OBJECT MOTION PERCEPTION AND TRACKING USING SIFT AND K-MEAN CLUSTERING

A PROJECT REPORT

Submitted by

DEEKSHITHA. P [Reg No: RA1411003020028]
B. TARUN REDDY [Reg No: RA1411003020045]
SHRAVANI BADADHA [Reg No: RA1411003020050]
K.K. DHRUTHI [Reg No: RA1411003020053]

Under the guidance of

Mr.S. PANDIARAJ
(M.E, Department of Computer Sciene & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM UNIVERSITY
(Under section 3 of UGC Act 1956)

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2018

SRM UNIVERSITY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**OBJECT MOTION PERCEPTION AND TRACKING USING SIFT AND K-MEAN CLUSTERING**" is the bonafide work of " **DEEKSHITHA. P [Reg No: RA1411003020028], B. TARUN REDDY [Reg No: RA1411003020045], SHRAVANI BADADHA [Reg No: RA1411003020050], K.K. DHRUTHI [Reg No: RA1411003020053],** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mr.S. PANDIARAJ
GUIDE
M.E
Dept. of Computer Sciene & Engineering

SIGNATURE

Dr. J. JAGADEESAN
HEAD OF THE DEPARTMENT
Dept. of Computer Science & Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Object motion tracking and re-identification is a very complex technique which plays vital task to be addressed in any video surveillance system. Detecting a dynamic moving object accurately in a video is a difficult job. To identify and implement a system that takes as input a sequence of images and generates clusters of SIFT features using the K-Means clustering algorithm. Every time the system processes an image it compares each new cluster to the clusters of previous images. When at least 25 percent of the SIFT features that compose a cluster match a cluster in the local cache, the system uses the centroid of both clusters in order to determine the direction of travel.In this proposed system, comparison of the frames analysis helps to track the motion object. Based on the motion information, key points corresponding to moving objects are extracted from every frame by using The Degree of Gaussian. SIFT (Scale Invariant feature transform) approach is specifically used for object recognition and to detect feature points when the scale and orientation are in place. Tracked SIFT capabilities provide the displacement of every feature point in the image, along with image coordinates and frame number constitute a feature vector. All the key points along with the feature vector are integrated and clustered in order to track the objects.Experimental results demonstrate that the proposed algorithm can deal with the challenging tracking situations such as: partial occlusion, illumination change, scale variations, object rotation and complex background clutter.Then,the number of moving objects in each frame is determined according to their motion-based information and position, and are later clustered using the k-means algorithm. Clustering of moving objects is performed using feature vectors made of pixels intensities, motion magnitudes, motion directions and feature point positions.

We would like to extend our gratitude to Dr.T.R.Pachamuthu,Founder,Dr. R.Shivakumar, M.D, Ph.D., Chairman and Dr. V. Subbiah Bharathi, M.E., Ph.D, Dean for their persistent endeavors towards our education.

We also extend our sincere thanks to Head of the Department, Computer Science and Engineering, Dr.J.Jagadeesan , M.Tech , Ph.D, for the constant support.

It is indeed a pleasure to mention about Mr. R.Augustian Isaac , M.E., Assistant Professor , project co-ordinator S.Pandiaraj, M.E., Assistant Professor(S.G) , project guide who have always been patient enough to make the complexities of the project and relentlessly supported us throughout the project.

Our thanks to the teaching and non-teaching staff of the Computer Science and Engineering department of SRM University, Ramapuram Campus, who provided necessary resources for this project.

Author

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CI	Confidence Interval Metric
SIFT	Scale Invarient Feature Transform
BP	Black Propagation
DMT	Diagonal Matrix Propagation
BTF	Brightness Transform Function
CCH	Conventional Color Histogram
FCH	Fuzzy Color Histogram
DoG	Difference of Gaussian
GMM	Gaussian Mixture Model

LIST OF SYMBOLS

α, β	Damping constants
θ	Angle of twist, rad
ω	Angular velocity, rad/s
b	Width of the beam, m
h	Height of the beam, m
$\{f(t)\}$	force vector
$[K^e]$	Element stiffness matrix
$[M^e]$	Element mass matrix
$\{q(t)\}$	Displacement vector
$\{\dot{q}(t)\}$	Velocity vector
$\{\ddot{q}(t)\}$	Acceleration vector

CHAPTER 1

INTRODUCTION

1.1 Overview

The system is proposed to develop an object motion perception for tracking and re-identification using SIFT and K- Means clustering algorithm. Object tracking and re-identification is a complex but it is an essential task to be addressed in any video surveillance application. This process is done through subtraction of background and current frame by XOR operation and propose to use a SIFT based method for tracking image features points across the frames and finally all similar features points are clustered into K groups of similarity. The coding part is done by the language MATLAB in the MATLAB version 7.5 to make the work easier and simpler to analyze, user interface development and for efficient system management. It had been demonstrated in our method in the context of a real-time surveillance application. There are immediate needs for automated surveillance systems in commercial, law enforcement and military applications. Mounting video cameras is cheap, but finding available human resources to observe the output is expensive. Although surveillance cameras are already prevalent in banks, stores, and parking lots, video data currently is used only "after the fact" as a forensic tool, thus losing its primary benefit as an active, real-time medium. What is needed is continuous 24-hour monitoring of surveillance video to alert security officers to a burglary in progress, or to a suspicious individual loitering in the parking lot, while there is still time to prevent the crime. In addition to the obvious security applications, video surveillance technology has been proposed to measure traffic flow, detect accidents on highways, monitor pedestrian congestion in public spaces, compile consumer demographics in shopping malls and amusement parks, log routine maintenance tasks at nuclear facilities, and count endangered species. Many real-time techniques proposed in the literature rely on a frame-to-frame matching of objects. This describes a technique which takes into consideration of the inherent temporal coherence that exists across frames, thus being able to robustly perform tracking while handling difficult situations such as object acceleration and partial occlusion. Scale Invarient Feature Transform (SIFT) approaches have been shown to perform well for object recognition, due to their robustness to noise, changes in illumination and viewpoint. In this work the propose to use a SIFT-based method for tracking image features across frames. Tracked SIFT features provide the displacement of each interest point in the image, which along with image coordinates and frame number constitute a feature vector. All feature vectors are added to a temporal buffer and clustered in order to identify and track coherently moving regions. The proposed clustering method uses an improved K-Means technique where K is determined using a Confidence Interval Metric (CI) metric. MATLAB 7.5 version is a software development platform which allows applications to be developed from a set of modular software components called modules. The advantages of MATLAB is that it offers greater performance Ratio. No handcrafted feature can be considered as universal, learning relations from data can be advantageous. Modeling complex distributions and functions have been a bottleneck in machine learning.

1.2 Introduction

The main aspect describing the achievement is by robust and efficient matching of image features or regions is an important prerequisite to many problems in computer vision. The success of higher level processes such as object recognition, classification or tracking depends heavily on the quality of image matching. A simple and fast approach to matching is through region correlation. This technique has little use in real world applications where changes in illumination, viewpoint or scale, as well as partial occlusions are common. Objects that undergo partial occlusion or even complete occlusion for a few frames pose particular challenges. Various techniques that rely on the continuous appearance of the entire object for the purpose of recognition fail in this context. Employing local features in matching helps alleviate this problem and has led to successful approaches in many applications including object recognition, texture recognition, and image retrieval. It has been shown that SIFT based descriptors perform particularly well in this respect. Visual object tracking appears as an essential component of many applications in Video conferencing, vision-based surveillance and monitoring, or image and video understanding. A wide range of methods have been presented in literature, targeting the problem of visual tracking. One of the most common methods with many variations is blob tracking, which usually emphasizes the difference between the current image observation and a model of the background. Such approaches assume that the image regions (blobs) extracted in this manner correspond to the actual foreground objects in the scene. These algorithms have particular difficulty handling shadows, non-stationary parts of the background and occlusions. A closely related approach to blob tracking is based on tracking active contours representing the boundary of an object. Active contour-based tracking algorithms represent the outline of moving objects as contours, which are updated dynamically in successive frames. These algorithms have drawbacks, such as their tracking accuracy is limited by a lack of precision in the location of the contour. Feature-based tracking algorithms perform tracking of objects by extracting elements, clustering them into higher level features and then matching the features between images. Feature-based tracking algorithms can further be classified into three subcategories: global feature-based methods, local feature-based methods, and dependence-graph-based methods. Global feature-based methods rely on features like color, centroid, and perimeters . The features used in local feature-based algorithms include line segments, curve segments, and corner vertices . In general, feature-based tracking methods can adapt successfully and rapidly to allow realtime processing and tracking of multiple objects with the exception of dependence-graph-based methods. In a different direction of research, recent efforts employ statistical models for representing video content. Our approach proceeds with a stage of background modeling and foreground extraction. SIFT features are extracted at points of interest inside the regions detected as foreground, then used for matching from one frame to the next. Each matched pair of SIFT features provides a displacement vector, equivalent to the image velocity of the point of interest. Within a temporal buffer the collect feature vectors that consist of this displacement, the image coordinates and the frame number. The frame number is normalized by the size of the temporal window considered. All feature vectors are processed using an improved K-Means algorithm to find the most dominant clusters, where the number of clusters is determined using a novel cluster statistics metric.

The contribution of this work is the development of a robust, real-time SIFT based approach for object tracking that enforces the inherent temporal coherence across image frames, therefore handling difficult situations caused by significant object acceleration and partial occlusion.

1.3 Objective

The main objective of this project is to strengthen the full potential of automatic surveillance systems for applications like content-based retrieval of surveillance video with the tracking and detection by predicting the abnormalities which can be achieved if the system can be applied in an unconstrained environment. Some of the major challenges that need to be handled in such an unconstrained environment include:

- Changes in illumination
- Background dynamics
- Camera setup
- Object motion
- Different types of objects
- Size of the monitored area

Often constraints are made in relation to these challenges but the resulting systems will also be less general and further away from the true potential of the applications. Surveillance is the monitoring of behaviour. Systems surveillance is the process of monitoring the behaviour of people, objects or processes within systems for conformity to expected or desired norms in trusted systems for security or social control.

1.4 Motivation

The state and security services still have the most powerful surveillance systems, because they are enabled under the law. The full potential of automatic surveillance systems for applications like content-based retrieval of surveillance video and detection and prediction of abnormalities will only be achieved. The state and security services still have the most powerful surveillance systems, because they are enabled under the law In an open area the objects will be able to move in any direction, and with a camera setup typical of surveillance systems, this will give movement in all directions of the surveillance video, and objects will enter and leave the field of view on all its boundaries.

1.5 Organization Of Report

The overview of the system is stated to propose a system which overcome the issue faced in the existing system. The problem is stated and analyzed in a well efficient manner. The introduction explains about the robust and efficient matching of image features or regions is an important prerequisite to many problems in computer vision.Furthermore the video will show some perspective, i.e. the size of an object will change when it moves towards or away from the camera. The object's freedom of movement also implies that they can move in a way where they occlude each other, or they may stop moving for a while. In the case of people the occlusion and stopping will be very likely when they are interacting, e.g. two people stopping and talking to each other and then shaking hands or hugging before departure. People may also be moving in groups or form and leave groups in an arbitrary fashion. These challenges could be solved by restricting the movement of the objects, but this would limit the system from being applied in many situations. Different types of objects: In some open areas many different types of objects will be present. A surveillance video of a parking lot for example will contain vehicles, persons, and maybe birds or dogs. People may also leave or pick up other objects in the scene. The most general surveillance system would be able to distinguish between these objects, and treat them in the way most appropriate to that type of object. Constraints in this respect would limit the system to areas with only a certain type of objects.The success of higher level processes such as object recognition, classification or tracking depends heavily on the quality of image matching. Based on the existing system, tried to enhance the object recognition, tracking and re-identification of image processing.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System

Object detection and tracking remains an open research problem even after research of several years in this field. A robust, accurate and high performance approach is still a great challenge today. The difficulty level of this problem highly depends on how one defines the object to be detected and tracked. If only a few visual features (e.g. color) are used as representation of an object, it is not so difficult to identify the all pixels with same color as the object. However, there is always a possibility of existence of another object or background with the same color information. Moreover, the change of illumination in the scene does not guarantee that the color will be same for the same object in all the frames. This leads to inaccurate segmentation based on only visual features (e.g. color). type of variability changes is quite obvious as video objects generally are moving objects. The images of an object may change drastically as it moves from one frame to another through the field of view of a camera. This variability comes from three principle sources namely variation in target pose or deformations, variation in illumination and partial/full occlusion of the target . There are two main techniques which are used in the existing system. The diagonal matrix transform is used to transform the pixel rate into a diagonal matrix by comparing the neighboring pixel values. There is no improvement in the affine model compared to the Diagonal Matrix Propagation (DMT) with spectral sharpening. The other technique used is the brightness transform function which is used to adjust the brightness in the image. By Brightness Transform Function (BTF) pixel level correspondence cannot be achieved.

2.2 Issues in Existing System

2.2.1 Illumination changes

It is desirable that background model adapts to gradual changes of the appearance of the environment. For example in outdoor settings, the light intensity typically varies during day. Sudden illumination changes can also occur in the scene. This type of change occurs for example with sudden switching on/off a light in a indoor environment. This may also happen in outdoor scenes (fast transition from cloudy to bright sunlight). Illumination strongly affects the appearance of background, and cause false positive detection. The background model should take this into consideration.

2.2.2 Dynamic Background

Some parts of the scenery may contain movement (a fountain, movements of clouds, swaying of tree branches, wave of water etc.) which should be regarded as background, according to their relevance. Such movement can be periodical or irregular (e.g., traffic lights, waving trees). Handling such background dynamics is a challenging task.

2.2.3 Occlusion

Occlusion (partial/full) may affect the process of computing the background frame.In real life situations, occlusion can occur anytime a subject passes behind an object with respect to a camera.

2.2.4 Clutter

Presence of background clutter makes the task of segmentation difficult. It is hard to model a background that reliably produces the clutter background and separates the moving foreground objects from that.

2.2.5 Camouflage

Intentionally or not, some objects may poorly differ from the appearance of background, making correct classification difficult. This is especially important in surveillance applications. Camouflage is particularly a problem for temporal differencing methods.

2.2.6 Presence of Shadows

Cast by foreground objects often complicate further processing steps subsequent to background subtraction. Overlapping shadows of foreground regions for example hinder their separation and classification. Researchers have proposed different methods for detection of shadows.

2.3 Literature Survey

2.3.1 Generalized Stauffer-Grimson background subtraction

Generalized Stauffer-Grimson background separation is an algorithm which is used to separate the background from that particular image or frame. Background separation is one of the most important steps in object detection. In this, use an online k-means algorithm for updating the parameters using the necessary statistics of the model. Then based on the analysis of the motion video the background separation is done. Usually the background separation algorithms are robust. There are various techniques like Dynamic textures, Background models, Background subtraction, mixture models and adaptive models used. It is suitable only for static scenes. This is a strong limitation for scenes with spatiotemporal dynamics.

2.3.2 Tracking and Counting People in Visual Surveillance Systems

The most difficult task in a visual Surveillance system is to count and track people. If there is only one person in the video then there would not be any problem but if there are multiple people appearing in the video then there exists various techniques to track them. Before a person is tracked the background separation is done in order to identify the object correctly. Once the background is separated to identify the object or person accurately there are decision points which are taken. The decision points are further refined to identify the object clearly. By this method the averaged detection ratio has been improved by about 10% when compared to the conventional method. The main drawback of it is very sensitive to light variations.

2.3.3 A People-Counting System based on BP Neural Network

A people-counting system is based on a Black Propagation (BP) neural network. The system uses cheap photoelectric sensor to collect data and introduces BP neural network for counting and recognition, and it is effective and flexible for the purpose of performing people counting. There are new methods for segmentation and feature extraction which are developed to enhance the classification performance. Promising results were obtained and the analysis indicates that the system based on BP neural network provides good results with low false rate and it is effective for people-counting. There are various techniques like segmentation, feature extraction and back propagation neural network used. To fully implement a standard neural network architecture would require lots of computational resources which is a major drawback. They are black box - that is the knowledge of its internal working is never known.

2.3.4 Fuzzy Color Histogram and Its Use in Color Image Retrieval

Conventional Color Histogram (CCH) considers neither the color similarity across different bins nor the color dissimilarity in the same bin. Therefore, it is sensitive to noisy interference such as illumination changes and quantization errors. Furthermore, CCH large dimension or histogram bins require large computation on histogram comparison. To address these concerns, a new color histogram representation, called Fuzzy Color Histogram (FCH), by considering the color similarity of each pixel's color associated to all the histogram bins through fuzzy-set membership function. A novel and fast approach for computing the membership values based on fuzzy -means algorithm is introduced. The FCH is further exploited in the application of image indexing and retrieval. Experimental results clearly show that FCH yields better retrieval results than CCH. Such computing methodology is fairly desirable for image retrieval over large image databases. There are various techniques used like Conventional color histogram, Fuzzy -means, Fuzzy color histogram, Illumination changes, Image indexing and retrieval, Membership matrix. There are various drawbacks. Representation is dependent of the color of the object being studied ignoring its shape and texture.High sensitivity to noisy interference such as lighting intensity changes and quantization errors.

2.3.5 Real-Time Human Motion Detection and Tracking

It describes a real-time system for human detection, tracking and motion Analysis. The system is an automated video surveillance system for detecting and monitoring people in both indoor and outdoor environments. Detection and tracking are achieved through several steps:Initially the design of a robust, adaptive background model that can deal with lightning changes, long term changes in the scene and objects occlusions. This model is used to get foreground pixels using the background subtraction method. Afterwards, noise cleaning and object detection are applied, followed by human modelling to recognize and monitor human activity in the scene such as human walking or running. The techniques used are Motion Detection, Tracking, Human Model, Surveillance, Motion Analysis, Image Processing. Difficult to recognize different types of human activities such as jumping, falling and entering secured area, identifying other moving objects like animals and vehicles.

2.4 Problem Statement

Object detection and tracking remains an open research problem even after research of several years in this field. A robust, accurate and high performance approach is still a great challenge today. The difficulty level of this problem highly depends on how one defines the object to be detected and tracked. So, presented a object detection and tracking with a greater performance ratio by reducing the existing overheads.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

The main aspect describing the achievement is by robust and efficient matching of image features or regions is an important prerequisite to many problems in computer vision. The success of higher level processes such as object recognition, classification or tracking depends heavily on the quality of image matching. A simple and fast approach to matching is through region correlation. However, this technique has little use in real world applications where changes in illumination, viewpoint or scale, as well as partial occlusions are common. Objects that undergo partial occlusion or even complete occlusion for a few frames pose particular challenges. Various techniques that rely on the continuous appearance of the entire object for the purpose of recognition fail in this context. Employing local features in matching helps alleviate this problem and has led to successful approaches in many applications including object recognition, texture recognition, and image retrieval. Our approach proceeds with a stage of background modeling and foreground extraction. SIFT features are extracted at points of interest inside the regions detected as foreground, then used for matching from one frame to the next. Each matched pair of SIFT features provides a displacement vector, equivalent to the image velocity of the point of interest. Within a temporal buffer, collect feature vectors that consist of this displacement, the image coordinates and the frame number. The frame number is normalized by the size of the temporal window considered. All feature vectors are processed using an improved K-Means algorithm to find the most dominant clusters, where the number of clusters is determined using a novel cluster statistics metric. The contribution of this work is the development of a robust, real-time SIFT based approach for object tracking that enforces the inherent temporal coherence across image frames, therefore handling difficult situations caused by significant object acceleration and partial occlusion.

3.2 Analysis of the Problem

Object detection and tracking remains an open research problem even after research of several years in this field. A robust, accurate and high performance approach is still a great challenge today. The difficulty level of this problem highly depends on how one defines the object to be detected and tracked. If only a few visual features (e.g. color) are used as representation of an object, it is not so difficult to identify the all pixels with same color as the object. There is always a possibility of existence of another object or background with the same color information. Due to the change of illumination in the scene does not guarantee that the color will be same for the same object in all the frames. This leads to inaccurate segmentation based on only visual features (e.g. color). type of variability changes is quite obvious as video objects generally are moving objects. The images of an object may change drastically as it moves from one frame to another through the field of view of a camera. This variability comes from three principle sources namely variation in target pose or deformations, variation in illumination and partial/full occlusion of the target . The typical challenges of background subtraction in the context of video surveillance have been listed below:

- Illumination changes
- Dynamic background
- Occlusion
- Clutter
- Camouflage
- Presence of shadow

3.3 System Architecture

The system architecture is given below:

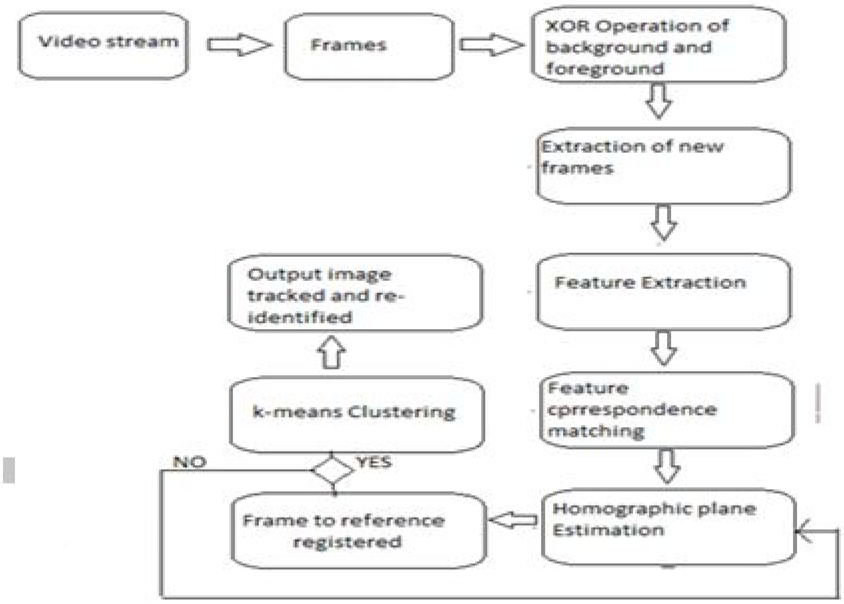

Figure 3.1: System Architecture

3.3.1 Architecture Description

Detecting moving objects and tracking in a dynamic scene is a difficult task in computer vision. Propose a moving object detection algorithm for advanced data driver assistance. The existing system use color as it is or address these challenges by designing color spaces focusing on a specific cue. In this proposed system, a data driven approach for learning color patterns from pixels sampled from video. Based on the motion-based information, feature points corresponding to moving objects are extracted from next frame. Then, the number of moving objects in each frame is determined according to their motion-based information and position, and are later clustered using the k-means algorithm. Clustering of moving objects is performed using feature vectors made of pixel's intensities, motion magnitudes, motion directions and feature point positions.

3.3.2 Flow chart

The flow chart is given below:

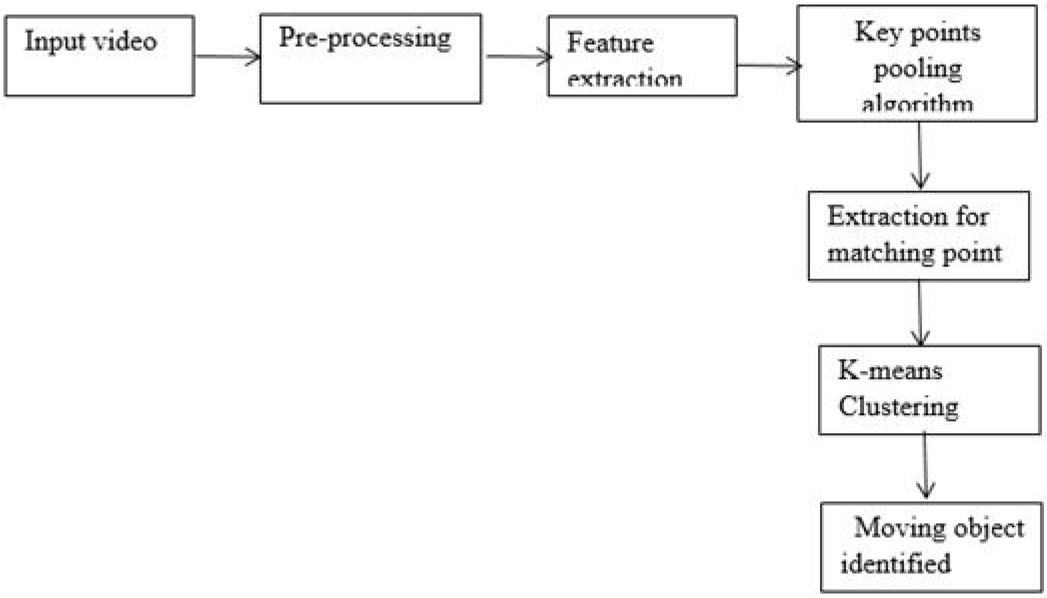

Figure 3.2: Flow Chart of Process Identification

3.3.3 Description

Detecting moving objects and tracking in a dynamic scene is a difficult task in computer vision. Proposed a moving object detection algorithm for advanced data driver assistance. The existing system use color as it is or address these challenges by designing color spaces focusing on a specific cue. In this proposed system, a data driven approach for learning color patterns from pixels sampled from video. Based on the motion-based information, feature points corresponding to moving objects are extracted from next frame. Then, the number of moving objects in each frame is determined according to their motion-based information and position, and are later clustered using the k-means algorithm. Clustering of moving objects is performed using feature vectors made of pixel's intensities, motion magnitudes, motion directions and feature point positions.

3.4 Mathematical Model for Overall System

SIFT-Scale Invariant Feature Transform The SIFT approach, for image feature generation, takes an image and transforms it into a "large collection of local feature vectors" Each of these feature vectors is invariant to any scaling, rotation or translation of the image. To aid the extraction of these features the SIFT algorithm applies a 4 stage filtering approach:

1. **Scale-Space Extrema Detection** This stage of the filtering attempts to identify those locations and scales that are identifiable from different views of the same object. This can be efficiently achieved using a "scale space" function.It has been shown under reasonable assumptions it must be based on the Gaussian function. The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.1}$$

Where $*$ is the convolution operator, G(x, y, σ) is a variable-scale Gaussian and I(x, y) is the input image. Various techniques can then be used to detect stable keypoint locations in the scale-space. Difference of Gaussians is one such technique, locating scale-space extrema, D(x, y, σ) by computing the difference between two images, one with scale k times the other. D(x, y, σ) is then given by: D(x, y, σ) = L(x, y, kσ) - L(x, y,σ)To detect the local maxima and minima of D(x, y,σ) each point is compared with its 8 neighbours at the same scale, and its neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

2. **Keypoint Localistaion** This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localised on an edge. This is achieved by calculating the Laplacian value for each keypoint found in stage 1. The location of extremum, z, is given by:
article z=

$$\frac{\partial D^{-}1}{\partial x^2}\frac{\partial D}{\partial x}$$

If the function value at z is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localisation it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the deference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the keypoint, the keypoint is rejected.

3. **Orientation Assignment** This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, described below, can then be represented relative to this orientation, achieving invariance to rotation. The approach taken to find an orientation is:
1. Use the keypoints scale to select the Gaussian smoothed image L, from above
2. Compute gradient magnitude, m
m(x,y)=$\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}$
3. Compute orientation, θ
θ(x,y)=tan((L(x,y+1)-L(x,y-1)/(L(x+1,y)-L(x-1,y)))
4. Form an orientation histogram from gradient orientations of sample points
5. Locate the highest peak in the histogram. Use this peak and any other local peak within 80
6. Some points will be assigned multiple orientations

7. Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position

4. **Keypoint Descriptor** The local gradient data is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is then used to create a set of histograms over a window centred on the keypoint. Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This results in a feature vector containing 128 elements. These resulting vectors are know as SIFT keys and are used in a nearest-neigbours approach to identify possible objects in an image. Collections of keys that agree on a possible model are identfied, when 3 or more keys agree on the model parameters this model is evident in the image with high probability. Due to the large number of SIFT keys in an image of an object, typically a 500x500 pixel image will generate in the region of 2000 features, substantial levels of occlusion are possible while the image is still recognised by this technique.

3.5 K-means Clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} (x - c)^2 \tag{3.2}$$

where $(x-c)^2$ is a chosen distance measure between a data point and the cluster centre,is an indicator of the distance of the n data points from their respective cluster centres. The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains. System is going to see, it is a good candidate for extension to work with fuzzy feature vectors.Suppose that one have n sample feature vectors x1, x2, ..., xn all from the same class, and one know that they fall into k compact clusters, k < n. Let mi be the mean of the vectors in cluster i. If the clusters are well separated, it can use a minimum-distance classifier to separate them. That is, one can say that x is in cluster i if || x - mi || is the minimum of all the k distances. This suggests the following procedure for finding the k means: Make initial guesses for the means m1, m2, ..., mk Until there are no changes in any mean Use the estimated means to classify the samples into clusters For i from 1 to k Replace mi with the mean of all of the samples for cluster i end for end until Here is an example showing how the means m1 and m2 move into the centers of two clusters. .

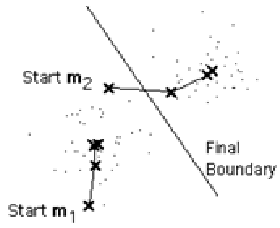

Figure 3.3: 2-mean clustering

This is a simple version of the k-means procedure. It can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers. It does have some weaknesses:

1.The way to initialize the means was not specified. One popular way to start is to randomly choose k of the samples.

2.The results produced depend on the initial values for the means, and it frequently happens that suboptimal partitions are found. The standard solution is to try a number of different starting points.

3.It can happen that the set of samples closest to mi is empty, so that mi cannot be updated. This is an annoyance that must be handled in an implementation, but that shall ignore.

4.The results depend on the metric used to measure || x - mi ||. A popular solution is to normalize each variable by its standard deviation, though this is not always desirable.

5.The results depend on the value of k.

This last problem is particularly troublesome, since, often have no way of knowing how many clusters exist. In the example shown above, the same algorithm applied to the same data produces the following 3means clustering. Is it better or worse than the 2means clustering?

Unfortunately there is no general theoretical solution to find the optimal number of

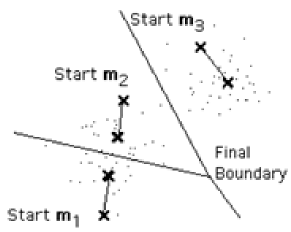

Figure 3.4: 3-mean clustering

clusters for any given data set. A simple approach is to compare the results of multiple runs with different k classes and choose the best one according to a given criterion but one need to be careful because increasing k results in smaller error function values by definition, but also an increasing risk of overfitting.

3.6 System Requirements

3.6.1 Hardware Specification

Hard disk : 40GB

RAM : 512MB

Processor : Pentium IV

3.6.2 Software Specification

OS : Windows 7/XP

Front End : MATLAB 7.1

Back End : MATLAB 7.1

3.7 Summary

Based on the problems in the existing system, the software used to program for object motion detection for both tracking and re-identification using SIFT based algorithm and k-means clustering to promote better performance strategy.

CHAPTER 4

MODULE DESCRIPTION

4.1 Introduction

The modules in the proposed software system are built so that each is a complete software sub system on its own performing a particular operation. In the proposed system the main modules are listed as follow:
1.Background Modeling and Foreground Detection Module
2.Estimation of a Feature Vector Module
3.Clustering Module
4.Object Tracking and re-identification Module

4.2 Background and Foreground Detection Module

4.2.1 Introduction

This process is done for detecting foreground regions in each image frame. This part of the system detects the moving objects (blobs) as simple image regions, without any assumption about the objects that they represent e.g., vehicle or non-vehicle. the system used a simple recursive learning method to model the background in order to satisfy these requirements.

4.2.2 Description

The background and foreground detection diagram is shown in figure 4.1. It is used to explain the background and foreground detection clearly. As it is shown in the figure the basic function is to employ an adaptive background model for the entire region of awareness, and for segmenting the novel objects that appear in foreground. Our approach involves learning a statistical color model of the background, and process a new frame using the current distribution in order to segment foreground elements. The algorithm has three main stages: learning, classification and post-processing. In the learning stage, establish the background model using recursive learning. In the classification stage, classify the image pixels into foreground and background pixels based on the background model. Finally, in a post-processing stage, group the detected foreground pixels into connected components and created a list of blobs (foreground objects) associated with the current image frame.

4.2.3 Flowchart

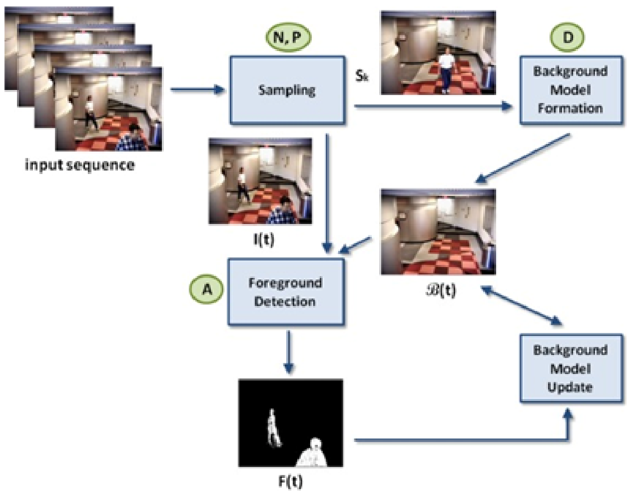

Figure 4.1: Background and forged detection

4.3 Estimation of a Feature Vector Module

4.3.1 Introduction

The interest points are determined by finding local extrema in the Difference of Gaussian (DoG) scale-space. The SIFT features are extracted for all such interest points detected. Each feature is classified as corresponding to one of the foreground blobs or to the background.

4.3.2 Description

The Estimation of a Feature Vector diagram is shown in figure 4.2. For any object there are many features, interesting points on the object, that can be extracted to provide a "feature" description of the object. This description can then be used when attempting to locate the object in an image containing many other objects. There are many considerations when extracting these features and how to record them. SIFT image features provide a set of features of an object that are not affected by many of the complications experienced in other methods, such as object scaling and rotation. For every feature corresponding to a particular blob and try to find a match with the features found in the next frame within a neighborhood window. All the matching features for a corresponding blob are averaged to find the displacement vector. The new position of the blob is determined by the previous position of the blob and the displacement vector.

4.3.3 Flow Chart

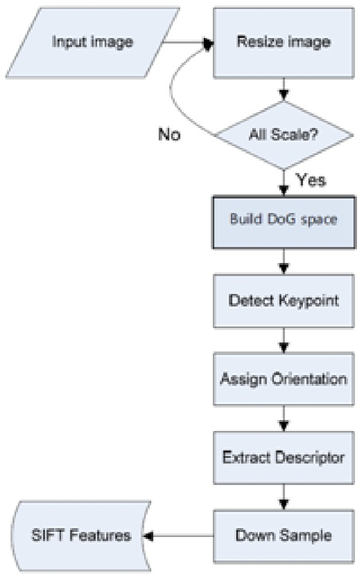

Figure 4.2: Flow Chart of SIFT based algorithm

4.4 Clustering Module

4.4.1 Introduction

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priority.

4.4.2 Description

The Clustering diagram is represented in figure 4.3. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point the need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After having these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop it may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.This algorithm aims at minimizing an objective function, in this case a squared error function.

4.5 Object Tracking Module

4.5.1 Introduction

Object tracking module is where the object is tracked and also re-identified. In order to allow high-resolution images of the people in the scene to be acquired it is reasonable to assume that such people move about in the scene. To monitor the scene reliably it is essential that the processing time per frame be as low as possible. Hence it is important

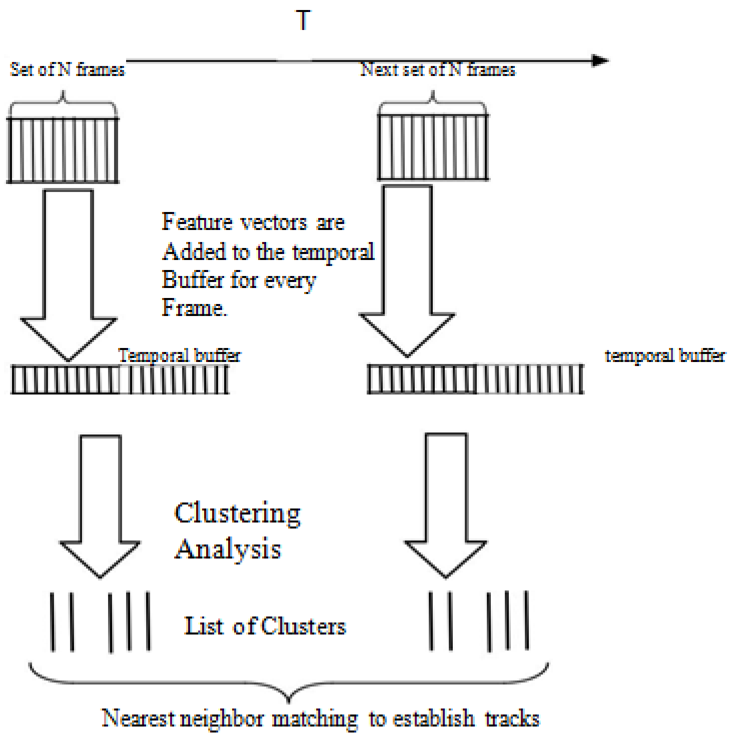

Figure 4.3: Block Diagram of K-Mean Clustering

that the techniques which are employed are as simple and as efficient as possible

4.5.2 Description

The object tracking diagram is represented in figure 4.4.Background subtraction allows moving objects to be detected by taking the point-by-point absolute difference of the current image and a background image. The system is an automated video surveillance system for detecting and monitoring people in both indoor and outdoor environments. Detection and tracking are achieved through several steps: Initially the design a robust, adaptive background model that can deal with lightning changes, long term changes in the scene and objects occlusions. This model is used to get foreground pixels using the background subtraction method. Afterwards, noise cleaning and object detection are applied, followed by human modeling to recognize and monitor human activity in the scene such as human walking or running.

4.5.3 Flow Chart

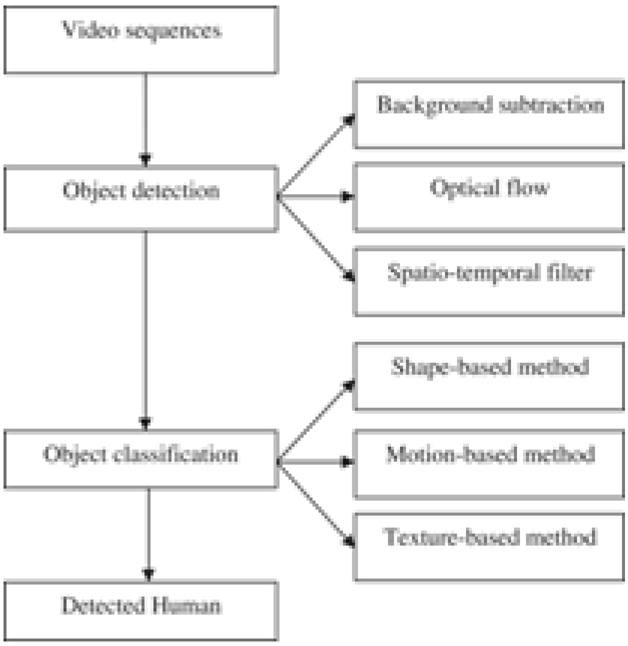

Figure 4.4: Flow chart of object training

4.6 Summary

Thus, there are the system is divided into four different modules. Each module operates separately and as a whole the system is used to detect the object in the video and also re-identify the object in the next frames.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Introduction

The software used for designing the file management systems is

1.MATLAB

The MATLAB technology is open source licensed and can be implemented for free. These software systems have been explained in detail in the following section.

5.2 Overview of Platform

5.2.1 MATLAB

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming. It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications. It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods. MATLAB's Power of Computational Mathematics MATLAB is used in every facet of computational mathematics. Following are some commonly used mathematical calculations where it is used most commonly :

1.Dealing with Matrices and Arrays

2.2D and 3D Plotting and graphics

3.Linear Algebra

4.Algebraic Equations

5.Non-linear Functions

6.Statistics

7.Data Analysis

8.Calculus and Differential Equations

9.Numerical Calculations

10.Integration

11.Transforms

12.Curve Fitting

13.Various other special functions

Features of MATLAB:

Following are the basic features of MATLAB

1.It is a high-level language for numerical computation, visualization and application development.

2.It also provides an interactive environment for iterative exploration, design and problem solving.

3.It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

4.It provides built-in graphics for visualizing data and tools for creating custom plots.

5.MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.

6.It provides tools for building applications with custom graphical interfaces.

7.It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

Uses of MATLAB

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including

1.Signal Processing and Communications

2.Image and Video Processing

3.Control Systems

4.Test and Measurement

5.Computational Finance

6.Computational Biology

5.3 Code Implementation

5.3.1 Sample Coding

LBP background subraction

```
clear all;
clc;
close all;
edgclrs=['r','g','b'];
nFrames = 25;
detarea=0;
rp=4;
count=1;
alpha = 0.05;
fmt = '.png';
figure(3);
title('humsn detected');
cropfin=1;

    for k = 1:1:nFrames
disp(['Processing frame: ' num2str(k)]);

    cd Frames1
b =num2str(k);
imagefile=strcat('1','(',b,')','.jpg');
Inn = imread(imagefile);
I = rgb2gray(Inn);
```

I = imresize(I,.25);

I = double(I);

cd ..

if(k == 1)

B = I;

end

FxRadius = 1;

FyRadius = 1;

TInterval = 2;

TimeLength = 2;

BorderLength = 1;

Compute uniform patterns

NeighborPoints = [8 8 8];

nDim = 2 $^(NeighborPoints(1));$

FLDP = XCSLBP(I, FxRadius, FyRadius, NeighborPoints, BorderLength);

Blbp = XCSLBP(B, FxRadius, FyRadius, NeighborPoints, BorderLength);

K = compute$_similarity(FLDP, Blbp);$

F = (K < 0.5);

F = medfilt2(F);

se = strel('line',12,90);

F=imdilate(F,se);

se = strel('line',5,0);

F=imdilate(F,se);

Boundary Label the Filtered Image [L, num]=bwlabel(F);

STATS=regionprops(L,'all');

cc=[];

removed=0;

[L2, num2]=bwlabel(L);

Trace region boundaries in a binary image.

B1,L,N,A

= bwboundaries(L2);

for kk=1:length(B1),

if(sum(A(kk,:)))

boundary = B1kk;

a$_mx = max(boundary(:,2));$

$a_mn = min(boundary(:,2));$

$b_mx = max(boundary(:,1));$

$b_mn = min(boundary(:,1));$

w(kk)=a$_mx - a_mn;$

$h(kk) = b_mx - b_mn;$

$ifw(kk) == 0$

$w(kk) = 1;$

end

$ifh(kk) == 0$

$h(kk) = 1;$

end

$x(kk) = a_mn;$

$y(kk) = b_mn;$

$rectangle('Position', [x(kk)y(kk)w(kk)$

$h(kk)],' EdgeColor',' r',' LineWidth', 2);$

$detarea(kk) = 2 * (w(kk) + h(kk));$

end

end

$figure(4);$

$imshow(Inn);$

$title('personidentitfied');$

$holdon$

$zx = 1;$

$forii = 1 : length(detarea)$

$ifdetarea(ii) > 80$

$II2 = imcrop(Inn, [x(ii) * rpy(ii) * rpw(ii) * rph(ii) * rp]);$

$count = count + 1;$

$filename = strcat((num2str(count)), fmt);$

$cdbg_img1$

$imwrite(II2, filename);$

$cd..$

$rectangle('Position', [x(ii) * rpy(ii) * rpw(ii) * rp$

$h(ii) * rp],' EdgeColor',' k',' LineWidth', 2);$

$datfinstr = [x(ii) * rpy(ii) * rpw(ii) * rph(ii) * rp];$

$datfinstrID(k, zx, :) = datfinstr;$

$zx = zx + 1;$

end

end

end

$disp('Finished');$

$cntmain = count;$

$savecntmaincntmain$

$savedatfinstrIDdatfinstrID$

$fork = 1 : 1 : 25$

$disp(['Processingframe :' num2str(k)]);$

$cdFrames1$

$b = num2str(k);$

$imagefile = strcat('1',' (', b,')',' .jpg');$

$Inn = imread(imagefile);$

$subplot(121);$

$imshow(Inn);$

$subplot(122)$

$imshow(Inn);$

$fortx = 1 : size(datfinstrID, 2)$

$DatX = datfinstrID(k, tx, :);$

$rectangle('Position', DatX,' EdgeColor',' k',' LineWidth', 2);$

end

$cd..$

$pause(.1);$

$holdon$

end

```
load datfinstrID datfinstrID
for k = 1:1:nFrames
disp(['Processing frame: ' num2str(k)]);
cd Frames1
b =num2str(k);
imagefile=strcat('1','(',b,')','.jpg');
Inn = imread(imagefile);
subplot(121);
imshow(Inn);
subplot(122)
imshow(Inn);
for tx=1:size(datfinstrID,2)
DatX=datfinstrID(k,tx,:);
rectangle('Position',DatX,'EdgeColor','k','LineWidth',2);
end
cd ..
pause(.1);
hold on
end
```

```
nFrames=15;
load datfinreidID datfinreidID
load datfinreidCLR datfinreidCLR
for k = 1:1:nFrames
disp(['Processing frame: ' num2str(k)]);
cd Frames2
b =num2str(k);
imagefile=strcat('1','(',b,')','.jpg');
Inn = imread(imagefile);
subplot(121);
imshow(Inn);
subplot(122)
imshow(Inn);

for tx=1:size(datfinreidID,2)
DatX=datfinreidID(k,tx,:);
CLRT=datfinreidCLR(k,tx,:);
if CLRT==1
edgclr='r';
else
edgclr='k';
end
rectangle('Position',DatX,'EdgeColor',edgclr,'LineWidth',2);
end
cd ..
pause(10);
hold on
end

nFrames=55;
load datfintrackID datfintrackID
```

```
load datfintrackCLR datfintrackCLR
edgclrs=['r','g','b'];
rp=4;
for k = 1:1:nFrames
disp(['Processing frame: ' num2str(k)]);
cd Frames1
b =num2str(k);
imagefile=strcat('1','(',b,')','.jpg');
Inn = imread(imagefile);
subplot(121);
imshow(Inn);
subplot(122);
I = imresize(Inn,.25);
imshow(Inn);

    for tx=1:size(datfintrackID,2)
DatX=datfintrackID(k,tx,:);
valper=datfintrackCLR(k,tx,:);
if valper>0
rectangle('Position',DatX*rp,'EdgeColor',edgclrs(valper),'LineWidth',2);
end
end
cd ..
pause(.5);
hold on
end

    function [descriptors, locs] = sift(img)
if size(img,3)>1
img = rgb2gray(img);
end
```

```
[rows, cols] = size(img);
f = fopen('tmp.pgm', 'w');
if f == -1
error('Could not create file tmp.pgm.');
end
fprintf(f, 'P5fwrite(f, img', 'uint8');
fclose(f);
if isunix
command = '!./sift ';
else
command = '!siftWin32 ';
end
command = [command ' <tmp.pgm >tmp.key'];
eval(command);
g = fopen('tmp.key', 'r');
if g == -1
error('Could not open file tmp.key.');
end
header, count
= fscanf(g, 'if count  = 2
error('Invalid keypoint file beginning.');
end
num = header(1);
len = header(2);
if len  = 128
error('Keypoint descriptor length invalid (should be 128).');
end

locs = double(zeros(num, 4));
descriptors = double(zeros(num, 128));
```

for i = 1:num
vector, count

= fscanf(g, 'if count = 4

error('Invalid keypoint file format');

end

locs(i, :) = vector(1, :);

[descrip, count] = fscanf(g, 'if (count = 128)

error('Invalid keypoint file value.');

end

descrip = descrip / sqrt(sum(descrip.^{2}));

$descriptors(i, :) = descrip(1, :);$

end

$fclose(g);$

delete('tmp.pgm');

function [matchLoc1 matchLoc2] = siftMatch(img1, img2)
des1, loc1

= sift(img1);
des2, loc2

= sift(img2);

distRatio = 0.6;

des2t = des2';

matchTable = zeros(1,size(des1,1));

for i = 1 : size(des1,1)

dotprods = des1(i,:) * des2t;
vals,indx

= sort(acos(dotprods));

if (vals(1) < distRatio * vals(2))

```
matchTable(i) = indx(1);
else
matchTable(i) = 0;
end
end
img3 = appendimages(img1,img2);


    figure('Position', [100 100 size(img3,2) size(img3,1)]);
colormap('gray');
imagesc(img3);
hold on;
cols1 = size(img1,2);
for i = 1: size(des1,1)
if (matchTable(i) > 0)
line([loc1(i,2) loc2(matchTable(i),2)+cols1], ...
    loc1(i,1) loc2(matchTable(i),1)
, 'Color', 'c');
end
end
hold off;
num11 = sum(matchTable > 0);
fprintf('Found

    idx1 = find(matchTable);
idx2 = matchTable(idx1);
x1 = loc1(idx1,2);
x2 = loc2(idx2,2);
y1 = loc1(idx1,1);
y2 = loc2(idx2,1);


    matchLoc1 = [x1,y1];
matchLoc2 = [x2,y2];
```

save num11 num11

end

addpath('../')

clear all; clc;

dir$_p ath = ['img/'];$

$png_f iles = dir(fullfile(dir_p ath, '*.png'));$

$nFrames = size(png_f iles, 1);$

detector = vision.ForegroundDetector(...

'NumTrainingFrames', 5, ...

'InitialVariance', 30*30);

for k = 1:1:nFrames

disp(['Processing frame: ' num2str(k)]);

imagefile = [dir$_p athnum2str(k)'.png'$];

$I = imread(imagefile);$

$I = rgb2gray(I);$

$I = imresize(I, 0.25);$

$I = double(I);$

FxRadius = 1;

FyRadius = 1;

TInterval = 2;

TimeLength = 2;

BorderLength = 1;

NeighborPoints = [8 8 8];

nDim = 2 $^{(}NeighborPoints(1));$

XCS = XCSLBP(I,FxRadius, FyRadius, NeighborPoints, BorderLength);

XCS = XCS*(255/16);
frame = medfilt2(XCS);

mask = step(detector, frame);
mask = medfilt2(mask);

h1 = figure(1);
subplot(1,2,1), imshow(I,[],'InitialMagnification','fit'), title('Input');
subplot(1,2,2), imshow(mask,[],'InitialMagnification','fit'),
title('Foreground');
pause(0.1);

end
disp('Finished');

clc
clear all
close all
mm=40;
fmt = '.png';
for iii = 1:mm
cd train1
a=sprintf('(filename=strcat(a,fmt);
im=imread(filename);
end

5.4 Screenshots

These are the few screen-shots from the running system:

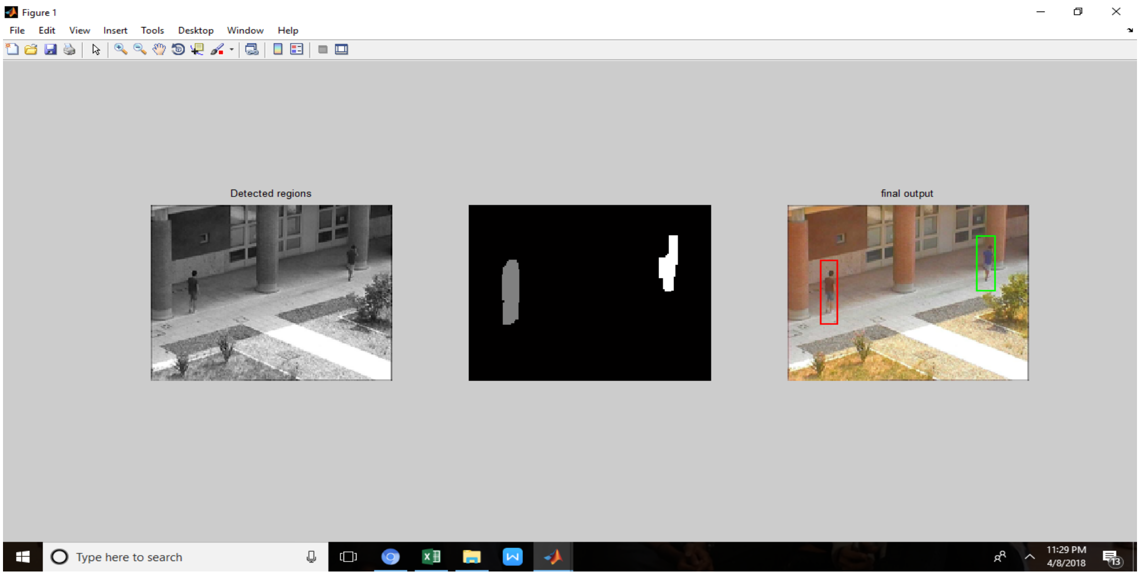

Figure 5.1: Median filter and Background Substraction

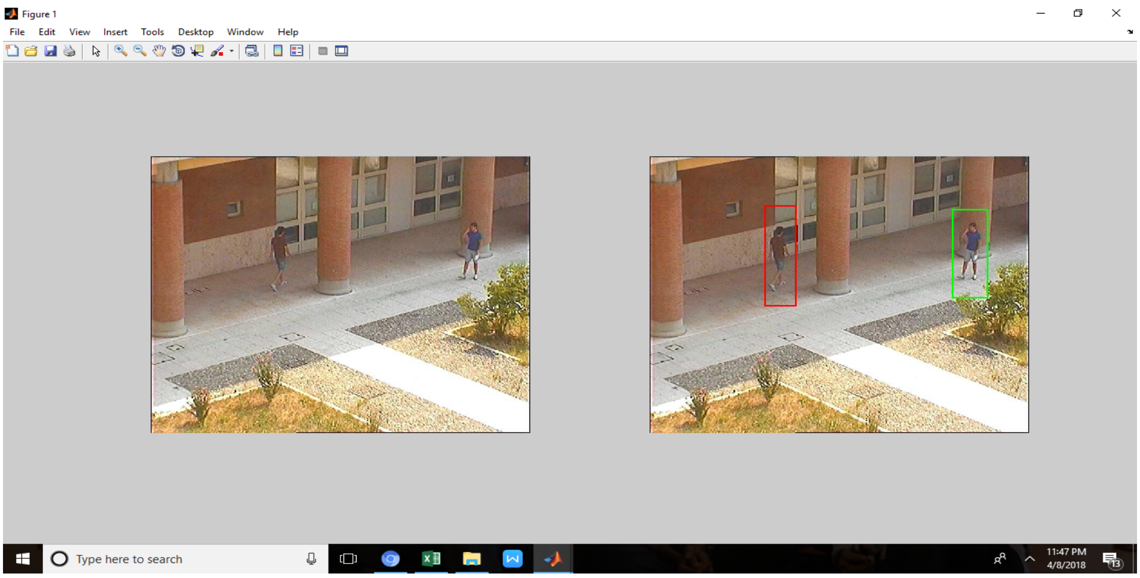

Figure 5.2: Image processing and detection

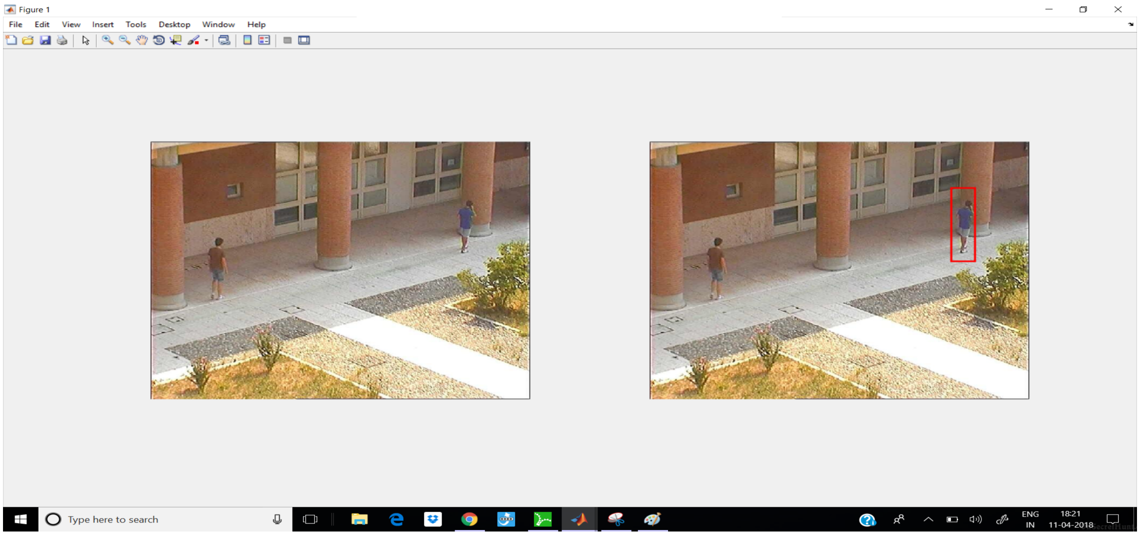

Figure 5.3: Re-identification process

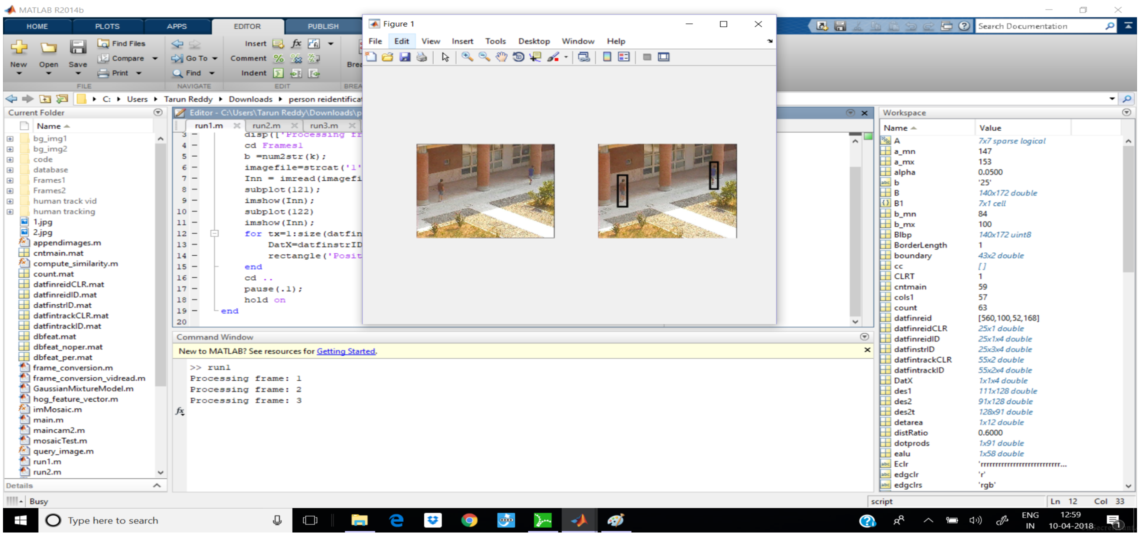

Figure 5.4: Detection process

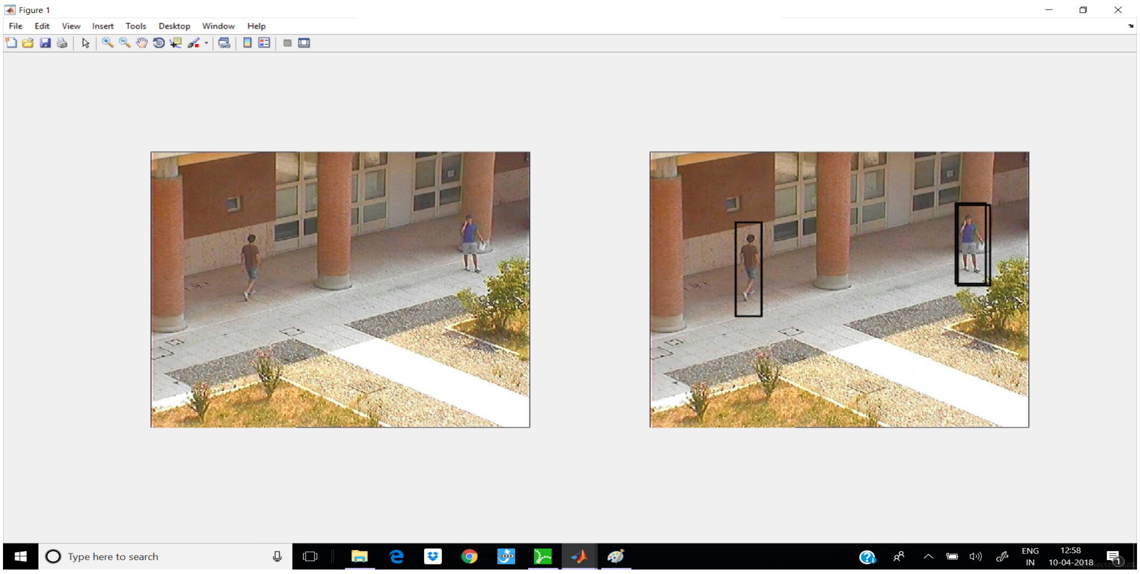

Figure 5.5: Detection

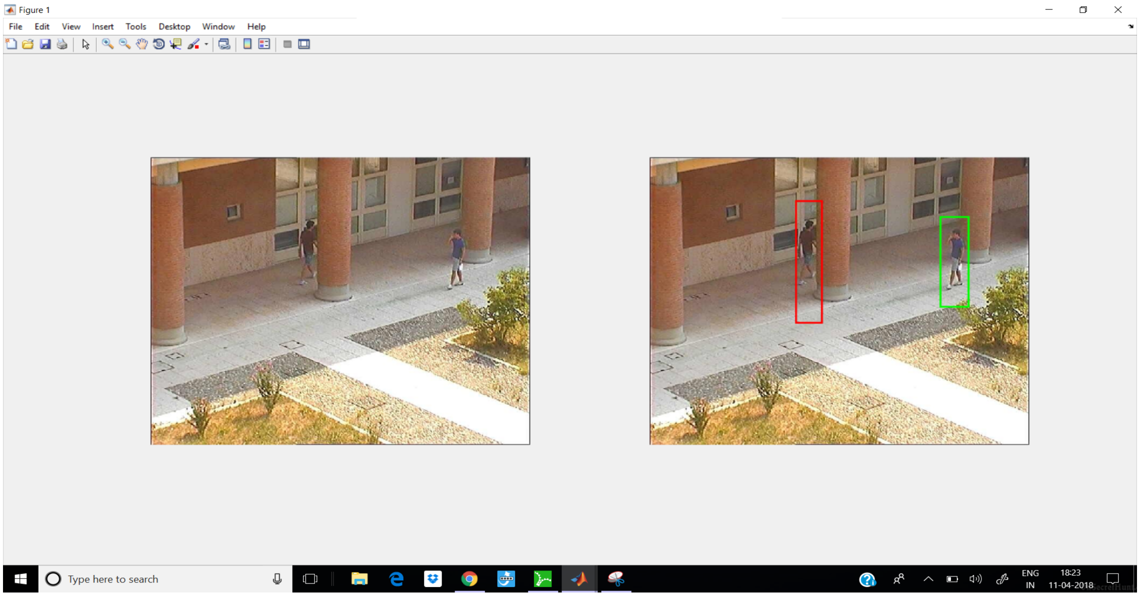

Figure 5.6: Object tracking

5.5 Summary

In the above system implementation description, details regarding the simulation and implementation of the project with sample coding for main module,search module were provided. Detailed screenshots of the application are also provided. The screenshots give a clear description of the applications various operations.

CHAPTER 6

SYSTEM TESTING

6.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product.It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

1.In system testing the behavior of whole system/product is tested as defined by the scope of the development project or product.

2.It may include tests based on risks and/or requirement specifications, business process, use cases, or other high level descriptions of system behavior, interactions with the operating systems, and system

3.System testing is most often the final test to verify that the system to be delivered meets the specification and its purpose.

4.System testing is carried out by specialists testers or independent testers.

5.System testing should investigate both functional and non-functional requirements of the testing.

6.2 Types of Testing

6.2.1 Acceptance Testing

Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.

6.2.2 Ad-Hoc Testing

Ad-Hoc testing involves practices that try to break the system on the scale of performance,robustness or scalability .

6.2.3 Assertion Testing

Assertion testing is a type of testing consisting in verifying if the conditions confirm the product requirements.

6.2.4 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

6.2.5 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.6 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2.7 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.8 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings,structure or language of the module being tested. It is a testing in which the software under test is treated, as a black box .you cannot. The test provides inputs and responds to outputs without considering how the software works.

6.3 Test Plan

A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from test engineers.Depending on the product and the responsibility of the organization to which the test plan applies, a test plan may include a strategy for one or more of the following:
1.Design Verification or Compliance test: to be performed during the development or approval stages of the product, typically on a small sample of units.
2.Manufacturing or Production test : to be perform ed during preparation or assembly

of the product in an ongoing manner for purposes of performance verification and quality control.
3.Acceptance or Commissioning test : to be performed at the time of delivery or installation of the product.
4.Service and Repair test : to be performed as required over the service life of the product.
5.Regression test : to be performed on an existing operational product, to verify that existing functionality did not get broken when other aspects of the environment are changed (e.g., upgrading the platform on which an existing application runs).

6.A complex system may have a high level test plan to address the overall requirements and supporting test plans to address the design details of subsystems and components.

6.4 Test Cases

6.4.1 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.Test Strategy and Approach Field testing will be performed manually and functional tests will be written in detail. Test Objectives
1. All field entries must work properly.
2.Pages must be activated from the identified link.
3.The entry screen, messages and responses must not be delayed.
Features to be tested
1.Verify that the entries are of the correct format
2.No duplicate entries should be allowed
3.All links should take the user to the correct page.

6.4.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications,e.g. components in a software system or one step up software applications at the test Results.All the test cases mentioned above passed successfully. No defects encountered.

6.4.3 Acceptance Testing

User Acceptance Testing is a critical phase of any system and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

6.5 Performance Testing

6.5.1 Existing System

The existing system uses mainly two techniques. The diagonal matrix transform is used to transform the pixel rate into a diagonal matrix by comparing the neighbouring pixel values. There is no improvement in the affine model compared to the DMT with spectral sharpening. The other technique used is the brightness transform function which is used to adjust the brightness in the image. By BTF pixel level correspondence cannot be achieved.

6.5.2 Proposed System

The proposed system is based on more efficient algorithms for the detection of object and reidentifying the object or person in the video. This system performs better than the existing system. The pixel level correspondence is achieved. It results in lesser overhead problems when compared to the existing system. There is improvement in the spectral sharpening also. The output obtained is as expected and very efficient. It also helps to improve the security and safety in the public places.

6.6 Summary

All the types of testing was performed on the system and the outputs were displayed without ant error. The path of tracking of the object is identified with a specific color and when the person is appearing in the other frame he is reidentified and the color of the tracked path is changed.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The system have described a novel approach for object tracking using unsupervised clustering, through an improved k-means algorithm with a confidence interval metric. Our experimental results demonstrate the contribution of this work, as a robust, real-time SIFT based approach that enforces the inherent temporal coherence across image frames, therefore handling difficult situations caused by significant object acceleration and partial occlusion. Although the proposed algorithm works well in general, its performance degrades when the scene contains a large number of moving objects. In the presence of many blobs that are close to each other, the method tends to group these clusters together. This situation sometimes leads to an increased variance for the newly grouped cluster which attracts more distinct clusters to group with it, ultimately grouping all the feature vectors into a single cluster. The reason is that the current approach uses the piecewise clustering technique where clustering is done separately for each piece and the cluster matching is done on the nearest neighbor basis. The problem of over-merging can be addressed if the information from the previous piece can be used while clustering a new piece of video. A Kalman filter can be used to estimate the position of cluster centers and these cluster centers can be used as seeds for clustering during the next set of frames. Gaussian Mixture Model (GMM) can also be used as a viable option to the improved k-means algorithm employed in this paper. The accuracy of the clustering can also be increased if the SIFT features are augmented to the feature vector used for clustering.

CHAPTER 8

REFERENCES

1. V. Ferrari, T. Tuytelaars, and L. Van Gool, Simultaneous Object Recognition and Segmentation by Image Exploration, Proc. Eighth European Conf. Computer Vision, pp. 4–54, 2004.

2. S. Lazebnik, C. Schmid, and J. Ponce, Sparse Texture Representation Using Affine-Invariant Neighborhoods, Proc. Conf. Computer Vision and Pattern Recognition, pp. 319–324, 2003.

3. K. Mikolajczyk and C. Schmid, Indexing Based on Scale Invariant Interest Points, Proc. Eighth Int l Conf. Computer Vision, pp. 525-531, 2001.

4. R. Polana and R. Nelson, Low level recognition of human motion, Proceedings of IEEE Workshop Motion of NonRigid and Articulated Objects, pp. 77–82, 1994.

5. T. J. Fan, G. Medioni, and G. Nevatia, Recognizing 3-D objects using surface descriptions, IEEE Trans. Pattern Recognit. Machine Intell. v. 11, pp. 1140–1157, 1989.

6. C. Bregler, Learning and recognizing human dynamics in video sequences,IEEE Computer Vision and Pattern Recognition (CVPR), June 1997.

7. Greenspan, J. Goldberger, A. Mayer, Probabilistic space-time video modeling via piecewise GMM, IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 26(3), pp. 384–396, March 2004.