Qualitative Evaluation of Detection and Tracking Performance

Swaminathan Sankaranarayanan
Graduate Student
TU Delft, Netherlands
S.Sankaranarayanan@student.tudelft.nl

Dr.Francois Bremond
Permanent Researcher
INRIA Sophia Antipolis, France
Francois.Bremond@inria.fr

Dr.David Tax
Assistant Professor, PRB lab
TU Delft, Netherlands
D.M.J.Tax@tudelft.nl

***Abstract*—A new evaluation approach for detection and tracking systems is presented in this work. Given an algorithm that detects people and simultaneously tracks them, we evaluate its output by considering the complexity of the input scene. Some videos used for the evaluation are recorded using the Kinect sensor which provides for an automated ground truth acquisition system. To analyze the algorithm performance, a number of reasons due to which an algorithm might fail is investigated and quantified over the entire video sequence. A set of features called Scene Complexity measures are obtained for each input frame. The variability in the algorithm performance is modeled by these complexity measures using a polynomial regression model. From the regression statistics, we show that we can compare the performance of two different algorithms and also quantify the relative influence of the scene complexity measures on a given algorithm.**

I. Introduction

The approach to performance evaluation can be majorly classified into two types: Evaluation with ground truth in which case it is performed offline ([1],[3]) and Online tracker performance evaluation without ground truth ([8]). Several methods have been proposed to evaluate the tracker performance in either case. The offline performance evaluation remains as a benchmark against which the online methods are compared with respect to accuracy whereas the online performance evaluation does not require any ground truth data which saves a lot of time and labour.

The existing approaches to tracking evaluation have limitations in providing qualitative information about the tracker performance. The metrics used for evaluation at best give out two types of information: how accurate the detection algorithm performs and how consistent the tracker is. They do not address the question of why a tracker has performed poorly (the question of 'why' generally arises in light of poor performance). Consider two tracking algorithms T1,T2 and an input video which has lighting changes over the entire sequence. Only T1 is capable of handling high variations in illumination but in other aspects T2 is much better than T1. In such a scenario, the metrics might report that T1 had performed exceedingly well but it cannot be inferred directly from the metrics why T1 has failed. This paper makes an attempt in providing such qualitative information by pointing out how a tracker performs in the frames that matter. The main contribution of this work is to explain the tracker performance by using a set of Scene Complexity measures that quantify the difficulty of each input frame. In addition, we also attempt to compare different tracking algorithms based on the regression model parameters that is obtained to describe their performance.

The issues faced by any tracking system can be classified into two types based on the input. The failures of the tracker due to occlusion/lighting changes etc., can be attributed to the variation in the input video. The issues that affect the tracker that are independent of the input video can be variation in encoding of the ground truth and limitations of the metrics which measure the tracker performance. This work mainly concentrates on describing how a tracker is affected by the input and compares the results with the state-of-art evaluation metrics used for tracking. The experiments are run on indoor video sequences recorded using the Kinect sensor. The rest of the paper is organized as follows: Section II explains the different approaches taken in the past for formulating ground truth including the methods for overcoming the subjectivity of the formulations. Section III presents the methodology of the proposed evaluation framework. Section IV explains the computation of the various complexity measures.

II. Related Work

With the development of many tracking algorithms over the years, the evaluation of tracking systems has become an important area to concentrate on. Various attempts have been made in the past to evaluate trackers based on spatial and temporal information that is given out by the tracker. The result of the evaluation system is usually a set of numbers based on the output of the algorithm and the metrics. This section surveys the related work in the field of tracking evaluation and states how the current approach can be seen as an improvement. The notation used in this section and the rest of the paper is defined as follows :

- GT refers to the ground truth information
- N_x refers to the count of x where x can be false positives (f_p), false negatives(f_n), true positives (t_p) or Ground Truth objects(g_t).
- N_{frames} is the total number of frames in the video sequence for which GT is available.

Bashir et al [1] provide two sets of tracking metrics, Tracker Detection Rate (TRDR) and False Alarm Rate (FAR) which are linear functions of N_{t_p},N_{f_p} and N_{g_t}.

Brown et al [3] propose a two pass scheme for matching output tracks and ground truth tracks. In the first pass, many to one association is performed from system tracks to (many) ground truth track(s) (not vice versa) based on temporal and spatial overlap. In phase two, multiple system tracks are matched to each GT track. After the matching procedure, the output metric is given as average spatial and temporal overlap over the entire sequence.

Bernardin et al [2] define a set of metrics for tracking and detection as part of the CLEAR workshop. The tracking performance is expressed by two numbers: "tracking precision", which expresses how well exact positions of objects are estimated, and "tracking accuracy", which shows how many mistakes the tracker made in terms of misses, false positives and mismatches. The mapping procedure makes sure that the Split or Merge errors committed by the tracker are accounted for.

All the above methods evaluate the tracker solely based on the tracker output and do not relate the characteristics of the input video to the tracking algorithm. Nghiem et al [6] present an evaluation approach in which they derive the upper bound for the algorithm capacity to handle different video processing problems. The most recent approach towards measuring the complexity of the video sequence was done by Chu et al [4] in their work describing thirteen hard cases for Visual tracking. They provide a very good account of the different factors, which we shall address as failure modes, that affect the tracking performance and have managed to quantify them and compare different trackers based on these factors. The input videos in [4] are preset videos and each video varies only in one failure mode. The evaluation framework developed in this work measures different complexity measures for any given video and is thus a more general approach to evaluate any tracker.

III. Qualitative Evaluation Framework

We propose an evaluation framework for tracking systems that give out qualitative information about the performance of the tracking system. The reasons for the errors of a detection/tracking algorithm cannot be understood clearly using the available evaluation metrics. The metrics usually quantify the magnitude of the errors being made and not why such errors are made. When the video sequences analyzed are larger in size, an Evaluation system that can point out specific frames that need to be looked at, would come in very handy. To analyze the algorithm's performance over a video sequence, the complexity of the frames has to be quantified. The cases of complexity that are considered in this work are Occlusion, Clutter, Changes in Lighting and Lower Contrast Objects. The following procedure is followed to perform this evaluation :

- The value of an evaluation metric which represents the performance of the algorithm is obtained for each frame. The metric value can be chosen as any of the framewise metrics used for tracking evaluation. For the current work, we choose the Frame Detection Accuracy (FDA) metric, which is defined in [2].
- Scene complexity measures SC_t^i are computed simultaneously for each frame.
- A polynomial regression model is fit with the metric value as dependent variable and the complexity measures as predictor variables. The model accounts for linear, interaction and quadratic effects.

$$MV_t = f(SC_t^i, (SC_t^i)^2, (SC_t^i * SC_t^j))_{i,j=1..4} \quad (1)$$

where MV_t is the metric value for the frame t and SC_t^i is the i^{th} Scene Complexity for frame t. Function f represents a polynomial regression equation that accounts for the interaction effects between the predictor variables.

The goodness-of-fit of the regression model can be evaluated using the F-score and/or the R-square statistic. Once the regression coefficients are obtained, we can compare them to identify which complexity measure affects the algorithm the most. It should be noted that we cannot expect the regression model to explain the entire variability in the metric values since the list of Complexity measures is by no way exhaustive. The computation of the Complexity measures is explained in detail in the following section.

IV. Measures of Complexity

A. Occlusion Complexity

An occlusion occurring in the scene can be classified as either Static or Dynamic. A static occlusion is the one in which a moving object in the scene overlaps with a static object, eg. a person overlapping with a piece of furniture. Dynamic occlusion is the one where two moving objects overlap eg. a person overlapping with another.

To measure Static Occlusion effectively, information about static objects in the scene should be provided. In the absence of this, the SOC measure cannot be used. For the current work, we are not considering the complexity that arises due to Static Occlusion.

A Dynamic occlusion situation consists of two mobile overlapping objects, typically two or more persons crossing each other's path in a video. Dynamic Occlusion is measured as the amount of overlap between the Occluding Objects output by the tracking system. When two people cross each other in the scene, it is inevitable that a part of one person is merged with the other. By observing the behavior of some detection systems we found that there are two specific cases which should be considered when computing the Dynamic Occlusion Complexity measure. A simple block diagram representing both the cases are shown in Fig.1.

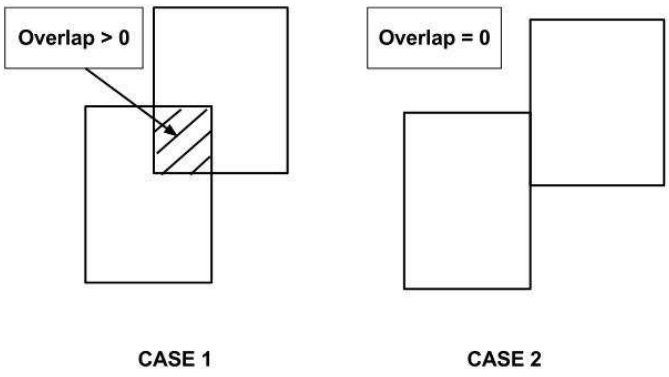

Figure 1: Different cases when to consider Dynamic Occlusion Complexity

- Case 1:
  This is the trivial case of Dynamic Occlusion where the objects in question overlap and hence the complexity measure is a straight-forward computation of the overlap between pairs of objects output by the tracking system for the entire frame. The complexity measure for this case can be formulated as:

$$M_1(t) = \frac{1}{N_t^2}\left(\sum_{i=1}^{N_t-1}\sum_{j=i+1}^{N_t}\frac{Overlap(i,j)}{area_i + area_j}\right) \quad (2)$$

  where $Overlap(i,j)$ is the common area of the bounding box enclosing the detected objects i and j, $area_i$ is the area of the bounding box enclosing detected object i and N_t is the number of objects in frame t.
- Case 2:
  This case involves a situation where the objects occur close enough to each other in a given frame but do not directly overlap. When people walk close enough to each other, this provides enough distraction for the detection algorithm to merge them as a single person. To account for these errors, we have to compute the Occlusion complexity for the current case as follows:

$$M_2(t) = 1 - \frac{1}{N_t^2}\left(\sum_{i=1}^{N_t-1}\sum_{j=i+1}^{N_t}\frac{|c_{ix} - c_{jx}|}{w_i + w_j}\right) \quad (3)$$

where c_{ix} is the x-coordinate of the centroid of the bounding box enclosing object i and w_i is the width of the same bounding box. This Occlusion measure is searched for in both x and y dimensions. The final measure is an arithmetic mean of the Static and Dynamic Occlusion Complexities.

B. Lighting Complexity

Changes in lighting conditions on a given scene presents a tough challenge to any tracking system. There have been several model based approaches in the past that try to predict the pattern in lighting change and find the best fit. We measure the complexity due to lighting changes by analyzing the intensity distribution of the input scene. The difficulty associated by a change in illumination can be analyzed by considering the following:

- A change in illumination between consecutive frames in the video can distract a detection algorithm very much.
- The amount of lighting incident on a given frame determines how good the detection algorithm can perform. If the intensity histogram is peaked at the extremes, ie if the lighting is either too high or too low, then most detection approaches cannot deliver a good performance. Fig. 2 gives an example of the case being discussed here.

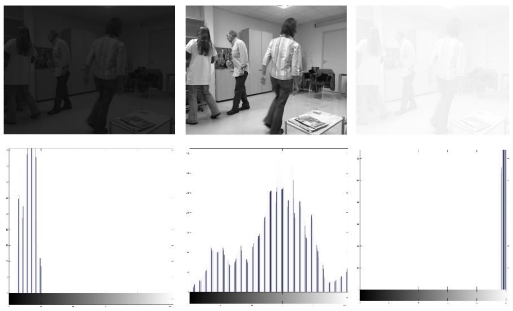

Figure 2: Illustration of Global changes in illumination. The image in the middle is the original image. The images at the left and right show peaked intensity version of the original image

Thus, to measure any illumination change completely both global and local changes have to be accounted for. In addition to this, the inherent complexity of the lighting intensity (peakedness of the histogram) should also be measured. The following procedure is followed to compute the final measure.

The Global change for illumination is measured by computing the mean difference in the intensity values between the grayscale version of the current and the previous images.

The next step is to check for any local changes in illumination using the corrected image from the previous step. To check for local changes in illumination, a given image frame t is split into multiple equally sized patches. These patches traverse the entire image and does not exclude any foreground pixels. For each image patch, an intensity histogram is computed and compared with the histogram of the corresponding image patch in frame $t-1$. The distance measure used here is the Earthmover's distance (EMD) which has a very good ability to differentiate intensity histograms. Let H_i^t be the intensity histogram of the patch

i. The final measure is computed as follows:

$$EMD(I_t, I_{t-1}) = \frac{1}{N}\sum_{i=1}^{N} EMD(H_i^t, H_i^{t-1}) \tag{4}$$

The final step is to compute the complexity due to the existing lighting in the image. If the intensity histogram is highly peaked, then it is highly likely that the detection algorithm would fail. To measure this complexity, the kurtosis measure of the intensity histogram is computed.

$$K(I_t) = E\left[\left(\frac{H(I_t) - \mu}{\sigma}\right)^4\right] \tag{5}$$

C. Quantifying Clutter

Measuring Clutter is more a subjective problem than a statistical one. It becomes very important in this case to correlate the statistical measure that we obtain to quantify clutter with how it describes the images that are being used. Clutter is a very broad term and it is important to specify what is being measured and hence what is to be expected of such a metric. Clutter can mean either textural changes in foreground or background or even Static or Dynamic occlusion that occurs in the foreground. The Clutter metric used here mainly deals with Variations in Texture. Any scene which is highly textured can be associated with a very high value for clutter. From an object detection perspective, it is very difficult to search for an object (person) in an highly textured environment. Thus, the metric for clutter should typically give higher score to highly cluttered environments and lower scores otherwise.

The metric for measuring Clutter that we have chosen to adopt is from Rosenholtz et al [7], who propose a clutter measure using the concept of feature congestion. Their feature congestion clutter measure is based on a Statistical Saliency Model (SSM), which models the degree to which a feature vector $\mathbf{T}$, is an outlier to the local distribution of feature vectors with mean μ and covariance Σ. The saliency is given as follows:

$$\delta = \sqrt{(\mathbf{T} - \mu)'\Sigma^{-1}(\mathbf{T} - \mu)} \tag{6}$$

The clutter value is an aggregate of the saliency values across scales.

D. Complexity due to Low Contrast Objects

The detection algorithm faces a very difficult task when it encounters objects which are very similar in contrast to the background. Not many algorithms are robust to this kind of complexity. A metric used to quantify low contrast object complexity should take into account the contrast level of the detected objects with respect to their neighbourhoods. We start by identifying a region around the each detected object which is a rectangular region bigger than the dimensions of the object. In case of Kinect recording, the detected object is a shape countour and only the relevant pixels are used for contrast computation. In general, the detector output is a rectangular bounding box. The Root Mean Square (RMS) contrast for both the image patches is computed in the three color channels (RGB). The resulting complexity measure for a detected object is the difference in RMS contrast between the object bounding box and its surrounding rectangle. This can be formulated as follows:

$$CC_t = 1 - \sum_{i=1}^{N_t} \frac{(|RMS_s - RMS_o|)_i}{C_{max}} \tag{7}$$

where RMS_o is the RMS contrast of the object, RMS_s is the RMS contrast of the region surrounding the bounding box , C_{max} is the maximum intensity value of the corresponding channel and N_t is the number of objects in frame t.

V. Experiments

The experiments were carried out over two video sequences both of which were recorded using Kinect sensor. Sequence **S1** consists of 300 frames with two people walking into a room with changes in lighting and a simple case of occlsuion. Sequence **S2** spans over 400 frames in a cluttered room consisting of two people initially and two others crossing the room in the middle of the video presenting a difficult case for occlusion. Three different tracking systems were evaluated on these sequences which can be listed as follows :

- T1 - Processing chain consisting of People Detection from [9] and tracking from [10].
- T2 - Raw tracking output from the OpenNi library using the depth map from the Kinect sensor.
- T3 - Processing chain consisting of People Detection from [5] and tracking from [10].

As explained earlier, the each tracking system is evaluated using anavailable evaluation metric and the variability in the metric is modeled using a polynomial regression fit. The metric used here is the Frame Detection Accuracy (FDA) which is a framewise measure of the detection accuracy over the successfully tracked objects. We have chosen this metric since it provides a compact representation of both detection inaccuracies and the tracking errors committed by the algorithm. The results we have obtained for the credibility (goodness-of-fit) of the regression models are shown in Figure 3. The results of the MOTP metric ([2]) for tracking evaluation is shown in Figure 4. It can be seen that the prediction ability of the model is directly related to how erroneous the tracking algorithm performs. Thus we already have a measure which represents the performance of the tracking system on the entire sequence.

It should be noted here that the *F-score* is the coefficient of determination which is used to evaluate regression models and the r^2 value explains the amount of variation in the dependent variable that the model is able to explain. The

	T1		T2		T3	
	F-score	r^2	F-score	r^2	F-score	r^2
S1	36.06	0.63	66.1	0.72	21.26	0.48
S2	28.66	0.58	91.77	0.78	43.74	0.66

Figure 3: Table showing the goodness-of-fit(GoF) measure for the regression models in terms of F-score and its associated r^2 value.

	T1	T2	T3
S1	0.73	0.81	0.49
S2	0.41	0.71	0.55

Figure 4: Table showing the MOTP values for sequence **S1**

polynomial regression model that we use consists of linear, interaction and quadratic terms. Before using the polynomial model, we should identify if all the predictor variables are relevant. To do this, we use the stepwise fitting approach. We start with the empty model and keep adding each predictor variable to the model and continue with the model if a given criterion is achieved. The criterion that we use here is the increase in F-score and its significance. This procedure is performed iteratively until the increase in the F-score is not significant. At the end of this procedure, we end up with the relevant complexity measures for the model.

Let us look at an example which demonstrates this procedure. Consider the evaluation of the output of algorithm **T2** on sequence **S1**. Some frames of interest of sequence **S1** are shown in Figure 5. The output data that we have are the FDA metric values for each frame and the four scene complexity measures for each frame which are the predictor variables for our regression model. Figure 3 shows the GoF values for the model using all the predictor variables. If we perform a stepwise regression to select the optimal set of predictor variables, we get the result shown in Figure 6.

The excluded column here (Column 3) is the Clutter Complexity measure. If we perform the polynomial regression fit with the updated model we get the following result: $F - score = 58.75, p \ll 0.001$. Though the F-score is slightly lesser in this case, this is achieved with a lesser number of parameters and the p-value is still very low indicating the credibility of the model is very high. If we look at the adjusted R-square coefficient for the two models, the full model has the value $r^2 = 0.72$, while the current model has the value $r^2 = 0.709$. Thus we do not lose any information at all but we have a simpler model now. This step, though not necessary, is generally recommended especially when there is a danger of over-fitting when using many parameters.

We run different regression models using the reduced set of predictor variables which include interaction and quadratic terms. The best possible fit that we obtain that

(a) Frame 75 (b) Frame 90

(c) Frame 105 (d) Frame 120

Figure 5: Interest Frames from sequence **S1**

Step	Column Added	p-value of model
1	4	4.4e-08
2	1	3.4e-08
3	1	0.01

Figure 6: Table showing the output of the stepwise regression procedure

we obtain for **S1** with algorithm **T1** is the following:

$$FDA = 0.29 - 0.39 * OC - 0.09 * (LCC) \quad (8)$$

This model shows that the algorithm is most influenced by the Occlusion complexity measure and slightly influenced by the presence of low contrast objects. The lighting effect was not observed as a significant factor from which we can infer that the algorithm copes quite well with Illumination Changes. For the purpose of comparing two algorithms, we shall run the same procedure for algorithm **T3** on the same sequence **S1**

$$FDA = 0.19 - 0.76 * OC - 0.71 * (OC * IC) \quad (9)$$

The magnitude of the regression coefficients and the terms present in the model equation show how the Scene Complexity measures affect the observed metric values. This prediction was made based on the model parameters and the input video. From the regression equations above, we can see that system **T3** is more sensitive to illumination changes and occlusion while system **T1** is prone to occlusion and weakly contrast objects but to a lesser extent. The information obtained from the two models can be further visualized by plotting the standard errors of the regression models in both the cases, as shown in Figure 7. We can observe that the model is able to explain the **T1** system better than the **T3** system. The pointers in the inset show the specific video frames where the model commits large

errors. For the **T3** system there are two such portions which signifies the portion of the video where Lighting changes and Occlusion occurs. For the **T1** system there is only a single region where Occlusion occurs. The relative magnitude of errors are also larger for the **T3** system. These results also confirm the predictions that were made earlier using the regression equations.

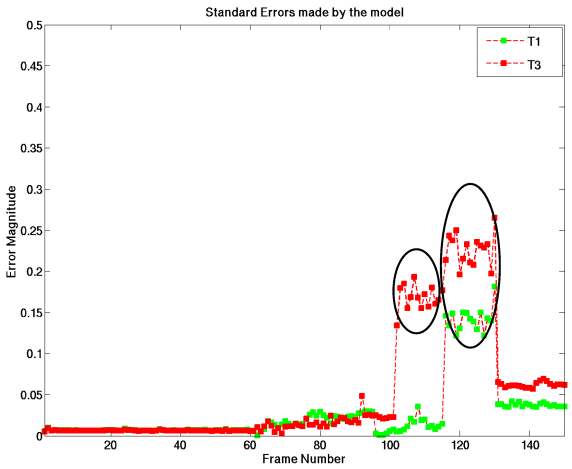

Figure 7: Plot of Errors (vs) Frame Number for algorithms T1 and T3.

This approach was further tested on longer sequences from the *PETS* workshops consisting of more than 1000 frames and the fitting accuracy was not as high as we obtained for shorter sequences. The low fitting accuracy does not allow us to make stronger inferences that we could make when we get significantly higher accuracy. This implies that this approach has a limitation when applied over longer and complex sequences where the tracking system commits more errors which are out of scope of the Scene Complexities considered here.

VI. Conclusions and Future Work

The main contribution of this work is the development of an evaluation framework that attempts to provide qualitative information about the tracker failure. We have motivated our choices of the different complexity measures and have modeled the output of the evaluation metrics using the Least Sum of Squares Regression model considering the interaction between the different complexity terms. We have tested our approach on the videos recorded with the Kinect consisting of reasonably difficult cases of the Scene Complexity. We have shown that this approach can successfully explain the output of a tracking algorithm and can also be used to differentiate between different algorithms. The amount of information that we obtain when performing such comparison is much more as compared to traditional evaluation methods.

However, the generalization ability of this model based approach is limited by the number of complexity measures that are employed, as seen in the r^2 values in Figure 3. If more complexities such as Camera Motion, Shadow Detection etc are added, the model will then be strong enough to be able to explain the entire variability in the metrics. Furthermore, to overcome the disability of this model to explain longer sequences a cascade based approach could be employed which can be implemented by splitting the longer sequence into several sub-sequences and creating a model for each sub-sequence. The use of only two tracking algorithms does not yet allow us to give a general conclusion about this approach. We intend to address these issues in the future.

References

[1] F. Bashir and F. Porikli. Performance Evaluation of Object Detection and Tracking Systems. In IEEE PETS, June 2006.

[2] K.Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. J. Image Video Process, January 2008.

[3] L.M.Brown, A. W. Senior, Y. li Tian, J. Connell. Performance evaluation of surveillance systems under varying conditions. In Proceedings of IEEE PETS Workshop, pages 18, 2005.

[4] D. Chu and A. Smeulders. Thirteen hard cases in visual tracking. In AVSS 2010, pages 103 110, 29 2010-sept. 1 2010.

[5] E. Corvee and F. Bremond. Haar like and LBP based features for face, head and people detection in video sequences. In ICVS11, page 10, Sophia Antipolis, France, Sept. 2011.

[6] A. Nghiem, F. Bremond, M. Thonnat, and R. Ma. A new evaluation approach for video processing algorithms. In Motion and Video Computing, 2007. WMVC 07.

[7] R. Rosenholtz, Y. Li, and L. Nakano. Measuring visual clutter. Journal of Vision, 7(2), Aug. 2007.

[8] H. Wu, A. C. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. IEEE Trans. Pattern Anal. Mach. Intell., 32(8):14431458, Aug. 2010.

[9] J. Yao and J. marc Odobez. Fast human detection from videos using covariance features. In 8th ECCV-VS workshop, 2008.

[10] D. P. Chau, F. Bremond, M. Thonnat, and E. Corvee. Robust mobile object tracking based on multiple feature similarity and trajectory filtering. In VISAPP11, pages 569574, 2011.