# Adaptive Histogram Equalization and Its Variations

**ARTICLE** *in* COMPUTER VISION GRAPHICS AND IMAGE PROCESSING · SEPTEMBER 1987

DOI: 10.1016/S0734-189X(87)80186-X

COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 39, 355–368 (1987)

# Adaptive Histogram Equalization and Its Variations

STEPHEN M. PIZER,*,† E. PHILIP AMBURN,‡ JOHN D. AUSTIN,* ROBERT CROMARTIE,*
ARI GESELOWITZ,§ TREY GREER,* BART TER HAAR ROMENY,"
JOHN B. ZIMMERMAN,# AND KAREL ZUIDERVELD"

*Department of Computer Science and †Department of Radiology, University of North Carolina, Chapel Hill, North Carolina; ‡United States Air Force; §School of Medicine, Pennsylvania State University, Hershey, Pennsylvania; "Roentgenafdeling, Academisch Ziekenhuis, Utrecht; and #Department of Computer Science, Washington University, St. Louis, Missouri

Received May 28, 1986; accepted October 10, 1986

Adaptive histogram equalization (ahe) is a contrast enhancement method designed to be broadly applicable and having demonstrated effectiveness. However, slow speed and the overenhancement of noise it produces in relatively homogeneous regions are two problems. We report algorithms designed to overcome these and other concerns. These algorithms include interpolated ahe, to speed up the method on general purpose computers; a version of interpolated ahe designed to run in a few seconds on feedback processors; a version of full ahe designed to run in under one second on custom VLSI hardware; weighted ahe, designed to improve the quality of the result by emphasizing pixels' contribution to the histogram in relation to their nearness to the result pixel; and clipped ahe, designed to overcome the problem of overenhancement of noise contrast. We conclude that clipped ahe should become a method of choice in medical imaging and probably also in other areas of digital imaging, and that clipped ahe can be made adequately fast to be routinely applied in the normal display sequence. 

## 1. INTRODUCTION

Adaptive histogram equalization (ahe) is an excellent contrast enhancement method for both natural images and medical and other initially nonvisual images. In medical imaging its automatic operation and effective presentation of all contrast available in the image data make it a competitor to the standard contrast enhancement method, interactive intensity windowing. In fact, observer studies [9, 8] indicate that for certain image classes, intensity windowing has no significant advantages in local contrast presentation in any contrast range, while ahe has advantages of being automatic and reproducible, and requiring the observer to examine only a single image.

The basic form of the method was invented independently by Ketcham *et al.* [6], Hummel [5] and Pizer [7]. In this basic form the method involves applying to each pixel the histogram equalization mapping based on the pixels in a region surrounding that pixel (its *contextual region*). That is, each pixel is mapped to an intensity proportional to its rank in the pixels surrounding it. But the basic method is slow, and under certain conditions the enhanced image has undesirable features. Therefore, this paper presents algorithms for ahe that increase its speed on various processors, and it presents variations on ahe that are intended to improve the enhanced image, along with summaries of the effectiveness of these variations.



0734-189X/87 $3.00

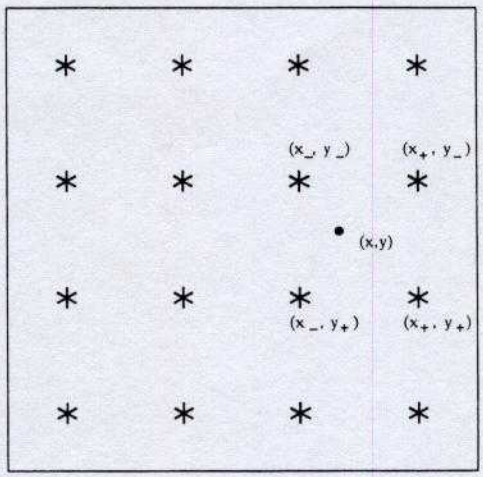


FIG. 1. Sample points (*) for mapping computation, and evaluation point (●).

## 2. SPEEDUP BY SAMPLING AND INTERPOLATION

In its basic form ahe requires time $O(n^2(m + k))$ for an $n \times n$ image with a range of $k$ intensity levels and an $m \times m$ contextual region size. A major speedup can be obtained by calculating the desired mapping only at a sample of pixels and interpolating the mapping between these sample locations. In our work the sample locations at which the mapping is computed are on a grid, and the resulting mapping at any pixel is interpolated from the sample mappings at the four surrounding sample-grid pixels (see Fig. 1). Thus, if the pixel mapped is at location $(x, y)$ and has intensity $i$, and $m_{+-}$ is the mapping at the grid pixel $(x_+, y_-)$ to the upper right of $(x, y)$ and similarly with subscripts $++$, $-+$, and $--$ for the mappings and locations of the grid pixels to the lower right, lower left, and upper left respectively of $(x, y)$, then the interpolated ahe result is given by

$$m(i) = a[bm_{--}(i) + (1 - b)m_{+-}(i)] + [1 - a][bm_{-+}(i) + (1 - b)m_{++}(i)],$$

where

$$a = (y - y_-)/(y_+ - y_-) \quad \text{and} \quad b = (x - x_-)/(x_+ - x_-).$$

Pixels in the borders of the image outside of the sample pixels need to be handled specially, using linear interpolation of the mappings at the two closest points or, in the corners where there is only a single close sample pixel, application of only one mapping.

With such an interpolated ahe there are two parameters, the size of the contextual regions and the spacing of the sample grid. We will discuss each of these in turn.

### 2.1. *Contextual Region Size*

In the interpolative mapping procedure each result pixel is derived by applying four mappings, those associated with four surrounding sample points. Each of those sample points has an associated contextual region, so it can be said that the result pixel in question has a region affecting its value that is the union of the four

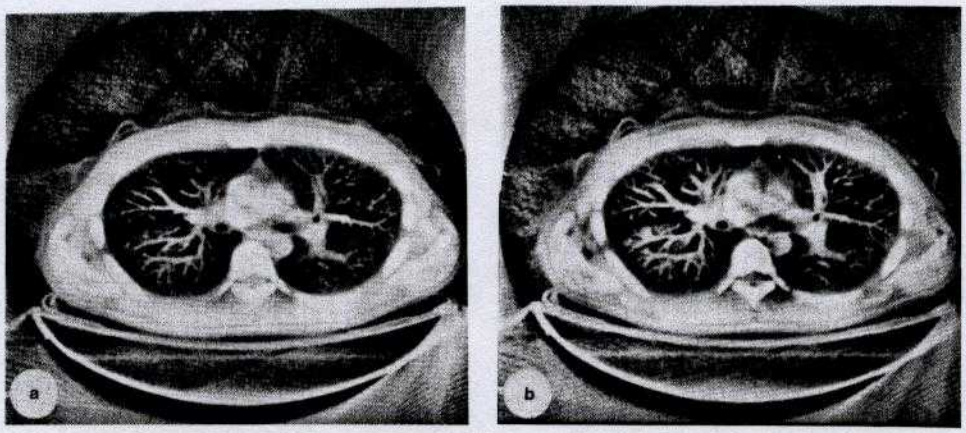


FIG. 2. Results for interpolative ahe of a chest CT scan, with the same ECRs: (a) full sampling (no interpolation); (b) mosaic sampling.

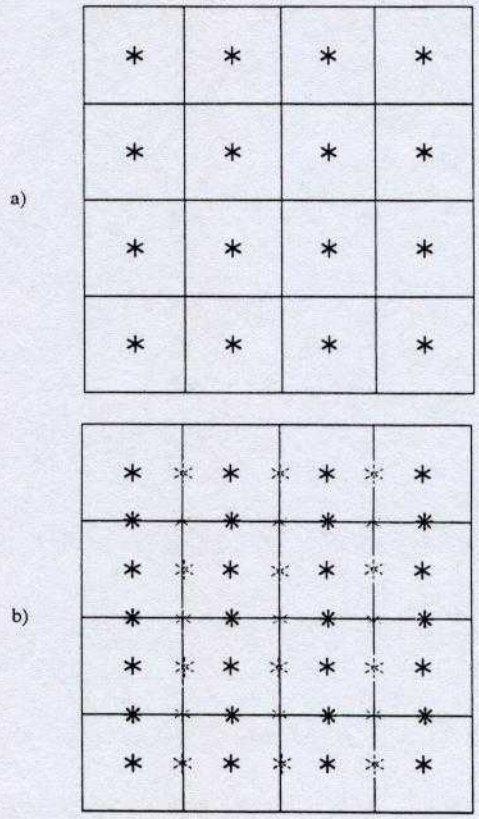


FIG. 3. Contextual regions and their centers: (a) mosaic; (b) half-overlapped.

associated contextual regions of its sample points. Let us call this affecting region the *equivalent contextual region* (ECR). We have found empirically (see Fig. 2) that different versions of the method with the same ECR produce approximately the same result. Thus, for example, if the sample grid point spacing is the same as the contextual region linear dimension, thus forming a mosaic of contextual regions within the image (see Fig. 3a), then uninterpolated (basic) ahe with contextual region area $A$ (and thus ECR $A$) produces approximately the same results as interpolated ahe with contextual region area $A/4$ (and thus ECR $A$). Similarly, if the sample grid point spacing is half that of the contextual region linear dimension (see Sec. 2.2 and Fig. 3b), interpolated ahe with contextual region area $4A/9$ will produce about the same result as the two cases just mentioned. As a result of this fact, we will henceforth refer to sample-based methods in terms of their ECR.

As the ECR area increases, the method becomes less and less locally sensitive but for interpolative ahe more and more efficient. As the ECR area decreases, the image contrast improves up to a point. For a wide range of medical images this optimal ECR area is between $\frac{1}{16}$ and $\frac{1}{64}$ of the image, with the smaller region chosen only when the feature size of interest is quite small. ECR areas intermediate between $\frac{1}{16}$ and $\frac{1}{64}$ of the image area produce results not importantly different from that using $\frac{1}{16}$ of the image, because the image appearance changes only slowly with ECR area. If the ECR is much less than $\frac{1}{64}$ of the image, the contrast becomes too sensitive to very local variations and, in particular, to image noise. This oversensitivity to local variations can cause artifacts, which have never been experienced with the preferred ECRs.

With these values for ECR interpolative ahe of a $512 \times 512$ image requires less than two minutes for a C program on a VAX 11/780 or an assembly language program on a small 16-bit minicomputer. This is a savings of well over an order of magnitude over ahe with full sampling.

### 2.2. *Contextual Region Sampling Rate*

We have evaluated sampling in each dimension at a distance equal to the contextual region linear dimension (so that the sample regions divide the image into a mosaic—see Fig. 3a), at half this distance (see Fig. 3b), and at one pixel (full sampling). This advantage of sampling at half the contextual linear dimension is that, unlike with mosaic sampling, each pixel contributes to the histograms of all four sample contextual regions whose mappings are applied to that pixel [4].

Of course, the coarser the sampling, the faster the results are computed. Although test pattern images can be created where finer than mosaic sampling is desirable to produce adequate quality, we have never found an image of the complexity of clinical medical images for which mosaic sampling was not effectively equivalent to the finer sampling for the same ECR.

## 3. ALGORITHMS FOR SPECIAL IMAGE PROCESSING DEVICES

### 3.1. *Feedback Architecture*

Image processing devices with a feedback architecture, such as those manufactured by Comtal, DeAnza, and Vicom, can do simple operations on one or more whole images in a frame display time, commonly $\frac{1}{30}$ s. The following algorithm for interpolated ahe appears to be especially well suited to such devices. The algorithm

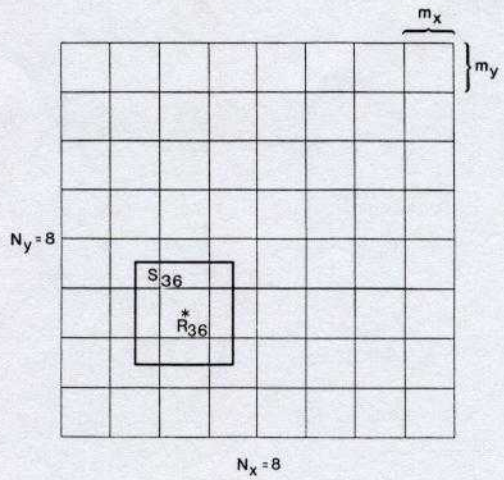


FIG. 4. Region and parameter definitions for Program 1. $R_{36}$ is a contextual region, and $S_{36}$ is the corresponding mapping region. $N_x = N_y = 8$ is equivalent in ECR to full ahe with $N_x = N_y = 4$.

is based on computing and applying each histogram equalization mapping from a contextual region $R_{ij}$ before moving on to the next. After all of the mappings have been applied, each pixel will contain the result of each of the four mappings applicable to it. A bilinearly weighted average of these four results is then calculated at each pixel. We will assume mosaic sampling, although the program can be modified to work for other samplings.

With mosaic sampling and bilinear interpolation of the mapping, the region made up of pixels to which a given mapping must be applied is a rectangle concentric with the corresponding contextual region but twice its size in each dimension (see Fig. 4). Let us call this region the *mapping region*, $S_{ij}$, corresponding to the $m_x \times m_y$ contextual region $R_{ij}$ and the mapping $L_{ij}$. Let $i = 1, 2, \ldots, N_x$ index the contextual regions in the $x$ dimension, and $j = 1, 2, \ldots, N_y$ index the contextual regions in the $y$ dimension.

Consider the following four sets of mapping regions: $\{S_{ij} | i \text{ odd}, j \text{ odd}\}$, $\{S_{ij} | i \text{ even}, j \text{ odd}\}$, $\{S_{ij} | i \text{ even}, j \text{ odd}\}$, $\{S_{ij} | i \text{ even}, j \text{ even}\}$. Each of these form an image that is a mosaic of mapped results from alternate contextual regions in each dimension (see Fig. 5). We name these four intermediate mosaic images: $M_{kl}(x, y)$, for $k = 0, 1$ and $L = 0, 1$; oddness or evenness of $k$ and $l$, respectively, corresponds to $S_{ij}$ with odd or even $i$ and $j$, respectively. The four mapped values $M_{kl}(x, y)$ at pixel $x, y$ are the four values to be combined with bilinear weights to produce the final value $M(x, y)$. The borders of the images, where some $M_{kl}$ is undefined must be treated specially.

In the center region $A$ where all of the $M_{kl}$ are defined (see Fig. 5), the bilinear weights as a function of $x, y$ are the same for each intermediate image $M_{kl}(x, y)$ except that each is shifted with respect to the other, in $x$ for different $k$ and in $y$ for different $l$, by the distance between two adjacent contextual region centers in the respective dimension. Also, for any $k, l$ the bilinear weights are cyclic with a period in each dimension equal to twice the distance between two adjacent contextual region centers in the respective dimension, $2m_x$ or $2m_y$. The weight function for one

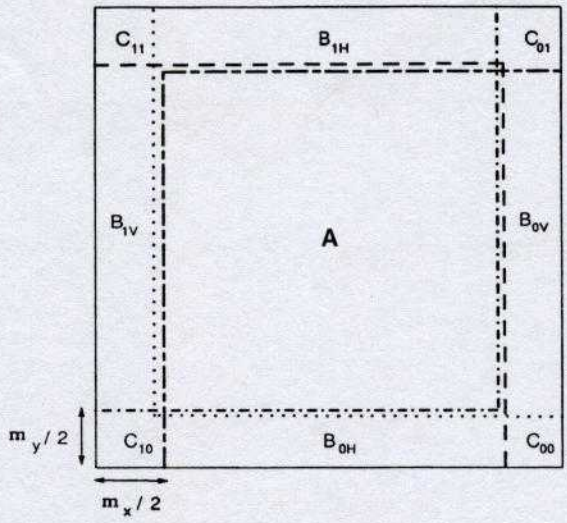


FIG. 5. The image area divided into the mosaic images $M_{00}$ (----), $M_{01}$ ( $\cdots$ ), $M_{10}$ (--), and $M_{11}$ (-·-), their common center area $A$, their border regions, $B_{kH}$ and $B_{kV}$, and the corner regions $C_{kl}$.

period is the product of triangles in $x$ and $y$, each going from 0 to 1 and then back to 0 across one period: the full two-dimensional period is defined by

$$xyperiod(x, y) = xperiod(x) * yperiod(y), \text{ where}$$
$$xperiod(x) = x/m_x, \text{ for } x \text{ between 0 and } m_x,$$
$$xperiod(2 * m_x - x) = xperiod(x), \text{ for } x \text{ between 1 and } m_x - 1,$$

and $yperiod(y)$ is defined by a similar expression, with $y$ replacing $x$ throughout.

Assume that a single two-dimensional period of the weights has been computed. Application of these weights can be accomplished directly by reference to this period, or from an image $W(x, y)$ consisting of $(N_x + 1)/2 \times (N_y + 1)/2$ periods of $xyperiod(x, y)$. The weighting functions for the $M_{kl}$ are each a subimage of $W(x, y)$, each specifiable by its upper left pixel (origin) in $W$.

In the $m_x/2 \times m_y/2$ corners of the image, $C_{kl}$, the result image has the values of the mosaic images $M_{kl}$. In the remaining border areas $B_{kH}$ and $B_{kV}$, the result is a weighted average of the two $M_{kl}$ that overlap in that border, with weights respectively being $xperiod(x)$ and a shifted version of $xperiod(x)$, for horizontal border areas, and $yperiod(y)$ and a shifted version of $yperiod(y)$, for vertical border areas.

Letting $I(x, y)$ be the intensity at $x, y$ in the input image, and assuming the "image" $W(x, y)$ as well as the arrays $xperiod(x)$ and $yperiod(y)$ have already been computed, the program to apply ahe on feedback processors is summarized in Program 1.

PROGRAM 1. Ahe for feedback processors.

```
/* Compute and apply mappings, histogram by histogram. */
k = 0
For i = 1 to N_x
        {k = k + 1 (modulo 2); l = 0
```

```
        For j = l to N_y
                {l = l + 1 (modulo 2)
                Compute H = histogram (R_ij)
                Compute L = lookup table which is histogram equalization
                    mapping corresponding to H = (cumulative
                    histogram * output display range/(m_x m_y))
                For all x,y ∈ S_ij
                        {M_kl(x, y) = L(I(x, y))}
}}

/* Weight temporary images by modified bilinear weighting function, and sum
   results. */
Zero M(x, y)
For k = 0 to 1 {For l = 0 to 1
        /* Multiply M_kl by the appropriate part of W, and add the result into
           M */
        {For all x,y
                {M(x, y) = M(x, y) + M_kl(x, y) * W(m_x/2 + k * m_x + x,
                                                    m_y/2 + l * m_y + y)}
}}

/* Fix up corners and borders. */
For k = 0 to 1 {For l = 0 to 1
        {For all x,y ∈ C_kl
                {M(x, y) = M_kl(x, y)}
        }
        For all x, y ∈ B_kH
                {M(x, y) = xperiod(m_x/2 + x) * M_k0(x, y)
                           + xperiod(3 * m_x/2 + x) * M_k1(x, y)}
        For all x, y ∈ B_kV
                {M(x, y) = yperiod(m_y/2 + y) * M_0k(x, y)
                           + yperiod(3 * m_y/2 + y) * M_1k(x, y)}
}
```

Note that only two consecutive rows of contextual regions are necessary to compute $m_y$ pixel rows of the final result $M$. Each additional consecutive row of contextual regions suffices to complete the computation of $m_y$ additional pixel rows of the $M_{kl}$ and thus $M$, as well as the computation of $m_y$ pixel rows of two additional $M_{kl}$. Thus with the storage necessary for $2m_y$ pixel rows of each of the $M_{kl}$, $M$, and $I$, the ahe result could be successively computed. Therefore, this method can apply ahe to an image $N_y/2$ times as large as would be needed if Program 1 were applied directly. In fact, since one could step one contextual region at a time in the horizontal direction, as well, while handling the second $m_y$ pixel rows of the various images, the factor by which the image size could be increased could be even $N_x N_y/(N_x + 2)$.

The speed of Program 1 on a feedback processor depends on whether each pixel in a single pass through the image can contribute to or be mapped by a different mapping or all the pixels must share the same mapping. On the latter, more conservative assumption, $N_x N_y$ passes through the image would be required to

calculate all of the histograms, and the same number of passes would be required to apply the mappings. Four additional passes would be necessary to compute the bilinearly weighted $M_{kl}$, and three more to sum these results. Neglecting the time in computing the mappings from the histograms, $2N_xN_y + 7$ passes would be required; for the common value for mosaic-sampled interpolative ahe of $N_x = N_y = 8$ and pass time of $\frac{1}{30}$ s, about 4.5 s would be required to apply this algorithm. If each pixel could contribute to or be mapped by a different mapping in a single pass through the image, the histograms could be computed in one pass and the mappings applied in four passes (both independent of $N_x$ and $N_y$), for a total of 12 passes or about 0.4 s.

### 3.2. *Processor-per-Pixel Architecture*

Another interesting architecture involves a small processor at each pixel and the ability to broadcast a value to all pixels simultaneously [1]. With such devices the following algorithm for uninterpolated ahe, based on computing each pixel's rank in its contextual region, is very fast.

Let $Q_{xy}$ be the set of pixels $u, v$ for which the pixel at $(x, y)$ is in the contextual region of that at $(u, v)$. (For contextual regions that are symmetric, $Q_{xy}$ is the same as the contextual region about $(x, y)$.) Let $\text{rank}_{xy}$ be the rank of pixel $x, y$ in its contextual region, the value to be computed.

PROGRAM 2. Ahe for per-pixel parallel processors.

```
For x, y in the image
        {Zero rank_xy}
For x, y in the image
        {For u, v in Q_xy
                {If i(x, y) ≤ i(u, v
                        then rank_uv = rank_uv + 1}
        }
```

The total algorithm takes a time proportional to the number of pixels, if one uses an engine in which each pixel value $I(x, y)$ is broadcast in parallel to all the other pixels, and those in $Q_{xy}$ which have greater intensity values have a rank counter incremented in parallel. We presently have a design of a VLSI-based engine that could accomplish ahe for a $512 \times 512$ image in under 1 s. This engine can operate with a memory large enough to hold only $m_y$ pixel rows of the image, i.e., 1 $N_y$ of the total image.

## 4. QUALITY IMPROVEMENTS

### 4.1. *Weighted Ahe*

It seems unattractive for the contextual region of ahe to end abruptly. If it does, one pixel is in the region, affecting the mapping of the pixel at the center of the region, and its neighbor has no effect on that mapping. Furthermore, it seems reasonable that the pixels near the pixel whose mapping is being calculated (the "center" pixel) should affect the mapping more than those farther away. Therefore, we have created and evaluated a form of ahe in which pixels' contribution to a histogram decreases with their distance from the center pixel.

In this weighted ahe the ECR was calculated with the area of each pixel weighted by $Nw_i/W$, where $N$ is the number of pixels in the contextual region, $w_i$ is the weight the pixel contributes to the histogram, and $W$ is the sum of the $w_i$. This method of calculating the ECR proved to give a value that made the effect of a contextual region with a given weighting scheme very close to that of an unweighted contextual region with the same ECR.

Weighted ahe with a conical weighting function was applied to a number of CT scans of various types, including those with sharp, strong boundaries. Little difference was noticeable when compared to ordinary ahe with the same ECR. Because weighted ahe is much more time consuming, we do not recommend its use.

### *4.2. Contrast Limitation*

Ahe has produced excellent results in enhancing the signal component of an image, but noise in the image is enhanced, too. There has been considerable debate about whether or not enhancing noise is really a problem. Controlled tests with simple test patterns indicate that enhancing noise proportionately with signal does not impair an observer's ability to detect information in an image [2]. However, clinicians who routinely examine images have indicated that noise enhancement is very disturbing and does cause problems. We decided to investigate the effects of limiting contrast enhancement in cases when the noise would otherwise become too apparent. This occurs when the range of image intensities in a contextual region is not a good deal greater than the noise level, i.e., in relatively homogeneous regions.

Contrast enhancement can be defined as the slope of the function mapping input intensity to output intensity (see Fig. 6). We will assume that the range of input and output intensities are the same. Then a slope of 1 involves no enhancement, and higher slopes give increasingly higher enhancement. Thus the limitation of contrast enhancement can be taken to involve restricting the slope of the mapping function.

With histogram equalization the mapping function $m(i)$ is proportional to the cumulative histogram (e.g., [3]):

$$m(i) = (Display\_Range) * (Cumulative\_Histogram(i)/Region\_Size).$$

Since the derivative of the cumulative histogram is the histogram, the slope of the mapping function at any input intensity, i.e., the contrast enhancement, is proportional to the height of the histogram at that intensity:

$$dm/di = (Display\_Range/Region\_Size) * histogram(i).$$

Therefore, limiting the slope of the mapping function is equivalent to clipping the height of the histogram.

High peaks in the histogram are normally caused by nearly uniform regions. In such a case, with the mapping due to ordinary histogram equalization a narrow range of input intensity values is mapped to a wide range of output intensity values, perhaps overenhancing the noise. But enforcing a maximum on the counts in the histogram will limit the amount of contrast enhancement and thus the enhancement of noise.

When contrast enhancement is reduced at one location it must be increased in other areas so that the entire input intensity range will be mapped to the entire output intensity range. This corresponds to renormalizing the histogram after

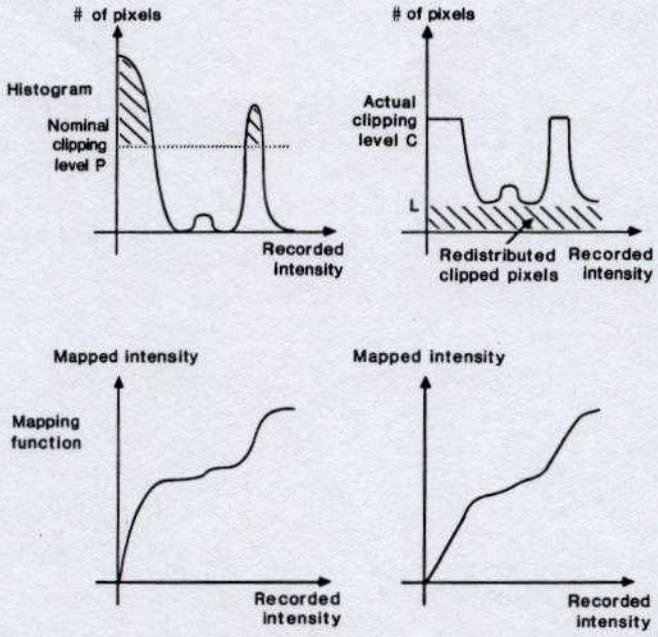


FIG. 6. Contrast mapping functions and their associated original and clipped histograms.

clipping so that its area returns to its original value. We think of this as redistributing the clipped pixels.

We have tried two means of redistributing the clipped pixels: uniformly distributing them in all histogram bins, and distributing them into bins with contents less than the clipping limit in proportion to their contents. The latter technique shares the intuitive advantage with ahe that contrast enhancement is in proportion to need for contrast enhancement, but it is complex and results in an undesired property: that the mapped intensity throughout the image can be strongly changed by moving one pixel from its original intensity to another in a bin that was formerly empty. Therefore, we have chosen the option of uniform redistribution of clipped pixels—across the full intensity range of the *whole* image. This option can be thought of as adding the contrast mapping due to the clipped histogram to a linear mapping that achieves just the height at the maximum image intensity such that the height of the sum is equal to the intensity range in the original image (see Fig. 6).

Incorporation of histogram clipping into the existing ahe algorithm is straightforward. One need only insert a histogram modification step into the algorithm. After each histogram for a contextual region is computed, it is clipped to some value before the mapping function is computed from it by calculating a cumulative histogram or, equivalently, ranks.

The user determines the clipping limit by specifying the limiting slope $S$ of the intensity mapping. The clipping limit $C$ can be shown to be $S$ times the average histogram bin contents, since a slope of 1 corresponds to all bins having the same (average) value and slope is proportional to the value of a histogram bin.

Adding a uniform level $L$ to the clipped histogram will push the clipped histogram again above the clipping limit, so the original histogram needed to be

clipped at a lower limit $P$ such that $P + L(P)$ is equal to the clipping limit ($L$ is written as a function of $P$ because it depends on $P$). The value of $P$ that satisfies this equation can be found by the binary search given in Program 3.

The resulting value of *bottom* is an integer, to which the remaining excess $S$ divided by

PROGRAM 3. Calculation of actual clipping limit.

```
Let C be the clipping limit and R the number of histogram bins in the total image
top = C
bottom = 0
while (top − bottom > 1)
        {middle = (top + bottom)/2
                S = sum over all histogram bins of the excess in that
                        bin over middle, if any
                if S > (C − middle)* R
                        then top = middle
                        else bottom = middle
        }
```

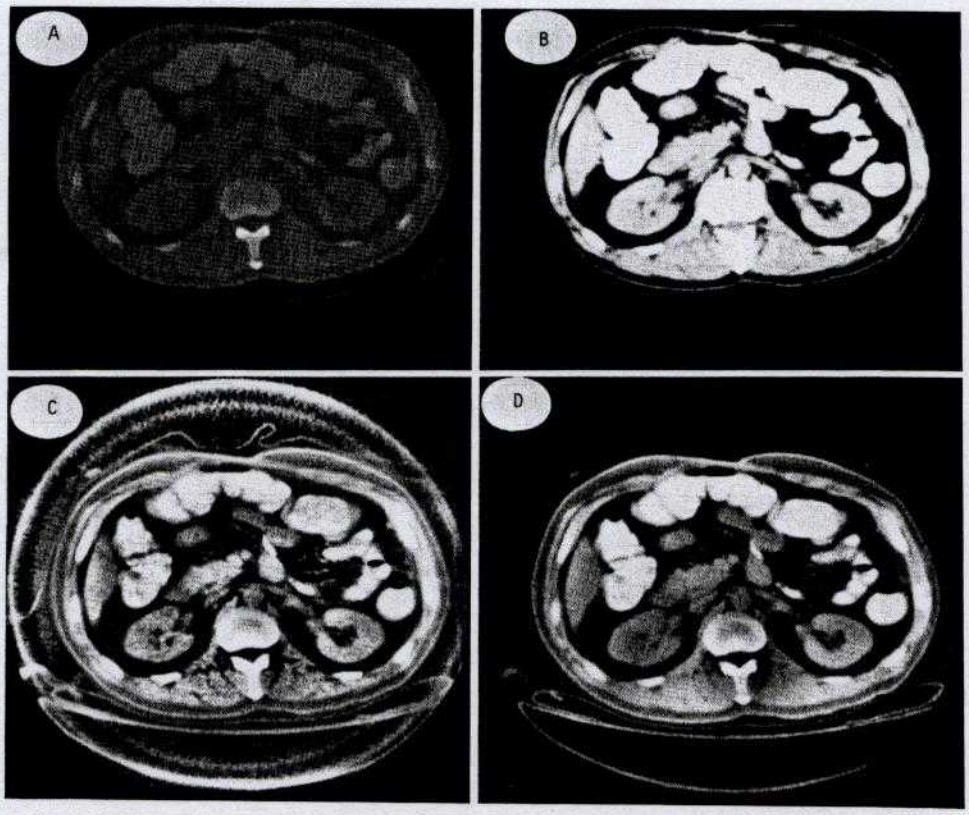


FIG. 7. (A) Original image; (B) interactively windowed result; (C) unclipped ahe result; (D) clipped ahe result with clipping limit 10, for a CT of the abdomen.

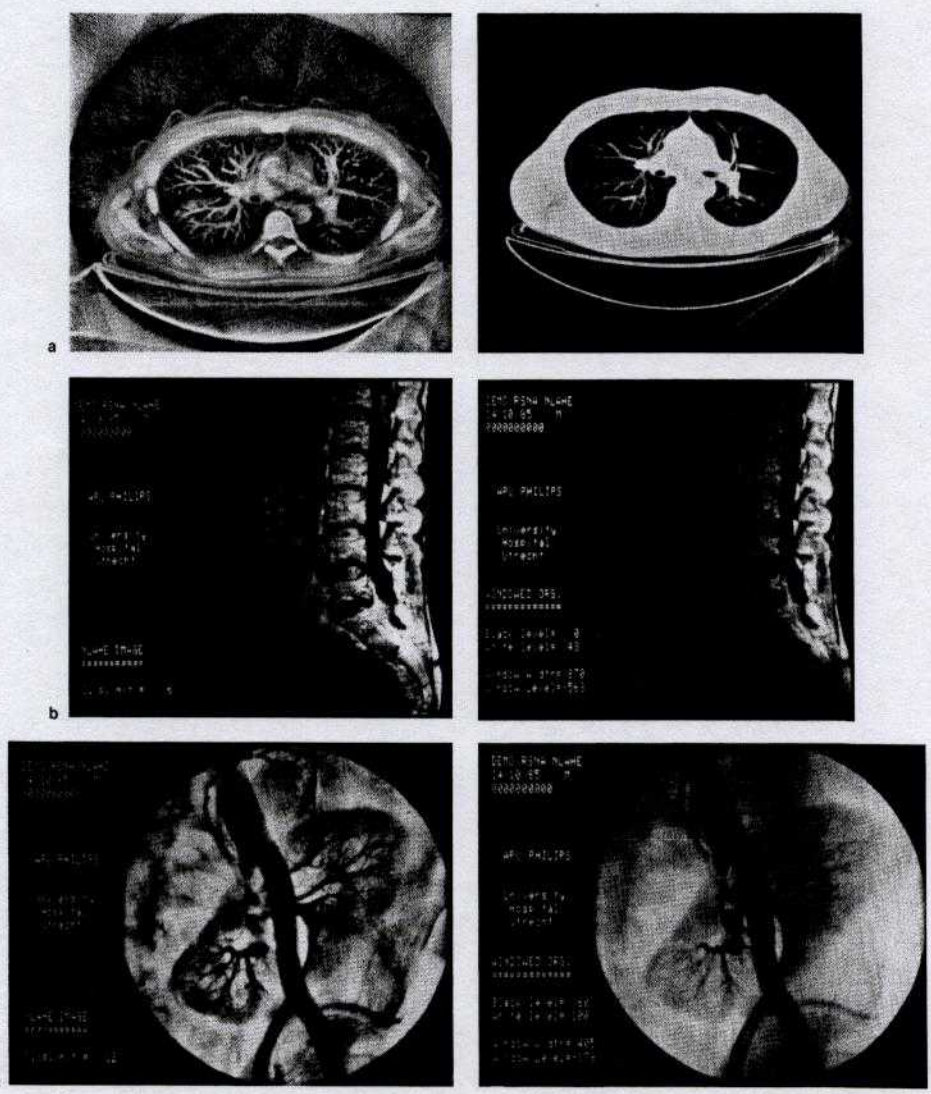

FIG. 8. Results of clipped ahe (left) vs results of intensity windowing (right): (a) CT of chest; (b) surface coil MRI of spine; (c) X-ray angiogram.

the number of bins $R$ must be added to produce the desired $P$. $L$ is equal to the clipping limit minus $P$. Then the modified histogram value $v$ in any bin is calculated from original value $v_{\text{orig}}$ by

$$v = \begin{cases} v_{\text{orig}} + L & \text{if } v_{\text{orig}} < P \\ C & \text{if } v_{\text{orig}} \geq P. \end{cases}$$

The clipped ahe technique has been applied to several different medical images, and the results to date have been encouraging. In Fig. 7 an original image and

images processed with ordinary and clipped ahe, and interactive windowing are compared. Figure 8 compares interactively windowed images with those processed with clipped ahe. Note not only the contrast in all organs simultaneously, but also the ability to lessen the effects of nonuniform sensitivity, as in MRI images from surface coils.

## 5. SUMMARY

Numerous improvements to adaptive histogram equalization in speed or quality have been suggested. As for quality, clipped ahe has had great success in both

(a) showing in a single image all contrast in electronically recorded images whose range is too wide for a nonadaptive mapping to succeed, and

(b) showing contrast hidden in images initially recorded on film.

This method seems to have the potential of being applicable to all medical images, although the clipping level must vary (apparently in a presettable way) with the imaging modality, body region imaged, and imaging variables. The method has been used with considerable success with light images as well.

As for speed;

(a) If you have a general purpose computer, interpolative clipped ahe with mosaic sampling is the method of choice, requiring a few minutes per megapixel on common minicomputers.

(b) If you have access to a system with a feedback processor, a considerable increase in speed can be obtained by using Program 1, with each histogram calculation followed by the clipping step before the corresponding mapping is computed. This requires around 20 s per megapixel with standard feedback processors doing arithmetic on a full frame in $\frac{1}{30}$ s.

(c) Looking toward the future, a VLSI-based processor-per-pixel design seems to be very attractive, because not only can it compute ahe in a few seconds per megapixel, but also it can do a wide range of other image processing operations fast [1].

## ACKNOWLEDGMENTS

We thank Paul F. G. M. van Waes, John R. Perry, Julian G. Rosenman, and Edward V. Staab for their collaboration on clinical application of ahe. We are indebted to Joan Savrock for manuscript preparation, to Leigh Pittman for figure preparation, and to Bo Strain and Karen Curran for photography. This paper was prepared with the partial support of NIH Grants R01-CA39059 and R01-CA39060.

## REFERENCES

1. J. D. Austin and S. M. Pizer, "An Architecture for Fast Image Processing," Technical Report TR86-002, Dept. of Computer Science, University of North Carolina, 1986.
2. A. E. Burgess, R. F. Wagner, and R. J. Jennings, Human signal detection performance for noisy medical images, in *Proceedings, IEEE Computer Society, International Workshop on Physics and Engineering in Medical Imaging*, Asilomar, March 10–18, 1982, pp. 99–105.
3. K. R. Castleman, *Digital Image Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1979.
4. G. Herman, personal communication, 1984.

5. R. A. Hummel, Image enhancement by histogram transformation, *Comput. Graphics Image Process.* **6**, 1977, 184–195.
6. D. J. Ketcham, R. W. Lowe, and J. W. Weber, Real-time image enhancement techniques, in *Seminar on Image Processing*, Hughes Aircraft, Pacific Grove, California, 1976, pp. 1–6.
7. S. M. Pizer, Intensity mappings for the display of medical images, in *Functional Mapping of Organ Systems and Other Computer Topics*, Society of Nuclear Medicine, 1981.
8. B. ter Haar Romeny, S. M. Pizer, K. J. Zuiderveld, J. B. Zimmerman, E. P. Amburn, A. Geselowitz, P. F. G. M. van Waes, and A. de Goffau, "Recent Developments in Adaptive Histogram Equalization," Exhibit at 71st Scientific Assembly and Annual Meeting of the Radiological Society of North America, November 17–22, 1985.
9. J. B. Zimmerman, *Effectiveness of Adaptive Contrast Enhancement*, Ph.D. dissertation, Department of computer Science, University of North Carolina, 1985.
10. J. B. Zimmerman and S. M. Pizer, Evaluation of the effectiveness of adaptive histogram equalization, in *Proceedings, Soc. of Photographic Scientists and Engineers, 25th Fall Symposia—Imaging, November 17–22, 1985*, pp. 189–190.