A Novel Framework for Real Time Object Tracking Systems

*Sanketh Datla, **Abhinav Agarwal, *Rajdeep Niyogi, **Barjeev Tyagi
*Department of Electronics and Computer Engineering
**Department of Electrical Engineering
Indian Institute of Technology Roorkee, India.
{sankipec,abhiapee,rajpfec,btyagfee}@iitr.ernet.in

Abstract— Object tracking is fundamental to automated video surveillance, activity analysis and event recognition. In real-time applications only a small percentage of the system resources can be allocated for tracking, the rest being required for high-level tasks such as recognition, trajectory interpretation, and reasoning. There is a desperate need to carefully optimize the tracking algorithm to keep the computational complexity of a tracker as low as possible yet maintaining its robustness and accuracy. This paper proposes a novel framework which attempts to attain a light weight tracking system by reducing undesirable and redundant computations. The frames of the video are passed through a preprocessing stage which transmits only motion detected blocks to the tracking algorithm. Further frames containing little motion in the search area of the target object are detected in preprocessing stage itself and are blocked from further processing. Our experimental results demonstrate that the throughput of the new proposed tracking system is exceptionally higher than the traditional one.

Index Term— Video Surveillance, Object Tracking, Appearance models, Block Cluster, Performance.

I. INTRODUCTION

The increasing availability of video sensors and high performance video processing hardware opens up exciting potential for tackling many video surveillance problems. It is important to develop robust video surveillance techniques which can process large amounts of data in real time. As an active research topic in computer vision, visual surveillance in dynamic scenes attempts to detect, recognize and track certain objects from image sequences, and more generally to understand and describe object behaviors. The aim is to develop intelligent visual surveillance to replace the traditional passive video surveillance that is proving ineffective as the number of cameras exceeds the capability of human operators to monitor them. In short, the goal of visual surveillance is not only to put cameras in the place of human eyes, but also to accomplish expressways, detection of military targets. A survey on visual surveillance of object motion and behaviors is given in [1]. Real-time object tracking is the critical task in many computer vision applications such as surveillance [2, 3], augmented reality [4], object-based video compression [5], and driver assistance. In its simplest form, tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene [6]. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video. Additionally, depending on the tracking domain, a tracker can also provide object-centric information, such as orientation, area, or shape of an object. In the recent decade, different techniques on object tracking are proposed [7,8,9]. Perhaps, the most extensive coverage of different object tracking techniques is covered in [6].

Object tracking is highly computationally intensive task. In real-time applications, only a small percentage of the system resources can be allocated for tracking, the rest being required for the preprocessing stages or to high-level tasks such as recognition, trajectory interpretation, and reasoning. Therefore, it is desirable to keep the computational complexity of a tracker as low as possible. There is a need to optimize tracking algorithms for its real time applications to be feasible. In this paper, we propose a novel framework for object tracking systems that aims at achieving a light weight, computationally inexpensive tracking system. As against traditional tracking systems, the proposed system works in two phases namely, preprocessing phase and iterative tracking phase.

The reminder of this paper is organized as follows. Section II describes the general framework of traditional and proposed tracking systems and outlines the differences between them. In Section III, the first stage of our proposed framework, the Pre-processing phase is described. In section IV, the second phase called Iterative tracking phase is described. Section V shows the experimental results demonstrating the superiority of our system over the existing one and section VI concludes the paper.

II. TRADITIONAL VS PROPOSED TECHNIQUE

We first describe the traditional framework of tracking systems briefly and then illustrate our proposed framework of tracking systems. We clearly outline the differences and advantages of the proposed technique over the existing one.

A. General framework of Traditional tracking systems

The general framework of a typical automatic VS system is shown in Figure 1. The main video processing stages include background modeling, object segmentation, object tracking, behaviors analysis. Every video surveillance system starts with Motion detection. It aims at segmenting the moving objects from the rest of the image. The process of Motion detection involves Environment modeling, Motion segmentation and object classification. Subsequent processes like tracking and behavior analysis are dependent on it. In [1,10], a good description of video processing in surveillance framework is presented. A brief review of these stages is given below:
1) Environment Modeling: The continuous construction and updating of background models are indispensable to any video surveillance system. The main problem in background modeling is to automatically recover and update background from a dynamically changing video sequence. Changes in the scene such as moved objects, parked vehicle etc. need to be carefully handled so that interesting foreground targets are detected. In paper [11], a framework is presented for recovering and updating background images based on a process in which a mixed Gaussian model is used for each pixel value. An online estimation is used to update background images in order to adapt to illumination variance and disturbance in backgrounds. Paper [12] proposes a simple layered modeling technique to update a background model. In addition, important issues related to background updating for visual surveillance are discussed.

2) Motion Segmentation: Motion Segmentation in video aims at detecting regions corresponding to moving objects such as humans and vehicles. Most of the segmentation methods use either spatial or temporal information in the image sequence. Some of the commonly employed approaches for motion segmentation are background subtraction, Temporal differencing and optical flow. Background Subtraction is the most popularly used method for motion segmentation, especially in a relatively static background. It detects moving regions in a video by taking the difference between the current frame and the reference background frame. It is simple in approach but is highly sensitive to background illumination changes. In Temporal differencing, the pixel-wise differences between two or more consecutive frames in an image sequence are used to extract moving regions.

3) Object Classification: The moving regions which are identified in the above steps may correspond to different targets. For example, the surveillance of road traffic scenes include humans, vehicles and other moving objects like flying birds and moving clouds, etc. To further track objects, object classification is essential. Object classification is purely a pattern recognition issue. There are two main approaches for object classification namely shape based classification and motion based classification. In Shape-based classification, Different descriptions of shape information of motion regions such as points, boxes, silhouettes and blobs are available for classifying moving objects. For example, in paper [13], the area of image blobs is used as classification metric to classify all moving-object blobs into humans, vehicles and clutter. In Motion based classification, Human motion shows a periodic property, so this can be used as strong cue for classification of moving objects. Paper [14] presents a similarity based technique to detect and analyze periodic motion. Here the self-similarity of a moving object is computed, as it evolves over time. For periodic motion, the self-similarity measure is also periodic. The tracking and classification of moving objects are implemented using periodicity.

4) Object Tracking: In a tracking scenario, an object can be defined as anything that is of interest for further analysis. The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. The tasks of detecting the object and establishing correspondence between the object instances across the frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm, and then the tracker corresponds to the objects across frames. In the latter case, the object region and the correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. Tracking methods are divided into four major categories: region-based tracking, active-contour-based tracking, feature based tracking, and model-based tracking[1]. Region-based tracking algorithms track objects according to variations of the image regions corresponding to the moving objects. For these algorithms, the background image is maintained dynamically and motion regions are usually detected by subtracting the background from the current image. Active contour-based tracking algorithms track objects by representing their outlines as bounding contours and updating these contours dynamically in successive frames. These algorithms aim at directly extracting shapes of subjects.

![Diagram](image.png)

The temporal difference technique is very adaptive to changes in dynamic environment and another advantage is that it does not make assumptions about the scene. Paper [13] detects moving objects in real video streams using temporal differencing. Here instead of consecutive frames, the video frames separated by a constant time $\delta t$ are compared to find regions which have changed. In Optical flow methods, The characteristics of flow vectors of moving objects over time are used to detect moving regions in an image sequence. Optical flow based methods can be used to detect independently moving objects even in the presence of camera motion.
and provide more effective descriptions of objects than region-based algorithms. Feature-based tracking algorithms perform recognition and tracking of objects by extracting elements, clustering them into higher level features and then matching the features between images. Model-based tracking algorithms track objects by matching projected object models, produced with prior knowledge, to image data. The models are usually constructed off-line with manual measurement or computer vision techniques. After successfully tracking the moving objects from one frame to another in an image sequence, behavior analysis is done. It involves the analysis and recognition of motion patterns, and the production of high-level description of actions and interactions.

B. Proposed framework for tracking systems

Here Tracking is implemented in two phases namely Preprocessing stage (Phase I) and Iterative tracking stage (Phase II). The frames are first passed to a preprocessing stage called Block Motion Detection phase, where motion in the frames are detected in block wise sense and the static blocks(i.e. the blocks containing no motion with respect to corresponding blocks in the previous frame) are removed from the frames for further processing and then passed to Iterative tracking phase. Further, if the search region in the frame (i.e. the region surrounding the object in the previous frame) contains all the static blocks, then the entire frame is removed for further processing and is blocked from passing to the second phase. The steps in the preprocessing stage are broadly divided into Frame Partitioning, Block-wise similarity detection and Block retrieval phase. These are shown in Fig. 2. These steps are elaborated in the next section.

The second phase is called Iterative tracking phase which operates only on the motion detected blocks that are passed on to it by the above pre-processing stage. It is so named because the object tracker first moves to the centre of every received block in search region and searches for the appearance similarity of the object tracked in the previous frame. If the appearance similarity is greater than a certain threshold in a block, then that object tracker iteratively updates the object position. The object position update is done in a direction such that the appearance similarity is increased. The stages in this iterative tracking phase are logically similar to the stages in the traditional tracking system but differ in some aspects which will be explained later in the next section.

III. PREPROCESSING (PHASE I)

A. Division of Frames into blocks

The incoming frame is divided into non-overlapping image blocks. The size of the block is an important characteristic and will affect the tracking performance. Bigger blocks are less sensitive to noise, while smaller blocks produce better contours. The next two factors are the amount of noise in the video frames and the texture of the objects and the background. The texture of the objects leads to the so called aperture problem( also found in block matching algorithms). The aperture problem appears in situations where the objects of interest have uniform color. The blocks which are inside the objects do not appear as moving because all of the blocks around them have the same color. When the uniform color regions consist of fewer blocks, there is a greater chance that their motion will be detected because some overlapping with non-uniform color regions is likely. A bigger block size can be used to overcome the aperture problem.

B. Block wise similarity detection (Scope for parallelization)

The similarity between the corresponding blocks in the successive frames is computed. Some of the similarity measures that can be used here are Sum of absolute difference (SAD), Mean Absolute Difference (MAD) and Normalized cross correlation (NCC). We have used Normalized cross correlation in our experiments. The correlation between two blocks is computed in MATLAB using the following equations.

\[
a = a - \text{mean}(a) \\
b = b - \text{mean}(b) \\
c = \frac{\text{sum}(a.*b))}{\sqrt{\text{sum}(a.*a))*\text{sum}(b.*b))}
\]

The function mean calculates the mean of the input matrix. The function sum calculates sum of all elements of the input vector. Here 'a' is the matrix representing the pixel intensities of the block in the present frame and 'b' the matrix representing the pixel intensities of corresponding block in the previous frame. 'c' gives the correlation between the two
blocks 'a' and 'b'. The correlation is computed between all the blocks in the present frame and corresponding blocks in the previous frame. This step can be done in parallel. Several parallel programming models like OpenMP, MPI on clusters and Nvidia CUDA on Graphics processing unit (GPU) can be used for parallelization.

C. Block Retrieval
The above process produces N (N is equal to the number of blocks in the image) values ranging from 0 to 1 depending on the absolute difference of the two correlated images. A minimum value of correlation called motion threshold is defined to detect motion. In normal cases, motion can easily be detected when the measured minimum cross correlation value of all the N values is used to set the threshold. However, detection fails when images contain global illumination variations or during camera movement. If the correlation of a block with respect to corresponding block in the previous frame is more than the motion threshold, then the block is considered to be static (i.e. motion less) and is removed from the image for further computations. If all the blocks in the search region around the object to be tracked are detected as static, then the corresponding frame is removed and is not passed to the tracking phase. This saves a lot of redundant computation in the tracking phase and facilitates real-time tracking. Thus, only a subset of frames that too containing only a subset of blocks (i.e. the motion detected blocks) are passed to the tracking phase.

IV. ITERATIVE TRACKING (PHASE II)
A. Adaptive Block Background modeling
In this step, the background model is estimated for the blocks which are continuous around the search region. The group of continuous blocks surrounding a search region is named a Block cluster. In the case of tracking multiple objects, the background model is computed separately for different block clusters surrounding the objects. This would improve the tracking performance significantly as globalized lightning changes are nullified.

B. Block Cluster Flow Segmentation
The technique used in this Block cluster Flow segmentation step is background subtraction. The background model calculated in the previous step is subtracted from the corresponding group of blocks. The image obtained after subtraction is used as a reference image for tracking in the next frame. Several other conventional motion segmentation approaches like temporal differencing and optical flow methods may not give satisfactory results because the block clusters around the target object may change from frame to frame.

C. Tracking
The position of the reference image obtained from the above segmentation phase is updated over time for every incoming frame. The object tracker moves to every block in the cluster and a similarity measure is computed. If the similarity in a block is less than a certain threshold, then the tracker moves on to the next block in the cluster. If the similarity is greater than the threshold, then the object tracker maximizes the similarity iteratively by comparing the appearance similarity of the object detected in the previous block cluster and the appearance in the window of the present block. At each iteration, the similarity is computed such that the appearance similarity is increased. This process is repeated until convergence is achieved, which usually takes four to five iterations. After the convergence is attained, the block number corresponding to the tracked object location is passed on to the preprocessing stage which will be used to evaluate motion in the new search region for the next frame.

Fig. 3. Tracking person using traditional tracker. The frames 5, 12, 15, 21, 32 and 56 are shown.
V. RESULTS

The visual tracker implemented according to the proposed framework was applied to many sequences and for several applications. Here we just present some representative results. We performed experiments both using the proposed tracking framework and the traditional framework for evaluating their relative performances. Compared to the traditional implementation, this new system performed exceptionally well in terms of throughput. For example, for the image sequences shown in this Fig. 3, the new system could track well up to 25 frames/second, whereas the traditional system could track the same sequence only until 15 frames/second. The object tracker used in this image sequence is Kalman filter which is composed of two steps, prediction and correction step. The prediction predicts the approximate location of the object. The correction step computes the exact location of the object. The prediction and correction steps are shown by the red and green circles in the Fig. 3 respectively. Fig. 4 shows the object clusters used for tracking in our system. Fig. 5 shows the tracking of two persons moving in an outdoor environment using the traditional multiple tracker using mean shift iterations. Fig. 6 shows the tracking of same sequence using our proposed framework. The actual position of the object with respect to the entire image is calculated based using the two dimensional block ID ($B_x, B_y$), the centroid ($C_x, C_y$) and the block size ($s_1 s_2$) as per the below equations:

$$
\text{X-coordinate of the object detected} = (B_x - 1) \times s_1 + C_x.
$$

$$
\text{Y-coordinate of the object detected} = (B_y - 1) \times s_2 + C_y.
$$

VI. CONCLUSION

This paper presents a novel generalized framework for lightweight tracking which greatly reduces the computational complexity and facilitates real time tracking. The system described here ran successfully using several different object trackers. However, though the system performed very well for
many conventional trackers, it has certain limitations. All prevailing techniques used for different stages like Environment modeling, Motion segmentation and tracking may not give satisfactory results. For example, Motion segmentation techniques like temporal differencing and optical flow methods may not give satisfactory results because of the block clusters dynamically change in position and shape in every frame.

REFERENCES


