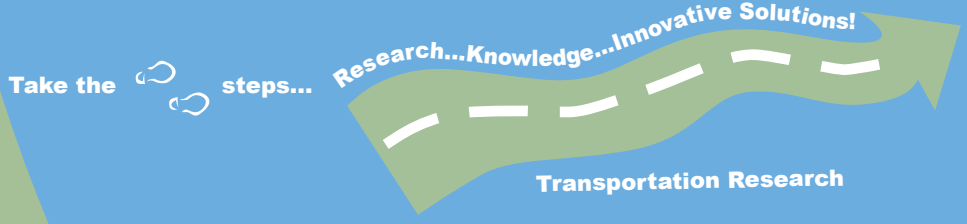
RESEARCH

2007-40

Freeway Network Traffic Detection and Monitoring Incidents



Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.	
MN/RC 2007-40			
4. Title and Subtitle Freeway Network Traffic Detection and Monitoring Incidents		5. Report Date October 2007 6.	
7. Author(s) A. Joshi, S. Atev, D. Fehr, A. Drenner, R. Bodor, O. Masoud, and N. Papanikolopoulos		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Computer Science and Engineering University of Minnesota 200 Union Street S.E. Minneapolis, Minnesota 55455		10. Project/Task/Work Unit No. 11. Contract (C) or Grant (G) No. (c) 81655 (wo) 98	
12. Sponsoring Organization Name and Address Minnesota Department of Transportation 395 John Ireland Boulevard St. Paul, Minnesota 55155		13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code	
15. Supplementary Notes http://www.lrrb.org/PDF/200740.pdf			

16. Abstract (Limit: 200 words)

We propose methods to distinguish between moving cast shadows and moving foreground objects in video sequences. Shadow detection is an important part of any surveillance system as it makes object shape recovery possible, as well as improves accuracy of other statistics collection systems. As most such systems assume video frames without shadows, shadows must be dealt with beforehand. We propose a multi-level shadow identification scheme that is generally applicable without restrictions on the number of light sources, illumination conditions, surface orientations, and object sizes. In the first level, we use a background segmentation technique to identify foreground regions that include moving shadows. In the second step, pixel-based decisions are made by comparing the current frame with the background model to distinguish between shadows and actual foreground. In the third step, this result is improved using blob-level reasoning that works on geometric constraints of identified shadow and foreground blobs. Results on various sequences under different illumination conditions show the success of the proposed approach. Second, we propose methods for physical placement of cameras in a site so as to make the most of the number of cameras available.

17. Document Analysis/Descriptors		18. Availability Statement	
Data collection, freeways, vision, tracking, shadows		No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161	
19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	47	

Freeway Network Traffic Detection and Monitoring Incidents

Final Report

Prepared by:

Ajay Joshi
Stefan Atev
Duc Fehr
Andrew Drenner
Robert Bodor
Osama Masoud
Nikolaos Papanikolopoulos

Artificial Intelligence, Robotics and Vision Laboratory
Department of Computer Science and Engineering
University of Minnesota

October 2007

Published by:

Minnesota Department of Transportation
Research Services Section
395 John Ireland Boulevard, MS 330
St Paul, MN 55155

ACKNOWLEDGEMENTS

This research has been supported by the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

We propose methods to distinguish between moving cast shadows and moving foreground objects in video sequences. Shadow detection is an important part of any surveillance system as it makes object shape recovery possible, as well as improves accuracy of other statistics collection systems. As most such systems assume video frames without shadows, shadows must be dealt with beforehand. We propose a multi-level shadow identification scheme that is generally applicable without restrictions on the number of light sources, illumination conditions, surface orientations, and object sizes. In the first level, we use a background segmentation technique to identify foreground regions that include moving shadows. In the second step, pixel-based decisions are made by comparing the current frame with the background model to distinguish between shadows and actual foreground. In the third step, this result is improved using blob-level reasoning that works on geometric constraints of identified shadow and foreground blobs. Results on various sequences under different illumination conditions show the success of the proposed approach.

Second, we propose methods for physical placement of cameras in a site so as to make the most of the number of cameras available. The ratio between the amount of information that can be collected by a camera and its cost is very high, which enables the use of cameras in almost every surveillance or inspection task. For instance, it is likely that there are hundreds to thousands of cameras in airport or highway settings. These cameras provide a vast amount of information. It would not be feasible for a group of human operators to simultaneously monitor and evaluate effectively or efficiently. Computer vision algorithms and software have to be developed to help the operators achieve their tasks. The effectiveness of these algorithms and software is heavily dependent upon a “good” view of the situation. A “good” view in turn is dependent upon the physical placement of the camera as well as the physical characteristics of the camera. Thus, it is advantageous to dedicate computational effort to determining optimal viewpoints and camera configurations.

CHAPTER 1
INTRODUCTION

The problem of moving shadow detection has had great interest in the computer vision community because of its relevance to visual tracking, object recognition, and many other important applications. One way to define the problem is to cast it as a classification problem in which image regions are classified as either foreground objects, background, or shadows cast by foreground objects. Despite many attempts, the problem remains largely unsolved due to several inherent challenges: (i) Dark regions are not necessarily shadow regions since foreground objects can be dark too; (ii) Self-shadows should not be classified as cast shadows since they are part of the foreground object; and (iii) A commonly used assumption that these shadows fall only on the ground plane is not valid to general scenes. In this paper, we address these challenges by proposing a shadow detection method which does not put any restrictions on the scene in terms of illumination conditions, geometry of the objects, and size and position of shadows. Results obtained using the proposed approach, in varied conditions, are very promising.

The report is organized as follows. In Chapter 2, we discuss various approaches proposed in the literature to deal with the problem of moving cast shadows. Chapter 3 describes our approach in detail. We present results on various scenes using our proposed method in Chapter 4. In Chapter 5, we introduce the problem of camera positioning and motivate its relevance. Chapter 6 discusses related previous work in the field and 7 gives a more formal explanation of the problem. Chapter 8 and 9 discuss extensions to the work by our research. Chapter 10 to 12 detail our research on new machine learning techniques applied to the problem of shadow detection. We introduce the learning framework, outline how our problem can be cast in a similar fashion, and show promising results. Chapter 13 concludes the report.

CHAPTER 2
RELATED WORK

There has been significant work done recently that deals with the problem of moving cast shadows. Reference [1] presents a comprehensive survey of all methods that deal with moving shadow identification. It details the requirements of shadow detection methods, identifies important related issues and makes a quantitative and qualitative comparison between different approaches in the literature. In [2], shadow detection is done using heuristic evaluation rules based on many parameters which are pixel-based as well as geometry-based. The authors assume a planar and textured background on which shadows are cast. They also assume that the light source is not a point source as they use the presence of penumbra for detection. Our objective was not to impose any such restrictions of planar or textured background, nor to assume a specific light source. Some methods use multiple cameras for shadow detection [3]. Shadows are separated based on the fact that they are on the ground plane, whereas foreground objects are not.

There has been an interest in using different color spaces to detect shadows. Reference [4], for example, uses normalized values of the *R*, *G*, and *B* channels, and shows that they produce better results than the raw color values. Their system relies on color and illumination changes to detect shadow regions. In [5], shadow detection is done in the HSV color space and automatic parameter selection is used to reduce prior scene-specific information necessary to detect shadows. A technique which uses color and brightness information to do background segmentation and shadow removal is outlined in [6]. A background model is maintained based on the mean and variance values of all color channels for each pixel. A new pixel is compared with the background model and a decision regarding shadow/ foreground is made. The method also suggests an automatic threshold selection procedure using histograms. In [7], the authors describe a technique which uses color and brightness values to differentiate shadows from foreground. They also deal with the problem of ghosts – regions which are detected as foreground but not associated with any moving object. However, the techniques based only on color and illumination are not effective when foreground color closely matches that of the background. In our work, we go a step further in pixel-based decisions by using edge magnitude and edge gradient direction cues along with color and brightness to separate foreground from background. The method in [8] models the values of shadowed pixels using a transformation matrix which indicates the amount of brightness change a pixel undergoes in the presence of a cast shadow. In case the background is not on a single plane, it is divided into flat regions so that a single matrix can be applied to each region. It does not use any other geometric constraints and is generally applicable. At a later stage, spatial processing is done using belief propagation to improve detection results. Another method which uses geometry to find shadowed regions is outlined in [9]. It produces height estimates of objects using their shadow positions and size by applying geometric reasoning. However, shadows need to be on the same plane so that the height estimates be valid. A sophisticated approach which works on multiple levels in a hierarchy is shown in [10]. Low-level processing is done at the first level and as we go higher up, processing controls parameters which change slower. This hierarchical approach takes care of both fast- and slow-changing factors in the scene. The authors describe an operating point module which has all the parameters stored at all times. However, geometric constraints which are applied make the algorithm applicable only to traffic sequences.

CHAPTER 3
APPROACH

In this work, we implement a three-stage shadow detection scheme. In the first step, background segmentation is done using the mixture of Gaussians technique from [11], as modified in [12]. The next step presents a parametric approach in which four parameters are computed for each pixel and a pixel-wise decision process separates shadows and foreground. In order to improve the results obtained in this step, further processing is done in the next step which works on the blob level to identify and correct misclassified regions. Both pixel-based and geometric cues are eventually used to make a shadow/ foreground decision. However, the geometric cues applied are general enough to allow their application to all commonly occurring scenes.

A. Step 1

This step implements a background update procedure and maintains a foreground mask, which the next two steps process. The learning rate of the background update is tuned depending on the type of sequence and the expected speed of motion. This foreground mask contains moving objects and their moving shadows. Static objects and static shadows are automatically separated and are not considered any further in the processing. This approach differs from a few other approaches which use statistical measures to do background segmentation and shadow removal simultaneously, i.e., the same measures are used to differentiate between background/foreground and foreground/shadow. An example of that is [5], which uses color and intensity measures for background segmentation. An advantage of our approach is that it uses a sophisticated background maintenance technique which takes care of static objects. Secondly, there are more parameters that differentiate shadows and foreground than shadows and background. Once we have a foreground mask, these parameters can be used effectively, without worrying about similarities between the measure for shadows and background, since background pixels can never be mistaken for shadow after this step.

B. Step 2

For effective shadow removal, we need to use features in which shadows differ from foreground. Such differences can lead to preliminary estimates for shadow/foreground separation. We experimented using different measures such as intensity, color, edge information, texture, and feature points, and found the following subset to be the most effective in shadow detection. Hence, for each pixel the following four parameters are computed by comparing the current frame with a constantly updated background model: a) Edge magnitude error (E_{mag}), b) Edge gradient direction error (E_{dir}), c) Intensity ratio (I_r) [5], and d) Color error (C_e) [5].

Edge magnitude error is the absolute difference between the current frame edge magnitude and background frame edge magnitude at each pixel. If the edge magnitudes at a pixel in the current frame and the background frame are m_1 and m_2 respectively, we then have

$$E_{mag} = \left| m_1 - m_2 \right|. \qquad (1)$$

Edge gradient direction images represent edge gradient direction (angle) for each pixel scaled between values of 0 and 255. Edge gradient direction error is the difference between gradient directions of current frame and background frame for each pixel. If d_1 and d_2 denote the gradient direction values for a pixel in current frame and background frame respectively, we then obtain:

$$E_{dir} = \min\left(\left|d_1 - d_2\right|, 255 - \left|d_1 - d_2\right|\right). \quad (2)$$

This gives the scaled angle between the edge gradient directions in the current frame and the background frame. If there are changes in edge magnitude, or if a new edge occurs at a pixel where there was no edge in the background model, or if an edge disappears, it is highly likely that the pixel belongs to a foreground object. Edge detection is carried out using simple forward difference in the horizontal and the vertical directions. Our experiments show that for our purposes, this works better than using any other edge detection operators like Sobel, Prewitt or others. Shadows do not significantly modify the edge gradient direction at any pixel. On the other hand, the presence of a foreground object will generally substantially modify the edge gradient direction at a pixel in the frame. These two edge measures are important cues in shadow detection. They work best along the edges of the foreground object while the other two measures of intensity and color work well in the central region of the object. These edge measures also work extremely well where foreground and background have significant difference in texture. For regions in which the foreground color matches that of the background, edge cues are the most reliable ones for shadow detection.

The Intensity ratio I_r can be easily explained using the color model in [5]. Given a pixel in the current frame, we project the point in RGB color space that represents the pixel's color on the line that connects the RGB space origin and the point representing the pixel background color according to the background model. The intensity ratio is calculated as the ratio of two distances: (a) the distance from the origin to the point projection, and (b) the distance from the origin to the background color point. Color error C_e is calculated as the angle between the line described above and the line that connects the origin and the point representing the current pixel color. Shadows show a lower intensity than background, while maintaining the background color. On the other hand, color change generally indicates the presence of a foreground object.

C. Probability mapping

We need to find the probability of a pixel being a shadow pixel based on the four parameters described above. Let these be represented by *A*, *B*, *C*, and *D* and the events that the pixel is a shadow pixel and foreground pixel be represented by *S* and *F*, respectively. Our goal is to make a decision whether a pixel is shadow or foreground. We use a maximum likelihood approach to make this decision by comparing the conditional probabilities $P(A,B,C,D|S)$ and $P(A,B,C,D|F)$. Assuming that these four parameters are independent, we get

$$P(A,B,C,D \mid S) = P(A|S)\cdots P(D|S). \quad (3)$$

We used a number of video frames along with their ground truth data and inspected the histograms of the edge magnitude error and the edge gradient direction error for shadow pixels and foreground pixels. E_{dir} histogram of shadow pixels, as in Figure 1, shows exponential behavior with significant peaks corresponding to values in degrees of 0, 45, 90, and 180. The

exponential curve decays fast initially, but has a long tail. The peaks are aberrations caused by dark regions, where edge gradient directions are highly quantized. Since we use the edge magnitude error, E_{mag}, as another measure apart from edge gradient direction error, the errors due to peaks do not lead to erroneous results. Our inspection also showed that E_{mag} exhibits similar statistical properties as E_{dir}, but without the peaks. In the case of foreground pixels, the histograms for E_{dir} and E_{mag} resembled a uniform distribution with E_{dir} showing similar peaks as mentioned above. The difference in the distributions is the basis for differentiating between shadows and foreground using these two features. Both these distributions are intuitively expected.

To model $P(E_{mag}|S)$ and $P(E_{dir}|S)$, we use the exponential functions in Equations (4) and (5). The variances of these exponentials (λ_1, λ_2) are parameters that can be tuned. For darker scenes, we expect the error caused by shadow to be lower compared to those for bright scenes. It follows that dark scenes will be better modeled using a lower variance for the exponential. In the equations, ω_1 and ω_2 are used for appropriate scaling.

The histograms computed for the intensity ratio measure (I_r) in shadow regions have sigmoid function-like shape as shown in Figure 2. Color error histograms show similar behavior except that shadow pixel frequency is high for small error values. These behaviors are modeled by Equations (6) and (7). β_1 and β_2 provide necessary shift in these equations. They depend on the strength of the shadows expected in the video. When stronger (darker) shadows are expected, β_1 is lower to account for a larger change in intensity due to shadow. Histograms of I_r and C_e were found to have a close to uniform distribution for foreground pixels.

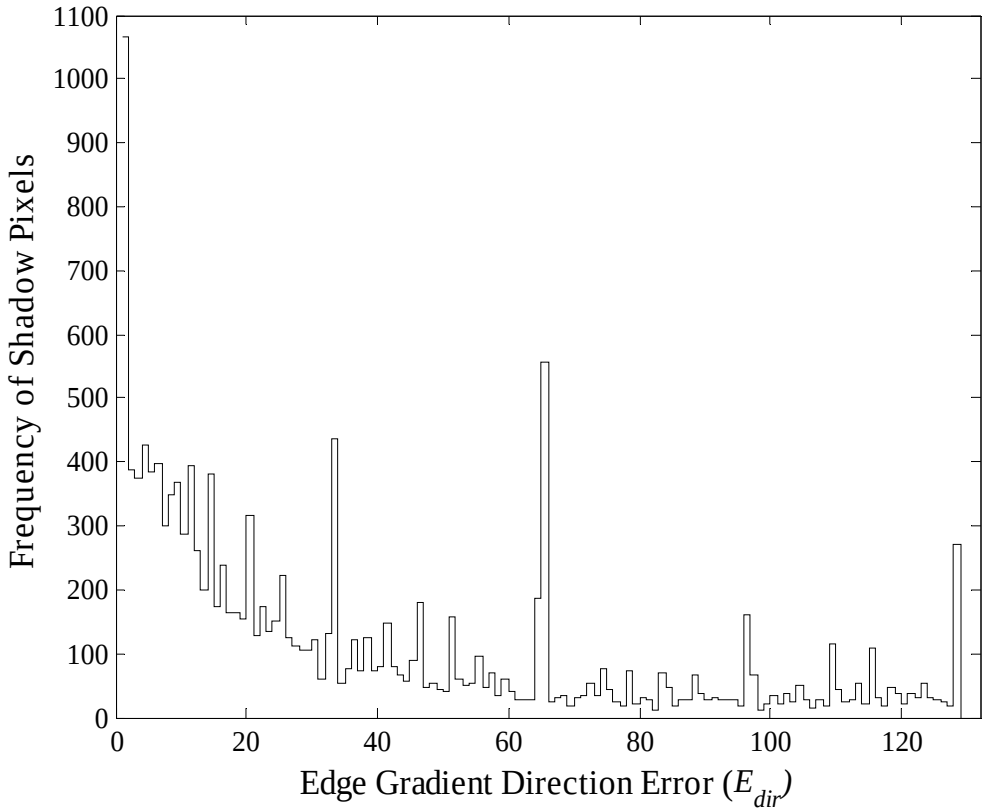


Figure 1: Histogram of E_{dir} values for shadow pixels.
Note the significant peaks corresponding to 0˚, 45˚, 90˚, and 180˚- aberrations caused due to dark regions of the image.

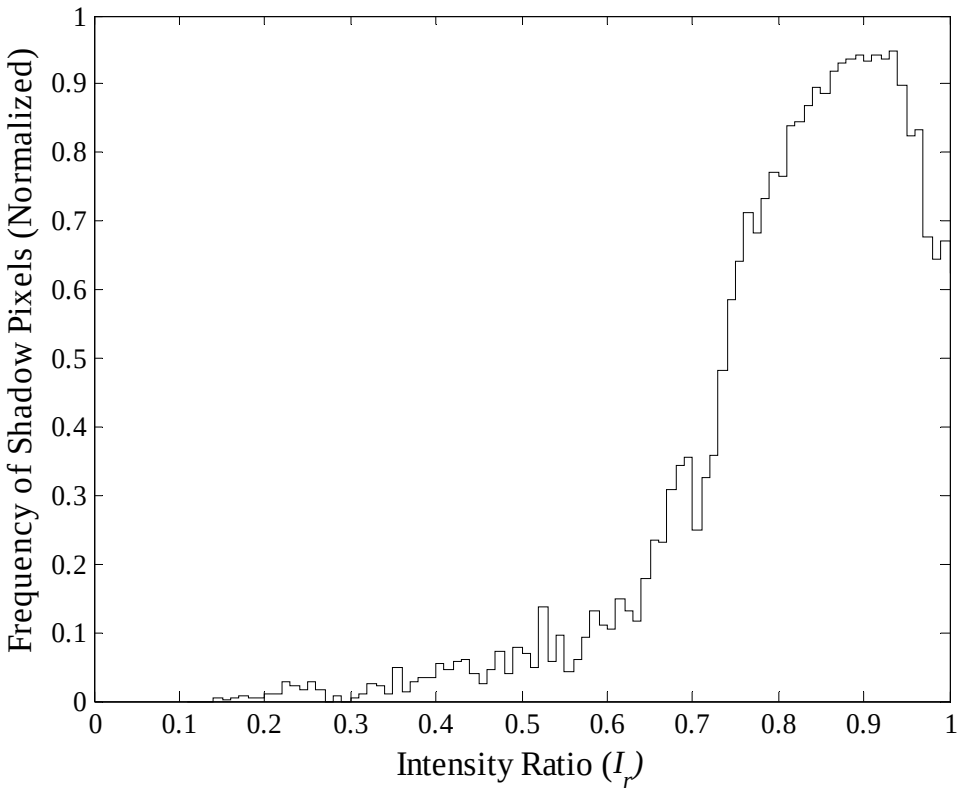


Figure 2: Normalized histogram of I_r values for shadow pixels.
This can be reasonably modeled by the sigmoid function in Equation (6).

$$P(E_{mag} \mid S) = (\omega_1 / \lambda_1) \exp(-E_{mag} / \lambda_1) \quad (4)$$

$$P(E_{dir} \mid S) = (\omega_2 / \lambda_2) \exp(-E_{dir} / \lambda_2) \quad (5)$$

$$P(I_r \mid S) = 1/(1 + \exp(-(I_r - \beta_1)/\sigma_1)) \quad (6)$$

$$P(C_e \mid S) = 1 - 1/(1 + \exp(-(C_e - \beta_2)/\sigma_2)). \quad (7)$$

Once we have probability maps using the above mapping for each parameter for every pixel, we blur the log-probability maps so as to account for the information of nearby pixels. This blurring is carried out for a box of 3x3 pixels around the given pixel. In case the conditional probabilities are around 0.5, blurring helps as it brings in neighborhood information to aid the decision making process. The probability maps are then multiplied together as in Equation (3). A final decision is made based on comparing the conditional shadow probability and conditional foreground probability thus obtained. Table I reports typical parameter values and issues for the above mapping.

Table 1: Typical parameter values for Step 2.

Parameter	Notes	Typical Values
λ_1	Lower for darker scenes – E_{mag} and E_{dir} decay fast for dark scenes	50 – 80
λ_2		30 – 50
σ_1	Lower for a steeper rise of sigmoid – when shadow and foreground intensity and color show distinct separation	0.10 – 0.15
σ_2		4 – 6
β_1	Control shift of the sigmoid functions. Significant tuning parameters based on shadow strength	0.4 – 0.6
β_2		0.85 – 0.9

D. Step 3

The second step is restricted to local pixels to make a decision as to whether a pixel is shadow or foreground. The farthest it goes is a 3x3 square of pixels around, which happens when log-probability maps are blurred. For good foreground object recovery, this is not enough. There are many regions of misclassification which make the object shape estimate erroneous. Thus, we need some technique to recover shape from frames obtained after step 2. Blob-level reasoning is a step further towards this objective. The processing is done at a higher level in order to identify connected components in an image and provide a better estimate of shadow positions.

At this stage, we have an image with shadow and foreground blobs marked by the previous step. Blob labeling is done along with other computations like blob area, perimeter, and the number of perimeter pixels that are neighbors to pixels of a blob of another type (shadow pixels neighboring foreground pixels and vice versa). In order to improve shadow detection accuracy, we propose to reason out misclassified blobs (e.g., flip shadow blob to foreground blob) based on the heuristic and metrics described below:

1) Blob area – the smaller the blob area, the easier it is to flip.
2) The ratio of the length in contact with another type blob to the length touching background– if this ratio is large for a blob, it is likely that the blob has been misclassified.
3) Whether flipping that blob connects two previously unconnected blobs – in case it does, it is less likely to have been misclassified.

Table 2: Parameter description and relations for Step 3.

Parameter	Description	Notes
BA	Blob Area	In pixels
P_{fg}, P_{bg}	Percent of perimeter of blob which is foreground, background respectively	Ratio P_{fg}/P_{bg} has an upper threshold T_h
T_h	Upper threshold for ratio P_{fg}/P_{bg}	Depends on C_{td} and whether $BA<T$ or $BA\geq T$
C_{td}	True if flipping connects two unconnected blobs, false otherwise	Influences T_h – higher T_h if C_{td} is true
T	Threshold for BA	Influences T_h – higher T_h if $BA\geq T$

Table 2 indicates the relations between different parameters and how they combine to flip misclassified regions in the image.

The ratio of lengths mentioned earlier (P_{fg}/P_{bg}) indicates what dominates the surroundings of the current blob. When this ratio is high, say for a shadow blob, it means that it is surrounded to a large extent by a foreground blob. Depending on the blob size (BA), we have different thresholds for the ratio. If the ratio falls above the threshold (T_h), the blob is likely to be flipped into other type. If flipping the blob connects two previously unconnected regions, we say that it is less likely to be misclassification in the first place. For example, consider the shadow region between two vehicles on a highway. Since the region touches foreground on both sides, the ratio mentioned has a high value. The above reasoning makes such a flip improbable. Thresholds on the ratio are such that smaller blobs are easier to flip as compared to larger ones. Theoretically, smaller blobs are equally likely to be misclassified as larger ones, but keeping in mind our objective of best possible foreground shape detection, this approach improves detection quality. An example of reasoning out such a blob is shown in a later section.

Especially when dealing with traffic videos, this approach improves results drastically as we rarely deal with objects which have holes. A significant problem for shadow detection in traffic scenes are the windshields of cars. All the four parameters mentioned above are incapable of consistently classifying them correctly. However, when we use blob-level reasoning as described above, that problem is substantially reduced. As windshields are surrounded by foreground for a large length, even if they are detected as shadow in the second step, they are generally flipped in the last step.

CHAPTER 4
RESULTS

The video clips *Highway I, Highway II, Intelligent Room, Laboratory, Campus*, and associated ground truth data for sequence *Intelligent Room* are courtesy of the Computer Vision and Robotics Research Laboratory of UCSD.

An important result of using edge-based measures in shadow detection is the improvement in performance at locations where the scene has highly reflective surfaces. Color change occurs on reflective surfaces even if they are under shadow and makes our premise on C_e invalid. Color-based and intensity-based measures cannot effectively differentiate between such reflective surfaces under shadow, and foreground. However, since the surface contains similar edge gradients even under shadow, it is possible to classify it as a shadow rather than foreground. In Figure 3 below, blue regions indicate foreground, red indicates shadow, and white indicates background as detected by our algorithm. The figure shows that a substantial part of the reflection on the table is detected as shadow and not foreground even though color changes occur.

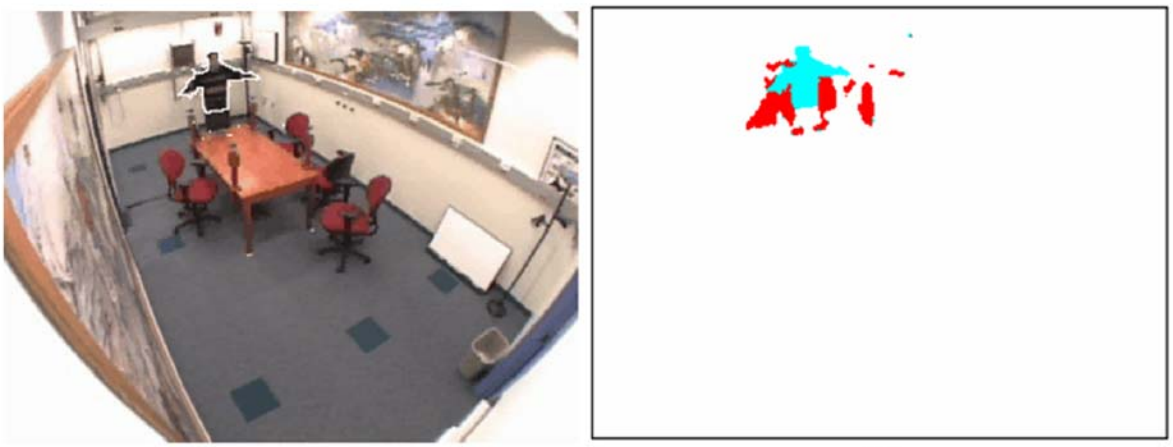


Figure 3: Result on a frame of the *Intelligent Room* sequence.
Note the shadow on table (reflective surface) (a) Original frame with a white contour of foreground as detected by the algorithm. (b) Detected shadow in red, foreground in blue, and background in white.

As mentioned earlier, the removal of holes results in cleaner images and better foreground object contour calculation. Figure 4 and Figure 5 show such a case. The figures on the left show the results of processing using only the first two steps of the algorithm. The ones on the right are the output of the final stage. In Figure 4(a), a number of foreground pixels are marked as shadow due to two reasons: the color similarity between the shirt of the person and the color of background, and the shadows of the person's hand on his shirt, which make the intensity similarity significant. Problems caused by indirect cast shadows are also reduced by the technique, as shown in Figure 4(b).

Windshields cause a significant problem in detecting the foreground correctly. Since windshields generally occur as dark regions, they are often misclassified as shadows. All the four parameters mentioned above cannot consistently classify windshields as foreground. Figure 5 shows such a case. Figure 5(b) shows how blob-level reasoning substantially reduces that problem.

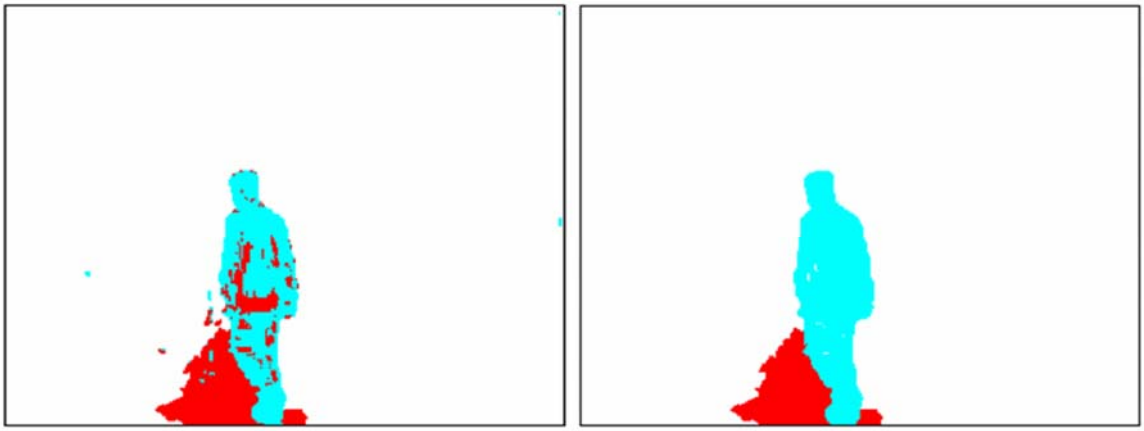


Figure 4: Results on a frame of sequence *Laboratory*.
Note how step three of the algorithm reduces problems due to intensity and color matching. (a) The output of the second step of our algorithm. (b) The output of the third step that shows the improvement.

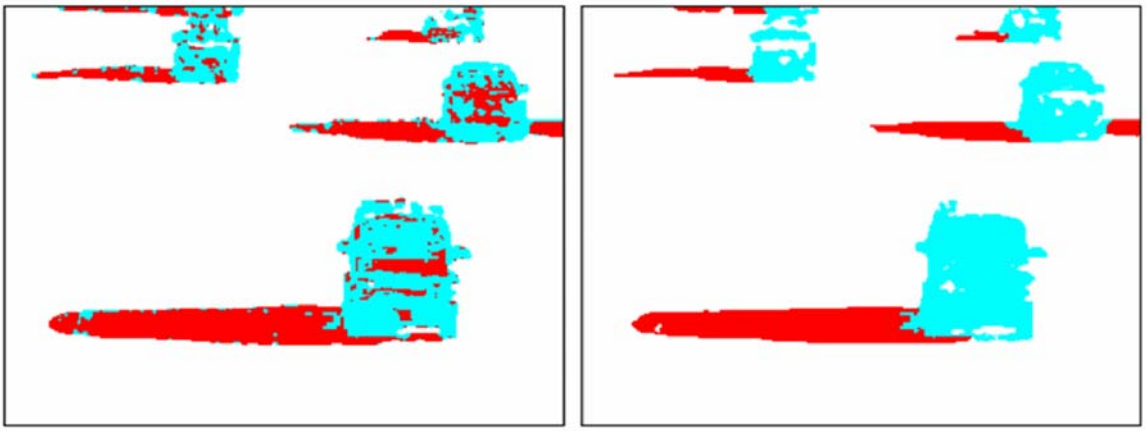


Figure 5: Results on a frame of sequence *Highway I*.
Note how step three of the algorithm reduces problems due to misclassified windshields of cars. (a) The output of the second step of our algorithm. (b) The output of the third step that shows the improvement.

In the following, we present results on test sequences for which ground truth data was available. For each sequence, we used the shadow discrimination accuracy [1] as the measure to quantify performance. Figures 6-11 show the original frames with foreground contours overlaid in the figure on the left along with the detected shadow, foreground and background regions in the figure on the right. To show how the performance of our method varies, we have shown two frames per sequence. The first shows one of the best results using our method, in terms of

discrimination accuracy and the second shows one of the worst.

Figure 6: One of the best results in terms of discrimination accuracy on the sequence *Highway III*.
Note how contours fit the foreground closely.

In Figure 7(a), road surface has a strip of dark color which is coincidentally aligned perfectly with the car windshield which makes distinction difficult and results in low discrimination accuracy. In Figure 10, we can see that the shadows are present on multiple planes of the background. Our algorithm does not put any restriction on the planes on which the shadow is cast. A strip in the background matches very closely in color with the person's shirt in Figure 10. The edges are also oriented along the same direction. A small strip is therefore classified as shadow after step two. However, it is further flipped in step 3 as it is completely surrounded by foreground regions.

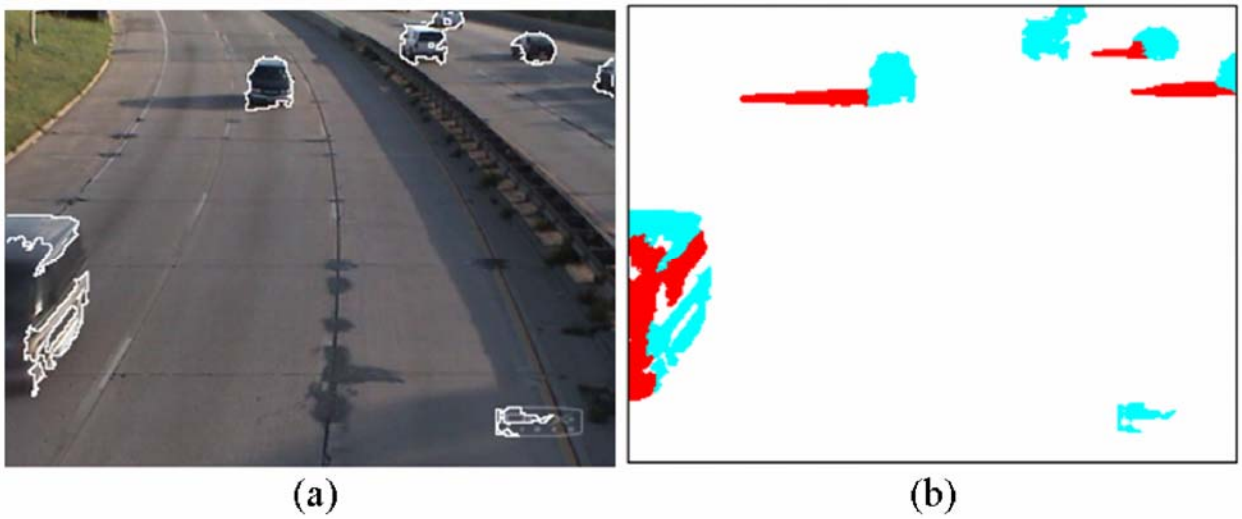
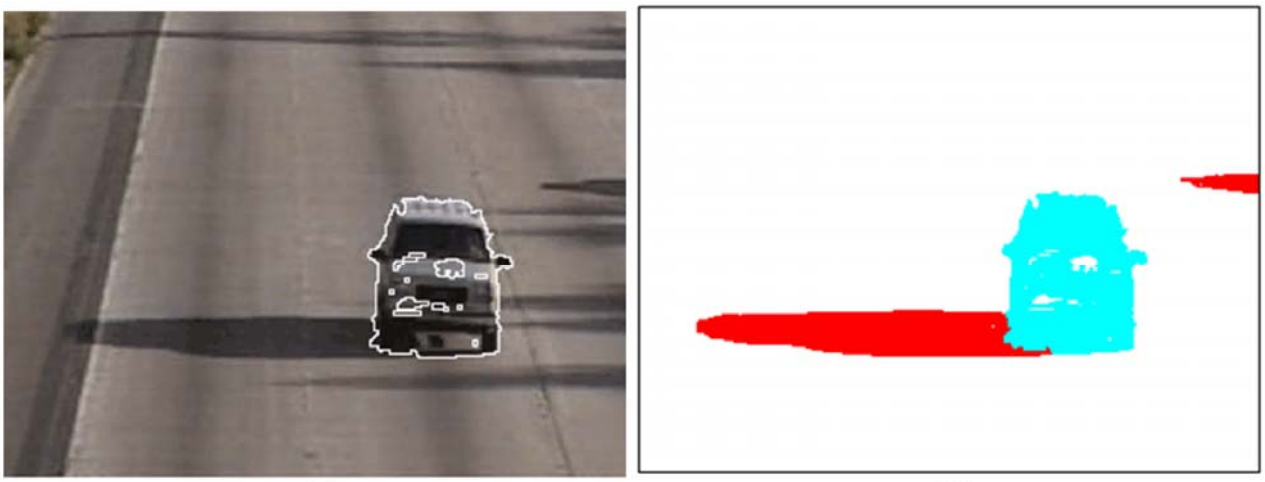


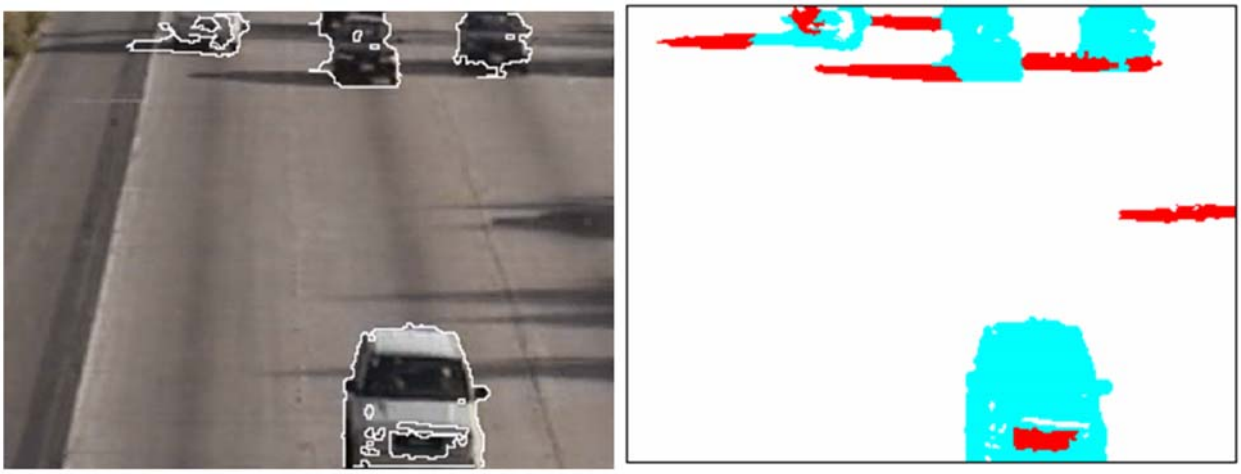
Figure 7: One of the worst results in terms of discrimination accuracy on the sequence *Highway III*.



(a) (b)

Figure 8: Best results on sequence *Highway I.*
Note that a central blob which is not detected as foreground, is not detected as shadow; however, it was classified as background by the background segmentation method.

Figure 11 shows a frame in which our method performs one of its worst. Small parts of the shirt are misclassified and a tiny part of the foot is classified as shadow. This worst result also produced a very good discrimination accuracy which shows how well our method performs on the given sequence.



(a) (b)

Figure 9: Worst results on sequence *Highway I.* (b) It shows distinctly, parts of vehicles identified as shadows.

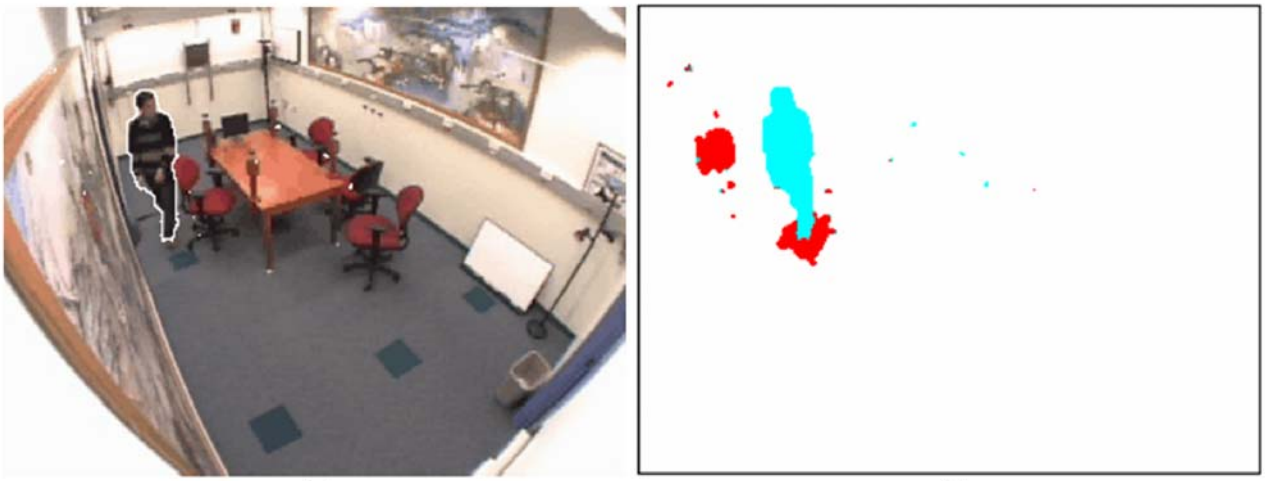


Figure 10: Best results on sequence *Intelligent Room.*
Note the shadows cast on different planes in the background.

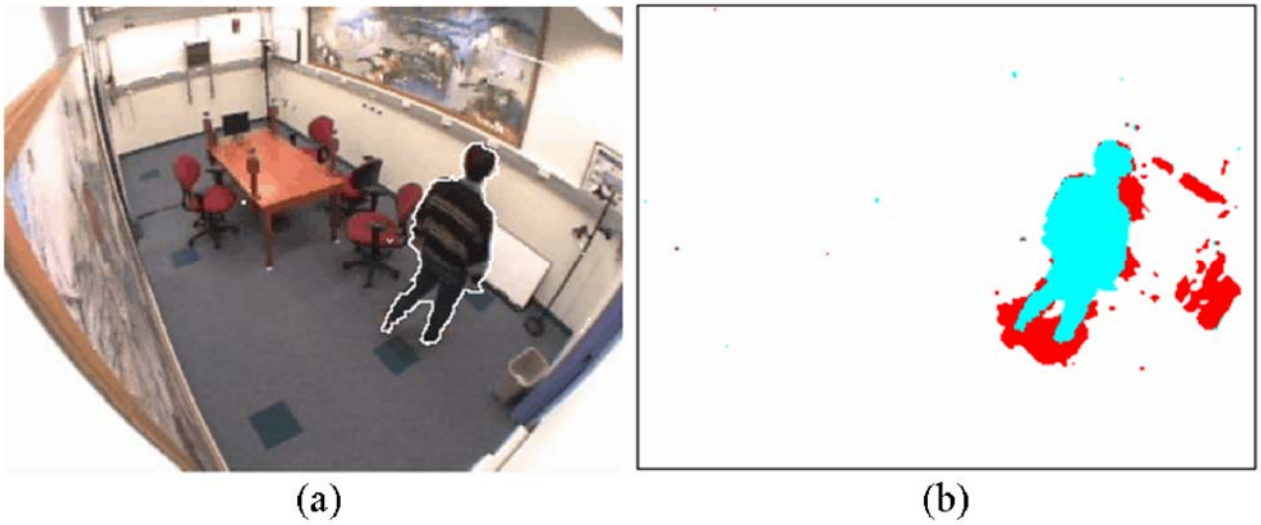


Figure 11: One of the worst results on the sequence *Intelligent Room.*

Challenges

Shadows pose some challenging problems for detection. This occurs especially when shadows are dark, which make them similar in intensity to dark colored vehicles. Also, in dark regions, edge gradient values are unreliable since they are highly quantized. Figure 12 shows results in such a case in which the method does reasonably well. Another problem in this sequence is that vehicles are small relative to frame size. This makes our resolution lower and distinction tougher. Figure 13 further demonstrates this. Vehicles in the image are even smaller; also the shadow of one falls on another vehicle in the frame shown. This is the problem of indirect cast shadows mentioned in [1]. Figure 13(b) shows that both vehicles are still separated correctly. As we go farther away from the camera, resolution becomes too coarse to make any distinction possible and shadows cannot be separated. Figure 14 presents results on the sequence *Highway I* on a frame with many vehicles present. Shadows again overlap; however in this case, vehicle size in

the frame is larger.

In Figure 14, shadow blobs are surrounded by foreground regions to a large extent. This still does not permit them to be flipped due to two reasons: 1) Their own large size which makes it unlikely to flip them; and 2) Our reasoning in step three which makes it unlikely to flip a blob, if flipping it connects regions which were previously unconnected.

(a) (b)

Figure 12: Results on sequence *Highway II*.
Note the similarity in the appearance of shadows and dark vehicles.

(a) (b)

Figure 13: Results on the sequence *Highway IV*.
Note the small size of vehicles and indirect cast shadows.

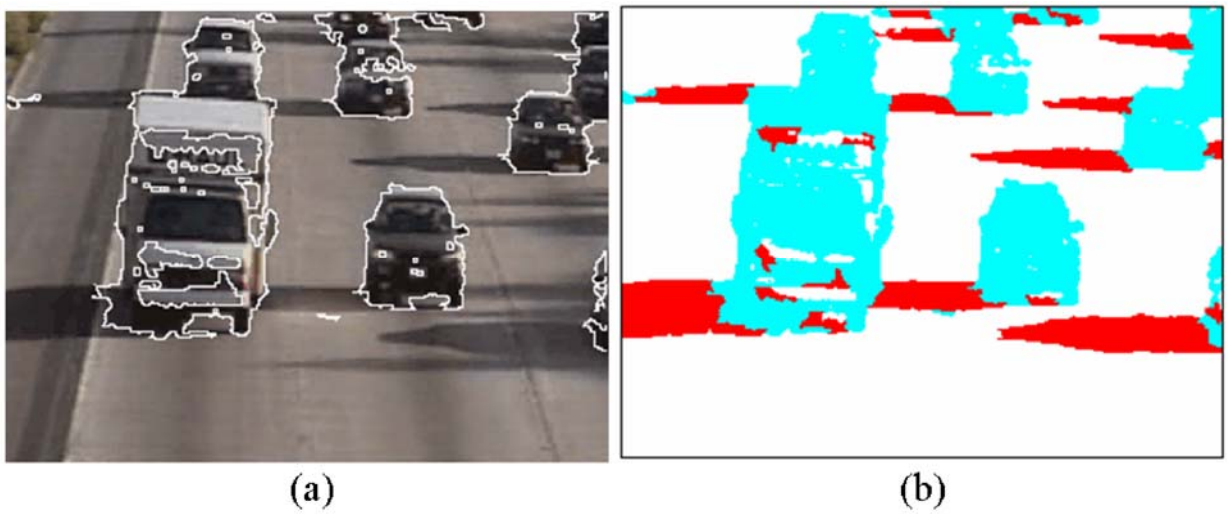


Figure 14: Results on a frame with different sizes and color of vehicles and overlapping shadows.

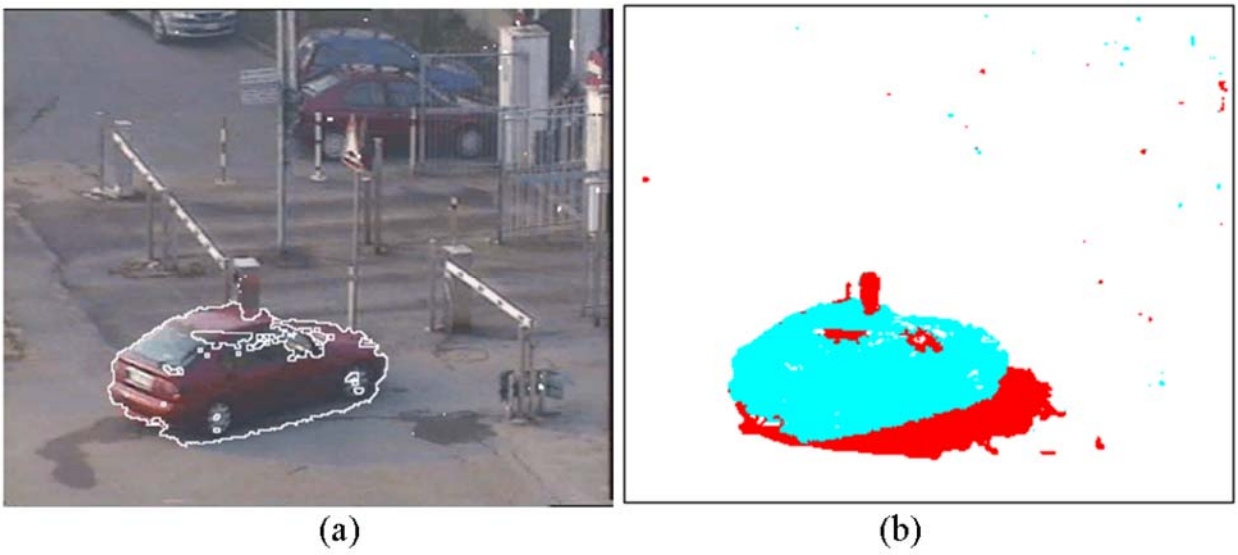


Figure 15: Results on a frame of sequence *Campus*.
Noise level of the sequence is high. Also shadows are cast on different planes.

Figure 14(b) shows that vehicles connected by shadow blobs are rendered separate after shadow detection. This promises to improve performance of automated vehicle counting on a highway. Another tough situation is depicted in Figure 15 on the sequence *Campus*. It is relatively noisy with objects of different sizes – vehicle and pedestrians. Figure 15(a) shows that object contour is detected fairly well.

In order to quantify results, we use shadow discrimination accuracy and detection accuracy [1]. Shadow detection accuracy (η) is the ratio of true positive shadow pixels detected to the sum of true positives and shadow pixels detected as foreground. Discrimination accuracy (ξ) is the ratio of total ground truth foreground pixels minus number of foreground pixels detected as shadow to the total ground truth foreground pixels. We manually segmented about 70 frames randomly from the sequence *Highway I* and about 50 frames (1 in every 5) from sequence *Highway III*.

Results are mentioned in Table 3. Table 3 also shows the results for *Highway I* with step 3 parameters tuned to *Highway III*, and vice versa. The respective accuracies show an expected decline; however, this decline is very small. This shows that even though step 3 requires fine tuning, performance without specific tuning also maintains good accuracy values.

Table 3: Detection and discrimination accuracy on various sequences.

Video Sequence	η (%)	ξ (%)	Number of pixels under test
Intelligent Room	87.99	97.08	246048
Highway I	84.11	96.69	903567
Highway I – Step 3 parameters tuned to *Highway III*	83.14	96.28	903567
Highway III	88.57	93.78	394524
Highway III – Step 3 parameters tuned to *Highway I*	88.39	92.58	394524

In Table 4, we provide best results out of those reported in [1] on the *Intelligent Room sequence*. From Table 4, it can be seen that our method performs better in terms of both η and ξ. An important point is that best results for η and ξ in Table 4 are produced by different algorithms as mentioned. Both η and ξ are conflicting in the sense that increasing one usually decreases the other. Our method outperforms both methods in both parameters together.

Table 4: Best detection and discrimination accuracies as reported in [1].
Note That Different Algorithms Perform Best On Each Parameter.

Sequence	η (%)	ξ (%)
Intelligent Room	78.61 (DNM1)	93.89 (DNM2)
Intelligent Room	87.99(our approach)	97.08 (our approach)

Table 3 indicates that our proposed method does very well on different sequences with varying illumination conditions, sizes of shadow, noise levels and whether the sequence is indoor or outdoor.

CHAPTER 5
POSITIONING OF CAMERAS

The ratio between the amount of information that can be collected by a camera and its cost is very high, which enables their use in almost every surveillance or inspection task. For instance, it is likely that there are hundreds to thousands of cameras in airport or highway settings. These cameras provide a vast amount of information which is not feasible for a group of human operators to simultaneously monitor and evaluate effectively or efficiently. Computer vision algorithms and software have to be developed to help the operators achieve their tasks. The effectiveness of these algorithms and software is heavily dependent upon a "good" view of the situation. A "good" view in turn is dependent upon the physical placement of the camera as well as the physical characteristics of the camera. Thus, it is advantageous to dedicate computational effort to determining optimal viewpoints and camera configurations.

In order to obtain better views, cameras can be mounted on moving platforms for data collection purposes. A system that would tell the moving platform where to place itself in order to collect the most information will enhance their usefulness. The same algorithms can be used in more restricted settings to identify positions and camera parameters for placement on pre-existing structures.

Our work is focused on the problem of task-specific camera placement, where we attempt to determine how to place cameras relative to vehicle/pedestrian trajectories, in particular highway data collection, in order to provide the maximal amount of information regarding these activities (motion recognition, measurements, etc.) with the minimal number of cameras. This work is an extension of Robert Bodor's work on optimal camera placement for automated surveillance tasks [1].

CHAPTER 6
THE INITIAL WORK

Bodor's *et al.* method [1] [2] focuses on optimizing the view of a set of target motion paths taken through an area of interest. As such the method considers dynamic and unpredictable environments, where the subjects of interest change in time. The considered paths are not known a priori. The method focuses on optimizing a set of paths as a whole and does not attempt to optimize the view of each individual path. The method does not attempt to measure or reconstruct surfaces and objects and does not use an internal model of the subjects for reference. In this Bodor's method differs significantly in its core formulation from camera placement solutions described earlier [1].

The data collection of the paths is achieved by using automated tracking algorithms [4] which identify targets and record their trajectories. After the collection of those paths, they are described in world coordinates using the method described in [5]. The paths are measured without any parametric assumption, and a line is fit via a least squares approximation, thus the paths are assumed straight (linear). However if there is high curvature in a path, this path can be split into smaller linear segments so Bodor can apply his method to arbitrary paths. This also explains why the paths are parameterized as being:

$$\overrightarrow{s_j} = [\phi_j \;\; x_j \;\; y_j \;\; l_j]^T$$

where ϕ_j is the orientation of the path j, (x_j, y_j) are the coordinates of the path's center and l_j is the length of the path.

CHAPTER 7
OPTIMIZATION CONSTRAINTS

Bodor *et al.* introduce two constraints. The first one states that the camera must maintain a minimum distance d_0 from each path in order to ensure the full view of each path (Figure 1). This distance d_0 is computed by the similar triangle theorem:

$$d_0 = \frac{l_j f}{w}$$

Equation 1: Definition of d_0.

where l_j is the length of the path, f the focal length and w the width of the camera sensor.

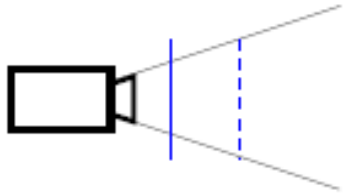
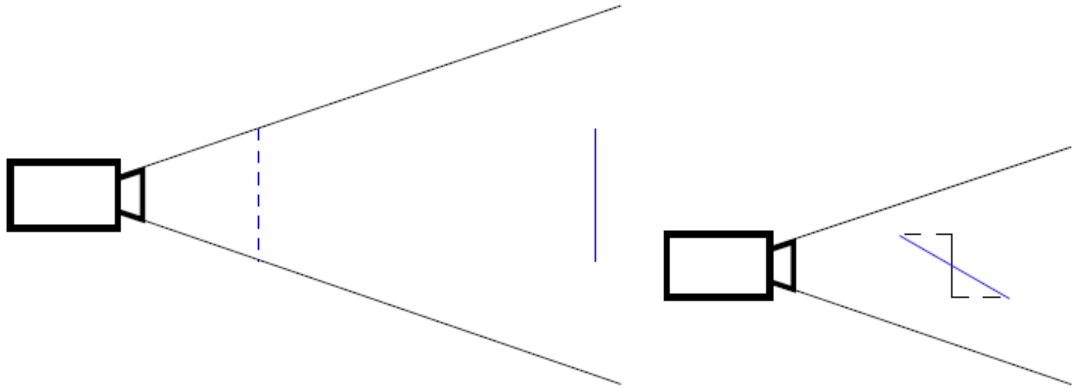


Figure 26: When d_{ij}(solid) < d_0 (dashed), the camera is unable to observe the full motion sequence and fails the first constraint [1].

The aim is to be as close as possible to the path so that a maximum amount of information can be retrieved (Figure 2a).



(a) Increasing object-camera distance. (b) Increasing foreshortening.

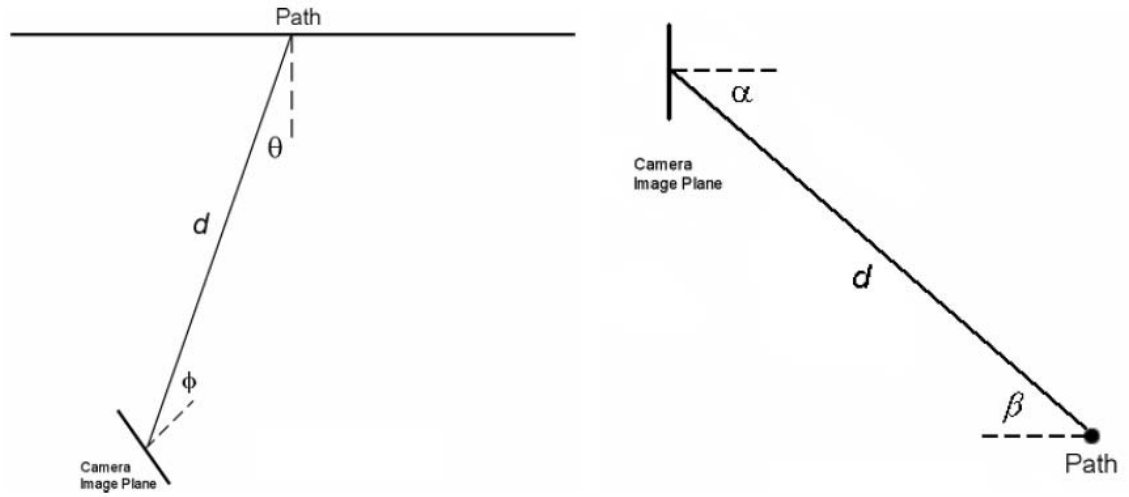
Figure 17: Configurations that decrease observability in pinhole projection cameras [1].

Another factor that reduces the observability of an object is the angle between the camera's view direction and the object itself (Figure 2b). Based on these assumptions Bodor defines an objective function for each path-camera pair G_{ij}, where each of these elements describes the observability of path j by camera i.

$$G_{ij} = \begin{cases} 0 & \text{if } d_{ij} < d_0 \\ \frac{d_0^2}{d_{ij}^2} \cos(\theta_{ij}) \cos(\phi_{ij}) \cos(\alpha_{ij}) \cos(\beta_{ij}) & \text{otherwise} \end{cases}$$

Equation 2: Objective function.

where the angles are defined by Figure 3.



(a) Top view (b) Side view

Figure 18: Definition of the angles used in the optimization function [1].

Optimizing this function over camera parameters will solve the observability problem for the single camera, single path case. The sum over all paths of these objective functions gives the placement of one camera.

CHAPTER 8
VIEWPOINTS

Bodor uses also "generalized" viewpoints as describes in [3]. However his viewpoints are described in a 7-dimensional space.

$$\overrightarrow{u_i} = [X_i \; Y_i \; Z_i \; \gamma_{xi} \; \gamma_{yi} \; \gamma_{zi} \; f]^T$$

where the first 6 parameters are the extrinsic parameters of the camera (position and orientation) and *f* is the focal length.

Robert Bodor considers that the roll (the rotation around the *y* axis) doesn't affect the observability. Furthermore his system is considered to have an overhead view of the scene (it is mounted on rooftops or on ceilings), which removes the height degree of freedom (*Z*). This assumption leads to the introduction of another constraint on the tilt of the camera (γ_x), which is strongly connected to *Z*. Finally, all the cameras are considered without zoom and thus with a fixed focal length *f*, which is also assumed the same for all the cameras. This enables to make the assumption that the computation of the camera positions is independent of the order in which the cameras are placed.
When taking into account all these constraints, the viewpoint vector is reduced to:

$$\overrightarrow{u_i} = [X_i \; Y_i \; \gamma_{zi}]^T$$

With this reduced vector only θ_{ij} and ϕ_{ij} remain relevant for further optimization purposes.

$$\Rightarrow \quad G_{ij} = \begin{cases} 0 & \text{if } d_{ij} < d_0 \\ \frac{d_0^2}{d_{ij}^2} \cos(\theta_{ij}) \cos(\phi_{ij}) & \text{otherwise} \end{cases}$$

Equation 3: Reduced form of the objective function.

In order to run the optimization on the viewpoints Bodor has to transform the θ_{ij} and ϕ_{ij} into *X*, *Y* and γ_z. This is achieved using basic geometry.

CHAPTER 9
EXTENTION OF PREVIOUS WORK

Angles and Optimization

The next step to take is to optimize the function with respect to Z and γ_x, which in Bodor's scheme were constant. We will still keep the assumption that a "roll" rotation around the focal axis of our system is still not important for our computations. At this point, we have to find new relations between Z and γ_x on the one hand and the angles α and β on the other defined earlier (Figure 3).

We also considered trying a different approach in the computation of the angles than the one Robert Bodor used [1]. Instead of making projections from above and from the side (Figure 3), we believe that the optimization angles can be described more effectively in three-dimensional space. When computing the same objective functions, we would be able to use only two angles in 3-D space as opposed to the four angles in Bodor's projections: two angles from the projection from above (Figure 3a) and two additional angles from the side projection (Figure 3b).

Figure 5 illustrates the basic idea of this new approach. Instead of using projections we are trying to define two new angles directly in three-dimensional space. One new angle would be between the normal vector of the path (thin blue line) and the main vector of the dashed line that links the center of the path to the center of the image plane (black square). The other angle would be between the normal to the image plane (thin black line) and the opposite vector used for the previous description of the angle.

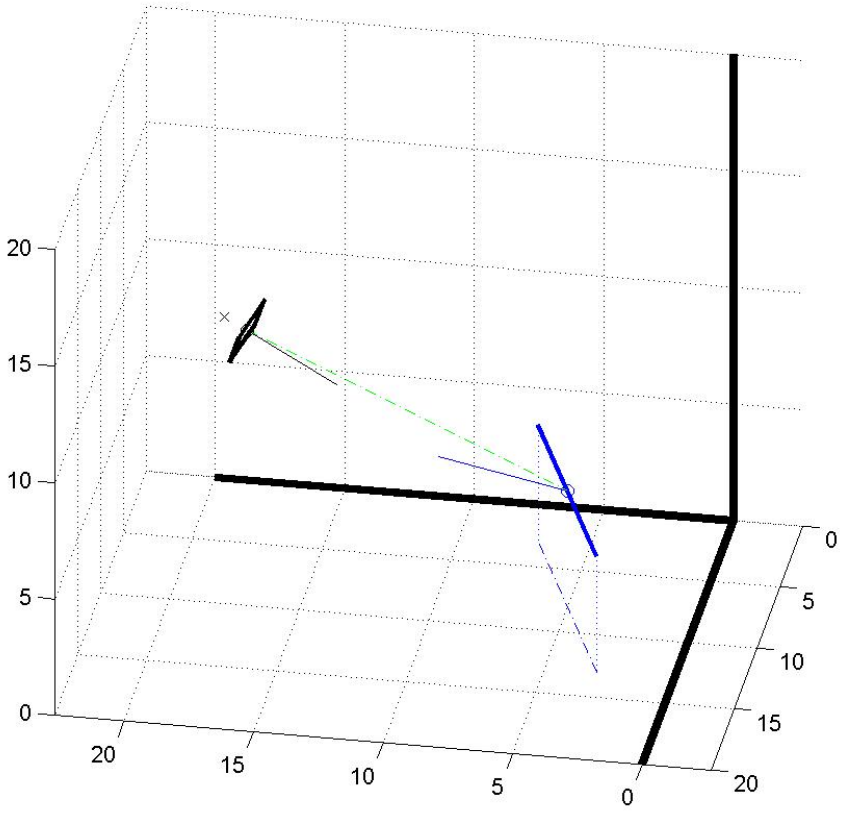


Figure 19: Angle calculations in 3-D space.

II. 4 APPLICATIONS

When we use our methods to determine the optimal camera placements at traffic intersections (Figures6 and 8), the results are promising (Figures 7 and 9).

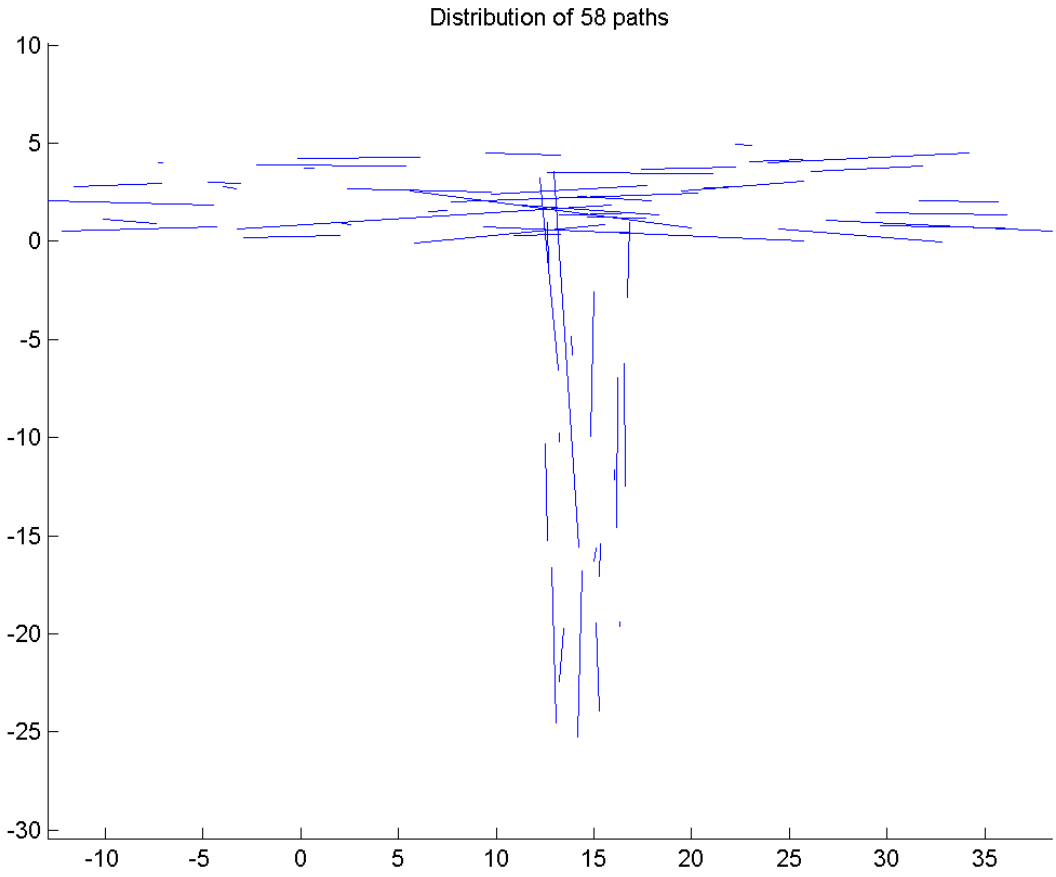
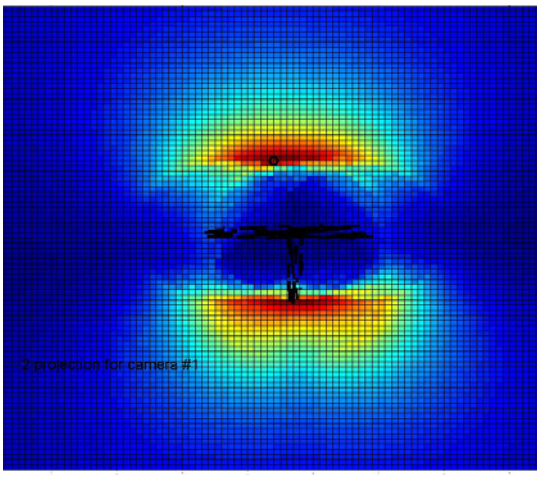
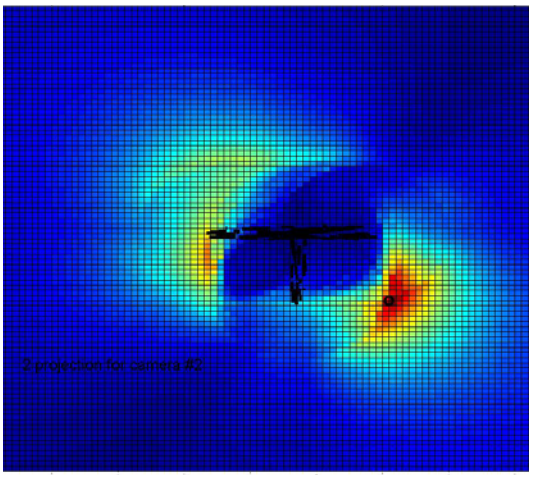


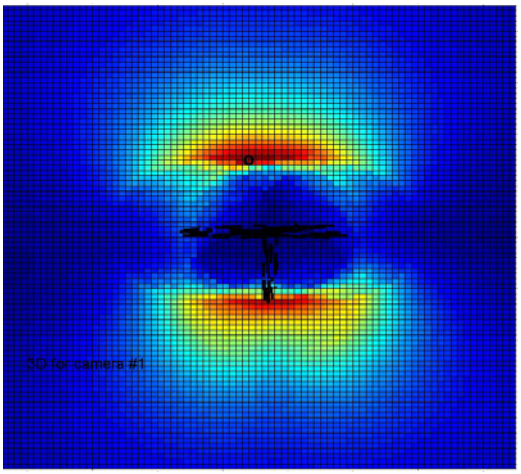
Figure 20: Paths distribution at a T-intersection.



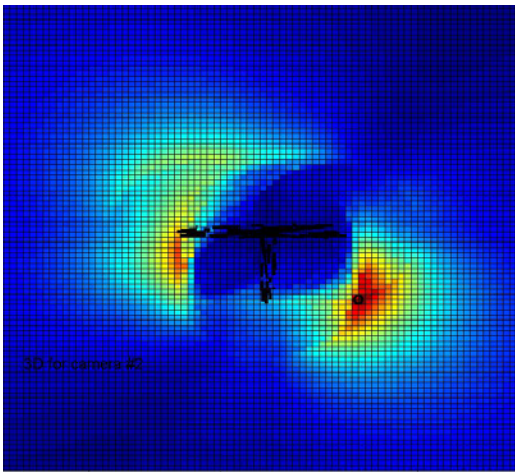
a) Camera position with 2D projection for camera 1. The red surface shows the best position. The small black circle shows the actual best camera position.



b) Camera position with 2D projection for camera 2. Our method enables the camera placement to observe paths that were not or not as well seen from the first camera.



c) Camera position with 3D calculation for camera 1. The optimal placement coincides with the one in the 2D projection



d) Camera position with 3D calculation for camera 2. Here again, the placement coincides.

Figure 21: Objective surface for paths at a T-intersection. a) and b) are the results for the 2D projections and c) and d) show the results for the 3D method.

All this positioning is realized without any constraints. Robert Bodor added the possibility of constraints for the camera placement. For instance the cameras could have to be fixed on rooftops. In the following figure the impact of this new constraint can be observed. In this figure, 3 cameras had to be placed on the rooftops of nearby buildings.

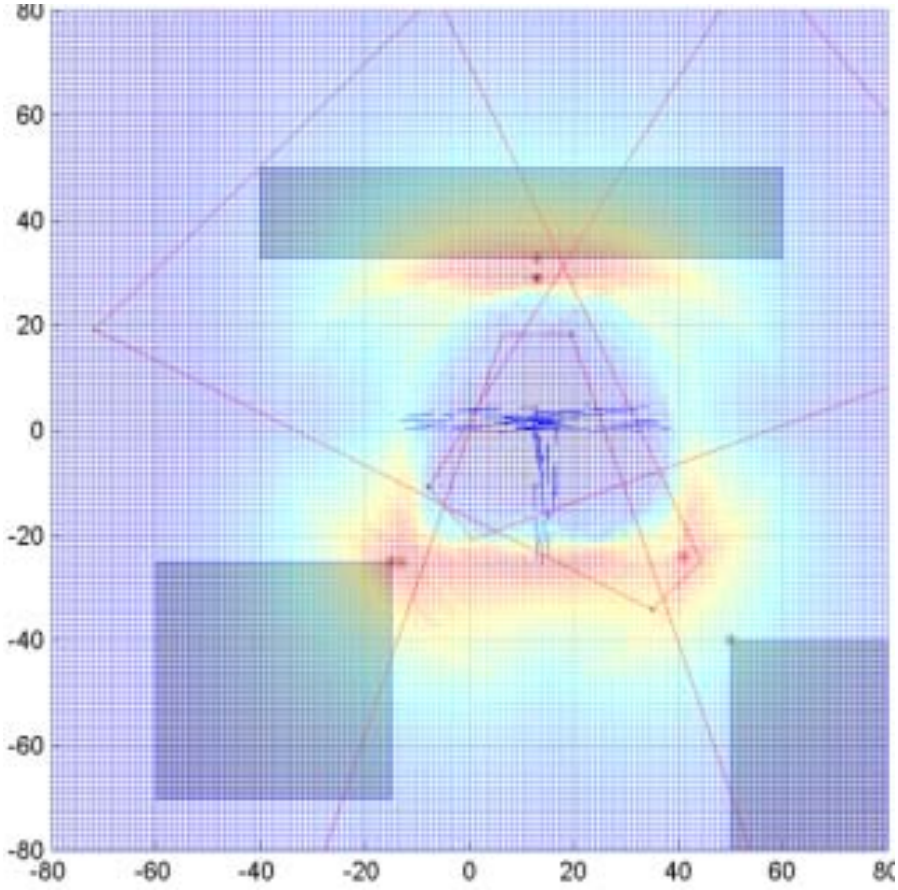


Figure 22: Objective surface at a T-intersection with rooftop constraint.

Figure 8 shows the result of the introduction of additional constraints. On this figure the different shaded rectangles represent the buildings, and the different trapezoids represent the projection of the view frustums of the cameras. What can also be seen are the camera positions without constraints, which are placed at the same locations we described earlier. Even after using these additional constraints the method is still giving the “intuitive” optimal position.

The following figures give the results for another intersection.

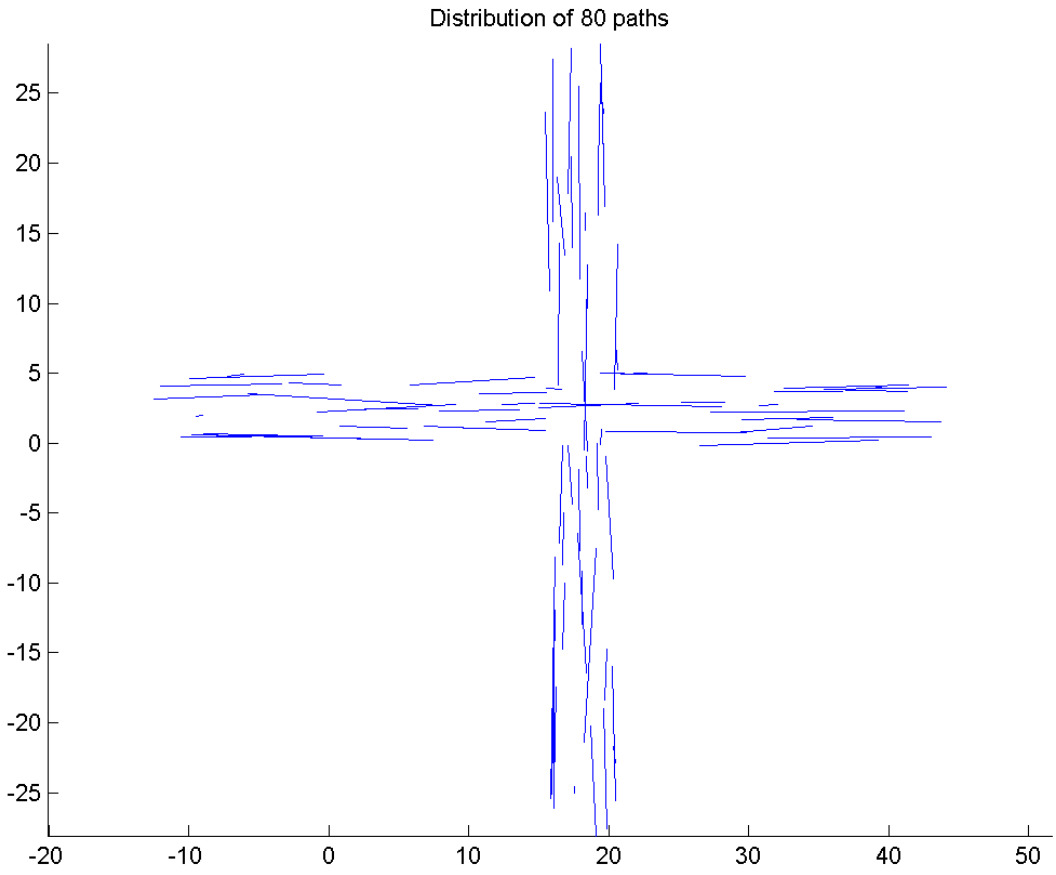
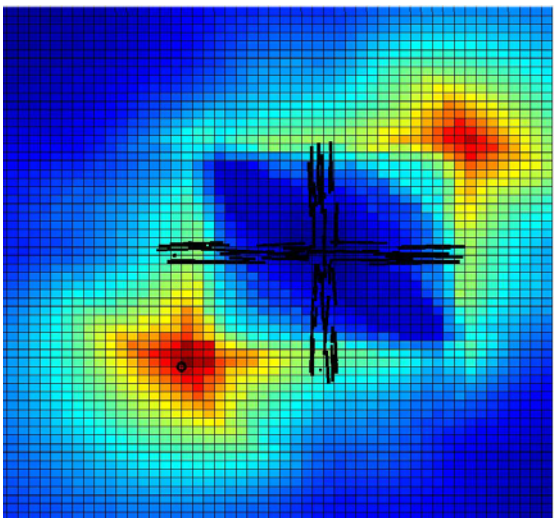
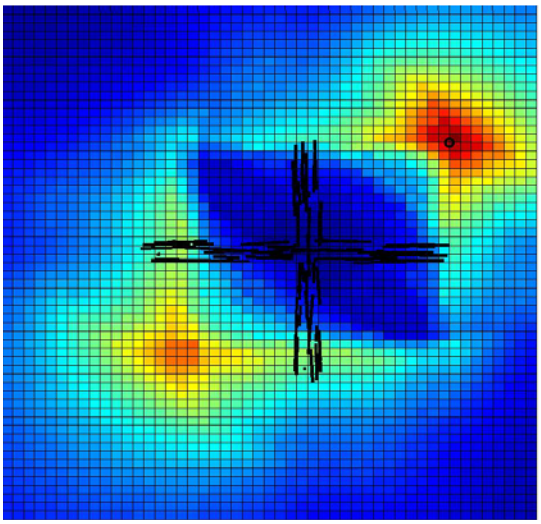


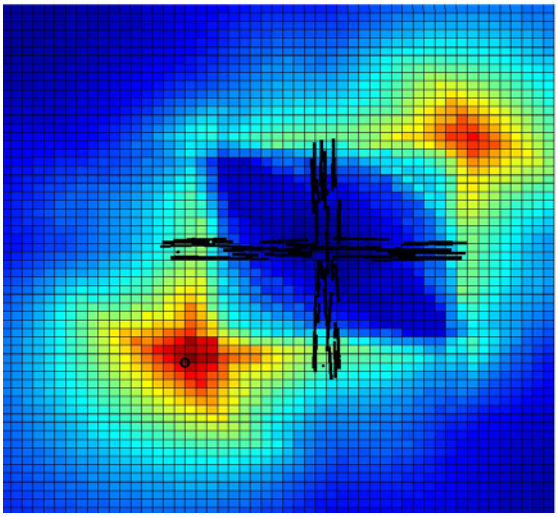
Figure 23: Paths distribution at a four-way intersection.



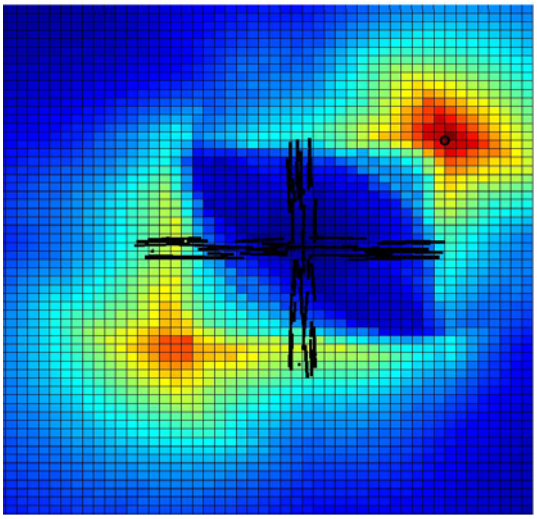
a) Objective surface with 2D projection for camera 1.



b) Objective surface with 2D projection for camera 2.



c) Objective surface with 3D computation for camera 1.

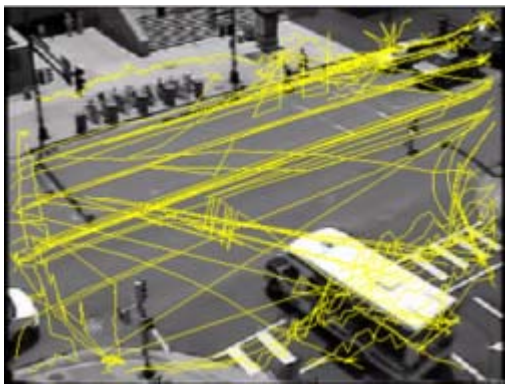


d) Objective surface with 3D projection for camera 2.

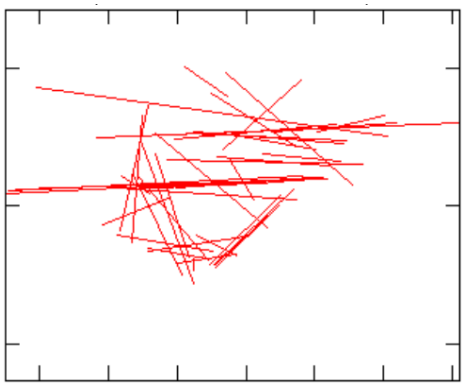
Figure 24: Objective surface for paths at a four-way intersection.

In the objective surfaces above, the red zones are the ones which suit our constraints described in section 2.1 best. The small black circles show the maximum of these zones and as such are the optimal points on which to position the cameras.

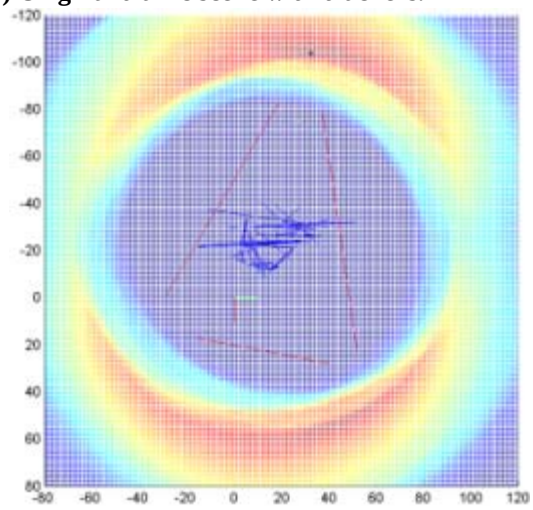
One final result we want to show is the surveillance of the intersection Washington Avenue and Union Street in Minneapolis.



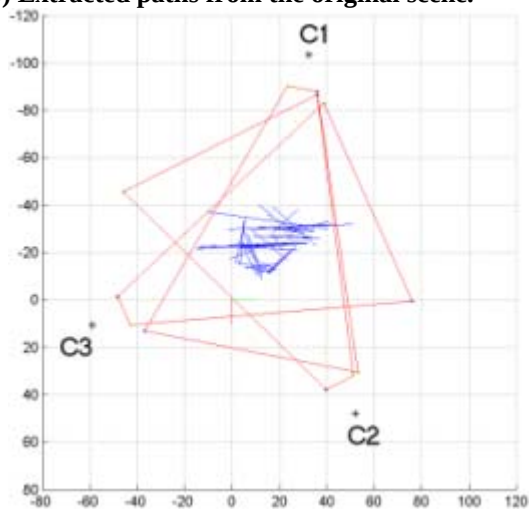
a) Original traffic scene with trackers.



b) Extracted paths from the original scene.



c) Objective surface for the first camera.



d) Final position for three cameras.

Figure 25: Camera placement for a traffic scene. a) It shows the original scene; b) It shows the extracted paths from the data in the original scene; c) It gives the objective surface for the placement of the first camera; and d) It supplies the final result of our method.

CHAPTER 10
NEW LEARNING TECHNIQUES FOR SHADOW DETECTION

We use a semi-supervised learning framework to differentiate between moving shadows and other foreground objects in video sequences. In this method, Support Vector Machines are used as the base classifier. Then applying co-training, the system improves its classification accuracy by using new data available at each incoming frame. Experimental results show that such an approach can produce favorable results as compared to current systems. Secondly, we observe that co-training can reduce the effects of certain types of `population drift' that can appear in many learning settings. Thus, when trained on a small labeled set of training examples, the system generalizes well to examples that come from a different distribution also. We show how this can result in a computer vision system tuning itself automatically, potentially taking the human out of the loop.

Our approach involves using effective pixel-based features from the frames of video to perform shadow detection. Specifically, we use four image features from [7], which have been shown to be effective in real world scenarios. These features, computed for each pixel are summarized in the following: (1) The difference between edge gradient direction at the pixel in the current frame and that in the background model (f1). Here, the background model is computed using the mixtures of Gaussians technique [9, 10]; (2) The difference between edge strength at the pixel in the current frame and that in the background model (f2); (3) The color distortion between the pixel in the current frame and that in the background model (f3). This distortion is computed from the 3D color model in the R, G, B space where each pixel is represented by a point; and (4) The ratio of intensity of the pixel in the current frame to that in the background model (f4). We wish to solve the following classification problem. For each pixel in a frame of video, given a feature vector f 2 <4, we want to find the corresponding label (what region the pixel belongs to) l 2 {S, FG} where S indicates shadow and FG indicates foreground. We use Support Vector Machines as the base classifier and then apply co-training in a semi-supervised setting with each incoming video frame.

Support Vector Machines:

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 + C\sum_i \beta_i, \\ &\text{subject to } \ y_i f(X_i) \geq 1 - \beta_i, \ \beta_i \geq 0. \end{aligned} \tag{1}$$

Here, w is the vector that defines the separating hyperplane in the feature space, and _0is are the slack variables. C is the penalty term that penalizes low margin classifications and misclassifications. This is necessary as we expect noisy data that might not be linearly separable. In that case, we need to allow for misclassification, however adding a penalty for such errors. f(X) = sign(wTX) is the decision function of the SVM. The optimization problem in (1) is usually solved by obtaining the Lagrange dual. A more general dual problem can be formulated as

$$\max \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathrm{K}(X_i, X_j), \quad \text{subject to} \quad 0 \leq \alpha_i \leq C, \ \sum_i \alpha_i y_i = 0. \tag{2}$$

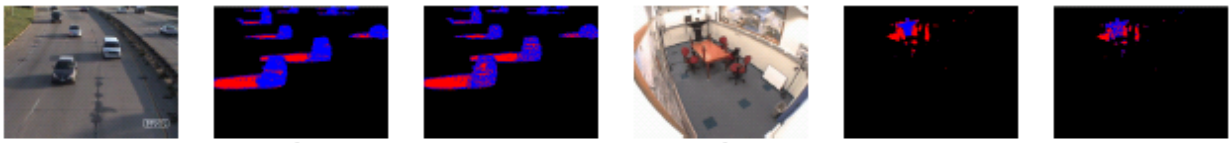
Co-training (Blum and Mitchell, 98)

The original motivation to develop co-training came from the fact that labeled data is scarce, whereas unlabeled data is usually plenty and cheap to obtain. In this algorithm, two classifiers are trained using two mutually exclusive feature sets (F1 and F2) on the initial labeled data. Then each classifier is deployed on the unlabeled data, and at each round, it chooses the example which it can label most confidently from each class, and adds it to the pool of labeled examples. This is carried out iteratively until a fixed number of rounds, or until all the originally unlabeled data is labeled. See [12, 8] for a detailed description of the co-training algorithm. Blum and Mitchell [12] discuss two assumptions under which theoretical guarantees on learning with co-training are proved. 1. The conditional independence assumption: In order for this assumption to hold, the two feature sets F1 and F2 should be conditionally independent given the class label. This assumption is not satisfied in most real-world data sets. Even in our case, this assumption is violated since none of the features are truly independent of each other given the class label. Nigam and Ghani [8] claim that even in such cases, there is an advantage to be gained by using co-training. Our results indicate the same as we will discuss later.
2. The compatibility of the instance distribution with the target function [12, 8]: This means that when the assumption holds, it is possible to predict the same labels on examples using each feature set independently of the other. In other words, both views of the example (each using a different feature set) would be independently enough to learn the target concept, if sufficient training examples were provided. This assumption is also violated in most practical settings. The idea behind co-training is the following. If the two classifiers are trained using conditionally independent feature sets, when one classifier labels an example, it is seen as a random training example by the other classifier. In this case, the other classifier benefits from this added example. In this way, different “views” of the example may help achieve better combined classification accuracy, even though individual classifier accuracy may be much weaker. We use the co-training approach to try and eliminate some problems caused by the supervised learning setting in our domain. Although scarcity of labeled data is one consideration while choosing to use co-training, we also look for the added advantages of possibly improved speed of operation, and tendency to reduce effects of population drift (discussed later).

From the feature vector f described earlier, we use features f1 and f2 for one classifier and f3 and f4 for the other. This split makes practical sense since the features in the first set are a result of image gradients, while the features in the second set are results of pixel colors. Our experiments also show that this feature split gives the maximum benefit with co-training. Henceforth, the semi-supervised method refers to the one using co-training, and supervised refers to the method using only one SVM and no unlabeled data.

CHAPTER 11
RESULTS

We performed experiments on real video sequences to compare results in both the supervised and the semi-supervised case. In the following figures, blue indicates regions classified as foreground by the algorithm while red indicates regions classified as shadows. An important aspect that should be considered when evaluating these results is that no global information is used for labeling the pixels. We use only local pixel-based cues. Postprocessing, as done in [7] may improve classification accuracy values further.



(a) (b) (c) (d) (e) (f)

Figure 26: Shadow detection performance using Support Vector Machines.

In Figure 26, we use labeled data (100 points) from one frame of the video to construct the base classifiers. Then each new frame is fed to the classifiers, and using 50 randomly picked unlabeled examples (pixels) from the new frame, the combined classifier is built using co-training. In Table 1, we compare our results to other pixel-based methods (methods in which no postprocessing is performed). The two metrics - Shadow Detection Accuracy and Shadow Discrimination Accuracy are defined in [1]. Detection accuracy aims to measure the percentage of true shadow pixels detected. Discrimination accuracy is an indicator of the false positive rate, higher the discrimination accuracy, lower is the false positive rate. We compare our results with those from [1] on the 'Intelligent Room' and the 'Highway' sequence. Ground truth data from the 'Intelligent Room' sequence was obtained from UCSD (109 manually marked frames), and it is the same that was used to obtain the results reported in [1]. For the other sequence, we manually marked shadow and foreground regions in 50 randomly selected frames. From Table 1, we can see that our approach gives favorable results compared to the best results
reported in the literature (these results are obtained from the survey paper [1]). The last row in the table will be discussed in the next section.

Table 5: Shadow detection and discrimination accuracy with learning.

Video Sequence	Intelligent Room		Highway	
Method	η	ξ	η	ξ
SNP [1]	72.82	88.90	81.59	63.76
SP [1]	76.27	90.74	59.59	84.70
DNM1 [1]	78.61	90.29	69.72	76.93
DNM2 [1]	62.00	93.89	75.49	62.38
Our approach (with Co-training)	**86.49**	**83.27**	**83.45**	**79.72**
Our approach (with Co-training) (adaptive)	**81.23**	**79.12**	**78.42**	**74.57**

CHAPTER 12
CO-TRAINING HELPS IN THE PRESENCE OF POPULATION DRIFT

A very common assumption in the design and evaluation of learning methods is that training data and test data come from the same underlying distribution. This assumption is not valid for most real-world settings since the underlying distributions may change and also the data gathering process is not perfect [17, 18, 19]. Although sample selection bias has been dealt with extensively in the literature, the aspect of population drift has not been studied much [19]. Population drift refers to the change in feature value distributions over time. In a learning scenario, any such change occurring between the learning and the prediction phases can cause the learning algorithm to perform poorly on the test data.

Our experiments show that this approach reduces the problems caused by population drift in many computer vision tasks. In Figure 5 we show shadow detection output using the proposed semi-supervised method. In all cases, initial labeled data are obtained from a video sequence that is different from the video on which the result is shown. Then using unlabeled data in each new frame, classification is performed. Thus, the system is not specifically tuned to any particular scene. For example, see the top row in Figure 5 (a)-(d). For the indoor sequence, training data are obtained from the outdoor sequence and vice versa.

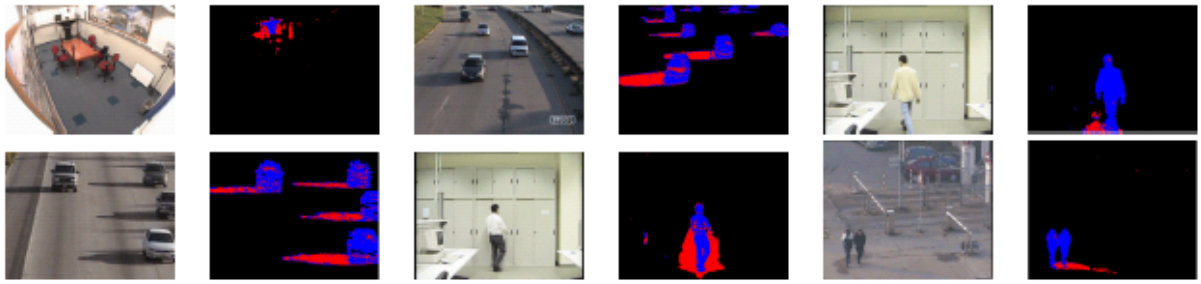


Figure 27: Shadow detection performance with co-training.

Observing the intensity of shadow and foreground regions in the two sequences, we see that they differ widely. One of the features we use incorporates this intensity information. For the outdoor sequence, shadow regions have this value in the range of 0.5 to around 0.75, whereas for the indoor video, this ratio ranges from 0.65 to about 0.95. Even with this difference in the feature values, training on the labeled set of one, and then using co-training on the other produces good results. We can see in all images that even though scene conditions differ widely, the detection quality is fairly good. For a comparison, see images from [1], keeping in mind that the results shown in that are obtained by tuning the system specifically for each video. We do not need to do any tuning in this semi-supervised method. The last row of Table 1 shows Detection and Discrimination Accuracy on the two video sequences, when training is performed using 100 examples from a different sequence. This still shows comparable values to other methods, showing adaptive behavior. Note that the other methods have been fine-tuned to work well for the particular sequence explicitly. In many vision tasks, we know what features best represent the knowledge based on which we wish to distinguish between different classes. Even though these features are the same, their values might change from one scene to another (based on illumination conditions, scene geometry, sensor variation, background etc.). This is countered by

tuning the system manually, usually by trial and error, or exhaustive search. A semi-supervised framework like the one presented in this paper holds potential for similar vision problems, in which the system needs to adapt online to scene changes.

CHAPTER 13
CONCLUSION

Results about the camera placement experiments coincide with the “intuitive” camera placement we would use to observe vehicles at highways or other traffic sites. Thus these results are promising in a way that our approach seems to provide us with good camera placements.

We are continuing to enhance our method at the moment. One future step to be taken is the inclusion of the optimization over the focal length. This inclusion will enable the use of cameras with zoom capability.

In terms of shadow detection, in this report, we presented a semi-supervised approach to detect moving shadows in video sequences using Support Vector Machines and the co-training algorithm. We show that co-training helps improve performance in the presence of specific types of population drift that can occur in many learning settings. The paper demonstrates how a computer vision system can tune itself automatically from one video to another since tuning can be posed as a population drift problem. Our results are preliminary and more extensive experiments are clearly needed to establish behavior across different types of population drift. Our future research will focus on these experiments and also on the relevant theory that can help understand these observations better.

REFERENCES

[1] A. Prati, I. Mikic, M.M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918-923, Jul. 2003.

[2] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 65-76, Mar. 1999.

[3] K. Onoguchi, "Shadow elimination method for moving object detection," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, pp. 583-587, Sydney, Australia, 1998.

[4] A. Elgammal, D. Harwood, and L.S. Davis, "Non-parametric model for background subtraction," *Proc. IEEE Int'l Conf. Computer Vision '99 Frame-Rate Workshop Kerkyra, Greece, 1999.*

[5] S. Tattersall, and K. Dawson-Howe, "Adaptive shadow identification through automatic parameter estimation in video sequences," *Proc. Irish Image Processing and Machine Conf*., pp. 57-64, Dublin, Ireland, Sept. 2003.

[6] T. Horprasert, D. Harwood, and L.S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," *Proc. IEEE Int'l Conf. Computer Vision '99 Frame-Rate Workshop Kerkyra, Greece, 1999.*

[7] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting objects, shadows, and ghosts in video streams by exploiting color and motion information," *Proc. Eleventh Int'l Conf. Image Analysis and Processing*, pp. 360-365, Copenhagen, Sept. 2001.

[8] I. Mikic, P. Cosman, G. Kogut, and M.M. Trivedi, "Moving shadow and object detection in traffic scenes," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, pp. 321-324, Barcelona, Spain, Sept. 2000.

[9] Y. Sonoda, and T. Ogata, "Separation of moving objects and their shadows, and application to tracking on loci in the monitoring images," *Proc. Fourth Int'l Conf. Signal Processing*, vol. 2, pp. 1261-1264, Seattle, Washington, 1998.

[10] M. Kilger, "A shadow-handler in a video-based real-time traffic monitoring system," *Proc. IEEE Workshop Applications of Computer Vision*, pp. 11-18, Vancouver, Canada 1992.

[11] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, vol. 2, 1999.

[12] S. Atev, O. Masoud, and N. Papanikolopoulos, "Practical mixtures of Gaussians with brightness monitoring," *Proc. IEEE Seventh Int'l Conf. Intelligent Transportation Systems*, pp. 423-428, Osaka, Japan, Oct. 2004.

[13] R. Bodor, A. Drenner., M. Janssen, P. Shrater, N. Papanikolopoulos, "Optimal Camera Placement for Automated Surveillance Tasks, International Journal of Robotics Research, in submission.

[14] R. Bodor, A. Drenner., M. Janssen, P. Shrater, N. Papanikolopoulos, "Mobile camera positioning to optimize the observability of human activity recognition tasks, *In Proc. Conference on Intelligent Robots and Systems*, Alberta, Canada, 2005.

[15] K. Tarabanis, R. Tsai, "Computing viewpoints that satisfy optical constraints, *In Proc. IEEE International Conf. Computer Vision and Pattern Recognition*, Hawaii, 1991.

[16] B. Maurin, O. Masoud, and N. Papanikolopoulos, "Monitoring crowded traffic scenes", *In Proc. IEEE Intelligent Transportation Systems Conference*, Essen, Germany, 2002.

[17] O. Masoud and N. Papanikolopoulos, “Using geometric primitives to calibrate traffic scenes”, In Proc. International Conf. on Intelligent Robots and Systems, Sendai, Japan, 2004.