SEMIANNUAL TECHNICAL REPORT

Covering Research Activity During the Period
1 October 1976 through 31 March 1977

Harry C. Andrews
Project Director
(213) 746-5514

Image Processing Institute
University of Southern California
University Park
Los Angeles, California 90007

31 March 1977

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract F-33615-76-C-1203, ARPA Order No. 3119.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Image Processing Institute
University of Southern California, University Park
Los Angeles, California 90007

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

IMAGE PROCESSING RESEARCH

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Semiannual Technical, 1 October 1976 through 31 March 1977

5. AUTHOR(S) (First name, middle initial, last name)

Harry C. Andrews (Project Director)

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
31 March 1977	200	42

8a. CONTRACT OR GRANT NO.

F-33615-76-C-1203

b. PROJECT NO.

ARPA Order No. 3119

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

USCIPI Report 740

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for release; distribution unlimited

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

13. ABSTRACT

This technical report summarizes the image understanding and image processing research activities performed by the Image Processing Institute at the University of Southern California during the period of 1 October 1976 to March 31, 1977 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency Information Processing Techniques Office.

The research program has as its primary purpose the development of techniques and systems for processing, transmitting, and analyzing images and two dimensional data arrays. Six tasks are reported: Image Understanding Projects; Image Processing Projects; Smart Sensor Projects; Institute Facilities Projects; Recent Ph.D. Dissertations and Recent Institute Personnel Publications. The Image Understanding project includes quite encouraging results on automatic scene segmentation by clustering using mathematical nonsupervised pattern recognition proced .es. In addition image synthesis, edge and boundary extension, symbolic chang · analysis and quantitative edge detector parameters are discussed. The image processing projects have concentrated on variable knot adaptive spline placement, image filtering and the psychovisual model, blind phase a posteriori restoration and digitally generated optical filters for image reconstruction. The smart sensor project covers simulations of adaptive 3 x 3 kernels, test equipment for the Sobel and adaptive CCD chips, and development plans for automatic segmentation implemented in CCD chip configuration. The Institute facilities section surveys the current USCIPI hardware software configuration while recent Ph.D. dissertations are discussed in

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
the following section, the report concluding with listings of recent publications. ********** 14. Key Words: Digital Image Processing, Image Restoration, Degrees of Freedom, Scene Analysis, Image Understanding, Edge Detection, Image Segmentation, CCD Arrays, CCD Processors.						

ABSTRACT

This technical report summarizes the image understanding and image processing research activities performed by the Image Processing Institute at the University of Southern California during the period of 1 October 1976 to 31 March 1977 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency Information Processing Techniques Office.

The research program has as its primary purpose the development of techniques and systems for processing, transmitting, and analyzing images and two-dimensional data arrays. Six tasks are reported: Image Understanding Projects; Image Processing Projects; Smart Sensor Projects; Institute Facilities Projects; Recent Ph.D. Dissertations; and Recent Institute Personnel Publications. The Image Understanding project includes quite encouraging results on automatic scene segmentation by clustering using mathematical nonsupervised pattern recognition procedures. In addition image synthesis, edge and boundary extension, symbolic change analysis and quantitative edge detector parameters are discussed. The image processing projects have concentrated on variable knot adaptive spline placement, image filtering and the psychovisual model, blind phase a posteriori restoration and digitally generated optical filters for image reconstruction. The smart sensor project covers simulations of adaptive 3 x 3 kernels, test equipment for the Sobel and adaptive CCD chips, and development plans for automatic segmentation implemented in CCD chip configuration. The Institute facilities section surveys the current USCIPI hardware software configuration while recent Ph.D. dissertations are discussed in the following section, the report concluding with listings of recent publications.

PROJECT PARTICIPANTS

Project Director	Affiliation
Harry C. Andrews	Computer Science & Electrical Engineering

Research Staff	Affiliation
Nasser E. Nahi	Electrical Engineering
Ramakant Nevatia	Computer Science
William K. Pratt	Electrical Engineering
Keith E. Price	Image Processing Institute
Erica M. Rounds	Computer Science
Alexander A. Sawchuk	Electrical Engineering
Timothy C. Strand	Image Processing Institute

Support Staff

Rosemarie Bell	Clay Olmstead
Marilyn Chan	Michael Patton
Gary Edwards	David Peck
Gregg Finn	James Pepin
Eva Huang	Hector Saucedos
Scott Johnston	Ray Schmidt
Eileen Jurak	Joyce Seguy
Ed Kasanjian	Dennis Smith
Toyone Mayeda	Pat Stoliker
David Nagai	Teh-Ling Sung
Shankar Narayan	Amy Yiu

Students

Ikram Abdou
Ahmad Armand
Chung-Ching Chen
Peter Chuan
Guy Coleman
Charles Hall
Chung-Kai Hsueh
Bijan Lashgari
Kenneth Laws
Simon Lopez-Mora
Lee Martin
Dennis McCaughey
John Morton
Firouz Naderi
Javad Peyrovian
Jin Soh

TABLE OF CONTENTS

Page

1. Research Overview

This document represents the third semi-annual report funded under the current ARPA Image Understanding contract and, as such, presents a certain amount of momentum and progress toward the goals originally undertaken a year and a half ago. I feel confident in stating that we clearly understand the Image Understanding problems in considerably greater depth. I also feel confident that we have made progress in the specific areas of quantitative scene segmentation by clustering, quantitative edge detection and evaluation, and (naturally with the arrival of Dr. Keith Price) have gained a good step toward general symbolic manipulation for the higher levels of many Image Understanding tasks.

Naturally we have also progressed on the traditional front of our expertise, that of Image Processing. The past six months have seen breakthroughs in the areas of variable sampling procedures for image approximations, advances in the a posteriori restoration problem as well as object detection in noisy images. Optical filters for image reconstruction have been designed and the foundations for research in the psychophysical characteristics of the human visual system have been laid.

On the "smart sensor" front considerable effort has been expended in two areas by USC personnel, that of 3 x 3 kernel definition for future sensor implementation, and the degrees of a real time CCD implementation of an on-board image segmentor. Both these projects represent study efforts for future designs. Naturally Hughes Research Laboratory personnel have also been progressing in the development of test circuitry for the CCD chips under fabrication, and it appears that as of this printing, the

Sobel chip is in production and is currently available for testing.

This semi-annual report also includes an overview of the current USCIPI laboratory configuration, numerous modifications having been implemented over the past two years. Finally a report of recent Institute Ph.D. dissertations are included as well as the listing of recent Institute personnel publications in the open literature.

2. Image Understanding Projects

This section presents recent results in the research area of Image Understanding. Progress has been achieved in the area of quantifying edge detector parameters by pattern recognition techniques as well as in edge elongation both in monochrome and color scenes. In addition to the above, higher level processes both in symbolic change detection and synthesis of adjacent regions are described. Finally considerable progress has been experienced in the area of automatic scene segmentation from signal processing (bottom up) procedures. The preliminary success of this algorithm is quite encouraging as it utilizes completely unsupervised pattern recognition clustering, feature selection, and cluster optimization techniques without the need for top-down or external guidance. The algorithm is based upon the inherent homogeneity concept of image segments but measured in N-dimensional vector space.

2.1 Scene Segmentation by Clustering

Guy Coleman

This effort is directed towards a method of automatically segmenting imagery. The method so far developed is autonomous and reasonably fast. A very general block diagram and some preliminary results are shown in figures 1 through 8.

One image shown is a 256 x 256, eight bit monochrome image. Features such as brightness and texture are computed at every pixel location in the scene. The output of the feature computation is a 225 x 225 map of vectors where the components of the vectors are the values of the features at the appropriate points in the scene and the size reduction is due to window effects at the scene edges. The next step is to perform a multi-dimensional (Karhunen-Loeve) rotation of the data such that the new features are a linear combination of the old features, but are statistically uncorrelated. This step is performed so that undesirable features may be discarded and the number of desirable features retained will be the minimum necessary. In other words, the decorrelation prevents retention of several good but highly correlated features.

A preliminary clustering is performed to evaluate the features for their usefulness in segmenting the scene. The evaluation is based on the pairwise average Bhattacharyya distance. Those features which are least useful are discarded and the clustering is performed again.

The clustering algorithm is performed for 2,3,4,... clusters. At each number of clusters, the product of the between and within cluster scatter average is computed. The

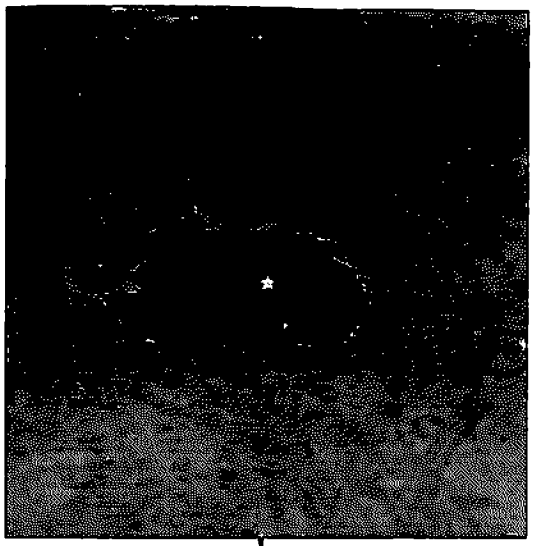
algorithm is stopped when this product reaches a maximum. The number of clusters and the cluster means are forwarded to the segmentation phase of the algorithm and the image is segmented.

Some preliminary results of this algorithm are shown on the following pages. The segmentations have been subjected to pseudo-coloring to improve the visibility of the different segments.

The first set of pictures resulted from using several variations of the basic procedures on an armored personnel carrier (APC). The first set of APC pictures, called "12 Non-Reduced Correlated Features" is the result of clustering the 12 original features. These features are considered very preliminary and were used to permit development of the clustering algorithm and to verify the ability of the algorithm to reject poor features. The algorithm rejected eight of the 12 features based on the pairwise average Bhattacharyya distance evaluated at the picture labeled "Best Number of Regions."

The data was reclustered, producing the second set of pictures labeled "4 Reduced Correlated Features." The picture labeled "Best Number of Regions" on the page labeled "4 Reduced Correlated Features" is the end product of the algorithm, having separated the vehicle from the background. The bushes in the top of the scene represent errors, that is, they were classified as being the same as the vehicle.

The next series of APC pictures is labeled "12 Non-Reduced Decorrelated Features." These images are the result of clustering the 12 features produced by the multi-dimensional (Karhunen-Loeve) rotation of the 12 original features. Except for the pseudo-color effects,



FEATURE COMPUTATION

potential focal plane implementation features are brightness, edges, texture, etc.

DECORRELATE REDUNDANT FEATURES

makes features independent (can be optional)

FEATURE REDUCTION

removes optional features not contributing to good clustering (can be optional)

CLUSTERING

finds inherent homogeneous regions of the image automatically

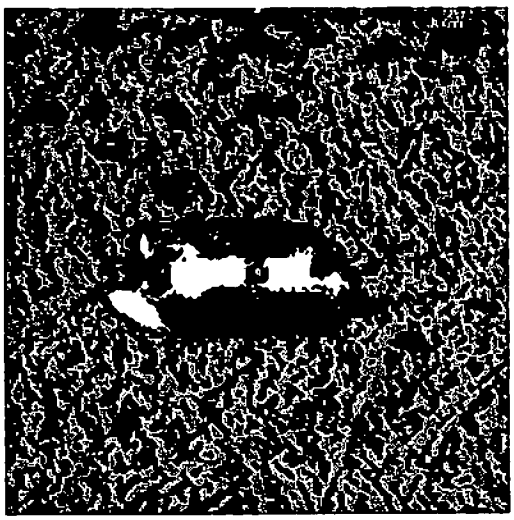
IDENTIFY BEST # OF CLUSTERS TO STOP SEGMENTATION

SEGMENTATION

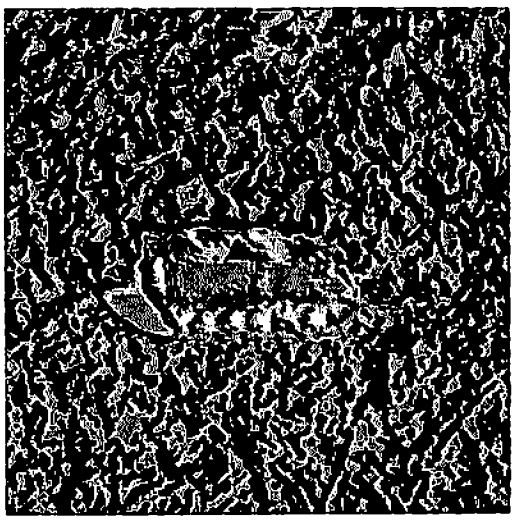
outputs regions of common homogeneity to a constant color

(see following pages)

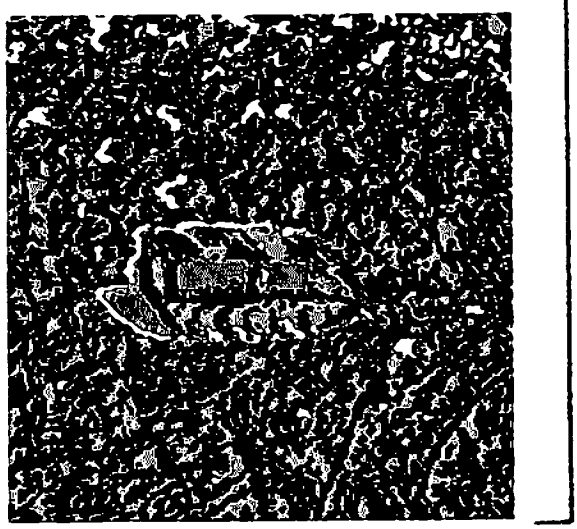
FIGURE 1.



7 REGIONS



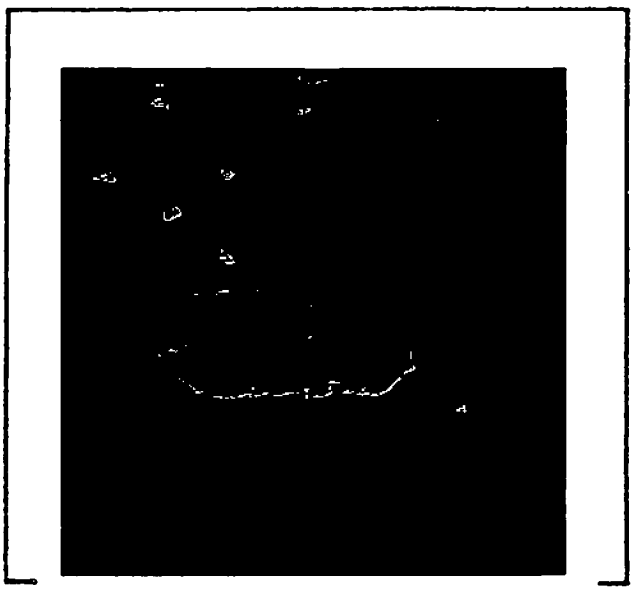
8 REGIONS



9 REGIONS
(BEST NUMBER OF
REGIONS)

10 REGIONS

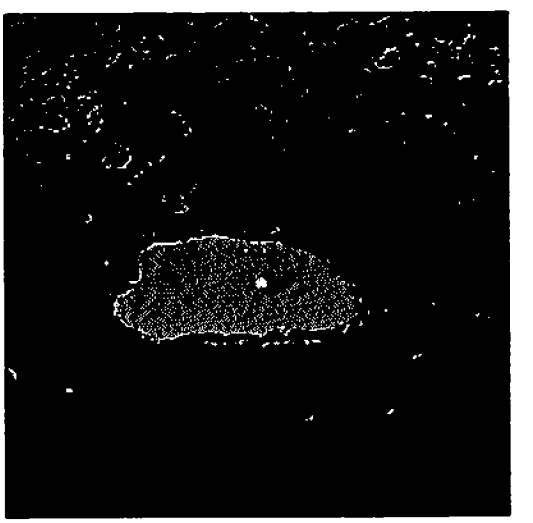
FIGURE 2. 12 NON REDUCED CORRELATED FEATURES



2 REGIONS
(BEST NUMBER
OF REGIONS)

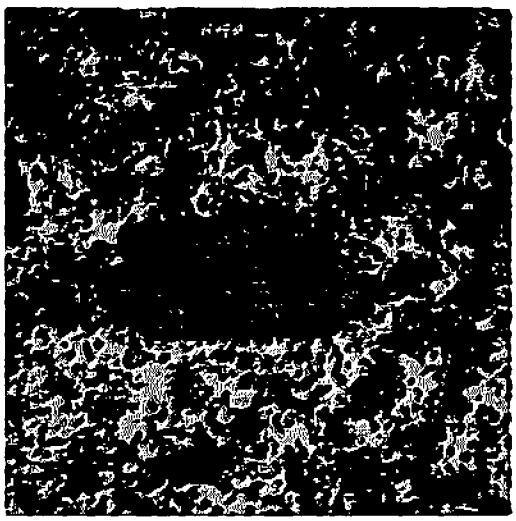
3 REGIONS

4 REGIONS

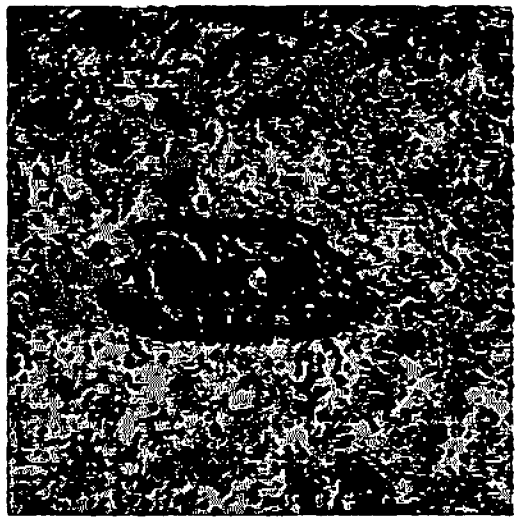


5 REGIONS

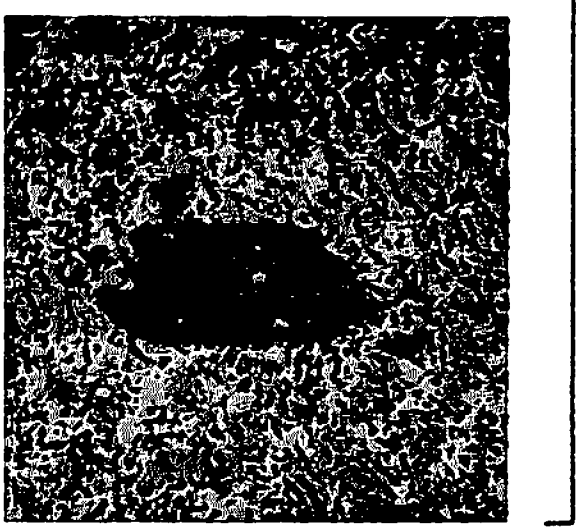
FIGURE 3. 4 REDUCED CORRELATED FEATURES



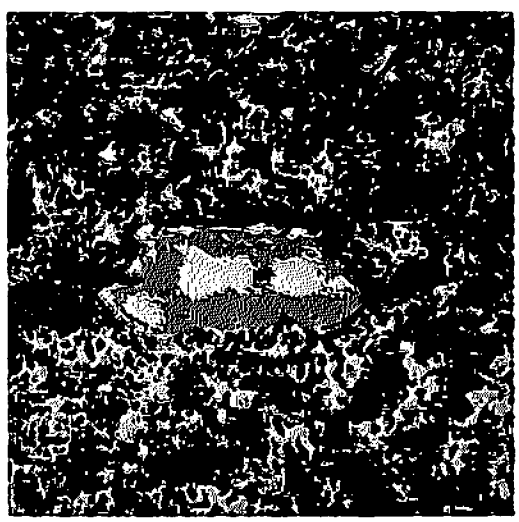
6 REGIONS



7 REGIONS

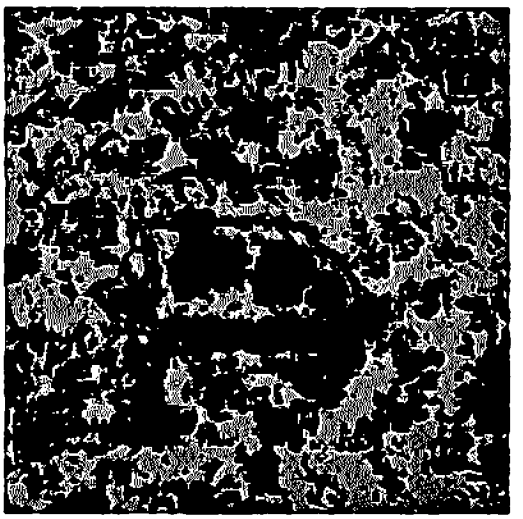


8 REGIONS
(BEST NUMBER
OF REGIONS)

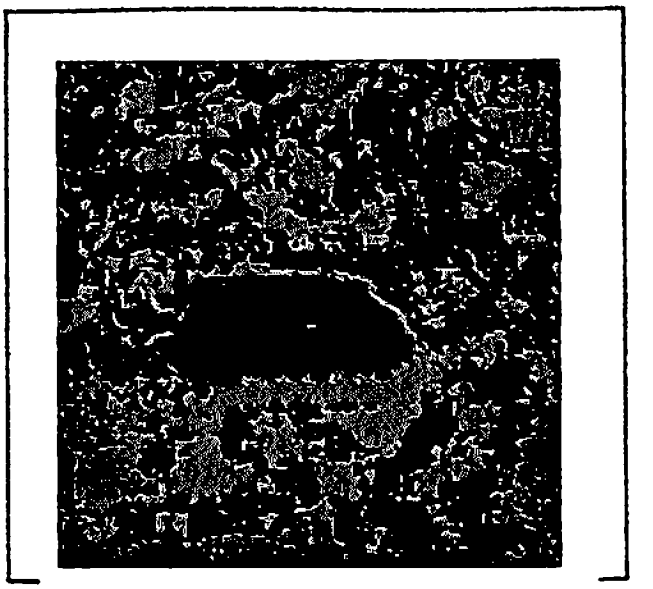


9 REGIONS

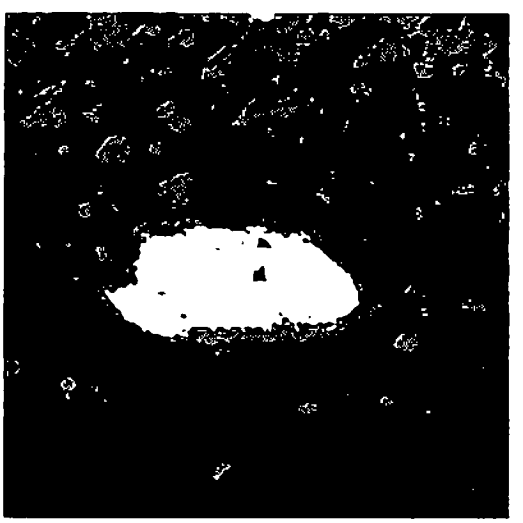
FIGURE 4. 12 NON REDUCED DECORRELATED FEATURES



2 REGIONS



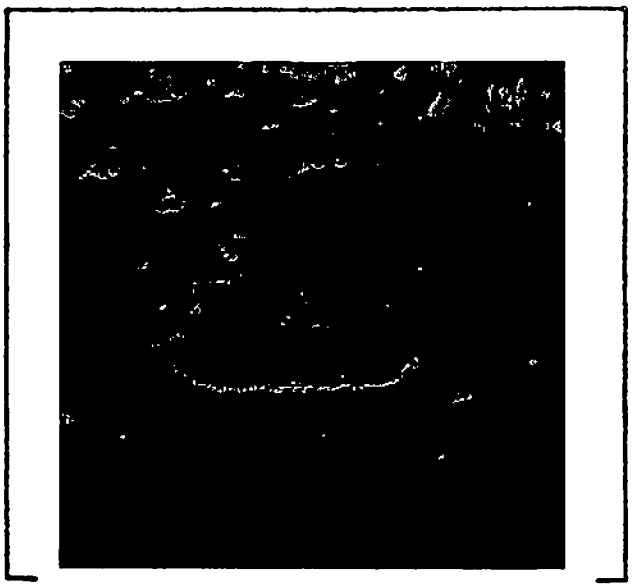
3 REGIONS
(BEST NUMBER
OF REGIONS)



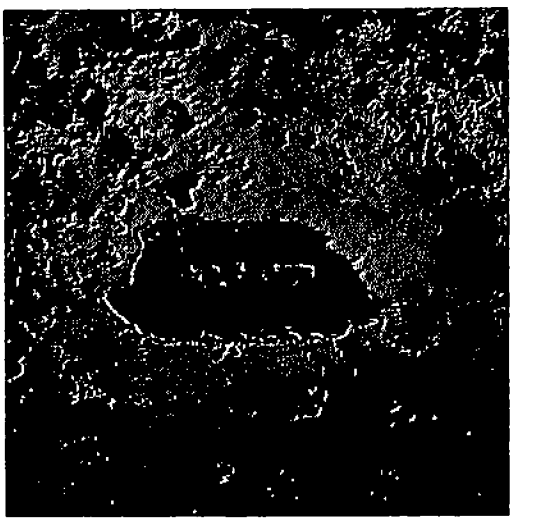
4 REGIONS

5 REGIONS

FIGURE 5. 4 REDUCED DECORRELATED FEATURES

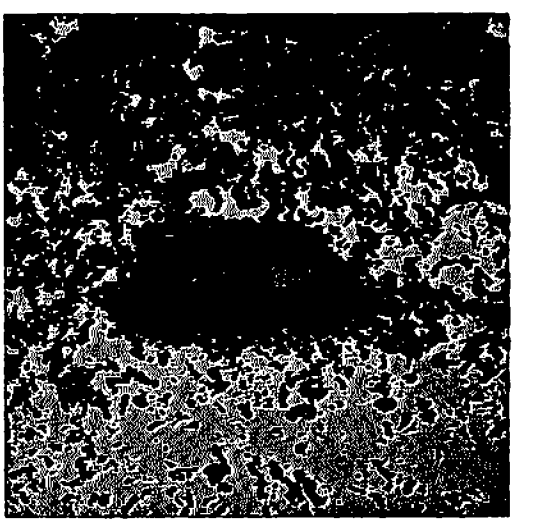


2 REGIONS
(BEST NUMBER
OF REGIONS)



3 REGIONS

4 REGIONS



5 REGIONS

FIGURE 6. SINGLE BEST DECORRELATED FEATURE

these images appear quite similar to the "12 Non-Reduced Correlated Features." This is so because rotation of the coordinate axes should not affect clustering.

The pairwise average Bhattacharyya distances for the rotated features were evaluated at the best number of regions (eight in this case) and clustering was performed on the above average features, in this case four. The results of this procedure are shown in the series of images titled "4 Reduced Decorrelated Features." The final result is shown in the image titled "Best Number of Regions," in this case three regions.

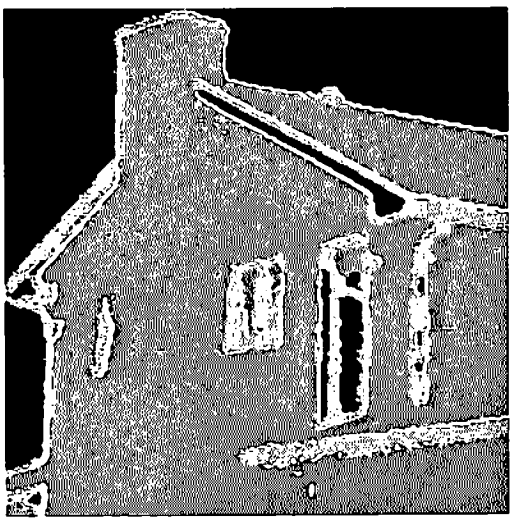
The pairwise average Bhattacharyya distances for the 12 rotated features were such that the average for one feature was substantially higher than any of the others. Accordingly, this feature alone was used to perform clustering. The results of this are shown in the final series of images titled "Single Best Decorrelated Feature." The best number of regions was two in this case. It can be observed that more errors were made in this segmentation than in previous ones due to the enormous reduction in dimension that has taken place.

The second series of pictures is the result of segmenting a color picture of a house. The features used are derived from the red, green and blue color planes of the image and there are a total of 15 (five per color plane). The first picture (two segments) was decided to be the best segmentation based on all 15 features. The additional segmentations are the result of permitting the algorithm to continue segmenting beyond the best number of clusters.

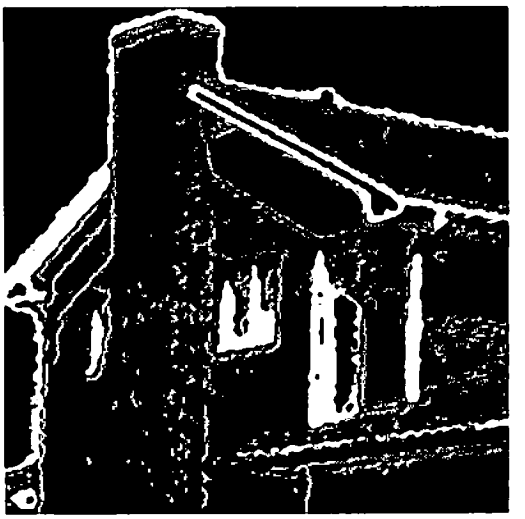
With suitable modification, the algorithm described above can be implemented at real time video rates. A more

(a) House Original

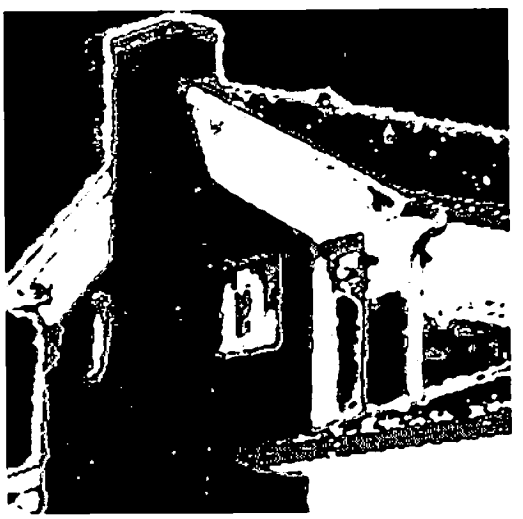
(b) 2 Regions (Best Number of Regions)



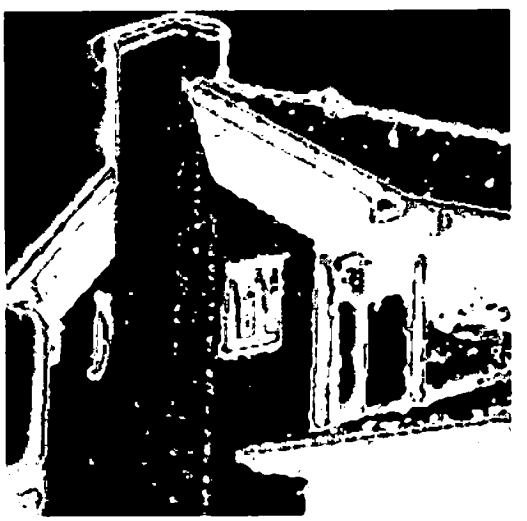
(c) 3 Regions



(d) 4 Regions



(e) 5 Regions



(f) 6 Regions

Figure 7. Segmentation of House Picture.

(g) 7 Regions

(h) 8 Regions

(i) 9 Regions

(j) 10 Regions

(k) 11 Regions

(l) 12 Regions

Figure 8. Segmentation of House Picture.

complete description of this concept is contained elsewhere in this report (see Section 4.2).

2.2 Symbolic Change Analysis

Keith Price

Recent work in image understanding has shown that symbolic techniques can be applied to a large class of images with a variety of change analysis tasks [1]. The system to perform this analysis is now operational at USC.

The symbolic change analysis system processes pairs of images of the same scene to produce a symbolic description of what regions in the scene have remained the same, what regions have changed between the views, and how these regions have changed. The analysis procedures work only with the symbolic description of the two images. A symbolic description of an image is composed of: the basic segments of an image which can be generated by machine processing, by human processing or a combination of both; a set of features of each of the basic regions, features such as size, location, neighboring regions, color, etc. The results of the analysis include: an indication of which segments in the two images correspond to the same object in the scene, an indication of global differences between the two images, an indication of changes in corresponding segments, and an indication of new objects in the scene.

The first operation, the location of corresponding regions in the two images, known as symbolic registration, is performed by comparing the features of a region in the first image with all regions in the second image to find which of the regions is most similar. The basic region to region comparison procedure produces a numeric rating for

the match between the two regions, the region with the best rating is the most similar. This rating is a function of the numeric ratings for matches with each individual feature.

Since this is a general system, some outside information about each set of images is necessary. This information includes a description of the task which is to be performed and any knowledge about the images which may be necessary to perform the analysis. In many cases this information is about global changes between the two views. For example, the information could include: there is a scale change between images, there is an orientation difference, the second image is brighter, etc. In this case, the global changes will affect the matching of regions in the two images since these global changes will alter the value of one or more of the features which are used. If the exact change is known such as: the second image is twice the scale of the first, then the feature values can be adjusted before a match is attempted. But, the magnitude of the change is not usually given, just that such a change may occur. The actual change amount can be computed from changes found in a fewer pairs of corresponding regions. Even with such adjustments some features will still have differences caused by global, or expected changes. Therefore, in the function which combines differences in all the features to produce the final region to region rating, the features which are expected to remain constant are given more strength than the features which may change. For example, if the task is to report on the changes in color of stationary regions in an image (e.g. crop analysis) then position, relative positions and size of shape would be expected to remain constant, but color should change.

This shows one example of the use of change results within the change analysis system, these same change results might be part of the ultimate task given to the system, or just a necessary by-product. Changes in features of the corresponding regions are derived directly from the symbolic registration results. Other change analysis can require additional processing: find the change in the number of ships in a pair of views of a dock area - the additional processing is required to determine which segments correspond to ships.

This has been a discussion of the current state of the art in the symbolic change analysis area. Work will continue in the areas of: additional use of knowledge in matching, additional task domains, the actual matching function, the use of the change results, change analysis in sequences of images, and the use of these techniques in more general image understanding systems.

Reference

[1] K. Price, "Change Detection and Analysis in Multi-Spectral Images," Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania, December 1976.

2.3 Synthesis of Adjacent Regions

Erica M. Rounds

I. Introduction

A digitized image can be partitioned into regions by either identifying all picture cells belonging to a given

region or by characterizing regions in terms of their closed boundaries. Depending on the analysis to be performed at later stages one or the other representation may be more desirable. Boundary coordinates typically require less storage and are more suitable for syntactic methods and for line and curve analysis. On the other hand, some shape analysis methods such as the medial axis transformation [1] operate on the interior points of a figure.

This paper describes an algorithm for reconstructing a digital image given the boundary lists of the regions contained in the image. Permissible topological relations between regions are adjacency and containment. Interior points are assigned to regions on the basis of a small set of "boundary types." These encode the shape of a contour segment connecting three adjacent vertices. The algorithm processes all regions together so that space and time requirements are minimized.

In [2] a method was presented for generating a single figure (possibly with holes) from its contour. The present paper extends this method to the general case of several adjacent image regions. For ease of reference, the basic algorithm is briefly described in the next section. In Section III the necessary modifications are introduced to handle common boundary points between adjoining regions.

In the following it is assumed that each region is described by a set of x and y coordinates representing a counterclockwise traversal of its contour. More precisely, the closed boundary for a simple region (no holes) is defined to be a set of N ordered grid points $\{(x_k,y_k)\}$, such that (x_{k-1},y_{k-1}) is adjacent to (x_k,y_k), $2 \leq k \leq N$, and $(x_N,y_N) = (x_1,y_1)$. Two points are said to be adjacent if they are one grid cell apart in a horizontal, vertical, or

diagonal direction (eight-neighborhood adjacency).

II. Basic Algorithm

The algorithm consists of two steps. During the first step, each boundary point (x_k,y_k) is assigned a type T which is stored in a corresponding element (i,j) of a picture matrix P (with dimension $(y_{max}-y_{min}+1)$ x $(x_{max}-x_{min}+1)$). Type T for (x_k,y_k) is determined by the relative positions of adjacent points in a 3 x 3 neighborhood centered on (x_k,y_k). T is computed from a vector $z = (z_1,z_2,z_3)^t$, where z_1 and z_3 are the number of neighbors above and below the reference point, and z_2 is the number of points on the same horizontal line. Figure 1 gives vector z and typical boundary configurations associated with each of the six types. Intuitively, types 1 and 3 represent local peaks (concave or convex vertices), and types 2 and 4 are end segments of a horizontal line. The latter always occur in pairs making up a concave or convex boundary segment. Figure 2 illustrates the assignment of boundary types for a single region with two holes.

During step 2, each row of P is scanned left to right starting at the top and interior points are assigned a specified value. (This may be an intensity level, color or texture value).

To determine the left and right ends of an interior row, two parameters are employed. Parameter INT specifies the interior/exterior condition. It is assigned value 1 when the boundary is crossed at the left (from outside to inside) and value 0 when the crossing occurs at the right (from inside to outside). Typically, that happens at type 5 points. Parameter HLINE has value 1 when a horizontal boundary segment is scanned and value 0 otherwise. It

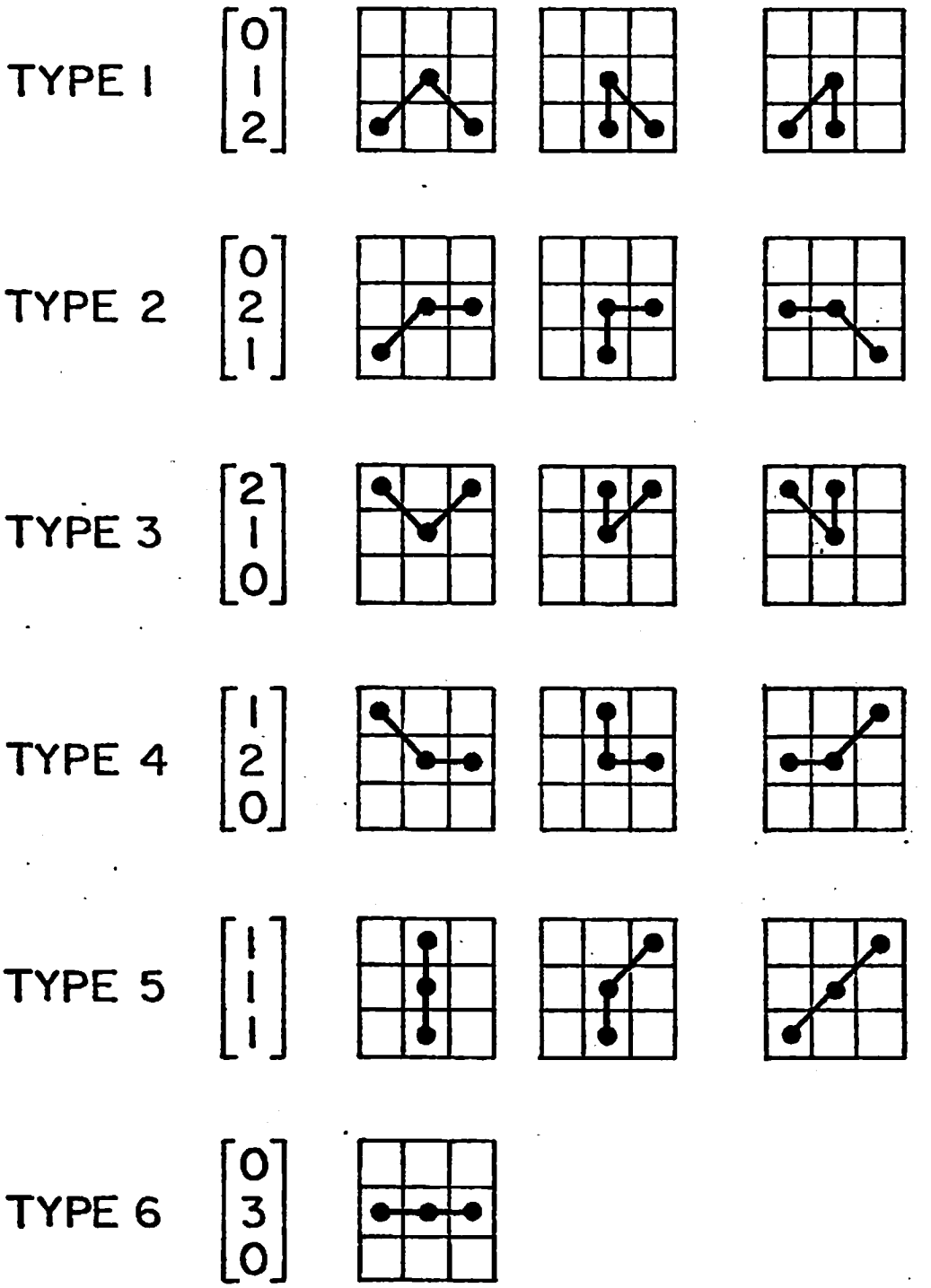


FIGURE 1. CLASSIFICATION OF BOUNDARY TYPES

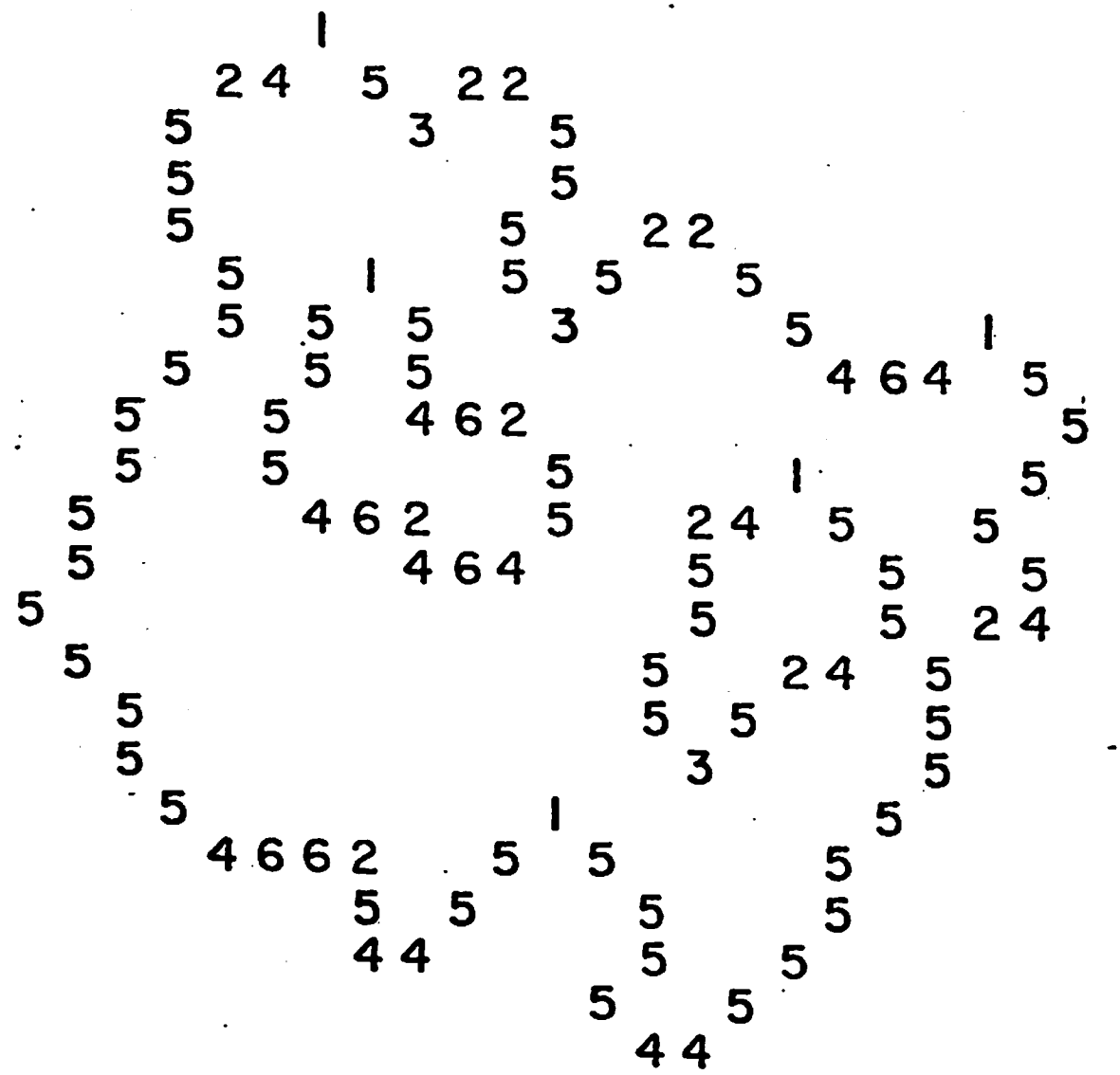


FIGURE 2. ASSIGNMENT OF BOUNDARY TYPES.

changes value at type 2 or type 4 vertices. While INT = 1 all points not part of a boundary are considered to be interior to the region. When a boundary point is encountered it is examined as follows:

(1) If INT = 0 then
    (a) types 1 or 3 and matched pairs of 2 and 4 are convex segments which are ignored;
    (b) type 5 and unmatched pairs of 2 and 4 are the beginning of an interior row (INT←1).
(2) If INT = 1 then
    (a) types 1 or 3 and matched pairs of 2 and 4 are concave segments which signify the continuation of an interior row;
    (b) type 5 and unmatched pairs of 2 and 4 are the end of an interior row (INT←0).

III. Algorithm for Adjacent Regions

In dealing with several adjacent regions one could proceed as above, i.e. by constructing matrix P_i for each region R_i. As a final step the submatrices would be appropriately combined to form the picture matrix P. This is straightforward but implies that (1) many picture cells would be examined more than once, and (2) the total required storage exceeds the actual area of the combined regions.

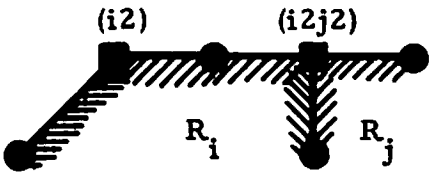
In this section the basic algorithm is modified so that all regions can be processed together. The first problem involves the storage of boundary types for joint vertices. A simple solution which requires no additional storage is to encode the region index R_i and boundary type T_i of each adjoining region in a single integer C of the form R2T2R1T1. For example, a common vertex having type 4 for region i and type 6 for region j is encoded as C = i4j6. Figures 3(a)

through (h) illustrate a few typical combinations of adjacent regions where the boundary encodings have been indicated for the square reference points. Note that types 1 and 3 are not included in the code for joint vertices (figures 3(c) and 3(f)), because they do not effect the labelling of interior cells. Also, vertices where three regions meet (see figure 3(h)) have only two associated region indices and boundary types. In fact, the following is asserted:

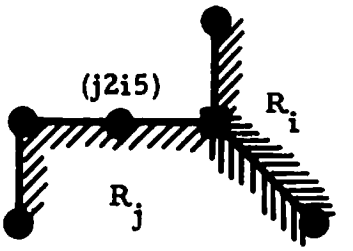
Claim For a joint vertex at which $k \geq 2$ regions meet at most two boundary types (and region indices) have to be stored.

Proof By eight-connectedness, $k \leq 8$. As noted above, types 1 and 3 can be omitted. Let k' be the number of adjoining regions minus type 1 or 3 regions. Thus, $k' \leq 4$. If $k' = 4$, then boundary types are adjacent pairs of types 2 and 4. Of these we need encode only two. Since type 5 (vertical) and type 6 (horizontal) are mutually exclusive at joint vertices the presence of either one implies $k' \leq 3$. For type 5 and $k' = 3$ the other types are complimentary types 2 and 4. For type 6 the remaining ones are either type 2 or type 4. Hence it suffices to encode type 5 (or type 6) together with one other type.

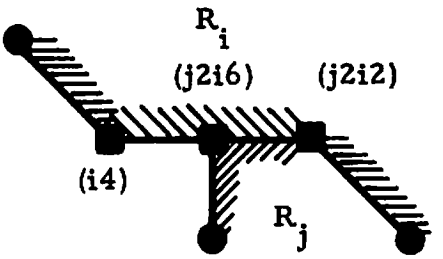
By the proceeding argument, the scheme for boundary encodings is sufficient for digitized contours based on eight-neighborhood adjacency. For ease of manipulation and to insure proper initial conditions at exterior joint vertices (for typical cases see figures 3(a) through (d) and 3(e)), boundary types are stored in the code in a certain order, namely, 5, 4, 6, and 2. For example, type 5 is always encoded in position T and any remaining types would be in position T . Figure 4 depicts the digitized contours



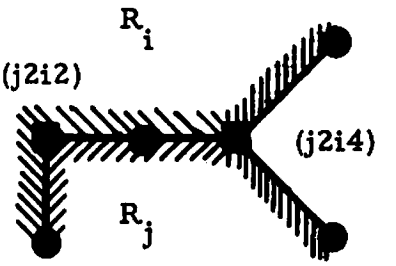
(a)



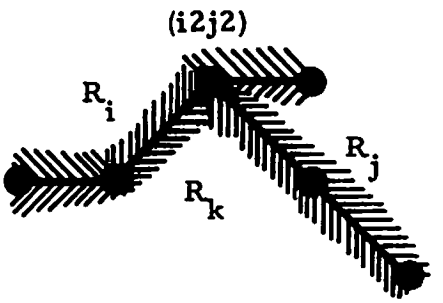
(b)



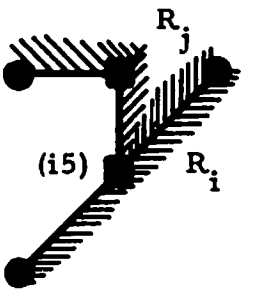
(c)



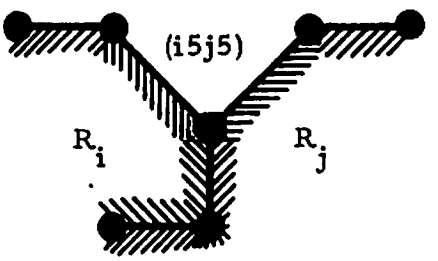
(d)



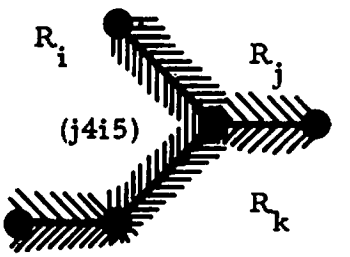
(e)



(f)



(g)



(h)

FIGURE 3. TYPICAL JOINT VERTICES

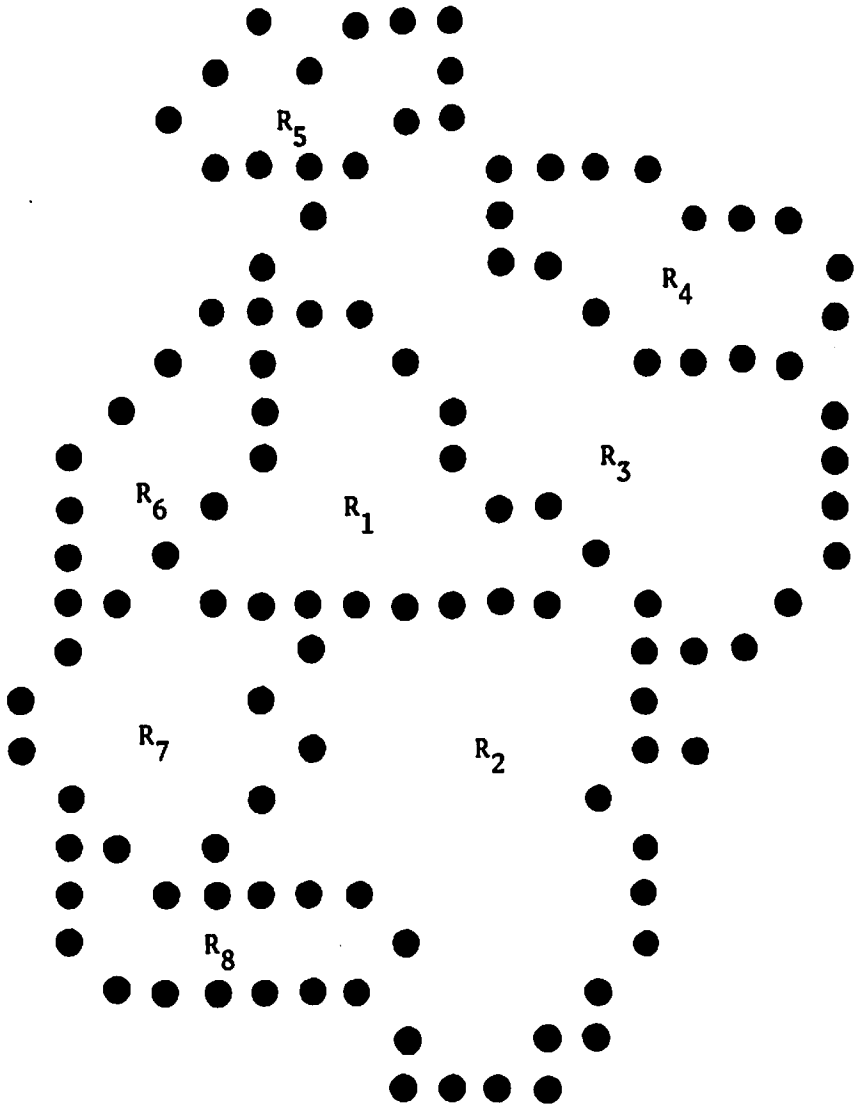


FIGURE 4. CONTOUR POINTS OF ADJACENT REGIONS

of several adjacent regions and figure 5 is a computer printout of the boundary codes for these regions.

Step 2 of the algorithm proceeds in a similar manner as discussed in Section II. At an exterior vertex the initial region is always R with type T . Parameter INT is assigned value i when region i is entered from the left (background is region 0), and HLINE is set to the index of the region under current consideration. The values of these two parameters also determine how code C is interpreted at a joint boundary point. Labelling of interior cells of the regions in figure 5 is shown in figure 6. For simplicity the region indices are used as labels.

IV. Conclusion

The algorithm presented in this paper is quite efficient in terms of storage and computational time. Each boundary point is examined once during step 1 and all cells of the entire image are examined once during step 2. Storage requirement is determined by the total area of the image. The encoding scheme devised for this algorithm permits considerable flexibility in that both adjacency and containment can be handled. The technique has potential application in image synthesis where the picture is generated from various parts having specified intensity, color or texture values.

References

[1] A. Rosenfeld and A.C. Kak, Digital Picture Processing, Academic Press, New York, 1976.

[2] E.M. Rounds, "Figure Construction from its Contour," USCIPI Report 720, Fall 1976.

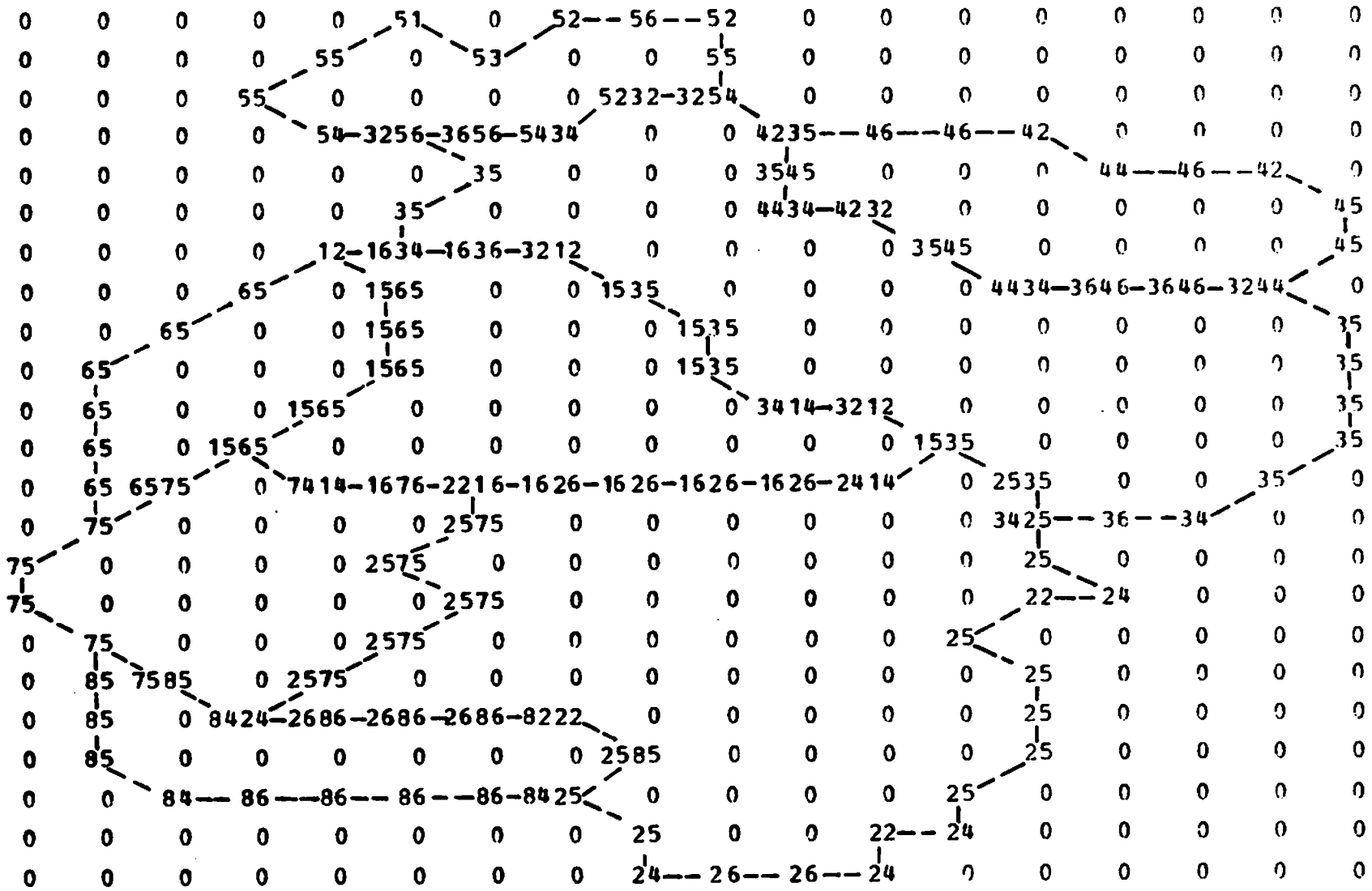


FIGURE 5. ASSIGNMENT OF BOUNDARY CODES

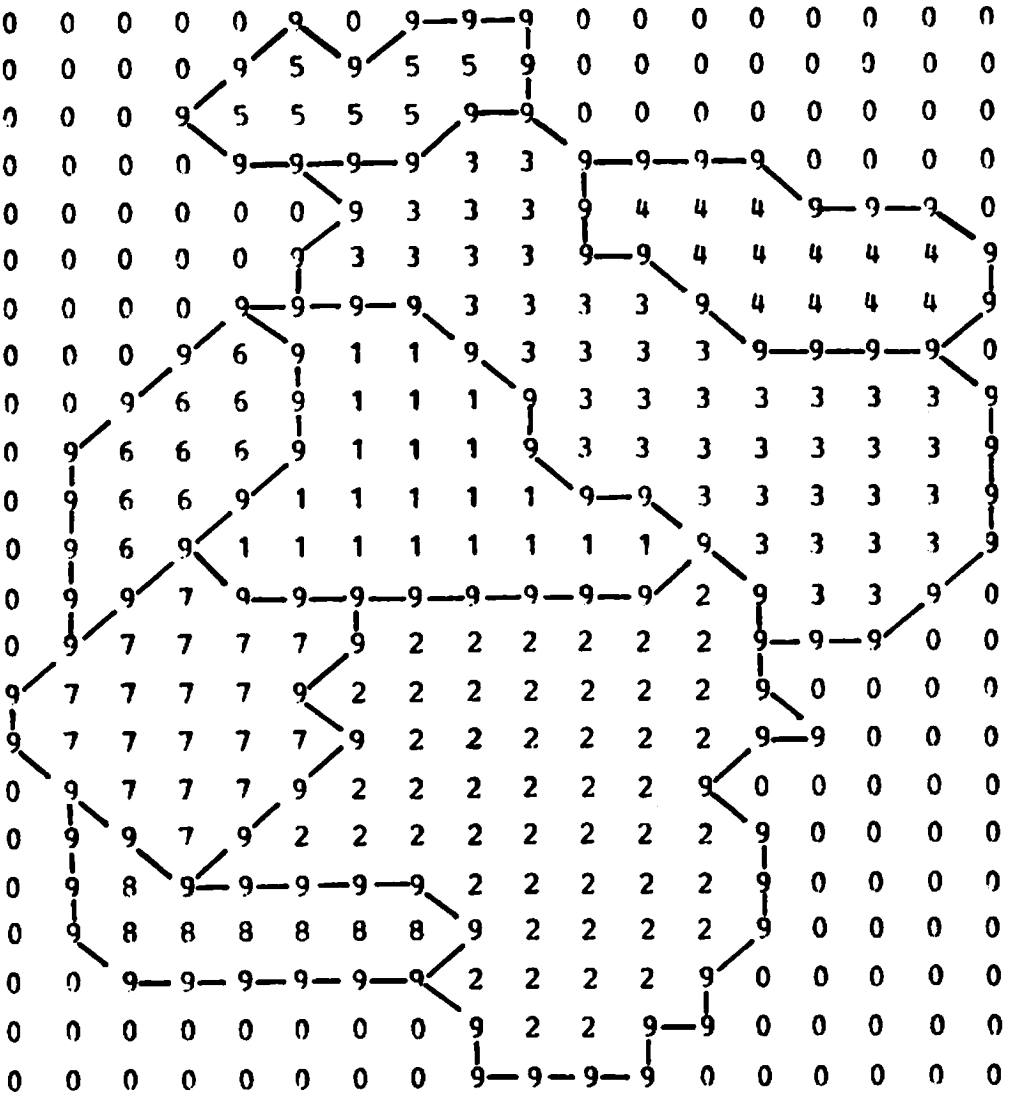


FIGURE 6. LABELLED INTERIOR POINTS

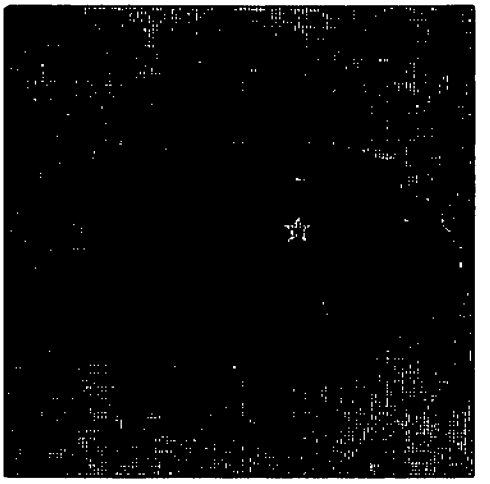
2.4 Extension of Boundary Segments in a Multi-Level System

Ramakant Nevatia and Kenneth Laws

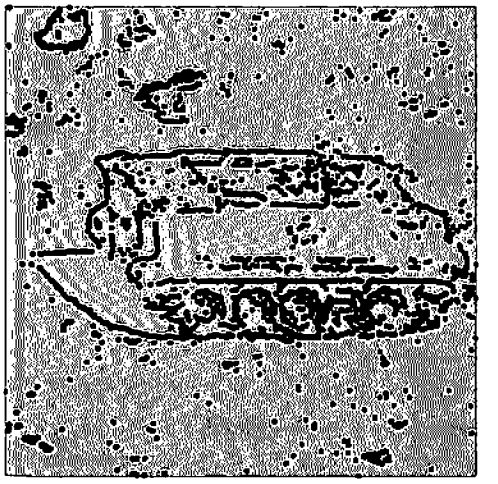
This section describes continuing efforts in our approach to scene segmentation by edge detection based methods. Obtaining boundaries of objects of interest is of central importance in analysis of a scene. Previously [1,2], we have described a technique that links local edges detected in an image into larger segments, providing partial boundaries for objects and removing much of the undesired textured background. Extension of such edge segments to yield more complete (longer segments) boundaries is described here.

Figure 1(a) shows an armored personnel carrier (APC) against a desert background. Figure 1(b) shows the edges detected in figure 1 by a Hueckel operator (only the edge positions are shown). Figure 1(c) shows the linked edge segments obtained by previously described linking techniques. Proximate edge elements having similar orientations are linked and only segments of a minimum length are retained.

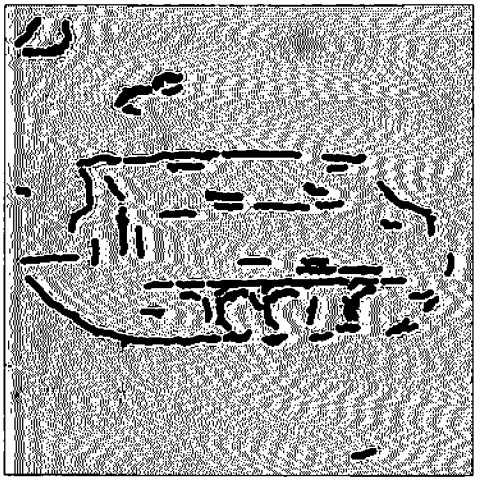
The outer boundary of the APC in figure 1(b) is continuous, however, the corresponding linked segments of figure 1(c) are disconnected. This is due to the stringent conditions required during the linking process. Now, these segments can be extended by relaxing the requirements of adding new elements to them. As only existing segments are to be extended, the relaxation will not cause the appearance of a large number of undesired segments, as would have been the case if these conditions were applied initially. We



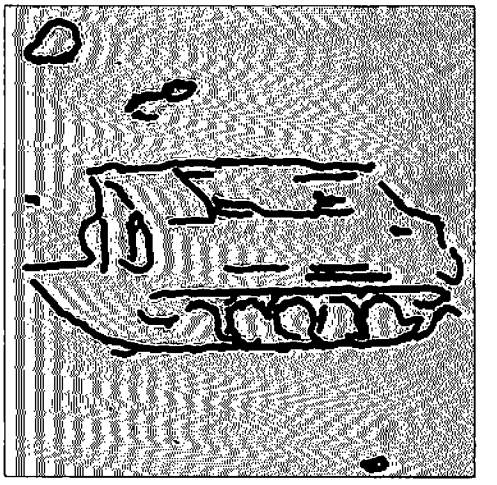
(a) Original



(b) Edges



(c) Linked segments



(d) Extended segments

Figure 1. An example of edge linking and extension.

thus have a multi-level boundary extraction process (some related issues are also found in Sections 2.5 and 2.6).

Each end of each segment is extended by incorporating one edge element after another until it meets another segment (or runs out of likely edge elements from which to choose). The segments are extended in the order of their lengths. The edge element chosen at each step is the one which most closely continues the direction and curvature of the segment.

The first step in the selection process is to define a square search neighborhood at the end of the segment, and to identify the edge elements within it.

Next a criterion function is computed for each edge element within the neighborhood, and the element producing the smallest value is chosen as the next extension. The criterion takes into account the change in segment trend (curvature) which the new extension would produce, as well as the difference between the new segment direction and the direction of the chosen edge element.

The problem of choosing a new segment direction and trend is similar to the statistical forecasting problem of recursively computing the mean and trend of a function as new points become available. It is not surprising that the same solution applies - exponential smoothing [3,4]. This is a moving average technique in which the most recent point is given the most weight. It requires very little storage or computation: the current direction and trend of each segment end are the only "historical" values which need to be stored.

Let the segment end have direction D_{SEG} and trend (or curvature) T_{SEG}. Initially D_{SEG} is the direction from the third edge element to the first (or end) element; $T_{SEG} = 0$. The values are computed adaptively as new edge elements are incorporated.

The program computes the direction, D_{EXT}, from the segment end to any edge element in the search neighborhood. Then, using modulo arithmetic:

$$D_{NEW} = D_{EXT} - BETA * (D_{EXT} - D_{SEG})$$

This new segment direction is compared to the direction of the edge element, D_{EDG}; if $|D_{EDG}-D_{NEW}|_{Mod}$ is less than some threshold value the program computes:

$$T_{EXT} = D_{EXT} - D_{SEG} .$$

Finally, for the edge producing the smallest change in trend, $|T_{EXT}-T_{SEG}|_{Mod}$, the new segment parameters are stored:

$$D_{SEG} = D_{NEW}, \; T_{SEG} = T_{NEW} = T_{EXT} - BETA * (T_{EXT} - T_{SEG}) .$$

If no suitable edge element has been found the extension of this segment end is terminated; otherwise a new search neighborhood is chosen and the process repeats.

The smoothing parameter, BETA, controls the speed with which the segment adapts to a new direction or curvature. A value of 1.0 forces the segment to retain its original direction and trend: the position of the segment end will change as new elements are incorporated, but the extender will not change the direction in which new elements are sought.

Values of BETA near 0.0 allow the segment to ignore past history and to adapt rapidly to a new direction or

curvature. This works well because straight segments in the image have already been removed; the tracing algorithm can thus be optimized for following curves and wavy edges.

Adaptive smoothing, like all moving average techniques, tends to lag behind constant trends. A simpler adjustment is to use a negative value of BETA (e.g. -0.3). This allows the extender to anticipate the segment direction a short distance ahead, provided that the curvature remains constant.

Figure 1(d) results from extension of segments in figure 1(c) by the above method. It was produced using a 6 x 6 pixel neighborhood, and BETA = -0.3. The threshold on the extended edge element direction described above was 45 degrees.

Our extension of edge segments differs from traditional edge followers in that the extension is from established boundary segments and only short extensions, hopefully to bridge gaps are desired. In many cases, the desired gaps to be bridged can be computed directly (such as between collinear segments). Edge elements within such gaps, if any, may then be examined directly. This approach is being considered for future research.

References

[1] R. Nevatia, "Locating Object Boundaries in Textured Environments," IEEE Transactions on Computers, November 1976, pp. 1170-1175.

[2] R. Nevatia, "Some Edge Segmentation Results," in USCIPI Report 720, University of Southern California, September 1976, pp. 78-84.

[3] C. McMillan and R.F. Gonzalez, Systems Analysis, Irwin, 1973, pp. 386-396.

[4] R.G. Brown, Smoothing, Forecasting and Prediction of Discrete Time Series, Prentice-Hall, 1963.

2.5 Detection of Edges in Elongated Neighborhoods

Ramakant Nevatia and Peter Chuan

Detection of object boundaries by edge detection requires the ability to perform sensitive local edge detection. The Hueckel operator normally performs well, but fails occasionally in the presence of fine texture. Edge operators using smaller neighborhoods, such as a Roberts or a Sobel operator, may pick up some of these edges but suffer from being sensitive to noise. We expect to use a variety of edge operators simultaneously for boundary construction. In a multi-level system, the more sensitive edge detectors may be used to fill in gaps in boundary segments derived from another edge detector.

Of particular interest are the elongated boundary segments. Here we describe a technique for detecting edges that belong to elongated segments. This restriction is expected to provide sensitivity to desired types of edges and not to fine texture or random noise. The technique is simply to convolve an image with elongated neighborhoods in various directions (masks for four directions are shown in figure 1). Each convolution gives a value indicating the magnitude of edge in that direction. The maximum value at each point and the associated direction are chosen as indicative of edge magnitude and direction at that point.

1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Horizontal

(a)

1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1

Vertical

(b)

0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0	0	-1
0	1	1	1	1	0	0	-1	-1	-1
1	1	1	0	0	-1	-1	-1	-1	0
1	0	0	-1	-1	-1	-1	0	0	0
0	-1	-1	-1	-1	0	0	0	0	0
-1	-1	-1	0	0	0	0	0	0	0

+30° Slant

(c)

1	1	1	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0
-1	0	0	1	1	1	1	0	0	0
-1	-1	-1	0	0	1	1	1	1	0
0	-1	-1	-1	-1	0	0	1	1	1
0	0	0	-1	-1	-1	-1	0	0	1
0	0	0	0	0	-1	-1	-1	-1	0
0	0	0	0	0	0	0	-1	-1	-1

-30° Slant

(d)

Figure 1. Edge Masks for Four Directions

The magnitude is thresholded for edge detection.

Figures 2 and 3 show results of such processing for two images. Figures 2(a) and 3(a) show the grey level images; figures 2(b) and 3(b) show the magnitude of edge at each pixel (the direction is not shown and masks in only four directions shown in figure 1 were used); and figures 2(c) and 3(c) show the thresholded edges.

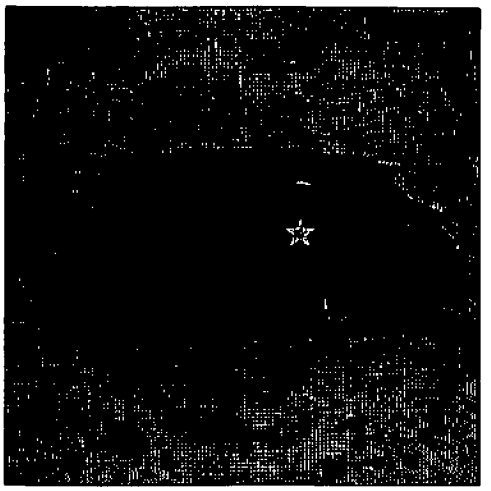
Note that most elongated edges of interest in the two images were detected. These edges may now be linked and other edge detectors used for filling small gaps or conversely the edges detected here could be used to fill in gaps left by another detector. Research is in progress to complete the design of such a multi-level program (see also Sections 2.4 and 2.6).

The concept of convolution by chosen masks has been widely used in the past and general selection of data obtained by using multiple masks is described in [1] and [2]. However, in the scheme suggested above, the results are to be used for specific goals and simpler selection procedures may suffice.

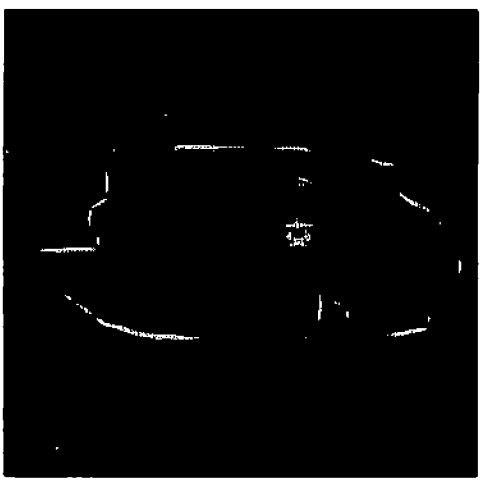
References

[1] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Transactions on Computers, Vol. C-20, May 1971, pp. 562-569.

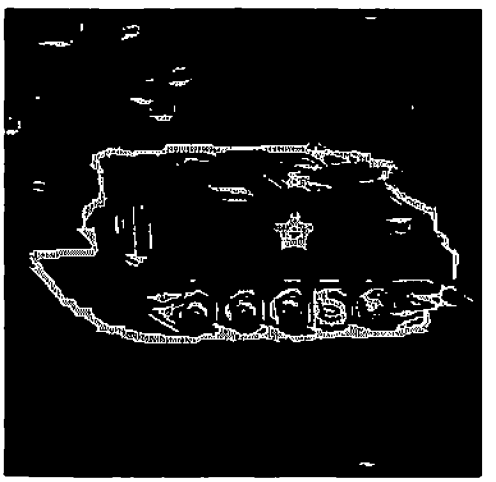
[2] D. Marr, "Early Processing of Visual Information," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, AI Memo No. 340, December 1975.



(a) Grey level image

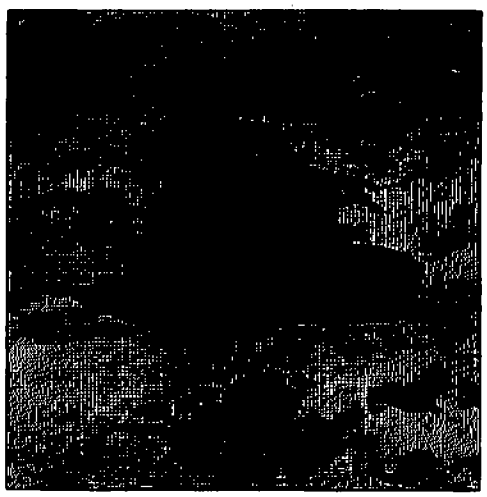


(b) Magnitude of edges

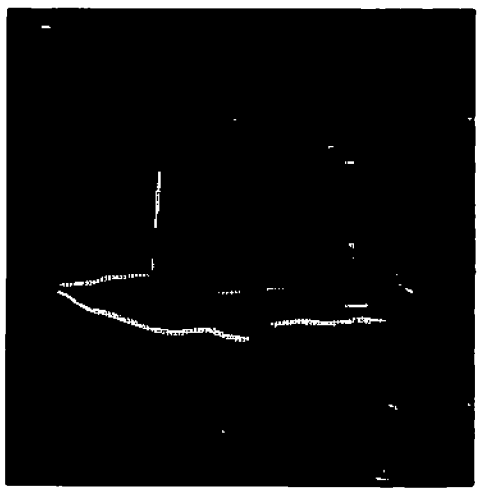


(c) Thresholded edges

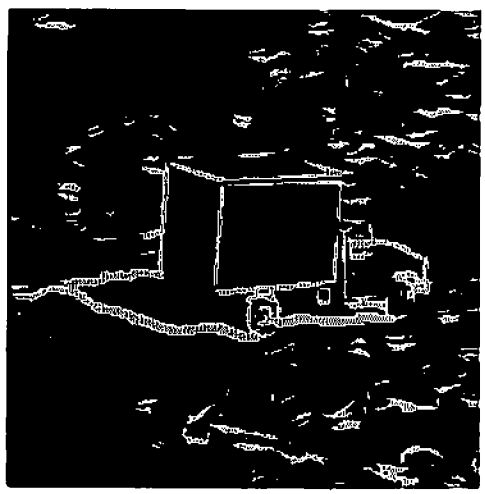
Figure 2. Edge detection results for one image.



(a) Grey level image



(b) Magnitude of edges



(c) Thresholded edges

Figure 3. Edge detection results for another image.

2.6 Color Edge Detection in Scene Segmentation

Ramakant Nevatia

A color edge detector, based on the achromatic Hueckel edge detector has been described previously [1-2]. This report discusses the usefulness of such color edges in scene segmentation in comparison to the use of achromatic edges, and provides an update of the previous results.

It is assumed here that the color edge detector is given three color components (in a chosen coordinate system) of a color image and returns three optimal step approximations in the corresponding components. It is required that the three steps have the same spatial orientation. Decision of the presence of an edge in each component is made independently as in the case of achromatic edge detection.

The R, G and B components of a color image are first transformed into Y, T_1 and T_2 defined as follows:

$$Y = c_1 R + c_2 G + c_3 B$$

$$T_1 = \frac{R}{R + G + B}$$

$$T_2 = \frac{G}{R + G + B}$$

where c_1, c_2 and c_3 are suitably chosen constants so that Y corresponds to the luminance of the image ($c_1 + c_2 + c_3 = 1$). T_1 and T_2 jointly describe the purely

chromatic information. If T_1 and T_2 are thought of as spanning a plane in color space then the polar coordinates of a point in this plane approximate the commonly used attributes of hue and saturation. We perform color edge detection in the Y, T_1 and T_2 components first and compute the edges in hue and saturation from them (because of technical difficulties in computing the hue edges directly, as hue is measured modulo 2).

Results of color edge detection on one image are presented here. The chosen image is a standard SMPTE test picture of a girl (chosen because of its general availability). The comments on results apply to the limited set of images that have been tested.

Chromatic versus Luminance Edges:

The first experiments are to test if the edges in certain components are more meaningful or useful than in other components. Figures 1(a), (b) and (c) show the R, G and B components of the test picture. Figures 2(a) and (b) show the edges detected in the luminance component and one of the T_1 and T_2 components respectively of the image of figure 1. The edge operator was applied at every other pixel along every other row. The number of edges occurring in the two figures is 6310 and 4068 respectively.

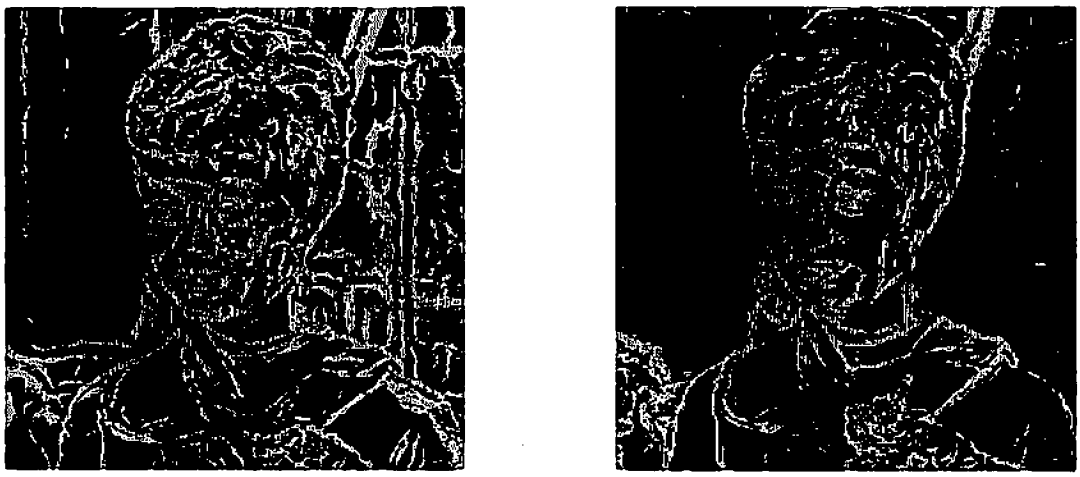
Edges detected in the T_1 and T_2 components are also separated into those occurring in the hue or the saturation components, as shown in figures 3(a) and (b) for the Girl image. The number of edges occurring in figures 3(a) and (b) is 3040 and 3195 respectively. Separation of the edges into hue and saturation components does not seem to be of any particular value, at least for the limited number of pictures analyzed. (The edges occurring in T_1 or T_2

(a)

(b)

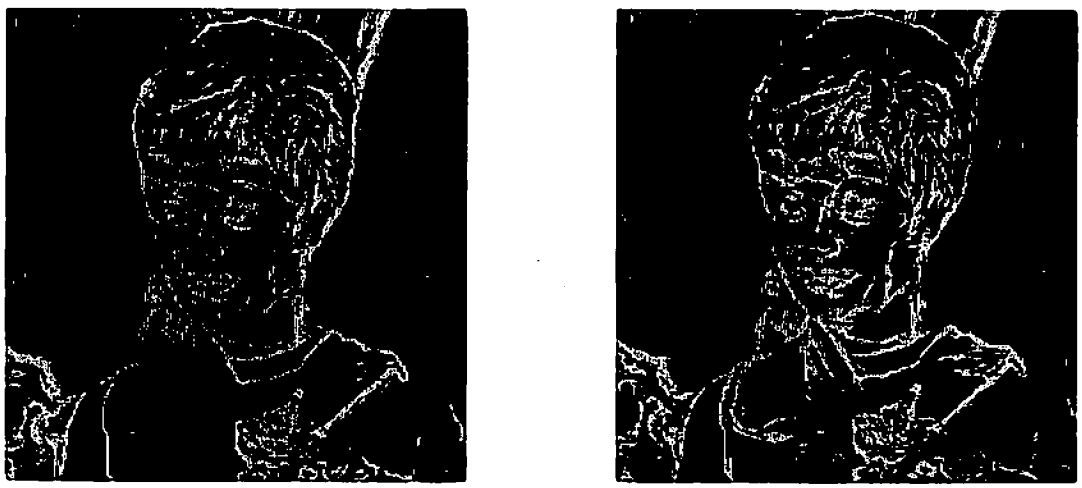
(c)

Figures 1a, b and c. Red, Green and Blue components of a color "Girl" picture.



(a) (b)

Figure 2. Color edges detected in the "Girl" picture
(a) Luminance component
(b) T_1, T_2 or both components.



(a) (b)

Figure 3. Hue and saturation edges detected in the "Girl" picture.

components do not all appear as edges in the hue or the saturation components, because of slightly different thresholding methods. For the remainder, we use the term chromatic edges to mean edges occurring in any of the chromaticity components.

The edges detected in the chromatic components are smaller in number than those detected for the luminance component. Further, a larger percentage of the edges in the chromatic components correspond to the desired edges, i.e. the edges corresponding to the boundaries of desired objects. This is in conformance to our intuitive expectations that the "color" in a picture should be relatively constant over large areas of the surface of a single object, whereas the luminance may change more erratically for various reasons including uneven illumination and reflections. However, note that the edges in the chromatic components have more gaps at the boundaries of objects of interest.

The observation of major importance is that a luminance edge is also present for a very large percentage of the cases where there is a chromatic edge present. This should not be surprising; in natural scenes, it is unlikely that objects of different hue will accidentally have the same luminance components. This implies that the luminance edges contain most information needed to obtain object boundaries, but that this information may be more difficult to isolate.

Of course, in some parts of some scenes, only chromatic edges will be present at boundaries of interest. This is more likely to occur in scenes with low contrast or poor illumination. Some instances of such edges are present in the edges around the scarf area of the Girl picture (compare figures 3(a) and (b)).

The computation time for color edge detection was approximately three minutes for each image using a DEC PDP-10, KI-10 processor. This is roughly three times the computation time required for detecting edges in an achromatic image of the same size.

Linking of Edges:

Evaluation of usefulness of the color edges is, of course, dependent on the further processing performed on them. Here, we limit the goal to obtaining boundaries of objects of interest in a scene. It has been suggested that one useful technique for extracting such boundaries is to link local edge elements into elongated segments [3,4]. It is hypothesized that the edges along objects align in elongated segments but the edges belonging to textured backgrounds and spurious edges are generally distributed at random.

One technique of linking edge elements into nearly straight line segments is described in detail in [3]. Only neighboring edge elements having orientations within specified limits of each other are linked. Further only edge segments that contain at least a certain minimum number of edge elements are preserved. The linking of color edges, each with descriptions of three edge components, uses additional constraints. In addition to the requirements of proximity in position and orientation, the two edges to be linked are required to have compatible color characteristics as defined below.

Each color edge is a step (or line) function in a three-dimensional color space. Such an edge may be considered to have an orientation in the color space, determined by the relative values of the step amplitude in

the three components. Then, an angle φ between two edges in the color space (to be differentiated from the spatial angle between them) is defined by

$$\cos(\varphi) = \frac{\underset{\sim}{S}_1 \cdot \underset{\sim}{S}_2}{\|\underset{\sim}{S}_1\| \cdot \|\underset{\sim}{S}_2\|}$$

where $\underset{\sim}{S}_1$ and $\underset{\sim}{S}_2$ are vectors representing the two edges, $\underset{\sim}{S}_1 \cdot \underset{\sim}{S}_2$ stands for their dot product and $\|\cdot\|$ stands for the Euclidean norm. Two edges are defined to be color compatible if the angle φ between them is less than a threshold (90^{o} for results presented below).

Consider the linking of color edges that occurs in any one or more of the three color components, i.e. the edges occurring in either of the figures 3(a) or (b) for example. The result of linking such edges is shown in figure 4. Here, only the edge segments consisting of more than seven edge elements are shown. (Some of the linked segments are wavy as no thinning or smoothing operations have been performed on the initial edge data. It is easier to perform a thinning operation now, as a direction for thinning is known.) The computation times for the linking operation were approximately one minute of CPU time for each picture.

It is clear that most of the edges belonging to object boundaries are retained while many undesired, spurious edge elements have been eliminated. However, many of the desired properties of the linked segments derive from the linking procedure rather than the use of color information. Strikingly similar results are obtained by merely applying the linking procedure to the luminance edges alone (figure 3). (The results are so similar that figures for linked luminance edges have been omitted.) Results of linking only the chromatic edges detected in the Girl picture (figure 3(b)) are shown in figure 5. There are now fewer spurious

Figure 4. Linked segments from edges in Figure 3 a.

Figure 5. Linked segments from edges in Figure 3 b.

linked segments, compared to figure 4, but larger gaps in the boundaries of interest.

Conclusions:

The important question is whether the use of color during edge detection aids the segmentation process (other uses of color, e.g. for classification of objects are not considered here). As the results presented earlier indicate, it is as naive to expect that edges in chromatic components would correspond solely to object boundaries as it is to expect the same for the edges in the luminance component. In experiments with a limited number of natural color images, the edges in chromatic components were found to be largely contained in the edges in the luminance component. This implies that most information of interest is embedded in the luminance component, though it may be more difficult to extract. Clearly, situations exist in pictures of low contrast where the luminance edges are absent but chromatic edges are present (chromatic content should not be strongly affected by the illumination, except for the sensor performance limitations). Also, for specific applications, edges in a particular chromatic attribute, such as hue may be of particular interest.

More generally, the use of color is likely to aid in building a more robust and reliable system. Consider a multi-level scheme for extracting object boundaries, utilizing a linking scheme of the type described earlier. For initial input, this system may start with edges in the chromatic components only, perhaps with thresholds set high to yield only edges with high color contrast, or only the edges co-occurring in the luminance and the chromatic components, as such edges seem to consistently contain fewer spurious edges. After linking of these edges, gaps in the

resulting segments may now be filled in with the luminance edges which are far more numerous, thus allowing guided use of such information. Further, in many cases, the initial segments may be sufficient to suggest a few, limited number of hypotheses for the identity of objects in the scene, thus allowing the powerful mechanism of using model specific knowledge in conjunction with other less reliable data to verify the presence of certain objects.

Such a scheme would be capable of using color information as available but capable of performing in absence of such data. This is consistent with the experience of human perception where achromatic pictures are generally sufficient for extracting most information of interest but the color pictures seem perceptually richer. Further experimentation is required to determine if the improved performance using color is worth the threefold increase in the requirements of storage and computation, at the current costs for these resources.

References

[1] R. Nevatia, "Hueckel Color Edge Detector," in USCIPI Report 660, pp. 70-81.

[2] R. Nevatia, "A Color Edge Detector," Proceedings of the Third International Joint Conference on Pattern Recognition, Coronado, California, November 1976, pp. 829-832.

[3] R. Nevatia, "Locating Object Boundaries in Textured Environments," IEEE Transactions on Computers, Vol. 25, No. 11, November 1976, pp. 1170-1175.

[4] D. Maar, "Early Processing of Visual Information," Massachusetts Institute of Technology, Artificial

Intelligence Laboratory, A.I. Memo No. 340, December 1975.

2.7 Calculation of Edge Detector Parameters by Ho-Kashyap Algorithm

William K. Pratt and Ikram E. Abdou

Introduction: In a recent paper [1], edge detection was formulated as the classical communication problem of signal detection in the presence of noise. In the following work edge detection is discussed as a problem of classifying patterns into two classes (edge and no edge). Many techniques have been developed in pattern recognition to solve this problem. One of them, the Ho-Kashyap algorithm, will be analyzed. The Ho-Kashyap algorithm is briefly reviewed, and the algorithm is then used to find parameters of the Roberts operator. Results obtained by these parameters are compared with probabilities of detection and false alarm derived theoretically.

Ho-Kashyap Algorithm: The Ho-Kashyap algorithm [2,3] can be used to find a weighting vector $\underline{w}$ which correctly classifies prototypes into classes Ω_1 and Ω_2 according to the rule

$$\underline{w}^T\underline{x} > 0 \Longrightarrow \underline{x} \in \Omega_1 \tag{1}$$

$$\underline{w}^T\underline{x} < 0 \Longrightarrow \underline{x} \in \Omega_2 \tag{2}$$

Equation (2) can be rewritten as

$$\underline{w}^T(-\underline{x}) > 0 \tag{3}$$

In the special case of single feature classification, for example the scalar edge gradient magnitude g, the augmented

vector $\underline{x}$ is given by

$$\underline{x} = \begin{bmatrix} g \\ 1 \end{bmatrix} \tag{4}$$

Hence

$$\underline{w}^T \underline{x} = [\, w(1) \quad w(2) \,] \begin{bmatrix} g \\ 1 \end{bmatrix} = w(1)\, g + w(2) \tag{5}$$

Equations (1) and (2) then reduce to

$$g > -\frac{w(2)}{w(1)} \Longrightarrow \underline{x} \in \Omega_1 \tag{6}$$

$$g < -\frac{w(2)}{w(1)} \Longrightarrow \underline{x} \in \Omega_2 \tag{7}$$

In this case $-w(2)/w(1)$ is equal to the threshold T commonly employed in edge detection [1].

In general, for classification tasks with more than one feature, eqs. (1) and (2) can be combined into

$$\underline{X}\,\underline{w} = \underline{b} \tag{8}$$

where

$$\underline{X} = \begin{bmatrix} \underline{x}_1^T \\ \vdots \\ \underline{x}_N^T \\ -\underline{x}_{N+1}^T \\ \vdots \\ -\underline{x}_{2N}^T \end{bmatrix} \tag{9}$$

and $\underline{x}_1,\ldots,\underline{x}_N$ are elements of Ω_1, while $\underline{x}_{N+1},\ldots,\underline{x}_{2N}$ are elements of Ω_2. The components $b = (b_1,\ldots,b_{2N})^T$ are all positive.

Equation (8) can be solved through the iteration formulae

$$\underline{w}(1) = \underline{X}^{\#}\ \underline{b}(1) \tag{10}$$

$$\underline{e}(k) = \underline{X}\,\underline{w}(k) - \underline{b}(k) \tag{11}$$

$$\underline{w}(k+1) = \underline{w}(k) + c\ \underline{X}^{\#}[\,\underline{e}(k) + |\,\underline{e}(k)\,|\,] \tag{12}$$

$$\underline{b}(k+1) = \underline{b}(k) + c\ [\underline{e}(k) + |\underline{e}(k)|\,] \tag{13}$$

$b(1) > 0$ but otherwise is arbitrary and where $\underline{X}^{\#} = (\underline{X}^T\underline{X})^{-1}\,\underline{X}^T$ is the pseudoinverse of $\underline{X}$, while c is a constant such that $0 < c < 1$. Equations (11) through (13) are repeated until $\underline{e}(k)$ converges to zero or when $\underline{e}(k)$ are less than a small number ϵ. The required parameters are then given by $\underline{w}(k)$.

Weight Parameters for Roberts Operator: The Ho-Kashyap algorithm can be used in the design of the Roberts edge detection operator. Prototypes are generated in two-by-two pixel groups according to the pattern

A(i)	B(i)
C(i)	D(i)

For an edge prototype A, B, C and D are given by

$$A = B = G(b,\sigma) \tag{14}$$

$$C = D = G(b + h,\sigma) \tag{15}$$

while for no edge

$$A = B = C = D = G(b,\sigma) \tag{16}$$

where $G(b,\sigma)$ is a Guassian random number generated with mean b and standard deviation σ representing image noise. A matrix $\underline{X}$ is formed as

$$X(i,1) = ((A(i) - D(i))^2 + (B(i) - C(i))^2)^{\frac{1}{2}} \tag{17}$$

$$X(i,2) = 1 \tag{18}$$

for i = 1,...,N and A, B, C and D are given by eqs.(14) and (15). For i = N+1,...,2N, eqs.(17) and (18) are used with negative signs and A, B, C and D are given by eq.(16). The number of prototypes N is chosen to almost guarantee separability of classes [3].

A series of experiments has been performed to evaluate the Ho-Kashyap procedure for the Roberts edge detector. The experimental parameters include: edge height h = 25 (10% magnitude) and SNR (signal-to-noise ratio) set at 1 and 10. The iteration has been repeated 250 times. The resultant weight vectors $\underline{w}$ are indicated in Table 1. It should be observed that with $\epsilon = 0.001$ and 250 iterations $\underline{w}$ reaches a steady state value. It appears that no more improvement can be anticipated for an increased number of iterations.

Evaluation of Results: The weighting vector $\underline{w}$ obtained by the Ho-Kashyap algorithm has been used to classify a new set of 250 prototypes corresponding to both cases of edge or no edge. In each case the probability of detection (if used with edge prototypes) or probability of false alarm (if used with no edge prototypes) is calculated and compared with theoretical results obtained in previous work [1]. The results are given in Table 1. It should be noticed that both practical and theoretical results agree. In addition, the parameters obtained are an optimum compromise between

SNR	$\underline{w}$	T	Probability of detection		Probability of false alarm	
			Experimental	Theoretical	Experimental	Theoretical
1	$\begin{bmatrix} 0.00394 \\ -0.189 \end{bmatrix}$	47.95	52%	55.9%	37.6%	39.9%
10	$\begin{bmatrix} 0.16 \\ -3.8 \end{bmatrix}$	23.75	91.2%	89.2%	11.6%	10.5%

Table 1

Comparison of Experimental and Theoretical Results

probabilities of detection and false alarm.

Conclusion: The results obtained show that the Ho-Kashyap algorithm can be a useful method for edge detector design. Although Gaussian noise is used in the experiment to simplify comparison with theoretical results, the same technique can be easily extended to general cases with any edge forms and noise models.

References

[1] I.E. Abdou and W.K. Pratt, "Statistical Design of Edge Detectors," USCIPI Semiannual Technical Report, September 1976, pp. 90-95.

[2] Y.-C. Ho and R.L. Kashyap, "An Algorithm for Linear Inequalities and its Applications," IEEE Transactions on Electronic Computers, Vol. EC-14, No. 5, October 1965, pp. 683-688.

[3] J.T. Tou and R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1974.

3. Image Processing Projects

This section surveys the progress made in the past six months on various image processing projects. Three new areas are discussed, those of image filtering based on the human visual system, optical filters from digitally constructed kinoforms (holograms) and spatial warp techniques. On-going projects include the estimation of object boundaries in noise, and a posteriori restoration. This latter project has experienced preliminary success in deriving the phase component of the OTF from spatially invariant distortions. Finally one project has reached fruition and completion, that of variable knot splines for image approximation. This technique has led to self-adaptive two-dimensional approximation methods which automatically sense the local activity of a region and apply enough knots (samples) locally to minimize a regional approximation. The technique has applicability in bandwidth compression, image understanding, and particularly in adaptive smart sensing. In the former case, adaptive compressions are available. In the latter case on-board high resolution sensor reduction is possible, and in the image understanding case, the knot density represents a useful feature for higher level processing.

3.1 Variable Knot Splines for Image Approximation

Harry C. Andrews and Dennis G. McCaughey

I. Introduction

This report presents a degree of freedom or information content analysis of images in the context of digital image processing. As such it represents an attempt to quantify the number of truly independent samples one gathers with imaging devices.

The degrees of freedom of a sampled image itself are developed as an approximation problem. Here bicubic splines with variable knots are employed in an attempt to answer the question as to what extent images are finitely representable in the context of digital sensors and computers. Relatively simple algorithms for good knot placement are given, and result in spline approximations that achieve significant parameter reductions at acceptable error levels. The knots themselves are shown to be useful as an indicator of image activity, and have potential as an image segmentation device as well as easy implementation in CCD signal processing and focal plane smart sensor arrays. Both mathematical and experimental results are presented.

Fundamentally, this concept of degrees of freedom can be viewed as an attempt to quantify the number of truly independent samples of data one gathers with photographic or other imaging devices. As image sensor technology grows, the quantity of data gathered increases, and it becomes reasonable to ask what the true increase in information content is as one increases image samples. This is especially important in military imaging applications where an increase in the quantity of data gathered, while not

producing a corresponding increase in image information, subjects the communication systems and end users to an unnecessary increase in bandwidth and data saturation without improving exploitation results.

Here the problem will be considered as a two-dimensional approximation problem and the concept of an "epsilon degrees of freedom" will be developed. By this it is meant that the degrees of freedom of an image at a level epsilon will be the minimum number of functions needed to approximate f(x,y) within an accuracy of epsilon assuming a particular metric.

From a "smart sensor" viewpoint, by way of motivation, if we consider a sampled image consisting of N^2 samples that could be approximated to an acceptable error by a least squares polynomial of M^2 variables with $M^2 << N^2$, it would be reasonable to say that this image had less than N^2 DOF in a least squares sense using a polynomial as the approximation technique. This approach is taken to circumvent the difficulty of associating a finite DOF to an image defined on a continuum which obviously has an uncountably infinite DOF if we desire to specify that image exactly. However, if we are willing to accept an approximation with a small but nonzero error, then the possibility exists in quantifying the DOF in this manner.

To illustrate the applicability of adaptive processing consider that advances in charge coupled device (CCD) sensors are such that some preprocessing within the sensor itself is not so unrealistic. This preprocessing could involve some evaluation as to what data constitutes information to the user and transmits only that data relevant to his needs. This increase in sensor sophistication coupled with the ability to gather large

quantities of data and to do adaptive sampling or some other more exotic processing to get at the real information content in the data, may provide fruitful results.

II. The Degrees of Freedom Viewed as an Approximation Problem

In characterizing the degrees of freedom of an image as an approximation problem we are confronted with two questions, namely: 1) to what extent is $f(x,y)$ finitely representable, and 2) the determination of the finite representation from a sampled version of $f(x,y)$.

Polynomial splines are chosen due to their approximation properties and the fact that they possess a basis namely, the normalized B-spline basis, which provides a local basis property allowing a rapid generation with the matrices involved in generating a B-spline fit to a function f being well conditioned. With DeBoor's algorithm for computations using normalized B-splines [1] no difficulties are encountered in handling multiple order knots. Hereafter we will consider a spline $S_{k,N_xN_y}(x,y)$ of the order k (the degree being equal to k-1) with N_x and N_y knots in the x and y directions respectively, to be of the following form:

$$S_{k,N_xN_y}(x,y) \triangleq \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} S_{i,j}N_{i,k}(\xi;x)N_{j,k}(\eta;y) = \hat{f}(x,y) \qquad (1)$$

where $N_{i,k}(\cdot)$ are the normalized B-splines of order k satisfying the following recursion relationship over the knot sets $\{\xi_1,\xi_2,\ldots,\xi_{N_x}\}$ and $\{\eta_1,\eta_2,\ldots,\eta_{N_y}\}$ [1].

$$N_{i,k}(\xi,x) = \frac{x - \xi_i}{\xi_{i+k-1} - \xi_i} N_{i,k-1}(\underline{\xi};x) + \frac{\xi_{i+k} - x}{\xi_{i+k} - \xi_{i+1}} N_{i+1,k-1}(\underline{\xi};x)$$

$$N_{i,1}(\underline{\xi},x) = \begin{cases} 1 & x \in [\xi_i, \xi_{i+1}) \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{i=1}^{N_x} N_{i,k}(\xi;x) = 1 \quad \forall\, x \in [\xi_1, \ldots, \xi_{N_x}] \qquad (2)$$

and

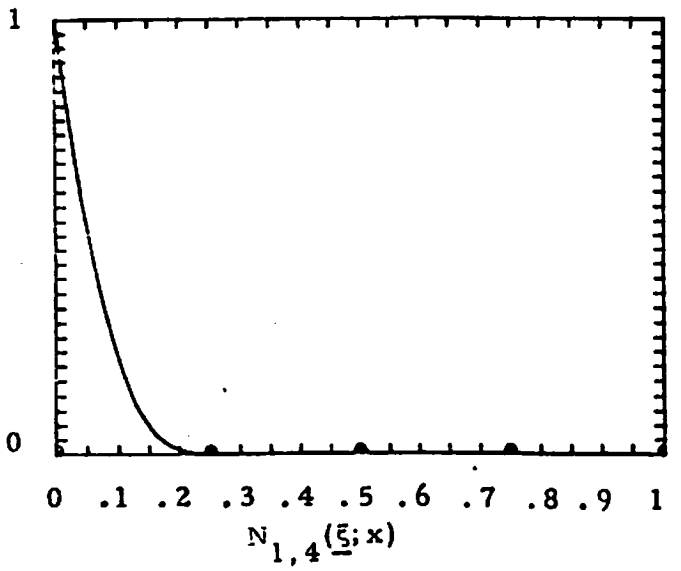
Note that the above indicates that $N_x + k$ knots are required to generate N_x normalized B-splines of order k, and that $N_{i,k}(\xi;x)$ is nonzero only over the interval $[\xi_i, \xi_{i+k})$. Also a knot may have multiplicity p, up to k in which the multiplicity indicates a discontinuity in the $(k - (p+1))$ derivative at that knot. We will follow Rice [2] and adopt the convention that the spline is differentiable of order 0 or -1 at the knot ξ_o if the spline is continuous or has a simple jump at ξ_o respectively. Thus a fourth order knot at ξ_o for a cubic (fourth order) spline indicates a simple jump in the spline at ξ_o. Figures 1 and 2 illustrate the normalized B-splines of order four for the knot vectors

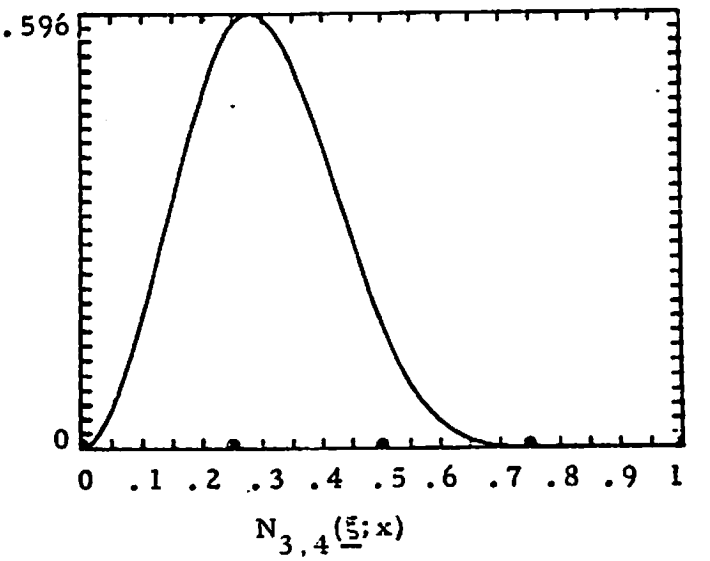
$$\underline{\xi}_1 = (0,0,0,0,.25,.5,.75,1,1,1,1)^t$$

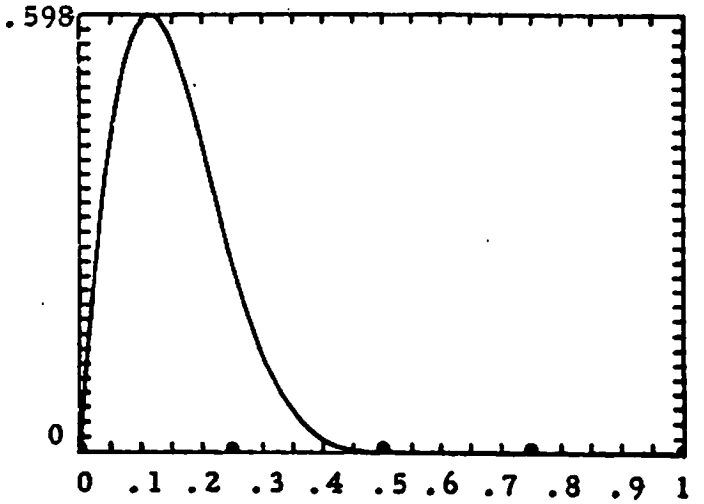
and

$$\underline{\xi}_2 = (0,0,0,0,.7,.8,.9,1,1,1,1)^t$$

respectively. $\underline{\xi}_1$ corresponds to the uniform knot case while $\underline{\xi}_2$ illustrates the effect of adaptively varying the internal knots towards 1. Note also that there are a total of 11 knots to define these seven nonzero normalized B-splines.







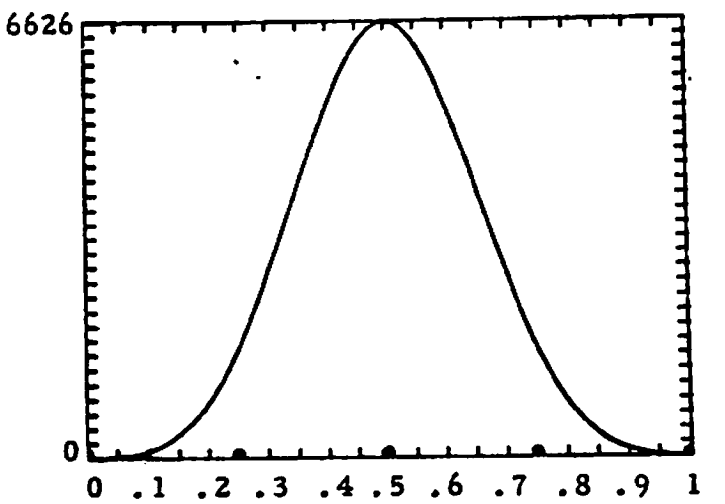
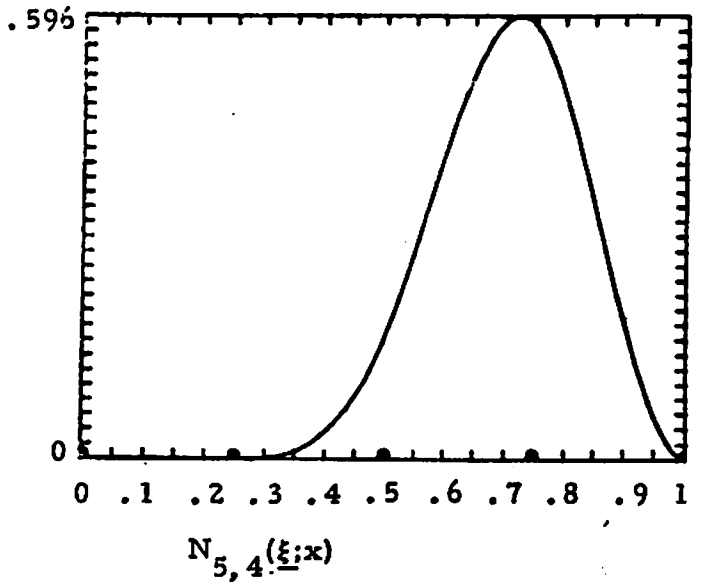
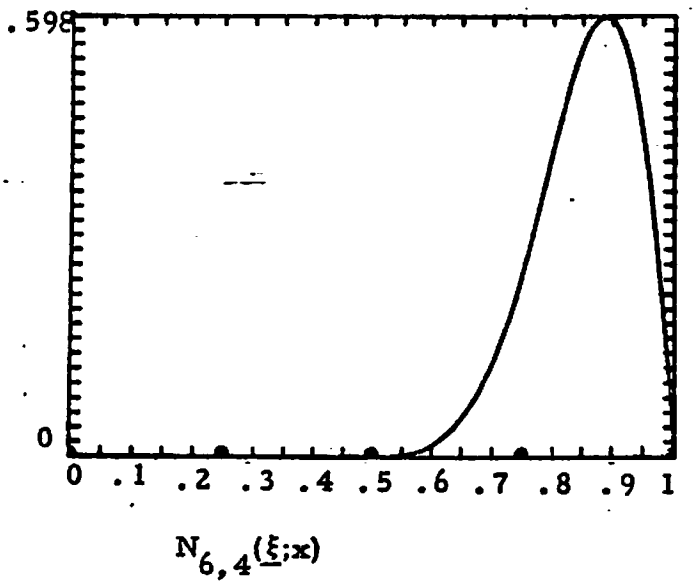


Figure 1 Normalized 4th Order B-Splines for Knot Vector $\underline{\xi}$ = (0,0,0,0,.25,.5,.75, 1,1,1,1)

Uniform Knots





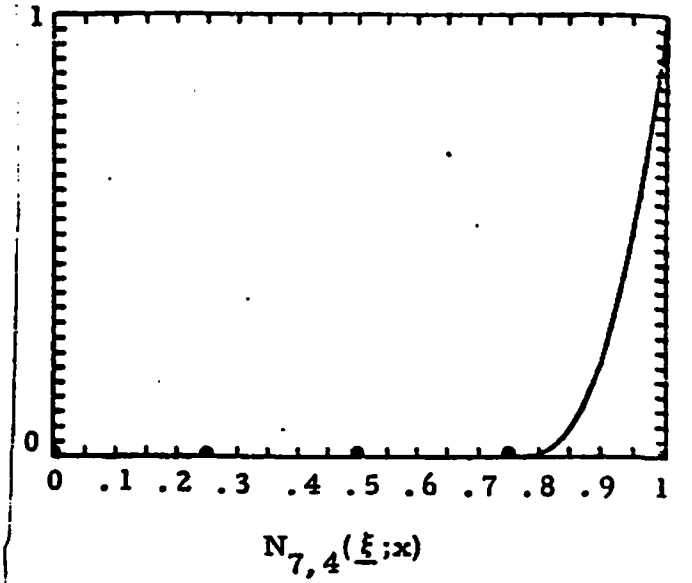


Figure 1 (continued) Normalized 4th Order B-Splines for Knot Vector $\underline{\xi}_1 = (0,0,0,0,.25,.5,.75,1,1,1,1)$

Uniform Knots

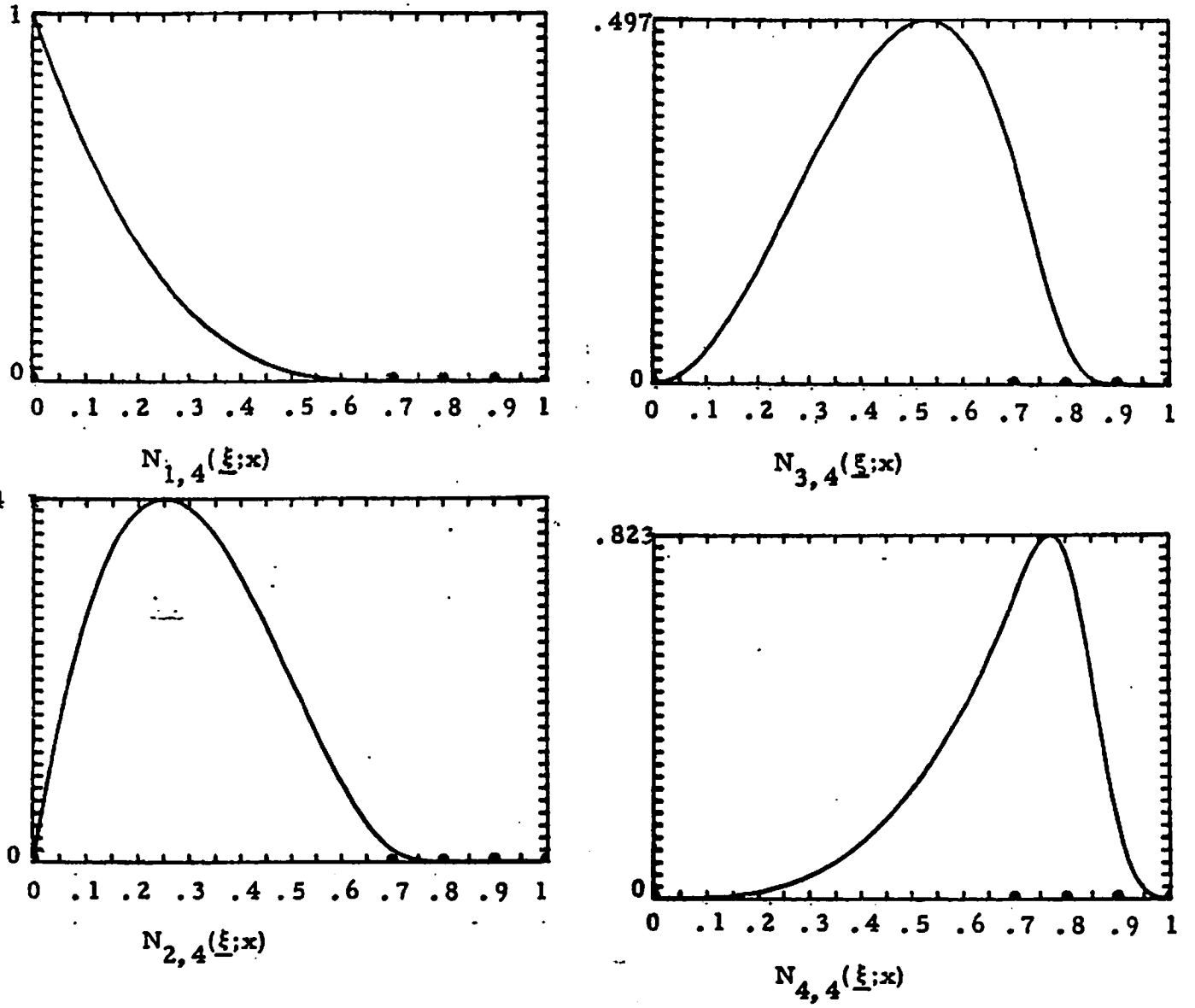
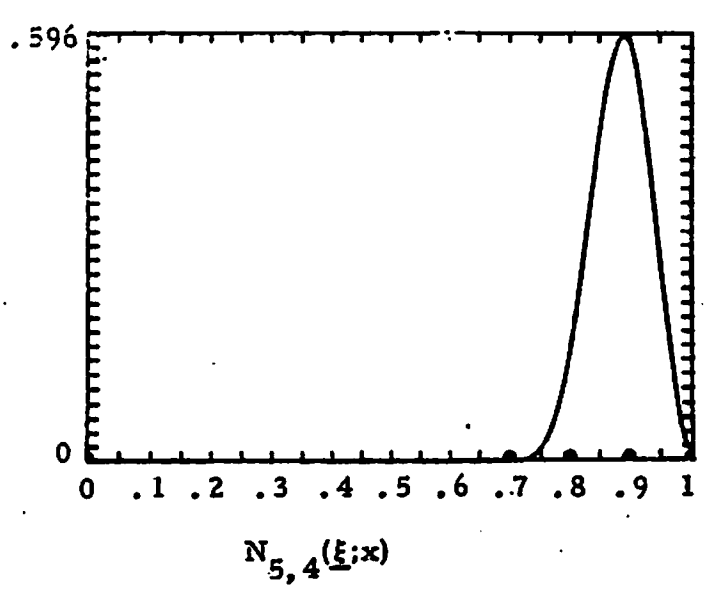
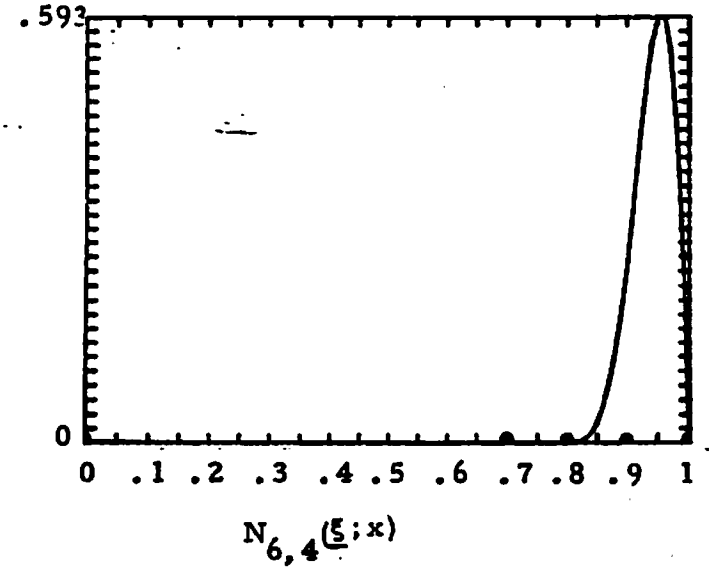


Figure 2 Normalized 4th Order B-Splines for Knot Vector $\underline{\xi}_2 = (0,0,0,0,.7,.8,.9,1,1,1,1)$

Variable Knots





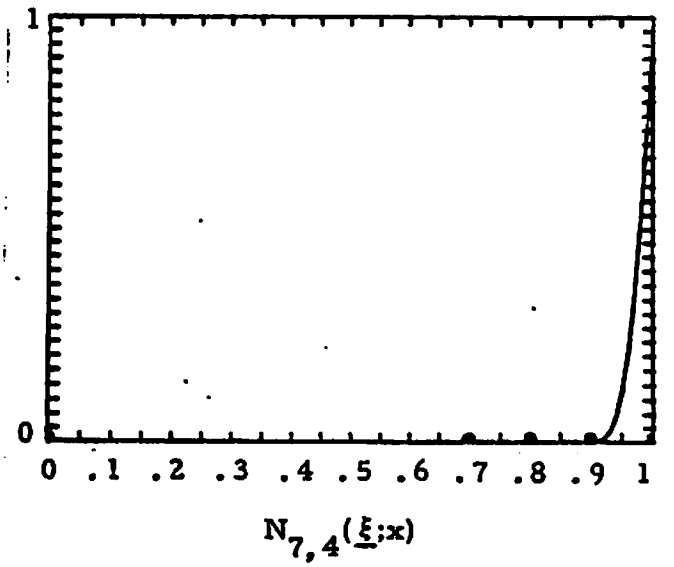


Figure 2 (continued) Normalized 4th Order B-Splines for Knot Vector $\underline{\xi}_2 = (0,0,0,0,.7,.8,.9,1,1,1,1)$

Variable Knots

Figure 1 serves to illustrate the relationship between multiple knots and the differentiability of the normalized B-splines. Note that $N_{1,4}(\underline{\xi}_1;x)$ involves a fourth order knot at $x = 0$ so that $k - (p+1)$ is equal to -1 and $N_{1,4}(\underline{\xi}_1;x)$ possesses a simple jump at $x = 0$. The third order knot at $x = 0$ for $N_{2,4}(\underline{\xi}_1;x)$ results in $k - (p+1)$ equalling 0 and from figure 1 it is clear that $N_{2,4}(\underline{\xi}_1;x)$ is merely continuous at $x = 0$. The second and first order knots at $x = 0$ for $N_{3,4}(\underline{\xi}_1;x)$ and $N_{4,4}(\underline{\xi}_1;x)$ respectively result in $N_{3,4}(\underline{\xi}_1;x)$ being once continuously differentiable and $N_{4,4}(\underline{\xi}_1;x)$ being twice continuously differentiable at $x = 0$. The same sequence of events is true for $N_{5,4}(\underline{\xi}_1;x)$, $N_{6,4}(\underline{\xi}_1;x)$ and $N_{7,4}(\underline{\xi}_1;x)$ at $x = 1$. Thus the polynomials in figures 1 and 2 become the non-orthogonal basis functions for our approximation methods. Thus finding the best approximation to $f(x,y)$ over the x and y knot sets can be stated as follows:

Minimize: $$\left[\iint_{\mathcal{S}} |f(x,y) - S_{k,N_x N_y}(x,y)|^2 \, dxdy\right]^{1/2}$$

over all possible x and y knot vectors $\{\xi_1,\ldots,\xi_{N_x}\}$, $\{\eta_1,\ldots,\eta_{N_y}\}$ subject to:

$$|\xi_i| \leq 1 \qquad \forall\, i = 1,\ldots,N_x$$
$$|\eta_j| \leq 1 \qquad \forall\, j = 1,\ldots,N_y$$

This is nothing more than nonlinear minimization over the possible knot sets in $\mathcal{S}$, the solution of which has been shown to exist by Rice [2].

Thus specifying an error ϵ, we can find the epsilon degrees of freedom by a sequence of minimizations decreasing the number of knots in the x and y directions until we reach a point at which the error will be exceeded with any further restriction in the number of knots.

For this to make sense we must be assured that for every $\epsilon > 0$ there exists N_x and N_y such that

$$\| f(x,y) - S_{k,N_x N_y}(x,y) \|_2 < \epsilon$$

and that a minimum over the knot sets exists. Addressing this convergence problem Schultz [3] has shown for k = 4 that

$$\| f(x,y) - S_{N_x N_y}(x,y) \|_2 \le \left(\frac{\bar{\rho}}{\pi}\right)^4 \left\{ \left\| \frac{\partial^4}{\partial x^4} f(x,y) \right\|_2 + \left\| \frac{\partial^2}{\partial x^2} \frac{\partial^2}{\partial y^2} f(x,y) \right\|_2 + \left\| \frac{\partial^4}{\partial x^4} f(x,y) \right\|_2 \right\} \tag{3}$$

where $\rho = \max\{\max(\xi_{i+1} - \xi_i),\ \max_j(\eta_{j+1} - \eta_j)\}$. Thus taking

$$N_x \ge \frac{2}{\bar{\rho}},\ N_y \ge \frac{2}{\bar{\rho}} .$$

We have as $N_x N_y \to \infty$ such that $\bar{\rho} \to 0$, the bicubic spline approximation will converge to $f(x,y)$ in an L_2 sense over the unit rectangle $\mathcal{S}$.

III. Experimental Results for Spline Approximation

The purpose of this section is to present some numerical results concerning the "epsilon degrees of freedom" concepts developed in the preceding sections. This was developed as an approximation problem whose solution was seen to involve the determination of a sequence of best approximating (in an L_2 sense) B-splines with variable knots. While the determination at each step of such a best approximating spline is simply a nonlinear minimization problem over the knots defining the spline, it is computationally infeasible. Thus we must follow DeBoor [4] and settle for spline approximations with good if not optimal knot placements. In what follows, two easily

implemented methods for placing the knots will be given that can result in a significant error reduction over the uniform knot case for the proper class of images.

Here the possibility of subsectioning the image and using different knot densities in each of the subsection is investigated. This method might provide fruitful results when one considers an L_2 error bound given by Schultz [5] for cubic splines. Recalling that the error is given by the L_2 norm, $\|\cdot\|_2$ of the difference between the function and its approximation, this bound is given by

$$\|f-\hat{f}\|_2 \leq 4\left(\frac{\bar{\rho}}{\pi}\right)^4 \left\{ \left\| \frac{\partial^4}{\partial x^4} f(x,y) \right\|_2 + \left\| \frac{\partial^2}{\partial x^2} \frac{\partial^2}{\partial y^2} f(x,y) \right\|_2 + \left\| \frac{\partial^4}{\partial y^4} f(x,y) \right\|_2 \right. \tag{4}$$

where

$$\bar{\rho} = \max \left[\max (\xi_{i+1} - \xi_i),\ \max (\eta_{j+1} - \eta_j) \right] \tag{5}$$

From previous discussions it follows that if the image derivative energy is large only over a small region, then using a uniform knot bicubic spline with knot mesh width equalling $\bar{\rho}$ given by eq.(5) should result in an overly good approximation of the image in those regions. Thus we should be able to obtain reasonable results by employing a different bicubic spline with uniformly spaced knots in each subsection, the knot density in each subsection being proportional to the value of

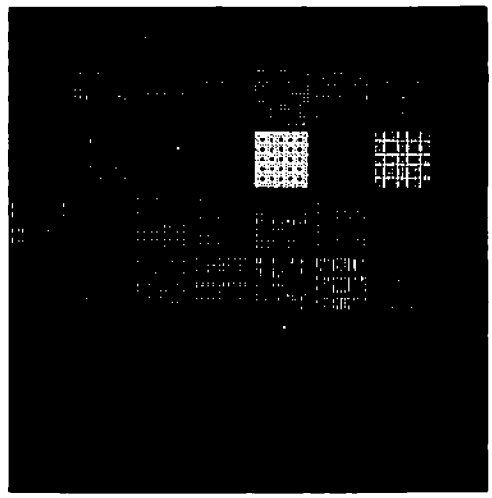
$$\left\| \frac{\partial^4}{\partial x^4} f(x,y) \right\|_2 + \left\| \frac{\partial^2}{\partial x^2} \frac{\partial^2}{\partial y^2} f(x,y) \right\|_2 + \left\| \frac{\partial^4}{\partial y^4} f(x,y) \right\|_2$$

in that subsection.

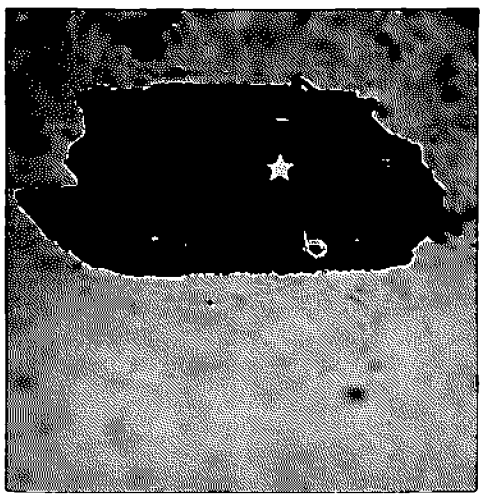
To explore this subpartitioning method a series of bicubic approximations involving image subsections of different sizes was run of the APC image, an aerial reconnaissance image and an image of Los Angeles International Airport (LAX). For this series the image dimension N was taken to be 256 and three subsection sizes of 32 x 32, 16 x 16 and 8 x 8 pixels were used for both images. For each subpartition size three knot density ranges were employed. In all cases the maximum knot density is taken to be such that the matrices of normalized B-splines are nonsingular thus resulting in the image being interpolated in at least one subsection. The lowest knot density in each subpartition sequence was taken to be that corresponding to a fourth order knot placed at each of the subregion boundaries for the x and y knot vectors respectively. The number of knots was then increased by raising the minimum number of knots employed. The results for the APC image along with the associated knot densities for subpartitions of size 32 x 32, 16 x 16 and 8 x 8 are shown in figures 3, 4 and 5 respectively. Here the fourth order knots at the subpartition boundaries are not displayed for aesthetic purposes. Note that all of the approximations are quite good and that the knot densities are quite adaptive for each subpartition size. Note that the error for the 32 x 32 case corresponding to the data reduction ratio of 5.62:1 is lower than the error for the 16 x 16 case for the 5.31:1 data reduction ratio. This would seem to indicate that for low error levels the 32 x 32 partition size is better than the 16 x 16 size partition which itself is better than the 8 x 8 case.

This same sequence was obtained for the reconnaissance image and the results along with the corresponding knot density patterns are shown in figures 6, 7 and 8 for the 32 x 32, 16 x 16 and 8 x 8 cases respectively. Note again

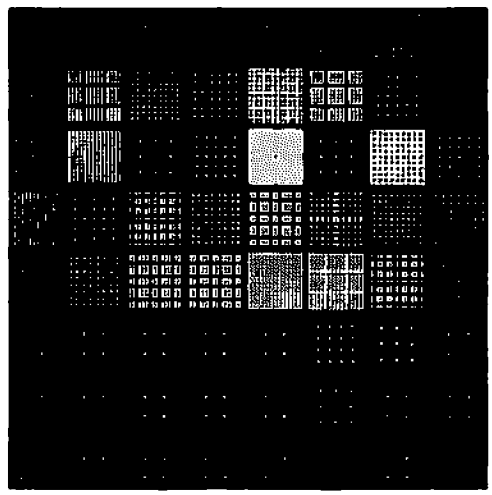
Parameter Reduction = 9.5:1



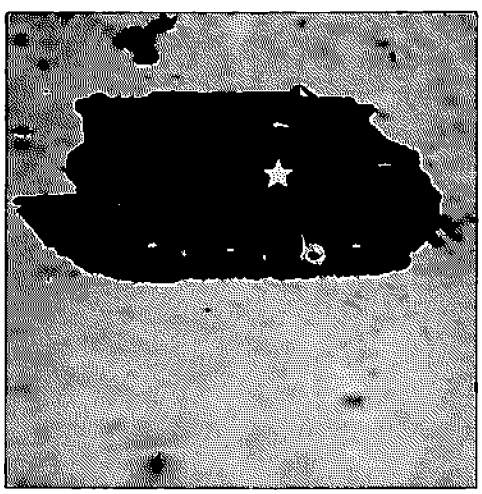
MSE = .38%



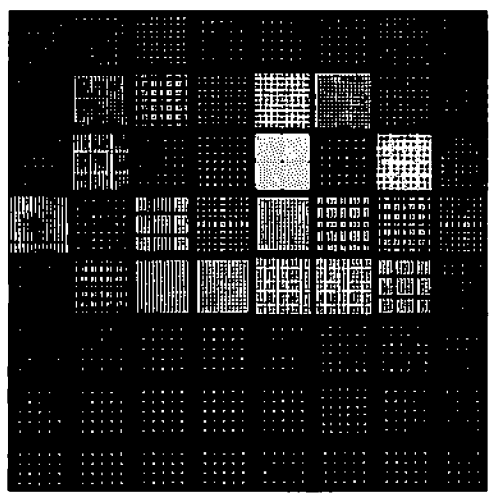
Parameter Reduction = 7.74:1



MSE = .32%

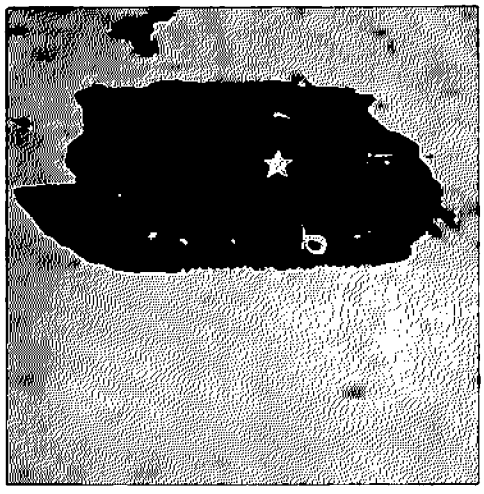


Parameter Reduction = 5.62:1

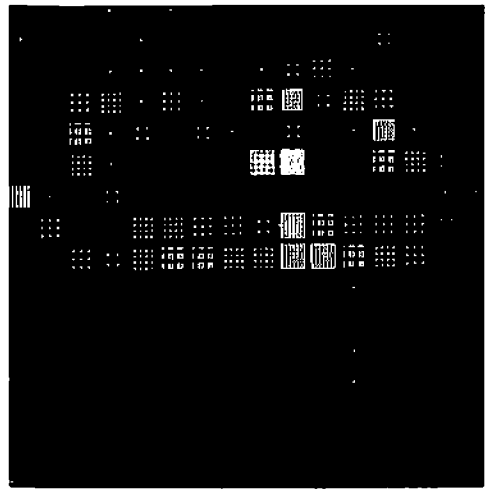


MSE = .25%

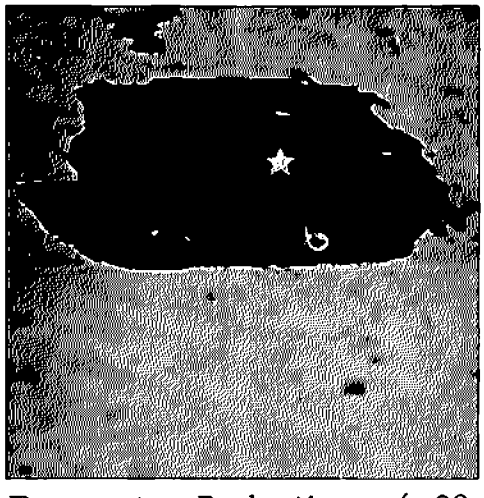
Figure 3. Bicubic Spline Reconstructions and Associated Knot Densities for an APC Photograph Using Subregions of Size 32 by 32.



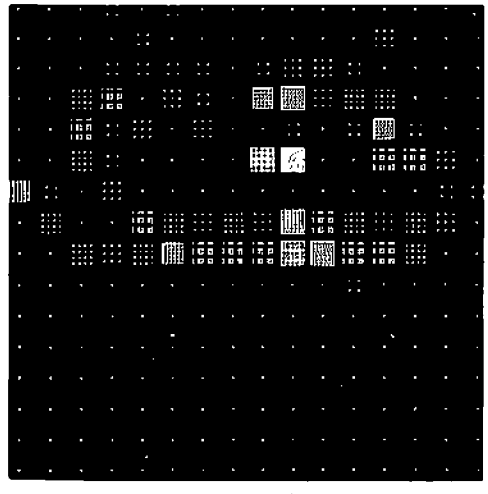
Parameter Reduction = 9.26:1



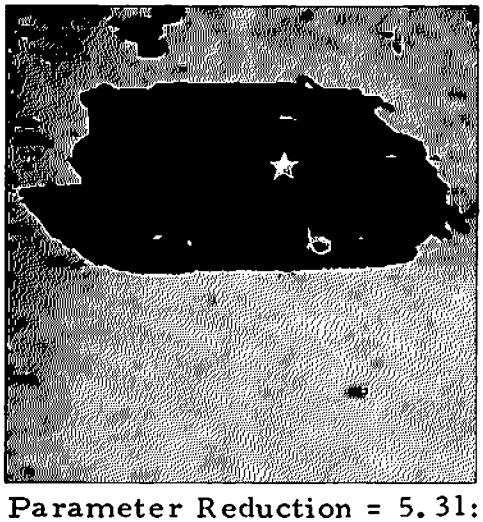
MSE = .37%



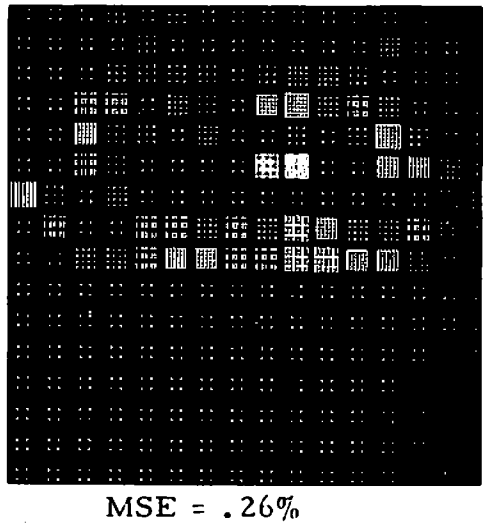
Parameter Reduction = 6.98:1



MSE = .31%

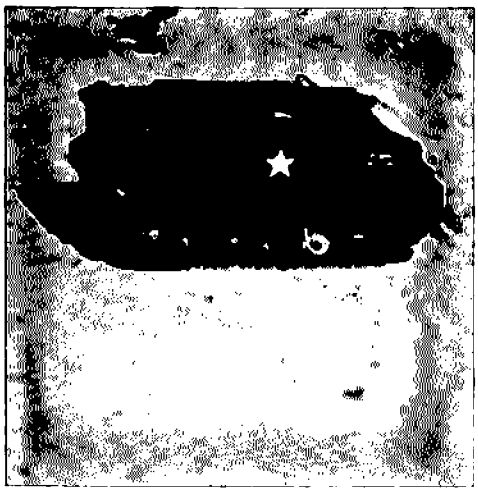


Parameter Reduction = 5.31:1

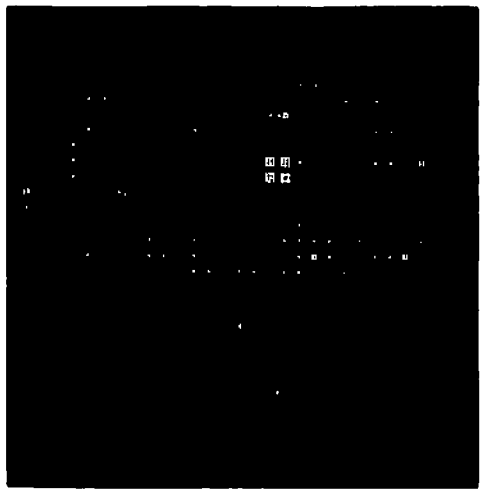


MSE = .26%

Figure 4. Bicubic Spline Reconstructions and Associated Knot Densities for an APC Photograph Using Subregions of Size 16 by 16.

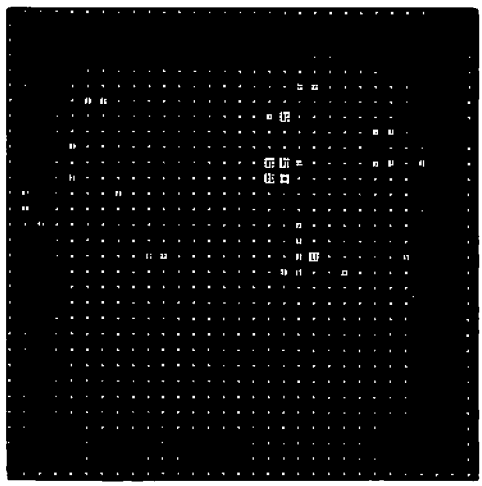


Parameter Reduction = 3.62:1

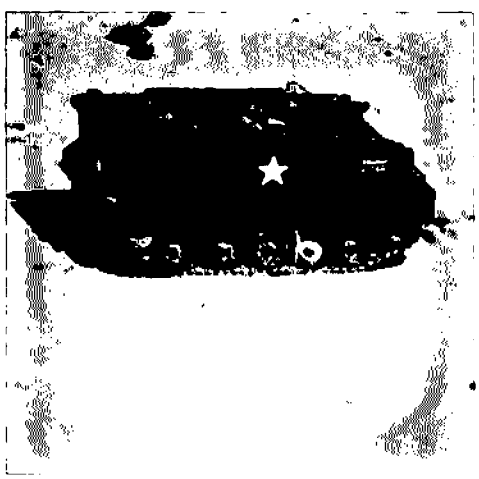


MSE = .24%

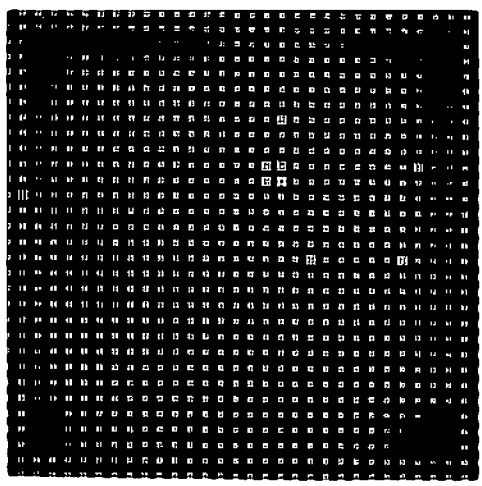
Parameter Reduction = 2.42:1



MSE = .17%



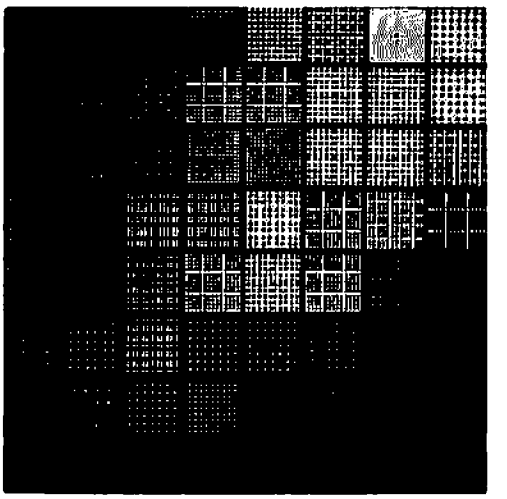
Parameter Reduction = 1.72:1



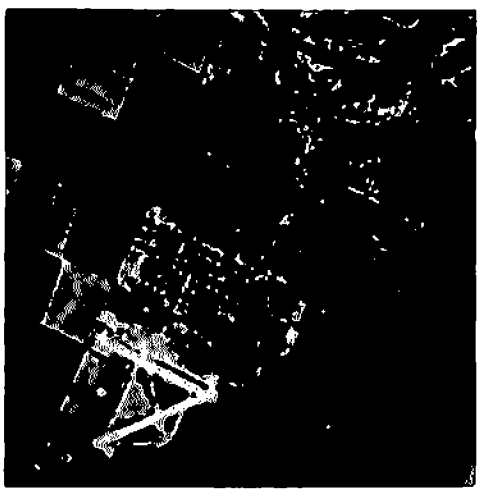
MSE = .1%

Figure 5. Bicubic Spline Reconstructions and Associated Knot Densities for an APC Photograph Using Subregions of Size 8 by 8.

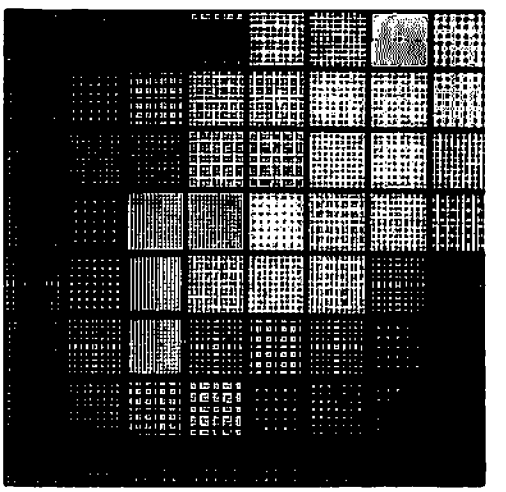
Parameter Reduction = 4.69:1



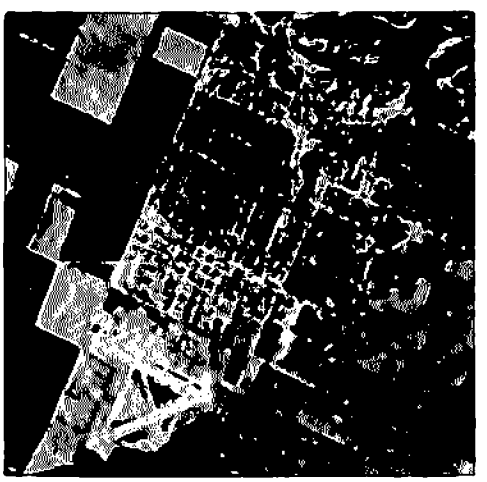
MSE = 1.4%



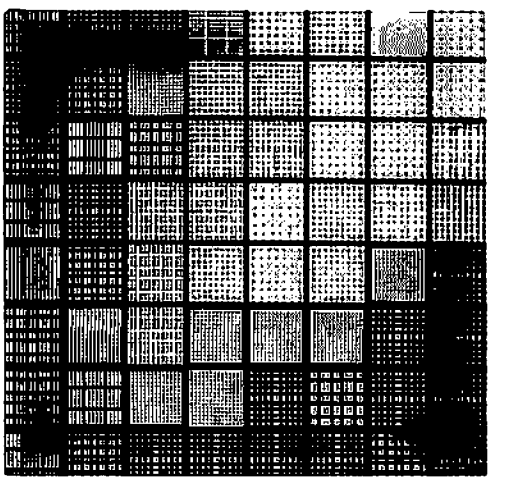
Parameter Reduction = 3.46:1



MSE = .63%



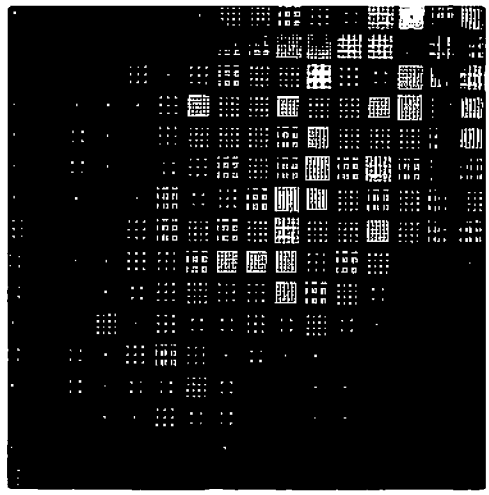
Parameter Reduction = 2.56:1



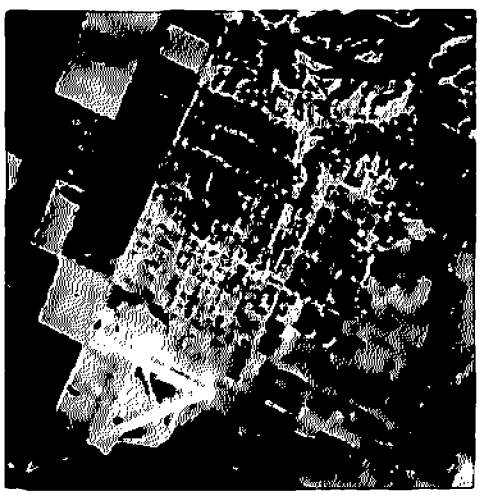
MSE = .38%

Figure 6. Bicubic Spline Reconstructions and Associated Knot Densities for a Reconnaissance Photograph Using Subregions of Size 32 by 32.

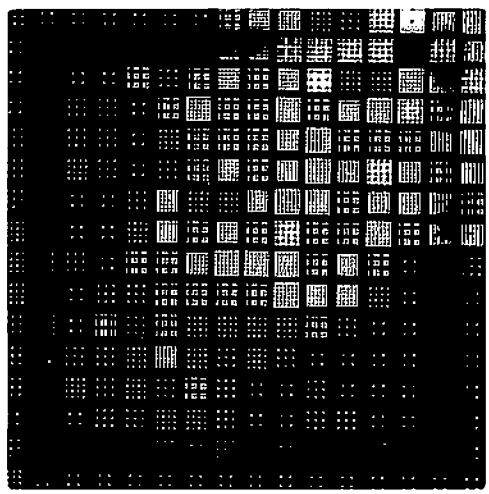
Parameter Reduction = 5.24:1



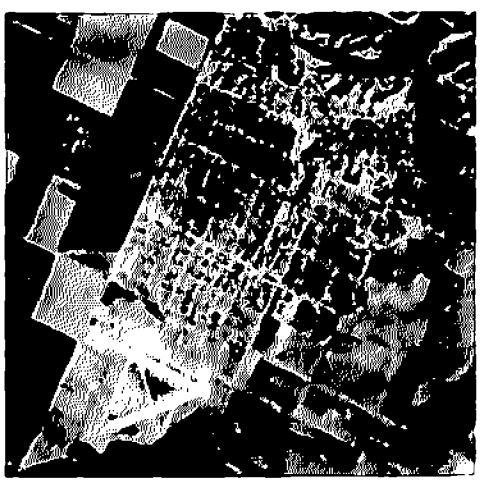
MSE = 1.2%



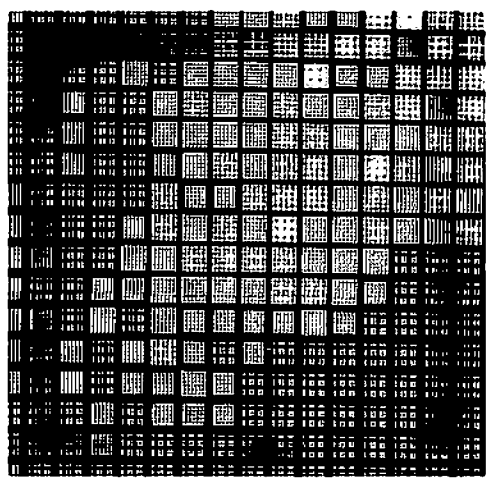
Parameter Reduction = 3.71:1



MSE = .70%



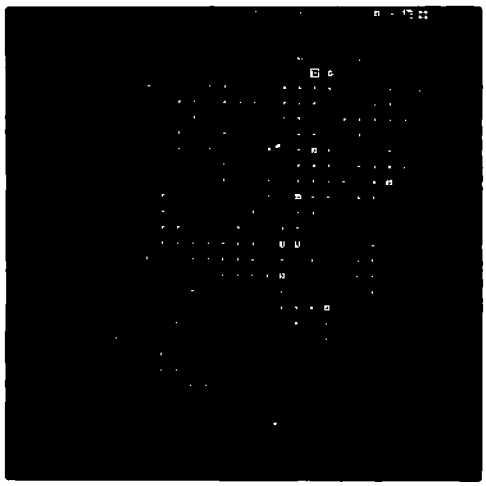
Parameter Reduction = 2.36:1



MSE = .39%

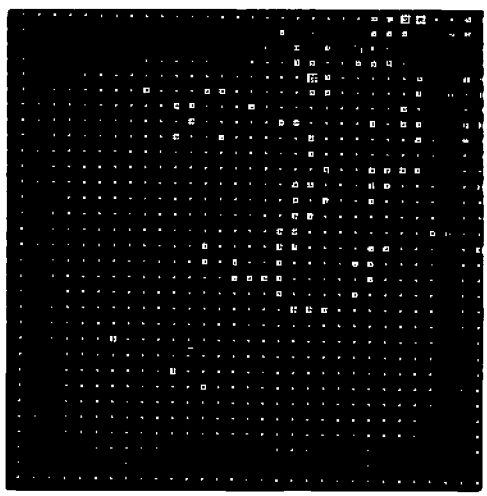
Figure 7. Bicubic Spline Reconstructions and Associated Knot Densities for a Reconnaissance Photograph Using Subregions of Size 16 by 16.

Parameter Reduction = 3.31:1



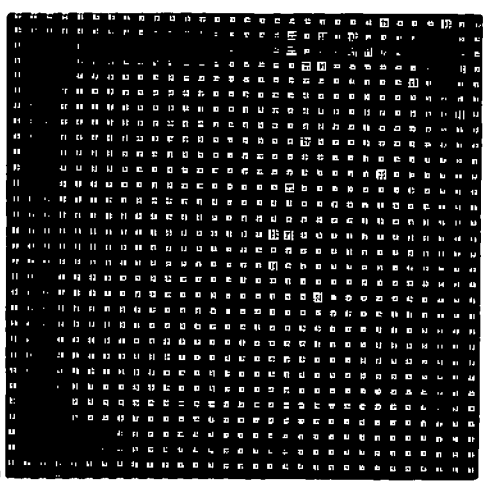
MSE = .77%

Parameter Reduction = 2.35:1



MSE = .49%

Parameter Reduction = 1.71:1



MSE = .31%

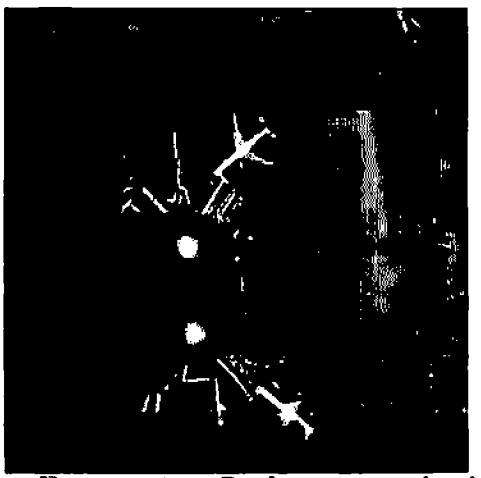
Figure 8. Bicubic Spline Reconstructions and Associated Knot Densities for a Reconnaissance Photograph Using Subregions of Size 8 by 8.

that the best reconstruction for the 32 x 32 case is at a lower error than the 16 x 16 case, and at a higher corresponding data reduction ratio. This would again seem to indicate that for low error levels the 32 x 32 size partition is better than the 16 x 16 partition.

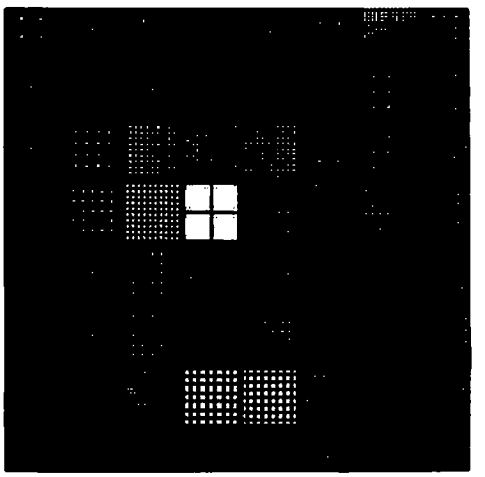
The sequence for the image of LAX is contained in figures 9, 10 and 11. Note here that while the errors are relatively high the results are quite good visually and that the knots serve fairly well in locating the areas containing the aircraft. Note also that the best results occur for the 8 x 8 subpartitioning case. This is most likely due to the relatively small size of the items of interest - in this case the aircraft.

IV. Summary and Conclusions

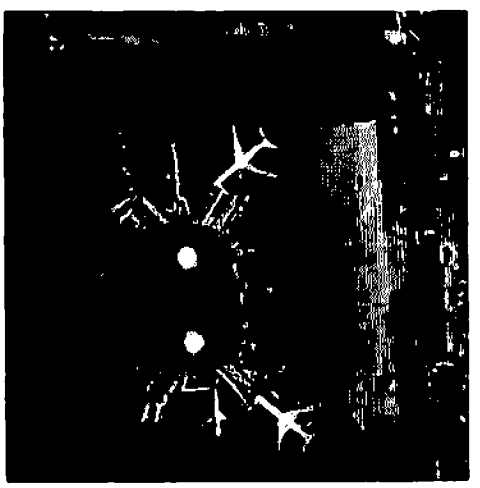
In this report the attempt has been to demonstrate the utility of variable knot splines in achieving a data reduction and quantifying the degrees of freedom of sample images by the number of variable knot bicubic splines necessary to approximate a particular image at an error level, epsilon. In dealing with the image itself the degrees of freedom was approached as an approximation problem where the degrees of freedom at a level epsilon was taken to be the minimum number of functions needed to approximate the image with an error epsilon. Since this minimum is difficult to find, the functions used were cubic splines with variable knots. By dividing the image into subregions a significant data reduction was achieved with reasonable errors. It was found that the number of knots and thus the degrees of freedom was higher in regions of higher image derivative energy than in those regions where the image was relatively constant. By subsectioning the image and employing a different spline approximation in each



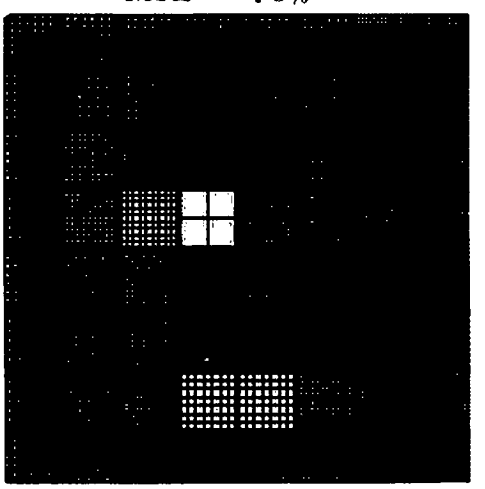
Parameter Reduction 9.19:1



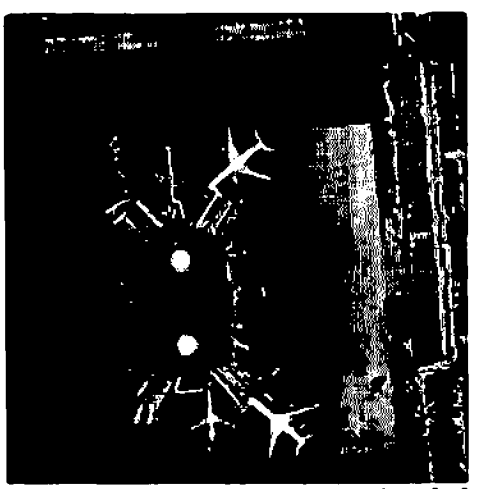
MSE = 2.8%



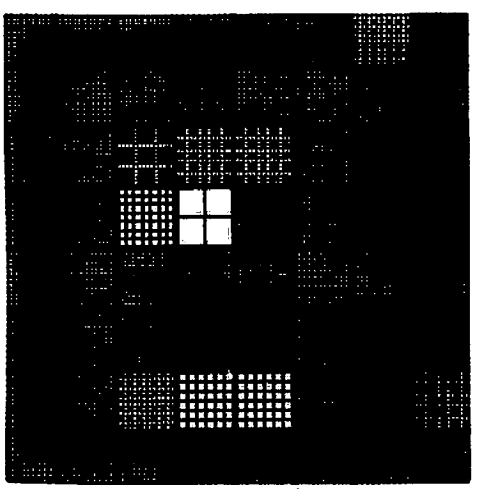
Parameter Reduction 5.48:1



MSE = 1.9%

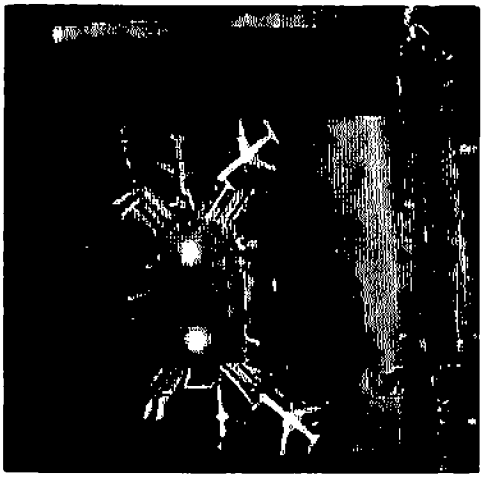


Parameter Reduction 3.41:1

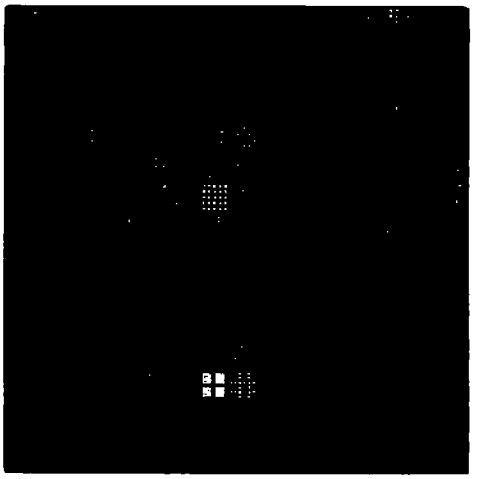


MSE = 1.3%

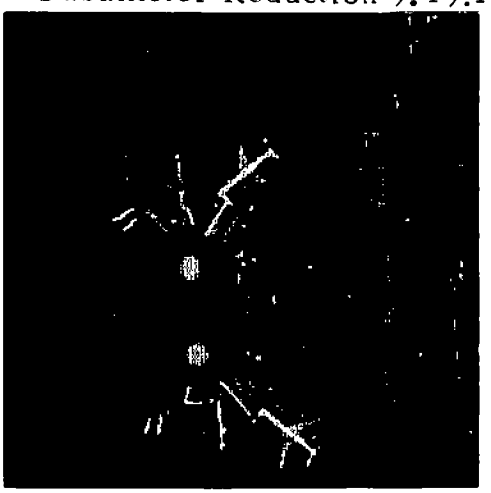
Figure 9. Bicubic Spline Reconstructions and Associated Knot Densities for LAX Photograph Using Subregions of Size 32 x 32.



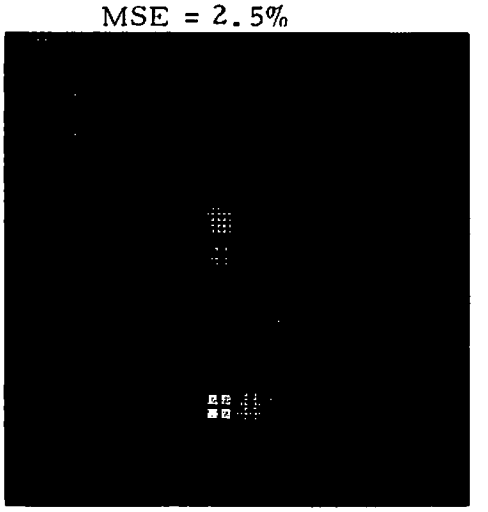
Parameter Reduction 9.19:1



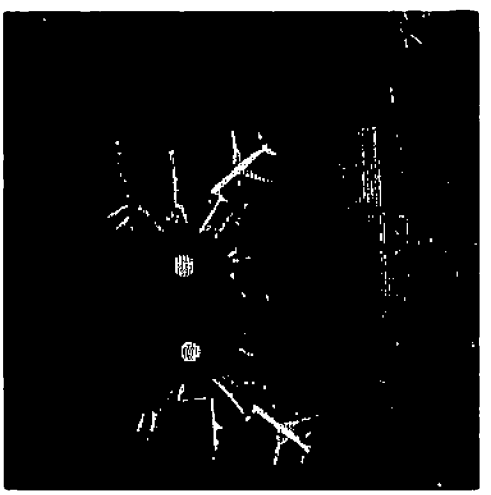
MSE = 2.5%



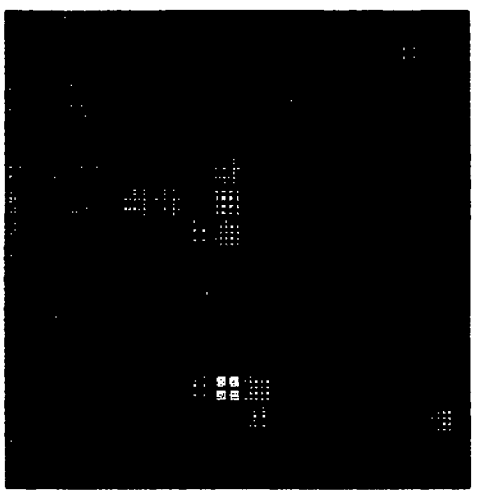
Parameter Reduction 5.35:1



MSE = 1.8%



Parameter Reduction 2.82:1



MSE = 1.1%

Figure 10. Bicubic Spline Reconstructions and Associated Knot Densities for LAX Photograph Using Subregions of Size 16 x 16.

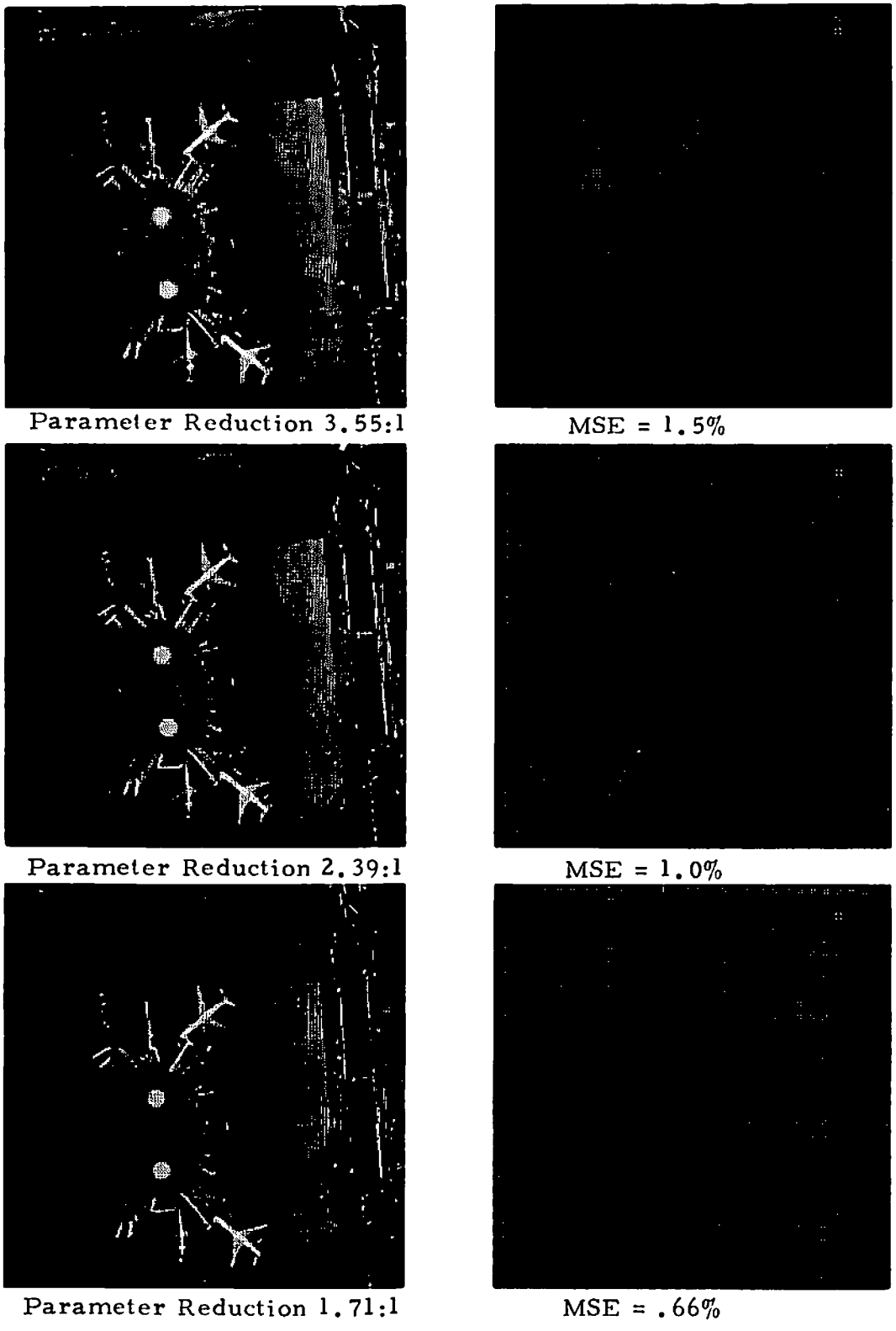


Figure 11. Bicubic Spline Reconstructions and Associated Knot Densities for LAX Photograph Using Subregions of Size 8 x 8.

subsection whose knot density was dependent on the fourth difference energy in that region, good results were obtained. A high degree of adaptability was in evidence through the knot density patterns with acceptable errors being obtained at reasonable data reduction ratios.

Finally it should be said that in effect this report represents an attempt to bridge the gap between the continuous domain upon which images are defined and the discrete grids upon which they are sampled and defined for analysis by digital techniques.

References

[1] C. DeBoor, "On Calculating with B-Splines," Journal of Approximation Theory, Vol. 6, 1972, pp. 50-62.

[2] J.R. Rice, The Approximation of Functions - Vol. II, Addison-Wesley, Reading, Massachusetts, 1969.

[3] M.H. Schultz, Spline Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[4] C. DeBoor, "Good Approximations by Splines with Variable Knots," from Spline Functions and Approximation Theory, A. Meir and A. Sharma, eds., Birkhauser Verlag Basel and Stuttgart, 1973.

[5] M.H. Schultz, Spline Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

3.2 Image Filtering Based on Psychophysical Characteristics of the Human Visual System

Charles F. Hall

In the past decade many physiological and psychophysical experiments which give some insight to the fundamental characteristics of the human visual system (HVS) have been performed [1]. These experiments indicate that the model shown in figure 1 is a good approximation for the HVS. The nonlinearity is primarily a result of the photon to electrical energy conversion which takes place in the photoreceptors of the retina. Recent results indicate that a cube root power law may be more appropriate than the logarithm, however, for this work the logarithmic nonlinearity was used. The low-pass filter is a consequence of the optics of the eye including the size of the pupil opening and structure of the retinal mosaic. Lateral inhibition (due to receptor output interconnectivity) produces the high-pass filter. Stockham has demonstrated the utility of a similar model in image processing [2].

The general shape of the entire filter function of the HVS can be viewed directly from the image shown in figure 2. The vertical bars in the figure are decreasing in contrast from the bottom to the top and increasing in spatial frequency from left to right. Both variations are logarithmic. When this figure is viewed one should perceive an outline of the HVS's modulation transfer function (MTF). The outline is formed by the points at which the contrast becomes too low to distinguish the bars. The central peak usually occurs at about six cycles/degree of visual field subtended.

Mannos and Sakrison have used the model of figure 1 in a coding experiment [3]. They found that a system containing a linear function of

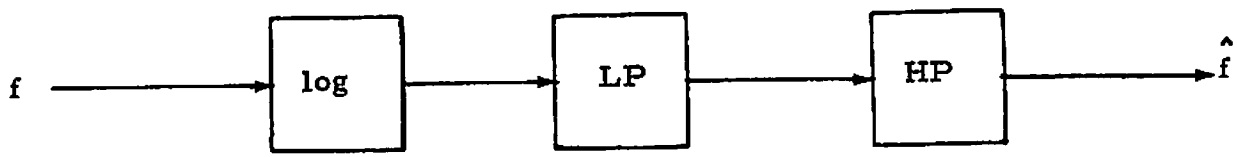


Figure 1. Spatial model for human visual system.

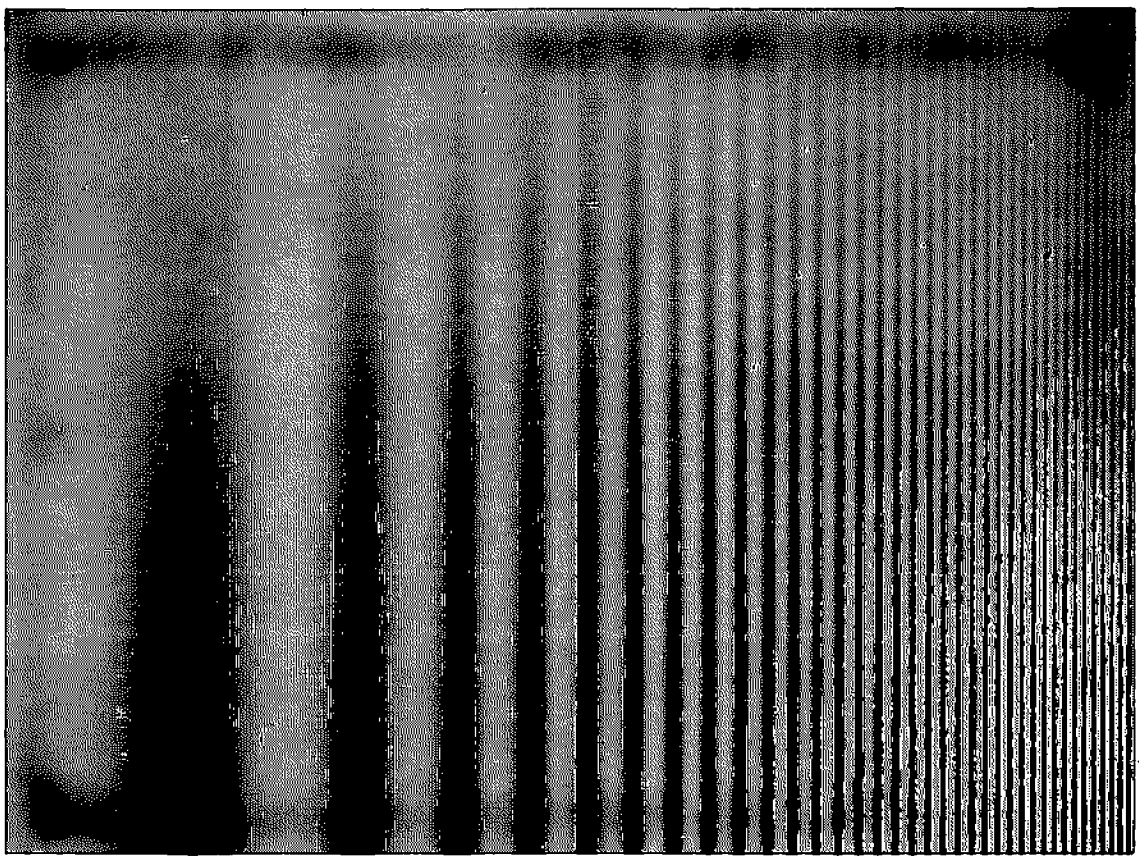


Figure 2. Sinewave spatial grating

$$A(fr) = 2.6[0.0192 + 0.114f_r] \exp[-(0.114fr)^{1.1}] \quad (1)$$

where f_r is radial frequency in cycles/degree, gave the best results when judged subjectively. This linear function peaks at eight cycles/degree. An isotropic representation of $A(f_r)$ is shown in figure 3. The nonlinearity and low frequency roll-off will reduce the intensity variations as noted by Stockham. Since $A(f_r)$ peaks at a frequency above that of the HVS, some edge enhancement should occur. The high frequency roll-off occurs at frequencies beyond the roll-off of the HVS and therefore should not affect the subjective quality of a filtered image.

In order to verify these statements two images were processed with the system shown in figure 4. Figure 5a contains the original image of a portion of L.A. International Airport (LAX). This image was generated from a 12 bit/pixel 256 x 256 array. Only eight bits/pixel have been displayed. The 12 bit/pixel version was processed and the eight bit/pixel result is shown in figure 5b. The wide dynamic range of the input intensity has been reduced and the details within the dark areas have been brought out at no expense to the lighter area details. In addition, all edges have been slightly enhanced.

The bridge scene shown in figure 6a was processed in the same way. This particular picture was digitized to only eight bits/pixel to begin with. The results are shown in figure 6b. Notice the shadow under the bridge has been reduced and stones on the bank are clearly visible.

Hall and Hall [1] have discussed the validity of the model in figure 1 and have shown the model in figure 7 to be

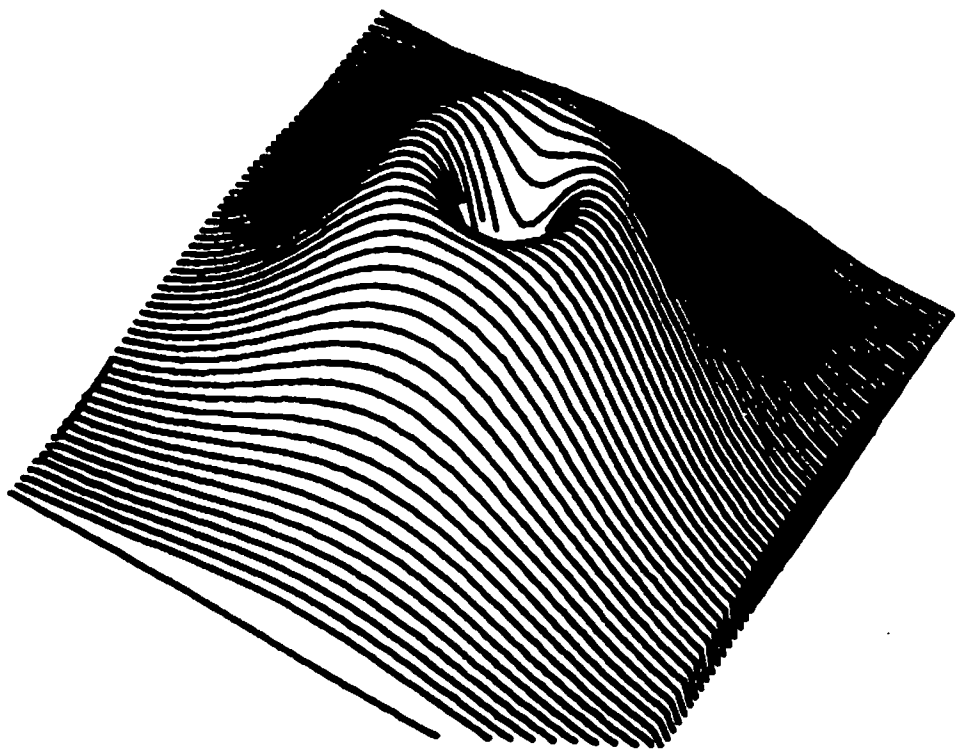


Figure 3. Isotropic filter function A(fr).

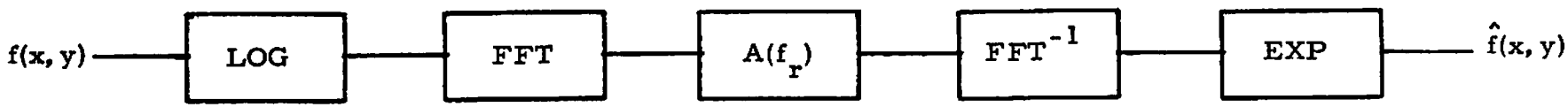
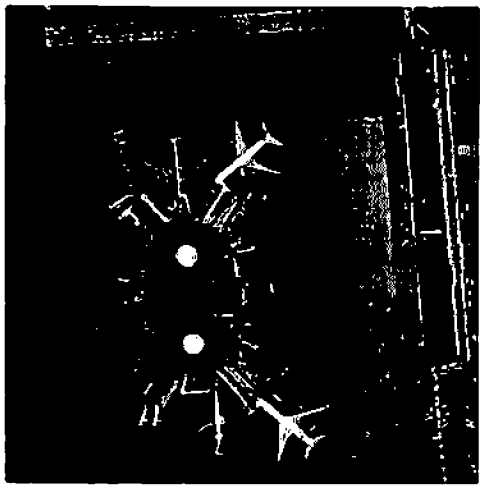
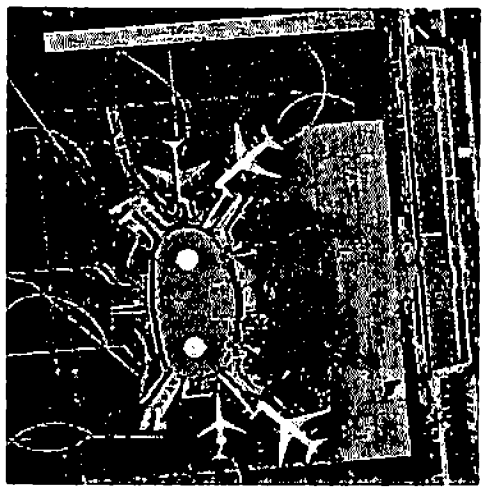


Figure 4. Processing system used to filter images.



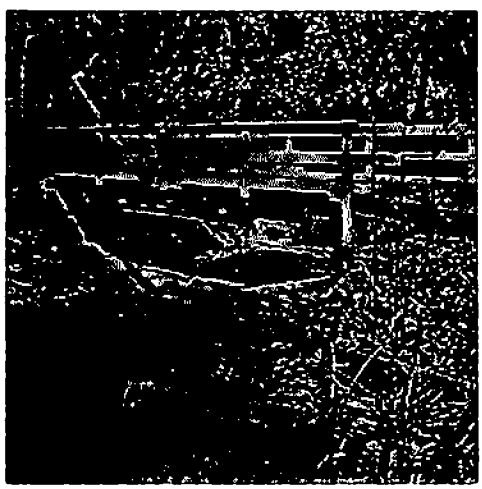
(a) Original



(b) Filtered by A(fr)

Figure 5. LAX scene

(a) Original



(b) Filtered by A(fr).

Figure 6. Bridge scene

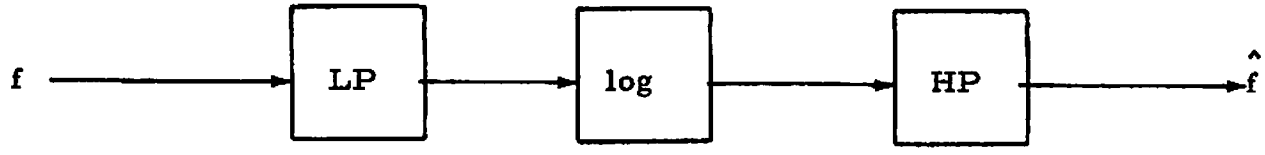
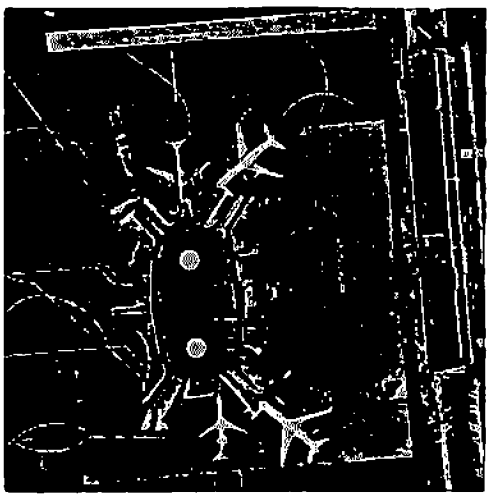
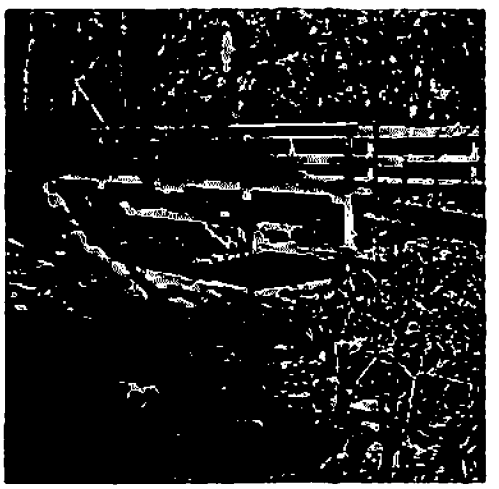


Figure 7. Hall model for H V S.



(a) LAX



(b) Bridge

Figure 8. Images processed by Hall model

more precise. This particular model was used to process the two images. The results are shown in figures 8a and 8b. Note that the contrast has been altered, as in the previous case, however the range is broader, i.e. there are more black and white points. In addition, since this model was based on the HVS properties, the processing produced an overall response which peaked at approximately six cycles/degree. As a result, the edges have not been enhanced as in the case of the model based on coding results. This difference is most apparent in the bridge scene. Another more subtle difference can be seen if one compares a high contrast edge (for example the light edge bordering the shadow under the bridge). The Hall model broadens this edge because the high spatial frequency response decreases as contrast increases [1].

References

[1] C.F. Hall and E.L. Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-7, No. 3, pp. 45-52, 1977.

[2] T.G. Stockham, Jr., "Image Processing in the Context of a Visual Model," Proceedings IEEE, Vol. 60, pp. 828-842, 1972.

[3] J.L. Mannos and D.J. Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images," IEEE Transactions on Information Theory, Vol. IT-20, pp. 525-536, 1974.

3.3 Optical Filters for Image Reconstruction

Alexander A. Sawchuk and Chung-Kai Hsueh

An incoherent system using a computer-plotted hologram as the spatial filter has been discussed in a recent report [1]. In the special situation when the hologram contains phase variations only, it is called a kinoform [2]. The kinoform is efficient in using the input light and the display covers the whole image field because it operates in the first diffraction order.

One problem with the kinoform is that it may not exist for a given impulse response. Iteration methods are used to obtain a kinoform which has a response very close to the desired one as discussed previously [1]. In fact, if we allow the kinoform to have a slow variation in amplitude as well as in phase, then a perfect impulse response can be obtained by setting the approximate response to the desired one and taking the Fourier transform to get the amplitude and phase of the "kinoform."

The "kinoform" with amplitude variation can be replaced by two kinoforms. A two-kinoform filtering system can achieve a general response by summing the responses from two separate systems, each of them with phase-only masks in the pupil as shown in figure 1. Suppose the desired transfer function $\underline{H}(f_x, f_y)$ at point (f_x, f_y) is $re^{j\theta}$. Here r and are functions of f_x, f_y and we drop these variables for simplicity. It can be shown (figure 2) that there exists exactly one pair of angles θ_1 and θ_2 such that

$$re^{j\theta} = \frac{1}{2} e^{j\theta_1} + \frac{1}{2} e^{j\theta_2} \qquad (1)$$

where

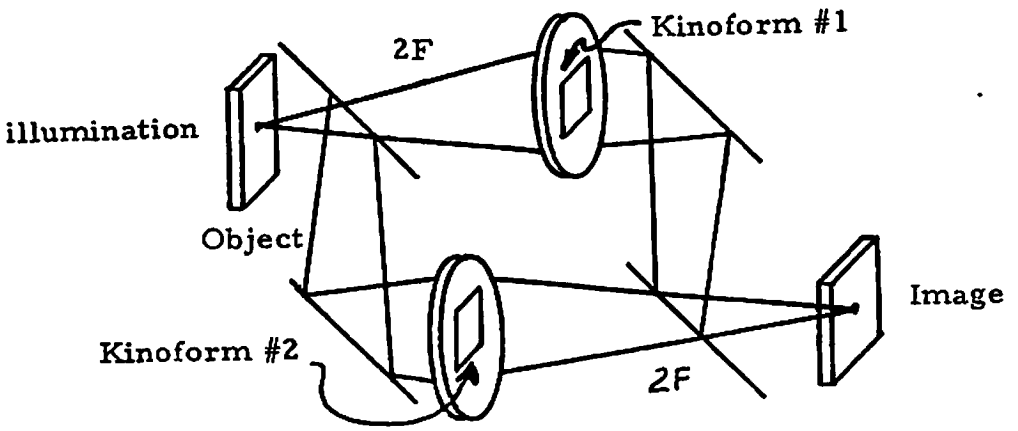


Figure 1. Two Kinoform Filtering System.

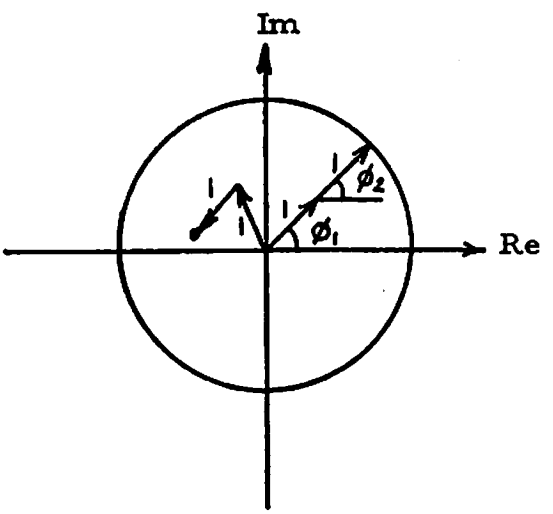


Figure 2. Complex Plane for Sum of Two Kinoform Functions.

$$\theta_1 = \theta - \frac{1}{2}\psi$$

$$\theta_2 = \theta + \frac{1}{2}\psi$$

and $\psi = \cos^{-1}(2r^2 - 1)$ has values between 0 and π. Thus the two-kinoform system can achieve any complex transmittance inside the fixed radius shown without the necessity of an amplitude part.

For the incoherent filtering system, $\underline{H}(f_x,f_y) = \mathcal{F}\{|h(x,y|e^{j\phi(x,y)}$, where $|h(x,y)|^2$ is the desired impulse response of the incoherent system and the phase $\phi(x,y)$ is arbitrary. Apparently, θ_1 and θ_2 need not be uniquely determined in this case due to the fact that $\phi(x,y)$ is uncertain. This gives us the flexibility in choosing θ_1 and θ_2 so that the resultant kinoforms are easier to plot. For the "kinoform" with slow amplitude variation discussed above, $r(f_x,f_y)$ is approximately constant. This implies that $\psi(f_x,f_y)$ and therefore $\theta_1(f_x,f_y)$ and $\theta_2(f_x,f_y)$ are smooth and are easy to plot by the computer-controlled microdensitometer.

One application of the above incoherent system is to give a continuous desampled output from the discrete spots on a CRT or other display devices. Two interesting impulse responses for achieving this operation are studied here. One is the cubic B-spline function and the other is the positive sampling reconstruction function [3]. The cubic B-spline function has been discussed [4,5] and applied to the interpolation of an image. However, only the digital interpolation has been considered. In this work, although simulation is done digitally, an optical filtering system will be used eventually. Comparing with the digital method, the optical system offers the advantages of inexpensive,

fast parallel processing of multidimensional signals.

One problem which arises in the optical incoherent system is that both input and impulse response have to be non-negative. The impulse response has been made to be non-negative by using one of the iteration methods. However, it is found that the coefficients in the cubic B-spline space may be negative, especially at rapid local transitions in the image. Fortunately, for a visual image there are not too many negative components and they can be eliminated by either discarding these components or by adding a constant to all of the coefficients to make them non-negative.

Another possible candidate in performing the desampling operation is the positive sampling reconstruction function. It is well known that a bandlimited function can be expanded in the form

$$f(x) = \sum_{n} f\left(\frac{n}{2W}\right) \frac{\sin[\pi(2Wx-n)]}{\pi(2Wx-n)} \tag{2}$$

where W is the highest frequency in the Fourier spectrum of f(x). Although it is a perfect desampling function for bandlimited signals, it has little application in the incoherent system due to the fact that the sinc function has negative lobes. Richards [4] has shown that eq.(2) can be reformulated as

$$f(x) = \sum_{n} (f_n - \alpha f_{n-1} - \alpha f_{n+1})\, g\,(2Wx-n) \tag{3}$$

where

$$f_n = f(\frac{n}{2W})$$

and g(z) is given by

$$g(z) = \frac{1}{\pi} \int_0^{\pi} \frac{\cos(zu)du}{1-2\alpha\cos u} \tag{4}$$

Equation (3) reduces to eq.(2) when $\alpha = 0$. For numerical evaluation eq.(4) can be evaluated as

$$g(z) = \begin{cases} \frac{\sin \pi z}{\sqrt{1-4\alpha^2}\,\pi z}\left[1 - 2z^2 \sum_{n=1}^{\infty} \frac{(-\beta)^n}{n^2 - z^2}\right] & z \neq \text{integer} \\ \\ \frac{(-\beta)^z \cos \pi z}{\sqrt{1-4\alpha^2}} & z = \text{integer} \end{cases} \tag{5}$$

where

$$\beta = (2\alpha)^{-1}[1-\sqrt{1-4\alpha^2}]$$

The results in figure 3 shown that g(z) is positive over most of the significant range provided that $\alpha > 0.34$. This is slightly different from the results by Richards in which he claimed that g(z) is positive if $\alpha > 0.2875$ approximately. The problem of negative components also occurs in the precombined function $f_n - \alpha f_{n-1} - \alpha f_{n+1}$. It is found that this precombined function generally contains more negative components than in the cubic B-spline case. The same remedies can be used to make the image non-negative.

An experiment has been conducted by using a 256 x 256 picture as the original (figure 4a). This original picture is averaged over a 4 x 4 block to simulate the effect of sampling a continuous picture and a 64 x 64 "sampled picture" is obtained as shown in figure 4b. The result of

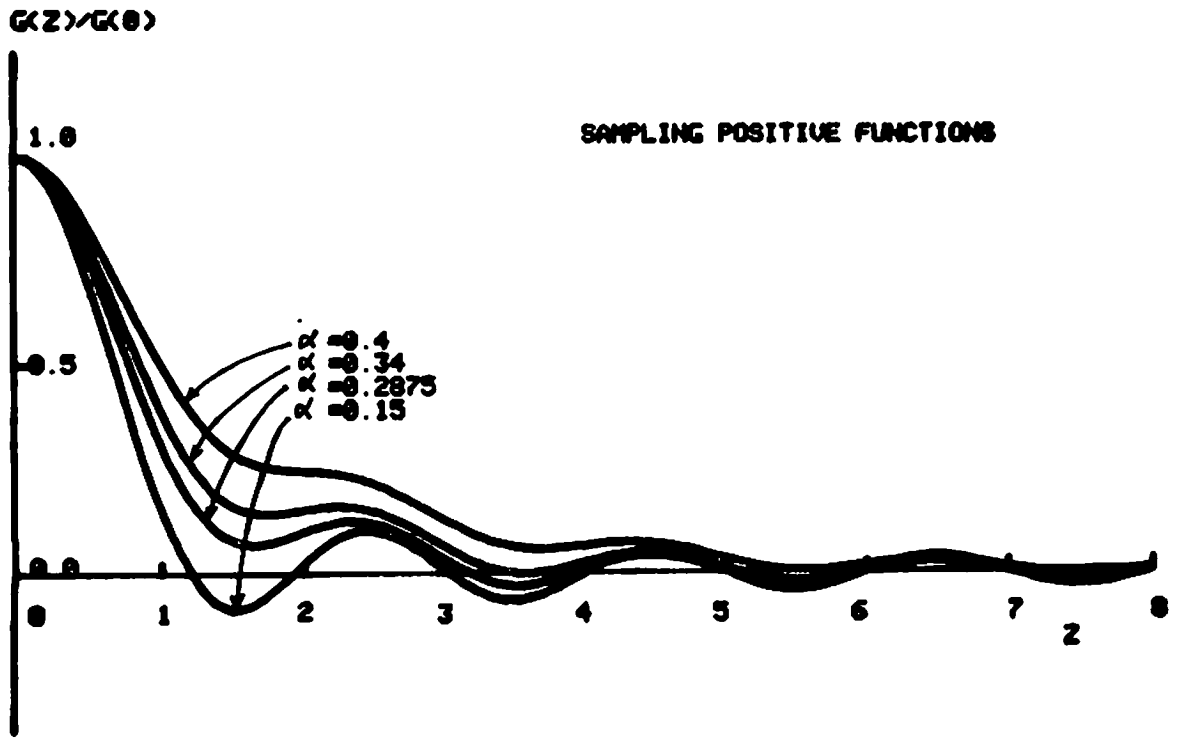
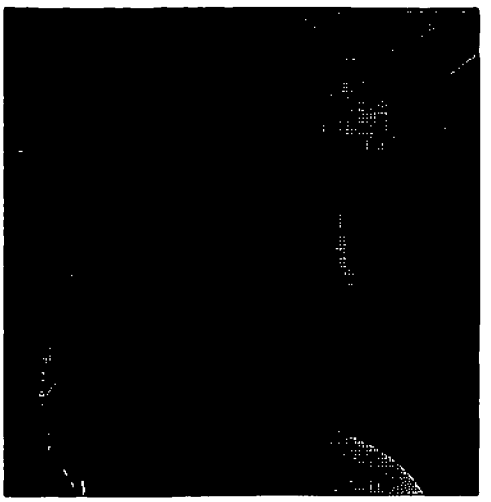


Figure 3. Sampling positive functions.

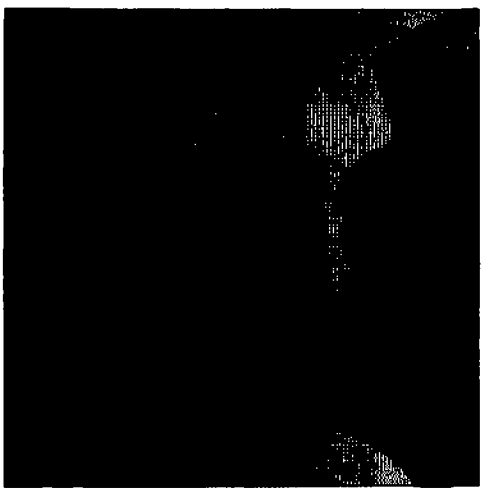


(a) (256 x 256)

(b) (64 x 64)

Figure 4. Original and Sampled Girl Picture

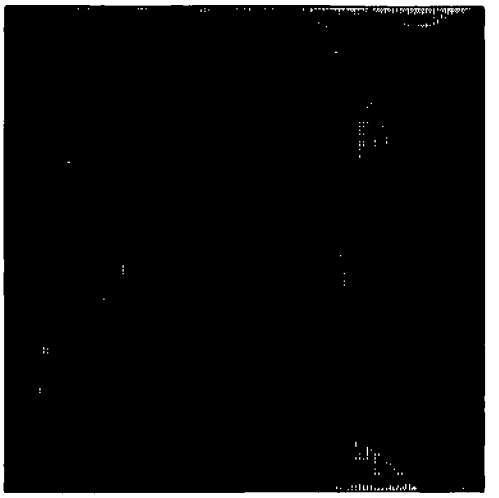
(c) Negative prints unaltered



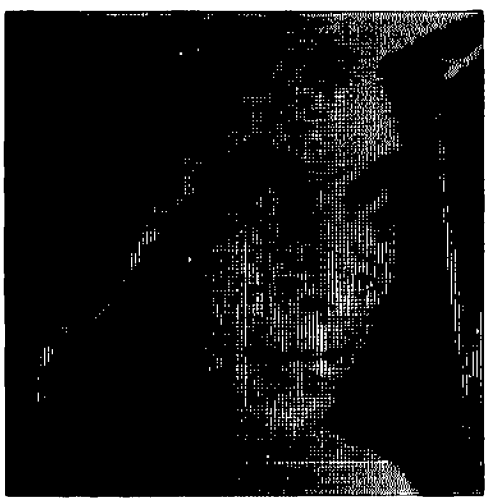
(d) Negative coefficients discarded

(e) Biased

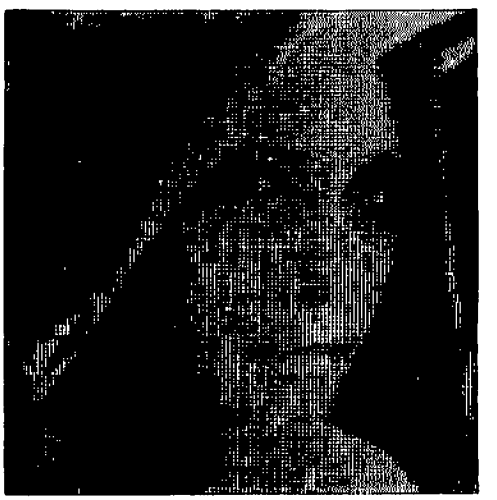
Figure 4 (cont'd) Reconstruction using a cubic B-spline function



(f) $\alpha = 0.2875$



(g) Negative coefficients discarded



(h) Negative coefficients discarded Impulse response modified to be non-negative

Figure 4 (cont'd) Reconstruction using a positive sampling function

cubic B-spline filtering is shown in figure 4c. In an incoherent system the input has to be non-negative and the result of figure 4c cannot be obtained exactly by an incoherent system. Modifications can be made by either setting the negative coefficients to zero or adding a constant to make the output non-negative. These results are shown in figures 4d and 4e respectively. Little degradation can be observed as compared with figure 4c. This indicates that optical implementation of the cubic B-spline filtering is possible.

Similar experiments have been done on the positive sampling function. Figure 4f shows the reconstruction by using the positive sampling function with $\alpha = 0.2875$. Although the results in figure 3 indicate that $g(z)$ is positive over most of the significant range when $\alpha > 0.34$, the computation results show that the number of negative components in the precombined image increases rapidly as α increases. Therefore no matter which modification is adopted, severe degradation would be expected for large α. One trade-off is to use smaller α and discard the small negative part in the positive sampling function. Figure 4g shows the result when negative components in the precombined image are set to be zero for $\alpha = 0.2875$ and figure 4h is the result when the negative part of the impulse response is also discarded. The oscillation occurring on the edges in figures 4f through 4h can be eliminated by applying a window function on the sampled data.

Future work will include making kinoforms by using a computer-controlled microdensitometer to produce the desired impulse responses and setting up the incoherent display system.

References

[1] C.-K. Hsueh and A.A. Sawchuk, "Phase Coding for Optical Image Processing," USCIPI Report 720, September 1976, pp. 133-145.

[2] J.C. Patau, L.B. Lesem, P.M. Hirsch and J.A. Jordan, Jr., "Incoherent Filtering Using Kinoforms," IBM J. Res. Develop., Vol. 14, 1970, pp. 485-491.

[3] P.I. Richards, "Sampling Positive Functions," Proceedings IEEE, Vol. 54, No. 1, January 1966, pp. 81-82.

[4] H.C. Andrews, "Digital Image Display Interpolation Techniques," USCIPI Report 560, March 1975, pp. 18-24.

[5] H.S. Hou, "Least Squares Image Restoration Using Spline Interpolation," USCIPI Report 650, February 1976.

3.4 A Technique of A Posteriori Restoration

John Morton

Review

Recall from the previous report [1] that this project is attempting to restore a blurred image with a minimum of a priori knowledge. Assumed as givens are

1) the blurred image itself,
2) the knowledge that the point-spread-function (PSF) is spatially invariant,

3) the extent of the PSF is small compared to the extent of the image,
4) and the image is not so severely blurred such that one cannot distinguish the general class to which the image belongs.

The general approach was outlined in the previous report [1] and will not be discussed herein.

Progress to Date

Assuming the image $f(x,y)$ is blurred by the spatially-invariant PSF $h(x,y)$, we have

$$g(x,y) = h(x,y) * f(x,y)$$

where $g(x,y)$ is the blurred image and '*' denotes convolution. In the Fourier domain we have

$$G(u,v) = H(u,v)\, F(u,v)$$

where $H(u,v)$ is termed the optical transfer function (OTF). Expressing the complex function $H(u,v)$ in magnitude and phase form, we have

$$H(u,v) = |H(u,v)|\, e^{j\theta(u,v)} .$$

This project can be divided into three tasks:

1) estimation of $|H(u,v)|$,
2) estimation of $\theta(u,v)$,
3) restoration of the blurred image given estimates of $|H(u,v)|$ and $\theta(u,v)$.

The computer programming for tasks (1) and (2) and for task (3) for an image of size 256 x 256 has been completed. The estimation of $|H(u,v)|$ is accomplished by techniques

developed by Cole [2] and Cannon [3] and the computer program is giving satisfactory results. In addition, assuming knowledge of H(u,v), the restoration program is giving satisfactory results. The estimate of $\theta(u,v)$ which is the emphasis of the project is encountering difficulties which are currently being investigated.

Let us consider the results of the following simulation. Shown in figure 1 is the undegraded image; figure 2 is the result of convolving figure 1 with a 5 x 5 matrix whose elements are of value 0.04.

Figure 3 is the Log of the estimate of the magnitude of the OTF via the method of Cole, while figure 4 is the Log of the estimate of the magnitude of the OTF via the method of Cannon. Although the known image was used to obtain $< F(u,v) >$ and $\theta(u,v)$ instead of prototypes as discussed in the previous report, it has been shown that the use of prototypes will also give satisfactory results [2-3]. Note that the results are fairly accurate except for the higher frequencies near the diagonals.

Figure 6 contains the results for the phase estimation for Path 1 whereas figure 7 contains the results for Path 2. From the previous report we have

$$\theta(u+\Delta u, v+\Delta v) \approx \theta(u,v) - \tan^{-1}\left\{\frac{R_G(u,v,\Delta u,\Delta v)\,|R_F(u,v,\Delta u,\Delta v|}{|R_G(u,v,\Delta u,\Delta v)\,|R_F(u,v,\Delta u,\Delta v)}\right\}.$$

where

Figure 1. Original image.

Figure 2. Degraded image.

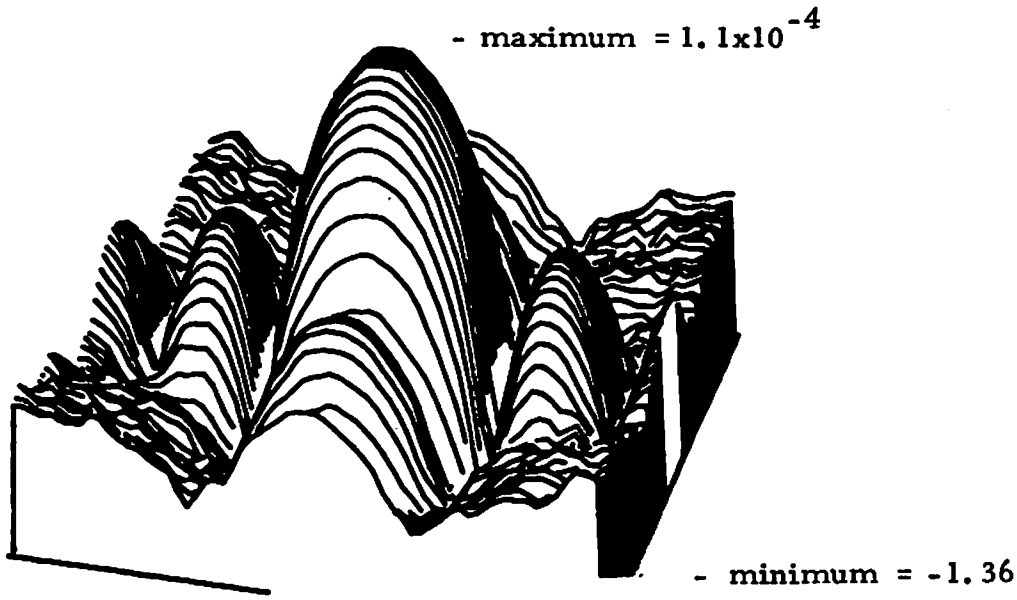


Figure 3. Log_{10} of the estimate of the magnitude of the OTF via the method of Cole. The known image was used instead of a prototype.

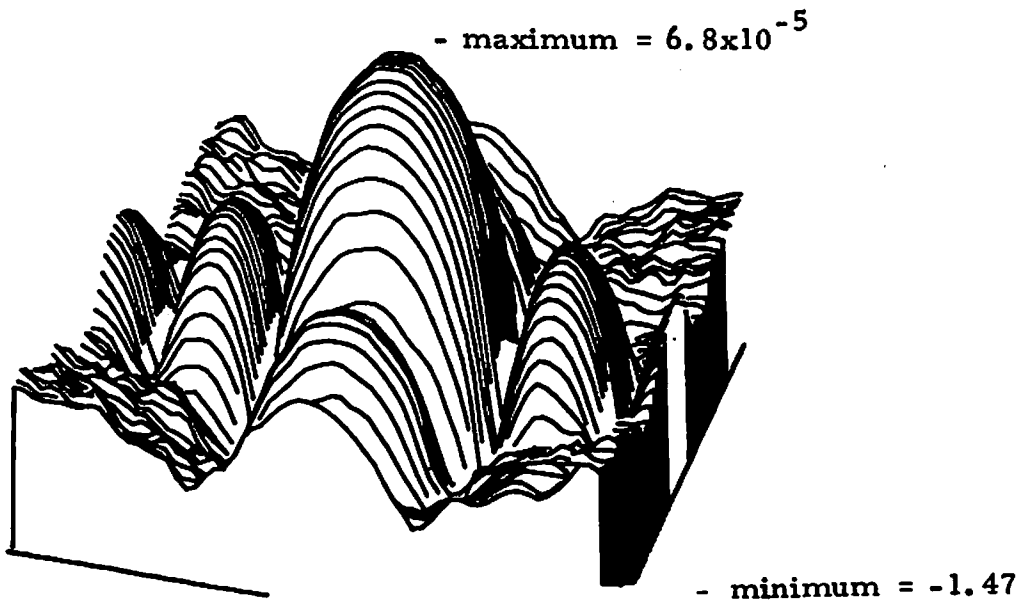


Figure 4. Log_{10} of the estimate of the magnitude of the OTF via the method of Cannon. The known image was used instead of a prototype.

$$R_G(u,v,\Delta u,\Delta v) = \sum_i G_i(u,v)\, G_i^*(u+\Delta u, v+\Delta v) \ .$$

$$R_F(u,v,\Delta u,\Delta v) = \sum_i F_i(u,v)\, F_i^*(u+\Delta u, v+\Delta v) \ .$$

Since $\theta(0,0) = 0.$, one may iterate along a great many paths. Path 1 and Path 2 are illustrated in figure 5 for the example $\theta(4,2)$.

Although the central lobes in figures 6 and 7 are essentially correct, the other values are more or less incorrect. For example plotting a cross-section of figure 6 along the u axis, we have figure 8. Note that the jumps occur approximately in the right places. However, each jump should be π , whereas, one jump is around 0.6 and the other is around 1.8. It is suspected that a windowing process is inhibiting the phase jumps from achieving their proper magnitudes.

The estimate of phase also used the known image for calculation of $R_F(u,v,\Delta u,\Delta v)$. This was to control the simulation better.

Regarding the use of prototypes to estimate $R_F(u,v,\Delta u,\Delta v)$, it was found that the phases of $R(u,v,\Delta u,\Delta v)$ for images in a prototype class were completely uncorrelated. For example given two images in a prototype class, the phases

$$\text{phase}\{R_{P_1}(u,v,\Delta u,\Delta v)\} \text{ and } \text{phase}\{R_{P_2}(u,v,\Delta u,\Delta v)\} \ .$$

where P_1 denotes one image of the prototype class and P_2 denotes another image in the prototype class, are uncorrelated. Nevertheless, preliminary results indicate that the phases of $R_F(u,v,1,0)$ and $R_F(u,v,0,1)$ do converge to 0. Note that $R_F(u,v,1,0)$ and $R_F(u,v,0,1)$ are the

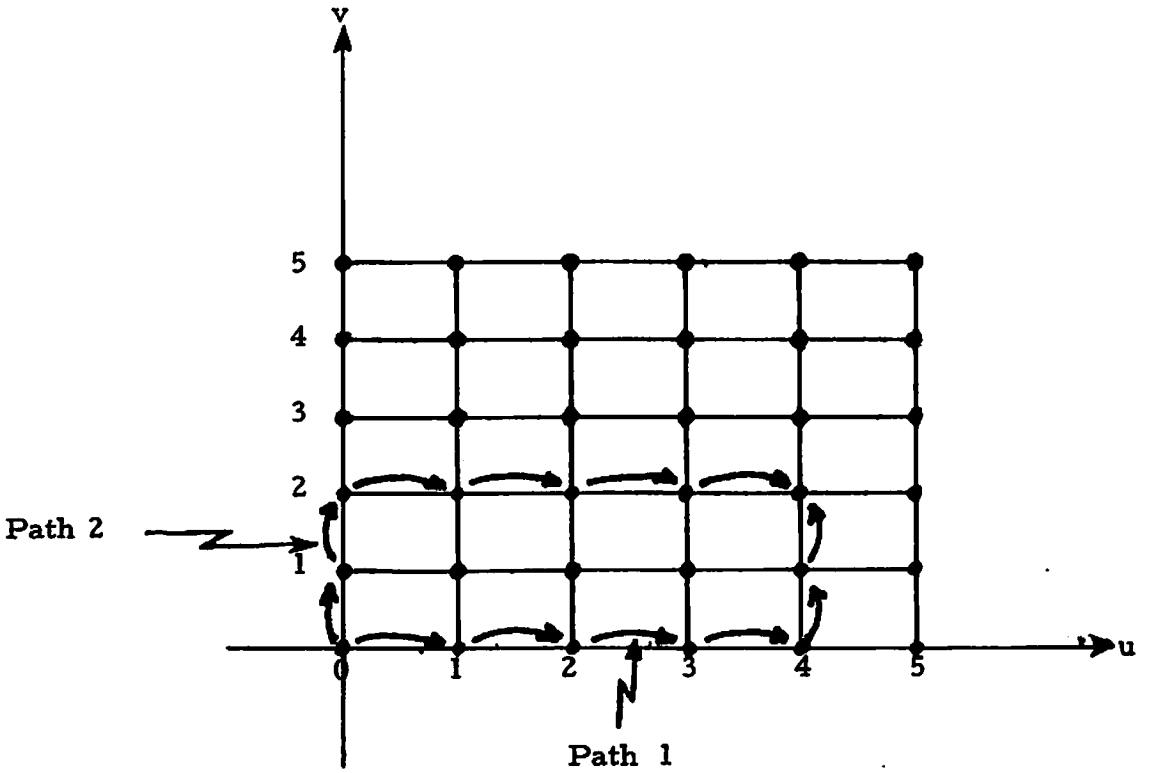


Figure 5. Illustration of two paths taken to interatively estimate $\theta(4, 2)$.

minimum value = -2.74

maximum value = 2.74

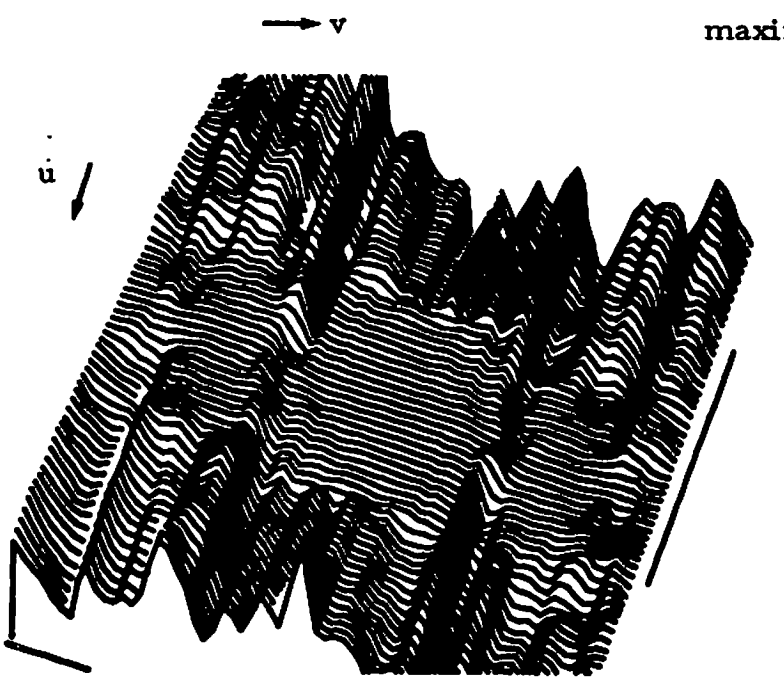


Figure 6. Estimate of OTF phase via Path 1. The known image was used to calculate $R_f(u, v, 1, 0)$.

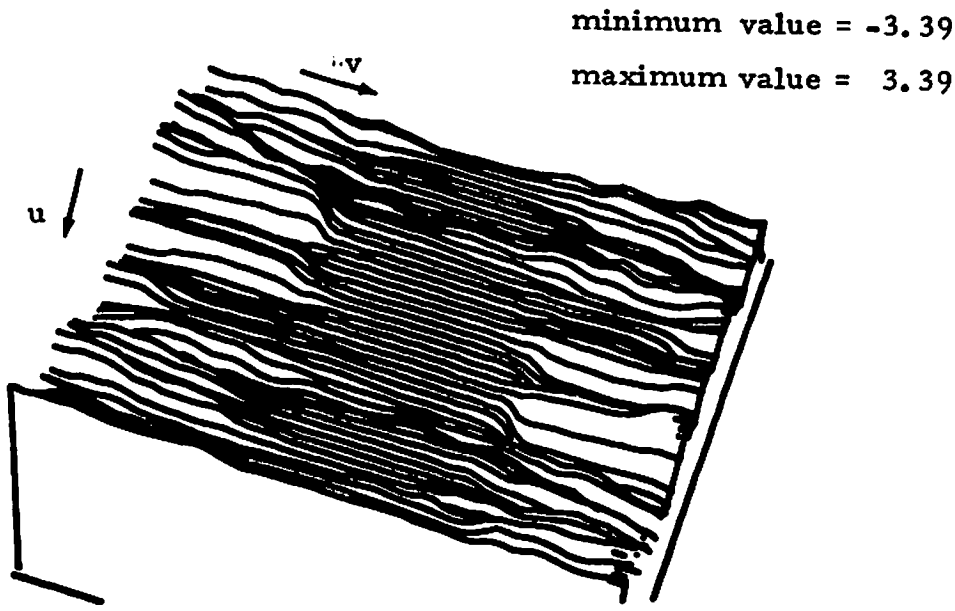


Figure 7. Estimate of OTF phase via Path 2. The known image was used to calculate $R_f(u, v, 0, 1)$

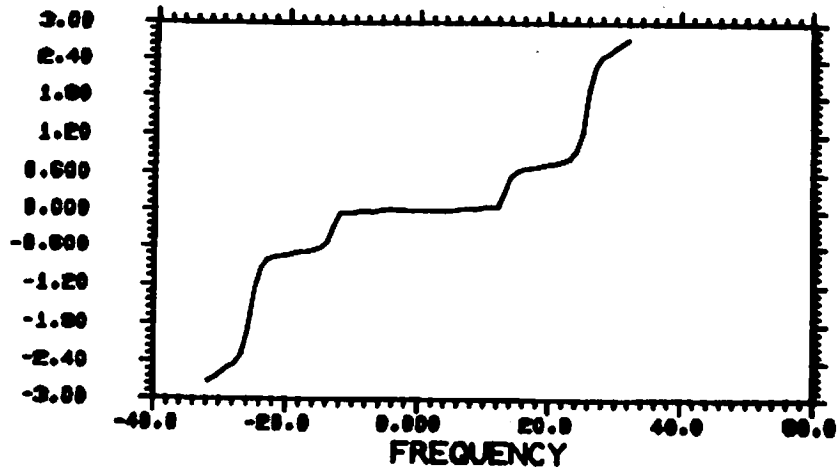


Figure 8. Cross-section along u-axis of Figure 6.

necessary quantities to traverse Path 1 and Path 2 respectively.

Shown in Table 1 are the standard deviations defined below in degrees for four similar images of text and the image in figure 1. Note that the operator E is assumed to refer to sample statistics opposed to mathematical expectation and the bar denotes sample mean. These images are all 512 x 512 images.

$$G_1^2 = \underset{uv}{E}[\, R_F(u,v,1,0) - \bar{R}_F(u,v,1,0)\,]^2.$$

$$G_1^2 = E[\, R_F(u,v,0,1) - \bar{R}_F(u,v,0,1)]^2.$$

$\bar{R}_F(u,v,1,0)$ and $\bar{R}_F(u,v,0,1)$ were essentially 0.

image	G_1	G_2
text1	4.8	4.9
text2	4.0	7.4
text3	7.2	6.4
text4	5.1	6.7
couple	4.2	3.7

Table 1. G_1 and G_2 in degrees for five different images.

Assuming the technique can tolerate the errors inherent in Table 1, the approximation below can be made.

$$\theta(u+\Delta u, v+\Delta v) \approx \theta(u,v) - \tan^{-1}\left\{ \frac{R_G(u,v,\Delta u,\Delta v)}{|R_G(u,v,\Delta u,\Delta v)|} \right\}$$

References

[1] Semiannual Technical Report, Harry C. Andrews - project

director, September 30, 1976, USCIPI Report 720.

[2] E.R. Cole, "The Removal of Unknown Image Blurs by Homomorphic Filtering," Department of Computer Science, University of Utah, ARPA Technical Report UTEC-CSc-74-029, June 1973.

[3] T.M. Cannon, "Digital Image Deblurring by Nonlinear Homomorphic Filtering," Department of Computer Science, University of Utah, ARPA Technical Report UTEC-CSc-74-091, August 1974.

3.5 Spatial Warp Interpretation Technique

William K. Pratt

Image interpretation consists of a description of a scene, or parts of a scene, based upon some symbolic scene representation. A new technique is described for image interpretation of segmented images containing perspective views of three-dimensional objects against a fixed background.

Roberts' Scene Analysis Method: The spatial warp image interpretation technique is an extension to complex, real world scenes of a method developed over a decade ago by L.G. Roberts [1]. In Roberts' scene analysis system scenes containing polyhedral objects are analyzed by matching polygonal parts of the scene to models of objects that might appear in the scene. This overall task is accomplished in several steps. First, edge detection is performed to determine points of convexity and concavity of objects within a test scene. These edge points are then linked together and short links are fit by straight line segments

using a priori knowledge that scene objects are of polyhedral form. Next, vertices between polygonal faces are detected and stored in a list.

A vertex of a polyhedral object is topologically invariant to translation, rotation, dilation, and changes in perspective. Such changes can be mapped in a four-dimensional coordinate system by a linear transformation. Let z,x,y,w denote a set of homogeneous components of an object point where z is depth, x is width, y is height, and w is a scale variable. These homogeneous components are related to the conventional three-dimensional components (X,Y,Z) of an object by the relations

$$X = \frac{x}{w} \tag{1a}$$

$$Y = \frac{y}{w} \tag{1b}$$

$$Z = \frac{z}{w} \tag{1c}$$

A vertex point can be represented as a vector $\underline{v}_1 = [z_1, x_1, y_1, w_1]^T$. After transformation the vector coordinate becomes

$$\underline{v}_2 = \underline{H}\,\underline{v}_1 \tag{2}$$

where $\underline{H}$ is a four-by-four transform matrix. Reference [1] provides several examples of the matrix structure for translation, rotation, dilation, and perspective movement.

The homogeneous coordinate system is a useful means of describing the movement of object points in a three-dimensional coordinate system in which depth information is available as well as spatial measurements. In a two-dimensional view of a scene containing three-dimensional objects depth measurements are not

directly available. Hence the z coordinate is unknown. Nevertheless, it is possible to "match" the vertices of three-dimensional objects defined by four component sets [z,x,y,w] by their three component projections [x,y,w] through a linear transformation. Consider a 4 x N matrix

$$\underline{A} = \left[\underline{a}_1 \vdots \underline{a}_2 \vdots \cdots \vdots \underline{a}_N\right] \tag{3}$$

containing N column vectors of vertices from a polyhedral object, and a 3 x N matrix

$$\underline{B} = \left[\underline{b}_1 \vdots \underline{b}_2 \vdots \cdots \vdots \underline{b}_N\right] \tag{4}$$

possessing column vectors of the two-dimensional scene projections of the same vertex points. A 3 x 4 transformation matrix $\underline{T}$ and an N x N scaling matrix $\underline{D}$ can then be found to minimize the transformation error

$$\underline{E} = \underline{T}\,\underline{A} - \underline{B}\,\underline{D} \tag{5}$$

Examination of the number of known and unknown elements of eq.(5) indicates that $N \geq 6$ is necessary to avoid an underdetermined solution. For an overdetermined system of equations the minimum mean square error solution assumes the form

$$\underline{T} = \underline{B}\,\underline{D}\,\underline{A}^T(\underline{A}\,\underline{A}^T)^{-1} \tag{6}$$

for a given scale matrix $\underline{D}$. The residual error can be measured by the matrix norm defined as

$$\mathcal{E} = \sum_{j=1}^{3}\sum_{k=1}^{N} E^2(j,k) \tag{7}$$

If the test points B match the model points A to sufficient accuracy, the corresponding object is deemed to be detected.

The basic limitations of the Roberts' scene analysis method is its dependence upon three-dimensional knowledge of visual models and its restriction to polyhedral objects. Attempts have been made to remove these limitations in the spatial warp interpretation technique.

Spatial Warp Image Interpretation: In the spatial warp interpolation system the three-dimensional nature of objects is not utilized explicitly. Rather, all scenes, test views and visual models, are considered as two-dimensional projections of objects related by a spatial coordinate warping transformation.

Figure 1 describes the basic methodology of spatial warping. A point (j,k) in an original image undergoes a physical spatial warp to be mapped into a distorted space at coordinate (p,q) according to the general relations

$$p = \mathcal{O}_p\{j,k\} \tag{8a}$$

$$q = \mathcal{O}_q\{j,k\} \tag{8b}$$

where $\mathcal{O}_p$ and $\mathcal{O}_q$ are arbitrary nonlinear operators. Next, a spatial warp correction is performed to compensate for the physical warp. If the physical warp tranformation of eq.(8) is known explicitly, then the correction inverse can be computed in principle. Usually, such information is not directly available, but it is often possible to approximate the physical warp by a polynomial transformation. As an example let

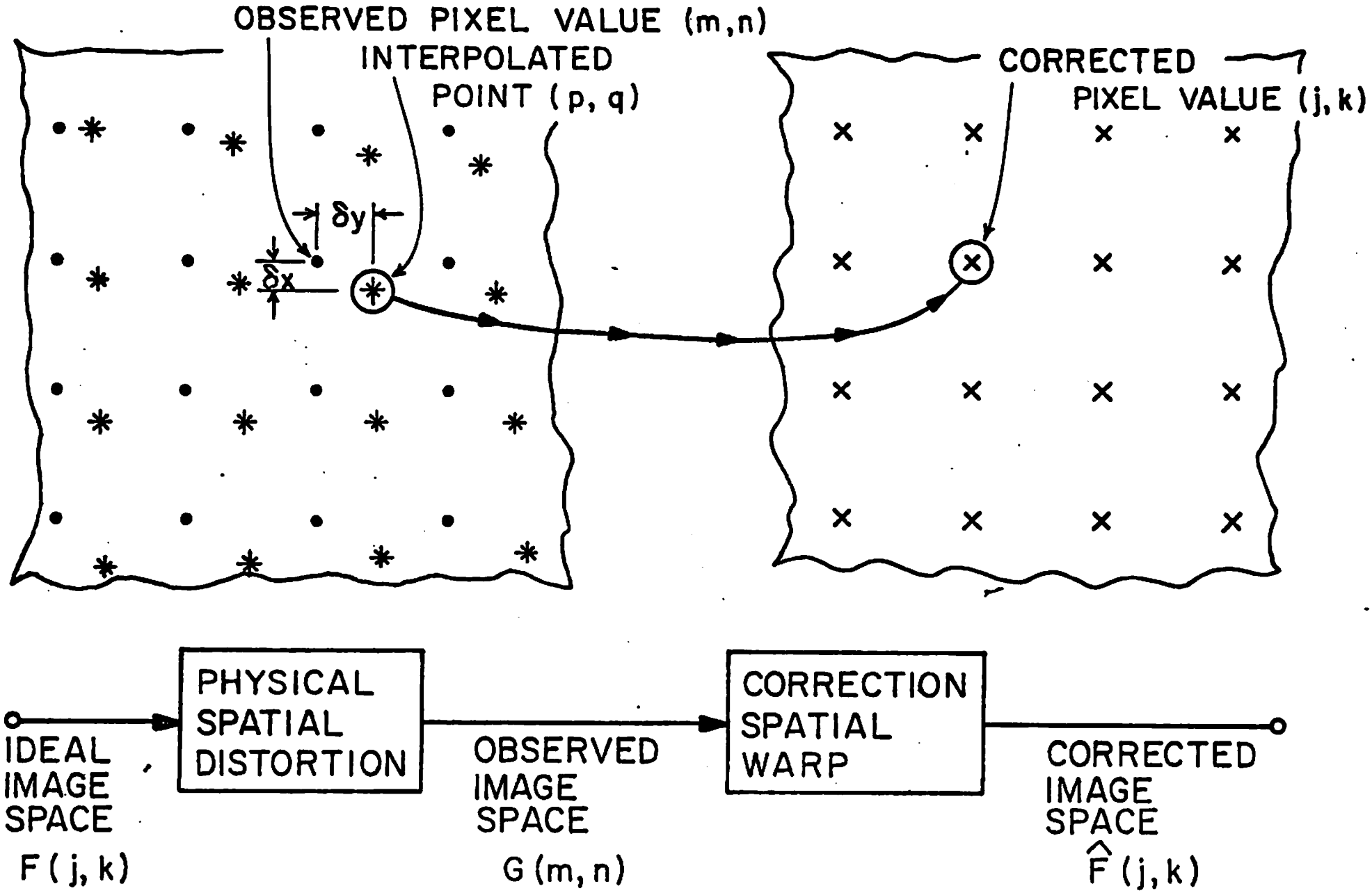


Figure 1. Spatial Warping Concept.

$$\hat{p} = a_o + a_1 j + a_2 k + a_3 j^2 + a_4 k^2 + a_5 jk \tag{9a}$$

$$\hat{q} = b_o + b_1 j + b_2 k + b_3 k^2 + b_4 k^2 + b_5 jk \tag{9b}$$

represent a fifth order spatial transformation. Then, the unknown constants

$$\underline{a} = [a_o\ a_1\ a_2\ a_3\ a_4\ a_5]^T \tag{10a}$$

$$\underline{b} = [b_o\ b_1\ b_2\ b_3\ b_4\ b_5]^T \tag{10b}$$

can be determined by measurement of a set of known vertices (control points) in the (j,k) and (p,q) coordinate systems. For M such points arranged in vector space form eq.(9) results in the relations

$$\underline{\hat{p}} = \underline{A}\,\underline{a} \tag{11a}$$

$$\underline{\hat{q}} = \underline{A}\,\underline{b} \tag{11b}$$

where

$$\underline{p} = [p_1\ p_2 \cdots p_u]^T \tag{12a}$$

$$\underline{q} = [q_1\ q_2 \cdots q_u]^T \tag{12b}$$

$$\underline{A} = \begin{bmatrix} 1 & j_1 & k_1 & j_1^2 & k_1^2 & j_1 k_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & j_M & k_M & j_M^2 & k_M^2 & j_M k_M \end{bmatrix} \tag{13}$$

For a given set of control points the mean square error

$$\delta = (\underline{p}-\underline{\hat{p}})^T (\underline{p}-\underline{\hat{p}}) + (\underline{q}-\underline{\hat{q}})^T (\underline{q}-\underline{\hat{q}}) \tag{14}$$

is minimized for weighting constants computed by

$$\underline{a} = \underline{A}^{-} \underline{p} \tag{15a}$$

$$\underline{b} = \underline{A}^{-} \underline{q} \tag{15b}$$

where $\underline{A}^{-}$ is the generalized inverse of $\underline{A}$. For an overdetermined system

$$\underline{A}^{-} = (\underline{A}^{T}\underline{A})^{-1} \underline{A}^{T} \tag{16}$$

Figure 2 describes the spatial warp technique for image interpretation. With this method each image is initially segmented into regions of common attribute, e.g. luminance, tristimulus values, texture. Next, a set of M control points $(\underline{j},\underline{k})$, chosen to be invariant under perspective transformation, are selected from each region and used to generate the warp transformation matrix $\underline{A}$. These control points are then warped to the space of the scene model data base, and matched to corresponding control points obtained from segmented images of the data base. The data base may contain primitive segment shapes (triangles, quadrilaterals, ellipses, etc.), or macro-segments (auto wheels, airplane wings, or house doors), or combination of both. The scene model control points $(\underline{p}_n,\underline{q}_n)$ for n = 1,2,...,N segments are used with the warp transformation matrix $\underline{A}$ to compute a set of N polynomial weighting factors $(\underline{a}_n,\underline{b}_n)$. With these weighting factors the control points are warped to the scene model space to produce the set of warped control points $(\hat{\underline{p}}_n,\hat{\underline{q}}_n)$ which are then compared to $(\underline{p}_n,\underline{q}_n)$. If the error is sufficiently small, the segments are subjected to a match over all points in their surfaces by warping the test image segment to the scene model space and performing a pixel-by-pixel match.

The major advantage of the spatial warp image interpretation system are its relative simplicity and

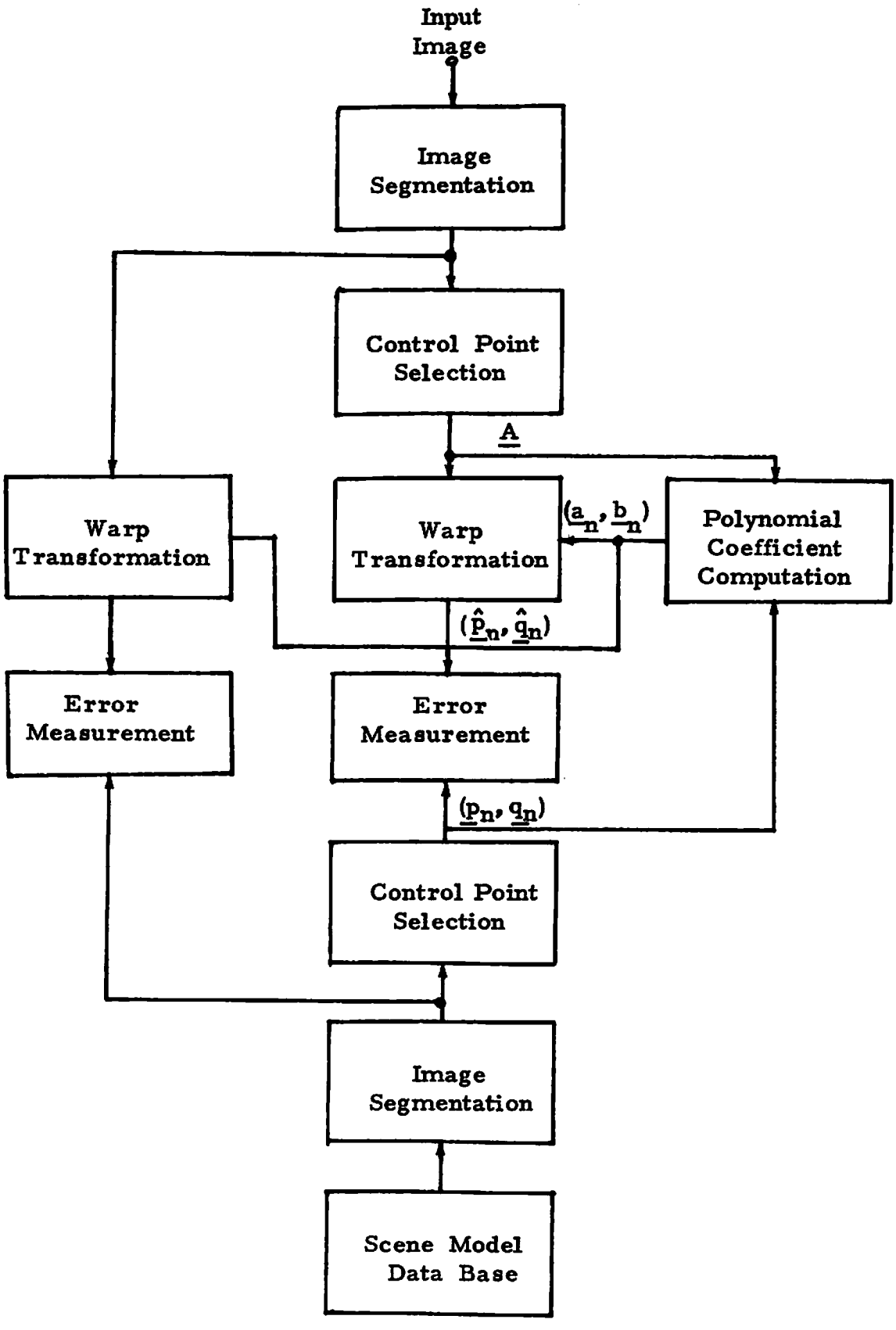


Figure 2. Spatial Warp Image Interpretation.

computational efficiency. The control points for scene model image segments can be pre-computed and stored. Then for each test image segment it is only necessary to compute a relatively small dimension generalized inverse. For each potential control point match it is necessary to perform a simple low dimensional vector multiplication to warp the test image control points to the space of the scene data. Total warping of the test image is performed only if the control point match is sufficiently close.

There are several important questions which must yet be explored. First, and perhaps most important is to determine the effect of inaccuracy in the detection of control points. If the number of control points is reasonably large, small errors should be insignificant. Another important point to investigate is the best measure of control point and image match error and means of evaluating if the error value is sufficiently small to judge a positive map. Work is underway on these theoretical questions in conjunction with a simulation study.

Reference

[1] L.G. Roberts, "Machine Perception of Three-Dimensional Solids," in Optical and Electro-Optical Information Processing, J.T. Tippett, et. al., eds., MIT Press, Cambridge, Massachusetts, pp. 159-197.

3.6 Estimation-Detection of Object Boundaries in Noisy Pictures

Nasser E. Nahi and Simon Lopez-Mora

An algorithm for successively estimating boundaries was introduced in [1]; the proposed estimator, unfortunately, assumed the presence of objects at every line in the picture and was susceptible to divergence. In [2,3] (see [2] for details) the algorithm was modified in such a way that before a new line was processed, a test to determine the acceptance or rejection of the estimated boundary was performed. Although the "refined estimator" as it was called, showed significantly better than the original one, the test procedure was ad hoc.

In the present report, the problem is formulated under a joint estimation-detection context [4] with the associated cost function. This framework permits us to obtain an optimal boundary estimation processor that includes a choice for the detector component as well as a procedure for optimal selection of the detection threshold.

Since the introduction of the joint scheme [4], several authors [4-10] have considered applications where signal parameters are estimated under the assumptions that they are defined at all times. This present formulation differs from the above mentioned applications in that the parameters to be estimated are hypothesis-dependent, signifying that their estimation is meaningful only under a given alternative.

Using the same nomenclature as in [2,3] let us express the set of observations ($k = 1,\ldots,N$) as

$$H_o:\ y(k) = S_b(k) + v(k),\ k \in I_\phi$$
$$H_1:\ y(k) = \lambda(k)S_o(k) + (1-\lambda(k))S_b(k) + v(k),\ k \notin I_\phi \qquad (1)$$

where λ is a binary sequence taking values 1 where the object is present ($S_b(k)$) and 0 elsewhere ($S_o(k)$), v is a white Gaussian noise sequence with zero mean and known variance, I_ϕ = {lines where the object is present}. In

turn, on these lines where the object is present, the boundary function λ is expressed in terms of two random variables w, c that represent the width and geometrical center of the object on the associated line.

Under the given setting, it can be seen that the processor must accomplish the following task: screen the two given hypotheses at the end of every line, and produce a decision at its output concerning the absence of the object (when H_o is decided). When H_1 is decided, proceed to estimate the width and center of the object and present these values at the output, see figure 1. In other words the boundary estimator output must be either a decision or an estimate. In order to optimize the structure in figure 1, it is necessary to define the cost function to minimize. Let

γ_i, i = 0,1 : be the decision associated with hypothesis H_i

$Y = \{ Y_{iN+1}, Y_{iN+2}, \ldots, Y_{(i+1)N}, \; 0 \leq i \leq N-1 \}$

$\delta(\gamma|Y)$: be the decision rule assigning a probability between (or equal to) 0 or 1 to each decision γ_i, i = 0,1 the distribution depending on Y.

p_i, i = 0,1: be the a priori probability of hypothesis H occurring

C_{ij}, i,j = 0,1: be the detection cost incurred when deciding γ_i when H_j happened

$f_{1,1}(\hat{\hat{r}}(Y), r)$: cost of estimating the object boundary, $\hat{\hat{r}}(Y)$, when γ_1 has been decided and when an object with boundary r was present (H_1 occurred)

$f_{1,0}$: cost of estimating a boundary when the object was not present

$F_n(Y)$: be the n-dimensional probability density of the set Y

$F(r|H_1)$: be the probability density for the object boundary r

It should be pointed out that the selection of $C_{0,1}$ and $f_{1,0}$ as constant values emphasizes the determination of the existence of the object as contrasted to non-constant costs which penalize missing or estimating objects in relation to their size and (or) positions.

The average cost of detection and estimation can now be written as

$$R_{O+E} + \int_Y dY\{\delta(Y_0|Y)[p_0 C_{0,0} F_n(Y|H_0) + p_1 C_{0,1} \langle F_n(Y|r,H_1)\rangle_r] + \delta(Y_1|Y)[p_0 C_{1,0} F_n(Y|H_0) + p_1 C_{1,1} \langle F_n(Y|r,H_1)\rangle_r] + \delta(Y_1|Y)[p_0 f_{1,0} F_n(Y|H_0) + p_1 \langle f_{1,1} F_n(Y|r,H_1)\rangle_r]\} \quad (2)$$

where

$$\langle F_n(Y|r,H_1)\rangle_r = \int F_n(Y|r,H_1) F(r|H_1) dr$$

Minimization of (2) (with respect to ζ, $\hat{\hat{r}}$) when a quadratic error cost is chosen for $f_{1,1}$ results in the following optimum decision rule and estimator

$$\hat{r}(Y) = E\{r|Y,H_1\} \quad (3)$$

$$\frac{p_1 F_n(Y|H_1)}{p_0 F_n(Y|H_0)} \equiv \Lambda \underset{\gamma_0}{\overset{\gamma_1}{\gtrless}} \frac{C_{1,0} + f_{1,0} - C_{0,0}}{C_{0,1} - C_{1,1} - E\{\tilde{r}^T \tilde{r}|H_1\}} \equiv K_\Omega \qquad (4)$$

For this particular cost function and in view of (3) and (4), figure 1 becomes figure 2. To illustrate the boundary processor performance, the armored personnel carrier (APC) of the USCIPI data base has been used. In order to emphasize the boundary estimation itself, the object and background luminances have been assumed known and constant. The luminance values were chosen from a histogram of the original picture. The noisy observations, figure 3b, figure 4a and figure 4b have been obtained from the original carrier by adding white Gaussian noise of standard deviation σ. The results appear in figures 3 and 4.

The parameters chosen for the boundary processors were

$$N = 256 \qquad \bar{w} = 75.8 \qquad \bar{C} = -4.6 \qquad p_0 = p_1 = .5$$

$$A_0(\text{mN}) = \begin{pmatrix} .876 & 0. \\ 0. & .924 \end{pmatrix} \qquad B_0(\text{mN}) = \begin{pmatrix} 4.02 & 0. \\ 0. & 2.67 \end{pmatrix}$$

For figures 3, 4a and 4c $K_\Omega = 1$

For figures 4b and 4d

$$C_{1,0} + f_{1,0} - C_{0,0} = 260$$

$$C_{0,1} - C_{1,1} = 260$$

Similar results have been obtained when the object luminance is unknown and will be reported in the near future.

References

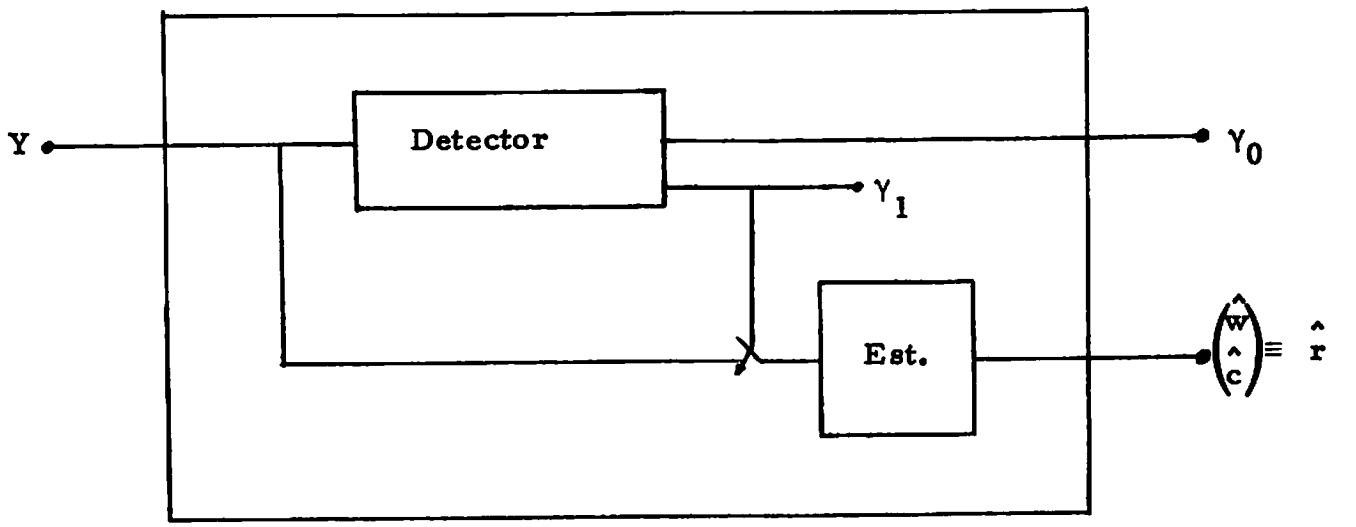


Figure 1. Boundary Processor.

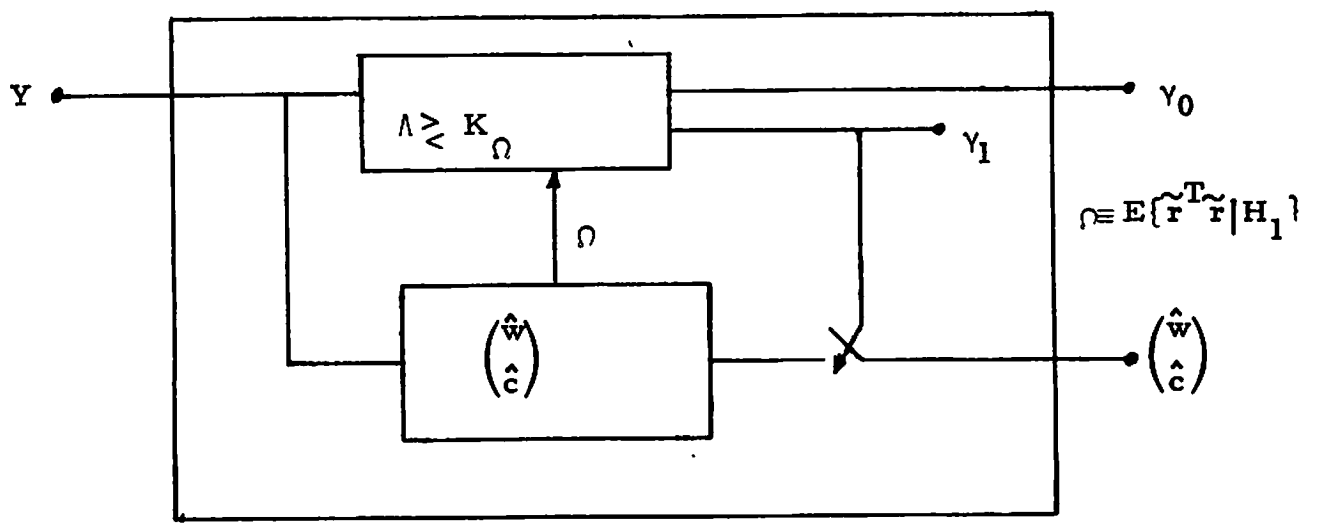
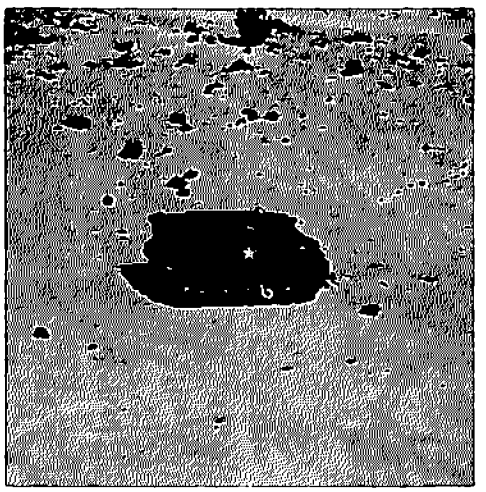
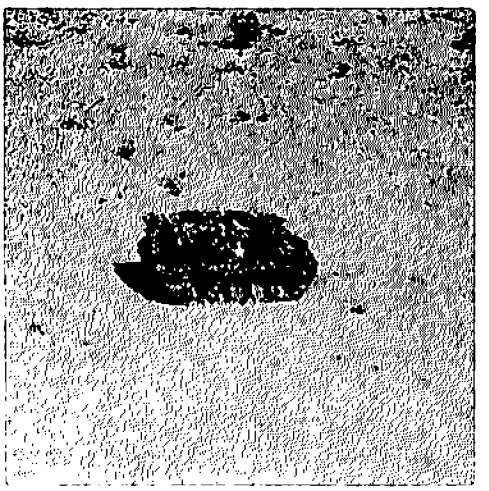


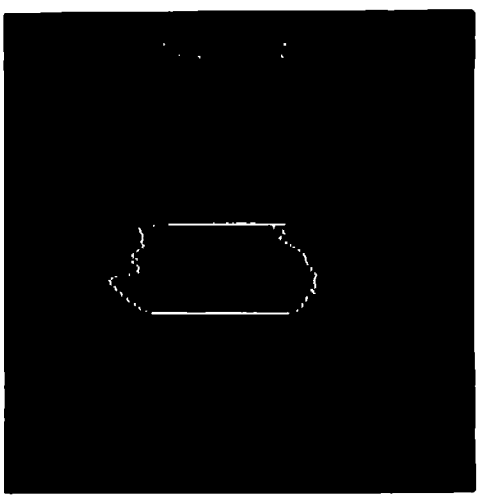
Figure 2. Boundary Processor for a particular Cost Funtion.



(a) APC original



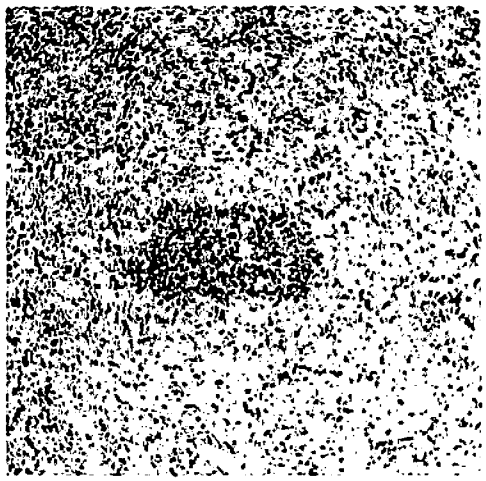
(b) Noisy APC, $\sigma = .1$



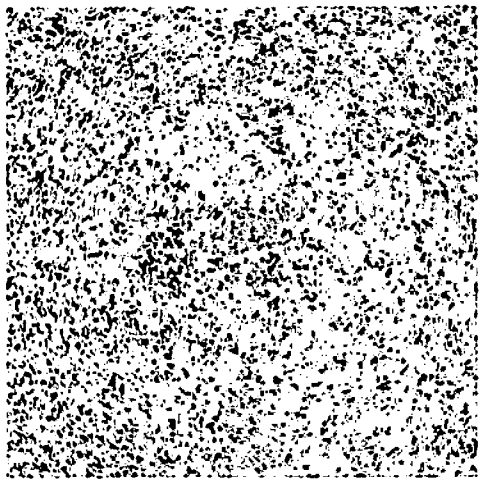
(c) Estimated APC boundary

(d) Original and estimated APC boundaries

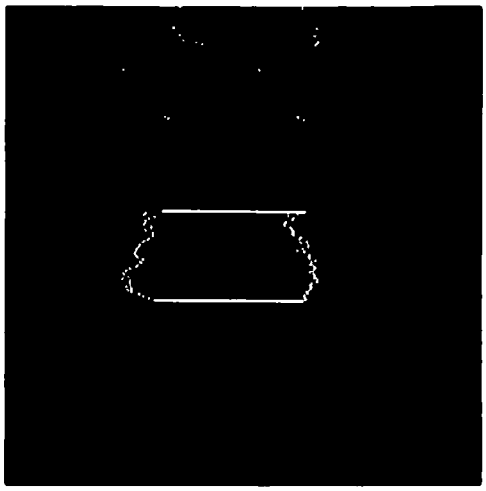
Figure 3.



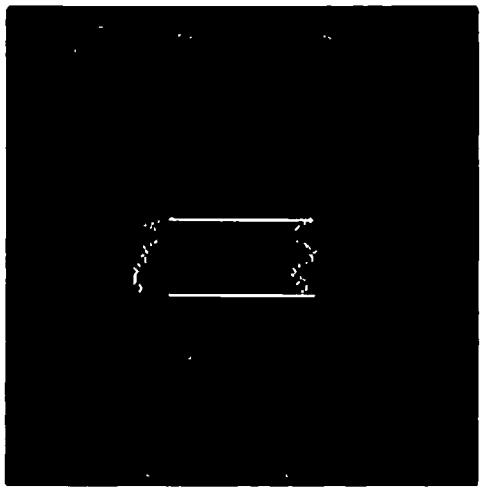
(a) Noisy APC, $\sigma = .6$



(b) Noisy APC, $\sigma = 1.2$



(c) Estimated APC boundary when $\sigma = .6$



(d) Estimated APC boundary when $\sigma = 1.2$

Figure 4.

[1] N. Nahi and S. Lopez-Mora, "Recursive Estimator for the Determination of Boundaries," University of Southern California, USCIPI Report 660, pp. 81-85.

[2] N. Nahi and S. Lopez-Mora, "Estimation of Object Boundaries in Noisy Images," Proceedings of the 1976 IEEE Decision and Control Conference, Clear Water, Florida, December 1976, pp. 607-612.

[3] N. Nahi and S. Lopez-Mora, "Recursive Estimation of Boundaries," University of Southern California, USCIPI Report 720, pp. 35-40.

[4] D. Middleton and R. Esposito, "Simultaneous Optimum Detection and Estimation of Signals in Noise," IEEE Transactions on Information Theory, Vol. IT-14, May 1968, pp. 434-444.

[5] N. Nahi, "Optimum Recursive Estimation with Uncertain Observation," IEEE Transactions on Information Theory, Vol. IT-15, July 1969, pp. 457-462.

[6] M.D. Srinath and P.K. Rajasekaran, "Estimation of Randomly Occurring Stochastic Signals in Gaussian Noise," IEEE Transactions on Information Theory, Vol. IT-17, March 1971, p. 206.

[7] A.G. Jaffer and S.C. Gupta, "Recursive Bayesian Estimation with Uncertain Observation," IEEE Transactions on Information Theory, Vol. IT-17, September 1971, pp. 614-616.

[8] A. Fredriksen, D. Middleton and D. Vandelinde, "Simultaneous Signal Detection and Estimation Under Multiple Hypotheses," IEEE Transactions on Information Theory,

Vol. IT-18, September 1972, pp. 607-614.

[9] D.W. Kelsey and A.H. Haddad, "Detection and Prediction of a Stochastic Process Having Multiple Hypotheses," Information Sciences 6, 1973, pp. 301-311.

[10] D.W. Kelsey and A.H. Haddad, "A Note on Detectors for Joint Minimax Detection-Estimation Schemes," IEEE Transactions on Automatic Control, October 1973, pp. 558-559.

4. Smart Sensor Projects

Our smart sensor effort is progressing nicely with a division of labor between USCIPI personnel and Hughes Research Laboratory personnel. As can be seen from the following, simulations at USC indicated very small adaptive convolving kernels can be quite useful for preprocessing close to the front end of a sensor. In addition, such processes, when implemented near the focal plane, provide potential for reduced subsequent dynamic range requirements in higher level processes. The test facility at HRL is progressing and the Sobel chip seems to be making the usual progress through the variety of production facilities necessary to configure such devices. Similar comments can be applied to the Circuit II, our first attempt at "adaptive on-chip" processing. Finally preliminary efforts are underway to design a real time CCD focal plane image segmentor. This represents our first entry into designing actual image understanding algorithms for potential on-board smart sensor implementation.

4.1 Enhancement with 3 x 3 Kernels

Harry C. Andrews

Progress in the development of smart sensor technology has led to the need for preliminary simulation of special algorithms prior to sensor design and construction. Parallel to this recent smart sensor effort has been the long-standing need for the development of high speed exploitation facilities for human interpretation of digital imagery. Typically such exploitation algorithms have been described as "enhancement" procedures but have traditionally been space-invariant in nature. More sophisticated modern-day digital image processing has led to the study of adaptive (space-variant) enhancement techniques. Coupled with the ability of both smart sensor and digital refresh technology to implement 3 x 3 convolutions within 1/30 second for 512 x 512 x 8 imagery, it was decided to undertake a study of the power and limitations that such 3 x 3 convolving kernel operations could be utilized to the tasks of both exploitation facility enhancement and smart sensor two-dimensional signal processing. Towards that end, this report represents preliminary results from such a study.

The underlying theme for this section is the utilization of 3 x 3 kernels for use as control signals to implement both linear and nonlinear as well as spatially invariant and variant (adaptive) signal processing functions in two dimensions. Coupled with this motivation is the fact that USCIPI and Hughes Research Laboratories are jointly embarking upon the construction of circuits which would potentially be able to implement these signal processing functions. A large variety of algorithms have been developed for these tasks, and probably those which are the

most successful would be labeled as nonlinear. However, all algorithms could be envisioned as being designed around three control signals or two-dimensional functions. These three signals are:

$$\begin{aligned} &\text{original image} = f(x,y) \\ &\text{blurred image} = f_m(x,y) \\ &\text{Sobel of image} = f_s(x,y) \end{aligned} \tag{1}$$

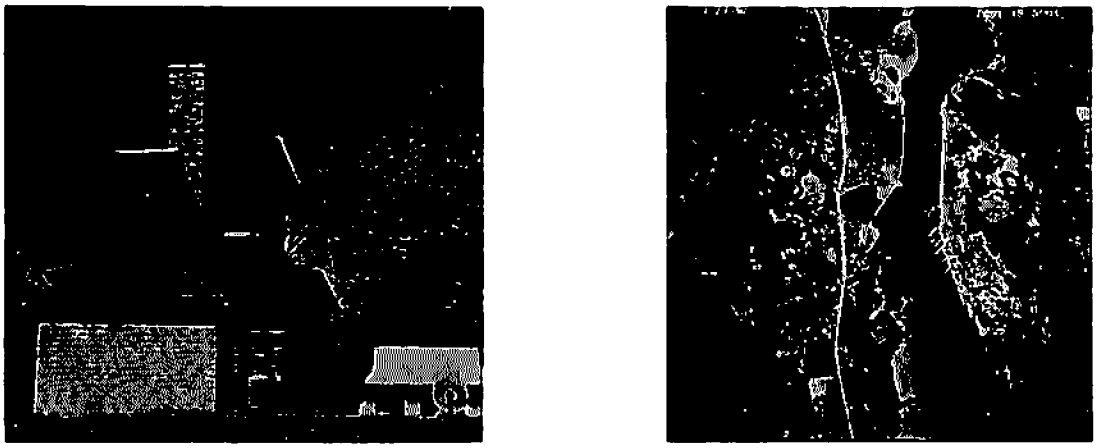
The blurred image is obtained by convolving f(x,y) with a 3 x 3 kernel whose entries are all unity. The Sobel image is obtained by passing the 3 x 3 Sobel operator over the original image. Thus

$$f_m(x,y) = f(x,y) \circledast \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

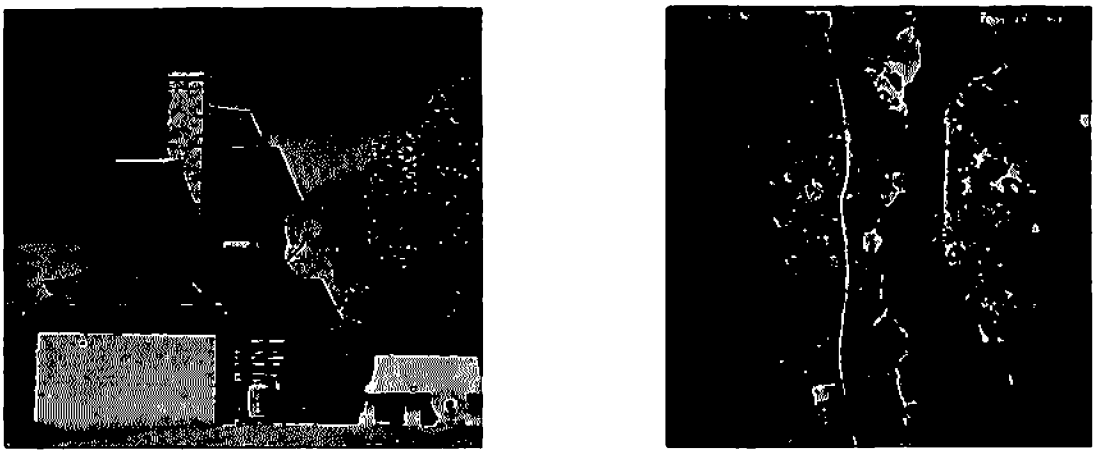
$$f_s(x,y) = \left| f(x,y) \circledast \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \right| + \left| f(x,y) \circledast \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \right|$$

Both these functions are easily implemented in either discrete high speed digital circuitry or analog CCD array technology. Figure 1 presents these three functions on two images, a "house" and an aerial "reconnaissance" scene. The former has quite a large dynamic range while the latter has a smaller dynamic range.

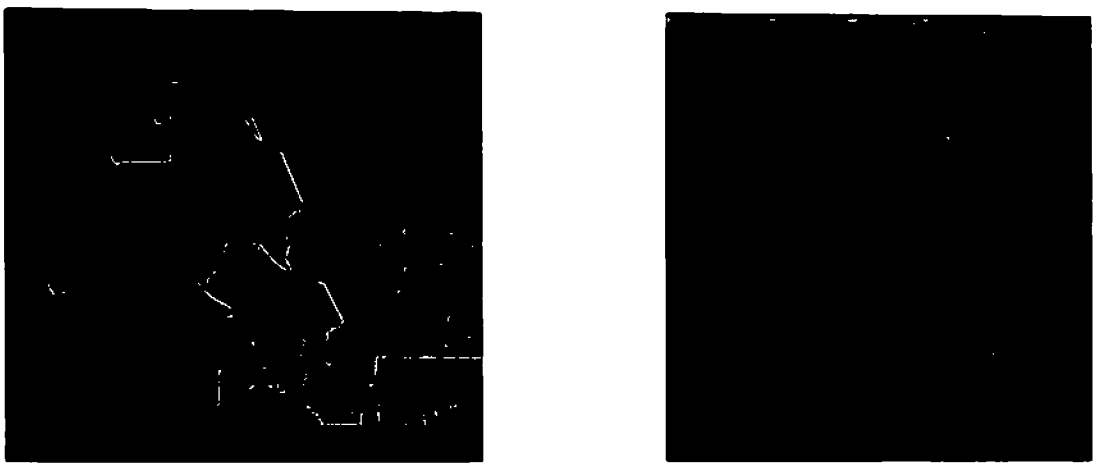
To exemplify a simple use of these control signals, consider the simple linear combination of the mean and Sobel



(a) Originals = $f(x, y)$

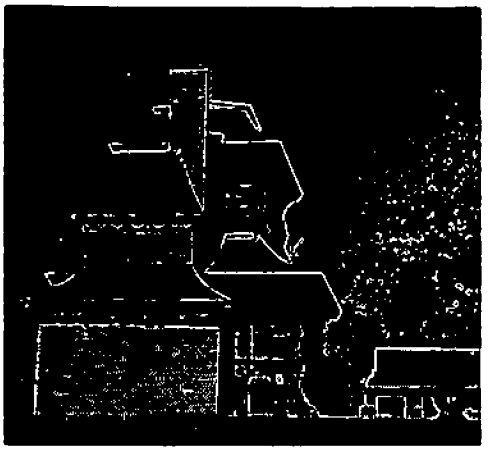


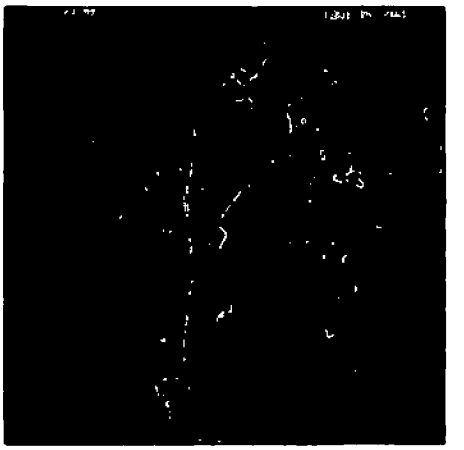
(b) Blurs = $f_m(x, y)$



(c) Sobels = $f_s(x, y)$

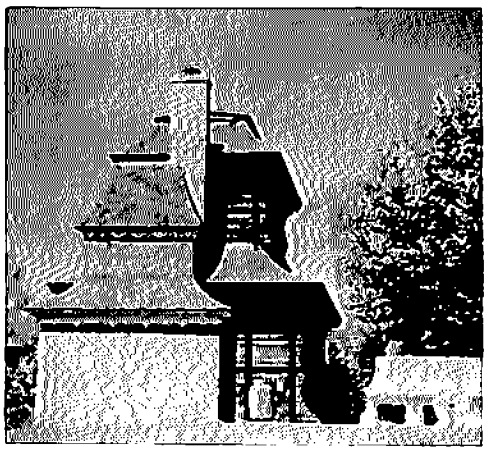
Figure 1. Control Signals

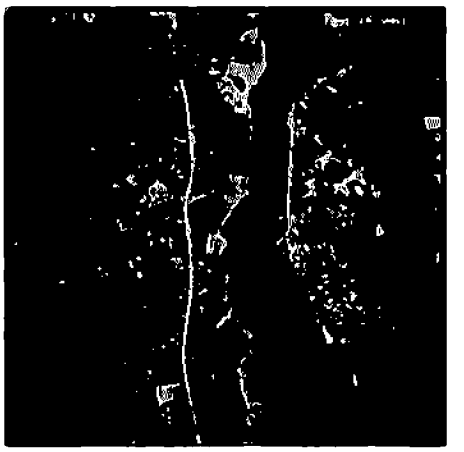




a) $\lambda = 0.75$

b) $\lambda = 0.50$





c) $\lambda = 0.25$

Figure 2. Nonadaptive $(1-\lambda) f_m(x, y) + \lambda f_s(x, y)$

a) $\lambda = 0.75$

b) $\lambda = 0.50$

c) $\lambda = 0.25$

Figure 3. Nonadaptive Unsharp Masking $f(x, y) - \lambda f_m(x, y)$

image, i.e.

$$g(x, y) = (1-\lambda)\, f_m(x, y) + \lambda\, f_s(x, y)$$

where

$$0 \leq \lambda \leq 1$$

Figure 2 illustrates this situation where in figure 2a, the Sobel edges are emphasized, while in figure 2c the mean image is emphasized. Probably a more familiar use of these control signals for edge enhancement is in the "unsharp masking" application of simply subtracting a percentage of the blurred image from the original. Thus

$$g(x, y) = f(x, y) - \lambda\, f_m(x, y)$$

This situation is illustrated in figure 3 for a variety of values for λ (1/4,1/2,3/4).

Both figures 2 and 3 are nonadaptive algorithms which are easily implemented but which indicated a lack of sophistication applied to the needs of the inherent nonstationary nature of imagery. A very simple adaptive stretching algorithm has been implemented in figure 4a in which the brightness of a grey level has been doubled depending on the value of $f_m(x,y)$. Thus

$$g(x, y) = 2\begin{cases} \min(f(x, y),\ 127) & f_m(x, y) < 128 \\ \max(f(x, y)-128,\ 0) & f_m(x, y) \geq 128 \end{cases}$$

Hence if the local mean is less than mid-grey, the center pixel is essentially passed through a function memory whose gain is a factor of two. The objective, of course, is to enhance the darks while simultaneously enhancing the brights without saturation in either case. Notice that this has indeed occurred in the shadows of the roofline of the "house" scene but unfortunately at the troublesome expense

of nonmonotonic artifacts due to the nonlinear decision process in the threshold of the output function.

A second adaptive algorithm, motivated by nonlinearities in most sensors, is illustrated in figures 4b and 4c. In both these cases it is assumed that the sensor (be it film, vidicon or CCD array) is optimized in terms of spatial frequency response in its linear brightness response range. However, in saturation, both lower and upper, it is anticipated that the spatial frequency response is decreased and as such could profit by localized enhancement. Consequently based upon the local mean, a linear proportion of edge emphasis is introduced depending upon how close to dark or bright saturation that local area of the image experiences. Figure 5 illustrates the central curves for the saturation for both the adaptive Sobel example and the adaptive unsharp masking. Thus the outputs are

<u>adaptive Sobel</u>

$$g(x, y) = \lambda f(x, y) + (1-\lambda) f_s(x, y)$$

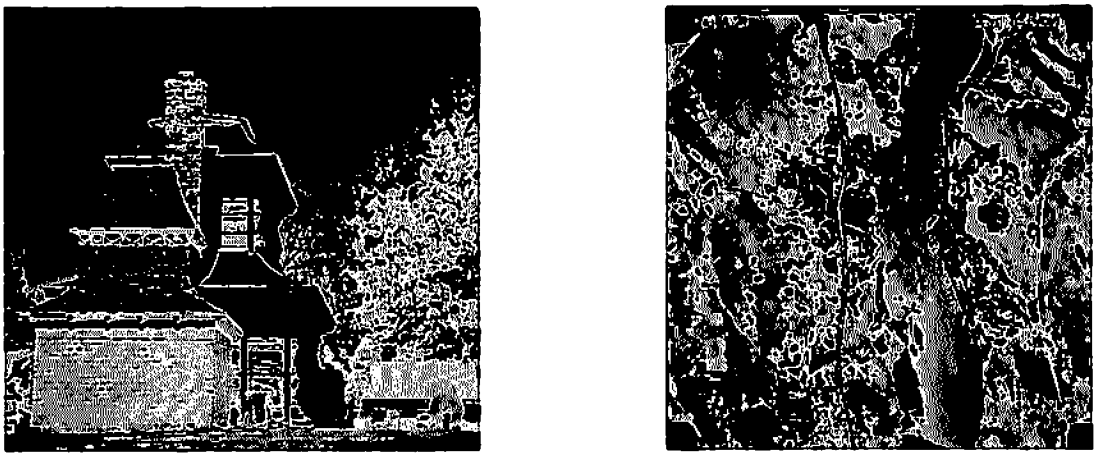
$$\lambda = \begin{cases} \frac{1}{2}(f_m(x, y) \,|\, 128) + \frac{1}{2} & f_m(x, y) < 128 \\ \frac{1}{2}((256-f_m(x, y)) \,|\, 128) + \frac{1}{2} & f_m(x, y) \geq 128 \end{cases}$$

<u>adaptive unsharp mask</u>

$$g(x, y) = f(x, y) - (1-\lambda) f_m(x, y)$$

$$\lambda = \begin{cases} \frac{3}{4}(f_m(x, y) \,|\, 128) + \frac{1}{4} & f_m(x, y) < 128 \\ \frac{3}{4}((256 - f_m(x, y)) \,|\, 128) + \frac{1}{4} & f_m(x, y) \geq 128 \end{cases}$$

Returning to figure 4 in both (b) and (c) it is clear that

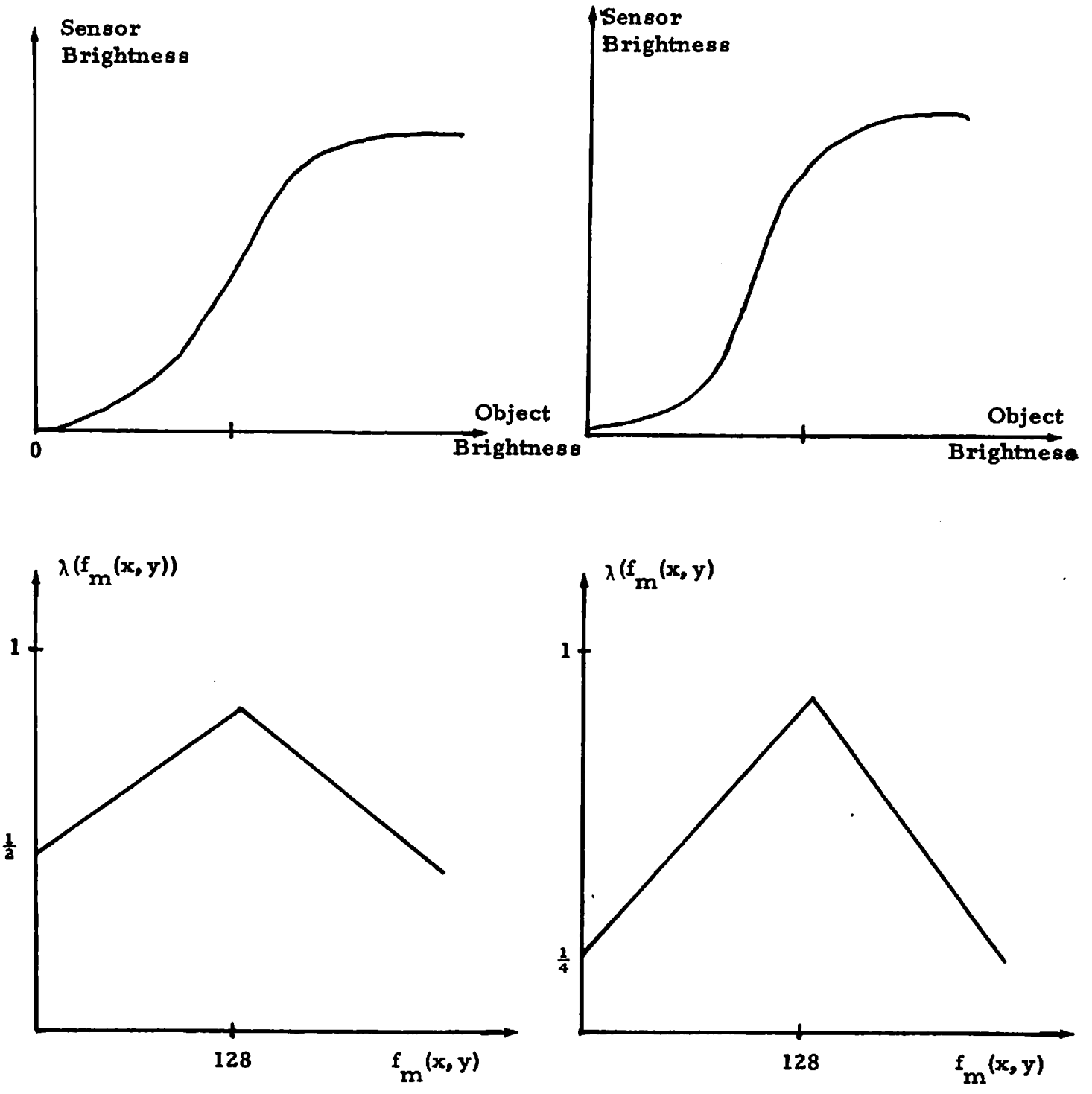


a) Adaptive Stretch

b) Adaptive Sobel

c) Adaptive Unsharp Mask

Figure 4. Some Adaptive Algorithms



a) Adaptive Sobel Control Functions

b) Adaptive Unsharp Mask Control Functions

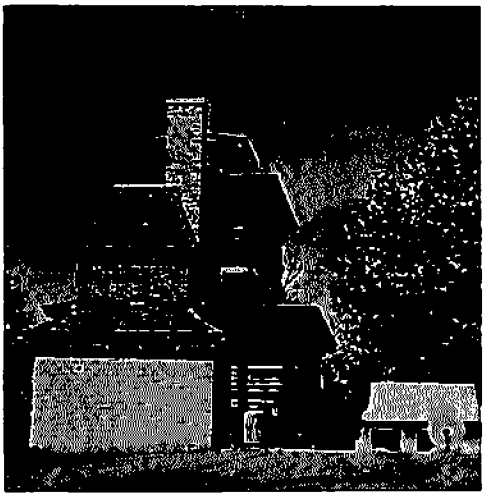
Figure 5. Adaptive Spatial Filtering

very bright regions are enhanced (see the bright roof in the lower right of the "house" scene). Unfortunately the edge enhancement in dark regions is still difficult to see because of the lack of much energy in these dark areas.

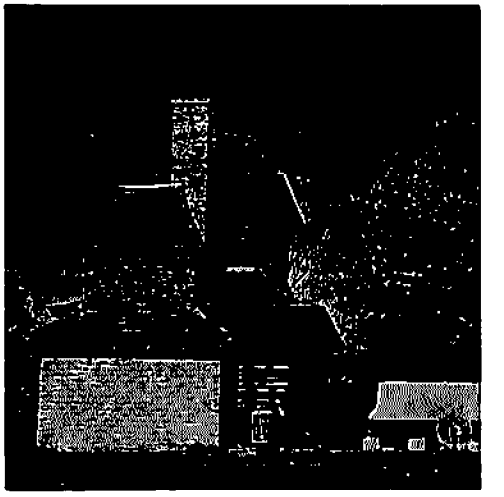
A final adaptive enhancement algorithm is illustrated in figures 6 and 7. This algorithm is motivated by the desire to develop zero mean unity variance random pixels which can then be passed through a common table look-up (function memory) for localized dynamic range adjustment and adaption. The variance of a neighborhood is assumed to be approximated by the Sobel edge energy. If we had a zero mean unity variance Gaussian random variable, the error function erf(•) would be used for localized table look-up dynamic range adjustment. However for simplicity the arctangent was substituted in its place. Thus the final output function becomes

$$g(x,y) = \lambda\, f(x,y) + (1-\lambda)\left\{\tan^{-1}\left\{\frac{f(x,y) - f_m(x,y)}{f_s(x,y)}\right\} + \pi\right\}\frac{255}{2\pi}$$

In figures 6 and 7 λ ranges from $\lambda = 1$ to $\lambda = 0$ providing the original all the way down to a very nearly binary image in which all shadow area and bright area region detail are simultaneously evident. Finally an adaptive binarizer is presented for comparison purposes in figures 6f and 7f. In these two cases we have a thresholded output to be black or white depending on the center pixel not exceeding or exceeding the local mean surround. Consequently

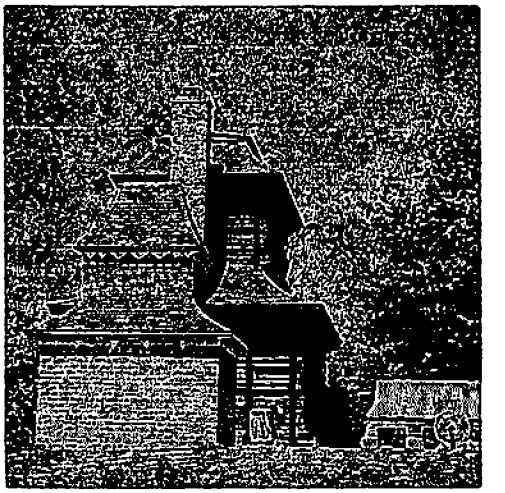


a) Original, $\lambda = 1.0$

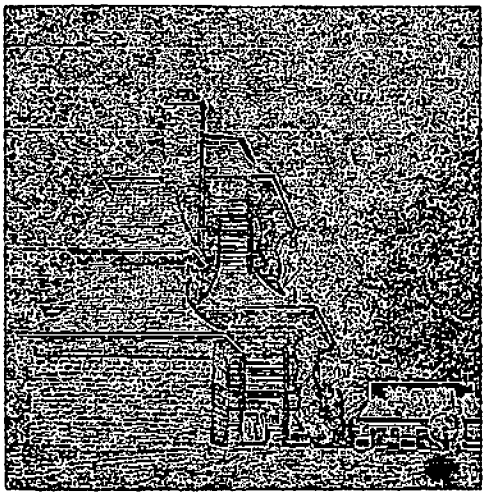


b) $\lambda = 0.75$

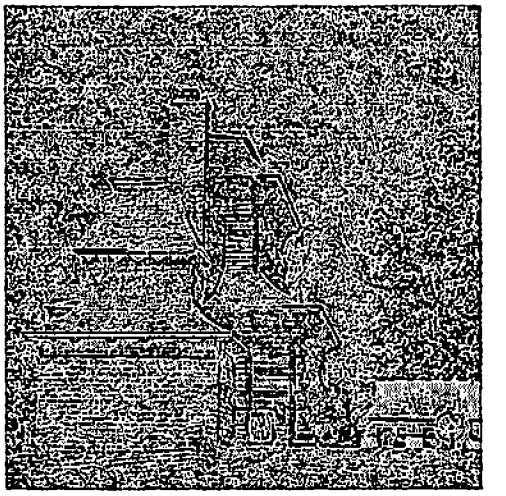
c) $\lambda = 0.50$



d) $\lambda = 0.25$



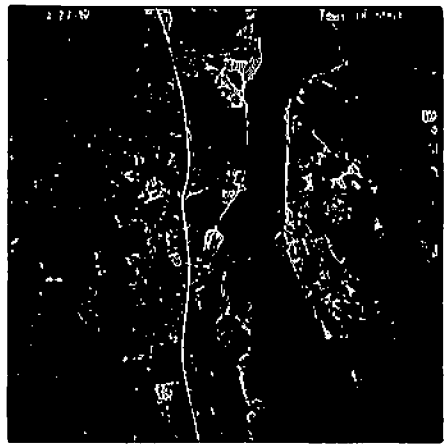
e) $\lambda = 0.00$



f) Binary

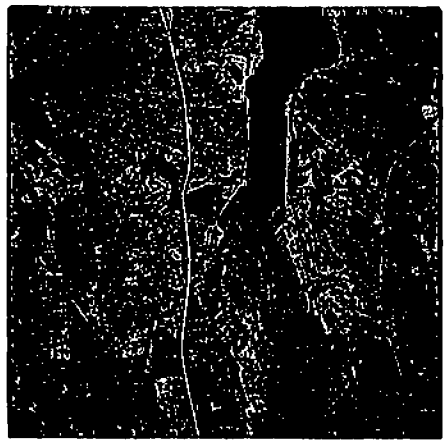
Figure 6. $\lambda f(x,y) + (1-\lambda)\left\{\text{Tan}^{-1}\left\{\frac{f(x,y) - f_m(x,y)}{f_s(x,y)}\right\} + \pi\right\}\frac{255}{2\pi}$

a) Original, $\lambda = 1.0$

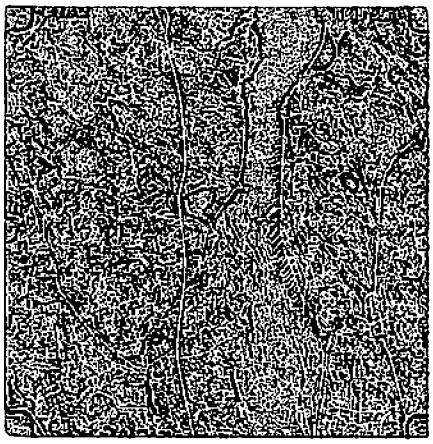


b) $\lambda = 0.75$

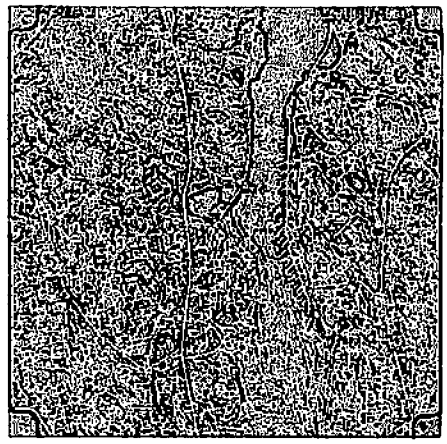
c) $\lambda = 0.50$



d) $\lambda = 0.25$



e) $\lambda = 0.00$



f) Binary

Figure 7. $\lambda f(x,y) + (1-\lambda)\left\{\text{Tan}^{-1}\left\{\frac{f(x,y) - f_m(x,y)}{f_s(x,y)}\right\} + \pi\right\}\frac{255}{2\pi}$

$$g(x, y) = \begin{cases} 255 & f(x, y) \geq f_m(x, y) \\ 0 & f(x, y) < f_m(x, y) \end{cases}$$

4.2 Real Time Implementation of Image Segmentation

Guy Coleman

The procedure described in Section 2.1 is currently being used to segment images on a general purpose digital computer. It is possible to implement this scheme, with some suitable modifications, to segment images in near real time, that is, at television rates.

Figure 1 is a conceptualization of such an implementation. Starting at the input to the system on the left side of the diagram, the features are computed in real time by the Feature Computer. Implementation of some of these features on CCD devices is already in progress as described elsewhere in this report.

The covariance matrix of the features is computed and diagonalized by the Statistical Computer. It is expected that the diagonalization process may take more than one frame, hence the rotation matrix that is sent to the Feature Rotator is based upon data that is several frames behind. Implicit in this approach is the assumption that the overall picture statistics do not change significantly in a small number of frames.

The rotated features go to the Segmentor, the Mean Computer and the Cluster Data Computer. Preliminary

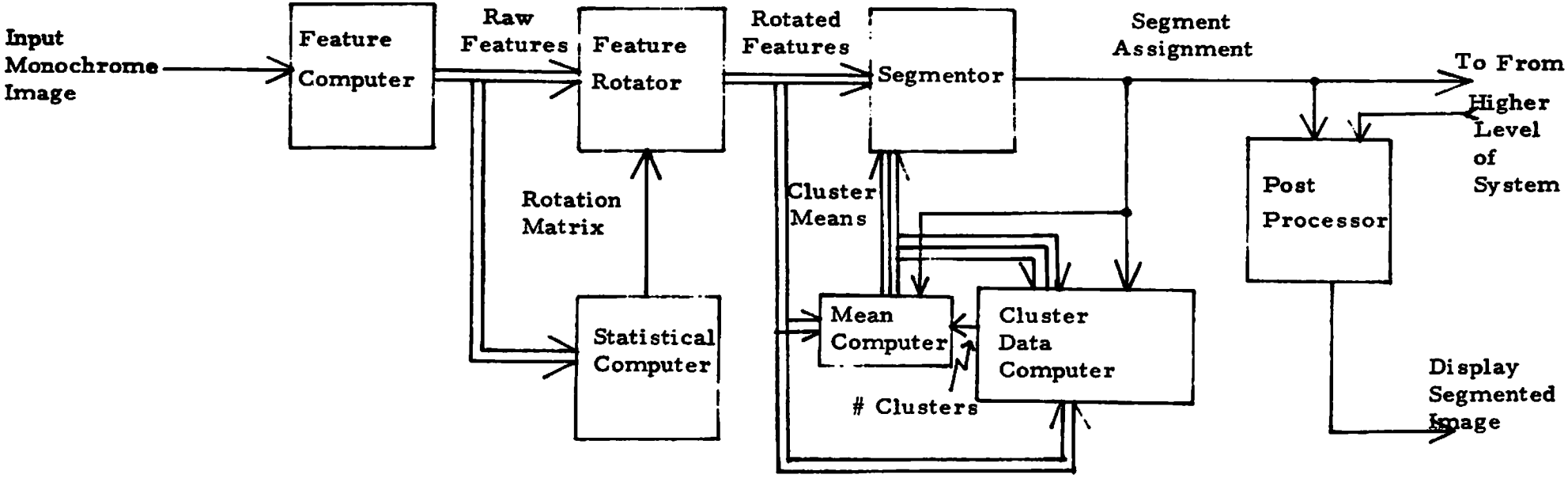


Figure 1.

Real Time Implementation of Scene Segmentor

investigation of implementing the Segmentor and the Mean Computer on CCD devices is currently in progress. If the Feature Computer, the Mean Computer and the Segmentor were available, real time nearest means (ISODATA) clustering could be performed on non-rotated features. The number of clusters would have to be manually selected.

The function of the Cluster Data Computer is to determine when the Segmentor/Mean Computer loop has converged and to then compute whether to change the number of clusters. At this point, the procedure will most likely be somewhat different than that currently being used in the algorithms on the general purpose computer. The current algorithm computes a product of between and within class scatter averages. The "correct" number of clusters is chosen to be the number for which this product is maximum. Implementation of this concept in real time would require three segmentors and three mean computers, one set for N clusters, one for N-1 clusters and one for N+1 clusters.

The Cluster Data Computer would have to compute the product for each number of clusters and decide whether or not to change N. A different procedure may be capable of producing satisfying results with less hardware and thus only one set of hardware is shown.

After turn-on, a number of frames of data will be required for the system to converge. If the scene changes relatively little from frame to frame, the system will "track" after the initial period of convergence.

The post processor is used to combine clusters for display purposes, based on analysis of the segmented images by the higher levels of the system. This is required because contextual interpretation of the scene may dictate

that certain segments are part of the same object even if they have different appearances.

4.3 CCD Image Processing Circuitry

Graham R. Nudd

I. OUTLINE

This report represents the work undertaken in the second phase of the subcontract to Hughes Research Laboratories from the USC Image Processing Institute. Participants in the work described are R. Harp, C.L. Jiang, W. Jensen, D. Maeding, P. Nygard, P. Prince, and G. Nudd.

In the first year of this program we investigated the possibility of performing specific image processing algorithms in real-time using special purpose CCD/MOS integrated circuits. The algorithms investigated were:

1. Chirp Transformations
2. Roberts' Cross Operation
3. Sobel Operator
4. Hueckel Operator
5. Histogramming

(For the purpose of this study, real-time operation was considered to be 5 MHz with accuracy equivalent to eight bits.) For each of the above operations (apart from the Hueckel operator) we developed circuit concepts and performance estimates.

During the period covered by this report we have concentrated our efforts principally on developing the integrated circuits necessary to demonstrate feasibility and to verify our concepts. Two circuits have been selected for implementation, each operating on a 3 x 3 array of picture elements.

The first circuit (Test Circuit I), an implementation of the Sobel Operator for edge detection, is fabricated as an-n channel surface CCD and is designed to operate at 10 MHz rate with accuracy of six bits or better. The detailed design and layout of this circuit has now been completed, and devices should be processed by April 1977.

Test Circuit II contains five separate algorithms; low pass filtering, edge detection, unsharp masking, binarization, and adaptive contrast enhancement. This circuit will be built on a second n-channel test chip, and we hope to have devices processed by mid-year. We anticipate that this chip will be approximately 190 mil x 190 mil, and if there is sufficient area, we will include other test circuits on the same chip. The exact space available for other circuits will not be known until a detailed layout has been completed in the next month or so.

Both circuits are analog implementations which perform arithmetic functions, such as the addition, intensity weightings, and the absolute value operation required in the Sobel, at rates equivalent to 200 MHz. Further, the relatively small size of these circuits offer the possibility of highly parallel operations. For example, by using the parallel arrangement of circuits shown in figure 1, edge detection on a full frame can be performed in a few microseconds, as compared with several minutes required by a general purpose computer. An example of this is shown in

the figure. A CCD imager or analog store is used to store a full frame, and the data from the N rows are clocked out in parallel into N parallel processing circuits. Each circuit might perform the Sobel operator, for example, and will process the data for an entire line, with the processed output appearing at the clock rate f_c (which for our circuits could be as high as 10 MHz). Thus, an entire frame would be processed in Nf_c seconds. For a 512 x 512 frame this would amount to 50 μsec. The advantages of this for direct focal plane processing are clear.

In addition to the detailed design and layout of the above circuits, we have spent some time developing the facilities necessary to perform the test and evaluation, and developing the software interface to access the USC data base from the IMSAI 8080 micro-processor which forms the basis of our present test system.

Each of the above tasks is discussed below, together with our plans and schedules.

II. TEST CIRCUIT I

The first test circuit is a CCD implementation of the Sobel edge detection algorithm. This circuit was chosen because it demonstrates two operations important to image processing; the possibility of achieving a two-dimensional convolution with arbitrary weightings and the ability to perform nonlinear functions such as the absolute magnitude operation.

The algorithm itself operates on an array of 3 x 3 picture elements with intensities f(i,j) as shown in figure 2, and evaluates

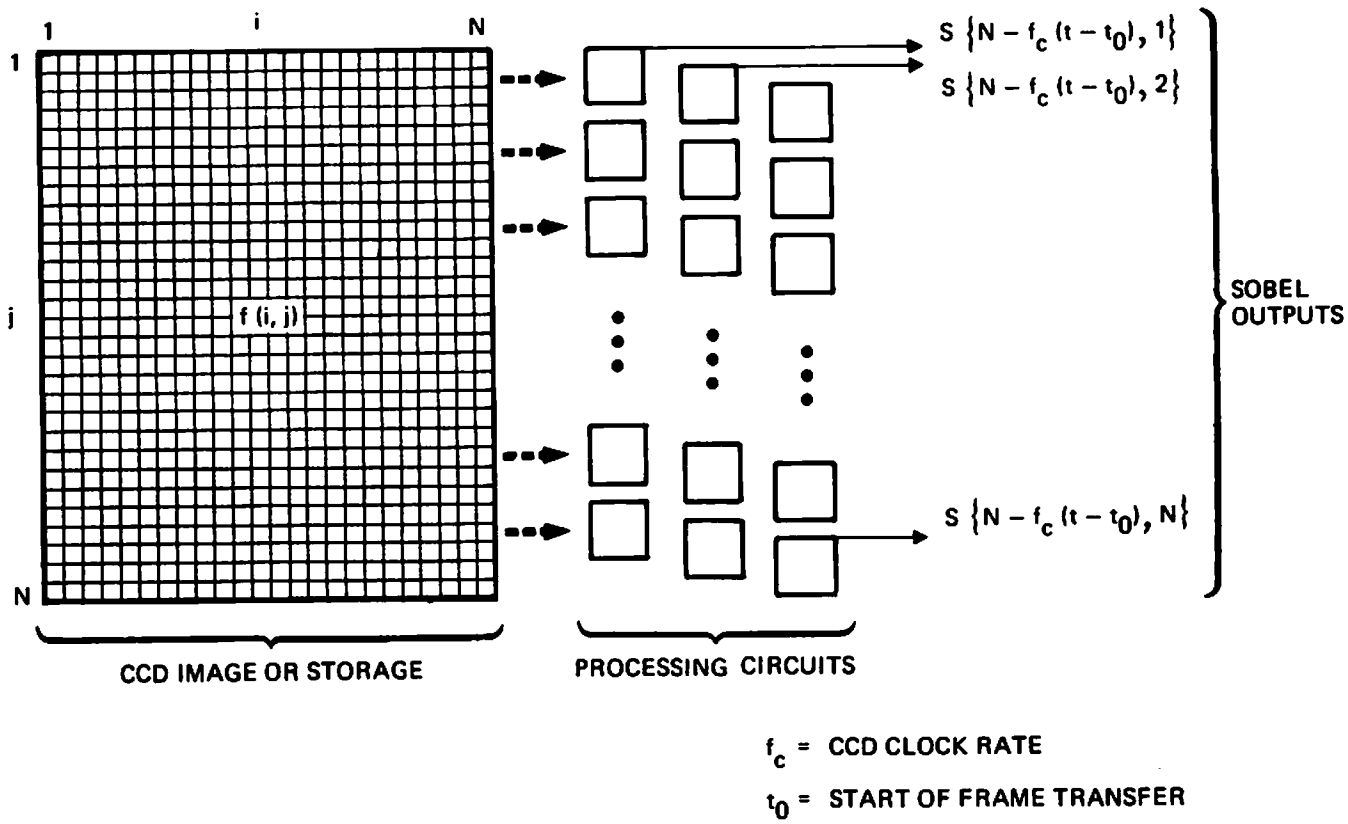


Figure 1. Concept of Parallel Pre-Processing Configuration

5931-5

f(i−1, j−1)	f(i, j−1)	f(i + 1, j − 1)
f(i − 1, j)	f (i, j)	f(i + 1, j)
f(i − 1, j + 1)	f(i, j + 1)	f(i + 1, j + 1)

$$S\{i,j\} = \frac{1}{8}\,[\,|(f(i-1,j-1) + 2f(I,j-1) + f(i+1,j-1)) - (f(i-1,j+1) + 2f(i,j+1) + f(i+1,j+1))|$$

$$+\ |(f(i-1,j-1) + 2f(i-1,j) + f(i-1,j+1)) - (f(i+1,j-1) + 2f(i+1,j) + f(i+1,j+1))|]$$

Figure 2. Unit Cell over which Sobel Operator Detects Edges

$$S\{i,j\} = \frac{1}{8}\Big[\Big|\Big(f(i-1,j-1) + 2f(i,j-1) + f(i+1,j-1)\Big) - \Big(f(i-1,j+1) + 2f(i,j+1) + f(i+1,j+1)\Big)\Big| + \Big|\Big(f(i-1,j-1) + 2f(i-1,j) + f(i-1,j+1)\Big) - \Big(f(i+1,j-1) + 2f(i+1,j) + f(i+1,j+1)\Big)\Big|\Big] \quad (1)$$

for each picture element. A schematic of the circuit concept is shown in figure 3. Three parallel lines of charge, proportional to the pixel intensities, are fed into the device using Tompsett potential equilibration inputs for linear operations. The top and bottom lines of charge are then divided into two parallel channels using a central implanted channel stop as illustrated, and floating gate electrodes are used to non-destructively sense the charge in each channel.

With the electrode configuration shown the voltage appearing on the top interconnection, for example is

$$V_1 = k\, C_{ox}\, \{f(i-1,j-1) + 2f(i,j-1) + f(i+1,j-1)\} \quad (2)$$

where C_{ox} is the oxide capacitance and k is a constant relating the charge generated by the input circuit to the pixel intensity. The weightings (1,2,1) are obtained directly by making the central electrodes twice the area of those on the corners. The voltages appearing on the other three interconnects are equivalent to the other expressions shown in eq.(1).

To calculate the full Sobel, S(i,j) pairs of these outputs are then subtracted and the absolute value of these operations taken prior to summation. In the direct implementation conventional MOS differential amplifiers (as shown in figure 9) can be used to perform this first

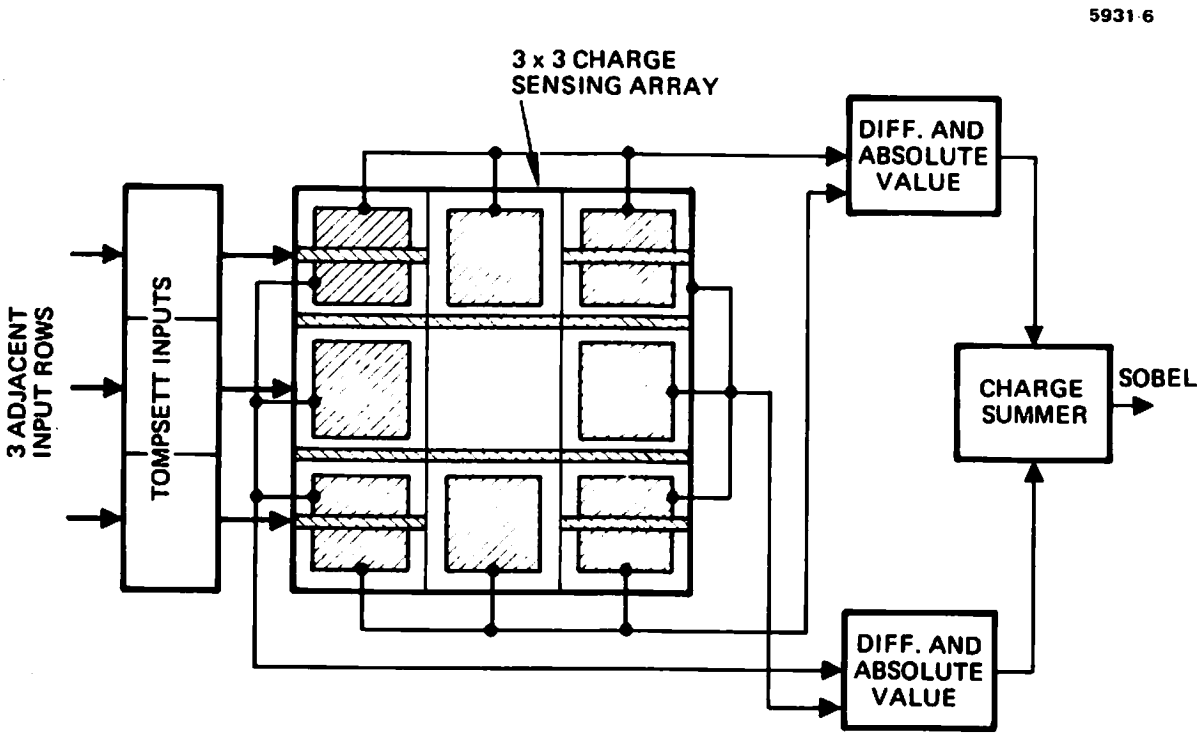


Figure 3. Schematic of CCD Sobel Circuit

operation, and the outputs from these fed to absolute value circuits as shown. We are currently investigating, on Test Circuit I, techniques to achieve all the arithmetic operations except the absolute value functions within the sensing array itself. We are also including on Test Circuit I two novel CCD absolute value circuits, and outputs will be available from both. Taking absolute values using charge coupled devices on the same IC chip that contains the charge-sensing matrices provides power savings, less temperature drift dependence and represents a very efficient analog processing method. A brief description of each circuit is given below.

A. CCD Absolute Value Circuits

1. Single Channel

Figure 4 depicts the circuit schematic and potential diagram of a single channel CCD absolute circuit. The circuit uses a fill and spill input system to generate a charge, Q, proportional to the magnitude of the voltage difference on gates SIG and B2, i.e., $Q = C_{ox} |(V_{sig} - V_{B2})|$.

In this way the B2 electrode is used as reference. For a negative input signal the potential profile at the silicon surface is as shown in the upper figure. When the diffusion ϕ_{INA} is pulsed, charge flows along the surface and fills the potential wells shown. When ϕ_{INA} drops, the excess charge flows across the potential barrier formed under the signal electrode back to the input diffusion. Then as the transfer gates, ϕ_{OUTA}, are clocked, the charge represented by the shaded area is clocked out.

For a positive input signal, the potential profile is shown in the lower figure. After the spill and fill

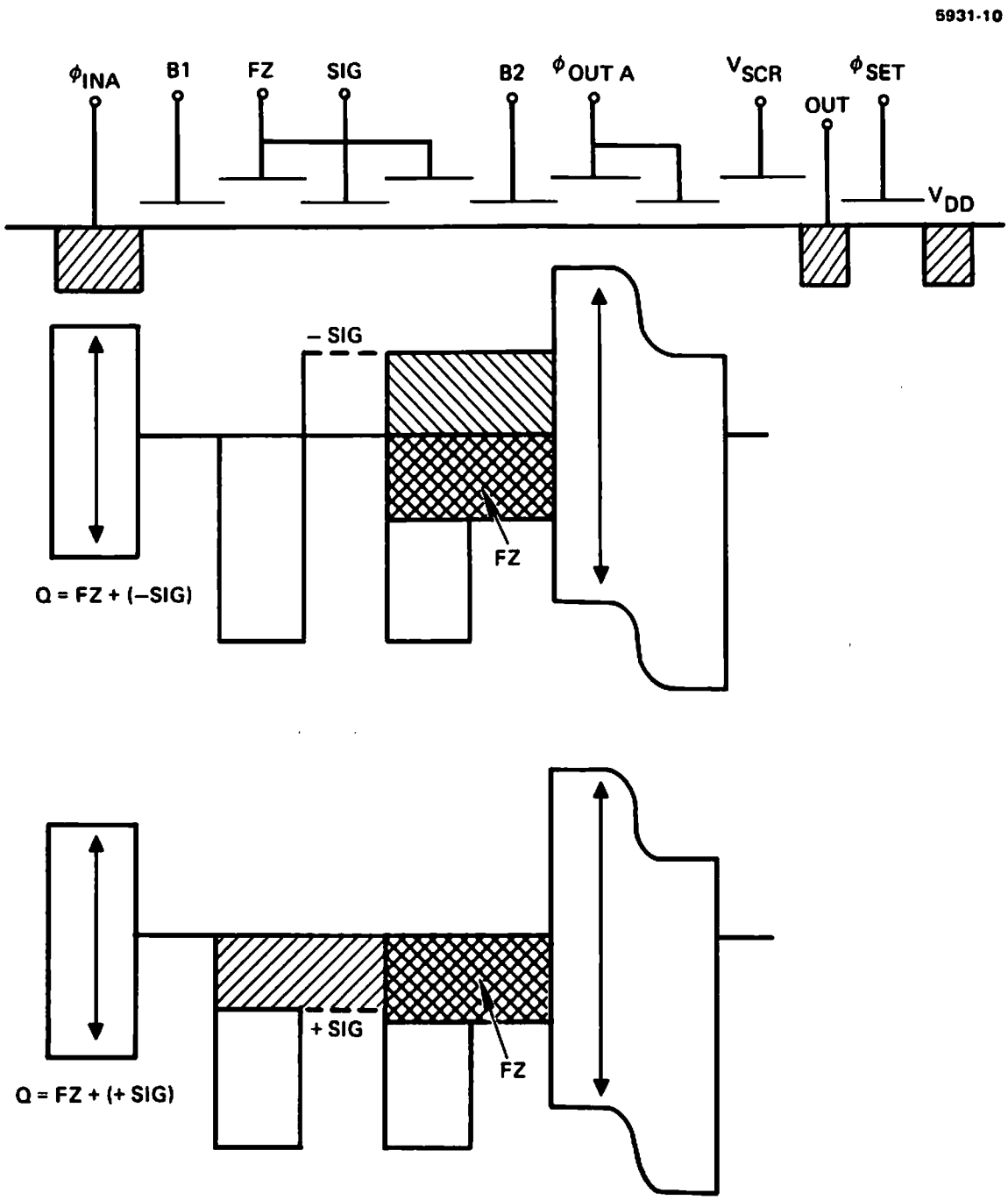


Figure 4. Absolute Value Circuit #1

operation is completed, by again pulsing the input diffusion, charge collects in the well as shown, and the charge indicated by the shaded area is clocked out.

If the total gate area of FZ and SIG is designed to be equal to that of B2 and FZ, equal amount of charge will be transferred for positive and negative signals of the same magnitude. Thus, an absolute value function in the charge domain is obtained.

This implementation has a number of advantages which will materially affect the performance and accuracy of the circuit. For example, it always provides a "fat zero" bias charge packet (indicated by the cross-hatched area) to decrease the transfer inefficiency caused by the surface states in the channel. The level of this "fat zero" is controlled by the d.c. bias applied to FZ.

A preliminary experiment was performed on a simple input circuit on Hughes CRC-100 CCD chip to demonstrate the functional concept described above. This circuit was not designed for performing absolute value functions. Hence, its input gates are not structured for this particular application (see figure 5). However, it illustrates the validity of the concept. Figure 6 is a scope photograph of both the input and output waveforms. It can be seen that the bottom half of the input waveform is inverted in the output. It is also noted that the output waveform is not symmetrical about the zero level due to the asymmetry of the input gate arrangement shown in figure 5. The circuit depicted in figure 4 will provide a more satisfactory operation.

2. Dual Channel

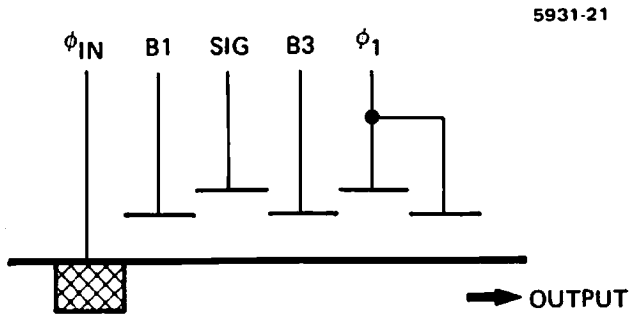


Figure 5. CCD Circuit used for Absolute Value Function Demonstration

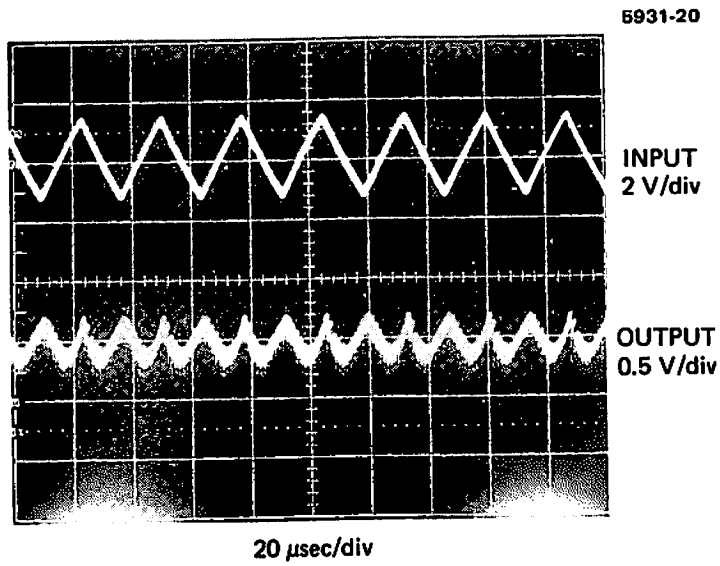


Figure 6. Input and Output Waveforms of a CCD Absolute Value Circuit

The basis of this circuit is two parallel CCD rectifier circuits, one operating on negative-going signals, and the other positive signals, as shown in figure 7. The electrode structure and potential profiles are shown in figure 7a for the positive going signals. Initially the d.c. and signal gates are clamped at a reference potential creating the potential well shown. A bias charge is then dumped, by pulsing ϕ_{IN}, over the input screen filling the well and excess charge spill over the barrier under the signal gate. The signal gate is next connected to an analog input signal. A positive input signal with respect to the d.c. voltage causes a charge proportional to the voltage difference to flow over the barrier. This represents a large dynamic range diode circuit with no forward voltage drop and, hence, rectifies the input signal on the signal gate. The opposite polarity circuit is realized by reversing the position of the d.c. and signal gates as shown in figure 7b. When the two circuits shown are implemented in parallel, a precision full-wave rectifier or absolute value circuit is achieved. The operation is based on the charge equilibration technique and, hence, good linearity, dynamic range and speed should be achievable. A slightly different geometrical arrangement of this concept was discussed in some detail in the previous semi-annual report.

B. Status of Test Circuit I

The detailed design and layout of this circuit was undertaken on a Hughes Aircraft Company IR&D chip CRC 111. This was completed in November 1976, and layout drawings sent to Micro Mask for digitization and mask fabrication at that time. Due to unexpected shortage of manpower at Micro Mask, the chip digitization was delayed for about four weeks and the decision was made to transfer it to Microfab in the first week of January 1977. Digitized cell plots were back

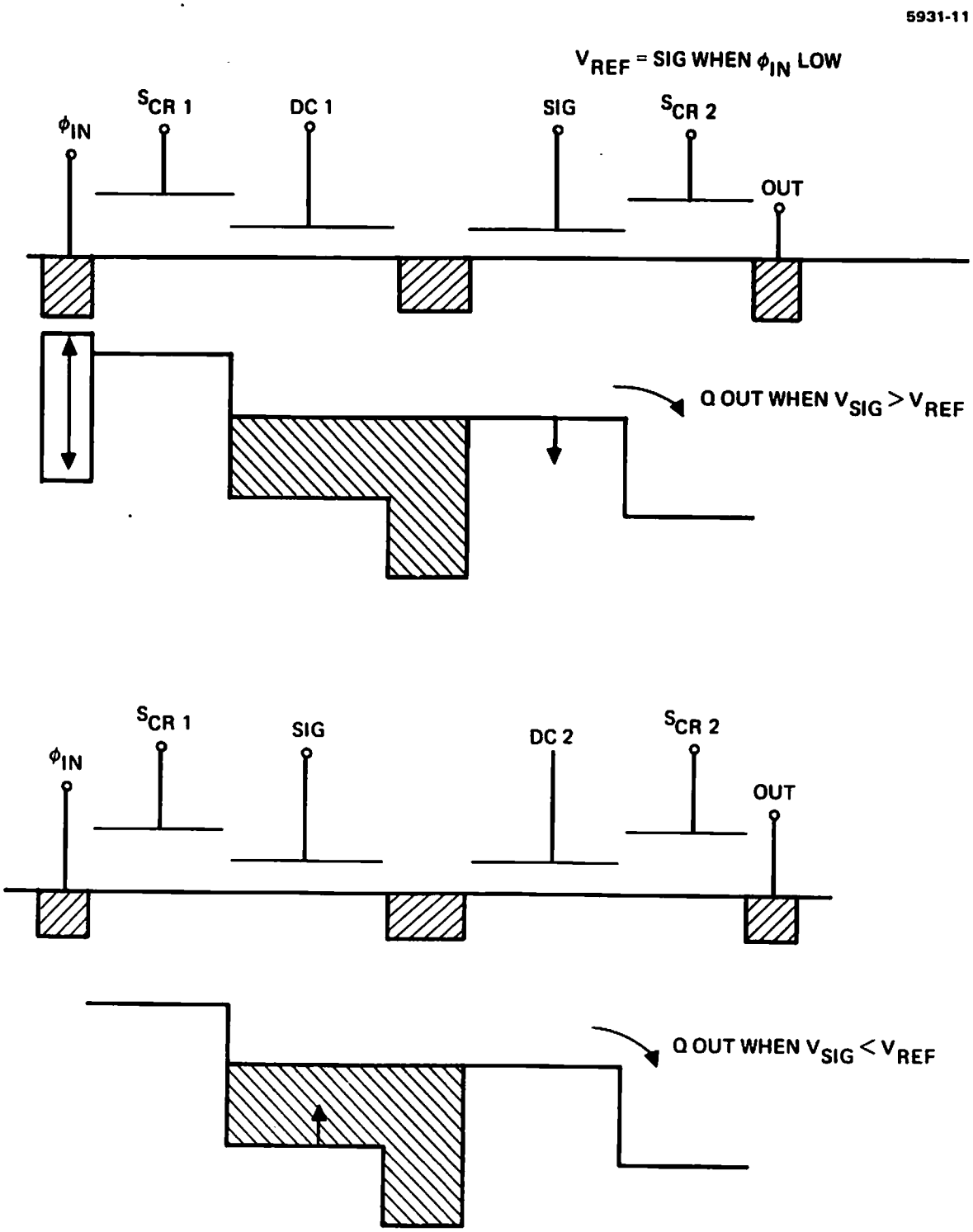


Figure 7. Absolute Value Circuit # 2

for a preliminary check on 9 February 1977. They were checked, corrected and sent back to Microfab for final plots. We anticipate masks to be delivered in March at the latest. Most of the circuits on the CRC 111 chip are DMOS circuits requiring thirteen mask levels, whereas the image processing circuits are n-channel CCD which requires only seven mask levels. We intend, therefore, to process the CCD circuits on a priority basis, and we anticipate processed circuits to be available in April.

III. TEST CIRCUIT II

The detailed design and layout for a second test circuit is currently in progress. The circuit will be processed on the CRC 115, Malibu Signal Processing Chip. This is an n-MOS CCD chip and devices are scheduled to be processed at mid-year. We anticipate that the circuit development cycle for Test Circuit II will be considerably shorter than for Test Circuit I since the primary emphasis of the CRC 115 chip is image processing and its schedule is geared to this program. (Test Circuit I was developed as an adjunct to an IR&D DMOS program.)

The circuit is designed to operate on a 3 x 3 array of pixels and perform the five operations defined in eqs.(1) through (5).

$$\text{Low Pass Filter} \quad f_m(i,j) = \frac{1}{9} \sum_{i-1}^{i+1} \sum_{j-1}^{j+1} f(i,j) \tag{2}$$

$$\text{Unsharp Masking} \quad S(i,j) = (1-\alpha)\, S(i,j) + \alpha\, f_m(i,j) \tag{3}$$

$$\text{Adaptive Binarizer} \quad f_b(i,j) = \begin{cases} 1 & f_m(i,j) \leq f(i,j) \\ 0 & f_m(i,j) > f(i,j) \end{cases} \tag{4}$$

Adaptive Stretching $f_a(i,j)$

$$= \begin{cases} 2 \text{ Min } \left| f(i,j),\ r/2 \right| & f_m(i,j) \leq r/2 \\ 2 \text{ Max } \left| \left(f(i,j) - r/2 \right),\ 0 \right| & f_m(i,j) > r/2 \end{cases} \qquad (5)$$

A block schematic of the circuit is shown in figure 8. It operates on three parallel lines of charge equivalent to three adjacent lines of image data and provides the five separate outputs 01-05 shown. The accuracy of each of these operations is anticipated to be equivalent to six bits. It is built using 5 m lithography and is designed to operate at 10 MHz. The circuit arrangement differs slightly from that shown in the previous semi-annual report in that the edge detection and low pass filtering are performed in parallel rather than serially. This approach avoids any timing problems associated with formation of the unsharp masked output, but requires greater attention to be paid to the linearity and matching of the Tompsett input structures. The circuit philosophy is to provide each of the five output functions independently, as shown, and make the interconnection either with wire bonds on the chip surface or external coax. In this way parallel techniques will be investigated and each function can be isolated and tested separately. For example, two Sobel circuits will be built (one using a HAC proprietary arithmetic technique for charge sensing and calculation), and a number of novel absolute value circuits developed. This will allow us in the initial testing phase to evaluate six different circuit arrangements for edge detection and determine the performance and accuracy of each approach. Then in the final image processing, we will select the optimum approach.

The detailed design and simulation of each of these devices has now been completed and the initial layout for

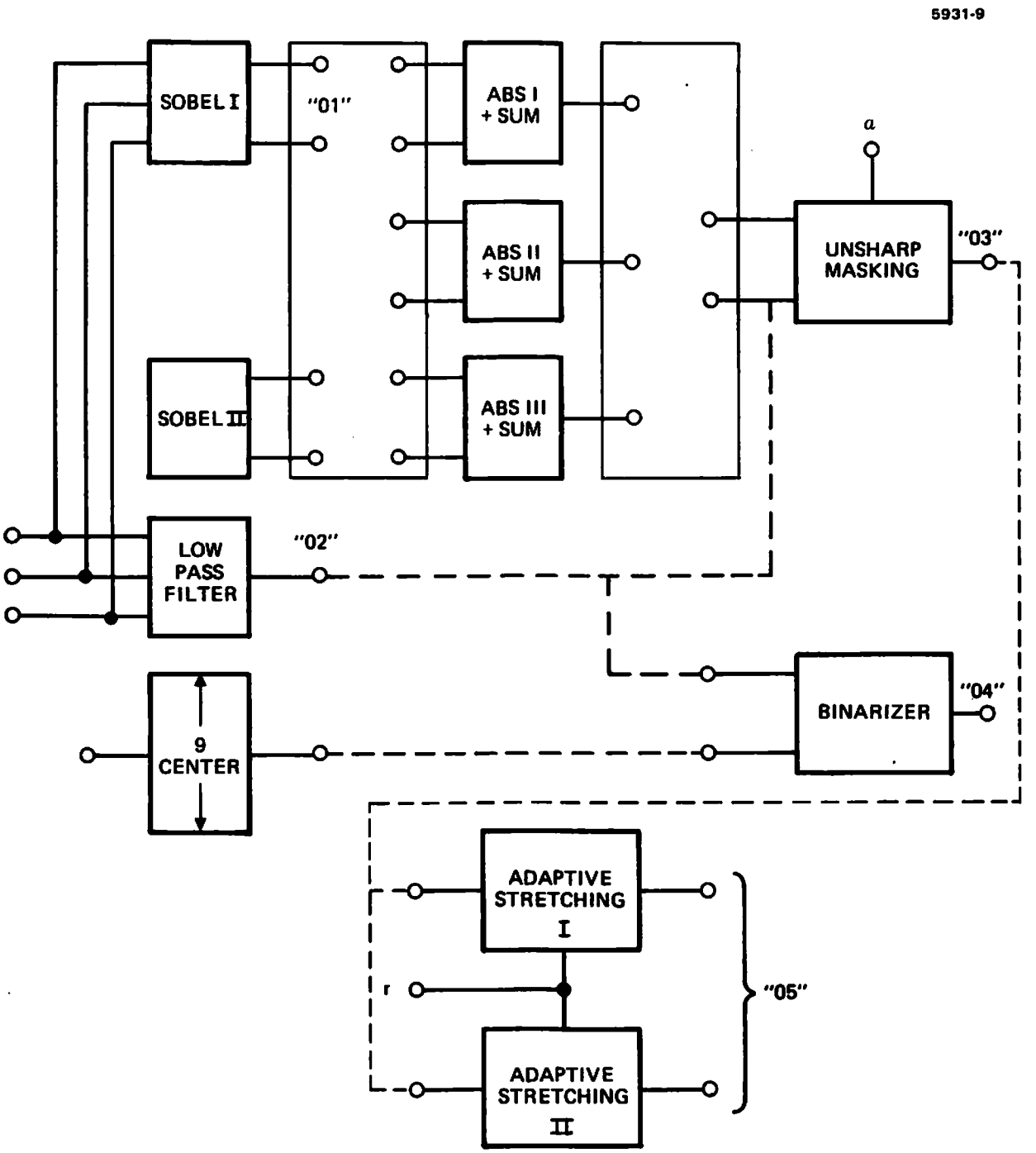


Figure 8. Block Schematic of Test Circuit II

the full circuit is to be completed by March 1st. A brief description of each circuit element is given below.

A. Edge Detection

The edge detection technique is again based on the Sobel operator, and two circuit concepts are being developed. The first is a novel circuit concept which avoids use of much of the MOS circuitry typically involved in the differential amplifiers at the outputs of the floating gate structures. The second technique is the more conventional approach illustrated in figure 3, using twelve floating gate electrodes and employing the area modulation shown to provide the outputs

$$f(i-1,j-1) + 2\,f(i,j-1) + f(i+1,j-1) \quad \text{etc.}$$

Then the differential amplifiers shown in figure 9 find each orthogonal edge component,

$$\left|f(i-1,j-1) + 2\,f(i,j-1) + f(i+1,j-1)\right| - \left|f(i-1,j+1) + 2\,f(i,j+1) + f(i+1,j+1)\right|$$

The advantages of this type of amplifier circuit is high common mode rejection and high speed. This design has been computer-simulated up to 20 MHz. It is estimated that an accuracy equivalent to seven bits will be achieved with a gain of 0.8. The balance achieved between the two input devices is crucial to accurate operation and in the device now being drawn particular emphasis is given to this issue.

Absolute Value Circuits

There are three absolute value circuits included on Test Circuit II. The first two are CCD implementations which operate so as to generate a charge equivalent to the magnitude of the input signal. These circuits have been

described in some detail as part of Test Circuit I. We are also including as part of Test Circuit II the MOS design shown in figure 10. The input in this circuit drives two matched transistors T1 and T2. As shown the input to T2 is inverted and hence the current drawn from V_{DD} flows either through T1 or T2 depending on the polarity of V_{in}. For matched transistors the magnitude of the output V_o is thus independent of the polarity of V_{in}. Again this circuit has been simulated and an accuracy of seven bits is estimated.

Low Pass Filter and Center Element

A schematic of the low pass filter is shown in figure 11. The three floating gate electrodes are used to sense and sum the charge magnitudes in nine adjacent cells. As shown the output represents

$$\sum_{i-1}^{i+1} \sum_{j-1}^{j-1} f(i,j)$$

and hence is nine times the mean. This has been done (rather than make each floating gate a ninth of the full cell size) to increase the sensitivity. It does require a CCD shift register with nine times the width to sense the center pixel as shown in figure 12 to achieve balanced signals.

Unsharp Masking Circuit

The concept of the unsharp masking circuit is shown in figure 13. It is based on the analog multiplier. Externally adjustable inputs (controllable by external power supplies) are fed to transistors T1 and T2 which control the gain of the two input devices T3 and T4. Since these are drawing current from a common source V the voltage of node, N, varies as $(1-\alpha)\, f_s(j,k) + \alpha f_n(j,k)$. The output

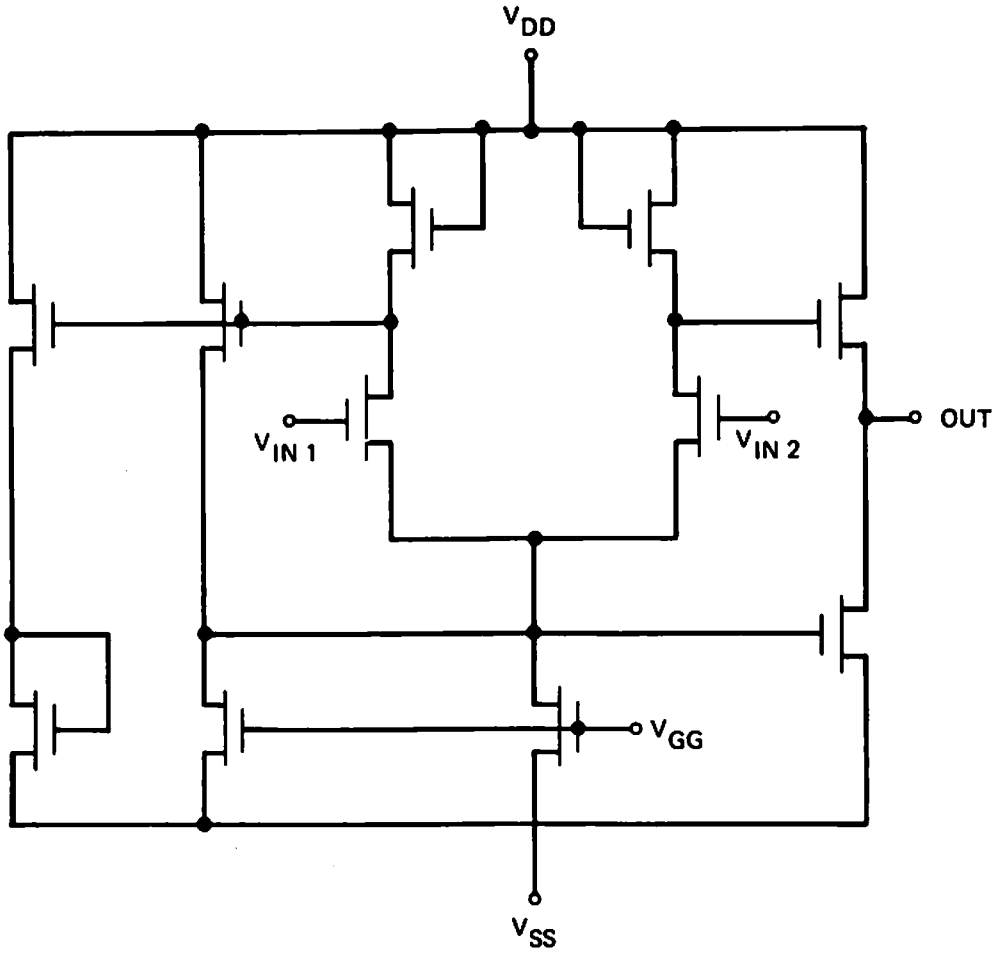


Figure 9. Differential Amplifier

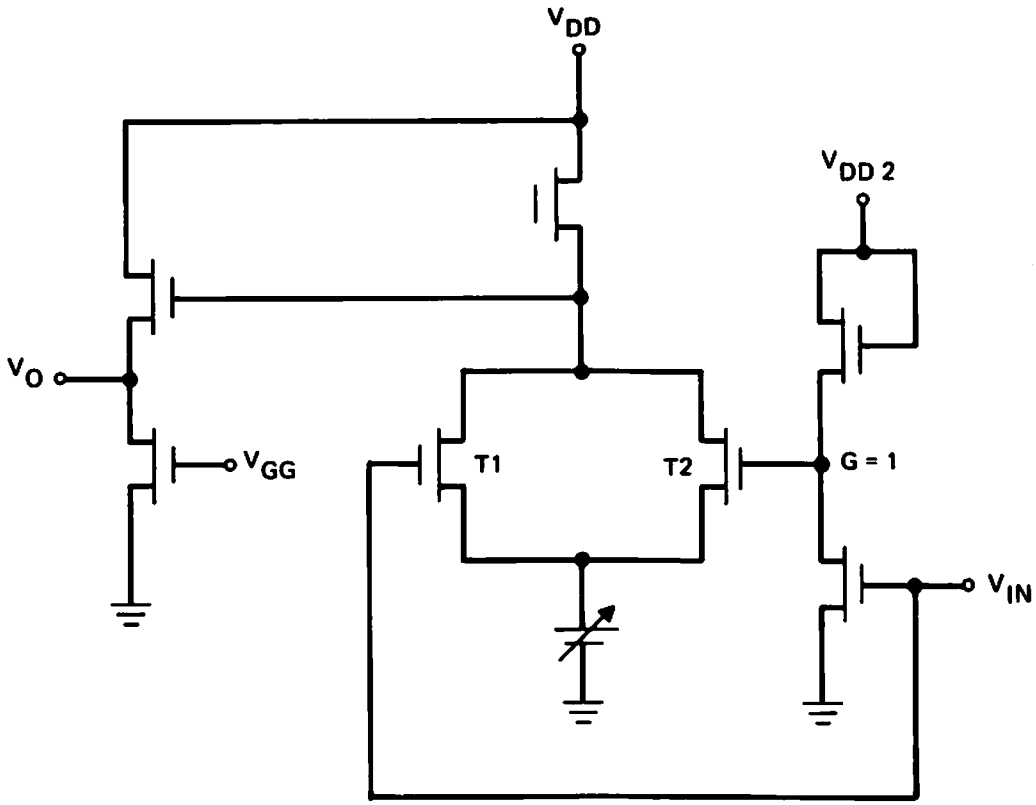


Figure 10. Absolute Value Circuit

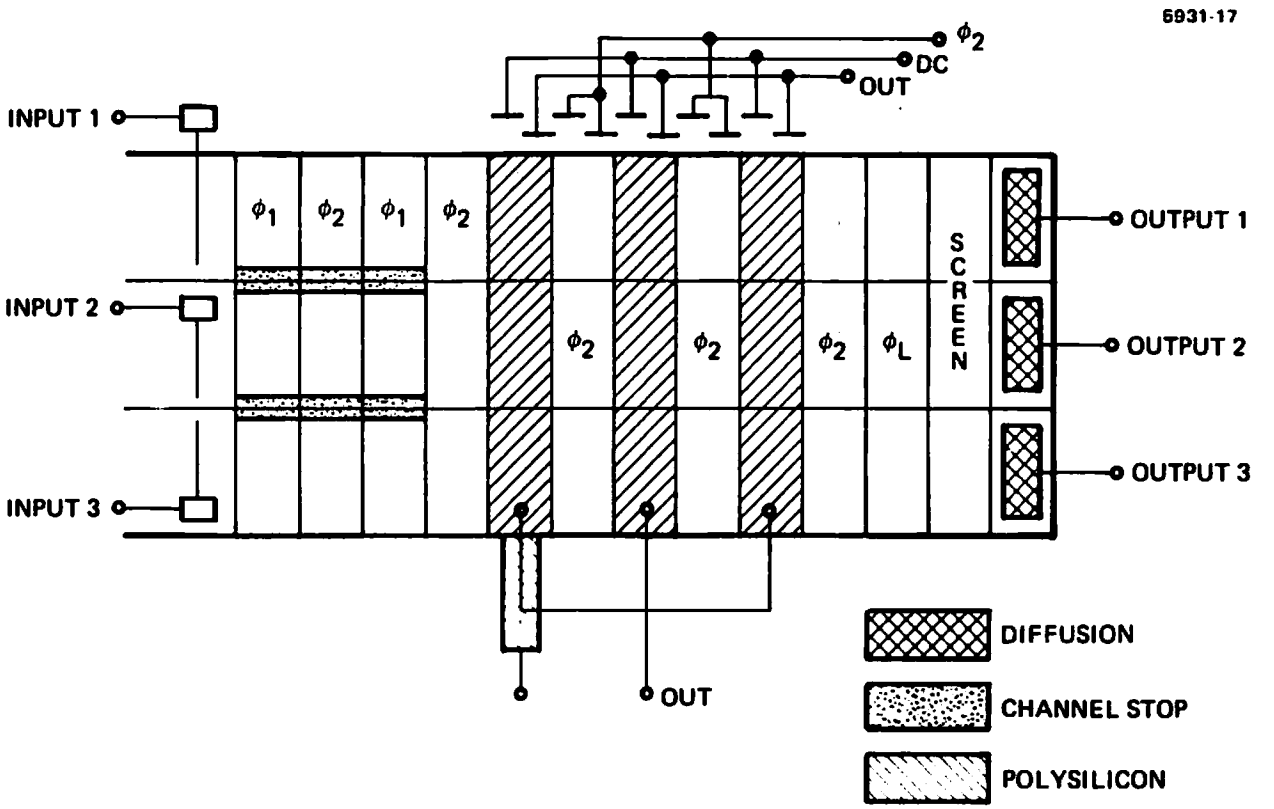


Figure 11. Low Pass Filter

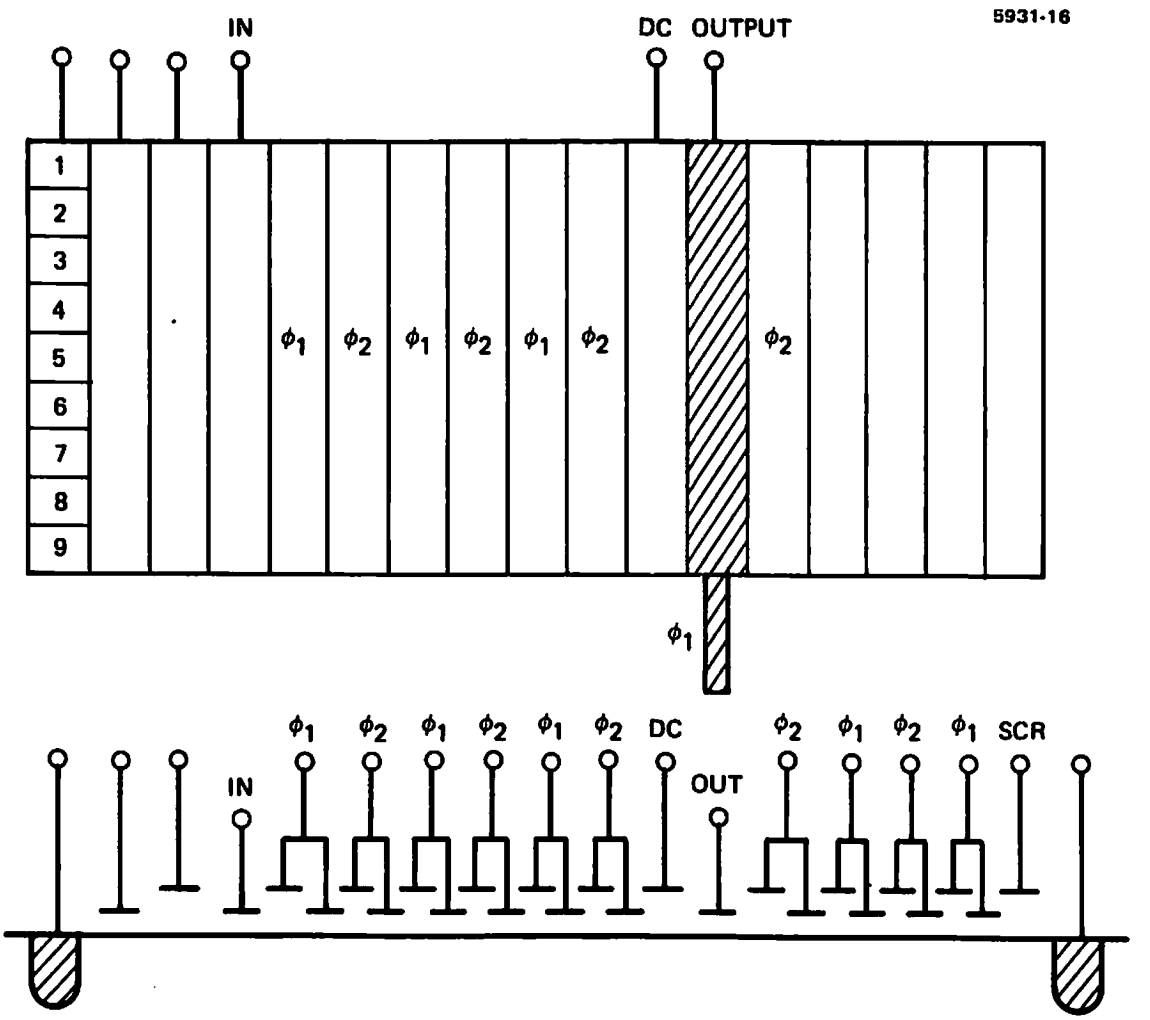


Figure 12. CCD Register to Access the Center Pixel

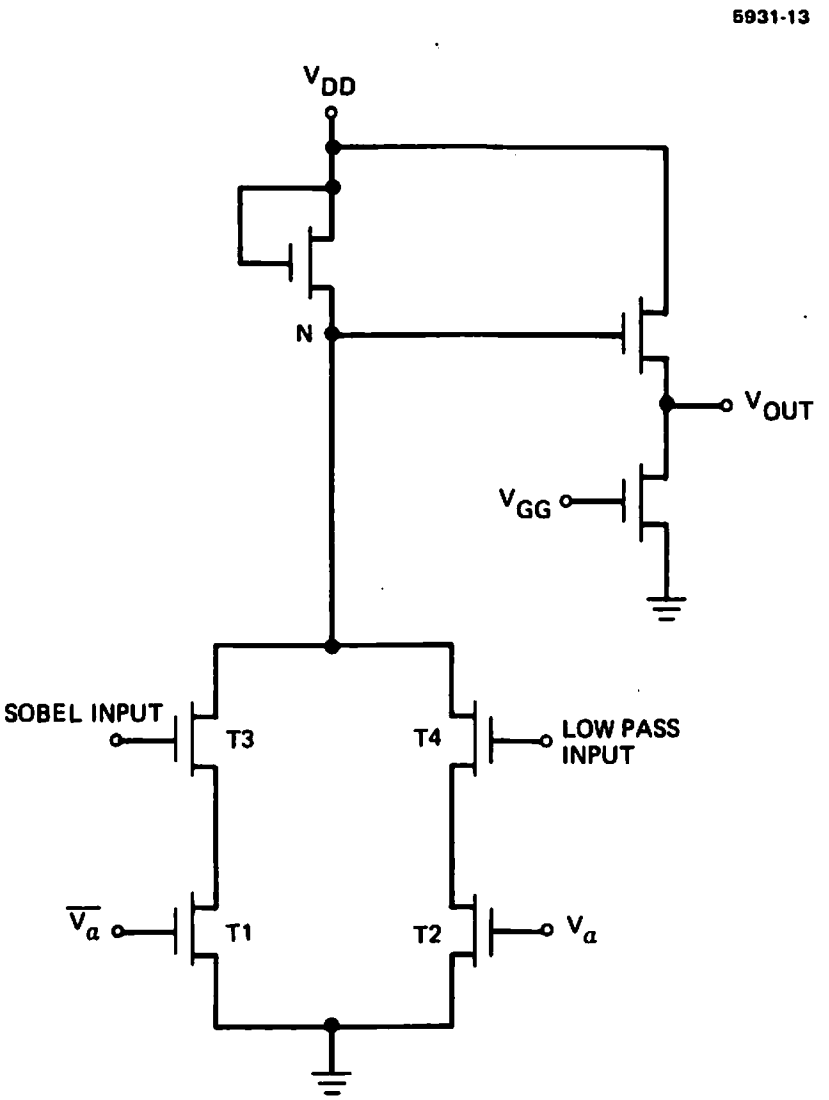


Figure 13. Analog Multiplier Used for Unsharp Masking

from the source follower is thus equivalent to the unsharp masked output as defined by eq.(3). The external control allows the output to vary from all edges to complete low pass output.

Binarizer

The concept of the binarizer has been employed widely as the refresh element for digital CCD memories. Its basic form is shown in figure 14a. The usual accuracy requirement for the digital refresh is comparatively low: merely sensing about a fixed threshold. The accuracy attainable is controlled by the matching of the two symmetrical halves of the circuit and is largely a geometric and threshold problem.

A photograph of a typical digital refresh circuit (taken from Hughes Aircraft CRC 100 chip) is shown in figure 14b where the required symmetry is immediately apparent. Typical MOS threshold variation might be approximately 20 mV and hence seven bit accuracy will require greater than two volts swings.

The binarization requires considerably more accuracy than direct refresh since the switching voltage itself is varying and is likely to be very close to the input signal (one being the center pixel, the threshold being the average of its nine neighbors). We are therefore currently considering using a pre-amplification stage prior to the cross coupled latch shown. An amplification of about five would be sufficient to achieve the necessary accuracy and provide correct latching. This problem is currently being analyzed.

Adaptive Stretching

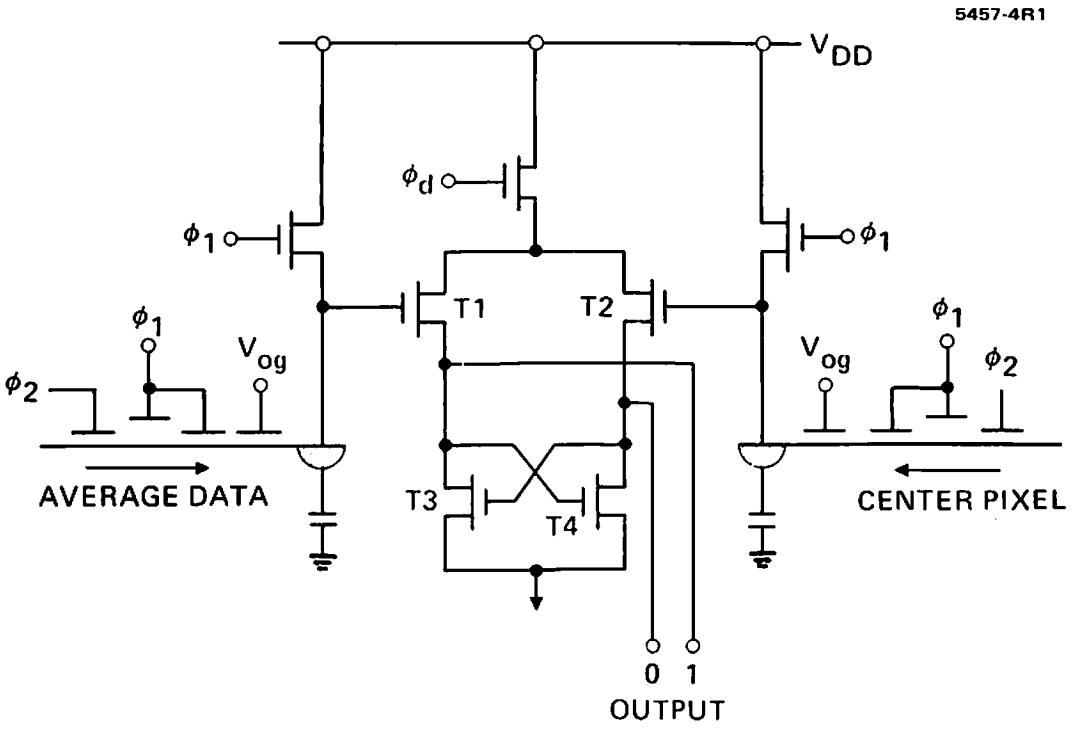


Figure 14 (a) Schematic of Binarizer

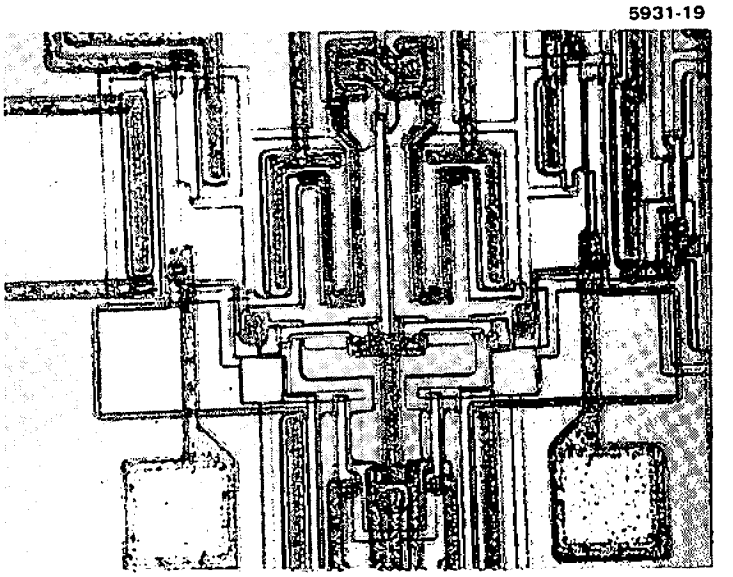


Figure 14 (b) Photo-micrograph of Binarizer

A circuit capable of providing the adaptive stretching function is shown in figure 15. The input signal equivalent to f(i,j) is a.c. coupled to an MOS transistor which is driven by an external voltage V_t. (This input can also be derived from the mean f_m by internal bonding on the chip.) The gain of this circuit is two and the output will be linear until the transistor T1 limits at $f(i,j)|_{max}$ (i.e., input magnitude $f(i,j)|_{max}/2$). The complement of this output is also available which provides a thresholded output (up to $f(i,j)|_{max}$) and then a linear gain of two. These two outputs provide the transfer function shown on page 160 of the September 1976 Semi-Annual Report, isolating the high brightness and shadow regions and can be externally varied by controlling the threshold voltage V_t and the gain, via the source follower input V_g.

Status of Test Circuit II

The detailed design and layout of each of the above functions, is currently underway. We are investigating the optimum binarizer circuit, and have scheduled a detailed design review by March 1st. We anticipate each circuit element will have been finalized at that time. Our present goals call for processed devices by July 1977.

IV. TEST FACILITIES

During the past six months we have spent considerable time developing the test facilities necessary to demonstrate the performance of our CCD circuits on the USC data base. The concept of the system is shown in figure 16. It is based on the IMSAI 8080 microprocessor and interfaces with the USC PDP-10 via a standard 300 baud telephone line. Image data, stored on magnetic tape at the Image Processing Institute, is read by the PDP-10 and transmitted to Hughes

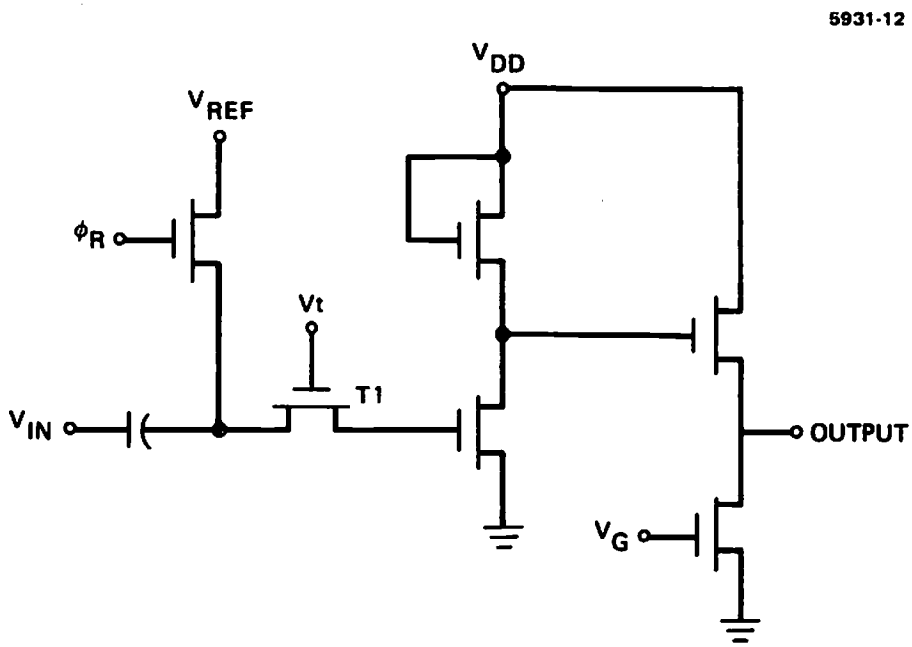


Figure 15. Adaptive Stretching Circuit

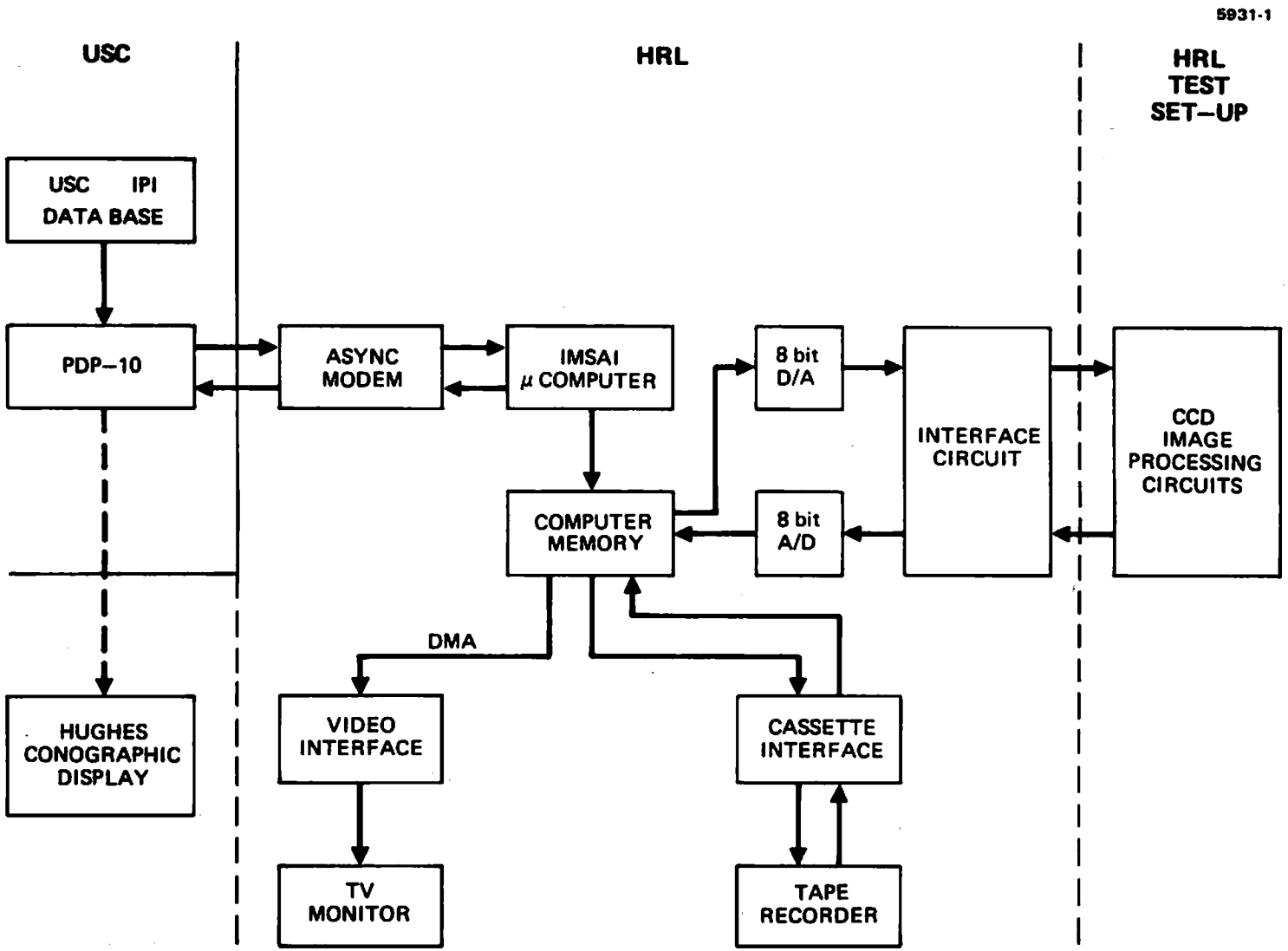


Figure 16. Schematic of Test Set-up

Research Laboratories via the existing telephone tie lines, and stored in the digital memory of the microprocessor. The data can then be displayed on the TV monitor shown, and if required, stored on a cassette tape recorder for later reference. An eight bit digital to analog converter is then used to access the data in the memory and interface with CCD circuits. The processed data from the circuits is then returned to the memory via an analog digital converter as shown.

The circuits themselves are bonded in a 40 pin dual in line package and mounted in a coaxial breakout box, through which the clocking pulses, biases and resets are applied. At the present time all the components shown in figure 16 have been built and interfaced to form the full system. A photograph of part of the system is shown in figure 17. We have also developed the necessary software to interface the PDP-10 with our system, and successfully accessed images from the USC system for both storage and display. The key elements of the system are discussed below.

The Microcomputer

An IMSAI 8080 microcomputer is used as an economical and flexible controller for the test facility. As configured it is capable of directly accessing 64K bytes of memory with a minimum instruction time of two microseconds. The system currently has 16K of static RAM memory with 500 nsec access time. A modem and an asynchronous serial communication interface at 300 baud are used to load the image data in the computer memory, from where it can be loaded into a tape cassette for permanent storage or displayed directly. Images can also be loaded from the magnetic tape at 1500 baud.

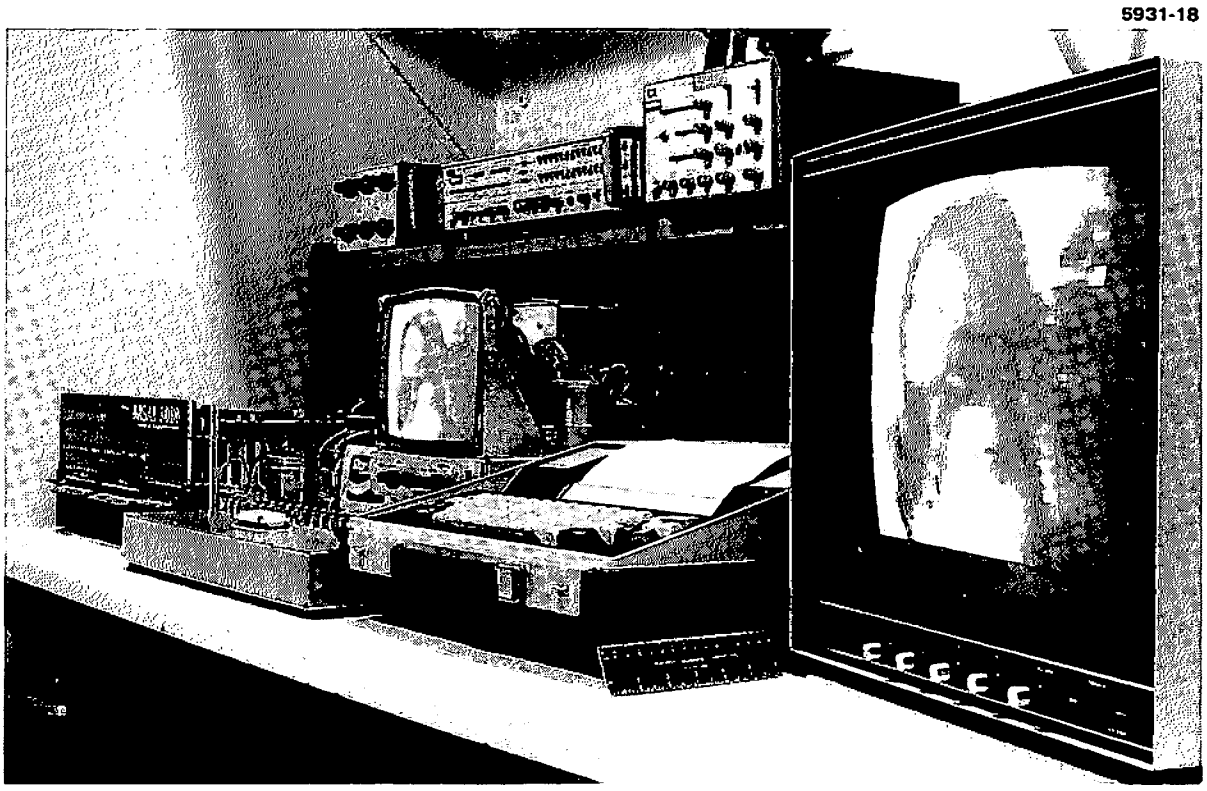


Figure 17. Photograph of Test Set-up

Video Display

At present the system uses a Cromemco TV Dazzler to enable a direct display of both unprocessed and processed images. Direct memory access is used to access the computer memory and read the image data sequentially at standard TV rates. The Dazzler is used to convert to the necessary video format in real time. The system is currently capable of displaying an image with 64 x 64 pixels and 16 grey levels, and we are currently increasing the resolution. (If necessary, we will display the processed data on the Hughes Conographics to obtain the full resolution while the necessary circuit changes are being made.)

Analog to Digital Converters

The system contains two types of analog to digital and digital to analog converters to interface with the CCD circuits. One device is capable of outputting two channels of analog signals with ten bit resolution and inputting eight channels with the same resolution. Another device will output and input seven channels with eight bit resolution with 25 sec conversion time per channel.

CCD Drivers

The necessary clocks and reset pulses for the Sobel circuit are currently being developed. All the waveforms will be generated from a basic square wave clock at 16x the CCD clock rate, and the phase of the diode pulses and resets will be programmable from external switches as illustrated in figure 18. Initially a clock rate of 100 kHz will be employed to interface with the computer memory, but the circuitry is designed to be capable of being run at 10 MHz

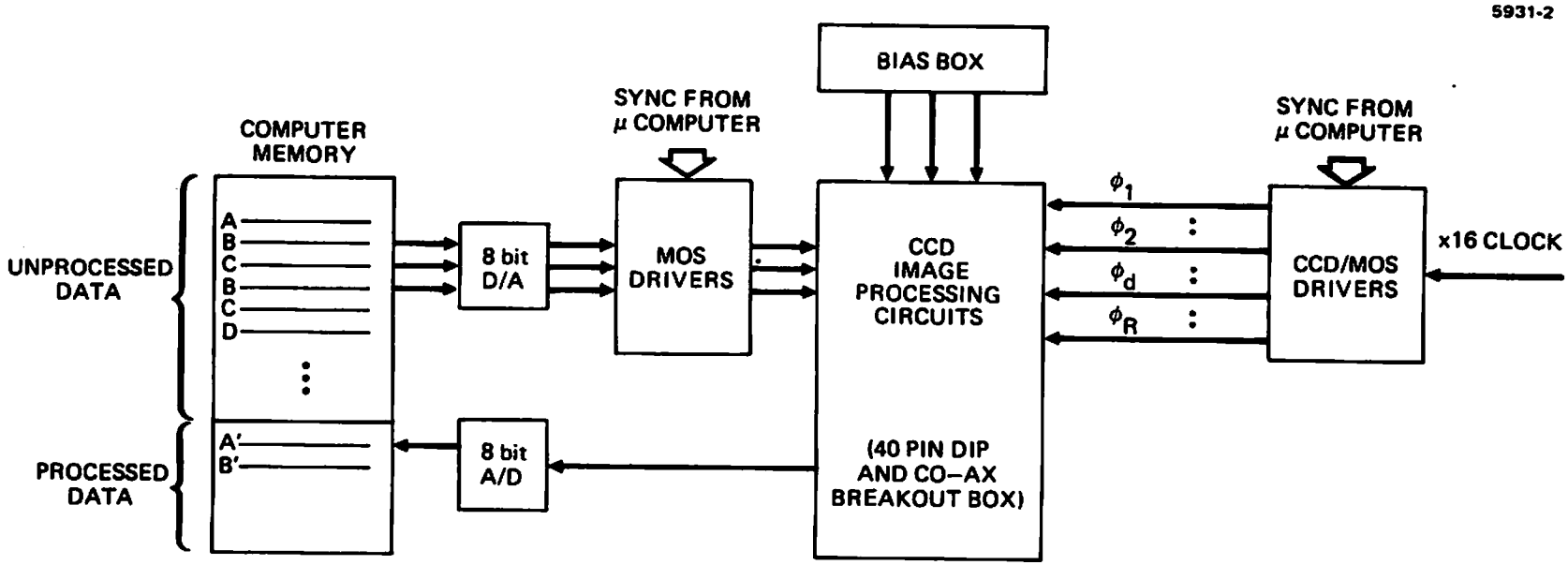


Figure 18. Data Flow in Test Set-up

rates (the normal design rate for our circuits) in later experiments. The basic logic circuits are TTL and high power TTL to MOS output drivers are included to provide the high voltage ≃ 20V required for the MOS circuits.

Software Development

The software for the system consists of a Basic Interpreter, an Assembler, Monitor and Editor to develop the assembly language programs. We have also developed the software necessary to access the USC data base and provide the necessary format translation for the CROMENCO display and the parallel data output required by 3 x 3 operators. The former is required to select the window of interest for our processing and convert from an eight bit format to the storage of two four bit pixels in one byte of the IMSAI memory suitable for display on the monitor. The latter is necessary to avoid having two lines of storage to access the three adjacent lines, as illustrated in figure 18. In a final implementation the storage can employ either an analog CCD shift register or the parallel approach of three adjacent operators each displaced by one line, as in the focal plane processing shown in figure 1.

In the initial phase of the testing program it is anticipated that we will use special test patterns generated on the PDP-10 to evaluate the circuit performance. We have started to generate a library of such images and a package of the computer simulated processed data.

V. PROGRAM STATUS AND FUTURE PLANS

Test Circuit I

We anticipate that processed circuits for Test Circuit I will be available in April at the latest, and our plans call for circuit testing to be commenced at that time. We do not foresee any other delays such as those encountered at the vendors in digitizing this circuit. The testing facilities are currently in place and we are continuing to develop special purpose drivers, etc. as described in Section IV.

Test Circuit II

Our progress on Test Circuit II so far has been better than anticipated and our schedule includes final drawing to be completed in March and masks delivered by mid-May. However we do not anticipate circuit testing to commence before the third quarter.

5. Institute Facilities

Recent interest and external visitor pressure has initiated the following report in this section. Essentially due to academic courses, summer short courses, research efforts and general interest in the USC Image Processing Institute, a brief description of the facilities developed to date are reported herein. A bit of the design philosophy as well as user oriented scenarios are presented for the reader to get a better feel for the capabilities (and limitations) currently available at the USCIPI. For additional details on the laboratories, please consult the various operating manuals and/or cognizant personnel respectively responsible for the various aspects of the Institute.

5.1 The Current Hardware/Software Architecture of the Image Processing Institute's Facilities

Harry C. Andrews

Abstract

The design philosophy of a digital image processing facility should be predicated upon the role such equipment is to ultimately perform. A large percentage of such facilities are dedicated toward production image processing in which parameters and variables are seldom changed. However, this report is devoted to describing an image processing facility whose main objective is the education and research development of graduate students in Electrical Engineering and Computer Science. Toward these goals, systems software, hardware architecture, and structural design philosophies tend to be radically different from production systems. One such digital image processing facility is described in which little, if any, production is experienced, but in which undergraduates, graduates, faculty, and staff users all have "hands on" access to rapidly processed digital imagery results.

I. Introduction

The subject of digital image processing has grown over the past ten years from a few research facilities to large scale computational, display, and interactive exploitation facilities scattered throughout the world. However underlying many such facilities lies the need for the education of competent individuals to effectively utilize the extremely sophisticated equipment and software that goes into the configuration of these facilities. Educational systems for digital image processing equipment are no less

sophisticated but often require different design philosophies and hardware architecture for effective utilization of preciously few available resources. This report then, is directed toward the genesis and continued development of one such facility solely dedicated to educational and research goals of students pursuing graduate degrees.

The general design philosophy behind this digital image processing facility has been predicated upon the need to service many users simultaneously, provide rapid visual access to processed pictorial results, handle large data arrays while simultaneously providing mass storage for easily accessible intermediate processed image results. In addition, due to the hectic pace of faculty, staff and student life, maximum efficiency of the user's personal time has been attempted to be optimized while still providing high quality hardcopy picture input and output results.

These design constraints have led to a central processing facility based on highly interactive time sharing, large core and fast disk storage with direct hardwired access to programming CRT terminals in user offices and hardwired access to remote digital refresh display devices for viewing of intermediate results prior to requests for hardcopy photographic prints. The software system that makes the facility useful is designed around a TENEX operating system with a few optimized routines directly related to image processing tasks. A parallel design philosophy has been utilized to allow minor image processing tasks to be implemented in an off-line mode for highly interactive fast turnaround exploitation using local processing capabilities in order to off-load the central facility from mundane but large bandwidth I/O tasks.

II. Hardware

Figure 1 presents the block diagram of the computing facility under discussion. The central processing unit centers around a PDP KI-10 with 512K words of core memory and fast disk storage of up to 128 million 36-bit words or the equivalent of approximately 2000 images each with 512 x 512 pixels of 8 bits of brightness per pixel. The computing facility is switched through a network of PDP 11 series mini-computers for communication to other peripheral pieces of equipment as well as the office terminals and real time digital display devices. The office terminals provide a unique convenience which coupled with the interactive text editors make keypunching and IBM cards obsolete. The real time digital display devices; two low resolution monochrome (256 x 256 x 6) and two high resolution color (512 x 512 x 8 x 3) monitors, provide viewable results of processing algorithms within seconds and, at most, minutes of program completion and output onto the large disk files. Consequently job turnaround from image in to image out can be experienced in half hour time frames rather than hours or even days experienced in earlier systems.

For off-line processing the equipment connected to the 100K bps line can be switched to a "local exploitation facility" mode in which highly interactive but computationally simple processing algorithms can be exercised. In addition the off-line processing mode provides for hardcopy output on the densitometer and flying spot display and hardcopy input on the densitometer and color facsimile scanner.

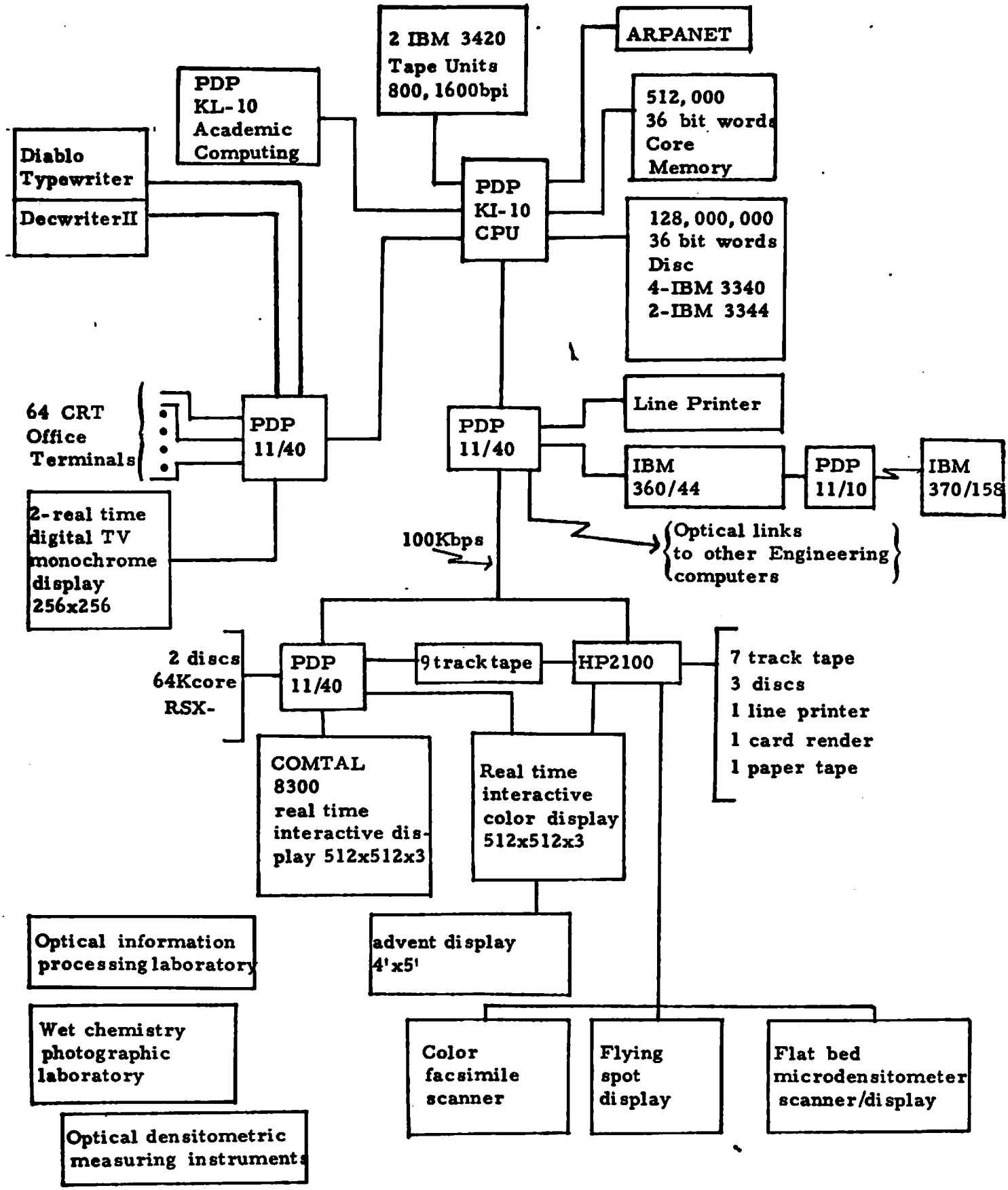

2 IBM 3420
Tape Units
800, 1600bpi
ARPANET
PDP
KL-10
Academic
Computing
512, 000
36 bit words
Core
Memory
Diablo
Typewriter
DecwriterII
PDP
KI-10
CPU
128, 000, 000
36 bit words
Disc
4-IBM 3340
2-IBM 3344
64 CRT
Office
Terminals
PDP
11/40
PDP
11/40
Line Printer
IBM
360/44
PDP
11/10
IBM
370/158
2-real time
digital TV
monochrome
display
256x256
100Kbps
Optical links
to other Engineering
computers
2 discs
64Kcore
RSX-
PDP
11/40
9 track tape
HP2100
7 track tape
3 discs
1 line printer
1 card render
1 paper tape
COMTAL
8300
real time
interactive dis-
play 512x512x3
Real time
interactive
color display
512x512x3
Optical information
processing laboratory
advent display
4'x5'
Wet chemistry
photographic
laboratory
Color
facsimile
scanner
Flying
spot
display
Flat bed
microdensitometer
scanner/display
Optical densitometric
measuring instruments

Figure 1

III. Software

The software system's design is based upon a TENEX time sharing operation in which users are serviced simultaneously. Because of the experimental nature of research software, very few programs are run consecutively without modification. Consequently an efficie t text editor and interactive CRT terminals allow for easy program modification with a minimal of effort. However, what one gives up in this software mode is the large batch number-crunching capability of sequential batch machines. Such jobs on this system are usually queued for third shift runs when the user load is down, larger time slices are available in the CPU, and the dollar accounting is more favorable.

Very little special purpose software has been developed on the system other than image file transfers to the display device. However one assembly language set of subroutines implemented on the PDP KI-10 has made considerable improvement both in case of user programming and in efficiency of mass storage. A typical use of this set of software might be:

```
CALL IPRESS(A,LENGTH,BITS)
CALL DSKIO(A,LENGTH,LINE,0,FILE,BITS)
CALL DSKIO(A,LENGTH,LINE,1,FILE,BITS)
CALL IXPAND(A,LENGTH,BITS)
```

This sequence packs a line of imagery in the array A of length equal to LENGTH into contiguous blocks of bits per pixel equal to BITS with subroutine IPRESS. DSKIO then writes out the packed array A into the line numbered LINE

and onto an image file named FILE. The second DSKIO reads into the array A from image row named LINE. Finally IXPAND expands the packed array into 36-bit PDP KI-10 computer words for conventional integer or floating point processing. This sequence of four subroutine instructions makes image processing software virtually available to even the most novice of users.

One additional aspect of the system's software that makes the configuration particularly useful for image processing is the user's ability to view his image files stored on the high speed disks with real time digital TV technology with simple "file-to-monitor" transfer routines. For display on the monochrome 256 x 256 monitors the real time television (RTTV) is called resulting in the transfer of the selected user image to the requested monitor in a matter of seconds. Similar instructions exist for transfer of 512 x 512 color digital images to either of the high resolution COMTAL systems depicted in figure 1. Once such transfer is accomplished, users usually like to go into the local mode for on-line additional processing of their results in the exploitation facility scenarios.

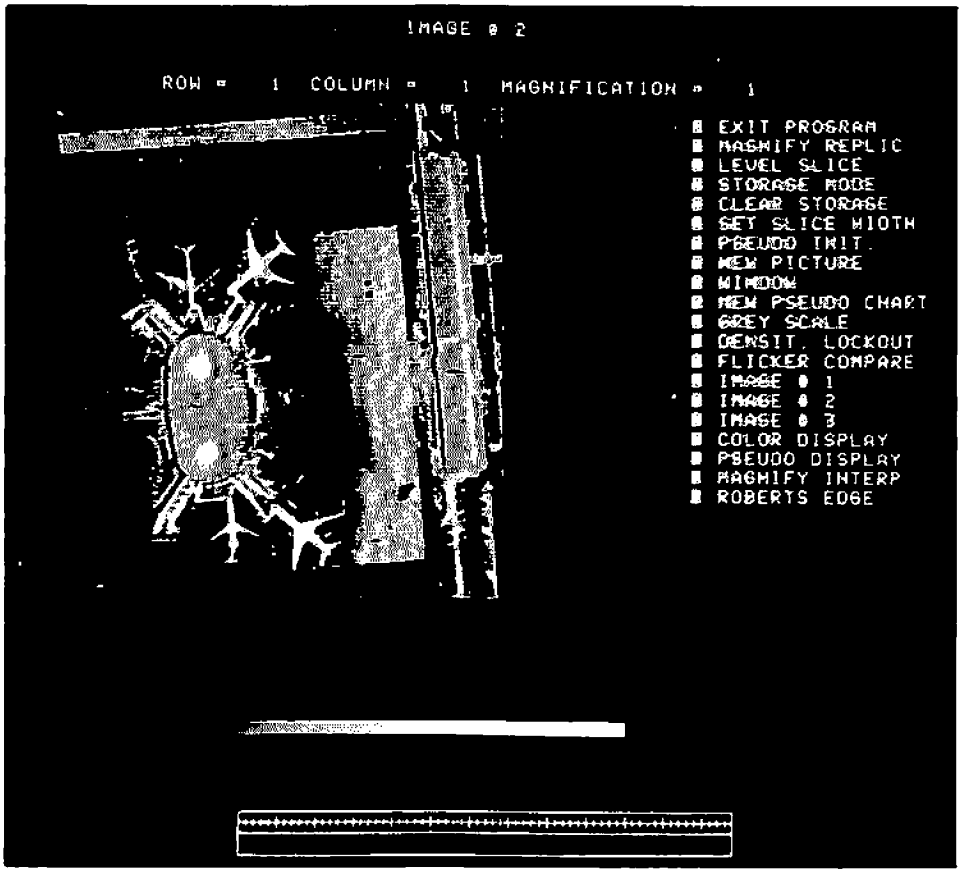
IV. Exploitation Facility

The items in the lower portion of figure 1 (connected to the 100K bps line) can be operated independently of the large computers and as such provide the possibility for highly interactive scenarios to be developed for local processing results. Such processes as histogram gathering, grey scale and color remapping, small convolutions, pseudocoloring, operator-defined object outlining, etc. are all easily implemented on the PDP 11/40 display stations. One particular sequence that allows for operator/user closed

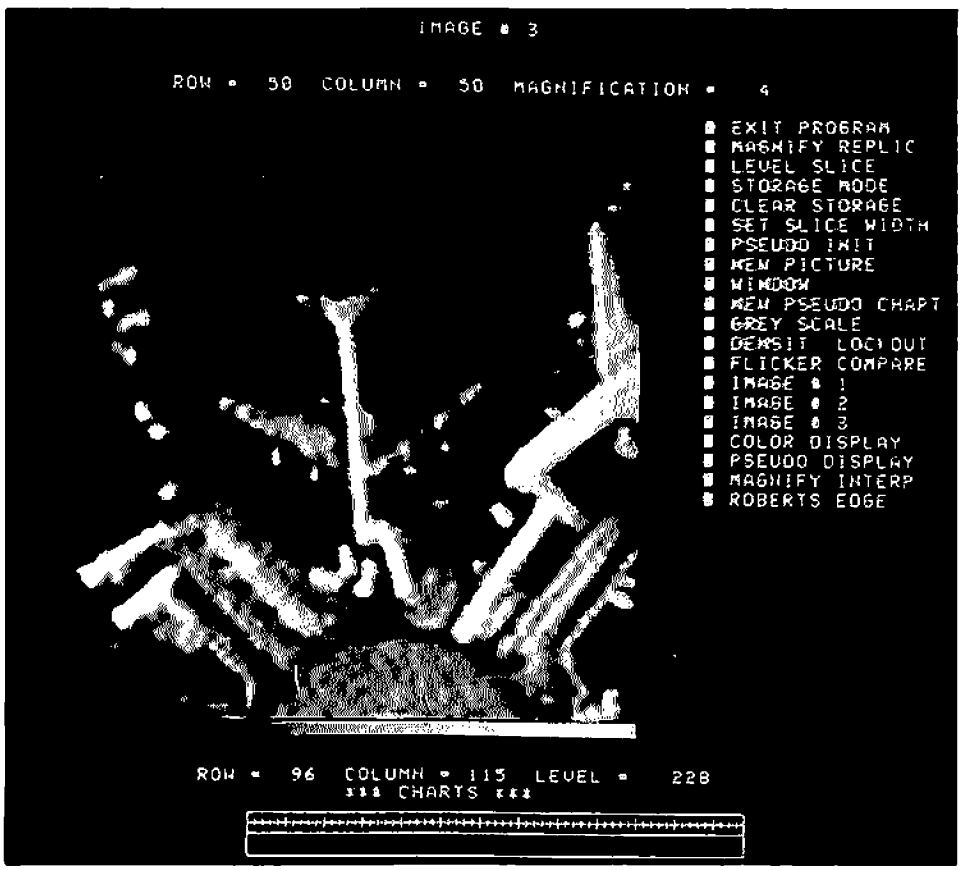
loop processing with the large computing facility is interactive file generation in the exploitation facility for retransmission back to the PDP KI-10 for incorporation in more powerful processing algorithms. One such application where this is particularly useful is in the situation where a user interactively segments an image (via manual trackball operations) for retransmission as "segmentation ground truth" for comparison with completely automatic algorithms being developed on the larger computing facility.

As mentioned previously, one of the major goals of the current systems configuration is the need for minimal training on the user's part to make early effective use of the facility. With this goal in mind, a picture analysis (PICAN) operating system has been developed for one of the exploitation scenarios on the interactive display system, an example of which appears in figure 2. This "menu" operating system has options which are presented on the monitor in a graphics overlay channel in a column on the right hand side of the screen. The user simply places the trackball in the appropriate box beside each option, and the computer implements the request. If quantitative data is desired as input, the computer makes the appropriate request on the terminal. For some variable inputs the grid at the bottom of the screen can also be used in a semi-quantitative fashion.

Returning to the figure, we see the original image in figure 2a with the trackball (the bright spot to the left of the square indicating "image 2") having selected the second image plane for display. Because this is the original, its reference point is row = 1, column = 1 and 1x magnification (see the top of the screen). In figure 2b we see a 4x magnification referenced to row = 50, column = 50 (in the original). The trackball is now within the image field of

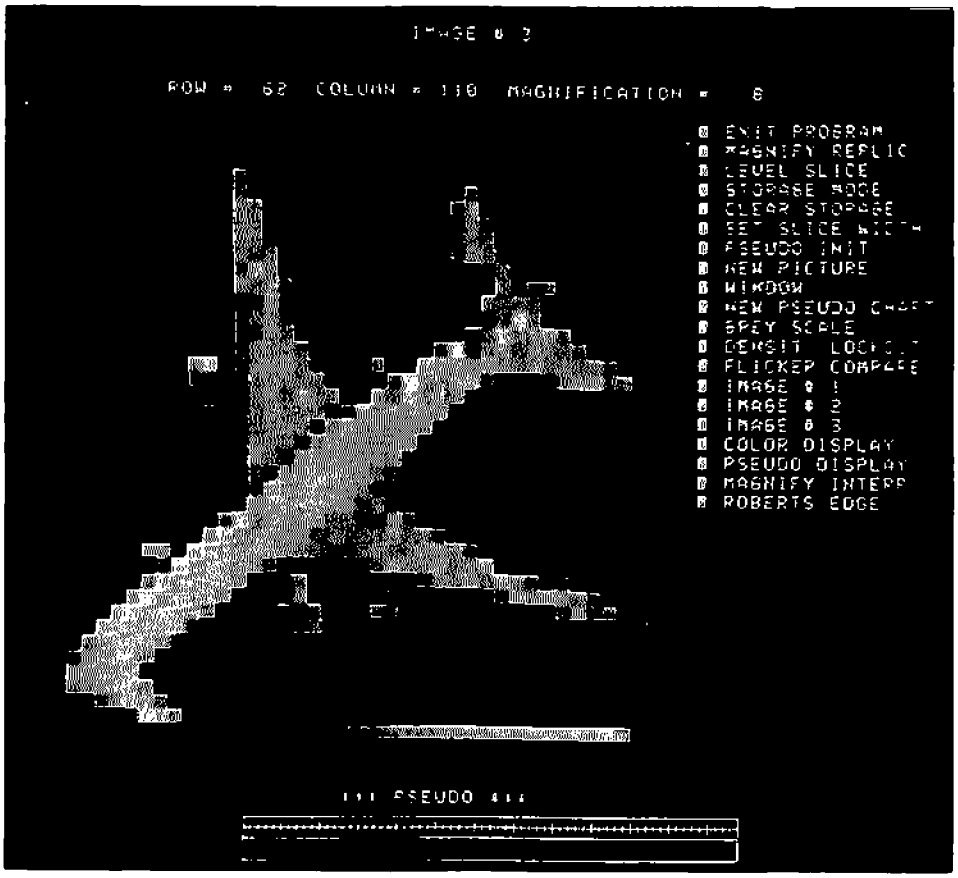


a) Original

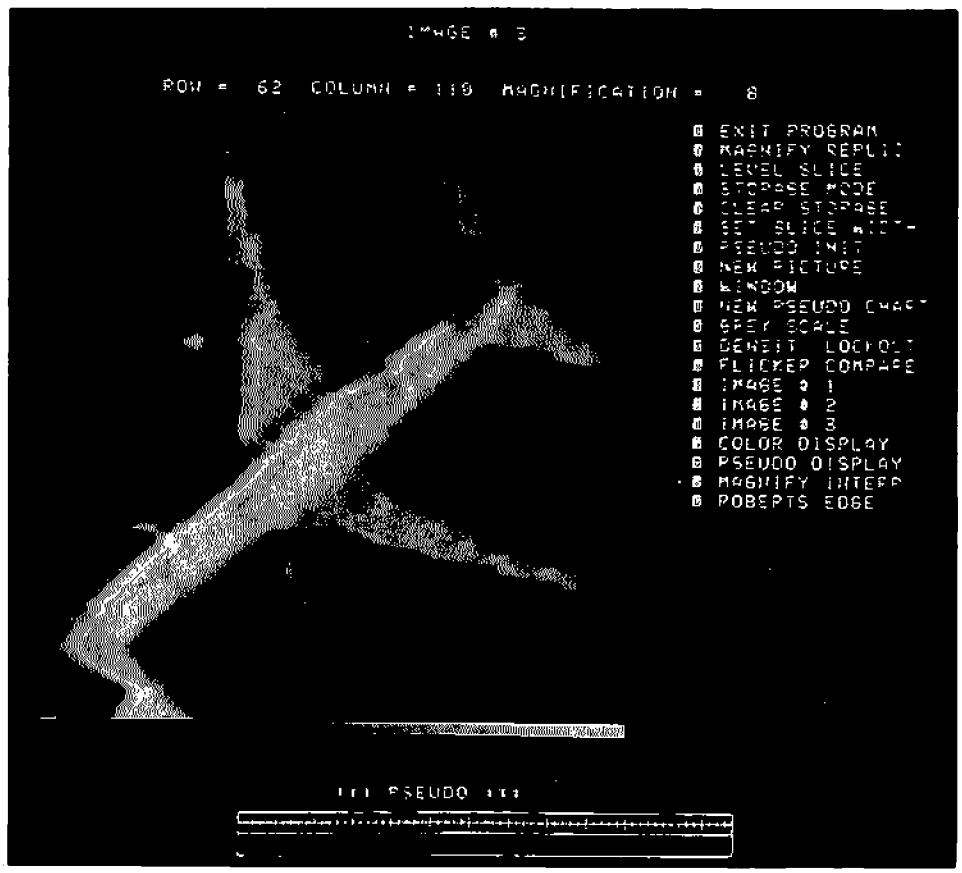


b) 4x Magnification

Figure 2. Interactive Scenarios



c) 8x Magnification - Replication



d) 8x Magnification - Interpolation

Figure 2 (cont). Interactive Scenarios

view at row = 96, column = 115 and is measuring the brightness value at that location to be 228 out of 255. In figure 2c, we see an 8x magnification of an aircraft in the original via replication. The trackball is adjacent to the "option box" entitled "magnify replicate". The reference image location of the aircraft is at row = 62, column = 110. Repeating the 8x magnification process on the same reference point via bilinear interpolation provides us with figure 2d (note the trackball position to the left of "magnify interpolate"). All of these options are interactively selected, require only seconds of PDP 11/40 time and could be easily constructed in digital video real time hardware (1/30 second implementation). Clearly many other "menu" selection options are available but these presented here are illustrative of some of the more simple techniques.

The interactive exploitation scenario developed under PICAN has evolved over the years to become a very effective picture analysis tool for relatively inexperienced users. However another aspect of digital image processing has similarly evolved during this time frame which capitalizes on such scenarios. Specifically when mankind's mathematical and analytic tools reach a useful limit, but technological problems still remain to be solved, most often further breakthroughs are accomplished by placing a human as intimately in the signal processing loop as possible and allowing his intuitive capabilities to take over. Such situations were extremely prevalent in one-dimensional signal processing applications in the past such as in sonar, radar, waveform analysis, and nonstationary signal detection in general. By analogy some of the truly complex "image understanding" tasks of today will require similar "human in the loop" procedures to effectively learn how and why a human is so expert at understanding images. However, until quite recently digital devices did not have the bandwidth

and speed to keep up with human visual processes, and as such effective use of his intuitive processes have not been made. However with responsive highly interactive exploitation scenarios, and with proper computer monitoring, it is now becoming possible to configure such systems for this next stage of image understanding.

V. Conclusions

This report has attempted to present the design philosophy behind the system and software configuration of a digital image processing facility devoted to educational and research objectives. Time sharing operating systems seemed to present the most efficient and economical software solution while large disk storage and direct image I/O became useful hardware devices. An off-line exploitation station for interactive image processing was also described for non-number-crunching objectives. The configuration thus described in this report has been in existence for three years with the result of minor modifications in evolving hardware and software improvements and currently represents an efficient economical facility for image processing, teaching and research.

VI. Acknowledgements

This facility described herein has had many individuals and organizations contribute to its configuration and growth. Throughout the major phases of its evolution both the Advanced Research Projects Agency of the Department of Defense and the School of Engineering of the University of Southern California have contributed their resources for both personnel and capital equipment expenses. This entire

effort was directed by Professor W.K. Pratt without whose guidance the facility would never have been developed.

6. Recent Ph.D. Dissertations

This section includes those dissertations completed since the last reporting period. The one listed here reflects an effort at utilizing two-dimensional approximation theory to much more effectively develop adaptive techniques for efficient image approximations. The results of the research are immediately applicable to high resolution sensors in which channel bandwidth does not permit transmission of the Nyquist resolution everywhere. By on-board variable knot sampling adaptive approximations to the high resolution image are obtained with low dynamic range coefficients. In addition the knot (or sample) density provides a valuable feature for potential on-board segmentation and higher level decision processes.

6.1 Degrees of Freedom of Images and Imaging Systems

Dennis G. McCaughey

Abstract

This dissertation presents a degree of freedom or information content analysis of images and imaging systems in the context of digital image processing. As such it represents an attempt to quantify the number of truly independent samples one gathers with imaging devices.

In quantifying the degrees of freedom of an imaging system it is necessary to develop an appropriate model. In this work the imaging system is modeled as a linear system through the continuous-discrete imaging equation. The associated gram matrix is then employed as an aid in defining the system degrees of freedom. The gram matrix eigenvalues are shown to be related to those of the associated continuous-continuous model and can be used to predict the discretized system performance. These ideas are then applied to the tomographic or projection imaging system; and result in the ability to predict the performance of this system by indicating where redundant data is achieved, and the best ways of increasing the degrees of freedom with a minimum sample increase.

The degrees of freedom of a sampled image itself are developed as an approximation problem. Here bicubic splines with variable knots are employed in an attempt to answer the question as to what extent images are finitely representable in the context of a digital computer.

Relatively simple algorithms for good knot placement are given, and result in spline approximations that achieve

significant parameter reductions at acceptable error levels. The knots themselves are shown to be useful as an indicator of image activity, and have potential as an image segmentation device.

7. Recent Institute Personnel Publications

1. S.R. Dashiell and A.A. Sawchuk, "Nonlinear Optical Processing: Analysis and Synthesis," Applied Optics, to appear in Vol. 16, April 1977.

2. S.R. Dashiell and A.A. Sawchuk, "Nonlinear Optical Processing: Non-Monotonic Halftone Cells and Phase Halftones," to appear Applied Optics.

3. S.R. Dashiell and A.A. Sawchuk, "Nonlinear Optical Processing: Effects of Input Medium and Precompensation," to appear in Applied Optics.

4. C.F. Hall and E.L. Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-7, No. 3, 1977, pp. 45-52.

5. H.S. Hou and H.C. Andrews, "Least Squares Image Restoration Using Spline Basis Functions," IEEE Transactions on Computers, 1977.

6. D.G. McCaughey and H.C. Andrews, "Degrees of Freedom for Projection Imaging," IEEE Transactions on Acoustics, Speech, and Signal Processing, February 1977.

7. D.G. McCaughey and H.C. Andrews, "Variable Knot Splines for Image Approximation," submitted for publication.

8. F. Naderi and A.A. Sawchuk, "Estimation of Images Degraded by Film-Grain Noise," submitted to Applied Optics.

9. N.E. Nahi and S. Lopez-Mora, "Estimation of Object

Boundaries in Noisy Images," Proceedings of the 1976 IEEE Decision and Control Conference, Clear Water, Florida, December 1976, pp. 607-612.

10. R. Nevatia, Computer Analysis of Scenes of 3-Dimensional Curved Objects, Birkhauser-Verlag, Basel, Switzerland, 1976.

11. R. Nevatia, "Locating Object Boundaries in Textured Environments," IEEE Transactions on Computers, November 1976, pp. 1170-1175.

12. R. Nevatia, "A Color Edge Detector," Proceedings of the Third International Joint Conference on Pattern Recognition, November 1976, pp. 829-832.

13. R. Nevatia and T.O. Binford, "Description and Recognition of Curved Objects," Artificial Intelligence Journal, Vol. 8, 1977, pp. 77-98.

14. R. Nevatia, "Evaluation of a Simplified Hueckel Edge-Live Detector," accepted for publication in Computer Graphics and Image Processing.

15. C.L. Patterson and H.C. Andrews, Special Issue on Digital Image Processing, Computer, Summer 1977.

16. W.K. Pratt and F. Davarian, "Fast Computational Techniques fo Pseudoinverse and Wiener Image Restoration," accepted for publication in IEEE Transactions on Computers, June 1977.

17. W.K. Pratt, Digital Image Processing, Wiley-Interscience, submitted for publication.

18. E.M. Rounds, "Image Synthesis of Adjacent Regions from Their Contours," accepted for the IEEE Conference on Image Processing and Pattern Recognition, June 1977.

19. E.M. Rounds and S. Ginsburg, "Dynamic Syntax Specification Using Grammar Forms," submitted for publication.

20. A.A. Sawchuk and H.C. Andrews, "Artificial Stereo Images," 1976 Annual Meeting, Optical Society of America, Tucson, Arizona, October 1976, Journal Optical Society of America, Vol. 66, p. 1111, October 1976.

21. A.A. Sawchuk, "Real-Time Correction of Intensity Nonlinearities in Imaging Systems," IEEE Transactions on Computers, Vol. C-26, January 1977, pp. 34-39.