# Color Image Compression Using Demosaicing and Optimized Color Spaces

Evgeny Gershikov

Department of Electrical Engineering, Ort Braude Academic College ofEngineering,Karmiel, Israel

and Department of Electrical Engineering, Technion – IIT, Haifa, Israel

eugeny11@braude.ac.il

***Abstract***

A new approach to image coding is presented. This method is based on recently introduced optimized color spaces for image demosaicing. These spaces can be used to transform the RGB color components to achieve desired properties of the new colors such as energy compactness or smoothness and thus to achieve better performance of the image reconstruction. In this work a new unified framework for color image compression is presented using demosaicing, where optimized color spaces are used both in the image coding and the demosaicing stages. A new coding algorithm based on the Discrete Wavelet Transform (DWT) is introduced and compared to presently available compression methods showing superior results both visually and quantitatively. It is concluded that the proposed unified framework and method of color space optimization are useful for storage and transmission of color images in band-limited information networks.

***Keywords***

*Color Image Compression; Optimized Color Spaces; Optimized Coding; Demosaicing; Discrete Wavelet Transform*

## Introduction

The high inter-color correlations present in most natural images (Kotera, H. and Kanamori, K. ,1990), (Limb J. O. and Rubinstein C. B., 1971), (Gershikov, E. and Porat, M., 2008) can be exploited for color image compression. Various methods have been proposed in order to reduce the amount of data that is actually coded, such as transformation ofthe RGB primaries to a new color space and then spatialtransformation ofthe new colorcomponents followed by a coding stage for them at different rates according to energy concentration or visual significance. Such a color space can be, for example, the YUV color space (Wallace, G. K., 1998), (Rabbani, M. and Joshi, R., 2002) or the KarhunenLoeve Transform (KLT) color space (Kouassi, R. K. et al., 2001). The new color components can be coded independently or using the remaining correlations (Shen, K. and Delp, E. J., 1997). Additional spatio-chromatic transforms can be applied to reduce the image data redundancy (Popovici, I. and Withers, W. D., 2005). Other approaches attempt to de-correlate the color components both spatially and chromatically at the same time (Leung, R. and Taubman, D., 2005), (Penna, B. et al., 2007) by using 3D transforming and coding, or utilize the inter-color correlations by choosing one of the components as the base and approximating the others as its function (Kotera, H. and Kanamori, K. ,1990), (Gershikov, E. et al., 2007).Here, however, we present a new image compression method based on image demosaicingas well asan efficient coding algorithm based on Rate-Distortion optimization (Gershikov, E. and Porat, M., 2007). The color processing of the proposed algorithm in the encoder and in the decoder has been optimized.

### *Image Demosaicing*

Many image acquisition devices are based on a single sensor using a color filter array (CFA), thus only partially sampled versions of the primary colors R, G, B are recorded. This is done in most cases according to the Bayer pattern (Bayer, B. E., 1976), as shown in Fig. 1. In this case, the green has twice as much samples as the red and the blue, making the green interpolation easier to be accomplished due to reduced potential of aliasing (Gunturk, B. et al., 2008). Then the red and the blue components can be reconstructed based on inter-color correlations which are usually high in natural images (Yamaguchi, H., 1984), (Roterman, Y. and Porat, M., 2007).Straightforward algorithms for demosaicing, such as bilinear or bicubic interpolation methods, however, do not use these inter-color correlations and operate on each color component independently. Better performance is achieved by algorithms that are based on the sequential scenario of the reconstruction of G first, followed by the reconstruction of R and B, e.g., (Hamilton, J. F. and

Adams, J. E., 1997), (Gunturk, B. K. et al., 2002), (Zhang, L. and Wu, X., 2005), (Chung, K.-H. and Chan, Y.-H., 2006), (Paliy, D. et al., 2007) and (Sher R. and Porat M., 2007). In such algorithms, the inter-color correlations are usually exploited by interpolating the differences $R-G$ and $B-G$.

However, since no optimization is performed, it can be shown that using these differences is not the best method to perform the task efficiently. It is better to do the image interpolation in optimized color spaces (Gershikov, E. and Porat, M., 2009). Such a color space can be, for example, the one where the High Pass (HP) energy of the color components is minimized as described in the next subsection.

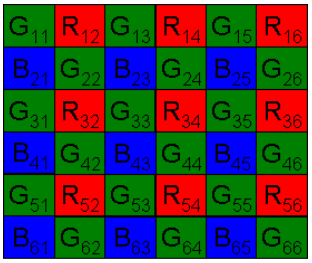



FIG. 1 THE BAYER CFA PATTERN

For the sake of completeness, it should be added that lots of effort has beenput into demosaicing research in recent years resulting in new techniques that are introduced every year. Non-sequential demosaicing methods have also been proposed, e.g. the iterative techniques of (Kimmel, R., 1999) or (Li., X., 2005) as well as vector CFA demosaicing (Gupta, M. R. and Chen, T., 2001). Most recent works can be found, for example, in (Fang, L. et al., 2012) and (Hu, C. et al., 2012).

***Minimal HP Energy Color Space***

A general color space for demosaicing after full reconstruction of the Green component can be written as the following relation between the new color components $C_1$, $C_2$, $C_3$ and the RGB primaries:

$$(1)\ C_1 = G, \quad C_2 = a_1 R + a_2 G, \quad C_3 = d_1 B + d_2 G.$$

The optimal coefficients $a_1$ $a_2$ $d_1$ and $d_2$ can be calculated based on different optimization criteria. For example, they can be found by minimizing the HP energy of $C_2$ and $C_3$, that is

$$\sum_i \sum_j \left(C_k^{HP_x}\right)_{ij}^2 + \sum_i \sum_j \left(C_k^{HP_y}\right)_{ij}^2, k = 2,3,$$

where $\left(C_k^{HP_x}\right)_{ij}$ is $C_k$ filtered by a horizontal high passfilter $HP_x$ at pixel $(i, j)$ of the image and similarly $C_k^{HP_y}$ is $C_k$ filtered by a vertical high pass filter $HP_y$ .Minimizing the component high pass energy results in higher smoothness of the image in the new color space and thus betterperformance of the demosaicing techniques is achieved. In fact the minimal HP color space is superior to other choices (Gershikov, E. and Porat, M., 2009). The optimal $a_1, a_2$ coefficients for this problem are

$$(2)\ a_1 = \frac{\alpha_{12} + \alpha_{22}}{\alpha_{11} + 2\alpha_{12} + \alpha_{22}}, \quad a_2 = -\frac{\alpha_{12} + \alpha_{11}}{\alpha_{11} + 2\alpha_{12} + \alpha_{22}},$$

where $\alpha_{11}$, $\alpha_{12}$ and $\alpha_{22}$ are calculated by applying the HP filters to R and G:

(3)

$$\alpha_{11} \triangleq \sum_i \sum_j \left[ \left(R^{HP_x}\right)_{ij}^2 + \left(R^{HP_y}\right)_{ij}^2 \right],$$

$$\alpha_{22} \triangleq \sum_i \sum_j \left[ \left(G^{HP_x}\right)_{ij}^2 + \left(G^{HP_y}\right)_{ij}^2 \right] \text{ and}$$

$$\alpha_{12} \triangleq \sum_i \sum_j \left[ \left(R^{HP_x}\right)_{ij} \left(G^{HP_x}\right)_{ij} + \left(R^{HP_y}\right)_{ij} \left(G^{HP_y}\right)_{ij} \right].$$

The solution to the $d_1$ and $d_2$ coefficients is the same as the solutionto $a_1$ and $a_2$, respectively, in (2) with B replacing R everywhere in (3). In this work $HP_x$ is the Sobel gradient filter given by

$$HP_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \text{and } HP_y = \left(HP_x\right)^T.$$

The structure of this work is as follows. The color image coding framework based on demosaicing is presented in the next section, where the stages of a compression algorithm based on it are discussed in detail as well. Simulation results for the proposed method are shown in Section "Compression Results" and compared to available methods. The last section provides summary and conclusions.

## Image Compression by Demosaicing

We present an application of an optimized demosaicing algorithm to color image coding. The idea is to create a Bayer pattern (Bayer, B. E., 1976) of a given color image and then to compress it. Coding this

single image instead of the full three color components has already reducedthe coded amount of bits significantly. The coding is performed considering the Bayer pattern (Fig. 1) as made of four components according to color: $RR$ for the red, $BB$ for the blue and $GR$ and $GB$ for the green (see Fig. 2). Each of these components is subband transformed and quantized followed by lossless post-processing. The reconstruction of the image is performed by decoding each of the four components and then running a demosaicing algorithm to reconstruct the full color image from the Bayer pattern. First the encoder and then the decoder are outlined.

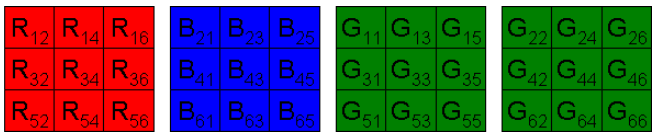



FIG. 2 THEBAYERPATTERN COMPONENTS: RR,BB, GR AND GB (FROM LEFT TO RIGHT).

### *The Encoding Algorithm*

As a first step, a Bayer pattern is created for a given image. This is done by sampling the image keeping only the pixels at the locations that are shown in Fig. 1. Then the compression method described next is applied to this pattern.

#### *1) The Compression Method*

The four channels $RR$, $BB$, $GR$ and $GB$ of the Bayer pattern (Fig. 2) are coded together using a Rate-Distortion model for subband transform coders (Gershikov, E. and Porat, M., 2007). The stages of the coding algorithm are given below.

1. Apply a color transform to the input channels to achieve better energy concentration. If we denote the channels at some pixel by $\mathbf{x} = [RRGRGBBB]^T$ and the color transform matrix by $\mathbf{M}$, then this stage is given by

$$\tilde{\mathbf{x}} = \mathbf{M}\mathbf{x}, \quad (4)$$

where $\tilde{\mathbf{x}} = [C_1 C_2 C_3 C_4]$ is the vector of the new color components at the same pixel. The DCT can be used as (Gershikov, E. and Porat, M., 2006) a color transform for the RGB color components and thus the three color componentsare suggested to be takenwhich correspond to the application of the following DCT matrix to the $BB, RR$ and $GB$ channels:

$$\mathbf{M}_{\mathbf{DCT}} = \begin{pmatrix} 0.333 & 0.333 & 0.333 \\ 0.500 & 0.000 & -0.500 \\ 0.250 & -0.500 & 0.250 \end{pmatrix}.$$

Note that this DCT matrix is normalized to $L_1$ norm of 1 for each row. The fourth color component can be taken simply as $GR - GB$. The resulting $\mathbf{M}$ (normalized and applied to $\mathbf{x} = [RRGRGBBB]^T$ is thus

$$\mathbf{M} = \begin{pmatrix} 0.333 & 0.000 & 0.333 & 0.333 \\ 0.000 & 0.000 & -0.500 & 0.500 \\ -0.500 & 0.000 & 0.250 & 0.250 \\ 0.000 & 0.500 & -0.500 & 0.000 \end{pmatrix}. \quad (5)$$

2. Apply the Discrete Wavelet Transform (DWT) to each of the new color components.
3. Quantize the DWT coefficients of each color component using quantization steps derived from optimal subband rate allocation (Gershikov, E. and Porat, M., 2007). The quantization steps are part of the output bit-stream.

Use a lossless post-quantization coding technique, such as in the Embedded Zerotree Wavelet (EZW (Shapiro, J. M., 1993)) algorithm to code the quantized DWT coefficients using the intra-subband and inter-subband correlations.

### *The Decoding Algorithm*

The decoder has to decompress the four colorchannels $RR$, $BB$, $GR$ and $GB$, to arrange them in a Bayer pattern and then a demosaicing algorithm is applied to it. The decompression technique and the proposed demosaicing method are depicted in the following subsections.

#### *1) Decompression*

To decode the four color channels, the following stages are performed.

1. Inverse post-quantization coding, corresponding to the one used in Step 4 of the compression method.
2. Inverse quantization of the DWT coefficients of the four color components.
3. Inverse DWT applied to the coefficients of each of the color channels.
4. Inverse color transform, which can be

described by

$$(6)\ \hat{\mathbf{x}} = \mathbf{M}^{-1}\hat{\hat{\mathbf{x}}}$$

where $\hat{\hat{\mathbf{x}}}$ and $\hat{\mathbf{x}}$ are the vectors of the reconstructed color components before and after the inverse color transform, respectively.

***2) Demosaicing***

Once the four colorchannels $RR$, $BB$, $GR$ and $GB$ have been decoded, they are arranged into a Bayer pattern and a demosaicing algorithm is performed. In this work a basic demosaicing algorithm consisting of the following stages has been selected:

1. The green color component is interpolated using edge preserving filtering (Hamilton, J. F. and Adams, J. E., 1997). Other more complex techniques can be used here for the reconstruction of the green, such as (Zhang, L. and Wu, X., 2005).

2. The interpolated green component $\hat{G}$ is used in thereconstruction of the red and blue colors. The linear combinations

$$(7)\ C_{RG} = a_1 R + a_2 \hat{G}, \quad C_{BG} = d_1 B + d_2 \hat{G}$$

are calculated at the known pixels of the red and the blue colors,respectively. Based on the results in Ref. 9, thedemosaicing algorithm used is optimized according to the minimal HP method(see Subsection "Minimal HP Energy Color Space"),which is proved to be best there. Then the red-green combination isinterpolated at the locations of the known blue samples, and theblue-green combination is interpolated at the locations of theknown red samples using a local polynomial approximation (LPA)filter (Paliy, D. et al., 2007).

3. The missing pixels in the red and blue - those at the locations ofthe known green pixels are reconstructed using bilinearinterpolation, resulting in full images $\hat{C}_{RG}$ and $\hat{C}_{BG}$.

4. The final red and blue components are calculated according to

$$(8)\ \hat{R} = \frac{\hat{C}_{RG} - a_2\hat{G}}{a_1} \text{ and } \hat{B} = \frac{\hat{C}_{BG} - d_2\hat{G}}{d_1}.$$

The reconstructed red, green and blue components of theimage $(\hat{R}, \hat{G}, \hat{B})$ are the output of thedecoder.

***Post-processing***

The result of the demosaicing algorithm can be refined using a post-processing method (Chang, L. and Tam, Y. P., 2004). In addition to that, the compression algorithm proposed in this work often results in "salt and pepper" type of noise in the reconstructed image. Thus median filtering can be applied in the smooth areas of the image.

## Compression Results

Here the performance of the proposed coding algorithmis compared to another DWT based method - the JPEG2000 standard(Rabbani, M. and Joshi, R., 2002), (JPEG 2000 Part I, 2000). We use the common objectivemeasure of PSNR (Peak Signal to Noise Ratio):

$$(9)\qquad PSNR \triangleq 10log_{10}\frac{255^2}{MSE},$$

where $MSE$ is the mean square error between the reconstructedimage $\hat{I}$ and the original one $I$. It is calculatedaccording to:

$$(10)\quad MSE \triangleq \frac{1}{3}\sum_{k\in\{R,G,B\}}\sum_i\sum_j\left(I_k(i,j) - \hat{I}_k(i,j)\right)^2.$$

$I_k(i,j)$ and $\hat{I}_k(i,j)$ here are the $k^{th}$ colorcomponents of $I$ and $\hat{I}$, respectively. The algorithms are also compared using the subjective PSPNR (Peak Signal toPerceptible Noise Ratio) measure, given by

$$(11)\qquad PSPNR \triangleq 10log_{10}\frac{255^2}{WMSE},$$

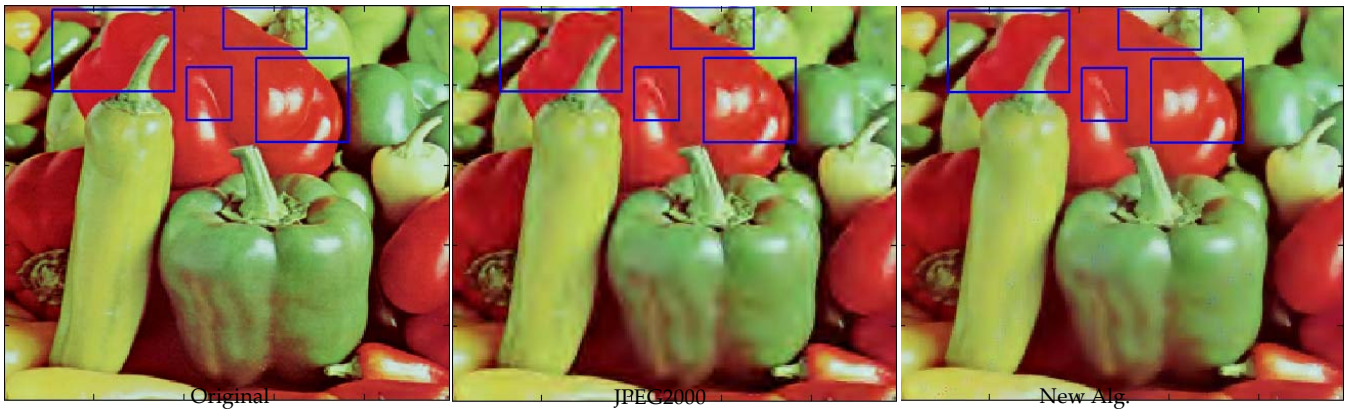
where $WMSE$ is the weighted mean square error of thereconstruction (different weights are assigned to differentfrequency bands). The results in terms of PSNR and PSPNR for thenew algorithm and JPEG2000 are summarized in Table 1 for the test images shown in Fig. 3. As it can be seen, the proposed method outperforms JPEG2000 for all the images witha gain of 1.65dB PSNR and 1.35dB PSPNR on average.

A visual comparison is given in Figs. 4 and 5. Once again the new algorithm issuperior. Note the color artifacts and blur introduced byJPEG2000, especially in the regions marked with a frame.

FIG. 3 THE COMPRESSION TEST IMAGES: LENA, PEPPERS,TREE, BABOON, FRUITS, CAT, TULIPS AND MONARCH.

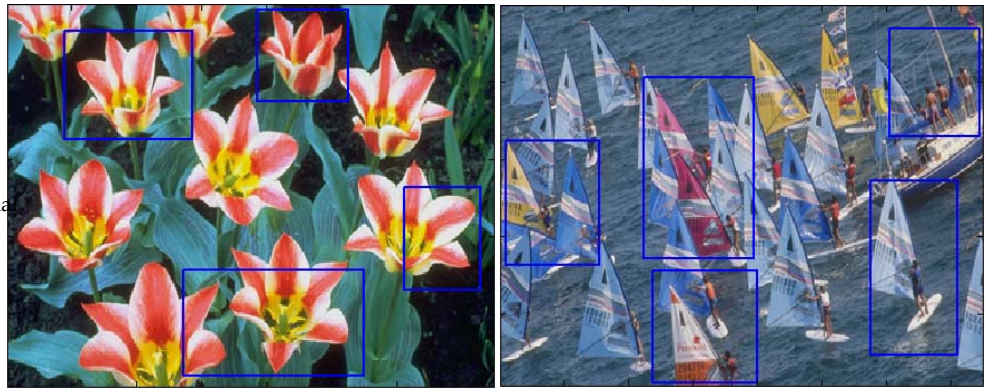
TABLE 1 PSNR AND PSPNR RESULTS FOR THE NEW ALGORITHM AND JPEG2000 AT THE SAME COMPRESSION RATE FOR THE TEST IMAGES.(BPP STANDS HERE FOR BIT PER PIXEL).

| Image | PSNR [dB] | | PSPNR [dB] | | Rate[bpp] |
|---|---|---|---|---|---|
| | New Alg. | JPEG2000 | New Alg. | JPEG2000 | |
| Lena | 28.12 | 26.63 | 36.14 | 34.88 | 0.25 |
| Peppers | 27.45 | 24.36 | 35.10 | 31.96 | 0.25 |
| Tree | 25.12 | 23.12 | 33.45 | 31.78 | 0.25 |
| Baboon | 23.30 | 21.90 | 32.93 | 31.77 | 0.25 |
| Fruits | 23.09 | 21.76 | 33.06 | 31.70 | 0.25 |
| Cat | 24.03 | 22.47 | 34.14 | 33.49 | 0.25 |
| Tulips | 24.28 | 22.15 | 32.57 | 30.58 | 0.25 |
| Monarch | 24.80 | 24.63 | 33.31 | 32.88 | 0.25 |
| Mean | 25.03 | 23.38 | 33.78 | 32.43 | |



Original JPEG2000 New Alg.

FIG. 4 COMPRESSION RESULTS FOR PEPPERS AT 0.63 BIT PER PIXEL (BPP): ORIGINAL, JPEG2000 AND THE NEW ALGORITHM.



Original

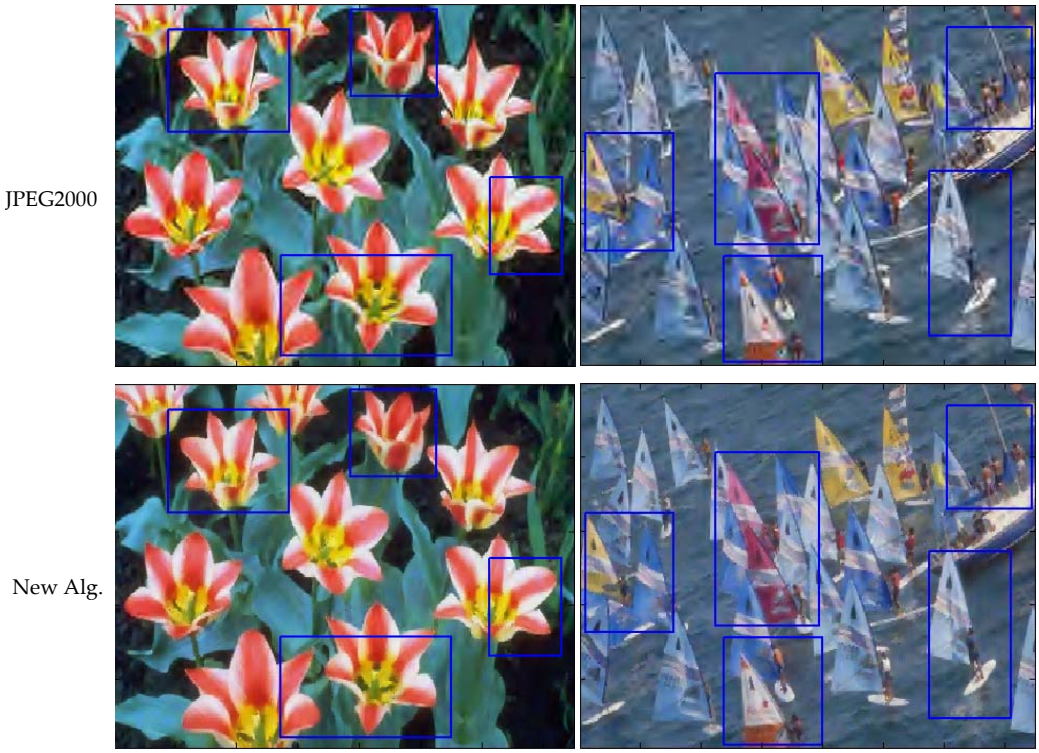



FIG. 5 COMPRESSION RESULTS FOR TULIPS AT 0.51BPP AND SAILS AT 0.37BPP (FROM TOP TO BOTTOM): ORIGINAL, JPEG2000 ANDTHE NEW ALGORITHM.

### *Summary and Conclusions*

A unified optimized framework of image compression usingdemosaicing is presented in this work. A color image is compressed by creating a Bayer pattern of it and then encoding it as four color channels following a color transform. The coding is based on a Rate-Distortion model (Gershikov, E. and Porat, M., 2007), from which optimal subband rate allocation is derived. The image is then decoded using a demosaicing algorithm operating in an optimized color space (minimal HP color space in this work). The comparison of the proposed compression algorithm to the JPEG2000 standard shows superior performance of our algorithm in terms of distortion measures aswell as visual quality. The simulations are performed at low rates too, which can be useful for transmission over slower communication channels. Our conclusion is that the proposed optimization framework for color images is useful for visual communication in band-limited information networks.

### ACKNOWLEDGMENT

The author would like to thank the administration of Ort Braude college and the Department of Electrical Engineering in Technion – IIT for providing the opportunity to conduct and publish this research.

### REFERENCES

Bayer, B. E. “Color imaging array”. U.S. Patent 3971065, July 1976.

Chang, L. and Tam, Y. P. “Effective use of spatial and spectral correlations for color filter array demosaicing”. IEEE Transactions on Consumer Electronics 50 (Feb. 2004): 355-365.

Chung, K.-H. and Chan, Y.-H. “Color demosaicing using variance of color differences”. IEEE Trans. on Image Processing 15 (2006): 2944-2955.

Fang, L., Au, O. C., Chen, Y., Katsaggelos, A.K., Wang, H. and Wen, X. “Joint demosaicing and subpixel-based down-sampling for Bayer images: a fast frequency-domain analysis approach”. IEEE Trans. on Multimedia 14 (Aug. 2012): 1359-1369.

Gershikov, E. and Porat, M. “A rate-distortion approach to optimal color image compression”, Proceedings of EUSIPCO 2006, Florence, Italy, September 2006.

Gershikov, E. and Porat, M. “On color transforms and bit allocation for optimal subband image compression”. Signal Processing: Image Communication 22 (2007): 1-18.

Gershikov, E., Lavi-Burlak, E. and Porat, M. “Correlation-based approach to color image compression”. Signal

Processing: Image Communication 22 (2007): 719--733.

Gershikov, E. and Porat, M. "Optimal color image compression using localized color component transforms". Proceedings of EUSIPCO 2008, Lausanne, Switzerland, August, 2008.

Gershikov, E. and Porat, M. "Image interpolation using optimized color transforms". Proc. of EUSIPCO, Glasgow, Scotland, August, 2009.

Gunturk, B. K. , Altunbasak, Y. and Mersereau, R. M. "Color plane interpolation using alternating projections", IEEE Trans. on Image Processing 11 (2002): 997-1013.

Gunturk, B., Li, X. and Zhang, L. "Image demosaicing: A systematic survey". Proc. of SPIE (2008): 68221J-68221J-15.

Gupta, M. R. and Chen, T. "Vector color filter array demosaicing". Proc. of SPIE, Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications II 4306 (2001): 374-382.

Hamilton, J. F. and Adams, J. E. "Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera". U.S. Patent 5629734, 1997.

Hu, C., Cheng, L. and Lu, Y. M., "Graph-based regularization for color image demosaicking", Proc. of IEEE ICIP2012, Orlando, Florida, Sep. 2012.

JPEG 2000 Part I: Final Draft International Standard (ISO/IECFDIS15444-1), NCITS ISO/IEC JTC1/SC29/WG1 N1855 (Aug. 2000).

Kimmel, R. "Demosaicing: image reconstruction from color ccd samples". IEEE Trans. on Image Processing 8 (1999): 1221-1228.

Kotera, H. and Kanamori, K. "A Novel Coding Algorithm for Representing Full Color Images by a Single Color Image." Imaging Technology 16 (Aug. 1990): 142-152.

Kouassi, R. K., Gouton, P. and Paindavoine, M. "Approximation of the Karhunen-Loeve transformation and its application to colour images". Signal Processing: Image Communication 16 (2001): 541-551.

Leung, R. and Taubman, D. "Transform and embedded coding techniques for maximum efficiency and random accessibility in 3-D scalable compression". IEEE Trans. on Image Processing 14 (Oct. 2005): 1632-1646.

Li., X. "Demosaicing by successive approximation". IEEE Trans. on Image Processing 14 (2005): 370-379.

Limb J. O. and Rubinstein C.B. "Statistical DependenceBetween Components of A Differentially Quantized Color Signal." IEEE Trans. on Communications Com-20 (Oct. 1971): 890-899.

Paliy, D., Katkovnik, V., Bilcu, R., Alenius, S. and Egiazarian, K. "Spatially adaptive color filter array interpolation for noiseless and noisy data". International Journal of Imaging Systems and Technology 17 (2007): 105-122.

Penna, B., Tillo, T., Magli, E. and Olmo, G. "Transform Coding Techniques for Lossy Hyperspectral Data Compression". IEEE Trans. Geoscience and Remote Sensing 45 (May 2007): 1408-1421.

Popovici, I. and Withers, W. D. "The eidochromatic transform for color-image coding". IEEE Transactions on Image Processing 14 (March 2005): 334-342.

Rabbani,M. and Joshi, R. "An overview of the JPEG 2000 still imagecompression standard". Signal Processing: Image Communication 17(2002): 3-48.

Roterman, Y. and Porat, M. "Color image coding using regional correlation of primary colors". Image and Vision Computing 25 (2007): 637-651.

Shapiro, J. M. "Embedded image coding using zerotrees of wavelet coefficients". IEEE Transactions on Signal Processing 41 (1993): 3345-3462.

Shen, K. and Delp, E. J."Color image compression using an embedded rate scalableapproach". Proceedings of IEEE ICIP (Oct. 1997): III-34-III-37.

Sher R. and Porat M. "CCD Image Demosaicing using Localized Correlations". Proc. of EUSIPCO, Poznan, Poland, September, 2007.

Wallace, G. K. "The JPEG still picture compression standard". IEEE Trans. Consumer Electronics 38 (1992): xviii-xxxiv.

Yamaguchi, H. "Efficient Encoding of Colored Pictures in R, G, BComponents", IEEE Trans. on Communications 32 (Nov.1984): 1201-1209.

Zhang, L. and Wu, X. "Color demosaicking via directional linear minimum meansquare-error estimation". IEEE Trans. on Image Processing14(2005): 2167-2178.

**Evgeny Gershikov** received his Ph.D. in ElectricalEngineering from Technion – Israel Institute of Technology in Haifa, Israel in 2010. His areas of interest are Signal and Image Processing, Color Processing and Vision, Computer Vision, Pattern Recognition and Speech Recognition.

# Eclipse Plugin Tool for Learning Programming Style of Java

Yuki Arakawa*[1], Masayuki Arai[2]

[1]Department of Human Information System, School of Science and Engineering, Teikyo University, Japan

[2] Graduate School of Science and Engineering, Teikyo University, Japan

[2]arai@ics.teikyo-u.ac.jp

***Abstract***

An Eclipse plugin tool contributive to learning the programming style of Java has been developed in this paper. The tool has the following functions: (1) recommendation on the use of CamelCase and English words for the names of classes, methods, and variables; (2) recommendation on setting the correct scope level of variables and the appropriate length of variable names; (3) recommendation on making comments in source programs; (4) showing sample source programs according to the programming style of Java.

***Keywords***

*Java; Learning Tool; Eclipse plugin; Programming Style; CamelCase; Scope*

## Introduction

Programming style is very important in the development of programs. Implementing a programming style makes source programs more readable and understandable, and can decrease the number of bugs in a program (D. D. Ward, 2004). Generally, novice programmers treat the programming style lightly. In spite of its importance, most universities and colleges do not teach the programming style to students in programming courses. Consequently, most students do not learn the programming style. To solve this problem, a tool contributive to learning the programming style of Java has been developed(M. Arai, 2012) and implemented as an Eclipse(Eclipse, 2012) plugin. In this paper, the Eclipse plugin tool to learn programming style of Java has been described.

Other learning systems have been proposed. For example, some learning systems detect ill-formed patterns in a program and diagnose the programming style (R. Sekimoto, 2000), while other learning systems detect the bugs or pitfalls of programming(M. Oda, 1994). These systems can detect programming codes that fail to follow the guidelines of the programming style registered in the systems in advance, and therefore users can check for improper programming style. However, the systems can only detect parts of a program, not a whole program.

In contrast, a system supporting module division using information on the active variables in a data flow analysis has been proposed to support a whole program (K. Suzuki, 2004). The system can support comments, names of variables, and whole programs written in the Pascal programming language.

In this paper, a tool has been proposed that has the following functions for novice Java programmers: supports indents and comments, recommends the use of CamelCase and English words for naming, and recommends setting the correct scope level of variables and the appropriate length of variable names. Many systems have the function of supporting indents (Eclipse, 2012) (mule, 2012) and comments (K. Suzuki, 2004). However, the other functions proposed above are new.

## Outline of the Tool

Fig. 1 depicts a view of the Eclipse implementing our tool, Fig 1(1) of which shows the editor by which users can write a program. Our proposed tool rewrites the program according to the proper Java programming style, which can be shown as fig. 1(2).

This section explains the outline of the tool using a sample program shown in fig. 2. The program fails to conform to the following guidelines for the programming style of Java:

(a)In the fourth line, the first letter of the class name “jugyou01kadai03” is not a capital letter.

(b)In the fourth line, the class name “jugyou01kadai03” is not an English word.

(c)In the eighth line, the variable “abc” does not have the correct scope.

(d)In the eighth line, the length of the variable name “abc” is too short.

(e)No comments are written in the program.

(f)No indents are inserted in the program.

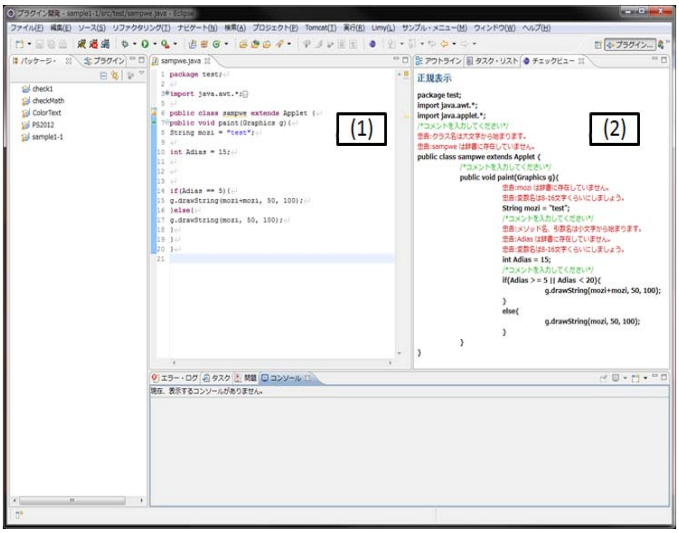



FIG. 1 A VIEW OF ECLIPSE IMPLEMENTED THE TOOL

Fig. 3 indicates the execution result after inputting the program shown in fig. 2 into the tooland shows the source program rewritten in the proper Java programming style as well as displays suggestions if the program has class names, method names and variable names thatis apart from conforming to the programming style. Users of the tool can learn the proper Java programming style by viewing the program as shown in fig. 3 and rewriting and inserting comments into the program in the editor area. Learners cannot edit, such as cut and paste, the program in fig. 3. The procedures are depitedto extract information and implement the functions mentioned above in the following sections.

## Extracting Information from the Source Program

The tool extracting information from the source program to implement the functions mentioned above, first performs a lexical analysis and then parses the source program. Finally, the tool extracts the following information:

(1)positions of declared classes, methods, and variables

(2)positions of used variables

(3)positions of instructions, including loops and conditional branches

(4)lines that have an improper indent

(5)lines that have multiple braces

```
1: import java.awt.*;
2: import java.applet.*;
3: import java.awt.event.*;
4: public class jugyou01kadai03 extends Applet
        implements ActionListener {
5: int xPosition = 50;
6: Button moveButton;
7: int numberOfClick = 0;
8: String abc="Executing Func() method";
9: public void init() {
10: moveButton = new Button("move");
11: moveButton.addActionListener(this);
12: add(moveButton);
13: }
14: public void paint(Graphics g) {
15: if (numberOfClick % 5 == 0)
16: xPosition = 50;
17: g.fillRect(xPosition, 50, 50, 50);
18: Func();
19: }
20: public void actionPerformed(ActionEvent e) {
21: numberOfClick++;
22: xPosition += 50;
23: repaint();
24: }
25: public void Func() {
26: System.out.println(abc);
27: }
28: }
```

FIG. 2 EXAMPLE OF A JAVA SOURCE PROGRAM

## Recommending the Use of Camelcase

CamelCase is one of Java’s naming guidelines for writing readable and understandable source programs. The following is required for the classes: if the name consists of several words, the first letter of the class name and the first letter of each word are in uppercase. If the first letter of a class name is in lowercase, the tool displays “Use a capital letter for the first letter of the class name,” as shown in fig. 3(a). In addition, the following is required for the methods and variables: if the name consists of several words, the first letter of the methods and variable names is in lowercase, and the first letter of each word is in uppercase. If the first

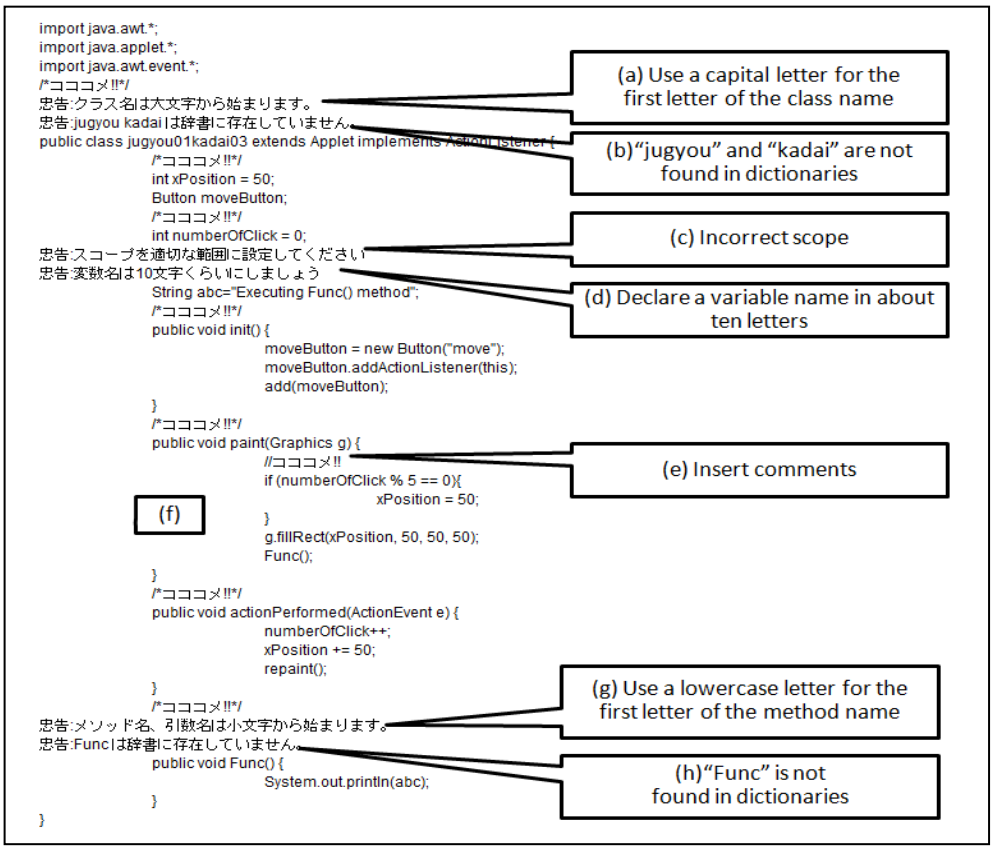



FIG. 3 EXAMPLE OF THE EXECUTION RESULT

letter of a method name is in uppercase, the tool displays "Use a lowercase letter for the first letter of the method name", as shown in fig. 3(g). The tool searches each class, method, and variable name for a uppercase letter and divides the name in front of the uppercase letter. For example, if the name of a class is "ClassName," the tool divides it into "Class" and "Name." Then, the tool searches two dictionaries—an English dictionary and an English abbreviation dictionary—for every word. If the words are not found in the dictionaries, it is likely that the class name does not follow CamelCase and thus, the tool displays "x is not found in dictionaries," as shown in fig. 3(b) and (h), where x is the original class, method, and variable name. The tool displays this message, for example, when the class name is "Classname". We will describe the method of searching dictionaries in the next section.

## Recommending the Use of English Words

In companies, computer systems are commonly developed by people from different countries. For this reason, the names of the variables, methods, and classes should be written in English so that everyone can understand them. In this section, we will describe the tool function to recommend the use of English words. If the name does not exist in English dictionaries, the tool displays "x is not found in dictionaries," as shown in fig. 3(b). The class name in "jugyou01kadai03" is shown as an example in fig 2. The two words—"jigyou" and "kadai"—do not exist ("jigyou" and "kadai" in Japanese mean "class" and "subject", respectively). Consequently, the tool displays "jigyou and kadai are not found in dictionaries."

We use Representational State Transfer (REST, 2012) to search English dictionaries. REST is a simple Web-based system with http and XML, and it can implement several functions using the Uniform Resource Locator (URL). Some websites, such as the EAST dictionary webservice (EASR, 2012), use REST to search English dictionaries. We insert a word into the URL to search English dictionaries. Fig. 4(a) and 4(b) show examples of the URL and XML data, respectively, received from the REST site in the case of searching for "camel." The line "<TotalHitCount>1</TotalHitCount>" in fig. 4(b) shows the number of hit counts in the dictionary. In

this case, the dictionary registers one "camel" record. If "<TotalHitCount>" is zero, the tool displays "x is not found in dictionaries."

In a similar manner, the tool can search for abbreviations in the abbreviation dictionary using REST.

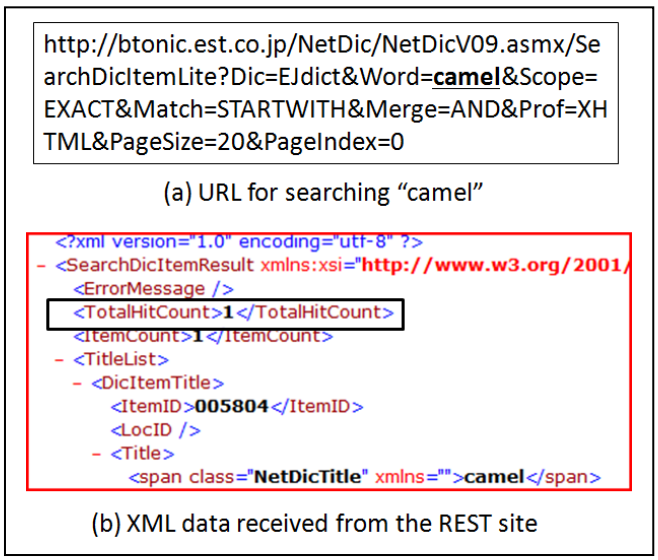



FIG. 4 EXAMPLE OF THE URL AND XML FOR SEARCHING "CAMEL"

## Recommending the Correct Scope Level Set up of Variables and the Appropriate Length of Variable Names

Correctly setting the scope level of variables enables easier debugging. Fig. 5 shows an example of the variable scope level. The variables—var1 and var2–are declared at the scope level n, as shown in fig. 5(a). The variable var1 is set at the correct scope level because it is used twice at the scope level n+1, as shown in Fig. 5(b) and (c). The variable var2, however, is used once, as shown in fig. 5(c). Therefore, variable var2 should be declared in area2. The tool can determine the incorrect scope level of variables, as shown in fig. 3(c).

Using the appropriate length of variable names makes source programs more readable and understandable. Gorla et al. stated that a variable name should consist of more than eight, but less than sixteen characters (N. Gorla, 1990). The tool can display "Declare a variable name in about ten letters" as shown in fig. 3(d). However, variable names often used by the loop counter and the array index, such as i, j, and k, are allowable.

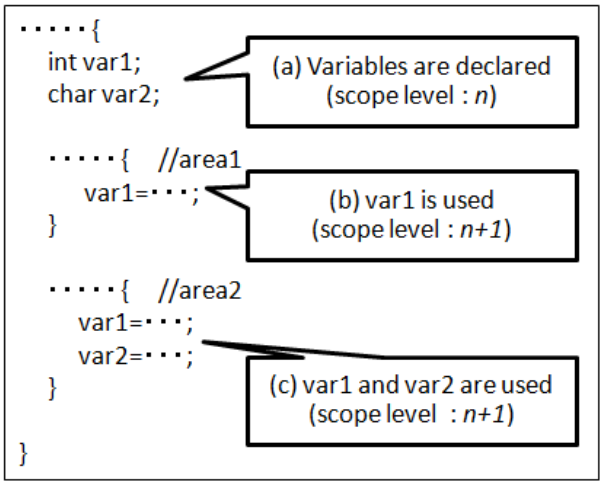



FIG. 5 EXAMPLE OF THE SCOPE LEVEL OF THE VARIABLES

## Additional Functions

Our proposed tool has the following additional functions:

### *Recommendation on Making Comments*

As mentioned above, the tool extracts the following positions from a source program: declared classes, methods, variables as well as instructions including loops and conditional branches. The tool displays "inserts comments" above these lines, as shown in fig. 3(e).

With this function, users can practice making comments such as for functions, objectives of classes, behaviors of methods, roles of variables, objectives of loops, and conditional branches.

### *Exception of File Saving Without Editing*

When the user saves the source program, the tool checks whether the user has edited the "insert comments" lines. If not, the tool prohibits saving of the source program.

### *The Display of a Sample Source Program Following the Programming Style of Java*

As mentioned above, the tool extracts lines that have improper indents and multiple braces from the source program and rewrites these lines according to the programming style. The corrected indent lines are shown in fig. 3(f).

## Conclusions

In this paper, we proposed an Eclipse plugin tool contributive to learning the proper programming style

of Java has been put forward. The tool has the following functions: recommendation on the use of CamelCase and English words for the names of classes, methods, and variables; recommendation on setting the correct scope level of variables and the appropriate length of variable names; recommendation on making comments in source programs; the showing of sample source programs according to the programming style of Java.

This tool is evaluated by practical use in actual classes.

## ACKNOWLEDGMENT

This study was supported in part by the Japan Society for the Promotion of Science, Grant Number (KAKENHI 24501150).

## REFERENCES

D. D. Ward and P. H. Jesty, "Guidelines for the use of the C language in critical systems," Motor Industry Research Association, 2004.

EAST dictionaries service http://www.btonic.com/ws/ (Dec. 21, 2012).

Eclipse http://www.eclipse.org/ (Dec. 21, 2012).

K. Suzuki, S. Yokoyama and Y. Miyadera, "Development and evaluation of automatic module division system for programming education," Technical reports of the Institute of Electronics, Information and Communication Engineers ET-2003(697), pp.143-148, 2004. (in Japanese).

M.Arai, "A Tool for Learning the Programming Style of Java", The 2012 International Conference on Software and Intelligent Information, S002, 2012.

M. Oda and T. Kakeshita, "Pitfall detection of C programs using pattern matching," Transactions of Information Processing Society of Japan 35(11), pp.2427-2436, 1994. (in Japanese).

mule http://www.m17n.org/mule/ (Dec. 21, 2012).

N. Gorla, A.C. Benander and B.A. Benander, "Debugging effort estimation using software metrics," IEEE Transactions on Software Engineering SE-16(2), pp.223-231, 1990.

REST http://ja.wikipedia.org/wiki/REST (Dec. 21, 2012).

R. Sekimoto and K. Kaijiri, "A diagnosis system of programming style," Transactions of Japanese Society for Information and Systems in Education 17(1), pp.21-29, 2000. (in Japanese).

**Yuki Arakawa** graduated from Department of Science and Engineering, Teikyo University, Japan in 2013. He is mainly engaged in developing programming learning tools.

**Masayuki Arai** is a professor in the Graduate School of Sciences and Engineering at Teikyo University. He received his B.E. degree from Tokyo University of Science in 1981 and Dr. Eng. degree from Utsunomiya University in 1995. His research interests include pattern recognition, natural language processing and information visualization. He is a member of the Information Processing Society of Japan and IEEE.

# Hand Shape Recognition Using 3D Active Appearance Models

Ryo Yamashita[1], Tetsuya Takiguchi[2], Yasuo Ariki[3]

Graduate School of System Informatics, Kobe University 1-1 Rokkodai, Nada, Kobe, 657-8501, Japan

[1]ryo@me.cs.scitec.kobe-u.ac.jp; [2]takigu@kobe-u.ac.jp; [3]ariki@kobe-u.ac.jp

***Abstract***

In this paper, a recognition method to discern complicated hand shapes has been proposed using 3D models as an interface for high-functionality TV. In such an interface, a user has to show his hand directly in front of the camera installed on the TV because it cannot recognize the hand shape when viewed in arbitrary directions. With this problem in mind, we have made it possible to track hand shapes in any direction by using 3D active appearance models (3D-AAMs). With the high-functional range image sensor Kinect, RGB images and depth-images of the targets can be obtained; by which hand shape models are constructed. Using multiple 3D AAMs, the robust recognition of such complicated hand shapes in any direction becomes possible.

***Keywords***

*Shape Recognition; Active Appearance Model; 3D Hand Model*

## Introduction

Recently, new interface techniques meeting high demand for use in combination with high-functionality TV or personal computers that use hand gestures in order to free users up from remote-controls. With [Kinect for Xbox 360], we can use our hands in place of a mouse to control a pointer, or play movies and music. There are many advantages of being free from a remote-control. For example, we can handle the main devises directly with our hands without keeping in mind how to use the controller buttons, or concerning that it runs out of batteries.

However, even if a high-functional range image sensor is in utilization, there will be some limitations in using hand gestures, compared to a conventional remote-control, namely limitations associated with the field of recognition and the ability to recognize finger motions. In the Kinect interface, users control the pointer on the TV display with their hands and keep the pointer still for a while on the icon they want. Then they can convey their commands to the TV. This method is not only time-consuming in regard to giving operation commands, but it also easily results in mistakes in controlling the pointer.

On the other hand, when images are used, taken through a camera, even the finger shape can be recognized by using appearance information, and the total cost will be reduced. However, this method requires user to put his hands in front of a camera and in some cases, this makes things difficult for the user.

With these things in mind, in this paper, a method to recognize gestures has been put forward (including the shape of fingers) using both depth information and appearance information. At first, multiple 3D hand and finger models are constructed using depth information and appearance information. Then the complicated shapes of the hand and fingers are recognized in any direction, and the 3D model is switched according to the shape change.

## System Flow

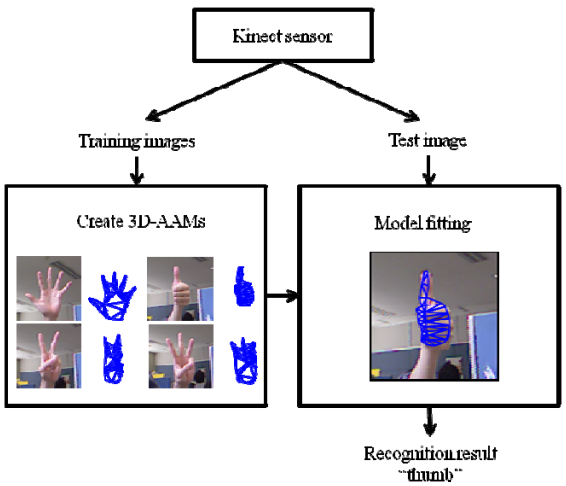



FIG. 1 SYSTEM FLOW

Fig. 1 shows the flow of the proposed system. In the learning phase, multiple 3D models are trained using depth information obtained from Kinect and RGB images. In this study, the 3D-AAM is applied, which combines active appearance models (AAMs) [T.F.

Cootes, 1995] [T.F. Cootes, 1998] [T. F. Cootes, 2002] and three-dimensional information as a three-dimensional model [J. Xiao, 2004] [M. Zhou, 2010] [V. Blanz, 1999]. The next step in the recognition phase is to fit each 3D model to the input images obtained from the Kinect sensor. Finally, the finger shape is output, which has a minimal difference between the model and the input finger.

## Hand Feature Extraction Using Multiple 3D-AAMs

Hand feature extraction using multiple 3D-AAMs [J. Xiao, 2004] is described in this section.

### *Active Appearance Models*

Cootes proposed an AAM that is mainly used to extract feature points of a face to represent the shape and texture variations of an object with a low dimensional parameter vector [T.F. Cootes, 1995]. The subspace is constructed by the application of principal component analysis (PCA) to the shape and texture of an object's feature points.

In the AAM framework, shape vector $x$ and texture vector $g$ of the object are given as follows:

$$x = (x_1, y_1, \cdots, x_n, y_n)^T \quad (1)$$

$$g = (g_1, g_2, \cdots, g_n)^T \quad (2)$$

where the shape $x$ indicates the coordinates of the feature points, and the texture vector $g$ indicates the gray-level of the image within the shape. For example, the AAM of the hand is constructed using 44 shape points as shown in Fig. 2. The texture consists of the intensities at pixels within triangular areas with feature points as shown in Fig. 3.

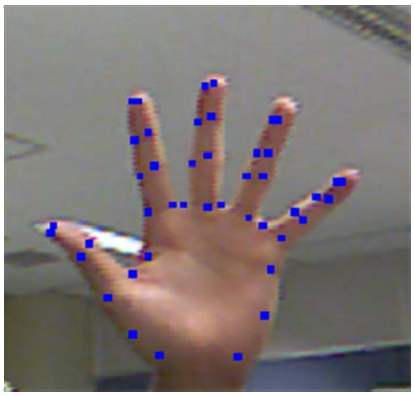


FIG. 2 HAND FEATURE POINTS

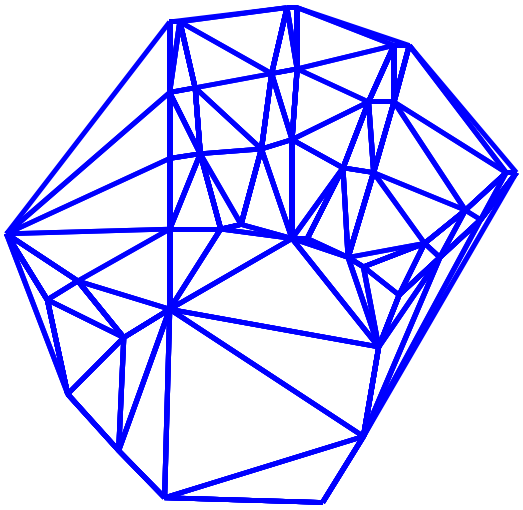


FIG. 3 TRIANGULAR AREAS IN SHAPE

Next, PCA is applied to the training data in order to obtain the normal orthogonal matrices, $P_S$ and $P_g$. Using the obtained matrices, the shape vector and texture vector can be approximated as follows:

$$x = \bar{x} + P_s b_s \quad (3)$$

$$g = \bar{g} + P_g b_g \quad (4)$$

where $\bar{x}$ and $\bar{g}$ are the mean shape and mean texture of the training images, respectively. $b_s$ and $b_g$ are the parameters of variation from the average. Further PCA is applied to the vector $b$ as follows:

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} + \begin{pmatrix} W_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{pmatrix} = Qc \quad (5)$$

$$Q = \begin{pmatrix} Q_s \\ Q_g \end{pmatrix} \quad (6)$$

where $W_s$ is a diagonal weight matrix for each shape parameter, allowing for the difference in units between the shape and texture models. $Q_s$ and $Q_g$ are the eigen matrices (including the eigenvectors). c is a vector of parameters controlling both the shape and gray-levels of the model. Finally, the shape and texture are approximated as functions of c.

$$x(c) = \bar{x} + P_s W_s^{-1} Q_s c \quad (7)$$

$$g(c) = \bar{g} + P_g Q_g c \quad (8)$$

### *Pose Parameter*

Using parameter $c$, it is possible to control variations in the shape and texture of the AAM. However, it is

not possible to express the position of the object in the image, the size of the object, or the object pose. The pose parameter $q$ is defined as the global posture change as follows:

$$q = [roll \quad scale \quad trans\_x \quad trans\_y] \tag{9}$$

where $roll$ indicates the rotation to the model plane, $scale$ indicates the size of the model, while $trans\_x$ and $trans\_y$ indicate the translation between $x$ and $y$, respectively.

### *Tracking AAM*

The goal of the AAM search is to minimize the error $E$ on the test image *Img* as shown in Eq. (10) with respect to $c$ and $q$,

$$E = \left[\left(\bar{g} + P_g Q_g c'\right) - I\left(Img, W\left(x; q', b_s'\right)\right)\right]^2 \tag{10}$$

where $W$ denotes the Affine warp function, and $I\left(Img, W\left(x; q', b_s'\right)\right)$ indicates the Affine transformed image controlled by the pose parameter $q$ on the test image. Thus, the most optimized $c$ parameter can be extracted from the test image.

### *3D-AAM*

In this paper, the 3D-AAM is used to extract the hand feature and to estimate the shape of the fingers. The AAM includes various fluctuation components (images) because the object images used as the training data include various changes, such as a left-side object image or right-side one, and an upturned or a downturned object. However, if so many changes are included in the training data of the object, the AAM cannot express their variation in PCA. Then the extraction accuracy of feature points will become lower. Since the variation of directions can be expressed as geometric change of shape, if a 3D shape with a right texture can be used, it can express the directional variations. This is the reason why the 3D-AAM is employed in the hand and finger shape recognition. The shape parameter is expanded into 3D using $z$ obtained from a depth image sensor, as shown in Eq. (11).

$$x = \left(x_1, y_1, z_1, \cdots, x_n, y_n, z_n\right)^T \tag{11}$$

### *RGBD Images Obtained Using Depth Image Sensor*

In the course of the creation of the 3D-AAM, three-dimensional shape information is required for a target. The 3D shape can be obtained, for example, using a 3D scanner or a stereo camera. However, we chose to use a Kinect sensor because this device is equipped with an RGB camera and infrared depth sensor. The 3D data expressed in Eq. (11) is obtained as a set of coordinate points on a target using a Kinect depth image sensor.

### *Elimination of Background Images in Triangular Areas Using Background Subtraction*

Background images are included in triangular areas of Fig. 3. Therefore, in order to eliminate the background images, background subtraction is performed using the depth data obtained from depth image sensor (Fig. 4).

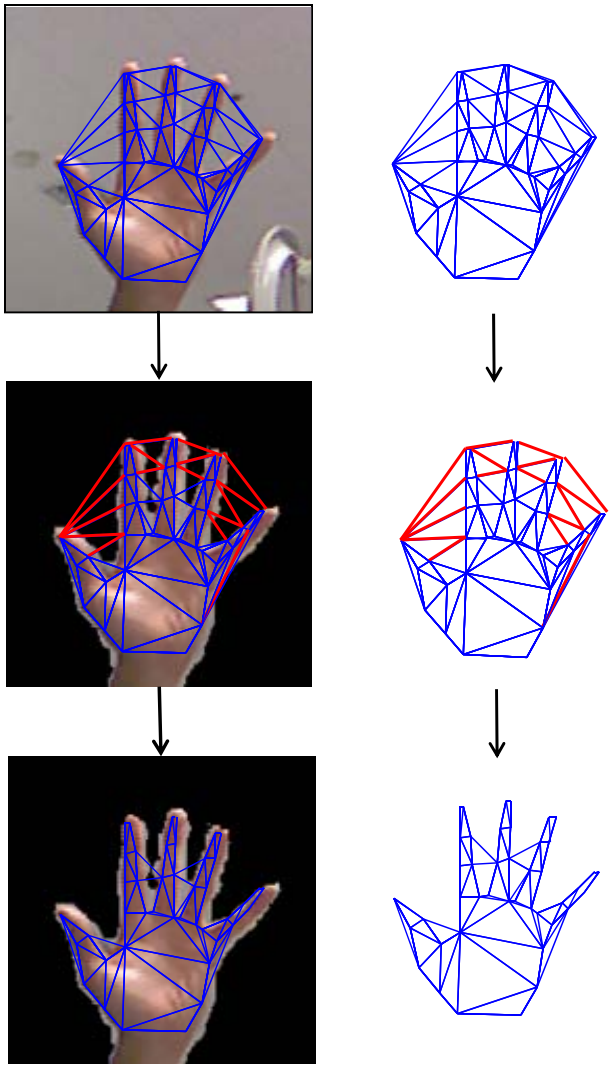


FIG. 4 BACKGROUND SUBTRACTION

### *Expanding of Pose Parameters*

The 2D pose parameters in Eq. (9) are expanded into 3D by adding yaw and pitch as shown in Eq. (12).

$$q = [yaw \quad pitch \quad roll \quad scale \quad trans\_x \quad trans\_y] \tag{12}$$

The moving variations of these parameters are shown

in Fig. 5.

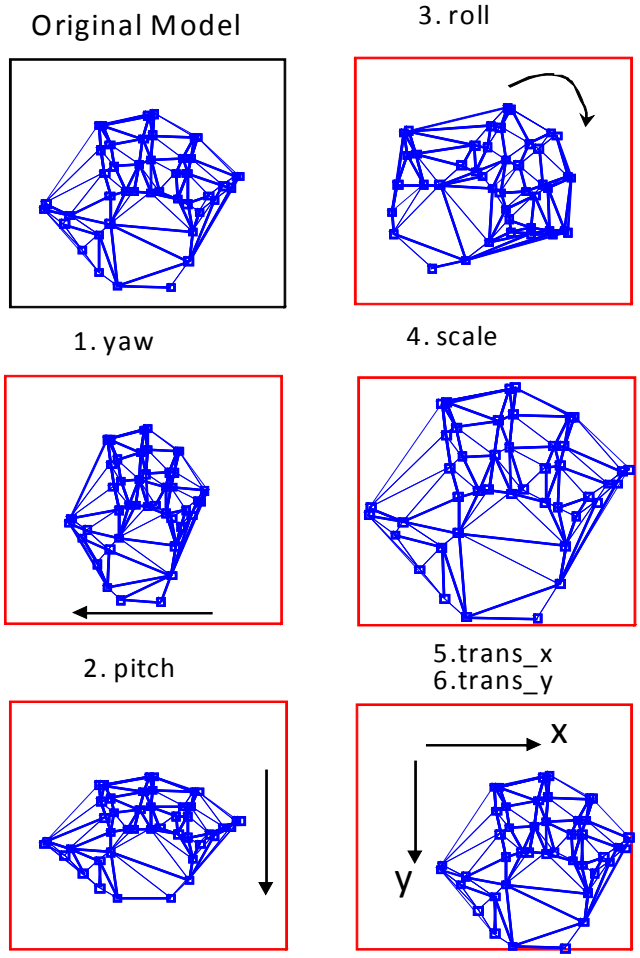



FIG. 5 3D POSE PARAMETERS

Using the six parameters, the 2D-AAM can be expanded into the three-dimensional AAM, and the parameters can transform the model viewed in all directions, angles, and positions. The transformation of the shape using this pose parameter is given as follows:

$$x_a = trans \cdot Scale \cdot RotZ \cdot RotY \cdot RotX \cdot x_b \qquad (13)$$

where $x_a$ and $x_b$ indicate the shape coordinate after and before transformation, respectively. Each transformation matrix is given by Eq. (14)-(18).

$$Trans = \begin{pmatrix} 1 & 0 & 0 & trans\_x \\ 0 & 1 & 0 & trans\_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (14)$$

$$Scale = \begin{pmatrix} scale & 0 & 0 & 0 \\ 0 & scale & 0 & 0 \\ 0 & 0 & scale & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (15)$$

$$RotX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(yaw*\pi/180) & -\sin(yaw*\pi/180) & 0 \\ 0 & \sin(yaw*\pi/180) & \cos(yaw*\pi/180) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (16)$$

$$RotY = \begin{pmatrix} \cos(pitch*\pi/180) & 0 & \sin(pitch*\pi/180) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(pitch*\pi/180) & 0 & \cos(pitch*\pi/180) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (17)$$

$$RotZ = \begin{pmatrix} \cos(roll*\pi/180) & -\sin(roll*\pi/180) & 0 & 0 \\ \sin(roll*\pi/180) & \cos(roll*\pi/180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (18)$$

### *Tracking the 3D-AAM*

Since the 3D-AAM consists of three-dimensional points and the input image consists of two dimensional pixels, it is necessary to project the 3D space into the 2D space when the error between the input image and the 3D model is calculated. Using the function P which projects the 3D space into the 2D space, the error can be calculated by Eq. (19). Then the optimized parameters are calculated using the same approach used in the 2D-AAM.

$$E = \left[\left(\bar{g} + P_g Q_g c'\right) - I\left(\mathrm{Im}\, g, P\left(W\left(x; q', b_s'\right)\right)\right)\right]^2 \qquad (19)$$

## Experiments

Multiple 3D-AAMs are constructed for the finger shapes and the model that produces the smallest error when the data is input into Eq. (19) is selected as the final result, as a sequence in the finger movie.

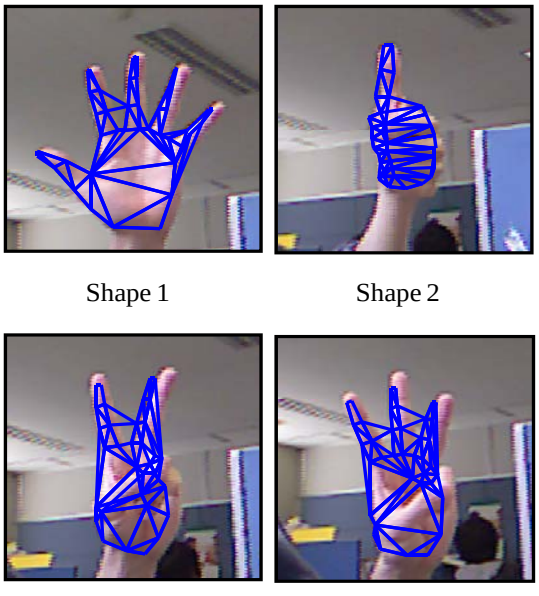



FIG. 6 FOUR MODELS OF FINGER SHAPE

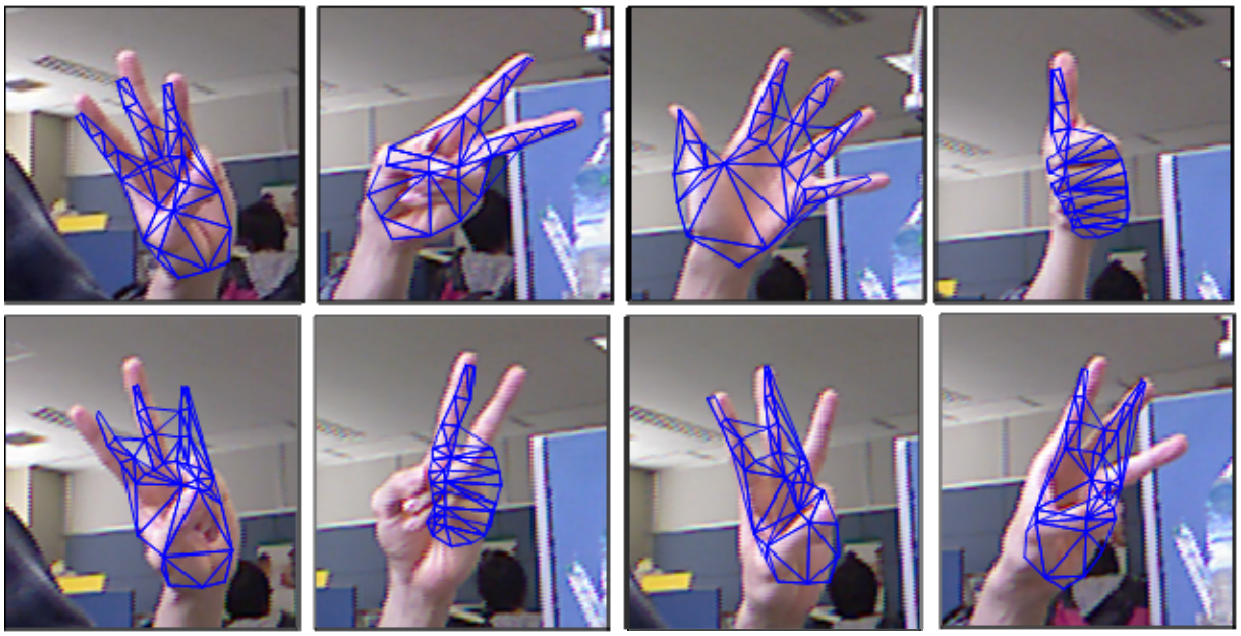


FIG 7. EXAMPLES OF EXPERIMENT RESULTS

### *Experiment Conditions*

Four types of three-dimensional models shown in Fig. 6 have been constructed in this experiment. The number of feature points included in respective model is 44, 34, 34 and 38 points, respectively. During the construction of each model, four depth images have been used to learn. Before learning, the background included in the training images has been excluded based on the background-subtraction. The training data and testing data collected from the same single person; included the finger shape with several variations taken from different directions.

### *Experiment Results*

Table 1 shows the confusion matrix of the finger shape recognition results. As a result of the experiment, it showed that the higher recognition rate has been obtained for shape 1 and shape 2. However, the false recognition can be seen between shape 3 and shape 4. It is thought that their finger shapes are similar so that the error of Eq. (19) becomes smaller. Fig. 7 shows the example of the outcome of this experiment.

TABLE 1 CONFUSION MATRIX AMONG FOUR SHAPE MODELS

| | Shape 1 | Shape 2 | Shape 3 | Shape 4 |
|---|---|---|---|---|
| Shape 1 | 1.0 | 0.0 | 0.0 | 0.0 |
| Shape 2 | 0.0 | 1.0 | 0.0 | 0..0 |
| Shape 3 | 0.0 | 0.011 | 0.966 | 0023 |
| Shape 4 | 0.0 | 0.0 | 0.327 | 0.673 |

## Conclusions

In this paper, by means of multiple three-dimensional models, the finger shape recognition method has been proposed. Using depth image sensor as a device, three-dimensional models were constructed by the acquisition of the depth information and RGB images of objects. The model recognized the finger shapes robustly against the various changes. The improvement in the finger shape models to be applied in the interface to TV or other display devices is the focus of further research.

## REFERENCES

Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade, "Real-Time Combined 2D+3D Active Appearance Models." CVPR, 535-542, 2004.

Mingcai Zhou, Yangsheng Wang, and Xiangsheng Huang, "Real-Time 3D Face and Facial Action Tracking Using Extended 2D+3D AAMs." ICPR, 3963-3966, 2010.

T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models – Their Training and Applications." Computer Vision and Image Understanding, Vol. 6, No. 1, 38-59, 1995.

T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models." ECCV, 484-498, 1998.

T.F. Cootes, K. Walker, and C.J. Taylor, "View-Based Active Appearance Models." Image and Vision Computing, 657-664, 2002.

V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces." SIGGRAPH, 187-194, 1999.

**Ryo Yamashita** received the B.E. in computer science from Kobe University in 2011. He is currently in graduate school of system informatics, Kobe University.

**Tetsuya Takiguchi** received the B.S. degree in applied mathematics from Okayama University of Science, Okayama, Japan, in 1994, and the M.E. and Dr. Eng. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 1996 and 1999, respectively.

From 1999 to 2004, he was a researcher at IBM research, Tokyo Research Laboratory, Kanagawa, Japan. Since 2004 he has been an Associate Professor at Kobe University. He stayed at University of Washington as visiting scholar from April 2012 to October 2012.

Dr. Takiguchi is a member of IEEE, Information Processing Society of Japan, and Acoustical Society of Japan.

**Yasuo Ariki** received his B.E., M.E. and Ph.D. in information science from Kyoto University in 1974, 1976 and 1979, respectively.

He was an assistant professor at Kyoto University from 1980 to 1990, and stayed at Edinburgh University as visiting academic from 1987 to 1990. From 1990 to 1992 he was an associate professor and from 1992 to 2003 a professor at Ryukoku University. Since 2003 he has been a professor at Kobe University. He is mainly engaged in speech and image recognition and interested in information retrieval and database.

Prof. Ariki is a member of IEEE, IPSJ, JSAI, ITE and IIEEJ.