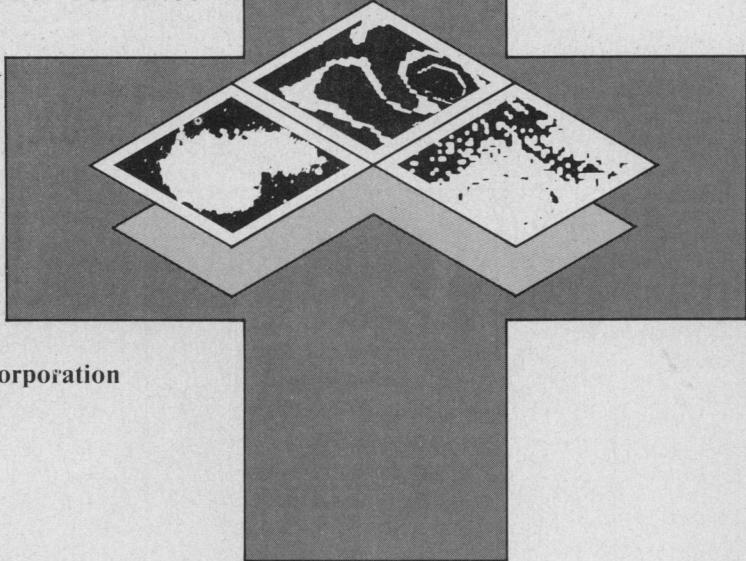


The burden on image analysts in medical fields has led to the automated processing of pictorial data. Here, a device called the cytocomputer searches for genetic mutations.

Biomedical Image Processing

Stanley R. Sternberg, CytoSystems Corporation



A computer revolution has occurred not only in technical fields but also in medicine, where vast amounts of information must be processed quickly and accurately. Nowhere is the need for image processing techniques more apparent than in clinical diagnosis or mass screening applications where data take the form of digital images. New high-resolution scanning techniques such as computed tomography, nuclear magnetic resonance, positron emission tomography, and digital radiography produce images containing immense amounts of relevant information for medical analysis. But as these scanning techniques become more vital to clinical diagnosis, the work for specialists who must visually examine the resultant images increases. In many cases, quantitative data in the form of measurements and counts are needed to supplement nonimage patient data, and the manual extraction of these data is a time-consuming and costly step in an otherwise automated process. Furthermore, subtle variants of shade and shape can be the earliest clues to a diagnosis, placing the additional burden of complete thoroughness on the examining specialist.

For the last five years, the University of Michigan and the Environmental Research Institute of Michigan have conducted a unique series of studies that involve the processing of biomedical imagery on a highly parallel computer specifically designed for image processing. System designers have incorporated the requirements of extracting a verifiable answer from an image in a reasonable time into an integrated approach to hardware and software design. The system includes a parallel pipelined image processor, called a *cytocomputer*, and a high-level language specifically created for image processing,

C-3PL, the cytocomputer parallel picture processing language.

These studies have involved a great many people from both the medical and engineering communities and have highlighted the interdisciplinary aspects of biomedical image processing. The methods have been tested in anatomy, developmental biology, nuclear medicine, cardiology, and transplant rejection. The general consensus is that quantification by automated image analysis not only increases diagnostic accuracy but also provides significant data not obtainable from qualitative analysis alone.

One study in particular, on which descriptions in this article are based, involves a joint effort by the University of Michigan's human genetics and electrical and computer engineering departments and is supported by a grant from the National Cancer Institute. Basically, automated image analysis is being applied via sophisticated biochemical and computer techniques to derive an accurate estimate of the mutation rate for the human species.

Study overview

We are becoming increasingly concerned about human exposure to environmental elements, particularly those agents that may be carcinogens or mutagens. The long-term effects of exposure to chemical poisons and low-level radiation must first be quantitatively understood before their ultimate costs can be taken into account in

the political process of their control. As a step in quantifying the impact of mutagenic agents on human populations, techniques are being developed to detect mutation and to estimate the mutation rate in sample populations.

In the study of human mutation rate at the University of Michigan, placental cord blood samples are collected from newborns as are venous samples from their mothers and fathers. Samples of lymphocytes, red blood cell membranes, red blood cell contents, and blood plasma are being subjected to *2-D gel electrophoresis*. A mutation is indicated when a variant of an identified protein appears in a child that is not present in either parent. The challenge is to develop that combination of specimens and techniques that will yield the greatest amount of information about mutation per unit of effort.

For the last 36 years, geneticist J. V. Neel of the University of Michigan School of Medicine has studied germ cell mutations in radiation-exposed Japanese populations.^{1,2} Using biochemical methods to detect protein variants at a cellular level, the approach was aimed at detecting single amino acid substitutions in a child's proteins that have been caused by mutations of parent's ova or sperm germ cells. Recently, 2-D electrophoresis systems in which cellular proteins are automatically mapped according to both their molecular weight and molecular charge have been developed. The 2-D gel electrophoresis system is sufficiently sensitive to detect protein charge changes caused by single amino acid substitutions. The significant advantage of the 2-D gel system over other biochemical methods is that many proteins are treated in parallel on a single gel, each protein a potential candidate for mutation. By contrast, traditional methods of estimating mutation rate have relied on the appearance of congenital defects, stillbirths, and infant survival rates, all of which severely limit the size of sample populations.

The method of separating proteins in two dimensions on a polyacrylimide gel medium is due mainly to O'Farrell,³ with substantial improvements to the method by Anderson and Anderson.⁴ First, the cellular material, usually blood, is fractionated into different cell types by centrifuging. Separation in the first gel dimension is by isoelectric focusing in which the proteins are separated on the basis of their molecular charge, or their isoelectric point. Separation in the second dimension is done by electrophoresis after the proteins have been treated by a detergent that masks the proteins' molecular charge and permits electrophoresis to resolve proteins on the basis of their molecular weights. At the end of the second step, each protein has migrated to a position (x, y) on the 2-D gel, where x reflects the molecular charge (pH) of the protein and y reflects its molecular weight. To see the proteins in the gel and perform qualitative and quantitative analysis, the proteins are either radioactively labeled during their synthesis and detected by autoradiography or stained at the end of electrophoresis. The individual proteins appear on the stained gel or autoradiogram as spots of different size and intensity. The integrated density of each spot is proportional to the amount of a given protein in the sample. Figure 1 illustrates a silver-stained 2-D electrophoretic gel.

The cyocomputer: a biomedical image processing tool

Development and use. In the University of Michigan mutation rate study, my colleagues and I are seeking rare events—only one mutation will occur in 100,000 samples—and thousands of mother, father, and child gel trios must be examined spot by spot for variations in protein patterns. This tedious task is clearly best suited to computerized visual comparison rather than human vision alone. (For more on computer analysis of 2-D electrophoretic gels in general, see Skolnick et al.⁵)

Searching for a rare protein mutation in a large number of gels requires that processing be very fast and that operator interaction be required only if a mutation is suspected. For this reason, we have directed efforts to the implementation of highly automated gel processing on a special-purpose computer characterized by a large capacity for parallel processing. The *cyocomputer*, which was developed at the Environmental Research Institute of Michigan, runs programs that apply sequences of neighborhood transformations to digitized gel images. Cytocomputer image processing operations are based on the concepts of cellular automata. Every cell or picture element (pixel) of an image is subjected to an identical sequence of time-discrete transformations, the transformed value of a pixel being determined by the values of a finite group of cells that make up its "neighborhood." Each neighborhood transformation is performed in an individual cytocomputer processing element referred to as a processing stage.

A cytocomputer is a serial pipeline of programmable processing stages, in which each stage performs a single

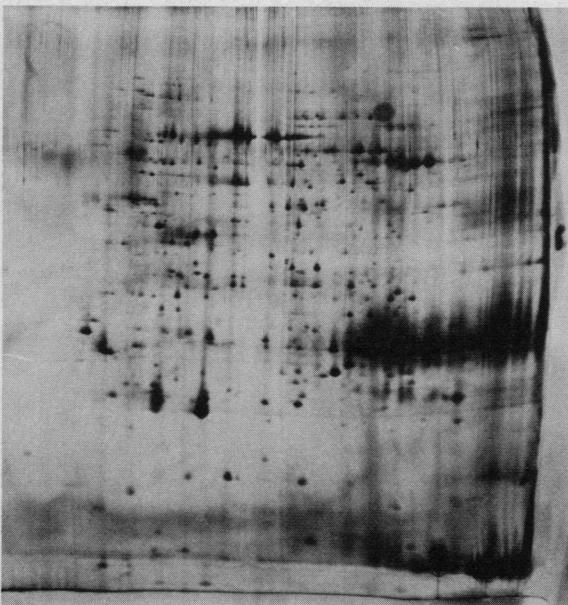


Figure 1. Two-dimensional silver-stained electrophoretic gel of red blood cell contents. The spots are individual proteins, spatially separated according to their molecular weight vertically and molecular charge horizontally. Prepared by Barnett Rosenblum of the University of Michigan.

transformation of the processing sequence on an entire image. Images are entered into a cytocomputer in a line-scanned format and progress through the pipeline of processing stages at a real-time rate. Following an initial delay to fill the pipeline, images can be processed at the same rate they are scanned.

Digital images and cellular automata have a common conceptual framework. Each pixel of a digital image can be thought of as a cell in a given state. If we define a neighborhood and a cell transition function on a digital image, then we can apply the transition function to modify or transform the configuration of cell states into new configurations. Of critical importance then is whether neighborhoods and transition functions exist that will cause images to be transformed in a predictable and useful manner when subjected to long sequences of neighborhood operations.

The image processing language we are investigating differs from conventional approaches in that the basic manipulative unit is pictorial and operations deal with images as wholes. Image processing is treated as a computation involving images as variables in algebraic expressions. These expressions may combine several images through both logical and geometrical relationships.

Architecture. Cytocomputer operations are implemented in highly efficient cellular computer architectures, and the computations are very fast. The Cyto I cytocomputer executes 140 million neighborhood operations per second in an 88-stage pipeline. Image processing algorithms are constructed as well-formed strings of primitives that are either variables representing images or image operations. The image being processed is referred to as the *active image*. Other images referred to in an image-algebraic expression are called *structuring elements*. In an image processing algorithm, we can modify the active image by probing it with structuring elements or combining it with other active images.

The neighborhood in a cellular space determines the set of structuring elements that can be employed in a single neighborhood transformation. Each pixel of a digital image belongs to a window of pixels composed of the given pixel and its neighbors. All structuring elements used in a neighborhood transformation must be sub-images of the window.

Consider a two-dimensional cellular array, where each cell of the lattice has connections with a finite collection of other cells that make up its input. The geometric pattern of the cells input to a given cell is the same as the pattern of the points in the neighborhood. Figure 2 illustrates a cellular array with the connection pattern for a 3×3 window configuration. In Figure 3, each cell of the array consists of a register for storing the state of the cell and a transition module that computes the new value of the cell state as a function of the states of the cells in the window. When a common clock pulse is applied to each cell in the array, all cell state registers pass from their previous state to a new state as determined by the programming of the transition logic module.

Although digitized biomedical image dimensions can often exceed 1000×1000 pixels, the largest arrays yet produced are only on the order of 100×100 . (See Potter's article on the MPP in this issue.) Large images must be partitioned into image segments, and each segment processed in turn. However, segment border effects propagate into the segment when multiple neighborhood transformations are applied, necessitating extremely costly and time-consuming I/O hardware and software subsystems for rapid segment swapping. The problem is remedied in the pipeline architecture where parallelism of image operations is used instead of pixel parallelism. (Danielsson and Levialdi give a good review of image processor architectures.⁶)

A cytocomputer consists of a serial pipeline of commonly clocked neighborhood processing stages (Figure 4).⁷ Shift registers within each stage store two contiguous

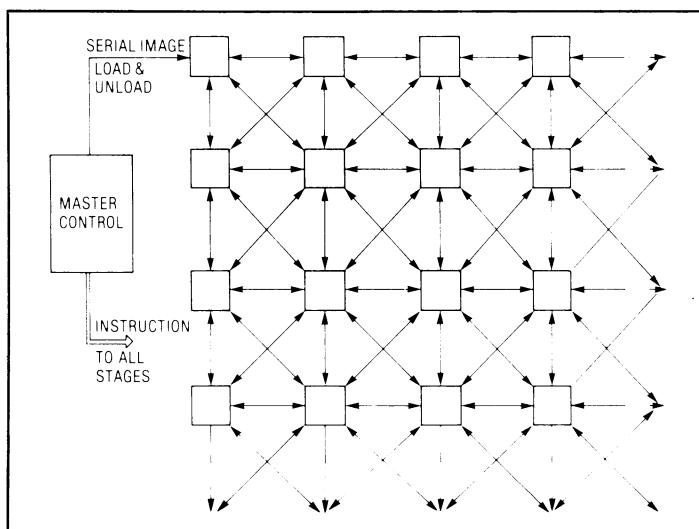


Figure 2. Array similar to cellular automata of identical cells connected to their nearest neighbor for iterative neighborhood processing of digital images.

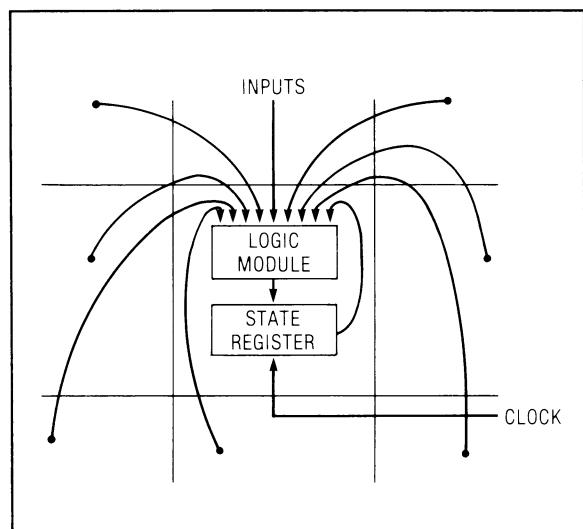


Figure 3. Cellular array block diagram. Inputs from neighborhood cells form the address to a RAM lookup tabulation of the neighborhood transition function.

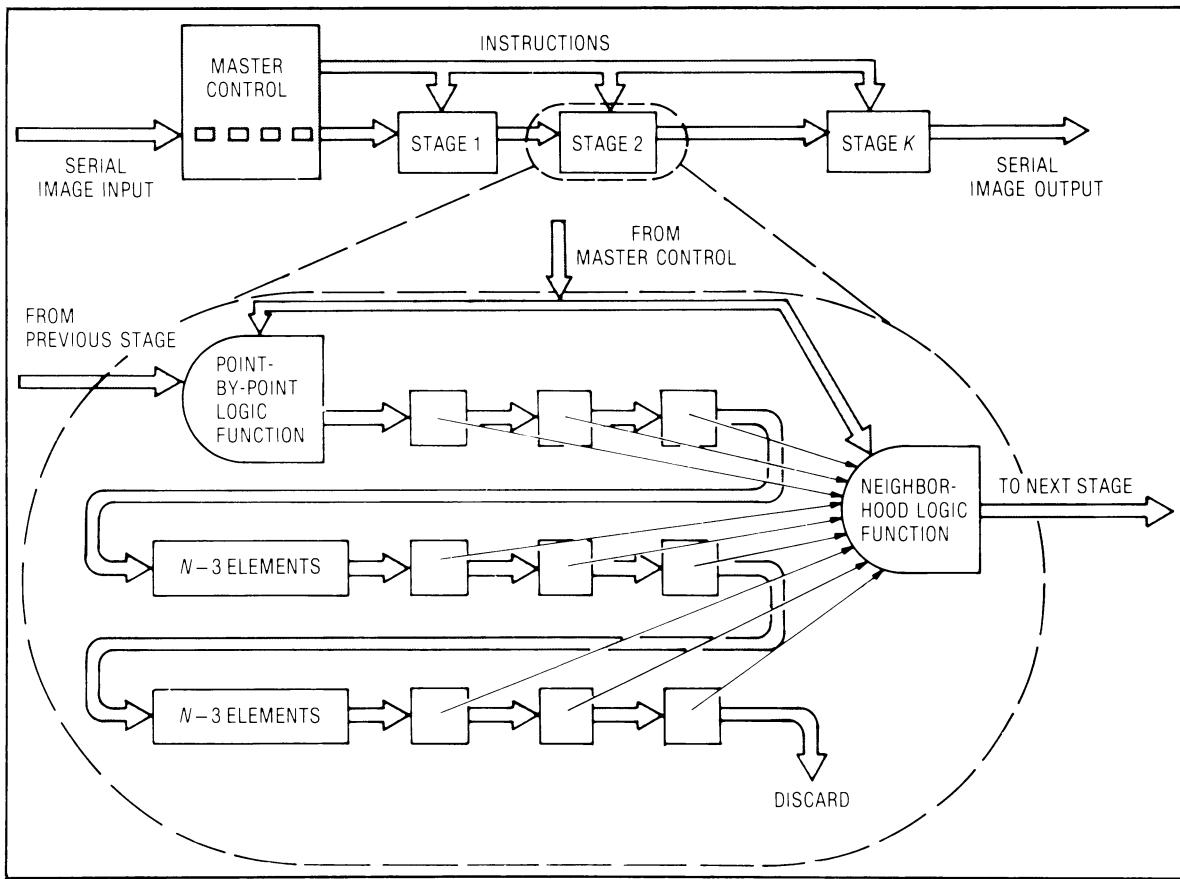


Figure 4. Cytocomputer block diagram.⁷ Serpentine shift register delays in each stage serially configure neighborhood inputs to the neighborhood logic module. A pipeline of K stages executes a K step neighborhood processing algorithm in real time. (Copyright IEEE, 1981.)

n pixel scan lines, and window registers hold the nine neighborhood pixels that constitute the 3×3 input to the neighborhood transition logic module. All neighborhood transformations and data transfers are computed within a single clock period. In each discrete clock period, a new pixel is clocked into the stage, and simultaneously the contents of all shift register delays are shifted by one element. In addition, operations that do not involve the states of the pixel's neighbors, such as gray-value scaling and bit setting, are performed in a separate point transition logic module to simplify the neighborhood transition logic circuit. Because stage input and output occur at the same rate, stages can be cascaded, with all stages operating in parallel. We can visualize a series of 3×3 windows following each other across the image, each processing the previous stage's output as shown in Figure 5.

Transformations implemented in a cytocomputer stage fall into two main categories. In the first, called the silhouette, or 2-D, transformation, image algebra operations are applied to planar binary images. In the second, called the umbra, or 3-D, transformation the same set of operations are performed on gray-scale images, where the gray value represents the brightness of a picture point or its height above an arbitrary reference plane. In umbra transformations the structuring elements are umbras of subimages in a $3 \times 3 \times 3$ window.

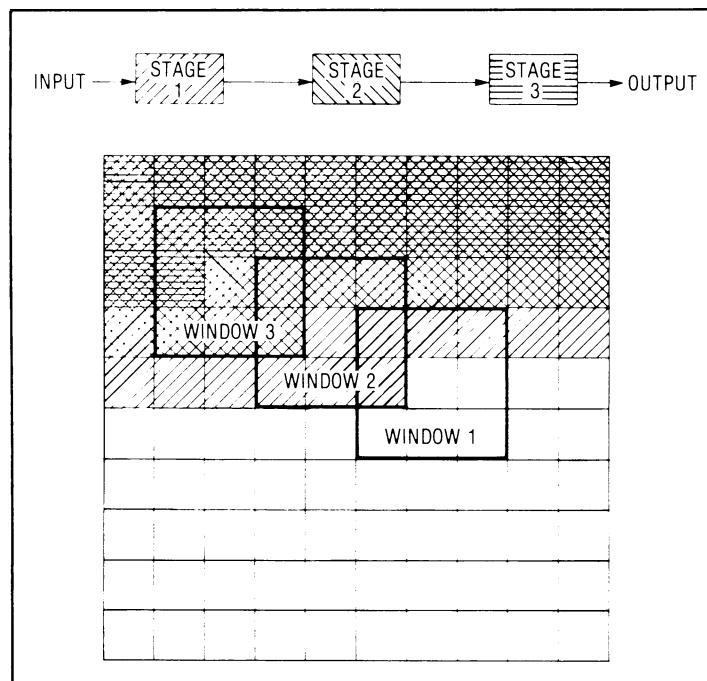


Figure 5. Cytocomputer stage windows sequentially scanning across a digital image. That portion of the image processed by the i th stage is immediately available for processing by the $(i + 1)$ th stage.

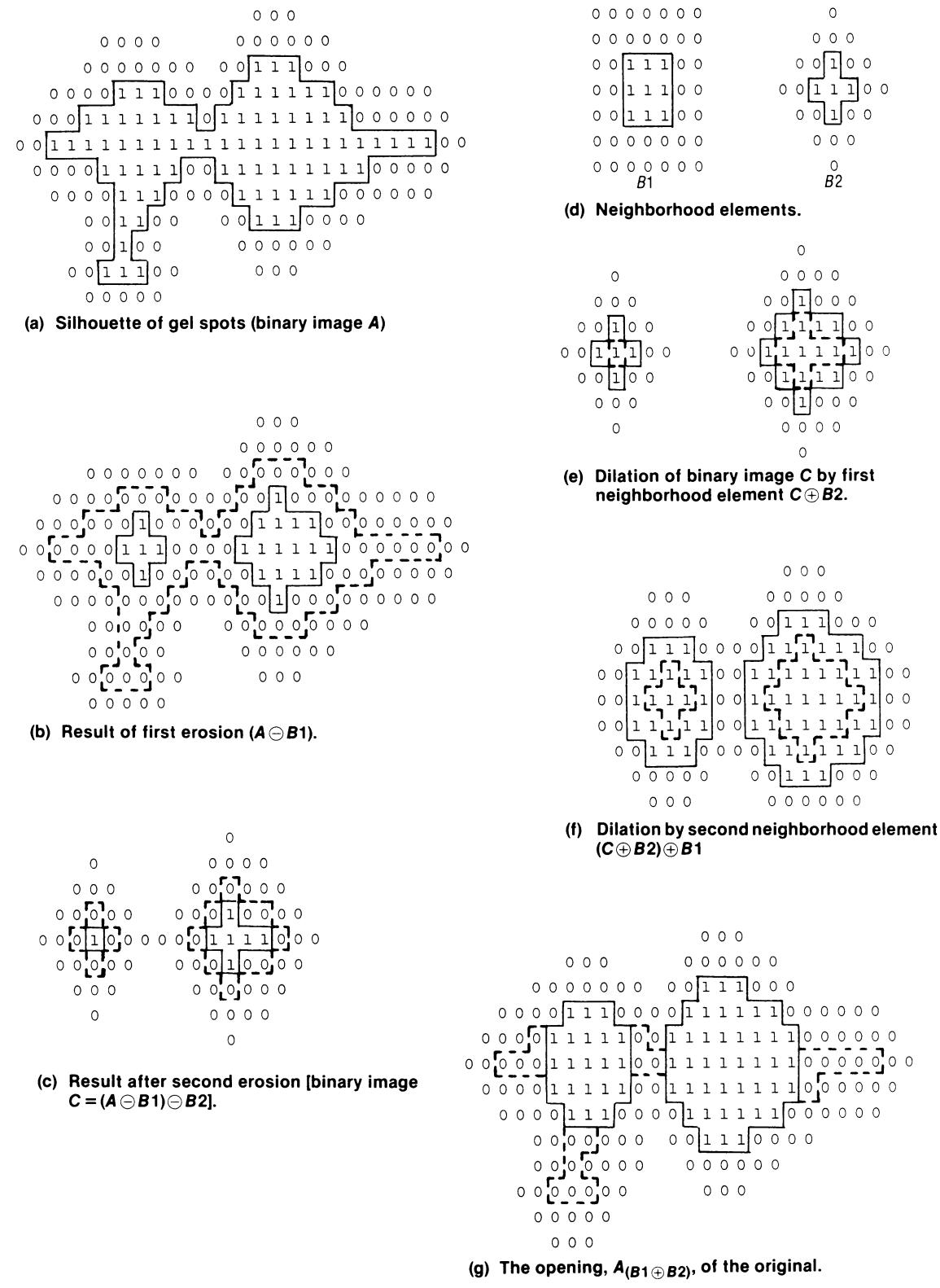


Figure 6. The morphological operations, erosion, dilation, and opening, illustrated on a binary example of electrophoretic gel spots. The process of opening separates the two connected spots and cleans them.

Image processing language: the processing link

The common thread in all biological image processing with cytocomputers is the belief that forms encountered and recorded in biology and medicine can be described in standard terms, or structuring elements. The structuring element probes the image, and in the process, transforms it. The transformed value of a pixel indicates the fit of the probe as applied to the image at that point.

Mathematical morphology is that part of image processing concerned with image analysis by structuring elements. The Center for Geostatistics and Mathematical Morphology of the Ecole des Mines in Fontainebleau, France, has been a traditional source of European image analysis expertise for more than 15 years. Mathematical morphology grew out of the early work of H. Minkowski,⁸ who used dilation to obtain appropriate measures on the parameters of certain ill-behaved sets. Later, the equivalence of sets and binary images led to the direct application of integral geometry to problems of estimating underlying physical parameters from samples in the image form.

Mathematical morphology entered the modern era through the work of G. Matheron⁹ and J. Serra¹⁰ of the Ecole des Mines. Motivated by problems in mining estimation, Matheron and Serra not only formulated the modern concepts of morphological image transformations but also designed and built a sophisticated image analyzer, the Texture Analyzer System,¹¹ for studying mineralogical samples.

The processing of an image by mathematical morphology is illustrated in Figure 6a, which is a binary silhouette visualized as a topographic slice at constant gray level within a gel section containing two closely adjacent spots. The sequence of neighborhood transformations involved is divided into a subsequence of *erosions* followed by a subsequence of *dilations*. Hence, mathematical morphology is useful in isolating and filtering individual spots.

A two-step erosion sequence is shown in Figures 6b and 6c. The neighborhood transformation rule applied in going from 6a to 6b is simply to set any pixel to 0 if any cell in its nine-cell window is 0—the “ANDing” of all the cells in the window. The transition rule used in the transformation from Figure 6b to 6c is similar except that only the four edge-connected pixels and their center are tested. These two transformation conditions are illustrated in Figure 6d as structuring elements $B1$ and $B2$, respectively. We can say that the binary image of Figure 6a, which we denote A , has been successively eroded by structuring elements $B1$ and $B2$ to produce result C (Figure 6c). This event can be expressed algebraically as $C = (A \ominus B1) \ominus B2$, where \ominus stands for the operation of erosion. The erosion of A by Bk can be defined as determining pixels p of A to which the origin of structuring element Bk can be translated, denoted Bk_p , such that it is entirely contained within A ; $A \ominus Bk = \{p : Bk_p \subset A\}$.

Two successive dilation steps involving the same structuring elements $B1$ and $B2$ yield the image in Figure 6f. The dilation rule states that any pixel having a 1 in a window position indicated by the structuring element is

transformed to 1—the “ORing” of the designated cells in the window. The results of successive dilations by structuring elements $B2$ and $B1$ are illustrated in Figures 6e and 6f, respectively. The dilation sequence of operations can be symbolized as $D = (C \oplus B2) \oplus B1$, where \oplus stands for the operation of dilation. The resulting processed image (Figure 6g) consists of two spots, which are clearly definable because extraneous pixels have been removed.

The dilation of C by Bk can be defined as the union of translations of the structuring element Bk to the points of C , or

$$C \oplus Bk = \bigcup_{p \in C} Bk_p$$

Since dilation is commutative ($C \oplus Bk = Bk \oplus C$), the order of the structuring element dilations is immaterial.

In mathematical morphology, the process of eroding and dilating by the same sequence of elemental structuring elements is called an *opening*. The result of an opening by a sequence $B1, \dots, Bk$ of structuring elements, as illustrated in the preceding example for $k=2$, can be expressed in an equivalent way in which the series of structuring elements can be combined into a single form B . Starting with $B1$ we successively dilate the structuring elements $B2, \dots, Bk$ to create a new structuring element B , or $B = [(B1 \oplus B2) \oplus B3] \oplus \dots$

The result of the opening process can be conveniently visualized in terms of the structuring element B alone. In this example, dilation of the structuring elements $B1$ and $B2$ gives the structuring element B shown in Figure 7a.

The opening of the active image is a new binary image consisting of only those pixels that were 1 in image A and that can be covered by at least one placement of B entirely inside A . That is, the program searches (in parallel) for all positions in A where we can place B and have all of the 1 pixels of B cover a 1 pixel of A . Figure 7b shows a possible position. Any pixel of A that can be covered by any

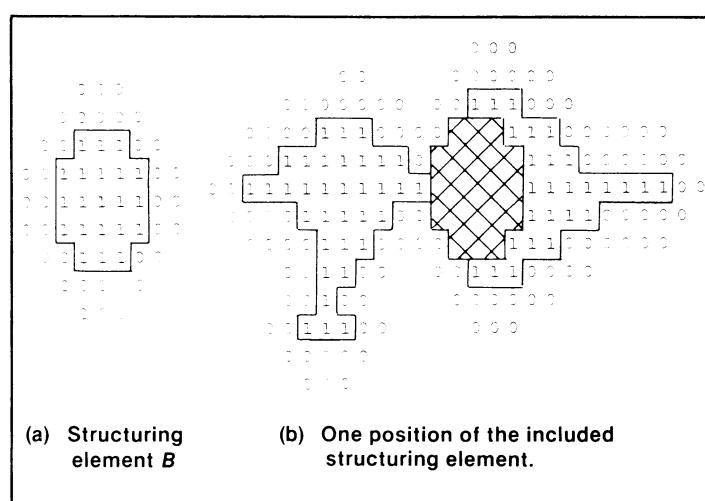


Figure 7. The opening of the binary spot image visualized as pixels that can be covered by the structuring element B as it slides around inside the spots. This result is implemented as a series of neighborhood transformations.

such included position of B belongs to the opening of A by B . Any 1 pixel of A that cannot be so overlapped becomes 0 after opening. The opening of A by B is the union of all translations of B that can be included in A , or $A_B = (A \ominus B) \oplus B$.

Thus, the transformation of opening an image allows us to determine where the given structuring element B

can fit as it “slides around” within A , the image being opened. I will now generalize opening to three dimensions and show how opening with an approximately spherical structuring element is useful in the process of correcting for variations in the background level in 2-D gels, the technique used in the human mutation rate study discussed earlier.

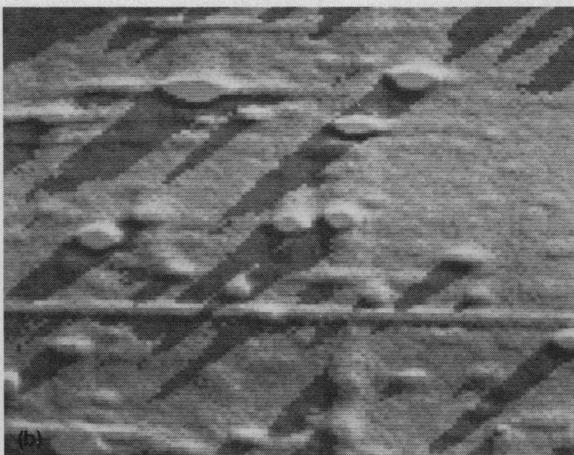
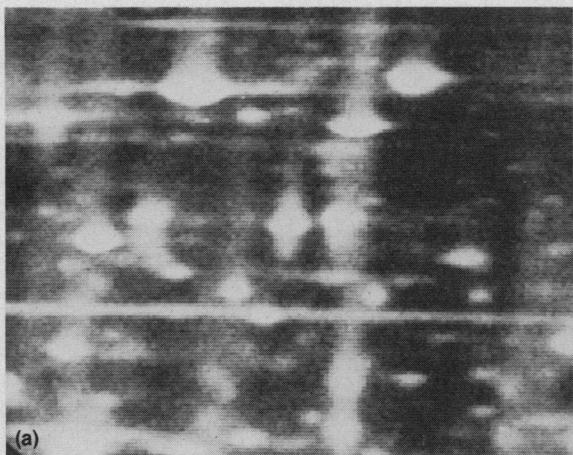


Figure 8. Section of a lymphocyte gel autoradiogram (a) and its shaded and shadowed representation (b) illustrating the 3-D umbra interpretation of the same data.

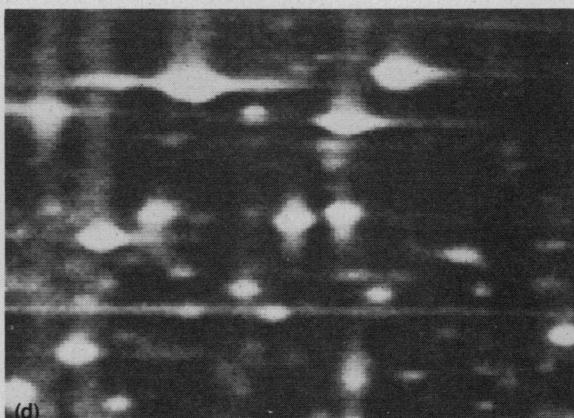
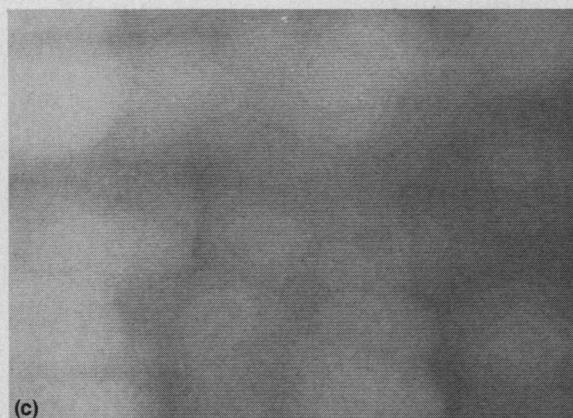
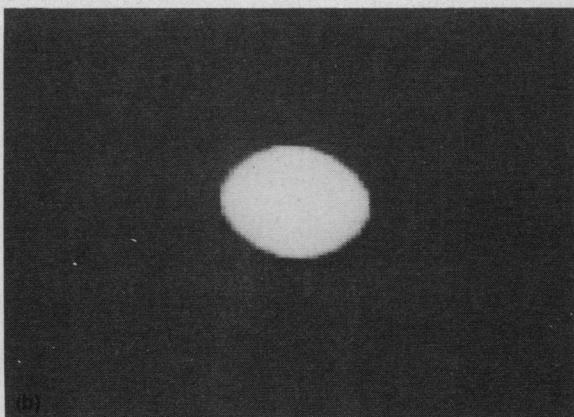
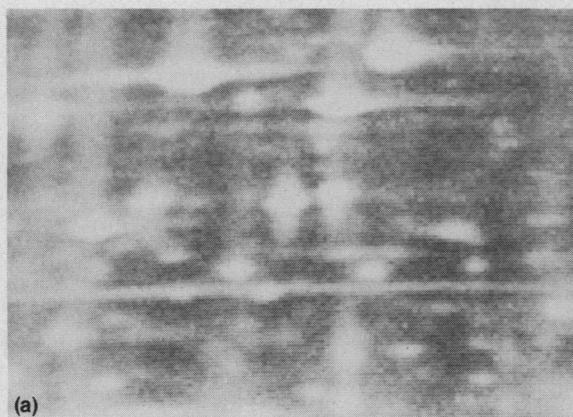


Figure 9. Process of background-normalization. Original gel section (a) is opened by the spherical structuring element (b) producing the gel background image (c), which is subtracted from the original, producing the result in (d).

Cytocomputer applications

Gray-scale morphological processing. Two closely spaced spots can often appear in the child's gel when each parent's gel displays only one. Here, the integrated density of each spot in the pair is half that of the parents. Thus, we place a genetic constraint on the processing of the gels by using a lower threshold on genetically screened proteins so that protein concentrations reduced by half are clearly discernible above background noise. The problem in implementing this step directly on the gel images is that the background intensity level, on which the spots appear, is not uniform over a gel and varies between gels. Thus, we must remove the background from the gel image before taking the threshold.

Programs have been developed that efficiently estimate the background level across a gel image. Subtracting the estimated background image from the original gel image gives the background-normalized gel image. The process of background estimation is an extension of the binary image opening process previously described. The only difference is that the digital spot images in the first example are arrangements of pixels whose values are either 1 or 0, and a gray-scale digital image has pixel intensity values of anywhere between 0 and 255. In defining gray-scale neighborhood transformations, we need to view the image as a set of "boxels," or cubical pixels in a 3-D volume. This representation, called the umbra of a gray-scale image, consists of rows and columns of vertical piles of boxels; the height of the pile (number of boxels) at position x,y in the umbra is equal to the gray level of the pixel at position x,y in the gray-scale image.

The umbra representation of a 2-D electrophoretic gel can be visualized as an extraterrestrial landscape of tall peaks and narrow ridges. The composited images in Figure 8 illustrate the umbra representation through appropriate shading and shadowing processing that makes the height interpretation of gel gray levels more visually apparent. Shading and shadowing, which are also implemented morphologically by neighborhood transforms, are detailed elsewhere.¹²

The gray-scale opening is a gray-scale erosion followed by a gray-scale dilation, expressed in terms of a gray-scale structuring element. This opening process is illustrated in Figure 9 for the previously shown gel section. The gray-scale gel image in Figure 9a is opened by the structuring element shown in Figure 9b to produce the image in Figure 9c, referred to as the background image. Figure 9d is the background-normalized image that results from subtracting the background image from the original.

The process of opening a gray-scale image by a gray-scale structuring element to produce the background image is understood in terms of the gray-scale image umbra. The opening of an umbra of a gray-scale gel image by a gray-scale structuring element is the union of all translations of the 3-D structuring element that can be entirely contained within the gel image umbra. We can visualize a solid sphere that moves freely within the solid volume of the gel image umbra but is constrained by the upper surface of the umbra, as shown in Figure 10. The umbra of the gel opening consists of only those boxels

that can be covered by at least one position of the translating sphere. The sphere diameter is selected so that it is considerably wider than any of the peaks formed by the gel spots and thus cannot enter the interior of the peaks. However, the diameter of the spherical structuring element is small enough to follow the smooth contours of the changing background intensity. The background is smooth with respect to the sphere, but the peaks are not.

A large digital sphere appropriately shaded is shown in Figure 11. The sphere is produced as the dilation of the umbras of 26 3-D structuring elements, each structuring element being a subset of a $3 \times 3 \times 3$ window. The sphere erodes or dilates a gray-scale image by a sequence of eroding or dilating neighborhood transformations, each neighborhood transformation being determined by the 3-D structuring elements composing the sphere.

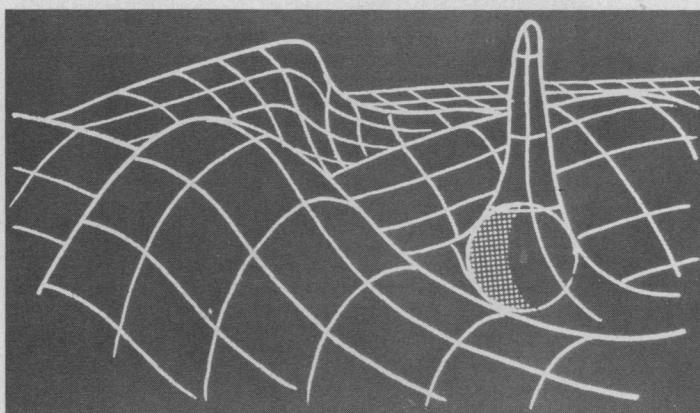


Figure 10. Schematic representation of the rolling ball algorithm used for background normalization. The ball follows the smooth background contours but does not penetrate the spot peaks. Rolling a ball is equivalent to eroding and dilating by a spherical structuring element.

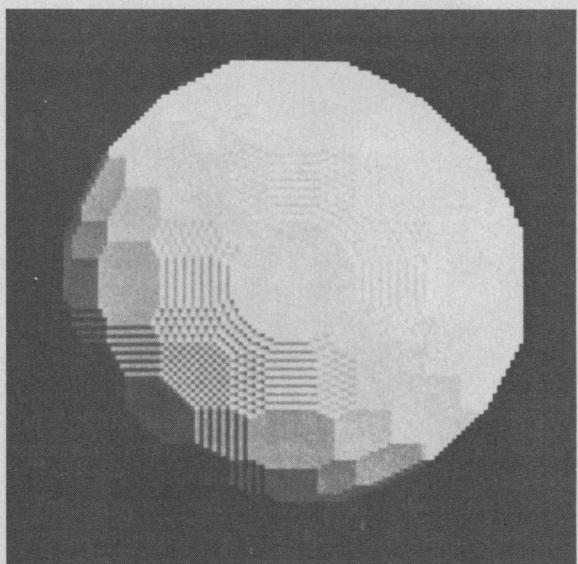


Figure 11. Digital ball. This shaded structuring element is obtained by dilating a single point by a sequence of 26 gray-scale neighborhood operations.

To implement the opening in a parallel processor by sequential gray-scale neighborhood operations, let B_1, B_2, \dots, B_k be a sequence of gray-scale neighborhood elements and let the gray level of the pixel at coordinate location (i, j) in the k th neighborhood element be $B_k(i, j)$, where i and $j = -1, 0$, or 1 and i varies over a 3×3 window. Denoting the gray levels of the active gray-

scale image as $A(x, y)$, the gray levels of the result $C = A \ominus B_k$ are given by

$$C(x, y) = \min_{i, j} [A(x-i, y-j) - B_k(-i, -j)].$$

As in the binary case, the opening of A by B is accomplished by first eroding A successively by the neighborhood elements composing B and then dilating the result by the same sequence. Again, the order of the neighborhood elements does not influence the final result. Dilation of a gray-scale image C by a gray-scale neighborhood element B_k is described by the relation

$$D(x, y) = \max_{i, j} [C(x-i, y-j) + B_k(i, j)].$$

where $D = (A \ominus B) \oplus B = A_B$

Streak removal. The background-normalized gel image exhibits a variety of horizontal and vertical streak patterns (Figure 12a). For genetic comparison, these streaks must be treated as noise, since low-intensity spots falling on a streak increase in intensity and may be detected as spots meeting the previously described concentration constraint. Thus, programs have been developed that remove the streaks while retaining the true intensities of the spots that lie on them. These programs rely on the same principle used in the background-normalization process; that is, 3-D structuring elements are used to open the image surface to define a new surface consisting only of the streaks. This new surface can then be subtracted from the original, effectively removing all the streaks.

Streak normalization is implementation by a pair of gray-scale openings of the background-normalized image. The structuring element of the first opening is a horizontal bar one pixel high whose length slightly exceeds that of the widest spot. The second opening is by a vertical bar one pixel wide whose length is similarly greater than the vertical extent of the largest spots. The two openings can be visualized as vertically and horizontally positioned rolling pins that slide under the image surface to define the vertical and horizontal streaks, respectively. The union of the horizontally and vertically opened images is then formed, as shown in Figure 12b. Subtracting the gel streak image from the background-normalized gel then gives the background-streak-normalized result shown in Figure 12c.

Detecting spot peaks. Thresholding removes spots with intensities that are too low to be clearly distinguishable from background noise. Sufficiently intense spots are then located by a local maxima program. (A local maxima program is adequate for spot detection in silver-stained gels but not in gel autoradiographs, which are much noisier.)

The parallel morphological local maxima program first labels pixels for which the gray level equals or exceeds that of all their neighbors. These labeled pixels are either local maxima or regions of local flatness located on a downslope of a spot or in a valley between spots. (Figure 8 is useful for visualizing these regions.) Next, a second label marks unlabeled pixels that adjoin the local maxima and regions of local flatness whose gray levels

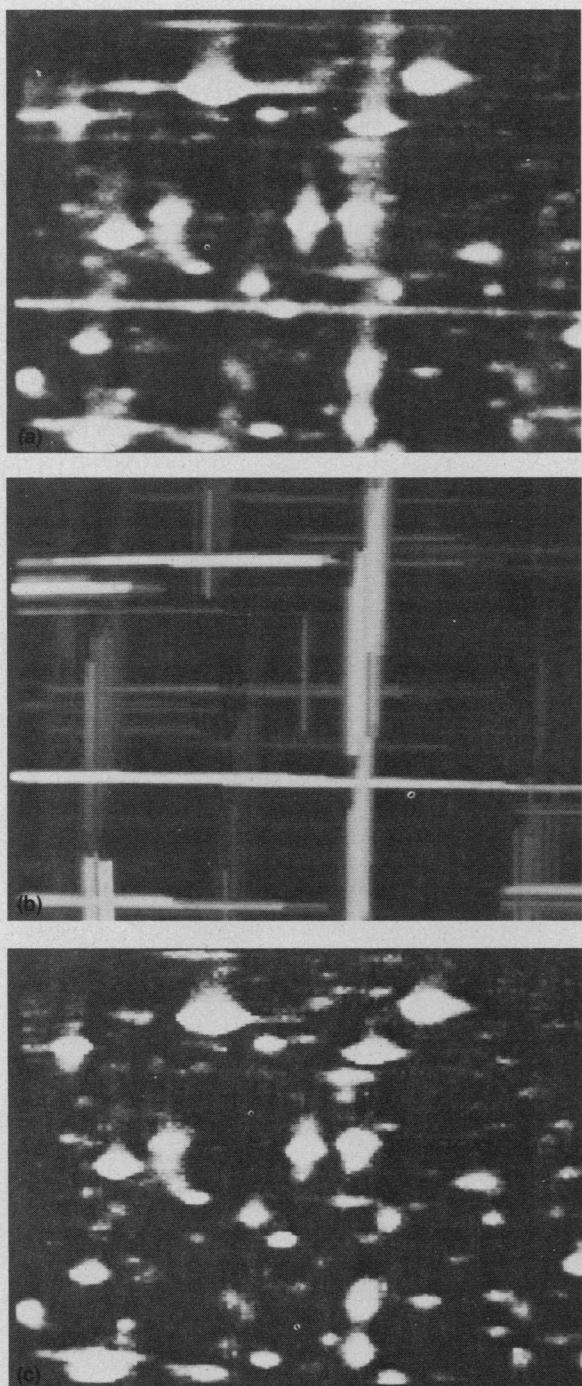


Figure 12. Streak removal. The streaks in the background-normalized gel section (a) are extracted by the opening operation (b) using rolling-pin-like structuring elements. The streakless result (c) is now ready for spot detection and trio matching.

are clearly less than those of their neighboring pixels. Finally, the second label iteratively propagates itself over any pixel, unlabeled or labeled, that is not uphill from the label. This last step changes all pixels marked by the first label that are not local maxima to the second label, leaving only the pixels marked by the first label as the true local maxima.

Spot domains. Occasionally, a pair of spots overlap in the gel image such that one of the spots does not possess a distinct local maximum. The elongated regions at the center of the pseudocolored, background-normalized gel images of Figure 13 are examples of such spots. The pseudocolored, multithresholded images of Figure 14 further illustrate spots (marked by arrows) that do not have local maxima yet must be ultimately matched to spots with local maxima. To resolve this potential difficulty, programs have been developed to locate spot domains on the basis of their shapes. These programs segment overlapping spots at regions of elongation or narrowing using a morphological iterative technique known as the watershed algorithm (Lantuejoul¹³). The result of watershed segmentation is shown in Figure 15.

Gel matching. The final step in the automated processing of the father, mother, and child, or FMC, trio is gel matching. Recall that a candidate mutant protein will be represented by a spot appearing in a child's gel that does not have a corresponding spot in the same location in either of the parent's gels. Candidate mutant proteins will be variants of normal proteins, in which a random insertion or deletion of an amino acid or amino acid chain yields a protein with a modified molecular weight, or more likely, a modified molecular charge. We expect to see mutant proteins as spots shifted from their normal positions by several millimeters, depending on where they occur in the gel.

Gels cannot be compared by direct superposition. Furthermore, global stretching cannot be applied to the gel images to align all similar protein spots because the gels exhibit strong local nonlinearities that are due to inhomogeneities in their manufacture. Humans can compare gels by locating a given spot within a local constellation of spots and searching accompanying gels first for the constellation and then for the given spot within the constellation. A computer compares gels by a similar procedure implemented as a graph matching program.

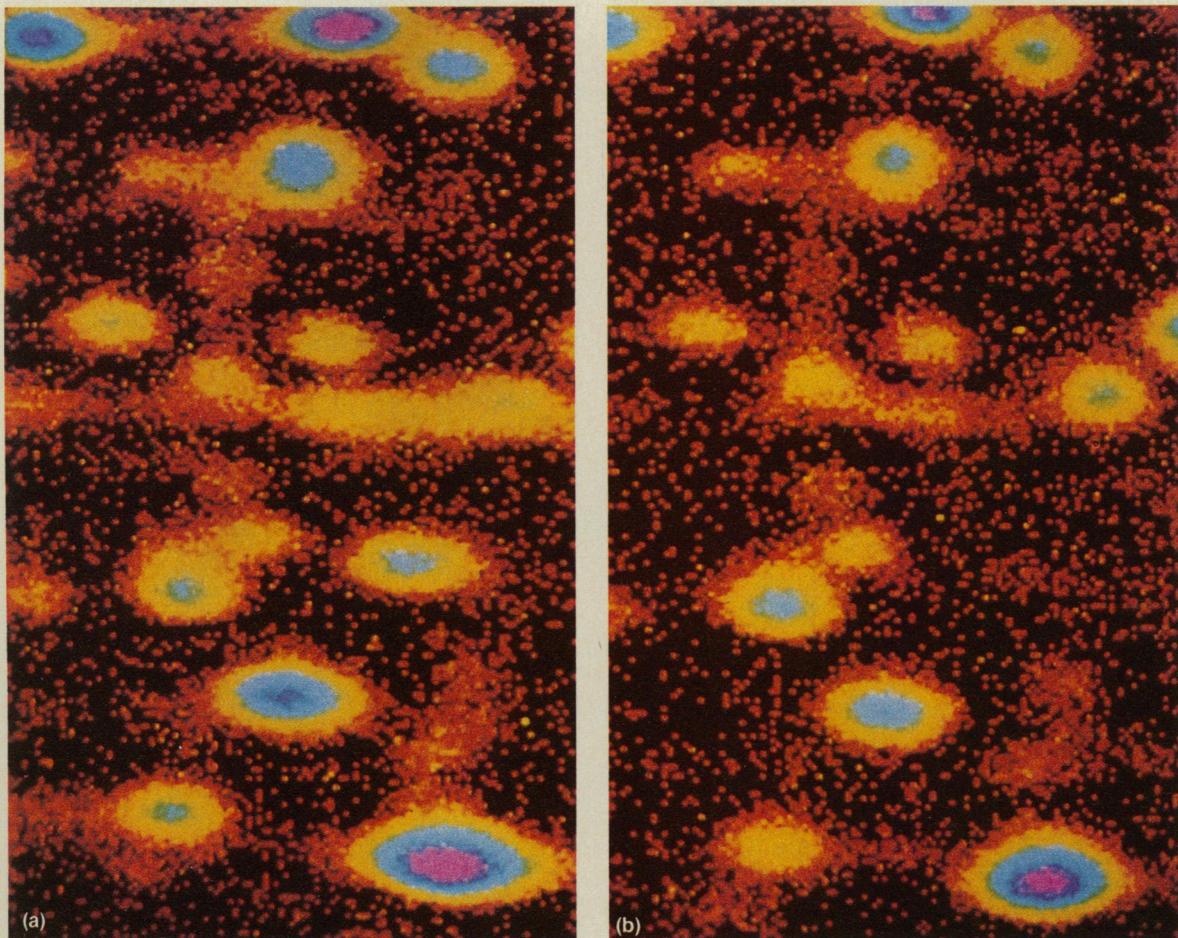


Figure 13. Pseudocolor composite of background-normalized gel autoradiograms. Autoradiograms are noisier than silver-stained gels and require greater care in processing.

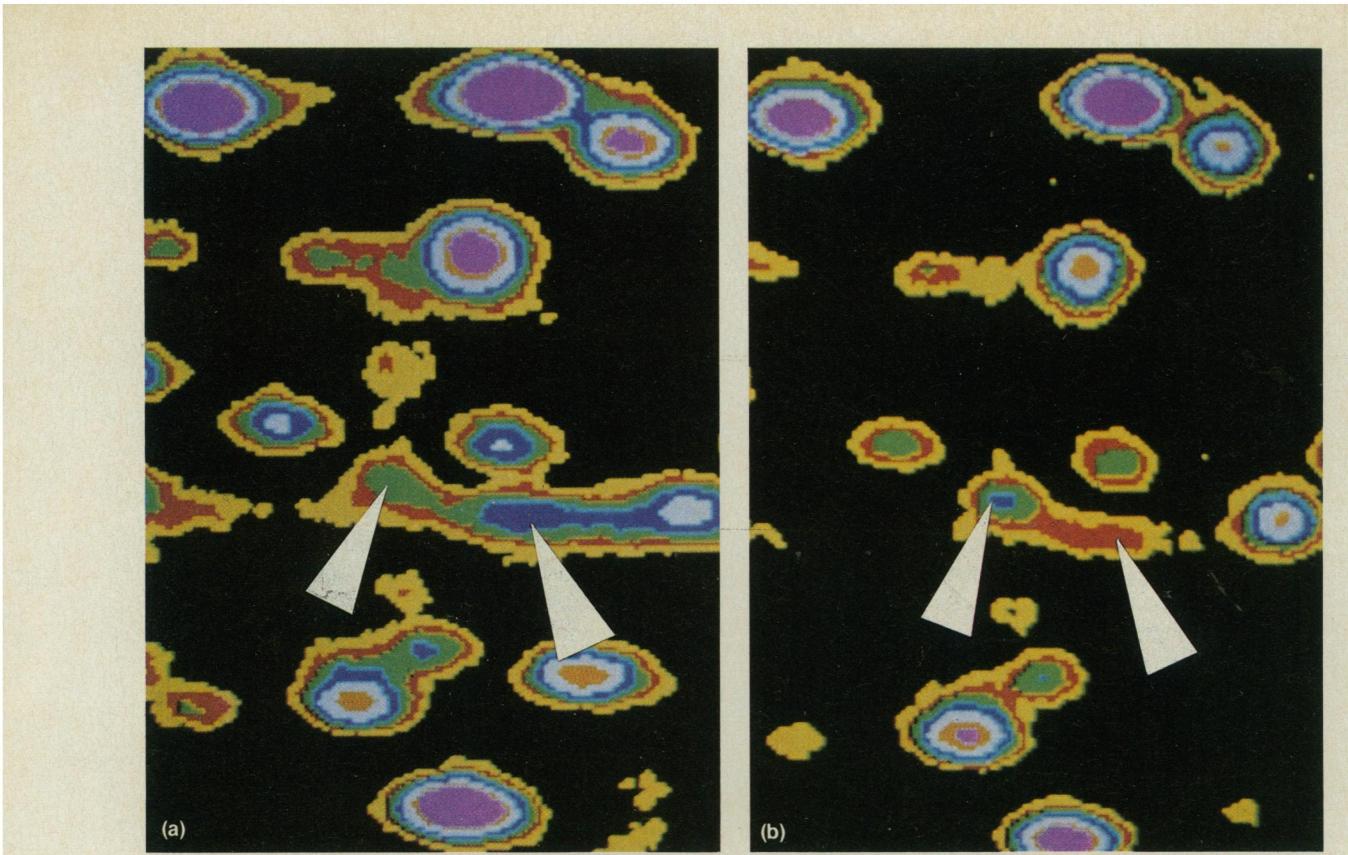


Figure 14. Pseudocolored, multithresholded composite from Figure 13. Arrows mark comparable spots in both gels, but not all spots are distinct. Shape recognition processing separates spots into distinct regions (Figure 15).

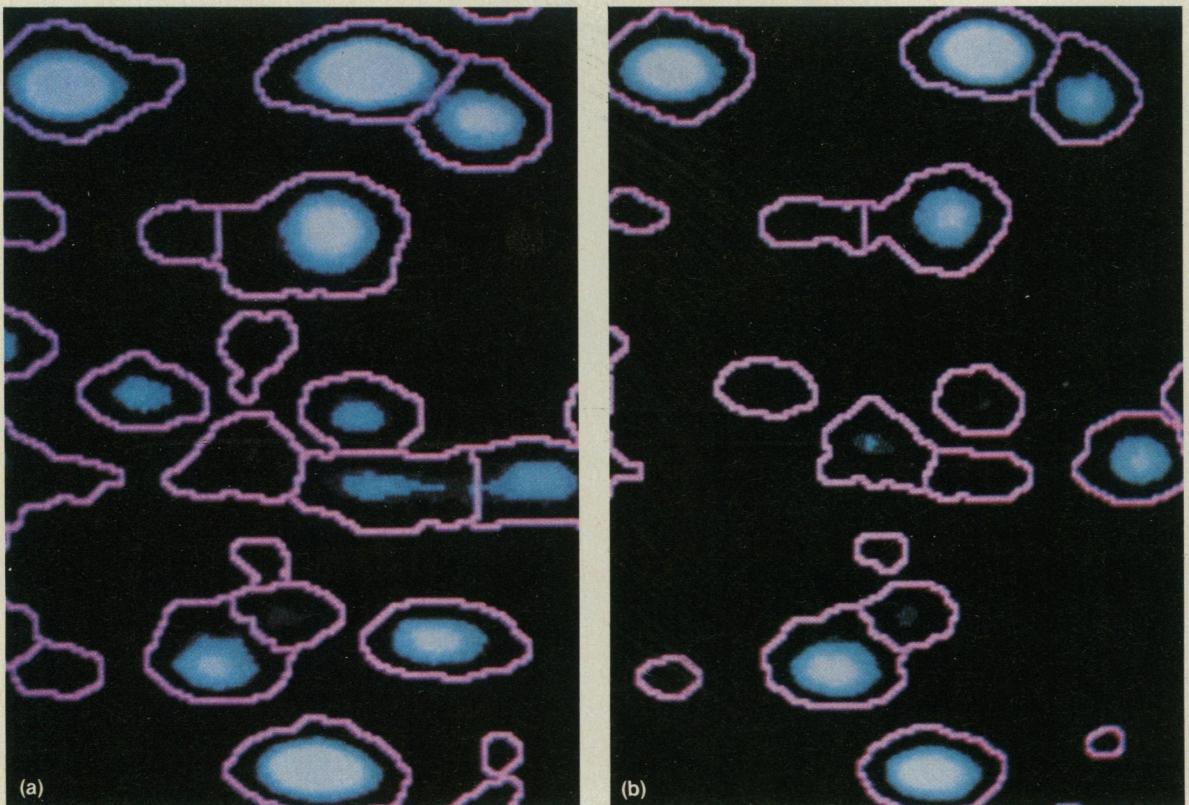


Figure 15. Extracted spot domains of the gel autoradiograph composite of Figure 13.

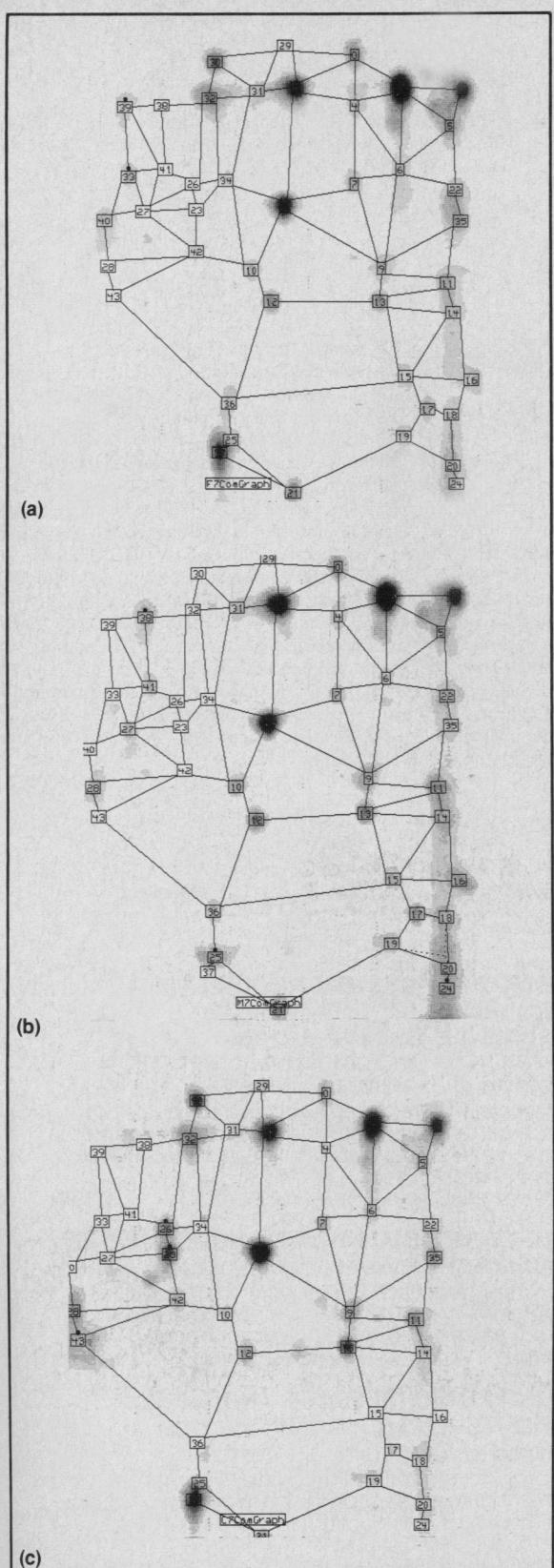


Figure 16. Matched father (a), mother (b), and child (c) trio after seven iterations of the Skolnick graph matching algorithm. Matching spots are given the same spot number. Virtual spots are created where no matching spot is found. Spots belonging only to one family member are marked with dots.

The graph matching program, which was developed by Skolnick,¹⁴ does not depend on any a priori knowledge of the spot spatial distribution. The program constructs a graph, the Gabriel graph, described by Toussaint,¹⁵ by using as nodes any spot cues that exceed a given intensity threshold. The graphs are then compared to determine an initial subset of spots common to all three. From this subset of common spots the program then determines whether the remaining graph differences are real or the result of quantitative variations caused by spots falling below the threshold. Through continued iteration the program resolves unmatched spots by spatially relating them via their connecting graph edges to previously matched spots until all graph differences from spot intensity variations are removed. The graph matching program does not run on the cytocomputer but on its host computer, a VAX 11/780. Figure 16 illustrates the Gabriel graphs of an FMC trio after seven iterations of Skolnick's algorithm. All graph differences have been resolved, and family protein matches can be listed.

Biomedical application has been the vehicle through which I have examined algebraic languages and computer architectures for image processing. I stress that the language and architecture presented here are not abstract constructs, but well-conceived solutions to practical problems that arise during biomedical applications of new computer technologies. Language, architecture, and application should not be segmented for individual study, however, for it is the parallel learning process that ultimately brings about a full understanding of computers and computer languages—and parallelism is after all what image processing is all about. ■

Acknowledgments

I thank the Environmental Research Institute of Michigan for generously providing the cytocomputer image processor that has been so valuable for biomedical studies, and the Computer and Image Processing Research Network for their extensive hardware and software support of the mutation study. Autoradiogram digital image data are from Argonne National Laboratories (courtesy of N. and L. Anderson). This work was supported in part by the National Cancer Institute through project 5-PO1-CA-26803-02.

References

1. J. V. Neel et al., "Search for Mutation Affecting Protein Structure in Children of Atomic Bomb Survivors: Preliminary Report," *Proc. Nat'l Academy Science*, Vol. 77, 1980, pp. 4221-4225.
2. J. V. Neel, H. W. Mohrenweiser, and M. M. Meisler, "Rate of Spontaneous Mutation at Human Loci Encoding Protein Structure," *Proc. Nat'l Academy Science*, Vol. 77, 1980, pp. 6037-6041.

3. O'Farrell, "High-resolution Two-dimensional Electrophoresis of Proteins," *J. Biology and Chemistry*, Vol. 250, 1980, pp. 4007-4021.
4. N. Anderson and L. Anderson, *Analytical Biochemistry*, Vol. 85, 1978, p. 33.
5. M. Skolnick, S. Sternberg, and J. V. Neel, "Some Algorithms for Adapting 2-D Gels to the Study of Mutation," *Clinical Chemistry*, Vol. 28, No. 4, Apr. 1982, pp. 969-978.
6. P. Danielsson and S. Levialdi, "Computer Architectures for Pictorial Information Systems," *Computer*, Vol. 14, No. 11, Nov. 1981, pp. 53-67.
7. S. R. Sternberg, "Parallel Architectures for Image Processing," *Proc. Compsac 79*.
8. H. Minkowski, *Volumen und Oberfläche*, *Mathematics Annual*, Vol. 57, 1903, pp. 447-495.
9. G. Matheron, *Random Sets and Integral Geometry*, John Wiley & Sons, New York, 1975.
10. J. P. Serra, *Mathematical Morphology and Image Analysis*, Academic Press, London, 1981.
11. J. C. Klein and J. Serra, "The Texture Analyser," *J. Microscopy*, Vol. 95, No. 2, 1972, pp. 349-356.
12. S. R. Sternberg, "Cellular Computers and Biomedical Image Processing," *Proc. US-France Seminar Biomedical Image Processing*, Springer-Verlag, New York, 1982.
13. C. Lantuejoul, "Skeletonization in Quantitative Metallography," *Issues in Digital Image Processing*, R. M. Haralick and J. C. Simon, eds., Sijhoff and Noordhoff, 1980.
14. M. Skolnick, "An Approach to Completely Automatic Comparison of Two-Dimensional Electrophoresis Gels," *Clinical Chemistry*, Vol. 28, No. 4, Apr. 1982, pp. 979-986.
15. G. T. Toussaint, "Pattern Recognition and Geometrical Complexity," *Proc. Fifth Int'l Conf. Pattern Recognition*, 1980, pp. 1324-1347.



Stanley R. Sternberg is president of Cyto-Systems Corporation in Ann Arbor, Michigan, and adjunct associate professor of electrical and computer engineering at the University of Michigan. From 1974 to 1981, he was senior research engineer at the Environmental Research Institute of Michigan, formerly the Willow Run Research Laboratories. There he created the cyto-computer and directed its development.

In 1979, he cofounded the Computer and Image Processing Research Network at the University of Michigan, a campus-wide image processing resource for use in the life sciences and as a testbed for engineers to design and evaluate new computers and distributed computer systems.

Sternberg received his PhD in industrial and operations engineering from the University of Michigan in 1971. He is a member of Tau Beta Phi, Eta Kappa Nu, Sigma Xi, and the Society of Manufacturing Engineers.

The Quality Choice in Endicott, NY!

Heading for Endicott on business? We have a Quality location. Just one block from IBM. 3 miles from GE. And 15 minutes from Singer-Link.

The Quality Inn Lodge offers great flexibility. Rooms for overnight. Or apartments for long term.

And if you're planning a meeting or banquet, the Quality Inn Lodge has 4 rooms that can accommodate from 25 to 800. Plus 6 rooms that handle 12 persons each.



Quality Inn Lodge
One Lodge Blvd.
Endicott, NY 13760

For reservations call 607-754-7570
or toll-free:

800-228-5151

The Quality Choice.



Reader Service Number 3

LexingtonBooks

COMPUTER-SYSTEM REQUIREMENTS

Techniques and Examples

Kenneth J. Thurber, Architecture Technology Corporation, and **Peter C. Patton**, University of Minnesota

Presents an integrated approach to the use of both hardware and software, including problem statements, specifications, linear tradeoff analysis, and graph theory.
128pp. ISBN 0-669-02958-0 \$15.95

SOFTWARE BLUEPRINT AND EXAMPLES

Yaohan Chu, University of Maryland

This new approach to software design applies engineering methodology for more efficient program development and productivity.
544pp. ISBN 0-669-05329-5 \$39.95

ELECTRONIC MESSAGE TRANSFER AND ITS IMPLICATIONS

Alfred M. Lee, Cornell University

Topics include technological developments, potential markets, possibilities for conservation of resources, and liability and privacy implications.
ISBN 0-669-05555-7 January 1983



LexingtonBooks
D. C. Heath and Company
125 Spring Street
Lexington, MA 02173
(617) 862-6650 (212) 924-6460

Call our toll-free number
800 428-8071

Reader Service Number 4