Abstract—Automatic video surveillance is seeing widespread use in the remote monitoring of people. When we are dealing with automatic video surveillance, classification of detected objects and tracking them are the most crucial processes. This work proposes a Human Tracking system for static background, which first classifies all incoming objects into human and non-human entities. Then, it tracks all human entities till they leave the surveillance area. The task of human detection is accomplished using a technique called part-based template matching, which matches pre-defined set of human templates with the detected objects. For tracking, extraction of colour and intensity features is done; salient features are calculated and used for tracking of the human entities. Experimental evaluation has shown that the proposed system gives very good result for human detection and satisfactory result for tracking. Hence, this system can be used for indoor surveillance with static background.

Index Terms—Human Tracking System, Blob Detection, Frame Formation, Background Calculation, Background Subtraction, Human Detection, Human Tracking, Handling Occlusion

I. INTRODUCTION

Since last many years, the need for strong, disciplined and organized security has increased. This increasing concern of humans taking the security aspects has led to huge demand of automatic video surveillance systems.

The human monitoring has been almost replaced by automatic surveillance systems, also continuous researches are taking place in the field of video processing and surveillance to make the automatic surveillance system more accurate, fast, easy, intelligent and user-friendly.

An automatic video surveillance system has processes like video capturing, identifying occurrence of any object, tracking the object, analyzing its actions and finally give output in terms of alarms, visual effects or reports.

II. HUMAN TRACKING SYSTEM

The proposed system takes a video as input and tracks all the humans till they leave the surveillance area. The block diagram of proposed Human tracking system (HTS) is shown below in Figure 1.

© Elsevier, 2013
Fig. 1 Block Diagram of HTS

In section 2 we will discuss about the frame formation. Section 4 discusses about Background Subtraction. Following which sections 5, 6, 7 and 8 discuss about blob detection, Human Detection, human tracking and occlusion handling respectively.

III. FRAME FORMATION

When we talk about processing a video, it is actually not a video that we are dealing with but the video must be divided into sequence of images which are further processed. The speed at which a video must be divided into images depends on the implementation of individuals. From [1] we can say that, mostly 20-30 images (frames) are taken per second which are sent to the next phases for further processing.

IV. BACKGROUND SUBTRACTION

Background subtraction is one of the most critical phases of any automatic surveillance system as it is not only responsible for the extraction of moving objects but all the remaining phases process based on the output of this phase. There are basically three conventional approaches to moving object detection: temporal differencing, optical flow and background estimation methods. Temporal differencing [2] is very adaptive to dynamic environments, but generally is inefficient in extracting all relevant feature pixels [4]. Optical flow[2], [3], [10] can be used to detect moving objects even if the scene is dynamic, however most optical flow
computation methods are very complex and are not efficient for real-time algorithms and they also need specialized hardware. Background subtraction is a particularly popular method for human detection especially under those situations with a relatively static background. It detects moving regions in an image by simply deriving the difference between current image and a reference image usually background image is used. The differenting is done in a pixel-by-pixel fashion. However, it is not suitable for dynamic scenes like changes due to lighting effect, changing background or any other extraneous events.

The proposed system is made for static backgrounds. As seen above simple background subtraction is very preferable. Thus, HTS uses simple background subtraction. This phase is composed of 2 processes, first is background frame calculation and second is background separation. In first part, an average of 1st ten frames is taken as background frame. This background frame is subtracted from all the incoming frames in a pixel by pixel fashion to detect presence of any object.

\[
|P_{\text{curr\_frame}} - P_{\text{ref\_frame}}| > \text{threshold} \tag{1}
\]

V. BLOB DETECTION

The detected blob in the background subtraction phase undergoes 2 checks. Firstly, it is checked whether the blob has completely entered the surveillance area. Secondly, there is possibility that tiny blobs might be detected due to illumination changes. Such invalid tiny blobs are neglected in this phase.

VI. HUMAN DETECTION

Before the complexity of the human activity can be understood, it is very crucial to identify humans in an image or video. Once the human is detected, depending on the requirements, system can do further processing. Human detection can be done using many ways as mentioned below:

A. Shape Based/Template Matching

Here, use of complete or partial object silhouettes/shape is made to classify objects and their actions. Here, full/partial template of an object say human (all the possible postures of human are required), car (taken from different angles), 2-wheeler, etc. are required to classify object into cars, humans or any other existing entity [9][10].

B. Feature Based Human Detection

HOG is one commonly used technique. Mostly Dalal and Trigs method is used to for the HoG Human detection [5], [7]. It uses the fact that shape of an object can be well represented by a distribution of local intensity gradients or edge directions. This is done by dividing the image in small spatial parts (cells) and finding the histograms of edge orientations over all the pixels of the cell. For classification, a dataset of human and non-human examples is created, and a linear classifier is trained using SVM (Support Vector Machine) on the gradient histogram features from the two classes.

Another used method for human detection is compare object’s facial features [8]. Features like angles, size, distance between eyes, nose and lips and many other features are matched to classify an object as human [11].

The proposed system is using a part-based template matching system. Here, we match the lower portion of detected object with our leg – templates. One of the provided leg templates has been shown by matrix “(2)”. If the object’s lower portion, which is 1/3rd of the object height, matches the defined templates more than threshold value than we classify the object as human otherwise it is classified as non-human entity.

HTS considers only human entities for tracking. Hence all the detected objects that are classified as humans are considered for tracking phase rest all are neglected.

\[
\begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \tag{2}
\]
VII. HUMAN TRACKING

Once an object has been detected in a video, the next step is tracking it. Tracking is a process of locating a moving object (or multiple objects) overtime in a video. Tracking algorithms can be classified into:

A. Point Tracking

Such techniques represent an object by a point mostly its centroid or by set of specific points chosen wisely from the object. The flow/movement of the chosen point(s) is studied for few subsequent frames and the analysis is stored about their movement pattern (like direction, speed, etc.). Based on the analysis done, the displacement of the object is estimated for the next incoming frame. When the next frame comes, if there is any object detected at the forecasted pixels than it is declared to be the same object. In general, the point representation is suitable for tracking objects that occupy small regions in an image and travel with constant speed and in uniform direction. Ex: Optical Flow tracking [3]

B. Kernel Tracking

A new approach toward target representation and localization - For both kernel based trackers[13] and particle filtering trackers[14], a popular method is to exploit color distributions in simple regions (region-wise density models) of pixel intensities. When images are to be directly compared in this vector space, the images are required to have the same number of pixels, and pixels with the same relative spatial position in different images are assumed to measure corresponding features. Usually Bhattacharya distance equation is used. It has good color histogram model and distance measure but due to use of the kernels, the center of the target in the current frame has to be covered by the target model in the previous frame. Otherwise, the local maximum of the Bhattacharyya coefficient would not be a reliable indicator.

C. Contour Tracking

Contour representation defines the boundary of an object [6]. Tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Silhouette and contour representations are suitable for tracking complex non-rigid shapes. Ex. Shape Matching [12]

D. Visual Feature Matching

Tracking is done using different features of the object. Features chosen for tracking include its color, intensity, orientation and any visible feature [15, 8].

E. Filtering and Data Association

Initially features are extracted based on particle filter (for a group of particle) than based on the feature at time $t$, the features of $t+1$ are estimated. In next frame the features are matched and updating of the features happen and continue. Ex: particle filter [14], kalman filter [13].

Through HTS we are making an effort to build system for indoor surveillance. While making surveillance system for indoor few things to be kept in mind are:

1) The background for indoor areas is mostly static (HTS has similar background constraint)
2) There are limited numbers of object that appear simultaneously in an indoor area. Current HTS implementation has limit on number of human entities that can be tracked.
3) In an indoor tracking system, the incoming object is very clearly visible and its features can be easily fetched. Hence, we can go for feature tracking.

As we are going with indoor surveillance system, the objects features can be clearly extracted and used for tracking. Hence we are going to explore Visual based tracking in HTS and try to track humans in the incoming video.

The tracking algorithm of HTS works in 2 phases:-

1) First phase calculates and stores the salient features.
2) Second phase, compares the stored features with the features calculated in subsequent frames and tracks the humans.

To accomplish the first phase which includes calculating salient features, below steps are performed:

1) Extract 4 colors-red, green, blue, yellow and Intensity features
2) For each color, perform below steps:
3) Convert rgb fetched color feature to lab format.
4) Subtract standard red, green, blue and yellow values from the “a” and “b” component of lab format.
5) Find Euclidean distance of the basic colors to the above derived values. This will give us the prominence factor of each
6) Find center – surround of each color. Thus, we get 4 features Cr, Cg, Cb and Cy.
7) Hue value is calculated.
8) For intensity, 2 features i.e. center–surround and surround–center are calculated. Center-surround is shown below:

\[
\begin{align*}
\text{Int}_{\text{On}, x, \sigma}(x, y) &= \max\{\text{center}(x, y, s) - \text{surround}(x, y, s, \sigma), 0\} \\
\text{Int}_{\text{Off}, x, \sigma}(x, y) &= \max\{\text{surround}(x, y, s, \sigma) - \text{center}(x, y, s), 0\}
\end{align*}
\]  

(3)

Surround-center is simple inverse of center-surround.

To accomplish the second phase which is comparing and tracking the human objects, below steps are performed on all subsequent frames:
1) Find all human blobs in subsequent frame.
2) For detected human blobs, calculate features as shown in first phase.
3) For each blob, compare its salient feature with the stored salient features.
4) If there is a match, then mark the object of subsequent frame with the same color as it was marked in the previous frame.

VIII. HANDLING OCCLUSION

There is a possibility that 2 or more objects might occlude. Occlusion is a state where one or more objects are hidden by other object. There can be 2 types of occlusion, complete and partial occlusion.

In partial occlusion, both the objects will be displayed partially. Even though the 2 objects are occluded but we will try to match the most salient features calculated in previous frames with the occluded blob in current frame. The occluded blobs will be considered as a single blob and whoever’s features match with the occluded blob, the blob will be highlighted with that colored rectangle.

In complete occlusion, one object will completely or almost hide the other object; hence there will be only one object visible. Hence that object will be highlighted with a rectangle of the color with which the visible object was highlighted in previous frame.

IX. EXPERIMENTAL RESULTS

A. Speed

For processes like frame formation, Background Subtraction and blob detection, we have used very fast and efficient methods. When it comes to human detection, we extract the lower portion of the detected blob, scale down and match with the given leg-templates. Once a human has been identified it must be tracked by the HTS. For tracking, feature extraction must be done for all the human blobs followed by salient features calculation which are matched with the salient features of the human blobs in subsequent frames. This process of human detection and tracking using feature matching in every frame is very time consuming. To overcome this frames at regular interval are taken for processing. Thus, this implementation works satisfactory at Speed.

![Figure 2 Result of HTS](image-url)
B. System Accuracy
The proposed system HTS has been tested on many data sets. Shown below are some of the output screenshots. The system is able to track single and multiple humans (tested maximum for 4). Fig. 2(a) shows tracking of single human. In Fig. 2(b), two humans are being tracked by different colors. When the two humans have crossed each other, the tracking is still working as the colored rectangles are still with the respective humans as shown in Fig. 2(c). Also, Fig. 2(c) shows that the 2-wheeler is not tracked as it is non-human entity. The overall system has shown 95% of success rate.

X. CONCLUSION
An attempt is made to design and develop a HTS (Human Tracking system) that can accepts a video of a surveillance area, detects the human entities, and track them till they leave the surveillance area. We are getting good results of tracking even if the human is changing its orientation in the input video. Due to complex processing like human detection and tracking, the system takes a bit more time. But by skipping few frames at regular interval results are up to the mark. So, the Human tracking system is achieving good results. It is showing accuracy above 90 % Overall the system is giving good performance for the tested video dataset.

XI. FUTURE WORK
The HTS works only for static background. In future we can expand it to work for dynamic background. HTS fails if the entity is wearing same color as the background. HTS also requires all its human entities to be of different color in order to in future it can be made more robust by adding few more features for tracking.

REFERENCES