff frequency at LPF. We also change sampling frequency and sampling phase at down sampler. The training data is resampled to create a mosaic image. In Bayer CFA case, there are four possible sampling patterns. Resampling block generates all possible patterns to increases phase-shift variation. The codevector in the codebook contains a small patch of training data (RGB patch \mathbf{c}) and the corresponding mosaic image (mosaic patch \mathbf{m}). \mathbf{c} and \mathbf{m} are a vector representation of patches. (\mathbf{c}, \mathbf{m}) denotes the codevector. The codebook must be divided for every CFA pattern of mosaic patch. A dense codebook is desirable to increase the probability of finding a close codevector. On the other hand, such a codebook takes a long time to search. Therefore, we propose that the given mosaic image should be divided into small regions. This is based on the assumption that similar texture distributes locally in the image. Every region has a 1/2 overlap with both adjacent regions. In addition, we use a tree-structured vector quantization (TSVQ) [7]. TSVQ creates a binary-tree-structured codebook. The codebook is easy to traverse and maintains sufficient search accuracy.

2.2. Reconstruction

The codebook generation step creates several codebooks which correspond with divided regions and CFA patterns. To estimate unknown color values at a certain pixel p in a given mosaic, a suitable codebook should be chosen. A local mosaic patch \mathbf{m}_L which includes p at a center position is compared to that of codevector, so the codebook must contain the same CFA pattern mosaic patch. In such codebooks, we chose the nearest codebook (Euclidian distance between p and the center of region is closest). We define a distance d between \mathbf{m}_L and mosaic patch in the codevector as L_2 norm (eq. 1).

$$d = \|\mathbf{m}_L - \mathbf{m}_C\| \quad (1)$$

where \mathbf{m}_C is the mosaic patch in codevector. Usually, there is no correct matching ($d = 0$) codevector and so we look for several close codevectors to approximate an ideal one. We apply locally linear embedding (LLE) based approximation [4, 8]. LLE computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs. LLE can recover a global nonlinear structure from locally linear fits. In our case, coefficients for a linear combination of mosaic patches in codevectors are computed to approximate \mathbf{m}_L , then these coefficients are used for the corresponding RGB patch estimation. The approximation is achieved by minimizing an error between \mathbf{m}_L and the linear combination of mosaic patches (eq. 2).

$$\arg \min_w \left\| \mathbf{m}_L - \sum_i w_i \mathbf{m}_{ci} \right\|^2 \quad \sum_i w_i = 1, \quad w_i \geq 0 \quad (2)$$

where \mathbf{m}_{ci} is a mosaic patch in i -th codevector. w_i is a coefficient for the linear combination of mosaic patches. w_i can compute by non negative least squares (NNLS) [9]. Ordinary least squares is not robust to outliers and sometimes computes large negative coefficients. This makes a bad RGB patch estimation, hence we introduce the constraint $w_i \geq 0$. RGB patch \mathbf{c}_L which correspond to \mathbf{m}_L is estimated as follows (eq. 3).

$$\mathbf{c}_L = \sum_i w_i \mathbf{c}_{ci} \quad (3)$$

where \mathbf{c}_{ci} is a RGB patch in the codevector. The center pixel in \mathbf{c}_L can be used as the estimated RGB pixel values at p . In addition to LLE-based method, we apply linear regression to refine the estimation result. The idea is the same as described in next section. A regression line is calculated from \mathbf{c}_L . The unknown color values at p are estimated by the known color value at p and the regression line.

2.3. Error detection

The existence of strong correlation between different color channels is well known. Fig. 3(a) shows a correlation between R and G channels at a tree region of campus image in Fig. 5. Every pixel value is normalized as [0,1] in this paper. In Fig. 3(a), a correlation coefficient is 0.994. The distribution of pixels at a local patch can be well approximated by a linear regression line (eq. 4).

$$C2 = \frac{Cov_{C1,C2}}{Var_{C1}}(C1 - M_{C1}) + M_{C2} \quad (4)$$

where $C1, C2$ are one of RGB color values. Var_{C1} is a variance of $C1$. $Cov_{C1,C2}$ is a covariance of $C1$ and $C2$.

M_{C1}, M_{C2} are a mean of $C1$ and $C2$. The equation means, $C2$ can be estimated from $C1$ and statistics of $C1$ and $C2$. Fig. 3(b) shows an error of estimated pixel value by linear regression. As a whole, error is less than 1% of pixel value range, therefore, we can trust the estimation result. We take advantage of this characteristic for the restoration error detection. The process estimates the mosaic image from the restored image in reverse. For example, the color at a certain pixel in the mosaic is red, we estimate the pixel value from green and blue pixel values at the same pixel of the restored image. If the restored image is not ideal, the reconstructed mosaic image is different from the original one. The restoration error ε_r is defined as follows (eq. 5).

$$\varepsilon_r = (C1_m - C1_{C2})^2 + (C1_m - C1_{C3})^2 \quad (5)$$

where $C1_m$ is one of RGB color values at a certain pixel in the mosaic. $C1_{C2}, C1_{C3}$ are estimated values of $C1$ by $C2$ and $C3$ in the restored image. If ε is close to 0, the restoration result has a high accuracy.

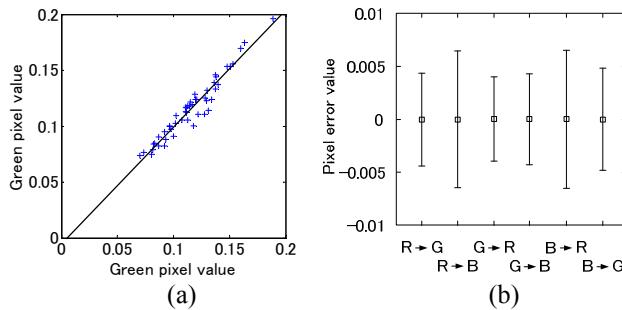


Fig. 3 (a) Correlation between R and G channels. (b) Error of estimated pixel value calculated by six images in Fig. 5. Error bar shows 2σ of error distribution. R → G means that green pixel value is estimated from red pixel value.

The proposed method makes use of two demosaicing methods. One is VQ-based method and the other is a traditional method. In both cases, restoration errors are calculated at every pixel and compared to each other. A simple way for combining two results is to select pixels that have smaller error. A binary mask can represent such selection. In our experiment, we applied blurring to the mask for a smooth blending (Fig. 7(d)).

3. EXPERIMENTAL RESULTS



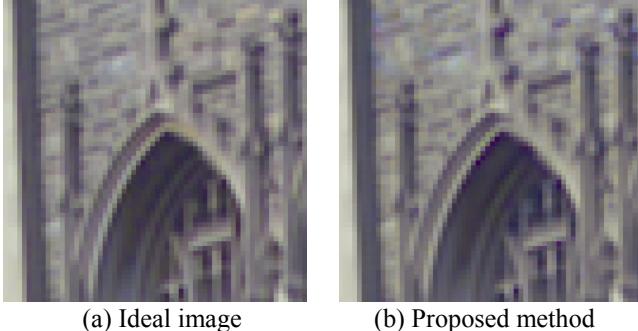
Fig. 5. Part of the test image set. From left, church, restaurant, campus, newspaper, Macbeth chart, bicycle.

We tested the proposed method on several images (Fig. 5). These images were captured by a SIGMA SD9 digital camera. The imager in the camera (FoveonX3) can capture the RGB image without any interpolation. We created mosaic images from these captured images.

The input mosaic was divided into 256x256 pixel regions for the codebook generation. We used a 3x3 patch for the codevector to make the codebook small and dense. At scaling block in codebook generation step, we created 180 variations of training data, 5 scaling variations ($1/2 - 1/4$), 4 phase-shifting variations (sampled at $0, \pi$ in the x-y direction), 9 blurring variations ($1/16N_f - 1/32N_f$ for 16 times upscaled image).

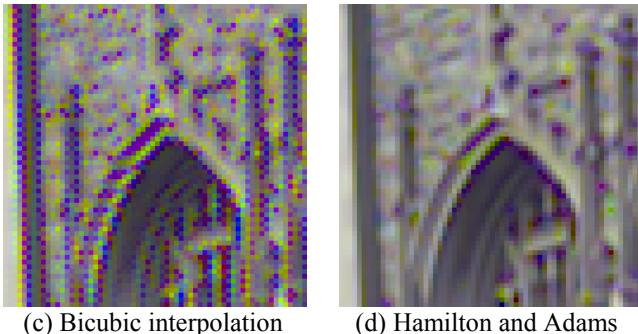
Fig. 6 shows the interpolation result of three different methods. Obviously, the proposed method achieves the best result. It can restore a high-resolution image and suppress a false color. Fig. 7 shows the effect of the error detection. The process blends two interpolation results, VQ-based result (c) and Hamilton and Adams result (d). The former does not have the false color, but it does have some artifacts around the edges, whereas the latter has the false color but a smooth edge. The blending result (b) and blend ratio (e) show that error detection selects better pixels from two results.

Fig. 8 summarizes the accuracy of demosaicing results. We use root mean square error between an estimated image and an ideal one for evaluation. This describes the efficacy of error detection. Comparison between VQ-based method (b) and One codebook (c) shows that the codebook division works well. Common codebook (d) does not make use of self-similarity. (d) uses 30 randomly selected images to generate the codebook. The result shows the usefulness of self-similarity for the codebook generation.



(a) Ideal image

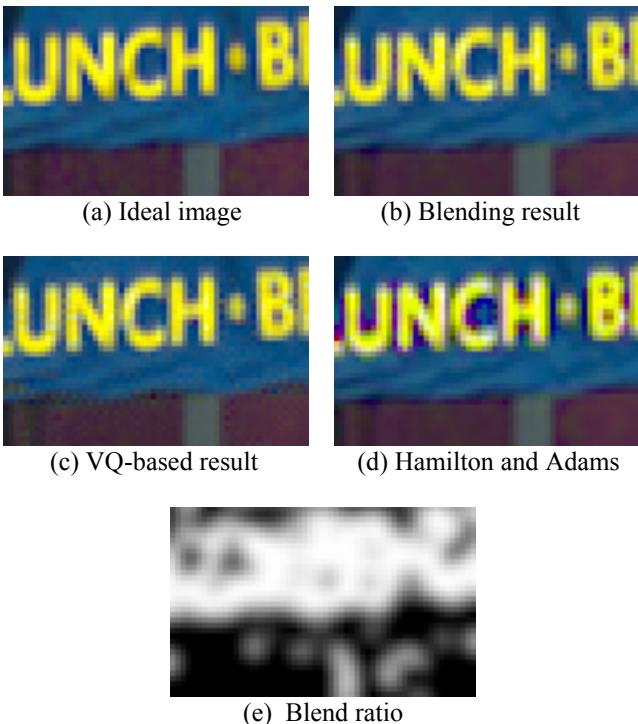
(b) Proposed method



(c) Bicubic interpolation

(d) Hamilton and Adams

Fig. 6. Interpolation results for three different methods. These images are a part of the church image in Fig. 5.



(a) Ideal image

(b) Blending result

(c) VQ-based result

(d) Hamilton and Adams

(e) Blend ratio

Fig. 7 Blending result. (e): Pixels in the white region comes from the VQ-based result and the other comes from Hamilton and Adams. These images are a part of the restaurant image in Fig. 5.

| RMSE (x10 ⁻³) | Church | | | Restaurant | | |
|------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | R | G | B | R | G | B |
| (a) Proposed method | 4.03 | 2.84 | 4.09 | 6.16 | 3.81 | 6.14 |
| (b) VQ-based method | 4.77 | 3.43 | 4.97 | 9.33 | 6.28 | 8.56 |
| (c) One codebook | 5.89 | 4.26 | 6.04 | 9.94 | 6.70 | 8.78 |
| (d) Common codebook | 9.67 | 7.45 | 11.7 | 18.7 | 12.9 | 17.5 |
| (e) Hamilton and Adams | 5.03 | 4.14 | 5.39 | 6.43 | 4.18 | 8.04 |
| (f) Bicubic | 17.9 | 9.20 | 17.1 | 19.7 | 9.22 | 16.2 |

Fig. 8. RMSE comparison. Only (a) uses error detection. (b) is the same as (a) except for error detection. (c) uses just one codebook generated by a whole region of input mosaic image. (d) uses a common codebook generated by 30 randomly selected images. (e) and (f) are the results of traditional methods.

4. CONCLUSION

We developed a VQ-based demosaicing method and an effective error detection method. Our proposed method utilizes the self-similarity in an image for the codebook generation in VQ. This idea improves codebook quality significantly. Furthermore, the error detection method removes an artifact that arises from the VQ-based method and yields a fine demosaicing result.

5. REFERENCES

- [1] W.T. Freeman, E.C. Pasztor and O.T. Carmichael, "Learning low-level vision," International Journal of Computer Vision, 40(1), pp.25-47, 2000
- [2] S.G. Narasimhan and S.K. Nayar, "Enhancing Resolution Along Multiple Imaging Dimensions Using Assorted Pixels," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.4, Apr 2005
- [3] R. Nakagaki and A.K. Katsaggelos, "A VQ-based Blind Image Restoration Algorithm," IEEE Transactions on Image Processing Vol. 12 No.9, pp.1044-1052, Sep. 2003
- [4] H. Chang, D. Yeung, and Y. Xiong, "Super-Resolution Through Neighbor Embedding," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004
- [5] M.F. Barnsley and L.P. Hurd "Fractal Image Compression," A.K. Peters,Ltd., 1993
- [6] J.F. Hamilton and J.E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," U.S. Patent No. 5,629,734, 1997
- [7] A. Gersho and R. M. Gray, "Vector Quantization and SignalCompression," Kluwer Academic Publishers, 1992
- [8] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, Vol.290, pp.2323–2326, 2000
- [9] C.L. Lawson and R.J. Hanson, "Solving Least Squares Problems," Prentice-Hall, New Jersey, 1974