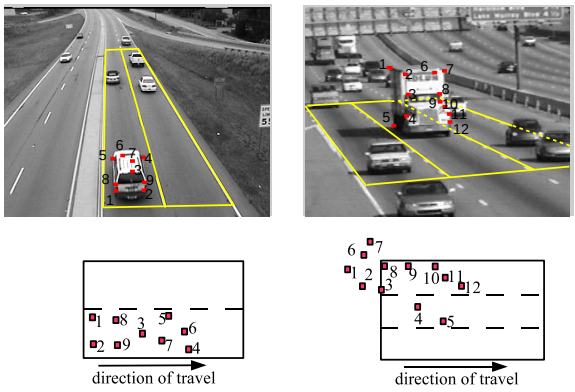
IEEE Conference on Computer Vision and Pattern Recognition (CVPR) San Diego, California, June 2005

Vehicle Segmentation and Tracking from a Low-Angle Off-Axis Camera

Neeraj K. Kanhere Shrinivas J. Pundlik Stanley T. Birchfield
Electrical and Computer Engineering Department
Clemson University, Clemson, SC 29634
{nkanher, spundli, stb}@clemson.edu

Abstract

We present a novel method for visually monitoring a highway when the camera is relatively low to the ground and on the side of the road. In such a case, occlusion and the perspective effects due to the heights of the vehicles cannot be ignored. Features are detected and tracked throughout the image sequence, and then grouped together using a multi-level homography, which is an extension of the standard homography to the low-angle situation. We derive a concept called the relative height constraint that makes it possible to estimate the 3D height of feature points on the vehicles from a single camera, a key part of the technique. Experimental results on several different highways demonstrate the system's ability to successfully segment and track vehicles at low angles, even in the presence of severe occlusion and significant perspective changes.



high-angle on-axis low-angle off-axis

Figure 1: High-angle (left) versus low-angle (right) scenarios compared. In the later, a single homography is not sufficient, because a vehicle may map to multiple lanes.

1 Introduction

Automatic traffic monitoring is becoming increasingly important as our need for highly reliable transportation systems grows. Among the many existing technologies, vision-based systems are emerging as an attractive alternative due to their ease of installation and operation, as well as their ability in principle to capture a rich description of the traffic parameters, including not only vehicle count and average speed but also parameters such as trajectories, queue length, and classification. Existing commercial solutions, however, have only begun to tap into this potential, with long setup times, expensive equipment, and impoverished extracted descriptions still common.

Much of the research on improving the capability of vision-based systems has focused upon the situation in which the camera is looking down on the road from a high vantage point. When the camera is high off the ground, the problem is greatly simplified because the problems of occlusion and temporal change in visual appearance are largely non-existent. It is not always feasible, however, to place the camera at a high vantage point. For example, to gain knowledge about the impact of, say, building a shopping center on neighboring roads and intersections, it is common to place a camera on a portable tripod on the side of the road to gather data about the current traffic patterns. The transient nature of such a study precludes expensive mounting equipment and strategic placement.

Figure 1 illustrates the difference between the two situations. When the camera is high above the ground and near the center of the road, a homography can be defined to map the road surface to the image plane, and the height of vehicles can be safely ignored because their appearance does not change significantly over time. In contrast, when the camera is at a low angle and/or off-centered from the road, the vehicles' height causes significant occlusion. A single homography will not suffice, as feature points on a vehicle may map to multiple lanes.

In this paper we present a method to segment and track vehicles in low-angle situations. The technique is based upon tracking feature points throughout a block of frames from the image sequence, then grouping those features using several motion-related cues. We present a novel combination of background subtraction, extending plumb lines, and multi-level homography to determine the height of features using a constraint we call the *relative height con-*

straint. In this manner the features are grouped, assigning one group per vehicle. We present experimental results on several sequences demonstrating the ability of the algorithm to successfully group vehicles even in the presence of severe occlusion.

2 Previous work

The past ten years has seen a number of systems proposed for the detection, segmentation, and tracking of vehicles on highways, most of which assume that the camera is at a high angle. When the vehicles are well-separated, background differencing is a powerful technique that has been explored in [5, 3, 11]. Active contour models with occlusion reasoning was presented in [10], with a primary limitation being the situation when the vehicles enter the scene already partially occluded. 3D wireframe models have also been successfully used [15, 9, 8, 13, 6, 4], but they require accurate models for many different types of vehicles.

Among these approaches, the one that is the most similar in spirit to our work is that of Beymer et al. [2]. Features are tracked throughout the sequence using Lucas-Kanade, then the features are grouped using motion cues to segment the vehicles. The distance between pairs of features, as well as their velocities, are calculated in a fixed world coordinate system using a single homography between it and the image plane. As such, it is applicable only to high-angle situations. Our work can be viewed as an extension of theirs to the low-angle scenario.

To our knowledge, the only other existing technique for the low-angle situation is that of Kamijo et al. [7]. In their work, the image is divided into 8×8 pixel blocks, and a spatio-temporal Markov random field (ST-MRF) is used to update an object map using the current and previous image. Motion vectors for each block are calculated, and the object map is determined by minimizing a functional combining the number of overlapping pixels, the amount of texture correlation, and the neighborhood proximity. The algorithm does not yield 3D information about vehicle trajectories in the world coordinate system, and to achieve accurate results it is run on the sequence in reverse so that vehicles recede from the camera.

3 Height estimation

As mentioned previously, the vehicle heights play an important role in the low-angle situation and cannot be ignored. This section describes our method for estimating the height of features on the vehicles.

3.1 Tracking features

Feature points are automatically selected and tracked using the KLT (Kanade-Lucas-Tomasi) feature tracker [1], which computes the displacement $\mathbf{d}$ that minimizes the sum of squared differences between consecutive image frames I and J:

$$\iint_W \left[I(\mathbf{x} - \frac{\mathbf{d}}{2}) - J(\mathbf{x} + \frac{\mathbf{d}}{2}) \right]^2 d\mathbf{x},$$

where W is a window of pixels around the feature point. This nonlinear error is minimized by repeatedly solving its linearized version:

$$Z\mathbf{d} = \mathbf{e},$$

where

$$\begin{aligned} Z &= \sum_{\mathbf{x} \in W} \mathbf{g}(\mathbf{x}) \mathbf{g}^T(\mathbf{x}) \\ \mathbf{e} &= \sum_{\mathbf{x} \in W} \mathbf{g}(\mathbf{x}) [I(\mathbf{x}) - J(\mathbf{x})], \end{aligned}$$

and $\mathbf{g}(\mathbf{x}) = \partial \frac{I(\mathbf{x}) + J(\mathbf{x})}{2} / \partial \mathbf{x}$ is the spatial gradient of the average image. These equations are identical to the standard Lucas-Kanade equations [14] but are symmetric with respect to the two images. As in [14], features are automatically selected as those points in the image for which both eigenvalues of Z are greater than a minimum threshold.

Features are tracked throughout a block of n image frames, overlapping with the previous block by $\frac{n}{2}$ frames, where n is determined by the average speed of the vehicles and the placement of the camera with respect to the road. The processing described in this and the following sections is performed on the features tracked throughout a single block. Correspondence between feature groups is established between blocks based upon proximity.

3.2 Multi-level homography

If the road can be approximated as a plane, then a point (u, v) on the image is related to the world coordinates (x, y, z) through a homography:

$$H_z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix},$$

where w is an arbitrary non-zero constant, and the z axis is perpendicular to the road. For a point on the road surface, $z = 0$.

When the camera is high above the ground, all the points on the vehicles can be assumed close to zero height, and this equation is sufficient to enable Euclidean reasoning about the motion of points in the image. At a low angle, however, the non-zero heights cannot be ignored. To handle this situation we employ two homographies, H_0 which maps the

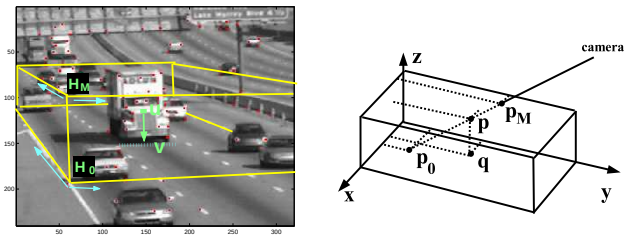


Figure 2: Multi-layer homography. If the pixel coordinates **u** and **v** are known of the projections of two points **p** and **q**, respectively, one of which rests on the road surface directly below the other, then the height of the other point can be computed.

image plane to the surface of the road and H_M which maps the image plane to a plane parallel to and above the road at a reference height M. See Figure 2. An image point (u, v) with height z is then mapped to the world point through

$$\left(\frac{z}{M}H_M + \left(1 - \frac{z}{M}\right)H_0\right)\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix}.$$

For convenience, we define the function f_z as the mapping between the homogeneous coordinates of a pixel and the 3D coordinates of its preimage at height z: $f_z(\mathbf{u}) = \mathbf{p}$, where $\mathbf{u} = [\,u \quad v \quad 1\,]^T$ and $\mathbf{p} = [\,x \quad y \quad z\,]^T$.

3.3 Using a single frame

Assume we know the image coordinates **u** of the projection of a point **p**, as shown in Figure 2. Let us drop a plumb line from **p** to get the point **q** directly below it resting upon the road surface. Suppose we also know the image coordinates **v** of the projection of **q**. (We will address the problem of obtaining the coordinates **v** in a moment.)

Since **q** lies on the ground plane, its coordinates can be computed as $f_0(\mathbf{v})$. If the point **p** were at one of the two extreme heights its coordinates would be $\mathbf{p}_0 = f_0(\mathbf{u})$ or $\mathbf{p}_M = f_M(\mathbf{u})$, respectively. Its true location, $\mathbf{p} = f_z(\mathbf{u})$, however, is unknown since its height z is unknown. This problem is solved by noticing from Figure 2 that the point **p** lies on the line joining $\mathbf{p}_0$ and $\mathbf{p}_M$, described by

$$\mathbf{p} = \mathbf{p}_0 + (\mathbf{p}_M - \mathbf{p}_0)\alpha,$$

where $\alpha \in \Re$ is the unknown fractional distance of **p** along the line. Breaking this equation into components yields

$$\begin{aligned} x &= x_0 + (x_M - x_0)\alpha \\ y &= y_0 + (y_M - y_0)\alpha \\ z &= z_0 + (z_M - z_0)\alpha \end{aligned}$$

which is a set of three equations with, in general, four unknowns (α, x, y, and z). In our case we know that **p** is directly above **q**, so their x and y coordinates are identical, providing two additional constraints. This overconstrained system can be solved for α, which can then be used to find the height z of **p**. To do this we minimize the error $\| \mathbf{p}' - \mathbf{q}' \|^2$, where $\mathbf{p}'$ and $\mathbf{q}'$ are 2×1 vectors containing the x and y coordinates of **p** and **q**. Algebraic manipulation yields

$$\alpha = \frac{(\mathbf{q}' - \mathbf{p}'_0)^T \Delta}{\Delta^T \Delta},$$

where $\Delta = \mathbf{p}_0 - \mathbf{p}_M$.

The coordinates of **v** are obtained automatically by background subtraction. Each feature point **u** that lies in a foreground region is projected vertically down in the image to the transition between the foreground and background regions, yielding the point **v**. This computation is repeated for every frame in the block, yielding n estimates for the image distance between **u** and **v** for a given feature point **u**.

When there is no occlusion, such a simple procedure alone would be accurate. When a vehicle is occluded, however, the point **v** may be the bottom of another vehicle, and thus **q** may not be directly below **p**. See Figure 3. To handle this problem, the algorithm distinguishes between stable features and unstable features. Stable features are declared as those for which the mean and variance of the estimated height z of the feature is small: $\mu(z) < \delta_1$ and $\sigma^2(z) < \delta_2$, where δ_1 and δ_2 are thresholds. Stable features are used in the next subsection to determine the height of the remaining, unstable features.

Why does this technique work? First, in the low-angle situation, many of the feature points lie on unoccluded, vertical surfaces of the vehicles, which is exactly the assumption being used. Secondly, features for which the assumption is violated will be declared unstable due to their high mean or variance. (Note that an incorrect height estimate will usually be higher than the true value, so that heights near the road surface are more reliable.) Thirdly, even for the stable features, the assumption does not need to hold for the entire block of image frames, but only for a small subset of them. Finally, only a small number of stable features are actually needed in practice, typically one per vehicle.

3.4 Refinement of height estimates

Suppose we have two features that are tracked from locations **u** and **v** in one image frame to **u**′ and **v**′ in another (not necessarily consecutive) image frame. (In this subsection **v** is another feature, not the ground point below **u**.) Their possible preimages at the extreme heights are given in the two frames, respectively, by $\mathbf{p}_z = f_z(\mathbf{u})$ and $\mathbf{q}_z = f_z(\mathbf{v})$, and by $\mathbf{p}'_z = f_z(\mathbf{u}')$ and $\mathbf{q}'_z = f_z(\mathbf{v}')$, for $z \in \Re$. Suppose **v** is a stable feature, so we know its actual preimage **q**, and from these data we wish to estimate the preimage **p** of **u**.

Figure 3: Left: Occlusion causes a wrong estimate for $\mathbf{v}$ and the height of the feature. Right: After five frames the occluding vehicle changes lanes, enabling an accurate height estimation.

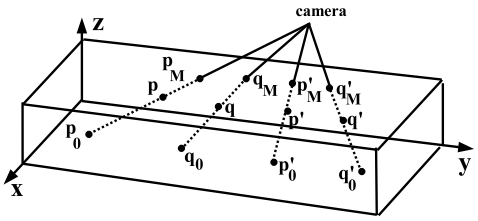


Figure 4: Refining height using stable feature.

From Figure 4 it is clear that, as before, all the possible preimages for a given point are collinear, leading to equations for $\mathbf{p}$, $\mathbf{p}'$, $\mathbf{q}$, and $\mathbf{q}'$, as before. If $\mathbf{p}$ and $\mathbf{q}$ are points on the same rigid vehicle that is only translating, then the motion vectors of the two points are the same: $\mathbf{p}' - \mathbf{p} = \mathbf{q}' - \mathbf{q}$. If we further assume that the road is horizontally flat, then the z component of $\mathbf{p}$ and $\mathbf{p}'$, as well as that of $\mathbf{q}$ and $\mathbf{q}'$, are equal. These assumptions lead to the following equation, which we call the *relative height constraint*, relating the heights of the two features:

$$z_p = \frac{\Delta y_{q_0} - \Delta y_{p_0}}{\Delta y_{p_M} - \Delta y_{p_0}} M + \frac{\Delta y_{q_M} - \Delta y_{q_0}}{\Delta y_{p_M} - \Delta y_{p_0}} z_q,$$

where $\Delta y_{q_0} = y_{q_0} - y'_{q_0}$, y_{q_0} is the yth component of $\mathbf{q}_0$, and so on. (This equation also holds for the x components.) We minimize the deviation from our assumptions over all frames in the block:

$$\epsilon_{p,q} = \sum_{i=2}^{n} \parallel (\mathbf{p}' - \mathbf{p}) - (\mathbf{q}' - \mathbf{q}) \parallel^2,$$

where $\mathbf{p}$ and $\mathbf{q}$ are the coordinates of the points in the first frame, and $\mathbf{p}'$ and $\mathbf{q}'$ are their coordinates in frame i. This error function is a quadratic surface in z_p and z_q with a minimum at

$$z_p = M \frac{\sum_{i=2}^{N} (\Delta\mathbf{q} - \Delta\mathbf{p}_0)^T (\Delta\mathbf{p}_M - \Delta\mathbf{p}_0)}{\sum_{i=2}^{N} (\Delta\mathbf{p}_M - \Delta\mathbf{p}_0)^T (\Delta\mathbf{p}_M - \Delta\mathbf{p}_0)}$$

when z_q is known, where $\Delta\mathbf{q} = \mathbf{q}' - \mathbf{q}$, $\Delta\mathbf{p}_0 = \mathbf{p}'_0 - \mathbf{p}_0$, and $\Delta\mathbf{p}_M = \mathbf{p}'_M - \mathbf{p}_M$. A typical example of this quadratic surface is plotted in Figure 5.

For any given unstable feature $\mathbf{p}$ we computes its height and hence its 3D world coordinates using each stable feature $\mathbf{q}$. Among all possible $\mathbf{q}$ we use the estimate which gives the lowest absolute trajectory error $\epsilon_{p,q}$ weighted by the Euclidean distance in x and y coordinates. This error is used in the next section to group the features.

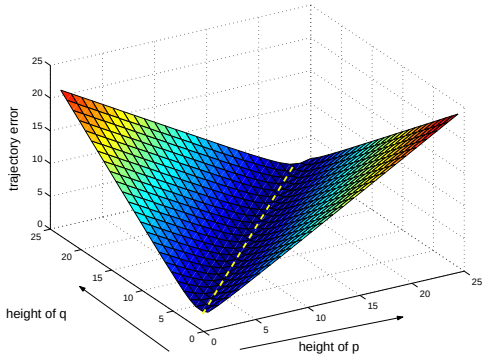


Figure 5: A typical example of the quadratic error surface associated with the relative height constraint.

4 Multiple cues for grouping

The features are grouped using normalized cuts [12]. We form the affinity matrix A as $A = A_B A_D A_E$, where $A_B(i,j) = e^{-N_b(i,j)/\alpha_B}$ measures the connectivity between features, with $N_b(i,j)$ the number of background pixels on the line connecting features i and j. This background-content cue effectively contributes to the segmentation when vehicles are disjoint. To handle occlusion, $A_D(i,j) = e^{-\epsilon_{ij}/\alpha_D}$ measures the trajectory error as defined in the previous section, while $A_E(i,j) = e^{-e_{ij}/\alpha_E}$ measures the Euclidean distance (in x and y) between the features in the world coordinate system. The parameters α_B, α_D, and α_E, as well as the normalized cut threshold τ, are determined by the type of image sequence.

After applying normalized cuts, a post-processing step removes groups that have less than a minimum number of features, and it enforces a minimum average feature height in each group. This step is necessary because, due to approximate calibration and tracking errors, a few features will have height estimates that deviate considerably from their true value.

5 Experimental Results

The proposed algorithm was tested on three grayscale image sequences, each containing 1200 frames. The videos were captured by a 30 Hz camera placed on an approximately 9 m pole on the side of the road and digitized at 320×240 resolution. No preprocessing was done to suppress shadows or to stabilize occasional camera jitter. For each sequence, two homographies were defined by hand on the generated background image, which was computed by averaging several images over the sequence.

Several frames of the first sequence are shown in Figure 6. The features tracked throughout the block are shown, with the color of each feature indicating its group. No attempt has been made to use unique colors, so groups of the same color that are obviously separated in the image are detected as separate groups by the algorithm.

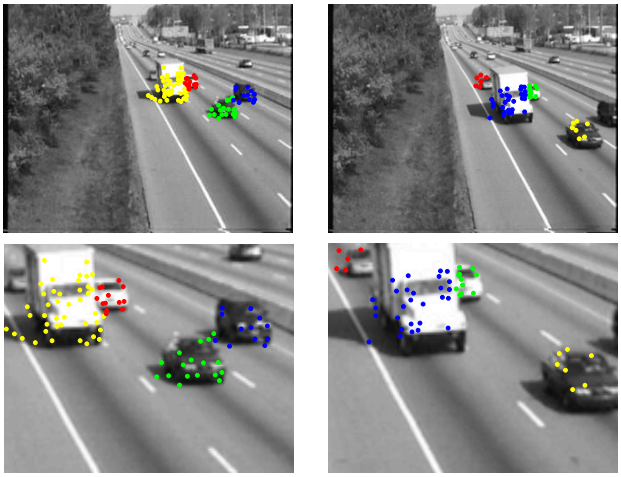
In Figure 6 the algorithm successfully groups features in the presence of severe occlusion, as seen with the trucks in the first three columns. In the third column, the yellow group of features just left of the big green group actually belongs to a small car just behind the large truck and thus were grouped successfully, although the car is not visible at the resolution shown here. Also notice that the lone yellow feature at the back of the flatbed truck in the second column has been successfully grouped with the rest of the truck despite its relatively large distance from the other features.

In the second and third sequences, shown in Figures 7 and 8, respectively, the algorithm was also able to correctly segment vehicles in the presence of occlusion. As in the first sequence, the lone green feature at the back of the large truck (Figure 8) was grouped successfully. One problem with these results is the grouping of feature points on the shadows of vehicles. More sophisticated processing will be necessary to remove these mistakes.

Table 1 provides the results of the algorithm on the three sequences, along with the parameters used. The segmentation accuracy ranged from 86% on the most challenging sequence to 98% on the easiest sequence, with at most $5/1200 = .004$ false positives per image frame on average. (Of the actual vehicles, the number of trucks in the sequences was 9, 13, and 5, respectively, all correctly segmented.) These results compare favorably with those of the only other low-angle system

6 Conclusion

Most approaches to segmenting and tracking vehicles on roads assume that the camera is high above the ground, thus simplifying the problem. We have presented a technique that works when the camera is at a low angle with respect to the ground and/or is on the side of the road, in which case occlusion and the height of vehicles cannot be ignored.



frames 25-35 frames 30-40

Figure 7: Two frames from the second sequence (top), with zoomed displays (bottom) for greater clarity.

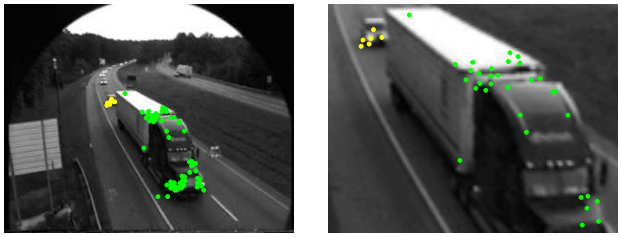
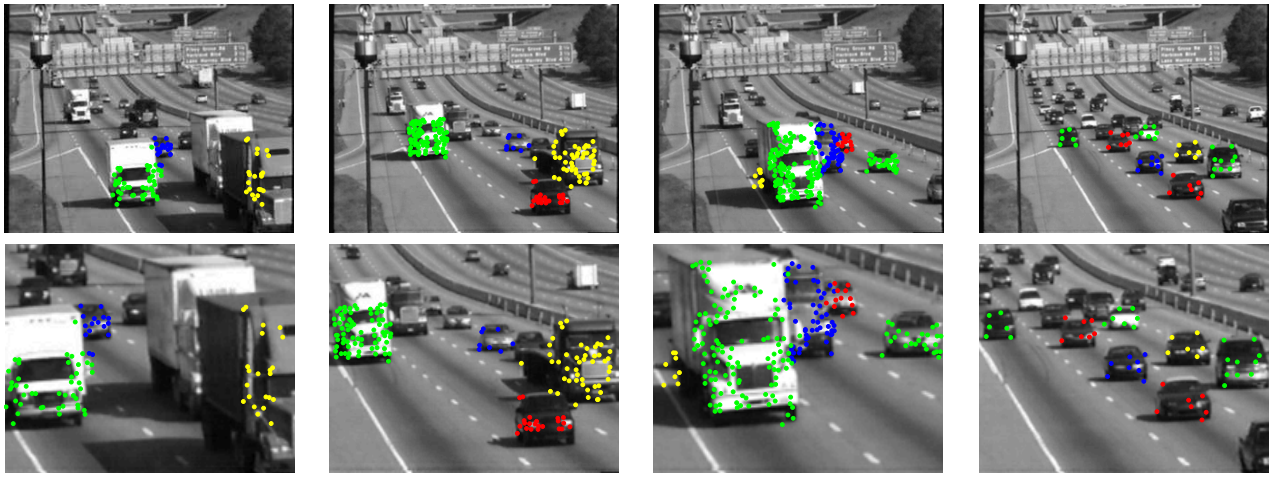


Figure 8: One frame from the third sequence (left), block 60-70, with zoomed display (right) for greater clarity.

seq.	α_B	α_D	α_E	τ	accuracy	FP
1	1	10	10	0.75	86% (97/113)	3
2	1	5	5	0.6	90% (113/125)	2
3	1	25	25	0.5	98% (47/48)	5

Table 1: Parameters used for processing the three sequences, along with the results. The penultimate column gives the segmentation accuracy as the number of vehicles correctly segmented divided by the actual number of vehicles. The last column gives the number of false positives (FP), i.e., the number of feature groups that did not correspond to actual vehicles.



frames 100-129 frames 150-179 frames 190-209 frames 510-529

Figure 6: Four frames from the first sequence (top), with zoomed displays (bottom) for greater clarity. Each feature's color indicates its group. Below the images are the frame numbers in the processed block of frames used to generate the results.

Our approach is based upon grouping tracked features using several motion-related cues. A novel part of the technique is the estimation of the height of features using a combination of background subtraction, plumb lines, and multi-level homography, employing a constraint we derive called the relative height constraint. Experimental results on several image sequences show the ability of the algorithm to handle the low-angle situation, including severe occlusion. Future work should be aimed at reducing the effects of shadows, incorporating explicit occlusion reasoning to handle complete occlusion, and using boundary information to improve accuracy.

References

[1] S. Birchfield. Kanade-Lucas-Tomasi feature tracker, http://www.ces.clemson.edu/~stb/klt.

[2] 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*, pages 495–501, 1997.

[3] D. Daily, F. W. Cathy, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. In *IEEE Conference for Intelligent Transportation Systems*, pages 98–107, 2000.

[4] J. M. Ferryman, A. D. Worrall, and S. J. Maybank. Learning enhanced 3D models for vehicle tracking. In *Proceedings of the British Machine Vision Conference*, pages 1998.

[5] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos. Detection and classification of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):37–47, Mar. 2002.

[6] M. Haag and H. Nagel. Combination of edge element and optical flow estimate for 3D-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision*, 35(3):295–319, Dec. 1999.

[7] S. Kamijo, K. Ikeuchi, and M. Sakauchi. Vehicle tracking in low-angle and front view images based on spatio-temporal markov random fields. In *Proceedings of the 8th World Congress on Intelligent Transportation Systems (ITS)*, 2001.

[8] Z. W. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 521–528, 2003.

[9] D. Koller, K. Dandilis, and H. H. Nagel. Model based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.

[10] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of the European Conference on Computer Vision*, pages 189–196, 1994.

[11] D. R. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing*, 22(2):143–155, Feb. 2004.

[12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000.

[13] T. N. Tan and K. D. Baker. Efficient image gradient based vehicle localization. *IEEE Transactions on Image Processing*, 9(8):1343–1356, 2000.

[14] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Apr. 1991.

[15] T. Zhao and R. Nevatia. Car detection in low resolution aerial image. In *ICCV*, pages 710–717, 2001.