Implementation of Generic Object Tracker based on TLD Framework, using Generic tools

Sheetal Balsaraf
University of Mumbai
K. J. Somaiya College of Engineering

Uday Joshi
Associate Professor, University of Mumbai
K. J. Somaiya College of Engineering

ABSTRACT
Security is becoming the primary concern of society due to a booming population, increasing scarcity of jobs, growing problems especially in urban cities, and the number of anti-social activities, etc. Having a security system is therefore becoming a requirement. Surveillance camera's output, when monitored, can track unauthorized objects from causing a menace. An important application of object tracking is video surveillance. In the proposed system, the object of interest is defined in each frame using a bounding box. The purpose is to determine the object's bounding box, automatically, in every frame that follows. It is a generic system for surveillance which indefinitely tracks an unknown bounded object, from online real time video or video file input, in an unconstrained environment. It works even in occlusions, illumination changes, and rotation and scale changes of object. Multiple packages offer tools, that can be used to implement computer vision systems, are available. One such package is EmguCV, which is used for developing this system. This is a C# wrapper for the OpenCV package which is in C++.

General Terms
Computer Vision, Video Surveillance.

Keywords
Long-term Tracking, Learning from video, Real-time.

1. INTRODUCTION
Security is becoming the primary concern of society due to a booming population, increasing scarcity of jobs, growing problems especially in urban cities, and the number of anti-social activities, etc. Having a security system is therefore becoming a requirement. They could save the lives of loved ones and will continue to pay for themselves by preventing damage and loss of valued property. Surveillance camera output when monitored, can track unauthorized objects from causing a menace. An important application of object tracking is video surveillance. Airports, train stations, departmental stores, religious places, courts and public buildings are primary examples of places where video surveillance has high priority. Additionally, military, astronomy, navigation, road/air traffic regulation, medical imaging, augmented reality and robotics are other major applications of object tracking. For surveillance purpose, real time video needs to be analyzed continuously. A video stream taken by a handheld camera, is considered, wherein objects are moving in and out of camera premises. The object of interest is defined in each frame using a bounding box. The purpose is to determine the object's bounding box, automatically, in every frame that follows. The processing of video stream is done frame to frame and it runs indefinitely long. This task is termed as long-term tracking. [1]. The crucial issues in long-term tracking are, the detection of the object when it reappears in the camera's premises, handling of scale, illumination changes, background clutter, partial occlusions and operating in real-time.

For long-term tracking, two approaches are followed widely, either from tracking or from detection perspectives. In tracking algorithms, estimation of the object motion is done. Trackers require initialization, are quick and produce smooth trajectories. But, they gather error during run-time (drift) and fail if the object is not present in the camera premises. Detection-based algorithms estimate the object location in every frame independently. Detectors need training, which means early knowledge of object to be tracked so cannot be applied to unknown objects. The fact is that tracking or detection both cannot solve the long-term tracking task independently. If both are used in combination, they may benefit each other. Tracker can provide weakly labeled training data for a detector and thus improve it during runtime. A detector can reinitialize a tracker and thus minimize the tracking failures. [1]

Tracking-Learning-Detection (TLD) [1] is a framework addressing long-term tracking of unknown objects in video streams. It decomposes the long-term tracking task into three subtasks: tracking, learning, and detection [1]. In the proposed system, the Bounding Box Tracker follows the object from frame to frame & estimates the object motion under the assumption that the object is visible and its motion is limited. It acts as a tracker. The Learner is provided weakly labeled data by the bounding box tracker block, Learner block uses classifier logic to classify the patches, which are verified by validation logic to eliminate false positive & false negative. Match Processing Detector acts as a detector which is provided with trainedImagesDb by learner. It outputs a matching value.

2. RELATED WORK
Video Surveillance is gaining importance. Content in a video, can be modeled as a hierarchy of abstractions. At granual level, video content are raw pixels with color information leading to lines, curves, edges, corners and regions. An important application of object tracking is video surveillance [2]. Military, astronomy, navigation, road/air traffic regulation, medical imaging, augmented reality and robotics are crucial & major applications of object tracking [3], [4], [5]. Object can be represented in various ways ,e.g as points [6], as articulated models [7], contours, as optical flow [8]. In proposed system objects are considered as geometric shapes & their motion is being estimated between consecutive frames. Simple approach to tracking is template tracking. The Template tracking can be as implemented as static, when the target template does not change [9] or adaptive, when the target template is extracted from the previous frame [6]. Combination of static and adaptive template tracking also ave een proposed [10]. Templates represent only single

appearance of object which reduces it's modelling capability. For modelling, more

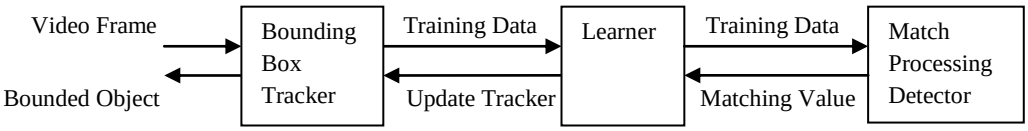

Figure 1: System Architecture for Generic TLD based object detector

appearance variations, the generative models have been proposed.[11],[12].

2.1 Template Matching [13][15]

Template matching is a technique in digital image processing for finding small parts of an image which match a template image [13]. The object, to be matched, is defined as a master template. With respect to video frame input, current video frame may be broken into parts. Each part is considered as a template to be matched with master template. The master template and current template from video frame is matched using a similarity threshold. If value returned by template matching is above the threshold it is considered as a match else not a match.

A basic method of template matching uses a convolution mask (template), tailored to a specific feature of the search image, which needs to be detected. This technique can be easily performed on grey images or edge images. The convolution output will be highest at places where the image structure matches the mask structure, where large image values get multiplied by large mask values.

2.1.1. Template Matching Algorithm

Let the search image **S(x, y)**, where **(x, y)** represent the coordinates of each pixel in the search image and the template **T(x_t, y_t)**, where **(x_t, y_t)** represent the coordinates of each pixel in the template.

Step1: Move the center (or the origin) of the template **T(x_t, y_t)** over each **(x, y)** point in the search image

Step2: Calculate the sum of products between the coefficients in **S(x, y)** and **T(x_t, y_t)** over the whole area spanned by the template.

Step3: The position with the highest score is the best position (a match).

2.2 The Generic TLD based Object Tracker

The system is influenced by the real-time, long-term tracking of unknown object using TLD framework from [1]. Generic tool OpenCV, is used in a specified manner, to implement TLD based object tracker. The architecture and pseudo code will be seen in following sections. It is assumed that frame to frame motion is limited. In above Figure 1. The system architecture for Generic TLD based object detector is given.

2.2.1 System Architecture

Video frames are accessed at 18 frames per second. The object is presented by its state, at any specific time. The state is a bounding box.

2.2.1.1 Bounding Box Tracker

The initial selection of the unknown object to be tracked, using a bounding box forms the initial state of the object. Bounding box has a fixed aspect ratio parameterized by it's initial bounding box location and scale. In-plane rotation is not considered. This bounding box forms the initial ROI (Region of Interest). Bounding Box tracker tracks the position of bounding box from frame to frame. It updates the co-ordinates of the new location of object, i.e the bounding box. The initial bounding box template is stored as a P-patch by the learner. The tracker estimates probable location for object using RectROI. The initial bounding box is the initial ROI. It is expanded by 10% on all side to form RectROI. RectROI is estimated for each video frame by tracker.

2.2.1.2 Learner

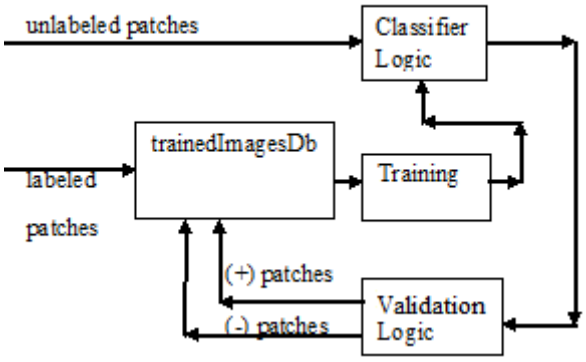

Figure 2. Block diagram of Learner

In a given video frame, when the object is bounded, the size of bounding box is measured and grids of same size are made all over the frame. This is the PN Grid. Each grid forms a patch which is classified either as object patch or a backgroung patch. Learner block belongs to semi-supervised learning that exploits both labeled & unlabeled data. The key idea is to separate the estimation of false positives from the estimation of false negatives. For this reason, the unlabeled set is checked two times, first based on the classification and second each template is analyzed by an independent expert in Validation Logic. Classifier logic classifies the unlabeled data into three classes, similar, some-what-similar, no match class depending on the matching value. For the proposed system matching value above 0.95 is a match, below 0.85 is no-match and in between 0.85-0.95 is some-what-match class. Validation Logic is the crucial element of Learner block, which estimates classifier errors. The validation logic gives final labels. Similar class is given P-patch label, no-match class is given N-patch label and patches which fall into some-what-similar class are checked using validation logic for a match or no match. The Validation logic, verifies the patch

that falls in some-what-similar class, whether it is a real match or not. Validation logic contains P-experts and N-experts. N-expert takes the respective patch/template and compares it with all other N-patches in training database. If it finds a match, it means it is a false positive and hence rejects it, and labels it as N-patch. P-expert takes patches that fall in no match class and compares the respective patch with all the P-images in database. If it finds a match, it means it is a false negative, hence it labels it as a P-patch. The trainedImagesDb consists of a label for every patch. It consists of labeled patches as P or N. P patches correspond to tracked object and N to background. P patches denotes various new features of the object learnt from previous video frames.New patches are appended at the end of the database. Hence, newly learnt features lie at higher indices of trainedImagesDb. Hence, matching of current template with the P-patches stored in the trainedImagesDb is done in reverse order.

2.2.1.3 Match Processing

Template matching is used as a detector. Training is provided to it from learner block. Learner provides the Match processing block with trained images database. Template Matching is applied for current video frame, to find a match from the trained images, in RectROI. Normalized Co-efficient Correlation template matching function is used for template matching, which returns a matching value between 0-1. Normalized correlation coefficient technique, matches the mean of the image relative to the mean of template, with a good match being 1, no match being 0. The matching value is returned by Match Processing block to the learner.

2.2.2 Implementation

There exist multiple packages that offer tools that can be used to implement computer vision systems. One such package is EmguCV. This is a C# wrapper for the OpenCV package which is in C++. OpenCV version 2.4.2 is used for developing the proposed system.

Pseudo *Code for Generic TLD based Object Tracker*

1: Copy the frame incoming from web-camera to imgMain.

2: Bound the object to be tracked using a bounding box.

3: Expand the bounding box by 10% on all sides to get rectROI.

4: Do

{

4.1: Divide the current video- frame into grids of same size as bounding box.

4.2: Apply template matching for current template from video frame with rectROI.

{

1. If (matching value > 0.95)

Bound the location of matched frame

2. ElseIf (matching value < 0.85)

No match

Increase ROI by 10% on all sides

Execute Step 4 again.

3. ElseIf (0.95 > matching value> 0.85)

{

3.1. Apply template matching of current template with N-images in trainedImagesDb

{

If (matching value > 0.95)

Put the current template in rejected frames

ElseIf (matching value < 0.85)

No match

ElseIf (0.95 > matching value> 0.85)

Store it as a P-patch in trainedImagesDb

}

3.2 Apply template matching of current template with P-patches in trainedImagesDb.

{

If (1 < matching value > 0.85)

Put the matched patch in P-patches Of trainedImagesDb

}

}

4.3: Execute step 4.2 for the next template in rectROI.

Until all the templates in rectROI are processed.

} while(video-capture is true).

5: Display the Bounded Object

6: Loop through Step 3 Until video frame input is finished.

2.3 Processing steps

The snapshots of execution of the template matching system and Generic TLD based Object Detector are given below.

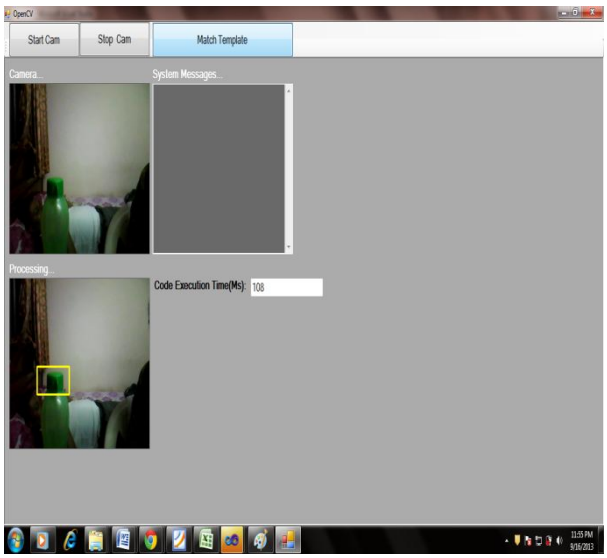

Figure 3. Snapshot of GUI of template matching system

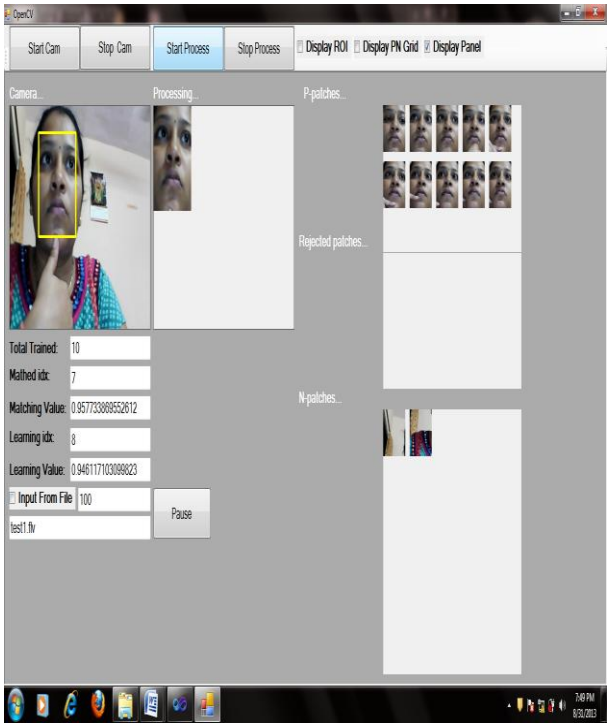

Figure 4. Snapshot of GUI of Generic TLD based Object Detector

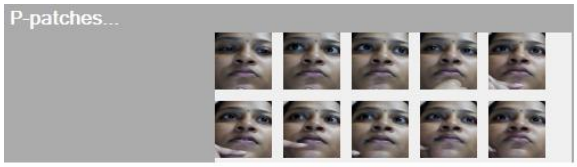

Figure 5. P-patches after processing

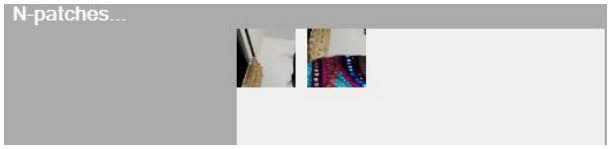

Figure 6. N-patches after processing.

Figure 7. Taking Input from a video file

Figure 8. GUI of Generic TLD based Object Detector Processing with PN Grid & RectROI

3. EXPERIMENTAL RESULTS

This section represents the implementation results of Template matching and Generic TLD based Object Detector. All experiments are performed on Intel Core i3-370M processor with 3 GB memory using OpenCV 2.2.4, with

consideration of work folder having handful of at least 100 videos obtained from internet [16] and some shooted according to varied requirements. Time required for execution of the two algorithms is as given in Table 2.

Table 2. Code execution Time

Algorithm	Elapsed Time (secs)
Template Matching	0.7 approx
Generic TLD based Object Detector	1.0 approx

The code execution time for Generic TLD based Object Detector varies for every video frame being processed. Initially it takes more time (less than 3 seconds always), if a match is found quickly then less time, if new feature is to be learned from the current video frame then again more time. Hence the average time required for processing the entire code for every frame in a video stream is calculated based on results obtained from processing of 100 videos.

4. DISCUSSIONS

As generic tools are used the cost of application reduces. As it is giving reliable results, it may be considered over other expensive systems. In this section performance analysis of template matching and Generic TLD based object tracking is done.

Table 1: Comparative study between Template Matching And Proposed system

Parameter	**Template Matching**	**Generic based TLD Object Tracker**
Rotation	Rotation is not handled efficiently.	As the degree of rotation increases it consistently shows better results upto a limit and then starts deteriorating.
Illumination Changes	Fails as variability in illumination increases.	Works well for variations in illumination, except drastic changes of light and dark.
Re-Detection time span	Takes more time to re-detect a template	Efficiently re-detects a template.
Noise Handling	Does not work well with noisy images.	Is not affected much by noise.

5. CONCLUSIONS

In the proposed system, video frame co-ordinate tracking and template matching as detector is enhanced with learning mechanism. The Bounding Box tracker using Learner provides weakly labeled data to the Match Processing Detector. Learner avoids false positives and false negatives being identified. This removes inherent drawbacks of template matching. Though Template Matching is fast, it is incapable of handling situations wherein the rotation, scaling changes of object are involved. Hence, it cannot be used tracking for surveillance. It can be concluded after testing 100 videos dataser that the proposed system, Generic TLD based Object tracker is robust, reliable and fast for unknown object tracking and surveillance. It is observed to perform well in cases where object behavior seems to change, with respect to rotation, presence of noise, illumination changes and re-detection of object.

6. REFERENCES

[1] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, July 2012.

[2] G. L. Foresti and F. Roli: Real-time Recognition of Suspicious Events for Advanced Visual-based Surveillance. In Multimedia Video-Based Surveillance Systems: From User Requirements to Research Solutions, G. L. Foresti, C. S. Regazzoni, and P.Mahonen, Eds. Dordrecht, The Netherlands: Kluwer, pp. 84 - 93, 2000.

[3] Y. Wang, R.E. Van Dyke and J F Doherty: Tracking Moving Objects in video Scene. Technical Report, Department of Electrical Engg, Pennsylvania State University, 2000.

[4] V.V. Vinod and H. Murase: Video Shot Analysis using Efficient Multiple Object Tracking. Proceedings of IEEE International Conference on Multimedia Computing and Systems, pp. 501 - 508, 1997.

[5] L. Wixson: Detecting Salient Motion by Accumulating Directionally-Consistent Flow. IEEE Transactions on PAMI, Vol. 22, No. 8, pp. 774 - 780, 2000.

[6] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proc. Seventh Int'l Joint Conf. Artificial Intelligence, vol. 81, pp. 674-679, 1981.

[7] L. Wang, W. Hu, and T. Tan, "Recent Developments in Human Motion Analysis," Pattern Recognition, vol. 36, no. 3, pp. 585-601, 2003.

[8] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," Artificial Intelligence, vol. 17, nos. 1-3, pp. 185-203, 1981.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-577, May 2003.

[10] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Int'l J. Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[11] M.J. Black and A.D. Jepson, "Eigentracking : Robust Matching and Tracking of Articulated Objects Using a

View-Based Representation," Int'l J. Computer Vision, vol. 26, no. 1, pp. 63-84, 1998.

[12] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental Learning for Robust Visual Tracking," Int'l J. Computer Vision, vol. 77, nos. 1-3,pp.125-141, http://www.springerlink.com/index/10.1007/

[13] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009

[14] OpenCV 2.4.2 Documentation [online]. Available: http://opencv.org/documentation.html

[15] Template Matching Overview [online]. Available: http://en.wikipedia.org/wiki/Template_matching

[16] Downloadable Videos for testing [online]. Available: www.vimeo.com

[17] Gary Bradski and Adrian Kaehler, "Learning OpenCV " O'Reilly Media, Inc. , 1st Edition, Sep '08.