LETTER

A Low-Cost Imaging Method to Avoid Hand Shake Blur for Cell Phone Cameras

Lin-bo LUO[†,††a)], *Nonmember and* **Jong-wha CHONG**[††b)], *Member*

SUMMARY In this letter, a novel imaging method to reduce the hand shake blur of a cell phone camera without using frame memory is proposed. The method improves the captured image in real time through the use of two additional preview images whose parameters can be calculated in advance and stored in a look-up table. The method does not require frame memory, and thus it can significantly reduce the chip size. The scheme is suitable for integration into a low-cost image sensor of a cell phone camera.
key words: *motion blur, hand shake blur, cell phone camera, frame memory*

1. Introduction

Hand-shaking usually introduces hand shake blur, which is detrimental to the performance of cell phone cameras. In the past, a deblurring algorithm that used a single blurred image was employed to obtain motion-blur-free images [1]. Recently, advances in image sensor have enabled fast image capture, and thus many algorithms now use multiple-exposure images to generate higher-quality images. For example, Lee et al. [2] enhanced a low-exposed image via tonal correction using a blurred but bright image. A combined method has also been employed to post-process a pair of multiple-exposure images [3]. Choi et al. [4] and Tico et al. [5] obtained high-quality images by fusing a short- and long-exposed image, while Tsuda et al. [6] used multiple-exposure images to restore a blurred image.

Compared to single image-based algorithms, multiple image-based algorithms can achieve better performance. However, algorithms that use multiple images must employ more than one frame memory when they are implemented by hardware. This will in turn lead to a significant increases in the chip area, especially for higher-resolution sensors. For example, in the TSMC 0.18 μm process, the frame memory for a full-sized image of a 3 M CMOS image sensor (CIS) (1/5 inch) will occupy an area of about 175 mm^2, which is about ten times the size of the CIS. Thus, frame memory should not be used for decreasing the chip size and cost.

Manuscript received February 20, 2012.
Manuscript revised March 5, 2012.
[†]The author is with the Faculty of Mechanical and Electronic Information, China University of Geosciences, Wuhan, 430074 China.
[††]The authors are with the Department of Electronics and Computer Engineering, Hanyang University, 17 Haengdang-Dong, Seongdong-Gu, Seoul, 133–791 Korea.
a) E-mail: luolb@hotmail.com
b) E-mail: jcchong@hanyang.ac.kr
DOI: 10.1587/transinf.E95.D.1702

In this letter, a low-cost imaging method to avoid hand shake blur that can be implemented by hardware without the need for frame memory is proposed. Two additional preview images are used to process a captured low-exposed image. Because the parameters of the two preview images can be calculated in advance, the processing of the captured image can be conducted in real time and frame memory can be removed.

2. Proposed Method

As shown in Fig. 1, the proposed imaging method uses three consecutive images: two preview images and a captured image. These images are exposed in long-short-long mode. The first preview image (P_1) was acquired under a long exposure time (T_{long}), while the second preview image (P_2) and first captured image (I_1) are acquired using a safe shutter speed (T_{short}). The brightness of P_1 is acceptable, although it is usually blurred because of the camera motion during the long exposure time. The motion blur of P_2 and I_1 is greatly reduced, but these images are low-exposed because of the short exposure time.

Considering that serious blur will occur with significant hand-shaking, we correct the low-exposed image I_1, but do not restore blurred image P_1, as shown in Fig. 1. First, we estimate the parameters of P_1 and P_2, and then we obtain the color mapping relationship between the short-exposed and long-exposed images. The mapping is subsequently stored in a look-up table (LUT). The LUT-based color correction method is then used to correct the short-exposed I_1. Because

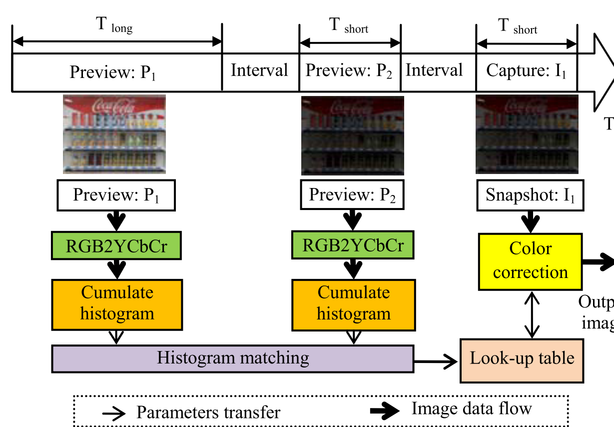

Fig. 1 Block diagram of the proposed system.

the final output image has the brightness of P_1 and the crisp edge and texture of I_1, it is a blur-free image of higher quality.

In contrast to the research conducted in the above-cited studies [2]–[6], we do not use P_1 to correct the color of I_1 directly. Such a step would require frame memory to buffer the data of I_1 when the color mapping function between P_1 and I_1 is estimated. Considering that P_2 and I_1 are temporally close, we use P_2 instead of I_1 to estimate the mapping function of the short- and long-exposed images. Because the parameters of P_1 and P_2 can be estimated in real time, the mapping can be obtained and stored it in a LUT before I_1 is processed. The color correction procedure for I_1 can be conducted in real time based on the pre-calculated LUT. Since both the mapping function estimation and color correction procedure are performed in real time, no frame memory is utilized, except for a small memory used to store the LUT.

2.1 Cumulative Histogram Matching

P_1 and P_2 are first transferred from RGB to the YCbCr color space. The histogram and cumulative histogram of the images are then calculated in the YCbCr color space. For image P_1, the histogram and cumulative histogram are calculated as follows:

$$h_{P_1}[L_i] = \frac{n_i}{W * H},\ i = 0, 1, \ldots, 255, \tag{1}$$

$$C_{P_1}[L_i] = \sum_{i=0}^{L_i} h_{P_1}[i], \tag{2}$$

where h_{P_1} is the histogram of P_1, L_i is the amplitude of the luminance signal (Y-channel of YCbCr), W and H are the width and height of P_1, respectively, n_i is the number of the pixels (i is the gray level), and C_{P_1} is the cumulative histogram of P_1. The cumulative histogram $C_{P_2}[L_j]$ of P_2 can be calculated in the same manner.

The image with the cumulative histogram of C_{P_2} can be mapped to an approximation of the desired levels of the image with the cumulative histogram of C_{P_1}. Through this approximation, we can determine the mapping function F between two images as follows [7]:

$$F[L_j] = L_i \text{ with } C_{P_1}[L_i] \le C_{P_2}[L_j] \le C_{P_1}[L_{i+1}] \tag{3}$$

As shown in Fig. 2, the mapping process consists of three steps: S_1, S_2, and S_3. First, through the L_j of P_2, we can find the corresponding $C_{P_2}[L_j]$. The approximation of $C_{P_2}[L_j]$ in $C_{P_1}[L_i]$ can then be determined. Finally, through the $C_{P_1}[L_i]$ of P_1, the mapped luminance L_i is obtained.

Because the captured image I_1 is temporally close to P_2, its histogram is very similar to that of P_2. Once the mapping F between the images P_2 and P_1 is determined, F can be applied to modify the histograms of the captured image I_1. The color correction is as follows,

$$L'_k = F[L_k],\ k = 0, 1, \ldots, 255, \tag{4}$$

where L_k is the original luminance level of I_1, and L'_k is the corrected luminance of the output image.

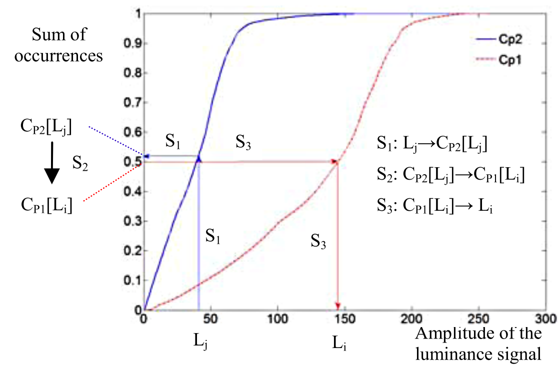

Fig. 2 Illustration of histogram matching based on the cumulative histograms of P_1 and P_2.

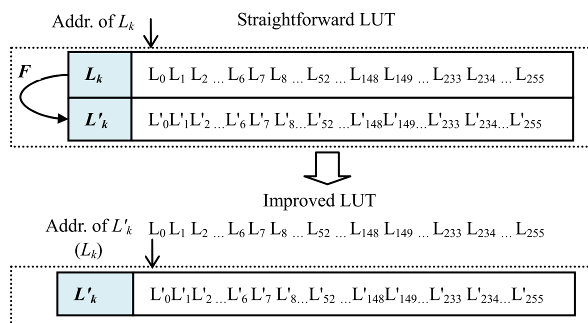

Fig. 3 Optimization of the LUT.

2.2 LUT-Based Histogram Matching

To implement the mapping of Eq. (4) in a straightforward manner, a 2-D LUT is constructed. One dimension of the 2-D LUT is L_k, while the other dimension is L'_k, as shown in the upper portion of Fig. 3. However, the straightforward LUT (SLUT) needs to store both the L_k and L'_k, and must also remember the mapping function F between L_k and L'_k. Thus, the SLUT is difficult to implement and uses more memory.

Here, we propose an improved LUT (ILUT). Considering that the maximum possible number of the luminance levels of both I_1 and the corrected I_1, i.e., L_k and L'_k, is 255, we used L_k as the address of L'_k directly, as shown in the bottom portion of Fig. 3. Thus, memory is conserved and methods to store the mapping function do not need to be considered. Furthermore, several optimization algorithms are proposed. First, the cumulative histograms are scaled by a scaling factor of 255. After the scaling, the ranges of C_{P_1} and C_{P_2} are changed from [0, 1] to [0, 255], and are denoted as C'_{P_1} and C'_{P_2}. When C'_{P_2} is mapped into C'_{P_1} and no mapping value exists for cumulative histogram C'_{P_1} at the integer location in [0, 255], the nearest rule is used to generate the output. Second, a strategy to address the problem of having n values of L_{P_1} mapping to the same C'_{P_1} was employed. For more than five L_{P_1} with the same C'_{P_1}, the middle value of the values is used as the final mapping; otherwise, if there

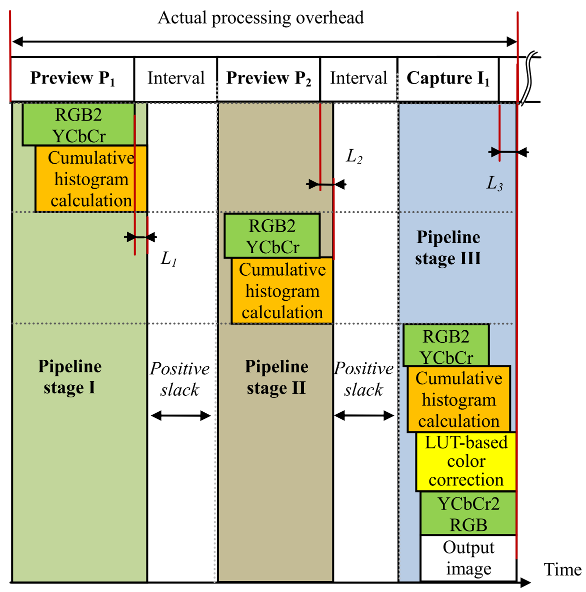

Fig. 4 Timing design of the proposed system.

are less than five L_{P_1} with the same C'_{P_1}, we use the maximum L_{P_1} as the final mapping.

After the optimizations, the SLUT is transformed into the ILUT, which is a 1-D table with the address of L_{I_1} and an output of $L'_{I_1}(L_{P_1})$. The histogram matching is simplified to retrieve a LUT in an on-chip memory.

2.3 Timing Design

As shown in Fig. 4, the pipelining framework mainly consists of three stages: the calculation of cumulative histogram of P_1, the calculation of cumulative histogram of P_2, and the LUT-based color correction of I_1.

In Fig. 4, we performed RGB2YCbCr modules via fixed-point calculation. The cumulative histograms of P_1 and P_2 are also calculated through fixed-point operation. Stages I and II are completed with the latencies of L_1 and L_2, respectively. Both of these latencies are smaller than the interval between the two images. In the third stage, the cumulative histogram calculation, RGB2YCbCr and YCbCr2RGB modules are used in the same manner with as in stage I. The LUT-based color correction is simplified to on-chip memory access since the ILUT is used. The latency of this stage, L_3, was also smaller than the interval between the two images and did not affect real time implementation.

Because we use the parameters of P_1 and P_2 to process I_1, the proposed method does not require a frame memory to buffer I_1. Furthermore, because both stages I and II can be conducted in real time, no frame memory is needed for this system.

3. Simulation and Results

To verify the similarity between the histogram of P_2 and I_1, we took 10 consecutive frames by a hand-held camera (shutter speed of 1/64 s, in a dimly lit office) and calculated their histograms. As shown in Fig. 5, almost all of the histograms of the 10 frames were very similar.

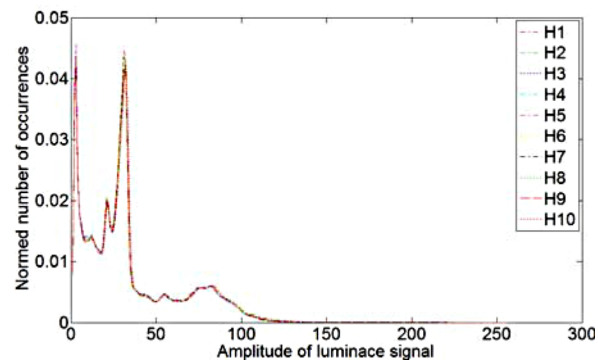

Fig. 5 Histograms of 10 consecutive frames.

One test chart and two different scenes (a folk museum and a vending machine) were used to validate the color correction process; the results are illustrated in Fig. 6. For subjective evaluation, Figs. 6 (m)–(o) shows that the resulting images of our method are as bright as Fig. 6 (a)–(c) and are as crisp as Fig. 6 (j)–(l).

For an objective evaluation, the CIELAB color distortion metric was used to determine the the efficiency of the LUT-based color correction algorithm. This metric may be expressed as follows:

$$\Delta E = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\sqrt{\Delta L_{ij}^{*2} + \Delta a_{ij}^{*2} + \Delta b_{ij}^{*2}}, \qquad (5)$$

where ΔL^*, Δa^*, and Δb^* are the differences in L*a*b* color space, and W and H are the width and height of the test image, respectively. According to the standard, the distortion is barely perceptible when ΔE is smaller than 3, while it is perceptible but acceptable when ΔE ranges from 3–6. In our simulation, the ΔE between the output images (Fig. 6 (m)–(o)) and the reference images (Fig. 6 (a)–(c)) are 0.23, 0.89, and 0.35, respectively.

To evaluate the degree of blurring, a modulation transfer function (MTF) [8] was employed as a metric. Both horizontal and vertical edges were used to estimate the MTF; the results are shown in Table 1. A comparison to the values of MTF50 revealed that the difference between the corrected images and reference images is very small, while the difference between the blurred images is great. These findings show that the proposed algorithm greatly decreases the blur level of a blurred image much and yields results that are similar to the reference images.

Different blur reduction schemes and their manner of implementation, the numbers of frames, and the final memory usage are listed in Table 2. For conventional multiple-exposure-based algorithms, one or more frame memories are needed if the algorithms are implemented by hardware in real time. Because the latencies of L_1, L_2, and L_3 are only a couple of clocks and the interval between the two images is longer than 1 ms, the proposed method can be im-

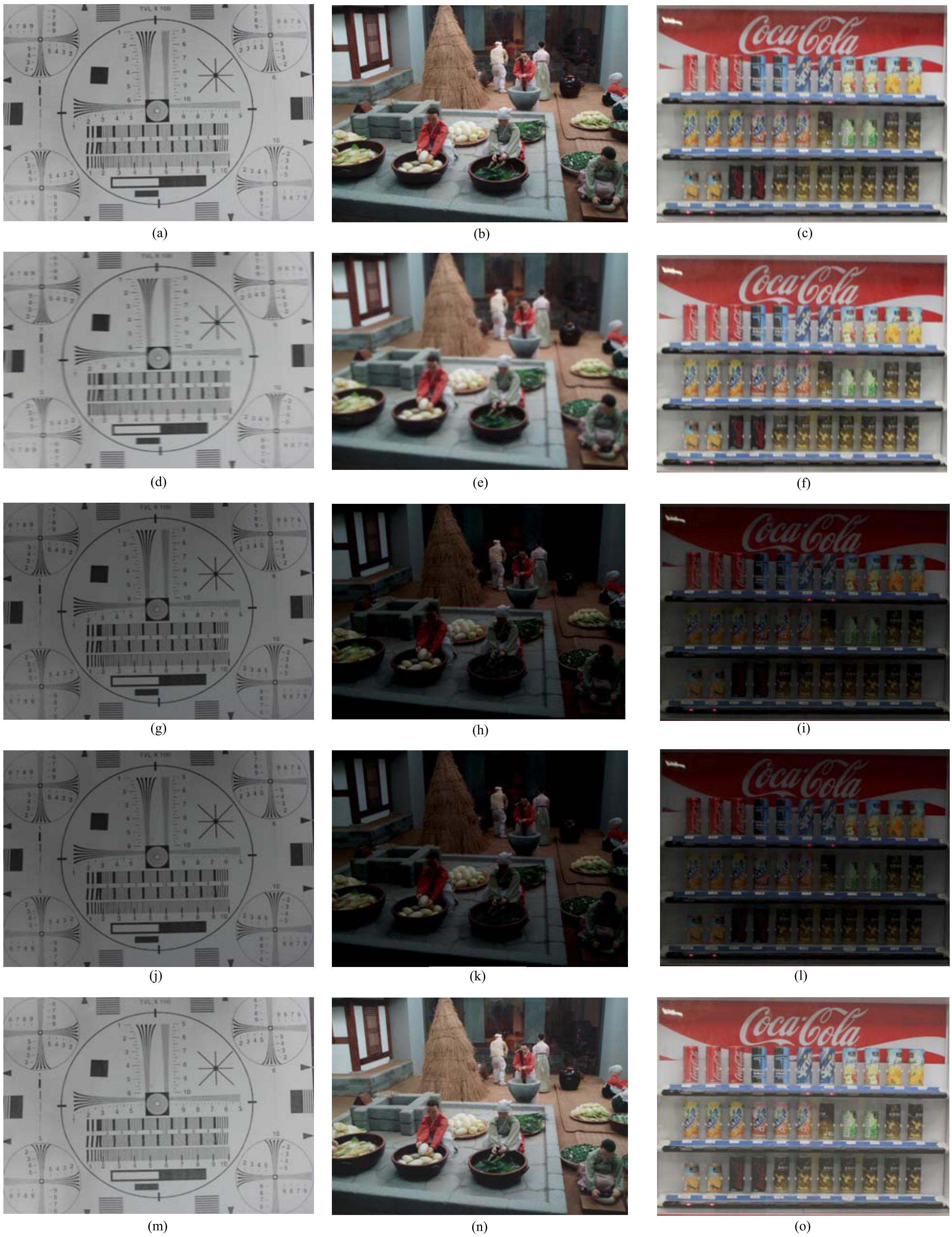

Fig. 6 Test images and the results of LUT-based color correction: (a)–(c) are reference images; (d)–(f) are blurred images previewed with long exposure time; (g)–(i) are low-exposed images previewed with short exposure time; (j)–(l) are low-exposed images captured with short exposure time; and (m)–(o) are the output images obtained by our proposed method.

Table 1 MTF50† (c/p) of reference, blurred, and corrected images.

	Test chart		Folk museum		Vending machine	
	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical
Reference	0.175	0.155	0.200	0.160	0.270	0.127
Blurred	0.047	0.045	0.055	0.055	0.025	0.102
Corrected	0.157	0.125	0.152	0.162	0.267	0.157

†n of MTF50 means: a 50% contrast spatial frequency equals n cycles per pixel.

Table 2 Comparison of the different algorithms.

Algorithm	[2]	[3]	[4]	[5]	[6]	Proposed
Conducted by	S/W	S/W& H/W	S/W	S/W	S/W	H/W
Method	Color correction	Combined	Image fusion	De-convolution	De-blurring	Color correction
Process time	14.7 s	1.4 s /mpix	Not real time	Not real time	Not real time	Real time
Frames used	2	2	≥2	2	2	3
Frame memory	1	4	≥2	2	1	0

plemented by hardware in real time and requires no frame memory. The only memory usage is for the ILUT, which occupied 256 bytes. However, this size is negligible and does not significantly increase the chip area.

4. Conclusions

A novel and efficient imaging method that can be implemented by hardware without the use of frame memory is proposed. The proposed scheme is suitable for low-budget cameras that have constraints with respect to chip area and cost.

Acknowledgments

This work was supported by Hynix Semiconductor Inc., Brain Korea 21 Project, MKE Korea, and the ITRC support program supervised by the NIPA (NIPA-2011-C1090-1100-0010). IDEC provides research facilities for this study.

References

[1] D. Huang, N. Fujiyama, and S. Sugimoto, "Blind image identification and restoration for noisy blurred images based on discrete sine transform," IEICE Trans. Inf. & Syst., vol.E86-D, no.4, pp.727–735, April 2003.

[2] J.-H. Lee, I.-Y. Shin, H.-G. Lee, T.-Y. Kim, and Y.-S. Ho, "Anti-shaking algorithm for the mobile phone camera in dim light conditions," Pacific Rim Conf. Multim.: Advan. in Multim. Infor. Proc., pp.968–973, 2009.

[3] White paper, "High quality photography on mobile phones," www.almalence.com, 2008.

[4] B.-D. Choi, S.-W. Jung, and S.-J. Ko, "Motion-blur-free camera system splitting exposure time," IEEE Trans. Consum. Electron., vol.54, no.3, pp.981–986, Aug. 2008.

[5] M. Tico, N. Gelfand, and K. Pulli, "Motion-blur-free exposure fusion," Proc. Int. Conf. Image Processing, pp.3321–3324, Sept. 2010.

[6] Y. Tsuda, H. Hatanaka, S. Fukumoto, M. Ueda, and K. Chihara, "Noise-robust image deblurring by blending regular- and short-exposure images," Proc. SPIE 7876, 78760S, 2011.

[7] U. Fecker, M. Barkowsky, and A. Kaup, "Histogram-based prefiltering for luminance and chrominance compensation of multiview video," IEEE Trans. Circuits Syst. Video Technol., vol.18, no.9, pp.1258–1267, Sept. 2008.

[8] P.B. Greer and T.V. Doorn, "Evaluation of an algorithm for the assessment of the MTF using an edge method," Med Phys., vol.27, no.9, pp.2048–2059, Sept. 2000.