UCRL- 88952
PREPRINT
Conf-830334--1

FUNDAMENTAL CONCEPTS OF DIGITAL IMAGE PROCESSING

R. E. Twogood

Invited Paper for the
International Symposium and Course on
Electronic Imaging in Medicine

San Antonio, Texas

March 1983

Lawrence Livermore Laboratory

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

FUNDAMENTAL CONCEPTS OF DIGITAL-IMAGE PROCESSING

R. E. Twogood

Lawrence Livermore National Laboratory, Livermore, Ca. 94550

BACKGROUND

The field of a digital-image processing has experienced dramatic growth and increasingly widespread applicability in recent years. Fortunately, advances in computer technology have kept pace with the rapid growth in volume of image data in these and other applications. Digital-image processing has become economical in many fields of research and in industrial and military applications. While each application has requirements unique from the others, all are concerned with faster, cheaper, more accurate, and more extensive computation. The trend is toward real-time and interactive operations, where the user of the system obtains preliminary results within a short enough time that the next decision can be made by the human processor without loss of concentration on the task at hand. An example of this is the obtaining of two-dimensional (2-D) computer-aided tomography (CAT) images. A medical decision might be made while the patient is still under observation rather than days later.

The reader is referred to several excellent texts which have recently been published covering the field of digital image processing, including those by Castleman[1], Andrews and Hunt[2], Huang[3], Pratt[4], Gonzalez and Wintz[5], and Hall[6]. Several collections of papers have been edited into bound versions, e.g. Mitra and Ekstrom[7], Aggarwal[8], Bernstein[9], and Andrews[10]. In addition to the voluminous conference and journal literature, there are several pertinent special journal issues devoted to 2-D digital signal processing and/or image processing[11-16].

The processing of two-dimensional data, or images, using a digital computer or other special digital hardware typically involves several steps. First, the image to be processed must be put in a format appropriate for digital computing - this image acquisition step can be accomplished in a number of ways, depending on the application. Then the processing must be performed in order to extract the information of interest from the image(s). Finally, the imagery must be reformatted for human or machine viewing, storage, or hardcopy documentation.

A simplified block diagram of a system for performing these digital image processing tasks is shown in Figure 1. Various issues associated with the image acquisition (leftmost block in the figure) are discussed briefly in the next two sections. The top and bottom blocks in the figure are simply the standard sort of data/program storage and command/control equipment commonly employed in digital computers. The rightmost block is needed in order to display the resulting imagery, typically on a CRT device or directly onto hardcopy such as film.
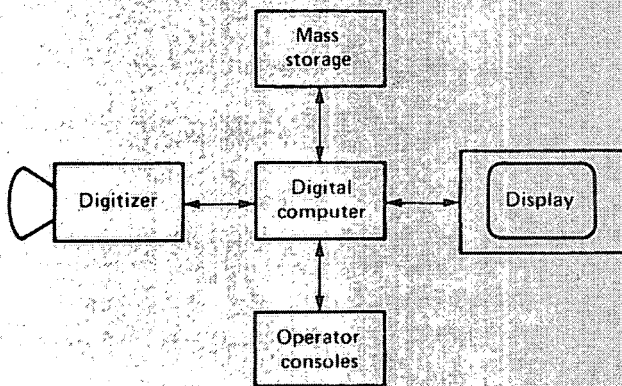
IMAGE DIGITIZATION

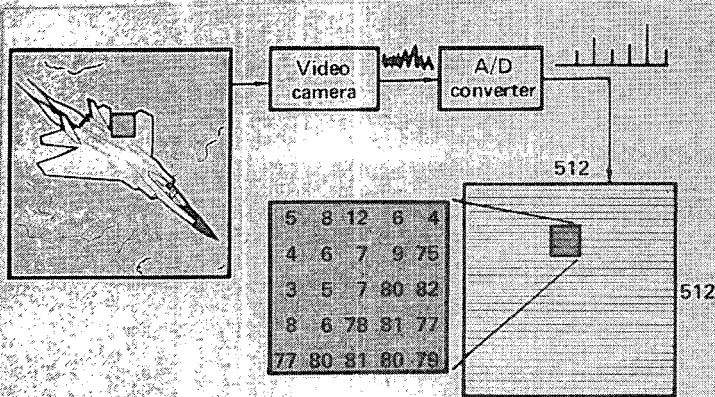Fig. 1. Typical components of a digital image processing system.

Fig. 2. Image acquisition and digital representation of image data.

Once one has determined that computer processing of an image is desired, the first requirement is, of course, to obtain a digital representation of the image or scene. This is accomplished in a number of ways, depending on the nature of the image to be processed. For instance, one example of using photoelectronic means to obtain a digital image is depicted in Figure 2. Here an image of an airplane, as viewed by a video camera, is digitized by simply running the video signal through an analog-to-digital (A/D) converter. Due to the nature of the video signal (analog signal in horizontal direction, but just a discrete set of signals "stacked" in the vertical direction), the digitizing step is a straightforward one.

One of the most common ways of obtaining a digital image from film or transparencies is through the use of a scanning microdensitometer. In such a device, a small beam of light is sent through the transparency or reflected off the picture. A photomultiplier (pm) tube gathers the light transmitted through the transparency or reflected from the film positive and computes a relative intensity value for that point. The stage of the microdensitometer is stepped incrementally through both x and y dimensions at specified step sizes, depending on the size of the picture being scanned and the desired sampling resolution. A device called a flying-spot scanner operates in much the same fashion as a microdensitometer, with the main difference being that the light spot itself is moved to sample the various picture elements in the image, as opposed to the fixed light spot and moving stage in the microdensitometer.

An important development impacting image acquisition schemes in recent years has been the implementation for semiconductor devices for light sensing. The most popular device of this kind is the charge-coupled device (CCD). There are two principle configurations

used in obtaining imagery using CCDs. In one case, an array of sensors (typically ranging from an array size of 32 x 32 to 512 x 512) is used, with the image illuminating the sensor array directly. The second configuration, more commonly used in off-line applications, consists of a single one-dimensional array of sensors (say 2048 x 1) that is stepped through the second dimension in order to obtain the two-dimensional digital image. The rapid improvements in this semiconductor imaging technology have led to increased applications usage of these devices.

The 512 x 512 array size shown in Figure 2 is a typical size for the dimensions of a digital image. The digital picture in that case consists of an array of over a quarter of a million (262,144) discrete data points, with each picture element, or "pixel", being an integer of some value. While 512 x 512 is probably the most common, the number of pixels used varies from application to application depending on the resolution required, data storage capabilities, processing throughput requirements, and the nature of the image itself. For computational and hardware reasons, it is often desirable to restrict the dimensions to be powers of 2. As a result, digital image dimensions typically vary from 64 x 64 up to 4096 x 4096 or even larger.

IMAGE QUANTIZATION

Once the sampling of an image as described above has been accomplished, one has a set of values sampled on a discrete two-dimensional grid. However, the individual pixel values themselves must also be discretized, preferably into as few bits per pixel as possible. This discretization of pixel values is referred to as image quantization. The quantization of each picture element may, in fact, be a natural step in the digitization scheme, e.g. in the analog-to-digital conversion of the video signal described in the above section the a/d converter has a specified number of bits per sample. The number of bits per sample needed is dependent on the nature of the image sensed, the intended processing steps to be performed via digital computer, the resolution of the output device to be viewed, the availability and cost of a/d conversion hardware, the data storage availability, and other factors. In some applications, a single bit per pixel may be sufficient, i.e., the individual pixel values are either 0 or 1 (black or white). In most cases, however, several more bits are needed, with 6 bits/pixel (= 64 discrete gray levels) to 12 bits/pixel (= 4096 discrete gray levels) being the range of quantization levels found in most applications.

IMAGE PROCESSING GOALS

Given that one has obtained digital images by some means such as those described above, the next problem is to perform the digital processing on the 2-D data. As mentioned earlier in this chapter, there are dozens of applications of digital image processing, including many military, medical, agricultural, and industrial

uses. The exact nature of the applications imagery will dictate the types of processing required, and there are literally hundreds of different algorithms and techniques used to perform such processing. In the next section, we discuss several of the more useful and general-purpose of these algorithms. In virtually all image processing applications, however, the goal is to extract information from the image data. Obtaining the information desired may require filtering, transforming, coloring, interactive analysis, or any number of other methods.

To be somewhat more specific, one can generalize most image processing tasks to be characterized by one of the following categories:

1) Image enhancement - This simply means improvement of the image being viewed to the (machine or human) interpreter's visual system. Image enhancement types of operations include contrast adjustment, noise suppression filtering, application of pseudocolor, edge enhancement, and many others.

2) Image restoration - Image restoration includes those techniques aimed at removal of image degradations caused by the image formation process. For instance, removal of image blur caused by camera motion, out-of-focus imaging, or atmospheric turbulence are all image restoration problems. Restoration is a difficult and ill-conditioned problem which has enjoyed only limited success in real-world applications. The algorithms employed are, in general, quite complex and tailored to the application at hand. For a thorough discussion of this subject, the reader should see the text by Andrews and Hunt[2].

3) Image coding - Image coding involves the compression of image data into an alternate (coded) form in order to reduce storage and/or data transmission requirements. Image coding will not be treated in depth in this paper, but the reader can consult the excellent tutorial review of this field by Jain[17].

4) Image analysis - Image analysis means different things to different people, and in some contexts, is used to denote virtually the entire field of image processing. In our use of the term, we assume the more restricted class of image processing techniques where measurements are made or parameters of interest are estimated from the image data. Examples of such image analysis applications include radiographic image mensuration for nondestructive evaluation purposes, fitting curves to image data, etc.

5) Feature extraction and recognition - In many image processing tasks, it is important to characterize objects in the scene or actually identify, typically via pattern recognition approaches, specific objects. Examples include finding any tanks in a battlefield scene, or identifying mutant cells in a biological cell sample. The algorithms for implementing such tasks are extremely varied, and range from simple edge detectors to matched filtering and sophisticated classification algorithms. Several comprehensive books have been published on pattern recognition and related topics[16,18,19], to which the reader is referred for further information.

6) Image understanding - The field of image understanding is a relatively young one in which a priori knowledge is used to try to help understand the relationships among objects in a picture. Motivated primarily by potential military applications, most of the research in this field has been funded through DARPA's Image Understanding program in the Department of Defense. The three major thrusts in this program have been a) smart sensors, b) iconics, or visual phonetics, which involves feature extraction and analysis of structural composition, and c) symbolic representation, or describing objects in a scene using mathematical descriptors, grammars, or strings of symbols[20]. The ultimate goal of this research is to maximize the degree of machine understanding of a scene, much as a human observer draws on his a priori understanding of the real world to interpret the scenes he views. As such, the field of image understanding incorporates aspects of many fields, including image processing, pattern recognition, artificial intelligence, neuropsychology, and computer science.

The above characterizations are, of course, generalizations of image processing tasks and goals. Rarely does a single application fall cleanly into one of these individual categories - but rather, there is a close synergism between several of these and indeed vast areas of overlap in some cases. In addition, many real-world applications involve more than one of the above classes of operations. For instance, a space exploration project might include remote image acquisition and coding, transmission of the coded data back to earth, image decoding, filtering and enhancement for noise suppression or feature enhancement, and then, interactive analysis by an image interpreter. Nonetheless, the descriptions above give a useful and broad overview.

DIGITAL IMAGE PROCESSING ALGORITHMS

This section presents a brief description of the types of algorithms commonly utilized for digital image processing. We will intentionally limit the discussion to the types of algorithms which are widely used in many different image processing applications. Relatively specialized topical areas in image processing, such as image coding, image restoration, tomography, unitary transforms, etc., will not be emphasized. Several good references contain detailed descriptions of these and other assorted image processing techniques, with the emphasis in many being based on the interests and experience of that particular author. For example, Pratt[4], Gonzalez and Wintz[5], and Castleman[1] have authored excellent texts covering, to various degrees, the general field of "Digital Image Processing," and each of those three books has that same title. The field of image restoration or deconvolution, which encompasses those techniques aimed at removing degradations of an image resulting from its being imaged through a system with a transfer function, has been thoroughly treated by Andrews and Hunt[2]. Transform techniques sometimes useful in image processing applications are investigated in another text by Andrews[21].

The techniques for image processing summarized in this section are categorized according to the nature of the region in the input image which determines the output pixel value. The first category is that of point operators, which map a single input pixel to the corresponding output pixel. While many of these operators are straightforward conceptually, several powerful mapping techniques exist.

The second category of image processing operators are those that compute an output pixel value as a function of a small neighborhood of pixels surrounding the corresponding pixel location in the input image. Several standard techniques which are essentially extensions from one-dimensional signal processing fall into this class of local operators, including linear filtering in two dimensions and spatial differentiation. However, there are a number of more or less ad hoc nonlinear and/or spatially variant local operators which turn out to be surprisingly powerful.

The final category discussed includes those operators that involve much, if not all, of the input image in the computation of the output pixel value. Primary examples of such "global" operators are the 2-D transform techniques.

Point Operators

One class of very simple, in general, yet very useful image processing operations is that of point operators. A point operator is one in which each output pixel is a function of only the single input pixel at the same (m,n) location in the input image. A generalization of this concept is the case when each output pixel is a function of the input pixels from the same (m,n) location in a set of input images.

In general, the implementation of point operators is efficient and straightforward. Indeed, many image processing systems available today realize a large number of point operators in hardware, allowing extremely fast throughput of results. Obvious examples of point operators are: addition, subtraction, multiplication, and division of multiple images or of single images with a constant; exponentiation or logarithm of an image; raising to a power, absolute value of an image, and many others.

In addition to the above-mentioned simple arithmetic type operators, there are a number of other point operations which can be of significant usefulness in image analysis and enhancement. Many techniques which perform operations based on the gray-level histogram of the input image have proven useful. The histogram is a function showing the number of occurrences of a particular gray-level value. The abscissa of the histogram is gray level value, and the ordinate corresponds to the number of pixels with that value. An example of a point operator that is based on histogram manipulations is that of clipping or thresholding, i.e.,

$$g(m,n) = \begin{cases} a & f(m,n) \le a \\ f(m,n) & a < f(m,n) < b \\ b & f(m,n) \ge b \end{cases}$$

where $f(m,n)$ is the input image and $g(m,n)$ is the output image.
Another useful technique based on the histogram is histogram equali-
zation, where gray-level values in the input image are mapped to
output values such that the resulting output image has an approxi-
mately uniform distribution, i.e., the gray levels are approximately
equally populated[22,23]. Variants of this concept have been
applied by Frei[24], including a histogram hyperbolization scheme
which incorporates a model of the response of the human eye's photo-
receptors.

Many of the hardware image processing systems available commer-
cially have histogram mapping hardware capabilities. As with the
histogram-based methods described above, these mappings are intended
to make more effective utilization of the display system's dynamic
range. In many cases, the built-in hardware capabilities allow
real-time human interaction to select, in some subjective sense, the
"best" mapping for the particular image being analyzed. One such
example of a nonlinear histogram mapping application for contrast
enhancement is shown in Figure 3. Figure 3a shows the original
image, which in this case, is a radiographic exposure of a material
through which the projectile from a rail gun has passed. Figure 3b
shows the nonlinear mapping used to map input pixel values to the
new output values — in this case, we are compressing the dynamic
range to emphasize gray-level differences over a very small range of
density values in the original image. The resultant contrast-en-
hanced output image is shown in Figure 3c.
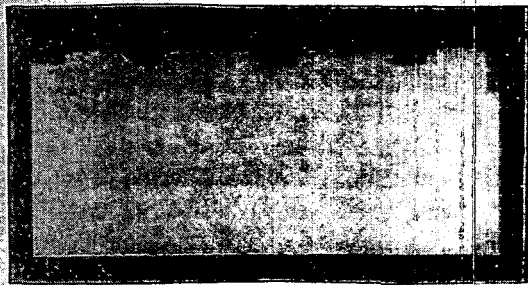
(a)
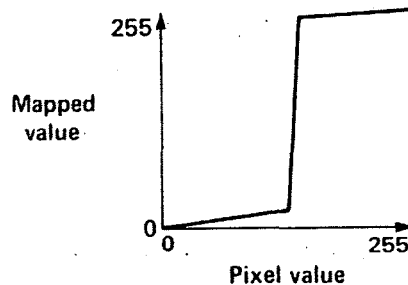
(b)

(c)

Fig. 3. Example of image enhance-
ment by simple nonlinear mapping
of gray-scale values.

a) Original rail-gun diagnostic
   radiograph.
b) Gray-value mapping used to en-
   hance contrast over a small
   range of gray levels.
c) Resulting enhanced image.

Pseudocolor. Pseudocolor display is a technique that maps each of the gray levels of a gray scale image into an assigned color[25]. As such, the pseudocolor operation can be viewed as a point, or pixelwise, operator as discussed above. The resultant colored image can make the identification of certain features easier for the observer. The mappings are computationally simple and fast. This makes pseudocolor an attractive technique for use on digital image processing systems that are designed to be used interactively. Furthermore, most commercially available image processing and display systems incorporate the necessary mapping registers and associated hardware (including color display capability) for pseudocolor implementations. Various color maps can give contrast enhancement effects, contouring effects, or gray-level mapping (depicting areas of a prescribed gray-level value). Pseudocolor schemes can also be designed to preserve or remove intensity information.

The relative success or failure of a particular pseudocolor scheme will depend on the nature of the image. One property that has a very noticeable effect is the noise in the image. Natural images such as digitized photographs of people, buildings, etc., tend to have variations of several gray levels in adjacent picture elements although the intensity may appear fairly uniform. The human visual system can filter out the high-frequency noise fairly well when the intensity is affected. However, when the intensity noise is converted to colors by a pseudocoloring operation, the image can be difficult, if not impossible, to see.

The effect of noise on pseudocolor schemes that change colors at high rates (such as a random color assignment) is severe. The eye will combine the many different colors in a small area into an intermediate color determined by color mixture laws. If these colors are not ordered in some fashion, it will be difficult to perceive features of interest in the image.

Change Detection via Pointwise Operators. A straightforward pointwise operator which can be extremely powerful in appropriate applications is that of image subtraction. The subtraction of similar images is particularly applicable to the general class of image processing problems called "change detection." In change detection, the objective is to identify differences between two variations of the same scene. Examples include detection of defects in LSI circuit masks (by comparing a test mask image with a master mask known to be good) and satellite monitoring of the movement of military equipment.

A major problem with change detection algorithms is that there may be large variations between the scenes being compared that are of no interest. For example, in the satellite monitoring of military equipment, variations in cloud cover, sensor performance, or scene illumination can make it virtually impossible to apply change detection in a meaningful way. Even in more well-controlled applications such as the comparison of circuit masks, a slight misregistration of the two images can cause severe problems for a change detection scheme.

One scenario in which change detection can be particularly effective is in real-time monitoring. In many real-time applications, the scenes being compared will not suffer from misregistration or illumination changes because they are typically from adjacent frames in a video picture, displaced by only 1/30 of a second in time. To illustrate this, consider the example shown in Figure 4. Figure 4a shows an aerial photograph of a weapon test site during an underground nuclear explosion. Figure 4b is the following frame from the videotape. The result of image differencing is shown in Figure 4c. The circular annulus observed in the enhanced difference image is a result of the so-called "slapdown" surface motion caused by the explosion. This example clearly illustrates that very simple pointwise operators can indeed be powerful image enhancement techniques.

(a)

(b)
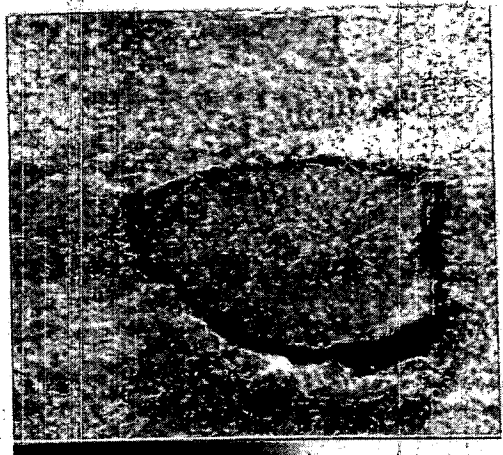
(c)

Fig. 4. Change detection application to video monitoring of weapons test site.

    a) One frame from video tape.
    b) The video frame immediately following that of (a).
    c) Difference image.

LOCAL OPERATORS

A very large class of image processing operations involve computing each output pixel as a function of a small neighborhood of input pixels surrounding the corresponding input pixel. Such neighborhood, or local, operators include many linear and nonlinear algorithms. The implementations of these operations are usually straightforward, at least with general-purpose computers, because only a small number of input image rows need be stored in the primary storage at a given time. In addition, many special-purpose image processing systems provide the hardware to perform a number of such local operations on the entire image in real time (video rates) or in near real time.

Perhaps the simplest local operator is the so-called nonrecursive linear filter. In two dimensions, this operator can be described by the equation

$$g(m,n) = \sum_{k=-K_1}^{K_2} \sum_{\ell=-L_1}^{L_2} h(k,\ell) f(m-k, n-\ell)$$

i.e., the output pixel at the point (m,n) is simply a weighted (by h) sum of the input pixels f.

The general nonrecursive filtering problem involves the design or choice of the appropriate filter coefficient array h(m,n). In many cases, certain ad hoc limited-extent masks are chosen because of their simplicity and ease of implementation. For instance, the low-pass filter operator

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

simply averages the pixels in a 3 x 3 neighborhood to compute the output pixel value. Similarly, the mask

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

performs a high-pass filtering operation on an image.

Although the 3 x 3 filters discussed above are often used in image processing applications, it is sometimes necessary to design filter arrays with much larger regions of support. A primary example of such a need arises when the desired filtering operation is used to perform some spatial-frequency domain selective filtering. For most typical filter responses, one needs filters that are 15 x 15, 25 x 25 or sometimes even larger to achieve the required characteristics. Several techniques for designing 2-D nonrecursive

filters based on frequency specifications have appeared in the literature. The most commonly used and/or versatile of these approaches include the windowing approach discussed by Huang[26], the McClellan transformation technique presented by Mersereau et al.[27], and Lodge and Fahmy's[28] nonlinear optimization-based technique.

Another effective neighborhood, or local, image processing operation is the so-called "unsharp masking" method. This technique consists of subtracting a weighted low-pass version of the input image from the image itself in order to enhance high-pass characteristics. A variation of the unsharp masking technique is to use a median filtered (discussed in the next section) version of the input image as $f_L(m,n)$, the low-pass filtered input. Additional examples of such local operators include recursive filters, several interpolation techniques, sectioned methods for convolution and image restoration, and many others. Two techniques involving nonlinear manipulations of pixels in a limited neighborhood which have found increasing applicability in a diverse set of image processing problems are the so-called median filter and contrast stretching using local statistics. We now briefly describe these two neighborhood operators and illustrate their utility on sample digital images.

Median Filtering. Median filtering is a very effective tool for digital image noise suppression filtering. The operation of this filtering technique is straightforward: at each pixel location, the output pixel is simply the median value of the pixels in an L x L neighborhood surrounding that pixel in the input image, where L is generally an odd number. Variations of this procedure include using a rectangular, rather than square, filter support, using a discrete approximation to a circular support, and using two one-dimensional median filters. Due to the nonlinear nature of the median filtering procedure, this technique is particularly successful in the filtering of images with impulsive noise or highly spatially-limited contaminating noise. From an image processing standpoint, it also possesses the particularly attractive tendency to successfully filter out random noise effects while blurring edges or other signal features very little. This is a clear advantage over linear noise suppression filtering techniques, which tend to blur image features in an objectionable manner.

Due to the unique capabilities of the median filter, this technique has been increasingly studied in terms of both functional usefulness and for fast implementations. As an example, Frieden[29] has incorporated the median filter into an iterative image deconvolution scheme for the restoration of edges in images.

Many papers have recently been published which describe fast implementation schemes for 2-D median filtering[30,31,32]. In general, these techniques incorporate fast sorting algorithms for median computation, take advantage of the commonality of data points in the filter window in adjacent output pixel computations, and/or utilize the limited number of bits per pixel used to represent the input.

<u>Contrast Stretching</u>. Another effective image enhancement technique which involves the use of only small local neighborhoods around a pixel in the computation is that of contrast stretching. The term contrast stretching implies an algorithm which enhances low-contrast details in images by adjusting local means and variances of the picture elements throughout the image. Typically, much of the dynamic range in an image is spanned by large differences in intensities of distinct regions. In radiographic imagery, for example, a region caused by an object which attenuated very little of the incident radiation would appear very dark in the radiograph. Another region, caused by an object of high attenuation, would appear very light. It is easy to notice these large differences in the image, but it may be difficult to see detail in either region alone. The gray levels of pixels in regions of low contrast generally are close to one another; in other words, their standard deviation is small. By decreasing the difference between the means of the gray levels of the two regions and increasing the standard deviations of the gray levels in each region, details may become more apparent. Such an adjustment of gray levels makes better use of the limited dynamic range that is available.

The contrast stretching algorithm is designed to equalize the local mean and standard deviation throughout the image. The transformation which computes the contrast stretched image $g(m,n)$ from the original $f(m,n)$ is the equation

$$g(m,n) = \alpha\mu_d + [1-\alpha]\mu(m,n) +$$

$$\sigma_d[f(m,n)-\mu(m,n)]/[\sigma(m,n)+\sigma_d/G_{max}]$$

where $\mu(m,n)$ is the local mean at the point (m,n), $\sigma(m,n)$ is the local standard deviation, μ_d is the desired mean, σ_d is the desired standard deviation, and α is a parameter in the range $(0,1)$ which determines the degree to which the mean value is equalized (i.e., $\alpha = 0$ implies the mean is unaffected). G_{max} is a term which limits the maximum multiplicative gain which the filter may impose (by preventing a divide by zero). Note that this transformation is nonlinear and space varying.

GLOBAL OPERATORS

Another class of image processing operations which can be characterized by the nature of the implementation is that of global operators. By global we will mean that the computation of an output pixel is dependent on a large part, if not all, of the input image. Many image processing algorithms which are global in nature involve two-dimensional transforms. By far, the most important of these transforms is the 2-D Fast Fourier Transform (FFT), which is an efficient algorithm for computing the 2-D discrete Fourier transform defined as

$$F(k,\ell) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)e^{-j2\pi(km/M + \ell n/N)} \; ,$$

$$\text{for } 0 \le k \le M - 1$$
$$0 \le \ell \le N - 1 \; ,$$

where f(m,n) is the M x N dimension image being transformed. Note that such a transform is global in the sense that for any (of the MN possible) choice of output pixel location (k,ℓ), all MN of the input pixels enter into the calculation. The 2-D FFT plays a pervasive role throughout the field of image processing and is encountered in such applications as image enhancement (implementation of 2-D nonrecursive filters; coefficient rooting), 2-D spectral estimation, and image coding. In addition, a number of image restoration[2] techniques including inverse filtering, Wiener restoration, deconvolution via spectral equalization, and others require the computation of the 2-D FFT and manipulation of the transforms in one way or another. In general, however, the implementation of the 2-D FFT algorithm is nontrivial because the digital image size far exceeds the memory capability of the processor used in the implementation. As a consequence, a number of efficient techniques for implementing a 2-D FFT of disk-based images have been proposed[33-36]. None of these techniques is substantially better than the others in most practical implementations due to the similarity of their input/output (I/O) structures. Recently, however, Ari[37] proposed what he calls a two-level matrix transposition algorithm with a particularly efficient I/O structure. When applied to the problem of implementing 2-D FFT's of images larger than the available primary memory, this technique gives an implementation which is superior, in general, to any other method proposed in the literature.

Many other image transform algorithms have found application to image processing problems, but to a far lesser extent than the 2-D FFT. The main application area for such techniques has been that of image coding. Examples of 2-D transform techniques with implementations not unlike the 2-D FFT which have been used for image coding include the Walsh transform, the cosine transform, the Hadamard transform, and the 2-D slant transform. In addition, several pattern recognition techniques utilize these 2-D transform algorithms.

REFERENCES

1. Castleman, K. R., Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1979.

2. Andrews, H. C. and Hunt, B. R., Digital Image Restoration. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1977.

3. Huang, T. S., Picture Processing and Digital Filtering. New York: Springer-Verlag, 1975.

14

4. Pratt, W. K., Digital Image Processing. New York: John Wiley and Sons, Inc., 1978.

5. Gonzalez, R. C. and Wintz, P., Digital Image Processing. Reading, Mass.: Addison-Wesley Publishing Co., Inc., 1978.

6. Hall, E. L., Computer Image Processing and Recognition. New York: Academic Press, Inc., 1979.

7. Mitra, S. K. and Ekstrom, M. P., Two-Dimensional Digital Signal Processing. Stroudsburg, PA: Dowden, Hutchinson & Ross, Inc., 1978.

8. Aggarwal, J. K. et al., Computer Methods in Image Analysis. New York: Institute of Electrical and Electronics Engineers, 1977.

9. Bernstein, R., Digital Image Processing for Remote Sensing. New York: Institute of Electrical and Electronics Engineers, 1978.

10. Andrews, H. C., Tutorial and Selected Papers in Digital Image Processing. New York: Institute of Electrical and Electronics Engineers, 1978.

11. Special Issue on Digital Picture Processing. Proc. IEEE, 60, 7 (1972).

12. Special Issue on Two-Dimensional Signal Processing. IEEE Trans. Computers, C-21, 7, (1972).

13. Special Issue on Digital Image Processing. IEEE Computer, 7, 5 (1974).

14. Special Issue on Digital Signal Processing. Proc. IEEE, 63, 4 (1975).

15. Special Issue on Image Processing. IEEE Computer, 10, 8 (1977).

16. Special Issue on Pattern Recognition and Image Processing. Proc. IEEE, 67, 5 (1979).

17. Jain, A. K., Image Data Compression: A Review. Proc. IEEE, 69, 3: 349-389.

18. Tou, J. T. and Gonzalez, R. C., Pattern Recognition Principles. Reading, Mass.: Addison-Wesley Publishing Co., Inc., 1974.

19. Gonzalez, R. C. and Thomason, M. G., Syntactic Pattern Recognition---An Introduction. Reading, Mass.: Addison-Wesley Publishing Co., Inc., 1978.

20. Druffel, L. E., Derivation of Information from Images. Proc. SPIE Vol. 205, Image Understanding Systems II: 53-57 (1979).

21. Andrews, H. C., Computer Techniques in Image Processing. New York: Academic Press, Inc., 1973.

22. Hall, E. L., Almost uniform distribution for computer image enhancement, IEEE Trans. Computers, C-23:207-209 (1974).

23. Andrews, H. C., Tescher, A. G. and Kruger, R. P., Image processing by digital computer. IEEE Spectrum, 9, 7:20-32 (1972).

24. Frei, W., Image enhancement by histogram hyperbolization. Computer Graphics and Image Processing, 6, 3:286-294 (1977).

25. Radewan, C. H., Digital image processing with pseudocolor. Proc. 18th Annual Tech. Meeting SPIE (1974).

26. Huang, T. S., Two-dimensional windows. IEEE Trans. Audio Electroacoustics, AU-20, 1:88-89 (1972).

27. Mersereau, R. M., Mecklenbräuker, W. F. G. and Quatieri, T. F., McClellan transformations for two-dimensional digital filtering: I-Design. IEEE Trans. Circuits Syst., CAS-23, 7:405-414 (1976).

28. Lodge, J. and Fahmy, M. M., An efficient optimization technique for the design of 2-D linear phase FIR digital filters. Intern. Conf. Acoust., Speech, Signal Processing Record, Vol. 80CH1559-4:737-740 (1980).

29. Frieden, B. R., A new restoring algorithm for the preferential enhancement of edge gradients, J. Opt. Soc. Am., 66, 3 (1976).

30. Huang, T. S., Yang, G. T. and Tang, G. Y., A fast two-dimensional median filtering algorithm. IEEE Trans. Acoust., Speech, Signal Processing, ASSP-27, 1:13-18 (1979).

31. Garibotto, G. and Lambarelli L., Fast on-line implementation of two-dimensional median filtering. Electron. Lett., 15:24-25 (1979).

32. Ataman, E., Aatre, V. K. and Wong, K. M., A fast method for real-time median filtering. IEEE Trans. Acoust., Speech, Signal Processing, ASSP-18, 4:415-421 (1980).

33. Eklundh, J. O., A fast computer method for matrix transposing. IEEE Trans. Computers, C-21:801-803 (1972).

16

34. Twogood, R. E. and Ekstrom, M. P., An extension of Eklundh's matrix transposition algorithm and its application in digital image processing. IEEE Trans. Computers, C-25:950-952 (1976).

35. Onoe, M., A method for computing large scale two-dimensional transform without transposing data matrix. Proc. IEEE, 63:196-197 (1975).

36. Harris, D. B., et al., Vector radix fast fourier transform. Proc. 1977 IEEE International Conf. on Acoustics, Speech, and Signal Processing: 548-551 (1977).

37. Ari, M. B., On transposing large $2^n x 2^n$ matrices. IEEE Trans. Computers, C-27:72-75 (1979).

DISCLAIMER