Joint Modeling of Algorithm Behavior and Image Quality for Algorithm Performance Prediction

Apurva Bedagkar-Gala
avbedagk@mail.uh.edu
Shishir K. Shah
shah@cs.uh.edu

Quantitative Imaging Laboratory
Computer Science Department
University of Houston
Houston, TX 77204-3010, USA

Abstract

In this paper, we propose a framework for predicting the performance of a vision algorithm given the input image or video so as to maximize the algorithm's ability to provide the desired output. This is achieved by learning a relationship between the algorithm's behavior characteristics and the quality of the input. In general, each algorithm has the ability to provide a successful output dependent on the input characteristics. Unfortunately, the acceptable input variability for each algorithm is not known *a priori*. Nonetheless, this can be modeled through effective assessment of input image/video quality and the vision algorithm's response to the input. The key benefit of such a model is in its ability to predict or select one of many algorithms given a new input so that the probability of achieving the desired output is maximized without the need for executing all available algorithms. Our proposed framework models the performance prediction process as one that accounts for the algorithm's behavioral properties as well as the quality of the algorithm's input. The input image/video quality is measured by a combination of objective image quality measures, while the algorithm's behavioral properties are captured using a performance evaluation technique. The overall framework is realized for the problem of object tracking for use in video surveillance applications.

1 Introduction

Estimation of an algorithm's performance given a particular input image/video is difficult for a computer but quite easy for a human observer. Humans can assess the ability of an algorithm to generate a positive outcome for a given input based on small number of observations since they can extract high level cognitive information from input-output observations. Simulation of this process can lead to automation of algorithm performance assessment and thus eventual prediction of algorithm performance given an input image. In any computer vision system, predicting the performance of a vision algorithm can prove valuable in optimizing the overall success of the application. It is well understood that an algorithm's probability of success or failure is dependent on the behavioral properties of the algorithm, which, in turn depends on the characteristics of the input that the algorithm receives. Thus, prediction of algorithm performance given a particular input relies on the ability to understand the

BMVC 2010 doi:10.5244/C.24.31

relationship between the algorithm's behavioral properties and input image/video characteristics. Such understanding can facilitate applications like optimal algorithm selection or optimal parameter selection that would maximize the probability of successfully completing a particular task.

Algorithm performance evaluation is basically the study of an algorithm's behavior under different input characteristics [11], wherein the algorithm's behavioral properties are learned by understanding the ability of the algorithm to complete its objective under varying inputs. Analytical examination can be used to evaluate the algorithm's performance [6]. Performance analysis of any algorithm includes a range of measures like processing quality, stability, time and computational complexity. In this paper, we focus on the quality of task performance, which is, how well the target task is performed based on the input image/video. In addressing the problem of algorithm selection, Shah [16] has presented a probabilistic framework for selection of a segmentation algorithm based on the information content of an input image. Toyoma *et al*. [17] have presented an approach that allows for selection of an appropriate model for background maintenance based on intermediate results. But here the selection process is not based on image quality but rather on processing results from an intermediate algorithm stage.

In this paper we present a generalized framework and a prototype system that is designed for optimal prediction of vision algorithm's performance given the inputs image/video quality and its application to an optimal algorithm selection process, specifically for the problem of object tracking. This system can be considered an intelligent system that combines algorithm's input's quality with knowledge based prediction of algorithm performance. More specifically, a framework for algorithm performance prediction is presented. Performance evaluation is used to obtain knowledge about algorithm's behavioral properties and intelligent system design is used to apply that knowledge. The quality of each frame is expressed as a function or combination of functions of image features that represent degradations present in the video frames. Algorithm's performance is characterized by performance metrics that capture its ability to provide a desired result.

The rest of the paper is organized as follows; Section 2 introduces the performance prediction framework and the learning and estimation phase for the model. Section 3 describes the image/video quality features that constitute the quality measure, and performance evaluation technique used to extract performance metrics. Results are discussed in section 4. Section 5 presents the conclusions and future directions.

2 Performance Prediction Framework

A general predictor learning framework for the proposed performance prediction system is shown in figure 1. The framework contains three main modules, *quality extractor*, *performance evaluator*, and *predictor*. The quality extractor is meant to quantify the degradations in the input image that affect the vision algorithm's performance. The performance evaluator, evaluates the algorithm's performance on a set of training input data with varying levels of image quality in order to capture its behavioral properties. In essence, the performance evaluator simulates the algorithm's perception of the input image. Finally, the predictor combines information from quality extractor and performance evaluator. It provides a mechanism to automatically acquire, store and utilize knowledge about algorithm's perception of image quality.

The framework is prototyped using 3 object tracking algorithms used in video surveil-

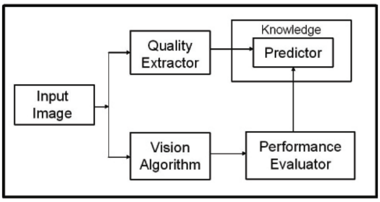


Figure 1: Performance Prediction Learning Framework.

lance applications. During the training phase, the input video is processed using each algorithm and performance evaluation on the results provides the performance metrics. The performance metric represents the quality of algorithm's performance associated with the input. Image quality measures are also calculated for every input frame in the video. Both image quality measure and performance measures are used to design the predictor. This allows for representation and capture of needed knowledge. When the predictor encounters a new input video, the knowledge learnt during training and the image quality are used to predict each algorithm's ability to succeed, without actual algorithm execution. The one expected to achieve maximum success is selected.

2.1 Learning and Prediction

Within the performance prediction framework, the predictor module predicts the vision algorithm's performance given each input image. In other words, it assigns each image a performance measure based on its quality. Thus, for every image I, image quality measure values extracted behave as a feature vector or image descriptor representing the image. The feature vector is given by $\mathbf{x} = (x_1, ..., x_n)$. Algorithm performance evaluation gives performance metrics that characterize the degree of success achieved by the algorithm in completing the task it was designed for, given a particular input image. This performance metric can be then treated as a class label, y, that denotes the performance class the input belongs to. Let there exist k different classes into which an input image can be classified. Thus, we can pose the problem of an algorithm's performance prediction as a classification problem. Predictor design can hence be thought of as the design of an optimal classifier. The predictor uses the image quality as the feature vector to assign the image to a particular performance class. For the rest of this paper, feature vector and class label will be used to represent image quality measure and performance measures, respectively.

The task of a classifier is to use the feature vector and assign new input data point to a class [7]. Classifiers can be expressed in terms of a set of discriminant functions like $g_i(\mathbf{x})$, where, $i = 1, ...k$. It assigns a feature vector $\mathbf{x}$ to a class y_i if:

$$g_i(\mathbf{x}) > g_j(\mathbf{x}), \tag{1}$$

for all $i \neq j$. Thus, the classifier computes k discriminant functions and selects the class with the largest discriminant.

3 Image Quality Measures, Performance Quality Measures, and Predictor Design

3.1 Image Quality Measures

The most prevalent definition of image quality measure is a measure that can quantify the extent and type of image degradation and correlates closely with human perception of image quality. There exists a large body of work that proposes different image quality measures that are consistent with this definition. The fundamental difficulty with all these measures is the assumption that the final consumers of these images/videos are humans, which is not necessarily the case in many vision applications. There have been some studies dedicated to task specific quality but are only restricted to linear estimators [3, 4]. Objective image quality measures that are consistent with perceptual image quality can reliably predict perceived quality. Many objective image quality measures have been proposed [2, 10, 21]. Most of these measures are reference quality measures, that is, they require a reference image/video to compute the quality measure. Such a reference image/video is usually not available in typical video processing applications. No-reference quality assessment metrics proposed over the last several years can be found in [14, 20, 22].

Since the application of interest in this paper is that of video surveillance and object tracking, we focus on metrics that capture the most common image/video degradations, which in turn could affect the ability of any tracking algorithm to successfully track objects in the video. Blurriness caused by sensor inadequacies or motion blur and blockiness caused by aliasing during compression are the most common degradations that affect real-world surveillance data. Besides these, errors during data transmission are also expected. Image quality measures that quantify these degradations, namely, signal activity and edge entropy are used. The signal activity measure proposed in [22] quantifies image blurriness and is computationally efficient. If the image is represented by $I(i,j)$, where i,j denotes a particular pixel position, then signal activity is given by:

$$A = \frac{A_h + A_v}{2} \tag{2}$$

where,

$$A_h = \frac{1}{m(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{n-1} |d_h(i,j)| \tag{3}$$

$$A_v = \frac{1}{(m-1)n} \sum_{i=1}^{m-1} \sum_{j=1}^{n} |d_v(i,j)| \tag{4}$$

where,

$$d_h = I(i,j+1) - I(i,j) \tag{5}$$

$$d_v = I(i+1,j) - I(i,j) \tag{6}$$

Image quality is space variant and content specific. To factor this into the signal activity measure, the quality is estimated locally. Signal activity is computed locally over fixed windows across the complete image and the overall image signal activity is then reported as the average of signal activity measures across all windows [22].

In video data there typically exists two layers of compression. One at the image level and the other at the video level. Typical image quality measures that quantify compression do so at the image level and are inadequate for quantifying compression artifacts in video. Thus, edge entropy is used as a quality measure, which not only accounts for compression artifacts but also quantifies degradations due to insufficient image resolution.

By definition, performance of a tracking algorithm depends heavily on the amount of correlation that exists between consecutive video frames. For a thorough assessment of a tracker's performance, we include a structural similarity measure that quantifies correlation between consecutive video frames. This gives us an idea of how the tracker's performance is affected by varying correlation between consecutive frames. The motivation behind this measure is based on the assumption that human visual system is adapted to extract structural information from the viewed scene and thus it can provide a good measure of perceived distortion. The structural similarity index metric (SSIM) proposed in [23] is used to measure the amount of correlation between consecutive frames, $f1$ and $f2$, such that:

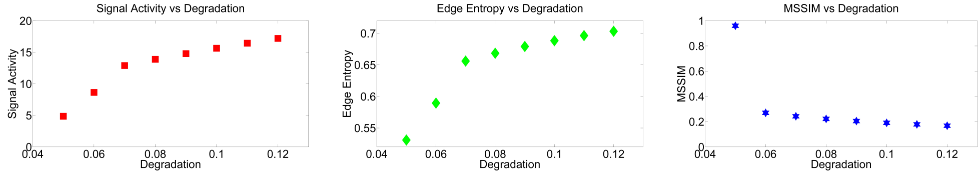
$$SSIM(f1,f2) = \frac{(2\mu_{f1}\mu_{f2}+C_1)(2\sigma_{f1f2}+C_2)}{(\mu_{f1}^2+\mu_{f2}^2+C_1)(\sigma_{f1}^2+\sigma_{f2}^2+C_2)} \tag{7}$$

where, μ_{f1}, μ_{f2} is the mean intensity of images $f1$ and $f2$, σ_{f1}, σ_{f2} is the standard deviation of intensities of respective images, σ_{f1f2} is the correlation coefficient between them, and, C_1 and C_2 are constants introduced to avoid instability. Default values for C_1, and C_2 are 0.01 and 0.03. The first component measures the closeness of mean luminance of the two frames and the second component combines the structural similarity and contrast similarity. SSIM is applied locally rather than globally to small windows across the image. The Mean Structural Similarity Index Metric (MSSIM) between the two images is given by:

$$MSSIM(f1,f2) = \frac{1}{W}\sum_{i=1}^{W} SSIM(f1_i, f2_i) \tag{8}$$

where, W is the number of local windows of the image.

Figure 2 shows the variations in each of the above described image quality measures as the degradation increases. It is clear that each of these measures captures the effect of the degradation on the image/video quality and hence are good objective quality measures to represent the input variability.



(a) (b) (c)

Figure 2: This figure shows the relationship between input degradation and the image quality measures used. The x axis shows the fraction of pixels affected by noise degradation. (a) Signal Activity Measure; (b) Edge Entropy Measure; (c) MSSIM.

3.2 Performance Quality Measures

Multiple Object Tracker performance evaluation is used to quantify a tracker's performance in terms of performance metrics. A systematic and objective performance evaluation of the tracker's characteristics proposed in [5] is used for this purpose. For every frame t, the tracker produces a set of hypotheses for a set of visible objects. The number of valid matches between the hypotheses and objects at time t are the true positives. Remaining hypotheses are considered as false positives. And, all remaining objects are misses or false negatives. We use F-measure [19] to quantify the tracker's correct responses and mistakes. It is thus a metric that corresponds to the trackers accuracy and it produces a bounded response.

$$\mathbf{F} = \frac{2c_t}{2c_t + m_t + fp_t} \tag{9}$$

where, c_t is the number of valid matches, m_t is the number of misses and fp_t is the number of false positives. The above measure is a fraction of the reliable results over the sum of reliable and unreliable results, where twice as much importance is attached to reliable results over unreliable ones. The metric output is then quantized into 4 bins, so in effect it describes the operating range of the tracker on its performance scale. The bin labels thus serve as the performance class label.

3.3 Predictor Design

Predictor design is essentially the estimation of discriminant functions for each class. The relationship between algorithms performance and input quality is multimodal in nature and highly non-linear. In general, we found that linear separability was not possible in the original feature space and hence non-linear subspace mapping was considered. We have used Kernel PCA for this problem. Kernel PCA, which is principal component analysis using kernel methods is used to transform the original feature space into the Hilbert space [15]. The non-linear mapping or the Hilbert space is defined by reproducing kernels as $\varphi_1(\mathbf{x})$. In this case, we have used the Gaussian kernel. Further, Support Vector Machines are trained in the transformed subspace and are used as the classifier. SVMs rely on preprocessing the data to represent patterns in much higher dimensions than the original feature space and then classify the data using hyperplanes. Thus, the discriminant function equation $g_i(\widehat{\mathbf{x}})$ is given by:

$$g_i(\widehat{\mathbf{x}}) = \mathbf{b}^t \widehat{\mathbf{x}} + b_{i0} \tag{10}$$

where,

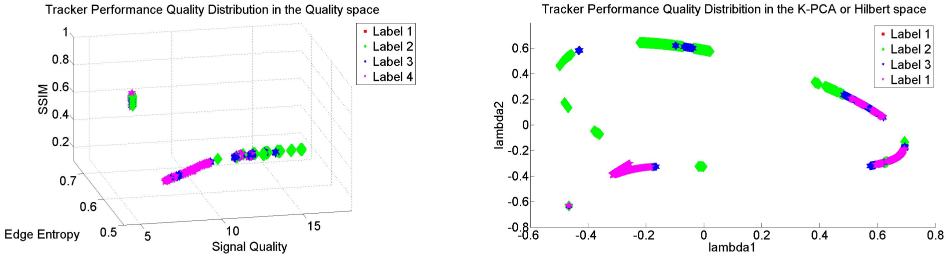
$$\widehat{\mathbf{x}} = \varphi_2(\varphi_1(\mathbf{x})) \tag{11}$$

$\mathbf{b}^t$ is a weight vector, b_{i0} is a bias factor and $i = 1, \ldots k$. In our case,φ_2 is Radial Basis Function Kernel and the input feature vector is given as:

$$\mathbf{x} = [SignalActivity, EdgeEntropy, MSSIM] \tag{12}$$

Figure 3 shows the effect of Kernel PCA on the 3D quality feature space.

In this paper, the predictor is in fact based on learning three distinct models, each emulating one of the 3 tracking algorithms used to prototype the framework. It uses the image quality measures extracted from each frame in the training dataset as the feature vectors. The performance class labels per frame are obtained by running each algorithm on every training video. This allows each model to learn the joint distribution between input quality and the corresponding algorithm's behavior.



(a) (b)

Figure 3: This figure shows the distribution of the video frames (data points) in 3-dimensional input quality Feature Space and 2-dimensional Hilbert Space: (a) Distribution in 3D feature space; (b) Distribution in 2D Hilbert space.

4 Experiments and Results

4.1 Algorithms and Training Data

To assess the feasibility of the proposed prediction framework, the performance prediction module is prototyped using 3 Object Tracking algorithms used in video surveillance applications. The algorithms used are Uniform Motion Connected Component tracking, Mean Shift tracking, and Particle filter tracking [9]. The background model used on which the tracker builds was proposed in [12]. In order to observe and learn the effect of the degradations in input data on the algorithm's performance, we use videos from the INRIA data set [1, 8]. The INRIA sequences have the associated ground truth available which enables us to quantify the tracking algorithms performance. The INRIA sequences have two sections of video clips, we use the first section filmed for the CAVIAR project. A subset of the first section was used for training. To incorporate the effect of typical degradations that occur during data transmission over surveillance networks, the videos are corrupted by salt and pepper noise of different densities using the 'imnoise' function in MATLAB's image processing toolbox. This gave us a set of 8 videos, which is about 4000 frames, with varying degrees of degradations as our training dataset. Three predictor were designed as discussed in section 3.

4.2 Testing Data

From the first section of INRIA sequences, 12 sequences amounting to about 6000 frames, were randomly selected from 3 basic scenarios, *Walking*, *Browsing*, and *People walking together* as the testing dataset. Each video in the test dataset is corrupted by varying degrees of salt and pepper noise to simulate data transmission errors.

4.3 Experiments

Since tracking algorithms are heavily dependent on temporal correlation between video frames, its performance is temporally variant. To factor this characteristic in the performance prediction process, the performance is estimated over a local temporal neighborhood. The scope of the neighborhood is defined as a temporal window representing the number of consecutive frames. The performance of the complete video is simply an average of the estimated performance metrics over all the windows in the video. Thus, the window size becomes an important parameter in estimation of the algorithms performance and by extension the predictors performance as well. Figure 4 shows the change in prediction accuracy due to varying temporal window sizes.

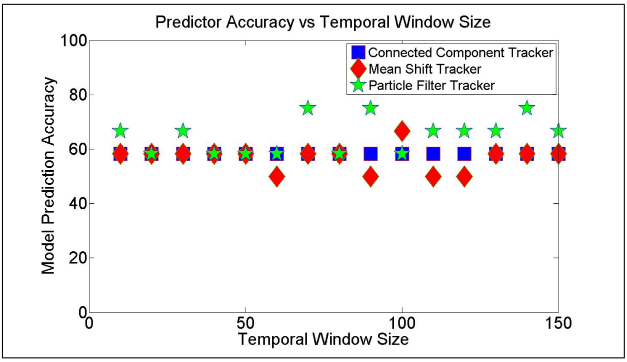


Figure 4: Effect of temporal window size on Predictor's Performance: x axis shows the different temporal window sizes while, the y axis shows the model's prediction accuracy with respect to each window size.

From the figure we can also see that the predictor behaves optimally for a window size of 70 and 140, resulting in the highest prediction accuracy for each of the algorithms.

Further, the utility of the predictor is evaluated by implementing an optimal algorithm selection system based on input quality in order to maximize the probability of successful object tracking given any input video. This is simply based on having each of the three predictors estimate the performance accuracy for a given input video and then selecting the tracker with the higher performance measure. Table 1 shows the performance of each individual tracker and the performance of the tracker selection based on prediction on the 12 test sequences. Labels CC, MS and PF represent the Connected Component Tracker, Mean Shift Tracker and Particle Filter Tracker, respectively. The tracker selection for a given video is based on the rank ordering of the performance labels predicted by each predictor. Rank 1, Rank 2 and Rank 3 are based on the ground truth performance of each of trackers. As can be seen from table 1, the predictor selects the best tracker 66.6% (8/12) of the times and one of the top two trackers 91.6% (11/12) of the times. This shows that the algorithm selector's overall accuracy exceeds any individual algorithm's accuracy. In 4 of the videos, the predictor does not pick the Rank 1 tracker, instead picks the Rank 2 tracker for 3 videos and Rank 3 tracker for 1 of the videos. Overall, the predictor outperforms any individual tracker and shows only a 8.33% chance of picking the worst tracker.

5 Conclusions and Discussion

In this paper, we have presented an algorithm performance prediction framework that models the relationship between a vision algorithm's behavior and characteristics of the input. The framework was prototyped using 3 object tracking algorithms for surveillance applications. Image quality measures that quantify image degradations were used as the input image quality features. Algorithm performance evaluation metrics are used to capture the ability

	Rank 1	Rank 2	Rank 3	Algorithm Selection
Video 1	MS	PF	CC	MS
Video 2	CC	MS	PF	CC
Video 3	PF	MS	CC	PF
Video 4	PF	MS	CC	PF
Video 5	PF	MS	CC	PF
Video 6	PF	MS	CC	PF
Video 7	PF	MS	CC	CC
Video 8	MS	CC	PF	MS
Video 9	CC	MS	PF	MS
Video 10	PF	CC	MS	CC
Video 11	PF	CC	MS	PF
Video 12	CC	MS	PF	MS

Table 1: Predictor based Tracker Selection (Window size = 80): The table shows the Predictor based Algorithm Selection compared to individual trackers performance.

of object tracking algorithms to successfully track objects in a given video. The framework was tested using a set of 12 test surveillance videos and its application to algorithm selection was demonstrated.

From the results, we found that the trained model for algorithm performance prediction is able to mimic the behavior of a vision algorithm relative to input quality. Such a performance prediction technique can be used to optimize the applicability of a vision algorithm to a particular application domain or serve to select an optimal algorithm from a bank of algorithms to optimize application performance. Other applications can be online algorithm switching for successful task completion or optimal parameter selection for given input in real-time. It also allows for the graceful performance degradation of a vision system.

In testing our framework, we used 3 video tracking algorithms that rely heavily on an underlying background model for effective object tracking. The background model used is dependent on the temporal consistency due to motion that exists between video frames to create a reliable background model. Image degradations that do not affect the temporal motion information in a video, do not affect the performance of any of the object tracking algorithms that rely on motion information. This observation is applied to enhance algorithm performance in [13]. Hence, even if degradations cause a reduction in the human perceptual quality of the image, its quality relative to the vision algorithm's perception does not reduce proportionally. The predictor exhibited the capability of learning this behavior. This concept is exploited for privacy preserving video surveillance applications in [18].

Future work in performance prediction of vision algorithms should deal with evaluating different image features that more closely quantify the image degradations and can capture the changes in the image due to content or scene changes. This will enable the capture and representation of the algorithm's response to scene content changes making the model more comprehensive. More realistic and efficient models for performance prediction also should be explored.

References

[1] Caviar: Context aware vision using image-based active recognition., 2004. URL http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[2] I. Avcibas, B. Sankur, and K. Sayood. Statistical evaluation of image quality measures. *Journal of Electronic Imaging*, 11:206–223, 2002.

[3] H.H. Barrett. Objective assessment of image quality: Effects of quantum noise and object variability. *Journal of Optical Society of America*, 7(7):1266–1278, 1990.

[4] H.H. Barrett, J.L. Denny, R.F. Wagner, and K.J. Myers. Objective assessment of image quality ii: Fisher information, fourier crosstalk anf figures of merit for task performance. *Journal of Optical Society of America*, 12(5):834–852, 1995.

[5] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008.

[6] P. Courtney and N.A. Thacker. Performance characterisation in computer vision: statistics in testing and design. pages 109–128, 2001.

[7] R. O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[8] R. B. Fisher. The pets04 surveillance ground-truth data sets. In *In Proc. 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–5, 2004.

[9] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, August 2002.

[10] S. Grgic', M. Grgic', and M. Mrak. Reliability of objective picture quality measures. *Journal of Electrical Engineering*, 55:3–10, 2004.

[11] R. Haralick. Performance characterization in computer vision. In *CAIP'93*, pages 1–9, 1993.

[12] L. Li, W. Huang, I. Y.H. Gu, and Q. Tian. Foreground object detection from videos containing complex background. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, 2003.

[13] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Workshop on Video-Oriented Object and Event Classification, ICCV 2009*, September 2009.

[14] E. Ong, W. Lin, Z. Lu, S. Yao, X. Yang, and L. Jiang. No-reference jpeg 2000 image quality metric. *International Conference on Machine E*, 14(1):234–778, 2003.

[15] B. Schölkopf, A. J. Smola, and K. Müller. Kernel principal component analysis. In *Artificial Neural Networks - ICANN '97*, pages 583–588, 1997.

[16] S. Shah. Performance modeling and algorithm characterization for robust image segmentation. *International Journal of Computer Vision*, 80(1):92–103, 2008.

[17] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. *Seventh International Conference on Computer Vision*, 1, 1999.

[18] M. Upmanyu, A.M. Namboodiri, K. Srinathan, and C.V. Jawahar. Efficient privacy preserving video surveillance. In *Proceedings of 2009 IEEE 12th International Conference on Computer Vision*, 2009.

[19] C. J. van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.

[20] Z. Wang, A. C. Bovik, and B. L. Evans. Blind measurement of blocking artifacts in images. In *In Proc. IEEE Int. Conf. Image Proc*, pages 981–984, 2000.

[21] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment using structural distortion measurement. In *in Proc. IEEE Int. Conf. Image Proc*, pages 65–68, 2002.

[22] Z. Wang, H. R. Sheikh, and A. C. Bovik. No-reference perceptual quality assessment of jpeg compressed images. In *Proceedings of IEEE 2002 International Conferencing on Image Processing*, pages 22–25, 2002.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibilty to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.