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

Joung-Youn Kim, Lee-Sup Kim, and Seung-Ho Hwang

***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. It can be classified into two branches according to the transformation function used: global or local. 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 complexity of computation required is very high due to its fully overlapped sub-blocks.**

In this paper, a low-pass filter-type mask is used to get a nonoverlapped sub-block histogram-equalization function to produce the high contrast associated with local histogram equalization but with the simplicity of global histogram equalization. This mask also eliminates the blocking effect of nonoverlapped sub-block histogram-equalization. The low-pass filter-type mask is realized by partially overlapped sub-block histogram-equalization (POSHE). With the proposed method, since the sub-blocks are much less overlapped, the computation overhead is reduced by a factor of about 100 compared to that of local histogram equalization while still achieving high contrast. The proposed algorithm can be used for commercial purposes where high efficiency is required, such as camcorders, closed-circuit cameras, etc.

***Index Terms*—Contrast enhancement, histogram equalization, POSHE.**

I. 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 [1]–[8] and among these, histogram equalization [1]–[5] is the most common method used due to its simplicity and effectiveness. The algorithms in [6] and [7] employ connected components and edge detection for contrast enhancement. These methods require very complex computation and hardware, so they cannot be applied to commercial real time purposes. The gamma-processing algorithm [8] is simple but produces low contrast-enhancement.

Histogram equalization can be categorized into two methods: global 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. In general, important objects have a higher and wider histogram region, so the contrast of these objects is stretched. On the other hand, the contrast of lower and narrower histogram regions, such as the background, is lost.

This global histogram equalization method is simple and powerful, but it cannot adapt to local brightness features of the input image because it uses only global histogram information over the whole image. This fact limits the contrast-stretching ratio in some parts of the image, 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]. In this method, a rectangular sub-block of the input image is first defined, a histogram of that region is obtained, and then its histogram-equalization function is determined. Thereafter, the center pixel of the region is histogram equalized using this function. The center of the rectangular region is then moved to the adjacent pixel and the histogram equalization is repeated. This procedure is repeated pixel by pixel for all input pixels. 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 computation complexity is very high. For example, for a 640×480 pixel image, the histogram equalization must be performed maximally 307 200 times.

To reduce this computation complexity and obtain the advantage of local adaptability of block-overlapped histogram equalization, sub-block nonoverlapped histogram equalization can be used. Even so, this nonoverlapped method will sometimes suffer from blocking effects. Therefore, in this paper, partially overlapped sub-block histogram-equalization (POSHE) is proposed. Using POSHE, the contrast of the input image can be enhanced at a similar rate to block-overlapped histogram equalization, while computation complexity can be reduced considerably and any blocking effects eliminated.

II. Conventional Histogram-Equalization Techniques

In histogram equalization [1], [2], the transformation function $T(r)$ is given by the following relation:

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

Manuscript received July 17, 1999; revised August 3, 2000. This paper was recommended by Associate Editor J.-N. Hwang.

The authors are with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Taejon, Korea (e-mail: jinee@mvlsi.kaist.ac.kr; lskim@ee.kaist.ac.kr; shwang@ee.kaist.ac.kr).

Publisher Item Identifier S 1051-8215(01)03011-7.

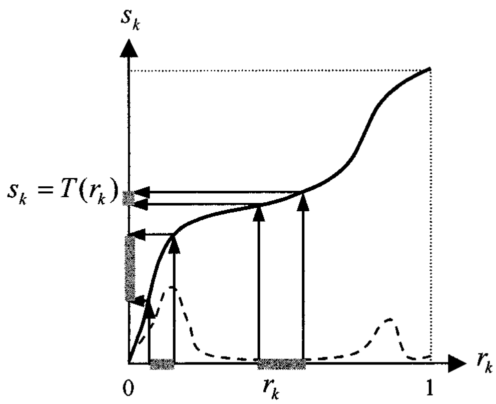


Fig. 1. Transformation function for histogram equalization.

where

$p_r(r_j) = n_j/n$ probability density function (pdf) of the input image level j;
n total number of pixels in the input image;
n_j input pixel number of level j.

Hence, the transformation function $T(r)$ represents the cumulative distribution function (cdf) of the input image levels. This transformation function is illustrated in Fig. 1. Using this cdf as a transformation function, data regions that have high concentration are transformed into wider regions (the concentration is reduced), while other regions are transformed into narrower regions (the sparse data become concentrated). Global histogram equalization uses the overall histogram as its transformation function, so it cannot adapt to the local light condition of the image. This feature results in the contrast deterioration of background and small objects.

Local histogram equalization, so-called block-overlapped histogram equalization [1], [5], can obtain overall contrast enhancement, regardless of the location in the input image. As discussed in the previous section, 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 the histogram equalization is performed with each sub-block's local histogram, this local histogram-equalization method can adapt well to the partial light condition of the image. Therefore, the difference of contrast enhancement between objects and background can be omitted.

III. POSHE

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

However, this nonoverlapped method cannot avoid a blocking effect. To eliminate this effect, additional operations are needed. 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 nonoverlapped sub-block histogram equalization is performed, their brightness values are changed by each transformation function. Since these transformation functions are obtained from each sub-block's histogram, they 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. This is known as the blocking effect. On the other hand, in block-overlapped histogram equalization, the transformation functions for each pixel are obtained from partially overlapped neighboring sub-blocks. Therefore, the shape difference of the transformation functions are very small and the blocking effect can be ignored.

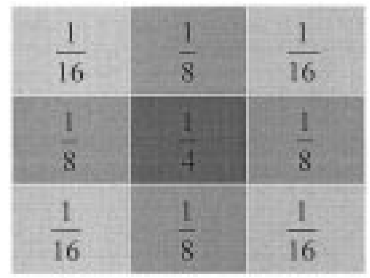


Fig. 2. pdf of POSHE.

To reduce shape difference in nonoverlapped sub-block histogram equalization, the weighted sum of neighboring sub-blocks' histograms can be used for generation of the transformation function for the current sub-block. Using a 3×3 mask, as shown in Fig. 2, the central sub-block's transformation function is obtained from the masked histogram of itself and its eight neighboring sub-blocks. This mask resembles a low-pass image filter (LPF), and its operation is in fact similar to that of an LPF. Image LPF reduces the difference in brightness between neighboring pixels, so that the resultant image is blurred. This low-pass filtering for sub-block histograms produces neighboring transformation functions that are similar to each other, resulting in a reduction of the blocking effect. As mask size is increased, blocking effects can be decreased by increasing the computation complexity. Simulations have shown that for mask sizes larger than 15×15, blocking effects are undetectable with simple filtering and performance is compatible with that of block-overlapped histogram equalization.

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 3×3 mask, as shown in Fig. 2, a sub-block is moved by a half of the sub-block size to partially overlap the next sub-block, and histogram equalization is performed for all pixels in the sub-block. This means, if the sub-block size is 120, then the step size is 60. So each sub-block is overlapped by 60, which is the half of sub-block size. Each pixel is histogram equalized more than once and the result for each pixel is accumulated. After partially overlapped sub-block histogram equalization is completed, each accumulated pixel

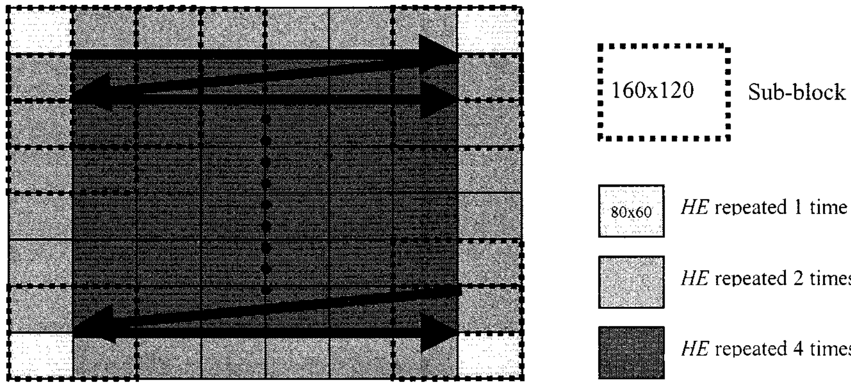


Fig. 3. Image plane for POSHE.

value is divided by its accumulation frequency. The image plane of POSHE is illustrated in Fig. 3. Through simple formula expansion illustrated in the following section, it can be seen that this procedure results in a 3×3 LPF mask. By decreasing the step size moved each time, the mask size can be increased. As the mask size is increased, the transformation function shape difference is decreased, so that the blocking effect is diminished. If the step size is taken as a quarter of the sub-block size, then the mask size is 7×7. Similarly, for a $1/8$ step size, the mask size becomes 15×15. Therefore, mask size can be changed easily by manipulating the step size moved. By taking a step size that is equal to the quotient of the sub-block size divided by a number that is a multiple of 2, most divisions can be performed by shift. Because sub-blocks are partially overlapped, this method is named 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.

The procedures of POSHE are as follows.

1) Define an $M \times N$-sized output image array for an $M \times N$ input image and set all values to zero.
2) Assign an $m \times n$ sub-block. For computational simplicity, a sub-block size is selected to be equal to the quotient of the input image size divided by a multiple of two. Assign the sub-block origin using the input image origin.
3) Perform local histogram equalization for the current sub-block. Unlike block-overlapped histogram equalization, the histogram equalization is performed over the whole sub-block, and the results are accumulated in the output image array.
4) Increase the horizontal-coordinate of the sub-block origin by the horizontal step size and repeat procedure 3. 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 input image plane.
5) After sub-block histogram equalization finished, divide each pixel value in the output image array by its sub-block histogram equalization frequency.
6) If a small blocking effect is generated at the sub-block boundaries, eliminate it with a blocking-effect reduction filter.

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 of the 15×15 mask, the blocking effect is almost completely eradicated and the average brightness of neighboring sub-blocks is almost equal. But at some sub-block boundaries, a few gray-level discontinuities may be generated and these discontinuities may show up as blocking effects. There is a small possibility that these small blocking effects may occur, so a protection method for small level difference is needed. When level discontinuity arises at sub-block boundaries, the filtering procedure starts. The occurrence of a blocking effect can be detected by the difference in brightness of adjacent pixels at the boundaries. To distinguish between the original edge and “POSHE-ed” boundary, the edge information of the original image can be used. BERF is a kind of low-pass filter, but it is applied to the blocked-boundaries.

The BERF procedures are as follows. First, blocking-effect detection is needed. Since blocking effects occur at each sub-block boundary, only these boundaries need to be checked. To exclude the original object edge on each sub-block boundary, the original edge information at sub-block boundaries must be checked as well. Namely, if there exists some level discontinuity at sub-block boundaries in POSHE-ed image but not in the original image, a blocking effect is generated. Second, when a blocking effect is detected, filtering starts from each sub-block boundary. An average of the two adjacent boundary pixels is calculated and the brightness values of these two pixels are replaced by this average. Third, in a perpendicular direction to the boundary, pixel values in an increasing/decreasing order are replaced by slowly changing values. This replacing value starts from the average, increases by a predetermined step size for the brighter sub-block, and decreases for the darker sub-block. This filtering is terminated when the termination condition is met. The increasing/decreasing brightness step size is selected such that the human eyes cannot differentiate the discontinuity in brightness at the sub-block boundaries. This unrecognizable brightness level is known to be about 3–4 levels in 256 gray-level image through experimentation [9]. This means that through BERF, the blocking effect at sub-block boundaries are eliminated. These procedures are repeated for all sub-block boundaries. Termination occurs under the follow conditions.

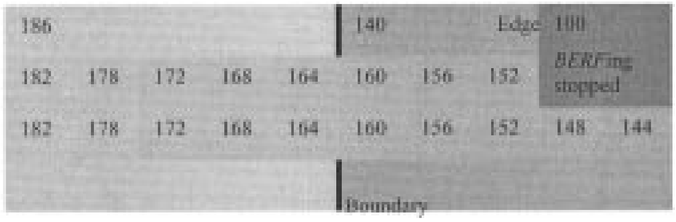


Fig. 4. Example of BERF.

1) When the brightness difference between pixels that have previously undergone the BERF procedures and the present pixel is smaller than the step size, the BERF procedure is terminated. This condition is satisfied when the brightness starts with the average value at the sub-block boundary and reaches the original "POSHE-ed" brightness.
2) When the brightness difference between the present pixel and the next pixel is large enough to indicate that there is an object edge (*not* on boundary), the BERF procedure is terminated.
3) If the brightness of the next pixel is smaller/bigger than the present pixel while BERF is being carried out, and the phase is increasing/decreasing, the increasing/decreasing rule is violated. BERF is therefore terminated.

This BERF procedure is performed maximally with a half of the step size in each direction. An example of BERF is shown in Fig. 4. The numbers indicate the pixel luminance. The levels in the figure are exaggerated to show the filtering effect clearly.

IV. Analysis of POSHE

A histogram equalization function is the cumulative distribution function of the input image as shown in (1). Let us suppose a region is composed of four partially overlapped sub-blocks for a 3×3 mask. Since the step size is a half of the sub-block size, the region is divided into nine sub-regions. POSHE is performed for the four sub-blocks with a step size of half the sub-block. This region is illustrated in Fig. 5. 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

$$s_k^e = \frac{1}{4}\left[T_1\left(r_k^e\right) + T_2\left(r_k^e\right) + T_3\left(r_k^e\right) + T_4\left(r_k^e\right)\right] \tag{2}$$

and each histogram equalization function can be written as

$$T_1\left(r_k^e\right) = \sum_{j=0}^{k} p_1(r_j)$$
$$p_1(r_j) = \frac{n_j^1}{\frac{4n}{9}} = \frac{n_j^a + n_j^b + n_j^d + n_j^e}{\frac{4n}{9}} \tag{3}$$

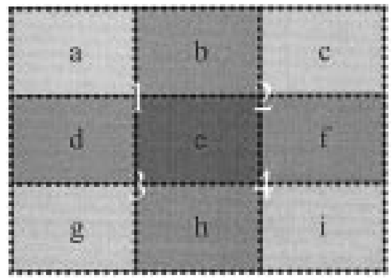


Fig. 5. Four partially overlapped sub-blocks.

$$T_2\left(r_k^e\right) = \sum_{j=0}^{k} p_2(r_j)$$
$$p_2(r_j) = \frac{n_j^2}{\frac{4n}{9}} = \frac{n_j^b + n_j^c + n_j^e + n_j^f}{\frac{4n}{9}} \tag{4}$$
$$T_3\left(r_k^e\right) = \sum_{j=0}^{k} p_3(r_j),$$
$$p_3(r_j) = \frac{n_j^3}{\frac{4n}{9}} = \frac{n_j^d + n_j^e + n_j^g + n_j^h}{\frac{4n}{9}} \tag{5}$$
$$T_4\left(r_k^e\right) = \sum_{j=0}^{k} p_4(r_j),$$
$$p_4(r_j) = \frac{n_j^4}{\frac{4n}{9}} = \frac{n_j^e + n_j^f + n_j^h + n_j^i}{\frac{4n}{9}} \tag{6}$$

where

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

From (2) to (6), the histogram equalization function for sub-region e is as in (7), shown at the bottom of the next page. From (7), the pdf for POSHE of the sub-region e can be written as follows:

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

where $p_x(r_j^e)$ represents the probability of level j in region x, where $x = a, b, \ldots, i$. The summation of this pdf is $(1/4) + 4\times(1/8) + 4\times(1/16) = 1$, which satisfies the condition of the probability density function, namely that the accumulation of pdf must be 1. This probability density function can be illustrated as in Fig. 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 $(2N-1)\times(2N-1)$ mask is generated. A 7×7 mask is illustrated in Fig. 6.

Complexity comparison between block-overlapped histogram equalization and POSHE is needed to verify the effectiveness of the proposed algorithm. Since an accurate comparison is difficult, only histogram-equalization repetition frequency is considered in this paper. In histogram equalization,

$\frac{1}{256}$	$\frac{2}{256}$	$\frac{3}{256}$	$\frac{4}{256}$	$\frac{3}{256}$	$\frac{2}{256}$	$\frac{1}{256}$
$\frac{2}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{8}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{2}{256}$
$\frac{3}{256}$	$\frac{6}{256}$	$\frac{9}{256}$	$\frac{12}{256}$	$\frac{9}{256}$	$\frac{6}{256}$	$\frac{3}{256}$
$\frac{4}{256}$	$\frac{8}{256}$	$\frac{12}{256}$	$\frac{16}{256}$	$\frac{12}{256}$	$\frac{8}{256}$	$\frac{4}{256}$
$\frac{3}{256}$	$\frac{6}{256}$	$\frac{9}{256}$	$\frac{12}{256}$	$\frac{9}{256}$	$\frac{6}{256}$	$\frac{3}{256}$
$\frac{2}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{8}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{2}{256}$
$\frac{1}{256}$	$\frac{2}{256}$	$\frac{3}{256}$	$\frac{4}{256}$	$\frac{3}{256}$	$\frac{2}{256}$	$\frac{1}{256}$

Fig. 6. 7×7 mask using 1/4 sub-block-sized step.

obtaining the transformation function is the most demanding computative task because the transformation is performed by just memory access. For block-overlapped histogram equalization, the sub-block histogram-equalization function must be obtained for every pixel in the input image frame. For an $M \times N$ image, this operation has to be repeated $M \times N$ times. Since block-overlapped histogram equalization transforms the center pixels of each sub-block, half sub-block sized regions of four edges are histogram equalized with other methods. If we assume that the local histogram equalization is not performed for the edge pixels, then its complexity C_{BOHE} takes the minimum value as follows for an $m \times n$ sized sub-block and it can be represented with a sub-block histogram equalization complexity C_{HE}

$$C_{\mathrm{BOHE}} = (M-m) \times (N-n) \times C_{\mathrm{HE.}} \tag{9}$$

In POSHE, sub-block histogram equalization frequency is proportional to the sub-block size $m \times n$ and step size $a \times b$. Since the sub-block histogram equalization is performed with increasing sub-block's origins by the step size until its origins reach to $(M-m)$ or $(N-n)$, the histogram equalization frequency can be obtained from the input image size $(M-m, N-n)$ divided by the step size (a, b). The histogram equalization frequency N_{POSHE} is represented in Fig. 7 and in (10)

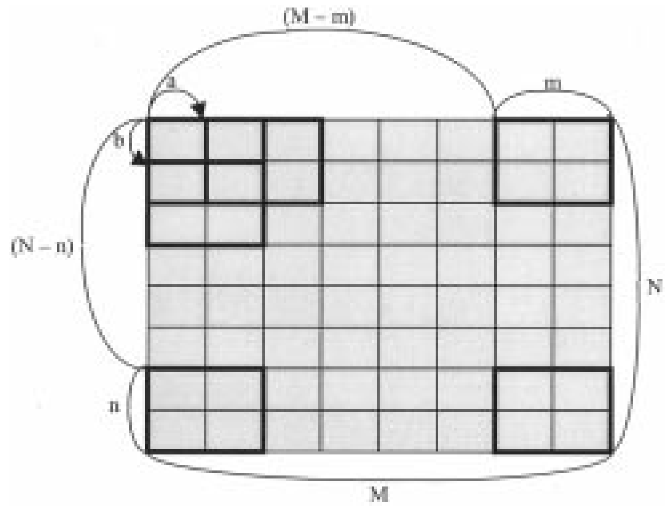


Fig. 7. Histogram-equalization frequency of POSHE.

$$\begin{aligned} N_{\mathrm{POSHE}} &= \left(\frac{M-m}{a}+1\right) \times \left(\frac{N-n}{b}+1\right) \\ &= \left(\frac{M}{a} - \frac{m}{M}\cdot\frac{M}{a}+1\right) \times \left(\frac{N}{b} - \frac{n}{N}\cdot\frac{N}{b}+1\right) \\ &= \left[\left(1-\frac{m}{M}\right)\frac{M}{a}+1\right] \times \left[\left(1-\frac{n}{N}\right)\frac{N}{b}+1\right] \\ &= \left[\left(1-\frac{1}{B}\right)S+1\right]^2 \end{aligned} \tag{10}$$

where $B = M/m = M/a$ is a sub-block divisor and $S = M/a = N/b$ is a step divisor.

From (10), it can be shown that the frequency of POSHE is proportional to B and S. This can be easily understood because when B and S are increased, the sub-block and step size are decreased, so more sub-block histogram equalization must be performed.

In addition, POSHE needs extra computation time for BERF and output scaling. Since final output scaling is of a fixed complexity that 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 that is formed by a step-sized sub-block moving and the number of boundaries is the multiplication of an image size $M \times N$, a step di-

$$\begin{aligned} s_k^e &= \frac{1}{4}\sum_{j=0}^{k}[p_1(r_j)+p_2(r_j)+p_3(r_j)+p_4(r_j)] \\ &= \frac{1}{4}\sum_{j=0}^{k}\left[\frac{4n_j^e+2n_j^b+2n_j^d+2n_j^f+2n_j^h+n_j^a+n_j^c+n_j^g+n_j^i}{\frac{4n}{9}}\right] \\ &= \sum_{j=0}^{k}\left[\frac{\frac{9}{4}n_j^e+\frac{9}{8}\left(n_j^b+n_j^d+n_j^f+n_j^h\right)+\frac{9}{16}\left(n_j^a+n_j^c+n_j^g+n_j^i\right)}{n}\right] \end{aligned} \tag{7}$$

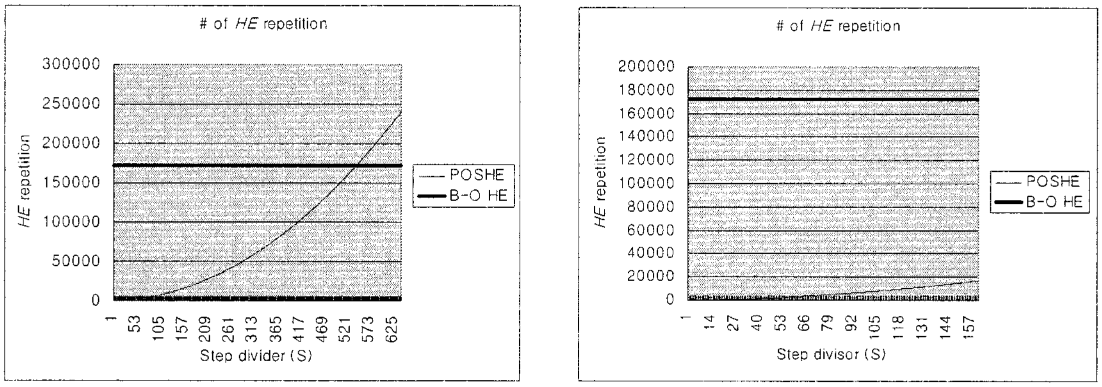


Fig. 8. HE repetition frequencies of block-overlapped HE and POSHE. (a) From $S = 0$ to $S = 640$. (b) Zoomed version of (a) from $S = 0$ to $S = 160$.

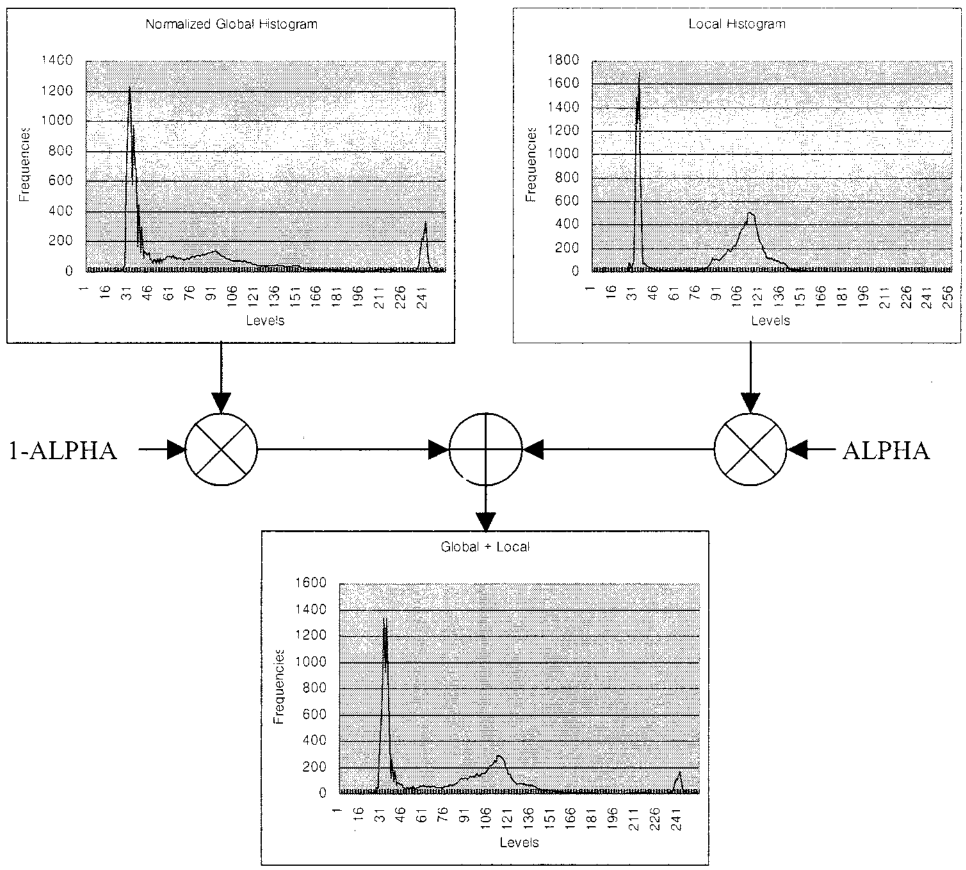


Fig. 9. Weighted summing of global and local histogram.

visor S and a BERF performing frequency. If we assume that the BERF procedure is always performed at the sub-block boundaries without conditions, the following equation can be formulated for BERF repetition frequency.

$$N_{\mathrm{BERF}} = M(S-1) + N(S-1) = (M+N)(S-1) \tag{11}$$

A BERF procedure is composed of two subtracts for checking level discontinuity (one for original and one for POSHE-ed) at boundaries, one add and one shift for averaging, and one to several increases and decreases. Sub-block histogram equalization is composed of 255 adds (for 8-bit image), 256 shifts, and $m \times n$ increases, so the BERF procedure for each pixel takes about $1/85$th of the computation of sub-block histogram equalization. Therefore, overall POSHE complexity C_{POSHE} can be obtained from (10) and (11) as a function of C_{HE}

$$\begin{aligned} C_{\mathrm{POSHE}} &= N_{\mathrm{POSHE}} \times C_{\mathrm{HE}} + N_{\mathrm{BERF}} \times C_{\mathrm{HE}} + C \\ &= \left[\left\{ \left(1 - \frac{1}{B}\right) S + 1 \right\}^2 + \frac{1}{85}(M+N)(S-1) \right] \\ &\quad \times C_{\mathrm{HE}} + C \\ &= \left[\left(1 - \frac{1}{B}\right)^2 S^2 + \left\{ 2\left(1 - \frac{1}{B}\right) + \frac{(M+N)}{85} \right\} S \right. \\ &\quad \left. + 1 - \frac{(M+N)}{85} \right] \times C_{\mathrm{HE}} + C \end{aligned} \tag{12}$$

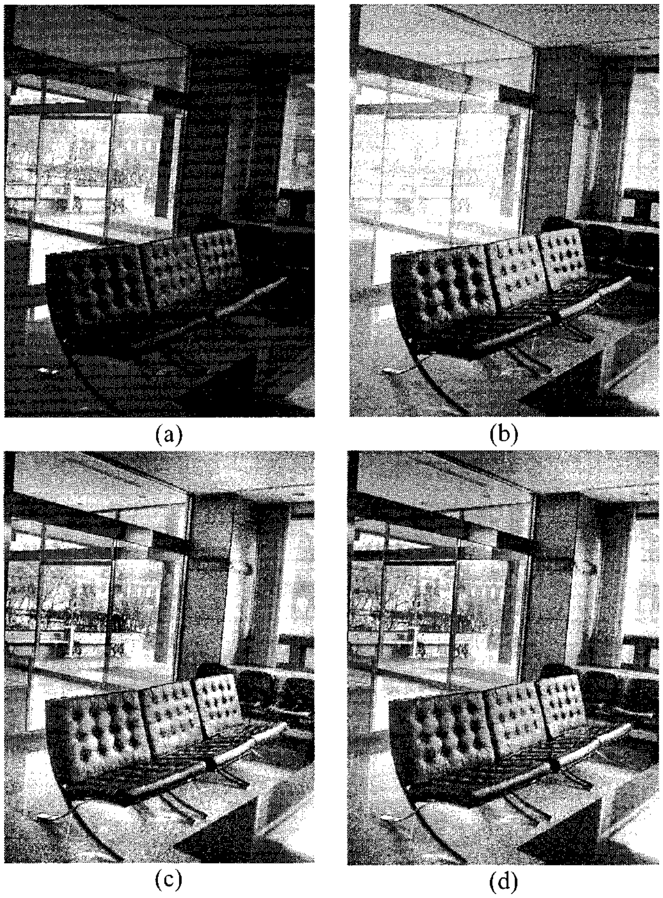


Fig. 10. Simulation results. (a) Original image. (b) Histogram equalized image. (c) Block-overlapped HEed image. (d) POSHEed image.

By using (9) and (12), setting the sub-block size of each method equal, 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 Fig. 8.

Fig. 8 shows that the proposed algorithm takes much less time compared to block-overlapped histogram equalization. From Fig. 8(a), it can be seen that two lines meet between $S = 480$ and $S = 640$. It is clear that when $S = 480$, the step size of POSHE 1.33×1 is larger than 1, so the histogram equalization frequency is smaller than that of block-overlapped histogram equalization. Similarly, when $S = 640$, the step size 1×0.75 is smaller than 1 and the opposite situation occurs. Through many simulations, it has been shown that when the step size is $1/8$th 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 1028 times. This is 0.59% of the block-overlapped histogram equalization frequency, which is performed 172 800 times.

V. Over-Enhancing Problem and Solution

The object of local histogram equalization is overall contrast enhancement regardless of location. As a result, local contrasts of both target object and background are enhanced, and therefore a much clearer image is generated from a numerical point of view. For special purposes, such as closed-circuit camera systems or image recognition of background in order to search for something, this overall high contrast is important. But, in the case of commercial camcorders or enhancement of general poorly-lit pictures, this feature may distort an image by over-enhancing the background. The human recognizes an image as only a target object and background. Since the uniform contrast enhancement feature of local histogram equalization makes the background as clear as the target object, the emphasized background confuses the human sense of distance. This image distortion is caused from the fact that local histogram equalization uses only local histogram information.

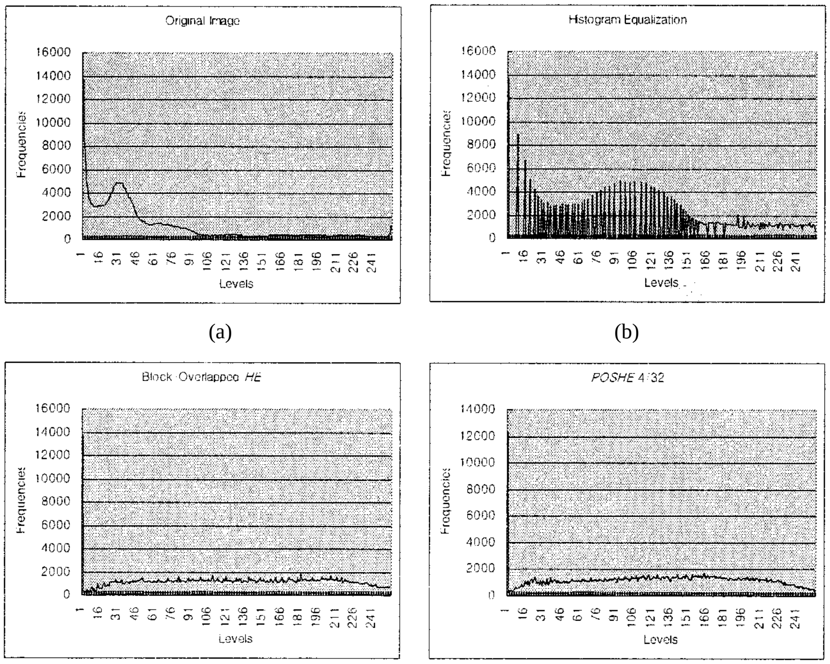
To overcome this over-enhancing problem, local histogram equalization must be modified to include global histogram information. When sub-block histogram equalization is performed, its sub-block histogram can be mixed with a global histogram in a proportional ratio. When this local/global histogram information-mixing ratio is increased, local information increases, so that the local contrast enhancement is emphasized. As a result, the contrast of the background becomes more emphasized. For the opposite case when global information is increased, the contrast of the background decreases. This histogram mixing can be performed easily by a proportional summing of sub-block histogram and global histogram. By changing the ratio, the contrast of the background/target objects of the transformed image can be varied to meet the purposes.

In Fig. 9, a global histogram, local histogram, and their weighted sum are shown. This weight can be represented as ALPHA. ALPHA is defined as the ratio of local histogram information over global histogram information. By increasing ALPHA, the background contrast can be increased. In this example, ALPHA is taken as 0.5 and the local histogram is taken from the sub-block in the background. Each histogram is normalized with a sub-block pixel number. This result is shown in Fig. 9. It can be seen that the resultant sub-block histogram also makes use of global information. Therefore, this summing effect can solve the problem of over-enhancing of the background in local histogram equalization by enabling the local sub-block histogram equalization to adapt to global light conditions of images.

VI. Simulation Results

Previous algorithms and the proposed algorithm are simulated on several images, and the results are illustrated to show the performance of each.

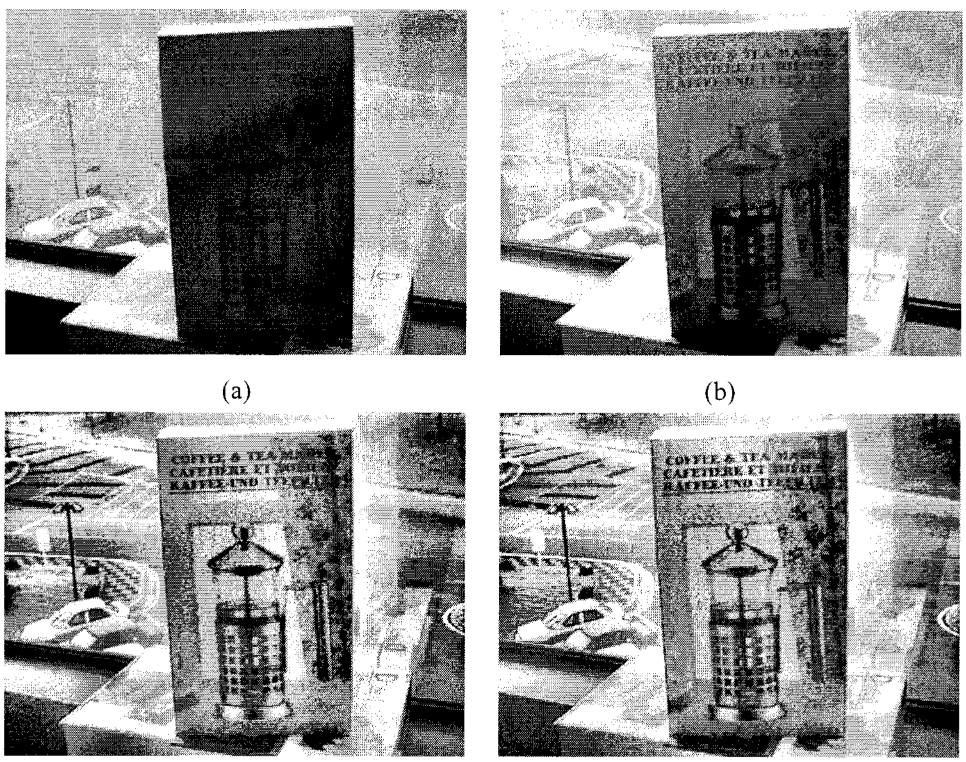
The results show common features as follows. Global histogram equalization enhances the contrast of dark objects and larger regions while it de-emphasizes the contrast of the background and smaller regions. Block-overlapped histogram equalization produces the best image, and POSHE generates a slightly less enhanced image than block-overlapped histogram equalization. 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



(a) (b)

(c) (d)

Fig. 11. Histograms of Fig. 10. (a) Original image. (b) Histogram-equalized image. (c) Block-overlapped HEed image. (d) POSHEed image.



(a) (b)

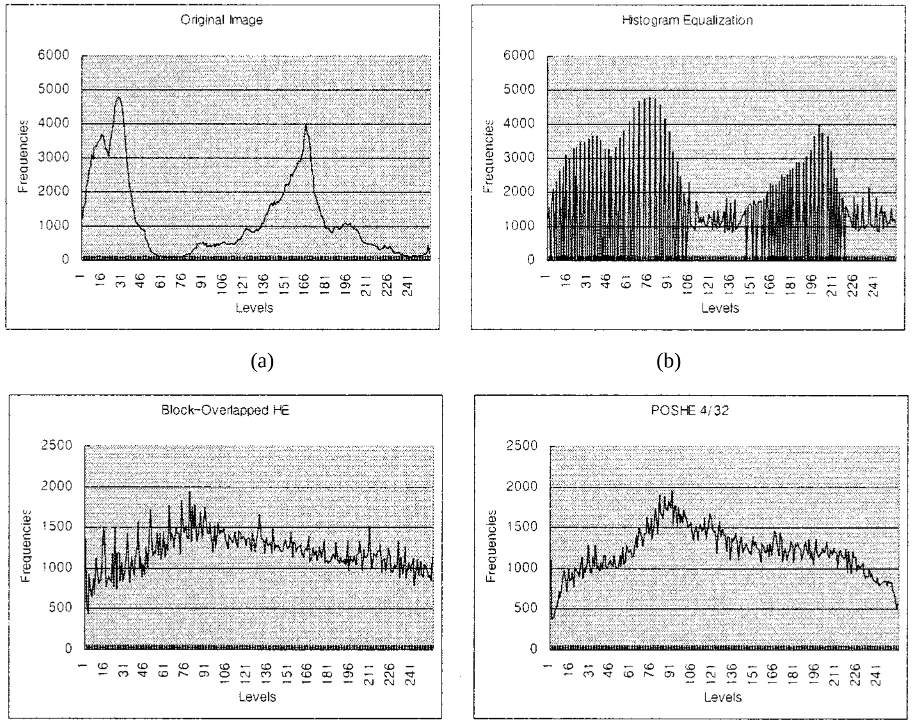
(c) (d)

Fig. 12. Simulation results. (a) Original image. (b) Histogram-equalized image. (c) Bock-overlapped HEed image. (d) POSHEed image.

methods, the contrast of the background is drastically enhanced compared with global histogram equalization.

Observing the histograms, some common features exist also. For all images, the global histogram equalized images give similar shaped histograms to the original images. Block-overlapped histogram equalization and POSHE produce histograms that are completely different from the original histograms. The resultant histograms have almost flat shapes. This, in effect, shows that the images have almost uniform pdfs.

Simulation results of a low-contrast image and its histograms are shown in Figs. 10 and 11. It can be observed simply that the contrast of the block-overlapped histogram equalized



(a) (b)

(c) (d)

Fig. 13. Histograms of Fig. 12. (a) Original image. (b) Histogram-equalized image. (c) Block-overlapped HEed image. (d) POSHEed image.

image and "POSHE-ed" image are enhanced more favorably compared to that of an original image and global histogram equalization. The indoor benches and outdoor background should be observed. 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. Although block-overlapped histogram equalization produces a better image, POSHE requires much less computation.

In Fig. 12 and Fig. 13, results of another simulation and histograms are shown. The input image is a backlit image. Through histogram equalization, the contrast of the dark object is enhanced, but the contrast of the background is rather worse than the original image. The block-overlapped histogram equalization enhances the contrast of both the dark object and the background. The POSHE also enhances the overall contrast.

VII. Conclusion

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 large contrast enhancements which global histogram-equalization methods cannot achieve, and proves to be simpler than local histogram equalization without incurring any blocking effects. As the simulation results show, the contrast is enhanced more than the global method, and the computation overhead is decreased drastically without incurring any blocking effects. 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.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Reading, MA: Addison-Wesley, 1992.
[2] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: 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. Medical Imaging*, pp. 304–312, Dec. 1988.
[4] Y.-T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE Trans. Consumer Electron.*, vol. 43, no. 1, pp. 1–8, Feb. 1997.
[5] T.-K. Kim, J.-K. Paik, and B.-S. Kang, "Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering," *IEEE Trans. on Consumer Electronics*, vol. 44, no. 1, pp. 82–86, Feb. 1998.
[6] G. Boccignone, "A multiscale contrast enhancement method," in *Proc. Int. Conf. Image Processing*, 1997, pp. 306–309.
[7] V. Caselles, J. L. Lisani, J. M. Morel, and G. Sapiro, "Shape preserving local contrast enhancement," in *Proc. Int. Conf. Image Processing*, 1997, pp. 314–317.
[8] S. Sakaue, A. Tamura, M. Nakayama, and S. Maruno, "Adaptive gamma processing of the video cameras for the expansion of the dynamic range," *IEEE Trans. Consumer Electron.*, vol. 41, pp. 555–562, Aug. 1995.
[9] S. A. Karunasekera and N. G. Kingsbury, "A distortion measure for blocking artifacts in image based on human visual sensitivity," *IEEE Trans. Image Processing*, vol. 4, pp. 713–724, June 1995.

Joung-Youn Kim received the B.S. degree in electronics engineering from Yonsei University, Seoul, Korea, in 1997, and the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1999,where he currrently working toward the Ph.D. degree in electrical engineering.

His research interests include 3-D graphics hardware design, 3-D graphics algorithms and architectures, and multimedia VLSI design.

Lee-Sup Kim received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1986 and 1990, respectively.

He was a post-doctoral Fellow at Toshiba Corporation, Kawasaki, Japan, during 1990–1993, where he was involved in the design of the high-performance DSP and single-chip MPEG2 decoder. Since March 1993, he has been with KAIST, where since 1997, he has been an Associate Professor. During 1998, he was on the sabbatical leave with Chromatic Research and SandCraft Inc., Silicon Valley, CA. His research interests are 3-D graphics hardware design, LCD display controller design, multimedia programmable processor design, and high-speed and low-power digital IC design.

Seung-Ho Hwang received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1979, the M.S. degree from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1981, and the Ph.D. degree from the University of California at Berkeley in 1989, both in electrical engineering.

In 1990, he joined the faculty of the Department of Electrical Engineering, KAIST, where he is currently an Associate Professor. His research interests include CAD for VLSI and video signal processing.