Anti-shaking Algorithm for the Mobile Phone Camera in Dim Light Conditions

Jong-Ho Lee[1], In-Yong Shin[1], Hyun-Goo Lee[2], Tae-Yoon Kim[2], and Yo-Sung Ho[1]

[1] Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea
{jongho,siy0808,hoyo}@gist.ac.kr
[2] Mobile Handset R&D Center, LG Electronics Inc.
327-23 Gasan-dong, Gumcheon-gu, Seoul 153-802, Korea
{dennylee,jrock}@lge.com

Abstract. A picture is blurred when it is taken by the mobile phone camera in dim light conditions. There are some hardware approaches for commercial use to solve this problem, but they cost high and occupy a large space in the camera. A software approach, called image deblurring, takes much time to process and its result is not suitable for real applications because of the ringing effect. In this paper, we introduce a fast anti-shaking algorithm that is suitable for the mobile phone camera which uses a full-size image with fast shutter speed, and a preview image. The proposed algorithm is relatively free from both cost and size restraints, and shows good ability to prevent the blur of the image. The processing time was 1.47sec and showed the blur correction rate of 65.2%.

Keywords: Anti-shaking, mobile phone camera.

1 Introduction

An anti-shaking algorithm is for reducing the blur in the picture caused by the camera shake. If you take a picture in dim light conditions, it takes some time to obtain enough light. Camera shaking by hands during this time results in the blurred image.

There are many technologies in commercial use to reduce the blur of the picture. Canon's image stabilizer or Konica Minolta's anti-shake shift a lens group or charge coupled device (CCD) to compensate for camera's motion. Although these hardware approaches show good performances for preventing blur and generating sharp images, they cost high and occupy much space in the camera. Since cost and size are very critical issues for mobile phone cameras, this implementation is not suitable for the mobile phone camera.

There is also an image processing technique called image deblurring. Since image deblurring is an ill-posed problem where the PSF(point spread function) and the sharp image is estimated from the blurred image alone, it requires many iteration for restoring the sharp image so that it takes much time to process. It takes 10 minutes with MATLAB for the 128×128 image by the algorithm proposed by Fergus *et al.*[1]. The deblurring method using blurred image and noisy image pairs introduced by Yuan *et al* [2] shows quite good result, but requires too much memory space because it has

P. Muneesawang et al. (Eds.): PCM 2009, LNCS 5879, pp. 968–973, 2009.

to handle two images simultaneously. The deblurring method proposed by Shan *et al* reduces the ringing artifacts significantly, but it takes about 20 minutes to deblur 1600×1200 color image with C source code [3].

Our approach uses advantages of both software and hardware approaches. We reduce the blur with the fast shutter speed of the camera, and then, we get the output image close to the original sharp image by denoising and histogram matching with the aim of the preview image.

This paper is organized as follows. The proposed algorithm composed of input image acquisition, adaptive bilateral filtering and color correction is addressed in Section 2. We will show the experimental results in Section 3. Finally, conclusions are drawn in Section 4.

2 Proposed Anti-shaking Algorithm

Figure 1 shows the overview of the proposed algorithm. The full-size image is taken with fast shutter speed to reduce the blur. Since the full-size image obtained with fast shutter speed is noisy and dark, we reduce the noise and correct the color of the full-size image. With these procedures, we can obtain the sharp image which has reduced blur.

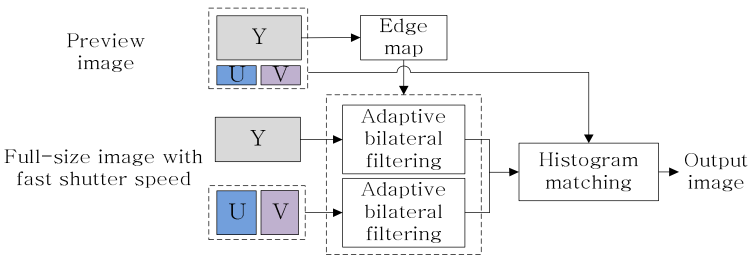


Fig. 1. Overview of the proposed algorithm

2.1 Input Image Acquisition

If the anti-shaking is activated, 1600×1200 full-size image with YUV 4:2:2 format is taken with the fast shutter speed of 70ms. This full-size image is not blurred but noisy and dark. 320×240 preview image with YUV 4:2:0 format is also available in the camera. The preview image is captured during enough time, so it is not noisy and contains right color information.

2.2 Adaptive Bilateral Filtering

If the picture is taken with fast shutter speed, noise flow residing in the camera becomes prominent due to the low signal intensity, so we get the noisy image. We use bilateral filter to remove the noise which is relatively fast and maintains edge well. Other state-of-the-art denoising algorithms like the non local means algorithm show better performance than bilateral filter, but they are two slow [6].

$$F[I]_P = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|I_p - I_q|) I_q, \quad (1)$$

where $F[I]_P$ is the restored pixel at position *p*, G_{σ_s} is the Gaussian function whose variable is the distance between *p* and *q* and whose standard deviation is σ_s, G_{σ_r} is the Gaussian function whose variable is the difference between the pixel value at *p* and the pixel value at *q* and whose standard deviation is σ_r, I_q is the pixel value at *q*, *S* is the neighborhood of *p* enclosed by window whose center is located at *p* and W_p is the sum of $G_{\sigma_s} G_{\sigma_r}$ in the window.

The bilateral filter requires three parameters, window size, σ_s and σ_r. If we change these filtering parameters adaptively according to the features on the image, we can get better denoised result. For these adaptive filtering, we need edge information of the image, but it is very difficult to extract edge information from the noisy image since both noise and edge are high frequency components. Thus, we use preview image which is free from the noise. Since the preview image has small size and does not contain noise, the edge information can be obtained efficiently and rapidly with the simple Sobel algorithm.

For adaptive filtering, the image is classified into three regions using the edge map extracted from the Y component of the preview image. We slide 3×3window on the edge map, and three regions are defined according to the edge information in the window. Region 1 is the edge-free region, region 2 is the region which has strong edge and region 3 is the region which has many edges. We filter the image with different filtering parameters set according to the classified region. A pixel of the edge map corresponds to 5×5 block. Since the noise in the chrominance components of the full-size image is more unpleasant to the eye than that of the luminance components of the full-size image, we generally filter the chrominance parts with stronger filtering parameters. The filtering parameters according to the classified region are presented in the table 1.

Table 1. Filtering parameters according to region classification

components	Classified region	Window size	σ_s	σ_r
Y	region 1,2,3	3 x 3	1	50
U,V	Region 1	11 x 11	1	50
	Region 2	7 x 7	1	30
	Region 3	3 x 3	1	5

2.3 Histogram Matching

The full-size image taken with fast shutter speed is dark due to the shortage of light. We correct the pixel intensity of the full-size image referring to the preview image which has right color information using the histogram matching.

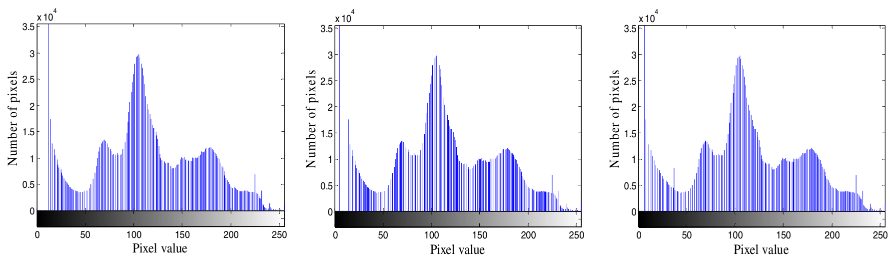
At first, we reduce the size of the full-size image to the size of preview image by down-sampling. After that, we get the cumulative histograms of two images, down-sampled full-size image and preview image. We get the mapping function from these cumulative histograms and substitute the pixel value of the full-size image with the values obtained from this mapping function. However, since all clipped pixel values below some threshold of the full-size image are mapped to the first active bin in the histogram like fig. 2. (a), this causes an unnatural image. We removed these artifacts using the Fecker's method [5]. If we define the mapping value of 0 as $M[0]$ and the histogram of pixel i of the preview image as $H[i]$, then the center of mass c of the values for the clipped interval can be calculated as follows.

$$c = \sum_{i=0}^{M[0]} iH[i] / \sum_{i=0}^{M[0]} H[i] \tag{2}$$

This value is used for the mapping of 0 values of the histogram. The result of relocating the first bin using this equation is represented in Fig. 2. (b). However you can see that there is unnatural blank interval between the first bin and the second bin in Fig. 2. (b). The quality of image can be improved by removing this interval. We moved $2(M[0]-c)$ bins of the histogram in Fig. 2. (a) by the equation 3.

$$M'[i] = M[i] - (M[0]-c)(2(M[0]-c)-i)/2(M[0]-c),\ i = 0 \ldots 2(M[0]-c)-1, \tag{3}$$

where $M[i]$ is the mapping value of the pixel value i of the full size image, $M'[i]$ is the rearranged mapping value, c is the center of mass in (2). The result of rearranging histogram is shown in Fig. 2. (c).



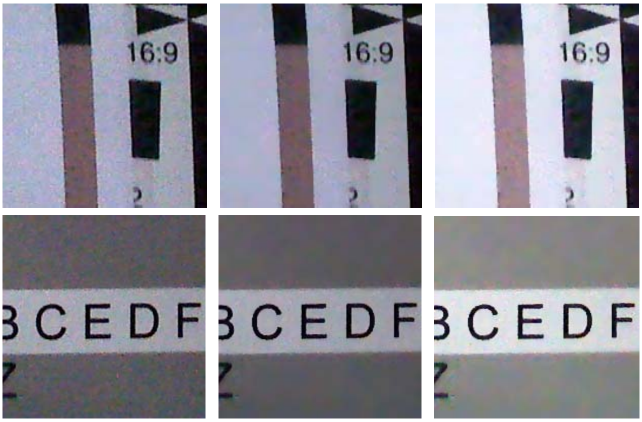
(a) (b) (c)

Fig. 2. (a) Histogram obtained with simple mapping. (b) Histogram correction using center of mass. (c) Histogram correction by rearrangement.

3 Experimental Results

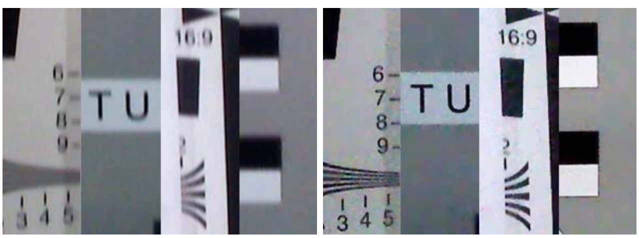
We experimented with the images supported by LG electronics. The images are taken with the camera moving vertically with 4 Hz. The full-size image is taken with the shutter speed of 70ms. In our test conditions with Pentium 4, 3.4 GHz and 3GB RAM, the total processing time was 1.47 sec.

Figure 8 shows the close up results for each procedure. Figure 3 (a) shows selected parts from the full-size image. There is a lot of noise in the image and the color of the image is dark. Figure 3 (b) represents denoised results for the corresponding parts. The noise is removed significantly while the edges are preserved. Figure 3 (c) shows the color-corrected results with histogram matching. After histogram matching, the color of the image becomes similar to the color of the preview image which has right color information.



(a) (b) (c)

Fig. 3. Close up of (a) Full-size image. (b) Denoised result. (c) Color-corrected result.



(a) (b)

Fig. 4. (a) Without anti-shaking. (b) With anti-shaking.

Figure 4 shows the results when the anti-shaking algorithm is not activated and activated respectively. We confirmed that the anti-shaking algorithm prevented the blur of the image effectively.

Table 2 shows a comparison between the blur-correction rates of the algorithms used in other devices and the blur-correction rate of our algorithm. The blur-correction rate is calculated as follows.

$$Blur-correction\ \ rate(\%) = (1 - \frac{l_B^{'}}{l_B}) \times 100 \tag{4}$$

where l_B' is the blurred length with the anti-shaking algorithm and l_B is the blurred length without the anti-shaking algorithm.

Table 2. Blur-correction rates with the test image

Model	Method A	Method B	Method C	Our method
Blurred length (*cm*)	0.186	0.127	0.154	0.074
Blur-correction rate (%)	8.6	42.5	45.5	65.2

* The exact names of camera models are replaced with alphabets for anonymity

4 Conclusions

In this paper, we have proposed the anti-shaking algorithm for the mobile phone camera which reduces the blur of the picture taken in dim light conditions. Our method removes the noise from the full-size image taken with fast shutter speed and corrects the color information of the full-size image to get the output which is close to the original image. The blur-correction rate was 65.2% and processing time was 14.7sec with our test image. Our algorithm also has no limitations on cost and size, so it is suitable for the mobile phone camera.

Acknowledgements

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute for Information Technology Advancement)" (IITA-2009-(C1090-0902-0017)) and in part by LG mobile handset R&D center.

References

1. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.: Removing camera shake from a single photograph. ACM Transactions on Graphics 25, 787–794 (2006)
2. Yuan, L., Sun, J., Quan, L., Shum, H.-Y.: Image deblurring with blurred/noisy image pairs. In: ACM SIGGRAPH, May 2007, vol. 26, pp. 1–10 (2007)
3. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Transactions on Graphics 27, 1–10 (2008)
4. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the ICCV, pp. 836–846 (1998)
5. Fecker, U., Barkowsky, M., Kaup, A.: Improving the prediction efficiency for multi-view video coding using histogram matching. In: Proceedings of the Picture Coding Symposium, April 2006, pp. 2–16 (2006)
6. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. Computer Vision and Pattern Recognition 2, 60–65 (2005)

Anti-shaking Algorithm for the Mobile Phone Camera in Dim Light Conditions

Jong-Ho Lee[1], In-Yong Shin[1], Hyun-Goo Lee[2], Tae-Yoon Kim[2], and Yo-Sung Ho[1]

[1]Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea
{jongho,siy0808,hoyo}@gist.ac.kr

[2]Mobile Handset R&D Center, LG Electronics Inc.
327-23 Gasan-dong, Gumcheon-gu, Seoul 153-802, Korea
{dennylee,jrock}@lge.com

Abstract. A picture is blurred when it is taken by the mobile phone camera in dim light conditions. There are some hardware approaches for commercial use to solve this problem, but they cost high and occupy a large space in the camera. A software approach, called image deblurring, takes much time to process and its result is not suitable for real applications because of the ringing effect. In this paper, we introduce a fast anti-shaking algorithm that is suitable for the mobile phone camera which uses a full-size image with fast shutter speed, and a preview image. The proposed algorithm is relatively free from both cost and size restraints, and shows good ability to prevent the blur of the image. The processing time was 1.47sec and showed the blur correction rate of 65.2%.

Key words: Anti-shaking, mobile phone camera

1 Introduction

An anti-shaking algorithm is for reducing the blur in the picture caused by the camera shake. If you take a picture in dim light conditions, it takes some time to obtain enough light. Camera shaking by hands during this time results in the blurred image.

There are many technologies in commercial use to reduce the blur of the picture. Canon's image stabilizer or Konica Minolta's anti-shake shift a lens group or charge coupled device (CCD) to compensate for camera's motion. Although these hardware approaches show good performances for preventing blur and generating sharp images, they cost high and occupy much space in the camera. Since cost and size are very critical issues for mobile phone cameras, this implementation is not suitable for the mobile phone camera.

There is also an image processing technique called image deblurring. Since image deblurring is an ill-posed problem where the PSF(point spread function) and the sharp image is estimated from the blurred image alone, it requires many iteration for restoring the sharp image so that it takes much time to process. It takes 10 minutes with MATLAB for the 128×128 image by the algorithm proposed by Fergus *et al.*[1]. The deblurring method using blurred image and noisy image pairs introduced by Yuan *et al* [2] shows quite good result, but requires too much memory space because it has to handle two images simultaneously. The deblurring method proposed by Shan *et al* reduces the ringing artifacts significantly, but it takes about 20 minutes to deblur 1600×1200 color image with C source code [3].

Our approach uses advantages of both software and hardware approaches. We reduce the blur with the fast shutter speed of the camera, and then, we get the output image close to the original sharp image by denoising and histogram matching with the aim of the preview image.

This paper is organized as follows. The proposed algorithm composed of input image acquisition, adaptive bilateral filtering and color correction is addressed in Section 2. We will show the experimental results in Section 3. Finally, conclusions are drawn in Section 4.

2 Proposed Anti-shaking Algorithm

Figure 1 shows the overview of the proposed algorithm. The full-size image is taken with fast shutter speed to reduce the blur. Since the full-size image obtained with fast shutter speed is noisy and dark, we reduce the noise and correct the color of the full-size image. With these procedures, we can obtain the sharp image which has reduced blur.

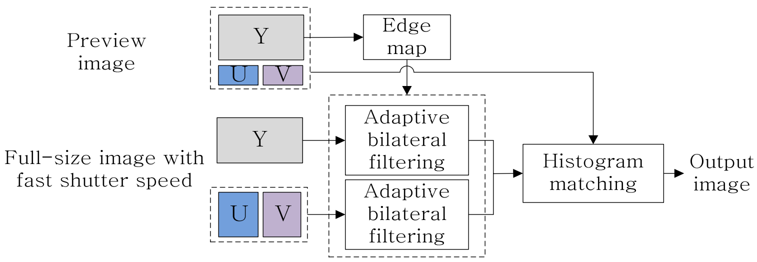


Fig. 1. Overview of the proposed algorithm.

2.1 Input Image Acquisition

If the anti-shaking is activated, 1600×1200 full-size image with YUV 4:2:2 format is taken with the fast shutter speed of 70ms. This full-size image is not blurred but noisy and dark. 320×240 preview image with YUV 4:2:0 format is also available in the camera. The preview image is captured during enough time, so it is not noisy and contains right color information.

2.2 Adaptive Bilateral Filtering

If the picture is taken with fast shutter speed, noise flow residing in the camera becomes prominent due to the low signal intensity, so we get the noisy image. We use bilateral filter to remove the noise which is relatively fast and maintains edge well. Other state-of-the-art denoising algorithms like the non local means algorithm show better performance than bilateral filter, but they are two slow [6].

$$F[I]_P = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|I_p - I_q|) I_q, \tag{1}$$

where $F[I]_P$ is the restored pixel at position *p*, G_{σ_s} is the Gaussian function whose variable is the distance between *p* and *q* and whose standard deviation is σ_s, G_{σ_r} is the Gaussian function whose variable is the difference between the pixel value at *p* and the pixel value at *q* and whose standard deviation is σ_r, I_q is the pixel value at *q*, *S* is the neighborhood of *p* enclosed by window whose center is located at *p* and W_p is the sum of $G_{\sigma_s} G_{\sigma_r}$ in the window.

The bilateral filter requires three parameters, window size, σ_s and σ_r. If we change these filtering parameters adaptively according to the features on the image, we can get better denoised result. For these adaptive filtering, we need edge information of the image, but it is very difficult to extract edge information from the noisy image since both noise and edge are high frequency components. Thus, we use preview image which is free from the noise. Since the preview image has small size and does not contain noise, the edge information can be obtained efficiently and rapidly with the simple Sobel algorithm.

For adaptive filtering, the image is classified into three regions using the edge map extracted from the Y component of the preview image. We slide 3×3window on the edge map, and three regions are defined according to the edge information in the window. Region 1 is the edge-free region, region 2 is the region which has strong edge and region 3 is the region which has many edges. We filter the image with different filtering parameters set according to the classified region. A pixel of the edge map corresponds to 5×5 block. Since the noise in the chrominance components of the full-size image is more unpleasant to the eye than that of the luminance components of the full-size image, we generally filter the chrominance parts with stronger filtering parameters. The filtering parameters according to the classified region are presented in the table 1.

Table 1. Filtering parameters according to region classification

components	Classified region	Window size	σ_s	σ_r
Y	region 1,2,3	3 x 3	1	50
U,V	Region 1	11 x 11	1	50
	Region 2	7 x 7	1	30
	Region 3	3 x 3	1	5

2.3 Histogram Matching

The full-size image taken with fast shutter speed is dark due to the shortage of light. We correct the pixel intensity of the full-size image referring to the preview image which has right color information using the histogram matching.

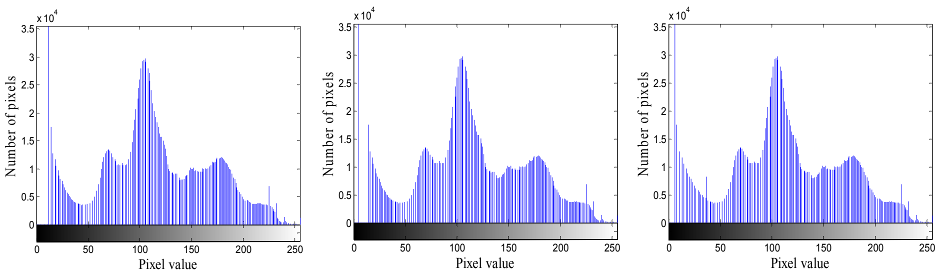
At first, we reduce the size of the full-size image to the size of preview image by down-sampling. After that, we get the cumulative histograms of two images, down-sampled full-size image and preview image. We get the mapping function from these cumulative histograms and substitute the pixel value of the full-size image with the values obtained from this mapping function. However, since all clipped pixel values below some threshold of the full-size image are mapped to the first active bin in the histogram like fig. 2. (a), this causes an unnatural image. We removed these artifacts using the Fecker's method [5]. If we define the mapping value of 0 as $M[0]$ and the histogram of pixel i of the preview image as $H[i]$, then the center of mass c of the values for the clipped interval can be calculated as follows.

$$c = \sum_{i=0}^{M[0]} iH[i] / \sum_{i=0}^{M[0]} H[i] \tag{2}$$

This value is used for the mapping of 0 values of the histogram. The result of relocating the first bin using this equation is represented in Fig. 2. (b). However you can see that there is unnatural blank interval between the first bin and the second bin in Fig. 2. (b). The quality of image can be improved by removing this interval. We moved $2(M[0]-c)$ bins of the histogram in Fig. 2. (a) by the equation 3.

$$M'[i] = M[i] - (M[0]-c)(2(M[0]-c)-i)/2(M[0]-c),\ i = 0 \ldots 2(M[0]-c)-1\ , \tag{3}$$

where $M[i]$ is the mapping value of the pixel value i of the full size image, $M'[i]$ is the rearranged mapping value, c is the center of mass in (2). The result of rearranging histogram is shown in Fig. 2. (c).



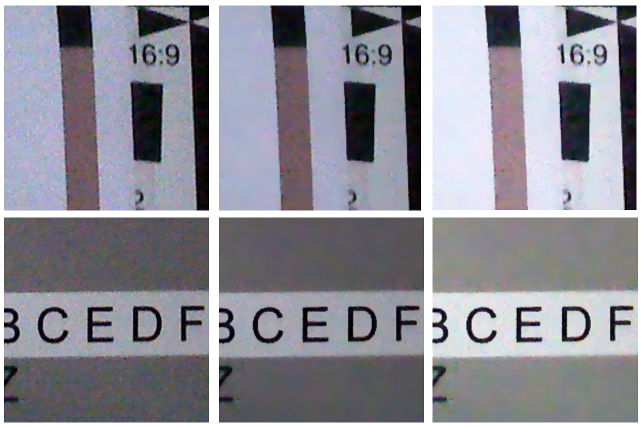
(a) (b) (c)

Fig. 2. (a) Histogram obtained with simple mapping. (b) Histogram correction using center of mass. (c) Histogram correction by rearrangement.

3 Experimental Results.

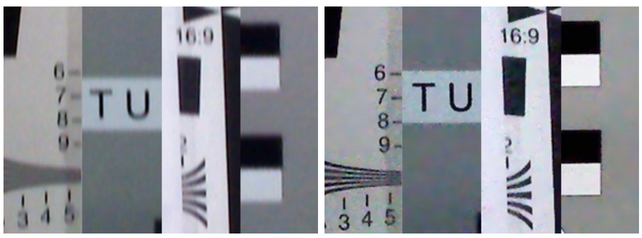
We experimented with the images supported by LG electronics. The images are taken with the camera moving vertically with 4 Hz. The full-size image is taken with the shutter speed of 70ms. In our test conditions with Pentium 4, 3.4 GHz and 3GB RAM, the total processing time was 1.47 sec.

Figure 8 shows the close up results for each procedure. Figure 3 (a) shows selected parts from the full-size image. There is a lot of noise in the image and the color of the image is dark. Figure 3 (b) represents denoised results for the corresponding parts. The noise is removed significantly while the edges are preserved. Figure 3 (c) shows the color-corrected results with histogram matching. After histogram matching, the color of the image becomes similar to the color of the preview image which has right color information.



(a) (b) (c)

Fig. 3. Close up of (a) Full-size image. (b) Denoised result. (c) Color-corrected result.



(a) (b)

Fig. 4. (a) Without anti-shaking. (b) With anti-shaking.

Figure 4 shows the results when the anti-shaking algorithm is not activated and activated respectively. We confirmed that the anti-shaking algorithm prevented the blur of the image effectively.

Table 2 shows a comparison between the blur-correction rates of the algorithms used in other devices and the blur-correction rate of our algorithm. The blur-correction rate is calculated as follows.

$$Blur-correction\ \ rate(\%) = (1 - \frac{l_B^{'}}{l_B}) \times 100 \qquad (4)$$

where l_B' is the blurred length with the anti-shaking algorithm and l_B is the blurred length without the anti-shaking algorithm.

Table 2. Blur-correction rates with the test image

Model	Method A	Method B	Method C	Our method
Blurred length (*cm*)	0.186	0.127	0.154	0.074
Blur-correction rate (%)	8.6	42.5	45.5	65.2

* The exact names of camera models are replaced with alphabets for anonymity

4 Conclusions

In this paper, we have proposed the anti-shaking algorithm for the mobile phone camera which reduces the blur of the picture taken in dim light conditions. Our method removes the noise from the full-size image taken with fast shutter speed and corrects the color information of the full-size image to get the output which is close to the original image. The blur-correction rate was 65.2% and processing time was 14.7sec with our test image. Our algorithm also has no limitations on cost and size, so it is suitable for the mobile phone camera.

Acknowledgements

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute for Information Technology Advancement)" (IITA-2009-(C1090-0902-0017)) and in part by LG mobile handset R&D center.

References

1. Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W.: Removing camera shake from a single photograph. ACM Transactions on Graphics, Vol. 25, July (2006) 787-794.
2. Yuan, L., Sun, J., Quan, L., and Shum, H.-Y.: Image deblurring with blurred/noisy image pairs. ACM SIGGRAPH, Vol. 26, May (2007) 1-10.
3. Shan, Q., Jia, J., and Agarwala, A.: High-quality motion deblurring from a single image. ACM Transactions on Graphics, Vol. 27, August (2008) 1-10.
4. Tomasi, C., and Manduchi, R.: Bilateral filtering for gray and color images. In Proceedings of the ICCV, (1998) 836-846.
5. Fecker, U., Barkowsky, M., and Kaup, A.: Improving the prediction efficiency for multi-view video coding using histogram matching, In Proceedings of the Picture Coding Symposium, April (2006) 2-16.
6. Buades, A., Coll, B., and Morel, J.-M.: A non-local algorithm for image denoising, Computer Vision and Pattern Recognition, Vol. 2, June (2005) 60-65.