ISSN 2229-5518

Object Tracking Based On Tracking-Learning-Detection

Rupali S. Chavan, Mr. S.M.Patil

Abstract—-In this paper; we present a novel tracking framework (TLD) approach for long-term tracking of unknown objects in a video stream. The object is defined by its location and extent in a single frame. 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 detector's errors and updates it to avoid these errors in the future. A novel learning method (P-N learning) which estimates the errors by a pair of "experts": (i) P-expert estimates missed detections, and (ii) N-expert estimates false alarms. The learning process is modeled as a discrete dynamical system and the conditions under which the learning guarantees improvement are found.

Index Terms— Long term object tracking, learning from video, real-time, P-N learning, TLD framework.

1 INTRODUCTION

IN computer vision develop mathematical techniques in order to extract information about physical objects based on camera images. Computer vision methods are applied to optical character recognition, quality inspection, robot guidance, scene reconstruction and object categorization. One domain of research in computer vision is object tracking, in which methods are studied that estimates the location of targets in consecutive video frames. The proliferation of high powered computers, the availability of high quality and inexpensive video cameras, and the need for automated video analysis have drawn interest to applying object tracking algorithms in automated surveillance, automatic annotation of video data, human-computer interaction, traffic monitoring and vehicle navigation .

Consider a video stream taken by a hand-held camera depicting various objects moving in and out of the camera's field of view. Given a bounding box defining the object of interest in a single frame, our goal is to automatically determine the object's bounding box or indicate that the object is not visible in every frame that follows. The video stream is to be processed at frame-rate and the process should run indefinitely long. This task is considered as long-term tracking task. To enable the long-term tracking, there are a number of problems which need to be addressed. The key

Tracking-Learning-Detection (TLD) is the long term tracking task into three sub-task tracking, learning and detection. Each sub-task is addressed by a single component and components operate simultaneously. 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 detector's errors and updates it to avoid these errors in the future. A novel learning method (P-N learning) which estimates the errors by a pair of "experts": (i) P-expert estimates missed detections, and (ii) N-expert estimates false alarms. The learning process is modeled as a discrete dynamical system and the conditions under which the learning guarantees improvement are found. Describe real-time implementation of the TLD framework and the P-N learning.

2 TRACKING

The long-term tracking can be approached either from tracking or from detection perspectives. Tracking is a algorithms which estimate the object motion. Trackers require only initialization, are fast and produce smooth trajectories. On the other hand, they accumulate error during run-time (drift) and typically fail if the object disappears from the camera view. The tracker follows the object from frame to frame.

Object tracking is the task of estimation of the object motion. Trackers typically assume that the object is visible throughout the sequence. Various representations of the object are used in practice, for example: points[1], articulated models [2], contours[3], or optical flow[4].

OBJECT TRACKING METHODS

2.1 Frame to frame tracking
2.2 Template tracking

2.1 FRAME TO FRAME TRACKING

The methods that represent the object by geometric shape and their motion is estimated between consecutive frames i.e. the so called frame.

2.2 TEMPLATE TRACKING

The object is described by a target template (an image patch, a color histogram) and the motion is define as transformation that minimizes mismatch between target template and candidate patch.

Template tracking can be either realized as static (when the target template does not change), or adaptive (when the target template is extracted from the previous frame). Methods that combine static and adaptive template tracking have been proposed as well as methods that recognize "reliable" parts of the template. Templates have

limited modelling capabilities as they represent only a single appearance of the object.

To model more appearance variations, the generative models have been proposed. The generative models are either build offline or during run-time [5]. The generative trackers model only the appearance of the object and as such often fail in cluttered background. In order to alleviate this problem, recent trackers also model the environment where the object moves. Two approaches to environment modelling are often used. First, the environment is searched for supporting object the motion of which is correlated with the object of interest. These supporting object then help in tracking when the object of interest disappears from the camera view or undergoes a difficult transformation. Second, the environment is considered as a negative class against which the tracker should discriminate. A common approach of discriminative trackers is to build a binary classifier that represents the decision boundary between the object and its background. Static discriminative trackers [6] train an object classifier before tracking which limits their applications to known objects.

Adaptive discriminative trackers [7] build a classifier during tracking. The essential phase of adaptive discriminative trackers is the update: the close neighborhood of the current location is used to sample positive training examples, distant surrounding of the current location is used to sample negative examples, and these are used to update the classifier in every frame. It has been demonstrated that this updating strategy handles significant appearance changes, short-term occlusions, and cluttered background. However, these methods also suffer from drift and fail if the object leaves the scene for longer than expected. To address these problems the update of the tracking classifier has been constrained by an auxiliary classifier trained in the first frame or by training a pair of independent classifiers.

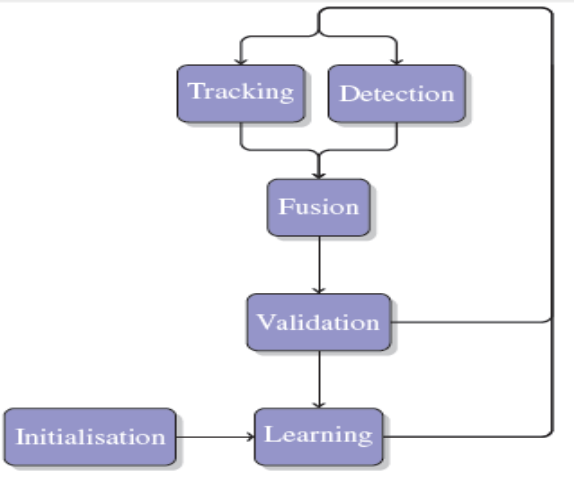

FIGURE 2.1:- The tracking process is initialized by manually selecting the object of interest. No further user interaction is required.

According to Fig. 2.1. First, an equally spaced set of points is constructed in the bounding box in frame t, which is shown in the left image. Next, the optical flow is estimated for each of these points by employing the method of Lucas and Kanade. This method works most reliably if the point is located on corners and is unable to track points on homogenous regions. We use information from the Lucas-Kanade method as well as two different error measures based on normalised cross correlation and forward-backward error in order to filter out tracked points that are likely to be erroneous. In the right image the remaining points are shown.

If the median of all forward-backward error measures is above a certain threshold, we stop recursive tracking entirely, since we interpret this event as an indication for drift. Finally, the remaining points are used in order to estimate the position of the new bounding box in the second frame by employing a transformation model based on changes in translation and scale. In the right image, the bounding box from the previous frame was transformed according to the displacement vectors from the remaining points.

FIGURE2.2:- Tracking is possible as long as the selected object is visible in the image. In the third frame an occlusion occurs.

3 DETECTION

Detection-based algorithms estimate the object location in every frame independently. Detectors do not drift and do not fail if the object disappears from the camera view.

Object detection is the task of localization of objects in an input image. The definition of an “object” varies. It can be a single instance or a whole class of objects.

3 OBJECT DETECTION METHODS

3.1 Local image features
3.2 Sliding window.

3.1 THE FEATURE-BASED APPROACHES

I) feature detection
II) feature recognition and
III)modelfitting

Planarity or a 3D model is typically exploited. These algorithms reached a level of maturity and operate in real-time even on low power devices and in addition enable detection of a large number of objects .The main strength as well as the limitation is the detection of image features and the requirement to know the geometry of the object in advance.

3.2 SLIDING WINDOW-BASEDAPPROACHES

The sliding window-based approaches scan the input image by a window of various sizes and for each window decide whether the underlying patch contains the object of interest or not. For a QVGA frame, there are roughly 50,000 patches that are evaluated in every frame. To achieve a real time performance, sliding window-based detectors adopted the so-called cascaded architecture. Exploiting the fact that background is far more frequent than the object, a classifier is separated into a number of stages, each of which enables early rejection of background patches thus reducing the number of stages that have to be evaluated on average. Training of such detectors typically requires a large number of training examples and intensive computation in the training stage to accurately represent the decision boundary between the object and background. Alternative approach is to model the object as a collection of templates. In that case the learning involves just adding one more template.

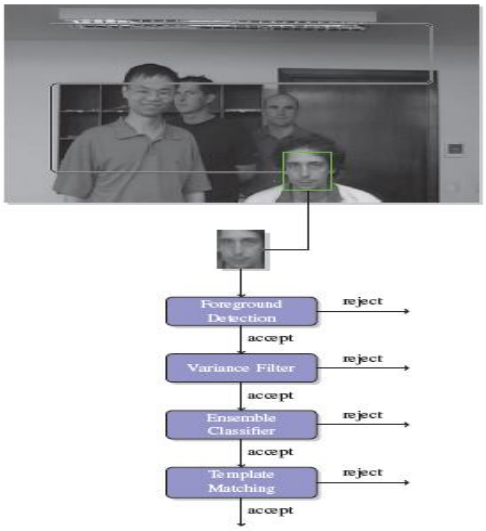

FIGURE 3.1:- In sliding-window-based approaches for object detection, sub windows are tested independently.

We employ a cascaded approach in order to reduce computing time.

Object detector is based on a sliding-window approach which is illustrated in Fig. 3.1. The image at the top is presented to the object detector, which then evaluates a classification function at certain predefined sub windows within each input image. Depending on the size of the initial object, we typically employ 50,000 to 200,000 sub windows for an image of VGA (640_480) resolution. Each sub window is tested independently whether it contains the object of interest.

Only if a sub window is accepted by one stage in the cascade, the next stage is evaluated. Cascaded object detectors aim at rejecting as many non-relevant sub windows with a minimal amount of computation. The four stages that we use for image classification are shown below the input image. First, we use a background subtraction method in order to restrict the search space to foreground regions only. This stage requires a background model and is skipped if it is not available. In the second stage all sub windows are rejected that exhibit a variance lower than a certain threshold. The third stage comprises an ensemble classifier based on random ferns. The fourth stage consists of a template matching method that is based on the normalised correlation coefficient as a similarity measure. We handle overlapping accepted sub windows by employing a non-maximal suppression strategy.

4 LEARNING

A tracker can provide weakly labelled training data for a detector and thus improve it during run-time. A detector can re-initialize a tracker and thus minimize the tracking failures.

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 detector's errors and updates it to avoid these errors in the future.

While a wide range of trackers and detectors exist, we are not aware of any learning method that would be suitable for the TLD framework.

Such a learning method should:

(i) deal with arbitrarily complex video streams where the tracking failures are frequent,
(ii) never degrade the detector if the video does not contain relevant information and
(iii) operate in real-time.

To tackle all these challenges, we rely on the various information sources contained in the video.

OBJECT LEARNING METHODS

4.1 Machine learning
4.2 P-N learning

4.1 MACHINE LEARNING

Object detectors are traditionally trained assuming that all training examples are labeled. Such an assumption is too strong in our case since we wish to train a detector from a single labeled example and a video stream. This problem can be formulated as a semi-supervised learning [8] that exploits both labeled and unlabeled data. These methods typically assume independent and identically distributed data with certain properties, such as that the unlabeled examples form "natural" clusters in the feature space. A number of algorithms relying on similar assumptions have been proposed in the past including EM, Self-learning and Co-training.

Expectation-Maximization (EM) is a generic method for finding estimates of model parameters given unlabeled data. EM is an iterative process, which in case of binary classification alternates over estimation of soft-labels of unlabeled data and training a classifier. EM was successfully applied to document classification and learning of object categories .In the semi-supervised learning terminology, EM algorithm relies on the "low density separation" assumption , which means that the classes are well separated. EM is sometimes interpreted as a "soft" version of self-learning.

Self-learning starts by training an initial classifier from a labeled training set, the classifier is then evaluated on the unlabeled data. The examples with the most confident classifier responses are added to the training set and the classifier is retrained. This is an iterative process. The self-learning has been applied to human eye detection in. However, it was observed that the detector improved more if the unlabeled data was selected by an independent measure rather than the classifier confidence.

It was suggested that the low density separation assumption is not satisfied for object detection and other approaches may work better.

Co-training is a learning method build on the idea that independent classifiers can mutually train one another. To create such independent classifiers, co-training assumes that two independent feature-spaces are available. The learning is initialized by training of two separate classifiers using the labeled examples. Both classifiers are then evaluated on unlabeled data. The confidently labeled samples from the first classifier are used to augment the training set of the second classifier and vice versa in an iterative process.

Co-training works best for problems with independent modalities, e.g. text classification (text and hyper-links) or biometric recognition systems (appearance and voice). In visual object detection, co-training has been applied to car detection in surveillance and moving object recognition.

We argue that co-training is suboptimal for object detection, since the examples (image patches) are sampled from a single modality. Features extracted from a single modality may be dependent and therefore violate the assumptions of co-training.

4.2 P-N LEARNING

The learning estimates detector's errors and updates it to avoid these errors in the future. How to identify detector's errors and learn from them. We develop a novel learning method (P-N learning) which estimates the errors by a pair of "experts":

(i)P-expert estimates missed detections, and (ii) N-expert estimates false alarms.

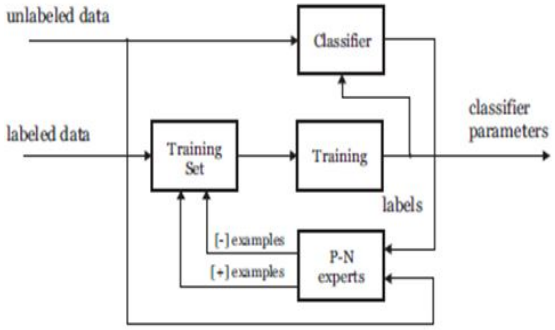

FIGURE 4.1: The block diagram of the P-N Learning

The goal of the TLD component is to improve the performance of an object detector by online processing of a video stream. In every frame of the stream, to evaluate the current detector, identify its errors and update it to avoid these errors in the future. The key idea of P-N learning is that the detector errors can be identified by two types of "experts". P-expert identifies only false negatives; N expert identifies only false positives. Both of the experts make errors themselves; however, their independence enables mutual compensation of their errors. The P-N learning is a semisupervised learning method.

Formalization

Let x be an example from a feature-space X and y be a label from a space of labels Y = {-1, 1}. A set of examples X is called an unlabeled set, Y is called a set of labels and L = f(x; y)g is called a labeled set. The input to the P-N learning is a labeled set Ll and an unlabeled set Xu, where l << u. The task of P-N learning is to learn a classifier f: X -> Y from labeled set Ll and bootstrap its performance by the unlabeled set Xu. Classifier f is a function from a family F parameterized by Θ. The family F is subject to implementation and is considered fixed in training, the training therefore corresponds to estimation of the parameters Θ.

The P-N learning consists of four blocks:
i) A classifier to be learned,
ii) Training set – a collection of labeled training examples,
iii) Supervised training – a method that trains a classifier from training set, and
iv) P-N experts–functions that generate positive and negative training examples during learning. See figure 3.1 for illustration.
The training process is initialized by inserting the labelled set L to the training set. The training set is then passed to supervise learning which trains a classifier, i.e. estimates the initial parameters Θ^0. The learning process then proceeds by iterative bootstrapping. In iteration k, the classifier trained in previous iteration classifies the entire unlabeled set, $y^k u = f(xu\ \Theta^{k-1})$ for all xu 2 Xu. The classification is analyzed by the P-N experts which estimate examples that have been classified incorrectly. These examples are added with changed labels to the training set. The iteration finishes by retraining the classifier, i.e. estimation of Θ^k. The process iterates until convergence or other stopping criterion.
The crucial element of P-N learning is the estimation of the classifier errors. The key idea is to separate the estimation of false positives from the estimation of false negatives. For this reason, the unlabeled set is split into two parts based on the current classification and each part is analyzed by an independent expert. P-expert analyzes examples classified as negative, estimates false negatives and adds them to training set with positive label. In iteration k, P-expert outputs n+(k)positive examples. N-expert analyzes examples classified as positive, estimates false positives and adds them with negative label to the training set. In iteration k, the N-expert outputs n(k) negative examples. The P-expert increases the classifier's generality. The N-expert increases the classifier's discriminability.

Relation to supervised bootstrap

To put the P-N learning into broader context, let us consider that the labels of set Xu are known. Under this assumption it is straightforward to recognize misclassified examples and add them to the training set with correct labels. Such a strategy is commonly called (supervised) bootstrapping. A classifier trained using such supervised bootstrap focuses on the decision boundary and often outperforms a classifier trained on randomly sampled training set. The same idea of focusing on the decision boundary underpins the P-N learning with the difference that the labels of the set Xu are unknown. P-N learning can therefore be viewed as a generalization of standard bootstrap to unlabeled case where labels are not given but rather estimated using the P-N experts. As any other process, also the PN experts make errors by estimating the labels incorrectly.
P-expert exploits the temporal structure in the video and assumes that the object moves along a trajectory. The P-expert remembers the location of the object in the previous frame and estimates the object location in current frame using a frame -to-frame tracker. If the detector labeled the current location as negative (i.e. made false negative error), the P-expert generates a positive example.
N-expert exploits the spatial structure in the video and assumes that the object can appear at a single location only. The N-expert analyzes all responses of the detector in the current frame and the response produced by the tracker and selects the one that is the most confident. Patches that are not overlapping with the maximally confident patch are labelled as negative. The maximally confident patch re-initializes the location of the tracker.

5 TLD FRAMEWORK

FIGURE 5.1: The block diagram of the TLD framework

TLD is a framework designed for long-term tracking of an unknown object in a video stream. Its block diagram is shown in figure 5.1. 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 any other detector, the detector makes two types of errors: false positives and false negative. Learning observes performance of both, tracker and detector, estimates detector's errors and generates training examples to avoid these errors in the future. The learning component assumes that both the tracker and the detector can fail. By the virtue of the learning, the detector generalizes to more object appearances and discriminates against background.

EXAMPLE NO**. 1**

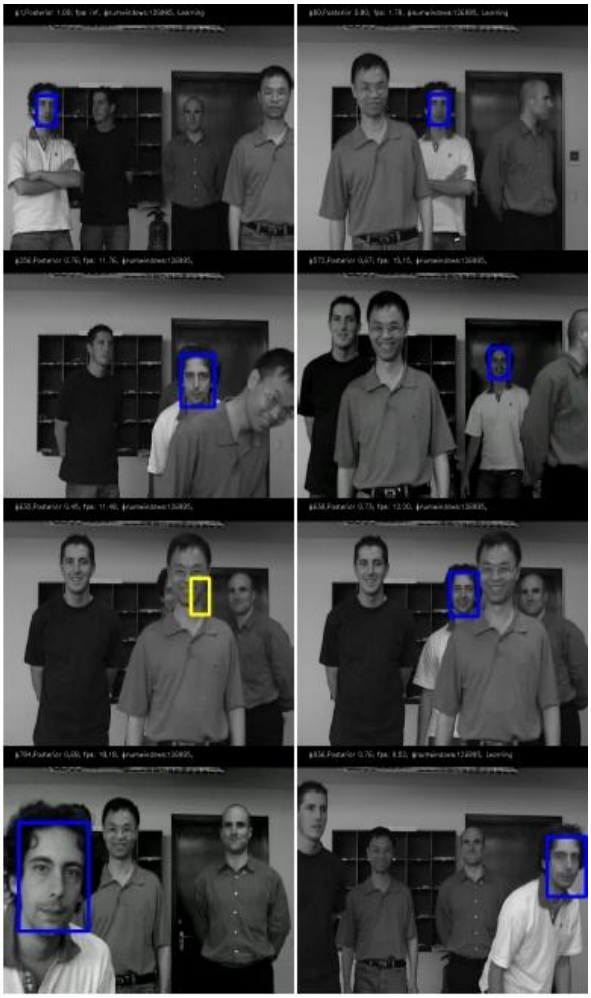

FIGURE a): Qualitative results for the sequence Multi Face Turning.

EXAMPLE NO. 2

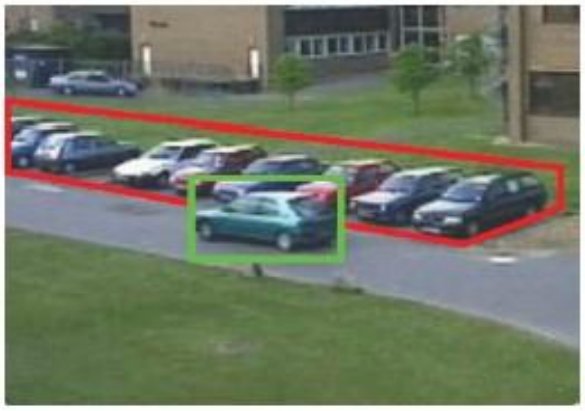

FIGURE b) Object tracking is to the distinguish the object of interest (green) from clutter in the background (red).

APPLICATIONS

1) Security
2) Domestic services
3) Human-Machine interaction
4) Video indexing
5) Sports competition
6) Traffic monitoring
7) Vehicle navigation

CONCLUSION

The problem of tracking of an unknown object in a video stream, where the object changes appearance frequently moves in and out of the camera view. A new framework exists that decomposes the tasks into three components: tracking, learning and detection. The learning component analysis shows that an object detector can be trained from a single example and an unlabeled video stream using the following strategy:
I) evaluate the detector,
II) estimate its errors by a pair of experts, and
III) update the classifier.
Each expert is focused on identification of particular type of the classifier error and is allowed to make errors itself. The stability of the learning is achieved by designing experts that mutually compensate their errors. The theoretical contribution is the formalization of this process as a discrete dynamical system, which allows specifying conditions, under which the learning process guarantees improvement of the classifier. The experts can exploit spatio-temporal relationships in the video. TLD framework is a real-time approach.

REFERENCES

[1] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," International Joint Conference on Artificial Intelligence, vol. 81, pp. 674–679, 1981.

[2] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," Pattern Recognition, vol. 36, no. 3, pp. 585–601, 2003.

[3] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," Conference on Computer Vision and Pattern Recognition,1998.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artificial intelligence, vol. 17, no. 1-3, pp. 185–203, 1981.

[5] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental Learning for Robust Visual Tracking," International Journal of Computer Vision, vol. 77,pp. 125–141, Aug. 2007.

[6] S. Avidan, "Support Vector Tracking," IEEE Transactions on Pattern
Analysis and Machine Intelligence, pp. 1064–1072, 2004.

[7] R. Collins, Y. Liu, and M. Leordeanu, "Online Selection of Discriminative
Tracking Features," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1631–1643, 2005.

[8] O. Chapelle, B. Sch¨olkopf, and A. Zien, Semi-Supervised Learning.
Cambridge, MA: MIT Press, 2006.

- *Rupali S. Chavan is currently pursuing masters degree program in Electronics and Telecommunication engineering in Mumbai University, India, PH-919821536853 E-mail: rup_sanj@tyahoo.com*
- *Mr. S. M. Patil is currently working in Electronics engineering dept. in DMCE, Mumbai University, India, PH-919819532579. E-mail:smpatil_99@rediffmail.com*