VEHICLE SEGMENTATION AND TRACKING FROM A LOW-ANGLE OFF-AXIS CAMERA

A Thesis

Presented to

the Graduate School of

Clemson University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Electrical Engineering

---

by

Neeraj K. Kanhere

August 2005

Advisor: Dr. Stanley Birchfield

August 5, 2005

To the Graduate School:

This thesis entitled "Vehicle Segmentation and Tracking from a Low-Angle Off-Axis Camera" and written by Neeraj K. Kanhere is presented to the Graduate School of Clemson University. I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science with a major in Electrical Engineering.

_____

Dr. Stanley Birchfield, Advisor

We have reviewed this thesis
and recommend its acceptance:

_____

Dr. Robert Schalkoff

_____

Dr. Wayne Sarasua

Accepted for the Graduate School:

_____

ABSTRACT

A novel method is presented 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 standard algorithm. The key part of the proposed system is to estimate the 3D world co-ordinates of feature points on the vehicles from a single camera. Experimental results on different highways demonstrate the system's ability to successfully segment and track vehicles even in the presence of severe occlusion and significant perspective changes.

DEDICATION

I dedicate this work to my parents who have struggled hard in life to give me the best.

ACKNOWLEDGMENTS

I am most grateful to my adviser, Dr. Stanley Birchfield for guiding me at every step of this endeavor. The freedom he gave me to pursue new ideas made this work a truly enjoyable learning experience. I would also like to thank Dr. Robert Schalkoff for his introductory course in image processing. Things I learned in his course have helped me at every step of this research. Many thanks to Dr. Wayne Sarasua for his generous financial support and encouragement.

Thanks to Prashant, Dinesh and Dinakar for being such good roommates. I have thoroughly enjoyed their company over the last couple of years.

Finally, I would like to thank my family and Uma for their love and patience.

TABLE OF CONTENTS

List of Tables

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

Introduction

Increase in demand for travel on highways has seen an explosive growth over the years. It is no longer feasible to merely build more roads to meet this ever increasing demand. Although it will be inevitable to augment the current infrastructure over the long term, to address the urgent needs it is necessary to utilize the existing infrastructure more efficiently. As a result, Intelligent Transportation Systems (ITS), which is an interdisciplinary technology that helps in design, analysis and monitoring of traffic networks, has received a lot of attention. Throughout the U.S., ITS technology is being used at several locations for automated traffic monitoring and incident detection applications (details can be found at http://www.its.dot.gov/). Most ITS are designed using readily available technology (sensors, communication etc.) which makes them reliable and useful. Proven credibility in recent years has made ITS an integral part of every modern transportation system.

In modeling traffic networks, parameters such as vehicle count, speeds, headway (distance between consecutive vehicles) and truck percentage play a key role. Using specialized sensors to estimate these parameters has gained popularity over manual data collection. A study of different type of sensors in traffic monitoring applications is reported in [18]. A summary of advantages and disadvantages of different types of sensors is presented in Table 1. Video sensors (cameras) are rich in information and offer wide area detection with

a single sensor. Unlike loop detectors, traffic interruption is not required for the installation and maintenance.

1.1 Related Work

The use of image processing for traffic surveillance was initiated in the mid 1970s in the United States and abroad, most notably in Japan, France, Australia, England, and Belgium [19]. The hardware and the algorithms used for estimating traffic parameters have seen a great improvement over the years. Existing commercial systems use a combination of incident detection techniques (detecting changes in image intensities at predefined locations) and heuristics to estimate quantities such as vehicle count and queue length. A much harder problem, but with more potential is tracking, which still remains an active research area. At this point we will review some of the commercial and research oriented systems related to traffic monitoring.

1.1.1 Commercial Systems

By the late 1980s, video imaging detection systems were marketed in the U.S. and elsewhere, generating sufficient interest to warrant research to determine their viability as an inductive loop replacement [20]. At present, there are a number of commercial systems being used throughout U.S. for manual as well as automatic traffic monitoring and incident detection. Two of the popular commercial systems are described below:

Autoscope Solo Pro: Autoscope has enjoyed popularity over the years for its reliability and accuracy. The Autoscope Solo Pro is the latest version of the integrated camera and processor from Image Sensing Systems, Inc. Autoscope video vehicle detection system has continued to set the standard for accuracy, reliability, and flexibility. Market success and market growth have attracted a handful of companies offering competing video systems.

Type	Advantages	Disadvantages
Inductive loop detector	• Low per-unit cost • Large experience base • Relatively good performance	• Installation and maintenance require traffic disruption • Easily damaged by heavy vehicles, road repairs, etc.
Microwave (Radar)	• Installation and repair do not require traffic disruption • Direct measurement of speed • Multilane operation • Compact size	• May have vehicle masking in multilane application • Resolution impacted by Federal Communications Commission (FCC) approved transmit frequency • Relatively low precision
Laser	• Can provide presence, speed, and length data • May be used in an along-the-road or an across-the-road orientation with a twin detector unit	• Affected by poor visibility and heavy precipitation • High cost
Infrared	• Day/night operation • Installation and repair do not require traffic disruption • Better than visible wavelength sensors in fog • Compact size	• Sensors have unstable detection zone • May require cooled IR detector for high sensitivity • Susceptible to atmospheric obscurant and weather • One per lane required
Ultrasonic	• Can measure volume, speed, occupancy, presence, and queue length	• Subject to attenuation and distortion from a number of environmental factors (changes in ambient temperature, air turbulence, and humidity) • Difficult to detect snow-covered vehicles
Magnetometer	• Suitable for installation in bridge decks or other hard concrete surfaces where loop detectors cannot be installed	• Limited application • Medium cost
Video processing	• Provides live image of traffic (more information) • Multiple lanes observed • No traffic interruption for installation and repair • Vehicle tracking	• Live video image requires expensive data communication equipment • Different algorithms usually required for day and night use • Possible errors in traffic data transition period • Susceptible to atmospheric obscurant and adverse weather

Table 1.1: Performance comparison among existing incident detection technologies [18]

In the larger competition for an above-ground detection solution alternative to in-ground loops, video detection systems have been a clear winner [12]. This system consists of a color camera, integrated machine vision processor, and a zoom lens. Autoscope is used as a video alternative to loop detectors for estimating traffic parameters such as vehicle counts, speeds, headways and turning counts. A user specifies detection zones in the image and the algorithm detects the presence of a vehicle in a detection zone. According to the Autoscope specifications, for optimal performance, the camera should be placed at 13 meters (40 feet) above the road surface. In situations of high traffic congestions however, the camera is usually mounted much higher.

Vantage : This technology is developed by Iteris. Similar to Autoscope, Vantage cameras are placed at an optimal location and virtual detection zones are monitored inside the image by the algorithm. From the results reported in [21], Autoscope was found to be more accurate in similar traffic conditions.

1.1.2 Research in Vehicle Tracking

Applying techniques of motion segmentation for tracking vehicles has been an interesting application of computer vision. A number of different approaches have been proposed in the past, each having its own advantages and shortcomings. Approaches which assume that objects to be tracked (vehicles) have already been initialized are not considered in the following discussions, since such systems can not be used in automatic traffic analysis. Techniques used for vehicle detection and tracking can be classified into following popular approaches:

Blob Tracking. In this approach, a background model is generated for the scene. For each input frame, the absolute difference between input frame and the generated background image is processed to extract foreground blobs corresponding to the vehicles on the road.

Variations of this approach have been proposed in [10, 17, 6]. Gupte et al. [10] use adaptive background subtraction to extract a foreground object mask. The threshold for binary image segmentation is chosen dynamically using the histogram of difference image. Vehicle tracking is performed at two levels: region level and vehicle level. The association problem between regions in consecutive frames is formulated as the problem of finding a maximal weight graph. The authors reported 90% detection accuracy and 70% classification accuracy for the test data which was acquired on an overcast day to remove the problem of shadows. Effectiveness of the algorithm in the case of significant heavy-vehicle traffic (large trucks, trailers etc.) is unclear.

The vehicle tracking algorithm proposed by Magee [17], utilizes combination of per pixel background model and a set of set of single hypothesis foreground models based on a general model of object size, position, velocity, and colour distribution. Each pixel in the scene is explained as either background, belonging to one of the foreground objects or as noise. Ground-plane calibration information is used to strengthen the object size and velocity consistency assumption. For improving tracking results, a prior model of typical road travel directions and speeds is built. This helps in initializing the tracker with the mean motion profile (as opposed to random value, or zero velocity) which is close to the ground truth. Using color information and optimal camera location, impressive results ($99\% - 100\%$) have been reported over a one minute sequence.

Active Contour Tracking. A closely related approach to blob tracking is based on tracking active contours (popularly knows as *snakes*) representing an object's boundary. Vehicle tracking using active contour models has been reported by Koller et al. [16]. The contour is tracked using intensity and motion boundaries. A contour is initialized for a vehicle using a background difference image. Tracking is achieved using two Kalman filters, one for estimating the affine motion parameters, and the other for estimating the shape of the contour.

An explicit occlusion detection step is performed by intersecting the depth ordered regions associated to the objects. The intersection is excluded in the shape and motion estimation. Results are shown on real world sequences without shadows or severe occlusions. The algorithm is limited to tracking cars.

3D-Model Based Tracking. Tracking vehicles using three-dimensional models has been studied by several research groups [15, 11, 8, 23]. Some of these approaches assume aerial view of the scene [23], and three dimensional wireframe models for different types of vehicles are used for matching with edges detected in the image. In [8], a single vehicle is successfully tracked through a partial occlusion. Applicability of the model based approach for congested traffic scenes is not clear.

Markov Random Field Tracking. An algorithm for segmenting and tracking vehicles in low-angle frontal sequences has been proposed by Kamijo et al. [13]. In their work, the image is divided into 8×8 pixel blocks, and a spatiotemporal 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. The accuracy increased twofold when the sequence was processed in the reverse order, thus it is not suitable for on-line processing when time-critical results are required. The authors found that the low-angle scenario is indeed a challenging problem.

Feature Based Tracking. In this approach, instead of tracking a whole object, sub-features of an object are tracked. The method is useful in situations of partial occlusions, where

only a portion of an object is visible. The task of tracking multiple objects then becomes the task of grouping the tracked features based on one or more similarity criteria. Beymer et al.[4] have proposed a feature tracking based approach for the task of traffic monitoring application in [4]. In their approach, point features are tracked throughout the detection zone specified in the image. Feature points which are tracked successfully from entry region to the exit region are considered in the process of grouping. Grouping is done by constructing a graph over time. Vertices represent sub-feature tracks and edges represent grouping relationship between tracks. A sub-feature is initially connected to all the neighboring within certain radius in the image plane. Through relative motion, edges representing motion disparity are broken. To compensate effects of depth, a single road-plane homogrpahy mapping is used. The algorithm was implemented on multi-processor digital signal processing (DSP) board for real-time performance. Results have been reported for day and night sequences with varying levels of traffic congestion.

1.1.3 Performance Factors

Some of the performance issues that exist in the problem of automated traffic monitoring are the following:

- *Location of the Camera.* In case of most systems, the camera needs to be placed high above the ground looking down at the vehicles. This serves two purposes, 1) field of view of the camera increases with its distance from the ground2) limits occurrences and severity of occlusions.

- *Traffic Conditions.* When the traffic is free flowing, vehicles are well separated. In that case, tracking is relatively easy. When traffic is moving slowly, vehicles travel close to each other resulting into more occlusion events. Performance of most of the systems degrade in such situations. The only algorithm that is designed specifically for such situations is the one proposed by Beymer et al.[4].

- *Truck Percentage.* A large vehicle often occludes nearby smaller vehicles. In the sequences used for testing the algorithm, it can be seen that a large truck or a trailer traveling in the first lane (the lane closet to the camera) occludes the vehicles traveling in the next lane almost completely. In addition, heavy vehicles often result into multiple counts in case of trip-detection type systems.

- *Number of Lanes.* For the situation in which camera is placed at the side of the road, results are more accurate for the lanes closer to the camera.

- *Moving Shadows.* Moving cast shadows of vehicles result into two or more vehicles merging into a single foreground object, thus reducing the accuracy of the system.

- *Lighting Conditions.* Different algorithms are required for daytime and nighttime. A notable exception to this is the algorithm proposed by Beymer et al [4].

- *Weather Conditions.* Reflection of the headlights on a wet road results into wrong vehicle counts. In case of a snow or a rain, segmenting the foreground objects becomes more difficult.

The commercial systems mentioned in the previous section are designed to be integrated with a traffic management system or similar specialized applications (e.g. toll-gate monitoring, surveillance). For such projects adding the required infrastructure (e.g. mounting poles for the cameras) is a viable option. 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 [14]. Absence of tall structures in rural areas is another situation where placing the camera at a high vantage point is difficult. When the camera is at a low angle, the planar motion assumption (motion in the road plane) is violated. All the techniques

High angle Low angle

Figure 1.1: High angle vs. low angle situation.

discussed in the previous section assume planar motion for the vehicles with the exception of the approach followed by Kamijo et al.[13].

1.2 Definitions

For better understanding of the rest of the material, some definitions are presented below:

Feature point A point location in the image having some kind of discernible quality (e.g. a corner). *Feature point* and *feature* will be used interchangeably in rest of this thesis.

Preimage A unique point in the world corresponding to a point location in the image.

World coordinate system A three dimensional Euclidean coordinate system defined by the user in the offline calibration process.

Low-angle view View from the camera closer to the ground and looking almost parallel to the road.

Frame-block A set of consecutive frames in the sequence, also referred to as a block of frames.

Chapter 2

Approach

The sequence is assumed to be taken from a single grayscale camera pointing at the road from the side. The task of segmenting and tracking vehicles in low-angle cluttered scenes is formulated as a feature tracking and grouping problem. Feature points are tracked in the image sequence using a standard technique followed by estimation of 3D world coordinates for those points, which are then grouped using a standard segmentation technique. The novelty of this work is the estimation of 3D coordinates. The rest of the chapter described this approach in detail.

2.1 Offline Calibration

Calibration is required to estimate 3D world coordinates for corresponding 2D points in the image. The calibration process described below is for a single camera and does not require knowledge about the camera specifications such as focal length or sensor dimensions. The only information that is needed is six or more point correspondences, which makes it possible to process pre-recorded sequences captured from unknown cameras.

2.1.1 Perspective-Projective Camera Model

We assume a perspective-projective pinhole camera model. The general relationship between an object point measured with respect to a user-selected world coordinate system and its image plane point is denoted by a 3×4 homogeneous transformation matrix [22]. This matrix will be referred as the camera calibration matrix **C**:

$$\widehat{\boldsymbol{p}} = \mathbf{C}\,\widehat{\mathbf{P}}, \tag{2.1}$$

where, $\widehat{\boldsymbol{p}} = [\,uw \quad vw \quad w\,]^T$ and $\widehat{\mathbf{P}} = [\,x \quad y \quad z \quad 1\,]^T$ are vectors containing homogeneous coordinates of image point, $\boldsymbol{p} = [\,u \quad v\,]^T$ and world point $\mathbf{P} = [\,x \quad y \quad z\,]^T$ respectively. Representing the matrix with corresponding entries, we get

$$\begin{bmatrix} u\,w \\ v\,w \\ w \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.2}$$

Because **C** is unique only up to a scale factor, we normalize it by fixing the scale factor $c_{34} = 1$.

Expanding the above equation then yields

$$u = \frac{c_{11}x + c_{12}y + c_{13}z + c_{14}}{w} \tag{2.3}$$

$$v = \frac{c_{21}x + c_{22}y + c_{23}z + c_{24}}{w} \tag{2.4}$$

$$w = c_{31}x + c_{32}y + c_{33}z + 1 \tag{2.5}$$

Substituting w into first two equations and rearranging leads to

$$u = x\,c_{11} + y\,c_{12} + z\,c_{13} + c_{14} - u\,x\,c_{31} - u\,y\,c_{32} - u\,z\,c_{33} \tag{2.6}$$

$$v = x\,c_{21} + y\,c_{22} + z\,c_{23} + c_{24} - v\,x\,c_{31} - v\,y\,c_{32} - v\,z\,c_{33} \tag{2.7}$$

These equations define a mapping from the world coordinates to the image coordinates.

2.1.2 Calibration Process

The image coordinates of a point can be calculated from its world coordinates and camera calibration matrix, **C**, which consists of 11 unknown parameters. Knowing the world coordinates and the image coordinates of a single point yields two equations of the form (2.6) & (2.7). Six or more points in a non-degenerate configuration lead to an over-determined system:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 x_1 & -u_1 y_1 & -u_1 z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1 x_1 & -v_1 y_1 & -v_1 z_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u_2 x_2 & -u_2 y_2 & -u_2 z_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v_2 x_2 & -v_2 y_2 & -v_2 z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 & -u_n x_n & -u_n y_n & -u_n z_n \\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_n x_n & -v_n y_n & -v_n z_n \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ c_{21} \\ \vdots \\ c_{33} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{bmatrix} \tag{2.8}$$

which can be solved using a standard least squares technique.

The offline calibration process depends upon the user-specified point correspondences for the calibration process. For improving the accuracy, it is desired that the world coordinates are derived from the actual measurements of the scene, for example, having place markers at known distances. For cases where this information is not available (e.g. pre-recorded data), an approximation can be done using standard specifications such the width of a lane and length of a truck. Gupte et al. [10] have developed a tool to calibrate the road surface with an arbitrary world coordinate system. Our calibration tool, which is shown in Figure 2.1, is similar to that developed by Gupte et al. [10], except that in their work the

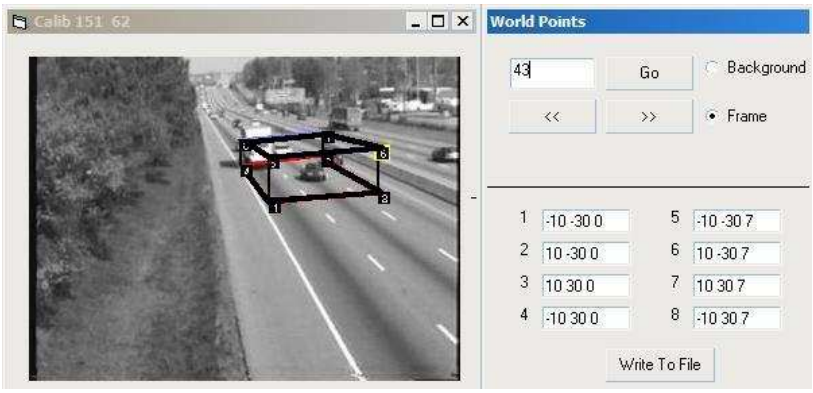


Figure 2.1: **Our calibration tool.**

tool was used to find a planar mapping between the points on the road and the image points; whereas in our case, the calibration tool is being used to estimate a perspective mapping.

An example of the calibration process is shown in Figure 2.2. First, the user places a marker across the width of the road and perpendicular to the lane markings as shown in Figure 2.2 (a). With the marker position unchanged, the sequence is advanced till the rear end of the truck appears to align with the marker position on the ground. A new marker is placed to align with the height of the truck (b). In the same frame a marker is placed on the ground to align with the front end of the truck (c). Once again, the sequence is advanced till the marker placed on the ground in (c) appears to align with the read end of the truck. This is shown in (d). For the same frame, the marker is realigned with the front end of the truck as shown in (e). A new marker is placed across the width of the road (f). One more time, the sequence is advanced for the new marker to appear aligning with the truck's rear end. An additional marker is placed as shown in (g) in such a way that it appears to be aligned with the height of the truck. The result looks as shown in (h). Using the dimensions of a known type of vehicle is an approximate method for estimating world coordinates of control points. Table 2.1.2 lists average lengths of some of the common vehicle types found on the road. In addition, the information about lane width (e.g. 12 feet on an interstate) and number of lanes is used. The calibration process is simple and usually takes around two minutes to complete. Figure 2.3 shows calibration results for different sequences.

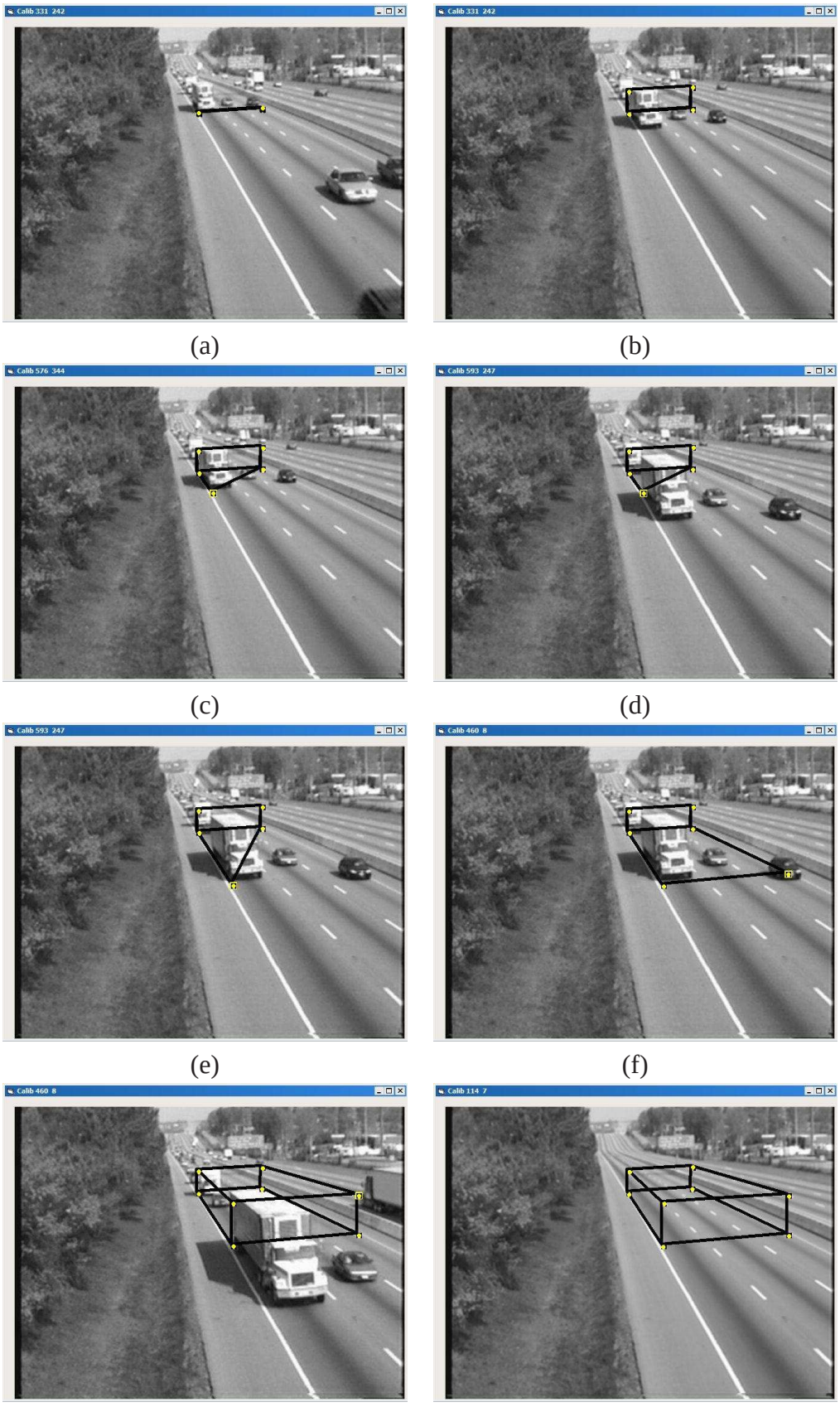


Figure 2.2: Camera calibration process.

1	Passenger Car	17.4 feet
2	Pickup Truck	19.1 feet
3	Buses	41.7 feet
4	4+ axle single units	51.2 feet
5	5-axle single trailer trucks	62.4 feet
6	6 or more axle single trailer trucks	71.2 feet
7	5 or less axle multi trailer trucks	70.0 feet

Table 2.1: Average lengths of standard vehicle classes, as reported in smart loop technology demonstration project webpage [1].

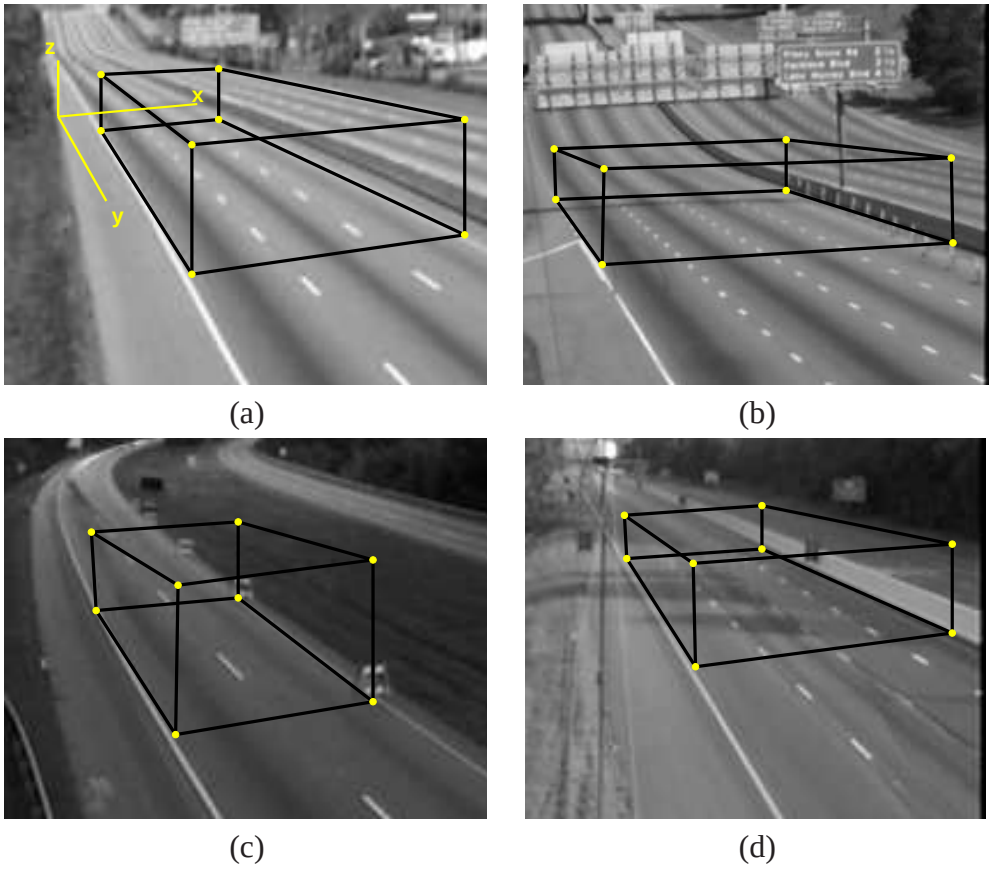


Figure 2.3: Offline calibration for different sequences.

2.1.3 Backprojections

The imaging process maps a point in three dimensional space into a two dimensional image plane. The loss of dimension results in a non-invertible mapping. Given the calibration parameters for the camera and the image coordinates of a single point, the best we can do is to determine a ray in space passing through the optical center and the unknown point in the world. Rearranging equations (2.6) & (2.7) yields equations for two planes in 3D space.

$$(u\,c_{31} - c_{11})\,x + (u\,c_{32} - c_{12})\,y + (u\,c_{33} - c_{13})\,z + (u - c_{14}) = 0 \tag{2.9}$$

$$(v\,c_{31} - c_{21})\,x + (v\,c_{32} - c_{22})\,y + (v\,c_{33} - c_{23})\,z + (v - c_{24}) = 0 \tag{2.10}$$

The intersection of these two planes is the ray in 3D passing through the point in the world **P**, which is projected as ***p*** in the image plane. The problem is under-constrained since we have two equations and three unknowns, namely *x*, *y* and *z*. If we know either *x*, *y* or *z*, we can solve for the other two using the image coordinates and **C**. In the sections to follow, we will explore a simple technique to achieve this.

2.2 Processing a Block of Frames

In the algorithm proposed by Beymer et al. [4], the point features tracked successfully from the entry region to the exit region are considered in the grouping step, which does not pose a problem when the camera is placed at a high vantage point looking down on the road. In the low-angle scenario in which we are interested, frequent occlusions and appearance changes (as vehicles approach the camera) result in losing a large number of features. As a result, the number of features that are tracked for the whole extent of the detection zone is not enough to achieve useful results. One way to overcome this problem is to process a block of frames (typically 5 to 20 frames per block) and to associate segmented vehicles between the successive blocks. Features are tracked throughout a block of F image frames,

overlapping with the previous block by N frames. The length of a block is determined by the average speed of the vehicles and the placement of the camera with respect to the road. If the number of frames in a block is too small, a large number of features will be tracked successfully throughout the frames in the block, but the motion information will be insufficient for effective segmentation. On the other hand, using more frames in a frame-block will yield more reliable motion information at the expense of losing important features. The proposed algorithm relies on human judgment to balance between these tradeoffs.

The steps described in the following sections are performed on the features tracked over a single block.

2.2.1 Tracking Features

Feature points are automatically selected and tracked using the Kanade-Lucas-Tomasi (KLT) feature tracker [2], 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 [26] but are symmetric with respect to the two images. As in [26], features are automatically selected as those points in the image for which both eigenvalues of Z are greater than a minimum threshold.

Among all the features that are tracked, those features which belong to the background are discarded. This requires knowledge of the foreground objects. The process of extracting foreground objects (blobs) is explained in the next section.

2.2.2 Background Subtraction

Background subtraction is a simple and effective technique for extracting the foreground objects from the scene. The process of background subtractions involves initializing and maintaining a background model of the scene, and subtracting the estimated background image from the frame being processed. This is followed by thresholding the difference image and morphological processing to yield foreground blobs. A review of several background modeling techniques is presented in [5].

A simple method of temporal median filtering produced satisfactory results for the test sequences. More elaborate methods like mixture of Gaussians [9] or nonparametric kernel density estimation [7] offer better accuracy for segmenting foreground in cluttered scenes. For the scope of this research, the median filtering technique was chosen for its simplicity and effectiveness. The median filter belongs to a general class of *rank filters*. It is frequently used in image processing for removing noise in an image. For background modeling, we will perform one dimensional median filtering in time domain. For each pixel in the background image, the median value is selected from the set of values observed at the same pixel location in the previous n frames. Sample frames from two of the sequences along with the generated background images are shown in Figure 2.4.

For each frame-block, a binary thresholding operation is performed on the absolute difference between background image and first frame of the block. The difference image

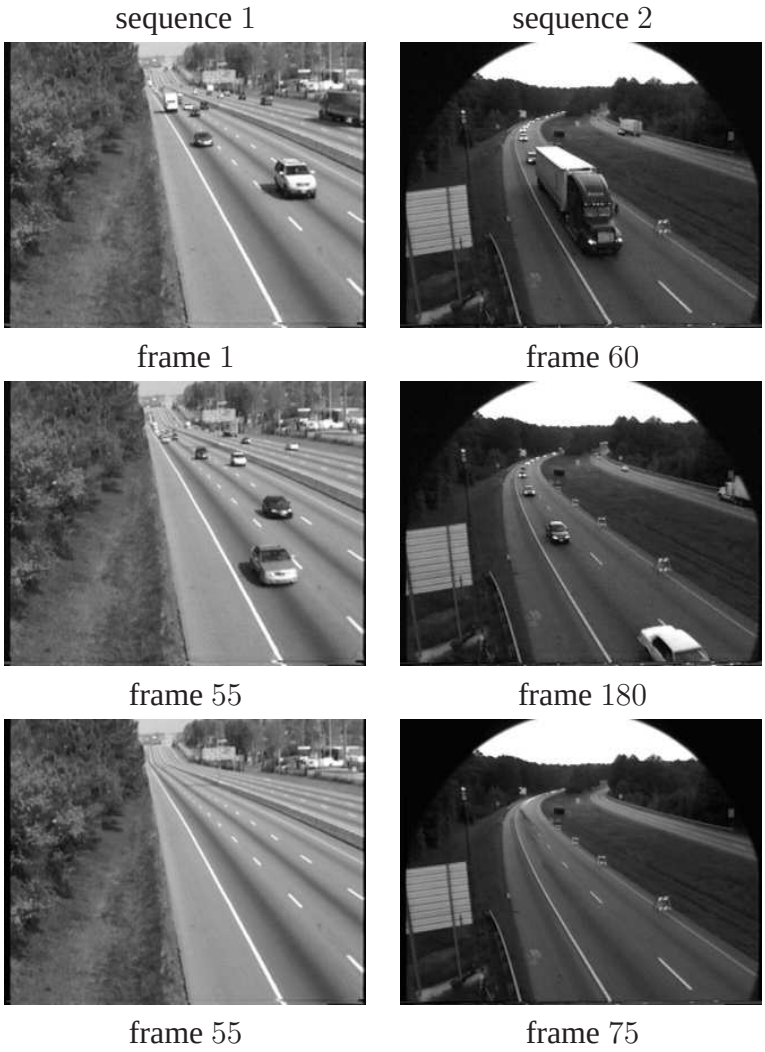


Figure 2.4: Sample frames and estimated background images using temporal median filtering.

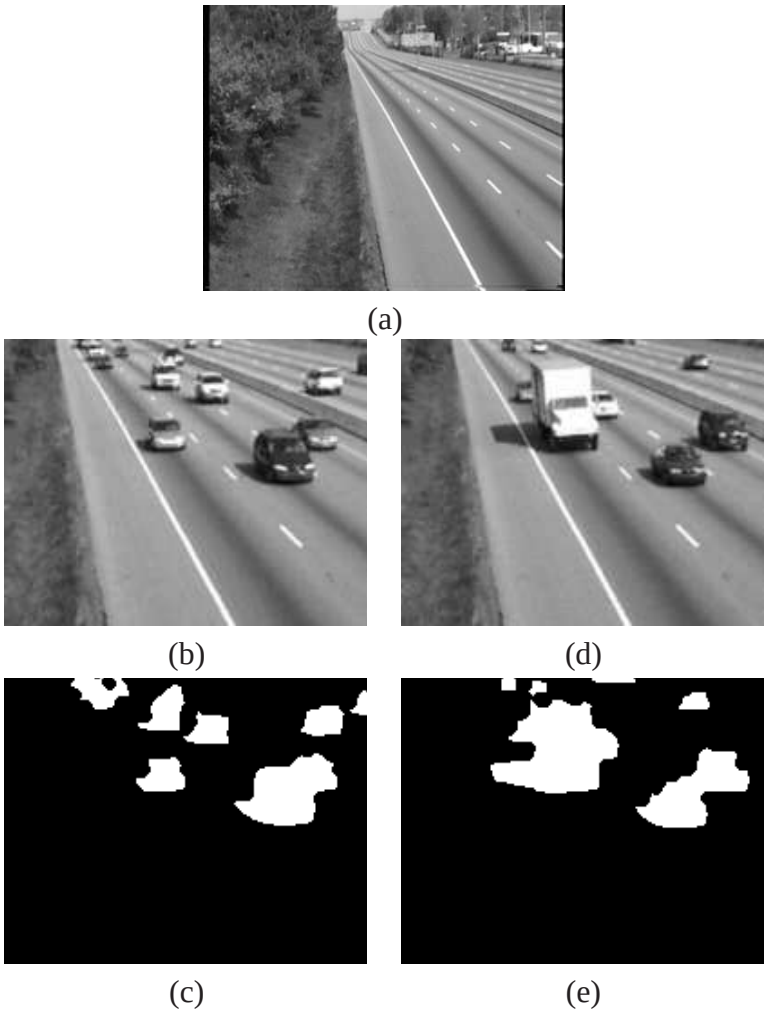


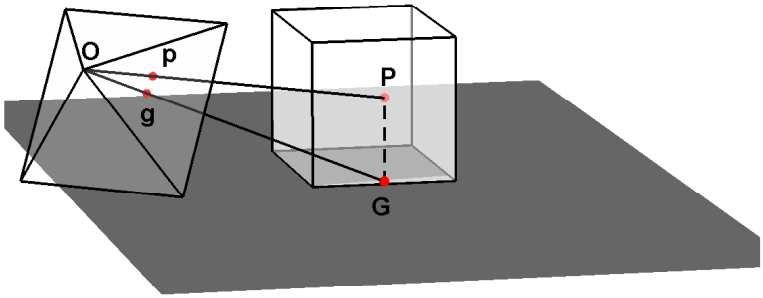
Figure 2.5: **Background subtraction:** (a) generated background, (b)-(c) and (d)-(e): input frame and resultant foreground mask

needs to be processed further (using morphological operations) to suppress false detections, and to obtain closed foreground regions.

2.2.3 Stable Features from a Single Frame

It was shown in section 2.1.2 & section 2.1.3 that for a point in the image, we can estimate the 3D coordinates of the corresponding world point using the calibration parameters and at least one component of its world coordinates. A simple technique to achieve the same is presented here which involves finding the vertical projection of a point on the road surface in the image. The foreground mask generated in the previous step is used to find the pro-

(a) (b)



(c)

Figure 2.6: **Road projection:** Projecting a feature on the road surface in the image for estimating its height. (a) Input frame (b) foreground mask is used for ground projection (c) 3D model. *p* and *g* are image points corresponding to *P* and *G* respectively. *O* is the optical center.

jection as shown in Figure 2.6(b). P is a 3×1 vector of world coordinates corresponding to the point p in the image. O is optical center of the camera. G is a 3×1 vector containing world coordinates of ground projection of P. Rearranging (2.6) and (2.7) yields

$$\begin{bmatrix} c_{31}\,u - c_{11} & c_{32}\,u - c_{12} \\ c_{31}\,v - c_{21} & c_{32}\,v - c_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c_{14} - u + z(c_{13} - c_{33}\,u) \\ c_{24} - v + z(c_{23} - c_{33}\,v) \end{bmatrix}$$

From the above equation it follows that

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c_{31}\,u - c_{11} & c_{32}\,u - c_{12} \\ c_{31}\,v - c_{21} & c_{32}\,v - c_{22} \end{bmatrix}^{-1} \begin{bmatrix} c_{14} - u + z(c_{13} - c_{33}\,u) \\ c_{24} - v + z(c_{23} - c_{33}\,v) \end{bmatrix} \tag{2.11}$$

Since G lies on the ground (or at least sufficiently close), we can compute its 3D coordinates by substituting $z = 0$ (corresponding to the road plane) in the above equation. P and G have same (x, y) coordinates. Now, we know the image coordinates p of the world point P along with its (x, y) coordinates, and the camera calibration parameters $\mathbf{C}$. Substituting these values into equations (2.6), (2.7), we solve for z:

$$z = \frac{h_p^T\,h_c}{h_p^T\,h_p} \tag{2.12}$$

$$h_p = \begin{bmatrix} u\,c_{33} - c_{13} \\ v\,c_{33} - c_{23} \end{bmatrix} \tag{2.13}$$

$$h_c = \begin{bmatrix} c_{14} - u\,c_{34} + (c_{11} - u\,c_{31})\,x + (c_{12} - u\,c_{32})\,y \\ c_{24} - v\,c_{34} + (c_{21} - v\,c_{31})\,x + (c_{22} - v\,c_{32})\,y \end{bmatrix} \tag{2.14}$$

For this technique to work, a simple box-model for the vehicles is assumed. A vehicles is modelled using five rectangular surfaces as shown in Figure 2.6(c). Two such models have been used to represent cars and heavy vehicles. Dimensions of corresponding models are computed using the calibration information (in proportion to the lane width). The calibration process described in section 2.1.2 is based on human judgment and therefore will not be perfect. Moreover, the objective of finding world coordinates of points is to be

(a) (b)

Figure 2.7: Error in height estimation caused by long shadows.

able to segment the vehicles based on the approximate location of the feature points in the world coordinate system. Estimates of the world coordinates under these conditions would not be accurate enough to use detailed shape models for the vehicles.

As shown in Figure 2.8, the rate of change of error in the location backprojected on the road increases non-linearly with increasing z. This relationship is derived in Appendix C. It can be seen that for the feature points which are closer to the road, an error in estimation of height z results in comparatively less error in the estimation of world coordinates, as compared to that of features higher up. The technique explained in this section works for points lying on any of the four surfaces of a vehicle which are orthogonal to the road plane. Thankfully, in practice, features that are successfully detected and tracked rarely belong to the top surface, primarily due to insufficient texture and a relatively small projection in the image. After estimating height of all the features using this technique, features which are close to the road surface (having $z \leq \delta$ where δ is a user defined parameter) are selected as *stable features*. In our previous work [14], stable features were selected based on an additional criterion of low variance in height estimation for each frame of the block. Neglecting the variance criterion reduced the number of computations without any noticeable degradation in the segmentation results.

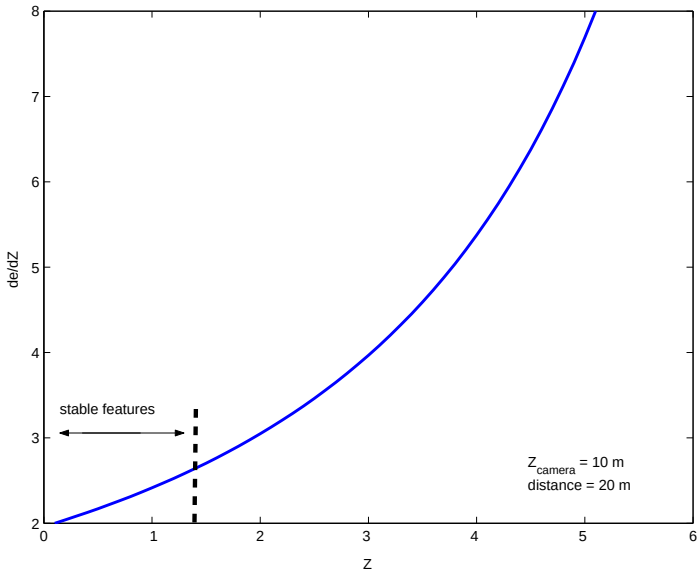


Figure 2.8: Rate of change of backprojection error as the function of z

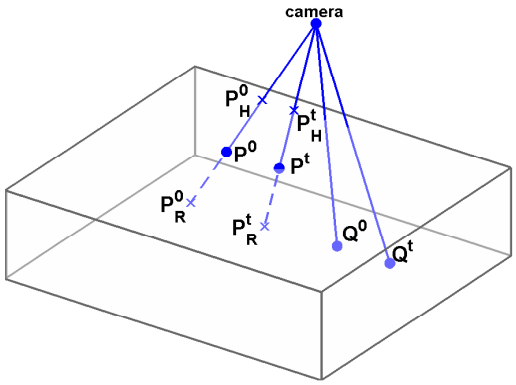


Figure 2.9: Estimating world coordinates using rigid motion. Coordinates of P are unknown. Q is a stable feature point with known world coordinates

2.2.4 World Coordinates from Multiple Frames

Factors like occlusion and shadows introduce significant error in the height estimates of the feature points obtained using the technique presented in the previous section (Figure 2.7). Stable features are used to estimate the world coordinates of the rest of the features using rigidity constraints and translational motion model.

A line in 3D can be represented in a parametric form as:

$$\mathbf{P} = \mathbf{P}_R + \alpha \left[\mathbf{P}_H - \mathbf{P}_R\right]$$

where, $\mathbf{P}_R$ and $\mathbf{P}_H$ are any two points on the line, and α is a scalar which defines location of a point along the line. The above representation simplifies the equations to follow.

As shown in Figure 2.9, we consider two points, **P** and **Q** which undergo a translational motion from $\mathbf{P}^0$, $\mathbf{Q}^0$ at the initial frame F^0 to $\mathbf{P}^t$, $\mathbf{Q}^t$ at frame F^t. If **Q** is one of the stable features, then its real world coordinates are known for both the frames. Let us represent world coordinates for **P** using parametric form of the equation of a line on which it lies.

$$\mathbf{P}^0 = \mathbf{P}_R^0 + \alpha^0 [\mathbf{P}_H^0 - \mathbf{P}_R^0] \tag{2.15}$$

$$\mathbf{P}^t = \mathbf{P}_R^t + \alpha^t [\mathbf{P}_H^t - \mathbf{P}_R^t] \tag{2.16}$$

Where, $\mathbf{P}_R$ and $\mathbf{P}_H$ are back projections of **P** at $z = 0$ and $z = h$ obtained using (2.11). Value of h is chosen as the z coordinate of one of the four upper control points used in the calibration process. For **P** to be on the same rigid body as **Q**, the following condition must be satisfied:

$$\mathbf{P}^t - \mathbf{P}^0 = \mathbf{Q}^t - \mathbf{Q}^0,$$

i.e., both points undergo the same translation. Representing in parametric form,

$$\{\mathbf{P}_R^t + \alpha^t [\mathbf{P}_H^t - \mathbf{P}_R^t]\} - \{\mathbf{P}_R^0 + \alpha^0 [\mathbf{P}_H^0 - \mathbf{P}_R^0]\} = \mathbf{Q}^t - \mathbf{Q}^0 \tag{2.17}$$

From our assumption that the road is flat, it follows that a feature point on a vehicle will travel parallel to the road surface. This implies that z coordinate of **P** has to be the same in both frames.

$$\mathbf{P}_z^0 = \mathbf{P}_z^t$$

which can be represented as

$$\mathbf{P}_{R_z}^0 + \alpha^0 [\mathbf{P}_{H_z}^0 - \mathbf{P}_{R_z}^0] = \mathbf{P}_{R_z}^t + \alpha^t [\mathbf{P}_{H_z}^t - \mathbf{P}_{R_z}^t]$$

By definition,

$$\mathbf{P}_{R_z}^0 = \mathbf{P}_{R_z}^t = 0$$

and

$$\mathbf{P}_{H_z}^0 = \mathbf{P}_{H_z}^t = h$$

Substituting, we get

$$\alpha^0 h = \alpha^t h \Rightarrow \alpha^0 = \alpha^t$$

The assumption that the road surface is flat is essential for the above relationship to hold. Substituting $\alpha = \alpha^0 = \alpha^t$ in the previous equation,

$$\mathbf{P}_R^t - \mathbf{P}_R^0 + \alpha\{[\mathbf{P}_H^t - \mathbf{P}_H^0] - [\mathbf{P}_R^t - \mathbf{P}_R^0]\} = \mathbf{Q}^t - \mathbf{Q}^0$$

Following the notation of Appendix A, let

$$\Delta_{P_R} = \mathbf{P}_R^t - \mathbf{P}_R^0 \tag{2.18}$$

$$\Delta_{P_H} = \mathbf{P}_H^t - \mathbf{P}_H^0 \tag{2.19}$$

$$\Delta_Q = \mathbf{Q}^t - \mathbf{Q}^0 \tag{2.20}$$

Substituting we get,

$$\Delta_{P_R} + \alpha[\Delta_{P_H} - \Delta_{P_R}] = \Delta_Q$$

Solving for α yields,

$$\alpha = \frac{[\Delta_{P_H} - \Delta_{P_R}]^T[\Delta_Q - \Delta_{P_R}]}{[\Delta_{P_H} - \Delta_{P_R}]^T[\Delta_{P_H} - \Delta_{P_R}]} \tag{2.21}$$

From α, the world coordinates of $\mathbf{P}$ at any time instant can be obtained as

$$\mathbf{P} = \mathbf{P}_R + \alpha[\mathbf{P}_H - \mathbf{P}_R] \tag{2.22}$$

Using the known world coordinates of the stable features, estimates for the non-stable feature points are obtained using the above relationship.

Let $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_s$ be the stable features and let $\mathbf{P}$ be a feature for which the world coordinates needs to be estimated. World coordinates of $\mathbf{P}$ are estimated with respect to each $\mathbf{Q}_k$.

$$\mathbf{P}_k = \mathbf{P}_R + \frac{[\Delta_{P_H} - \Delta_{P_R}]^T [\Delta_{Q_k} - \Delta_{P_R}]}{[\Delta_{P_H} - \Delta_{P_R}]^T [\Delta_{P_H} - \Delta_{P_R}]} [\mathbf{P}_H - \mathbf{P}_R] \tag{2.23}$$

Finally, we choose,

$$\mathbf{P} = \min_k \left\{ w_d \; || \; \widetilde{\mathbf{P}}_k - \widetilde{\mathbf{Q}}_k \; ||_2 + w_e \, [\Delta_{P_k} - \Delta_{Q_k}]^T [\Delta_{P_k} - \Delta_{Q_k}] \right\} \tag{2.24}$$

$$\text{where, } k = 1, 2, \ldots, s$$

In the above equation, $\widetilde{\mathbf{P}}$ represents 2×1 vector of x and y world coordinates of $\mathbf{P}$. The first term weighted by w_d is the Euclidean distance in x and y between $\mathbf{P}_k$ and $\mathbf{Q}_k$. The second term, weighted by w_e, is the squared trajectory error between $\mathbf{P}_k$ and $\mathbf{Q}_k$ at estimated world coordinates. Only the x and y coordinates are used for calculating the Euclidean distance to avoid penalizing a feature point for being at a higher elevation from the road surface. World coordinates for all the unstable features are estimated in the same manner.

2.2.5 Affinity Matrix and Normalized Cuts

We form the affinity matrix composed of three components, namely, the 3D Euclidean distance in world coordinates, difference in trajectory and the *background content measure*. Euclidean distance and background content are measured using coordinates of feature points in the first frame of the block. The affinity matrix A is computed as:

$$A = A^D \otimes A^E \otimes A^B \tag{2.25}$$

$$A^D_{i,j} = e^{\frac{\{\|P_i^0 - P_j^0\|_2\}}{\sigma_d^2}} \tag{2.26}$$

$$A^E_{i,j} = e^{\frac{\{[\Delta_{P_i}-\Delta_{P_j}]^T[\Delta_{P_i}-\Delta_{P_j}]\}}{\sigma_e^2}} \tag{2.27}$$

$$A^B_{i,j} = e^{\frac{\{\lambda(\mathbf{p}_i^0,\mathbf{p}_j^0,\beta_f)\}}{\sigma_b^2}} \tag{2.28}$$

$$\tag{2.29}$$

$A^D_{i,j}$ is the 3D Euclidean distance between $\mathbf{P}_i^0$ and $\mathbf{P}_j^0$ at $t = 0$, i.e. the first frame of a block.

$A^E_{i,j}$ is the error in trajectories of features P_i and P_j. Trajectories are computed in the world coordinate frame. Consider two points which belong to the same vehicle. If the estimated world coordinates for those two points are close to the true values, then world-trajectories for the points would match in spite of possibly different image velocities. This is observed more frequently in case of a heavy vehicle like trailers.

$A^B_{i,j}$ is the measure of background content between two features. $\lambda(\mathbf{p}_i^0, \mathbf{p}_j^0, \beta_0)$ is a function which measures number of background pixels that lie on a line connecting $\mathbf{p}_i^0$ and $\mathbf{p}_j^0$ in the image. β_0 is the background image at $t = 0$.

The contribution of each factor to the affinity matrix is controlled by corresponding σ parameters. In Shi et al.[25, 24], it is mentioned that for the normalized cut algorithm to be computationally efficient, the affinity matrix (also called weight matrix) should be sparse. Shi et al. [25, 24] achieve this by limiting the computation of edge weights to a local neighborhood. In this work, feature points, rather than all image pixels, represent nodes in the graph. In addition, separate affinity matrices are formed for each of the connected component in the segmented foreground mask image. This results in affinity matrices of a reasonable size for applying the normalized cut algorithm. Experiments were performed using sparse affinity matrices, i.e. using only local edge connections for a feature, but it was observed that using full matrices produced better results without a significant increase in the computing time.

2.2.6 Grouping With Incremental Cuts

Image segmentation based on low level cues cannot and should not aim to produce a completely correct segmentation. The objective should instead be to use the low-level coherence of brightness, color, texture or motion attributes to sequentially generate hierarchical partitions. Mid-level and high-level knowledge can then be used to either confirm these groups or select some for further attention. This attention could result in further repartitioning or grouping [24]. The same can be said for motion segmentation.

In this section a grouping procedure that we call *incremental cuts* will be explained for segmenting a set of features into meaningful groups. The key part of this step is to use the calibration information to accept or reject a feature group based on its spatial properties. Fig. 2.2.6 shows the steps for grouping with incremental cuts.

```
function IncrementalCuts()
01    V = [ ]
02    for each label ∈ L
03        List = {k | L(p_k) = label}
04        while L not empty,
05            A_List ← {A_{i,j} | i ∈ List, j ∈ List}
06            increment ← true
07            c ← 0
08            while increment
09                c ← c + 1
10                [G_1, G_2, ..., G_c] = NormalizedCuts(A_List, c)
11                for each G ∈ {G_1, G_2, ..., G_c}
12                    if ValidGroup(G)
13                        V ← V ⊕ G
14                        List ← List ⊖ List_G
15                        increment ← false
```

V is the set of valid segmented groups for the block. A is the affinity matrix of all features in the block. L is the labelled foreground mask corresponding to the first frame in the block. Image coordinates of a feature point k are represented by p_k.

`NormalizedCuts(`$A_{List},\ c$`)` is a function which applies normalized cuts on the affinity matrix A_{List} to give c disjoint groups $\{G_1, G_2, \ldots, G_c\}$. $\oplus$ and $\ominus$ are set addition and set subtraction operations. $List_G$ is the list of features in group G. `ValidGroup` is a function which returns `true` if all of the following conditions are satisfied for a group G:

1. Number of features in G is more than a threshold value.
2. The centroid (in 3D coordinates) lies inside the detection zone.
3. Dimensions of G are within a valid range.

The range of valid dimensions for the two vehicle models are calculated using the calibration information. For simplicity, only two possible classes are assumed; cars(car, SUV, pick-up truck) and heavy vehicles (trailers, buses).

2.3 Correspondence Between Frame Blocks

In previous sections, we looked at how to track feature points through a block of F frames, estimate corresponding world coordinates, and in the end, how to group features using incremental cuts. With the same set of parameters, we segment consecutive blocks of frames. Blocks overlap by $F-1$ frames. For long term tracking, it is necessary to find correspondence between detections within consecutive frame blocks. This section describes an approach for finding the correspondence.

Consider two consecutive frame-blocks **A** and **B** with F frames in each block and overlapping by N frames. Let $\{a_1, a_2, \ldots, a_{n_1}\}$ denote feature groups segmented in a frame-block **A**. Similarly, let $\{b_1, b_2, \ldots, b_{n_2}\}$ denote feature groups segmented in frame-block **B**. An undirected graph is formed with the segmented feature groups in both frame blocks

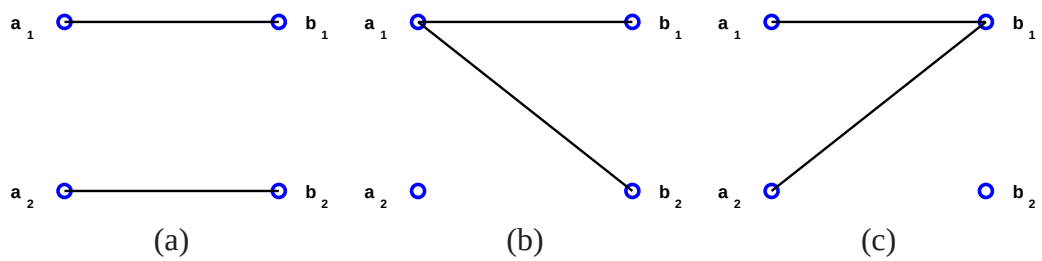


Figure 2.10: correspondence events: (a) one-to-one (b)a_1 splits into b_1 and b_2. a_2 is declared missing. (c)a_1 and a_2 merge into b_1. b_2 is declared as new detection.

as nodes and the number of common feature points shared by a pair of groups as the weight of an edge connecting the respective nodes. If a group in the previous block shares features with only a single group in the current block, then we call this a one-to-one unique correspondence. A group in **A** sharing features with more than one group from **B** indicates splitting. Similarly, two or more groups in **A** sharing common features with a group in **B** indicates merging. A group in **A** having no association is considered a missing event, and a group in **B** having no association with any of the groups in the previous block is considered as a new detection. If a group is associated with a one-to-one correspondence over β_r consecutive blocks, it is labelled as a reliable group. If a group is missing for β_m consecutive blocks, it is labelled as inactive. During initialization, each group in the first frame-block is assigned a unique label. For each consecutive frame-block, a graph is constructed as mentioned above. To neglect minor segmentation errors, all the edges having weights $w < w_{min}$ are removed. This is followed by searching for the unique one-to-one correspondences between the groups of previous and current frame-blocks. Groups of the current block having unique correspondences are assigned the labels of respective groups in the previous block. After processing all the unique associations, the graph is searched for splits. For a split event, the edge with maximum weight is used for correspondence and the remaining edges are removed. Merge events are handled the same way. Groups in **A** which are no longer connected to any of the groups in **B** and are labelled as reliable, are declared missing. Groups in **B** which are not connected with any of the groups in **A** are declared as new detections. Each group that is declared as a new detection is matched with

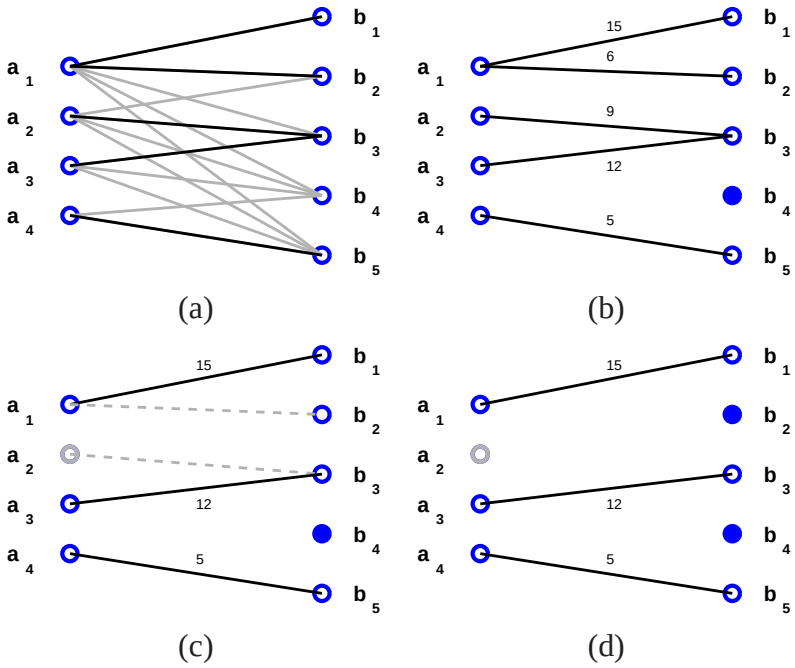


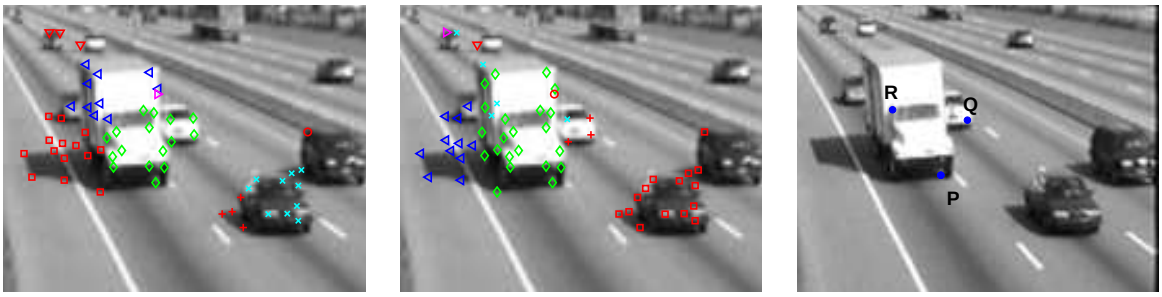
Figure 2.11: (a)Initialized graph, gray edges with $w < w_{min}$ (b)b_4 is matched with missing groups, otherwise assigned a new label. (c)Maximum-weight edges are selected for split and merge events. (d)b_2 is matched with missing groups, otherwise assigned a new label. a_2 is declared missing if it has been labelled as reliable, discarded otherwise.

all the active missing groups to find a possible correspondence. If a correspondence with missing groups is not found, the group is assigned a new label.

Chapter 3

Experimental Results

To judge the improvement in segmentation by using 3D coordinates, a sample frame-block was analyzed using planar motion assumption. In this case, the affinity matrix was computed with the assumption that all the feature points lie in the road plane. Feature points higher up on the truck lie far in the back from their true location when backprojected on the road. Using only the image velocities, or the planar motion assumption, features which are closer to the ground are not grouped together with the features which are higher up on the vehicle. On the right, using the estimates of the world coordinates, most features that belong to the truck are grouped together correctly. Results of the computations for the three points shown in Figure 3.1(c) are presented in Table 3



(a) (b) (c)

Figure 3.1: Better segmentation with world coordinates. (a) using planar motion assumption. (b) using estimated world coordinates. (c) three points for analysis.

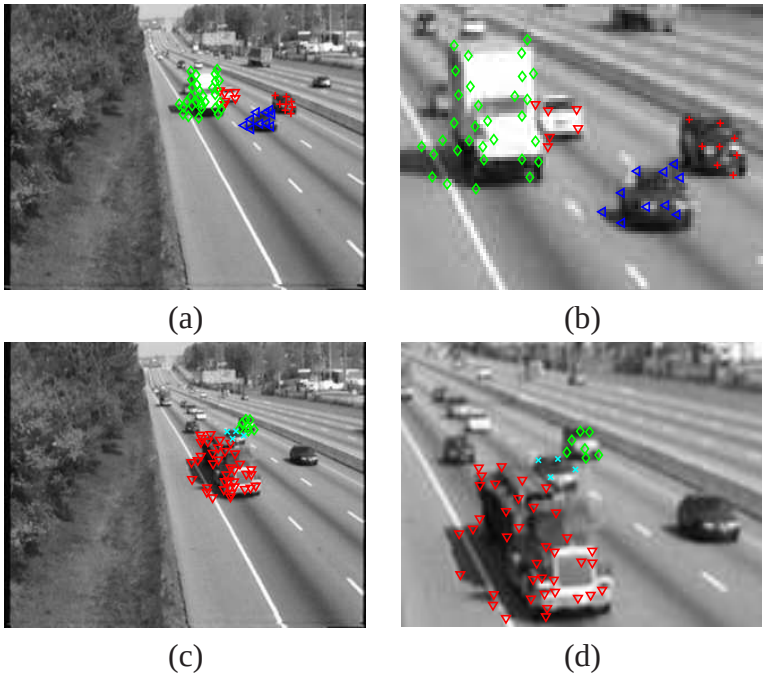
Quantity	Planar	Image	3D
Distance between P and Q	13.48 units	22.44 pixels	11.21 units
Distance between P and R	16.13 units	28.87 pixels	6.31 units
Trajectory difference in P and Q	0.1 units	7.43 pixels	0.2 units
Trajectory difference in P and R	4.2 units	6.22 pixels	0.05 units

Table 3.1: Improved segmentation with 3D information.

The columns show the values computed using the planar motion assumption, image coordinates, and the estimates of world coordinates respectively. With planar motion assumption, both the distance and the trajectory difference between *P* and *Q* is less than *P* and *R*. This explains the grouping of P and Q together in Figure 3.1(a). The distances computed using world coordinates are closer to the true values. This explains the grouping of *P* and *Q* in (a). Using the 3D information, *P* and *R* are grouped together correctly and *Q* belongs to a different group as shown in (b).

The algorithm was tested on four grayscale image sequences, each containing 1200 frames captured at 24 frames per second. The camera was placed on an approximately 9 m pole on the side of the road. The sequences were digitized at 320×240 resolution. No preprocessing was done to suppress shadows or to stabilize occasional camera jitter. For each sequence, offline camera calibration was performed as explained earlier.

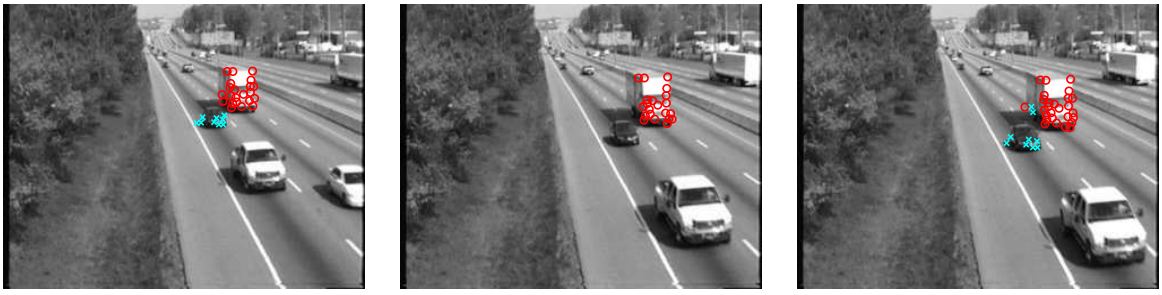
The first sequence was captured on a clear day. Vehicles are travelling in three lanes and there are moderate moving shadows. Some results from the sequence are shown in Figure 3.2. Frame 592 demonstrates the ability of the algorithm to correctly detect and track a vehicle which is severely occluded by another vehicle (a small vehicle is occluded by a large trailer in the adjacent lane). The vehicle is occluded throughout the detection zone, and appears to be moving with almost the same speed as that of the trailer. In frame 178, a truck and a car travelling next to each other are segmented correctly even when the shadow of the truck results in merging of the two vehicles in the foreground mask. In frame 182, some of the features on the car are lost, and the car is missing due to lack of



(a) (b)

(c) (d)

Figure 3.2: Sequence 1 (a) frame 35 (b) frame 35 zoomed (c) frame 592 (d) frame 592 zoomed.



(a) (b) (c)

Figure 3.3: Sequence 1 (a) frame 178 (b) frame 182 (c) frame 183.

(a) (b) (c)

Figure 3.4: Sequence 1 (a) frame 295 (b) frame 303 (c) frame 310.

sufficient reliable features. However, in frame 183, the new detection is matched with the missing groups and associated with the correct missing group. Although both the vehicles are detected and tracked, the segmentation is not perfect. Notice the two feature points on the truck that are grouped with the car. The estimates of world coordinates for both the features are incorrect. When computing the world coordinates, minimum value for the equation (2.24) was obtained for a stable feature belonging to the car. However, the feature at the back end of the truck is correctly grouped with the rest of the features. In frames 295-310, the two vehicles travelling in the middle lane, are not detected. The reason for these missed detections is that neither of the vehicles has the minimum required number of features. Having a low threshold on this number results in overs-segmentation. Setting a higher threshold avoids detection of spurious groups at the cost of missing a vehicle occasionally. During the experiments, it was observed that most of the missed detections were for dark colored vehicles (due to lack of sufficient texture in the image).

The second sequence shows a four-lane highway with the last lane blocked for maintenance work. The lane closure results into a slow moving traffic with vehicles traveling close to each other. The sequence was captured during data collection for studying the effect of a workzone on freeway traffic [3]. Some of the frames from the sequence are shown in Figure 3.5.

In frame 72, the algorithm successfully segments the trailer and the smaller vehicle traveling close to it. It might appear that a vehicle in the last lane has been grouped with

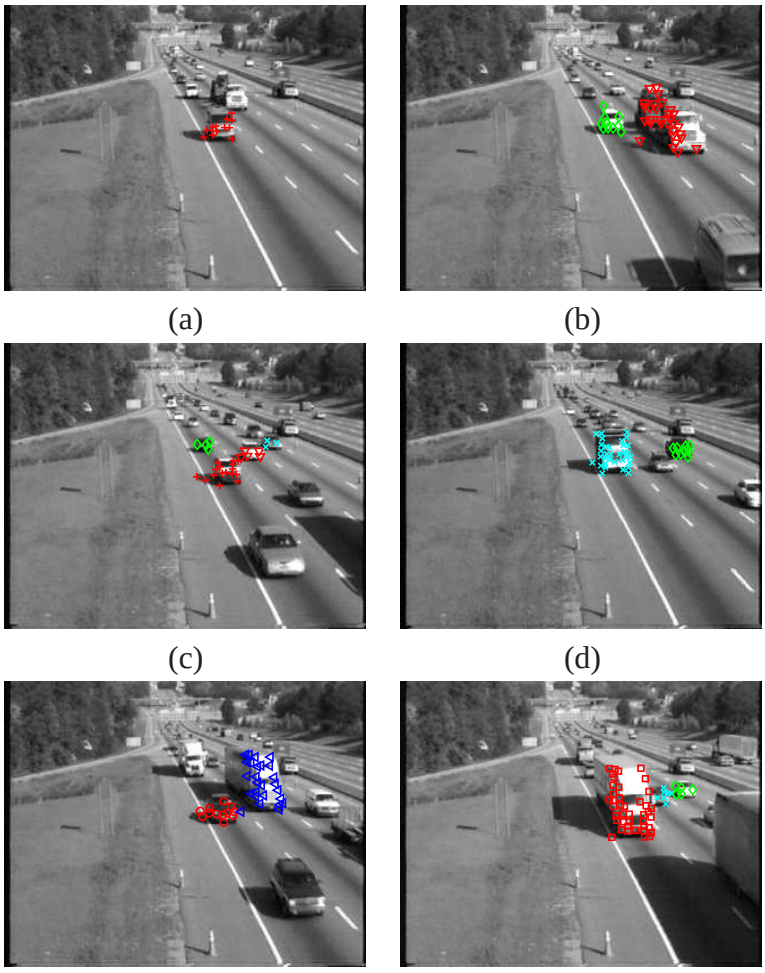
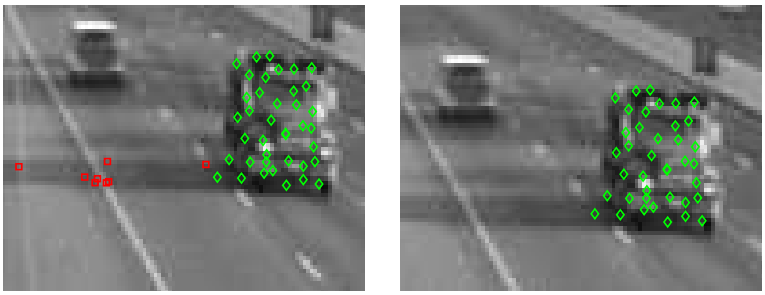


Figure 3.5: Sequence 2 (a) frame 58 (b) frame 72 (c) frame 184 (d) frame 240 (e) frame 310 (f) frame 330.



(a) (b)

Figure 3.6: Sequence 3 frame 10 (a) moving shadow of the truck is detected as a vehicle, (b) setting a threshold on minimum height of a group removes the group formed by shadow.

the trailer, but it is the load that the trailer is carrying and not a different vehicle. Frames 310 and 330 show segmentation in the presence of large vehicles. In frame 310, the white suburban is occluded for most of the frame-block by the trailer and the vehicle traveling ahead of it resulting in a missed detection.

The third sequence was found to be more challenging. Vehicles cast long shadows making the process of segmentation based of size-constraints harder. One simple method was tested for detecting and removing groups that belong to shadows as shown in Figure 3.6. If the height of a group is below a threshold value, it is classified as a shadow group and is discarded. Having zero as the threshold (which is theoretically correct) does not yield the desired results, since the estimation process is based on the approximate calibration along with simple assumption for the shape of vehicles resulting in height estimation error. If the threshold is set higher, more shadow-groups are detected and discarded at the cost occasionally detecting a small vehicle (e.g. a compact sports car) as a shadow group.

Segmentation results are shown for frames 300 to 315 in Figure 3.7. In frame 302, a truck is correctly segmented. By frame 308, enough features are reliably tracked to segment the occluded trailer. In frame 311, the pickup truck is detected. Note that the entire pickup truck is in the shadow cast by the trailer. All the three vehicles are detected as a single foreground object as a result of long shadows.

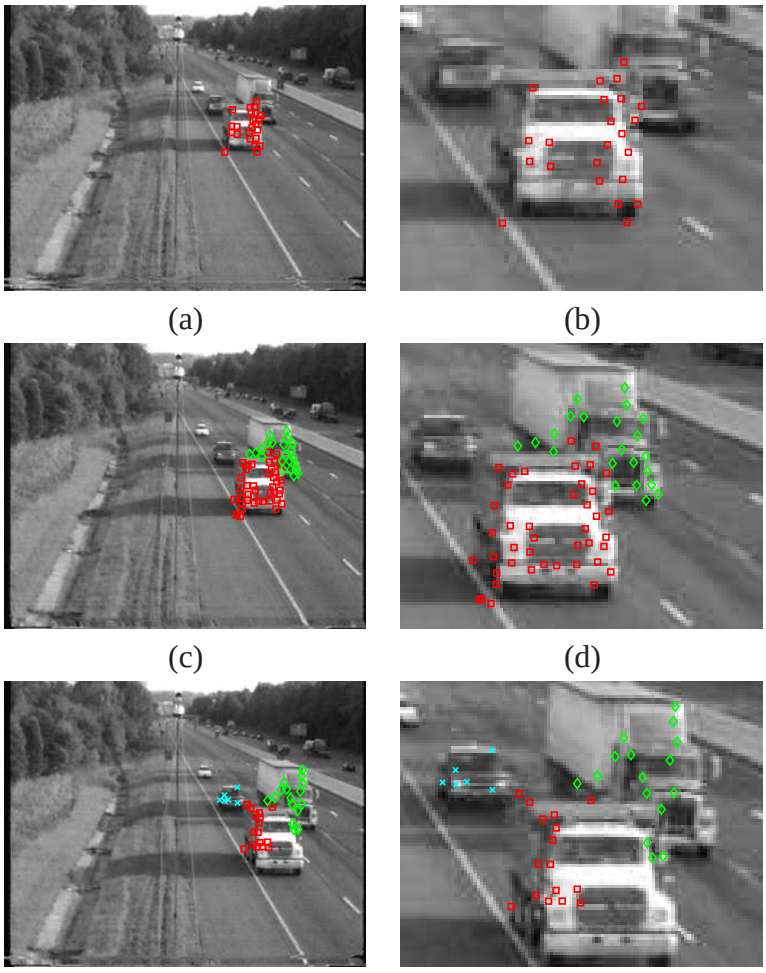


Figure 3.7: Sequence 3 (a) frame 302 (b) frame 302 zoomed (c) frame 308 (d) frame 308 zoomed (e) frame 311 (f) frame 311 zoomed.

Sequence	C	T	O	DC	DT	DO	FP
1	116	9	19	114 (98%)	9 (100%)	16	4
2	120	8	17	115 (96%)	7 (88%)	11	4
3	57	7	11	53 (93%)	6 (86%)	6	5
4	43	3	9	43 (100%)	3 (100%)	6	2

Table 3.2: Accuracy on sequences. The columns show the sequence, number of cars (C), number of trucks (T), number of occluded vehicles (O), number of cars tracked (DC), number of trucks tracked (DT), number of occluded vehicles detected and tracked (DO) and number of false detections (FP) respectively.

The fourth and the last sequence was captured for the workzone study. The images are noisy compared to the previous sequences due to the presence of fog. Vehicles are traveling close to each other at low speeds. Three frames of the result are shown in Figure 3.8. In all the three cases shown here, vehicles under partial occlusion are segmented correctly. Quantitative assessment of the results on all the sequences is presented in Table 3.

Figure 3.8: Sequence 4 (a) frame 415 (b) frame 415 zoomed (c) frame 560 (d) frame 560 zoomed (e) frame 756 (f) frame 756 zoomed.

Chapter 4

Conclusion

Most approaches to segmenting and tracking vehicles from a stationary camera assume that the camera is high above the ground, thus simplifying the problem. A technique has been presented in this thesis 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 occlusions are more frequent. In such a situation, planar motion assumption for vehicles is violated, especially in case of heavy vehicles like trailers. The approach proposed is based upon grouping tracked features using a standard segmentation algorithm. A novel part of the technique is the estimation of the 3D world coordinates of features using a combination of background subtraction, offline camera calibration (for a single camera), and rigidity constraints under translational motion. Experimental results on real sequences show the ability of the algorithm to handle the low-angle situation, including severe occlusion.

Some of the aspects of the proposed algorithm need further analysis and improvement. At the heart of the algorithm is the feature point tracker. Improving the tracker to handle intensity changes resulting from static or moving shadows will ensure more features that are tracked reliably. Explicit shadow suppression step will improve the accuracy of the segmentation. A very simple approach has been adopted for associating the results between the frame-blocks. The approach is based solely upon the number of common features and

is susceptible to errors easily. Using the spatial proximity and motion information is likely to help the association step in making correct decisions. The algorithm was implemented and tested in Matlab, except for the feature tracking code [2]. Since the bulk of the core computations are performed using nested loops, implementing the algorithm in a compiled environment is expected to yield a better performance.

APPENDICES

Appendix A

Notation Used

$\boldsymbol{p}$: 2×1 vector of image coordinates of a point.

$\widehat{\boldsymbol{p}}$: 3×1 vector of homogeneous image coordinates of a point.

$\mathbf{P}$: 3×1 vector of world coordinates of a point.

$\widehat{\mathbf{P}}$: 4×1 vector of homogeneous world coordinates of a point.

$\mathbf{A}$: 3×4 Camera calibration matrix.

P^t : 3×1 vector of world coordinates of point P at time t.

Δ_P : 3×1 translation vector for point P between first and last frames of a frame block.

Appendix B

Assumptions

For the proposed approach, following assumptions have been made.

1. Two classes of vehicles are assumed (cars, SUVs etc. and trailers)
2. Road is assumed to be straight and flat.
3. Translational motion has been assumed to model motion of vehicles.
4. A perspective-projective pinhole camera model is assumed.
5. It is assumed that at least one point feature, which is close to the road surface (low-height), is successfully tracked for each vehicle.
6. Absence of long shadows

Out of these assumptions, the first two assumptions are reasonable in case of vehicles traveling on a highway. The fourth assumption, which appears to be a strong one, is found to be satisfied in practice. The last assumption has been made for convenience. The issue of detecting and suppressing static as well as moving shadows has been postponed for future work.

Appendix C

Mapping error as the function of height from road surface

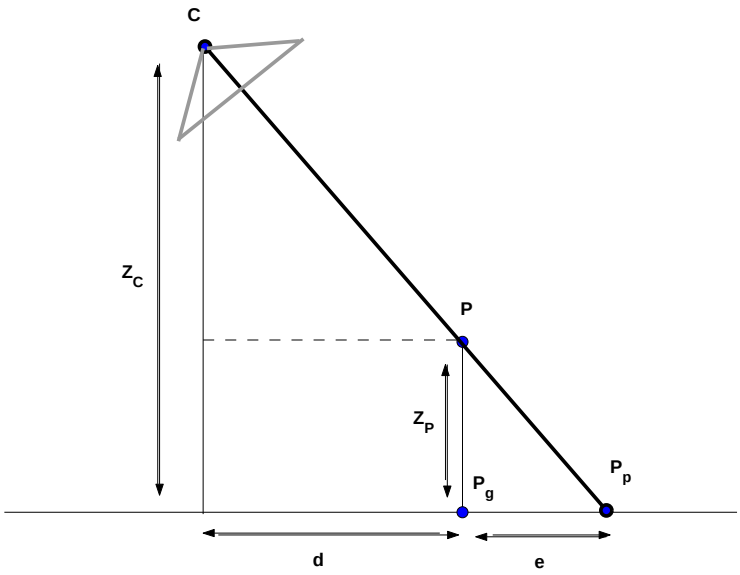


Figure C.1: Mapping error *e* as the function of *Z*

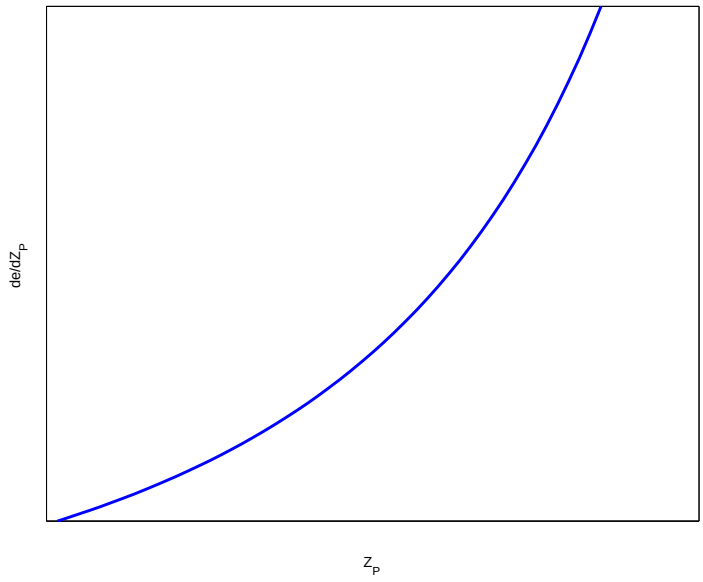
Consider a point *P* in the scene with its ground projection P_g. P_g is at a distance *d* from the base of the camera. Z_P and Z_c are distances of *P* and *C* respectively from the ground plane. Assuming (erroneously) that all the points lie on the road surface, the image of *P* is assumed to correspond with the world point P_P.

$$\frac{d+e}{Z_C} = \frac{d}{Z_C - Z_P}$$

$$e = \frac{Z_P\, d}{Z_C - Z_P}$$

The error due to violation of the planar motion assumption increases with the distance of a point from the road surface and the distance between that point and the camera measured along the road surface. Differentiating above equation with respect to Z_P yields:

$$\frac{\partial e}{\partial Z_P} = \frac{Z_c\, d}{(Z_c - Z_P)^2} \tag{C.1}$$



Bibliography

[1] Smart Loop Technology Demonstration Project at Otay Mesa, San Diego, http://www.dot.ca.gov/dist11/operations/tsps/sloops1.htm.

[2] S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker, http://www.ces.clemson.edu/~stb/klt/.

[3] Wayne Sarasua. Traffic Impacts of Short Term Interstate Work Zone Lane Closures: The South Carolina Experience, http://ops.fhwa.dot.gov/wz/workshops/accessible/Sarasua.htm.

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

[5] S. C. Cheung and Chandrika Kamath. Robust techniques for background subtraction in urban traffic video. In *Proceedings of Electronic Imaging: Visual Communications and Image Processing*, 2004.

[6] 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.

[7] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of IEEE International Conference on Computer Vision Frame-rate workshop*, Sept 1999.

[8] J. M. Ferryman, A. D. Worrall, and S. J. Maybank. Learning enhanced 3d models for vehicle tracking. In *British Machine Vision Conference*, pages 873–882, 1998.

[9] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.

[10] 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, March 2002.

[11] 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.

[12] Graham Heywood. Autoscope: Design of a decade. *Traffic Technology International Annual Review 2004*.

[13] 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.

[14] Neeraj K. Kanhere, Shrinivas J. Pundlik, and Stanley T. Birchfield. Vehicle segmentation and tracking from a low-angle off-axis camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1152–1157, 2005.

[15] 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.

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

[17] D. Magee. Tracking multiple vehicles using foreground, background and motion models. In *Proceedings of ECCV Workshop on Statistical Methods in Video Processing, (2002)*. http://citeseer.ist.psu.edu/magee01tracking.html.

[18] Hani S. Mahmassani, Carl Haas, Sam Zhou, and Josh Peterman. Evaluation of incident detection methodologies. Technical Report FHWA/TX-00/1795-1, Center of Transportation Research, The University of Texas at Austin, Oct 1998.

[19] P. G. Michalopoulos. Vehicle detection video through image processing: the autoscope system. *IEEE Transactions on Vehicular Technology*, 40(1):21–29, Feb 1991.

[20] Dan Middleton, Deepak Gopalakrishna, and Mala Raman. Advances in traffic data collection and management (white paper). http://www.itsdocs.fhwa.dot.gov/JPODOCS/REPTS_TE/13766.html.

[21] Dan Middleton and Ricky Parker. Vehicle detection evaluation. Technical Report FHWA/TX-03/2119-1, Texas Transportation Institute, Texas A&M University, Oct 2002.

[22] Robert J. Schalkoff. *Digital Image Processing and Computer Vision*. John Wiley & Sons, Inc., first edition, 2002.

[23] C. Schlosser, J. Reitberger, and S. Hinz. Automatic car detection in high resolution urban scenes based on an adaptive 3D-model. In *EEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, Berlin*, pages 98–107, 2003.

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

[25] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *IEEE International Conference on Computer Vision*, pages 1154–1160, 1998.

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