OBJECT RECOGNITION USING VIDEO SEQUENCES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ÇANKAYA UNIVERSITY

BY

ARZU BURÇAK SÖNMEZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE 2008

Title of the Thesis : **Object Recognition Using Video Sequences**
Submitted by **Arzu Burçak Sönmez**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Özhan Ç. Uluatam
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

M.R. Tolun

Prof. Dr. Mehmet R. Tolun
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Abdülkadir Görür
Supervisor

Examination Date : 18.07.2008

Examining Committee Members

Asst. Prof. Dr. Abdülkadir GÖRÜR (Çankaya Univ.)

Asst. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.) R. Hassanpour

Asst. Prof. Dr. Tansel ÖZYER (TOBB ETU) T. Özyer

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Arzu Burçak Sönmez

Signature :

Date : 18.07.2008

ABSTRACT

OBJECT RECOGNITION USING VIDEO SEQUENCES

Sönmez, Arzu Burçak

M.S.c., Department of Computer Engineering

Supervisor : Asst. Prof. Dr. Abdülkadir Görür

June 2008, 58 pages

The aim of this study is to construct a system which recognizes specified objects from video sequences. First the moving foreground objects are separated from the background by object segmentation. Then an object template is tried to be located in that segmented foreground by means of matching process. Edge points are used as feature points and measure of edge distance is used to localize the template in the image.

Keywords: Edge matching, Object segmentation, Object recognition.

ÖZ

VİDEO GÖRÜNTÜLERİNDE NESNE TANIMA

Sönmez, Arzu Burçak

M.S.c., Bilgisayar Mühendisliği Bölümü

Danışman : Yrd. Doç. Dr. Abdülkadir Görür

Haziran 2008, 58 sayfa

Bu çalışma, video görüntüleri içinde belli nesnelerin tanınması amacıyla gerçekleştirilmiştir. Öncelikle ön plandaki hareketli nesneler, arka plandan ayrılır. Daha sonra, kenar nokta özellikleri kullanılarak taslak objenin arka plandan ayrılmış nesneyle olan uyumu hesaplanır.

Anahtar Kelimeler: Kenar noktası eşleştirme, Nesne tanıma.

ACKNOWLEDGEMENTS

I would like to thank, first, Assistant Professor Dr. Reza Hassanpour for his guidance and support throughout the completion of thesis and also my graduate career. I would also like to thank my supervisor Assistant Professor Dr. Abdülkadir Görür for his support and encouragements.

Next, I want to thank to my colleagues for their informative comments, technical and moral support throughout the thesis. And I want to thank Cihan Onay for encouraging me about completing my thesis.

Additionally, I want to thank to my family for their support, encouragement and reliance throughout my life.

TABLE OF CONTENTS

CHAPTERS:

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. Introduction to Object Recognition

Humans have high level visual processing skills. We can recognize objects by only looking at them. It is a complex task to identify the human vision recognition process step by step. However, some basic criteria about human vision system are used to achieve object recognition for computer vision applications [1]. Today, these applications are used for various purposes. Industrial applications such as pick and place operations, quality control, flaw detection or military applications such as automatic target recognition are some common examples [2]. Although it is difficult to expect a computer vision system performs as well as a human vision system, there are many successful applications. For example industrial flaw detection application can recognize details that are hard to be noticed by humans. This is because the specific features are searched and the application can be designed according to these features only. Also we can control the environmental variables such as viewpoint and lighting. But recognizing everyday objects is not easy as in defect detection application since controlling natural environment is not a simple task. Figure 1.1 is good example to show the difficulties of recognition process for computer vision applications. The same chair with different sizes and from various viewpoints is shown in the figure. We can easily understand that all chair figures are the same, but it is a complex task to construct an application that recognizes all chair figures in the image. This is because the figures appear in different sizes, in different orientations, under different illuminations, in different location or from different viewpoints [3]. Human vision system performs well since it is invariant to all these variations. For example, different portion of the retina is stimulated when the location of the object changes. Changes in position do not disturb recognition accuracy in human vision system. It means that people do not learn to recognize an object according to its position in the environment or its position relative to other objects. This explains why object recognition can occur even when objects are partially occluded [4].

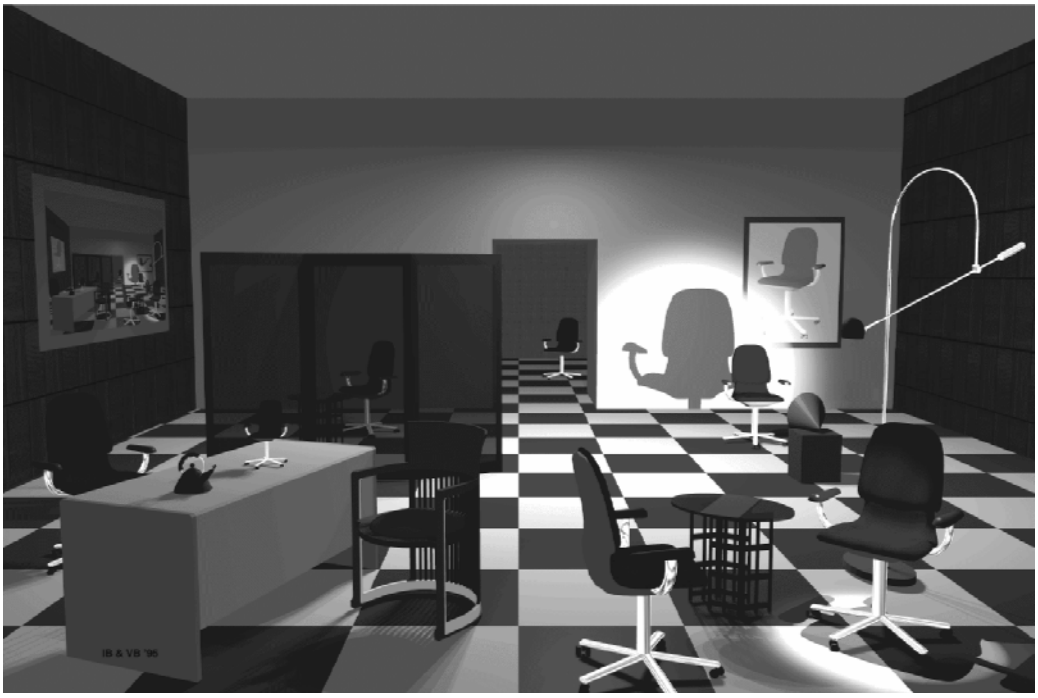


Figure 1.1: Room View Including Same Object in Different Forms

1.2. Main Contributions

There exist a lot of object recognition techniques. These methods are used to recognize objects in images by matching predetermined models. There are various features used by these methods. For example, shape is one of the most powerful features to recognize the object [5]. Apart from the shape there are other features which are used frequently like color, texture [6], invariant moments [7], depth, topology, etc. More complex systems use Bayesian methods and aspect graph methods. These methods provide robustness and accuracy. As mentioned before, image may include noisy objects or it may be cluttered with several different objects. Sometimes the target objects may be occluded or hidden so it can be seen partially. Apart from the occlusion problem, the object location, orientation or size may change in the image. Color and texture of the object may also differ in different parts of the same image or in subsequent frames of the video because of the illumination changes. Because of all these variations, the recognition systems have high computational complexity, long decision latency and vulnerability to error [8].

This work is focused on a powerful method that is computationally cheap. This method is involves segmentation process followed by a matching process. The detailed explanation of the method is given in following chapters.

1.3 Organization of Thesis

The rest of the thesis is organized as follows. Chapter 2 gives the basic steps and widely used methods for these steps of object recognition process. Chapter 3 presents the application steps of object segmentation from video sequences and technical details of edge matching process. Then Chapter 4 gives the conclusion and future work.

CHAPTER 2

BACKGROUND ON OBJECT RECOGNITION

The object recognition problem can be explained such that some knowledge of how certain objects may be explained and an image of a scene possibly containing those objects are given. Then it is necessary to find out which objects are present in the scene and where. Knowledge about the object is provided by an explicit model of the object. Recognition process is performed by finding a correspondence between certain features of the image and comparable features of the model. The two most important problems that a method must deal with are what constitute a feature, and how is the correspondence found between image features and model features. Various methods are developed to overcome this problem. To perform recognition, these methods follow a sequence of steps. First step is feature detection. Feature is important piece of data to show the information content of the image and in this way it helps dimension reduction. Features should be discriminative enough for a successful consecutive classification process. Features should also be robust enough to adapt to changes like viewpoint, scale, or orientation. After feature detection process is completed, next step is the classification step. The goal of classification step is to group items that have similar feature values into groups. These groups are more informative than individual features and help us better understanding for selection and matching of models [9]. A classifier achieves this by making a classification decision. There are two main models for computing a classifier [10]:

2.1. Discriminative Models

The goal of this model is to find optimal decision boundaries by means of the training data and the corresponding labels given before. Linear discriminant analysis and support vector machines are some examples.

2.2. Generative Models

The goal of this model is gathering as much information as possible to represent the data in a suitable way. Therefore an object can be represented by geometrical attributes, by means of contours or shapes or by means of different two-dimensional views. Based on the applied features, these methods can be classified into two main classes:

2.2.1. Local Approaches

This approach is based on local features. This type of feature is a piece of distinctive information belonging to the object of interest. Color, texture, intensity, etc. can be an example of a local image feature. For the success of the recognition systems, image features should be invariant to illumination, scale, rotation, viewpoint etc. But a feature itself is so simple to be invariant to all of these conditions. Thus, several features of a single interest point or region are combined to form more complex image description of the image. This can be referred as image descriptor. Descriptors describe the region that is detected by region of interest detectors before. Region of interest includes significant information for representation of the image and they are extracted by means of detectors. The performance of detectors directly affects the performance of descriptors. There are some examples of common and widely used detectors and descriptors below:

2.2.1.1. Region of Interest Detectors

The region of interest detectors can be classified as below:

2.2.1.1.a. Harris Corner-Based Detectors

This is the most popular region of interest detector. It is based on the Equation 2.1:

$$\mu = \begin{bmatrix} I_x^2(p) & I_x I_y(p) \\ I_x I_y(p) & I_y^2(p) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad \textbf{(2.1)}$$

The response of the detector is called corner response and calculated by Equation 2.2:

$$c = Det(\mu) - k \times Tr(\mu)^2 = (AC - B^2) - k \times (A+C)^2 \qquad \textbf{(2.2)}$$

The main advantage of the method is the minimization of the computational time. The disadvantage of the detector is the extraction of spatial locations only. The detector is only rotationally invariant but no region of interest properties such as scale and orientation are extracted to be used for the later steps such as descriptor calculation [11].

2.2.1.1.b. Hessian Matrix-Based Detectors

Hessian matrix detectors are based on similar idea like Harris detectors. Since Hessian matrix approach uses second derivative of the image, it gives strong responses on blobs and ridges. Hessian matrix is only rotational invariant [12].

2.2.1.1.c. Difference of Gaussian Detector

This detector is used by David Lowe [13]. It is based on approximation of Laplacian by calculating the differences of Gaussian blurred images at several adjacent local scales s_n and s_{n+1}. DoG's scale invariant property is its most important advantage but it is a time consuming process. But it can be calculated in a pyramid much faster than the Laplacian scale space and show comparable results [10].

2.2.1.1.d. Maximally Stable Extremal Regions

It is a watershed-like algorithm based on intensity values. The algorithm can be implemented efficiently with respect to runtime. According to the algorithm, an image is thresholded repeatedly and the regions remaining same are referred as maximally stable extremal regions. Acquired regions are robust against continuous transformations and monotonic intensity changes. Although the detector detects small number of regions, it shows high performance because of its repeatability [14].

2.2.1.1.e. Entropy Based Salient Region Detector

It is based on gray value entropy of a circular region in the image by the Equation 2.3. By means of entropy value, remarkable feature estimation is achieved [15].

$$H_f(s,x) = -\int p(f,s,x) \times \log_2(p(f,s,x))df) \qquad \textbf{(2.3)}$$

p is the probability density function for the entropy and it is estimated by the intensity histogram values as features(f) of a region(F) for a given scale(s) and location(x). The detector is scale and rotational invariant. Affine invariance of the algorithm is supplied by an extension but affine invariant implementation needs longer runtime [16].

2.2.1.2. Region of Iinterest Descriptors

The region of interest descriptors can be classified as below:

2.2.1.2.a. Scale Invariant Feature Transform

Scale invariant feature transform method is very popular and efficient method. The features extracted by the method are invariant to image scaling and rotation. They are also partially invariant to change in illumination and viewpoint. These features are well localized in both frequency and spatial domains. This means that the effect of the occlusion, clutter or noise can be minimized. Another important property of the method is its capability of extracting large number of features which are highly distinctive. The cost of feature extraction is also decreased by cascaded filtering approach. First, an initial test is applied and the high cost operations are applied on the locations passing this test. Scale invariant feature transform method is scale invariant. To be able to identify the locations with scale invariance property, the features that are stable across all possible scales should be searched. The experiments over this subject [17], [18] shows that the most suitable scale-space kernel is Gaussian kernel. So, scale space of an image can be computed by convolving the Gaussian kernel with various scales by the input image as in Equation 2.4 and 2.5.

$$\mathrm{L}(\mathrm{x},\mathrm{y},\sigma)=\mathrm{G}(\mathrm{x},\mathrm{y},\sigma)*\mathrm{I}(\mathrm{x},\mathrm{y}) \qquad \textbf{(2.4)}$$

$$\mathrm{G}(\mathrm{x},\mathrm{y},\sigma)=(1/2\sigma^2)\times \mathrm{e}^{-(x^2+y^2)/2\sigma} \qquad \textbf{(2.5)}$$

In order to detect stable key points in scale space, first the input image should be convolved with two Gaussian functions with two nearby scales and then these two convolved images should be subtracted from each other. This process helps finding scale space extrema.

$$\mathrm{D}(\mathrm{x},\mathrm{y},\sigma)=(\mathrm{G}(\mathrm{x},\mathrm{y},\mathrm{k}\sigma)-\mathrm{G}(\mathrm{x},\mathrm{y},\sigma))*\mathrm{I}(\mathrm{x},\mathrm{y}) \qquad \textbf{(2.6)}$$

$$\mathrm{D}(\mathrm{x},\mathrm{y},\sigma)==\mathrm{L}(\mathrm{x},\mathrm{y},\mathrm{k}\sigma)-\mathrm{L}(\mathrm{x},\mathrm{y},\sigma) \qquad \textbf{(2.7)}$$

Equation 2.6 and 2.7 shows the scale space difference of the same image convolved by different scale gaussian kernels. This process leads to scale space extrema detection.

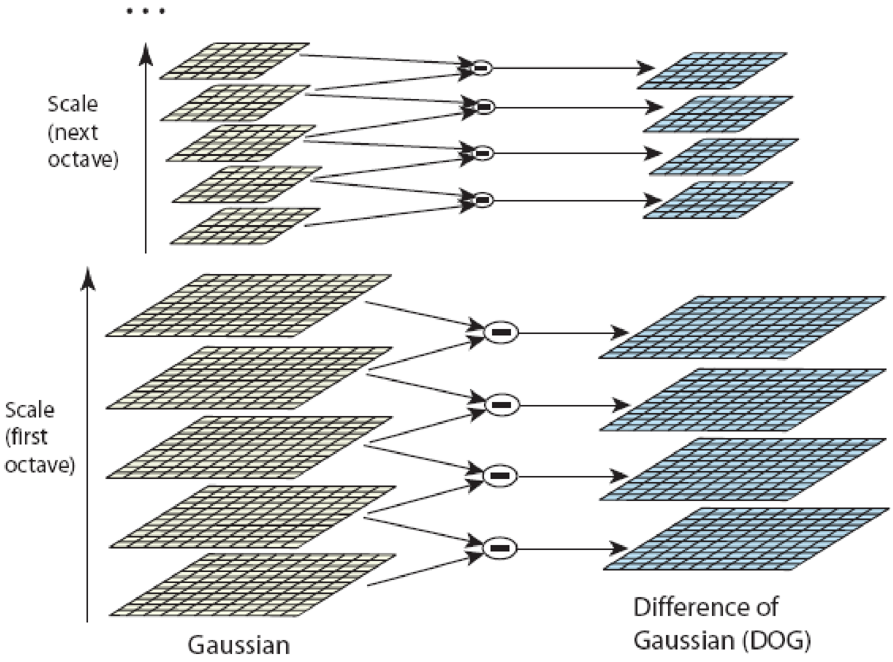


Figure 2.1: Construction of Difference of Gaussian

According to the figure, the original image is incrementally convolved with Gaussians. Gaussian kernels are separated each other by a constant factor *k*. This is shown on the left side of the figure. Then the adjacent image scales are subtracted from each other to produce DoG images shown on the right. Once complete octave is processed, the Gaussian image is downsampled and the process is repeated. Scale-space extrema can be found by local extrema detection. This process is achieved by comparing sample points with its eight neighbors in current image and nine neighbors in other images above and below with different scales. At the end of comparison, if the sample point is maximum or minimum among the neighbors, it can be selected as keypoint. This process is shown in the Figure 2.2.

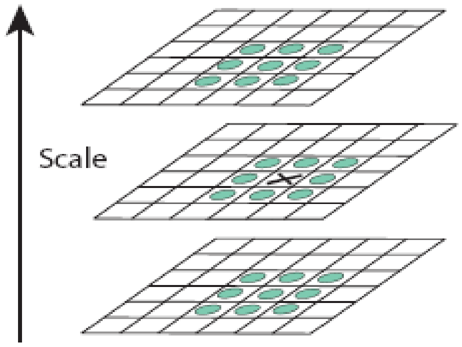


Figure 2.2: Extrema Detection Process

After keypoint candidate has been found, the next step is to reject the points that are sensitive to noise and are poorly localized. For this reason, 3D quadratic function is used to detect the interpolated location of the maximum. This method supplies a significant improvement in matching and stability. Elimination of points with low contrast is not enough to extract stable keypoints. As mentioned before, candidate points are tested initially and the points passing the test are used for high cost applications. Next test is about the poorly localized points along the edges. The approach from [11] is used to calculate the ratios of principal curvatures. The keypoints with the ratio between principal curvatures under some threshold can not pass the test and are eliminated.

There are former studies on rotational invariant properties [19]. This is not an efficient approach since all descriptors are limited as they are based on rotationally invariant measure. The approach used in Scale invariant feature transform method is generating more stable result than the previous works. In this approach:

- L(x,y) = Gaussian smoothed image at scale of keypoint
- m(x,y) = the gradient magnitude

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \qquad \textbf{(2.8)}$$

- $\theta(x,y)$=orientation

$$\theta(x,y)=\tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))) \qquad \textbf{(2.9)}$$

By using these equations, orientation histogram is constructed based on the gradient orientations of sample points within a region around the keypoint. Peaks in the orientation histogram correspond to the dominant directions of the local gradients. The highest peak in the histogram is identified first. Then, the local peaks in the range of the predetermined percentage of the highest peak are also used to create keypoints. That means, if the locations with multiple peaks exist, multiple keypoints with same scale and location but different orientations are created. Figure 3.3 shows the keypoint extraction process step by step. According to the figure the keypoint candidates from the original image (a), including scales and orientations are extracted initially (b). Then, the ones with low contrast(c) and poor edge response (d) are eliminated.

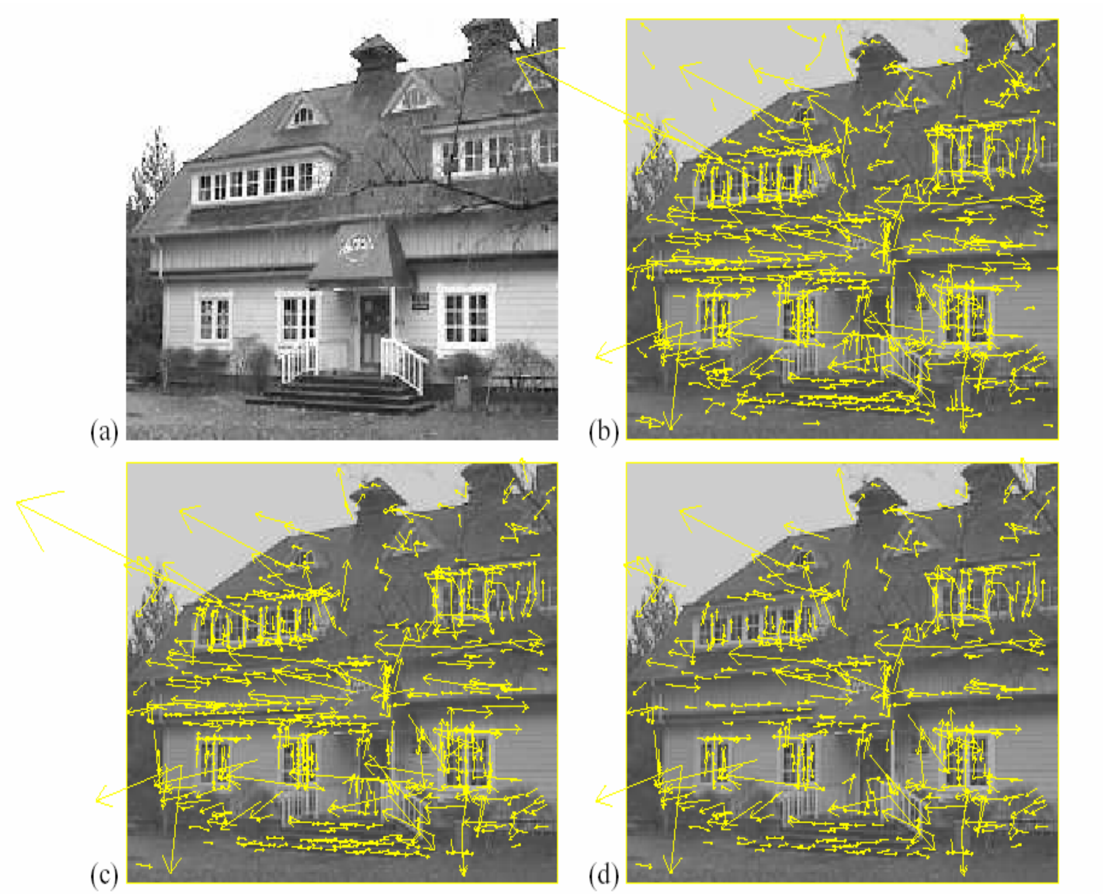


Figure 2.3: Feature Extraction Process

The extracted keypoints are used to describe the local image region. First of all, image gradient magnitudes and orientations are sampled around the keypoint location. A gaussian weighting function is used to assign a weight to sample point magnitudes. When the position of the window changes slightly, the descriptor may have big changes. Gaussian window does not let this happen. It also focuses on the sample point near the keypoint location. So the misregistrations of the points far from the center caused by affine changes can be prevented. For descriptor representation, orientation histograms are constructed over 4x4 sample regions in[13] because of the results of the efficiency measures over 16x16 sample array used in[13]. Each histogram has eight directions and length of each arrow is magnitude of that histogram entry. Another important point is to make right decisions for gradient values about the histograms or orientations they belong to. In order not to be affected by boundary conditions, the gradients should be distributed by using trilinear interpolation. The figure 2.4 shows the 8x8 sample points and 2x2 descriptor respectively.

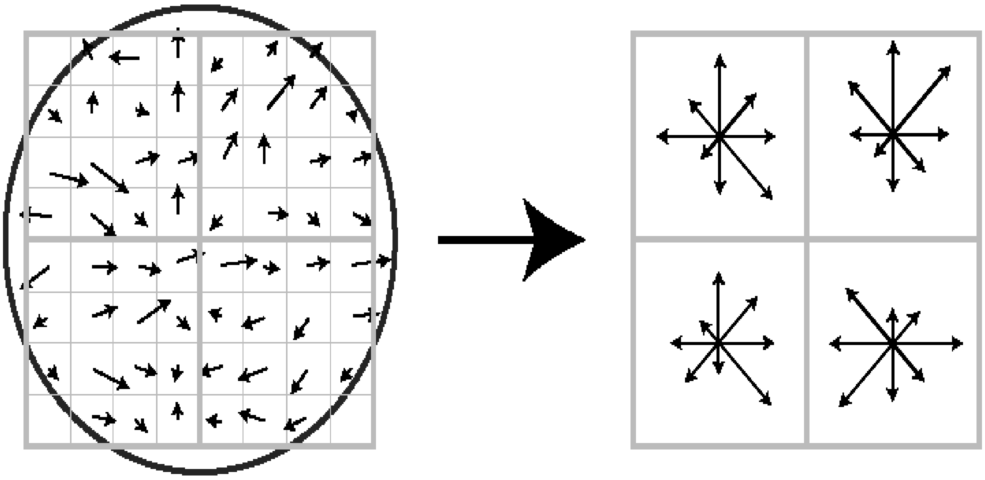
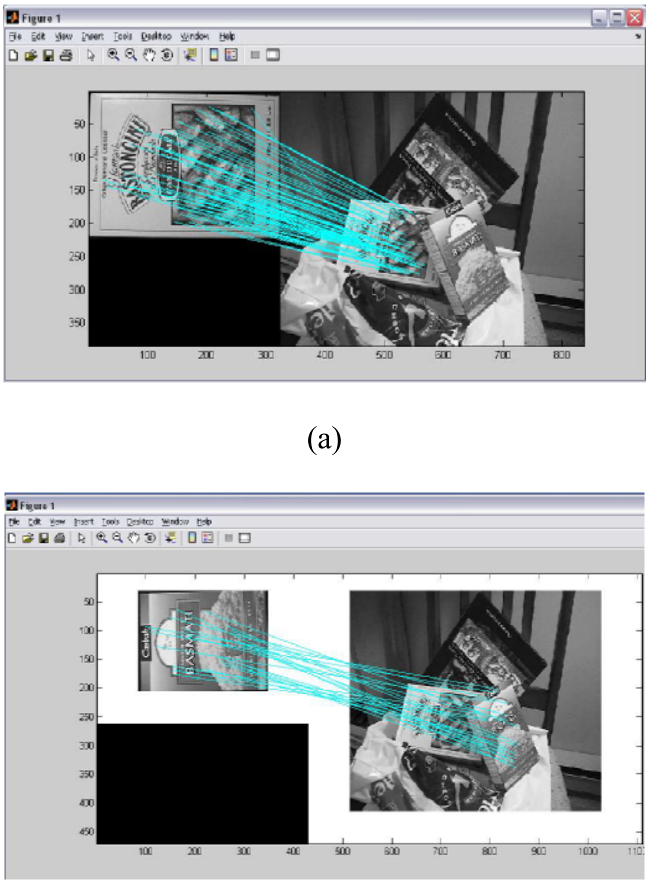


Figure 2.4: The Sample Points around Keypoint and Orientation Histograms over 2x2 Sample Regions.

The experiments on Scale invariant feature transform method shows that best results are achieved with a 4x4 array of histograms with 8 orientations that are computed from 16x16 sample array. So this Scale invariant feature transform method uses a 4x4x8=128 element feature vector for each keypoint.

After we derived the invariant keypoints, we can use them for recognizing objects. The best candidate match for each keypoint is found by identifying nearest neighbor in the database of keypoints from training images. Scale invariant feature transform algorithm extracts keypoint descriptor that is a high dimensional feature vector. It means that high amount of computation is required. So the best algorithms such as k-d tree algorithms have no efficiency over exhaustive searches to find nearest neighbor [20].

However, Best Bin First Search method is developed by modifying k-d tree algorithm, so that bins in feature space are searched in the order of their closest distance from the query location. BBF algorithm identifies the nearest neighbors with high probability and reduced amount of computation [21]. Some experimental results of Scale invariant feature transform methods are shown in the images below:



(a)

(b)

Figure 2.5: Application of Scale Invariant Feature Transform Method on Still Images

In Figure 2.5(a), 638 keypoints are found for the object image on the left, and 1021 keypoints are found for the scene image on the right. And 78 matches are found.

In Figure 2.5(b), 657 keypoints are found for the object image on the left, and 1098 keypoints are found for the scene image on the right. And 24 matches are found.

As observed from the examples in above figures, Scale invariant feature transform method is successful for still images but when Scale invariant feature transform method is used for the images segmented from the video sequences, it is not as successful as applied for still images. This may happen since the segmented images are not appropriate for the method. The images may need some additional operations to obtain more suitable images to be processed by scale invariant feature transform .method.

Figure 2.6 illustrates the application of Scale invariant feature transform for matching process of the same objects segmented from the different frames of the video:

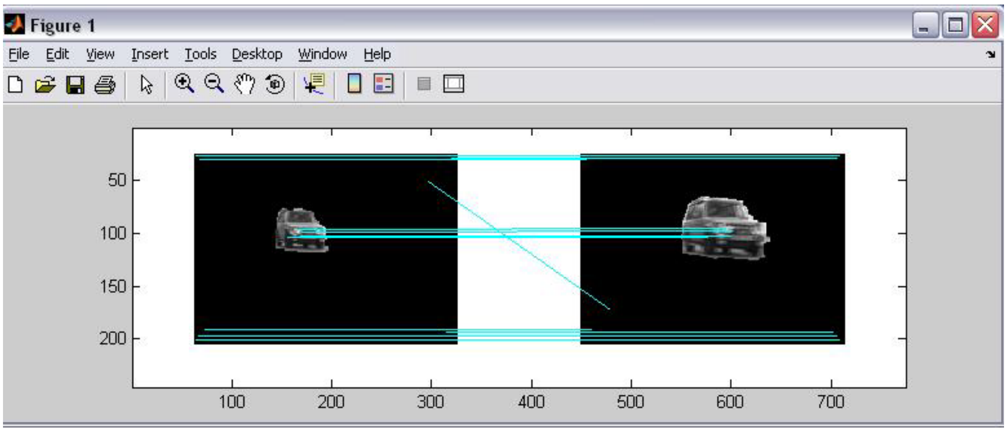


Figure 2.6: Application of Scale Invariant Feature Transform Method on Video Sequences

2.2.1.2.b. Spin Images

It is used for 3-D shape based recognition applications for concurrent recognition of multiple objects in cluttered scenes [22]. This method is then used for 2-D models in [23] for texture matching. This algorithm works on intensity domain. The spin image histogram descriptor includes the histogram of intensity values and the distance of the values from the center of the detected region. Finally smoothing and normalizing processes occur to achieve affine and illumination invariance.

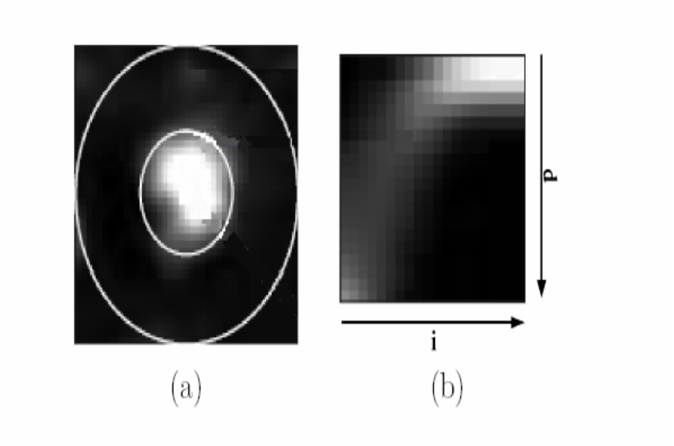


Figure 2.7: (a) Sample Region of Interest and (b) Corresponding Spin Image

2.2.1.2.c. Shape Context

This method is introduced in [24]. According to the method, two primary points of the searched object or region are detected by an edge detector. The contour points of the curve between these primary points are also sampled by an edge detector. So descriptor of the object or region is constructed by focusing on the relative positions and pair wise joint orientations of the points between the primary points. As the region or object grows, descriptor suffers from high dimensionality. Hence coarse histogram of the relative shape sample points is computed for dimension reduction. This is called shape context. Experiments through this method show that using 5 bins for radius and 12 bins for orientation introduce good results. Figure 2.8 shows the histogram pattern constructed by shape context method.

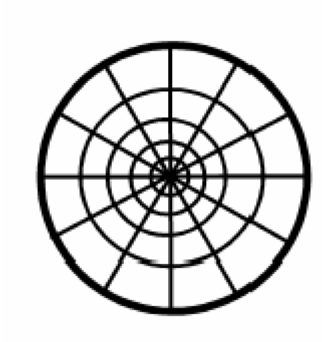


Figure 2.8: Histogram Bins Used for Shape Context

2.2.1.2.d. Locally Binary Pattern

This method is proposed by Ojala et al.[25]. This method is used to threshold intensity values for binary coding. Method uses 3x3 neighborhood for its simplest form but it can be extended to include all circular neighbors with any number of pixels by using interpolation. Method can be described by its simplest form by using 3x3 neighborhood. The neighborhood pixels are labeled by notation $p_{i,}$ that i=1,....,8. The central point is labeled by p_o. These neighborhood pixels are then signed(S) according to Equation 2.10:

$$S(p_0,p_i)=\left\{\begin{matrix} 1, \left[I(p_i-I(p_0)\right]>=0 \\ 0, \left[I(p_i-I(p_0)\right]<0 \end{matrix}\right\} \qquad \textbf{(2.10)}$$

Then, locally binary descriptor value for central point is computed by summing these signs which are weighted by a power of 2 as in Equation 2.11:

$$LBP(p_0)=\sum_{i=1}^{8}W(p_i)S(p_0,p_i)=\sum_{i=1}^{8}2^{i-1}S(p_0,p_i) \qquad \textbf{(2.11)}$$

Locally binary patterns are invariant to monotonic gray value transformations. But it is not invariant to rotation. Rotational invariance can be achieved by rotating the neighborhood points. Also, partial scale invariance can be achieved by combining the method with scale invariant detectors.

2.2.1.2.e. Filter-Based Descriptors

This type of descriptor may based on local derivative properties of the images or linear combination of the basic filters in order to have rotational invariant property. They are all rotational invariant but they may not produce efficient information against other type of distortions such as affine or scale distortions.

2.2.1.2.f. Cross-Correlation

This method estimates statistically the similarities between image regions. The descriptor only involves the intensity or color component values. These values are matched by means of similarity score. These scores are calculated by cross-correlation. The dimensionality of descriptor depends on the number of points in the region detected by region detector but if a region detector does not exist, an exhaustive search is required. So the most important disadvantage of the method is its high computational cost. It also has no invariance to any image transformations.

2.2.1.2.g. Moment Invariants

Van Gool first introduced the color and intensity moments to characterize the intensity, color and shape distributions for a region. Combination of these moments has geometric and photometric invariant property. Also combined with the affine invariant regions, it gains a powerful detector/descriptor property [26].

2.2.2. Global Approaches

This approach is based on global features that try to cover the information content of all images. Global methods may be more robust but less resistant to occlusion and clutter.

2.2.2.1. Principal Component Analysis

It is originally well-known and widely used method in statistics. But it is also used for computer vision applications. It is first introduced to computer vision by Kirby and Sirovich [27] and became popular when Turk and Pentland [28] are used the method for face recognition identifying pattern in data by focusing the similarity and differences. The most important property of the method is dimension reduction. Method reduces the dimensionality without much loss of information [29].

2.2.2.2. Linear Discriminant Analysis

The goal of the method is to make efficient classification by increasing the separation of the data. To achieve this, the classes that the data belongs to should be distant enough with respect to their variations [10].

After completing grouping task successfully, recognition process continues with matching. First we have to select a model among group of object models that is likely to match the features found in the image. Instead of trying each model in turn, indexing approach can be used. This approach involves a table that is indexed either by individual features or by small groups of them. Each table entry shows a model that could produce the corresponding feature or group. Before recognition, the table entries are created for various viewpoints of each model by analyzing the model library, by rendering each model from a sample of viewpoints, or by processing a representative set of training images. During recognition, features chosen from the image are used to index the table, and producing hypotheses about what objects are present in the image. Each hypothesis indicates a possible, match between model and image. Then this hypothesis will be tested to determine the matching quality [9].

Various forms of this basic design are implemented. Breuel index the table by using all features from images [30]. Lamdan et al. used randomly chosen subset of features [31]. Clemens and Jacobs used just features that are judged particularly likely to be derived from single objects and they tested every hypothesis retrieved from the table for a possible match [32]. Lamdan et al. treat the hypotheses as votes, and only test hyphothesis that receive the most votes [31]. Stein and Medioni uses the likely manner but they tested the hypothesis that receive some minimum number of votes[33]. Lastly Lamdan et al. assumed that all features are not used for indexing or all most voted hypothesis is not tested. They indicate an error threshold. Then repetitive testing is processed to estimate the probability of missing a model with each test and testing repeats until cumulative probability drops below that error threshold [31]. Another indexing approach developed by Basri [34] and Sengupta et al. [35] which are based on same scheme. According to that approach, models of similar objects are clustered, and each cluster is represented in the library by a single prototype. This clustering process may be repeated several times to form a model abstraction hierarchy. Recognition continues by descending this hierarchy while refining an object's identity along the way. However, a full search for a match between image and model features need only be performed at the hierarchy's top level such that at each lower level, the match result from the level above provides an advanced starting point in the search for a more complete match [28].

CHAPTER 3

OBJECT RECOGNITION USING VIDEO SEQUENCES

Recognition of objects in video can offer significant benefits for various applications such as video retrieval and video based surveillance systems used for moving objects. A lot of methods exist for recognition process. The most common for these systems is localizing the object and representing it. So we have to segment the object first and then extract the feature points. The ways to be followed for these two methods are specified by the characteristics of the application. Such that, the camera and object motion are important for video surveillance systems or focusing on faces is more important than focusing on motion for face detection systems. Detailed information about the methods is given below. [8].

3.1. Problem Definition

The system constructed here is used to detect the objects moving through the camera scene and recognize them according to the feature points. The objects are detected from grayscale image sequences taken from the stationary camera. The system first segments the moving object and then extracts the feature points. Then these feature points are compared with the ones belong to the template objects. At the end, the system tries to find the best possible matches of the object among the other templates

Detailed explanation of the steps through the recognition process is given by the following sections.

3.2. Background Subtraction

Background subtraction is important to distinguish the moving foreground from the stationary background. There are a lot of background subtraction methods but we can classify them as pixel-based and region based methods. The simplest method is to differentiate the corresponding pixels in grayscale images. If the difference is over a predetermined threshold, the pixel is treated as foreground pixel since the difference over the threshold symbolizes the change in the scene. So, determination of threshold is very important. If threshold is chosen high, it causes not to detect some changes. Also, if it is chosen small, it may cause a lot of noise. Pixel-based methods have low computational cost, since the values for only one pixel is computed for each time.

The method implemented for the thesis follows the same simple way for background subtraction and object segmentation. Figure 3.1 illustrates the steps followed here. As it is seen, both preprocessing step and background modeling step construct background subtraction process. First step is preprocessing step. This step is important for noise removal. Second step is background modeling step. It is the most important step. It involves the process of distinguishing moving pixels from the stationary ones. These stationary pixels are used to model the background initially. In the thesis, background modeling is achieved by taking the median of the array containing the pixels from each frame. Let A is an array of N consecutive frames.

A^k (i,j) is the intensity of pixel (i,j) in the k-th frame. The median of the intensity values of the same pixel location in all frames determines the intensity value of the same pixel location of background model [8].

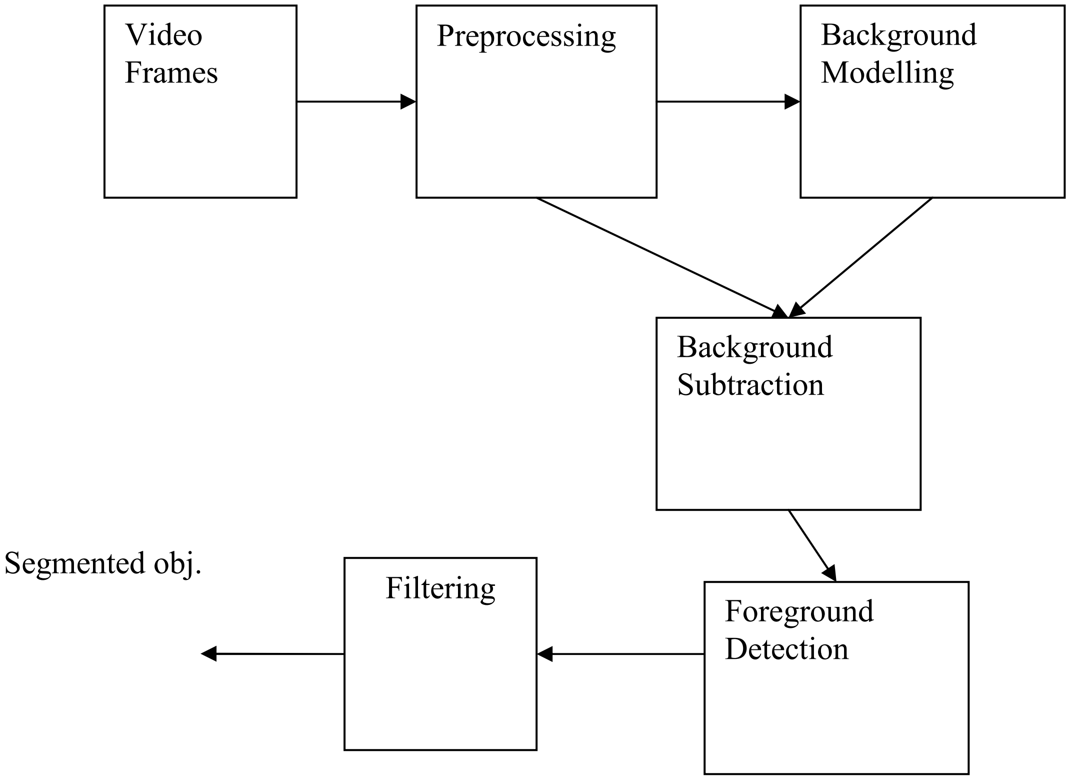


Figure 3.1: Background Subtraction and Object Segmentation Steps

3.3. Foreground Detection

After background is modeled, foreground detection process starts. After a single image is handled as background image, foreground detection is achieved by taking the difference of the background pixel and the corresponding pixel in the original imageunder analysis. Equation 3.1 shows that the difference should be over a predetermined threshold as mentioned before [8].

$$| I_t(x,y) - B_t(x,y) | > T \tag{3.1}$$

Foreground detection is not a simple task as it seems. There may remain non-stationary pixels caused by moving leaves of trees in the stationary background. These non-stationary items may be sensed by the system as foreground objects. Also false improper determination of threshold value may cause segmentation of small sized noise blobs as foreground objects. So we need to filter them out for proper segmentation. Otherwise, recognition process will be unsuccessful. In the thesis, morphological operators are used for filtering.

3.4. Filtering Noise

Mathematical morphology is used to extract the image components in order to represent skeleton, boundary, etc of an object. Mathematical morphology can be applied directly to binary images. Morphological operations can be described in terms of set operations. In the binary image :

- Foreground pixel is a point in the set
- Background pixel is not in the set
- Set operators (intersection, union, inclusion, complement,etc) can be applied to them [36].

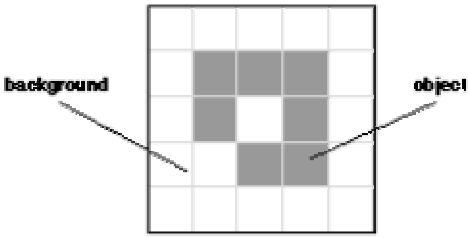


Figure 3.2: Binary Image Including Object and Background

Morphological image operations involve two sets: Image and structuring element. The structuring element used in practice is generally much smaller than the image, often a 3x3 matrix. Some of the morphological operations are given below [37]:

- Erosion reduces the size of the object by eliminating the components smaller than the structuring element. That means, areas of foreground pixels shrink in size, and holes within those areas become larger.

  Figure 3.3 illustrates the erosion operation. Let A becomes the input image and B is the structuring element with the middle point chosen as the origin. In binary image A, foreground pixels are logical 1s, and background pixels are logical 0s. To compute the erosion, we have to consider each pixel by superimposing the structuring element so that the origin of the structuring element and the input pixel from the binary image coincides. If the input pixel is foreground pixel and all its neighbors are also foreground pixels, then the pixel remains same, but if at least one neighbor is not a foreground pixel, it becomes background pixel. If the considered pixel is background pixel, it remains same.

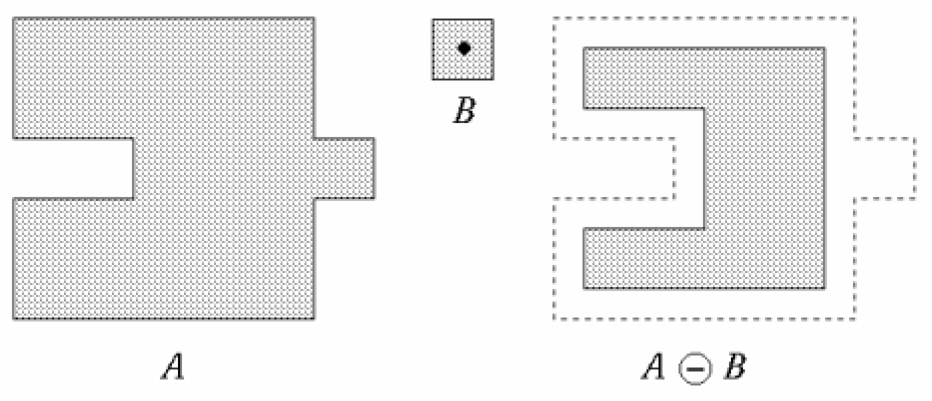


Figure 3.3: Erosion of Binary Image A with 3x3 structuring element B

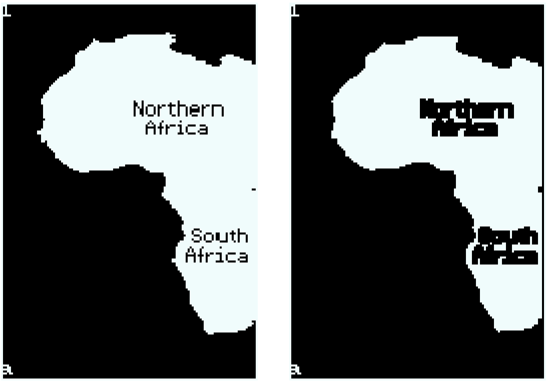


Figure 3.4: Original Image and Eroded Image

- Dilation expands the size of the foreground objects. So, background holes in the object region shrink in size.

Again we can take a 3x3 matrix for the structuring element, with the center pixel used as the origi. Each input pixel coincides with the origin considered as in erosion operation. If the pixel is a foreground pixel, it remains same. If the pixel is background pixel and none of its neighbor is a foreground pixel, then it remains as background pixel. If the pixel is a background pixel but at least one of its neighbors is foreground pixel, then it becomes foreground pixel.

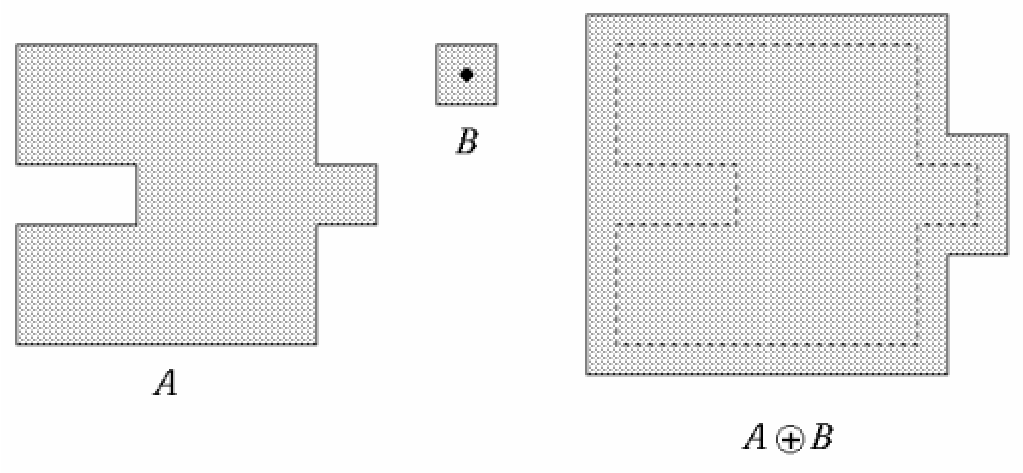


Figure 3.5: Dilation of Binary Image A with 3x3 Structuring Element B

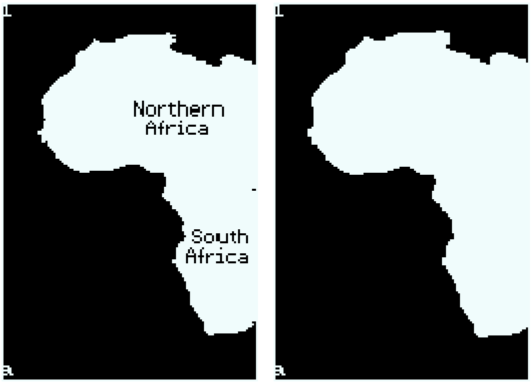


Figure 3.6: Original Image and Dilated Image

- Opening operation is composition of the operations described above. For opening, erosion is applied on the image in order to remove smaller features according to the structuring element. Then the eroded image is dilated to handle smother edges. This operation may be used to remove noise pixels.

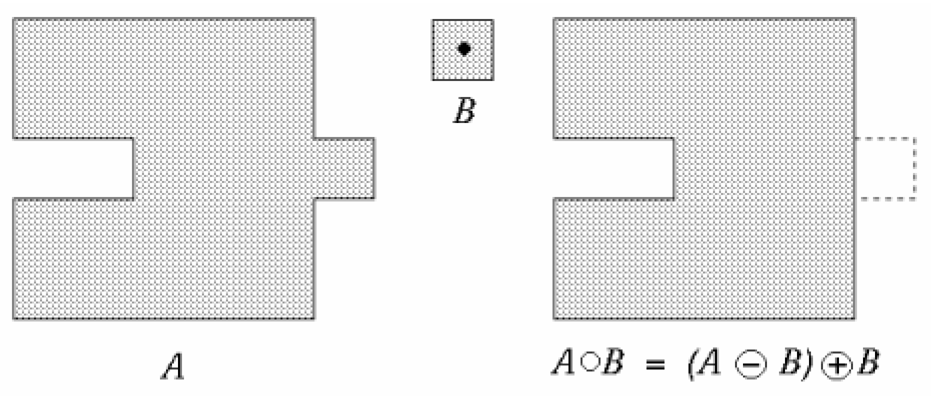


Figure 3.7: Opening Operation

- The closing, like opening, is also a composite operator. The closing of set A by set B is achieved by first dilating of set A by B, then eroding the resulting set by B as in Figure 3.8.

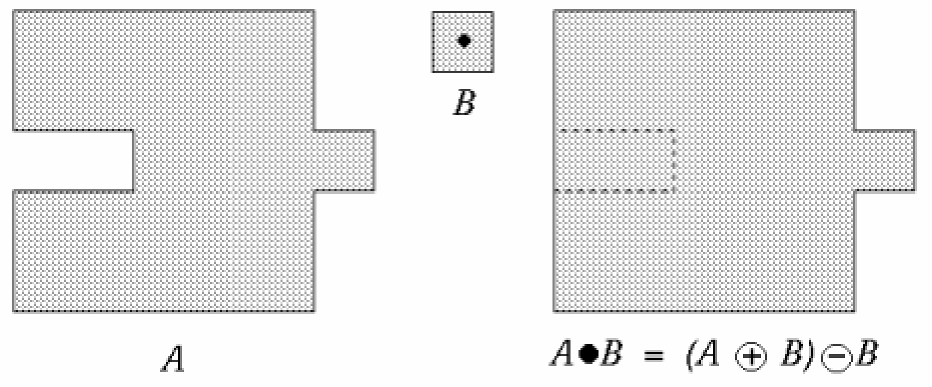


Figure 3.8: Closing operation

The operations followed in thesis for filtering starts by first Gaussian filtering of the frames for reducing noise as preprocessing step. Then background modeling process is achieved as mentioned in section 3.2. And a single background image is handled. Figure 3.9 shows the background.

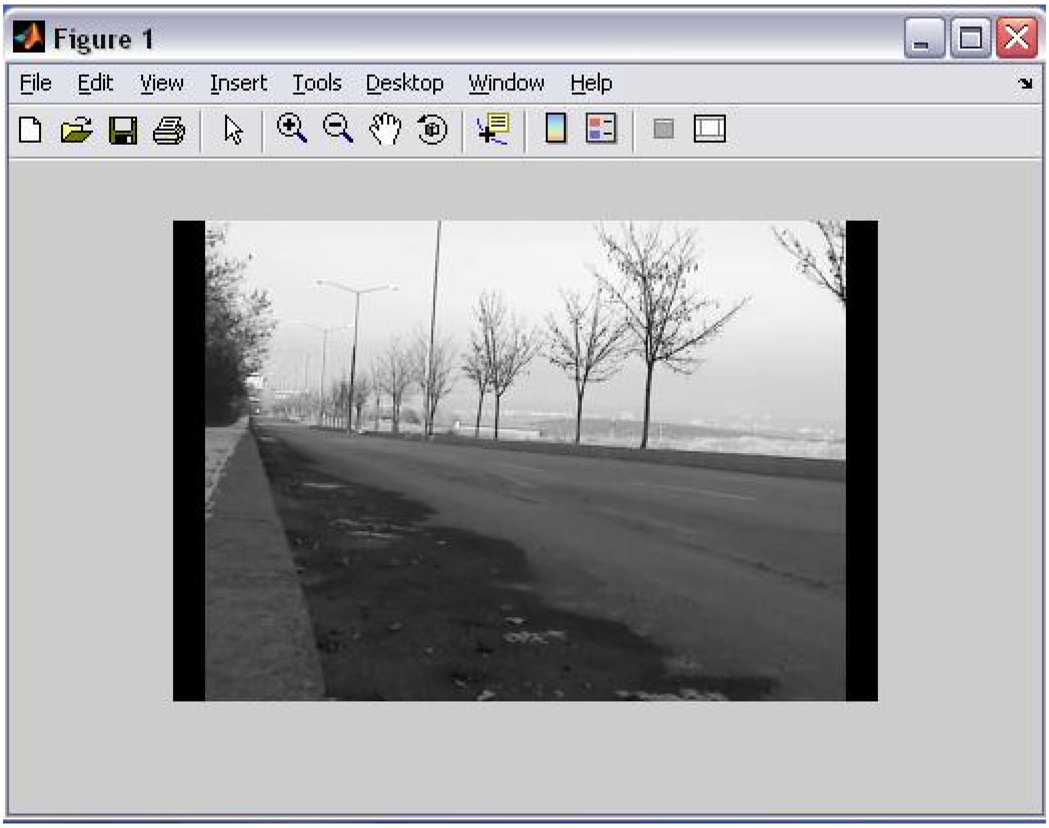


Figure 3.9: Background Image

After the background is handled, this background model is subtracted from the image under analysis. By this process, non-stationary foreground object is detected and segmented from the image. Figure 3.10 illustrates the binary image of initially segmented foreground [8].

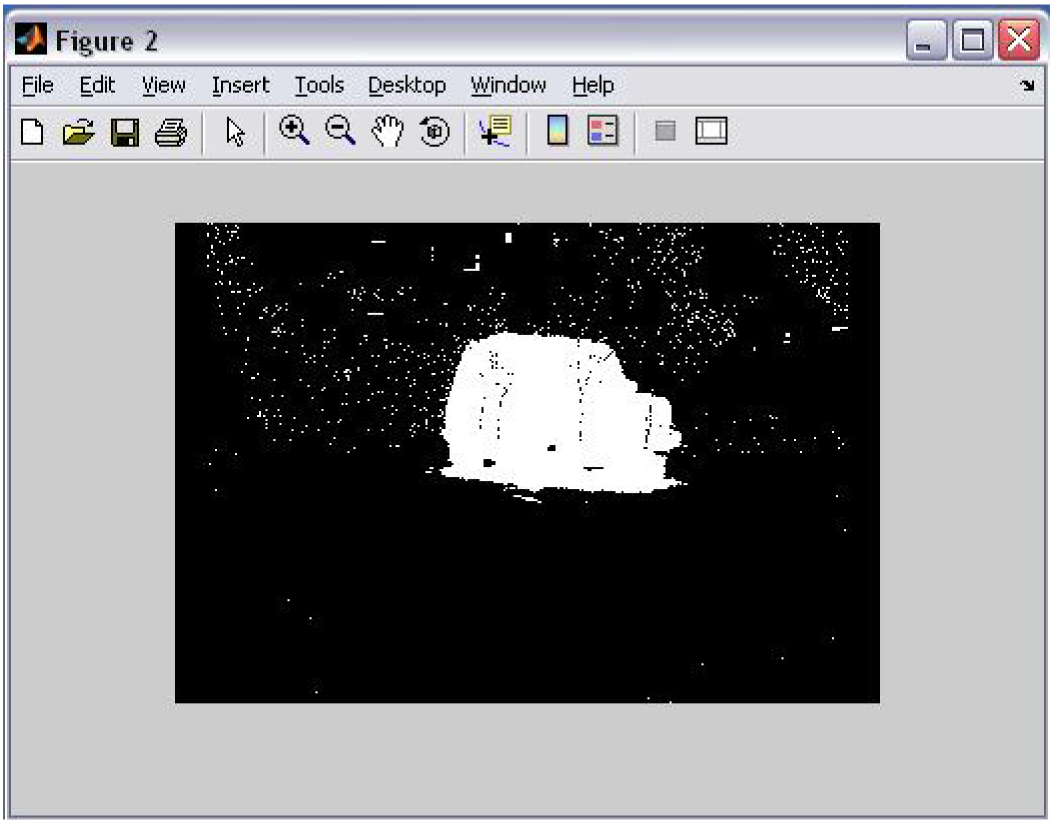


Figure 3.10: Binary Image of Foreground Object

As it is seen in Figure 3.10, there exist some smaller objects extracted as foreground object. These noises may be caused by the non-stationary background objects such as leaves moving because of the wind. The next step of the application is removing these particles. We use morphological operators for this step. The image in Figure 3.10 is subject to opening operation first. This opening operation smoothes the contours breaks narrow isthmuses and eliminates little knobs. Figure 3.11 illustrates the result of that opening operation.

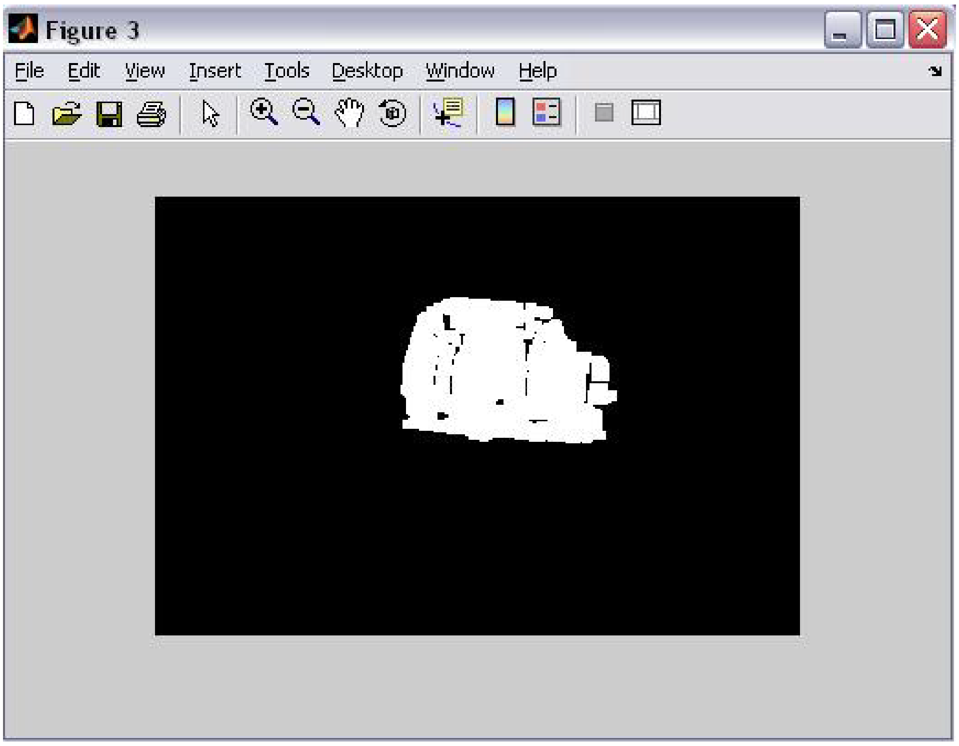


Figure 3.11: Foreground Object after Opening Operation

As it is seen in Figure 3.11, there still exist some breaks on the segmented object. To remove these, closing operation is applied to the morphologically opened image. Figure 3.12 shows the resultant image.

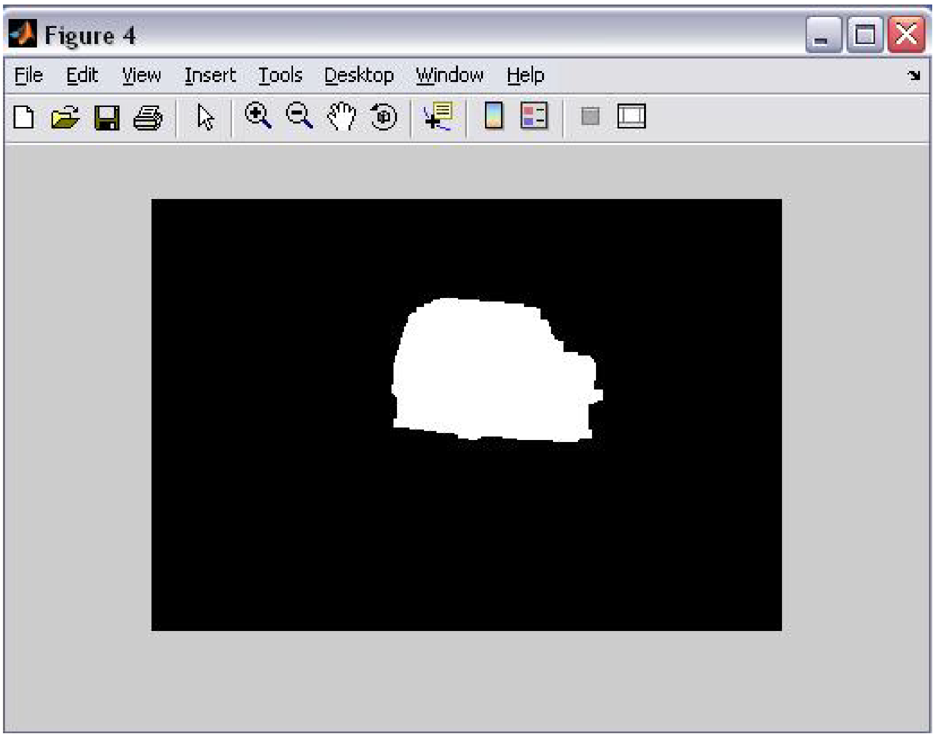


Figure 3.12: Foreground Object after Closing Operation

After all these morphological operations applied, binary image of foreground object is acquired. At last, foreground object is converted to grayscale image by using the original gray values of the image under analysis. Figure 3.13 shows the last form of the segmented object.

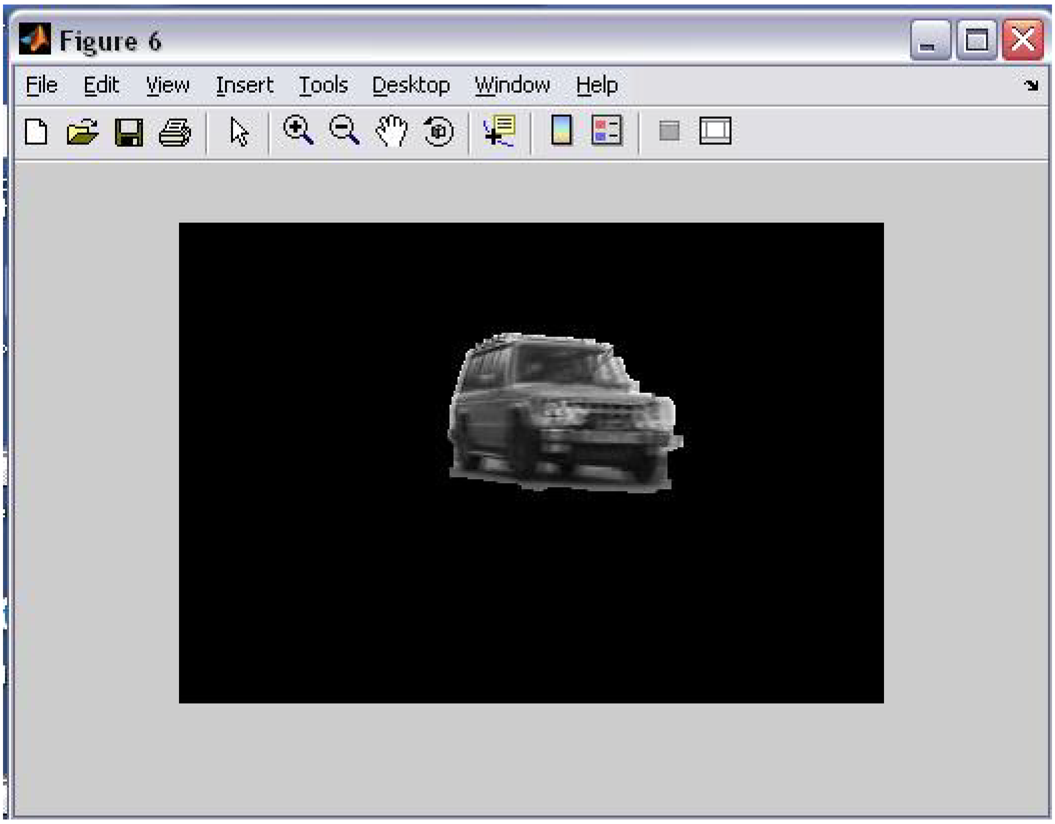


Figure 3.13: Grayscale Image of Foreground Object

3.5. Edge Matching

Object recognition process includes various complicated steps. Matching process is very important among all. Since matching process is used for different applications, it has to be robust against the parameters changing according to the applications.

Some types of applications are: [38]

- Image to image matching
- Template to image matching
- Image to symbolic image matching

Stereo image matching and motion detection are the examples of image to image matching. Template to image matching is used for specific object recognition tasks. And registration of aerial images can be example of image to symbolic image matching.

Matching algorithms differ according to the features used for matching process. One type of algorithm such as correlation may use only pixel values. But this type of algorithm is very sensitive to changes. For example matching process of same images in different illumination conditions may produce false matches. The other type of matching algorithm focuses on the edges and corners. The last type includes high level matching features such as identified objects or relations. Graph-theoretic methods can be shown as example of this kind. Unlike the methods considering pixel values, high-level methods are very insensitive to changes but extracting high-level features is not an easy task.

The method used in the thesis uses the edge points for matching on gray level images. It also gives the matching measure to interpret the matching process easily. The other most important problem that is figured out here is the transformation of the image. The image may be distorted by rotation, translation, etc. The matching algorithm should also identify these transformations. The steps followed for the application are explained by the following sections. [38]:

3.5.1. Edge Extraction

Edges are used as feature points in the thesis. The edges of two images are used for matching. One of the image is called pre-distance image, the other one is called pre-polygon image. Pre-distance image is used to acquire an edge image in order to use for the distance transformation.

For this step, the pre-distance image is determined first. Then the image is subject to an edge detection algorithm. All available edge detection operators in MATLAB are used and it is observed that canny edge detector gives the best results of matching measures for the same set of examples.

When edges are extracted, a binary image is formed. The value of the edge locations is logical 1 and the non-edge pixel locations have value of logical 0.

3.5.2. Distance Transformation

Distance transform is used to identify the distances of non-edge pixels to the nearest edge pixels. To calculate the distances, we may use Euclidean distance method but it causes high computational load. Instead of Euclidean distance method, we may use its approximation. Distance transform process is very important for computation of the matching measure.

Distance transformation identifies global distances. In order to achieve this , local distances are considered in iteration. 3x3 neighborhood is used to evaluate the distance of the pixels to approximate Euclidean distance. In 3x3 neighborhood, the distances between horizontal, vertical and diagonal neighbors are considered. There are more than one algorithm for distance transformation. These algorithms differ according to the distance units.

- 2-3 Distance Transform: uses 2 and 3 respectively as distances. This algorithm shows %13 difference from the Euclidean distance.

- 3-4 Distance Transform: uses 3 and 4 respectively as distances. The maximum difference from Euclidean distance is %8. So 3-4 Distance Transform is the most suitable one.

Distance Transformation is computed by following steps:

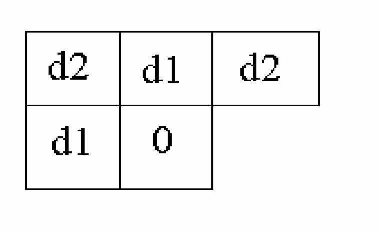
- The edge image is constructed
- In the binary edge image, the edge pixels are set to 0 and non-edge pixels are set to infinity.
- d1 and d2 are added to the pixel values in the distance map as in Figure 3.14 and the new value of the pixel is the minimum of the five sums.

d1	d2	d1
d2	$V_{i,j}$	d2
d1	d2	d1

Figure 3.14: 3x3 Neighborhood for Distance Transform. d1=4 and d2 =3 for 3-4 Distance Transform algorithm

For sequential 3-4 DT algorithm we make 2 passes over the image :

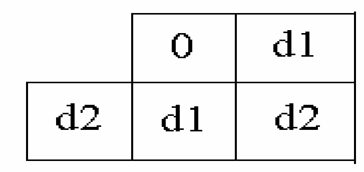
- One pass from left to right and top to bottom(forward)



d2	d1	d2
d1	0	

Figure 3.15: Forward Passing Mask

- One pass from right to left and from bottom to top(backward)



	0	d1
d2	d1	d2

Figure 3.16: Backward Passing Mask

The sequential algorithm is below [2]:

Forward:

for i=2,……,rows-1 do

for j=2,…….,columns -1do

$$v_{i,j} = \min(v_{i-1,j-1} + 4, v_{i-1,j} + 3, v_{i-1,j+1} + 4, v_{i,j-1} + 3, v_{i,j}$$

Backward:

for i=rows-1,…..,1 do

for j= columns-1,…..,1 do

$$v_{i,j} = \min(v_{i,j}, v_{i,j+1} + 3, v_{i+1,j-1} + 4, v_{i+1,j} + 3, v_{i+1,j+1} + 4)$$

3.5.3. Polygon Definition

In this step, the edges are extracted from pre-polygon image. They are extracted by the same edge extraction operator used for the pre-distance image and the edge image of the pre-polygon image is acquired. Then the coordinate pairs of the edge pixels are listed. Normally, all of these coordinates are not used for matching. We do not need all edges. Instead, we choose coordinate pairs according to some criteria. For example we may need the shape of the object in the image so we use only coordinates of the contour points. For the application in the thesis, the threshold for canny edge detection value is determined manually to eliminate unnecessary edges inside and keep as much contour points as possible.

These chosen points among all edge pixels form a polygon. That polygon is superimposed on the distance image. In order to understand if two images are matched, we calculate the correspondence of the images by averaging the pixel values in the distance image that the polygon hits. This matching measure is called edge distance. As edge distance minimizes, the matching becomes better. So the perfect fit between two images occurs when edge distance is zero.

3.5.4. Polygon Transformation

The polygon may be tried to be searched in a pre-distance image that is geometrically distorted by parametric transformation equations such as translation and rotation. So, the polygon coordinates should be recomputed according to the Equation 3.2 and the new coordinates should be acquired in order to be projected to the pre-distance image. In equation 3.2, (x,y) is the polygon coordinates, (X,Y) is the new position of the polygon in the distance image, c_x and c_y are the translation parameters in X and Y direction and θ is the rotation angle.

$$X= c_x + \cos(\theta)x - \sin(\theta)y \qquad Y= c_{xy}+ \sin(\theta)x + \cos(\theta)y \qquad \textbf{(3.2)}$$

In the thesis, only the translation parameters are used. The step-length of c_x and c_y is one pixel. The polygon position is changing for one pixel at each iteration to find the optimal position. The edge distance is computed for new coordinates at each iteration and the minimum edge distance value among all gives the optimal position of the polygon.

3.5.5. Matching Measure

In order to find the optimal position of geometrically distorted polygon in the pre-distance image, the global minimum of a multidimensional function, where each dimension corresponds to a parameter in the transformation equations, should be evaluated. That means, after the position of the polygon is transformed by predetermined step lengths for each transformation parameter, in order to find the optimal position of the polygon in the pre-distance image, edge distance at each step length is evaluated to find the minimum edge distance. The coordinates of the global minimum carries information about the location of the polygon in the pre-distance image. For example, according to the application in this thesis, the coordinates of the two dimensional matrix which the global minimum is found can be interpreted as the shifting value of the polygon in the pre-distance image.

According to [38] the root mean square average measurement is the most suitable average computation since it gives the fewer false minima. So it is used to compute the matching measure by Equation 3.3.

$$\frac{1}{3}\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} v_i\right)} \qquad \textbf{(3.3)}$$

The performance of the implementation can be improved by decreasing the computational load. Instead of taking the squares of the distance values (v_i), taking absolute values decreases the source requirement of the matching process .

3.6. Experimental Results of Matching

Some still images are used in the thesis to show the results of the edge matching applications. These images are taken from the Amsterdam Library of Object Images which is a color image collection of one-thousand small objects, recorded for scientific purposes. In order to capture the sensory variation in object recordings, viewing angle, illumination angle, and illumination color for each object are systematically varied. Figure 3.17 shows these variations.

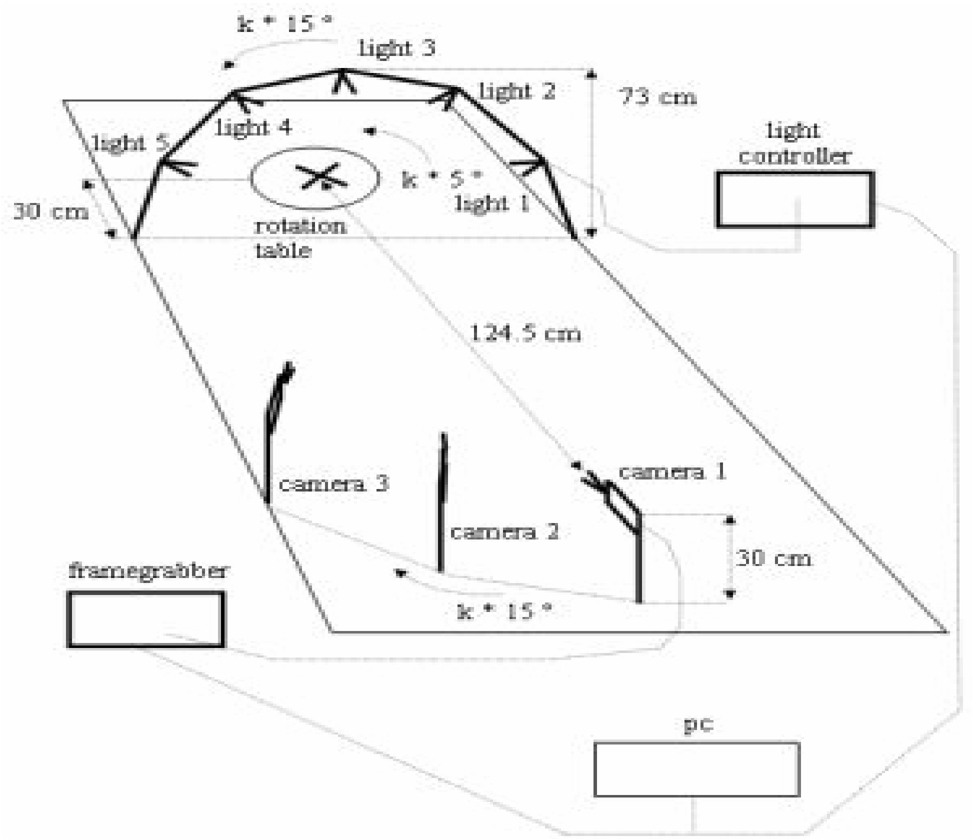


Figure 3.17: Illustration of Technical Details of Image Capturing for Amsterdam Library of Object Images

According to the technical details about the construction of Amsterdam Library of Object Images taken from the website of Information and Communication Department of University of Amsterdam, there are 24 different illumination direction. These variations are formed by with only one out of five lights turned on, yielding five different illumination angles (conditions l1-l5). By switching the camera, and turning the stage towards that camera, the illumination bow is virtually turned by 15 (camera c2) and 30 degrees (camera c3), respectively. Hence, the aspect of the objects viewed by each camera is identical, but light direction has shifted by 15 and 30 degrees in azimuth. In total, this results in 15 different illumination angles. Furthermore, combinations of lights were used to illuminate the object. Turning on two lights at the sides of the object yielded an oblique illumination from right (condition l6) and left (condition l7). Turning on all lights (condition l8) yields a sort of hemispherical illumination, although restricted to a more narrow illumination sector than true hemisphere. In this way, a total of 24 different illumination conditions were generated, conditions c(1..3)l(1..8) [39].

Figure 3.18 illustrates the toy duck image taken from the Amsterdam Library of Object Images. Figure 3.18(a) is the original image used as pre-distance image for the application. Figure 3.18(b) shows the result of edge extraction process over the pre-distance image. As mentioned before, edge extraction process is followed by distance transformation of the edge image in order to give the non-edge pixels a measure of the distance to nearest edge pixels. Then pre-polygon image is processed for edge extraction. The same image with pre-distance image is selected here as pre-polygon image. Edge extraction process is different from the previous one since all of the edges are not used. Some edges are selected according to some criteria. The criterion is a threshold value here. The threshold value is selected in order to acquire more contour points as possible. Figure 3.18(c) shows the polygon including these selected edges. At last Figure 3.18(d) shows the result of the matching process of the polygon and the distance image. The edge distance computed here is 0.0263 that means nearly zero and perfect matching.

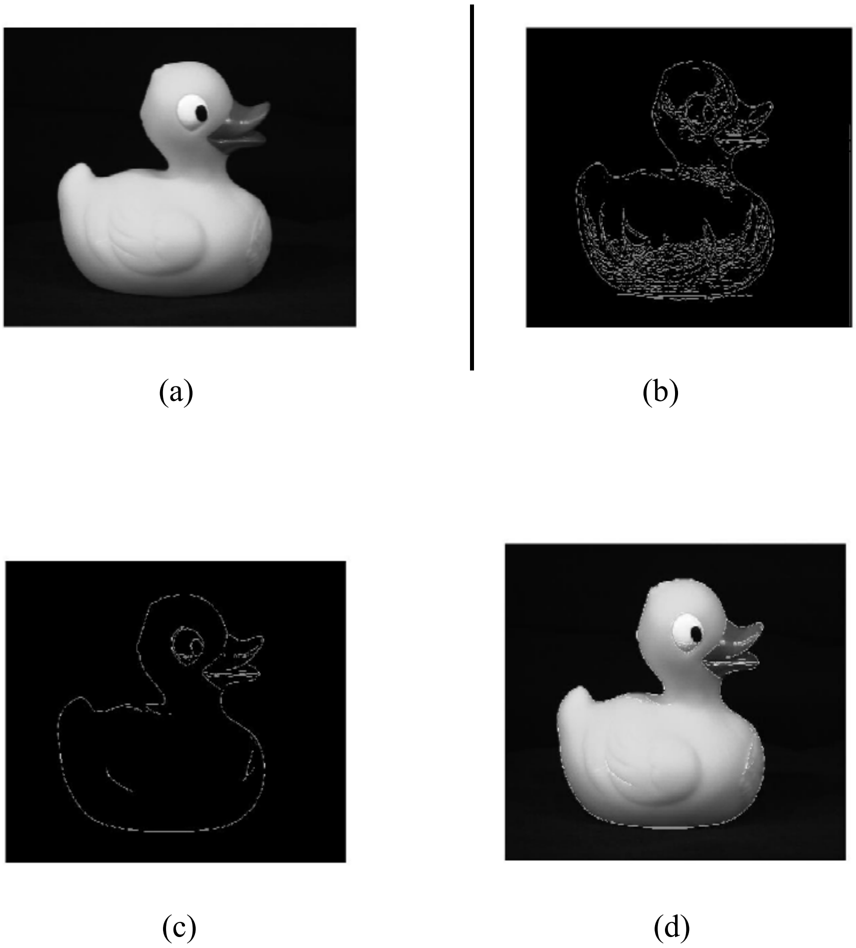
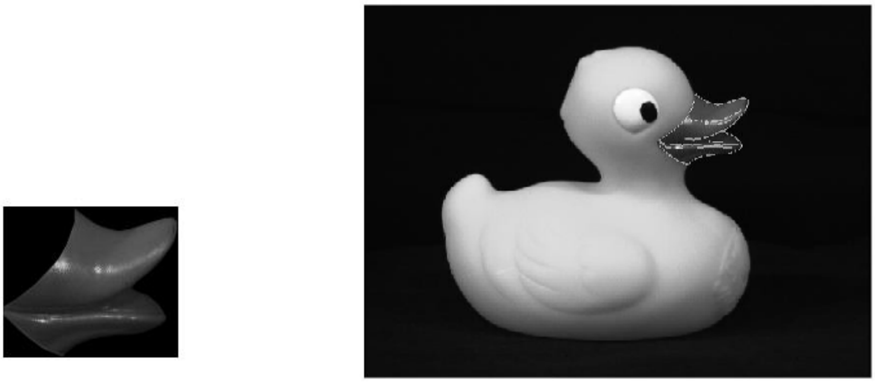


Figure 3.18: (a) Original Image (b) Extracted Edges of Original Image (c) Polygon (d) Result of Matching Process of Original Image and Polygon

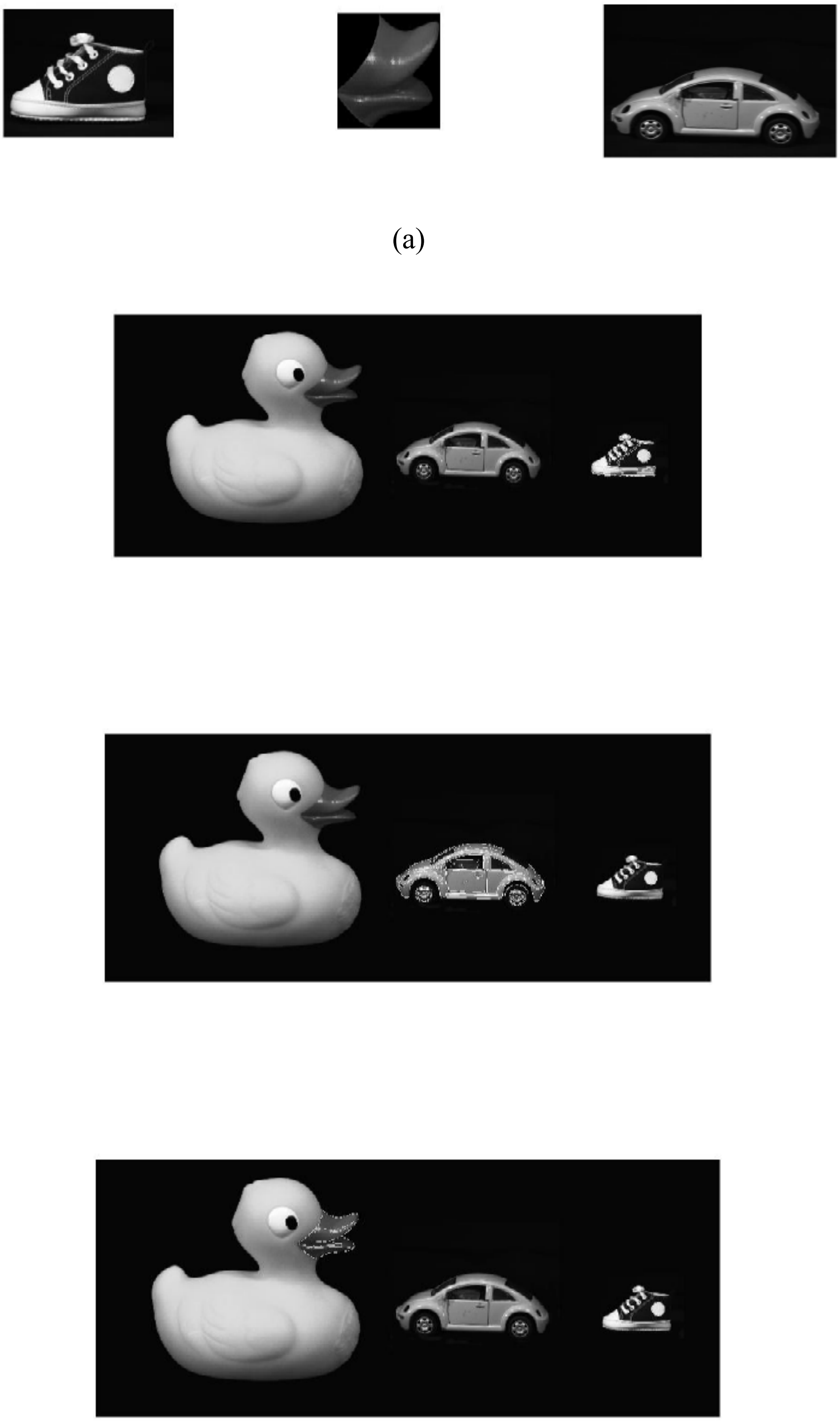
The example given in Figure 3.18 is used the image itself as polygon. Figure 3.19 shows an example which is using a part of the pre-distance image as polygon. The same toy duck image in Figure 3.18(a) is used as pre-distance image again and the beak of the duck is extracted from the pre-distance image by help of Photoshop. Then that extracted part is saved as a new image and used as pre-polygon image to be searched in the pre-distance image. Firstly, the polygon extracted from the pre-polygon image and the edge coordinates are listed. Since the polygon is only a part of the pre-distance image, the optimal position of the polygon in the pre-distance image should be found. The polygon is superimposed on the distance image and edge distance is computed but this time it is computed for every possible translational polygon position in the distance image. As a result of this computation, a 2-D matrix, with a dimension representing the translation in x-direction and the other dimension representing the translation in y-direction, includes all the matching measures. The location of the minimum matching measure shows how much the polygon is translated in the original image. Figure 3.19(c) shows the exact position of the polygon in the original image.



(a) (b)

Figure 3.19: (a) Pre-polygon Image (b) Result of the Matching Process

In Figure 3.20, same process is illustrated but a scene image is constructed by using the images of three objects: a shoe, a car and a duck. Then these images are used as a polygon to be searched in the scene. Figure 3.20(a) shows the pre-polygon images. And Figure 3.20(b) shows the resultant images of matching process.



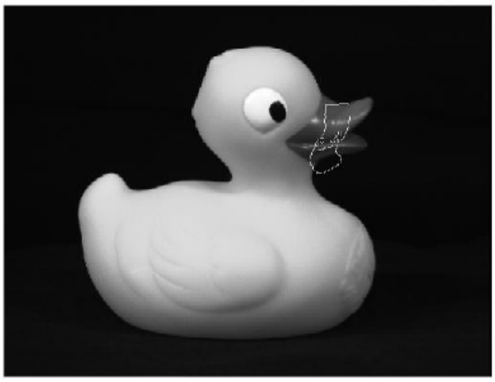
(a)

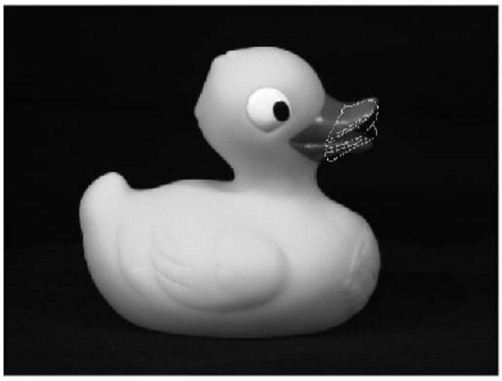
(b)

Figure 3.20: (a) Different Pre-polygon Images (b) Results of the Matching Process

Figure 3.21 shows the success of matching of the different pre-polygon images of the same object. Figures 3.21(a) shows other beak images extracted from different duck images with the same scale of the pre-distance image. The application finds the location of the polygon in the pre-distance image nearly true. Since the shape of other polygons are different than the one in the pre-distance image, it does not fit perfectly.

(a)





(b)

Figure 3.21: (a) Different Beak Images Used as Polygons (b) Matching Results

The last illustration shows the robustness of the edge matching application against illumination changes. As mentioned before, there are 24 different illumination directions. These variations are acquired by means of 5 different lights and 3 cameras located at the different angles. In this way, a total of 24 different illumination conditions were generated, conditions c(1..3)l(1..8) [39]. 1 out of 24 illumination angle affects the matching process and the polygon is located wrongly in the image. The reason of this misregistration is the illumşnation angle that affects the edge extraction process. Difference in the illumination condition causes the edges in the body of the duck to be matched with the edges of the peak. The other illumination angles does not affect the process badly. So the application is robust against illumination. Figure 3.22 shows some successful results of the matching process for the pre-distance images with different illumination angles.

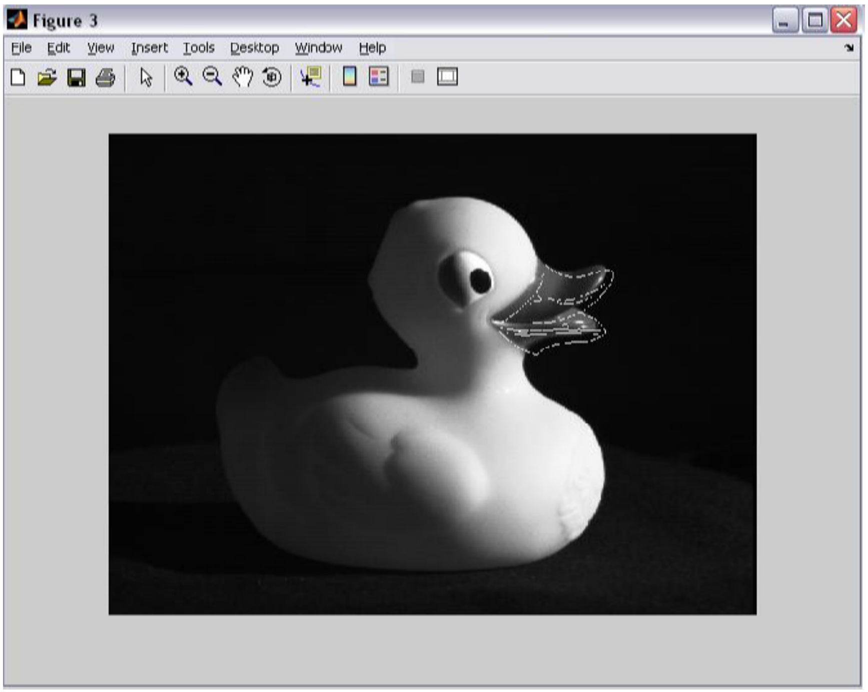


Figure 3
File Edit View Insert Tools Desktop Window Help

(a)

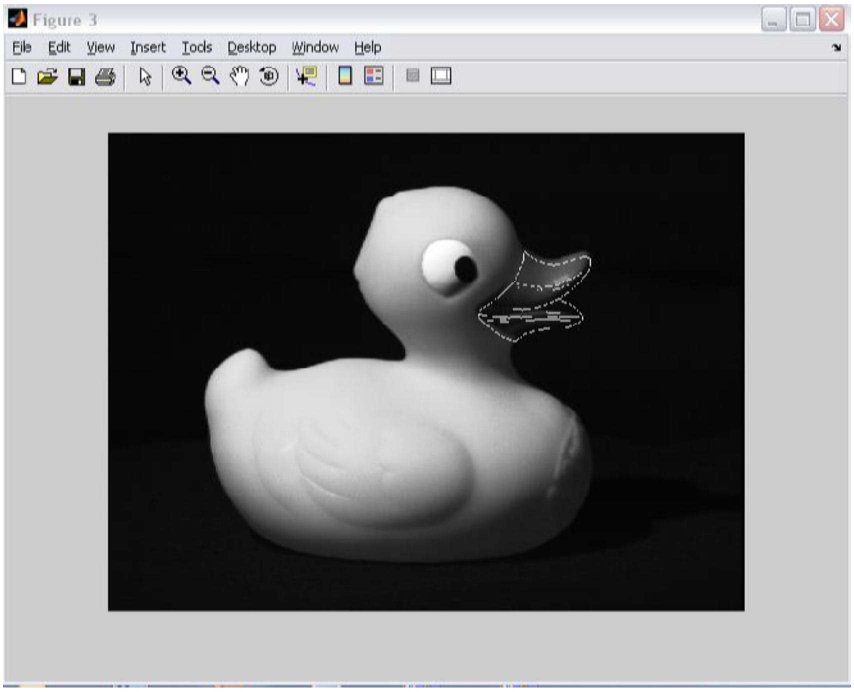
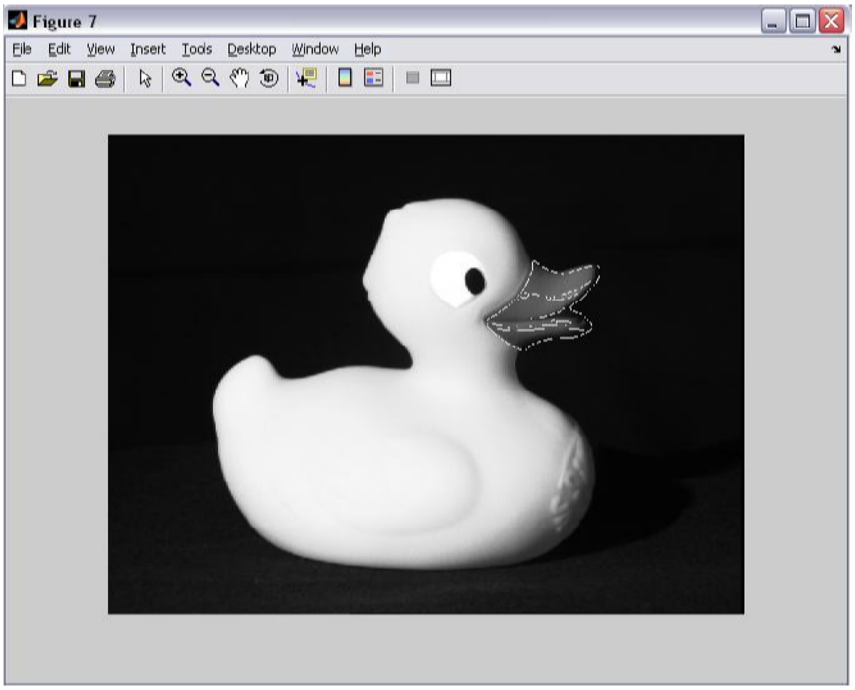


Figure 3
File Edit View Insert Tools Desktop Window Help

(b)



(c)

Figure 3.22: (a) l1c3 (b) l4c1 (c) l5c3

CHAPTER 4

CONCLUSION AND FUTURE WORK

The method described in the thesis focuses on recognition of specific objects in video scenes. Since we use cameras frequently in our daily lives, these kinds of applications become widespread. These applications include homeland security, surveillance systems, video retrieval from movie achieves, etc. In this study, I try to find some solutions for object segmentation to distinguish moving foreground from stationary background. Mathematical morphology is used to establish finer results from segmentation process. Morphological operators are used to eliminate noise and some moving particles in the background. But there are some more problems to be solved as future work. One of them is the removing shadows. Shadows are formed by projection of the object in the direction of incident light. This situation misleads the system by illustrating the shadow as a part of the object and the original shape of the object is changed. The improper object segmentation causes the unsuccessful recognition process.

In the thesis, only the translational parameters are considered for transformation equation but there are some other parameters. Rotation parameters also change the position of the polygon as in translational parameters. The application in the thesis can also be improved to work successfully for also rotational transformations. The coordinates of the polygon can be recomputed according to the Equation 3.2 for rotational transformation. For the applications in the thesis, the polygons and the images from same scales are used but scaling is also another geometrical deformation which changes the size of the image. In order to achieve matching process for images in varying scale, we can use fixed size polygon and variable size pre-distance images. By computing the edge distance for each scale, the appropriate scale of the polygon can be found. Image pyramid approach is suitable for this application.

This application improved for the thesis consumes large portions of CPU time. It is normal because the images and especially videos include large volumes of information. In order to decrease the computational load image pyramids can also be used.

.

REFERENCES

[1] **TÜRKOĞLU, İ.** (2003), *Örüntü Tanıma Ders Notları*, Elektronik ve Bilgisayar Eğitimi Bölümü, Fırat Üniversitesi, Elazığ.

[2] **STEGER, C**. (2002) *Occlusion, Clutter, and Illumination Invariant Object Recognition,* International Archives of Photogrammetry and Remote Sensing, 345-350, Vol. 34(3A).

[3] **LITER, J.C., BULTHOFF, H.H.** (1996), *Introduction to Object Recognition*, Report no: 43, Max-Planck-Institut für biologische Kybernetik, Germany.

[4] **QHUSRO, A.A.M.** (2004), *Invariant Object Recognition*, Msc.Thesis, King Fahd University, Saudi Arabia.

[5] **DIPLAROS, A.** et. al. (2006) *Combining Color and Shape Information for Illumination-Viewpoint Invariant Object Recognition*, IEEE Transactions on Image Processing, 1–11. Vol. 15.

[6] **HAHNEL, D.** et al. (2004) Color and Texture Features for Person Recognition, *IEEE International Joint Conference on Neural Networks*, 647-652. Vol. 1.

[7] **ROCHA, L.** et. al. (2002) Image Moments-Based Structuring and Tracking of Objects, *SIBGRAPI '02: Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing (Washington, DC, USA), IEEE Computer Society*, 99–105.

[8] **PATI, N.** (2007) *Occlusion Tolerant Object Recognition Methods for Video Surveillance and Tracking of Moving Civilian Vehicles*, unpublished Msc.Thesis, University Of North Texas.

[9] POPE, A.R. (1994), *Model-Based Object Recognition: A Survey of Recent Research*, Report no: 94–04, University of British Columbia.

[10] ROTH, P.M., WINTER, M. (2008), *Survey of Appearance-Based Methods for Object Recognition*, Report no: ICG-TR-01/08, Graz University of Technology, Austria.

[11] HARRIS, C., STEPHENS M. (1988) A Combined Corner and Edge Detector, *Proc. of Alvey Vision Conference,* 147-151.

[12] MIKOLAJCZYK, K. et.al. (2005) A Comparison of Affine Region Detectors, *Intern. Journal of Computer Vision*, 43-72. Vol. 65(1-2).

[13] LOWE, D.G. (2004) Distinctive Image Features from Scale-Invariant Keypoints, *Intern. Journal of Computer Vision*, 91-110. Vol. 60.

[14] MATAS, J. (2002) Robust Wide Baseline Stereo from Maximally Stable Extremal Regions, *Proc. British Machine Vision Conf.*, 384-393. Vol. 1.

[15] KADIR, T., MRADY M. (2003) Scale Saliency : A Novel Approach to Salient Feature and Scale Selection, *Intern. Conf. on Visual Information Engineering*, 25-28.

[16] KADIR, T. et al. (2004) An Affine Invariant Salient Region Detector, *Proc.European Conf. on Computer Vision*, 228-241

[17] KOENDERINK, J.J. (1984) The Structure of Images, *Biological Cybernetics*, 363-396. Vol. 50.

[18] LINDEBERG, T. (1994) Scale-Space Theory:A Basic Tool for Analysing Structures at Different Scales, *Journal of Applied Statistics*, 224-270. Vol. 21(2).

[19] SCHMID, C., MOHR, R. (1997) Local Gray Value Invariants for Image Retrieval*, IEEE Trans. on Pattern Analysis and Machine Intelligence*, 530-534, Vol. 19(5).

[20] **TURITZIN, M., HUI, A., GUSTAVSSON, C.** (2004), Interim Report: Improving Scale Invariant Feature Transform Features, Stanford University, Stanford.

[21] **LOWE, D.G.** (1999) Object Recognition from Local Scale-Invariant Features, *International Conference on Computer Vision*, 1150-1157.

[22] **JOHNSON A.E., HEBERT, M.** (1999) Using Spin-Images for Efficient Multiple Model Recognition in Cluttered 3-D Scenes, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 433-449. Vol. 21(5).

[23] **LAZEBNIK, S., SCHMID, C., PONCE, J.** (2003) A Sparse Texture Representation Using Affine-Invariant Regions, *Proc. IEEE Conf. On Computer Vision and Pattern Recognition*, 319-324, Vol. 2.

[24] **BELONGIE, S., MALIK, J., PUZICHA, J.** (2002) Shape Matching And Object Recognition Using Shape Contexts, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 509-522. Vol. 24.

[25] **OJALA, T., PIETIKAINEN, M., HARWOOD, D.** (1996) A Comparative Study of Texture Measures with Classiffication Based on Featured Distributions, *Pattern Recognition*, 51-59. Vol. 29(1).

[26] **GOOL, L.V., MOONS, T., UNGUREANU, D.** (1996) Affine/ Photometric Invariants for Planar Intensity Patterns, *Proc. European Conf. On Computer Vision*, 642-651, Vol. 1.

[27] **KIRBY, M., SIROVICH, L.** (1990) Application Of The Karhunenloeve Procedure for the Characterization of Human Faces, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 103-108. Vol. 12(1).

[28] **TURK, M., PENTLAND, A**. (1991) Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, 71-86. Vol. 3(1).

[29] **SMITH, L.I.** (2002), *A Tutorial on Principal Component Analysis,* Cornell University, USA.

[30] **BREUEL, T.M.** (1990), *Indexing for Visual Recognition from a Large Model Base*, Report No: AIM-1108, Massachusetts Institute of Technology, MA, USA.

[31] LAMDAN, Y., SCHWARTZ, J., WOLFSON, H. (1988) Object Recognition by Affine Invariant Matching, *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 335–344.

[32] CLEMENS, D., JACOBS, D. (1991) Model Group Indexing for Recognition, *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 4–9

[33] STEIN, F., MEDIONI, G. (1992) Structural Indexing: Efficient 2-D Object Recognition, *IEEE Trans. Patt. Anal. Machine Intell.*, 1198–1204. Vol. 14(12).

[34] BASRI, R. (1993) Recognition by Prototypes*, Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 161–167.

[35] SENGUPTA, K., BOYER, K.L. (1993) Information Theoretic Clustering of Large Structural Model Bases, *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.* , 174–179.

[36] www.ph.tn.tudelft.nl

[37] www.inf.u-szeged.hu/

[38] BORGEFORS, G. (1988) Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligience*, 849-865. Vol.10.

[39] www.staff.science.uva.nl