Performance Evaluation of Object Detection and Tracking Systems

Faisal Bashir, Fatih Porikli
Mitsubishi Electric Research Laboratories, Cambridge, MA 02139
fatih@merl.com

Abstract

This paper presents a set of metrics and algorithms for performance evaluation of object tracking systems. Our emphasis is on wide-ranging, robust metrics which can be used for evaluation purposes without inducing any bias towards the evaluation results. The goal is to report a set of unbiased metrics and to leave the final evaluation of the evaluation process to the research community analyzing the results, keeping the human in the loop. We propose metrics from statistical detection and estimation theory tailored to object detection and tracking tasks using frame-based as well as object-based evaluation paradigms. Object correspondences between multiple ground truth objects to multiple tracker result objects are established from a correspondence matrix. The correspondence matrix is built using three different methods of distance computation between trajectories. Results on PETS 2001 data set are presented in terms of 1st and 2nd order statistical descriptors of these metrics.

1. Introduction

The issue of object detection and subsequent tracking from video sequences is a fundamental problem in computer vision. The problem started to gain attention in the wider community of computer vision more than a decade ago. Today, the issue receives more intense pursuit from the narrower but more focused visual surveillance community. In video surveillance domain, object detection and tracking is of central importance for any modern video surveillance and behavioral analysis system. Automated video surveillance systems constitute a network of video sensors observing people as well as other moving and interacting objects in a given environment for patterns of normal/abnormal activities, interesting events, and other domain-specific goals. A vital role envisioned for the modern video surveillance systems is their use as an active tool towards crime prevention, law-enforcement, and pre-emptive interest protection [1]. This is in sharp contrast with most existing systems used mainly as forensic tools for "after the fact" investigation. Automated video surveillance is attractive because it promises to replace the more costly option of staffing video surveillance monitors with human observers. This promise of automated video surveillance can easily turn into its pitfall if the systems don't perform to the desired level. This expected performance level tends to be quite high in case of trained and attentive human operators on a well-staffed facility. On the other hand, the problem of robust object detection and tracking is even harder to address given the requirement that the video surveillance systems have to operate in widely varying weather conditions and all time periods. This situation of high performance expectations and stringent requirements places a minimal margin of error on the performance of these video surveillance systems.

The issue of evaluating the performance of video surveillance systems is becoming more important as more and more research effort is drawn into object detection and tracking. It's a natural question to ask whether there has been quantifiable progress in the form of robust, commercial-grade video surveillance systems as a result of past and ongoing research in this direction. This paper addresses the issue of comprehensive performance evaluation of automatic object detection and tracking systems. We propose several performance evaluation metrics for quantitative assessment of the performance of video surveillance systems. Based on signal detection and estimation theory, these metrics are extended for correct application towards performance evaluation of video surveillance systems. These metrics are evaluated after establishing correspondences between ground truth and tracker result objects. These many-to-many correspondences are established based on association matrices computed from three different methods of trajectory distance computations. The rest of this paper is organized as follows: Section 2 presents a review of the recent and ongoing activity in the domain of performance evaluation for tracking systems; Sections 3 details our evaluation metrics as well as algorithms of data association using correspondence matrices; Section 4 briefly outlines the two different tracker modules developed by us that will be evaluated using the metrics; results of this performance evaluation setup are also reported in this Section; finally, Section 5 concludes the report with summary and future work directions.

2. Related Work

Initial efforts towards performance evaluation of video detection and tracking systems began with the workshops dedicated to the topic, namely PETS (performance evaluation of tracking and surveillance) series of workshops. The main focus of the workshop in early stages was to provide a standard benchmark datasets for participants to evaluate their systems and report results on industry-standard datasets. Later on, the emphasis has shifted more towards standard metrics used to compute the results of evaluation process. Finally, a recent trend is providing online access portals for research community to submit their intermediate results based on object detection and tracking. The online portal administrators then evaluate the system performance based on standard metrics to compare the candidate algorithms.

As alternative to manual ground truth generation, Black *et al* [3] propose to use pseudo synthetic video to evaluate tracking performance. They synthetically vary the perceptual complexity of tracking task by adding occlusions and inserting increasing number of agents in the scene. Results of object detection and tracking are presented based on metrics derived from a number of sources. These metrics include 'tracker detection rate' (TRDR), 'false alarm rate' (FAR), 'track detection rate' (TDR) and 'track fragmentation' (TF). Lisa *et al* [6] propose algorithms for matching ground truth tracks and system generated tracks and compute performance metrics based on these correspondences. They measure the performance of their system under several different conditions including: indoor/outdoor, different weather conditions and different cameras/view-points. Results on background subtraction and tracking evaluation are reported on the above as well as on the standard PETS 2001 datasets. Stefan *et al* [5] form the correspondence between ground truth and detected objects by minimizing distance between the centroids of ground truth and detected objects. They compute a set of performance metrics including false positive track rate, false negative track rate, average position error, average area error, object detection lag, etc.

An emerging trend in the performance evaluation systems is online portals and websites where contributors can upload the results of their detection and tracking systems in a standard format (mostly in eXtensible Markup Language, XML). The results of various algorithms are then tested for standard performance evaluation metrics to generate results for the comparison of different systems. Collins *et al* [4] report a tracking test-bed to run and log tracking algorithms and their results in real-time. The testbed allows a tracking experiment to be repeated from the same starting state using different tracking algorithms and parameter settings, thereby facilitating comparison of algorithms. On the associated website, tracking results can be uploaded for evaluation using the standard test metrics. A similar approach is taken in [2], where the online service allows researchers to submit their algorithm results on video segmentation to view their algorithm's performance against a set of standard metrics. The approach has been used towards the problem of motion segmentation using seven motion segmentation algorithms to date.

3. Performance Evaluation Metrics

This section outlines the set of performance evaluation metrics we have implemented in order to quantitatively analyze the performance of our object detection and tracking system. We propose a set of both frame-based and object-based metrics for the evaluation. The ground truth information is represented in terms of the bounding box of object for each frame. Similarly, the results of object detection and tracking systems are in terms of the detected or tracked object's bounding box. At the time of evaluation, we employ different strategies to robustly test if the overlap between ground truth and system's results occurs. The simplest form of overlap is testing to see if the system result's centroid lies inside the ground truth object's bounding box. This issue is discussed later in the Section.

Frame-based metrics are used to measure the performance of surveillance system on individual frames of a video sequence. This does not take into account the response of the system in preserving the identity of the object over its lifespan. Each frame is individually tested to see if the number of objects as well as their sizes and locations match the corresponding ground truth data for that particular frame. The results from individual frame statistics are then averaged over the whole sequence. This represents a bottom-up approach. On the other hand, the object-based evaluation measures take the whole trajectory of each object into consideration. Here, the individual tracks of objects which are automatically detected and then tracked over their lifespan are analyzed as separate entities. The various ways of finding the best correspondence (association) between individual ground truth tracks and tracker result tracks are analyzed. Finally, based on a particular association, success and error rates are computed and accumulated for all the objects. This represents a top-down approach. We propose metrics for both the approaches in this section and then present results of the evaluated detection and tracking system in the next section.

3.1. Frame-based Metrics

Starting with the first frame of the test sequence, frame-based metrics are computed for every frame in the sequence. From each frame in the video sequence, first a few true and false detection and tracking quantities are computed.

True Negative, TN: Number of frames where both ground truth and system results agree on the absence of any object.

True Positive, TP: Number of frames where both ground truth and system results agree on the presence of one or more objects, and the bounding box of at least one or more objects coincides among ground truth and tracker results.

False Negative, FN: Number of frames where ground truth contains at least one object, while system either does not contain any object or none of the system's objects fall within the bounding box of any ground truth object.

False Positive, FP: Number of frames where system results contain at least one object, while ground truth either does not contain any object or none of the ground truth's objects fall within the bounding box of any system object.

In the above definitions, the two bounding boxes are said to be *coincident* if the centroid of one of the boxes lies inside the other box. Also, total ground truth TG is the total number of frames for the ground truth objects and TF is the total number of frames in the video sequence. Once the above defined quantities are calculated for all the frames in the test sequence, in the second step, the following metrics are computed:

$$\text{Tracker Detection Rate (TRDR)} = \frac{TP}{TG} \quad (1)$$

$$\text{False Alarm Rate (FAR)} = \frac{FP}{TP+FP} \quad (2)$$

$$\text{Detection Rate} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{Specificity} = \frac{TN}{FP+TN} \quad (4)$$

$$\text{Accuracy} = \frac{TP+TN}{TF} \quad (5)$$

$$\text{Positive Prediction} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Negative Prediction} = \frac{TN}{FN+TN} \quad (7)$$

$$\text{False Negative Rate} = \frac{FN}{FN+TP} \quad (8)$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN} \quad (9)$$

Figure 1 shows two of the metrics, TRDR and FAR computed from the six combinations of trackers and detectors on the PETS data set evaluated in this paper. The notched box plots in this figure clearly show the high detection rate and low false alarm rate of the video surveillance system evaluated in this paper. Each vertical bar represents one tracker – detector combination evaluated for the whole data set.

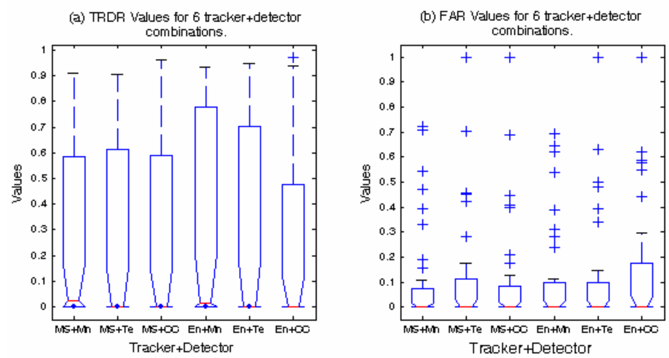

Figure 1: TRDR (a) and FAR (b) for 6 combinations of trackers and detectors.

3.2. Object-based Metrics

Object-based evaluation computes the metrics based on the complete trajectory and lifespan of the individual system and ground truth tracks. Since a given ground truth track could correspond to more than one system tracks and likewise, a correspondence mapping has to be established first. Based on this mapping between object tracks, the frame-based as well as object-based metrics are computed. Figure 2 shows the procedure to compute the core metrics (TN, TP, FN and FP) from object correspondences.

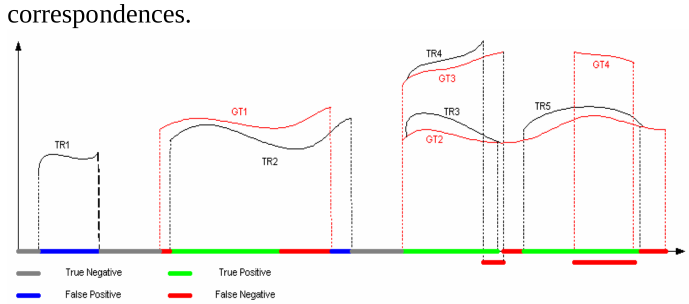

Figure 2: Definitions of 'true negative', 'true positive', 'false negative' and 'false positive'. Note how metrics for multiple objects in a single frame are computed.

The first set of metrics we present are based on simple threshold-based correspondence. For each common frame between a system track and ground truth track, the Euclidean distance between their centroids is computed. The cumulative Euclidean distance is then normalized by the total number of overlapping frames between the ground truth – system track pair under investigation. Finally, two ground truth – system track pairs are declared corresponding if their total normalized distance is within a threshold. Once the correspondence is established, we compute the true positive (TP), false positive (FP) and total ground truth (TG) as explained previously in the context of frame-based metrics. The tracker detection rate

(TRDR) and false alarm rate (FAR) are then computed as before. We also compute the object tracking error which is the average discrepancy between the ground truth bounding box centroid and the centroid of the system result:

Object Tracking Error (OTE)=

$$\frac{1}{N_{rg}} \sum_{i \in g(t_i) \wedge r(t_i)} \sqrt{\left(x_i^g - x_i^r\right)^2 + \left(y_i^g - y_i^r\right)^2} \qquad (10)$$

where N_{rg} represents the total number of overlapping frames between ground truth and system results, x_i^g represents the x-coordinate of the centroid of object in i^{th} frame of ground truth, x_i^r represents the x-coordinate of the centroid of object in i^{th} frame of tracking system result.

The above approach is good enough for quick and simple evaluations. A detailed analysis of the kind of errors most frequently made by practical object tracking systems calls for more sophisticated measures of object correspondence. To provide motivation for our approaches towards object correspondence, imagine the hypothetical scenario in Figure 3. This figure shows two scenarios of error highlighted in a hypothetical yet realistic setting. In (a) and (b), the figure on left hand side shows ground truth tracks and labeling while the figure in the right hand side shows system output on the same sequence. Each of the two figures also shows its binary correspondence map in figures (c) and (d), which reveal the four types of errors consistently. In each binary correspondence map, columns represent the system detected and tracked objects, while rows represent the ground truth detected and tracked objects. A dark cell in the correspondence map shows correspondence between the two objects. We have two algorithms to establish correspondence: many-to-many correspondence associates the tracks in ground truth and tracker results with multiple associations as long as the temporal and spatial overlap between the two tracks is within a threshold; unique correspondence associates the two tracks with most overlap allowing one ground truth track to associate with only one system track and vice versa. For many-to-many correspondence, we look at the binary correspondence map from two views. First we look at the rows of the binary map for each ground truth track to obtain all the matching system tracks. This procedure captures track false negatives and fragmentation errors. In the second pass, we look at the columns of the same binary map to associate each system tracks with all ground truth tracks it matches to. This procedure reveals track false positive errors and track merge errors. For unique correspondence, we use the same two-pass approach, but this time on the two different correspondence maps. In the first pass, each ground truth track is matched against all the tracker result tracks. The resulting correspondences are shown in left hand side of the figure for unique correspondence. This reveals track false negative but fails to capture track fragment errors (no multiple associations allowed). In the second pass, each tracker result is matched against all ground truth tracks. The resulting correspondences are shown in the binary maps on the right hand sides. This reveals track false positive but fails to capture track merge errors.

Figure 3(a) shows an example situation where track merge error (object 1 and 2 joined as 1) and track false positive error (system detects an extra object, 2) occurs. Also, there is no track fragmentation error (one ground truth object split into two) or track false negative error (a ground truth object missed). Here, object 1 leaves the scene in the right hand center corner of the camera field of view (FOV). After that, a second object enters from a close location and moves towards the left hand side of the FOV. A tracker system which bases its tracking on the history of motion pattern (using Kalman prediction or particle filtering based approach), always has a certain time lag to allow for consistent tracking in the event of occlusion. This property can result in object labeling error, where the tracker system mistakes the object 2 entering the scene from neighboring location not too long after object 1 has left. In this situation, the tracker system can mistake object 2 for object 1 and merge the two objects into 1. This error resulting from object label ambiguity causes the corresponding system tracks to be labeled as merged tracks. In this particular example, ground truth tracks 1 and 2 will be labeled as merged tracks. Similarly, Figure 3(a) shows another error visible in the system track's output. The system track 2 has no corresponding ground truth. It could have resulted from some noise in the video sequence data. Both these errors are captured in many-to-many correspondence maps of (c). Here, column 1 reveals that tracker track 1 is matched to both ground truth tracks 1 and 2, resulting in track merge error. Also, tracker track 2 has no corresponding ground truth track, resulting in track false positive. In the unique correspondence results, the left hand correspondence map is the same as for many-to-many correspondence resulting in the same false positive, but no merge errors as expected. Also, both these algorithms yield no track fragment and false negative errors.

On similar lines, Figure 3 (b) shows a situation where track fragment error and track false negative errors occur, but there are no track merge and track false positive errors. In this situation, tracker loses the object 1 for a brief period of time due to occlusion. After a while it detects it again, but this time assigns it a different label due to object label ambiguity. Also, ground truth track 2 is not even detected. Both the errors resulting in this scenario are captured in the many-to-many binary correspondence maps shown in figure (d). Unique correspondence misses the track fragment error as expected. Also, no track false positive or track merge errors are detected.

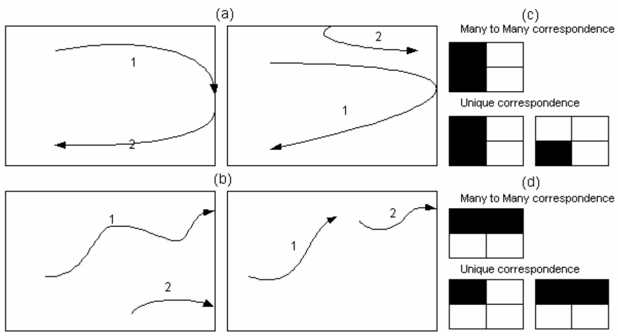

Figure 3: (a) Example of a track merge error scenario. (b) Example of a track fragment error scenario. (c) & (d) Object correspondence maps for the two scenarios.

Figure 4 outlines the algorithm used for generating correspondences. For simplicity, it shows the complete portion for first pass detailing system results to ground truth mapping. The second pass, which matches ground truth to system results, is a straightforward symmetrical implementation and is skipped for brevity. In this algorithm, the core problem is detecting whether the two bounding boxes, one from system track results and the other from ground truth coincide or not. To answer this question, we have implemented three strategies as explained below.

The first strategy is based on Euclidean distance between centroids of the two boxes. If the distance is between a preset threshold, the two boxes are said to be coincident and an association declared between the two objects. The second approach tests if the centroid of one of the boxes is within the bounds of the other box. It inflates the first box by a fixed percentage in both x- and y- directions, and then tests if the centroid of second box lies inside the first one or not. Finally, the third approach computes the ratio of the intersection and union of the two boxes. If this ratio is with a fixed threshold between 0 and 1, the two objects are declared to have a match.

4. Simulation Environment and Results

This section outlines our simulation test environment, test dataset and results obtained using the metrics discussed in the previous section.

4.1 Automatic Object Detection and Tracking Systems

This section very briefly outlines the automatic object detection and tracking system for video surveillance that we have tested. These systems will be evaluated based on metrics detailed in the previous section. The video surveillance system consists of a background generation module coupled with several object detection and tracking modules in an easy-to-bind API (application programming interface). This allows us to provide a convenient user interface for testing various combinations of object detection and tracking modules. We have tested three object detection modes and two tracking modules for our surveillance system. The first mode is manual detection where the user draws an object template (bounding rectangle) on an initial frame; the second detection mode is template detection from background estimation; the third detection mode is connected component-based detection from background estimation. We have tested two of recently reported tracking systems for performance evaluation. The first tracking system is ‘multi-kernel mean-shift’ which uses multiple meanshift kernels centered at the high motion areas. Details about this system can be found in [9]. The second system is ‘ensemble tracker’ which poses tracking as a binary classification problem, where an ensemble of weak classifiers is trained online to distinguish between object and background [10]. All the results in this paper are tested on the output generated by these two trackers in conjunction with three detectors to yield six combinations for evaluation.

```
for each tracker track i
{
  for each GT track j
  {
    for each overlapping frame k
    {
      tov[i][j] ++
      if (box[i][k] coincides box[j][k])
        sov[i][j] ++
    }
    tov[i][j] /= totalframes
    sov[i][j] /= totalframes
    fov[i][j] = α1 tov[i][j] + α2 stov[i][j]
    if (fov[i][j] > T1)
      TR_GT[i] ++
  }
  if (TR_GT[i] == 0)
    TFP ++
  if (TR_GT[i] > 1)
    TME += TR_GT[i]
}
```

Figure 4: Algorithm for many-to-many tracker to ground truth matching.

4.2. Simulation test-bed

In order to perform the evaluation tests of our system against ground truth data, we have developed a GUI-driven desktop application (different from the video surveillance system application mentioned in Section 4.1). For a test sequence, the user can load object localization information from ground truth and tracker results files. The data from ground truth and tracker results is stored in the eXtensible Markup Language (XML) format. The specific XML format followed is that proposed by Computer Vision Markup Language (CVML) [8]. This xml-based file format ensures that results from different teams can be tested on our systems without any portability issue. The cvml-parser module of the application parses the input data files from both tracker results and ground truth to fill in internal data structures. The two results are then compared according to the rules and metrics discussed in Section 3. An attractive feature of the application is batch processing mode, where a set of sequences to be evaluated can be specified in a separate file and the system performs evaluation on all these sequences automatically. Although the processing time for each sequence is quite manageable (around 15-20 seconds per sequence on a Pentium-IV, 3.0 GHz desktop), running the application for hundreds of sequences becomes tedious. In batch processing mode, the application can be left running unsupervised to generate results. The outputs of individual sequences are written out as soon as they are available for each sequence. At the end, the output of batch processing is written out along with means and variances for each metric.

4.2. Dataset

We have used a mix of both standard and in-house datasets for our evaluation purposes. The standard dataset is based on the PETS 2001 dataset. We have also tested our system on some other in-house and publicly available sequences of varying length. The total number of sequences tested in our evaluation experiments is around 40. The test sequences consist of more than 50,000 frames and depict both indoor and outdoor scenarios; partial and full occlusions; various object types, such as pedestrians, vehicles, bicycles, etc. The results of these experiments are detailed next.

4.3. Results

We have tested automatic object detection and tracking systems using test metrics discussed in the previous section on the dataset in batch processing mode. Partial results of this evaluation based on TRDR and FAR are reported in Figure 1. The full results of this evaluation in the form of means and variances of each metric are presented in Table 1 for 'multi-kernel meanshift' tracker and in Table 2 for 'ensemble tracker'.

5. Summary and Conclusions

In this report, we have addressed the issue of unbiased performance evaluation of object detection and tracking systems. We have made contributions in two areas: a set of novel performance evaluation metrics have been proposed for detection and tracking; novel methods of establishing correspondence between ground truth tracks and system generated tracks have been proposed. Our set of metrics contains both frame-based metrics (to test the performance of detection system) as well as object-based metrics (to test the tracking capabilities including consistent object labeling). Experiments have been conducted on a standard dataset containing more than 50,000 frames. The cumulative results of these experiments in terms of mean and variance values for each metric are reported.

References

[1]. Robert T. Collins, Alan J. Lipton, Takeo Kanade, "Introduction to the Special Section on Video Surveillance", IEEE Transactions on PAMI, Vol. 22, No. 8, August 2000, pp. 745 - 746.

[2]. David P. Young, James M. Ferryman, "PETS Metrics: On-Line Performance Evaluation Service", Proceedings 2nd Joint IEEE International Workshop on VS-PETS, Beijing, October 15-16, 2005.

[3]. James Black, Tim Ellis, Paul Rosin, "A Novel Method for Video Tracking Performance Evaluation", The Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, October, Nice, France, pp. 125-132. (2003).

[4]. Robert Collins, Xuhui Zhou, Seng Keat The, "An Open Source Tracking Testbed and Evaluation Web Site", IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), January 2005.

[5]. Stefan Muller-Schneiders, Thomas Jager, Hartmut S. Loos, Wolfgang Niem, "Performance Evaluation of a Real Time Video Surveillance Systems", Proceedings 2nd Joint IEEE International Workshop on VS-PETS, Beijing, October 15-16, 2005.

[6]. Lisa M. Brown, Andrew W. Senior, Ying-li Tian, Jonathan Connell, Arun Hampapur, "Performance Evaluation of Surveillance Systems under Varying Conditions", IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance, Colorado Jan., 2005 .

[7]. Vladimir Y. Mariano, Junghye Min, Jin-Hyeong Park, Rangachar Kasturi, David Mihlacik, Huiping

Li, David Doermann, Thomas Drayer, "Performance Evaluation of Object Detection Algorithms", International Conference on Pattern Recognition, pp. 965-969, 2002 .

[8]. Thor List, Robert B. Fisher, "CVML – An XML-based Computer Vision Markup Language", 17th International Conference on Pattern Recognition (ICPR 2004), Vol. 1, pp. 789-792. 2004.

[9]. Fatih Proikli, Oncel Tuzel, "Multi-Kernel Object Tracking", IEEE International Conference on Multimedia and Expo (ICME 2005), pp. 1234-1237, July 2005.

[10]. Shai Aviden, "Ensemble Tracking", IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), ISSN: 1063-6919, Vol. 2, pp. 494-501, June 2005.

Table 1: Tabular results of performance evaluation on 'Multi-kernel Meanshift Tracking System' with various automatic object detection methods for around 40 video sequences from PETS and in-house data set.

Metric			**Manual Meanshift**		**Template Meanshift**		**CC Meanshift**	
			Mean	**Var**	**Mean**	**Var**	**Mean**	**Var**
Frame-based	TRDR		0.480929	0.060065	0.545417	0.040047	0.513411	0.07145
	FAR		0.225161	0.043955	0.2133	0.038476	0.225849	0.051534
	Detection Rate		0.480929	0.060065	0.545417	0.040047	0.513411	0.07145
	Specificity		0.654807	0.054624	0.581161	0.059394	0.617162	0.085862
	Accuracy		0.547126	0.04946	0.589513	0.026975	0.551378	0.050211
	Positive Predictive Value		0.774839	0.043955	0.7867	0.038476	0.774151	0.051534
	Negative Predictive Value		0.314004	0.046597	0.285508	0.029245	0.296549	0.051285
	False Negative Rate		0.519071	0.060065	0.454583	0.040047	0.486589	0.07145
	False Positive Rate		0.345193	0.054624	0.418839	0.059394	0.382838	0.085862
Object-based	Euclidean Threshold	TRDR	0.646242	0.068983	0.717953	0.030324	0.813559	0.035763
		FAR	0.203123	0.043281	0.166291	0.023183	0.17383	0.04095
		Detection Rate	0.696041	0.061334	0.610564	0.038849	0.787851	0.047174
		Specificity	0.688858	0.04373	0.627918	0.048623	0.635475	0.078635
		Accuracy	0.734413	0.031448	0.779488	0.012852	0.834226	0.020197
		Positive Predictive Value	0.796877	0.043281	0.833709	0.023183	0.82617	0.04095
		Negative Predictive Value	0.528668	0.062497	0.36017	0.061787	0.561818	0.079523
		False Negative Rate	0.29265	0.056773	0.381619	0.037059	0.204333	0.042613
		False Positive Rate	0.311142	0.04373	0.372082	0.048623	0.364525	0.078635
		Track False Positive	0.212707	0.167463	1.251522	6.453521	1.069031	3.504285
		Track False Negative	0.391796	0.508262	0.168998	0.15607	0.060968	0.072883
		Track Merge Error	0	0	0.05093	0.048336	0.10803	0.09636
		Track Fragment Error	0.235736	0.180164	0.82442	0.765946	0.286737	0.420579
	Centroid in Rectangle	TRDR	0.597409	0.084026	0.675086	0.042262	0.65172	0.101447
		FAR	0.255427	0.070712	0.209608	0.046633	0.33233	0.103508
		Detection Rate	0.699214	0.071628	0.65212	0.048812	0.703334	0.100348
		Specificity	0.715978	0.02892	0.643301	0.049125	0.603593	0.063744
		Accuracy	0.803126	0.033313	0.804321	0.015131	0.842247	0.022605
		Positive Predictive Value	0.744573	0.070712	0.790392	0.046633	0.66767	0.103508
		Negative Predictive Value	0.621375	0.087312	0.470762	0.074644	0.658076	0.08775
		False Negative Rate	0.252056	0.049838	0.340063	0.046372	0.192914	0.047391
		False Positive Rate	0.284022	0.02892	0.356699	0.049125	0.396407	0.063744
		Track False Positive	0.807649	0.162651	1.911387	8.703765	1.735889	4.240596
		Track False Negative	0.885987	0.53496	0.545911	0.672115	0.582771	0.926545
		Track Merge Error	0	0	0	0	0	0
		Track Fragment Error	0.248278	0.299929	0.70882	0.780362	0.105397	0.094289
	Area Ratio Overlap	TRDR	0.235838	0.120703	0.431792	0.103333	0.251769	0.102233
		FAR	0.754213	0.13532	0.491049	0.132944	0.738237	0.111596
		Detection Rate	0.307273	0.187835	0.448761	0.105656	0.353636	0.143692
		Specificity	0.669942	0.014249	0.565374	0.041517	0.532402	0.02464
		Accuracy	0.971789	0.004212	0.827823	0.026912	0.899033	0.022285
		Positive Predictive Value	0.245787	0.13532	0.508951	0.132944	0.261763	0.111596
		Negative Predictive Value	0.920536	0.029111	0.613843	0.123858	0.814865	0.070679
		False Negative Rate	0.033428	0.007289	0.265242	0.05317	0.171212	0.057013
		False Positive Rate	0.330058	0.014249	0.434626	0.041517	0.467598	0.02464
		Track False Positive	1.829033	0.899422	2.618809	8.816763	2.591081	7.236438
		Track False Negative	1.885679	0.745644	0.981076	1.285967	1.476057	1.550727
		Track Merge Error	0	0	0	0	0	0
		Track Fragment Error	0	0	0	0	0	0

Table 2: Tabular results of performance evaluation on 'Ensemble Tracking System' with various automatic object detection methods for around 40 video sequences from PETS and in-house data set.

Metric			Manual Ensemble		Template Ensemble		CC Ensemble	
			Mean	Var	Mean	Var	Mean	Var
Frame-based	TRDR		0.583182	0.084726	0.613543	0.060189	0.447507	0.065313
	FAR		0.238632	0.049738	0.210794	0.044481	0.282596	0.056876
	Detection Rate		0.583182	0.084726	0.613543	0.060189	0.447507	0.065313
	Specificity		0.610320	0.051911	0.598695	0.059042	0.532792	0.080033
	Accuracy		0.625534	0.06033	0.642713	0.041356	0.486041	0.051821
	Positive Predictive Value		0.761368	0.049738	0.789206	0.044481	0.717404	0.056876
	Negative Predictive Value		0.359372	0.048083	0.340331	0.036629	0.242104	0.035727
	False Negative Rate		0.416818	0.084726	0.386457	0.060189	0.552493	0.065313
	False Positive Rate		0.389680	0.051911	0.401305	0.059042	0.467208	0.080033
Object-based	Euclidean Threshold	TRDR	0.74025	0.059105	0.736054	0.038256	0.779085	0.069626
		FAR	0.205657	0.047088	0.182218	0.033953	0.22252	0.069671
		Detection Rate	0.792952	0.053303	0.772397	0.033471	0.741613	0.079402
		Specificity	0.647741	0.043084	0.660785	0.043207	0.576553	0.062554
		Accuracy	0.826679	0.018761	0.819638	0.014937	0.854983	0.015251
		Positive Predictive Value	0.794343	0.047088	0.817782	0.033953	0.77748	0.069671
		Negative Predictive Value	0.618142	0.048588	0.531993	0.044368	0.541479	0.097811
		False Negative Rate	0.182036	0.038022	0.219787	0.029152	0.201533	0.048697
		False Positive Rate	0.352259	0.043084	0.339215	0.043207	0.423447	0.062554
		Track False Positive	0.353382	0.319173	1.260737	5.025566	1.548544	5.271998
		Track False Negative	0.493006	0.717945	0.402995	0.562953	0.200675	0.448046
		Track Merge Error	0	0	0.05093	0.048336	0.05093	0.048336
		Track Fragment Error	0.188662	0.153069	0.331578	0.323494	0.627201	1.042936
	Centroid in Rectangle	TRDR	0.657232	0.094717	0.692611	0.054343	0.581208	0.149388
		FAR	0.290587	0.091443	0.225234	0.057475	0.414578	0.149851
		Detection Rate	0.827868	0.062887	0.803629	0.040309	0.583884	0.152444
		Specificity	0.680205	0.031417	0.679881	0.03387	0.565269	0.053782
		Accuracy	0.876542	0.017114	0.830707	0.022513	0.861936	0.027627
		Positive Predictive Value	0.709413	0.091443	0.774766	0.057475	0.585422	0.149851
		Negative Predictive Value	0.741183	0.051225	0.61244	0.052718	0.658277	0.109339
		False Negative Rate	0.139386	0.040342	0.188554	0.035502	0.182695	0.058798
		False Positive Rate	0.319795	0.031417	0.320119	0.03387	0.434731	0.053782
		Track False Positive	0.992348	1.80733	1.835692	7.703544	2.464621	8.711641
		Track False Negative	0.943311	1.853212	0.697301	1.099339	0.832812	2.121629
		Track Merge Error	0	0	0	0	0	0
		Track Fragment Error	0.049037	0.046633	0	0	0.292332	0.265949
	Area Ratio Overlap	TRDR	0.230048	0.095952	0.384071	0.122701	0.179283	0.059764
		FAR	0.746713	0.106427	0.563954	0.154915	0.821946	0.062503
		Detection Rate	0.392012	0.197961	0.505314	0.180036	0.330953	0.159903
		Specificity	0.591268	0.011419	0.627483	0.017292	0.516166	0.013807
		Accuracy	0.948076	0.013274	0.910729	0.014953	0.888639	0.0217
		Positive Predictive Value	0.253287	0.106427	0.436046	0.154915	0.178054	0.062503
		Negative Predictive Value	0.904852	0.045538	0.805123	0.062678	0.888601	0.028055
		False Negative Rate	0.070059	0.024773	0.097039	0.017687	0.154731	0.069269
		False Positive Rate	0.408732	0.011419	0.372517	0.017292	0.483834	0.013807
		Track False Positive	1.587296	1.775537	2.368109	8.785345	3.405052	11.655499
		Track False Negative	1.587296	1.775537	1.229719	1.468704	1.480912	0.905387
		Track Merge Error	0	0	0	0	0	0
		Track Fragment Error	0	0	0	0	0	0