An Advanced Contrast Enhancement Using Partially Overlapped Sub-Block Histogram Equalization

Joung-Youn Kim, Lee-Sup Kim, and Seung-Ho Hwang
Dept. of Electrical Engineering Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon, Korea

ABSTRACT

In this paper, an advanced histogram equalization algorithm for contrast enhancement is presented. Histogram equalization is the most popular algorithm for contrast enhancement due to its effectiveness and simplicity. Global histogram equalization is simple and fast, but its contrast enhancement power is relatively low. Local histogram equalization, on the other hand, can enhance overall contrast more effectively, but the computational complexity is very high due to its fully overlapped sub-blocks.

For high contrast and simple calculation, a low pass filter type mask is proposed. The low pass filter type mask is realized by partially overlapped sub-block histogram equalization (*POSHE*). With the proposed method, the computation overhead is reduced by a factor of about one hundred compared to that of local histogram equalization while still achieving high contrast.

1. INTRODUCTION

Enhancing the contrast of images is one of the major issues in image processing, especially backlit images. Contrast enhancement can be achieved by stretching the dynamic range of important objects in an image. There are many algorithms for contrast enhancement and among these, histogram equalization [1 - 5] is the most common method used due to its simplicity and effectiveness.

Histogram equalization can be categorized into two methods: global histogram equalization and local histogram equalization. Global histogram equalization uses the histogram information of the whole input image as its transformation function. This transformation function stretches the contrast of the high histogram region and compresses the contrast of the low histogram region. This global histogram equalization method is simple and powerful, but it cannot adapt to local brightness features of the input image. This fact limits the contrast-stretching ratio and causes significant contrast losses in the background and other small regions. To overcome this limitation, a local histogram equalization method has been developed, which can also be termed block-overlapped histogram equalization [1, 5]. This method allows each pixel to adapt to its neighboring region, so that high contrast can be obtained for all locations in the image. However, since local histogram equalization must be performed for all pixels in the entire image frame, the computational complexity is very high.

To reduce this computation and obtain the advantage of local adaptability of block-overlapped histogram equalization, Partially Overlapped Sub-block Histogram Equalization (*POSHE*) is proposed in this paper. Using *POSHE*, the contrast of the input image can be enhanced at a similar rate to block-overlapped histogram equalization while the complexity can be reduced considerably and any blocking effects eliminated.

2. CONVENTIONAL HISTOGRAM EQUALIZATION TECHNIQUES

Let input images be composed of L discrete levels. The transformation function can be written as follows.

$$s_k = T(r_k), \qquad k = 0,1,\ldots,L-1 \tag{1}$$

which produces level s_k as the output of the normalized original image level r_k. In histogram equalization [1, 2], the transformation function *T(r)* is given by the following relation,

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, 0 \le r_k \le 1 \text{ and } k = 0,1,\ldots,L-1 \tag{2}$$

where $p(r_j)$ is the probability density function (*pdf*) of the input image level j, n is the total number of pixels in the input image, and n_j is the input pixel count of level j. Hence, the transformation function *T(r)* represents the cumulative distribution function (*cdf*) of the input image. When an image is transformed by this function, the contrast of large objects increases, while the contrast of small objects such as background decreases because their dynamic ranges are increased or decreased respectively. This result is acceptable for general images such as photographs, but if we want a high contrast in the background, this result is not so desirable.

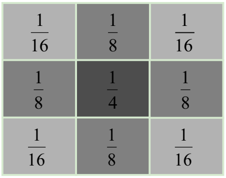
Local histogram equalization, so-called block-overlapped histogram equalization [1, 5], can obtain overall contrast enhancement regardless of location in the input image. Block-overlapped histogram equalization defines a sub-block and retrieves its histogram information. Then, a histogram equalization is performed for the center pixel of the sub-block using the *cdf* of that sub-block. Next, the sub-block is moved by one pixel and sub-block histogram equalization is repeated until the end of input image is reached. Since each pixel is histogram equalized using its neighboring sub-block, the result adapts to the local light condition well. So the local contrast is enhanced maximally. But, its computational complexity is very high because the local histogram equalization must be performed for each pixel.

3. PARTIALLY OVERLAPPED SUB-BLOCK HISTOGRAM EQUALIZATION (*POSHE*)

In order to make the histogram equalization locally adaptive for higher contrast, and reduce the computation complexity, non-overlapped sub-block histogram equalization is essential. In this method, all pixels in each sub-block are histogram equalized using the sub-block's histogram and these sub-blocks are not overlapped with adjacent sub-blocks, so the computation complexity is reduced considerably.

However, this non-overlapped method cannot avoid a blocking effect. Blocking effects occur due to shape differences between histogram equalization functions of neighboring sub-blocks. In the input image, adjacent sub-blocks have similar brightness and the gray levels of neighboring pixels at the boundaries of the sub-blocks change gradually. However, after non-overlapped sub-block histogram equalization, their brightness values are changed by each sub-block's histogram, which have shape differences. As a result, the difference in brightness after transformation between adjacent sub-blocks becomes large, so that sudden level change can occur at the boundaries. On the other hand, in block-overlapped histogram equalization, the transformation functions for each pixel are obtained from mostly overlapped neighboring sub-blocks. Therefore, the shape difference of the transformation functions are very small and the blocking effect can be ignored.

So, to eliminate the blocking effect, the shape difference between neighboring sub-blocks must be reduced. For this, the weighted sum of neighboring sub-blocks' histograms can be used for generation of the transformation function for the current sub-block. Using a 3x3-mask, as shown in Figure.2, the central sub-block's transformation function is obtained from the masked histogram of itself and its 8 neighboring sub-blocks. This mask resembles a image low pass filter (*LPF*), and its operation is in fact similar to that of an *LPF*. This low pass filtering effect for sub-block histograms reduces the transformation function difference of neighboring sub-blocks, resulting in a reduction of the blocking effect. As mask size is increased, blocking effects can be decreased with increasing the computation complexity.



$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

Figure.2. *pdf* of *POSHE*

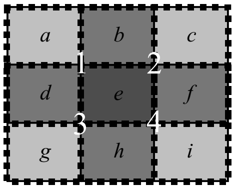


Figure.3. Four partially overlapped sub-blocks

This *LPF* can be realized by moving sub-blocks using a partially overlapped scheme and performing histogram equalization for those sub-blocks. To make a 3x3 mask, as shown in Fig. 2, a sub-block is moved by a half of the sub-block size to overlap the next sub-block by half, and histogram equalization is performed for all pixels in the sub-block. The procedures are as follows.

- procedure 1 : Define an $M \times N$ sized output image array for an $M \times N$ input image and set all values to zero.
- procedure 2 : Assign an $m \times n$ sub-block. The sub-block size that is equal to the quotient of the input image size divided by a multiple of 2 is used. This reduces most division into shift. Assign the sub-block origin using the input image origin.
- procedure 3 : Perform histogram equalization for the current sub-block. The histogram equalization is performed over the whole of the sub-blocks and the histogram equalized results are accumulated to the output image array.
- procedure 4 : Increase the horizontal-coordinate of the sub-block origin by the horizontal step size. The step size that takes the quotient of the sub-block size divided by a multiple of 2 is normally used. Repeat the previous steps. When the horizontal coordinate equals the horizontal input image size, increase the vertical coordinate of the sub-block origin by the vertical step size and repeat horizontal *POSHE*. Repeat these steps until *POSHE* covers the whole of the input image plane.
- procedure 5 : Divide each pixels in the output image array by their histogram equalization frequency.
- procedure 6 : If a small blocking effect is generated at the sub-block boundaries, eliminate it with a blocking effect reduction filter.

The image plane for 3 × 3 mask is illustrated in Figure.4. Through simple formula expansion illustrated in the following section, it can be seen that this procedure results in a 3x3 *LPF* mask. By decreasing the moving step size, the mask size can be increased. Therefore, mask size can be changed easily by manipulating the step size moved. Because sub-blocks are partially overlapped, this method is named Partially Overlapped Sub-block Histogram Equalization (*POSHE*). It will be shown in the following section that the computation overhead for a 15 × 15-sized mask *POSHE* is less than 1% of block-overlapped histogram equalization.

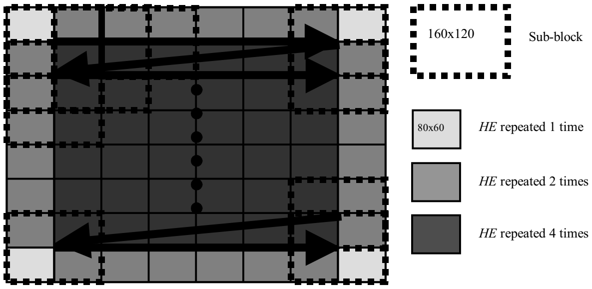


Figure.4. Image plane for *POSHE*

In procedure 6, a blocking effect reduction filter (*BERF*) is used. This is a simple filter to eliminate small blocking effects that sometimes occur. With the low pass filtering effect, the blocking effect is almost completely eradicated. But at some sub-block boundaries, a few gray level discontinuities may be generated. *BERF* is a protection for small level difference is needed.

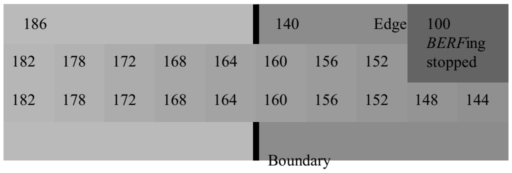


Figure.5. Example of *BERF*

The *BERF* procedures are as follows. First, when a blocking effect is detected at a sub-block boundary, an average of the two adjacent boundary pixels is calculated. Second, in a perpendicular direction to the boundary, pixel values are replaced with an increasing/decreasing value starting from the average luminance. This filtering is terminated when the termination condition is met. The increasing/decreasing step size is selected such that the discontinuity cannot be distinguished. An example of *BERF* is shown in Figure.5. The numbers indicate the pixel luminance.

4. ANALYSIS OF *POSHE*

A histogram equalization function is the *cdf* of the input image as shown in (1) and (2). Let's suppose a region is composed of four partially overlapped sub-blocks for a 3x3 mask. The region is divided into nine sub-regions in Figure.3. There are four sub-blocks, 1, 2, 3, and 4, and nine sub-regions, *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*, and *i*.

The sub-region *e* is "*POSHE*-ed" by four sub-blocks 1, 2, 3, 4. The peripheral sub-regions *a*, *b*, *c*, *d*, *f*, *g*, *h*, and *i* affect the histogram equalization of the target sub-region *e*. Let the histogram equalization functions of each sub-block be $T_1(r_k)$, $T_2(r_k)$, $T_3(r_k)$, and $T_4(r_k)$, respectively. Then, the *POSHE* function of the sub-region *e* is written as,

$$s_k{}^e = \frac{1}{4}\left[T_1(r_k{}^e) + T_2(r_k{}^e) + T_3(r_k^e) + T_4(r_k^e)\right] \quad . \tag{3}$$

And each histogram equalization function can be written as

$$T_1(r_k^e) = \sum_{j=0}^{k} p_1(r_j), \quad p_1(r_j) = \frac{n_j^1}{\frac{4n}{9}} = \frac{n_j^a + n_j^b + n_j^d + n_j^e}{\frac{4n}{9}} \tag{4}$$

$$T_2(r_k^e) = \sum_{j=0}^{k} p_2(r_j), \quad p_2(r_j) = \frac{n_j^2}{\frac{4n}{9}} = \frac{n_j^b + n_j^c + n_j^e + n_j^f}{\frac{4n}{9}} \tag{5}$$

$$T_3(r_k^e) = \sum_{j=0}^{k} p_3(r_j), \quad p_3(r_j) = \frac{n_j^3}{\frac{4n}{9}} = \frac{n_j^d + n_j^e + n_j^g + n_j^h}{\frac{4n}{9}} \tag{6}$$

$$T_4(r_k^e) = \sum_{j=0}^{k} p_4(r_j), \quad p_4(r_j) = \frac{n_j^4}{\frac{4n}{9}} = \frac{n_j^e + n_j^f + n_j^h + n_j^i}{\frac{4n}{9}} \quad , \tag{7}$$

where n is the number of pixels in the entire region, n_j^x represents the number of pixels in sub-region x with j-th level, and $x = a, b, \ldots, i$.

From (3) through (7), the histogram equalization function for sub-region e is

$$\begin{aligned} s_k^e &= \frac{1}{4}\sum_{j=0}^{k}\left[p_1(r_j) + p_2(r_j) + p_3(r_j) + p_4(r_j)\right] \\ &= \frac{1}{4}\sum_{j=0}^{k}\left[\frac{4n_j^e + 2n_j^b + 2n_j^d + 2n_j^f + 2n_j^h + n_j^a + n_j^c + n_j^g + n_j^i}{\frac{4n}{9}}\right] \\ &= \sum_{j=0}^{k}\left[\frac{\frac{9}{4}n_j^e + \frac{9}{8}(n_j^b + n_j^d + n_j^f + n_j^h) + \frac{9}{16}(n_j^a + n_j^c + n_j^g + n_j^i)}{n}\right] \end{aligned} \tag{8}$$

From (8), the *pdf* for *POSHE* of the sub-region *e* can be written as follows,

$$\begin{aligned} p(r_j^e) &= \frac{1}{4}p_e(r_j^e) + \frac{1}{8}[p_b(r_j^e) + p_d(r_j^e) + p_f(r_j^e) + p_h(r_j^e)] \\ &+ \frac{1}{16}[p_a(r_j^e) + p_c(r_j^e) + p_g(r_j^e) + p_i(r_j^e)] \end{aligned} \tag{9}$$

where $p_x(r_j^e)$ represents the probability of level j in region x, where $x = a, b, \ldots, i$. This *pdf* is illustrated in Figure.2. As mentioned previously, when we decrease the step size, the *pdf* mask size increases. In general, for a step size which is 1/N of a sub-block size, a (2N – 1) × (2N – 1) mask is generated.

Complexity comparison between block-overlapped histogram equalization and *POSHE* is performed by the histogram equalization repetition time. For block-overlapped histogram equalization, the sub-block histogram equalization must be performed for every pixel in the input image frame. For an M × N image, this operation has to be repeated M × N times. The half sub-block sized regions of four edges are histogram equalized with other methods. So, for m × n sub-block, the computational complexity is as follows.

$$Freq_{Block-overlapped\ HE} = (M-m)\times(N-n) \tag{10}$$

In *POSHE*, sub-block histogram equalization frequency is proportional to the sub-block size and step size. If we obtain the sub-block size by dividing the input image size with a sub-block divisor B and the step size by dividing with a step divisor S, the histogram equalization frequency is represented as a function of B and S as follows.

$$Freq_{POSHE} = \left[\left(1-\frac{1}{B}\right)S + 1\right]^2 \tag{11}$$

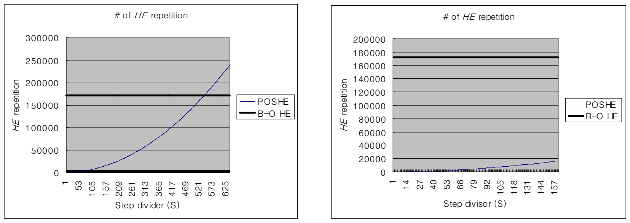
From (11), it can be shown that the frequency of *POSHE* is proportional to B and S. In addition, *POSHE* needs extra computation time for *BERF* and output scaling. Since output scaling is of a fixed complexity which is small compared with histogram equalization and *BERF*, it can be replaced with a constant C. Because the *BERF* procedure must be performed for each boundary, the execution frequency is proportional to the step size and the image size, so the following equation can be formulated.

$$Freq_{BERF} = M(S-1) + N(S-1) = (M+N)(S-1) \tag{12}$$

A *BERF* procedure is composed of three adds, one shift and one to several increases and decreases. Since a sub-block histogram equalization is composed of 255 adds (for 8-bit image), 256 shifts and m × n increases, the *BERF* procedure for each pixel takes about 1/85th of the computation of sub-block histogram equalization. Therefore, overall histogram equalization frequency can be obtained from (11) and (12).

$$\begin{aligned} Freq_{Overall\ POSHE} &= \left[\left(1-\frac{1}{B}\right)S+1\right]^2 HE + (M+N)(S-1)BERF + C \\ &= \left[\left(1-\frac{1}{B}\right)S+1\right]^2 + \frac{1}{85}(M+N)(S-1) + C \\ &= \left(1-\frac{1}{B}\right)^2 S^2 + \left[2\left(1-\frac{1}{B}\right) + \frac{(M+N)}{85}\right]S + 1 - \frac{(M+N)}{85} + C \end{aligned} \tag{13}$$

Using (10) and (13), a complexity comparison between the two algorithms can be done. Assuming the image size is 640 × 480 and the sub-block divisor is fixed at 4, histogram equalization frequency of *POSHE* becomes a second-order polynomial of S. Their graphs for S are illustrated in Figure.6.



(a) (b)

Figure.6. *HE* repetition frequencies of block-overlapped *HE* and *POSHE*
(a) From S = 0 to S = 640,
(b) Zoomed version of (a) from S = 0 to S = 160

Figure.6 shows that the proposed algorithm takes much less time compared to block-overlapped histogram equalization. Through many simulations, it has been shown that when the step size is 1/8th of the sub-block size, the blocking effect cannot be seen. In this case, the step size is 20 × 15, and the step divisor S is 32. Sub-block histogram equalization is performed only 1,028 times. This is 0.59% of the block-overlapped histogram equalization frequency, which is performed 172,800 times.

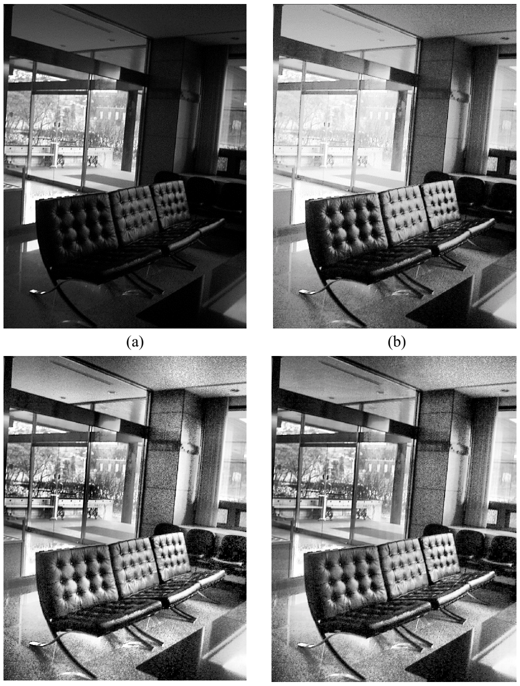
5. SIMULATION RESULTS

Previous algorithms and the proposed algorithm are simulated on several images. The results show common features as follows. Global histogram equalization enhances the contrast of dark objects and larger regions while it de-emphasizes the contrast of the background and smaller regions. Block-overlapped histogram equalization and *POSHE* produce similar enhanced results. These results are simulated on 640 × 480-sized input images with a 160 × 120-sized sub-block. *POSHE* step size is set to be 20 × 15 to make a 15 × 15 sized mask. Using these two methods, the contrast of the background is drastically enhanced compared with global histogram equalization.

The simulations are performed with many images having various histograms. POSHE enhances not only bad-contrast images such as back-lit, fore-lit but also normal-lit images, and all the results are acceptable for human eyes. Example in Figure 7 is a general back-lit image. In particular, the background information of the histogram equalized image is nearly lost. But contrasts are enhanced in block-overlapped histogram equalization and *POSHE*, so the background can be recognized clearly.

6. CONCLUSIONS

In this paper, a new contrast enhancement algorithm, termed *POSHE*, is proposed. *POSHE* is derived from local histogram equalization, but it is more effective and much faster compared to this. The effectiveness results from its local adaptability, and its speed from the partial overlapping feature. The most important feature of *POSHE* is a low-pass filter shaped mask that obtains a sub-region probability density function, and the fact that the mask size can be varied to achieve quality improvements at the expense of calculation complexity. *POSHE* gives overall large contrast enhancements that the global histogram equalization methods cannot do, and achieves drastically low computation overhead compared to the block-overlapped histogram equalization without incurring any blocking effects paying a few additional computation with *BERF*. The powerful contrast enhancement capability of *POSHE* is useful in many consumer electronics fields, such as commercial camcorders, digital still cameras, and especially closed circuit cameras. Due to its simplicity, *POSHE* can be realized in simple hardware and processed in real-time.



(a) (b) (c) (d)

Figure.7. Simulation result
(a) original image, (b) histogram equalized image,
(c) block-overlapped *HE*ed image, (d) *POSHE*ed image

7. REFERENCES

[1] Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, 2nd edition. Addison-Wesley Publishing Co., 1992

[2] Anil K. Jain, “Fundamentals of Digital Image Processing”, Prentice-Hall, 1989

[3] J. Zimmerman, S. Pizer, E. Staab, E. Perry, W. McCartney, and B. Brenton, “Evaluation of the Effectiveness of Adaptive Histogram Equalization for Contrast Enhancement”, IEEE Trans. On Medical Imaging, December. 1988, pp.304-312

[4] Yeong-taeg Kim, “Contrast Enhancement Using Brightness Preserving Bi-Histogram equalization”, IEEE Trans. On Consumer Electronics, Vol. 43, No 1, February 1997, pp.1-8

[5] Tae-keun Kim, Joon-ki Paik, and Bong-soon Kang, “Contrast Enhancement System Using Spatially Adaptive Histogram equalization with Temporal Filtering”, IEEE Trans. On Consumer Electronics, Vol. 44, No. 1, February 1998, pp.82-86