LOCAL ADAPTIVE TONE MAPPING WITH COMPOSITE MULTIPLE GAMMA FUNCTIONS

Sohsuke Shimoyama, Masaki Igarashi, Masayuki Ikebe, Junich Motohisa

Hokkaido University
Graduate School of Information Science and Technology
Kita 14, Nishi 9, Kita-ku, Sapporo 060-0814, Japan

ABSTRACT

We describe a high-speed method of correcting and compressing the dynamic range of images that can be operated intuitively and naturally. Adaptive operations are conducted for shadow, middle, and highlight tones in the local areas of images. Although natural image processing can be achieved with default parameters, we can set the parameters for each of these three tones individually and intuitively. We attained two million pixels per 0.3 sec of operation and real-time VGA movie processing without the need for SIMD instruction, multi-thread operation, or additional hardware by using the Athlon64 X2 3800+ CPU.

Index Terms— HDR compression, composite multiple gamma functions, intuitive image correction, local histogram equalization (LHE)

1. INTRODUCTION

Tone mapping is a key technique in methods of correcting and compressing the dynamic range of images. Moreover, the technique is also used for tuning contrast such as compensation for underexposed images.

According to Retinex theory [1], the human retina does not tend to be affected by illumination. Numerous methods derived from Retinex theory have been proposed. However, these methods are too complex for real-time processing and cannot completely eliminate artifacts at the edges of images.

Histogram Equalization (HE) is one means of non-linear tone mapping and is used to enhance contrast in images. HE can be classified into two types: Global Histogram Equalization (GHE) and Local Histogram Equalization (LHE). LHE needs large amounts of calculation. Both GHE and LHE have problems with over-enhancement.

Apical's Iridix®[2] has recently attracted a great deal of attention of the LHE-like methods. Cameras produced by Sony and Nikon presently use this technique, where the mapping function is reconstructed from local histograms transformed orthogonally. However, as the mapping function may generate ripples, the output image needs to be blended with the original image. Moreover, 2D low-pass filtering is required in the orthogonal transform. Therefore, it is necessary to use additional hardware to apply Iridix®to video-image processing.

Here, we propose an approach to generating a tone-mapping function by using composite multiple gamma functions derived from local histograms. The multiple gamma functions represent the shadow, middle, and high-light tones of local-image areas. We can intuitively control the parameters of these three tones and suppress over-enhancement of images with LHE by using these multiple gamma functions. Moreover, we achieved two million pixels per 0.3 sec of operation and real-time movie correction (resolution: VGA, frame rate of 30fps) without the need for dedicated instructions or additional hardware by calculating the local histogram independent of the pixel-box size and using a look-up table for functions.

2. GENERATING FUNCTION FOR TONE MAPPING

2.1. Proposed tone mapping

2.1.1. Blending multiple gamma functions

Here, we consider a way of imitating an accumulative histogram by using gamma correction. Our method is based on a local histogram with four bins. It adjusts the gamma function so that it corresponds to the accumulative local histogram on the mapping tone surrounding a pivot point. To accomplish this, we assumed that a curve of the gamma function passed through a point near the pivot point of the accumulative histogram.

Figure 1 shows three gamma functions defined by three pivot points on the accumulative local histogram. G_s is the gamma function for the shadow tone, and G_m and G_h are those for the middle and highlight tones. It is necessary for each gamma function to only be allocated on a corresponding tone. We blended the three gamma functions through the use of blending functions to run such partial transformations.

simoyama@impulse.ist.hokudai.ac.jp

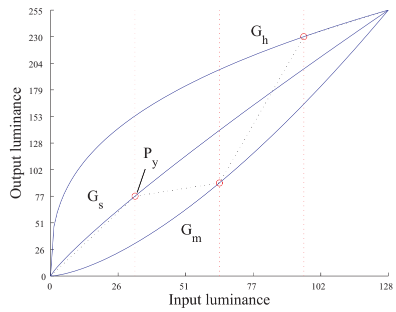

Fig. 1. Gamma functions

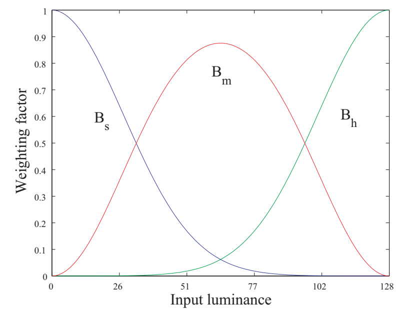

Fig. 2. Blending functions

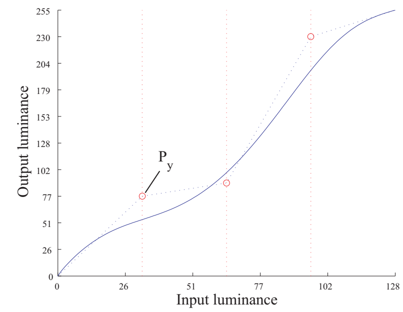

Fig. 3. Mapping function with our method

These functions are Gaussian windows. Figure 2 shows the three blending functions. B_s and B_h denote Gaussian functions that weight the shadow and highlight tones respectively. B_m is a blending function that weights the middle tones to ensure that these blending functions are consistent for synthesizing functions. The final mapping function using our method and is represented as:

$$Y_{out} = G_s(Y_{in}) \times B_s + G_m(Y_{in}) \times B_m + G_h(Y_{in}) \times B_h \quad (1)$$

Figure 3 shows the mapping function with our method; we can see that its curve smoothly follows that of the accumulative local histogram.

2.1.2. Restricted parameter of gamma function

Here, we discuss how to controlled the contrast and luminance of the three tones. We limited the variable range of the pivot point, p_y, to achieve optimum control. The restricted size of each bin affects the accumulation of bins when calculating accumulative histograms in LHE with restrictions. Therefore, some restriction that takes the entire accumulation of local histogram bins into account is required.

The gamma function range is guaranteed to be (x, y) = (0, 0) to (255, 255). Because blending is done by using a Gaussian window with our method, the range of the composite functions is also guaranteed to be (x, y) = (0, 0) to (255, 255). This also takes into account the fact that no interference of any function needs to be considered. Therefore, we can control each tone independently and obtain a continuous luminance-mapping function with only a few local histogram bins.

It is necessary to set the maximum and minimum values for a limited range to identify the restriction parameters. The method of limitation is to linearly compress the pivot point, p_y, and this is represented by:

$$pl_y = 255 \times \left(min + (max - min) \times \frac{p_y}{255} \right) \quad (2)$$

,where pl_y is the modified p_y and the max and min values denote the maximum and minimum values representing the limited variability of normalized p_y. The default values for min and max are set as:

$$max_s = 0.4, \quad max_m = 0.7 \quad , max_h = 0.9$$
$$min_s = 0.05, \quad min_m = 0.4 \quad , min_h = 0.7$$

,where the suffixes s, m, and h denote the shadow, middle, and highlight tones in which p_y exists. The restriction parameters of each tone represent the luminance between the upper and lower limits and their difference represents spatially contrast. Therefore, we can intuitively control the contrast and luminance of all tones.

2.2. Fast calculation of local histogram

The conventional method (Fig. 4) uses overlapping pixels between the current pixel block and the previous block (left of current block). This method requires $O(N)$ steps. Non-overlapping blocks accessed with interpolation are used (Fig. 5)[3] for high-speed processing. Although the non-overlapping approach can process quickly, it causes block noise. A method that uses partially overlapping blocks is therefore proposed[4] to supress block noise. However, the block noise cannot be completely suppressed and the calculation time is increased.

Here, we focus on pixel access with the "Box-filtering Method"[5]. The method achieves 2D-moving-average-filter processing in $O(1)$ time with only 2 pixels to access each pixel block. A $O(1)$-median-filter has recently been proposed with this pixel access[6].

We improved the "Box-filtering method" and here propose one pixel-access to calculate the local histogram in $O(1)$ time. Our method is outlined in Fig. 6. Checkerwise pixel access is carried out. We use a line buffer to store the column local histograms of the previous line, and calculate the column histograms of the current line from the histograms of the previous line. Let us consider a case of moving a pixel block from the left of one pixel to the next. The column histogram with N pixels is updated by adding one new pixel below it. After the next step, the neighboring column histogram

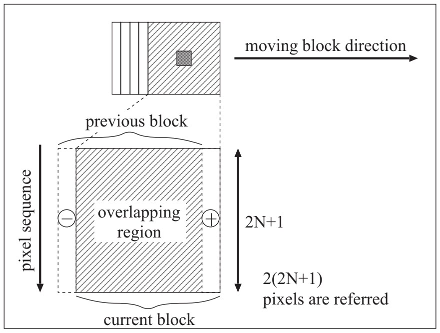

Fig. 4. Conventional method

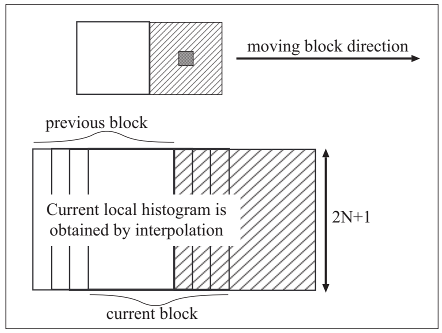

Fig. 5. Fast conventional method

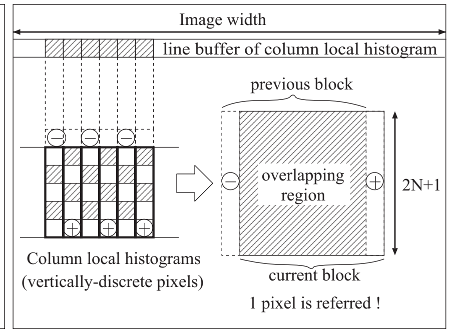

Fig. 6. Our method

with $N + 1$ pixels is updated by subtracting its topmost pixel. The column histogram is calculated using only one addition or subtraction.

We can also obtain the current target histogram by adding and subtracting one-column histogram. Finally, our local histogram is calculated by only referencing one pixel and by performing three arithmetic operations. These operations are processed in time $O(1)$ and are independent of the size of the pixel block.

We implemented our algorithm in C++ and measured the processing time for local-histogram equalization on our PC (CPU: Athlon64 X2 3800+). Table 1 compares this processing time with fast conventional methods. The dependence on the size of the pixel block and the number of bins are listed in the table. Non-overlapping processing is the faster method of calculating histograms. However, the methods require numerous other calculations for interpolation. Most of the calculations were done for the histograms because our method is based on pixel-wise operation without interpolation. Therefore, the time for histogram equalization decreased significantly as the number of bins decreased. When four bins were present, our method was four times faster than non-overlapping processing.

Reducing the number of bins had both advantages and one disadvantage. The advantages were improved processing speed and reduced over-enhancement. The disadvantage was that artifacts may have been generated in particular cases. If there is a pixel group in the boundary of the bins when the histogram is equalized, bright pixels on the boundary will be mapped from lower luminance and dark pixels under the boundary will be mapped from higher luminance. Therefore, there is the possibility of generating stripe artifacts. In particular, images with gradation on the bin boundary may generate these artifacts.

However, artifacts can be supressed by severely restricting parameters $max_{s,m,h}$ and $min_{s,m,h}$. The contrast of corresponding tones in the converted image is almost the same as that in the original image by restricting these parameters.

Table 1. Processing time of histogram calculation and histogram equalization (msec)

blocksize	64	128	256
nonoverlapped method	62	62	62
equalization [3]	(767)	(768)	(768)
pixel-wise proposed	1114	1115	1136
method equalization	(1902)	(1904)	(1922)

number of bins	4	16	64	256
nonoverlapped method	57	57	59	62
equalization [3]	(753)	(759)	(764)	(768)
pixel-wise proposed	39	93	282	1136
method equalization	(185)	(267)	(602)	(1922)

histogram calculation time and (histogram equalization time)
Image size: 1920x1080, Athlon™64 X2 Processor 3800+

3. IMAGE PROCESSING OF OUR METHOD

The results from calculations are shown in Figs. 7-10 to evaluate the proposed method. Figure. 7 compares the image quality with the conventional and proposed methods. The pixel-box size depends on the resolution of the images. For example, the box size was set to 128 $\times$ 128 for SXGA. The left image is the original with optimum gamma correction, and the center-left image has been modified by global HE. The center-right image was modified by using Apical's Iridix ®[2] and the images at right were processed with the proposed method. The parameter $\alpha = 0.5$, which is the default blending ratio with the original image, is defined in Chesnokov[2]. Compared to Iridix and the global HE, our proposed method makes the girl's face clearer. Figure 8 compares DR compression image and that with our proposed method. Compared to logarithmic compression, the shape of the whole building can be represented with our method. Figure 9 shows how the tone was controlled with our method. Only the contrast of the clouds has changed by controlling restriction parameters of the middle and highlight tones. Figure 10 shows the control of artifacts caused by reducing the number of bins. We could obtain natural images by appropriately

Fig. 7. Comparison with other methods. Left: original image (1296x972). Center left: GHE. Center right: Apical's Iridix ®($\alpha = 0.5$). Right: Proposed method ([$min_s : max_s, min_m : max_m, min_h : max_h$]= default)

Fig. 8. DR compression. Left: logarithmic compression (949x708). Right: proposed method (default)

Fig. 9. Control of highlight and middle tones. Left: original image (640x480). Center: proposed method ([0.1:0.5, 0.3:0.6, 0.6:0.9]). Right: ([0.1:0.5, 0.65:0.7, 0.85:0.9])

adjusting the restriction parameters.

4. CONCLUSION

We proposed a method of correcting and compressing the dynamic range of images that was intuitive and natural in operation. Adaptive operations were done for the shadow, middle, and highlight tones in local areas of images. We demonstrated that our method of tone mapping greatly improved image quality. This method is also high-speed and we achieved two million pixels per 0.3 sec of operation and real-time VGA movie processing.

Acknowledgements

The authors would like to extend their appreciation to Messers. Kensuke Yamaoka and Yuji Osumi of Dai Nippon Printing

Fig. 10. Effect of few bins. Left: original image (1205x803). Right: proposed method. upper artifact part (default), and lower corrected part ([0.3:0.4, 0.55:0.5, 0.7:0.9])

Co., Ltd. for their helpful guidance and insightful discussions regarding the development of the processing system we used in this research.

5. REFERENCES

[1] E. H. Land, "The retinex theory of color vision.," *Scientific American*, vol. 237, no. 6, pp. 108–128, December 1977.

[2] Vyacheslav Chesnokov, "Image enhancement methods and apparatus therefore," United States Patent, November 2007, US 7302110 B2.

[3] Jie Zhao and Shawmin Lei, "Automatic digital image enhancement for dark pictures," *ICASSP 2006 Proceedings.*, vol. 2, pp. 953–956, 2006.

[4] Joung-Youn Kim, Lee-Sup Kim, and Seung-Ho Hwang, "An advanced contrast enhancement using partially overlapped sub-block histogram equalization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 4, pp. 475–484, 2001.

[5] M. J. McDonnell, "Box-filtering techniques," *Computer Graphics and Image Processing*, vol. 17, pp. 65–70, 1981.

[6] S. Perreault and P. Hebert, "Median filtering in constant time," *Image Processing, IEEE Transactions on*, vol. 16, no. 9, pp. 2389–2394, 2007.