Learning From Disagreements: Discriminative Performance Evaluation

Christina Pavlopoulou David Martin Stella Yu Hao Jiang
Computer Science Department
Boston College
Chestnut Hill, MA 02467
{pavlo,dmartin,syu,hjiang}@cs.bc.edu

Abstract

Selecting test cases in order to evaluate computer vision methods is important, albeit has not been addressed before. If the methods are evaluated on examples on which they perform very well or very poorly then no reliable conclusions can be made regarding the superiority of one method versus the others. In this paper we put forth the idea that algorithms should be evaluated on test cases they disagree most. We present a simple method which identifies the test cases that should be taken into account when comparing two algorithms and at the same time assesses the statistical significance of the differences in performance. We employ our methodology to compare two object detection algorithms and demonstrate its usefulness in enhancing the differences between the methods.

1 Introduction

Evaluation of computer vision algorithms is challenging and in the past few years the computer vision community has seen an increasing effort in establishing protocols for rigorous quantitative comparisons among various methods. Research so far has been geared towards investigating measures of performance and creating databases of annotated images and videos to serve as ground truth. While undeniably these are important components of an evaluation scheme, no attention so far has been paid to the selection process of the individual test cases; yet it is an important problem.

Because of the richness of the visual stimuli and the difficulty in obtaining ground-truth annotations, only a small set of test-cases may be employed at a time and currently there are no rules of thumb about what constitutes "good" evaluation test cases. The images or videos in a test set are often dependent since they are collected from the same individuals and from the same locations. As a result, an algorithm is likely to perform similarly in many test cases and such a behavior makes comparable studies unreliable. Additionally, if we employ test cases where all the methods perform very well or very poorly, then any measure will fail to robustly characterize performance differences.

A good evaluation test set should consist of representative examples that are neither too easy nor too difficult for the methods under evaluation. Such selection cannot be performed by users reliably; the level of difficulty of an example depends on the algorithm and not on the human visual perception. Instead, we can use the methods themselves to assess whether an image or video is a good test case. By choosing test cases on which the methods mostly disagree, we focus on those aspects of performance that really differentiate the methods and we can draw more reliable conclusions regarding the superiority of a method versus another.

Figure 1 illustrates this point. Assuming that we order the test cases based on the performance of one method, then the leftmost and rightmost examples will be the ones for which the methods perform similarly (either very well or very poorly) and hence should be excluded from the study. In essence, we define a window through which differences in performance are enhanced. As computer vision algorithms improve, we expect this window to slide towards the right side of the graph. Previously difficult cases will be more successfully handled by new state-of-the-art methods and previously challenging cases will become increasingly easy.

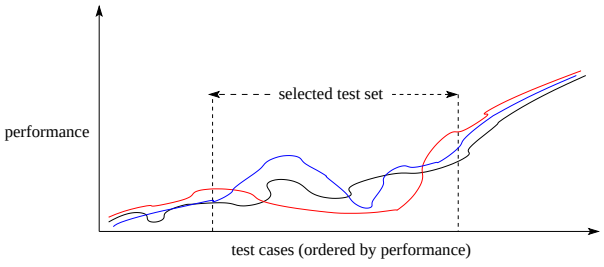


Figure 1: The test cases selected should be ones for which the algorithms disagree.

To computationally assess which examples should be removed from the test set, we propose a simple method which

at the same time assesses the statistical significance of the differences in performance. We successively remove test cases from the test set, so that cases for which the algorithms perform similarly are removed first. After each removal, a statistical significance test is performed and the p-value of accepting that the methods are indeed different is computed. When the p-value permanently falls below the desired threshold and the differences among the outputs of the methods for the examples removed are small, then we conclude that the difference in performance is significant.

Our methodology could be applied as is to a variety of computer vision problems including segmentation and tracking. In this paper we employ it to evaluate two representative object detection algorithms. Well-known precision-recall and average measures fail to give a confident answer regarding the superiority of an algorithm versus the other for the particular test set employed. By looking at the examples the two methods mostly differ in, we gain better insight in understanding the caveats of the algorithms as well as evaluate more confidently their quality.

2 Previous Work

Performance evaluation of object detection and tracking is a challenging problem and we could distinguish two lines of thinking when it comes to designing evaluation measures.

The first, relies on precision and recall values obtained by comparing the result of a method against ground truth data. Precision measures how faithfully the results match the ground truth whereas recall penalizes the false positives produced. The ideal method has very high precision and very high recall, that is, it produces the desired output with no false positive detections. Precision-recall curves like the ones of Figure 2 have been proposed by various researchers for evaluating motion, tracking, and detection algorithms (refer to [7, 1, 5, 13] and references therein).

Precision-Recall curves may not offer a definite answer as to whether a method A is better than method B since it is often the case that different methods perform well on different types of images. To this end, average measures like the ones discussed in [10] are useful because they allow easy comparisons at a glance. Their more accurate interpretation though requires statistical significance testing [9]. Significance assessment though relies on many assumptions and may not offer conclusive evidence regarding the superiority of an algorithm over another.

Obtaining ground truth data is a laborious process. For problems like segmentation and object recognition ground truth data are more widely available ([8] and [11]). Tracking typically requires annotation of each frame and is more difficult to obtain. The PETS workshop and the PASCAL challenge offer a range of datasets sometimes accompanied with ground truth that can be employed for evaluation of object detection and tracking systems.

3 Method

Our goal is to select test cases in order to reliably compare methods A and B. The difficulty in this task lies in that we can easily get a degenerate solution consisting of the single test case where A and B differ most. To avoid this situation, we impose the additional constraint that for the test cases removed the performance of methods A and B should be below a given threshold. Note that this constraint does not eliminate the empty set solution, that is, all the test cases should be removed. This is desirable though, since it could be the case that the particular test set is too easy for the methods and hence should not be employed.

We will address this question in the context of statistical significance testing [2] and we will seek to find whether it is possible to remove examples in order to establish statistical significance between methods A and B. Ultimately, the goal is to establish whether the differences between two methods are important. If the methods are evaluated on examples where they mostly perform similarly, this question cannot be answered reliably; the power of a statistical test is diluted.

Our algorithm is shown in Figure 3. The inputs required are the results of two methods A and B on a common set of examples S, as well as a similarity measure between the performances of A and B on a single test case. Such a similarity measure can be computed either by directly comparing the outputs of the methods or by measuring the compliance of the result of each algorithm with ground truth.

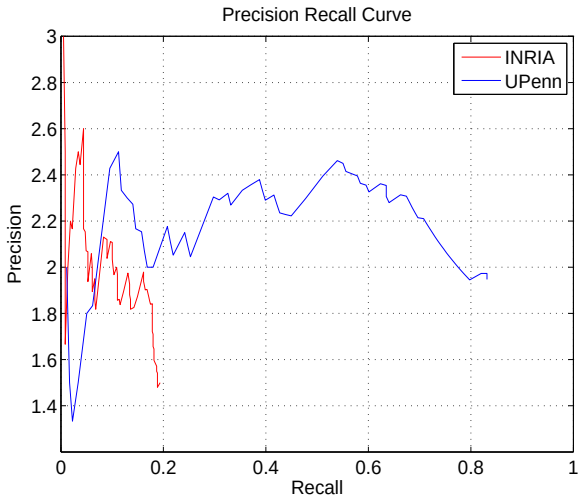
We sequentially remove test cases from the test set so that cases for which the algorithms perform most similarly are removed first. This is done by employing a set of increasingly higher thresholds `all_t`, and removing examples whose similarity falls below the given threshold. For each threshold, we compute the p-value of a statistical significance test for the remaining results. In the end the algorithm computes a set of p-values for the corresponding thresholds.

To decide whether the results of A and B are indeed different we check whether the following statements are true.

1. There is a performance similarity threshold t_0 for which the corresponding p-value,p_0, falls below a certain confidence level α (usually 0.05).

2. p-values corresponding to similarity thresholds $t > t_0$ remain below the confidence level α.

3. t_0 has to be small. That is, we should allow removal of test cases for which the difference between the performances of the algorithms are small enough.



(a)

Method	Mean Det. Acc.
INRIA	0.3486
UPenn	0.2728

(b)

Figure 2: **Known Evaluation Criteria:** (a) precision-recall curves (b) mean values using the Frame Detection Accuracy measure of [10].

It is not enough to simply find some threshold which gives rise to a low p-value; the calculation of this value should be robust. Condition 2 essentially ensures that the test cases removed are indeed outliers while significance test can be reliably performed.

To assess whether a threshold t_0 is small, we normalize all thresholds in `t_all` so that they sum up to 1. This way what we consider "small", does not depend on the difference range between the particular algorithms.

Other computational techniques could be employed at this stage. For example, one could use RANSAC [4] to find the outliers among the differences in the performance measures of the two methods. These outliers should be the test cases to evaluate the methods on. We could extend the current approach to compare more than two methods at a time by using ANOVA [6] instead of t-test.

4 Case Study: Object Detection

Our goal in this section is not to provide a rigorous evaluation between the methods selected, but rather demonstrate how our methodology can be employed to establish reliably the performance superiority of a method vs. another for a given set of images.

We compare two representative object detection methods whose source code is available on the web. The first method, to which we will refer to as "INRIA" [3], employs gradient orientation histogram features and a linear SVM classifier to determine the existence of people in a scene. The second method [12], which we will refer to as "UPenn", uses a codebook constructed from the silhouettes of a variety of pedestrians and tries to match them against candidate pedestrians in a new scene. Shape context is used to represent the silhouettes and the hypotheses are generated using the edge map and the segmentation of the image. We used the training data sets that the authors provided with their code and tested on a subset of the annotated INRIA-person database which was also part of the PASCAL 2005 challenge. The subset consisted of 78 images randomly selected.

Table 2 shows common performance measures that can be used to describe the results of the two algorithms. The calculation of the precision-recall curves requires assessing whether an image area identified by an algorithm indeed corresponds to a person. A detection was considered a true positive if the area of the intersection of the estimated region with the ground truth was more than 50% of the area of the union. The closer both precision and recall are to 1, the better an algorithm is and thus the topmost curve in a precision-recall graph corresponds to the best performance. For the methods at hand such conclusion is not possible since in some cases "INRIA" performs better and in some others "UPenn".

To compute a single quantity characterizing the performance of a method, we can combine precision and recall in a single measure, and subsequently compute the mean over the entire test set. To this end we used the Frame Detection Accuracy (FDA) measure described in [10], and defined as follows:

$$FDA = \frac{\text{Overlap Ratio}}{\frac{N_G+N_D}{2}} \quad (1)$$

```
function discriminative_evaluation(S_A, S_B)

% S_A: results obtained from method A on testset S
% S_B: results obtained from method B on testset S

all_p = {}
for t in all_t
    S_At = {I_A: d(I_A,I_B) < t}       % difference in performance < t
    S_Bt = {I_B: d(I_A,I_B) < t}
    S_A = S_A - S_At                   % remove results from results of A
    S_B = S_B - S_Bt                   % remove results from results of B
    p = assess_significance(S_A, S_B)  % compute p-value for reduced result set
    add p in all_p
 end

% decision procedure
if (p_0 < 0.05) & (t_0 small) & (p < 0.05 for t > t_0)
   S_A and S_B different
```

Figure 3: Our algorithm for removing test cases on which methods A and B perform similarly as well as assessing the significance of the difference between the methods' performances.

where N_G is the number of ground truth objects and N_D the number of detected objects. The overlap ratio is computed as follows:

$$\text{Overlap Ratio} = \sum_{i=1}^{N_{mapped}} \frac{|G_i \cap D_i|}{|G_i \cup D_i|} \qquad (2)$$

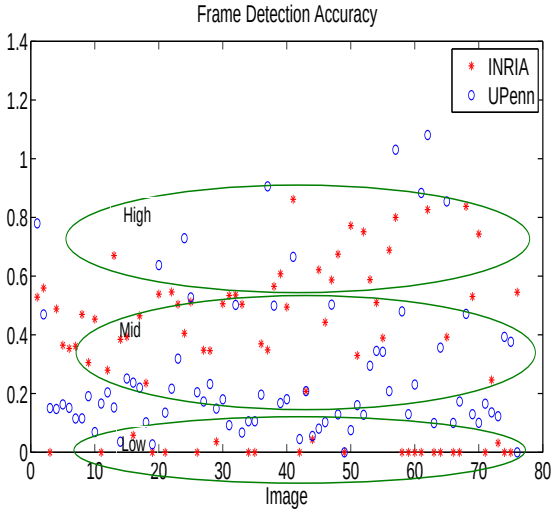
where G_i denotes the i-th ground truth object and D_i the i-th detected object. The calculation of the overlap ratio requires to find a mapping between the detected and the ground truth objects in an image. For that, we employed a simple greedy procedure: to each of the detected objects we assigned the ground truth object with the largest area overlap.

Figure 2 shows the mean FDA's for the methods evaluated. The mean of the "INRIA" method is higher than the mean of "UPenn", but it is not clear whether such a difference is important. By using a t-test on the detection accuracies for the two methods we do not obtain statistical significance for confidence level $\alpha = 0.05$; in fact $p = 0.0735 > 0.05$.

However, we can gain valuable insight regarding the performance of the algorithms by looking at the detection accuracies for the individual images. Such a plot is shown in Figure 4. The red stars belong to performance measures of "INRIA" and the blue circles to those of "UPenn". The plot immediately leads to interesting observations that cannot be so readily inferred from the measures of Figure 2: the "INRIA" algorithm has a tendency to underestimate the number of objects detected in an image and its performance is either really good or really bad. On the other hand the performance of the "UPenn" algorithm lies mostly in the mid level area due to its tendency to produce many false positives. The images in this table present some representative results for the different performance levels. By qualitatively assessing the plot, one could argue that the performance of the "INRIA" algorithm is indeed better from the "UPenn". However, how can we establish that, since common statistical significance criteria argue against it?

Our method proceeds by iteratively removing data points for which the algorithms perform increasingly less similarly. Similarity is assessed in two steps. First, the output of each method is compared against the ground truth using the FDA measure of Equation 1. Second, the absolute value of the difference between the two FDA measures is computed. At each iteration, images for which the similarity between the outputs of the two methods lies below a given threshold are removes The plots at the second row of Figure 5 illustrate how the removal of points affects the configuration of performance values. The images below each plot are representative examples of data points removed. For small thresholds, the images removed are ones for which the algorithms perform very similarly. The specific case illustrated is an image where both algorithms entirely fail: the "INRIA" method misses the person, whereas the "UPenn" produces too many false positives. As the threshold increases the output of the algorithms become less and less similar.

For the methods at hand, a small threshold is sufficient for achieving statistical significance, as shown in the topmost plot of Figure 5. The results for which the performance similarity is less than 0.1 are visually very similar

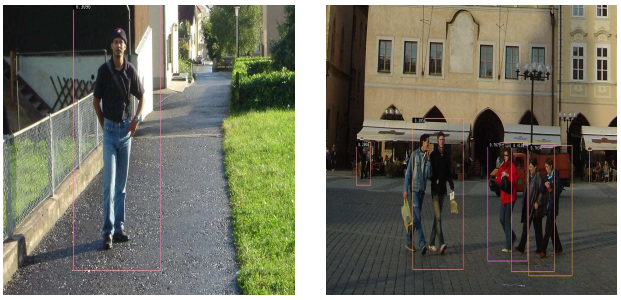


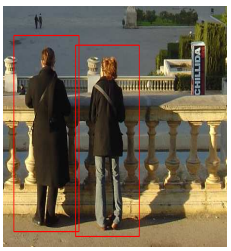
INRIA UPenn

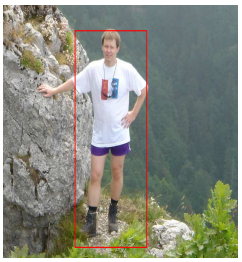
High

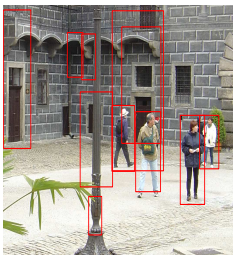
Med

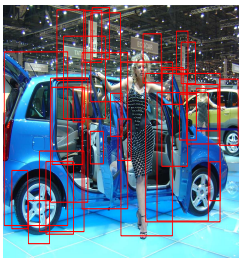
Low











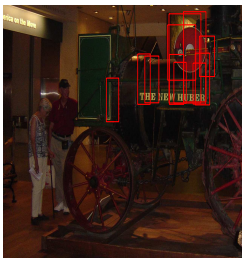


Figure 4: Performance levels for the object detection algorithms examined. The top plot shows the detection accuracy for each image for the two algorithms. The ellipses outline images for which the objects are detected with high accuracy, medium accuracy and low accuracy. Example results of each performance category for both methods are also shown.

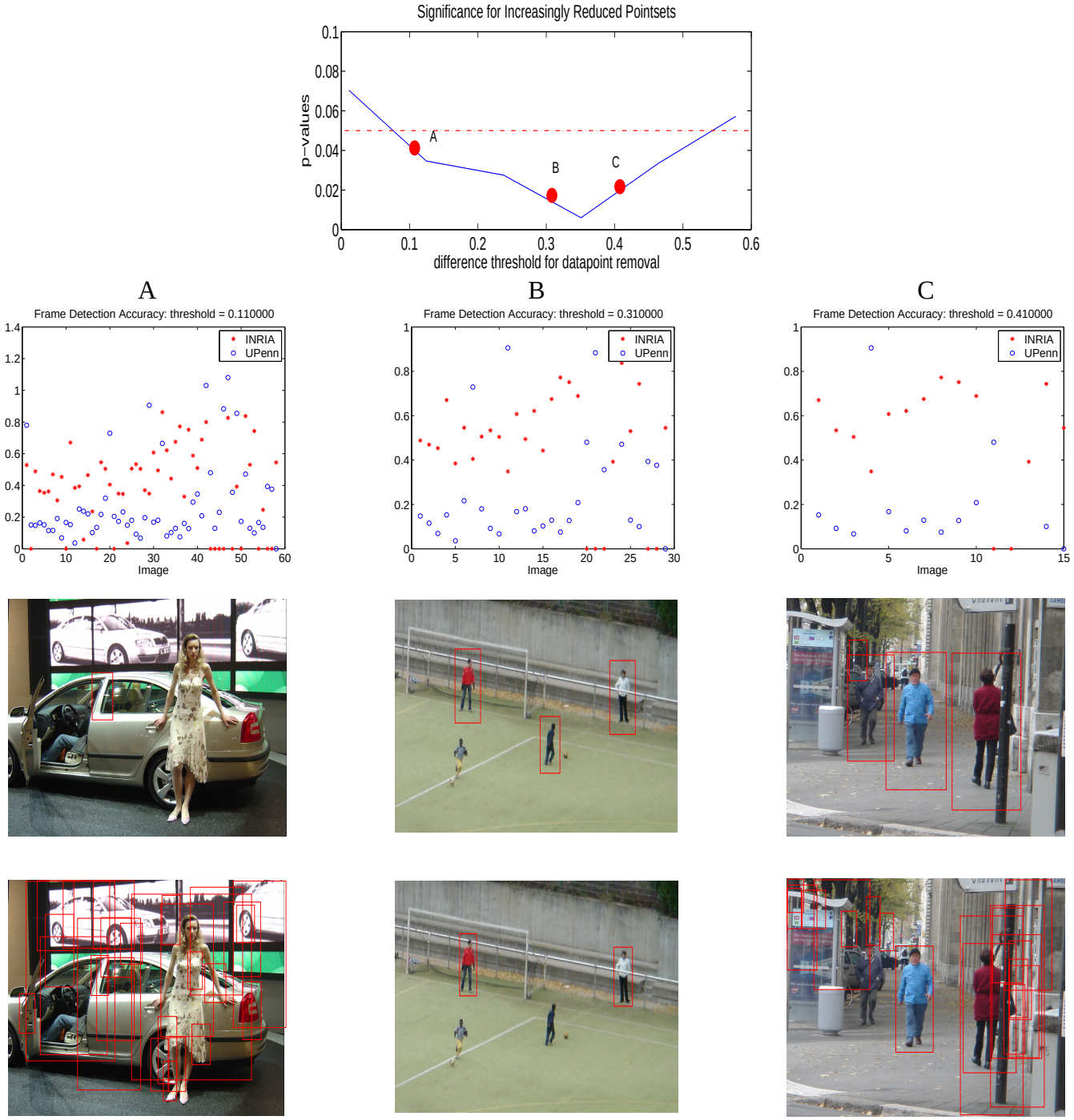


Figure 5: **First Row:** The plot at the top of the figure was produced by successively removing the images for which the algorithms performed similarly and subsequently performing a t-test on the remaining data. For example, point A on the graph was produced by removing the images for which the difference between the performances was less than 0.11. **Second Row:** The plots A, B and C show the configuration of the detection accuracies when increasingly less similar performances are removed. The points A, B and C on the topmost plot were obtained by using t-test on the corresponding performance configurations. **Bottom Rows:** Examples of images removed in order to obtain configurations A, B and C. The top images show the detection results obtained with the INRIA algorithm while the bottom detections were obtained with the UPenn method.

and thus the particular threshold is acceptable for this case. Furthermore, the p-value falls beneath 0.05 and remains there for higher thresholds. This way, one can numerically establish the qualitative observation that the "INRIA" algorithm performs better on average for the given test set.

5 Conclusions and Future Work

Despite advances in evaluation measures and ground truth datasets, the problem of what constitutes a good test case has not been addressed in the past. In this paper we have advocated, that algorithms should be evaluated on test cases they disagree most and we have provided a methodology of identifying those cases and assessing the statistical significance between performance differences.

Central to our methodology has been the similarity between the outputs of the methods under evaluation. In the work presented, the computation of similarity required ground-truth data. However, this need not be the case; one could compare directly the outputs of the methods. Such an approach might not be as reliable as when ground truth is employed, however it can provide guidance as to what test cases should be annotated. For problems like tracking and surveillance, where annotations are particularly laborious and time-consuming, knowing what frames to annotate, without affecting the reliability of an evaluation method, would be very beneficial.

References

[1] F. Bashir and F. Porikli. Performance Evaluation of Object Detection and Tracking Systems. In *9th IEEE International Workshop on PETS*, 2006.

[2] G. Casella and R. Berger. *Statistical Inference*. Duxbury Press, 2nd edition, 2001.

[3] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2005.

[4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[5] D. Gray, S. Brennan, and H. Tao. Evaluating Appearance Models for Recognition, Reacquisition and Tracking. In *10th IEEE International Workshop on PETS*, 2007.

[6] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 1998.

[7] N. Lazarevic-McManus, J. Renno, D. Makris, and G. A. Jones. Designing Evaluation Methodologies: The Case of Motion Detection. In *9th IEEE International Workshop on PETS*, 2006.

[8] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[9] M. Pound, A. Naeem, A. French, and T. Primdore. Quantitative and Qualitative Evaluation of Visual Tracking Algorithms Using Statistical Tests. In *10th IEEE International Workshop on PETS*, 2007.

[10] R. Kasturi and D. Goldgof and P. Soundararajan and V. Manohar and J. Garofolo and M. Boonstra and V. Korzhova and J. Zhang. Framework for Performance Evaluation of Vace, Text and Vehicle Detection and Tracking in Video: Data, Metrics and Protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2), 2009.

[11] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[12] L. Wang, Shi J, G. Song, and I-F. Shen. Object Detection Combining Recognition and Segmentation. In *8th Asian Conference on Computer Vision (ACCV)*, 2007.

[13] F. Yin, D. Makris, and S. Velastin. Performance Evaluation of Object Tracking Algorithms. In *10th IEEE International Workshop on PETS*, 2007.