IN-59
58215
P-76

NASA Contractor Report 4420

Analysis of Objects in Binary Images

Desiree M. Leonard

CONTRACT NAS1-19038
DECEMBER 1991

NASA Contractor Report 4420

Analysis of Objects in Binary Images

Desiree M. Leonard
Computer Sciences Corporation
Hampton, Virginia

Prepared for
Langley Research Center
under Contract NAS1-19038

National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1991

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

Figure **Page**

PREFACE

The purpose of this document is to describe blobtool - an image analysis application package developed for the NASA/Langley Research Center's Image Processing Laboratory (IPL). Detailed are the techniques and algorithms utilized by the package.

The document is divided into seven major sections with several appendices. Following this preface, the first section introduces the field of image analysis and its relationship to the discipline of image processing. Section two defines image processing terms used throughout the document. Sections three, four, and five address, in turn, each of the image analysis techniques implemented in the package: image segmentation, object recognition, and quantitative analysis. Finally, section six describes the capabilities and features of the application package. A bibliography of references and related subjects is provided. Appendix A provides supplemental formulas for the discussions in section three. Results from the various image segmentation techniques are located in Appendix B, and Appendix C provides examples of the object recognition algorithm implemented and presents typical quantitative measurements used to describe images.

1.0 Introduction

Image processing is a discipline that encompasses a class of techniques that are used to enhance and analyze digital images. Image processing systems typically consist of three components as shown in Figure 1-1. Image acquisition devices convert a continuous scene into a numerical representation, an image, of that scene so that it may be processed by a digital computer. Image processing provides the capability to enhance the image for a desired effect or generate data about the image. Finally, the image display component provides a means to output the resultant image so that it may be viewed.

Image processing applications can be utilized to perform a variety of tasks for researchers. Many are used either to generate quantitative data about the objects within a given image (image analysis) or to produce improved digital images through application of successive enhancement techniques to that image (image enhancement). Image analysis techniques center on generating quantitative measurements that describe the physical characteristics of the objects in an image. These techniques require little or no user interaction. The most common difficulty encountered when utilizing these techniques is the degradation of image data, usually introduced during the image acquisition process. However, through utilization of image enhancement techniques, which are primarily a series of "image in - image out" processes, the visual appearance of the image data can be modified and improved. As examples, contrast enhancement, high-pass filtering, and color enhancement techniques all involve the generation of new images to accentuate specific features through manipulation of a pixel's intensity value. The application of these techniques is often through subjective evaluation of the image data and is highly interactive.

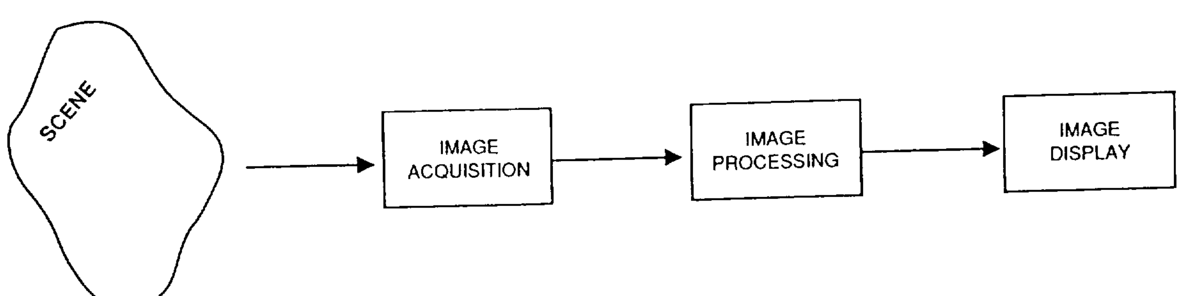


Figure 1-1. Components of an image processing system

In support of and to assist researchers in a wide range of disciplines, e.g., interferometry, heavy rain effects on aerodynamics, and structure recognition research, it is often desirable to count objects in an image and compute their geometric properties. The focus of this application has, therefore, concentrated on a subset of image analysis techniques used for object recognition in binary images. The three main phases are categorized as: image segmentation, object recognition, and quantitative analysis.

Images typically consist of several shades of grey, but extraction of geometric properties from a grey scale image is difficult. Grey scale images are often transformed, through image segmentation techniques, into a binary format, producing images with a uniform background and specifically defined object pixels. These binary images display silhouettes of objects so that the geometry properties of the objects may be extracted. After the binary images have been produced, object recognition techniques are employed to locate and distinguish a particular object from the background or other objects present. The concept of neighborhood connectivity determines how objects are located and influences the accuracy of their geometric properties. Sequential labeling algorithms, border fill algorithms and line-adjacency fill algorithms are examples of object recognition techniques currently utilized in this field. When the objects located in a given image have been identified, processing can be applied to each one on an individual basis. Some of the quantitative measurements that can be generated include the area, perimeter, Euler number, compactness, and the radii of an object. In conjunction with the process of calculating the relative position of the object and its orientation to its surroundings, moments of the image are also determined.

2.0 Image Processing Terminology

In this document, the *image, f,* will refer to a two-dimensional array where $\boldsymbol{M}$ and N denote the spatial dimensions of the image and the value of $\boldsymbol{f}(\boldsymbol{m},\boldsymbol{n})$ at any point $(\boldsymbol{m},\boldsymbol{n})$ is the brightness intensity function. The smallest unit that is addressable in an image is defined as a picture element or *pixel.* The digital value of each pixel represents its intensity value. Both spatial sampling and brightness quantization influence how a value is assigned to a pixel. Spatial sampling determines the intervals at which a scene is "sampled" to acquire a good approximation of that scene. The higher the sampling rate (resolution), the greater the detail. Some of the more common image sizes consist of 512 x 512, 1024 x 1024, or 2048 x 2048 resolutions.

Brightness quantization imposes a limit on the number of discrete grey levels that a pixel value may be assigned. As the number of bits associated with each pixel increases, a larger range of grey levels may be used. For example, if the pixel only contained one bit for intensity values then the image would be comprised of zeros and ones - a true binary image. It is more common, however, to associate 8 bits per pixel, thereby allowing each pixel to contain one of up to 256 different values at any one time.

Conventions that will be utilized throughout this document include:

1) The coordinate system used will have its origin oriented so that the upper left corner of the image is position **(0,0)**.

2) Addressing of elements in the two-dimensional array will consist of $\boldsymbol{m}$ and $\boldsymbol{n}$ locations; where $\boldsymbol{m}$ represents the scanline and $\boldsymbol{n}$ represents the column position. The total scanlines and columns permissible are represented as $(\boldsymbol{M}\text{-}1)$ and $(N\text{-}1)$, respectively. The largest grey level value is represented by $(\boldsymbol{L}\text{-}1)$, using 8-bits per pixel. Thus an image has $\boldsymbol{MN}$ resolution with $\boldsymbol{L}= 2^8$ grey levels, e.g., $\boldsymbol{L}$=256 where $\boldsymbol{l} = 0,1,...255$ gives the different intensity values.

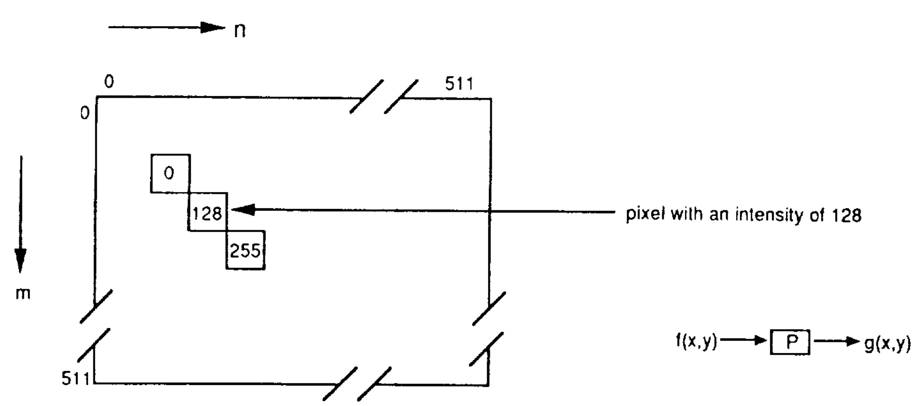
3) Low digital intensities in an image represent low intensities observed in the actual scene, with the intensity value of zero representing black and 255, white.

4) Images illustrated in the discussions concerning thresholding techniques are of M=N=8 resolution with $L=2^3$ of which eight different intensity values will be used, l=0,1...7. Images illustrated in the discussions of object recognition and object statistics are of M=N=16 resolution with $l = 0$ or $l = 255$. Images of M=N=512 resolution where $l = 0, 1,...255$, were used to test and verify many of the features discussed herein and are shown in Appendix B and C.

5) Input pixel values are referred to as $f(m,n)$ and their associated output pixel values, after applying an image processing technique, are referred to as $g(m,n)$.

6) In discussing threshold techniques, t, represents a threshold value; t' represents a possible candidate for the optimal threshold value utilized in an algorithm; and t^* represents the optimal threshold value selected by the algorithm on the basis of a criterion function.

Figure 2-1 illustrates some of these conventions.



a. Components of an image

b. Components of an image processing function

Figure 2-1. Specifying pixels in an image

3.0 Image Segmentation

In order to describe the quantitative measurements, i.e., shape features or geometric properties of objects in an image, it is often necessary to decompose the image into two specifically defined classes - foreground and background. Image analysis techniques which support this decomposition process include a wide variety of image segmentation schemes.

The purpose of segmentation is to divide the image into meaningful regions so that descriptions of the regions can be computed. Segmentation schemes are typically categorized as region-dependent or point-dependent techniques. The former technique utilizes the properties that are exhibited in a defined region, or *neighborhood*, while the latter technique examines the image on a pixel-by-pixel basis. Although there are several region-dependent approaches, this report concentrates on various point-dependent techniques. Additional information of image segmentation approaches can be found in Hall [3] or Gonzalez and Wintz [2]. A survey of image segmentation techniques is described in Haralick and Shapiro [4].

The most commonly used point-dependent approach for image segmentation is *thresholding*. To facilitate understanding of the thresholding methods, an image's grey level distribution, or histogram, is discussed. Histograms reveal the frequency distribution of the pixels within the image and are typically represented as a plot of the grey levels (the horizontal axis) and the quantity of pixels at each specific grey level (the vertical axis). Although image histograms do *not* depict any spatial information, i.e., several different images can have the same grey level distribution, they do convey, at a glance, the pixel distribution of the image. Histograms are generated by literally counting all the pixels in the image and incrementing the count value associated with each intensity value. By this method a count array, **c**[ ], of length equal to L would initially contain all zeros and as the image is "walked-through" left to right, top to bottom, the count array is adjusted accordingly. Once the total count of each intensity value is known, the relative frequency of the intensity values can be computed - that is, the measure of the probability that a pixel in the image would have a specific intensity value. Relative frequency is calculated utilizing the formula:

$$p[l] = \frac{c[l]}{MN}$$

where $l = 0,1..L\text{-}1$ and MN is the total number of pixels in the image.

Generally, when computing thresholds, it is often useful to determine, early on, the cumulative distribution of the image:

$$P[l] = \sum_{i=1}^{l} p[i]$$

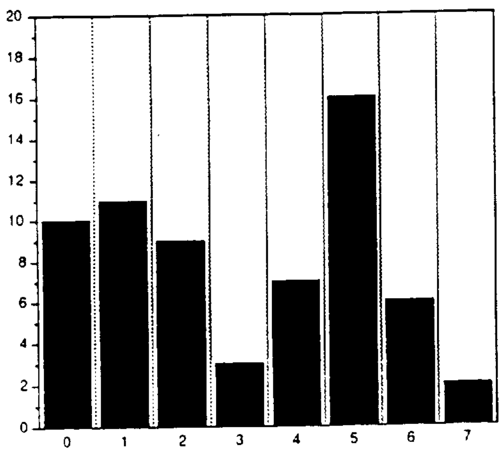
where $P[0] = p[0]$ and $l = 1,2...L\text{-}1$. This represents the percentage of pixels with intensity values at or below the specific intensity, l.

Additionally, the maximum and minimum intensity values represented in the image are located at the largest and smallest l values where $p[l] > 0$ and $l = 0,1...L\text{-}1$.

Other single image statistics that can be inferred from an image's histogram include: *mean*, the global brightness of the image; *standard deviation*, the amount of grey level variation about the mean; *mode*, the grey level where the associated relative frequency is largest; *median*, the smallest grey level where at least half the pixels in the image are defined; and *entropy*, i.e., the measure of the degree of randomness of the set of random variables [2]. Formulas utilizing the histogram and describing the mean, standard deviation, and entropy are included in Appendix A. To clarify, an example using an 8x8 resolution image with L=8 different grey levels ranging from 0 to 7 is illustrated in Figure 3-1, on the following page. Its associated histogram and statistics are also shown.

0	1	1	1	2	2	2	2
0	0	1	1	2	2	2	2
0	0	0	1	1	1	2	3
0	0	0	0	1	1	1	3
5	5	5	5	5	4	3	4
6	6	6	5	5	5	4	4
7	6	6	5	5	5	5	4
7	6	5	5	5	5	4	4

Image



Histogram

l =	0	1	2	3	4	5	6	7
c[l] =	10	11	9	3	7	16	6	2
p[l] =	0.16	0.17	0.14	0.05	0.11	0.25	0.09	0.03
P[l] =	0.16	0.33	0.47	0.52	0.63	0.88	0.97	1.00

mean = 3.0625
standard deviation = 2.1641
entropy = 2.7856
median = 3
mode = 5

Statistics

Figure 3-1. Grey level histogram of an image

3.1 Image Segmentation - Thresholding

Thresholding is an effective technique for image segmentation. If an image exhibits a histogram that consists of two peaks (a bimodal histogram) and the height associated with each peak is proportioned to the area of the corresponding region in the image, then the selection of the threshold (*t*) is simply chosen to separate the two peaks. This allows segmenting of the image into two distinct classes - foreground and background [3]. Using Figure 3-1, a threshold equal to 3 could be selected as the grey level separating the two peaks. To produce a binary image using this *t* value, the criterion function

$$g(m,n) = \begin{cases} 0 & \text{if } f(m,n) < t \\ 1 & \text{if } f(m,n) \geq t \end{cases} \qquad \text{(eq. 3.1)}$$

is applied to every pixel in *f*. The objective of selecting a *t* value is for one class to contain, as much as possible, all the pixel values related to the background, while the other contains pixel values associated with the foreground - the objects of interest. In practice, it is often difficult to ensure that an image exhibits a bimodal histogram. Other methods for determining an optimum threshold value generally depend on a criterion function being used to define the property of the classes into which the image will be divided.

Threshold techniques may be categorized as indicated in Figure 3-2. Multiple level selection techniques, for the most part, are an extension of the bi-level techniques and are beyond the scope of this document. Bi-level thresholding can be further broken down into global or local thresholding techniques. Global techniques apply the optimally selected threshold to an entire image, whereas local techniques divide the image into subimages and determine an *optimal threshold* for each of the subimages. Using the local techniques, the processed image (i.e., the thresholded image) evidences grey level discontinuities at the boundaries of these different subimages. A smoothing technique would then be applied to the processed image to eliminate these discontinuities.

Whether global or local approaches are utilized, the thresholding techniques applied to the entire image or subimage are further classified as point-dependent or region-dependent methods. Point-dependent methods determine a threshold value solely from the grey level of each pixel. Region dependent methods determine ***t*** by utilizing some local property in the neighborhood of each pixel [12]. For the most part, region-dependent methods rely on second order grey level statistics e.g., a grey level co-occurrence matrix or a transformation of the grey level histogram.

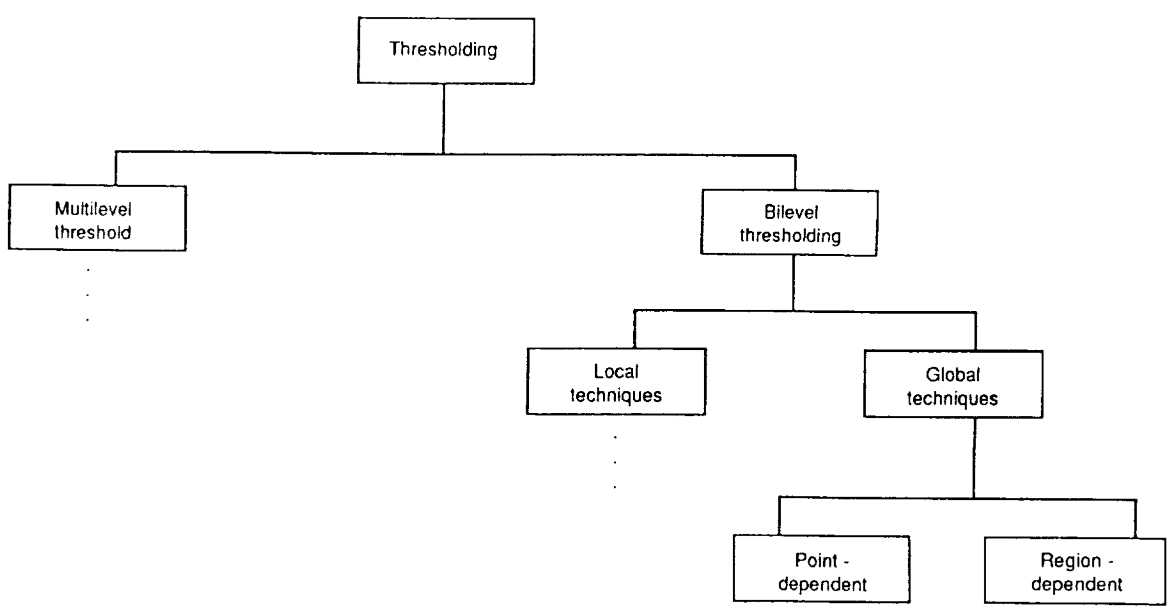


Figure 3-2. Thresholding categories

In the interest of providing researchers with the broadest possible capabilities to process their data, the global, point-dependent thresholding techniques rather than the region-dependent were examined. As stated previously, these techniques can be extended to local thresholding methods by applying the same point-dependent method to each of the subimages. For example, global techniques do not perform satisfactorily if the image's histogram is unimodal. The image, therefore, can be subdivided and then global thresholding techniques applied to each of the subimages.

3.2 Thresholding Using Global Techniques

The proper selection of the threshold is very important if the image is to have its objects correctly recognized. Several techniques utilize the information contained in the image's histogram, allowing a two-dimensional problem to be reduced to a one-dimensional problem by treating the images as patterns of brightness. Thresholding methods may be applied to an image on a subjective basis, such as allowing the researcher to manually enter t (hopefully after evaluating the image's histogram) or through automatic selection. Automatic selection consists of computing an image's histogram and determining the most suitable threshold value, t^*, based on an appropriate criterion function.

Investigations conducted for this application were directed at four specific automatic thresholding methods: Otsu's, Entropy Analysis, Moment Preservation, and Minimum Error thresholding methods. These methods were selected based on their overall applicability to the research being conducted in the Image Processing Laboratory and their potential for expansion into multilevel thresholding techniques.

3.2.1 Otsu's Thresholding Method

Several thresholding methods base their selection of t^* on computing a possible candidate, t' for every possible grey level. Often these techniques select a threshold that results in a function that has been minimized or maximized. For example, Otsu's method [11] utilizes discriminant analysis to select an optimum threshold value. Using the image's grey level distribution, the criterion function selects the threshold, t^*, such that the between class variance value is maximized.

This results in maximizing the measure of separability between the two classes obtained by segmenting the image at point t'. The optimum threshold value is selected by maximizing:

$$\sigma^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \qquad \text{(eq. 3.2)}$$

where:

- ω_0 is the area occupied by the background of the image after it has been thresholded;
- ω_1 is the area occupied by the foreground (objects) of the image after it has been thresholded;
- μ_0 is the background's average brightness; and
- μ_1 is the foreground's average brightness.

Stated in terms as discussed in Section 3.0, ω_0 is the summation of all the pixel value's relative frequencies up to and including the currently selected threshold t' that constitutes the background; ω_1 is similarly defined, but for those pixel values that constitute the foreground. In essence, every grey level, $t'=0,1,..,L-1$, is chosen and the one satisfying equation 3.2 is selected as the optimum threshold value, t^*. After the optimum threshold is selected, a binary image can be produced using equation 3.1. Primary formulas that support this method are located in Appendix A. Using the Figure in 3-1, the criterion function for every candidate threshold is illustrated in Figure 3-3a.

3.2.2 Entropy Analysis Thresholding Method

A criterion function based on the entropic features of information theory, as applied to image processing, was suggested by Kapur, et al. [8]. As with Otsu (3.2.1), this method also utilizes the global properties of image histograms. Using the image's grey level distribution, the optimum threshold is selected by obtaining the maximum information between two class distributions:

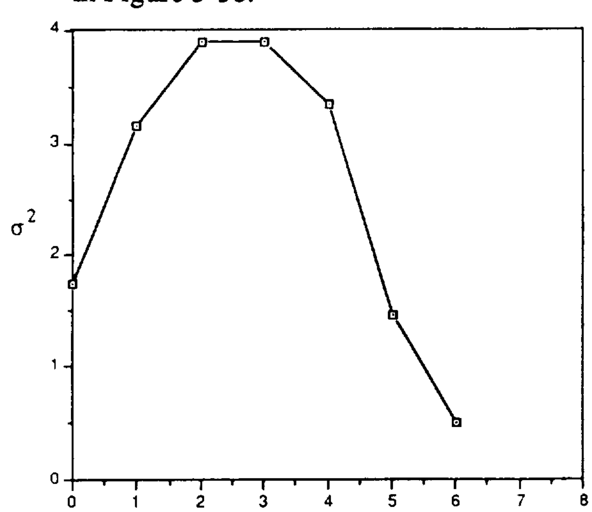
t^*=max(H(A)+H(B)) for every t' (eq. 3.3)

$$\text{where } H(A) = -\sum_{i=1}^{t'} \frac{p[i]}{P[t']} ln\left(\frac{p[i]}{P[t']}\right)$$

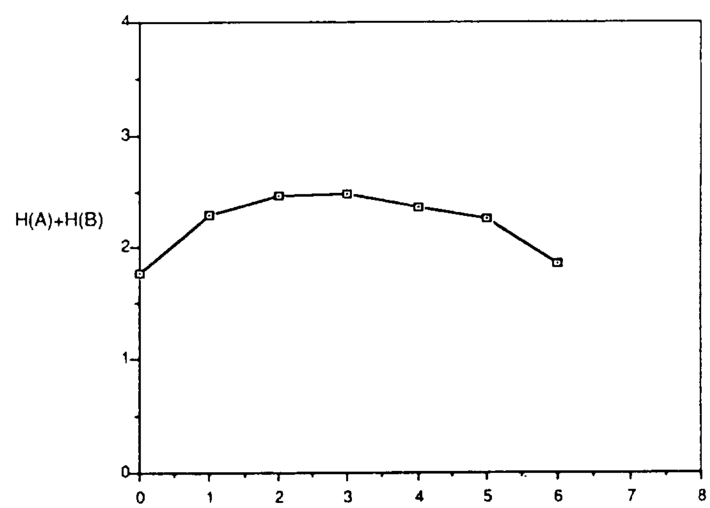
$$\text{and } H(B) = -\sum_{i=t'+1}^{L} \left(\frac{p[i]}{1-P[t']}\right) ln\left(\frac{p[i]}{1-P[t']}\right).$$

$P[t']$ represent the cumulative frequency and $p[i]$ is the relative frequency as described in Section 3.0. All grey levels, t', are selected as candidates for the optimum threshold value. For each grey level selected, the image is split into two classes and the entropy of each class is determined. The optimum threshold value is selected where the entropy between the classes is maximized.

Again, using Figure 3-1, the criterion function for every candidate threshold is illustrated in Figure 3-3b.



a. Criterion Function for Otsu's Method, t^*=3, max (σ^2)



b. Criterion Function for Entropic Analysis Method t^*=3, max (H(A)+H(B))

Figure 3-3. Criterion functions

3.2.3 Moment Preservation Thresholding Method

Another automatic threshold selection technique that utilizes a criterion function is based on deterministic approaches to moment-preserving principles [13]. Essentially, the grey level moments of an input image are computed from its histogram with the optimum t^* being selected in such a way that the moments of the output image, the thresholded image, are not modified. Moments may be generated from an image's histogram through:

$$m_i = \sum_{l=0}^{L-1} p[l]\, l^i \qquad \text{(eq. 3.4)}$$

where $p[l]$ is the relative frequency of a grey level and $(L\text{-}1)$ is the largest grey level value.

This method selects t^* using

$$p_0 = \frac{1}{MN} \sum_{l=0}^{1} c[l] \qquad \text{(eq. 3.5)}$$

where $c[l]$ is the number of pixels at the grey level l. For all candidates, t', the optimum threshold value selected is the highest grey level which maps at least $(100\text{-}p_0)\%$ of the pixels into the p_0 class. p_0 is the class that consists of grey levels below the selected candidate. Appendix A contains additional formulas for this approach.

3.2.4 Minimum Error Thresholding Method

In the last automatic threshold selection technique, a criterion function was employed based on statistical decision theory. Again the image's histogram is used and an optimum threshold is selected such that the average classification error rate is minimized. For each candidate threshold, the criterion function reflects the amount of overlap between the background and foreground classes. The optimum threshold is selected as the grey level that minimizes this overlapped area. The criterion function is:

$$t^* = \text{minimize}\ [1 + 2\ [P_1(t')\log\sigma_1(t') + P_2(t')\log\sigma_2(t')] - 2\ [P_1(t')\log P_1(t') + P_2(t')\log P_2(t')]]$$

where $P_1[t']$ and $P_2[t']$ are the cumulative frequency of each class, and σ_1 and σ_2 is the variance associated with each class, respectively. See Appendix A for supporting formulas.

3.3 Comparison of the Threshold Methods

In comparing the automatic threshold techniques discussed, the success of the technique applied to an image is dependent on the objects that are to be extracted. For the best results, it was evident that the grey level composition of the objects should occupy a range of grey levels that are distinct from the background. There appeared to be a trade-off between maximizing or minimizing the criteria functions and maintaining a reasonable ratio of black to white pixels. For example, if the optimum threshold value, t^*, is too high then information may be lost. Similarly, if t^* is too low, an increase in background clutter is observed.

Otsu's method and the moment-preserving method are best suited for images that reveal a bimodal histogram. However, the optimum threshold value computed using Otsu's method deteriorates as the two modes become further apart - as observed in broad, flat valleys. The minimum error method is more appropriate in images that exhibit unequal variance between the two classes or have very unequal class sizes. The minimum error method along with the entropic method should be used if the intent is to preserve the detail in the image. The entropic method is more suitable to use on an image that is multi-modal in nature. Finally, the entropic, moment-preservation, and Otsu methods should be applied to an image to retain the uniformity and shape of the image. Appendix B illustrates these methods on typical research data.

4.0 Object Recognition

After thresholding techniques have been applied to an image, several *object recognition* schemes may be utilized to determine the existence of objects in an image. In this document, the term object recognition implies object detection. Conceptually, this means as an algorithm scans through an image, objects are detected (recognized) as being different from their surroundings. In other words, objects are *not* recognized as specific identifiable items, but rather as an anomaly to the background. Once an object has been detected and its boundaries defined, classification techniques can be utilized to identify the object as a specific entity.

In order to detect objects, we must understand what defines an object, its boundaries, and its neighbors. If these terms are not well defined, the accuracy of the geometric properties associated with each object will be impaired.

In a binary image, an object is represented by pixels whose values are not the background value and are contained by some sort of boundary. Conceptually, for binary images, there are two types of pixels that comprise an object. An interior pixel is surrounded such that all its immediate *neighbors* have the same grey level value. Whereas, a pixel belonging to an object that has at least one of its immediate neighbors as part of the background is defined to be a perimeter pixel. Together, the perimeter pixels of an object comprise its boundary. This distinction is needed for the determination of several geometric properties. In distinguishing between perimeter and interior pixels, the term *neighborhood* is used to clarify how these pixels might be connected. Although different authorities cite differing definitions of neighborhood connectedness, such as Horn [5] who advocates a six connectedness neighborhood, this report will address the problems associated with 8-connected and 4-connected neighbors. Figure 4-1 illustrates these types of connectedness.

If connectivity of the interior pixels is defined to be an 8-connected neighborhood, then the perimeter pixels of the object must be connected using 4-connectedness. Similarly, if the interior pixels of the object are 4-connected, then the boundary pixels must use 8-connectedness. This limitation is necessary so that interior pixels are not connected to pixels outside the perimeter that defines the object. Consider Figure 4-2, the dilemma of connectedness arises when trying to classify pixel A as an interior pixel (using 4-connectedness for the object) or a perimeter pixel (using 8-connectedness for the object). For purposes of consistency, the 4-connectedness neighborhood definition is the basis for discussion herein relating to object detection and quantitative measurement schemes.

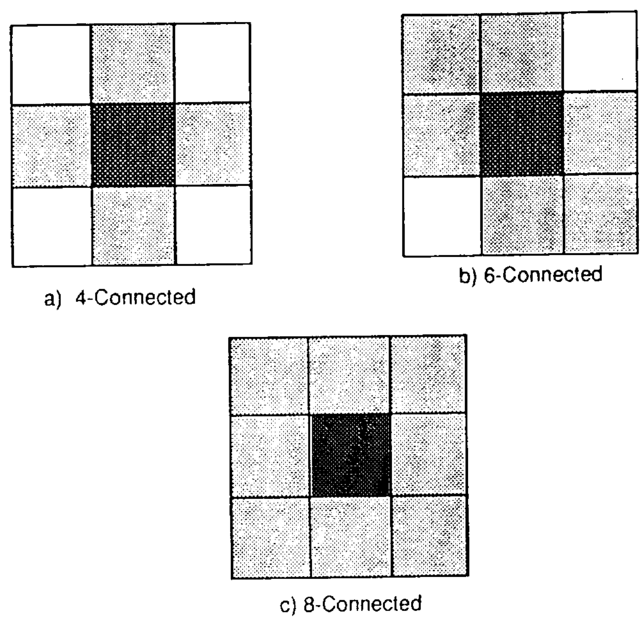


Figure 4-1. Connectedness

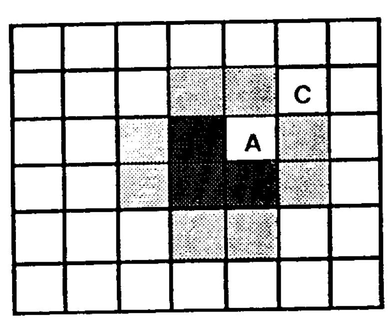


Figure 4-2. Connectivity dilemma

4.1 Object Detection Techniques

Several techniques exist to detect objects in an image. Some techniques scan an image from left to right assigning labels to each subject or assigning an intensity value that differs from the foreground and background intensities. Other techniques assign labels to the objects by following the connected border pixels of the objects. The various techniques that were examined include: a pixel-by-pixel fill algorithm, a line-adjacency fill algorithm, and a sequential labeling algorithm. The intent of all these algorithms is to "fill regions" defined as objects with a specific fill value or a label so that object statistics can be computed. Usually, compilation of certain statistical information may be accomplished as the object is being filled. This reduces what would be a two pass process over the image into one pass. In describing each of the algorithms, it is assumed that the scanning process (the "walk through" process) is from left to right and top to bottom.

4.1.1 Single Pixel Algorithm

The Single Pixel Algorithm is a conceptually simple, but inefficient recursive algorithm that may be employed to detect objects in a binary image [15]. Three intensity values are used in this method, a background value (0 in the binary image), a foreground value (1 in the binary image) and a fill value. As the algorithm "walks through" the image, each pixel is examined to determine if it needs to be filled. If the pixel value is neither the background value nor the fill value then it is *recognized* as an object or part of an object. The pixel value is changed to the fill value and the process is repeated with the "next" pixel. The next pixel examined is actually determined by the connectivity neighborhood scheme utilized. This method uses at least four recursive calls, one for each immediate neighbor.

4.1.2 Line-Adjacency Fill Algorithm

An expansion of the single pixel approach is to view a group of pixels as a line segment. This is the premise behind the Line-Adjacency Fill Algorithm (LAF) [15]. An object is then depicted as a group of line segments connected vertically. Although this method is also recursive, its determinant for repeating a recursive call is based on the connectivity of the line segments that are vertically connected. Similar to the pixel-by-pixel object fill method, this algorithm uses the same three intensity values to distinguish between background, foreground, and a filled object. When a new object is encountered, the initial call to the LAF routine identifies the "anchor" pixel as the extreme left pixel of the "highest" scanline in the object. The algorithm then scans left and right to locate the endpoints of subsequent line segments. Note that on the initial call in detecting a new object, the *scan left routine* is not necessary - that is, it is presumed that the current object pixel has a background pixel as its immediate left neighbor. This routine still needs to be called, however, so that some geometric property variables are correctly tabulated. After each line segment is identified, the line segment is filled. As the algorithm continues, the next iteration of the LAF routine will locate the next group of horizontally connected pixels that are vertically adjacent to the line segment just filled. The algorithm is recursively called in the downward search direction. When the "lowest" line comprising the object is filled, the recursion process terminates. As each of the levels in the recursive routine return, a search in the upward direction ensues, again filling horizontal line segments that are vertically connected, thus ensuring that the line segments around any hole contained in an object are filled correctly. Figure 4-3 demonstrates the pattern direction of the algorithm.

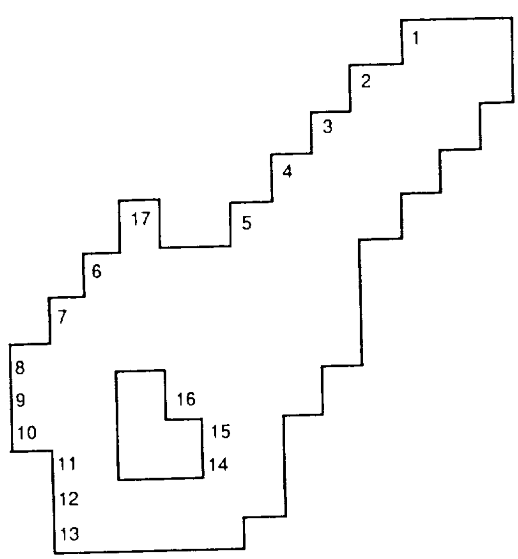


Figure 4-3. Pattern of the Line-adjacency fill algorithm

4.1.3 Sequential Labeling Algorithm

The third method for object identification is the Sequential Labeling Algorithm (SLA) where each pixel is assigned a label as it is encountered in the scanning process [15]. During the scanning process, if the current pixel value is identified as an object, the connectivity of its neighboring pixels is examine. A label is assigned to the pixel on the basis of the other object(s) to which it is connected. Using 4-connectedness, if the current pixel has an object neighbor immediately above it, that object label is used as the current pixel's label. If there is no such neighbor, then the current pixel's immediate left neighbor is examined. If this neighbor contains an object label, the current pixel's associated label is the same as that of its left neighbor. A problem arises if both immediate neighbors (above and left) have different object labels, as shown in Figure 4-4. Initially, there appears to be two separate objects, although further examination of neighbors shows the two objects are connected through the current pixel. If this scenario exists, then the current object's label is assigned from *one* of its neighbor's object labels and the two labels are said to be equivalent. A new object label is issued whenever a transition from the background to an object which appears to be isolated is detected.

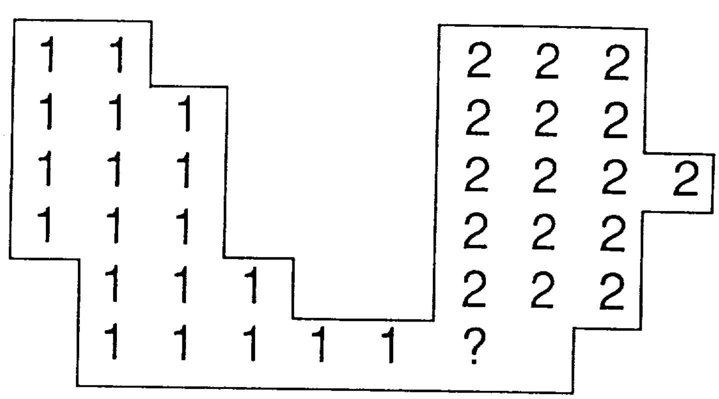


Figure 4-4. Sequential labeling algorithm conflict

4.2 Comparison of the Object Recognition Algorithms

The algorithms examined were compared on the basis of the complexity of the algorithms, speed of the compiled code, memory limitation of the devices on which the code would be implemented, data structures and memory requirements utilized for algorithmic implementation, and the suitability for realistic implementation in a research environment. Test cases, as well as typical image data, i.e., the number of pixels comprising each object (referred to as object pixels) and the number of objects in an image, were considered in evaluating the methods and determining which method was most appropriate.

The single pixel algorithm, although conceptually simple, is highly recursive and inefficient. The depth of the recursive calls can become exceedingly large when considering the resolution of typical images and the size of objects in various images. The algorithm is also inefficient in that every pixel is examined more than once, sometimes as many as four separate times (based on 4-connectedness).

The LAF algorithm is more complex than the single pixel method, but still recursive in nature. The algorithm executes faster because groups of pixels are examined before a recursive call is issued - thus reducing the runtime memory requirements, i.e, the activation records needed to support recursion. Using this method, with the exception of the end points of the line segment and interior pixels adjacent to holes that are contained in the object, pixels belonging to a line segment are never re-examined if they were already inspected in a previous invocation of the LAF algorithm.

The SLA algorithm differs from the preceding algorithms in that it does not require recursion to detect an object. Therefore, the memory requirements that are usually needed to support recursion are not applicable to this method. This algorithm does, however, have other limitations to consider. Theoretically, the total number of possible labels needed to "label" an image is $\left(\frac{MN}{2}\right)$; an image exhibiting a checkerboard appearance. Typical images in the IPL consist of 8-bit pixel data. The labeling "pool", therefore would normally be 256 distinct labels. Of these, two are already used because the images are binary. This situation limits the researcher to being able to process only data that was comprised of 254 or fewer objects. The "pool" of labels could be increased if the image data was extended to 16-bit or 32-bit pixel quantities, but the memory constraints inherent with the workstation require consideration. If a scenario as described in Figure 4-4 exists, the runtime execution is impinged upon because of the merging process that must transpire. There is also the potential that pixels will be classified as perimeter pixels when they are actually interior pixels. These false perimeter points will decrease the accuracy of the computed geometric properties of the objects.

If there are a relatively low number of object pixels per image, the efficiency of the LAF and SLA algorithms is similar. As the number of object pixels increase, the runtime of the LAF is less efficient than the SLA method. As the number of objects per image increases, however, the cost of merging object labels deteriorates this slight advantage of the SLA method. Based on the various properties and the performance of these algorithms, it was determined that the LAF would be the most appropriate algorithm to implement in the image analysis application package.

5.0 Quantitative Analysis

Once an object in an image has been identified, several geometric properties can be computed. As stated earlier, the shape of the object is defined by the area it occupies, which in turn is itself defined by the connectivity that is used. These properties are useful for identifying the global characteristics of an object and for further shape analysis research.

The *area* of an object is one of the simplest measurements to obtain. It is calculated by counting all the pixels that belong to the object. Similarly, the *perimeter* is computed through counting the "perimeter" pixels - as defined in Section 4.0, object pixels adjacent to background pixels. There are several ways in which the perimeter count may be computed. For example, assume that we are computing the perimeter of an isolated pixel, i.e, an object containing exactly one pixel. One way to compute the perimeter is to define its value on the basis of what constitutes a pixel - here a pixel is the smallest entity definable in an image. Alternatively, a pixel may be thought of as containing four perimeter edges. In our example, the isolated object could then have a perimeter value of one or four. Finally, if the perimeter value is a measurement of distance around an object and *not* just the pixel count and an object contains more than one pixel, it is possible that some of the perimeter pixels will be diagonal to one another. In this instance, counting the pixel values as in the first method would only generate an estimated result. James [7] recommends counting two sets of perimeter values for an object, using 4-connected and 8-connected neighborhoods. The square root of this product is then used as an approximation of the true perimeter value. To eliminate the ambiguity surrounding the approximation, the perimeter is calculated using the first method.

After the area and the perimeter of an object are determined the *roundness* or compactness can be computed and is defined as the ratio:

$$\frac{(\text{perimeter})^2}{4\pi(\text{area})}$$

Note that a circle is valued with a compactness of 1 since $\frac{(2\pi r)^2}{4\pi(\pi r^2)} = 1$

The number of connected partitions as well as holes that exist in an object may also be calculated from simple counting methods. The difference between these numbers is defined as the *Euler number* which may be used to further classify an object. Using a 4-connected neighborhood to locate objects, the number of connected partitions and the number of holes in an object can be identified by using the following patterns, respectively:

$$\begin{array}{|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & 0 \\ \hline 0 & 1 \\ \hline \end{array} \qquad \qquad \begin{array}{|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & 1 \\ \hline 1 & 1 \\ \hline \end{array}$$

The object in Figure 4-3 has an Euler number of 0 - that is, one connected partition minus one hole. The object in Figure 4-4 has an Euler number of 1; the object contains no holes.

Many geometric properties can be represented as measurements of moments. In shape analysis, moments assist in defining the direction and orientation of an object, and its displacement within the image. By determining the first and second moments of an object we can compute the center of area, the orientation and eccentricity of an object and its bounding rectangle. The center of area of an object is located at position $(\bar{m}, \bar{n})$ such that:

$$\bar{m} = \frac{1}{area} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \bullet m \qquad \bar{n} = \frac{1}{area} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \bullet n$$

where $f(m,n)$ is zero, if background and 1, if an object.

Therefore, the center $\bar{m}$ position is the sum of all the coordinates in the object divided by the total area of the object. The center's $\bar{n}$ position is similarly defined. The moments of an object may be affected by the shift or change in scale of the image; therefore, moments are made shift invariant by defining the ijth moment, ***Mij***, as

$$M_{ij} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(m-\bar{m})^i(n-\bar{n})^j f(m,n)$$

The orientation, θ, of an object is the angle of axis of least moment of inertia and is given by:

$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2M_{11}}{M_{20}-M_{02}}\right]$$

where M_{11} is the XY moment about the center, $(\bar{m}, \bar{n})$, M_{02} is the second moment about the horizontal line through the center and M_{20} is the second moment about the vertical line through the center of the object. Figure 5-1 illustrates the various moments.

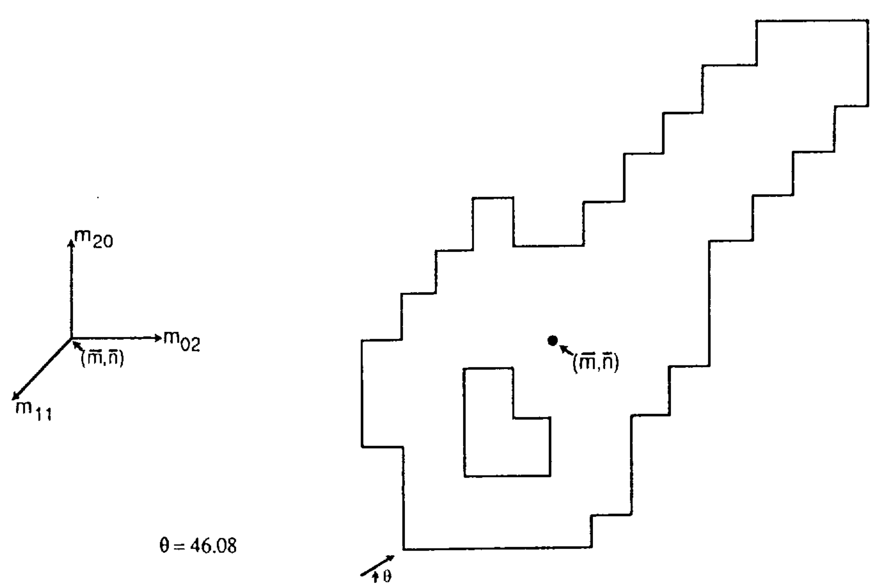


Figure 5-1. Moments and orientation

There are several methods to measure the eccentricity of an object [1] [6]. The eccentricity of an object is a measurement that determines the elongation of the object. One method to compute eccentricity is given by:

$$\frac{(M_{20} - M_{02})^2 + 4M_{11}}{area}$$

where M_{20}, M_{02}, and M_{11} are as defined previously.

Another measure for eccentricity is the ratio of the minimum and maximum distances from a perimeter pixel to the center of the object. For each perimeter pixel, the distance from it to the center of area is calculated. From all these distances the maximum and minimum distance is determined and their ratio in turn determines an eccentricity measurement. Figure 5-2 illustrates this method.

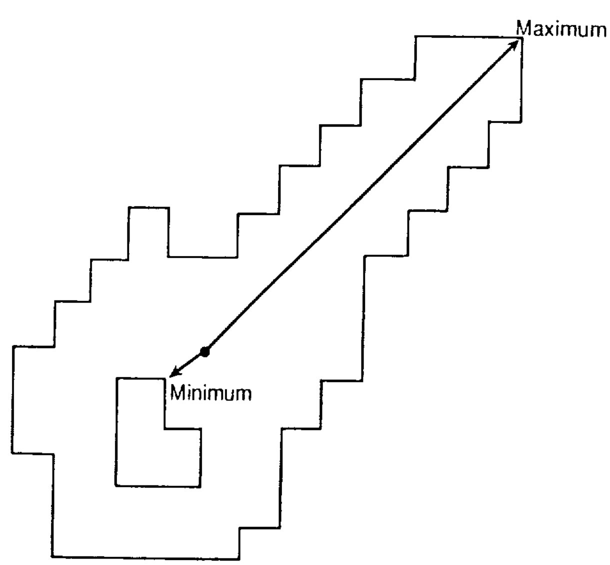


Figure 5-2. Minimum and maximum radii using 8-connectedness

6.0 Package Implementation

The Image Processing Laboratory at the NASA Langley Research Center exists to provide researchers with the interactive capability to produce, enhance, process, and analyze digital images. The lab is an open shop, research facility supported by personnel who demonstrate system capabilities, provide consultation support, and develop applications to increase the capabilities of the lab or to support research projects. Currently, the lab consists of a network of Sun Microsystem Family 3 color workstations which interface with several peripheral devices. Although the workstations support many programming languages and windowing environments, the language primarily utilized in the lab, and, hence, for this package was C. The package was developed in the SunView window environment.

The primary purpose of this package was to provide image analysis capabilities which would permit the computation of quantitative measurements derived from the analysis of a given image. The package development was initially in support of research in the areas of structural recognition and heavy rain effects on aerodynamics. The work performed in the development of this package also provided the basis for another image processing application package supporting research in the field of interferometry. Due to the open shop nature of the lab, however, a further objective in implementing this package was to provide a means for researchers to operate and manipulate their data with minimal support from the lab's personnel. To achieve this goal, a user-friendly graphical user interface was designed and implemented.

In implementing the various image segmentation and object recognition techniques, trade-offs involving accuracy, speed, and memory limitations were evaluated. For example, the accuracy of the object recognition techniques were influenced by the connectivity utilized to distinguish an object. The efficiency of the various thresholding techniques was improved upon through the employment of the minimum and maximum grey levels in an image as indices for the algorithm, rather than forcing the algorithm to search the entire histogram array structure. As stated earlier (Section 4.1), the object recognition and object analysis phases could and, in this case, have been combined. Conceptually, the two techniques each require a pass through the image, one pass to identify the objects (either through labeling or filling techniques) and another pass to collect statistical information about each object in the image. These two techniques were combined to improve the efficiency of the code. The memory limitations of the workstation impinged upon the implementation of the object recognition technique and also had a negative impact on the use of the SunView data structures. The recursive nature of the LAF algorithm required that the length of each activation record associated with a given call be minimized. This was accomplished through the definition and employment of global variables, limiting the parameters passed into each of the invocations.

A representation of the image analysis package is shown in Figure 6-1. As illustrated, the package utilizes the SunView window environment. It generates a large base window that contains a panel region and two canvas regions. The panel region allows the researcher to select items through a mouse. Each item represents an option to be modified or a function to be performed. The items are selected through button and slider manipulation or through text entry. The panel region also displays diagnostic and informative messages to assist the researcher. The two canvas regions are capable of simultaneously displaying an image through a 512 x 512 resolution viewport. Scroll bars are attached to each of the canvas viewports, allowing for the viewing of 512 x 512 resolution subsections of larger images.

Other options incorporated into this package provide basic I/O capabilities, color table manipulations and help features. The capability to display an image stored on a disk file in one of the canvas regions or store a processed image from the canvas region to disk is supported through the "Load Image >>" and "Save Image >>" options. Both of these options concatenate the names entered on the "Directory:" and " File:" lines together to form the complete pathname of the file to be manipulated. Both options also utilize the "walking menu" mechanism of the SunView environment to allow the user to specify from or to which canvas region image data will be stored or displayed. If the intent is to write an image to a disk file and that file already exists, the researcher is prompted for verification of an overwrite. To allow images processed in this package to be utilized by other packages, image files are stored on hard disk in the standard Sun rasterfile format. The "List Directory" option provides the capability to display a directory from within the package. The directory displayed is the name entered on the "Directory:" line of the panel region. The directory's contents are displayed in a pop-up Sun window and can be scrolled. The "Histogram" option generates a graphical display of the histogram for the image located in either the left or right canvas regions. The histogram, along with other single image statistics (see

Section 3.0) are displayed in a pop-up Sun window. The "ViewColor" option allows several different color palettes to be applied to the images that are displayed in the canvas regions. The actual pixel data of each image is not modified, however, red, green, and blue intensity values associated with each pixel are changed. This allows the researcher to view his data through color table enhancement methods. When this option is "exited" the original color palettes associated with each canvas are reinstated. Additionally, specific color palettes may be applied to or stored from the canvas regions through the "Load LUT" and "Save LUT" options. The "Help" option is another scrollable, pop-up Sun Window which contains a description of the overall functionality of the image analysis package, describing the purpose of each of the panel items and providing suggestions for applying the various thresholding techniques.

To obtain the geometric properties of objects it is first necessary to produce a binary image. This is accomplished through use of the "Threshold >>" option. The researcher may select any of the automatic selection methods as discussed in Section 3 or manually enter a specific threshold value. The image located in the left canvas is taken as input, processed using equation 3-1, and then displayed in the right canvas. To improve the visual display on the right canvas following thresholding, the pixel values used in Equation 3-1 are 0 and 255. The "Threshold >> User Supplied" option allows for the selection of one or two threshold values through slider bar or cursor manipulation. If two threshold values are entered, equation 3-1 is slightly modified, that is, pixel intensity values below the first threshold and above the second threshold are set to 0; otherwise the values are set to 255. The "Threshold >> Automatic" option implements the algorithms as defined in Section 3.2. By default, the Minimum Error, Entropic Analysis, and Otsu thresholding methods use the range between the minimum and maximum intensity values of the image's histogram. The researcher may elect to restrict this range by utilizing the clip feature associated with these options.

Prior to invoking the object detection option the researcher may elect *not* to have some of the geometric properties reported. This is provided by the "Object Statistics >> Select Stats" option and is illustrated in Figure 6-2. After the geometric properties have been selected, the object detection algorithm is invoked through the "Object Statistics >> Generate Stats" option. The technique implemented is the LAF algorithm as described in Section 4.1.2 and utilizes 4-, 6-, or 8-connectedness neighborhoods. As each object is detected, several partial sums for calculating the geometric properties of an image are compiled. By default, the geometric properties selected are displayed in a pop-up Sun window. Within this package, the researcher has the capability to save the statistics compiled in a permanent disk file or send the information to a printer.

The "Paint Pixels >>" option allows the researcher, through use of a bitmap editor, to edit the image located in the right canvas region. Pixels may be written in either black, intensity value 0, or white, intensity value 255.

Finally, the "Quit" option exits the application package. Figure 6-3 is included to illustrate a typical image in the left canvas and its thresholded binary image in the right canvas.

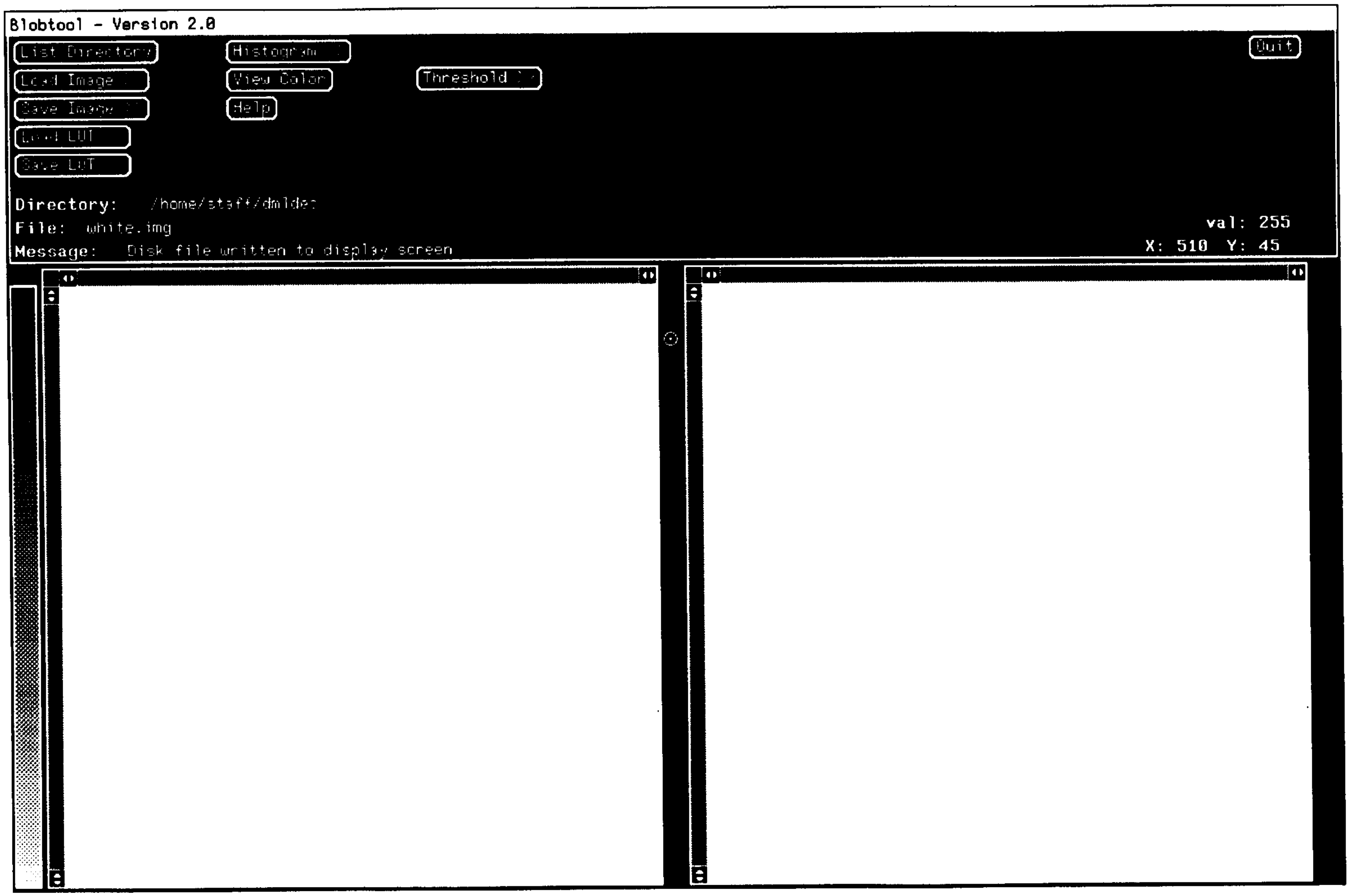


Figure 6 - 1. Blobtool

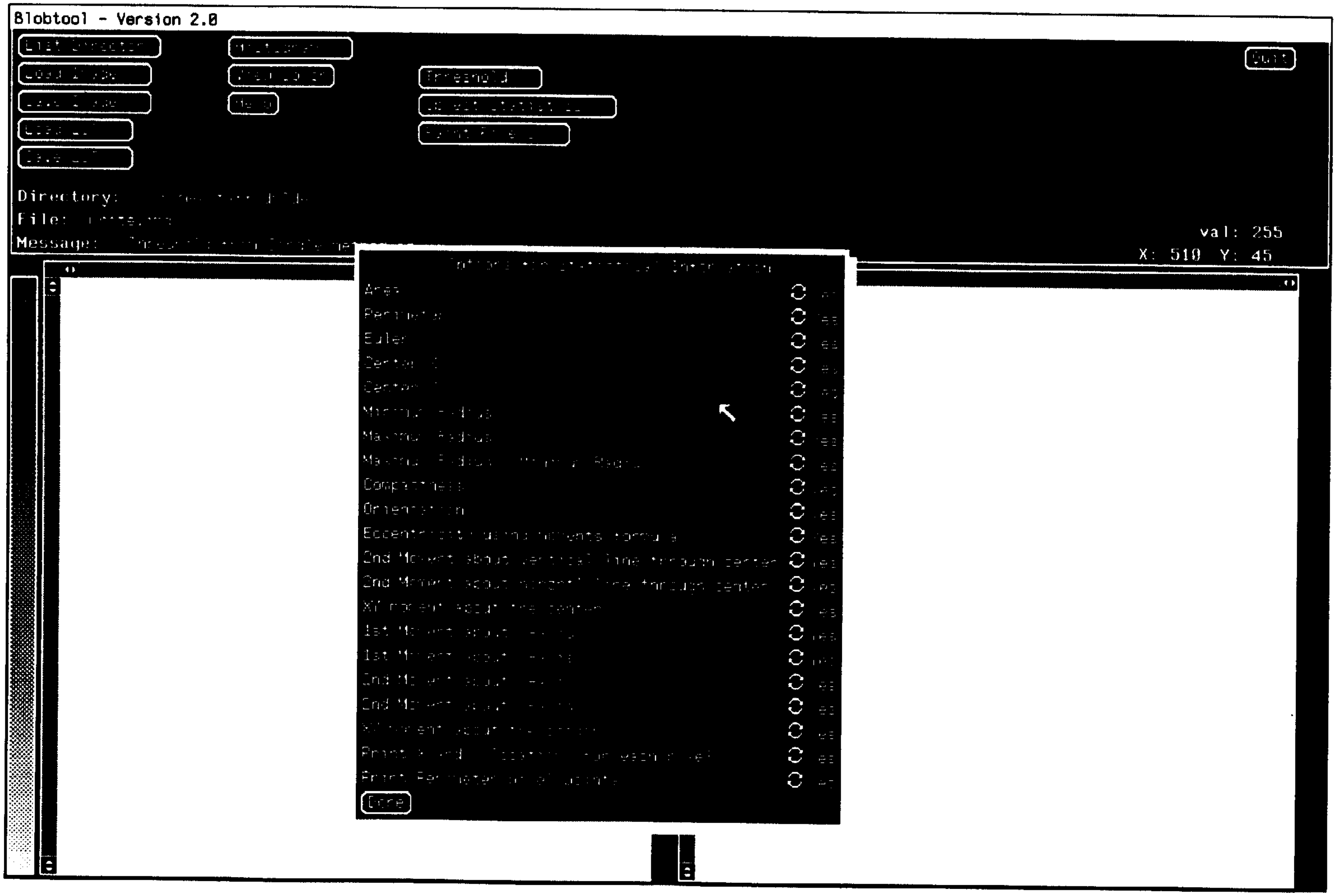


Figure 6 - 2. Blobtool: Statistic Information Selection

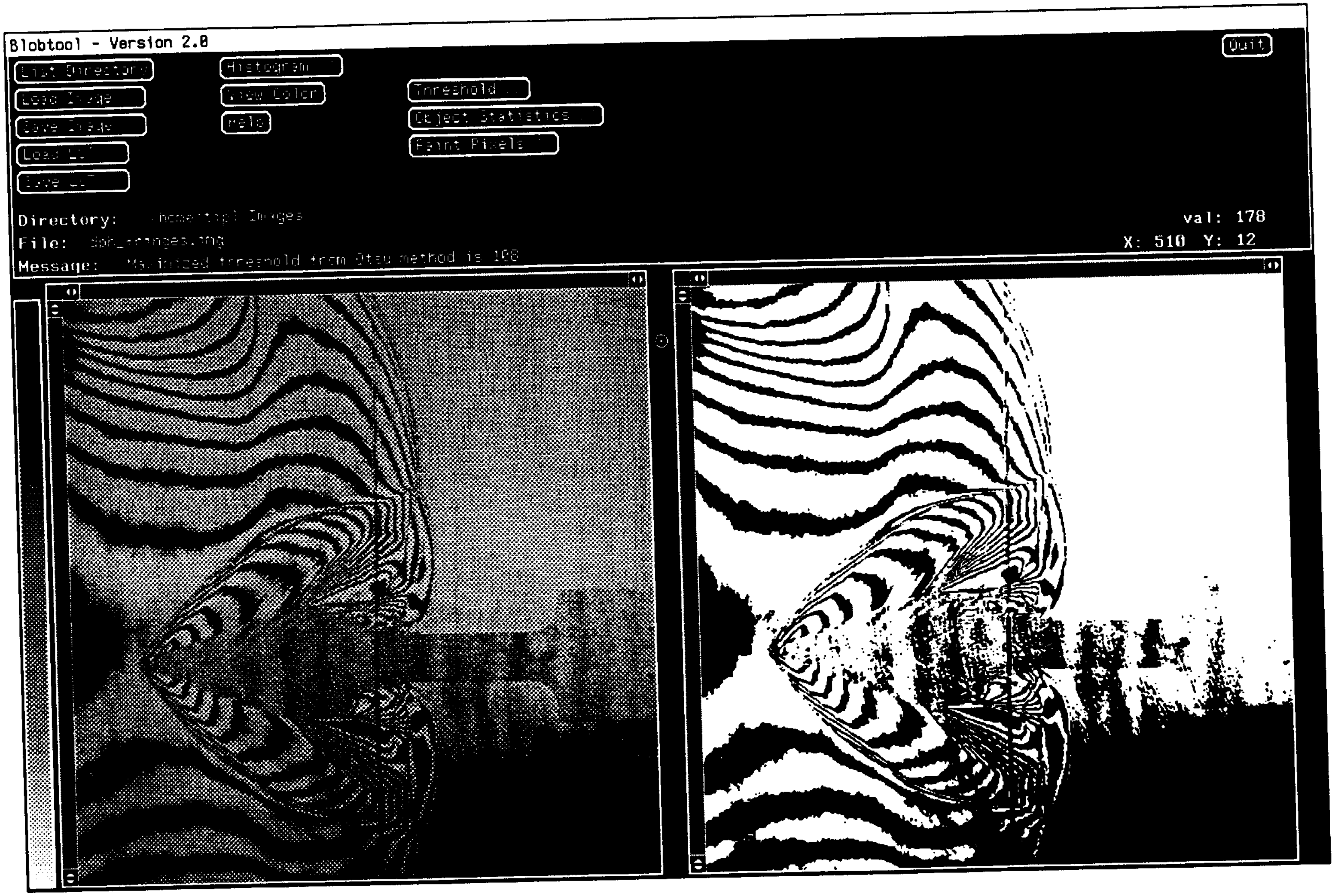


Figure 6 - 3. Blobtool: Binary Image

7.0 CONCLUSION

This application was to provide various tools and techniques that support the analysis of images. In order to achieve this goal, several image processing and computer vision techniques were examined, evaluated, and, if determined to be feasible, implemented. The package reduces researchers' dependence of staff consulting availability and provides access to various image processing tools, such as, thresholding, object recognition, and object analysis. These features are tied together in an easy to learn and understand, mouse-driven, graphically represented environment.

Acknowledgment

This work was performed for NASA/Langley Research under Contract NAS1-19038. The work also supported the requirements for a master's project in the Computer Science Department at Old Dominion University, Norfolk, Virginia and was, in part, directed by Dr. James Schwing (Associate Professor, Old Dominion University). The software developed by Charles Viles, programmer/analyst for Virginia Institute of Marine Science, Gloucester, Virginia provided the basis for the object recognition algorithm implemented and some of the quantitative measurements generated. Images utilized to test this application were provided by Earl Booth, Acoustics Division (ACOD)-Aeroacoustics Branch, Al Burner and William Goad, Instrument Research Division (IRD)-Photo-Optical Instrumentation Section, and Bryan Campbell and John Fabijanic, Applied Aerodynamics Division (AAD)-Subsonic Aerodynamics Branch. The Image Processing Lab's personnel provided detailed technical discussions and constructive comments during the package's development.

BIBLIOGRAPHY

1. Ballard, D.H. and Brown, C.M. Computer Vision. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 1982.

2. Gonzalez, R. C. and Wintz, P. Digital Image Processing. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts. 1977.

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

4. Haralick, R. and Shapiro, L. "Image Segmentation Techniques". Computer Vision, Graphics, and Image Processing. Vol. 29, 1985, 100-132.

5. Horn, B.K.P. Robot Vision. The MIT Press: McGraw-Hill Book Company, New York, New York. 1986.

6. Jain, A. K. Fundamentals of Digital Image Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 1989.

7. James, Mike. Pattern Recognition. Wiley-Interscience, John Wiley and Sons, New York, New York. 1988

8. Kapur, J., Sahoo, P.K., and Wong, A.K.C. "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram". Computer Vision, Graphics, and Image Processing. Vol. 29, No. 3., March 1985, 273-285.

9. Kittler, J. and Illingworth, J., "Minimum Error Thresholding". Pattern Recognition. Vol. 19, No. 1. 41-47, 1986.

10. Moik, J. Digital Processing of Remotely Sensed Images. U. S. Government Printing Office. Washington, D.C. 1980.

11. Otsu, N. "A Threshold Selection Method from Gray-Level Histogram". IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1. January 1979, 62-66.

12. Sahoo, P. K., Soltani, S., Wong, A.K. and Chen, Y.C. "A Survey of Thresholding Techniques". Computer Vision, Graphics, and Image Processing. Vol. 41. 1988, 233-260.

13. Tsai, W. "Moment-Preserving Thresholding: A New Approach". Computer Vision, Graphics and Image Processing. Vol. 29, No. 3. March 1985, 377-393.

14. Weszka, J. and Rosenfeld, A. "Threshold Evaluation Techniques". IEEE Transactions on Systems, Man and Cybernetics, Vol. 8, No. 8, August 1979, 622-629.

15. Wilton, Richard. Programmer's Guide to PC and PS/2 Video Systems. Microsoft Press, Redmond, Washington. 1987.

16. Winston, Patrick and Horn, B.K.P. , Lisp. Addison-Wesley Publishing Company, Reading Massachusetts. 1981.

APPENDIX A

APPENDIX A contains additional formulas.

MISCELLANEOUS PLANS

APPENDIX A

Single Image Statistics:

$$\text{mean} = \sum_{l=0}^{L-1} lp[l]$$

$$\text{standard deviation} = \sqrt{\sum_{l=0}^{L-1} (l - mean)^2 p[l]}$$

$$\text{entropy} = -\sum_{l=0}^{L-1} p[l]\, lg(p[l])$$

where p[l] is the relative frequency array and l=0,1...L-1. L is the number of grey levels in the image.

Otsu's Thresholding Method:

$$\sigma^2 = \omega_0\omega_1(\mu_1 - \mu_2)^2$$

$$\sigma^2(t') = \frac{[\mu T\omega(t') - \mu(t')]^2}{\omega(t')[1 - \omega(t')]}$$

where t' is the current candidate threshold value and the optimal threshold, t^*, is defined as

$$\sigma^2(t^*) = \text{maximized } (\sigma^2(t'))$$

$$\omega_0 = \omega(t') = \sum_{i=1}^{t'} p[i] \qquad \mu_0 = \frac{\sum_{i=1}^{t'} ip[i]}{\omega_0} \qquad \mu_1 = \frac{\sum_{i=t'+1}^{L} ip[i]}{\omega_1}$$

$$\omega_1 = [1 - \omega(t')] = \sum_{i=t'+1}^{L} p[i]$$

Moment Preserving Thresholding Method:

The first three moments of the output image, **g** are:

$$m'_i = \sum_{j=0}^{1} p[j]\,(z[k])^i \qquad \text{i=1,2,3}$$

where p[*j*] is the relative frequency of the class and z[*k*] is the number of pixels with the grey level *k*.

$$p_0 z_0^0 + p_1 z_1^0 = m_0$$

$$p_0 z_0^1 + p_1 z_1^1 = m_1$$

$$p_0 z_0^2 + p_1 z_1^2 = m_2$$

$$p_0 z_0^3 + p_1 z_1^3 = m_3$$

The linear equations are obtained through

$$c_0 m_0 + c_1 m_1 = -m_2$$

$$c_0 m_1 + c_1 m_2 = -m_3$$

Minimum Error Thresholding Method:

$$P_i(t') = \sum_{l=0}^{t'} p[l]$$ where i=0,1 and p[*l*] is the relative frequency of the grey level *l*.

$$\mu_i(t') = \frac{\sum_{l=0}^{t'} p[l]\,l}{P_i(t')}$$ where i=0,1 and each μ represents the mean of each class

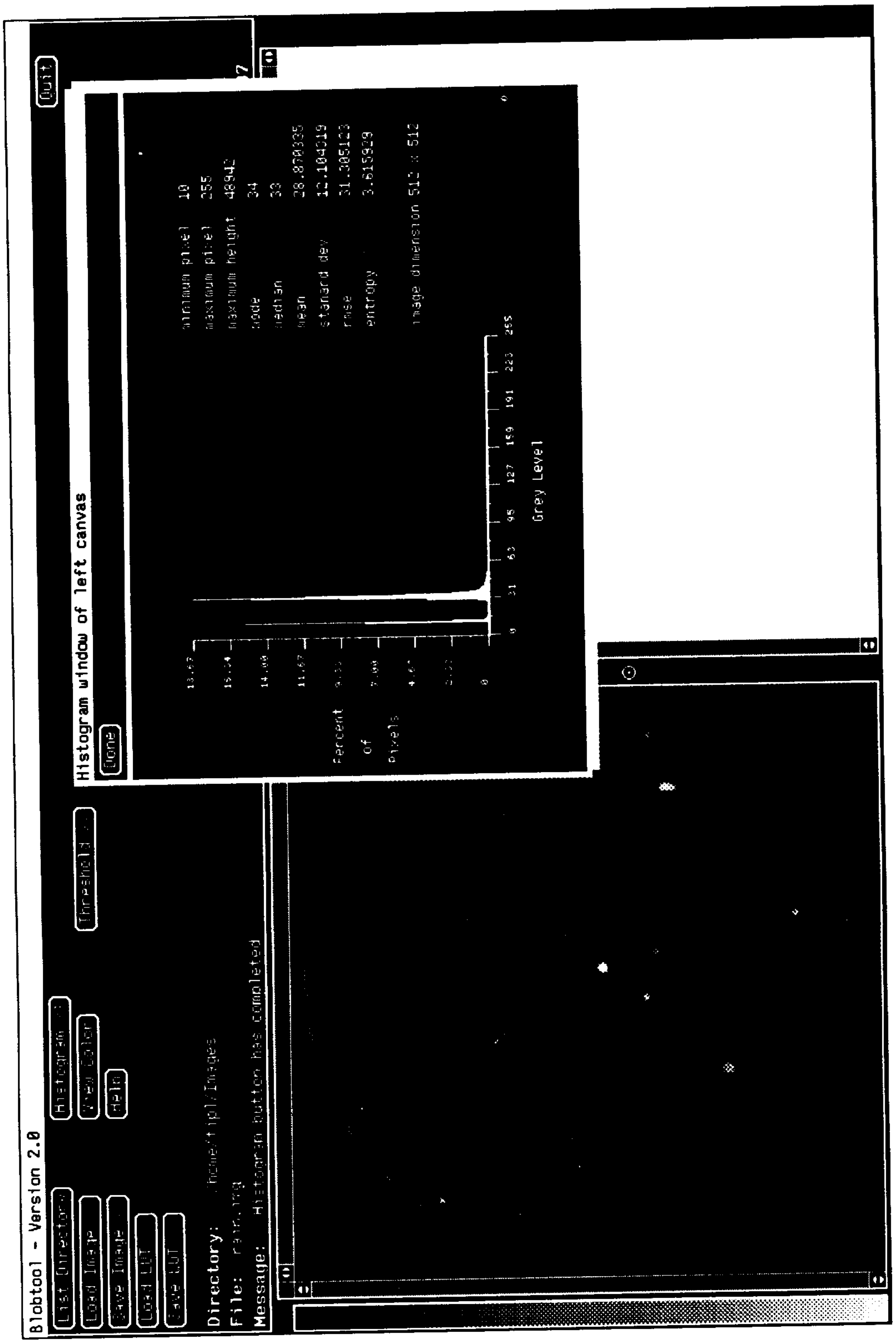
$$\sigma^2_i(t') = \frac{\sum_{l=0}^{t'} (l - \mu_i(t'))^2 p[l]}{P_i(t')}$$

where i=0, 1 and each σ represents the variance of each class

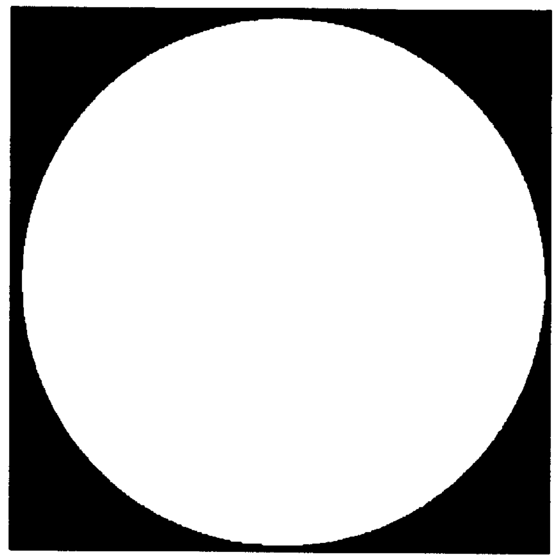
APPENDIX B

APPENDIX B contains test image data that illustrates image segmentation using various thresholding techniques.

PAGE________ INTENTIONALLY BLANK

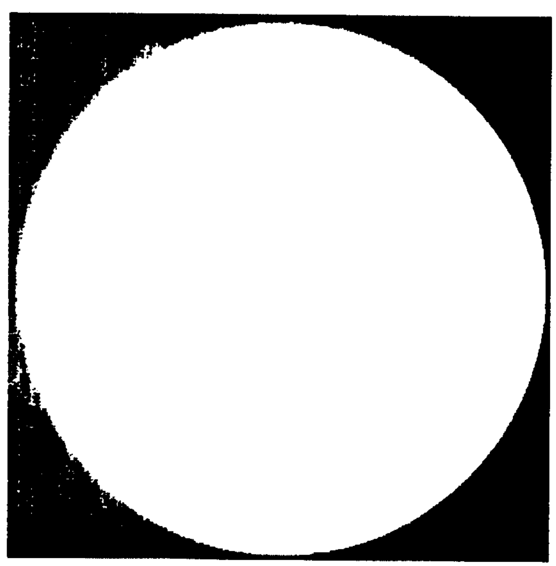


Blobtool: rain.img

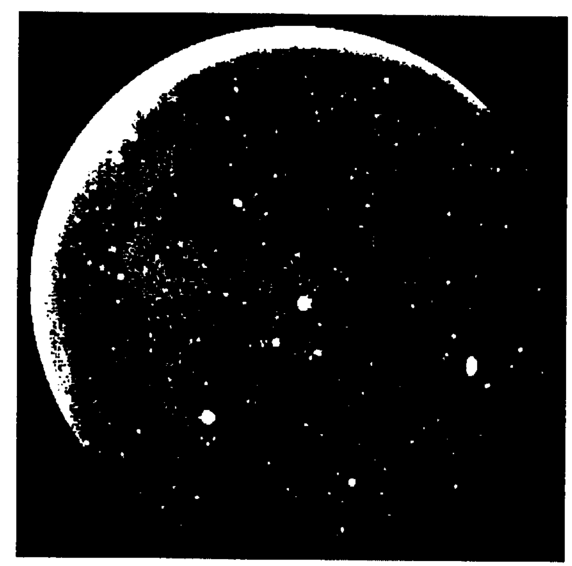


Otsu's Thresholding method, $t^* = 23$

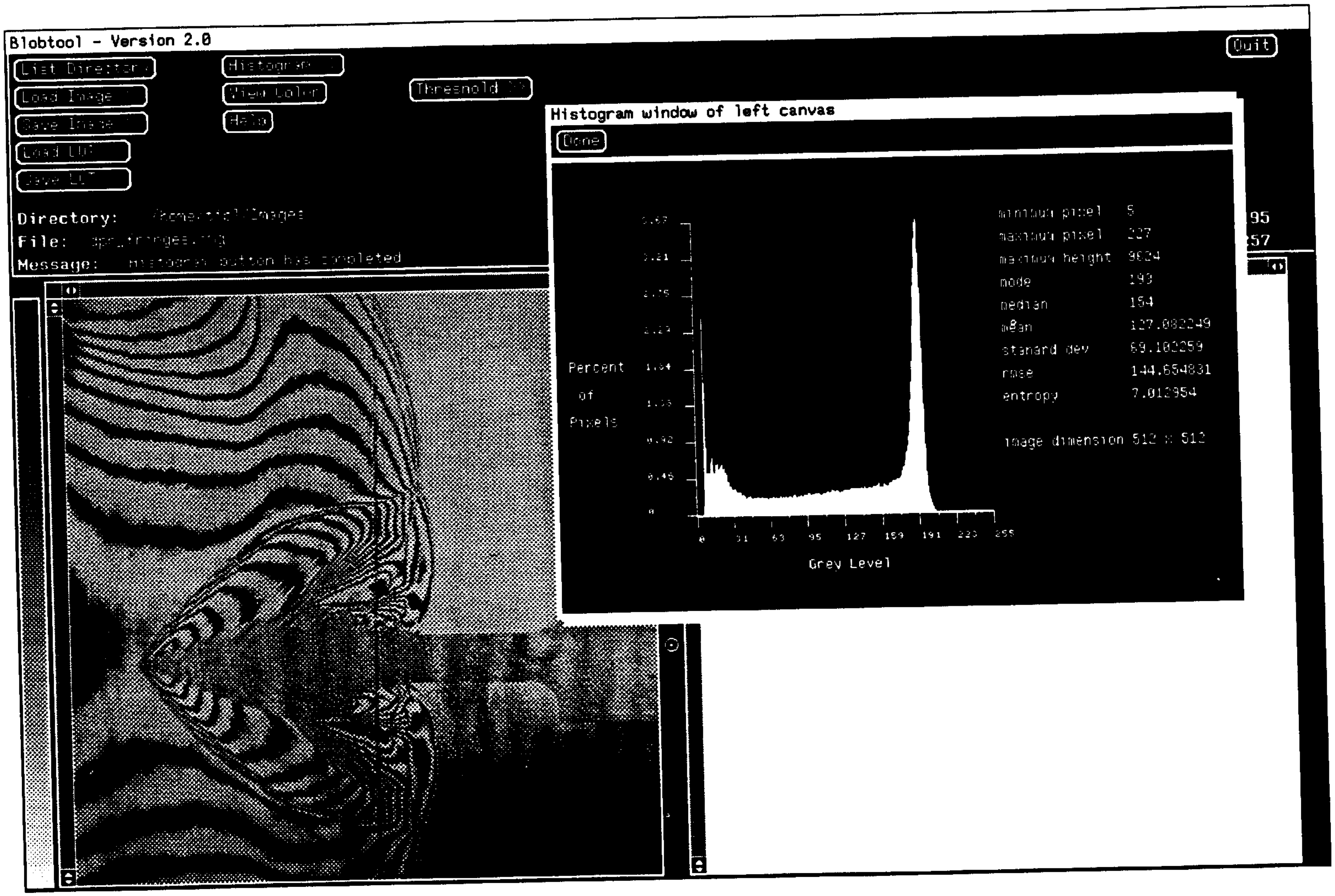
Entropic Thresholding method, $t^* = 82$



Minimum Error Thresholding method, $t^* = 13$



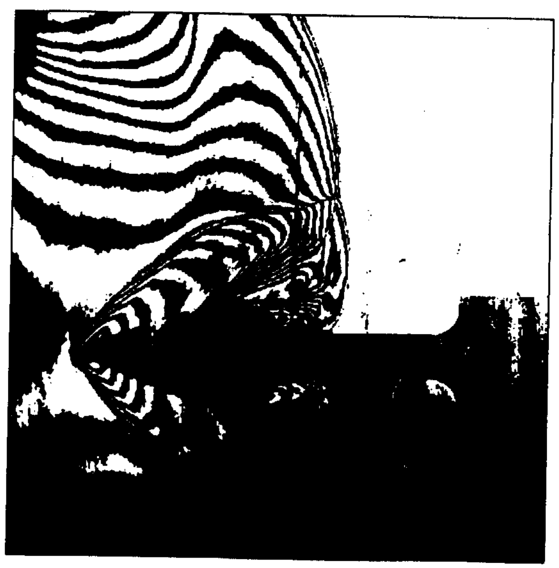
Moment Preservation Thresholding method, $t^* = 37$



Blobtool: fringes.img

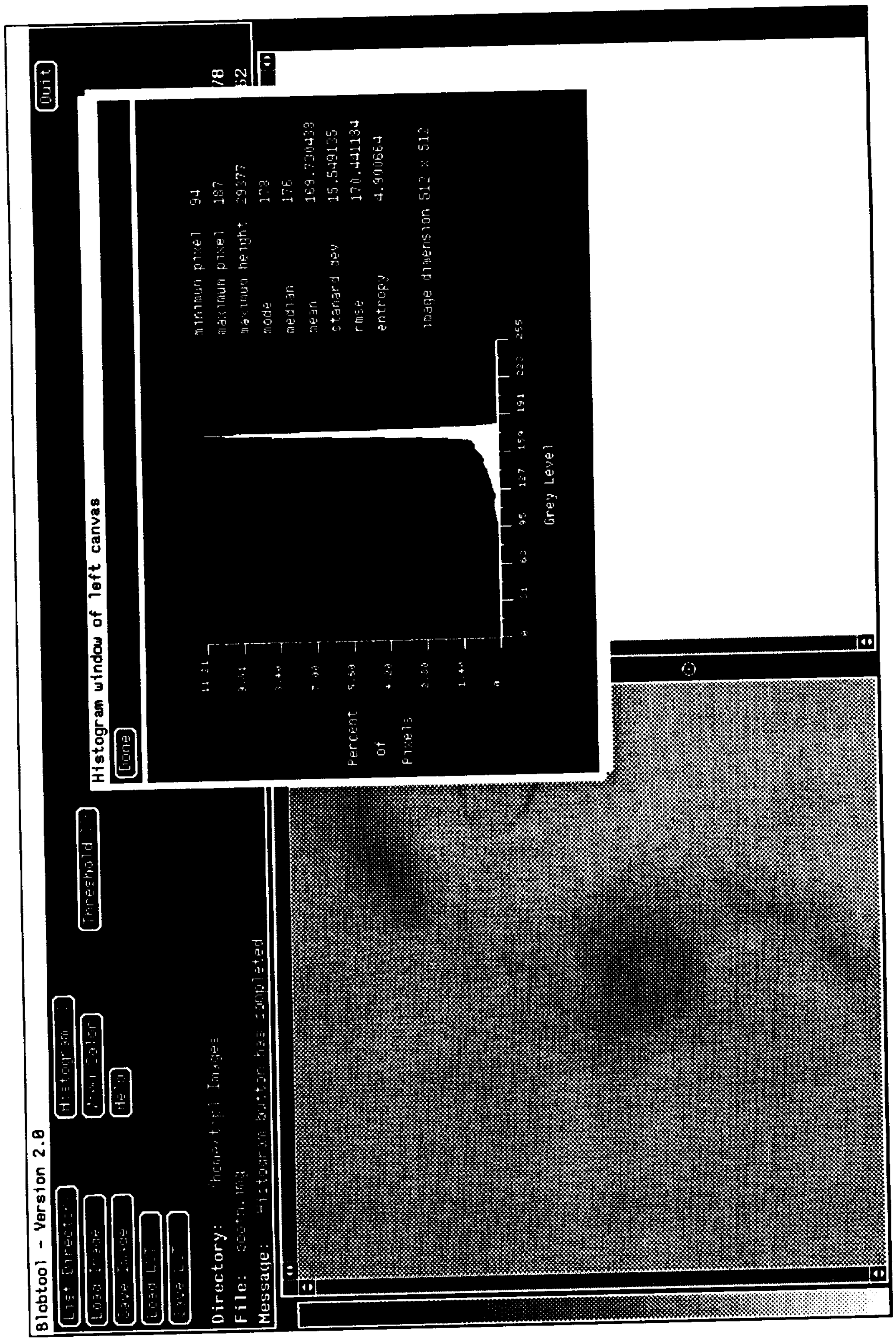
Otsu's Thresholding method, $t^* = 108$

Entropic Thresholding method, $t^* = 112$

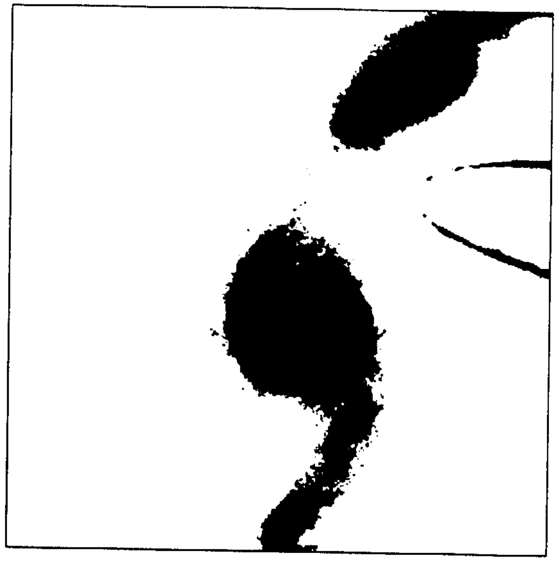


Minimum Error Thresholding method, $t^* = 176$

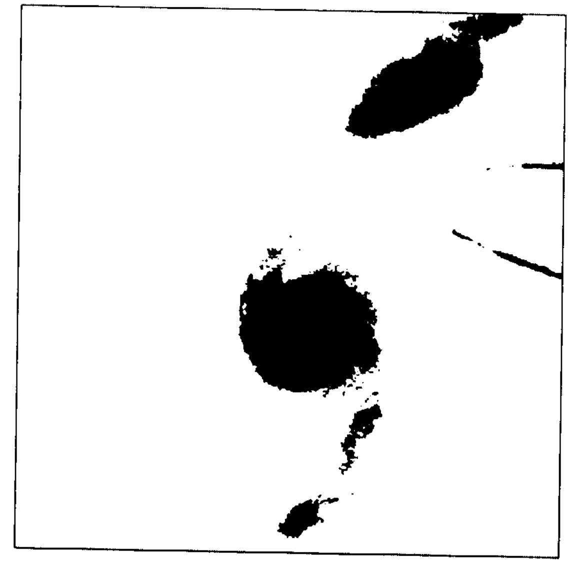
Moment Preservation Thresholding method, $t^* = 110$



Blobtool: booth.img

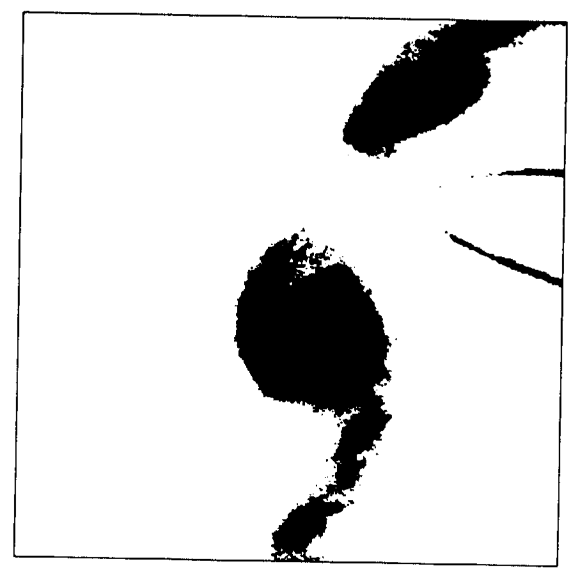


Otsu's Thresholding method, t* = 156



Entropic Thresholding method, t* = 145

Minimum Error Thresholding method, t* = 170



Moment Preservation Thresholding method, t* = 153

APPENDIX C

APPENDIX C contains test image data that illustrates the LAF object recognition technique and the various quantitative measurements that might be generated. The files with extensions ".txt" and ".img" are the input image files, and their respective results are stored in the ".stats" files.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0
0	0	0	0	0	0	0	0	0	0	255	255	255	255	0	0
0	0	0	0	0	0	0	0	0	255	255	255	255	0	0	0
0	0	0	0	0	0	0	0	255	255	255	255	0	0	0	0
0	0	0	0	255	0	0	255	255	255	255	0	0	0	0	0
0	0	0	255	255	255	255	255	255	255	0	0	0	0	0	0
0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0
0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
0	255	255	255	0	255	255	255	255	0	0	0	0	0	0	0
0	255	255	255	0	0	255	255	0	0	0	0	0	0	0	0
0	0	255	255	0	0	255	255	0	0	0	0	0	0	0	0
0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
0	0	255	255	255	255	255	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Image filename: obj4.img
Threshold: Threshold from Single method is 128
Neighborhood: 4 connectedness

object number = 1

2	12	2	13	3	10	3	11	3	12	3	13	4	9	4	10
4	11	4	12	5	8	5	9	5	10	5	11	6	7	6	8
6	9	6	10	7	3	7	4	7	5	7	6	7	7	7	8
7	9	8	2	8	3	8	4	8	5	8	6	8	7	8	8
8	9	9	1	9	2	9	3	9	4	9	5	9	6	9	7
9	8	9	9	10	1	10	2	10	3	11	1	11	2	11	3
12	2	12	3	13	2	13	3	13	4	13	5	13	6	13	7
14	2	14	3	14	4	14	5	14	6	12	6	12	7	11	6
11	7	10	5	10	6	10	7	10	8	6	4				

****Stats for object 1:

Area:	70
Perimeter:	42
Euler:	0
Center X:	6.23
Center Y:	8.53
Minimum Radius:	1.55
Maximum Radius:	9.41
Maximum Radius / Minimum Radius:	6.09
Compactness:	2.01
Orientation:	46.08
Eccentricity:	-0.42
2nd Moment about vertical line through center:	10.23
2nd Moment about horzntl line through center:	10.79
XY moment about the center:	-7.39
1st Moment about x-axis:	597
1st Moment about y-axis:	436
2nd Moment about x-axis:	5847
2nd Moment about y-axis:	3432
XY Moment about the origin:	3201

Writing perimeter points:

2	12	2	13	3	10	3	11	3	13	4	9	4	12	5	8
5	11	6	7	6	10	7	3	7	5	7	6	7	9	8	2
8	9	9	1	9	4	9	9	10	1	10	3	11	1	11	3
12	2	12	3	13	2	13	4	13	5	13	7	14	2	14	3
14	4	14	5	14	6	12	6	12	7	11	6	11	7	10	5
10	8	6	4												

Trilevel Thresholding, t* = 69 and 113
booth.img

Image filename: booth.img
Threshold: Thresholds from Trilevel method are 69 and 113
Neighborhood: 8 connectedness

****Stats for object 1:

Area:	1442
Perimeter:	311
Euler:	-3
Center X:	365.64
Center Y:	77.82
Minimum Radius:	10.28
Maximum Radius:	48.58
Maximum Radius / Minimum Radius:	4.73
Compactness:	5.34
Orientation:	34.22
Eccentricity:	5.84

****Stats for object 2:

Area:	1
Perimeter:	1
Euler:	1
Center X:	404.00
Center Y:	55.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 3:

Area:	1
Perimeter:	1
Euler:	1
Center X:	404.00
Center Y:	67.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 4:

Area:	4
Perimeter:	4
Euler:	1

Center X:	393.50
Center Y:	71.75
Minimum Radius:	0.56
Maximum Radius:	1.68
Maximum Radius / Minimum Radius:	3.00
Compactness:	0.32
Orientation:	162.39
Eccentricity:	0.66

****Stats for object 5:

Area:	1
Perimeter:	1
Euler:	1
Center X:	347.00
Center Y:	75.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 6:

Area:	1
Perimeter:	1
Euler:	1
Center X:	342.00
Center Y:	101.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 7:

Area:	1941
Perimeter:	548
Euler:	-18
Center X:	267.62
Center Y:	302.35
Minimum Radius:	2.40
Maximum Radius:	37.91
Maximum Radius / Minimum Radius:	15.77
Compactness:	12.31
Orientation:	128.90
Eccentricity:	0.25

****Stats for object 8:

Area:	1
Perimeter:	1
Euler:	1
Center X:	239.00
Center Y:	292.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 9:

Area:	1
Perimeter:	1
Euler:	1
Center X:	237.00
Center Y:	306.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 10:

Area:	1
Perimeter:	1
Euler:	1
Center X:	236.00
Center Y:	308.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 11:

Area:	5
Perimeter:	5
Euler:	1
Center X:	236.40
Center Y:	312.00
Minimum Radius:	0.40
Maximum Radius:	1.40
Maximum Radius / Minimum Radius:	3.50
Compactness:	0.40
Orientation:	-0.00

Eccentricity:	0.01
****Stats for object 12:	
Area:	1
Perimeter:	1
Euler:	1
Center X:	271.00
Center Y:	314.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00
****Stats for object 13:	
Area:	2
Perimeter:	2
Euler:	1
Center X:	268.00
Center Y:	318.50
Minimum Radius:	0.50
Maximum Radius:	0.50
Maximum Radius / Minimum Radius:	1.00
Compactness:	0.16
Orientation:	90.00
Eccentricity:	0.03
****Stats for object 14:	
Area:	4
Perimeter:	4
Euler:	1
Center X:	243.25
Center Y:	320.25
Minimum Radius:	0.35
Maximum Radius:	1.27
Maximum Radius / Minimum Radius:	3.61
Compactness:	0.32
Orientation:	161.57
Eccentricity:	0.25
****Stats for object 15:	
Area:	1
Perimeter:	1
Euler:	1
Center X:	265.00
Center Y:	320.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 16:

Area:	5
Perimeter:	5
Euler:	1
Center X:	264.20
Center Y:	323.20
Minimum Radius:	0.28
Maximum Radius:	1.70
Maximum Radius / Minimum Radius:	6.00
Compactness:	0.40
Orientation:	135.03
Eccentricity:	0.29

****Stats for object 17:

Area:	1
Perimeter:	1
Euler:	1
Center X:	244.00
Center Y:	324.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 18:

Area:	1
Perimeter:	1
Euler:	1
Center X:	242.00
Center Y:	327.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 19:

Area:	3
Perimeter:	3
Euler:	1
Center X:	255.00
Center Y:	328.00
Minimum Radius:	0.00
Maximum Radius:	1.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.24
Orientation:	-0.00
Eccentricity:	0.15

****Stats for object 20:

Area:	1
Perimeter:	1
Euler:	1
Center X:	259.00
Center Y:	331.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 21:

Area:	2
Perimeter:	2
Euler:	1
Center X:	262.50
Center Y:	337.00
Minimum Radius:	0.50
Maximum Radius:	0.50
Maximum Radius / Minimum Radius:	1.00
Compactness:	0.16
Orientation:	-0.00
Eccentricity:	0.03

****Stats for object 22:

Area:	1
Perimeter:	1
Euler:	1
Center X:	264.00
Center Y:	339.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 23:

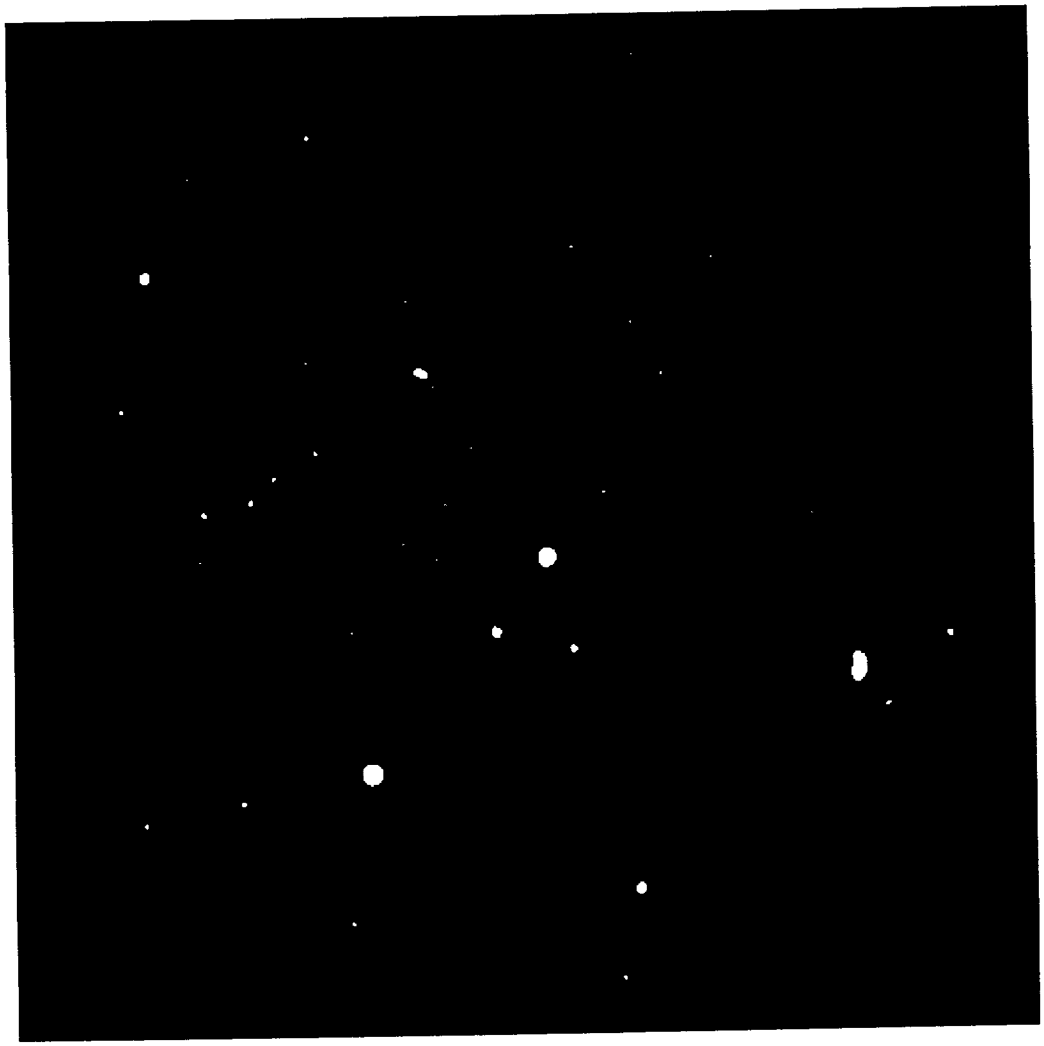
Area:	3
Perimeter:	3
Euler:	1
Center X:	267.00
Center Y:	339.00
Minimum Radius:	0.00
Maximum Radius:	1.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.24
Orientation:	-0.00
Eccentricity:	0.15

****Stats for object 24:

Area:	1
Perimeter:	1
Euler:	1
Center X:	259.00
Center Y:	340.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00
Eccentricity:	0.00

****Stats for object 25:

Area:	3
Perimeter:	3
Euler:	1
Center X:	262.33
Center Y:	340.67
Minimum Radius:	0.47
Maximum Radius:	0.75
Maximum Radius / Minimum Radius:	1.58
Compactness:	0.24
Orientation:	134.23
Eccentricity:	0.15



Single Thresholding, t* = 84
rain.img

```
Image filename: rain.img
Threshold:      Threshold from Single method is 84
Neighborhood:   8 connectedness



****Stats for object 1:

Area:                                          1
Perimeter:                                     1
Euler:                                         1
Center X:                                      313.00
Center Y:                                      20.00
Minimum Radius:                                0.00
Maximum Radius:                                0.00
Maximum Radius / Minimum Radius:               0.00
Compactness:                                   0.08
Orientation:                                   -999.00
Writing perimeter points:
   20  313

****Stats for object 2:

Area:                                          5
Perimeter:                                     5
Euler:                                         1
Center X:                                      149.00
Center Y:                                      60.00
Minimum Radius:                                0.00
Maximum Radius:                                1.00
Maximum Radius / Minimum Radius:               0.00
Compactness:                                   0.40
Orientation:                                   -999.00
Writing perimeter points:
   59  149   60  148   60  149   60  150   61  149

****Stats for object 3:

Area:                                          1
Perimeter:                                     1
Euler:                                         1
Center X:                                      89.00
Center Y:                                      80.00
Minimum Radius:                                0.00
Maximum Radius:                                0.00
Maximum Radius / Minimum Radius:               0.00
Compactness:                                   0.08
Orientation:                                   -999.00
Writing perimeter points:
   80   89
```

****Stats for object 4:

Area:	1
Perimeter:	1
Euler:	1
Center X:	352.00
Center Y:	122.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00

Writing perimeter points:

```
  122  352
```

****Stats for object 5:

Area:	25
Perimeter:	18
Euler:	1
Center X:	67.04
Center Y:	128.60
Minimum Radius:	1.60
Maximum Radius:	2.77
Maximum Radius / Minimum Radius:	1.73
Compactness:	1.03
Orientation:	79.08

Writing perimeter points:

```
  126   67  126   68  127   65  127   66  127   67  127   68  127   69  128   65
  128   69  129   65  129   69  130   65  130   66  130   68  130   69  131   66
  131   67  131   68
```

****Stats for object 6:

Area:	1
Perimeter:	1
Euler:	1
Center X:	198.00
Center Y:	142.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00

Writing perimeter points:

```
  142  198
```

****Stats for object 7:

Area:	1
Perimeter:	1
Euler:	1
Center X:	311.00
Center Y:	154.00
Minimum Radius:	0.00

```
Maximum Radius:                                   0.00
Maximum Radius / Minimum Radius:                  0.00
Compactness:                                      0.08
Orientation:                                      -999.00
Writing perimeter points:
  154  311

****Stats for object 8:

Area:                                             1
Perimeter:                                        1
Euler:                                            1
Center X:                                         147.00
Center Y:                                         172.00
Minimum Radius:                                   0.00
Maximum Radius:                                   0.00
Maximum Radius / Minimum Radius:                  0.00
Compactness:                                      0.08
Orientation:                                      -999.00
Writing perimeter points:
  172  147

****Stats for object 9:

Area:                                             26
Perimeter:                                        20
Euler:                                            1
Center X:                                         205.12
Center Y:                                         178.08
Minimum Radius:                                   0.93
Maximum Radius:                                   3.47
Maximum Radius / Minimum Radius:                  3.73
Compactness:                                      1.22
Orientation:                                      156.63
Writing perimeter points:
  176  203  176  204  176  205  177  202  177  203  177  205  177  206  177  207
  178  202  178  203  178  207  178  208  179  203  179  204  179  205  179  208
  180  205  180  206  180  207  180  208

****Stats for object 10:

Area:                                             2
Perimeter:                                        2
Euler:                                            1
Center X:                                         326.00
Center Y:                                         179.50
Minimum Radius:                                   0.50
Maximum Radius:                                   0.50
Maximum Radius / Minimum Radius:                  1.00
Compactness:                                      0.16
Orientation:                                      90.00
Writing perimeter points:
  179  326  180  326
```

```
****Stats for object 11:

Area:                                      1
Perimeter:                                 1
Euler:                                     1
Center X:                                  211.00
Center Y:                                  185.00
Minimum Radius:                            0.00
Maximum Radius:                            0.00
Maximum Radius / Minimum Radius:           0.00
Compactness:                               0.08
Orientation:                               -999.00
Writing perimeter points:
  185  211

****Stats for object 12:

Area:                                      4
Perimeter:                                 4
Euler:                                     1
Center X:                                  54.50
Center Y:                                  195.50
Minimum Radius:                            0.71
Maximum Radius:                            0.71
Maximum Radius / Minimum Radius:           1.00
Compactness:                               0.32
Orientation:                               -999.00
Writing perimeter points:
  195   54  195   55  196   54  196   55

****Stats for object 13:

Area:                                      1
Perimeter:                                 1
Euler:                                     1
Center X:                                  230.00
Center Y:                                  216.00
Minimum Radius:                            0.00
Maximum Radius:                            0.00
Maximum Radius / Minimum Radius:           0.00
Compactness:                               0.08
Orientation:                               -999.00
Writing perimeter points:
  216  230

****Stats for object 14:

Area:                                      3
Perimeter:                                 3
Euler:                                     1
Center X:                                  151.33
Center Y:                                  217.67
Minimum Radius:                            0.47
Maximum Radius:                            0.75
```

```
Maximum Radius / Minimum Radius:                1.58
Compactness:                                    0.24
Orientation:                                    135.24
Writing perimeter points:
  217  151  218  151  218  152

****Stats for object 15:

Area:                                           3
Perimeter:                                      3
Euler:                                          1
Center X:                                       130.33
Center Y:                                       230.33
Minimum Radius:                                 0.47
Maximum Radius:                                 0.75
Maximum Radius / Minimum Radius:                1.58
Compactness:                                    0.24
Orientation:                                    45.07
Writing perimeter points:
  230  130  230  131  231  130

****Stats for object 16:

Area:                                           2
Perimeter:                                      2
Euler:                                          1
Center X:                                       296.50
Center Y:                                       239.00
Minimum Radius:                                 0.50
Maximum Radius:                                 0.50
Maximum Radius / Minimum Radius:                1.00
Compactness:                                    0.16
Orientation:                                    -0.00
Writing perimeter points:
  239  296  239  297

****Stats for object 17:

Area:                                           5
Perimeter:                                      5
Euler:                                          1
Center X:                                       118.60
Center Y:                                       242.20
Minimum Radius:                                 0.45
Maximum Radius:                                 1.26
Maximum Radius / Minimum Radius:                2.83
Compactness:                                    0.40
Orientation:                                    71.63
Writing perimeter points:
  241  119  242  118  242  119  243  118  243  119
```

```
****Stats for object 18:

Area:                                          6
Perimeter:                                     6
Euler:                                         1
Center X:                                      95.17
Center Y:                                      248.17
Minimum Radius:                                0.24
Maximum Radius:                                1.18
Maximum Radius / Minimum Radius:               5.00
Compactness:                                   0.48
Orientation:                                   135.16
Writing perimeter points:
  247   95  248   94  248   95  248   96  249   95  249   96

****Stats for object 19:

Area:                                          1
Perimeter:                                     1
Euler:                                         1
Center X:                                      401.00
Center Y:                                      251.00
Minimum Radius:                                0.00
Maximum Radius:                                0.00
Maximum Radius / Minimum Radius:               0.00
Compactness:                                   0.08
Orientation:                                   -999.00
Writing perimeter points:
  251  401

****Stats for object 20:

Area:                                          1
Perimeter:                                     1
Euler:                                         1
Center X:                                      195.00
Center Y:                                      264.00
Minimum Radius:                                0.00
Maximum Radius:                                0.00
Maximum Radius / Minimum Radius:               0.00
Compactness:                                   0.08
Orientation:                                   -999.00
Writing perimeter points:
  264  195

****Stats for object 21:

Area:                                          72
Perimeter:                                     34
Euler:                                         1
Center X:                                      267.78
Center Y:                                      271.39
Minimum Radius:                                3.43
Maximum Radius:                                4.94
```

Maximum Radius / Minimum Radius:	1.44
Compactness:	1.28
Orientation:	75.95

Writing perimeter points:

```
  267  266  267  267  267  268  267  269  268  265  268  266  268  269  268  270
  268  271  269  264  269  265  269  271  270  264  270  271  270  272  271  264
  271  272  272  264  272  272  273  264  273  271  273  272  274  264  274  265
  274  270  274  271  275  265  275  266  275  268  275  269  275  270  276  266
  276  267  276  268
```

****Stats for object 22:

Area:	1
Perimeter:	1
Euler:	1
Center X:	93.00
Center Y:	272.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00

Writing perimeter points:

```
  272   93
```

****Stats for object 23:

Area:	1
Perimeter:	1
Euler:	1
Center X:	212.00
Center Y:	272.00
Minimum Radius:	0.00
Maximum Radius:	0.00
Maximum Radius / Minimum Radius:	0.00
Compactness:	0.08
Orientation:	-999.00

Writing perimeter points:

```
  272  212
```

****Stats for object 24:

Area:	22
Perimeter:	17
Euler:	1
Center X:	241.77
Center Y:	308.73
Minimum Radius:	1.26
Maximum Radius:	2.83
Maximum Radius / Minimum Radius:	2.25
Compactness:	1.05
Orientation:	107.49

```
Writing perimeter points:
  306  241  307  240  307  241  307  242  307  243  308  240  308  243  308  244
  309  240  309  243  309  244  310  240  310  241  310  243  311  241  311  242
  311  243

****Stats for object 25:

Area:                                         1
Perimeter:                                    1
Euler:                                        1
Center X:                                     169.00
Center Y:                                     308.00
Minimum Radius:                               0.00
Maximum Radius:                               0.00
Maximum Radius / Minimum Radius:              0.00
Compactness:                                  0.08
Orientation:                                  -999.00
Writing perimeter points:
  308  169

****Stats for object 26:

Area:                                         8
Perimeter:                                    8
Euler:                                        1
Center X:                                     469.12
Center Y:                                     312.88
Minimum Radius:                               0.18
Maximum Radius:                               1.43
Maximum Radius / Minimum Radius:              8.06
Compactness:                                  0.64
Orientation:                                  -45.00
Writing perimeter points:
  312  468  312  469  312  470  313  468  313  469  313  470  314  469  314  470


****Stats for object 27:

Area:                                         11
Perimeter:                                    11
Euler:                                        1
Center X:                                     280.64
Center Y:                                     317.45
Minimum Radius:                               0.58
Maximum Radius:                               1.70
Maximum Radius / Minimum Radius:              2.92
Compactness:                                  0.88
Orientation:                                  108.23
Writing perimeter points:
  316  280  316  281  317  279  317  280  317  281  317  282  318  280  318  281
  318  282  319  280  319  281
```

****Stats for object 28:

Area:	95
Perimeter:	42
Euler:	1
Center X:	423.59
Center Y:	329.00
Minimum Radius:	2.59
Maximum Radius:	7.18
Maximum Radius / Minimum Radius:	2.77
Compactness:	1.48
Orientation:	86.53

Writing perimeter points:

```
322 422 322 423 322 424 323 421 323 422 323 424 323 425 323 426
324 421 324 426 325 421 325 426 325 427 326 421 326 427 327 421
327 427 328 421 328 427 329 421 329 427 330 420 330 421 330 427
331 420 331 427 332 420 332 426 332 427 333 420 333 421 333 426
334 421 334 425 334 426 335 421 335 422 335 424 335 425 336 422
336 423 336 424
```

****Stats for object 29:

Area:	4
Perimeter:	4
Euler:	1
Center X:	438.00
Center Y:	347.50
Minimum Radius:	0.50
Maximum Radius:	1.12
Maximum Radius / Minimum Radius:	2.24
Compactness:	0.32
Orientation:	31.72

Writing perimeter points:

```
347 438 347 439 348 437 348 438
```

****Stats for object 30:

Area:	89
Perimeter:	37
Euler:	1
Center X:	178.49
Center Y:	379.56
Minimum Radius:	4.25
Maximum Radius:	5.46
Maximum Radius / Minimum Radius:	1.29
Compactness:	1.22
Orientation:	84.80

Writing perimeter points:

```
375 176 375 177 375 178 375 179 375 180 375 181 376 175 376 176
376 181 376 182 377 174 377 175 377 182 377 183 378 174 378 183
379 174 379 183 380 174 380 183 381 174 381 183 382 174 382 175
382 182 382 183 383 175 383 176 383 181 383 182 384 176 384 177
384 178 384 179 384 180 384 181 385 178
```

```
****Stats for object 31:

Area:                                        4
Perimeter:                                   4
Euler:                                       1
Center X:                                    113.50
Center Y:                                    393.50
Minimum Radius:                              0.71
Maximum Radius:                              0.71
Maximum Radius / Minimum Radius:             1.00
Compactness:                                 0.32
Orientation:                                 -999.00
Writing perimeter points:
  393  113  393  114  394  113  394  114

****Stats for object 32:

Area:                                        4
Perimeter:                                   4
Euler:                                       1
Center X:                                    64.75
Center Y:                                    404.00
Minimum Radius:                              0.25
Maximum Radius:                              1.03
Maximum Radius / Minimum Radius:             4.12
Compactness:                                 0.32
Orientation:                                 90.00
Writing perimeter points:
  403   65  404   64  404   65  405   65

****Stats for object 33:

Area:                                        24
Perimeter:                                   18
Euler:                                       1
Center X:                                    313.12
Center Y:                                    438.67
Minimum Radius:                              1.31
Maximum Radius:                              2.81
Maximum Radius / Minimum Radius:             2.15
Compactness:                                 1.07
Orientation:                                 70.68
Writing perimeter points:
  436  313  436  314  437  312  437  313  437  314  437  315  438  311  438  312
  438  315  439  311  439  315  440  311  440  312  440  314  440  315  441  312
  441  313  441  314

****Stats for object 34:

Area:                                        3
Perimeter:                                   3
Euler:                                       1
Center X:                                    168.33
Center Y:                                    454.67
```

Minimum Radius:	0.47
Maximum Radius:	0.75
Maximum Radius / Minimum Radius:	1.58
Compactness:	0.24
Orientation:	134.52

Writing perimeter points:
  454 168 455 168 455 169

****Stats for object 35:

Area:	3
Perimeter:	3
Euler:	1
Center X:	304.67
Center Y:	483.33
Minimum Radius:	0.47
Maximum Radius:	0.75
Maximum Radius / Minimum Radius:	1.58
Compactness:	0.24
Orientation:	136.06

Writing perimeter points:
  483 304 483 305 484 305

PAGE ______ INTENTIONALLY BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Contractor Report

4. TITLE AND SUBTITLE
Analysis of Objects in Binary Images

5. FUNDING NUMBERS
C NAS1-19038
WU 992-16-04-05

6. AUTHOR(S)
Desiree M. Leonard

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Sciences Corporation
3217 N. Armistead Avenue
Hampton, Virginia 23666

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-4420

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Mary-Anne Posenau
New Technology Report: Interactive Image Analysis Program, (July 1991)
Cosmic Release: BLOBTOOL - Interactive Image Analysis Program (LAR-14767)

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 59

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Digital image processing techniques are typically used to produce improved digital images through the application of successive enhancement techniques to a given image or to generate quantitative data about the objects within that image. In support of and to assist researchers in a wide range of disciplines, e.g., interferometry, heavy rain effects on aerodynamics, and structure recognition research, it is often desirable to count objects in an image and compute their geometric properties. Therefore, an image analysis application package, focusing on a subset of image analysis techniques used for object recognition in binary images, was developed. This report describes the techniques and algorithms utilized in the three main phases of the application and are categorized as: image segmentation, object recognition, and quantative analysis. Appendices provide supplemental formulas for the algorithms employed, as well as, examples and results from the various image segmentation techniques and the object recognition algorithm implemented.

14. SUBJECT TERMS
Digital Image Processing
Image Analysis
Image Processing
Image Segmentation
Object Recognition

15. NUMBER OF PAGES
72

16. PRICE CODE
A04

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

PRECEDING PAGE BLANK NOT FILMED

NASA-Langley, 1991

National Aeronautics and
Space Administration

NASA