Volume 3, Issue 2, February 2018 ISSN (online): 2456-0006
International Journal of Science Technology Management and Research
Available online at: www.ijstmr.com

Tracking Learning Detection Survey

Mr. Rohan S Gopale
HOD, Dept. of Computer Technology
Kala Vidya Mandir Institute of Technology
Mumbai, India

Abstract: ***The goal of this paper is to review tracking methods, classify them into different categories, and identify new trends. Object tracking, in general, is a challenging problem. Difficulties in tracking objects can arise due to abrupt object motion, changing appearance patterns of both, object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions, and camera motion. Tracking is performed in the context of higher-level applications that require the location and/or shape of the object in every frame. Typically, assumptions are made to constrain the tracking problem in the context of a particular application. In this survey, we categorize the tracking methods on the basis of the object and motion representations used, provide detailed descriptions of representative methods in each category, and examine their pros and cons.***

Keywords-:Object tracking; 3D; 2D

I. INTRODUCTION

Object tracking is an important task within the field of computer vision. The proliferation of high-powered computers, the availability of high quality and inexpensive video cameras, and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms.

In the 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. 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. Tracking object can be complex due to loss of information to projection of 3D world on 2D image, noise, complex object shape, nature of object, partial and full object occlusions etc.

II. TRACKING APPROACHES

There are three key steps in video analysis: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior. There are two sub task in tracking:

1. Build some model of what you want to track
2. Use what you know about where the object was in the previous frame(s) to make predictions about the current frame and restrict the search.

Repeat the two sub tasks possibly updating the model. [1]

- **Tracking People by Learning Their Appearance**

One of the great open challenges in computer vision is to build a system that can track people. A reliable solution opens up tremendous possibilities, from human computer interfaces to video data mining to automated surveillance. This task is difficult because people can move fast and unpredictably, can appear in a variety of poses and clothes, and are often surrounded by clutter. Because of the technical challenge and the attractive rewards, there exists a rich body of relevant literature.

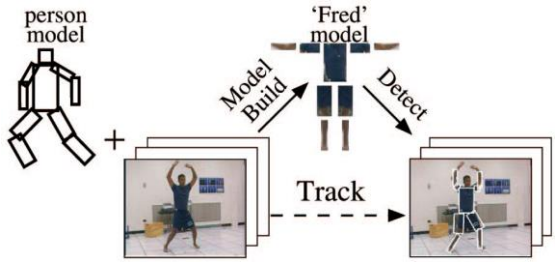

Fig. 1. Model-based people tracker.

This work describes a system that, given a video sequence of possibly multiple people, automatically tracks each person. The system first builds a puppet model of each person's appearance and then tracks by detecting those models in each frame. Since methods for detecting pictorial structures are well understood, it focuses primarily on algorithms for learning them (from a given video sequence). This approach describes two methods of building appearance models. One is a bottom-up approach that looks for candidate body parts in

each frame. It clusters the candidates to find assemblies of parts that might be people. Another is a top-down approach that looks for an entire person in a single frame. It assumes people tend to occupy certain key poses, and so it builds models from those poses that are easy to detect. Once we have learned an appearance model by either method, it detects it in each frame to track a person. [2]

- **Object tracking using SIFT features & mean shift**

In this the proposed tracking algorithm is an effective integration of mean shift and SIFT feature tracking. The proposed approach will apply a similarity measurement between two neighboring Frames in terms of color and SIFT correspondence. Technically, a track will be made if mean shift and SIFT feature tracking lead to approximate probability distributions (e.g., intensity and color) within the corresponding region in the next image frame. An expectation–maximization algorithm is employed in order to pursue a maximum likelihood estimate using the measurements from mean shift and SIFT correspondence**.** Object tracking is equivalent to similarity search across two neighboring image frames. Given the predicted target's position in the current frame and its uncertainty, the measurement task assumes the search of a confidence region for the target candidate that is the most similar to the target model. The similarity measure conducted here is based on color information. The sample points in the current frame are denoted by

$$Ix = (Xi, Ui)^{N}{}_{(i-1)}$$

where Xi is the 2D coordinates and Ui is the corresponding feature vector (e.g., RGB colors of sample points). The sample points of the target image are

$$Iy = (Yj, Vj)^{M}{}_{(j-1)}$$

encoding the 2D coordinates and the corresponding feature vector. The targets can be in the joint feature-spatial space.

- **Proposed algorithm:**

The entire algorithmic flowchart can be summarized as follows:
(1) Define a rectangle on the region of interest in the first frame of a video sequence.
(2) Compute the color histogram of this region, whilst extracting SIFT features within this region.
(3) In the second frame, start from the former location and examine the surroundings for similarity measure .The sum of squared difference (SSD) method is applied for SIFT feature correspondence across frames.
(4) Launch the proposed expectation–maximization algorithm to search for an appropriate similarity region whilst minimizing the distance between the detected locations by mean shift and SIFT correspondence,
Respectively.
(5) Iterate the above steps till the difference between two mean shifts is smaller than a threshold. [3]

- **Fast Occluded Object Tracking by a Robust Appearance Filter**

This paper is concerned with tracking rigid objects in image sequences, using template matching. In essence, object tracking is the process of updating object attributes over time. The complete set of attributes includes position, motion, shape, and appearance. The appearance is comprised of a set of photometric features representing the object region in a frame. To suppress noise and to achieve tracking stability, the attributes are smoothed by a temporal filter like the Kalman filter or Monte Carlo filters.
This method presents a new template updating algorithm that satisfies the two qualities: simplicity and robustness. Simplicity implies that the algorithm is easy to implement and has the minimum number of parameters. Robustness implies the ability of the algorithm to track objects under difficult conditions which include: severe occlusions and lighting changes, changing of object orientation or viewpoint, background clutter and the presence of other moving objects in the scene, a moving camera, and non translational object motion like zooms and rotations.

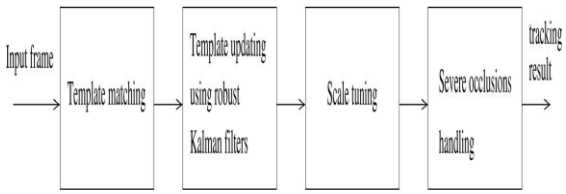

Fig. 2. The data flow diagram of one tracking iteration.

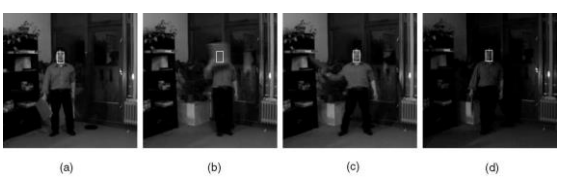

Fig. 3. Tracking results with occlusions and abrupt lighting changes. Color invariants were used.

The multivalve appearance template is smoothed temporally by robust Kalman filters during tracking. In particular, outliers due to partial occlusions are down weighted by an observation model using a robust error norm and the Mahalanobis distance. The residual information is exploited to tune the scale parameters automatically. When photometric invariants are used, the method can achieve the insensitivity to shadow and abrupt changes of illumination conditions. [4]

➢ Fast Multiple Object Tracking via a Hierarchical Particle Filter

A very efficient and robust visual object tracking algorithm based on the particle filter is presented. The method characterizes the tracked objects using color and edge orientation histogram features. While the use of more features and samples can improve the robustness, the computational load required by the particle filter increases.

Particle Filter

The particle filter is a Bayesian sequential importance sampling technique, which recursively approximates the posterior distribution using a finite set of weighted samples. It consists of essentially two steps: prediction and update.

Color Rectangle Features

In this method, the grayscale image was converted to integral image format. i.e. an image in which at each pixel the value is the sum of all pixels above and to the left of the current position. The sum of the pixels within any rectangle can then be computed in four table lookup operations on the integral image.

Edge Orientation Histogram

Combination of color with a contour model gives faster and more robust tracking. The color histogram and an ellipse shape model are combined for object tracking. In this, it uses the edge orientation histogram for the purpose of simplicity, efficiency and generalization. To detect edges, it first converts color images to grayscale intensity images.

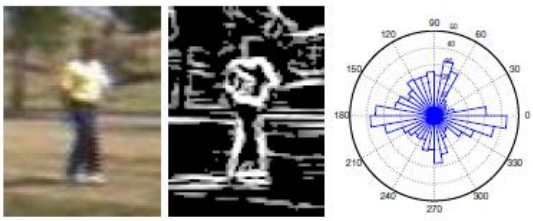

Fig 4. Edge orientation histogram. (Left) Example image. (Center) Edge strength image. (Right) Polar plot of edge orientation histogram.

Cascade of Features

The combination of the color information and edge orientation histogram achieves excellent performance in term of speed and accuracy.

Particle Filter Tracking

In each iteration, the particle filter tracking algorithm consists of two steps: prediction and update. The state of the particle filter is defined as X =(X, Y, Sx, Sy), where X, Y indicate the location of the target, Sx, Sy the scales in the x and y directions. In the prediction stage, the samples in the state space are propagated through a dynamic model. The update stage applies the observation models to estimate the observation likelihood for each samples, i.e., the weights of samples in the case of the bootstrap filter. [5]

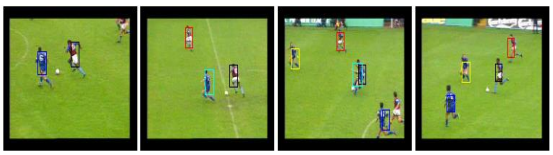

Fig.4 Results of the proposed particle filter based multiple object tracking for the football game sequence.

➢ A Linear Programming Approach for Multiple Object Tracking

Linear programming (LP) is another approach that can be used for more efficient search in object tracking. At each frame it represents all the possible spatial locations of each object from the observations as nodes based on attributes of the objects. However, if one object occludes another, there is a break in the track of one object. It is having a special occlusion node that allows the path for an occluded object to be accounted for in that particular frame if there is no other non-overlapping location for the potentially occluded object. This graph forms the basis for formulating a cost function based on all the possible paths and constraints, leading to a linear program that may be efficiently solved. The algorithm optimizes the states for all the objects together. Thus, it finds consistent paths for all the objects over a window of video frames and assigns a meaningful interpretation of location or status of occlusion to each object as described more formally below.

In Fig. 5, an object's possible location and appearance states are represented as round nodes. For a given frame, hypothesized locations (i.e., observations) for each object may be different, and therefore the sub-network for each object may contain a different number of nodes. The rectangular nodes in Fig. 5 are the occlusion nodes that provide a node to represent that an object is occluded and does not have a spatial location. A source node and a sink node, shown as diamond nodes in Fig. 1 are also included for each object sub-network to represent the start and end of the object tracking sequence. Sink nodes are included just for convenience; they do not correspond to states of objects. The solid arcs between nodes indicate possible state transitions. A connected set of nodes between a source and sink node represents the spatial trajectory of an object. [6]

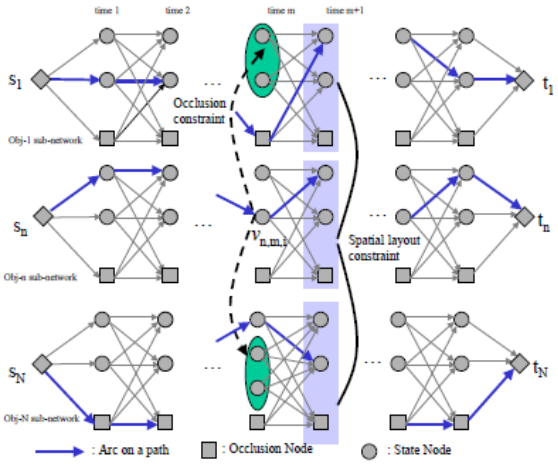

Fig.5 The network model for multiple object tracking.

- **Tracking the Invisible: Learning Where the Object Might be**

This techniques propose a method to learn supporters which are, be it only temporally, useful for determining the position of the object of interest. This approach exploits the General Hough Transform strategy. It couples the supporters with the target and naturally distinguishes between strongly and weakly coupled motions. By this, the position of an object can be estimated even when it is not seen directly (e.g., fully occluded or outside of the image region) or when it changes its appearance quickly and significantly The core idea is depicted in Fig. 7. First, local image features from the whole image are extracted (yellow points). Given the position of the object of interest in the frame, these image features are usually divided into object points and points belonging to the background (see Fig. 7b). Object points lie on the object surface and thus always have a strong correlation to the object motion (green points). Background points, e.g., points on other independently moving objects or in the static background, are considered to carry no information about the object position (blue points).

Supporters: They are features which are useful to predicting the target object positions. They at least temporarily move in a way which is statistically related to the motion of the target (red points). A supporter can be very strong (comparable to an object feature), e.g., a wristwatch on a hand holding the target; or quite weak when the coupling with the target motion is not that outspoken. The goal of our algorithm is, in other words, to find such local image features which help to predict the position of the target. [7]

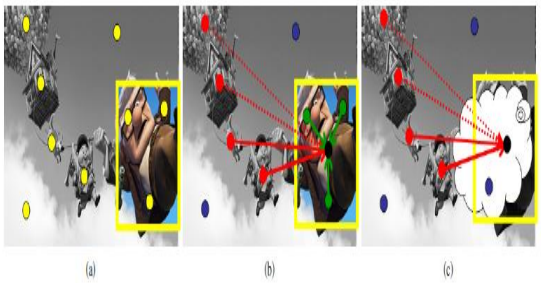

Fig.6 (a) A frame with the target object marked. (b) Supporters are features that vote for the position of the object, since their motion appears correlated. They can belong to the object itself (green) or not (red). Uncorrelated features (blue) are discarded. (c) Even if the object cannot be tracked based on its appearance

Tracking Learning Detection

In this technique object is defined by its location and extent in a single frame. In every frame that follows, the task is to determine the object's location and extent or indicate that the object is not present. This method build a novel tracking framework (TLD) that explicitly decomposes the long-term tracking task into tracking, learning, and detection. The tracker follows the object from frame to frame. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The learning estimates the detector's errors and updates it to avoid these errors in the future.

Fig. 7 Given a single bounding box defining the object location and extent in the initial frame (LEFT), our system tracks, learns, and detects the object in real time. The red dot indicates that the object is not visible.

The detector is evaluated in every frame of the video. Its responses are analyzed by two types of "experts":1) P-expert— recognizes missed detections, and 2) N-expert—recognizes false alarms. The estimated errors augment a training set of the detector, and the detector is retrained to avoid these errors in the future. TLD is a framework designed for long-term tracking of an unknown object in a video stream.

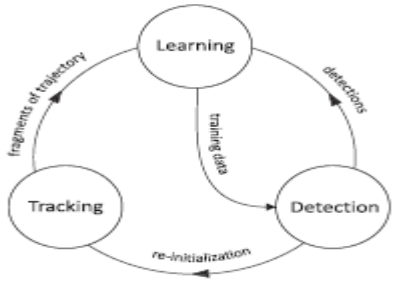

Fig 8 The block diagram of the TLD framework.

Its block diagram is shown in Fig. 7. The components of the framework are characterized as follows: Tracker estimates the object's motion between consecutive frames under the assumption that the frame-to-frame motion is limited and the object is visible. The tracker is likely to fail and never recover if the object moves out of the camera view. Detector treats every frame as independent and performs full scanning of the image to localize all appearances that have been observed and learned in the past. As with any other detector, the detector makes two types of errors: false positives and false negative. Learning observes the performance of both tracker and detector, estimates detector's errors, and generates training examples to avoid these errors in the future.

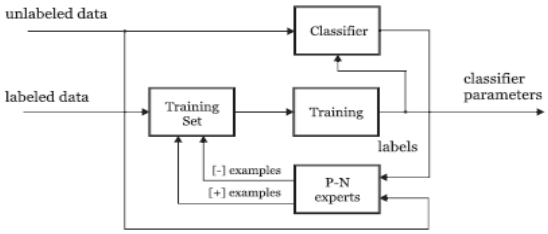

Fig. 9 The block diagram of the P-N learning.

The P-N learning consists of four blocks.

1. A classifier to be learned,
2. Training set—a collection of labeled training examples,
3. Supervised training—a method that trains a classifier from a training set,
4. P-N experts—functions that generate positive and negative training examples during learning.[8]

CONCLUSIONS

In this paper, many different techniques for single/multiple object tracking have been discussed. Note that all these methods require object detection at some point. We provide detailed summaries of object trackers, including discussion on the object representations, motion models, and the parameter estimation schemes employed by the tracking algorithms.

REFERENCES

[1] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey." Acm Computing Surveys (CSUR) 38.4 (2006): 13.

[2] Ramanan, Deva, David A. Forsyth, and Andrew Zisserman. "Tracking people by learning their appearance." Pattern Analysis and Machine Intelligence, IEEE Transactions on 29.1 (2007): 65-81.

[3] Zhou, Huiyu, Yuan Yuan, and Chunmei Shi. "Object tracking using SIFT features and mean shift." Computer Vision and Image Understanding 113.3 (2009): 345-352.

[4] Nguyen, Hieu Tat, and Arnold WM Smeulders. "Fast occluded object tracking by a robust appearance filter." Pattern Analysis and Machine Intelligence, IEEE Transactions on 26.8 (2004): 1099-1104.

[5] Yang, Changjiang, Ramani Duraiswami, and Larry Davis. "Fast multiple object tracking via a hierarchical particle filter." Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Vol. 1. IEEE, 2005.

[6] Jiang, Hao, Sidney Fels, and James J. Little. "A linear programming approach for multiple object tracking." Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007.

[7] Grabner, H., Matas, J., Van Gool, L., & Cattin, P. (2010, June). Tracking the invisible: Learning where the object might be. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 1285-1292). IEEE.

[8] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-learning-detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.7 (2012): 1409-1422.