LETTER

Improvement of Ringing Artifact Reduction Using a *K*-Means Method for Color Moving Pictures

Wonwoo JANG[†], **Hagyong HAN**[†], **Wontae CHOI**[††], **Gidong LEE**[†], ***Nonmembers***, ***and* Bongsoon KANG**[†a)], ***Member***

SUMMARY This paper proposes an improved method that uses a *K*-means method to effectively reduce the ringing artifacts in a color moving picture. To apply this improved *K*-method, we set the number of groups for the process to two (K=2) in the three dimensional *R*, *G*, *B* color space. We then improved the *R*, *G*, *B* color value of all of the pixels by moving the current *R*, *G*, *B* color value of each pixel to calculated center values, which reduced the ringing artifacts. The results were verified by calculating the overshoot and the slope of the light luminance around the edges of test images that had been processed by the new algorithm. We then compared the calculated results with the overshoot and slope of the light luminance of the unprocessed image.

key words: *ringing artifacts reduction, K-means method, color moving picture, image processing*

1. Introduction

When viewing images on video devices, a vivid image is desirable. Therefore, many new techniques are emerging for enhancement of image qualities in devices such as mobile camera phones, digital cameras, and televisions. In order to obtain the goal, it is necessary to enhance blurred images and highlight fine details in images by sharpening edges. Image sharpening can be achieved by high pass filtering, since edges and other abrupt changes in images are associated with high-frequency components. However, visual artifacts are generated during high pass filtering, and the overshoots and undershoots may give rise to ringing artifacts [1]. These artifacts can also arise from Gibbs-like oscillations in the vicinity of image discontinuities in wavelet coding schemes. In general, this results from the quantization of wavelet coefficients that represent high spatial frequencies [1]–[4]. Therefore, to overcome the serious ringing artifacts encountered during the image processing, we need to use novel technologies to achieve a clear image.

A number of strategies have been tried thus far. For example, to reduce artifacts occurring during the coding process, C. Wang et al. presented a fast edge-preserved post-processing algorithm that used the Sobel operator to search for impulse-like structure noise. A bilateral filter with a central 6×6 window was then used, while keeping sharp edges [1]. S.H. Oguz et al. proposed a morphological post-filtering algorithm that employed a spatially localized non-linear filtering operation based on the local knowledge principle of morphology [2]. I. Popovici et al. employed a frequency-domain analysis algorithm that split a whole image into several blocks of 8×8 pixels, and then applied Gaussian filtering to alleviate ringing artifacts near the shape edges and other details in the images [3].

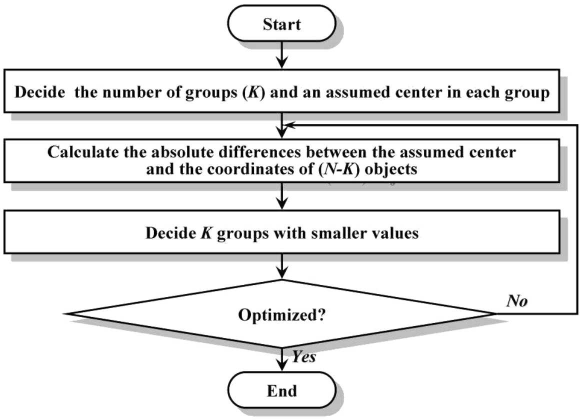

Start
Decide the number of groups (*K*) and an assumed center in each group
Calculate the absolute differences between the assumed center and the coordinates of (*N*-*K*) objects
Decide *K* groups with smaller values
Optimized?
No
Yes
End

Fig. 1 Simplified flowchart for a *K*-means algorithm.

Still others have investigated the ringing artifacts that occur in distorted images after the image has been processed. S. Yang et al. proposed a maximum-likelihood algorithm that employed a parameter estimation method based on the *K*-means algorithm with the number of clusters determined by a cluster-separation measure [4].

The original *K*-means algorithm clusters *N* objects into *K* mutually exclusive groups [4]–[7]. A simplified flowchart of the *K*-means algorithm is depicted in Fig. 1. The first step is to decide the number of groups to be denoted as *K* and then to choose an object in each group as an assumed center. The original *K*-means algorithm then calculates the absolute differences between this center and the coordinates of (N-K) objects. Next, the objects need to be divided into *K* groups based on smaller values among the calculated differences. After processing the (N-K) objects with the above method, new centers can be generated in *K* groups by averaging the coordinate values in each group. Unless new centers are equal to the previous centers, the process in Fig. 1 iterates by itself.

The calculation time of the original *K*-means algorithm

Manuscript received May 27, 2009.
Manuscript revised September 8, 2009.

[†]The authors are with the Dong-A University, 840, Hadan 2-dong, Saha-gu, Busan, 604-714, Korea.
[††]The author is with the SAMSUNG Electro-Mechanics Co. Ltd., 314 Matan-3-dong Young-tong-gu, Suwon, 443-803, Korea.
a) E-mail: bongsoon@dau.ac.kr

DOI: 10.1587/transfun.E93.A.348

is intrinsically dependent on the K number and iteration number. The original K-means algorithm may require very long processing time if the use of a high K number required. Therefore, in order to improve the ringing artifacts of a color moving picture, we need to apply a small K number and iteration number during the process. This would allow real-time processing, in which received data would be processed almost immediately [8]–[10].

In this paper, we propose a process method that can reduce the ringing artifacts of the color moving picture using the K-means algorithm. In this case, the method applies two K numbers and a single iteration number, because a small K number can also provide the desired performance [4], [5], [11]. In addition, a small K number and single iteration number can permit the improvement of the moving picture in real time.

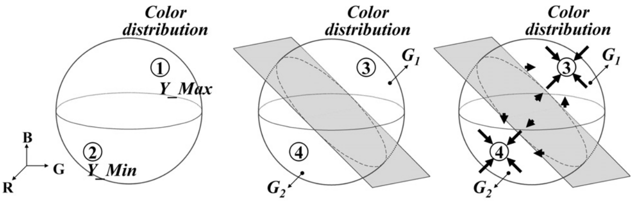
We moved the R, G, B values of each pixel to the desired positions in the R, G, B space using the proposed method. For verification, we calculated overshoots and slopes of three test images processed by moving the R, G, B color value of the each pixel to center values. In order to verify the calculated results, we compared the calculated overshoot and slope by the proposed method, with the same values of the unprocessed test images. We used an overlapped block structure of 8 × 8 pixels to prevent the block-discontinuity. The overlapped block was derived by using four 16 × 16 pixel-blocks in [12]. We also designed a real-time imaging processing system using Verilog-HDL for the digital video camera and high-quality display.

2. Proposed Algorithm

Figure 2 shows the representation of the color improvement for reducing ringing artifacts on the three-dimensional R, G, B space [13]. For each block in the image, we have to decide on two center colors using the K-means algorithm (K = 2). Figure 2(a) shows the sphere for the R, G, B color distribution of the each block.

We first select two reference colors with maximum and minimum luminance values among all of the pixels in the each block in Fig. 2(a). The luminance of R, G, B data can be calculated as follows [14]:

$$luminamce = 0.3 \times R_{in} + 0.59 \times G_{in} + 0.11 \times B_{in} \tag{1}$$



(a) (b) (c)

Fig. 2 Representation of the color improvement for reducing the ringing artifacts on the three-dimensional RGB spaces: (a) two reference colors of the unprocessed image, (b) two center colors by K-means processing, and (c) approach of the color positions to two center color positions for improvement of ringing artifacts.

where R_{in}, G_{in}, B_{in} denotes the component values of R, G, B data in each pixel. From Eq. (1), the two reference colors can be used as first positions for calculation [4], [5], [11]. ① and ② in Fig. 2(a) represent the color positions with the maximum *Y_Max* and the minimum luminance *Y_Min* in the block, respectively.

We then generate two new center colors, indicated as ③ and ④ in Fig. 2(b), using the K-means method. We also divide all colors of the pixels in each group into two groups (G_1 and G_2), again by the K-means method, as shown in Fig. 2(b) [4]. The slanted gray surface penetrating the color sphere represents the calculated virtual boundary surface. From the calculated two new center colors and the virtual boundary surface, we can then calculate the new color values of all pixels to effectively remove the ringing artifacts.

The calculation process for making the current colors in each pixel approach the center colors on the color sphere consists of two criteria, the *Proximity_ratio and Local_color_cohesion* [12]. In this paper, the *Proximity_ratio* represents the ratio of the distance between one center color position and another center color position from the current color position in each pixel in the three dimensional R, G, B color space. The *Proximity_ratio* can be calculated as follows:

$$Proximity\_ratio = \sqrt{\frac{(R_{in}-C_a(R))^2 + (G_{in}-C_a(G))^2 + (B_{in}-C_a(B))^2}{(R_{in}-C_b(R))^2 + (G_{in}-C_b(G))^2 + (B_{in}-C_b(B))^2}} \tag{2}$$

where C_a and C_b represent center color positions ③ and ④, when the current color of the pixel exists in the G_1, respectively. In contrast, C_a and C_b represent the center color position ④ and ③ when the current color of the pixel exists in the G_2.

The criterion *Local_color_cohesion* implies the accumulated color distances between the center color position and the current color position of given pixels in each block. Using this parameter, we can control the approaching length in each block for vivid images. In order to derive *Local_color_cohesion*, we define *Center_dist*, which represents the distance between two center colors in the three dimensional R, G, B color space, as shown below:

$$Center\_dist = \Big((C_1(R)-C_2(R))^2 + (C_1(G)-C_2(G))^2 + (C_1(B)-C_2(B))^2\Big)^{1/2} \tag{3}$$

where $C_1(R)$, $C_1(G)$, and $C_1(B)$ denote R, G, B color value of the center color position in G_1, and $C_2(R)$, $C_2(G)$, and $C_2(B)$ represent R, G, B color value of the center color position in G_2. We also define the parameter Interval as the sum of three absolute differences between the R, G, B color values of the center colors and the current color values in each pixel:

$$Interval = |R_{in} - C_1(R)| + |G_{in} - C_1(G)| + |B_{in} - C_1(B)| \quad \text{(in the Group } G_1\text{)} \tag{4}$$

or

$$Interval = |R_{in} - C_2(R)| + |G_{in} - C_2(G)| + |B_{in} - C_2(B)| \quad \text{(in the Group } G_2\text{)}$$

In this paper, we calculate the parameter *Local_color_cohesion* in a given range, as indicated below, because we need to consider color values of the pixels that are farther from the center color positions. Before the calculation of *Local_color_cohesion*, therefore, we use a parameter *Far_interval* for the appropriate calculation, as below, which is defined as the filtered *Interval* that is larger than one fourth of the *Center_dist*.

$$Far\_interval = \begin{cases} Interval, & Interval > Center\_dist/4 \\ 0, & else \end{cases} \tag{5}$$

Consequently, the summation of *Far_interval* implies the summation of the variation between the center color position and the current color positions. As this summation value of all *Far_interval* gets larger, more high frequency details, except the ringing artifacts, could be present in the image in each block. The parameter *Local_color_cohesion* can then be calculated, as below:

$$Local\_color\_cohesion = \left(\frac{\sum Far\_interval}{Center\_dist} \right)^{-1} \tag{6}$$

Using the calculated parameters *Proximity_ratio* and *Local_color_cohesion*, we can calculate the new improved color values of the pixels in the pane, as follows:

$$\begin{aligned} \Delta R &= c \times Local\_color\_cohesion \\ &\quad \times (1 - Proximity\_ratio) \times (C_i(R) - R_{in}) \\ \Delta G &= c \times Local\_color\_cohesion \\ &\quad \times (1 - Proximity\_ratio) \times (C_i(G) - G_{in}) \\ \Delta B &= c \times Local\_color\_cohesion \\ &\quad \times (1 - Proximity\_ratio) \times (C_i(B) - B_{in}) \end{aligned} \tag{7}$$

where c is a weight constant for the control of light luminance. ΔR, ΔG and ΔB indicate the calculated color distances from R_{in}, G_{in}, B_{in} to a color position of the center position for improvement, respectively. C_i is the center position ③ if the current color positions of the pixel exist in group G_1. C_i is the center position ④ if the current color positions of the pixel exist in group G_2. The lengths of the arrows in Fig. 2(c) represent the amplitude of the color variations in each color position for the improvement in each pixel. In Fig. 2(c), the calculation from Eq. (7) shows that the color positions closer to the center position have a larger color difference value compared to the color positions farther from the center color position. If the color values of the pixels exist on the virtual boundary surface, the color of the pixels will never move to the desired positions. Even when the proposed algorithm is applied to an ideally flat image with all pixels of same color values, the algorithm generates the same center colors of (③ and ④) in Fig. 2(b) for three dominant colors in each block. In this case, the *Center_dist* of Eq. (3) and the *Local_color_cohesion* of Eq. (6) become zero. Thereby, the calculated color distances of Eq. (7) are also zero. This implies that there are no color variations in each color position.

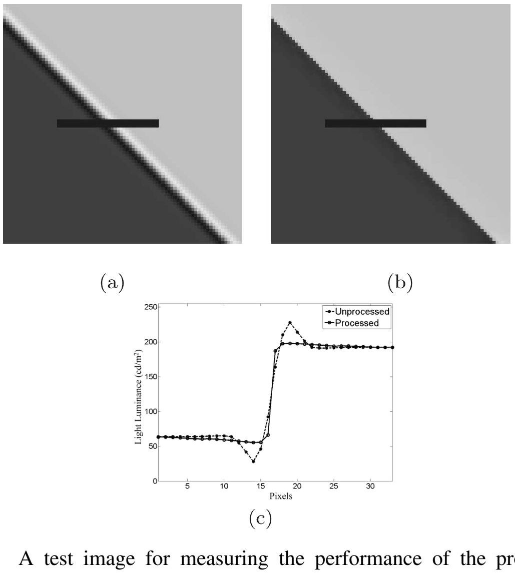
3. Results and Discussion

In order to verify the performance of the proposed method, we have evaluated three images that were captured from a color moving picture. These were processed by the proposed method and compared to the unprocessed images by measuring the overshoot and the slope of the light luminance on the edge of the three test images. First of all, we selected a gray test pattern with a ringing artifact in order to confirm the effect of the proposed method, as shown in Figs. 3(a) and (b). In order to measure the quantitative and qualitative performance for reducing ringing artifacts, we measured the overshoot and the slope of the light luminance on the edge of both the processed image and the unprocessed image. The overshoot and the slope of the light luminance can be expressed as below:

$$\begin{aligned} Overshoot &= \frac{y_{max} - y_{high}}{y_{high} - y_{low}} \\ Slope &= \frac{[(y_{high} - y_{low}) \times 0.9] - [(y_{high} - y_{low}) \times 0.1]}{x_{high} - x_{low}} \\ &= \frac{[(y_{high} - y_{low}) \times 0.8]}{x_{high} - x_{low}} \end{aligned} \tag{8}$$

where y_{max}, y_{high} and y_{low} represent the positive overshoot value, the high value, and the low value toward light convergence in the edge profile, respectively. The x_{high} and x_{low} represent the x positions for which light luminance y corresponds to 90% of the y_{high} and 10% of the y_{low}, respectively.

In Fig. 3, we obtained one-dimensional cross-sectional



(a)

(b)

(c)

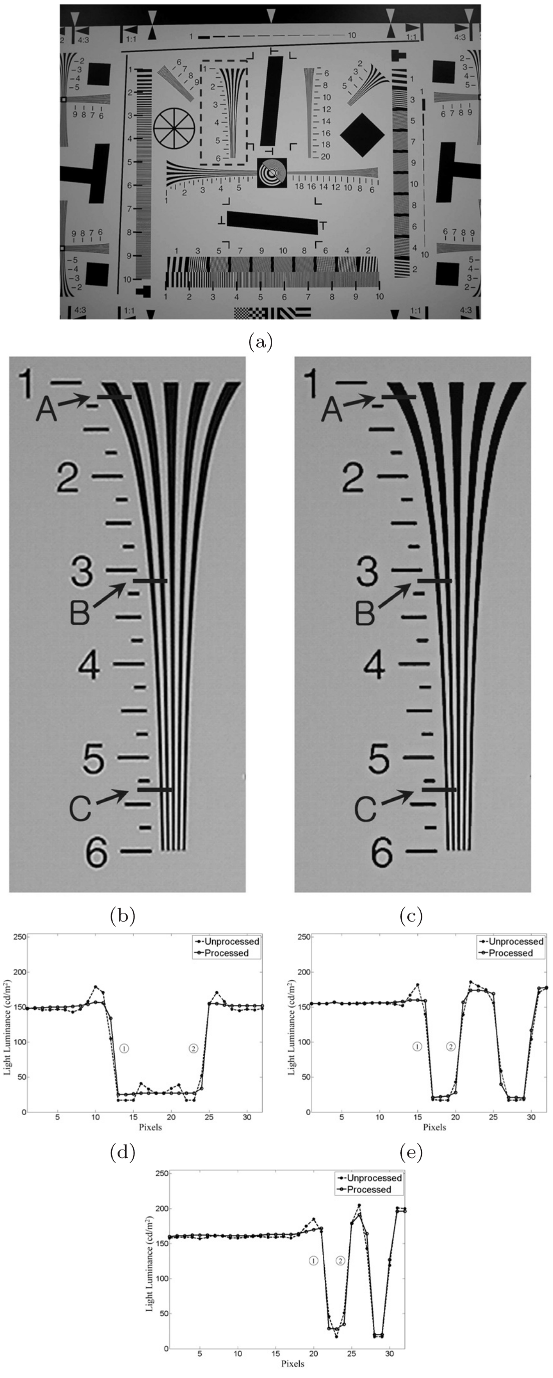
Fig. 3 A test image for measuring the performance of the proposed method: (a) unprocessed gray test pattern (80 × 80), (b) processed gray test pattern (80 × 80), (c) the calculated light luminance along the cross-sectional line at the center solid line.

Table 1 Comparisons of the performance on ringing artifacts in Fig. 3.

Position	Unprocessed		Processed	
	Overshoot	Slope $\left(\frac{\text{cd}\cdot\text{pixel\_width}}{\text{m}^3}\right)$	Overshoot	Slope $\left(\frac{\text{cd}\cdot\text{pixel\_width}}{\text{m}^3}\right)$
	227.87 (28.01%)	61.4	197.98 (4.67%)	120.84

pixel values, as shown in Figs. 3(a) and (b) to compare the optical performance of the images. The solid line in Fig. 3 shows the cross-sectional line for calculation. Figure 3(c) shows the calculated light luminance along the cross-sectional line of the images as shown in Figs. 3(a) and (b). In Fig. 3(c), we can recognize that the overshoot and undershoot of the unprocessed image can be effectively removed by applying the proposed method. Table 1 shows the calculated results of the overshoot and the slope of the light luminance in the two test images, as shown in Figs. 3(a) and (b). The overshoot on the edge of the processed image is reduced to 4.67% from the 28.01% in the unprocessed image. The slope of the light luminance provides a sharper line in the processed image compared to the unprocessed image. From Fig. 3 and Table 1, we can confirm that the ringing artifacts were effectively reduced, as shown in Fig. 3(b), using the proposed method.

The second test image is the International Organization for Standardization (ISO) resolution chart captured from the verification board as shown in Fig. 4(a) [13]. The dotted rectangular line indicates the region for testing the ringing artifacts in the two processed and the unprocessed image. Figures 4(b) and (c) show the unprocessed image and the image processed by the proposed method. We measured the overshoot and the slope at the same three positions in the images. Cross-sectional solid lines (*A*, *B*, and *C*) in Figs. 4(b) and (c) show the measured line position for comparing the optical performance. Figures 4(d), (e) and (f) show the calculated optical luminance at 3 cross-sectional line position *A*, *B*, and *C*, respectively. Positions ① and ② in Fig. 4 show the edge position in each line. From Figs. 4(d), (e), and (f), we can perform the calculation of the overshoot and the slope for each position. Table 2 shows the calculated results of the overshoots and the slopes of the light luminance in the two test images, as shown in Figs. 4(b) and (c). In this table, the average value of overshoot on the edges is reduced from 20.22% to 8.35% upon applying the proposed method. The ringing artifacts around the thick line (position *A* and *B*) can be more effectively removed compared to the thin line (position *C*). Consequently, we expect that the ringing artifacts around larger objects in a moving picture can be easily removed by applying this proposed method. The average absolute value of the calculated slope along the cross-sectional lines is also improved from 86.53 to 112.3.



(a)

(b)

(c)

(d)

(e)

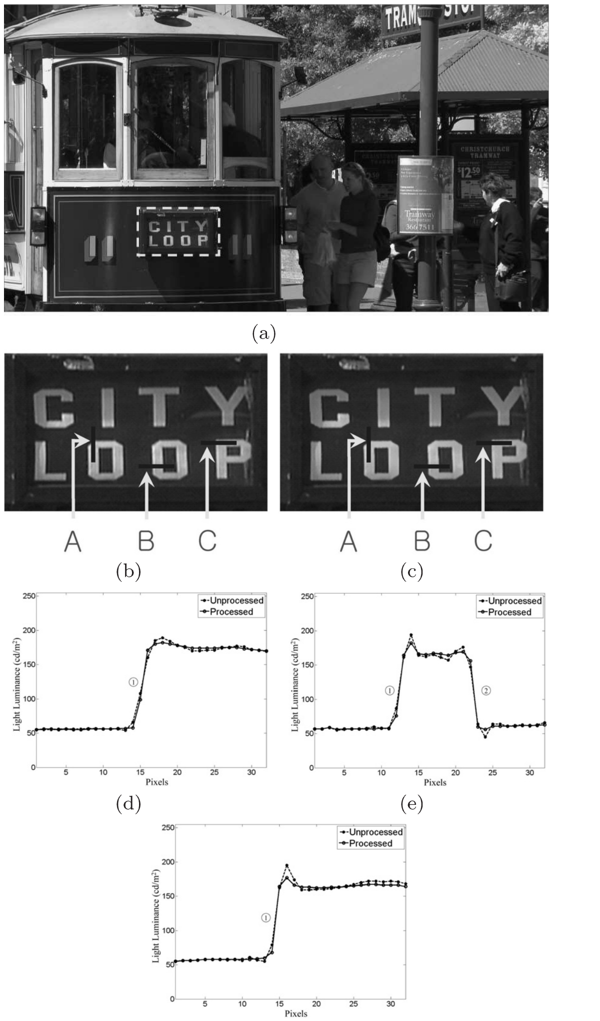
(f)

Fig. 4 International Organization for Standardization (ISO) resolution chart for evaluating the proposed method: (a) unprocessed ISO resolution chart (2048 × 1536), (b) zoomed measured region of the unprocessed ISO resolution chart (220 × 500), (c) zoomed measured region of the processed ISO resolution chart (220 × 500), (d) the calculated light luminance along the cross-sectional line *A*, (e) the calculated light luminance along the cross-sectional line *B*, (f) the calculated light luminance along the cross-sectional line *C*.

The last test image is the complex scene shown in Fig. 5(a) captured from the EPSON Demo DVD. The dotted rectangular line in Fig. 5(a) shows the measured region for testing the ringing artifacts. We also selected 3 cross-sectional line positions (*A*, *B*, and *C*) for measuring the overshoot and the slope in the processed and unprocessed images as shown in Figs. 5(b) and (c). Figures 5(d), (e), and (f) also show the calculated optical luminance at 3 cross-sectional line positions, *A*, *B*, and *C*, respectively. Table 3 shows the

Table 2 Comparisons of performance on ringing artifacts in each measured position in Fig. 4.

Positions		Unprocessed		Processed	
		Overshoot	Slope $\left(\frac{\text{cd}\cdot\text{pixel\_width}}{\text{m}^3}\right)$	Overshoot	Slope $\left(\frac{\text{cd}\cdot\text{pixel\_width}}{\text{m}^3}\right)$
A	①	179 (20.00%)	−79.75	157 (3.70%)	−93.58
	②	171 (14.07%)	73.86	155 (2.22%)	79.2
B	①	182 (19.42%)	−119.74	160 (3.60%)	−138
	②	186 (22.30%)	71.81	174 (13.67%)	129
C	①	185 (15.86%)	−86.64	172 (6.90%)	−143
	②	205 (29.66%)	87.39	191 (20.00%)	90.54



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 5 A complex scene captured from the EPSON Demo DVD: (a) unprocessed complex scene (1920 × 1080), (b) zoomed measured region of the unprocessed complex scene (250 × 150), (c) zoomed measured region of the processed complex scene (250 × 150), (d) the calculated light luminance along the cross-sectional line A, (e) the calculated light luminance along the cross-sectional line B, (f) the calculated light luminance along the cross-sectional line C.

Table 3 Comparisons of performance on ringing artifacts in each measured position in Fig. 5.

Positions		Unprocessed		Processed	
		Overshoot	Slope $\left(\frac{\text{cd}\cdot\text{pixel\_width}}{\text{m}^3}\right)$	Overshoot	Slope $\left(\frac{\text{cd}\cdot\text{pixel\_width}}{\text{m}^3}\right)$
A	①	189 (15.52%)	47.01	182 (9.48%)	56.96
B	①	194 (26.85%)	55.81	182 (15.74%)	64.28
	②	176 (10.19%)	−71.85	169 (3.70%)	−96
C	①	195 (22.12%)	64.29	177 (6.19%)	96

calculated overshoot and the slope obtained from Figs. 5(d), (e), and (f). In this table, a smaller overshoot (8.78%), compared to the overshoot (18.67%) of the measured positions in the unprocessed image, can be provided by applying the proposed method. The slope of the measured position in the processed image also becomes more sharpened compared to the slope of the measured positions in the unprocessed image, as shown in Table 3. The designed real-time image processing system was also experimentally proven using a FPGA device with a USB interface board and a CMOS image sensor.

4. Conclusion

In this paper, we proposed a method to reduce ringing artifacts in a color moving picture that uses a K-means method. For practical purposes, in the color moving picture, we set the number of groups to two (K=2) and a single iteration number in the three dimensional R, G, B color space. In order to verify the performance of the proposed method, we evaluated the images processed by the proposed method and compared these to the unprocessed images by calculating the overshoot and slope of the light luminance in the test images. The performance of the proposed method for a color moving picture was verified by comparing the captured images from a color moving picture. We confirm that the proposed method can be applied to many visual devices, such as mobile camera phones, digital cameras, and televisions.

Acknowledgments

This paper was supported by research funds from Dong-A University.

References

[1] C. Wang, P. Xue, W. Lin, W. Zhang, and S. Yu, "Fast edge-preserved postprocessing for compressed images," IEEE Trans. Circuits Syst. Video Technol., vol.16, no.9, pp.1142–1147, Sept. 2006.

[2] S.H. Oguz, Y.H. Hu, and T.Q. Nguyen, "Image coding ringing artifact reduction using morphological post-filtering," Proc. IEEE 2nd Workshop Multimedia Signal Processing, pp.628–633, Dec. 1988.

[3] I. Popovici and W.D. Withers, "Locating edges and removing ringing artifacts in JPEG images by frequency-domain analysis," IEEE Trans. Image Process., vol.16, no.5, pp.1470–1474, May 2007.

[4] S. Yang, Y.H. Hu, T.Q. Nguyen, and D.L. Tull, "Maximum-likelihood parameter estimation for image ringing-artifact removal," IEEE Trans. Circuits Syst. Video Technol., vol.11, no.8, pp.963–973, Aug. 2001.

[5] T.N. Pappas, "An adaptive clustering algorithm for image segmentation," IEEE Trans. Signal Process., vol.40, no.4, pp.901–914, April 1992.

[6] D. Charalampidis, "A modified k-means algorithm for circular invariant clustering," IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.12, pp.1856–1865, Dec. 2005.

[7] M.C. Su and C.H. Chou, "A modified version of the K-means algorithm with a distance based on cluster symmetry," IEEE Trans. Pattern Anal. Mach. Intell., vol.23, pp.674–680, June 2001.

[8] S.M. Chai, A. Gentile, W.E. Lugo-Beauchamp, J. Fonseca, J.L. Cruz-Rivera, and D.S. Wills, "Focal-plane processing architectures for real-time hyperspectral image processing," Appl. Opt., vol.39, pp.835–849, 2000.

[9] O. Matoba, E. Tajahuerce, and B. Javidi, "Real-time three-dimensional object recognition with multiple perspectives imaging," Appl. Opt., vol.40, pp.3318–3325, 2001.

[10] D. Doswald, J. Hafliger, P. Blessing, N. Felber, P. Niederer, and W. Fichtner, "A 30-frames/s megapixel real-time CMOS image processor," IEEE Trans. Solid-State Circuits, vol.35, no.11, pp.1732–1743, Nov. 2000.

[11] D.L. Pham, "Edge-adaptive clustering for unsupervised image segmentation," Proc. Intl. Conf. Image Processing, pp.816–819, Sept. 2000.

[12] J. Kim, W. Jang, B. Kwak, S. Kim, and B. Kang, "A new image-scaling algorithm eradicating blurring and ringing to apply to camera phones," Proc. Int. Conf. Consumer Electronics, p.5.4–1, Jan. 2007.

[13] W. Jang, J. Park, J. Kim, B. Kwak, and B. Kang, "Ringing artifacts removal system for mobile application camera by modified K-means algorithm," Proc. Intl. Conf. ICSPCS, pp.1–5, Dec. 2007.

[14] K. Jack, Video Demystified: A Handbook for the Digital Engineer, HighText Pub., p.42, 1996.