

**SEGMENTATION OF INFRARED IMAGES**

by

**ANUJ DHUNGANA, B.E.E.**

**A THESIS**

**IN**

**COMPUTER SCIENCE**

Submitted to the Graduate Faculty  
of Texas Tech University in  
Partial Fulfillment of  
the Requirements for  
the Degree of

**MASTER OF SCIENCE**

Approved

6-20  
2002

AC  
805

T3

2002

No. 51

Lop. 2

## ACKNOWLEDGMENTS

I am thankful to Dr. Gopal Lakhani, my thesis advisor, for his constant support, without which it would have been impossible to complete this work. I would also like to thank my thesis committee members Dr. Eric D. Sinzinger and Dr. Hamed Sari-Sarraf for their insightful suggestions.

I am also thankful to graduate school, Texas Tech University, for providing financial support as the summer thesis award for the completion of this thesis.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
ABSTRACT	vi
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
I. INTRODUCTION	1
1.1. Overview	1
1.2. Goals of This Work	3
1.3. Organization of Thesis	5
II. LITERATURE REVIEW	6
2.1 Infrared (IR) Images	6
2.2 Segmentation	8
2.2.1 Thresholding Based Segmentation	9
2.2.2 Boundary Based Segmentation	13
2.2.3 Region Based Segmentation	13
2.2.4 Hybrid Technique	14
2.3 Application of Image Segmentation	15
III. INFRARED IMAGE SEGMENTATION	16
3.1 A Region-Growing Segmentation Algorithm	16
3.1.1 Performance on IR Images	19

3.2 Fuzzy Logic-Based Segmentation	20
3.2.1 Concept of Fuzziness	21
3.2.2 Using Fuzzy Logic for Segmentation of Infrared Images	22
3.2.3 Limitations of Sun-Gu Sun's Fuzzy Logic-Based Segmentation	28
3.3 Fujimura's Approach	29
3.4 Otsu's Thresholding	30
3.5 Adaptive Global Thresholding	30
3.6 Our Implementation	34
3.6.1 Implementation for Multiple Object Refinement	34
3.6.2 Adaptive Global Thresholding with Maxima-Based Mode Detection	36
3.7 Experiments and Results	39
3.8 Limitations of Our Method	52
<b>IV. OBJECT MATCHING/TRACKING IN IR IMAGE SEQUENCES</b>	<b>53</b>
4.1 Introduction	53
4.2 Feature-Based Inter-Frame Object Matching	55
4.3 Feature-Based Object matching for Synthesized Video Sequence	56
4.4 Experiments	59
4.4.1 Development of a Video Sequence for Object Tracking	59
4.5 Results	61
4.5.1 Results of Feature-Based Matching	62
4.5.2 Another Result of Feature-Based Matching	64
4.5.3 Results on Object Tracking	68

4.6 Summary	70
V. SUMMARY AND CONCLUSIONS	71
5.1 Summary	71
5.2 Conclusions	72
REFERENCES	75
APPENDIX	78

## ABSTRACT

In this thesis, we study some segmentation methods for infrared images. First we summarize the characteristics of infrared images and survey different approaches of segmentation. Our objective is to develop an image segmentation method for surveillance images, which helps isolate all major objects in the image. We study fuzzy logic based segmentation for this purpose. Our study shows that use of a suitable thresholding method is important prior to use of the fuzzy logic based method for refinement of boundaries of objects of interest. The existing fuzzy logic based method requires prior knowledge of the size of the target objects. We remove this requirement by using adaptive global thresholding heuristic based on image histogram. Next we extend this segmentation method for object matching between frames of surveillance image sequence and for object tracking.

## LIST OF TABLES

2.1	Frequency Band and Image Capturing Phenomenon	6
3.1	Global Threshold with Different Approaches	40
4.1	Object Data Structure	57
4.2	Region Attributes of Frames 1 and 2 for Triangle Image	62
4.3	Overall Match Score of Triangle Image Sequence	63
4.4	Object Attributes of Frames 1 and 2 for Plate Image	65
4.5	Overall Match Score of Plate Image Sequence	65
4.6	Region Attributes of Distant Frames of Triangle Image	67
4.7	Overall Match Score for Distant Frames of Triangle Image	67

## LIST OF FIGURES

2.1	Infrared Image of a Tank	7
2.2	Histogram of Tank Image	8
2.3	Bimodal Thresholding	9
2.4	Unimodal Thresholding	12
3.1	IR Image with General Segmentation	19
3.2	Image with ROI and Reference Region	23
3.3	Tank Image Binarized and Labeled at Threshold 176	26
3.4	Membership Histogram for Region $R_b$ in ROI	26
3.5	Membership Histogram for Region $R_r$ in ROI	27
3.6	Membership Histogram for Region ROI	27
3.7	Segmented Result after Refinement with Fuzzy Thresholding	28
3.8	Bi-Modal Histogram with High Contrast	31
3.9	Unimodal Small Size Object	32
3.10	Multimode Histogram with Complex Background and Small Object	32
3.11	Original Infrared Image ‘mecycle.pgm’	41
3.12	Histogram of Image ‘mecycle.pgm’	42
3.13	Image ‘mecycle.pgm’ Thresholded with Our Approach at 157	42
3.14	Image ‘mecycle.pgm’ Thresholded with Fujimura’s Approach at 220	43
3.15	Image ‘mecycle.pgm’ Thresholded using Otsu’s Approach at 74	43

3.16	Image ‘mancycle.pgm’ before Applying Refinement at 157 Threshold	44
3.17	Image ‘mancycle.pgm’ before Applying Refinement at 220 Threshold	44
3.18	Image ‘mancycle.pgm’ after Refinement at 157 Threshold	45
3.19	Image ‘mancycle.pgm’ after Refinement at 220 Threshold	45
3.20	Original Infrared Image ‘horses.pgm’	46
3.21	Histogram of Image ‘horses.pgm’	47
3.22	Image ‘horses.pgm’ Thresholded with Our Approach	47
3.23	Image ‘horses.pgm’ Thresholded with Fujimura’s Approach	48
3.24	Image ‘horses.pgm’ with Otsu’s Threshold	48
3.25	Image ‘horses.pgm’ after Refinement (Our Approach)	48
3.26	Image ‘horses.pgm’ after Refinement (Fujimura’s Approach)	49
3.27	Refined Image of Tank with Our Approach	50
3.28	Refined Image of Tank with Otsu’s Approach	50
3.29	Refined Image of Tank with Fujimura’s Approach	51
3.30	Refined Image of Tank with Sun-Gu Sun Approach	51
4.1	Two Consecutive Frames with Three Objects	62
4.2	Consecutive Frames Containing Different Bright Regions	64
4.3	First Frame of Figure 4.2 after Segmentation	65
4.4	Two Frames of a Video Sequence 90 Frames Apart	66
4.5	Sequence of Frames with Object Leaving and Entering the Scene	68
4.6	First and Last Frame of 17-Frame Image Sequence	69
4.7	Centroid of the Matched Object Tracked for 17 Consecutive Frames	69

A.1	A Sample Frame from Simulated Video Sequence	79
A.2	Original Image ‘swimmer.pgm’	79
A.3	Image ‘swimmer.pgm’ Segmented with Our Approach	80
A.4	Image ‘swimmer.pgm’ Segmented with Fujimura’s Approach	80
A.5	Segmentation of Tank Image with General Homogeneity Based Criterion	81

# CHAPTER I

## INTRODUCTION

### 1.1 Overview

Use of information stored in the form of images and video sequences is growing very fast. Therefore, development of efficient methods for acquisition, analysis, compression and manipulation of visual data in digital format is now a focal point of the ongoing revolution in information technology. As more and more image data are acquired managing and manipulating them as information has become more important in order to take full advantage of the information. An image contains information in the form of objects, colors, texture, edges, motion, size, shape, etc. Human perception is highly complex; it can extract this information easily by analyzing and relating different attributes of image simultaneously. There is no single image-processing program that automatically extracts information all by itself. Therefore, a number of image processing applications have been developed, which can extract information like objects boundaries, textures and motion separately. Some of the major image and video processing tasks are edge detection, motion detection, object recognition and object tracking. These tasks are often automated and used in the way that suits different fields of their applications like medicine, space exploration, entertainment and security.

One of basic and important task in the course of extracting information from an image is segmentation. No matter whether the goal is to do object recognition, tracking, data compression or scene interpretation, image segmentation is very desired. No

segmentation technique is universally applicable, which works equally well for all kinds of images. Therefore, various segmentation approaches have been developed which perform differently for different applications as well for different types of images.

Segmentation approaches can be divided into four categories: thresholding, region-based, boundary-based and hybrid. These approaches are described in Chapter II. Images can be classified on the basis of their acquisition from the nature. This diversity holds, because images can be acquired from every type of radiation. Radiation in the visible range of spectrum is used to capture photographic images; other types are infrared, UV, X-Rays, gamma rays, cosmic rays, etc. All objects live or dead and of any color, emit infrared radiation by virtue of their temperature; the exact degree of radiation is determined by the absolute temperature and the thermal characteristics of the material from which the object is made [29]. Infrared imaging technology has been applied for the last 20 years to a variety of applications. Infrared imaging is widely used in military services for general-purpose surveillance as well as for combat missions. Infrared imaging is receiving much greater attention of the U.S national defense, due to increased security concerns, and it is a motivation for our study.

As mentioned before image segmentation is application dependent and success of the segmentation is measured by the needs of the application. For example, in medicine, segmentation is used to automatically isolate segments containing tissue and organs from the images. In this application; the result of a segmentation program is often compared with the outline of organs drawn by hand. In object recognition and tracking applications, segmentation results are assumed to be good, only if they give positive results for

tracking when compared with the actual data. In other applications, where perceptual recovery or enhancement of image is the purpose of segmentation, results are often compared among similar other methods on the basis of visual inspection.

## 1.2 Goals of This Work

There are many approaches for image segmentation, and no single approach is suitable for all applications. Different features and/or homogeneity criteria generally lead to different segmentations of the same data; for example, color segmentation, textures segmentation and motion segmentation usually result in different segmentation maps. Furthermore, there is no guarantee that the resulting segmentations will be semantically meaningful, since a semantically meaningful region may contain multiple colors, textures, or motion. Various types of segmentation techniques have been applied to images on the basis of need of applications in the past [3, 4, 8, 11, 13, 19, 23, 12, 33]. In this thesis, we study application of two segmentation approaches on IR images, namely, region growing, and thresholding. We study three problems,

- i. Given an IR image identify regions containing objects of interest,
- ii. Given a sequence of IR images, match objects of interest in successive frames,  
and,
- iii. Track moving objects after they are recognized in consecutive frames.

Since infrared images map temperature of objects to different colors or gray levels, the objects of interest are brighter than their background. Therefore, first task is to segment images by thresholding, applicable to gray images. In infrared images, the gray

level distribution within objects is usually not uniform. Therefore, fuzzy logic based approach may be used to refine the regions found by thresholding to obtain objects contained in them. We tune our study to infrared surveillance images. We assume that these images contain a small number of objects and most of them are stationary. A surveillance image generally covers a large area and only a portion of the image shows important information. Therefore, the first task is to determine regions of interest in surveillance image. We will concentrate on this problem initially. We note that automatic object recognition and/or tracking moving objects is important to US military and border patrol.

To attain above goals, we reviewed several segmentation techniques and implemented some to study their performance on infrared images. We use a simple homogeneity criterion for segmenting images and study different approaches for thresholding in depth. In infrared images, although most part of the object of interest is usually brighter than the background, determination of gray level distribution in and around the objects is a complex task, because the images contain a lot of noise. Therefore, we will use a combination of thresholding-based method with fuzzy logic based method to obtain better segmentation results. Thresholding based methods are faster and efficient, and therefore they suit for tracking problems. By reviewing and suggesting improvements in the existing fuzzy logic based thresholding technique to automatically identify regions of interest, we hope to develop clear understanding of general surveillance images.

### 1.3 Organization of Thesis

This thesis comprises of five chapters. The first chapter gives brief introduction to the need of infrared image processing, segmentation and the goals of the thesis. Chapter II reviews literature on image segmentation. In Chapter II, first we present our view of infrared images, then give a detailed description of thresholding-based segmentation and finally review some other segmentation techniques.

In Chapter III, first we describe image segmentation algorithm of [24] developed for general photographic colored image; it is based on the region growing approach. Then we present results of its application on an IR image and note down its limitations for our study. Next we present segmentation based on thresholding using fuzzy logic of [6], and again list its limitations for our application of surveillance images. Then we study characteristics of histograms and develop an adaptive heuristic for obtaining suitable thresholds. Experimental results are presented to compare our implementation with [6, 33, 26].

Purpose of the study of Chapter III is to obtain region of interest in IR images. In Chapter IV, we explore how segmentation results can be used to match objects in video sequence for motion detection. We created a video sequence for our experiments. Results are presented for this sequence only. We conclude in Chapter V and present some future works.

## CHAPTER II

### LITERATURE REVIEW

#### 2.1 Infrared Images

All objects emit electromagnetic radiation. Thermal energy given off by an object is spread across all wavelengths peaking at a wavelength determined by the object's temperature. Spectrally infrared radiation is located between the visible and radio frequencies. Hot objects emit peak radiation at short wavelengths, while cooler objects do so at longer wavelengths. For example sun emits mostly visible radiation, and people emit mostly thermal infrared radiation. Thermal infrared cameras record this emitted radiation. Non-living objects also emit radiation. The variation of reflectance with wavelength helps determine what an object looks like in visible and near-infrared bands.

Table 2.1 shows the different frequency band and image-capturing phenomenon.

Table 2.1 Frequency Band and Image Capturing Phenomenon

Image Type	Thermal IR	Near IR	Visible
Frequency band	$10^{-4}$ - $10^{-6}$ m	$10^{-6}$ m(1200-700nm)	700-400 nm
Way of Capturing	Thermal Radiation	Reflectance	Scattering

An infrared image is captured by sampling the thermal radiation from a scene to produce a 2D map of the observed irradiance representing temperature [1], emissivity and reflectivity variations in the scene. Thermal radiation is produced by all surfaces with

temperature above zero degrees Kelvin. Therefore, a thermal image may be produced without any active visible light source. This lack of requirement of illumination makes infrared images popular for military application, surveillance and other night vision applications. Figure 2.1 shows a typical infrared image with a tank in the foreground. Some infrared cameras even produce color images but here we shall consider only grey level images for study of image segmentation. Figure 2.2 shows the histogram of the image in Figure 2.1. Histogram of the digital image is a plot or graph of the frequency of occurrence of each gray level. Thus a gray level histogram consists of 255 bins each bin corresponding to a different gray level. The histogram in Figure 2.1 contains basically two peaks. One of the peaks corresponds to the background and the other to the foreground or object. The mode corresponding to the object is brighter or warmer and contains sharp peak at the higher gray level. Another peak corresponding to background contains large number of pixels at lower gray level because it is cooler.

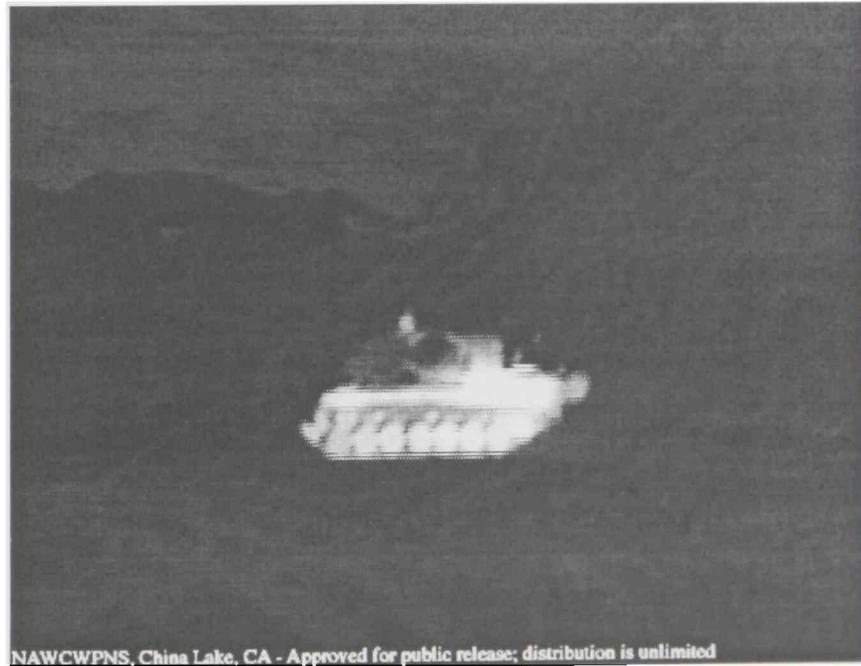


Figure 2.1 Infrared Image of a Tank

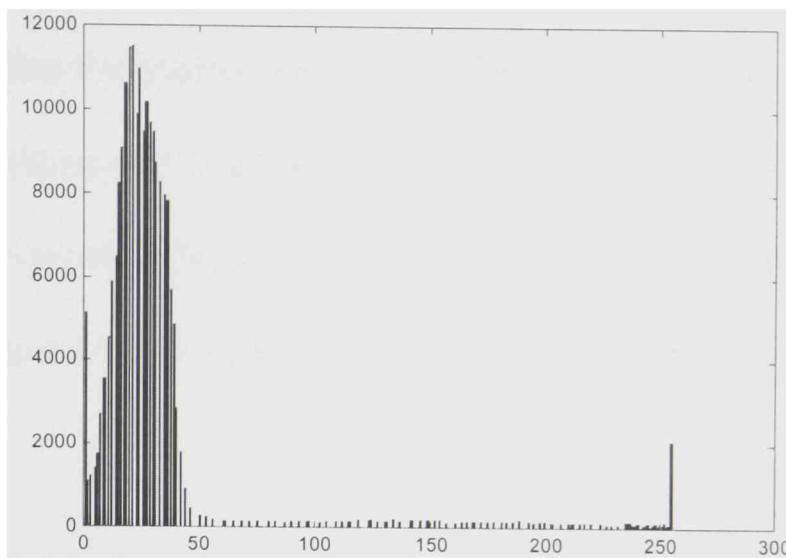


Figure 2.2 Histogram of Tank Image

## 2.2 Segmentation

Image segmentation is a low level image processing task that aims at partitioning an image into homogenous regions before performing any other high-level tasks such as object recognition or scene interpretation. Homogeneous here can be quantified as similar in color, texture, brightness, or motion with some tolerance. The human eye and its intelligence helps to identify objects by relating various features such as, texture, color, and motion simultaneously. But in digital image processing, we must first define the criterion of homogeneity and then proceed to segment on that regard. In some photographic images colors can be good criterion of homogeneity, in agriculture field survey texture might be better criterion for segmentation. In infrared images gray level may be chosen as homogeneity criterion. In some of infrared images simple binarization with certain gray level threshold even gives good result of segmentation. Segmentation can basically be done with two different approaches data driven and model based. Data

driven approach uses low-level image information while model based approach uses high-level information for segmentation. In the following, we discuss some commonly used data driven techniques for image segmentation. Due to the simplicity and wide applicability of thresholding-based segmentation in infrared images, a detailed description of various ways of thresholding is discussed along with the other methods of segmentation.

### 2.2.1 Thresholding Based Segmentation

For gray images, separation of darker background from brighter foreground is thresholding. Thresholding is process of binarization of the image into two levels (bi-level): foreground and background. In the simplest kind of images, this can be done by thresholding at the valley between the two modes of histogram (see Figure 2.3) [14].

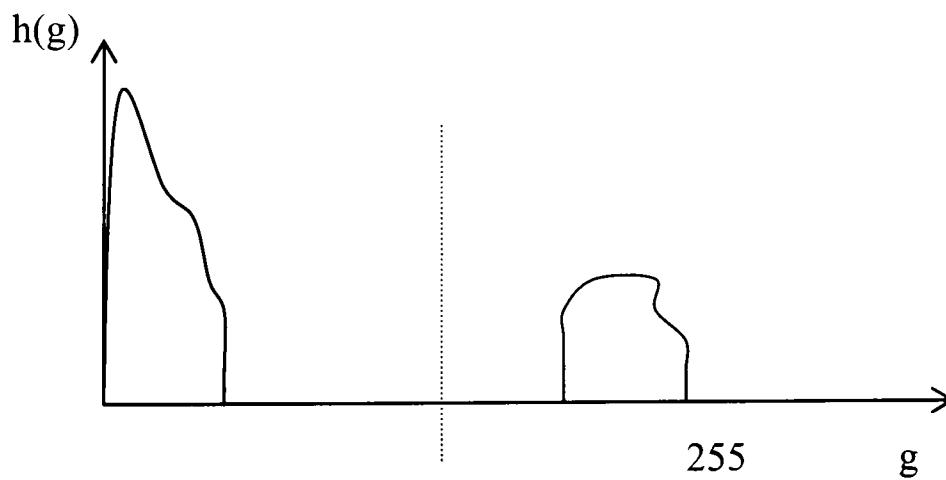


Figure 2.3 Bimodal Thresholding

Thresholding of complex images that do not have such simple histograms require some other techniques. Chow and Kaneko [18] suggested using a threshold that is determined from the spatially local-area around the pixels to which the threshold applies.

They propose neighborhood of size 33 by 33 or 65 by 65 to compute the local histogram. To reduce the computational cost, they divide the image into mutually exclusive blocks, compute the histogram considering pixels in each block, determine an appropriate threshold for each histogram, and then spatially interpolate the threshold values to obtain a spatially adaptive threshold.

Weszka, Nagel, and Rosenfeld [21] suggested determining the histogram for only those pixels, which have high Laplacian magnitude. They argue that there should be a shoulder of the gray tone intensity on all sides of the boundary. The shoulder should have high interior pixels just next to the interior border of the region; it should not involve those pixels in between regions that cause the histogram valley shallow. It may contain some number of pixels from the foreground and from the background. This makes two histogram regions about the same size. Thus, the valley seeking method for threshold selection has a higher chance to work suitably on the new histogram. Weszka and Rosenfeld [21] presented a method for segmenting white blobs in dark background based on busyness of pixels. Busyness is the percentage of pixels having a neighbor whose thresholded value is different from their own thresholded value. A good threshold is that point in a histogram valley, which lies between two peaks that minimizes the busyness. Watanabe [20] suggested choosing a threshold that maximizes the sum of gradients taken over all pixels whose gray level equals the threshold. Kohler suggested a modification of the Watanabe's approach, i.e., instead of choosing a threshold, which maximizes the sum of gradient magnitudes, he suggested choosing the threshold, so that more high contrast edges and fewer low contrast edges are selected. Kohler defined the set  $E(T)$  of edges

detected by a threshold  $T$  to be the set of all pairs of neighboring pixels, one of whose gray tone intensity is less than or equal to  $T$  and other greater than  $T$ ;

$E(T) = \{((i, j), (k, l)) \mid \text{pixels } (i, j) \text{ and } (k, l) \text{ are neighbors, and}$

$$\min \{I(i, j), I(k, l)\} \leq T < \max \{I(i, j), I(k, l)\}\}.$$

The total contrast  $C(T)$  of edges detected by threshold  $T$  is given by

$$C(T) = \sum_{((i, j), (k, l)) \in E(T)} \min\{|I(i, j) - T|, |I(k, l) - T|\}$$

The average contrast of all edges detected by threshold  $T$  is then given by  $C(T)/\#E(T)$ . Value of  $T$  that maximizes  $C(T)/\#E(T)$  is chosen as the best threshold.

Milgram and Herman [22] argue that pixels, which are in between regions probably have in-between gray tone intensities. If it is these pixels, which are the cause of the shallow valleys, then it should be possible to eliminate their effect by only considering pixels having small gradients. They take this idea further and suggest that by examining clusters in the 2-dimensional measurement space consisting of gray tone intensity and gradient magnitude it is even possible to determine multiple thresholds when more than one kind of object is present.

Panda and Rosenfeld [19] suggested a related approach for segmenting a white blob against a dark background. Consider the histogram of gray levels for all pixels, which have small gradients. If a pixel has a small gradient, then it is not likely for it to be an edge. If it is not an edge, then it is either a dark background pixel or a bright blob pixel. Hence, the histogram of all pixels having small gradients will be bimodal and for pixels with small gradients, the valley between the two modes of the histogram is an appropriate threshold point. Next consider the histogram of gray levels for all pixels,

which have high gradients. If a pixel has a high gradient, then it is likely for it to be an edge. If it is an edge separating a bright blob against a dark background and if the separating boundary is not sharp but somewhat diffuse, then the histogram will be unimodal, the mean being a good threshold separating the dark background pixels from the bright blob pixels. Thus Panda and Rosenfeld suggested determining two thresholds; one for low gradient pixels and one for high gradient pixels. By this means they performed the clustering measurement space consisting of gray tone intensity and gradient.

Paul L. Rosin [10] proposed an algorithm for thresholding unimodal distribution. He proposed a bi-level thresholding algorithm. The paper assumed that there is one dominant population in the image that produces one main peak located at the lower end of the histogram relative to the secondary population. This latter class may or may not produce a discernible peak, but needs to be reasonably separated from the large peak to avoid being swamped by it. Figure 2.4 shows the approach:

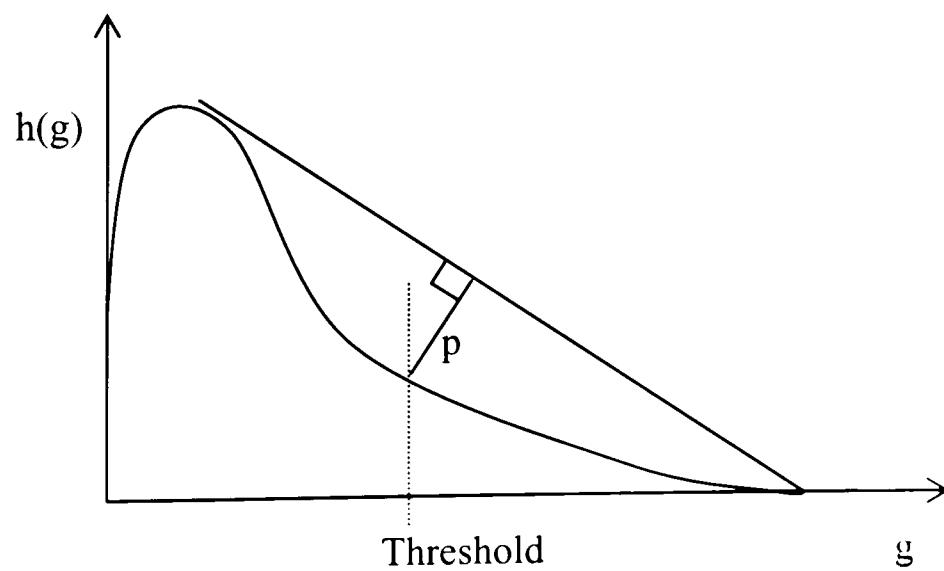


Figure 2.4 Unimodal Thresholding

### 2.2.2 Boundary Based Segmentation

Boundary or edge based segmentation techniques assume that the pixel change sharply at the boundary of a region. Much of the research on edge detection has been devoted to the development of optimal edge detectors, which provide the best trade-off between the detection and localization performance [3]. A common strategy in designing such operators is to find the filter, which optimizes the performance with respect to the three criteria: good detection, good localization, and unique response to a single edge. Canny showed that the optimal detector can be approximated by the first derivative of Gaussian. By convolving the image with this filter, the problem of edge detection is equivalent to finding the maxima in gradient magnitude of a Gaussian-smoothed image in the appropriate direction.

In [13], boundary based technique edges are detected and then grouped into contours/surfaces that represent boundaries of image objects. Most implementations use a differentiation filter in order to approximate the first order image gradient or image Laplacian. Then the candidate edges are extracted by thresholding the gradient or Laplacian magnitude. During the edge grouping stage, the detected edge pixels are grouped in order to form continuous, one-pixel wide contours.

### 2.2.3 Region Based Segmentation

In this approach, regions consisting of adjoining pixels that satisfy certain predefined homogeneity criteria are detected. In region-growing or merging approach, the input image is first tessellated into a set of homogeneous primitive regions, and then by

using an iterative merging process, similar neighboring regions are merged according to a decision rule. In splitting techniques, the entire image is initially considered as one rectangular region, and each heterogeneous region is divided into four rectangular segments; this process repeats until all regions are homogeneous [13].

#### 2.2.4 Hybrid Technique

This is a combination of boundary and region growing techniques. In this technique the image is first divided into different regions and then it is refined using some edge based segmentation method. Region-based techniques often fail to yield the desired structure due to the difficulty of choosing a reasonable starting “seed” point, an appropriate growing rule or stopping rule. Similarly, edge-based techniques may fail due to the similarity of objects within the scene or indistinct boundaries between image objects. In [31], the authors combined region growing and edge detection for magnetic resonance (MR) brain image segmentation. They start with a simple region growing algorithm which produces an over segmented image, then they applied a sophisticated region merging method which is capable of handling complex structures. Edge information is then integrated to verify and correct the region boundaries.

## 2.3 Application of Image Segmentation

- Tracking targets

Image segmentation is an important task performed during target tracking. In [12], Parry, Marshall and Markham used a histogram-based optimization method and identify regions to track target.

- Image compression

Image compression is another major application of image segmentation. Segmentation helps retrieving contextual information for object identification. It leads to compression by providing object based encoding of image [36].

- Obstacle detection

Autonomous system like robotics and automatic transportation segment floors, analyze obstacles and interpret scene.

- Remote sensed image segmentation for oil spill detection, cartography and crop monitoring.

Synthetic Aperture Radar (SAR) images are widely used for surveillance purpose. They perform better than visible or infrared images, because of their capability of acquiring images even in cloudy and dark situations are used for surveillance purposes. SAR images are helpful in detecting oil spills and crop monitoring. These images must be segmented before analyzing the regions of interest.

- Computer vision

Object recognition and scene interpretation and also used in medicine for brain image segmentation and tissue classification.

## CHAPTER III

### INFRARED IMAGE SEGMENTATION

The goal of image segmentation depends on the type image and its application. Segmentation is a complex image-processing task. A generic segmentation algorithm might not produce good results for a given application, although the same is known to provide good results for some other applications. In the absence of any preset criterion, image segmentation uses a simple homogeneity criteria and partitions the image into “puzzle pieces” so that content of each piece is uniformly the same in appearance. Several pieces together may form an object of interest. In this chapter, we discuss first segmentation of colored images, and then present its application to infrared images. Our first goal is to identify objects of interest in infrared images.

#### 3.1 A Region-Growing Segmentation Algorithm

We summarize here the algorithm of [24], which we selected to show how a general segmentation algorithm performs on infrared images. In [24], image domain is mapped into feature space. Image domain is basically described by pixel values, i.e., RGB value and pixel position as stored in a standard digital image format. We choose image files in ppm and pgm standards. Human visual perception is not as simple as the one represented in standard digital image, and RGB description is not suitable for segmentation. Different image representation models have been developed, which provide better representative of human perception of an image than RGB. One of them is

YUV, where Y denotes the luminance (brightness) component and U and V are the chromatic components (chrominance) [35]. Homogeneity in YUV space preserves perceptual quality of image. Therefore, YUV space represents feature space of the image in [24]. As the first step, the image domain information (RGB) is converted to feature space (YUV). This space is divided into random number of search windows and the centers of high-density regions are found by applying mean shift algorithm to each window, i.e. if we randomly select a point and this point lies on the left of some high-density region, then the windowed mean should also lie on the left. A high-density region corresponds to a cluster of objects, having similar features. The mean shift vector, the vector of difference between the local mean and the center of the window, is proportional to the gradient of the probability density at  $x$ , where  $p(x)$  is the density function for feature vector  $x$ . The proportionality factor is reciprocal to  $p(x)$ . The highest density region corresponds to smaller mean shifts and the low-density region corresponds to higher mean shift. The shifts are always in the direction towards the center with zero near to the mode. In this approach, the mean shift algorithm is applied in four steps:

- choose the radius  $r$  of the search window,
- choose the initial location of the window,
- compute the mean shift vector by taking the difference between the local mean and the center of the window and translate the search window by that amount,
- repeat till convergence when we are at the mode of cluster.

All high-density regions are detected in feature space. The tolerance set in the feature space determines the homogeneity criterion. All pixels around a high-density

region, which do not lie within the chosen tolerance, are grouped into separate clusters. This way all high-density regions are detected. For pixels that lie in the same cluster, 8-connected component labeling is applied to find corresponding regions in the image domain. Next, this algorithm validates the extracted centers with some image domain constraints to provide the feature palette. In image domain, checking if enough number of connected pixels is present does validation. If there are enough pixels of similar features, then a color is assigned. This approach thus uses automatic detection of the modes by using mean shift algorithm. Finding the modes corresponding to object or clusters in an image is a challenging task.

By controlling two parameters, this algorithm computes three different kinds of segmentation called under-segmentation, over-segmentation, and quantization; the controlling parameters are: homogeneity and the region size. Homogeneity determines which similar features are treated alike; it is expressed by choosing some tolerance in feature space. For automatic segmentation, value of tolerance should be adaptive, i.e., it should adjust as per image properties. If the visual activity in the image is high, homogeneity should be defined with lower tolerance to preserve the image quality. Activity in an image can be measured by calculating the global covariance matrix of the image. The square root of trace of the global covariance matrix, which is related to the power of the signal, gives a measure of activity in the image. The region size is measured in terms of number of adjoining pixels having similar features. Under-segmentation produces images with lowest resolution. In this case homogeneity is maintained by choosing a large tolerance margin and by retaining most significant colors

as the feature palette. Over-segmentation provides intermediate resolution. Quantization yields the highest resolution and considers all the features as significant.

### 3.1.1 Performance on IR Images

To demonstrate characteristics of this algorithm, we downloaded an implementation provided by the author of [24]. We choose an IR image, however, which perhaps is not best candidate for this algorithm. The results are given in Figure 3.1.

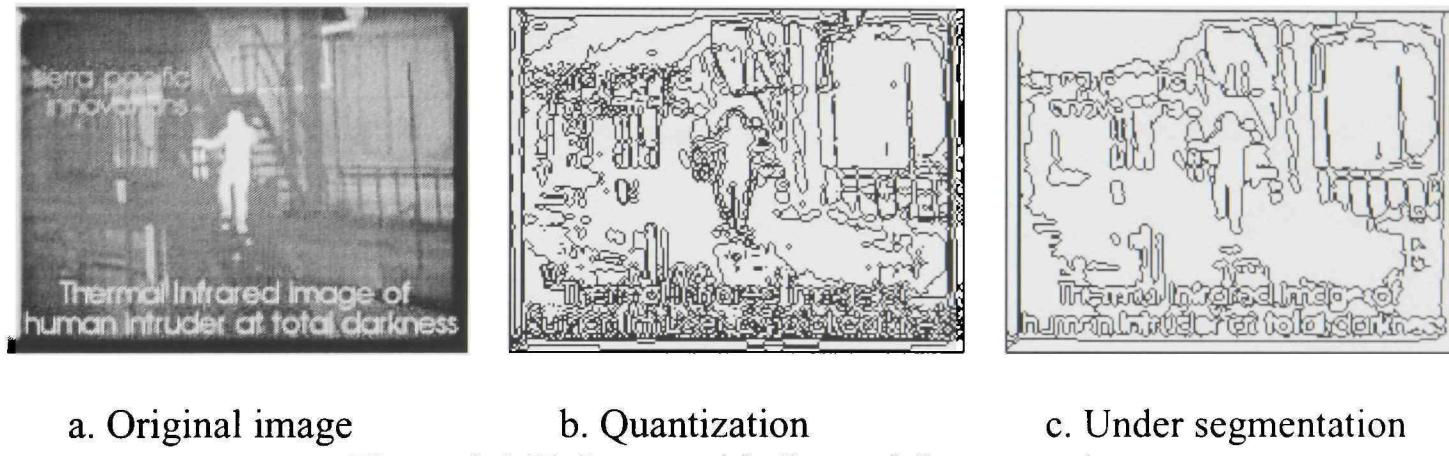


Figure 3.1 IR Image with General Segmentation

From Figure 3.1(c), we can see that even the under-segmented image contains a lot of details of the background; for most applications, this detail is redundant and unnecessary. Figure A.5 shows the result of segmentation on tank image. From which we can observe that the image is highly over-segmented, producing multiple regions for same object. Thus, the method does not help to identify regions of interest directly. Furthermore, this method is computationally expensive, because it calculates the global covariance matrix and the algorithm iterates until it captures all high-density centers for homogeneous region growing. Moreover, since feature space processing is done in YUV

space, the method converts the image first to YUV for segmentation and converts the segmented image back to RGB for reproducing the results. Conversion also takes extra time for computation. Hence, this method may not be applicable for real-time processing such as problem of object tracking.

The above method considers the homogeneity on the basis of color or gray level. Using only this as homogeneity criterion is not suitable for the applications where the object of interest has varying gray levels. Thus, this approach is better suited for low-loss image compression where a lot of details need to be preserved, but it is not suitable for infrared image segmentation. Further, the tolerance to distinguish homogeneous regions is calculated on the basis of the visual activity by calculating global covariance matrix. The noise in infrared image due to varying gray levels within and around the object of interest gives higher measure of visual activity, which leads to over segmentation. Finally, this algorithm is 6-7 times slower than the approach based on thresholding discussed next.

### 3.2 Fuzzy Logic-Based Segmentation

Determining a criterion, which is strong enough to separate two adjoining regions, is not easy, due to fuzziness between them. Due to this reason, fuzzy logic has been applied for image segmentation. Fuzzy sets are used to represent vagueness of region of interest in the image [17]. This approach avoids the uncertainty caused in a conventional approach, which divides the image into meaningful regions by making crisp decisions. In [17], fuzzy rule-based reasoning is applied to obtain a coarse segmentation map. A coarse

segmentation usually generates an over-segmented image. In order to identify the main object, the regions identified by a fine segmentation are merged. Adjoining regions are merged if they share features such as colors, edge strength of the common boundary and the spatial proximity.

### 3.2.1 Concept of Fuzziness

Fuzzy sets deal with the type of uncertainty that arises when class of objects are not sharply defined. Uncertainties such as the outcome of an experiment of rolling a pair of dice are randomness. Outcomes of rolling dice can therefore be observed by random variables, which follow some distribution laws. For many problems, these laws may not be known a priori. Furthermore, probability of an event is distinct from degrees of membership to a fuzzy set. Probability logic is used in situations, where events of interest are either true or false, but the information available is incomplete and prevents such a determination. Given an arbitrary set  $X$ , a fuzzy set (on  $X$ ) is a function from  $X$  to the unit interval  $I=[0,1]$ ,

$$\mu : X \rightarrow I \quad (3.1)$$

The function helps in describing subsets of  $X$ . In many applications, a subset of  $X$ , say  $A$ , is described by giving a supplementary property  $P$ , satisfied by the points of  $A$ , i.e.,

$$A = \{x \in X \mid x \propto P\} , \quad (3.2)$$

where  $x \propto P$  stands for “ $x$  fulfills the property  $P$ .” If  $P$  is well formulated, determination of subset  $A$  is not difficult. But in many problems whether a point say  $x$  satisfies the given property may not be deterministic. In this case, we assign  $x$  a value between 0 and

1; a higher value close to 1 indicates that  $x$  strongly satisfies the property  $P$ . For example a question, “Do you feel young?”, could be described by using a membership function. Fuzzy logic helps to quantify vague questions.

### 3.2.2 Using Fuzzy Logic for Segmentation of Infrared Images

We report next the approach proposed in [6] for target detection in FLIR (forward looking infrared images), which employs fuzzy logic to identify region of interest. The objective is to determine regions (sets of pixels), which have similar gray pixels and are not far from the center. This approach uses fuzzy logic to measure similarity and adjacency of pixels in a region. It assumes that these measures are sufficiently different for pixels belonging to the target.

Let  $X$  denotes an  $MXN$  pixel image with  $L$  gray levels, and let  $g(x,y)$  denote the gray level of pixel  $(x, y)$ . The general definition of a fuzzy subset given in equation 3.2 can be defined for image as

$$X' = \{[g, \mu_{x'}(g)] \mid g \in X\}, \quad (3.3)$$

where

$$\mu_{x'} : X \rightarrow [0,1]$$

$\mu_{x'}$  is the membership function of  $X'$  and it can be represented by a scale from 0 to 1 according to some property of pixels. This property is mathematically represented in [6] as the function of similarity and adjacency. In the following, we describe how the concept of fuzziness is applied to infrared images in [6].

Consider Figure 3.2, it represents an IR image with  $R_r$  representing an object of interest. Let the bound on the largest target object be known and let this bound be not

more than (MaxWidth . MaxHeight) and not less than (MinWidth . MinHeight). This information can be used to determine a suitable threshold for segmentation of the image. A global thresholding is done on the histogram of the whole image by selecting a threshold in such a way that the total number of pixels beyond the threshold in the histogram is just greater than the minimum size of the object. The threshold satisfies the following:

$$\text{Threshold} = \max t \text{ such that } \sum_{g=0}^t h(g) \leq (M * N - \text{MinWidth} * \text{MinHeight}) \quad (3.4)$$

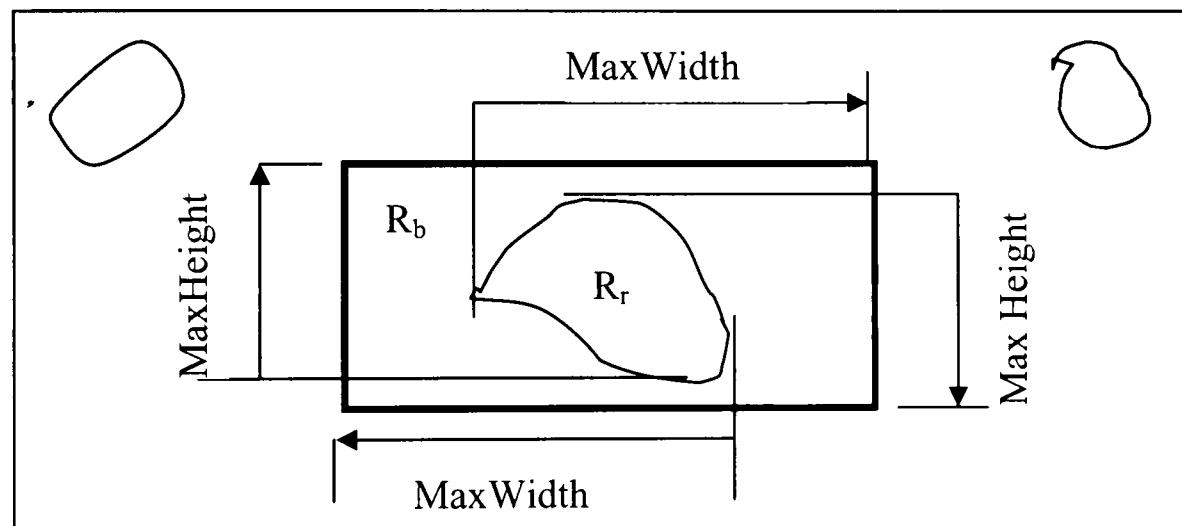


Figure 3.2 Image with ROI and Reference Region

To segment the image, initially the pixels are divided into two classes; one belonging to background (i.e., gray level is below threshold) and the other belonging to foreground whose gray level is above threshold. Connectivity component labeling is applied to the pixels above the threshold. All connected foreground pixels are put together to form a region. Connectivity of a pixel is obtained by testing any of its eight

neighbor is also member of foreground; if so, it belongs to the same region. This is recursively applied to each pixel in the foreground, until all the foreground pixels are considered. The largest region obtained after labeling is chosen as to contain the target. This region is called reference region  $R_r$  as shown in Figure 3.2.  $R_r$  is usually not the whole object and may denote visible portions of the object. In infrared image, objects have varying gray levels; therefore a simple thresholding may prune portions of the object. Without reconsidering these thresholded portions, the object cannot be identified properly. The thresholded portions of the object are adjacent to the reference region. Therefore, we refine the region of interest (ROI) that surrounds the object. ROI is found by inflating the reference region as shown in Figure 3.2, along both horizontal and vertical directions, so that it can contain the maximum possible size of the object. The bold rectangle in Figure 3.2 represents ROI. Region  $R_b$  is the region in ROI except  $R_r$ . To determine any adjoining part of object, [6] defines a similarity function as follows.

$$S(g, g_r) = 1 - |g - g_r| / C_1 \quad (3.5)$$

$$C_1 = g_r - g_{\min}$$

Where  $g_r$  is the average gray level in the reference region and  $g_{\min}$  is the minimum gray level in the whole image. Thus the values of  $S(g, g_r)$  lies between 0 and 1. To calculate the adjacency, we first determine the centroid of the reference region.

Let point  $(x', y')$  in the image denote the centroid of the reference region. Then

$$A[(x, y), (x', y')] = 1 / (1 + \max(|x-x'|, |y-y'|)) \quad (3.6)$$

where  $A[(x, y), (x', y')]$  is a measurement of adjacency between two points  $(x, y)$  and  $(x', y')$ . The following function is a weighted sum of adjacency and similarity between two points.

$$\mu_{x'}[g(x, y)] = \beta * S(g, g_r) + (1 - \beta) * A[(x, y), (x', y')] \quad (3.7)$$

where  $\beta$  is a constant. The higher value of  $\beta$  gives more emphasis on the gray level information of the object than the spatial distance. Since both similarity and adjacency values are between 0 and 1 their weighted sum or the membership value ( $\mu$ ) is also between 0 and 1. The membership values are scaled from [0,1] to integer values in the range 0-100. Let  $\mu'$  represents the scaled membership values, it is computed for all the pixels in the region ROI. Then we determine histogram of membership for each regions  $R_b$ ,  $R_r$  and ROI separately. Let  $h_r$ ,  $h_b$  and  $h$  be the membership histogram for regions  $R_r$ ,  $R_b$ , and ROI respectively. Let  $m_r$  and  $m_b$  denote the maxima in the histograms  $h_r$  and  $h_b$ , respectively. These values of  $m_r$  and  $m_b$  represent the dominant membership values in the reference and surrounding background region  $R_b$ , respectively. Then we determine a valley between  $m_b$  and  $m_r$  in the overall histogram  $h$  of the whole region ROI, i.e.,

$$\begin{aligned} t_f &= \min h(\mu') \\ m_b &\leq \mu' \leq m_r \end{aligned} \quad (3.8)$$

All the pixels having  $\mu'$  greater than  $t_f$  are considered to satisfy the property and hence taken to be part of the object.

Next we demonstrate the algorithm using an example of the tank image shown in Figure 2.1. Figure 3.3 shows the result after applying thresholding and labeling. Labeling finds the connected pixels. With the help of known minimum object size, the threshold is

calculated to be 176. The rectangular box around the image in Figure 3.3 represents ROI. Figure 3.4 and Figure 3.5 shows the histogram of integer membership values for region  $R_b$  and  $R_r$ , respectively. Membership histogram of the all the pixels in ROI is shown in Figure 3.6. From each histograms for region  $R_b$  and  $R_r$ , we obtain the membership value corresponding to highest bins as  $m_b$  and  $m_r$  respectively. Now we apply equation (3.8) to the overall histogram in Figure 3.6 to obtain the decision threshold, which decides whether a pixel having certain membership value belongs to object or the background.



Figure 3.3 Tank Image Binarized and Labeled at Threshold 176

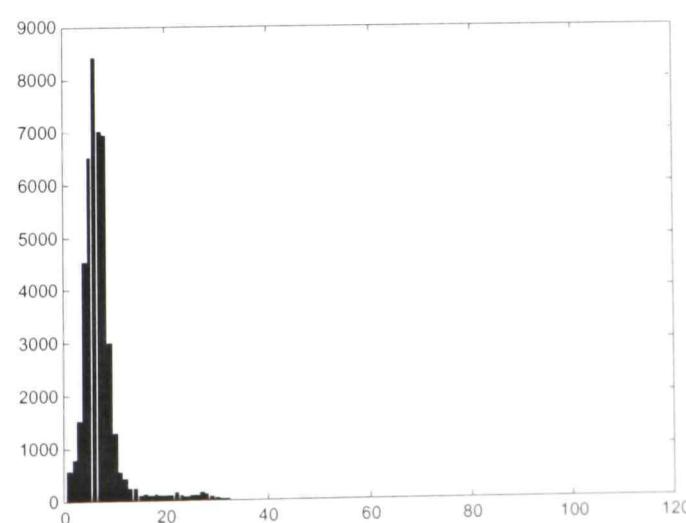


Figure 3.4 Membership Histogram for Region  $R_b$  in ROI.

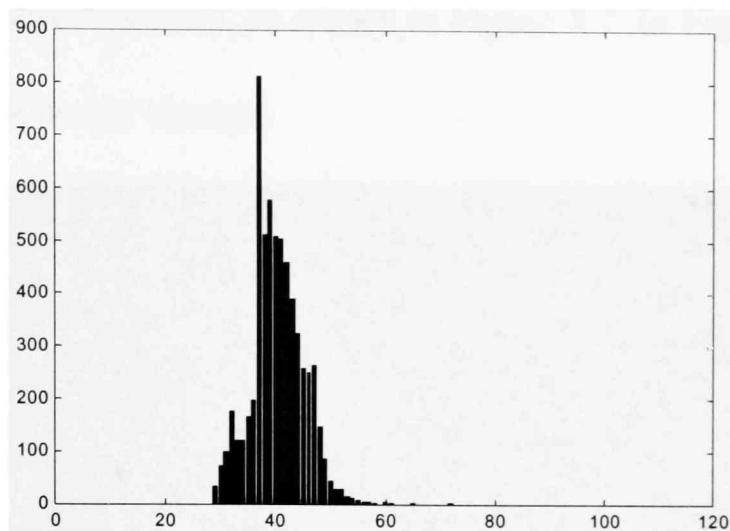


Figure 3.5 Membership Histogram for Region  $R_f$  in ROI.

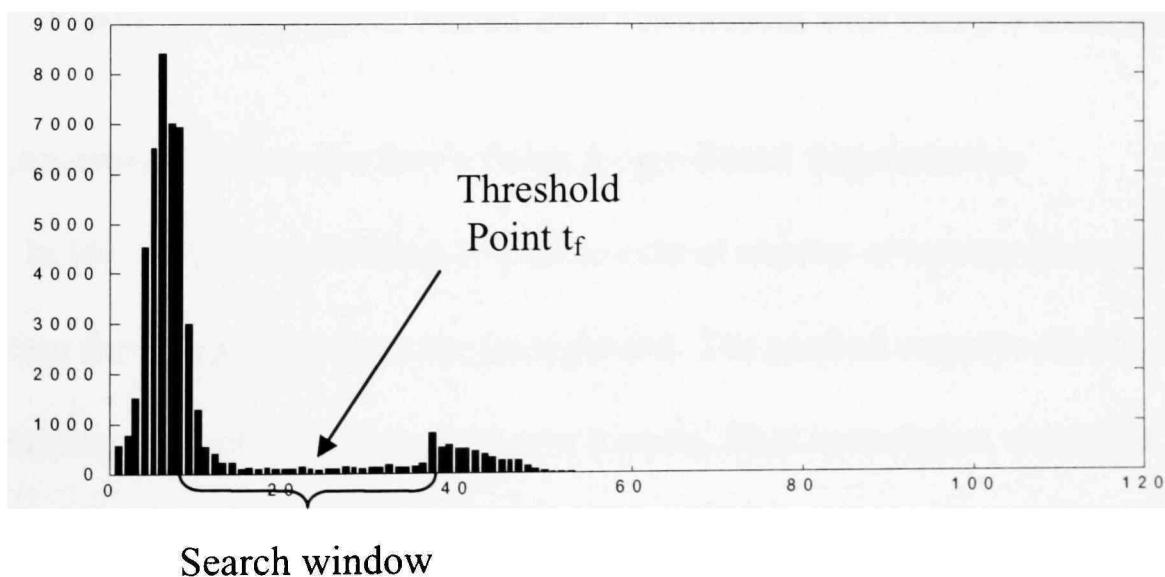


Figure 3.6 Membership Histogram for Region ROI.

From Figure 3.3, we can see that thresholding and connected component labeling only cannot precisely segment the object, and use of fuzzy logic is able to determine adjacent pixels as object portion, as shown in Figure 3.7. In Figure 3.7, the region outside the ROI is left without any change.

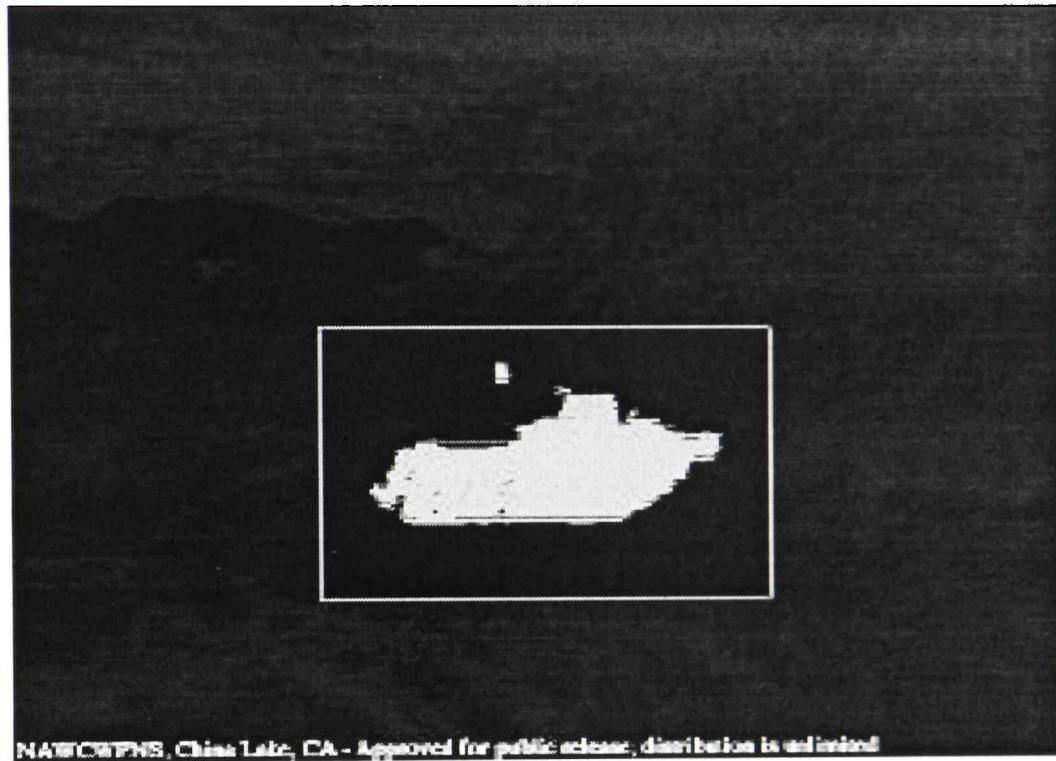


Figure 3.7 Segmented Result after Refinement with Fuzzy Thresholding

### 3.2.3 Limitations of Sun-Gu Sun's Fuzzy Logic-Based Segmentation

In [6], fuzzy thresholding is used to extract objects of known dimensions, by separating the foreground from the background. The method requires that the maximum and minimum size of the object is known a priori. This assumption was made on the basis that the size of the tank could be estimated from the knowledge of distance between the tank and the camera onboard. The minimum size of the object is used for global thresholding and to locate where in the image the object of interest. The maximum size of the object is used for identifying ROI. Once the ROI is identified, a fuzzy logic based

refinement is performed in ROI. They used this refined object to further identify the edges of the tank. They also assume that there exists sufficient contrast between the foreground and background.

We applied the above method to general surveillance images, which may contain multiple objects. We assume that no knowledge about types, shape, distance or dimensions of object in the scene is given. However we do assume that there exist sufficient contrast between the foreground and background. Since the size of the object is not known both global thresholding and ROI identification cannot be performed as formulated in [6]. Thus we have two major problems, one is to determine global threshold and the other is to find the ROI around each object in the image.

To avoid these limitations, we first discuss some other global thresholding methods for infrared images, and then adapt one of them to expand [6]. In section 3.7, we present the results and compare the expanded method with [6].

### 3.3 Fujimura's Approach

Fengliang Xu and Kikuo Fujimura [33] faced the above problem, where they considered identification of multiple pedestrian in night vision images. They used the following to compute a dynamic threshold:

$$D\text{Threshold} = 1/5 * d\text{Mean} + 4/5 * \text{Maxgrey} \quad (3.9)$$

where  $d\text{Mean}$  is the average gray level of the image and  $\text{Maxgrey}$  is the maximum gray-level present in the object.

Our experiments using the above formula to calculate global threshold exhibited that the formula does not compute appropriate threshold for different types of images. The reason is that it computes higher threshold, which if used over segments object in the foreground. Results are presented in section 3.7.

### 3.4 Otsu's Thresholding

Otsu's method selects a threshold from the histogram following some discriminant analysis [26]. In this method, pixels are divided into two classes at the threshold point, which maximizes a discriminant measure. Discriminant function is the measure of separability of the resultant class. This method measures the separability of all the gray levels and finally chooses one that maximizes the separability as the proper threshold. Threshold based in Otsu's method can be used for thresholding bi-modal histograms.

### 3.5 Adaptive Global Thresholding

One of the limitations discussed in section 3.2.3 is the problem of determining a proper threshold to process the histogram without any prior knowledge of the size of the main object in the image. Therefore, Sun-Gu Sun's approach is inapplicable for most surveillance images. To solve this problem, we propose an alternative approach to compute threshold on the basis of histogram modes. A number of approaches have been purposed for histogram-based thresholding [8, 10, 26, 33, 34]. Most of these approaches are designed for photographic images. For color images, each mode of the histogram

corresponds basically to a different color and therefore it very likely identifies a different object. Histogram of an infrared image may also contain a number of modes with different gray levels indicating the presence of complex background and noises. For the purpose of object extraction, we consider the mode of the highest gray level only. Because of the way the infrared images are acquired, they vary widely. Infrared cameras capture the object by virtue of temperature of object, and therefore an object may contain various gray levels within it. Therefore, the main objective of the adaptive thresholding is to threshold all the complex background modes and preserve the mode representing the main object. Figures given next show various possibilities of histogram distributions. The purpose is how to determine suitable gray level for thresholding.

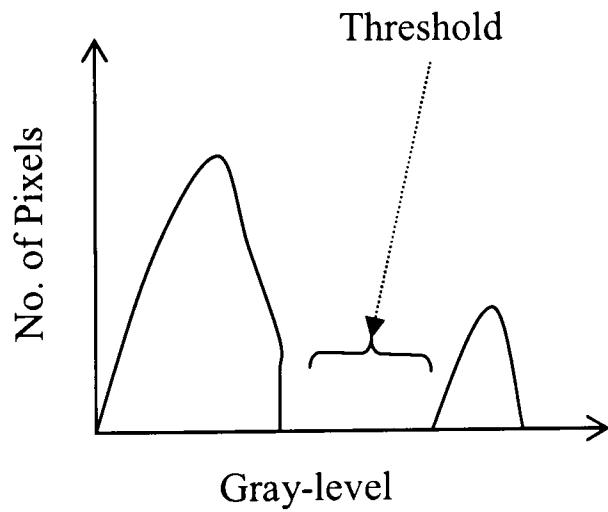


Figure 3.8 Bi-Modal Histogram with High Contrast

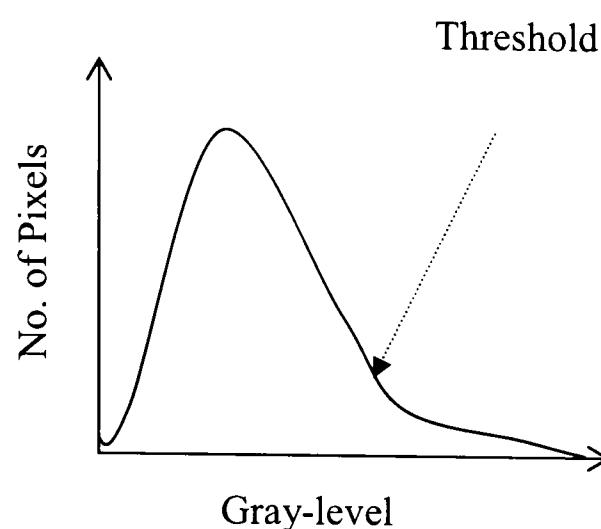


Figure 3.9 Unimodal Small Size Object

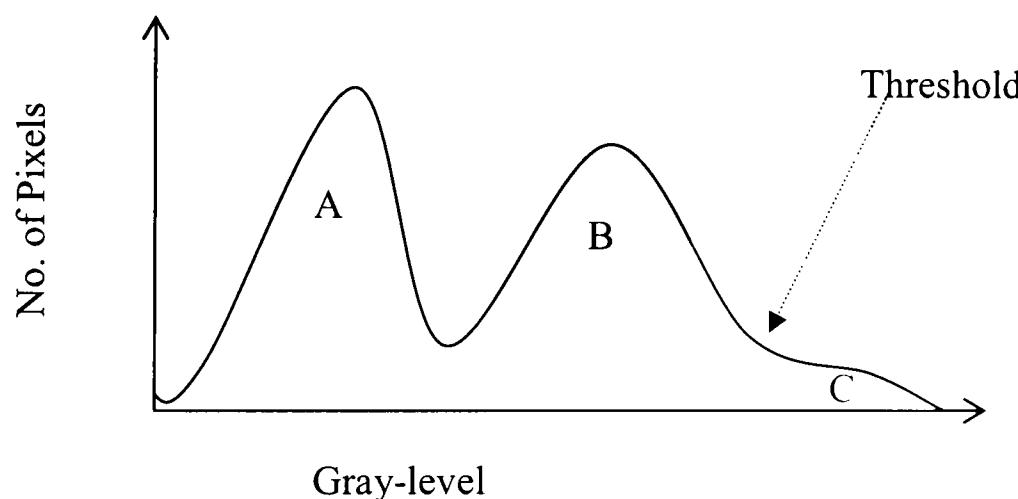


Figure 3.10 Multimode Histogram with Complex Background and Small Object

The mode of interest always lie on the rightmost part of the histogram, because it pertains to brighter areas of the image, which should correspond to radiating objects in an IR image. We should always try to separate this mode from other modes regardless of its size. If we select any gray level on the brighter side as the threshold, it might not work

suitably because the gray level assignment depends entirely on the relative temperature (radiation) difference between the foreground and background. Hence, it is necessary to determine a cluster (mode) in order to be able to separate foreground from background.

Figure 3.8 shows a bimodal histogram. Otsu's [26] approach works well here to separate foreground from background. The histogram in Figure 3.9 is unimodal. The falling tail shows that the foreground is small. Therefore, the foreground or the object does not have distinct mode, instead it is swamped by the background mode. In this case, selection of threshold is more difficult, but a gray level pointed at by the arrow may be tried as the threshold. Thus, if histogram is either bimodal or unimodal, then we can threshold the image by carefully selecting a gray level. But if there are three or more modes as shown in Figure 3.10, we can determine the number and position of the modes and then discard all background such as (mode A) from consideration. The rest of the histogram now contains mode B and C as shown in Figure 3.10, where mode C is mixed with B and mode C is not distinct. A major portion of C is swamped by mode B. To recover these portions of the mode C, we can treat the histogram as unimodal after discarding mode A, and apply unimodal thresholding on B to obtain suitable threshold. On the other hand, if B and C are clearly separable, we can threshold at gray level between the modes B and C. The simplest way of detecting the modes is to locate all maxima and minima in the histogram.

### 3.6 Our Implementation

We will first discuss the overall implementation for multiple regions processing in section 3.6.1. In section 3.6.2, we present the implementation of our approach of adaptive global thresholding. We also implemented the global thresholding approach described in sections 3.3 and 3.4 for comparison of our results.

#### 3.6.1 Implementation for Multiple Object Refinement

We implemented the algorithm of [6] and then enhanced it to address problems discussed in section 3.2.3. Our program first loads the image, computes its histogram and then determines an appropriate global threshold without any user assistance. The threshold divides the image into foreground and background pixels. Next, we apply connected component labeling on the foreground pixels to find connected regions. From the list of regions developed, those containing less than 10 pixels are discarded, assuming that the connected regions of very small size represent most probably noise in the image. The regions are then ranked on the basis of average gray level and their size. Since pixels in the same region can have different gray levels greater than threshold, we calculate the average threshold of each region. Since we consider an image to have multiple objects of interest, we process all regions in the region list to get the best result. But not all the regions in the list correspond to an object of interest. We do not know if a criterion exists to determine the region corresponding to every object. Also, due to the randomness of the connected component labeling method, the region list can contain regions in any order.

We developed a simple heuristic to rank the regions on the basis of the information that the objects in infrared image have higher gray levels and are considerable in size.

$$\text{Weight of region} = 0.75 * \text{average gray level} + 0.25 * \text{size of region} \quad (3.10)$$

Using single feature, say gray level alone, may lead us to select the small noisy regions with bright pixels instead of the regions corresponding to the objects for processing. If we consider only size for ranking regions, we might discard the regions that are very bright in the image. Since in an infrared image, objects of interest are usually very bright, brighter regions are given higher priority. This ranking helps us to process generally the regions that correspond to objects. Our implementation also provides an option to select the maximum number of objects that are expected in the scene. If we know the maximum of the numbers of regions, we can discard the regions from further processing on the basis of their rank. Next we apply the fuzzy logic based refinement of [6] recursively to each region in order of its rank, until all the regions are processed. The refinement process is described in section 3.2.2. It requires knowledge of the maximum size of the object to identify any ROI. We used the following approach for finding the ROI for each region.

- i. Find the all-extreme points of the region in consideration, to determine its width along x-axis and the maximum height along y-axis (see Figure 3.2).
- ii. Next calculate a rectangle bounding the current region. This rectangle is extended by 10% in all four sides, right, left, top and bottom. Thus we insure that the ROI completely surrounds the reference region. Extending the region by more than

10% might interfere with other nearby objects. This extension percentage is chosen at random.

Thus finding ROI in this way requires no knowledge of the object size. For simplicity, our implementation does not process regions whose ROI overlap (we assume that the objects in the image are located sufficient apart). For each region in the foreground, we apply the fuzzy logic based refinement and no processing is done to the background.

### 3.6.2 Adaptive Global Thresholding With Maxima-Based Mode Detection

Sun-Gu Sun's approach [6] assumes that size of the object is known; it uses this information to separate background from the foreground. In absence of any such knowledge, we need an adaptive method of thresholding for images that might contain number of objects of interest. In our method, we calculate the threshold by analyzing the histogram. This approach uses simplified bi-modal and unimodal thresholding with maxima based mode detection. We make following assumption for our implementation:

- i. All objects of interest are brighter. They occupy a small portion of the image compared to background.
- ii. The histogram might contain multiple modes, but we consider only three major modes. Thus the multiple mixed modes representing the background is considered as a single mode. For the simplicity, we assume that there are at the most two dominant modes as shown in Figure 3.10 representing the background and one for the foreground or it is swamped by the background modes. Figure 3.8 shows a

finely separated bimodal image histogram. In Figure 3.9 and Figure 3.10, the foreground is swamped by the background modes, where objects are smaller.

- iii. For simplicity, we assume that the modes in the histogram follows a Gaussian distribution, where the distribution falls down to the 10% of the maximum when it reaches the base of the mode.

We follow the following steps to determine the modes and threshold. Let,  $g$  denotes the gray level, and  $h(g)$  be the size of  $g^{\text{th}}$  bin.

- i. Find all the local maxima in the histogram.

Usually, before finding the maxima and minima in the histogram, histogram is smoothed to avoid excessive number of local maxima. But in our approach we do some post smoothing by discarding the local maxima that lies in the proximity of the global maxima of the corresponding mode. The list of local maxima consists gray level  $g$  and  $h(g)$ , that satisfies the relation, (except first and last bin)

$$h(g - 1) \leq h(g) \geq h(g + 1). \quad (3.11)$$

- ii. Find the greatest maxima from the list of all maxima.

The global maxima correspond to the main background mode, because in surveillance images, background of image occupies more part of an image. Find the base of this mode by finding the gray level at which the value settles below 10% of the maximum. We check 10 bins to decide if it is settled or not. We also discard any local maximum falling below the gray level corresponding to the base of the mode.

- iii. Determine the largest maxima among the rest of the maxima list. This maximum corresponds to either the foreground mode or to the second background mode.

Assuming that this mode corresponds to the foreground and the histogram is bimodal, we determine the base of this mode in reverse direction i.e. towards lower gray level by checking when it settles below 10%.

- a. If it does not settle on the lower side of the base of the previous mode and the gray level at the point is higher than average gray level, we apply unimodal thresholding. We apply the unimodal thresholding on this mode, which maximizes the distance  $p$  shown in Figure 2.4. This threshold point also separates the foreground from background. To apply unimodal thresholding, we first find the equation of the line from the top of the mode and the maximum gray level satisfying:

$$a * x + b * y + c = 0, \quad (3.12)$$

where  $x$  represents the axis along gray value and  $y$  is  $h(g)$ .

For any point  $(x_1, y_1)$  given by  $(g, h(g))$ , the perpendicular distance from the point is calculated using,

$$p = (a * x_1 + b * y_1 + c) / \sqrt{a^2 + b^2}. \quad (3.13)$$

The gray level corresponding to the point, which maximizes this value, is chosen as the threshold.

- b. If the base of this mode is above the average gray level in the image, the histogram is most likely bi-modal and the mode corresponds to the foreground. Then threshold at the middle of base of the mode found in (ii) and the base of the mode is calculated.

- c. If the base of the mode found in (b) is below the average gray level in the histogram, then this cannot be foreground mode. Because a surveillance image can have a small number of pixels at higher gray levels, the average gray level of the histogram is biased towards the background rather than foreground.
- d. Now we seek another base of the second mode on the brighter side now. We again remove the local maxima within this mode. We have two possibilities. First is that the base of the mode is at the highest gray level, i.e. the mode swamps or eats away the foreground: in this case, we apply the simplified unimodal thresholding on the second mode. Second possibility is that we have a distinct valley between the foreground mode and the second background mode. In this case, we threshold at the middle of these modes.

### 3.7 Experiments and Results

For our experiments, we obtained some sample surveillance infrared images from various resources web sites. These images contain text identifying the sources for public distribution. Hence, we ended up processing images with different histogram characteristics denoting different resolutions and a numbers of objects and regions of interest.

In this section, we present the results of our experiment. We compare our results of global thresholding with the thresholds obtained using Otsu's and Fujimura's methods. We present the results of each method in detail. A few more results are presented in the appendix. Our program takes approximately 0.4 seconds to segment the image. Table 3.1

shows the threshold we obtained using different approaches. Figures 3.11-3.30 show the results with each threshold.

Table 3.1 Global Threshold with Different Approaches

Index	Image	Average gray level	Threshold with different Approaches		
			Fujimura's	Our	Otsu
1	Ship	51	214	137	123
2	Mancycle	82	220	157	74
3	Tank	31	210	147	111
4	Intruder	108	225	160	120
5	Swimer	106	225	145	105
6	Horses	106	225	187	66

Figure 3.11 shows the original image ‘mancycle.pgm’. The image contains multiple bright regions each representing different activity (region of interest). We cannot use Sun-Gu Suns’s [6] approach directly, because we do not know any size information. Figure 3.12 represents the histogram of the image. From this histogram, we see that there are three different modes. The first one is very close to zero and denotes lots of pixels of the background. There is a small mode near gray level of 70 and another major mode near gray level of 120. Since only a small portion of the whole image corresponds to the objects of interest, both of these modes have to be thresholded. In Figure 3.12, when we seek for the base of the mode near gray level 120, we find that the base of this mode is

below the average gray level, if we consider this mode as the foreground mode, we will be considering a large number of pixels. Since we process the modes with the highest maxima first, the mode between the first and the third mode automatically gets discarded, which also belongs to background. Since there is no other discernable mode above the mode near 120, we apply unimodal thresholding to this mode, which lead us to threshold at 157 near the base of the mode. The result of thresholding is shown in Figure 3.13. Figures 3.14 and 3.15 show the results of thresholding using Fujimura's and Otsu's method. Figures 3.16 and 3.17 show the objects and their ROI identified. Figures 3.18 and 3.19 show the final results after applying fuzzy logic based refinement. In Figure 3.19, we see that some portions of the head and the leg of the man are missing because of higher threshold, which leads to a small ROI.

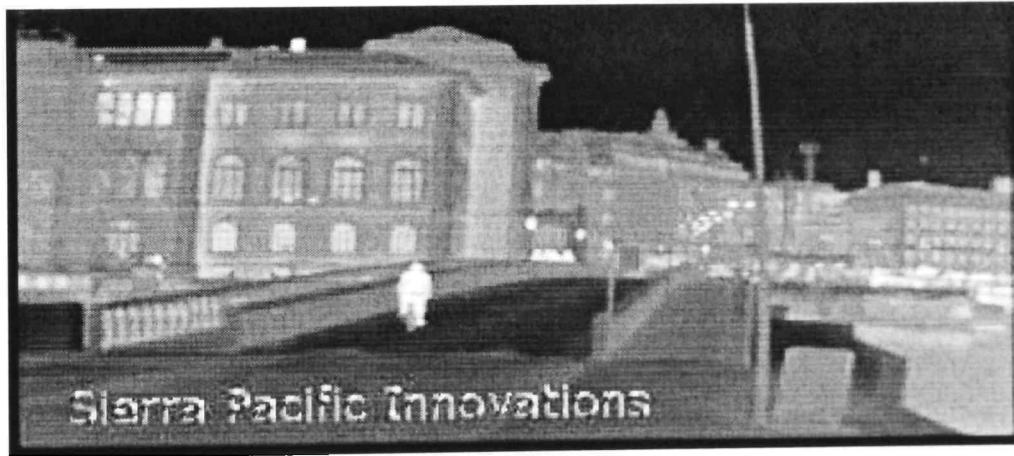


Figure 3.11 Original Infrared Image 'mancycle.pgm'.

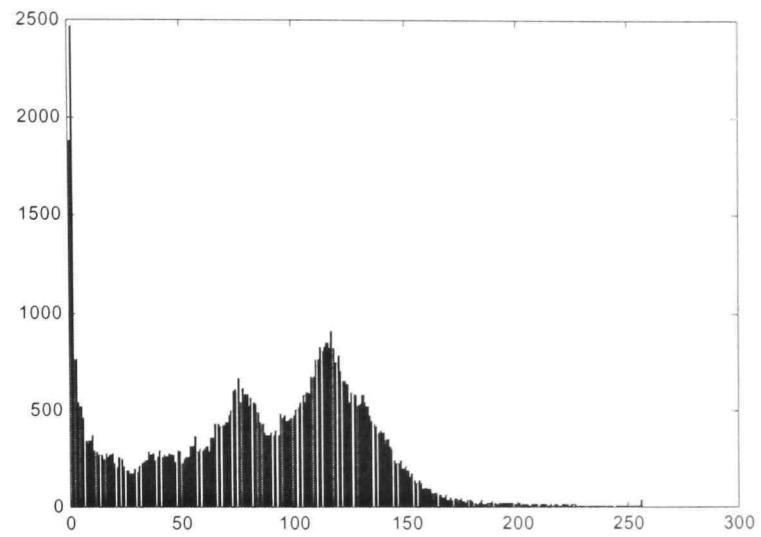


Figure 3.12 Histogram of Image ‘mancycle.pgm’

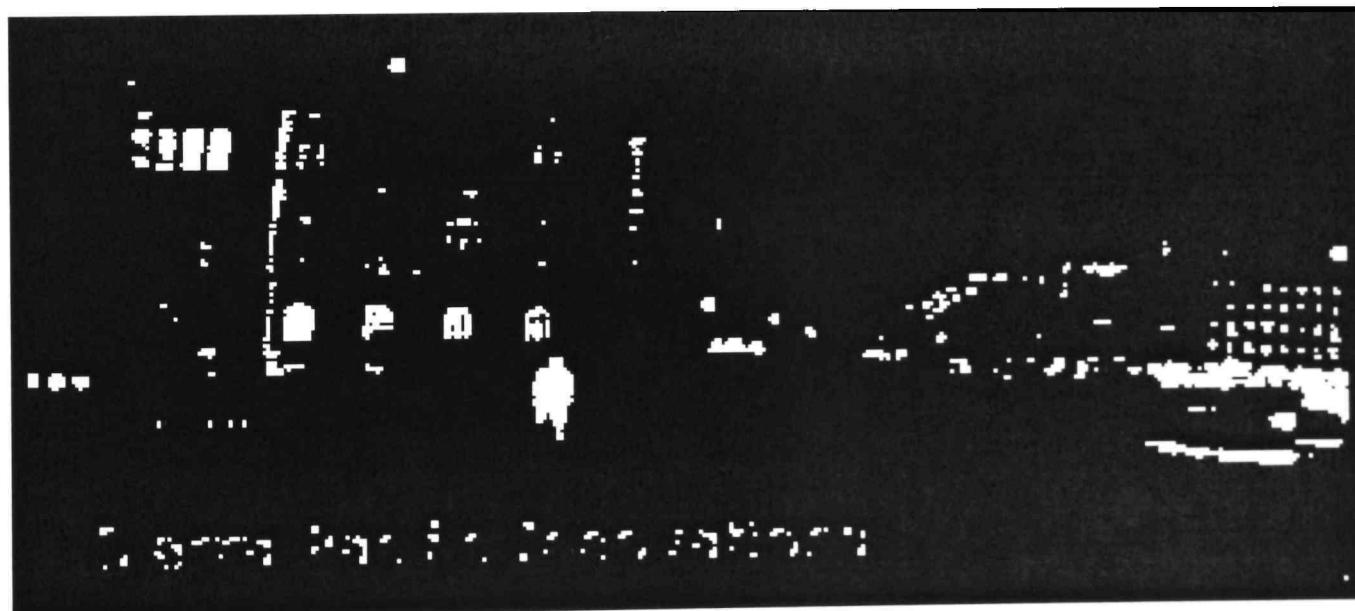


Figure 3.13 Image ‘mancycle.pgm’ Thresholded with Our Approach at 157

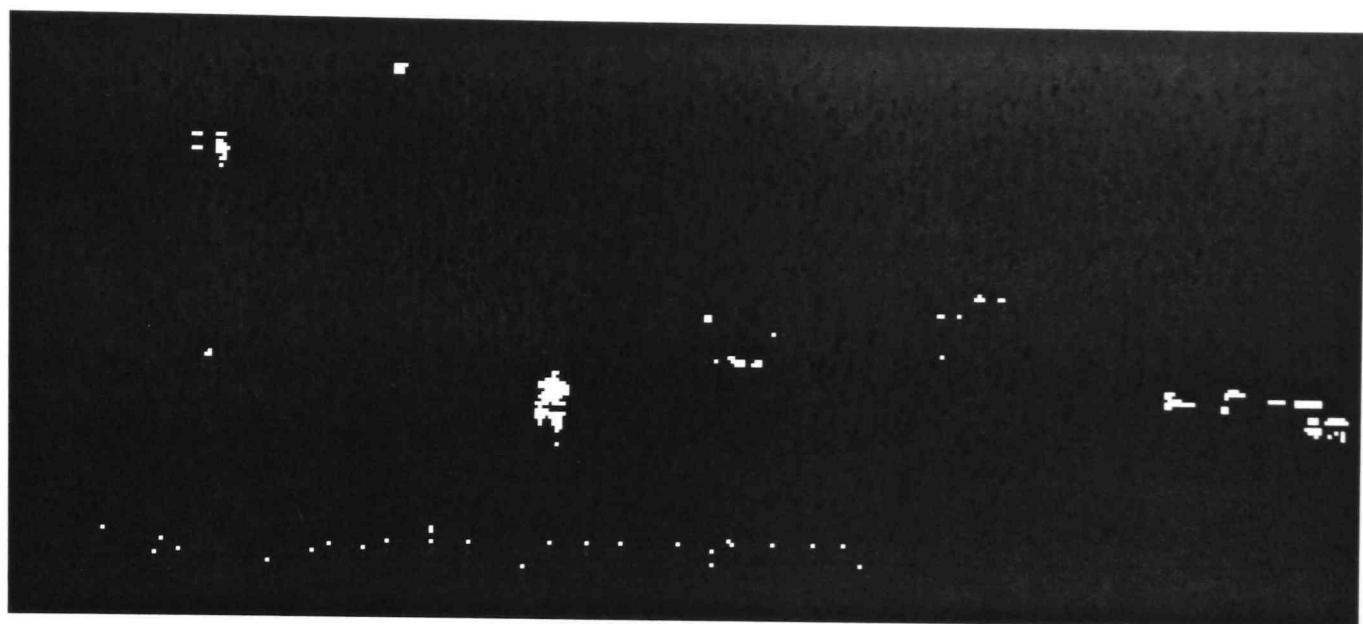


Figure 3.14 Image ‘mancycle.pgm’ Thresholded with Fujimura’s Approach at 220

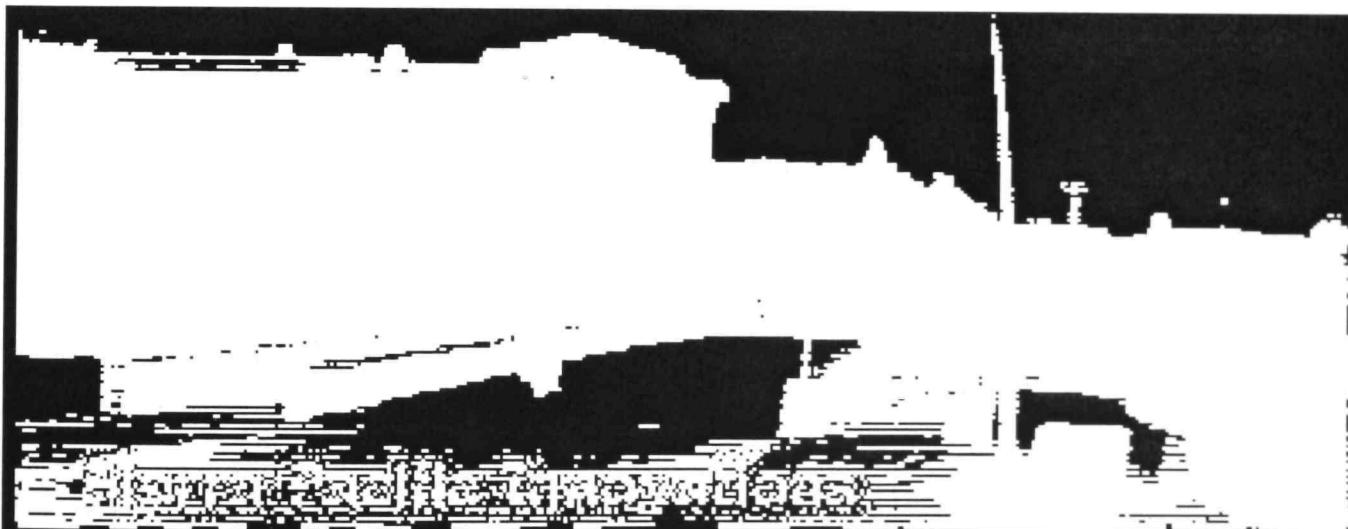


Figure 3.15 Image ‘mancycle.pgm’ Thresholded using Otsu’s Approach at 74

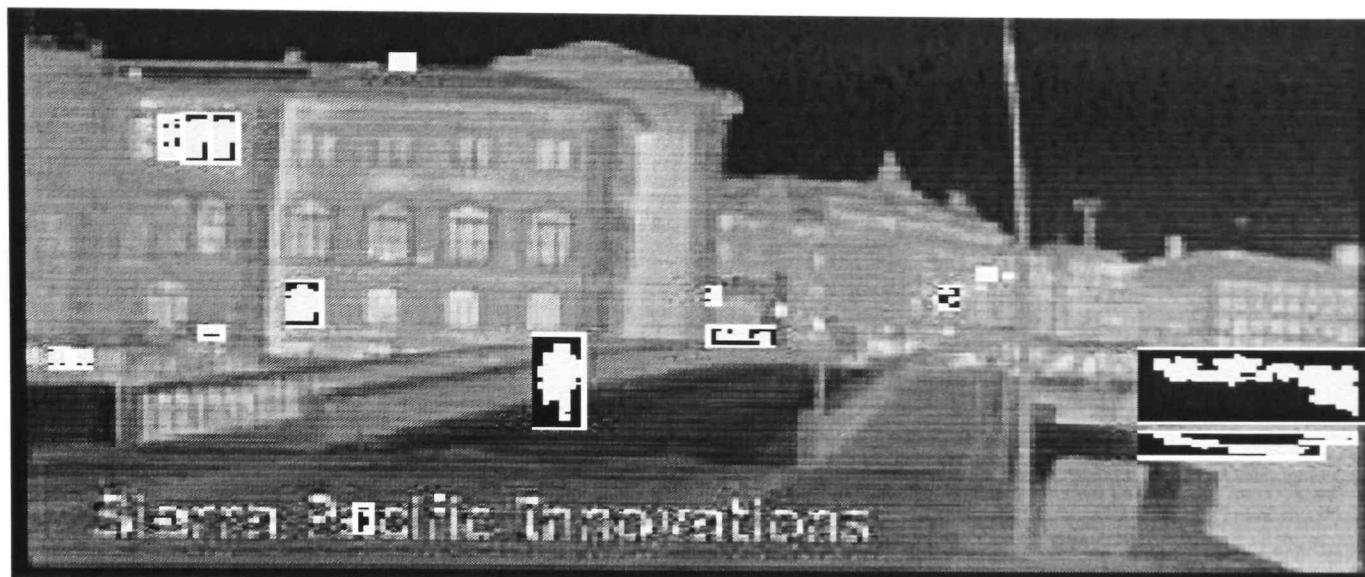


Figure 3.16 Image ‘mancycle.pgm’ before Applying Refinement at 157 Threshold

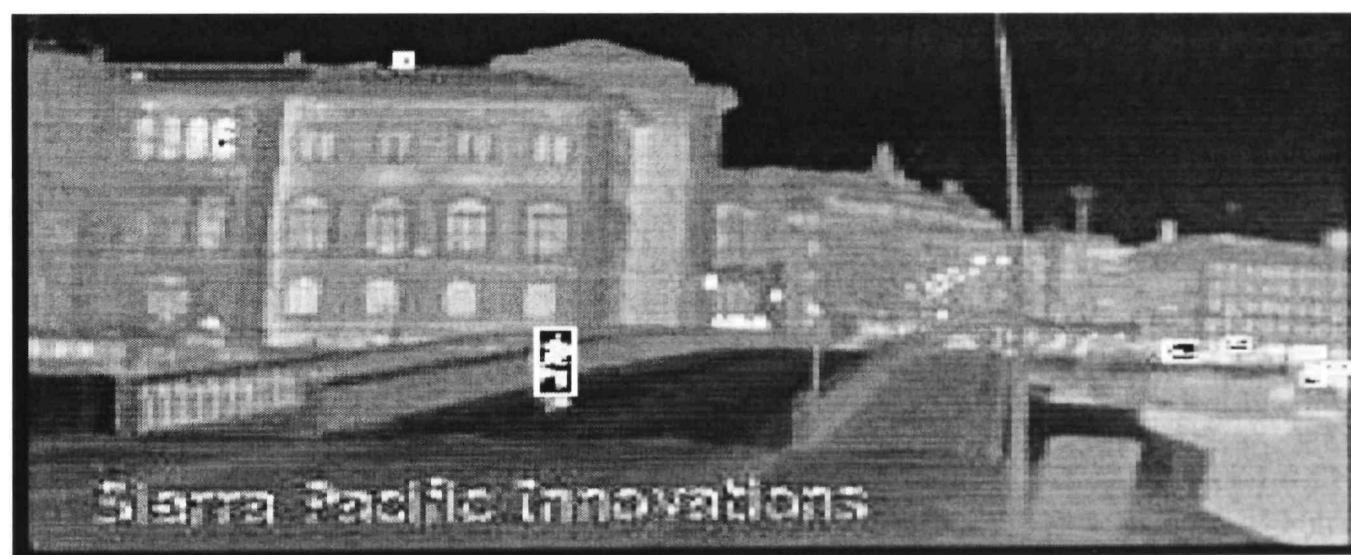


Figure 3.17 Image ‘mancycle.pgm’ before Applying Refinement at 220 Threshold

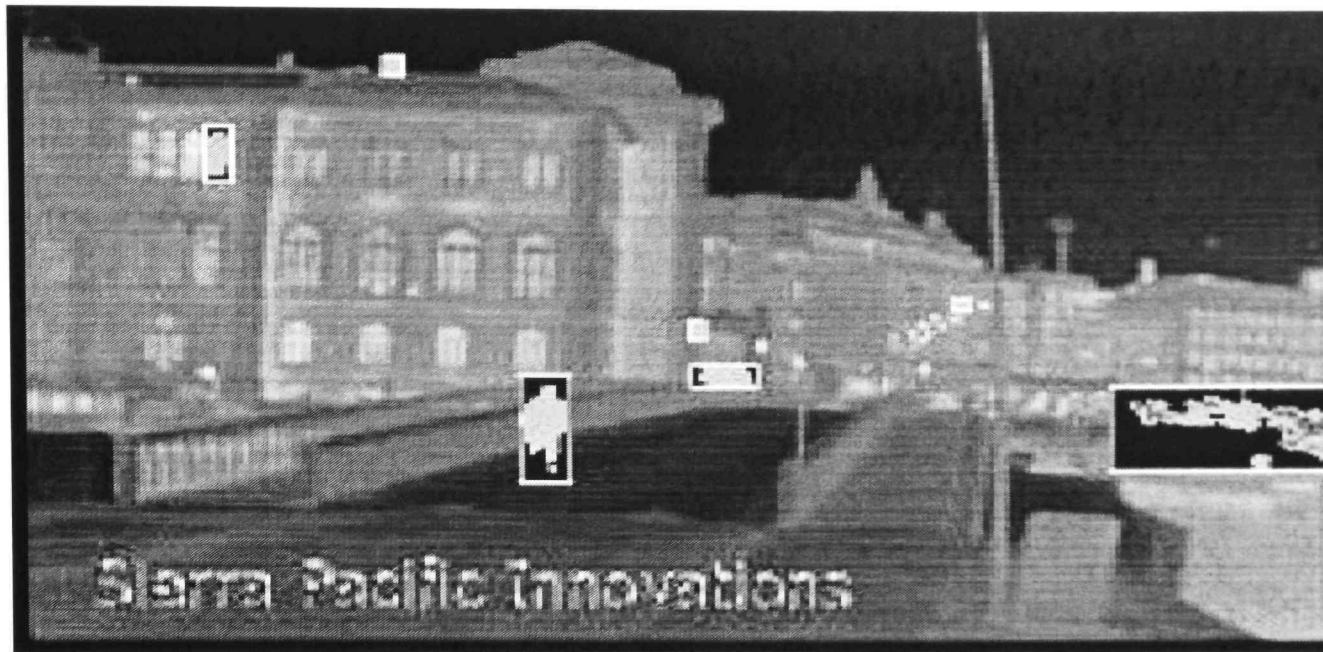


Figure 3.18 Image ‘mancycle.pgm’ after Refinement at 157 Threshold

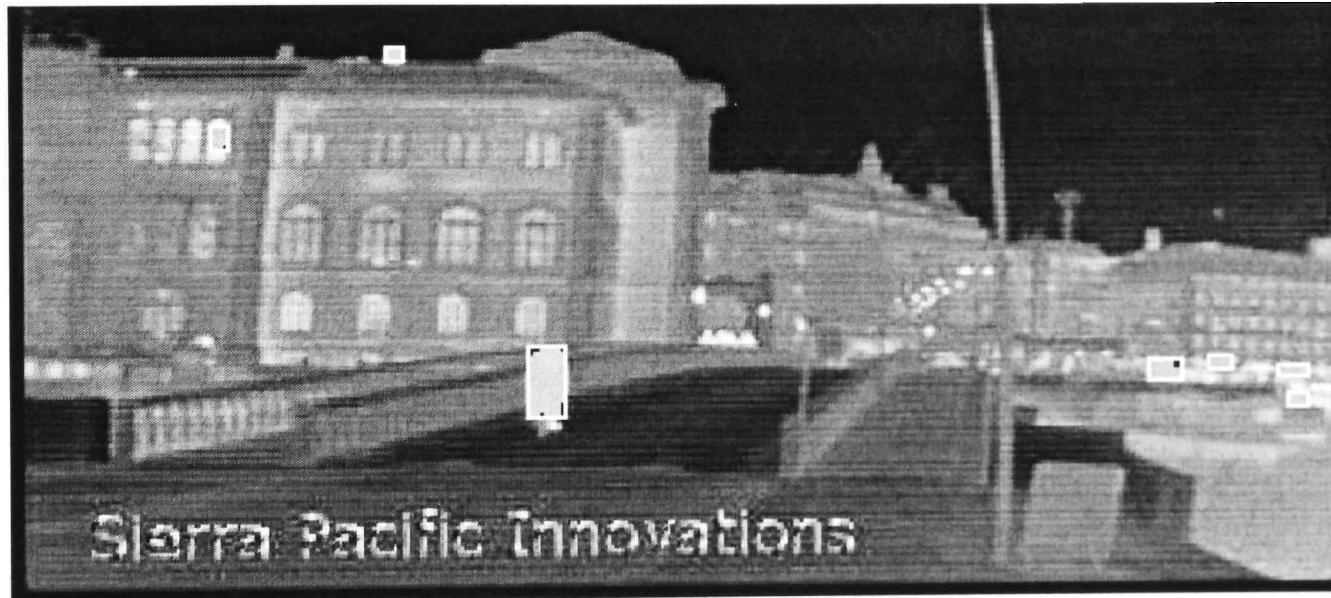


Figure 3.19 Image ‘mancycle.pgm’ after Refinement at 220 Threshold

We now present similar results for another sample image. Figure 3.20 shows the original image ‘horses.pgm’. Figure 3.21 shows the histogram of image ‘horses.pgm’. The histogram in Figure 3.21 shows three major modes with first two representing the background. Since the base of the second mode on the left side is above the average gray

level we can say that this mode belongs to the background. Also, since second and third modes are easily discernible, we select a threshold between base of the second and base of the third mode. Figures 3.22-3.24 shows the image thresholded with our approach, Fujimura's, and Otsu's approach, respectively. In Figure 3.23, because of selection of very high threshold, the body of horse in the middle of the image is over segmented in three regions. We consider each region and find that ROI containing the back of the horse is discarded (see portions outside box in Figure 3.26). Also, Figure 3.26, we see that the head of the horse is not selected. From Figures 3.22 and 3.25, we see that, we are able to select a good estimate of ROI for the horse in the middle by thresholding; Although some portion of the horse is not selected by thresholding alone, the fuzzy logic based refinement to the ROI is able to capture good segmentation. Figures A.3 and A.4 in the appendix also shows the effect of over-segmentation. Figure A.4 shows that the single object is refined as three different object with Fujimura's approach whereas with our thresholding the whole object can be captured.



Figure 3.20 Original Infrared Image 'horses.pgm'

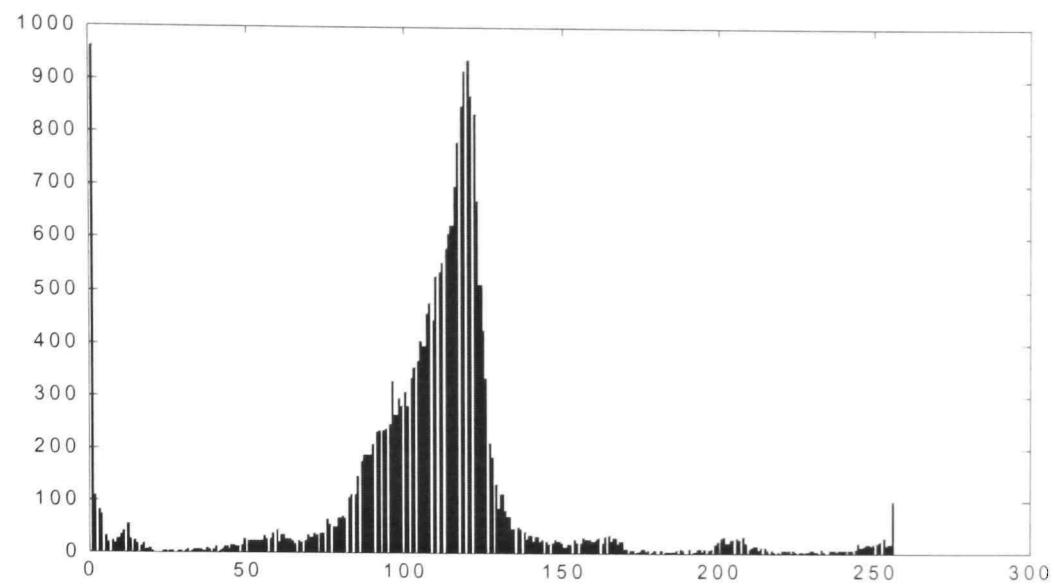


Figure 3.21 Histogram of Image ‘horses.pgm’

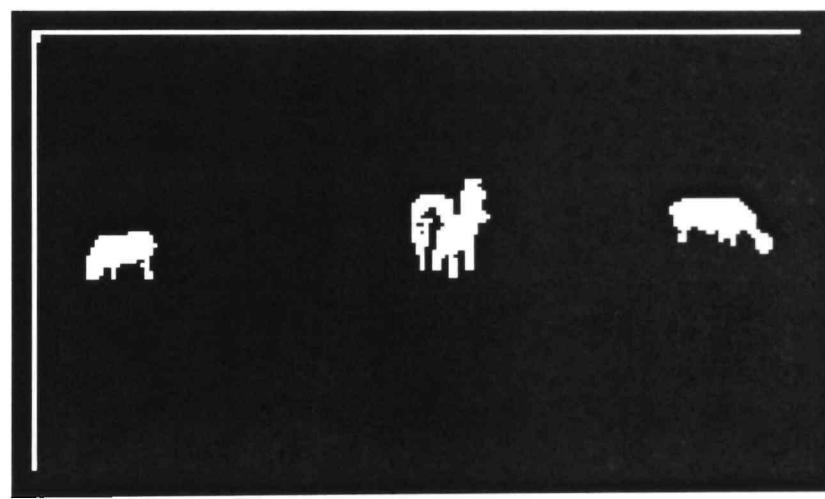


Figure 3.22 Image ‘horses.pgm’ Thresholded with Our Approach

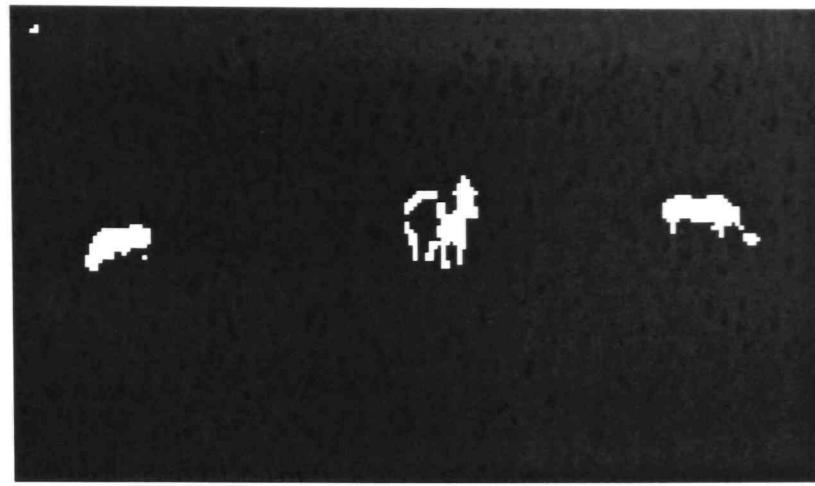


Figure 3.23 Image ‘horses.pgm’ Thresholded with Fujimura’s Approach

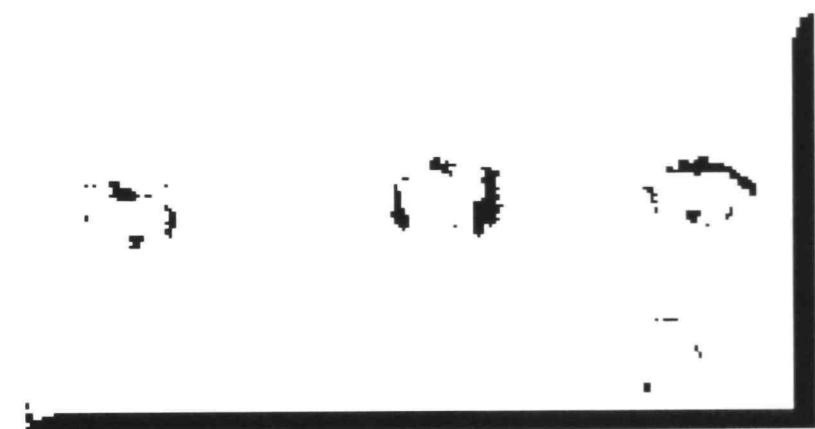


Figure 3.24 Image ‘horses.pgm’ with Otsu’s Threshold

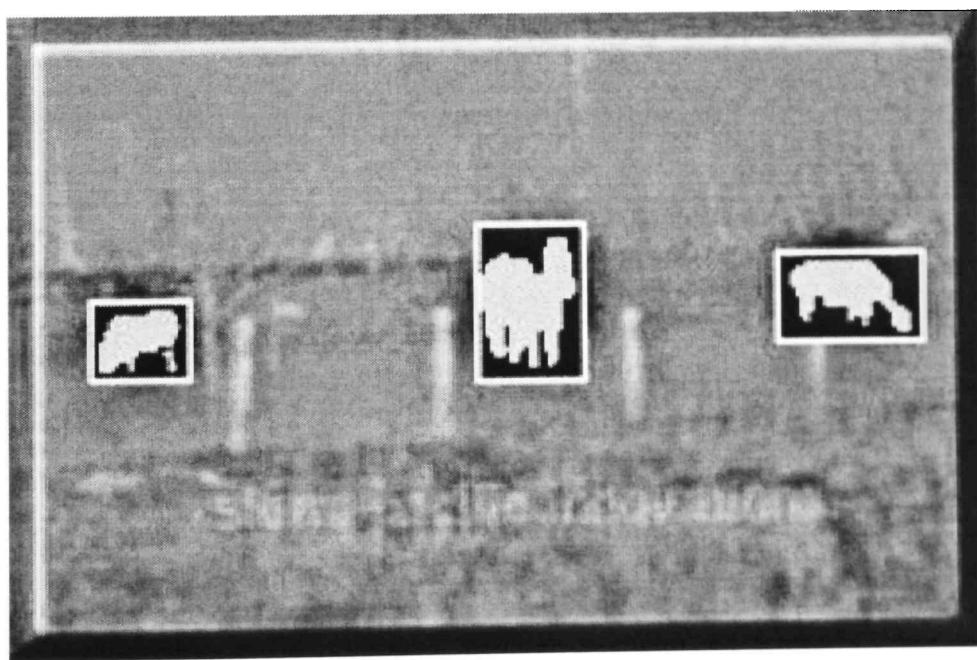


Figure 3.25 Image ‘horses.pgm’ after Refinement (Our approach)

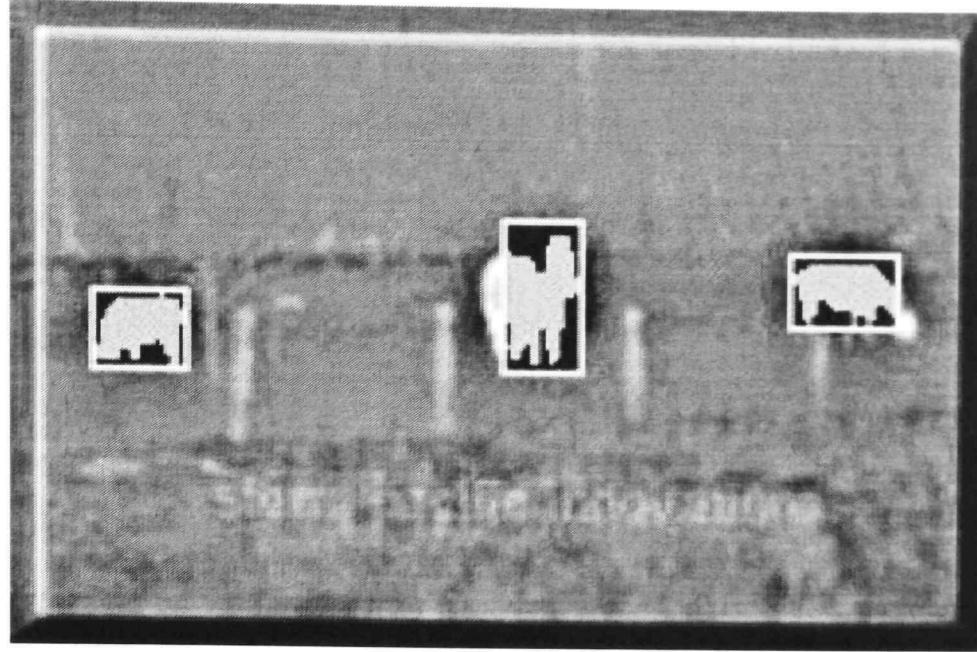


Figure 3.26 Image ‘horses.pgm’ after Refinement (Fujimura’s approach)

Finally, we present segmentation results of the tank image, which led to this study.

Figures 3.27-3.30 shows the result for tank image obtained using our approach, Otsu’s, Fujimura’s and Sun-Gu Sun’s methods. Figures 2.1 and 2.2 show the original image of the tank and its histogram, respectively. First, we compare Figure 3.27 (our thresholding method followed by refinement of [6]), with Figure 3.30([6]). The only difference in the results is due to assumption of [6] that the minimum and maximum size of the object for thresholding and ROI identification. The bounding rectangle in each figure shows the identified ROI. Comparing Figure 3.27 with Figure 3.30, we can say that segmentation results obtained without the knowledge of the size of the object, of our method is acceptable. Figure 3.28 show that Otsu’s method of thresholding coupled with the refinement results are nearly as good as presented in Figure 3.27, because the histogram of tank is bimodal. Figure 3.29 shows the result for the same using Fujimura’s method, it shows that a small portion of the tank at the top is not captured as part of tank.

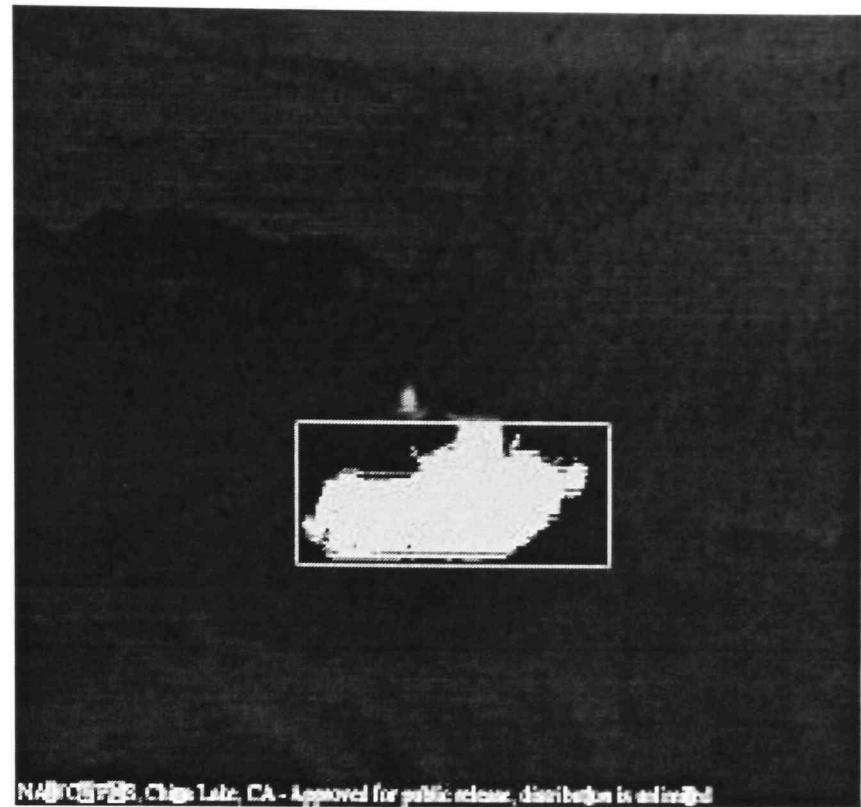


Figure 3.27 Refined Image of Tank with Our Approach

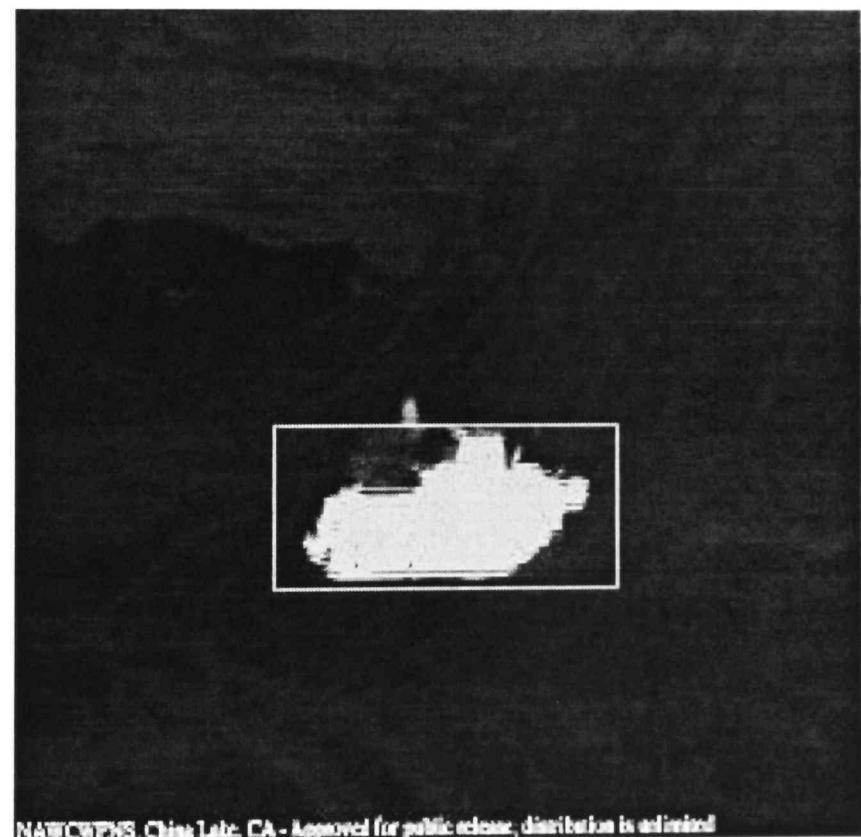


Figure 3.28 Refined Image of Tank with Otsu's Approach

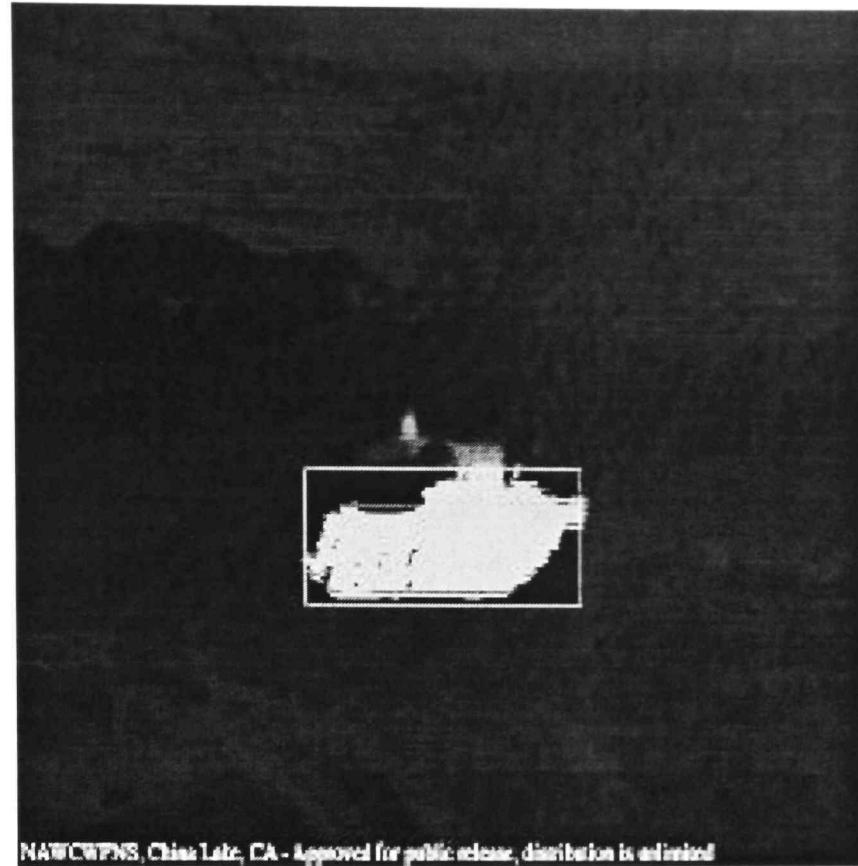


Figure 3.29 Refined Image of Tank with Fujimura's Approach

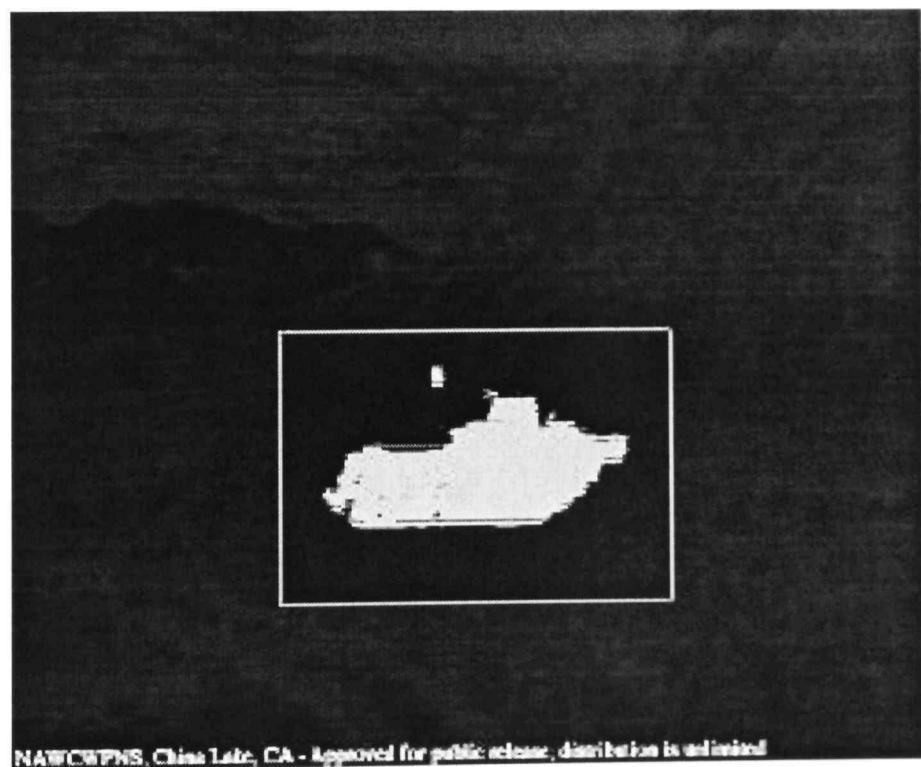


Figure 3.30 Refined Image of Tank with Sun-Gu Sun Approach

### 3.8 Limitations of Our Method

In this section, we discuss the limitations of our implementations. Since we placed main focus on the thresholding of bimodal and unimodal histograms in this thesis, the performance of our method is not as good when applied to an image with multiple modes of histogram. If the histogram consists of at most two major modes representing the background and only small size objects, we can skip the first background mode and apply shifted unimodal thresholding on the second mode. For multi-modal complex histogram, finding the base becomes difficult and our approach do not work well; a histogram smoothing before mode detection might give better results in this situation. Also the maxima based mode detection can be replaced by some other better method of mode detection techniques discussed in [34] and [37].

## CHAPTER IV

### OBJECT MATCHING/TRACKING IN IR IMAGE SEQUENCES

#### 4.1 Introduction

Identification of moving objects is important for surveillance. It requires us to match the objects in successive image and then track their motion. In this thesis we consider matching of objects in two dimensions as part of our study of motion detection, which eventually leads to the object based tracking. This is sometimes referred to as block or region matching. For successful tracking, we must first isolate each object in every two consecutive frames, and then establish correspondence between the objects. This process should be continuously performed over a small time interval to successfully track all objects. Given a video sequence, human eye can easily isolate objects from background by combining past information and the prediction even though their positions may change in the next frame. Human eye can easily detect the boundary of object. If the object boundaries are not easily distinct for human eye, motion of objects helps in finding the boundaries. In digital video processing, the same should be done very fast. If this process is slow, the object might leave the scene before it can be identified. Due to the requirement of fast processing and randomness present in the segmented images due to noise in image sequence, matching becomes difficult. Segmentation basically isolates objects from background, where it is possible to represent an object by a separate and distinct region. Due to changes in position and attributes of the objects, it is not necessary that the regions corresponding to the objects appear exactly in the same positions in two

consecutive frames. Usually in problem of matching, it is assumed that the object moves slowly; hence, the distance between any two objects is not much different in consecutive frame. Thus, we can use Euclidean distance between the positions of the objects for matching. But using only this feature may lead to error when one object is moved near to other object. Therefore we should also consider other attributes of objects for matching. Using more than one feature for matching can reduce the effect of variations of attributes. Even some attributes, which do not change much for an object, can be given higher emphasis than the other for better match.

The tracking process depends on matching a set of image features present in one frame with features found in the next frame. In [32], a feature based matching technique is used for real-time closed-world tracking. A feature based tracking method matches objects or regions in image frames using some parameters, such as size of object, gray level, position of the object and velocity and shape of objects. Tracking algorithm in [32] uses four parameters, namely size, color, and velocity and current and the past position. This information is used for matching each object in last frame to a blob or region in the new frame. The metric for matching is weighted sum of several individual feature distance measures. This technique is further discussed in section 4.2.

In [12], the authors have dealt with the problem of tracking an object in a sequence of images captured by a camera moving towards object. They used a technique based on tracking regions obtained by image segmentation. First stage is to segment the frames into homogenous regions. As the camera approaches the object, the size of the object as well as the neighboring regions proportionately increase. A region template of

the target is acquired in the first frame and correlated over segmented image of the future frames. Thus, for object tracking, not only the object identification in the first frame is important, but also matching of segments in each time step. Therefore, the process of matching is highly sensitive to the success of any object or region based tracking task.

This chapter describes our study of tracking objects as presented above for IR images. Section 4.2 discusses how different features of objects can be considered for object matching. Section 4.3 formulates a metric for object matching. Section 4.4 describes how we developed a video sequence for our experiments and then presents experimental results.

## 4.2 Feature-Based Inter-Frame Object Matching

In this thesis, we use feature-based approach for matching used in [32]. In [32], an approach of adaptively weighing different features based on contextual information is used to relate objects. Authors used this approach to track children in a closed room with a camera mounted at the top in the center of the room. Collision between children, children coming in and out of the room, and their movements in the room are used as the contextual information for matching. For each frame, they maintained the object's estimated size, color, velocity, and current and past position, and used this information to match each object in last frame to a blob in the new frame. Blobs are regions computed by background differencing. For each blob, its size, position, and average color are computed during clustering. Difference between the average color of the blob in the current frame and object in the last frame, Euclidean distance between their centroid,

difference between the size and also the Euclidean distance between predicted position and the current position are used for matching. For each feature, a match score matrix is prepared. Each score matrix is then normalized and combined into an overall match matrix with different weight. The weight for each feature is determined on the basis of the contextual information.

Once the match score matrix is prepared, a non-iterative greedy algorithm is used to find the match from match score matrix. Greedy algorithm is suitable, when a large numbers of objects have to be matched.

We used this approach to match the objects in a synthesized video sequence.

#### 4.3 Feature-Based Object Matching for Synthesized Video Sequence

We segment each frame separately to obtain regions containing objects in each frame. It remains to determine which object of the previous frame corresponds to an object in the current frame. Matching is a difficult task. The difficulty is caused due to various factors such as non-uniformity in segmentation, some objects may be moving fast, or some new objects may be entering in the scene.

In each frame, there can be a numbers of objects or regions. We first isolate the regions and then refine them. A dynamic data structure is maintained, which contains information on each object in the frame. For inter-frame matching, we also maintain a data structure for the current frame and previous frame.

Because of the possibility of new objects coming into the scene, old objects disappearing or inaccuracy in segmentation, consecutive frames may contain different number of regions. Table 4.1 shows the data structure for each object in the frame.

Table 4.1 Object Data Structure

Field	Data/Feature	Data Type	Description
1	Object Identifier	Integer	Integer
2	Average gray level	Floating point	Average gray level of refined object
3	Size	Integer	No of pixels selected
4	ROI	Structure containing four integers	Structure containing four integers
5	Position of object centroid	Integers	As X and Y
6	Motion Flag	Integer	0 if stationary, 1 if moving and 2 if new object
7	Motion Vector	Integers	Contains both displacement along x and y-axis.

For object matching, a match score matrix is computed for each feature. The match score of a feature is the measure of differences in the region in the current frame with respect to the previous frame. Let  $n$  be the number of objects in previous frame and  $m$  denote the number of regions in new frame and  $k$  denotes different features. Let  $S^k$  denote the match matrix for feature  $k$ ; each element  $S_{ij}$  of this matrix is the measure of how closely the  $k^{th}$  feature of object  $i$  in previous frame match with the region  $j$  in the current frame. For all the features, each row of the matrix  $S_{ij}$  is normalized, i.e.,

$$\sum_{j=0}^m S_{ij} = 1. \quad (4.1)$$

After normalizing the score matrix of each feature, the score matrices are combined to a single matrix by taking the weighted sum of each element of the matrices. The weights are assigned to each feature on the basis of application and context. Thus the final match matrix  $M$  is given by,

$$M_{ij} = \sum_{f=0}^k w_f * S_{ij}^f \quad (4.2)$$

where  $\sum w_f = 1$

$$\sum_{j=0}^m M_{ij} = 1. \quad (4.3)$$

Thus,  $M_{ij}$  is the overall match score matrix. The value of the overall match score depends on the selection of weights to feature. Since our objective was to apply this method to a video sequences describing surveillance images, where only few objects of different types are present; some of which may be moving and some of them may be stationary. Even if an object moves, the position of its centroid between two consecutive frames does not change much. Therefore, a higher weight is assigned to the position of the object. In infrared images, the objects are brighter, and the average gray level of two different objects in the same frame does not differ much. Therefore, a low weight is given to the average gray level. In our experiment, we considered only three features for object matching. These features are the average gray level, position of the centroid and size of

the object. The position of the centroid is assigned a weight of 0.5, and the size and average gray level are assigned the weight of 0.3 and 0.2, respectively.

The object  $i$  match with region  $j$  fully if the corresponding score,  $M_{ij}$  is 0. Since our input video sequence contained a small number of objects, a simple selection criterion is chosen. Also, if an object cannot be matched between the frames then it is considered as new object if more than one region in the current frame matches with the object in previous frame, one having better match is preserved. Once the objects are matched, moving objects are identified and motion vector is calculated of moving objects by comparing objects between two frames. We can also use this motion vector as new feature for matching object in the consecutive frames.

#### 4.4 Experiments

Due to the unavailability of the true infrared video surveillance image, we produced a video sequence for our experiments. Our goal was to simulate surveillance image and achieve characteristics of IR images in our video. This process is described in section 4.4.1. The results of object based matching performed on this synthesized video sequence is discussed in section 4.5.

##### 4.4.1 Development of a Video Sequence for Object Tracking

Since we have focused on IR surveillance images, which are captured to track activity at night, we shot a video using a home camcorder with night shot option. The camera captured images in near-infrared range. In the total dark, it emitted a beam of red

light acting as the source and then captured reflected light as the radiation emitted by the objects. For this reason, we chose some highly reflective material to compose some simple objects. We used car license plates and some light emitting balls to simulate objects. Some objects were moved manually while others remained stationary. The video frames were loaded as ‘pgm’ files and were processed using our program for matching.

Figure A.1 in the appendix shows an example of one such frame. For a successful synthesis, we tried to meet the following requirements,

- i. We required total darkness for taking shots so that frame appear as IR images and not resemble as photographic images. Therefore, we shot the video in a totally closed room.
- ii. Small size objects were used to give an impression that images were captured from far distance to give impression of IR images. Different shape and size objects were used for the same reason.
- iii. Some objects of high reflectance were used so that they appear as hot objects in real surveillance images. A non-reflective background was used to represent the cooler background.
- iv. We also required that some portions of the object have different gray level, i.e. their reflectance is different.
- v. During the simulation, the video must provide some of objects as stationary and some of objects in motion. The camera position is fixed.

It was difficult to obtain reflective objects of different shapes, and we were not able to get the varying gray levels within each object as we saw in the tank image. These

constraints made object matching much more difficult. Figure 4.1 represents two frames of a video sequence we shot for our experiments. Another example is given in Figure 4.3. We were able to introduce some fuzziness on object boundaries, but shapes of objects are synthetic and not natural. Our main objective of matching was to demonstrate that the object features extracted from the real infrared images in chapter 3 could be used for objects tracking. In chapter 3, we have already shown how different objects are isolated and refined. This allowed us to extract certain features of each object such as its size (total number of pixels), average gray level, and a rectangular region bounding refined object (ROI).

## 4.5 Results

In this section, we present three different sets of results. (i) First set describes results of an implementation of the method presented in section 4.3. First we present values of different features, size, centroid and average gray levels of objects considered for matching. Then we present resulting score matrix and analyze matching decisions. (ii) Then we present similar results of another experiment to describe difficulties that arose due to noise in the images. (iii) Next we present results of tracking experiment where our matching program is able to identify objects that enter and/or leave scenes. Finally, we present results of tracking by plotting centroids of moving objects through the whole video sequence.

#### 4.5.1 Results of Feature-Based Matching

Figure 4.1 presents two frames used for our first experiment. These were segmented and values of the three features were computed; these are presented in Table 4.2.

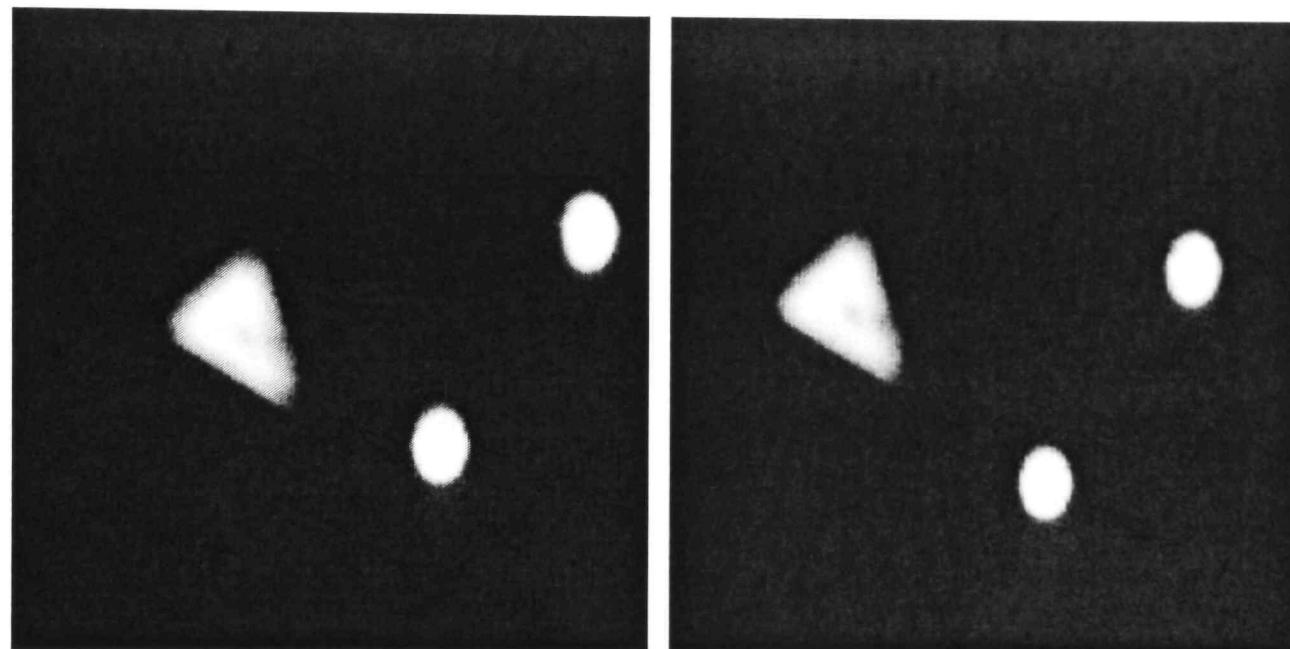


Figure 4.1 Two Consecutive Frames with Three Objects

Table 4.2 Region Attributes of Frames 1 and 2 for Triangle Image

Frame	Region no.	Size	Centroid (X,Y)	Average Gray level
First	1	7702	221, 235	203.7
	2	2039	576, 160	222.51
	3	1962	425,322	223.5
Second	1	7704	174, 214	206.03
	2	1855	529, 185	221.2
	3	1987	380, 347	200.06

Table 4.2 was used to compute score matrix of each feature separately. They were combined to produce the overall match score matrix; feature weight of, 0.5, 0.3 and 0.2 were assigned for the Euclidean distance, size and gray level, respectively. This matrix is given in Table 4.3. From Table 4.2, we see that average gray level of the all three objects are very near and using them for matching should not have much effect.

Table 4.3 Overall Match Score of Triangle Image Sequence

Objects in First frame	Overall match score for regions in second frame		
	Region 1	Region 2	Region 3
1	0.06578	0.58035	0.35387
2	0.64782	0.05236	0.29982
3	0.64966	0.18731	0.16303

From Table 4.3, we can see that object 1 in the first frame matches to the region 1 of the second frame, which has the minimum weighted difference. Likewise, object 2 of frame 1 can be easily matched with region 2 of frame 2. However matching object 3 is not easy. Assuming that the value in third row 0.18731 and 0.16303 are nearly equal. Object 3 could be matched with either of region 2 or region 3. If we consider object 3 first and match with region 2, and then consider object 2 and try to match with region 2, we will conclude that object 2 is missing from the frame 2. This exhibits difficulty of the feature based matching algorithm. We note that even for such simple images, where objects are not moved much, matching is difficult, because results may depend on order of selection of objects for matching. If we must match all objects at the same time, then

we should choose the best match first and then match the rest. Problem arises because new objects can appear in the next frame, these objects should be identified as new objects.

#### 4.5.2 Another Result of Feature-Based Matching

For our second experiment, we used car license plates as objects in the video. These are shown in Figure 4.2. Although the images seem to show that there are three objects, our segmentation program extracted five objects in frame 1 and four objects in frame 2. This is due to the noise in the images caused by high reflectance of plates. This is result of over-segmentation. Thus, the noisy regions further increase the problem of matching. The regions that are not present in the first frame appearing in the next frame are to be treated as the new objects in the scene, which is not the case here. The features of frames are tabulated in Table 4.4 and the corresponding score matrix is given in Table 4.5.

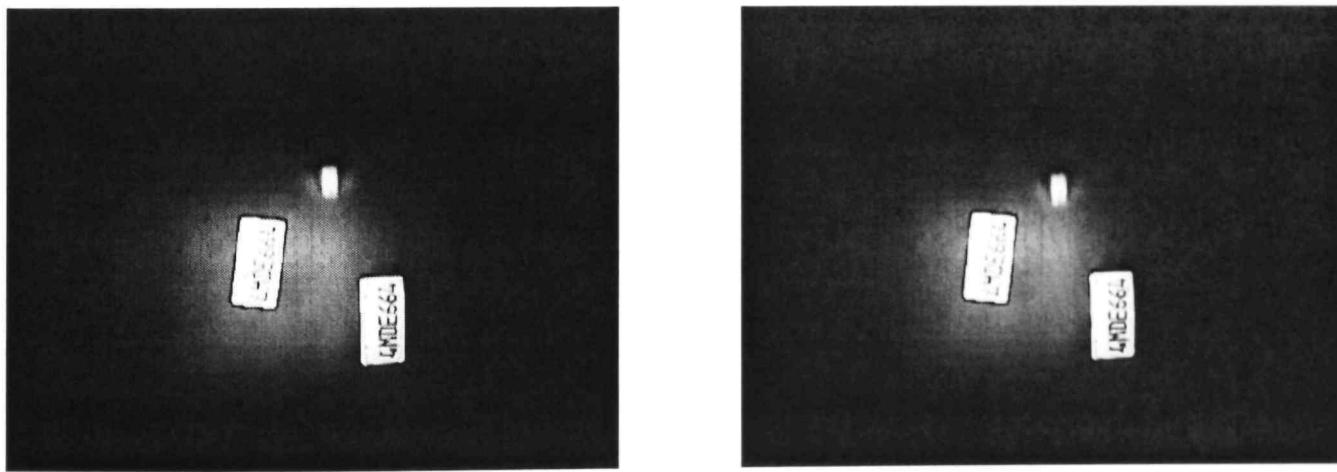


Figure 4.2 Consecutive Frames Containing Different Bright Regions

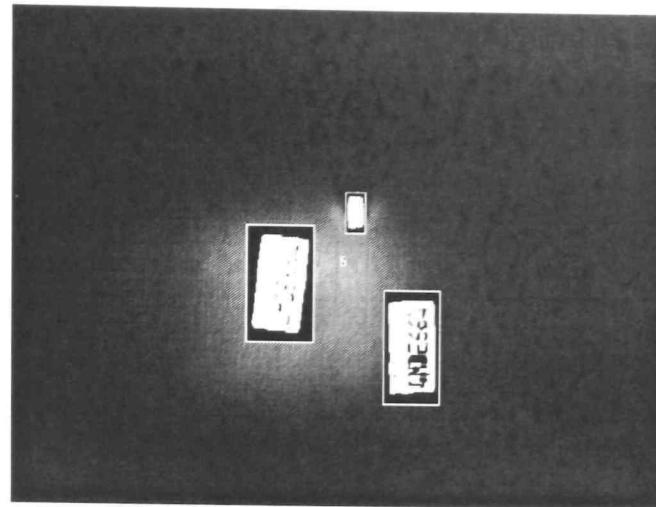


Figure 4.3 First Frame of Figure 4.2 after Segmentation

Table 4.4 Object Attributes of Frames 1 and 2 for Plate Image

Frame	Region no.	Size	Centroid (X,Y)	Average Gray level
First	1	4090	262, 263	227.37
	2	3634	389, 321	201.3
	3	361	335, 180	204.99
	4	86	307,245	135.23
	5	9	322, 250	134.39
Second	1	4018	262, 263	227.51
	2	3508	390, 321	215.29
	3	368	335, 193	200.49
	4	9	324, 240	133.66

Table 4.5 Overall Match Score of Plate Image Sequence

Objects in First frame	Overall match score for regions in second frame			
	Region 1	Region 2	Region 3	Region 4
1	0.00277	0.26711	0.33680	0.39332
2	0.24590	0.03217	0.31504	0.40688
3	0.35916	0.37606	0.02793	0.23686
4	0.33201	0.43669	0.18973	0.04157
5	0.36462	0.41586	0.19658	0.02294

We see that only first three regions can be matched with the corresponding objects and, the others must be tagged as new objects. Both objects four and five match with region 5 since it is better than the other one.

Figure 4.4 shows an example where matching of objects between two frames is difficult. These frames are not consecutive frames, and hence, the triangular object has moved away considerably. Since we have given higher weight to the position rather than size of the objects, it may be difficult to match these objects based on match score matrix, which is given in Table 4.7. Furthermore, if we refer to Table 4.6, we find that the average gray level of the objects have also changed a lot, which also aids in failure to match. From Table 4.7, we can see that both objects 2 and 3 in the first frame match with region1 in next frame. Consequently object 3 is matched with region 1, and the others are recognized as new objects.

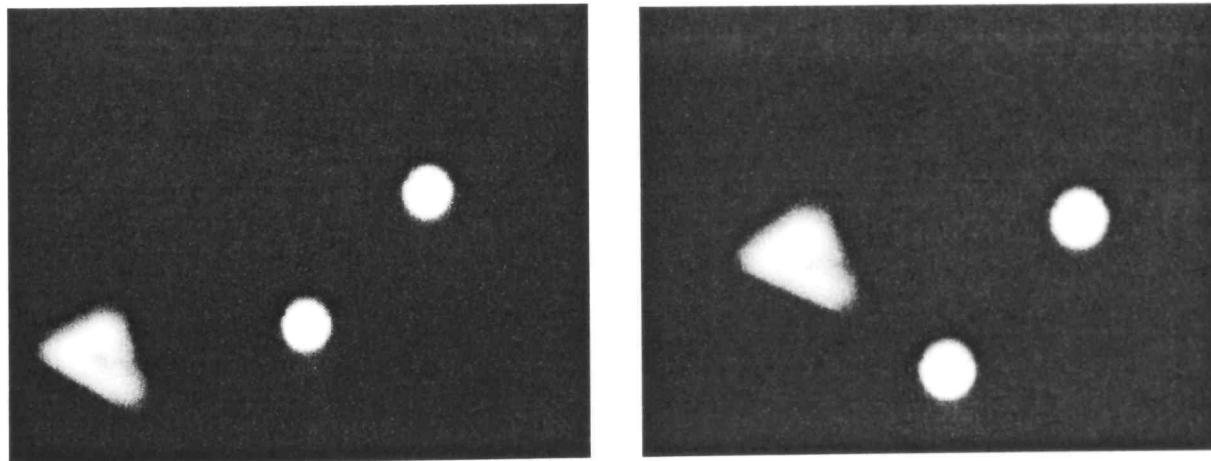


Figure 4.4 Two Frames of a Video Sequence 90 Frames Apart

Table 4.6 Region Attributes of Distant Frames of Triangle Image

Frame	Region no.	Size	Centroid (X,Y)	Average Gray level
First	1	6145	95, 364	198.696228
	2	2031	325, 335	225.640488
	3	1943	459, 196	221.585892
Second	1	2497	485, 229	225.844589
	2	8163	176, 268	181.925430
	3	2842	337, 390	194.927750

Table 4.7 Overall Match Score for Distant Frames of Triangle Image

Objects in First frame	Overall match score for regions in second frame		
	Region 1	Region 2	Region 3
1	0.47561	0.23104	0.29335
2	0.17189	0.65005	0.17807
3	0.07055	0.67325	0.25620

Our experiments showed that if two frames chosen for match are close in the video sequence is mostly successful. This is because only a small number of objects move other objects remain fixed. Also, the movement is slow enough so that the changes between consecutive frames are very little. Since the camera takes about 30 frames per second, the movement introduced manually on video sequence is not significant to produce mismatch between the consecutive frames. For this reason we compared frames, which are 50 or more, frames apart (taken over minimum 2 seconds) in order to introduce significant changes so that mismatch occurs.

#### 4.5.3 Results on Object Tracking

Tracking of the objects entirely depend on the result of matching of objects in every two successive frame. We performed experiments, with two different video sequences, one with triangle, and other with plates. Matching results are propagated from one frame to another frame in the form of a list of object identifier. We assign each object a distinct ID at the beginning and then in each step of matching, the object ID is assigned on to the matched objects in the new frame. If new frame contains objects that do not matched with any of the objects in previous frame, they are assigned new object ID. We verified the tracking results by inspecting the attributes and position of the objects in final frame and their object ID. We conducted experiments with consecutive frames, alternate frames and even skipping four frames with in the video sequence; all objects were tracked correctly. Only skipping a large number of frames results in mismatch and therefore improper tracking. Figure 4.5 shows a series of frames considered for one such experiment. Our matching program was able to determine one of the objects leaving the scene in third frame. It also determines new object entering in frame 4.

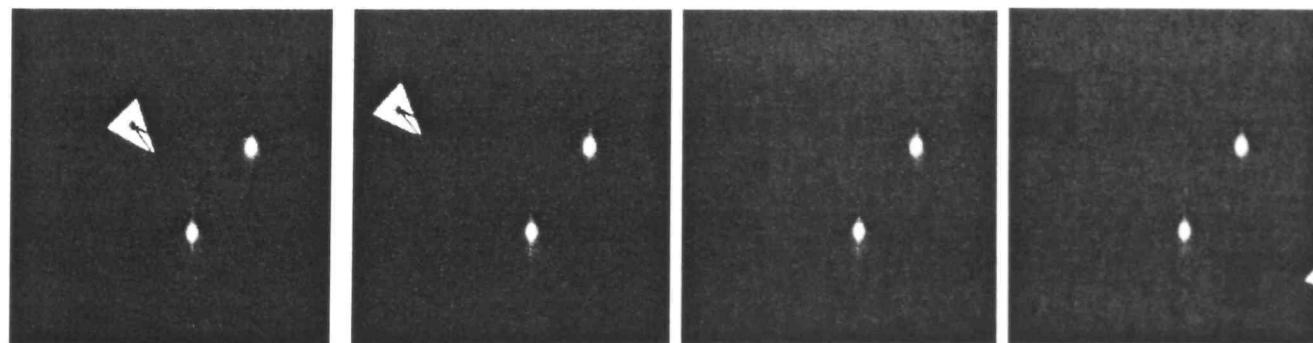


Figure 4.5 Sequence of Frames with Object Leaving and Entering the Scene

Figure 4.6 shows the two end frames of a sequence considered for another experiment, Figure 4.7 shows the result of tracking. The three objects are assigned unique ID, which are represented by ‘\*’, ‘+’, and ‘o’ in the plot. The plot shows the tracking of centroid of each object. Results show that two of the objects are almost stationary but the triangular object has moved significantly. The trace of ‘\*’ represents the path of its centroid.

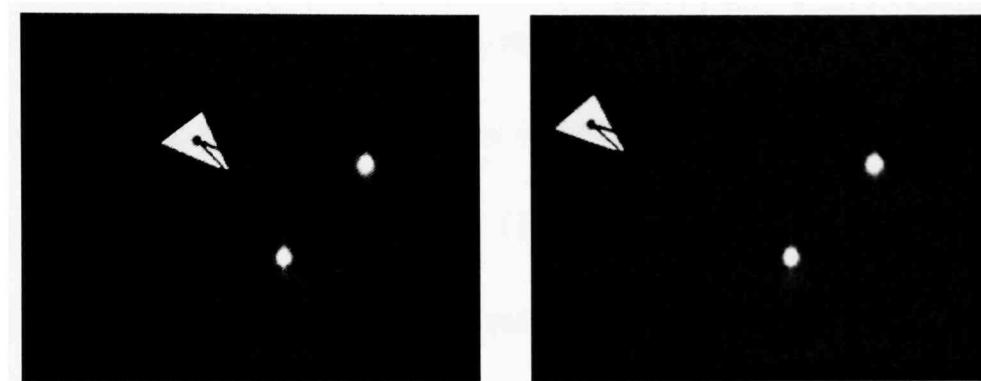


Figure 4.6 First and Last Frame of 17-Frame Image Sequence

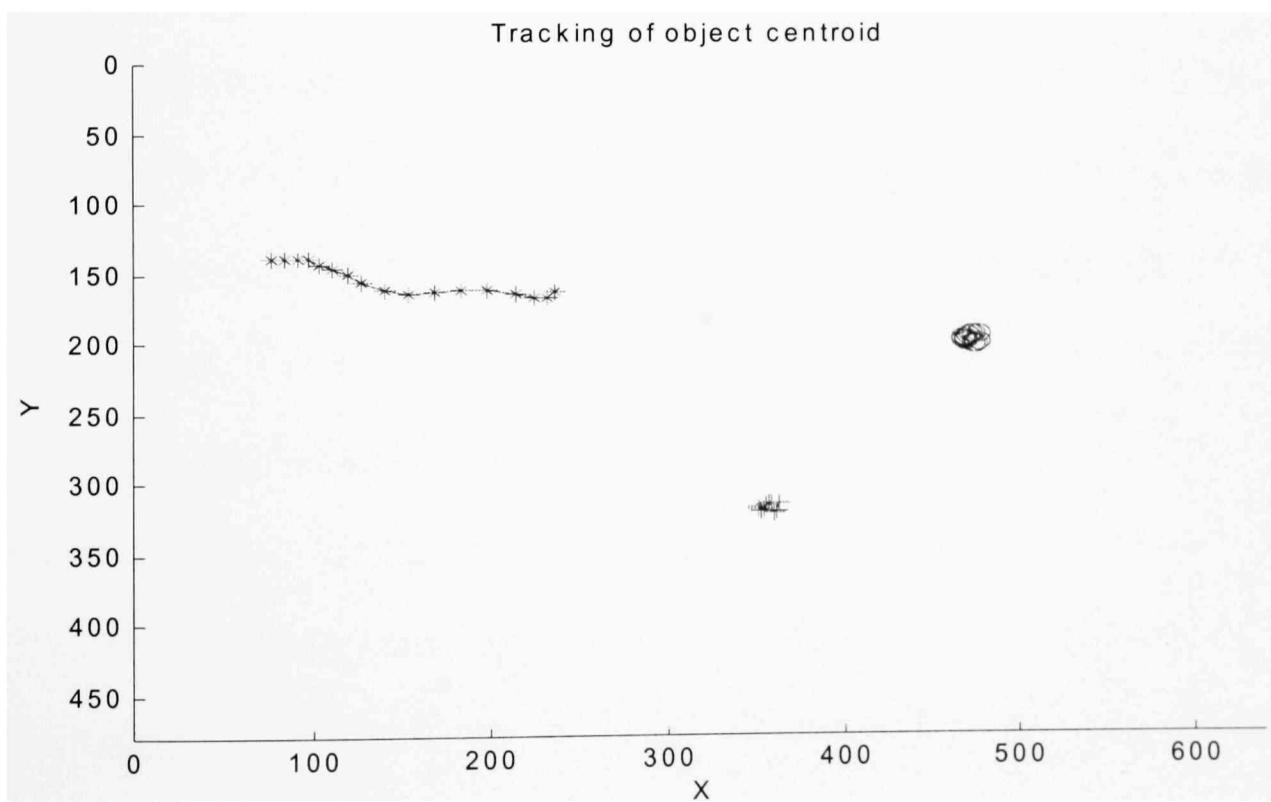


Figure 4.7 Centroid of the Matched Object Tracked for 17 Consecutive Frames

## 4.6 Summary

Above described results of object matching and motion tracking for simple video sequences. We assumed that camera is fixed and all motion is in two dimensions, which constitute the view of the camera. We considered two-dimensional objects and translatory motions, because our goal was to simulate surveillance images. Results show that our programs were able to match and track objects satisfactorily for small data sets we could develop. Due to lack of time and resources, we could not test or develop any other matching algorithm or test them on complex images. Given time, we would have taken many other factors into account. A few are: (i) independent movement of multiple objects; (ii) more exhaustive set of object features; (iii) speed of segmentation program; and (iv) matching metric and criterion.

## CHAPTER V

### SUMMARY AND CONCLUSIONS

#### 5.1 Summary

In this thesis, we have considered segmentation of infrared images for detection and tracking of moving objects in surveillance image. For this purpose, we outlined characteristics of gray-level IR images and observed that the main task is to capture objects of interests. We first applied a general-purpose image segmentation program developed for colored photographic image. We found that any such program caused over segmentation. Then we studied some thresholding segmentation methods, which compute histogram of gray level image and split the image into foreground and background by choosing a gray level from the histogram as the threshold. This exposed us to two problems: (1) selection of proper threshold is a complex task, and (2) thresholding alone is not sufficient for IR images. For these reasons, we studied and implemented the fuzzy logic based approach of [6]. In this approach, computation of threshold requires that the maximum and minimum sizes of objects be known before hand. This is unrealistic. Therefore, we studied histograms of some IR images and developed an adaptive global thresholding method. We compared segmentation results of this heuristics with methods of [33] and [26]. Our method captures regions of interest (ROI) more closely for all sample images we considered. The objective is to include all components of object of interest within ROI. For histogram of these images we determined the global threshold using maxima based mode detection method. We assumed that intensity of foreground

and background areas is sufficiently different and that the histogram distribution or modes are simple. Another parameter that we used in computation was the maximum number of objects in the image. This allowed us to select objects of most interest and limit the processing only for these objects. We have also shown that better ROI can be found by studying characteristics of the histogram modes than by thresholding based on the average and maximum gray level of the image as used in the pedestrian tracking [33].

In chapter IV, we applied our segmentation program for object matching in IR image. The objective was to match objects between successive video images and then use this information to track motion of objects. We studied a feature-based object tracking approach of [32]. Due to unavailability of the surveillance IR image sequence, we shot some clips, which have somewhat characteristic of IR images. Our program was able to match all objects successfully between consecutive frames for the synthesized image with up to four objects. However, when we used two frames, which were far apart in the sequence for object matching, the problem became more complex. Then we extended our segmentation/matching program for tracking position of objects in IR image sequence. In the experiment we conducted, we tracked the position of one moving object with two other objects in stationary position matched in consecutive frames.

## 5.2 Conclusions

Image segmentation is known to be a complex task; its effectiveness depends on image characteristics and the end application. For this purpose, we have considered IR images as input and object extraction, matching and tracking as the end application. We

assume that IR images are such that foreground object are brighter than background. This enabled us to develop threshold-based heuristics for segmentation. From the results given in Chapter III, we conclude that such IR images can be automatically segmented and regions of interests (ROI) containing objects of interest can be determined. From results of Chapter IV, we conclude that segmented IR images can be used to match common objects automatically and track movements of objects. In other words, the task of object tracking can be automated for surveillance images. We also conclude that fuzzy-logic based method captures object boundaries more precisely, but that assumption made in [6] that bound on the size of all objects be known beforehand is not necessary.

Our program finds the ROI of objects or regions in images. This information can be used for image compression. First we can compress the background severely while preserving the ROI containing the objects of interest. Then we can compress the objects by capturing its edges. Also, in this study, we have used a raw histogram for mode detection. Therefore, the method is not applicable to a wide range of images. A histogram smoothing applied before mode detection could give better results. But an optimal smoothing technique should be used for better results. Smoothing with arbitrary window size might give adverse results. Due to the unavailability of the true infrared video sequence, various features that can be exploited for matching could not be used in this work. Adding extra features in the matching process will further increase the accuracy of matching. If we can predict or know some contextual information regarding the participating objects, contextual weighing of various features can be used to aid in

matching. For example, if an object is moving in the previous few frames, giving more weight to the predicted position in the next frame might help us improve matching.

## REFERENCES

- [1] L. F. Pau and M. Y. Nahas, *An Introduction to Infrared Image Acquisition and Classification Systems*, Research Studies Press LTD, England (1983).
- [2] R. O. Duda and P. E Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New-York (1973).
- [3] W. Y. Ma and B. S. Manjunath, “Edge flow: a framework of boundary detection and image segmentation,” *IEEE Int. Conf. On Computer Vision and Pattern Recognition* (1997). Retrieved from <http://vision.ece.ucs.edu/publications/97CVPREdgeFlow.pdf>
- [4] Jianping Fan, David. K. Y. Yau, Ahmed. K. Elmagarmid, Walid G. Aref, “Automatic image segmentation by integrating color-edge extraction and seeded region growing,” *IEEE Transactions on Image Processing*, **10** (10) (October 2001).
- [5] Masataka Kagesawa, Shinichi Ueno, Katsushi Ikeuchi, Hiroshi Kashiwagi, “Recognizing vehicles in infrared images using IMAP parallel vision board”, *IEEE Tran. Intelligent Transportation Systems*, **2**(1) (March 2001).
- [6] Sun-Gu Sun, “Segmentation of forward-looking infrared image using fuzzy thresholding and edge detection,” *Optical Engineering*, **40**(11) (November 2001).
- [7] Wei-Ying Ma, B. S. Manjunath, “Edge flow: a technique for boundary detection and image segmentation,” *IEEE Transactions on Image Processing*. **9**(8) (Aug. 2000)
- [8] Andreas E. Savakis, “Adaptive document image thresholding using foreground and background clustering,” *Proceedings of International Conference on Image Processing* (1998). Retrieved from <http://www.rit.edu/~axseec/papers/icip98adt.pdf>
- [9] Roger F, Philippe M., Armand L., Frank S., Danielle D, Eliane C, “Region-based enhancement and analysis of SAR images,” *Proceedings of International Conference on Image Processing* (1996).
- [10] Paul L. Rosin, “Unimodal thresholding,” *Proc. Scand. Conf. Image Anal.* (1999). Retrieved from <http://www.cs.cf.ac.uk/User/Paul.Rosin/resources/papers/simple2.pdf>

- [11] P. H. Batavia and Sanjiv Singh, “Obstacle detection using adaptive color segmentation and color stereo homography,” *IEEE International Conference on Robotics and Automation* (May 2001). Retrieved from [http://www.ri.cmu.edu/pub\\_files/pub2/batavia\\_parag\\_2001\\_1/batavia\\_parag\\_2001\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub2/batavia_parag_2001_1/batavia_parag_2001_1.pdf)
- [12] H. S. Parry, A.D. Marshall, and K. C. Markham, “Region template correlation for FLIR target tracking,” British Machine Vision Conference (1996). Retrieved from <http://www.bmva.ac.uk/bmvc/1996/index.html>.
- [13] Kostas Haris, Serafim N. Efstratiadis, “Hybrid image segmentation using watersheds and fast region merging,” *IEEE Trans. on Image Processing* **7**(12) (Dec. 1998).
- [14] Robert M. Haralick and Linda G. Shapiro, “Survey, image segmentation techniques,” *Computer Vision, Graphics and Image Processing* **29**, pp. 100-132 (1985).
- [15] Hung T. Nguyen, Elbert A. Walker, *A First Course in Fuzzy Logic* (Second Edition) Chapman and Hall/CRC, Boca Raton, FL (2000).
- [16] R. Lowen, *Fuzzy Set Theory*, Kluwer Academic Publishers, Norwell, MA (1996).
- [17] Xin Yang, S M Krishnan and K L Chan, “Color Image Segmentation Based on Fuzzy Rule-Base Reasoning Applied to colonoscopic Images,” *Critical Reviews™ in Biomedical Engineering* **28** (3-4), pp. 355-361 (2001).
- [18] C. K. Chow and T. Kaneko, “Boundary detection of radiographic images by a thresholding method” In *Frontiers of Pattern Recognition* (S. Watanabe, Ed.), pp. 61-82, Academic Press, New York (1995).
- [19] D. P. Panda and A. Rosenfeld, “Image segmentation by pixel classification in (gray level, edge value) space,” *IEEE Trans. Comput. C-27*, 875-879 (1978).
- [20] S. Watanabe and the CYBEST group, “An automated apparatus for cancer prescreening” *CYBEST, Computer Graphics Image Process.* (1974).
- [21] J. S. Weszka, R. N. Nagel, and A. Rosenfeld, “A threshold selection technique,” *IEEE Trans. Comput. C-23* (1974).
- [22] D. L. Milgram and M. Herman, “Clustering edge values for threshold selection,” *Comput. Graphics Image Process.* **10** (1979).
- [23] Aishy Amer and Eric Dubios, “Image segmentation by robust binarization and fast morphological edge detection,” Vision Interface (VI2000), Montreal, Quebec Canada (May 2000).

- [24] Dorin Comaniciu and Peter Meer, "Robust analysis of feature spaces: color image segmentation," *Proc. IEEE Conference CVPR*, S 750-755. Puerto Rico (1997).
- [25] Kyujin Cho and Peter Meer, "Image segmentation from consensus information," *Computer Vision and Image Understanding: CVIU* **68** (1), pp. 72-89 (1997).
- [26] Nobuyuki Otsu, "A threshold selection method from gray-level histogram," *IEEE Transactions on Systems, Man and Cybernetics*, SMC-**9**(1) (January 1975).
- [27] Mona Sharma, "Performance evaluation of image segmentation and texture extraction methods in scene analysis," Thesis Report, University of Exeter (January 2001).
- [28] C. H. Chen, L. F. Pau and P. S. P. Wang (Eds.), *Handbook of Pattern Recognition and Computer Vision* (Second Edition) World Scientific, New Jersey (1999).
- [29] A. L. Bovik (Ed.), *Handbook of Image and Video Processing*, Academic Press, SanDiego, CA (2000).
- [30] Borko Furht, Stephen W. Smoliar and Hongjiang Zhang, *Image and Video Processing in Multimedia Systems*, p. 225, Kluwer Academic Publishers, Norwell, MA (1995).
- [31] Jianhua Xuan, Tulay Adalr, and Yue Wang, "Segmentation of magnetic resonance brain image integrating region growing and edge detection," *International Conference of Image Processing*, pp. 544-547 (1995).
- [32] Stephen S. Intille, James W. Davis and Aaron F. Boboc, "Real-time closed-world tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 697-703 (June1997).
- [33] Fengliang Xu and Kikuo Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Intelligent Vehicle Symposium*, Versailles, France (June 2002).
- [34] Jeng-Horng Chang, Kuo-Chin Fan and Yang-Lang Chang, "Multi-modal gray-level histogram modeling and decomposition," *Image and Vision Computing* **20**, pp. 203-216 (2002).
- [35] Charles Poynton, "Frequently asked questions about Color." Retrieved from <http://www.inforamp.net/~poynton>
- [36] C. S. Won, "A block-based MAP segmentation for image compressions," *IEEE Trans. Circuit Syst. Video Technol.* **8**, pp. 592-601 (1998).

**APPENDIX**  
**ADDITIONAL RESULTS**

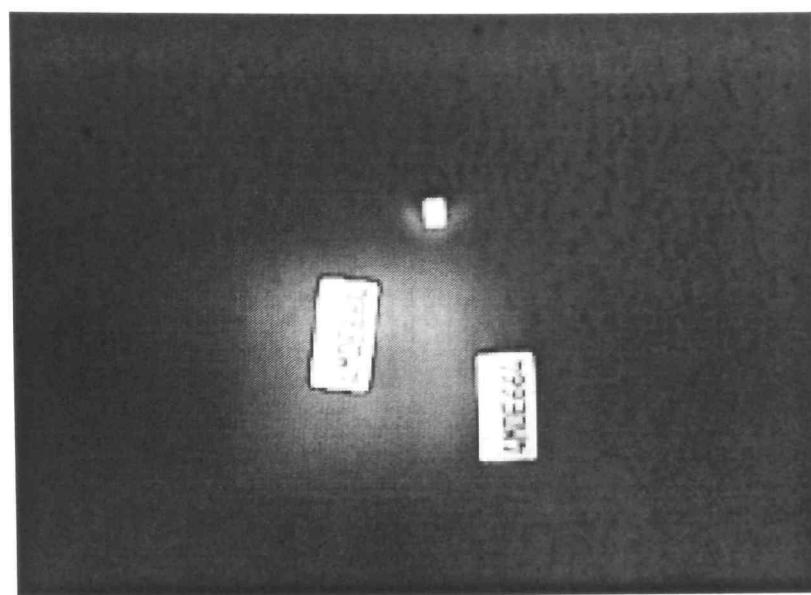


Figure A.1 A Sample Frame From Simulated Video Sequence

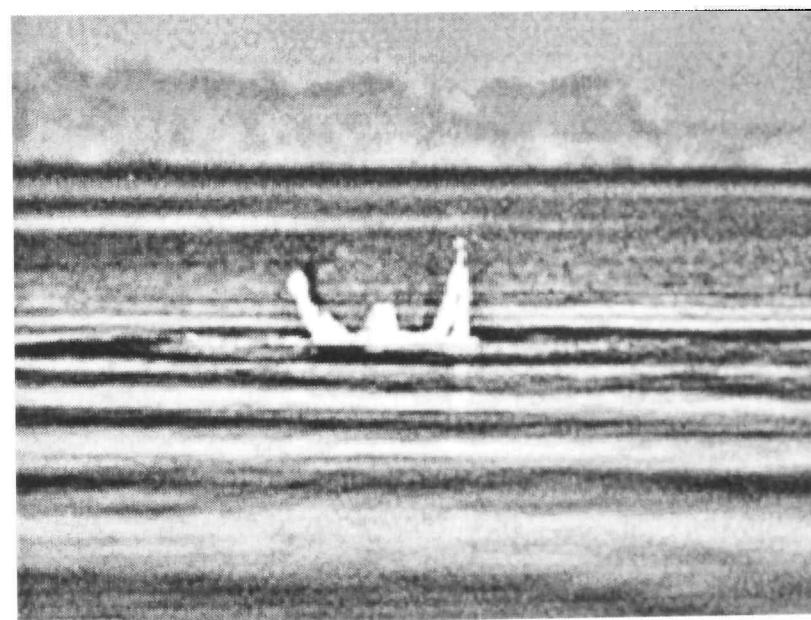


Figure A.2 Original Image swimmer.pgm

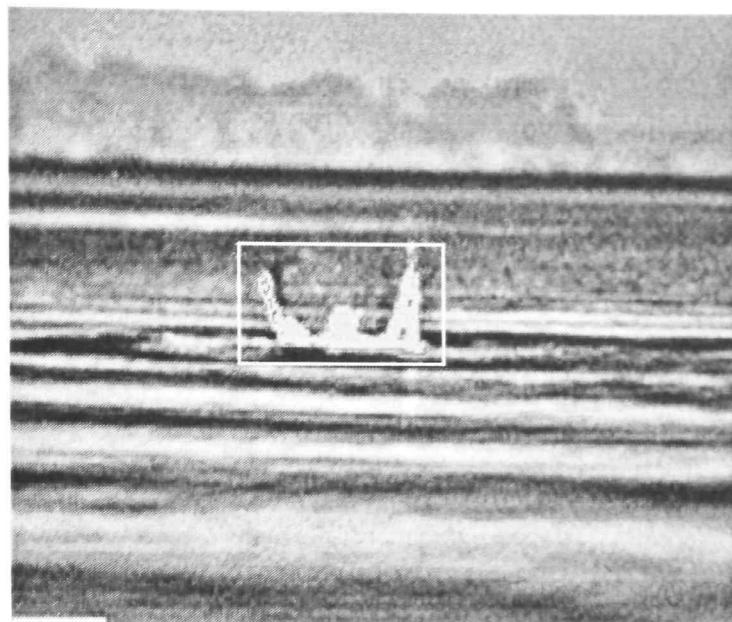


Figure A.3 Image swimmer.pgm Segmented with Our Approach

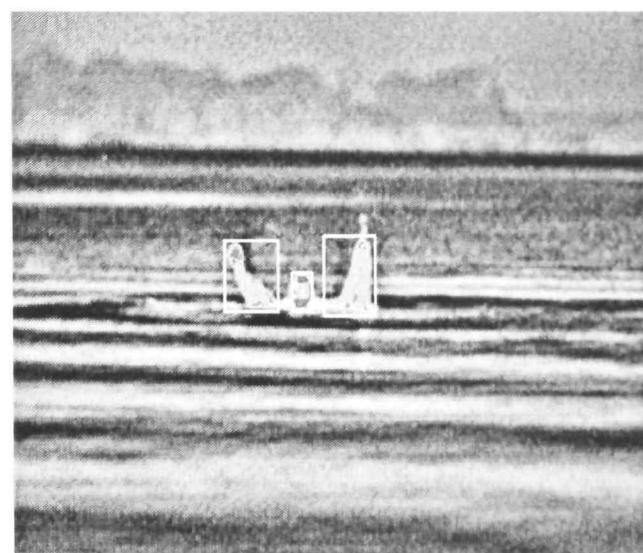
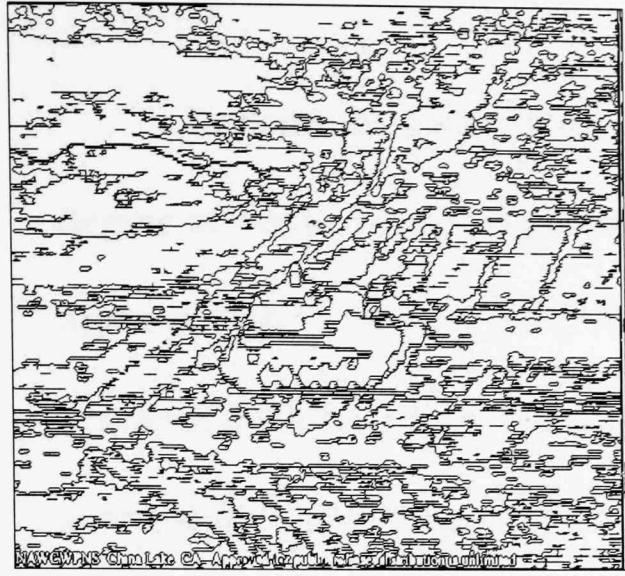
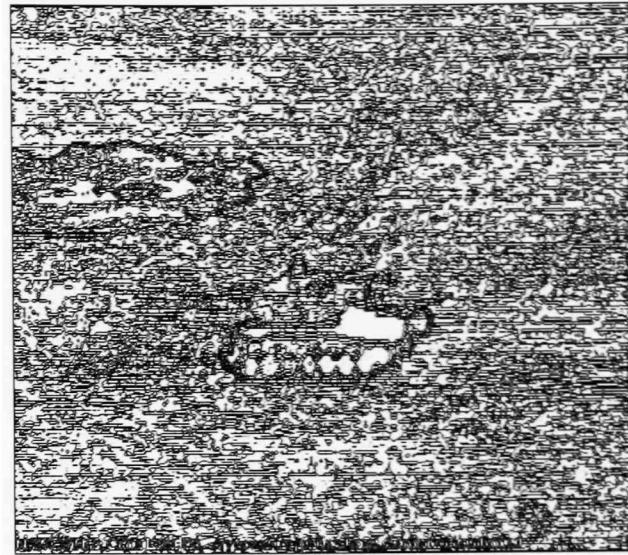


Figure A.4 Image swimmer.pgm Segmented with Fujimura's Approach



(a) Under-segmented



(b) Quantization

Figure A.5 Segmentation of Tank Image with General Homogeneity Based Criterion

## PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University or Texas Tech University Health Sciences Center, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Agree (Permission is granted.)

Disagree (Permission is not granted.)

---

Student Signature

---

Date