# Real-Time Digital Image Enhancement

R. E. WOODS, MEMBER, IEEE, AND R. C. GONZALEZ, MEMBER, IEEE

*Invited Paper*

*Abstract*—A programmable system for enhancing monocular and stereographic images at video rates is presented. The system provides both automatic and interactive enhancement modes based on histogram modification and intensity mapping techniques. Experimental results which illustrate enhancement capabilities under a variety of scene types and conditions are described.

## I. Introduction

REAL-TIME enhancement techniques have application in areas of digital image processing that involve human operator interaction and in autonomous systems with high data throughput requirements. In interactive applications, such as biomedical image analysis [1] and industrial inspection [2], real-time techniques are important not only in terms of improving productivity, but also in reducing operator errors associated with visual feedback delays.

In autonomous applications, such as robotics and military systems [3], [4], image enhancement techniques are often used in a preprocessing stage in order to increase the probability of correct pattern detection and recognition. These applications are generally characterized by data throughput requirements that can only be met by hardware capable of operating in real time.

In this paper we discuss the design and implementation of a programmable system for enhancing monocular and stereographic video images. The system operates in real time in the sense that its throughput delay is well below the standard television refresh rate of 30 frames/s. It is capable of simultaneously processing two video sources with a combined resolution of $2^Q$ pixels per scan line, with $Q$ being an integer in the range $5 \leqslant Q \leqslant 10$. The intensity resolution of each digitized video input is 256 discrete levels.

The system is capable of automatic and interactive enhancement based on histogram equalization, function processing, and histogram specification techniques. Experimental results are included to demonstrate system performance under a variety of scene types and conditions.

## II. Enhancement Approach

Digital image enhancement techniques may be divided into two principal categories: 1) transform-domain methods and 2) spatial-domain methods [5]. Approaches based on the first category consist basically of computing a two-dimensional transform (e.g., Fourier or Hadamard transform) of the image to be enhanced, altering the transform, and computing the inverse to yield an image that has been enhanced in some manner. Examples of transform-domain techniques include low- and high-pass filtering for image smoothing and sharpening, respectively, and homomorphic filtering for manipulating the effects of illumination and reflectance in an image [5], [6].

Spatial-domain techniques consist of procedures that operate directly on the pixels of the image in question. Examples of spatial-domain enhancement techniques include smoothing by neighborhood averaging, sharpening by using gradient-type operators, and global enhancement by means of histogram modification techniques [5]–[9]. The enhancement procedures discussed in this paper deal with applications of the latter approach to video images.

Suppose that each frame of a time-varying video signal to be processed is digitized to form an $N \times M$ array of pixels. Let $x$ be a variable which represents the gray level (intensity) of each pixel. It is assumed for simplicity that $x$ has been normalized to the interval $0 \leqslant x \leqslant 1$, where 0 denotes black and 1 denotes white in the gray scale. Attention will be focused in the following sections on transformations of the form

$$y = T(x), \qquad 0 \leqslant x \leqslant 1 \tag{1}$$

which map a gray level $x$ into a level $y$.

### A. Histogram Modification Techniques

*1) Foundation:* The gray-level mapping methods developed in this section are based on transforming the probability density function of the gray levels in an image to be enhanced. The density function of the levels in the original image will be denoted by $p_x(x)$, while $p_y(y)$ will be used to denote its counterpart in the enhanced image. Although these quantities are discrete functions for a digital image, the following development will first be carried out in continuous mathematics to simplify the explanation. The discrete equivalents will then be obtained by straightforward extensions of these results.

The functions $p_x(x)$ and $p_y(y)$ are of fundamental importance in describing the visual characteristics of the original and enhanced images. For example, the average brightness of the original image is given by

$$\bar{x} = \int_0^1 x p_x(x)\, dx \tag{2}$$

while the variance of the intensity (which is a measure of contrast) is given by

$$\sigma_x^2 = \int_0^1 (x - \bar{x})^2 p_x(x)\, dx. \tag{3}$$

For a given image, we are interested in obtaining an enhanced image with a specified density function $p_y(y)$. The transformation $T$ is then determined from the desired $p_y(y)$.

The transformation function $y = T(x)$ and its inverse $x = T^{-1}(y)$, are guaranteed to be strictly monotonically increasing

Manuscript received May 19, 1980; revised August 28, 1980. This work was supported by NASA under Contract NAS8-29271.

The authors are with the Electrical Engineering Department, University of Tennessee Knoxville, TN 37916.

0018-9219/81/0500-0643$00.75 

in the interval [0, 1] if the function $T(x)$ is assumed to satisfy

a) $T(x)$ is single-valued and strictly monotonic in the interval $0 \leqslant x \leqslant 1$, and
b) $0 \leqslant T(x) \leqslant 1$ for $0 \leqslant x \leqslant 1$. (4)

The monotonicity condition preserves the order from black to white in the gray scale of the enhanced image, while condition b) guarantees a mapping that will be consistent with the allowed range of pixel values. Under these conditions, $p_y(y)$ can be written in terms of $p_x(x)$ and $T(x)$ as follows [10]:

$$p_y(y) = \left[ p_x(x) \frac{dx}{dy} \right]_{x = T^{-1}(y)}. \tag{5}$$

The function $p_x(x)$ is obtained from the original image, $p_y(y)$ is specified, and the problem is to determine the transformation function $T(x)$ which will yield the desired $p_y(y)$.

A density equalization technique [6]–[9] is obtained from (5) by using the transformation function

$$y = T(x) = \int_0^x p_x(s)\, ds, \qquad 0 \leqslant x \leqslant 1 \tag{6}$$

which is the cumulative distribution function of $x$. The variable $s$ in (6) is a dummy variable of integration. From this equation we have that $dx/dy = 1/p_x(x)$, and (5) reduces to

$$p_y(y) = 1, \qquad 0 \leqslant y \leqslant 1. \tag{7}$$

In other words, the use of (6) yields an image whose gray levels have a uniform density. Intuitively, it is not unreasonable to assume that the quality of an image is improved when its pixels are uniformly distributed over the entire gray-level range.

Although density equalization can be quite useful in some applications, this particular method is not suited for interactive image enhancement since all it can do is produce a density function $p_y(y)$ that is uniform. As will be seen below, however, this approach can be used as an intermediate step in a transformation which will actually yield a specified $p_y(y)$. This transformation procedure is referred to as density specification [5], [9].

Suppose that the gray levels $x$ of an image are transformed using (6) to yield a new set of levels $z$; that is,

$$z = H(x) = \int_0^x p_x(s)\, ds. \tag{8}$$

From the above discussion, we have that $p_z(z)$ is a uniform density function. If the inverse of (8), $H^{-1}(z)$, is applied to the $z$'s, we obtain the original image with $p_x(x)$ back. It is noted, however, that if we specify a density function $p_y(y)$ and apply (6) we would obtain

$$z = G(y) = \int_0^y p_y(s)\, ds. \tag{9}$$

Although $G(y)$ is in general different from $H(x)$, $p_z(z)$ is the same if either (8) or (9) is used. If we now apply the inverse function $G^{-1}(z)$ to the $z$'s, the result would be a set of $y$ levels with the specified density $p_y(y)$. In other words, if the $z$'s have a uniform density, the desired $p_y(y)$ can be obtained by using the inverse mapping $G^{-1}(z)$. Therefore, if the original image is first density equalized and the new (uniform) levels are inverse mapped using the function $G^{-1}(z)$, the result would be an image whose gray levels have the desired density $p_y(y)$.

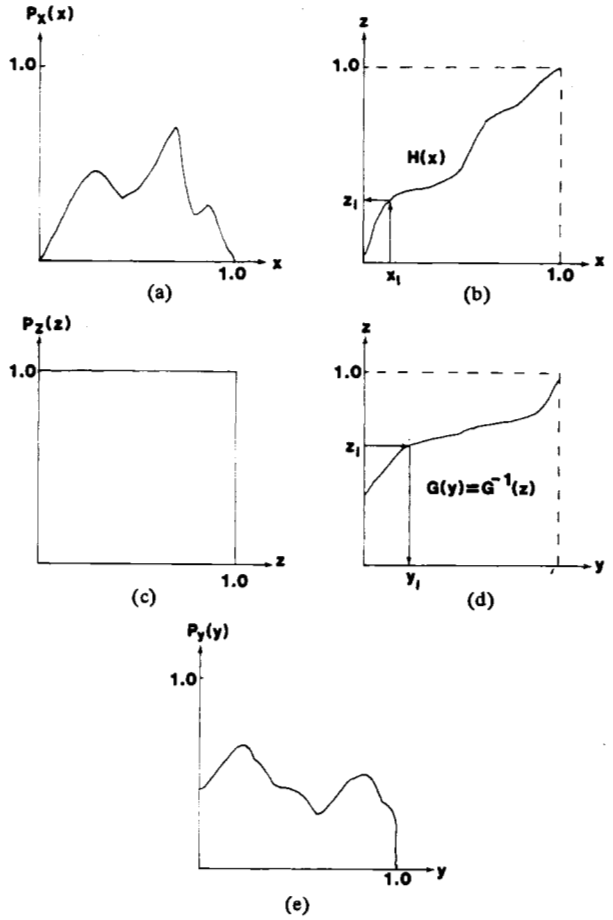


Fig. 1. Histogram specification procedure.

The above results can be expressed in terms of a transformation function $T(x)$ from $x$ to $y$ by noting that $y = G^{-1}(z)$ and, since $z = H(x)$,

$$y = T(x) = G^{-1}\,[H(x)] \tag{10}$$

The density equalization technique is a special case of (10) obtained by letting $G^{-1}\,[H(x)] = H(x)$.

The transformation procedure is summarized in Fig. 1. Fig. 1(a) shows the original density function which is calculated from the image to be enhanced. Fig. 1(b) shows the transformation function $H(x)$ which maps $x$ into $z$. Fig. 1(c) is the result of this transformation. Fig. 1(d) shows the inverse transformation from $z$ to $y$. The result of this transformation, shown in Fig. 1(e), is the desired probability density function $p_y(y)$. The problem in using (10) for continuous variables lies in obtaining the inverse function analytically. In the discrete case, this problem is circumvented by the fact that the number of distinct gray levels is usually relatively small, and it becomes feasible to calculate and store a transformation value for each discrete pixel value.

*2) Discrete Formulation:* For gray levels that assume discrete values, we deal with probabilities defined by the equation

$$p_x(x_k) = \frac{n_k}{n}, \qquad 0 \leqslant x_k \leqslant 1,$$

$$k = 0, 1, \cdots, L-1 \tag{11}$$

where $L$ is the number of levels, $p_x(x_k)$ is an estimate of the probability of the $k$th gray level, $n_k$ is the number of times this level appears in the digital image, and $n$ is the total number of pixels in the image. A plot of $p_x(x_k)$ versus $x_k$ for $k = 0, 1, \cdots, L-1$, is usually called a *histogram*, and the equalization procedure obtained from a discrete formulation of (6) is called *histogram equalization*. The terms *histogram linearization* and *histogram flattening* are also often used to describe this method.

The discrete form of (6) is given by the equation

$$y_k = T(x_k) = \sum_{j=0}^{k} \frac{n_j}{n}$$

$$= \sum_{j=0}^{k} p_x(x_j), \qquad 0 \leq x_j \leq 1$$

$$k = 0, 1, \cdots, L-1 \qquad (12)$$

and the inverse transformation is given by $x_k = T^{-1}(y_k)$. Similarly, the discrete equivalent of the density specification technique is referred to as *histogram specification* and is given by the following equation:

$$y_k = T(x_k) = G^{-1}[H(x_k)], \qquad 0 \leq x_k, y_k \leq 1$$

$$k = 0, 1, \cdots, L-1. \qquad (13)$$

Since the gray levels are predefined discrete quantities, (e.g., 0 to 255 for 8 bits of intensity resolution), equations (11)–(13) are normally scaled to the range of allowed values (instead of the interval [0, 1]) and the transformation functions are rounded off to the closest integer value in this range. This allows all possible discrete integer values of the transformation functions to be computed and stored in a look-up table for use in mapping all the pixels in the input image. It is also worth noting that the discrete formulations given in (12) and (13) do not guarantee a perfectly uniform or specified histogram. The reason for this is based on the fact that transformations of the form $y_k = T(x_k)$, for example, do not allow redistribution of pixels in the histogram; that is, *all* pixels with value $x_k$ are assigned the value $y_k$. The power of the technique lies in the fact that $x_k$ and $y_k$ are in general different so that $p(x_k)$ and $p(y_k)$ correspond to different values in the intensity scale, causing spreading and shrinking effects in the original histogram. Since the transformation function is computed from a known histogram shape, the result is that the modified histogram has a tendency to approach the shape of the specified histogram in terms of the intensity distribution of the latter.

### B. Specification of Histograms

There are two principal approaches to the problem of specifying histograms for image enhancement. The first involves *a priori* knowledge about the desired histogram shape based on some known properties of the image (or class of images) to be enhanced. The second is to specify the histogram interactively and then evaluate the results visually until a "good" enhanced image is obtained.

An interesting example of the first approach is found in some of the digital processing techniques for improving the quality of radiographs [12]. By studying a set of radiographs of the same type (e.g., chest radiographs) that have been developed properly, it is possible to arrive at an average or representative histogram for that particular type of image. The histogram

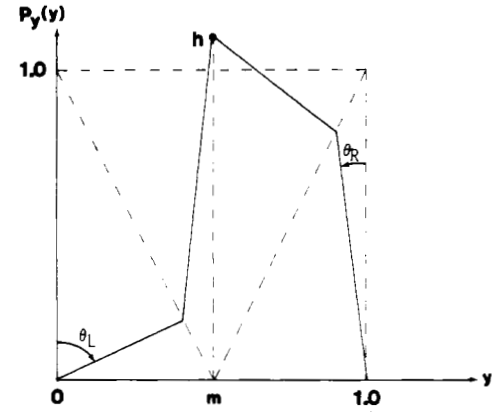


Fig. 2. A four-parameter approach for specifying $p_y(y)$.

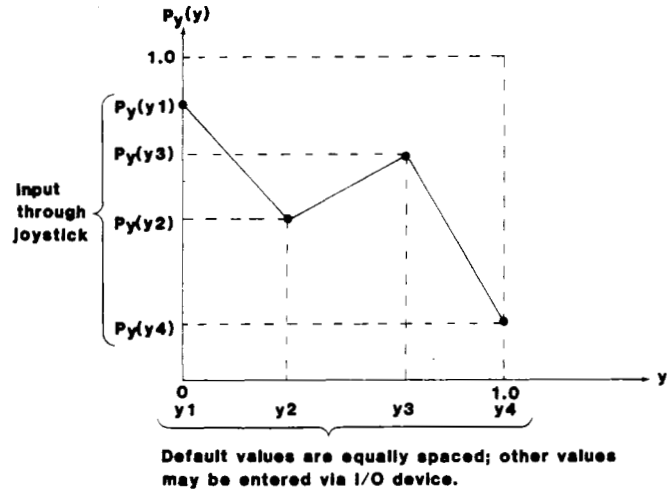


Fig. 3. Another approach for specifying $p_y(y)$.

found in this manner can then be used in the histogram specification method of (13) for mapping radiographs of the same type, but which have been improperly developed. The resulting images will then have histograms which resemble more closely the histogram representative of the control set.

When no *a priori* information exists about the image to be enhanced, it is often possible to specify a histogram interactively by using a prespecified set of parameters that define the shape of a desired histogram [9]. A four-parameter approach for specifying histograms is shown in Fig. 2. The first parameter, denoted by $m$, is related to the mean of the pixel values and thus controls average brightness. The second parameter, denoted by $h$, specifies the peakedness about $m$ and is thus related to the kurtosis of the histogram. The third and fourth parameters, labeled in the figure as $\theta_R$ and $\theta_L$, control the spread and symmetry of the histogram. The spread is related to the variance and thus controls image contrast. The symmetry controls the brightness bias toward dark or light shades of gray.

Another procedure for specifying histograms is shown in Fig. 3. In this case the specified parameters are $p_y(y_1)$ through $p_y(y_4)$, with the points along the $y$-axis being fixed. This approach is simpler to implement from a computational point of view, but has the disadvantage that the specified parameters do not have the physical meaning associated with those used in Fig. 2.

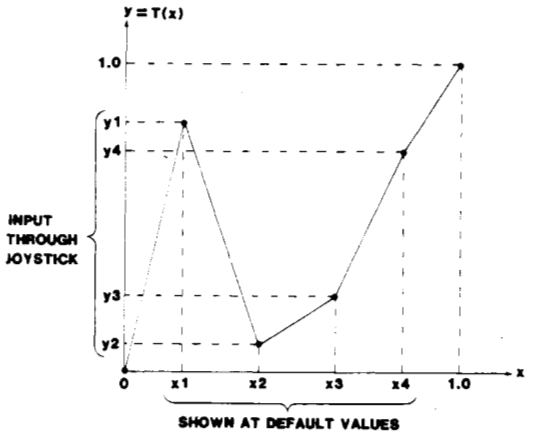


Fig. 4. A method for specifying $T(x)$.

Although the number of parameters for specifying the histogram in Figs. 2 and 3 could easily be increased, we have found that more than four parameters are generally difficult to control and interpret interactively, particularly when enhancement speed is an important consideration.

### C. Direct Specification of $T(x_k)$

The gray-level mapping methods described in this section are based on directly specifying the transformation $T(x_k)$ of (12). The transformations need not conform to the monotonicity condition of (4) and are independent of the gray-level content of the image being processed. Procedures for generating such transformations are commonly referred to as function processing algorithms.

A four-parameter approach to specifying transformations is shown in Fig. 4. This method directly parallels the procedure for specifying histograms in Fig. 3. Both linear and nonlinear transformations may be generated by varying each of the four parameters shown in Fig. 4 and, since monotonicity is not required, this approach can often be used to highlight objects in an image by the process of isolating and spreading different bands in the intensity scale.

Direct specification of $T(x_k)$ does not necessarily require operator interaction. Two transformations which do not involve such interaction are depicted in Fig. 5. Fig. 5(a) results in the negative of an image while Fig. 5(b) distributes the grey levels of an image into four evenly spaced levels. Transformations similar to the one shown in Fig. 5(b) are useful in controlling the number of gray levels used in displaying an image.

## III. Real-Time Computation of Transformation Functions

The enhancement methods considered in the previous sections represent an important class of spatial-domain techniques commonly referred to as position-invariant transformations. Such transformations operate directly on each pixel of an image to be enhanced without regard to its spatial position. In this section, we develop a method for implementing these transformations without temporary image storage and examine the conditions under which they can be implemented in real time.

Suppose that a time-varying video signal is digitized to form a sequence of $N \times M$ images $x(i, j, t_0)$, $x(i, j, t_0 + \Delta t)$, $\cdots$ where $i$ and $j$ denote spatial coordinates and $\Delta t$ represents the standard television refresh period of 33.3 ms. Let $x(i, j, m)$ repre-

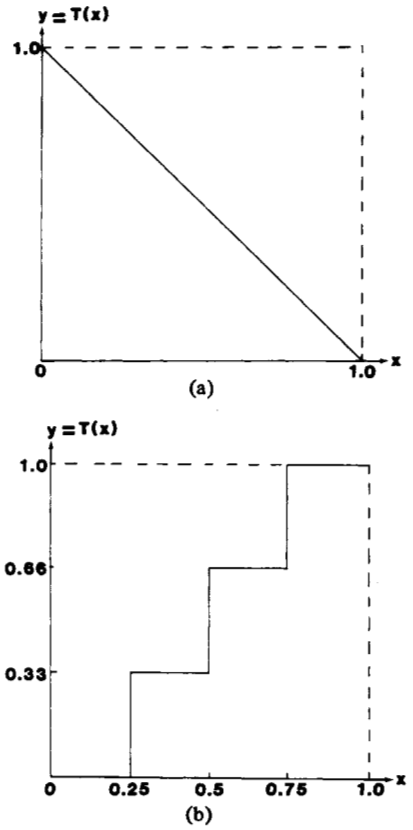


Fig. 5. Transformations to (a) create the negative of an image, and (b) reduce the number of gray levels.

sent an arbitrary member of this sequence, where $i$ and $j$ are allowed to range through the values

$$1 \leqslant i \leqslant N$$
$$1 \leqslant j \leqslant M \tag{14}$$

and $m$(an integer) identifies the position of an image in the sequence; that is,

$$t = t_0 + m\ \Delta t. \tag{15}$$

Then, for a particular $m$, (12) can be written as

$$y_k(m) = T[x_k(m)] \tag{16}$$

where $x_k(m)$ is any pixel in the $m$th image and $y_k(m)$ is its corresponding transformed value. As was indicated in Section II, we will restrict our attention to transformations $T$ that can be expressed as a function of the gray-level distribution of $x_k(m)$, denoted by $p_x[x_k(m)]$ and/or four interactively specified parameters (ISP's); that is,

$$T = f\{x_k(m), \text{ ISP's}\}. \tag{17}$$

Equations (16) and (17) suggest a convenient decomposition of our basic enhancement procedure: 1) computation of the mapping function $T$, and 2) transformation of the pixels in $x(i, j, m)$. Any digital implementation of these operations requires a finite expenditure of time. We will denote the time required to generate $T$ by $D_E$ and to transform a single pixel of $x(i, j, m)$ by $D_T$. Since $D_T$ is independent of the method used to generate $T$, $D_T$ is the same for all enhancement procedures discussed above. The time required to generate $T$, however, depends on the particular enhancement procedure being used.

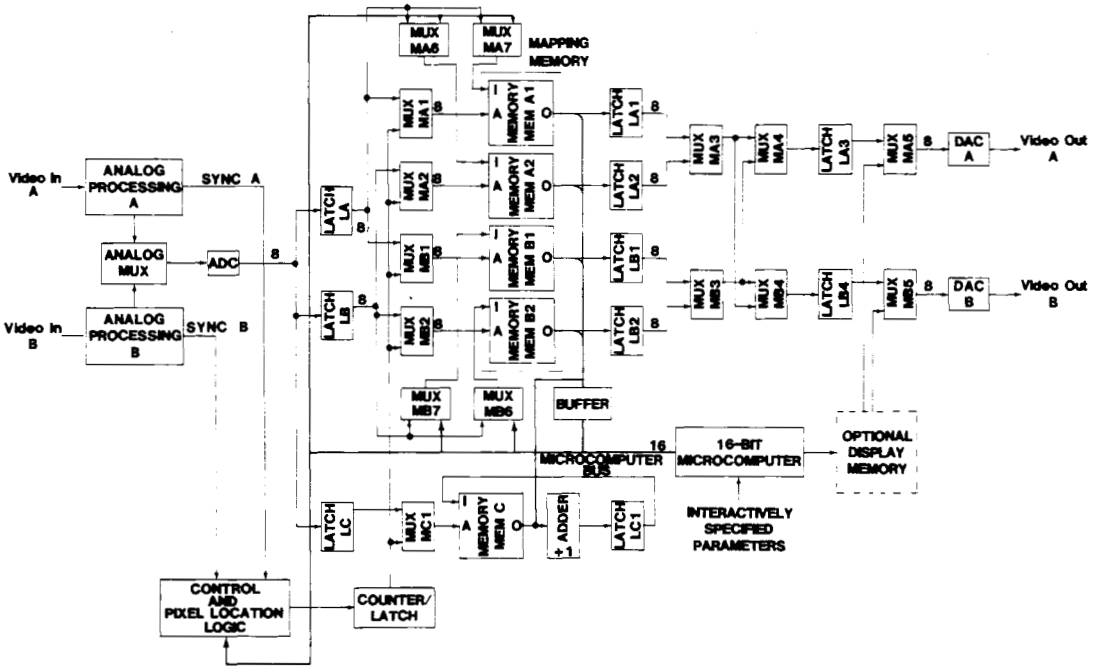


Fig. 6. Basic system architecture.

For histogram equalization, the minimum obtainable value of $D_E$ is $\Delta t$; that is, $T$ cannot possibly be determined before $x(i, j, m)$ is examined and $p_x[x_k(m)]$ is computed. When $D_E \leqslant \Delta t$, the entire image being enhanced must be temporarily stored.

To circumvent the need for temporary image storage, we use an approximation of (17) given by

$$T' = f\{x_k(m - \tau), \text{ISP's}\} \tag{18}$$

where $\tau$ is a nonnegative integer. In other words, we let the transformation used to map $x_k(m)$ depend on pixel values from a previous image $x_k(m - \tau)$. The success of this enhancement approach depends on the degree of interframe correlation in the video signal being processed. More specifically, its validity is related to the degree of similarity between the gray-level distributions of successive frames. For a sequence of images that represent an object in motion, one would expect a greater amount of correlation between gray-level distributions than between images themselves. As $\tau$ approaches zero, this correlation should increase so that in the limiting case where $\tau = 0$, (18) is equivalent to (17).

The use of (18) also facilitates a real-time implementation of the overall enhancement procedure. This is because $T'$, being a function of $x_k(m - \tau)$ rather than $x_k(m)$, can be computed in advance of the image $x(i, j, m)$ to be transformed. Thus (16) guarantees a mapping of $x_k(m)$ to $y_k(m)$ in time $D_T$. Since $D_T$ can be made small (e.g., on the order of the time between consecutive image pixels), the entire enhancement procedure may be considered real time.

When applying the above concepts to stereo imagery, two approaches may be followed. One obvious approach is to process the left and right views independently. In terms of histogram equalization, two gray-level distributions are computed and two independent transformations are generated. Such a technique can compensate for mismatches between the camera associated with the left and right views (e.g., cameras with different aperture settings). Another approach is to create a single $T'$ for mapping both views. For histogram equalization, $T'$ can be a function of the right and/or the left distributions. This approach, which is based on the (reasonable) assumption that the right and left views are highly correlated, has the advantage of computational simplicity.

## IV. System Design and Implementation

In this section we introduce the basic organization of an image enhancement system based on the concepts of Section II and explain the internal processes by which enhancement algorithms are executed. A functional block diagram of the enhancement system is used to illustrate basic operational concepts. Although the actual system has a more complicated logical structure than that indicated by the functional block diagram considered here, the chosen structure is sufficiently representative to demonstrate the basic organizational properties of the system.

### A. Architecture

A block diagram of the system we have implemented is shown in Fig. 6. This system consists of analog-to-digital (A/D) and digital-to-analog (D/A) converters, gray-level mapping logic, a 16-bit microcomputer, and a gray-level histogram computation stage. The operation of the system is controlled by a sequence of instructions residing in the program memory of the 16-bit microcomputer. Under the direction of this program, the microcomputer generates transformation functions (as described in Section II) which can be based on operator specified parameters and/or gray-level distributions provided by the histogram computation stage. These transformations are loaded into memories that are responsible for the actual gray-level mapping. As the appropriate video signals are periodically sampled by the A/D converter, their discretized

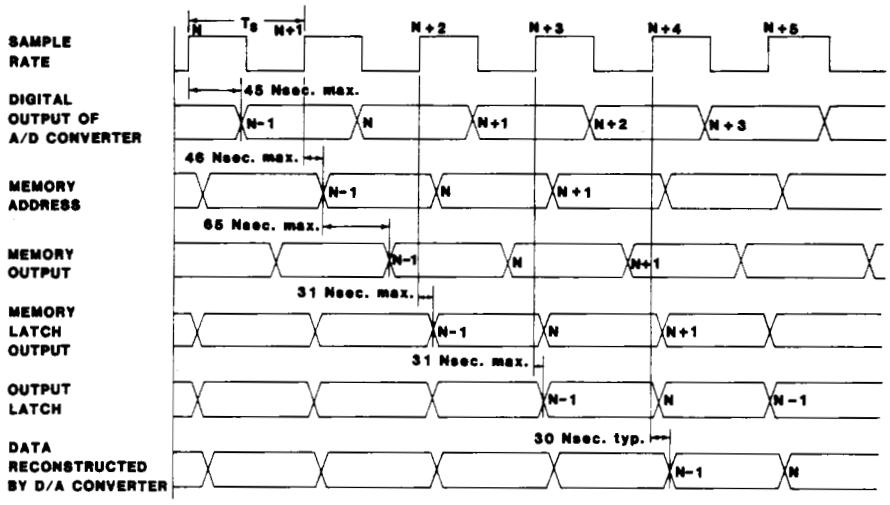


Fig. 7. Examples of system output image quality (right column) as compared to the precision analog input test patterns shown on the left.

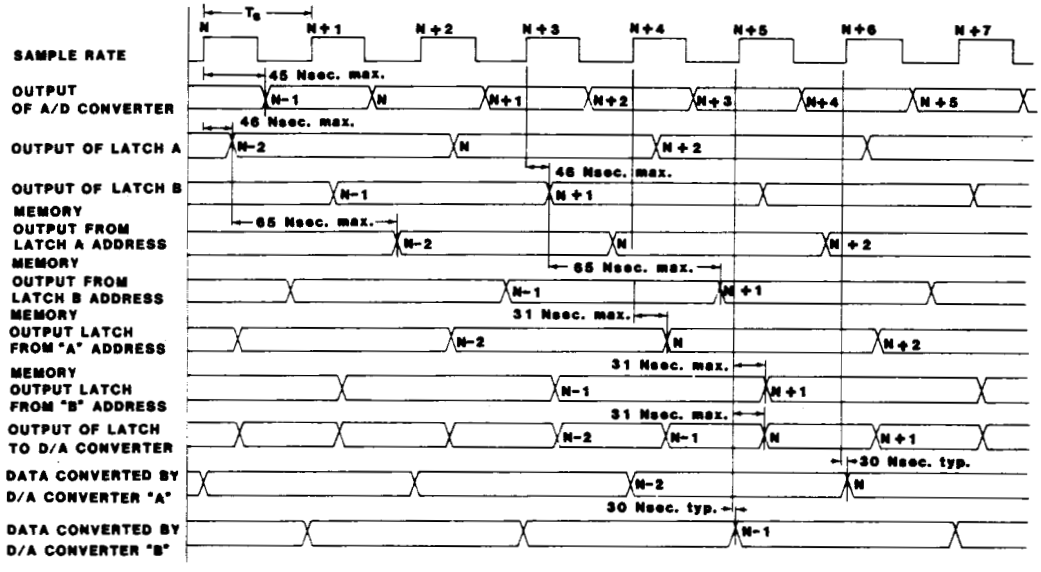


Fig. 8. Example of an image digitized at 1024, 512, 256, 128, and 64 points/line.

values are used as addresses of the memories containing the desired transformations. The addressed contents of these memories become the transformed gray levels which are converted to analog form by D/A converters.

*1) Analog-to-Digital and Digital-to-Analog Conversion:* The A/D and D/A converters used in the system of Fig. 6 determine both its gray-level and spatial resolution. The gray-level resolution is a function of the number of bits utilized by the converters. Since 8-bit converters are employed in the system, each pixel of a sampled video waveform is represented as one of 256 possible discrete intensity levels. According to previous studies, this intensity resolution is well beyond the 64 gray levels required over the span from black to white to minimize false contouring, a condition caused by insufficient intensity levels in a picture [5].

The system's spatial resolution is determined by the sampling period (time between samples) $T_s$ used for A/D and D/A conversion. Although a 4-MHz video signal requires a maximum Nyquist sampling period of 125 ns, the choice of $T_s$ (in practice) should be well below this figure to minimize the effects caused by spatially limiting the data. The enhancement system contains a 19.2 MHz master clock within the control logic portion of the system to provide a 52 ns minimum sampling period $T_s$. This corresponds to a spatial resolution of 1024 points per horizontal line for one video source of 512 points per horizontal line for each or two sources. Other spatial resolutions are obtained as multiples of this minimum $T_s$.

The video quality obtainable with the system is compared against precision test patterns in Fig. 7. Figs. 7(a)–(d) show the original analog patterns, while Figs. 7(e)–(h) illustrate the corresponding output images after digitization at 1024 points per scan line, mapping with a one-to-one transformation, and reconstruction. All images are displayed on the same television monitor with identical control settings. It is noted that there is no discernable difference between the two sets of images. Fig. 8 shows a natural scene displayed at several spatial resolutions, including 1024, 512, 256, 128, and 64 samples per scan line. As is usually the case, there is no noticeable difference between the 1024 and 512 resolution images. The image sampled at 256 points per line shows a slight degrada-

tion around the cup handle. The other two pictures illustrate the familiar effects of low-resolution sampling.

*2) Gray-level Mapping:* The gray-level mapping logic of Fig. 6 is a collection of four 256 × 8 bit mapping memories, together with the necessary latches and multiplexers needed to transfer information to and from these memories. The various clocking and steering signals necessary for the operation of these components are self-explanatory and have been omitted from Fig. 6. As can be seen in this figure, the system's four gray-level mapping memories are partitioned into two functionally interchangeable groups of two memories. These two groups of memories and their associated logic form two distinct digital processing paths from the output of the A/D converter to the two D/A converters. Each of these processing paths, which are designated by the symbols A and B, is capable of transforming one or two video sources at any available sampling rate, but only one may be used for gray-level transformation at any given time. For the purposes of the following discussion, we will represent the path that has been selected for gray-level mapping by the symbol $X$; the other path will be denoted by $\overline{X}$.

Transfer of data along path $X$ is a function of sampling rate and the number of video signals to be manipulated. The sequence of components involved in this pipelined transfer, however, is a function of sampling rate alone. (The timing of data transfers among the various components depends on the number of signals being transformed.) Each time video source $A$ is sampled at any rate other than 1024 points per horizontal line, the output of the A/D converter is stored in latch LA. Multiplexer MX1 selects the contents of this latch as the address of memory MEM X1, whose output (the transformed value of the grey level forming the address) is loaded into latch LX1. Multiplexers MX3 and MA4 are then used to route the transformed value to latch LA3, where it is selected for D/A conversion by MA5 and DAC A. Using the notation "$\alpha \rightarrow \beta$" to denote "the output of component $\alpha$ is transferred to component $\beta$," the above transformation process can be compactly represented as INPUT A → ADC → LA → MX1 → MEM X1 → LX1 → MX3 → MA4 → LA3 → MA5 → DAC A → OUTPUT A. A similar expression can be obtained for video source B at sampling rates other than 1024 points per scan line and is given by INPUT B → ADC → LB → MX2 → MEM X2 → LX2 → MX3 → MB4 → LB3 → MB5 → DAC B → OUTPUT B.

When processing one video source ($A$ or $B$) at 1024 points per horizontal line, the transformation process described above must be altered to account for the fact that the access time of a single mapping memory in our system is not short enough (i.e., less than 52 ns) to transform successive samples of the source. Two mapping memories must be loaded with the same transformation function and alternately used to transform consecutive samples. In other words, the $N, N+2, N+4, \cdots$ samples are transformed by one mapping memory while samples $N+1, N+3, N+5, \cdots$ are transformed by another memory. If we let $Y$ denote the video source to be sampled at the 1024 rate, the component sequence utilized to transform the even samples of $Y$ can be expressed as INPUT $Y$ → ADC → LA → MX1 → MEM X1 → LX1 → MX3 → MY4 → LY3 → MY5 → DAC $Y$ → OUTPUT $Y$; the odd samples are altered by the sequence INPUT $Y$ → ADC → LB → MX2 → LX2 → MX3 → MY4 → LY3 → MY5 → DAC $Y$ → OUTPUT $Y$.

While processing path $X$ is being used for gray-level transformations, the mapping memories of path $\overline{X}$ are under direct microcomputer control. As long as these memories are under the control of the microcomputer, they may be employed as a buffer for digitizing and storing a scan line of either video source or loaded with transformation functions and selected for video mapping. (In the latter case, the memories become part of processing path $X$ and are no longer under direct access of the microcomputer.) When the memories are utilized to digitize a line of either source $A$ or $B$, the microcomputer must supply the control and pixel location logic with 1) the number of pixels to be stored and 2) the coordinates of the initial pixel for storage. Upon receiving this information, the control and pixel location logic initializes the counter/latch, which is selected as the address source of MEM $\overline{X}1$ by M$\overline{X}1$, to zero. When the first pixel to be stored is sampled and loaded into latch L$\overline{X}$, the contents of L$\overline{X}$ are selected for input to MEM $\overline{X}1$ by M$\overline{X}7$—thus storing the first pixel in MEM $\overline{X}1$. When the next pixel is digitized and clocked into L$\overline{X}$, it is selected as the input of MEM $\overline{X}2$ by M$\overline{X}6$. After this pixel is stored in MEM $\overline{X}2$, the counter/latch that is being used as the address source of both memories is incremented. As sampling continues, image pixels are alternately stored in consecutive locations of MEM $\overline{X}1$ and MEM $\overline{X}2$ until the requested number of pixels have been placed in the memories.

To read a digitized line from MEM $\overline{X}1$ and MEM $\overline{X}2$ or load a transformation function into them, the microcomputer must provide the control and pixel location logic with the address for the desired read or write operation. This address is loaded into the counter/latch and routed to the address inputs of the appropriate memories. When a memory write operation is performed, data from the microcomputer bus is selected as the memories' inputs by multiplexers M$\overline{X}6$ and M$\overline{X}7$; during a memory read, the outputs of the appropriate memories are placed on the microcomputer bus. Both 8- and 16-bit read/write operations can be performed.

*3) Histogram Computation:* Although the microcomputer can obtain a histogram of either video source by repeated application of the line digitization procedure described above, a common circuit for making amplitude histogram computations is included in the system to relieve the microcomputer of this time-consuming task. This circuit, which includes MEM C and its associated logic, can compute a 256-level amplitude histogram in one video frame. When forming a histogram, periodic samples of the desired video signal are loaded into latch LC. The output of this latch is selected as the address of MEM C by multiplexer MC1. Each time a memory location is addressed, its contents are incremented by one. Since every memory location corresponds to a given conversion interval, a histogram will grow in the memory as sampling goes on. Of course, all locations of the histogram memory must be initialized to zero before the histogram is formed.

### *B. Processing Rate*

The enhancement system of Fig. 6 possesses an inherent throughput delay resulting from the process of transforming the input video signal(s). This processing time is, in fact, the time required to transform a single pixel ($D_T$) noted in Section III. In the system of Fig. 6, $D_T$ is a function of sampling rate and the number of video signals to be digitized, transformed, and reconstructed. Two basic throughput delay modes exist. Figs. 9 and 10 illustrate the system timing associated with each of these operational modes. As will be seen below, all system throughput delays are negligible with respect to the standard video update rate of 30 frames/s.

An overall timing diagram for the system of Fig. 6 when one

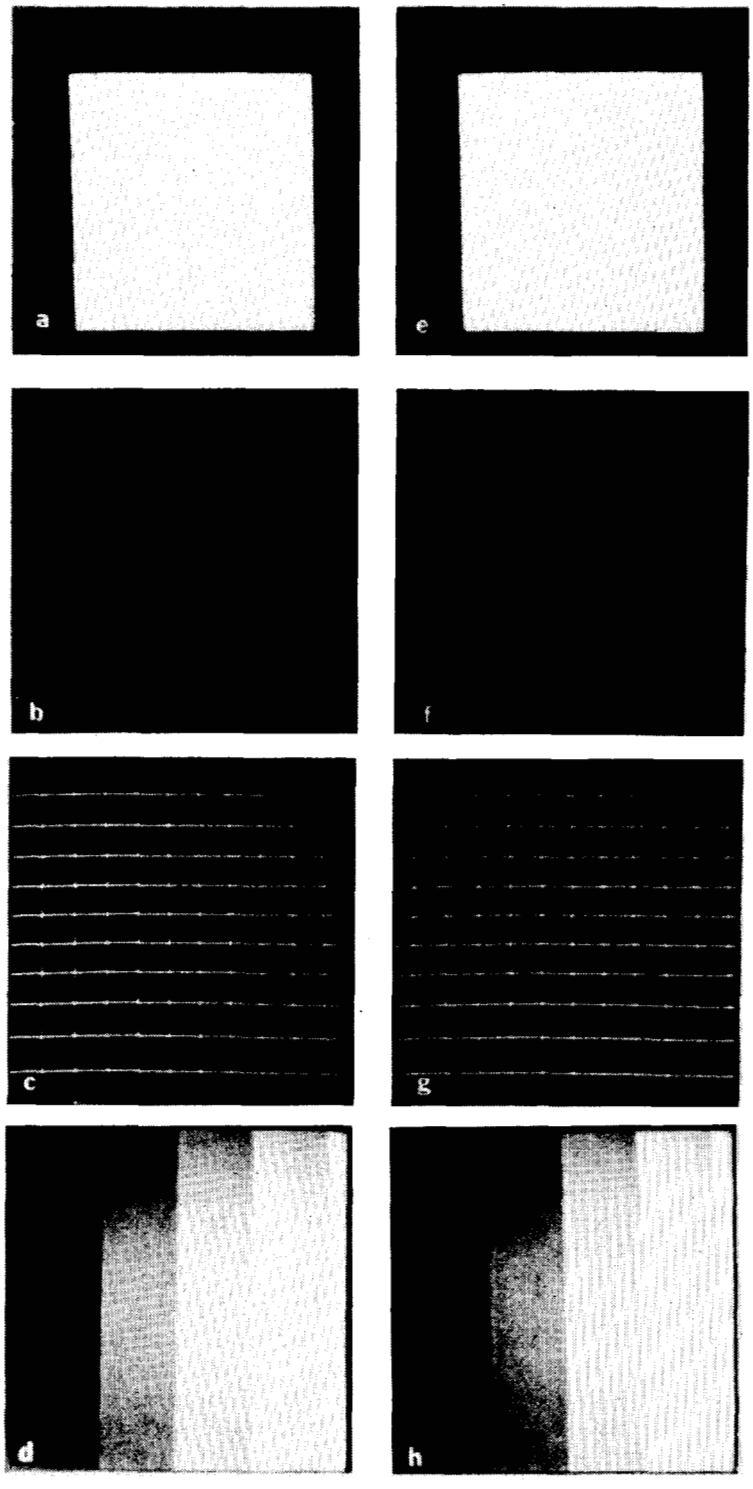


Fig. 9. *Timing diagram for one video source sampled at rates less than 1024 points/line.*

video source is sampled at 512 points or less per horizontal line is shown in Fig. 9. This diagram traces a sample point $N$ along path $X$ from the output of the A/D converter to its reconstruction by the D/A converter. Delay times shown in the figure correspond to propagation delays associated with components used in the system. These values represent the delay from the time a data transfer is initiated to the time it is completed. For example, 46 ns are needed from the edge of the pulse initiating the transfer of data from the output of the A/D converter to insure a valid memory address. As can be seen by tracing a sampled point $N$ through Fig. 9, five sampling periods are required to transfer a single pixel in this mode of operation. The A/D converter uses the equivalent of two sampling periods since its method of conversion produces the digital representation of the previously sampled point when a new sample pulse occurs, i.e., when the pulse to sample point $N$ is given, the digital representation of point $N-1$ is available at the output of the A/D converter. The $N+2$ sample pulse

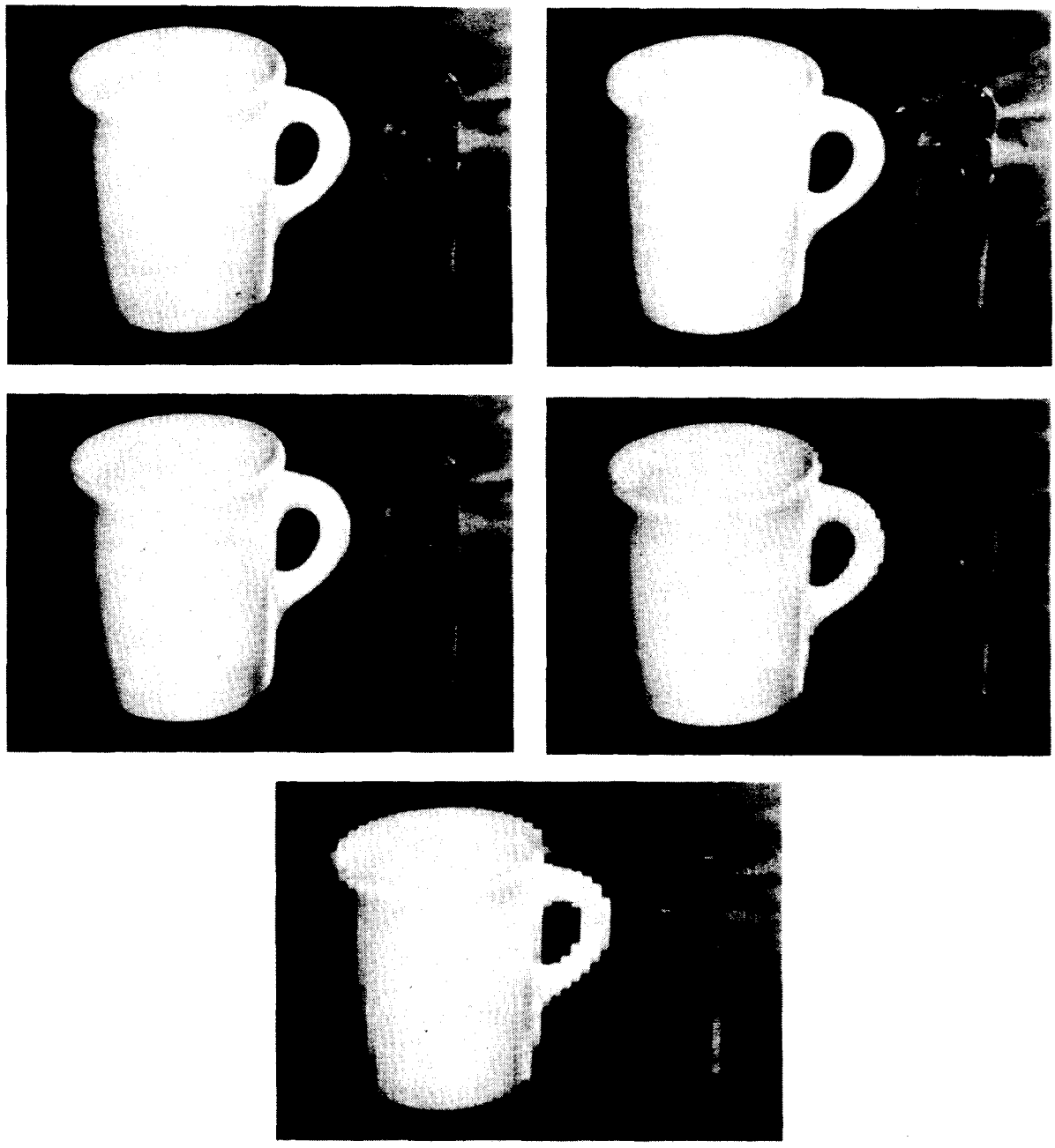

Fig. 10. Timing diagram for one video source sampled at 1024 points/line or two video sources sampled at a maximum of 512 points/line.

clocks the output of the A/D converter into the latch (LA or LB) used to hold the memory address. Next, the $N+3$ sample pulse latches the output of the memory location addressed, which is the transformed data, into a memory output buffer (LX1 or LX2). Sample pulse $N+4$ clocks the output of this buffer into latch LX3, which is used by the D/A converter for reconstruction. Finally, the $N+5$ sample pulse strobes the D/A converter to reconstruct the analog signal value corresponding to the transformed sample point $N$.

Fig. 10 depicts the overall timing of the system when one video source is sampled at 1024 points per horizontal line or two sources are sampled at any available rate. Transfer of data along path $X$ occurs by the same basic procedure mentioned above, with the exception that the output of the A/D converter is alternately clocked into two latches (LA and LB) that are selected as the addresses of MEM X1 and MEM X2, respectively. This allows the system of Fig. 6 to run at a rate faster than the access time of the memory chips used for mapping would allow, since each of the two memories is only addressed by every other sampled point. Thus the $N$, $N+2$, $N+4$, $\cdots$ sampled points are clocked into latch LA while the $N-1$, $N+1$, $N+3$, $\cdots$ points are clocked into latch LB. The same procedure described in regard to Fig. 9 is used to transfer the transformed data to the latches (LX1 and LX2) at the memories' outputs. The contents of these latches are then alternately clocked into the appropriate latches for D/A conversion. Six sampling periods are required to transfer the data from the A/D converter's output to the outputs of the D/A converter(s) in this mode of operation. The transfer follows the same pattern as that of Fig. 9, except that the data requires

TABLE I
ENHANCEMENT SYSTEM DELAY TIME FOR ONE VIDEO SOURCE

| Sample Rate | Sampling Period ($T_s$) | Delay time ($D_T$) |
|---|---|---|
| 1024 | 52 ns | 350 ns |
| 512 | 104 ns | 570 ns |
| 256 | 208 ns | 1.14 $\mu$s |
| 128 | 416 ns | 2.28 $\mu$s |
| 64 | 832 ns | 4.56 $\mu$s |
| 32 | 1.664 $\mu$s | 9.12 $\mu$s |

TABLE II
ENHANCEMENT SYSTEM DELAY TIME FOR TWO VIDEO SOURCES

| Sample Rate[a] | Sampling Period ($T_s$)[a] | Delay Time ($D_T$) |
|---|---|---|
| 512 | 104 ns | 350 ns |
| 256 | 208 ns | 700 ns |
| 128 | 416 ns | 1.4 $\mu$s |
| 64 | 832 ns | 2.8 $\mu$s |
| 32 | 1.664 $\mu$s | 5.6 $\mu$s |
| 16 | 3.328 $\mu$s | 11.2 $\mu$s |

[a]Sample rate and sampling period per camera

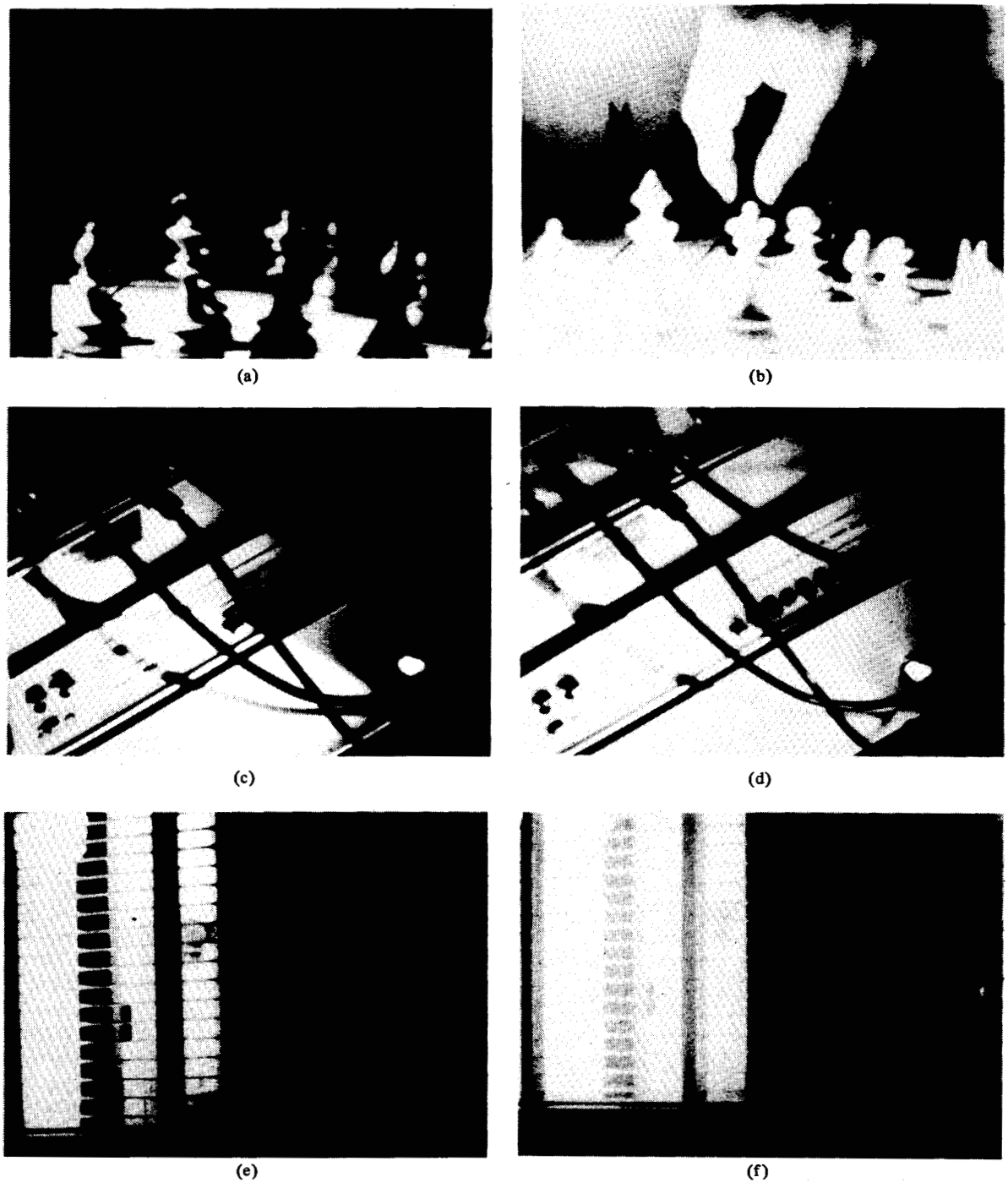


Fig. 11. Examples of images before (left) and after (right) histogram equalization.

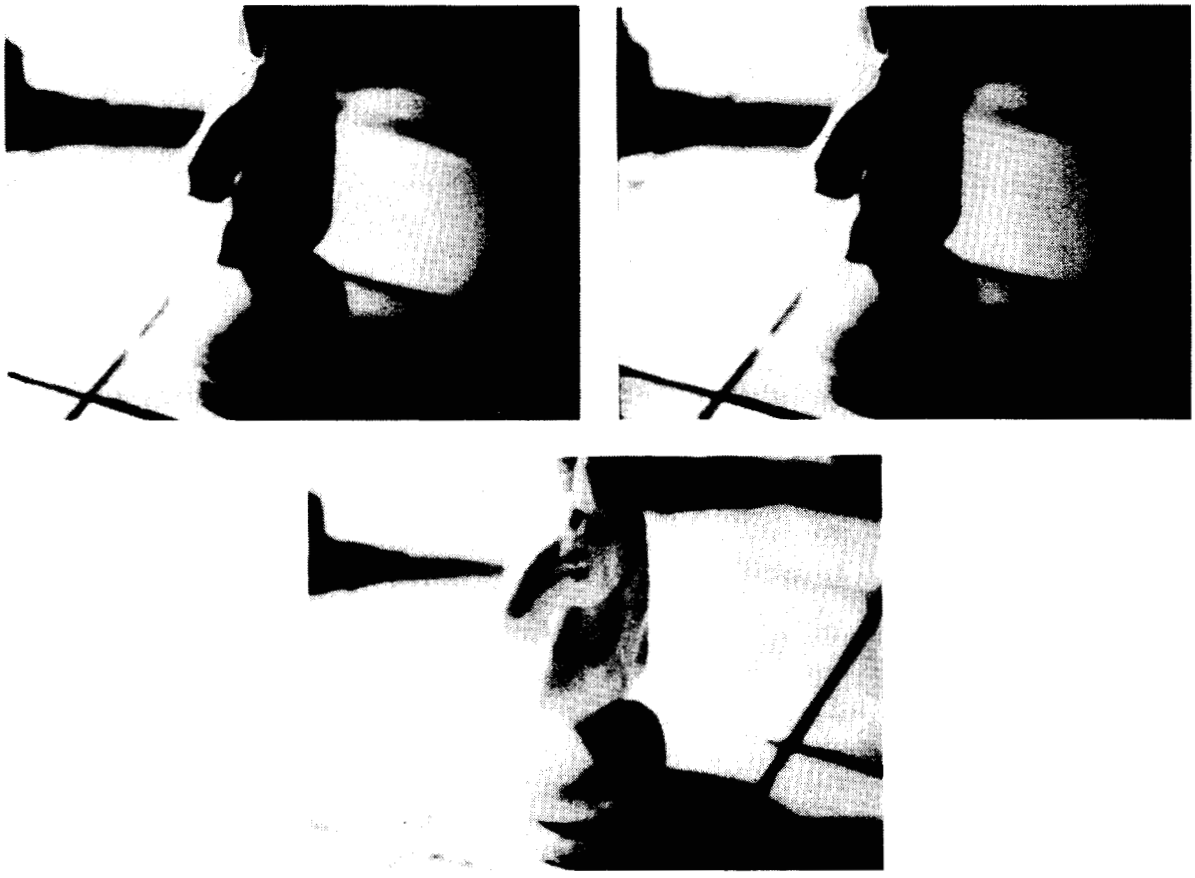

Fig. 12. Original image (top left) and results after histogram equalization and function processing.

two sampling periods to be transformed by the memory or memories (rather than one).

Tables I and II specify the delay times incurred in the two basic modes of operation for various sampling rates or spatial resolutions. As indicated in these tables, the system of Fig. 6 possesses a delay time $D_T$ of 350 ns at the fastest sampling rate and 11.2 $\mu$s at the slowest rate. These delays correspond to 0.05 and 17.6 percent of the time required to represent a horizontal line of video information in the analog signal, respectively.

## V. Results

In this section we present a number of results using the enhancement system under a variety of experimental conditions. Because of the difficulties in duplication and viewing stereographic images we have included only monocular enhancement results. In stereo applications, we have found the same degree of enhancement as that shown in the following results with the exception, of course, of the psychovisual phenomena associated with depth perception.

Fig. 11 illustrates the enhancement capabilities of the histogram equalization mode of operation, which is automatic since no operator interaction is required. The histograms of the images on the left side of Fig. 11 would be expected to span two relatively narrow gray-level ranges; the pixels of the corresponding enhanced images on the right side of Fig. 11 span a considerably larger spectrum of the gray-level scale. As is characteristic of this method, a simple spreading of the gray-level histogram can have a remarkable effect on the output image.

Although, as shown in Fig. 11, histogram equalization can yield excellent enhancement results, this approach often fails to bring out high contrast details in an image. An example of this is given in Fig. 12. The image on the top left is the original and the image on the top right is the result after histogram equalization. For all practical purposes, this result is of little use because it failed to bring out the obscure side of the knight. The image displayed at the bottom of Fig. 12 was obtained interactively using function processing. It is noted that the detail on the obscure side of the knight was vividly brought out by this approach. The transformation functions used in this approach are generated by the system in less than one video frame.

As a final illustration, consider Fig. 13 which shows, on the top row, the original image and the histogram equalized result. Although the latter image is an improvement over the original, it still has several areas that are barely discernable. For example, most of the telephone cord is obscured. The result shown in the bottom left picture was obtained by interactive function processing. It is noted that the cord, as well as several other objects that were not readily visible in the histogram equalized image, were brought out by the function processing approach. Finally, the image on the bottom right was obtained using histogram specification. This image presents a more balanced appearance than any of the other results. For example, the grooves in the wall were clearly brought out using

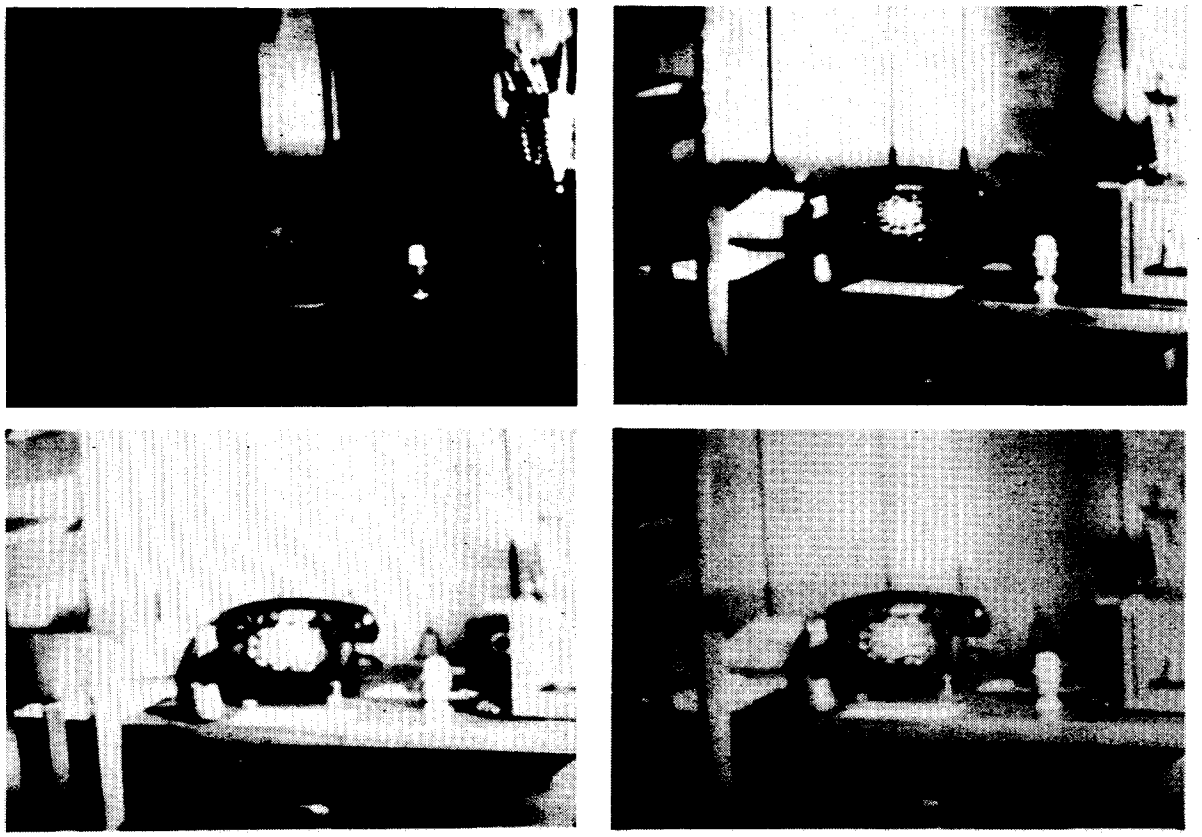

Fig. 13. Original image (top left) and results (clockwise) after histogram equalization, histogram specification, and function processing.

this method. Transformation functions for histogram specification require about 30 video frames or 1 s to be generated. The interactive enhancement process is therefore slower than that required by function processing. In general, we have found the function processing approach to be quite acceptable for most interactive experiments. It is often superior to histogram equalization and, unlike histogram specification, has essentially no delay before the enhanced result is available on the television screen following a change in parameters.

## VI. Summary

This paper has presented the design and implementation of a digital image enhancement system based on histogram modification techniques. The system is capable of enhancing an image by one of three basic methods: histogram equalization, function processing, and histogram specification. The first two approaches were implemented in real time in the sense that sampling and enhancement delays are well below the standard television refresh rate of 30 frames/s. Histogram specification, while more powerful, requires computations that introduce a delay of approximately 1 s. Although this delay is significant with respect to the refresh rate, it is still well below human response time for many interactive applications.

The experimental results presented in the previous section illustrate the enhancement capabilities of the system in the three enhancement modes. Histogram equalization is an automatic mode in the sense that it does not require operator assistance. The other two modes are automatic when *a priori* information is used to compute the transformation functions and interactive when they must be specified by an operator. All three methods were shown to be capable of producing significant enhancement results. The interactive modes proved in several cases to be quite useful in highlighting details that were beyond the enhancement capabilities of the histogram equalization technique.

## References


[1] H. K. Huang, "Fundamentals of biomedical image processing," in *Proc. 2nd Annu. Symp. Computer Applications Medical Care* (Washington, DC), pp. 8–14, Nov. 1978.

[2] B. G. Batchelor, "Interactive image analysis as a prototyping tool for industrial inspection," *IEEE J. Comput. Digital Technol.*, vol. 2, pp. 61–70, 1979.

[3] "Shuttle free-flying teleoperator system experiment definition," Bell Aerospace Company, Final Rep. D7425-953004, vol. 2, June 1972 (under NASA contract NAS8-27895).

[4] J. R. Tewell, *et al.*, "Teleoperator visual system simulations," *J. Spacecraft*, vol. 11, no. 6, pp. 418–423, 1974.

[5] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1977.

[6] T. G. Stockham, Jr., "Image processing in the context of a visual model," *Proc. IEEE*, vol. 60, pp. 828–842, 1972.

[7] R. A. Hummel, "Image enhancement by histogram transformation," *Comput. Graph. Image Proc.*, vol. 6, pp. 184–195, 1977.

[8] W. Frei, "Image enhancement by histogram hyperbolization," *Comput. Graph. Image Proc.*, vol. 6, pp. 286–294, 1977.

[9] R. C. Gonzalez and B. A. Fittes, "Grey-level transformations for interactive image enhancement," *Mechanism Mach. Theory*, vol. 12, pp. 111–122, 1977.

[10] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1965.

[11] E. L. Hall, "Almost uniform distributions for computer image enhancement," *IEEE Trans. Comput.*, vol. C-23, pp. 207–208, 1974.

[12] J. J. Hwang, W. J. Nalesnik, E. L. Hall, and C. R. Archer, "Computer-aided evaluation of radiographic film quality," in *Proc. Computer Aided Analysis of Radiological Images* (New Port Beach, CA), pp. 132–136, June 1979.

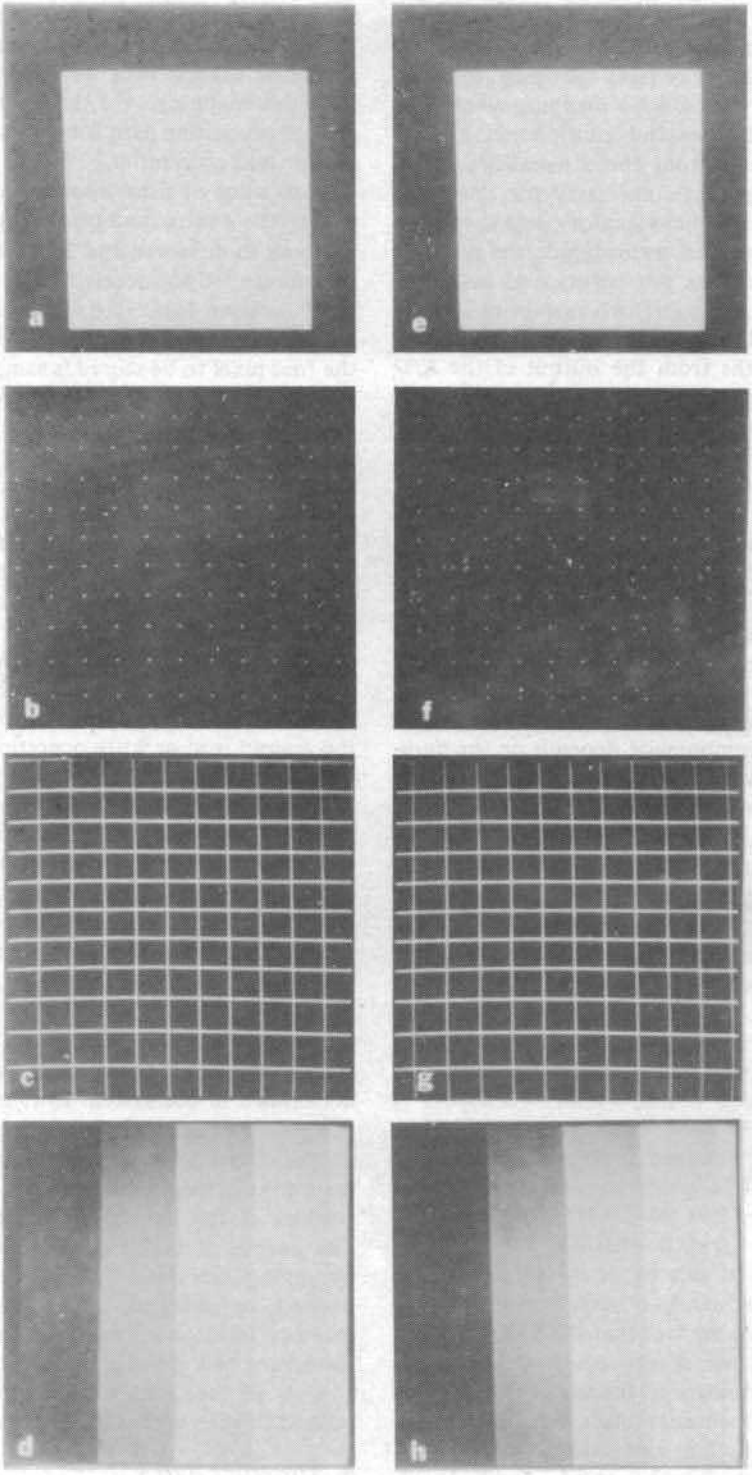


Fig. 9. Timing diagram for one video source sampled at rates less than 1024 points/line.

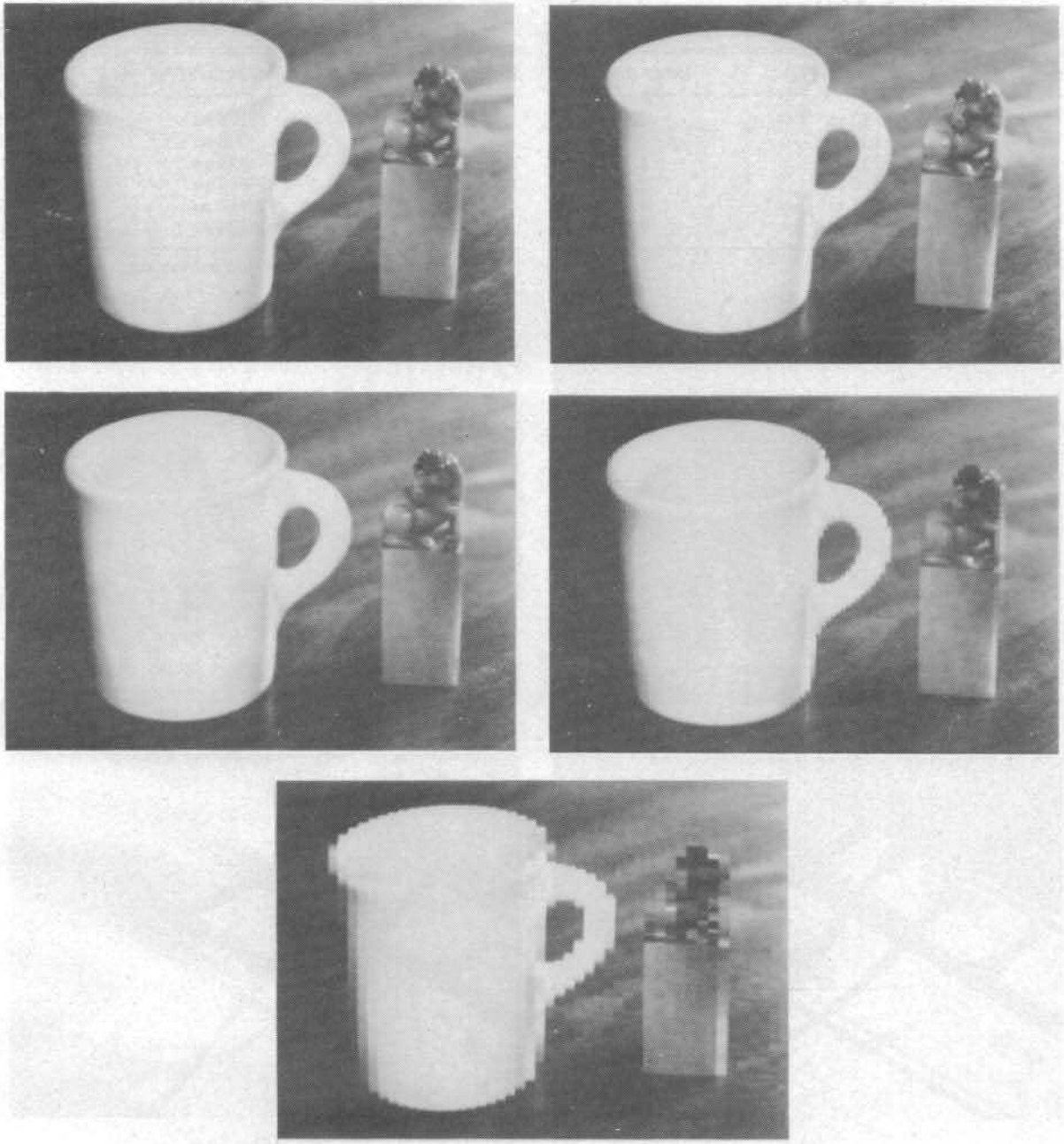

Fig. 10. Timing diagram for one video source sampled at 1024 points/line or two video sources sampled at a maximum of 512 points/line.

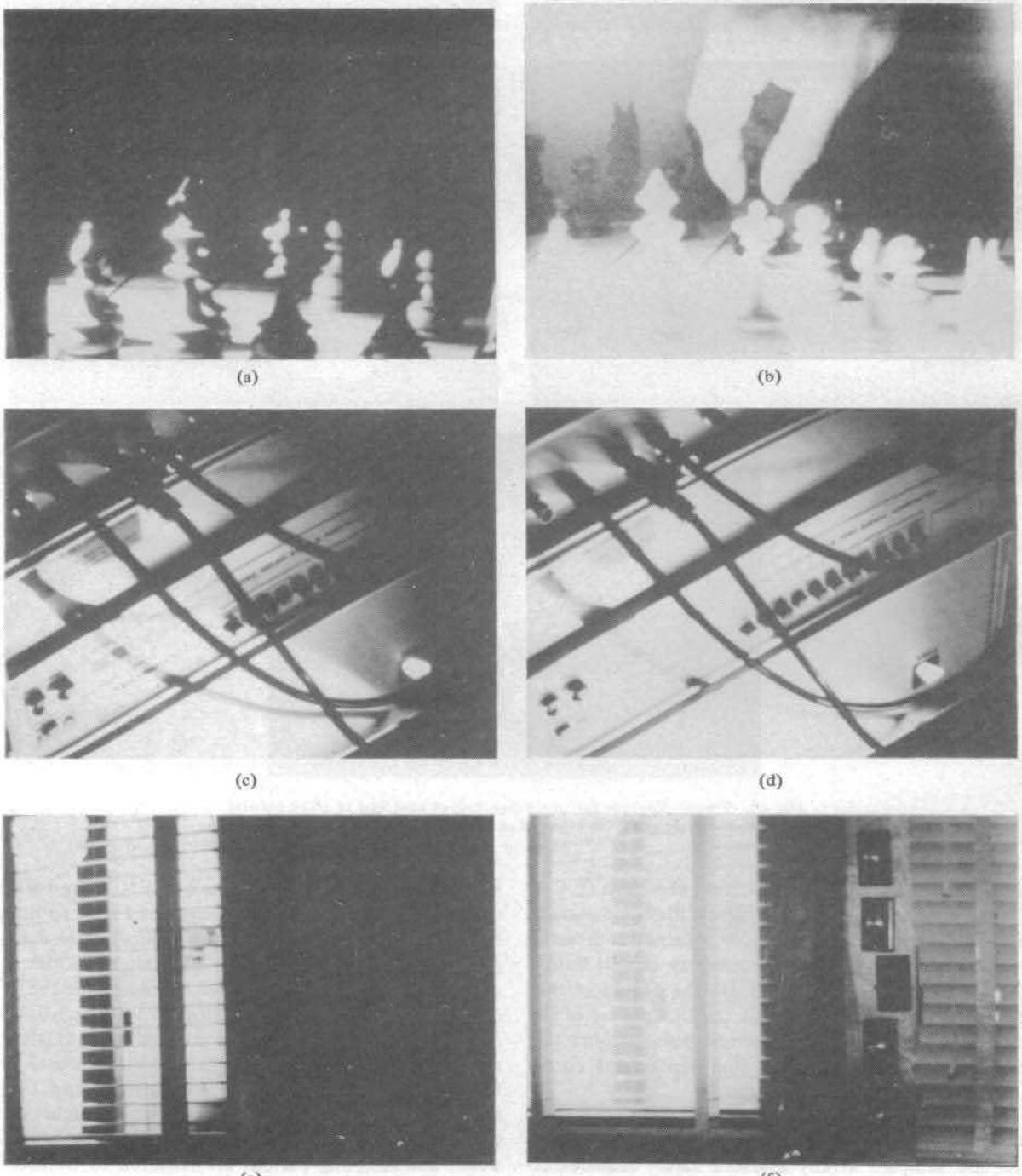

Fig. 11. Examples of images before (left) and after (right) histogram equalization.

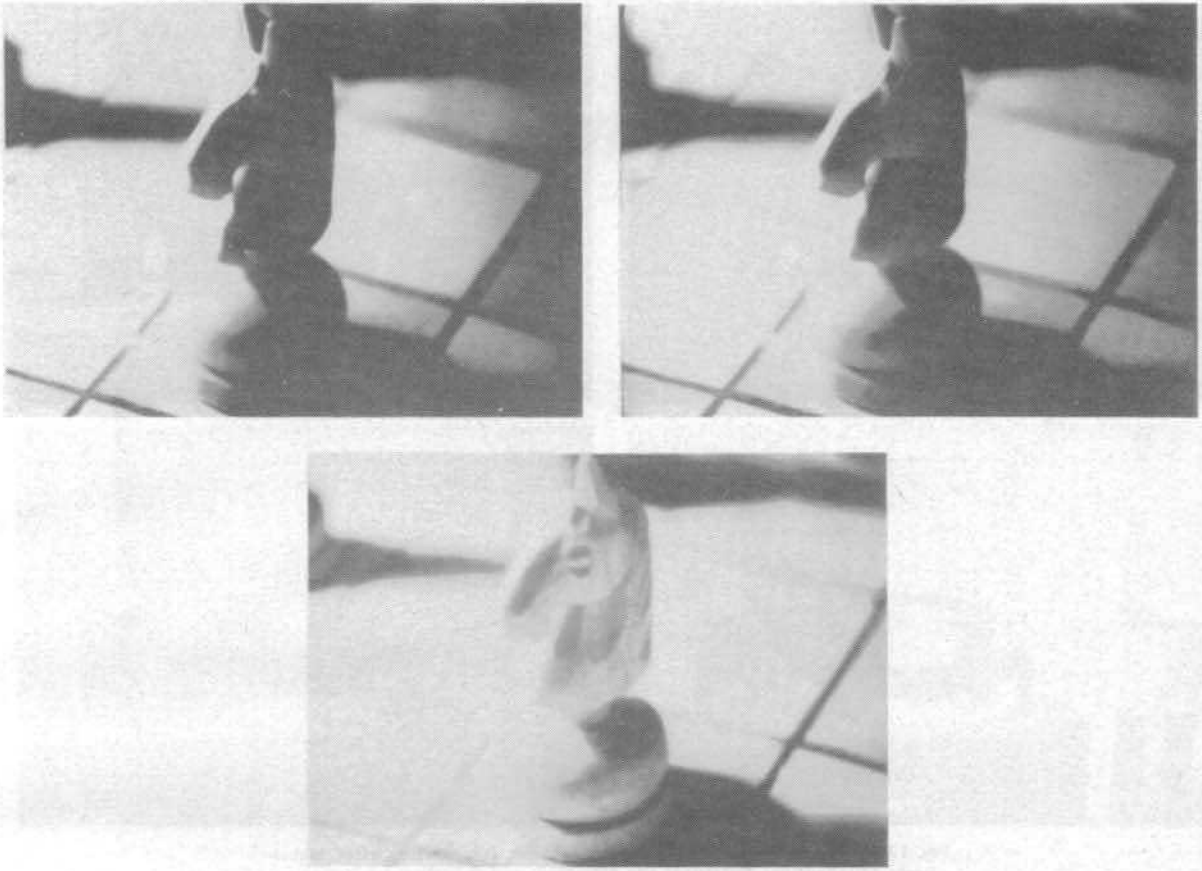

Fig. 12. Original image (top left) and results after histogram equalization and function processing.

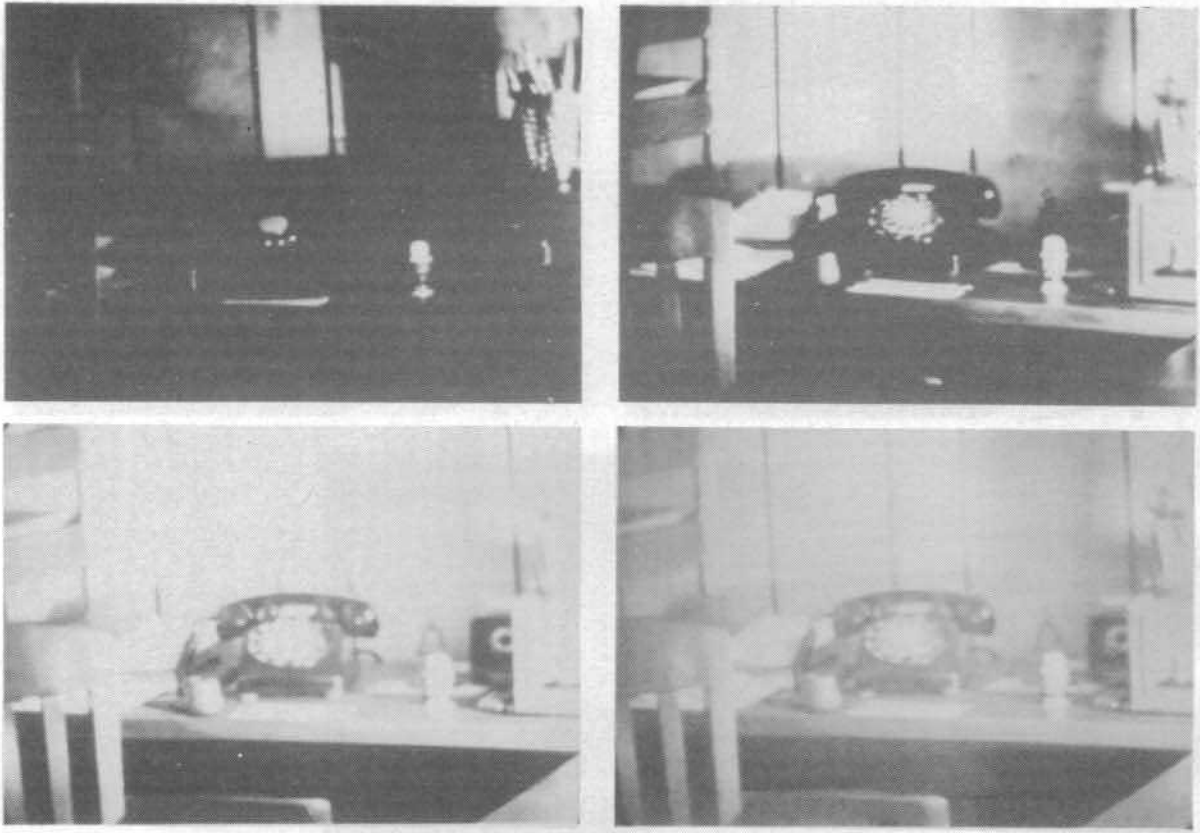

Fig. 13. Original image (top left) and results (clockwise) after histogram equalization, histogram specification, and function processing.