Detecting and Tracking Tractor-Trailers Using View-Based Templates

---

A Master's Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Vinay Gidla
May 2010

Accepted by:
Dr. Stanley Birchfield, Committee Chair
Dr. Robert J. Schalkoff
Dr. Damon L. Woodard

Abstract

Vehicle tracking and classification is an important application of computer vision. There exist mainly two types of tracking techniques: Appearance-based tracking and Feature-based tracking. Appearance-based techniques generally require a model or template of the target to be tracked. However the model needs to be robust to the multitude of deformations possible in the course of the target's movement. This research explores the use of a collection of view based templates, instead of a single template, to accurately trace the contours of the tractor-trailers in moving traffic.

The tracking begins with creation of a template sequence by manually processing a segment of traffic which contains an exemplar of the desired class of tractor-trailers. The processing involves manually segmenting the tractor-trailer from the rest of the scene in each frame. The template sequence is intended to capture the appearance of the vehicle in all possible poses as it moves through the lane. This collection is then used to detect and track similar tractor-trailers during their movement through the lane. Salient features, such as gradient magnitude information, are incorporated for better alignment of the contour during tracking.

Dedication

To my mother, father and Jesus.

Acknowledgement

I am extremely grateful to Dr. Birchfield for his patient guidance all through this research. I also thank Dr. Schalkoff and Dr. Woodard for their invaluable review inputs.

I would like to thank my computer vision research group for the friendly and conducive environment in which I was able to conduct my research. I thank Nikhil for his timely advice and valuable inputs. A special thanks to Dr. Kanhere for sharing his expertise on the traffic database on which I could analyze my algorithm.

Finally, I wish to thank my friends at Clemson for all the great time I spent with them, especially Pari, Anup, Keyur, Aesha, Harshad, Akshay, Tanzy and Ani.

Table of Contents

List of Figures

Chapter 1

Introduction

Computer vision is the science and application of obtaining models, meaning and control information from visual data. It enables artificial intelligence systems to effectively interact with the real world. The advance of technology has made video acquisition devices better and less costly, thereby increasing the number of applications that can effectively utilize digital video such as:

- Traffic Management: To extract statistics about the traffic information from the cameras and automatically direct traffic flow based on the statistics [9]
- Automobile driver assistance: In lane departure warning systems for trucks and cars that monitor position on the road [2].
- Sports analysis: To track sports action for enhanced broadcasts and also to provide real-time graphics augmentation [11].
- Retail video mining: To track shoppers in retail stores and determining their trajectories for optimal product placement [10].
- Games and gesture recognition: To track human gestures for playing games or

interacting with computers.

- Automated Video Annotation and Retrieval: To automatically annotate temporal sequences with object tags and index them for efficient content-based retrieval of videos [1]

An important application of computer vision is to track the motion of desired objects in a video. The primary aim of tracking is to establish the location and trajectory of the object movement in a captured video. The tracking can be online, during the video capture or offline, in the post processing stage of video analysis.

1.1 Computer vision in road traffic analysis

Computer vision and pattern recognition techniques play an important role in traffic flow monitoring and traffic analysis. The monitoring involves collecting parameters such as volume of traffic, the detection and classification of vehicles, their individual speeds etc. This data can be analyzed to detect lane changes, speed violations, traffic congestions, accidents etc.

There are numerous sensor technologies to acquire the traffic data. Inductive loop detectors are an established technology in the United States and have a well defined zone of detection. However their installation and maintenance invariably delays or disrupts traffic. Non-intrusive sensing techniques involve the use of technologies such as radar and video image processing. Radars have a compact size and are effective for speed measurements, but they lack precision and their output is a very limited description of traffic events. On the other hand, video sensors provide a rich description of the traffic and also have a wide area coverage. Because of their low cost, non-intrusiveness and high effectiveness, they have emerged as a viable tool for automated traffic surveillance. The systems used for automated vision-based traffic monitoring can be broadly classified into two categories: 1) Systems based on localized incident detections, and 2) Systems tracking individual vehicles. Vehicle tracking systems offer more accurate estimates of traffic parameters compared to localized incident detections.

1.2 Previous work

Vehicle tracking is commonly approached in two different ways: Feature-based tracking and Appearance-based tracking.

1.2.1 Feature-based tracking

Feature-based tracking approaches the task of tracking a vehicle by not focusing on the vehicle as a whole, but instead detecting features on the vehicle such as distinguishable points or lines. This is followed by grouping the detected features into potential vehicle objects. Beymer et al. [3] explored this feature point approach by tracking features throughout the video sequence and considering only those features which are tracked consistently from entrance to exit of the detection zone. They then group these features by using motion cues in order to segment the vehicles. Kanhere et al. [8] also implemented a vision-based tracking using stable features, but with a low-angle camera. To tackle the 3D perspective effects that occur at low angles, they employ a 3D perspective mapping from the scene to the image. This mapping identifies a subset of features whose 3D coordinates can be accurately estimated. These features are grouped for subsequent standard feature tracking to monitor traffic flow.

Feature-based approaches have low computational cost and are robust to partial occlusion as some of the features of the vehicle are always visible. However, as a vehicle can have multiple features, constraints have to be imposed to identify features belonging to the same vehicle. Also, in the event of an occlusion, the feature-based approach fails to retrieve the shape of the vehicle.

1.2.2 Appearance-based tracking

In appearance-based tracking, a cascade of classifiers is generally trained to learn the appearance of a vehicle under different deformations. One approach is to perform a density matching based on the vehicle model distribution and the image distribution. The probability density estimates of the object appearance can either be parametric, such as a mixture of gaussians [4], or nonparametric, such as Parzen windows [6]. Chockalingam et al. [4] divide the object into multiple fragments and represent the object by a gaussian mixture model in a joint feature-spatial space. Freedman et al. [14] further combine density matching with shape priors by using PDE-based curve evolutions implemented via level sets.

An alternative technique is to use a model of the desired object in terms of a 2D template of image intensities. The object is then searched for by comparing the search image with the template image. A commonly used tool for comparison is discrete cross-correlation, which identifies correspondences across two frames. Quantitatively, this correspondence or similarity is measured by the SSD distance measure (i.e., squared Euclidean distance),

$$SSD = \sum_{x,y} (I(x + \delta x, y + \delta y) - R(x, y))^2 \tag{1.1}$$

where I is the search image and R is the reference image representing the template, δx and δy are the displacement between the reference image and the search image in the x and y directions, respectively. The summation is over all the pixels of the region of interest. By varying δx and δy, a probability map can be generated which identifies the most probable locations of the template in the search image.

The advantage of template-based approaches are that they embed both spatial and appearance information in the tracking process. However, the primary limitation they suffer from is that they encode the object appearance only from a single viewpoint. As such they do not adapt to changes in the appearance of the object over time and the tracker output gradually drifts away from the target. This constraint is generally handled by choice of an appropriate motion model to transform the search image before performing discrete correlation. For the traffic vehicles which are rigid and compact, the similarity transformation is a good approximation of the motion model. However, this approach still does not completely eliminate the drift problems such as when the object is rotating out of plane. Wang et al. [13] proposed to handle the problem of out-of-plane rotation by a technique they termed as backward correlation. It involves augmenting the standard correlation by performing a motion segmentation and grouping pixels around the target with similar image velocities. Ho et al. [7] proposed an alternative technique to update an object model in a video frame by defining a subspace that continually approximates the appearance of the object in a set of previous frames.

1.3 Tracking approach

In this thesis, it is proposed to overcome the limitations of a single template by instead using an ensemble of templates, where each template encodes the spatial location and shape from a different viewpoint. This view-based template collection approach is tested by choosing to track tractor-trailers in a multi-lane traffic. Tractor-trailers are large and have an uniform shape which helps to easily embed their appearance in a template.

The traffic feed is streamed from a stationary camera positioned high above the road surface. This minimizes the effects of occlusion and spillover and makes it suitable for vehicle tracking and speed measurements. The video is sampled to obtain a succession of video frames which are input to the tracking algorithm. Prior to tracking, a segment of traffic which contains a tractor-trailer movement is identified. Ideally, a segment which has relatively less traffic is preferred for easier template creation. This sparse traffic segment captures the motion of the tractor-trailer from the entrance to the exit of the lane. The video frames in the chosen traffic segment are used as training data to create a template sequence, which will be used in the actual tracking stage. The template sequence contains a manual segmentation of the tractor-trailer from the rest of the traffic during the course of its transit in the lane. Each template in the template sequence represents the tractor-trailer shape at a particular spatial location.

An automated search for these templates is then performed in every video frame for the rest of the traffic video. If a potential match is identified, then the contour of the template is overlaid onto the scene in the video frame to highlight the desired silhouette.

1.4 Outline of this thesis

- Chapter 2 formulates the algorithm and the novel approach of collective template-based tracking proposed in this research.
- Chapter 3 displays the effectiveness of this approach in different conditions.
- Chapter 4 summarizes this work and proposes future additions to this work.

Chapter 2

View-Based Template Tracking Using Template Sequences

This chapter describes the algorithm developed for tracking the motion of tractor-trailers in a multi-lane traffic. The algorithm uses a sequence of view-based templates to model the appearance of a tractor-trailer at different points in the lane.

2.1 Overview of algorithm

The first step in the algorithm is to create a template sequence representing the motion of a tractor-trailer through the lane. After the template sequence is created, the algorithm processes the traffic video to track similar tractor-trailers. The tracking stage starts with background subtraction to identify the non-static objects in the scene. The blobs obtained are then compared for both shape and gradient matching with every element in the template sequence. The best match identifies the template which represents the pose of the tractor-trailer in the input video frame. This frame by frame processing traces the position, trajectory and contour of the vehicle during its transit in the lane.

2.2 Template sequence creation

The approach explored in this thesis is to use the human visual system to manually create view-based templates of a desired vehicle. As the objective of this work is to track tractor-trailers, a portion of the traffic video containing a tractor-trailer is processed for template creation. The traffic segment chosen captures the motion of the vehicle from entrance to the exit of the lane. In each video frame of such a traffic segment, the tractor-trailer is manually cropped from the rest of the scene. The cut image is pasted onto a blank background of the same size as the traffic video frames. The foreground is then thresholded to create a binary template which embodies the shape and spatial location of the tractor-trailer at a particular position in the lane. The process is repeated for every frame in the traffic segment, yielding a template sequence containing binary template images corresponding to the tractor-trailer in different poses.

The number of templates required for this technique is proportional to the choice of the detection zone in the lane, generally between 30-50. Here, their number N is chosen to be 35. The procedure for a single lane is repeated for the second lane creating a separate template sequence and can be extended to all the lanes. Figure 2.1 shows an example creation of an element of the template sequence. The contour of the vehicle is marked by six red edges for cropping. This skeleton is processed to create the template shown adjacent to it.

The input video frame used for the creation of the binary template of the tractor-trailer is also used to compute its gradient magnitude. The idea behind using the gradient magnitude is to identify salient points which would exist in any other

Figure 2.1: Example of a template creation.

similar tractor-trailer. The gradient magnitude image is thresholded to retain pixels of high gradient magnitude. The pair of binary template and the thresholded gradient magnitude image are ANDed together to determine points of high gradient magnitude that correspond only to the tractor-trailer. This gradient pixel set is stored alongside the binary template in the template sequence. Figure 2.2 gives shows the thresholded gradient magnitude of the input video frame shown in Figure 2.1 and the pixel set obtained after ANDing it with the corresponding template.

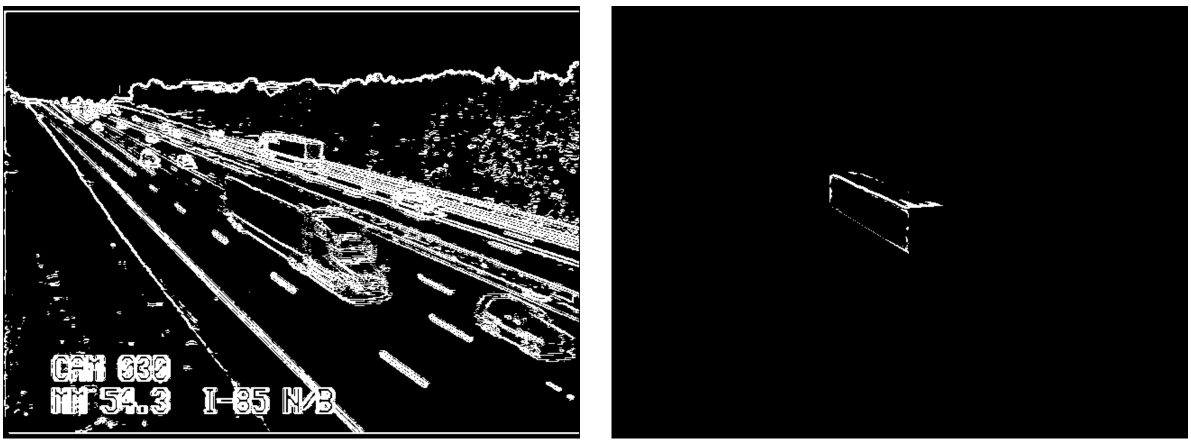


Figure 2.2: Example of gradient magnitude-based salient point extraction. Left: Thresholded gradient magnitude of the input frame. Right: Logical AND with corresponding binary template.

2.3 Background subtraction

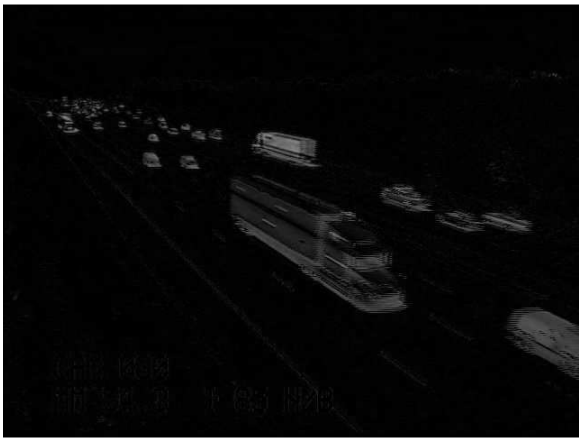
Background subtraction is the first step in identifying the spatial location of moving objects in the input video. It involves the subtraction of the input video frame from a reference frame. Ideally the reference frame contains only a static background which is constant over the range of the video capture. In the current scenario, a frame in which no traffic flows through the desired lanes is chosen as the reference frame.

During tracking, the reference frame is constantly subtracted from each video frame. Figure 2.3 shows an example video frame in the traffic sequence and the corresponding background subtracted image. The background subtracted image is then thresholded for subsequent comparison with the binary templates in the template sequence. The thresholding operation results in clusters of binary data, commonly referred to as blobs. These blobs are the possible candidates for the vehicle being tracked. The blob of motion obtained after thresholding is shown at the bottom in Figure 2.3.

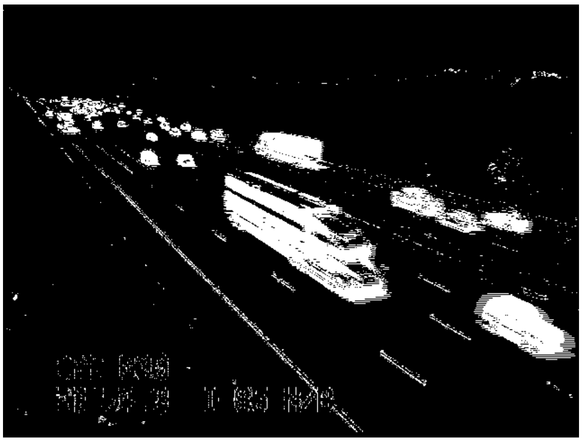
2.4 Template selection

The next step in the tracking process is to probabilistically identify templates corresponding to the various blobs. Two match scores are computed to estimate the best template match for a blob. The first is based on a shape comparison between the blob and all of the templates. The second score is based on detecting high gradient magnitude points in the area covered by the blob which are at an identical location in the thresholded gradient magnitude image stored in the template sequence.

Video Frame



Background subtracted frame



Thresholded background subtracted frame

Figure 2.3: Example of background subtraction and blob generation.

2.4.1 Blob-Template matching

Each blob in a lane is compared with all the templates in the template sequence created for that lane. A match score is computed to determine the degree to which a blob matches the shape of an element in the template sequence. The blob and the template, both of which are binary patches of data, are ANDed to obtain the pixels that are effectively portions of the blob which fall in the area bounded by the template. The match score is obtained by normalizing the number of matching pixels by the number of pixels in the template. It is computed as:

$$Match\% = 100 * \frac{\sum_{x,y} T(x,y) * B(x,y)}{\sum_{x,y} T(x,y)} \tag{2.1}$$

where $T(x, y)$ is the binary template image and $B(x, y)$ is the thresholded gradient magnitude image containing the blobs. The operator * refers to the multiplication of these two images. A high match score is a good indicator of a vehicle at that pose in the input video frame.

Figure 2.4 shows a example plot of the match percentage of a sequence of input frames with various templates in a template sequence of 35 templates. The frames considered are 0337–0372 in a 5000 frame traffic sequence and their images can be seen in the Results section in Figure 3.4. In the course of the vehicle's movement along the lane, the template curves reach a peak match at a unique frame number and drop monotonically on either side. For example, the curve in Figure 2.4 corresponding to template 20 (blue curve) reaches a peak match score of 86% at frame 0356. Hence it can be correctly concluded that there is a vehicle at an identical pose as template 20 in the input video frame 0356. Similar conclusions can be derived from the other curves.

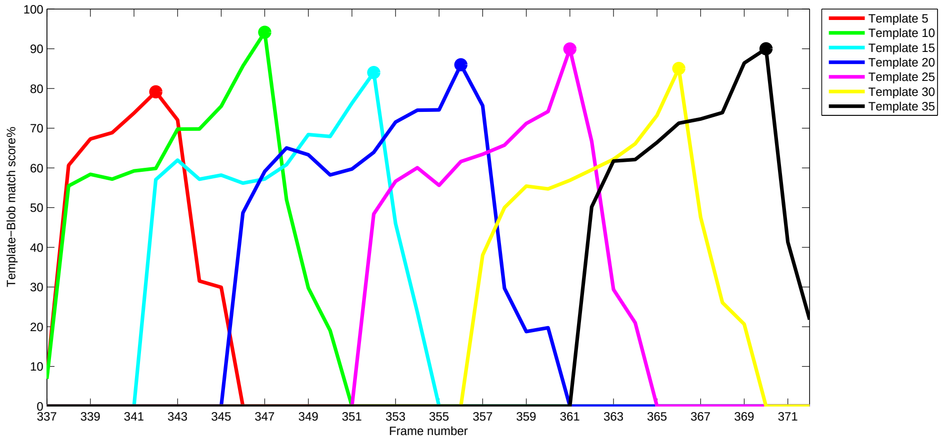


Figure 2.4: Plot of Match% between the template sequence and the blobs in a traffic segment.

2.4.2 Gradient magnitude-based matching

The template chosen in the previous stage marks a possible candidate for the position of the vehicle. In addition to the match score computed earlier, a secondary match score is based on gradient magnitude information. The template is ANDed with the thresholded gradient magnitude of the current input video frame. This operation yields pixels of high gradient magnitude that fall in the vicinity of the template. This pixel set is matched with the training pixel set stored alongside the template during template creation. A match score is computed by normalizing the number of matching pixels by the size of the training pixel set. It is computed as:

$$Match\% = 100 * \frac{\sum_{x,y} G_r(x,y) * G_i(x,y)}{\sum_{x,y} G_r(x,y)} \tag{2.2}$$

where,

$$G_r(x,y) = T(x,y).\{|\nabla R(x,y)| : |\nabla R(x,y)| > threshold\}$$

$$G_i(x,y) = T(x,y).\{|\nabla I(x,y)| : |\nabla I(x,y)| > threshold\}$$

where $R(x,y)$ is the training image from which the template $T(x,y)$ was created, $I(x,y)$ is the input video frame and the thresholds for the computation of $G_r(x,y)$ and $G_i(x,y)$ are manually chosen to a certain common value.

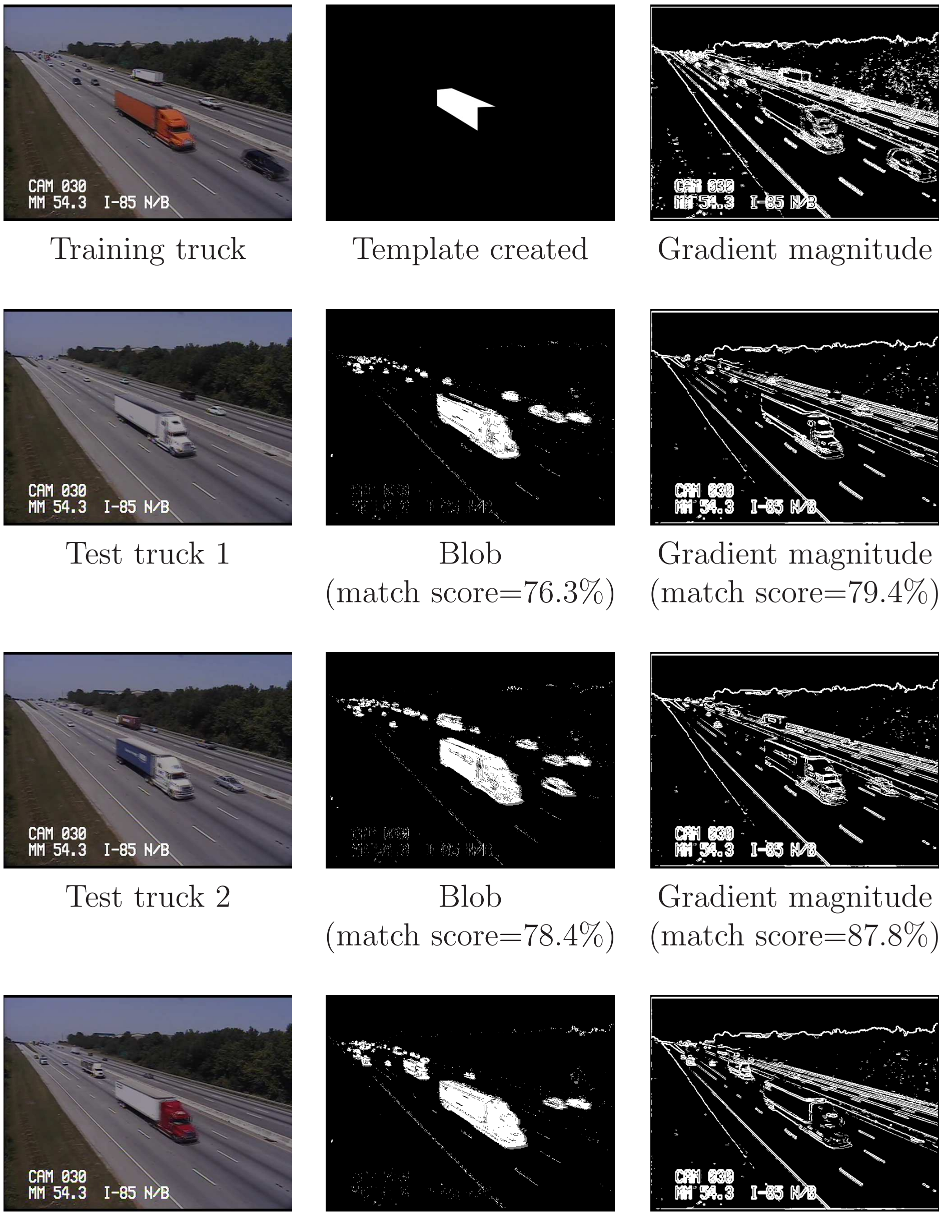
2.5 Template matching summary

The final match score to determine a fit between a blob and a template is computed as a weighted mean of the two match scores computed by blob-template correlation and gradient magnitude-based matching.

Figure 2.5 shows examples of how an element in the template sequence is searched for in the input video frame. The first row shows the tractor-trailer used in the training stage at a specific location in the lane. The template created corresponding to this instance and the gradient magnitude information are shown adjacent in the same row. The succeeding rows give examples of instances in the traffic video sequence containing different tractor-trailers. The blobs obtained after background subtraction and the thresholded gradient magnitudes of the video frame are shown adjacent. The match scores are based on the computation described in sections 2.4.1 and 2.4.2.

2.5.1 Display contour

The template for which the match score is the highest and above a certain threshold is decided to match the vehicle pose in the input video frame. The final step in the tracking process is to overlay the selected template contour onto the input video frame. This highlights the vehicle's contour in the traffic video.



Training truck | Template created | Gradient magnitude

Test truck 1 | Blob (match score=76.3%) | Gradient magnitude (match score=79.4%)

Test truck 2 | Blob (match score=78.4%) | Gradient magnitude (match score=87.8%)

Test truck 3 | Blob (match score=81.6%) | Gradient magnitude (match score=88.4%)

Figure 2.5: Examples of template matching during tracking.

2.6 Algorithm summary

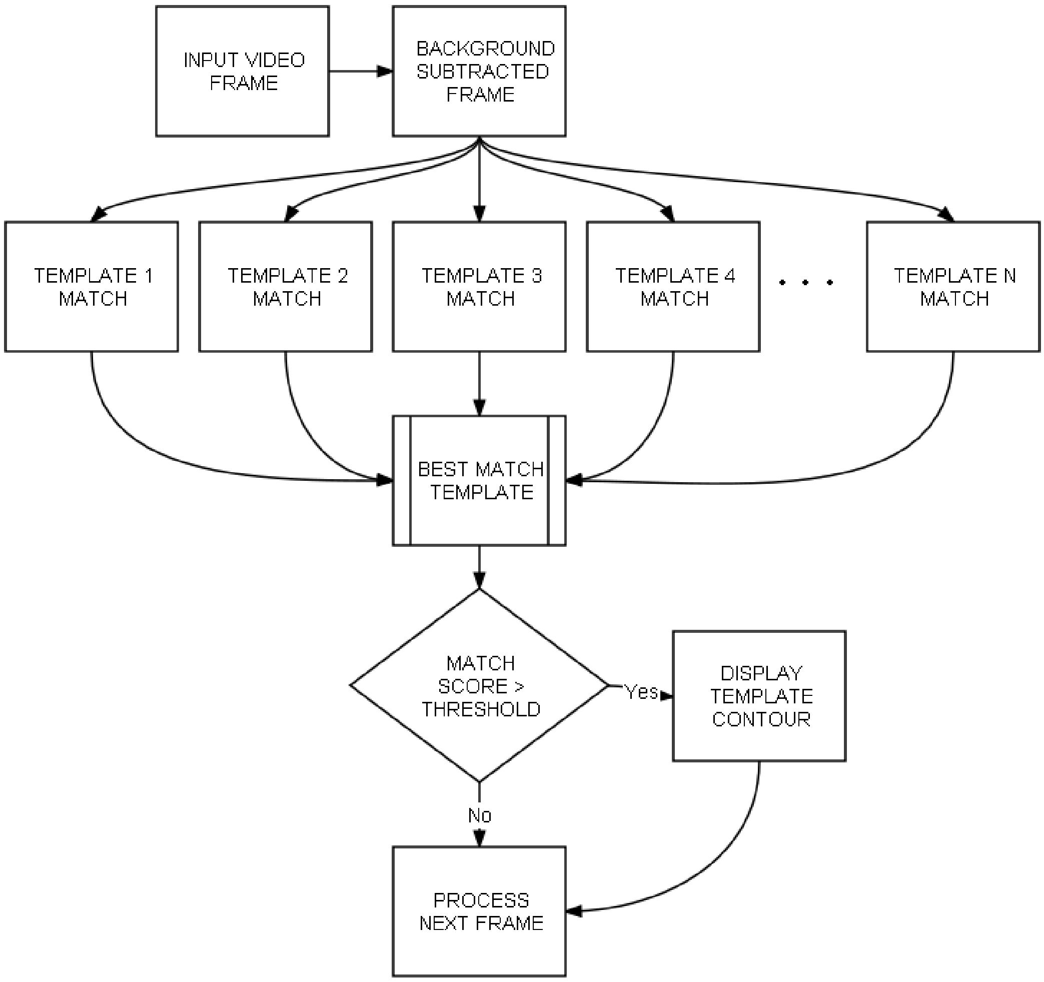


Figure 2.6: Template collection-based tracking algorithm summary.

Chapter 3

Experimental Results

The algorithm was implemented in Visual C++ on an Intel Core 2 Duo 2.0GHz machine. The input traffic video was captured at a specific viewpoint on the interstate highway. The traffic flows in four lanes and the lanes are numbered as Lane 1 being closest to the median and the lane numbers increasing outward. The tractor-trailers move only in Lanes 2 and 3 and their occurrences during the course of the video are shown in Figure 3.1.

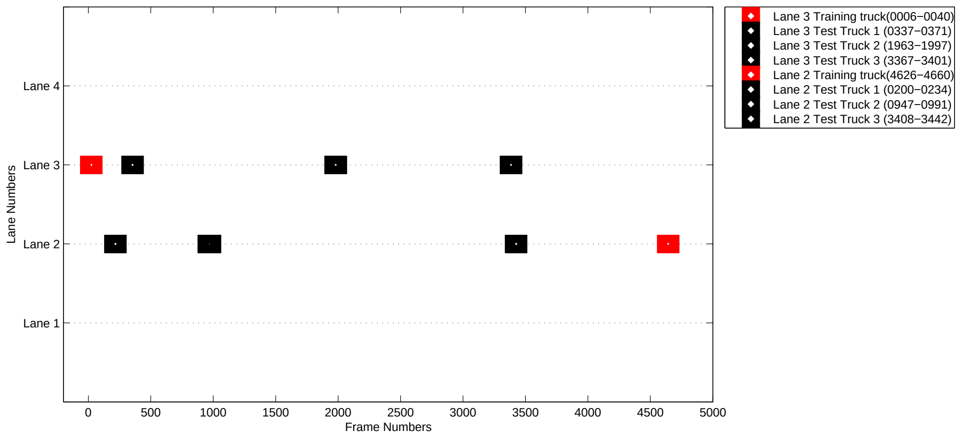


Figure 3.1: Plot depicting traffic flow in the input video.

3.1 Results of the tracker framework

Figure 3.2 lists the template sequence created for Lane 3. Figure 3.3 shows the result of the algorithm on the tractor-trailer used for the training stage. Figures 3.4, 3.5 and 3.6 list the results for tractor-trailers different from the tractor-trailer used in the training stage. Similarly, Figure 3.7 lists the template sequence for the Lane 2. Figure 3.8 shows the result of the algorithm on the tractor-trailer used for the training stage. Figures 3.9, 3.10 and 3.11 list the results for other tractor-trailers in Lane 2.

3.2 False positives

While the tracker output is accurate for frames involving the desired tractor-trailer movement, it also falsely detects and marks certain events which do not correspond to a tractor-trailer movement. Such detections are termed here as false positives. Figure 3.12 shows the frames for which such false positives were detected in the course of processing the entire 5000 frames.

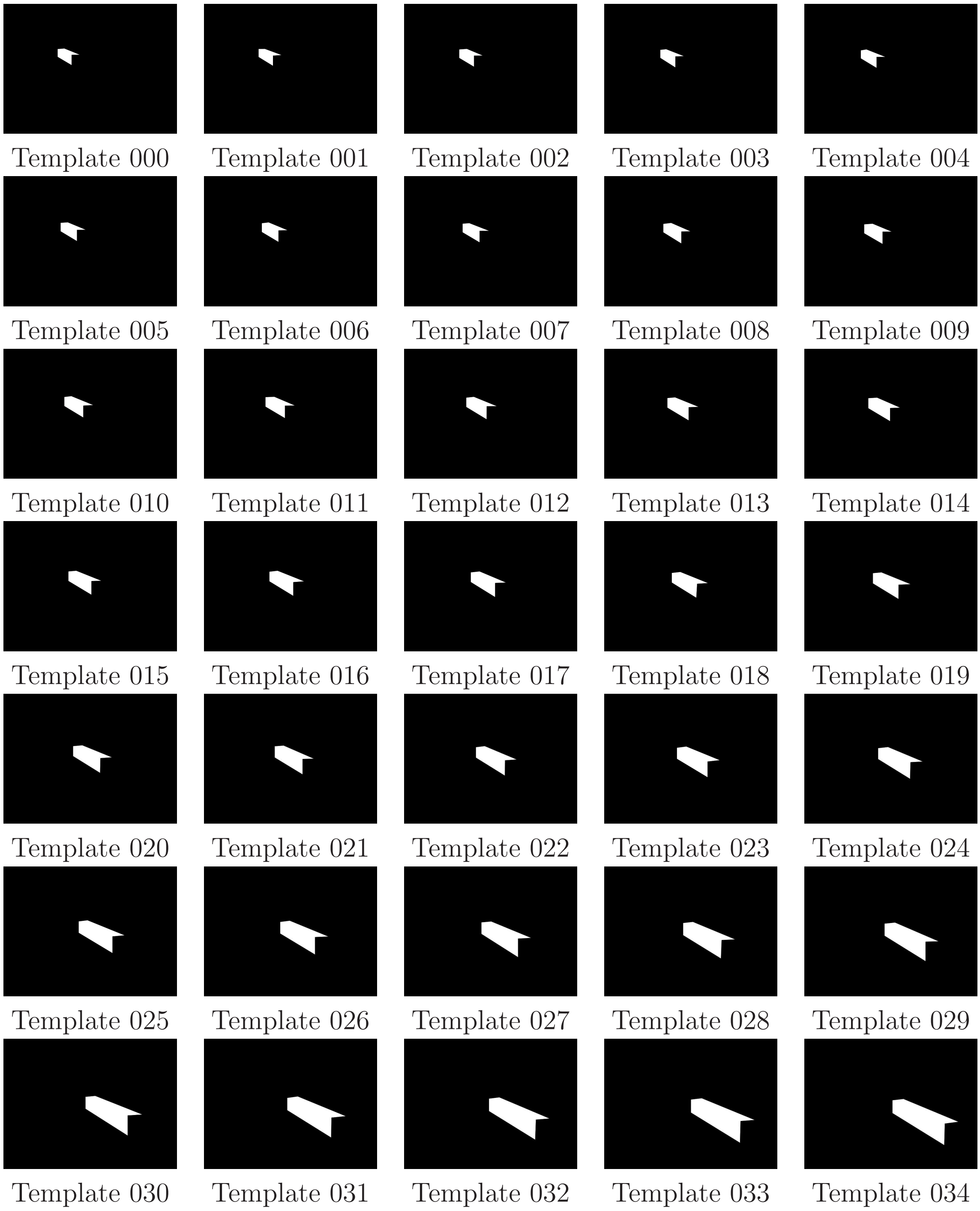


Figure 3.2: Template sequence for Lane 3.

Figure 3.3: Results of algorithm on Lane 3 for tracking the same tractor-trailer as used for template generation.

Figure 3.4: Results of algorithm on Lane 3 for tracking a different tractor-trailer.

Figure 3.5: Results of algorithm on Lane 3 for tracking another tractor-trailer.

Figure 3.6: Results of algorithm on Lane 3 for tracking yet another tractor-trailer.

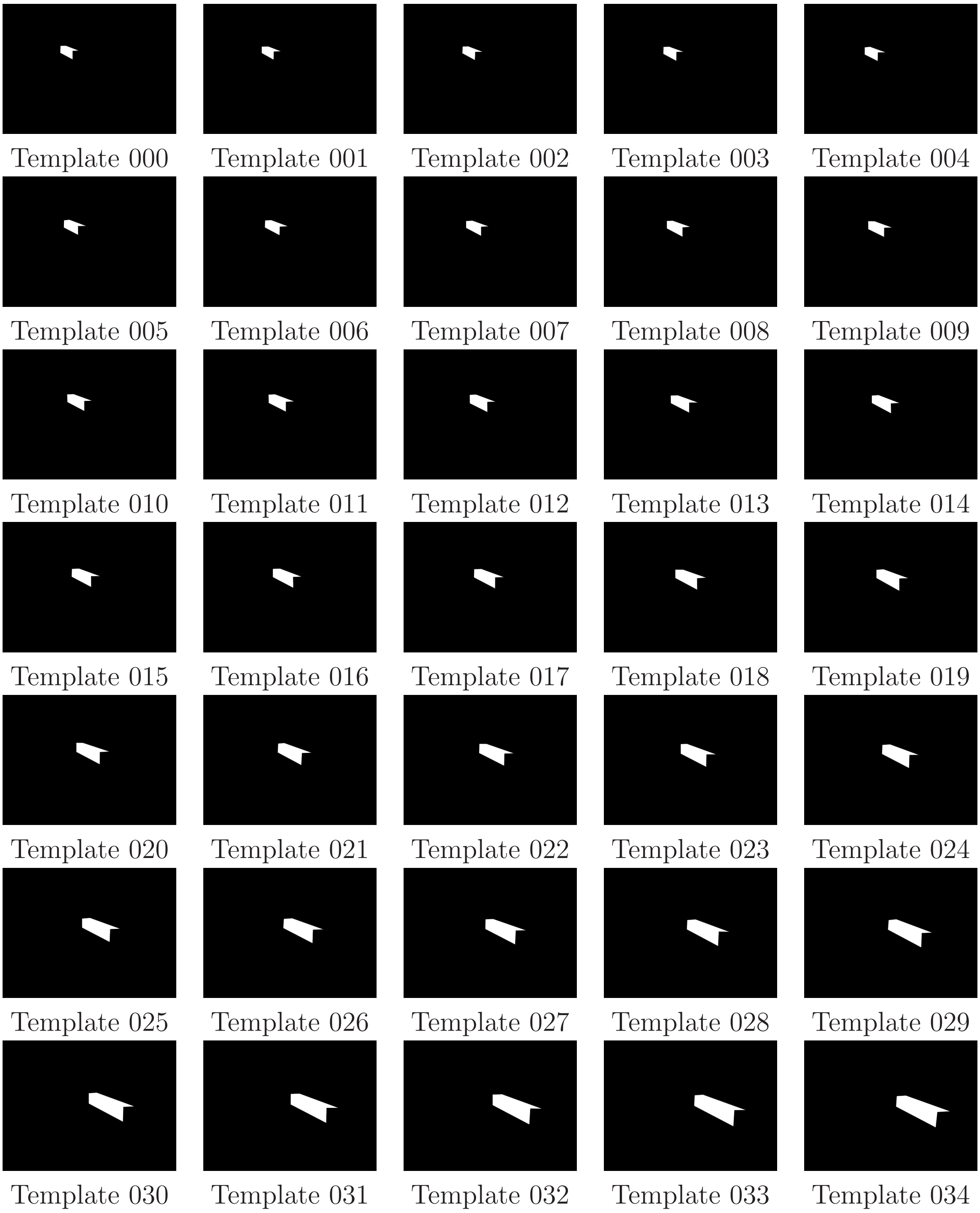


Figure 3.7: Template sequence for Lane 2.

Figure 3.8: Results of algorithm on Lane 2 for tracking the same tractor-trailer as used for template generation.

Figure 3.9: Results of algorithm on Lane 2 for tracking a different tractor-trailer.

Figure 3.10: Results of algorithm on Lane 2 for tracking another tractor-trailer.

Figure 3.11: Results of algorithm on Lane 2 for tracking yet another tractor-trailer.

Frame 0625 Frame 2554

Frame 2925 Frame 2926 Frame 2927 Frame 2928 Frame 2929

Frame 2930 Frame 2931 Frame 2932 Frame 2933 Frame 2934

Frame 2935 Frame 2936 Frame 2937 Frame 2938 Frame 2939

Frame 3123 Frame 3124 Frame 3125 Frame 3126 Frame 3127

Frame 3128 Frame 3129 Frame 3130 Frame 3131 Frame 3132

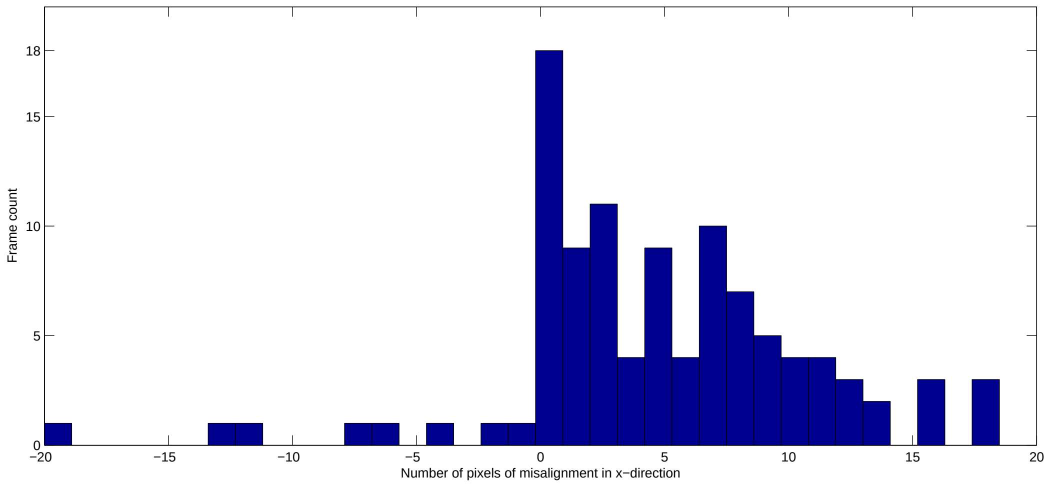
Frame 3133 Frame 3134 Frame 3135 Frame 3136 Frame 3137

Frame 3138 Frame 3139 Frame 3140 Frame 3141 Frame 3142

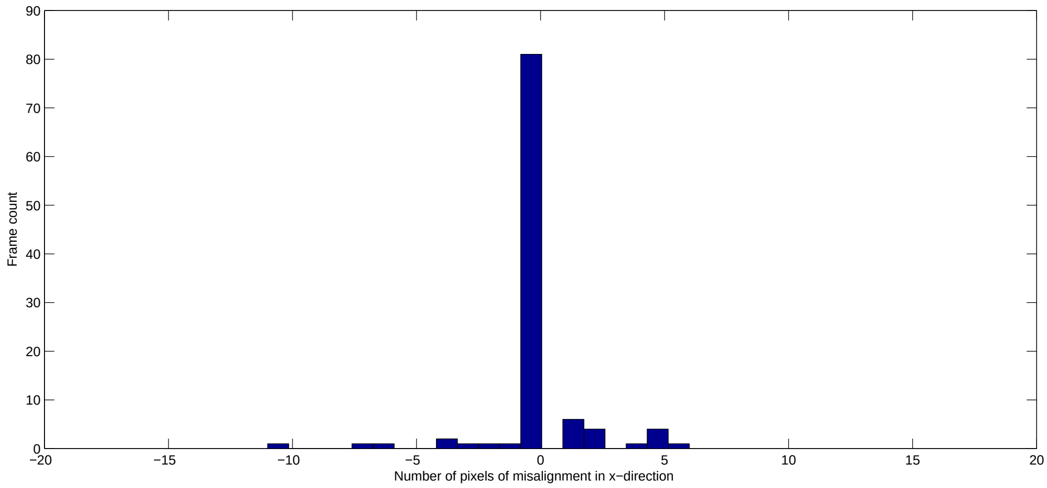
Figure 3.12: List of false positive detections during the entire tracking sequence.

3.3 Influence of gradient magnitude-based matching

The decision on a suitable template match for a blob in the background subtracted frame is influenced by the salient points. This section explains the need for salient point matching to supplement the match based on correlation between the blob and an element in the template sequence. Figure 3.13 shows a histogram depicting the misalignment of the final displayed contour relative to the ground truth if salient points were not used. As seen in the figure, the addition of salient points to the matching process greatly reduces the misalignment error. It might seem that we might as well discard correlation based match score and perform matching based solely on the salient points. However, this approach is flawed as it is seen to increase the number of false positives over the entire training sequence.



Only template correlation-based matching



Only salient points-based matching

Figure 3.13: Effect of salient points on misalignment.

Chapter 4

Conclusion

In this research, a view-based template collection method for tracking vehicles was proposed. The templates were manually created using a sparse segment of traffic containing the desired class of vehicles (tractor-trailers). The template collection comprised a sequence of vehicle poses over the course of the vehicle's movement in a multi-lane traffic. This template sequence was then used to scan the rest of the traffic video, to track the movement of similar class of vehicles. During tracking, every frame in the video sequence was subtracted from a reference frame to identify the moving objects. The resulting blobs were then matched against the template sequence to identify a possible candidate for a vehicle instance. If a potential match exists and the matching criterion is above a certain threshold, then the template contour is overlaid on the current video frame. This displayed contour in effect, traces the contour of the vehicle in the input video.

The tracker was tested to track tractor-trailer movement in a two lane traffic at a particular viewpoint on the interstate highway. The tracker results were accurate for all the tractor-trailers present in the entire video frame sequence. However, there were a few false negatives where the tracker completely missed the tractor-trailer towards the exit of the lane.

These results encourage consideration of future work in the directions of:

- More robust verification of the tracking algorithm in the presence of occlusion and heavy traffic.
- Tracking of multiple classes of vehicles such as passenger cars, buses etc.
- Condensing the set of templates so as to have minimal redundancy in the template sequence, while maintaining tracking accuracy.
- Using level sets to incorporate manually created shape priors such as by Cremers [5].
- Incorporating level set techniques such as Karl-Shi [12] for faster tracking.

Bibliography

[1] A. Anjulan and N. Canagarajah. Object based video retrieval with local region tracking. *Signal Processing: Image Communication*, 22:607–621, Aug. 2007.

[2] S. Avidan. Support vector tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, pages 1064–1072, Aug. 2004.

[3] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real time computer vision system for measuring traffic parameters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 495–501, 1997.

[4] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *Proceedings of the International Conference on Computer Vision*, Oct. 2009.

[5] D. Cremers. Dynamical statistical shape priors for level set-based tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1262–1273, 2006.

[6] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. In *Proceedings of the IEEE*, pages 1151–1163, July 2002.

[7] J. Ho, K.-C. Lee, M.-H. Yang, and D. Kriegman. Visual tracking using learned linear subspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 782–789, June 2004.

[8] N. K. Kanhere and S. T. Birchfield. Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):148–160, Mar. 2008.

[9] N. K. Kanhere, S. J. Pundlik, and S. T. Birchfield. Vehicle segmentation and tracking from a low-angle off-axis camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1152–1157, June 2005.

[10] A. Leykin. *Visual Human Tracking and Group Activity Analysis: A Video Mining System for Retail Marketing*. PhD thesis, Dept. of Computer Science, Indiana University, 2007.

[11] G. Pingali, Y. Jean, and I. Carlbom. Real-time tracking for enhanced tennis broadcasts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 0, pages 260–260, June 1998.

[12] Y. Shi and W. C. Karl. Real-time tracking using level sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 34–41, June 2005.

[13] X. Wang. Forward-backward correlation for template-based tracking. Master's thesis, Clemson University, May 2006.

[14] T. Zhang and D. Freedman. Tracking objects using density matching and shape priors. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1056–1062, Oct. 2003.