Final Report for
Traffic Surveillance and Detection Technology Development
New Traffic Sensor Technology

Jitendra Malik, Stuart Russell

December 19, 1996

Contents

1 Description **3**

2 Tracking Approach **4**
- 2.1 3D Model based tracking . . . 4
- 2.2 Region based tracking . . . 4
- 2.3 Active contour based tracking . . . 5
- 2.4 Feature based tracking . . . 5

3 Motion-Based Grouping **6**

4 Tracking and grouping procedures: Detailed analysis **9**
- 4.1 Off-line camera definition . . . 9
- 4.2 On-line tracking and grouping . . . 9
  - 4.2.1 Sub-feature Detection . . . 10
  - 4.2.2 Feature Tracking . . . 10
  - 4.2.3 Grouping . . . 11
- 4.3 Preliminary analysis of parameter measurement errors . . . 14

5 Vehicle Classification **14**
- 5.1 Vehicle Class Definitions . . . 15
- 5.2 Significant Attribute Definitions . . . 15
- 5.3 Measurement of Significant Attributes . . . 15
  - 5.3.1 Downstream Velocity and Lateral Position . . . 15
  - 5.3.2 Vehicle Height . . . 15
  - 5.3.3 Vehicle Length and Width, Camera Position Known . . . 17
  - 5.3.4 Vehicle Length and Width, Camera Position Unknown . . . 19
- 5.4 Classification Using Measured Data . . . 19

6 Hardware Port **20**
- 6.1 Overall Architecture . . . 20
- 6.2 Updating tracks . . . 21
- 6.3 Software development . . . 21
- 6.4 Field Deployable system . . . 22

7 Parameters computed at the TMC **23**
7.1 Overall design . . . 23
7.2 Vehicle matching . . . 23
7.2.1 Computing most likely matches . . . 25
7.2.2 Measured vehicle features . . . 25
7.2.3 Sensor and motion models . . . 26
7.3 Computation of travel times and O/D counts . . . 27

8 Testing methodology **28**
8.1 Introduction . . . 28
8.2 Individual parameter testing . . . 28
8.2.1 Vehicle flow . . . 28
8.2.2 Average speed . . . 29
8.2.3 Link travel time . . . 29
8.2.4 Vehicle classification . . . 29
8.2.5 Lane changes . . . 29
8.2.6 Queue length (distance) . . . 30
8.2.7 Queue length (number of vehicles) . . . 30
8.2.8 Spatial headway (inverse of time average density) . . . 30
8.2.9 Acceleration . . . 30
8.2.10 O/D tracking . . . 31
8.3 Test conditions . . . 31
8.4 Primary test sites . . . 31
8.4.1 Sacramento Test Site, Northbound . . . 32
8.4.2 Sacramento Test Site, Southbound . . . 33
8.4.3 Walnut Creek Test Site 1, I-680 Southbound . . . 34
8.4.4 Walnut Creek Test Site 2, Hwy 24 Westbound . . . 35
8.4.5 San Jose Intersection Test Sites . . . 36

9 Results **37**
9.1 Vehicle level results . . . 37
9.2 Single site level results . . . 38
9.2.1 Results for parameters with automated calibration . . . 38
9.2.2 Results for parameters with manual calibration . . . 69
9.2.3 Acceleration detection . . . 90
9.2.4 Lane change detection . . . 95
9.2.5 Classification results . . . 101
9.2.6 Queue length detection . . . 102
9.3 Multiple site level results . . . 102

A Camera Calibration **105**

B Interfaces **107**

C Computation of 3-D Shape **111**

D Parameter Measurement **112**

E Data Sources **115**

F Classification confusion matrix **117**

G Acknowledgements **118**

1 Description

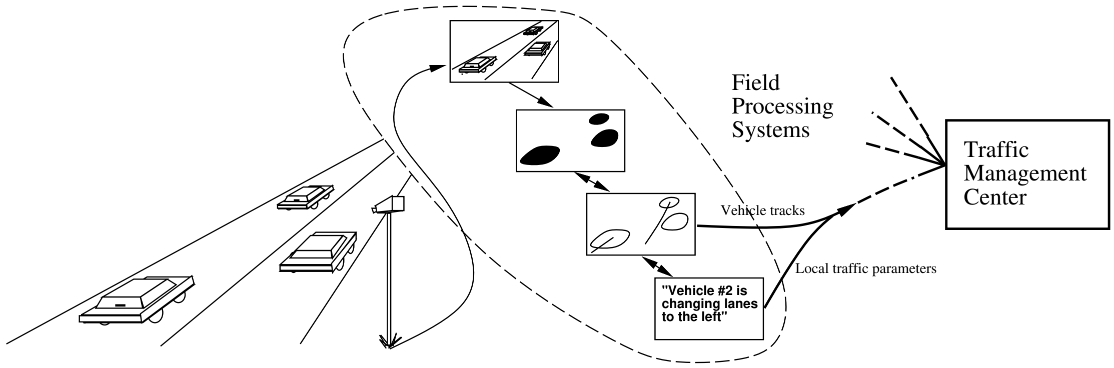


Figure 1: Our system concept.

Our system concept is illustrated in Figure 1. The core idea is to have video cameras mounted on poles or other tall structures looking down at the traffic scene. Video is captured, digitized, and processed by onsite computers, and then transmitted in summary form to a Transportation Management Center (TMC) for collation and computation of multi-site statistics such as link travel times.

Processing occurs in three stages:

1. Segmentation of the scene into individual vehicles and tracking each individual vehicle to refine and update its position and velocity in 3D world coordinates, until it leaves the tracking zone.

2. Reasoning from the track data in order to compute local traffic flow parameters including vehicle counts per lane, average speeds, incidents, lane change frequencies, etc. These parameters, together with track information (timestamp, vehicle type, color, shape, X–Y position), are communicated to the TMC at regular intervals.

3. At the TMC, local traffic parameters from each site are collated and displayed as desired, and/or used in controlling signals, message displays, and other traffic control devices. Computers at the TMC also process the track information from neighboring camera sites to compute long-distance parameters such as link times and origin–destination counts.

For reference, we include the functional specifications from the RFP (Table 1 on page 2 of RFP Exhibit III) as Table 1 below.

Parameter	Units	Range	Reporting rate	Error
Vehicle flow rate	veh/lane/hr	0-2500	variable	± 2.5%
Average speed	mph	0-90	variable	± 1 mph
Link Travel time	min	0-60	variable	± 5 %
Vehicle classification	Type count	0-2400	variable	± 5 %
Lane Changes	changes by lane	as measured	variable	± 5 %
Queue length	veh /type/lane	as measured	variable	± 5 %
Spatial headway	feet per vehicle	as measured	variable	± 5 %
Acceleration	mph/sec	as measured	variable	± 5 %
Origin/destin. tracking	enter/exit loc.	0-500 veh/hr/loc	tracked vehicle	± 10 %

Table 1: Table of recovered parameters.

2 Tracking Approach

Tracking moving objects in video streams has been a popular topic in the field of computer vision in the last few years; earlier contributions to the areas of multi–target tracking and data association were made by control and aerospace engineers. Our application context entails several stringent requirements for a proposed scheme:

1. Automatic segmentation of a vehicle from the background and other vehicles so that there can be a unique track associated with each vehicle.
2. Deal with variety of vehicles – motorcycles, passenger cars, buses, construction equipment, trucks, etc.
3. Deal with range of traffic conditions – light midday traffic, rush-hour congestion, varying speeds in different lanes.
4. Deal with variety of lighting conditions – day, evening, night, sunny, overcast, rainy days.
5. Real-time operation of the system.

The different tracking approaches in video data that have been studied in computer vision can be classified as follows.

2.1 3D Model based tracking

Three-dimensional model-based vehicle tracking systems have previously been investigated by several research groups, the most prominent being the groups at Karlsruhe [9] and at the University of Reading[1, 14]. The emphasis is on recovering trajectories and models with high accuracy for a small number of vehicles. The most serious weakness of this approach is the reliance on detailed geometric object models. It is unrealistic to expect to be able to have detailed models for all vehicles that could be found on the roadway.

2.2 Region based tracking

The idea here is to identify a connected region in the image – a "blob" – associated with each vehicle and then track it over time using a cross-correlation measure. Initialization of the process is most easily done by the background subtraction technique. A Kalman filter-based adaptive background model[7, 8] allows the background estimate to evolve as the weather and time of day affect lighting conditions. Foreground objects (vehicles) are detected by subtracting the incoming image from the current background estimate, looking for pixels where this difference image is above some threshold and then finding connected components.

This approach works reasonably well in free-flowing traffic. Difficulties arise in two important situations. The first is that of long shadows that may result in connecting up blobs that should have been associated with separate vehicles. This can be dealt with to some extent by making use of color information or by exploiting the fact that shadow regions tend to be devoid of texture. The more serious, and so far intractable, problem for video based traffic surveillance systems has been that of congested traffic. Under these conditions, vehicles partially occlude one another instead of being spatially isolated, which makes the task of segmenting individual vehicles difficult. Such vehicles will become grouped together as one large blob in the foreground image. This would obviously lead to undercounting vehicles.

2.3 Active contour based tracking

A dual to the region based approach is tracking based on active contour models, or *snakes*. The idea is to have a representation of the bounding contour of the object and keep dynamically updating it. The previous system for vehicle tracking developed in our group [6, 10, 11] was based on this approach. The advantage of having a contour based representation instead of a region based representation is reduced computational complexity (though multi–resolution approaches can considerably reduce complexity of region matching).

However, the problems stated previously arising from shadows linking different vehicles and the inability to segment vehicles that are partially occluded remains. If somehow one could initialize a separate contour for each vehicle, then one could keep tracking even in the presence of partial occlusion[10]. However it is the initialization that is the difficult part!

2.4 Feature based tracking

Finally, yet another approach to tracking abandons the idea of tracking objects as a whole but instead tracks sub-features such as distinguishable points or lines on the object (e.g. portion of rear bumper, lower corner of rear windshield). The advantage of this approach is that even in the presence of partial occlusion, some of the sub-features of the moving object remain visible. The technology of tracking points and line features in a Kalman filtering formalism is quite well developed in the computer vision community. Since a vehicle could have multiple sub-features, the new problem then is that of grouping – what set of features belong to the same object.

3 Motion-Based Grouping

The grouping of vehicle sub-features will be based on a common motion constraint, a concept known to Gestalt psychologists as common fate. Point features that are seen as moving rigidly together will be grouped together into a single vehicle. But since there are many vehicles in traffic scenes, there is also an important segmentation aspect to the problem. One does not want to link together sub-features from neighboring vehicles. The grouping process must be sensitive enough to pick up a motion that distinguishes a vehicle from its neighbors, a motion such as a slight acceleration or lane drift.

To make the grouping system robust enough to segment different vehicles, the spatial information guiding the grouper will be integrated over a period of time, utilizing as many image frames as possible. Only the sub-features that are tracked from a detection region at the bottom of the image to an exit region near the top will be allowed to participate in the final grouping. Thus, in order to fool the grouper, two vehicles would have to have identical motions during the entire time they were being tracked. In congested traffic, vehicles are constantly changing their velocity to adjust to nearby traffic, thus giving the grouper the information it needs to perform the segmentation. In free-flowing traffic, vehicles may be more likely to maintain constant spatial headways over time, thus making the grouping constraint less useful. But in this scenario, there is more space between vehicles, so a spatial proximity cue is added to aid the grouping/segmentation process. The spatial proximity cue also helps to reduce the grouper's computational load by limiting the number of sub-feature pairs that are examined.

Since most road surfaces are flat, the grouper exploits an assumption that vehicle motion will be parallel to the road plane. To describe the road plane, the user simply specifies four or more line or point correspondences between the image road and a separate "world" road plane, as shown in Fig. 2. Based on this off-line step, the system can compute a projective transform, or homography, between the image coordinates (x, y) and world coordinates (X, Y). By writing points in homogeneous coordinates, this is a simple linear transform

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \propto H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

The scaling of H is arbitrary, so $H(3,3)$ is often chosen to be 1.

The grouper considers sub-feature points in pairs. That is, the basic grouper computation is whether or not to group together the 2D point features $p_a(t)$ and $p_b(t)$. The dependence on time t is written to emphasize that the grouper is working with sub-feature tracks, and hence has access to the time history of points. The 3D coordinates of these points in the real world will be written in upper case $P_a(t)$ and $P_b(t)$.

Consider the simple case where P_a and P_b are at the same distance to the camera (e.g. both on the back face of a truck). In this scenario, the grouper only needs to look at a simple function of the displacement vector $p_a(t) - p_b(t)$. Since P_a and P_b are both at the same distance from the camera d, $p_a(t)$ and $p_b(t)$ are both scaled by the same scale factor $1/d$. Thus, for points on the same vehicle, $p_a(t) - p_b(t)$ will be constant over time if we can simply compensate for the $1/d$ scaling. Fortunately, the homography can be used for this compensation. Given a point (x, y) in the image, we can estimate the scale factor s that transforms the region around that point to world coordinates. The difference vector $p_a(t) - p_b(t)$ can then simply be scaled by s.

Now consider the more general case where P_a and P_b are not at the same distance from the camera. We can no longer compensate by using the estimated scale factor s. However, consider the relationship between a point P in 3D and its projection onto the road Q, as shown in Fig. 3. The camera origin O, $P(t_1)$, and $P(t_2)$ define a plane. Where $\overline{O\,P(t_1)}$ and $\overline{O\,P(t_2)}$ intersect the road plane define $Q(t_1)$ and $Q(t_2)$, and where $\overline{P(t_1)\,P(t_2)}$ and $\overline{Q(t_1)\,Q(t_2)}$ intersect the X-Z

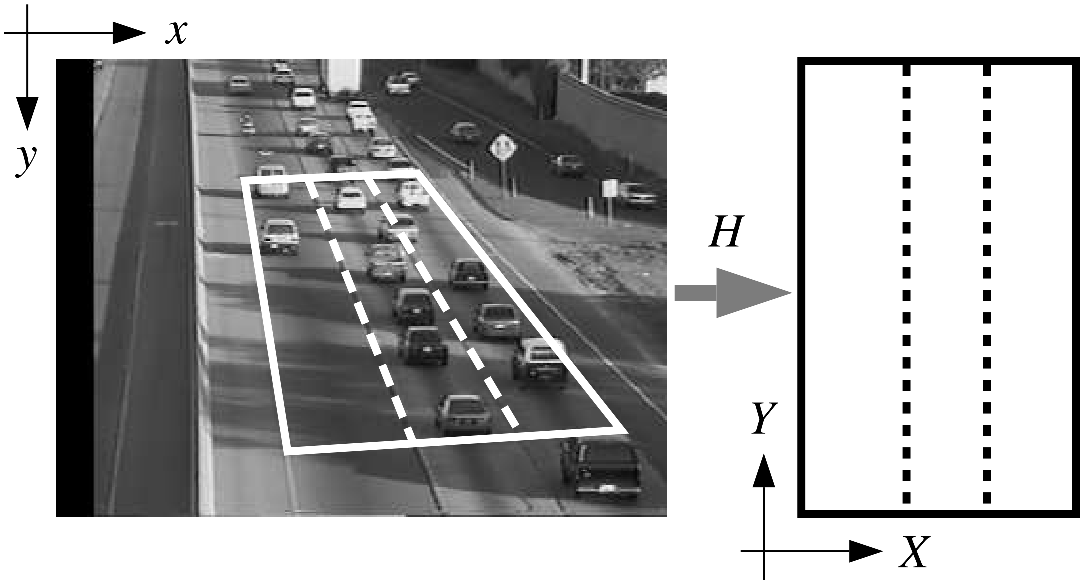


Figure 2: A projective transform H, or homography, is used to map from image coordinates (x, y) to world coordinates (X, Y).

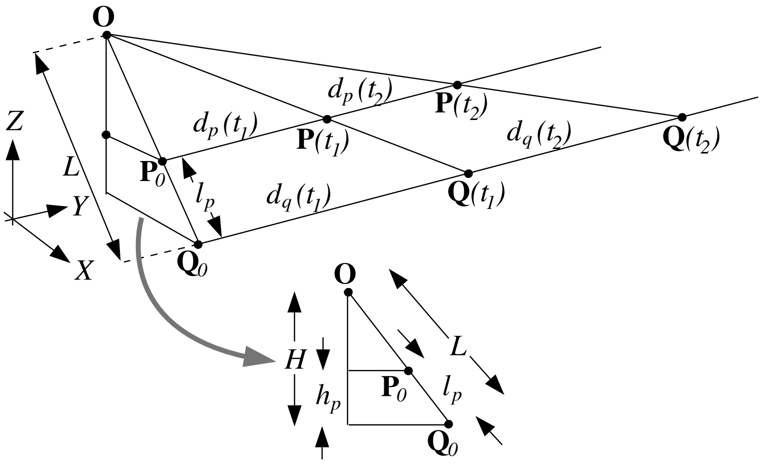


Figure 3: When a point P moves from $P(t_1)$ to $P(t_2)$, the distance the projection on the road surface Q moves is a scaled version of the true distance, where the scaling factor is related to the height h_p above the road plane.

plane defines P_0 and Q_0. The distance from P_0 to $P(t_1)$ is denoted by $d_p(t_1)$ and the distance from P_0 to $P(t_2)$ is denoted by $d_p(t_2)$; $d_q(t_1)$ and $d_q(t_2)$ are defined similarly. Using similar triangles, one can easily show

$$\frac{d_p(t_1)}{d_q(t_1)} = \frac{L - l_p}{L} = \frac{d_p(t_2)}{d_q(t_2)}$$

and

$$\frac{L - l_p}{L} = \frac{H - h_p}{H}.$$

Rearranging terms, one gets

$$\frac{d_p(t_2) - d_p(t_1)}{d_q(t_2) - d_q(t_1)} = \frac{H - h_p}{H}.$$

This means that in world coordinates, the actual distance traveled $d_p(t_2) - d_p(t_1)$ is the measured distance (assuming zero height) $d_q(t_2) - d_q(t_1)$ multiplied by a function of the true height $\frac{H-h_p}{H}$. Thus, points on the same vehicle but at different heights will move at slightly different speeds in world coordinates, where higher points move faster than lower ones. For points projected onto the road surface, differences in height are now more important than differences in depth.

This suggests the following for the grouper. When comparing points $P_a(t)$ and $P_b(t)$, one should scale the lengths of the tracks to equalize them in world coordinates, and then look at whether $Q_a(t) - Q_b(t)$ is constant over time. Scaling the tracks to give them equal length effectively places then at the same height. Adding this flexibility to the grouper may enable it to group points at different heights, but this comes at the cost of allowing some overgrouping. For instance, if one vehicle moves at a velocity $v(t)$ and a nearby vehicle moves at exactly $c\, v(t)$, where c is any constant, then the track scaling will group the vehicles together. But notice that the constant c must not vary over time.

Empirically, we have found that the first method of simply compensating for depth performs better than the track scaling method. Most of the features that survive from detection to exit regions are from the rear of the vehicle, and these features are from roughly the same depth. This does cause some oversegmentation of trucks, however, since features on trucks are usually tracked at a variety of depths.

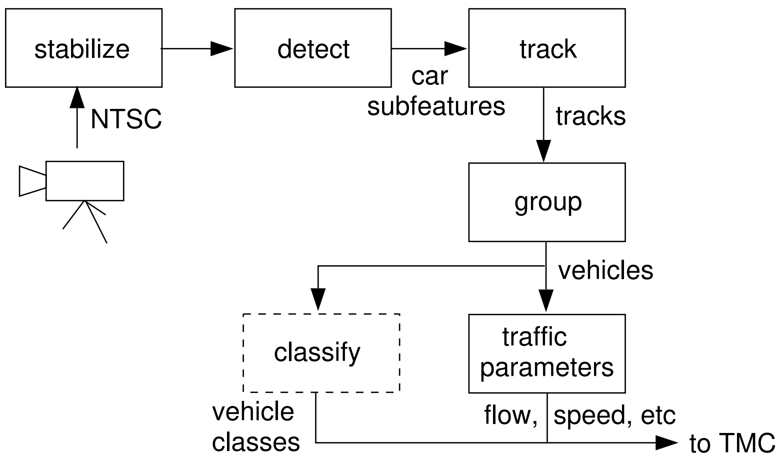


Figure 4: Block diagram of our vehicle tracking and grouping system.

4 Tracking and grouping procedures: Detailed analysis

4.1 Off-line camera definition

Before running the tracking and grouping system, the user specifies some camera-specific parameters off-line. These parameters include:

1. line correspondences for the homography (Fig. 2),
2. a detection region near the image bottom and an exit region near the image top, and
3. a fiducial point for camera stabilization.

4.2 On-line tracking and grouping

A block diagram for our vehicle tracking and grouping system is shown in Fig. 4. First, the raw camera video is stabilized by tracking a manually chosen fiducial point to subpixel accuracy. Next, the stabilized video is sent to a detection module, which locates corner features in a detection zone near the bottom of the image. These corner features are then tracked over time in the tracking module, which employs Kalman filtering. Next, sub-feature tracks are grouped into vehicle hypotheses in the grouping module. Finally, traffic parameters such as flow rate, average speed, and average spatial headway are computed from the vehicle tracks. In the future, we intend to add a vehicle classification module that will identify vehicles as automobiles, motorcycles, trucks, buses, etc. In this section, we describe the detection, tracking, and grouping modules.

Figure 5: Example corner features located by the system.

4.2.1 Sub-feature Detection

Vehicle sub-features are detected and tracked in order to be insensitive to partial occlusion. Even if part of the vehicle is obscured due to congested traffic conditions, some of the vehicle's sub-features should still remain visible.

Corner features are the chosen sub-features since they can be reliably tracked. Corners are typically defined as regions in the intensity image $I(x, y)$ where there is gradient energy in both the x and y directions. Our corner detector is based on the second moment matrix

$$A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix},$$

where A is computed at each pixel in the detection region and the derivatives I_x and I_y are averaged over small 5x5 neighborhoods. One can relate the rank of A at a point (x, y) with the type of feature present at (x, y). At corners, where there is gradient energy in both directions, A has rank 2. In edge regions, A has rank 1, and in featureless regions, A has rank 0. Thus, our corner measure is taken to be the smaller eigenvalue of A – when the smaller eigenvalue is large, the matrix tends to have rank 2.

Fig. 5 shows some example corner features detected by the system. When a corner sub-feature is detected, a small 9x9 template of the grey level image is extracted and used for correlation in the tracking module. Also, while there are some undesirable corners present near the vehicle boundaries and background, these corners will be pruned away by the feature tests employed by the tracker.

4.2.2 Feature Tracking

The tracking module tracks corner sub-features from the detection region at the bottom of the image to the exit region near the top. To address the problem of noisy measurements, we employ the Kalman filter[5] formalism to provide most likely estimates of the state of a vehicle sub-feature based on accumulated observations. In our system, the state vector contains sub-feature positions and velocities $(X, Y, \dot{X}, \dot{Y})$; vehicle acceleration is captured in the system dynamics noise process. Notice that tracking is performed in the world coordinate system. The advantage of tracking in world coordinates is that physical constraints of a vehicles motion model can be used to guide tracking. For example, the knowledge that vehicles have finite acceleration will limit the range of motion a vehicle can have in the image from frame to frame.

The measurement process in the Kalman filter is based on normalized correlation. At each time frame, the Kalman filter predicts where to search for each corner feature. This prediction is mapped back to the image plane, and then the template extracted when the corner was originally detected is correlated in a window around the prediction. The template is scaled down over time to reflect the fact that vehicles are getting smaller as they move down the road surface. We can use the position in world coordinates to predict the proper scale of the template. Once we have located the correlation peak, this measurement is mapped back onto the road plane. Finally, the standard Kalman filter equations for updating the state and error variance are employed.

Two tests are used to eliminate bad sub-feature tracks:

1. *Kalman filter innovations.* The distance between the Kalman filter prediction and the current measurement is computed and the track is rejected if the distance is above a threshold. A high value for this distance is indicative of an unstable track.

2. *Imprecise measurement test.* If the correlation values form a broad, undefined peak around the correlation maximum, then the measurement process is probably not localizing the sub-feature within the needed precision. To measure the peak's curvature, we compute the number of pixels in the correlation peak that are within a certain fraction of the peak. The track is rejected if the count is over a threshold.

Fig. 6 shows the time evolution of some example tracks, plotted as position over time. The image shown is the frame when the corners were originally detected.

4.2.3 Grouping

The purpose of the grouping module is to group together sub-features that come from the same vehicle. The central cue used by the grouper – common motion – was described already in section 3. In this section, we discuss the details of how the common motion constraint is applied to the sub-feature tracks.

The grouper organizes its task by constructing a graph over time. The vertices are sub-feature tracks, edges are grouping relationships between tracks, and connected components correspond to vehicle hypotheses. When a new sub-feature is detected and is added to the grouping graph, it is initially connected to all neighboring tracks within a certain radius in the image plane. The attitude of the grouper is that nearby tracks are compatible until they prove otherwise through relative motion. For all pairs of tracks $p_a(t)$ and $p_b(t)$ joined by an edge, the grouper keeps track of the relative displacement $d(t) = p_a(t) - p_b(t)$ as scaled by the depth-compensating factor computed from the homography. Upon each time frame, another d value is computed for each edge, and the edge is broken if either

$$\begin{aligned} \max_t d_x(t) - \min_t d_x(t) &> x \text{ threshold, or} \\ \max_t d_y(t) - \min_t d_y(t) &> y \text{ threshold.} \end{aligned}$$

This breaks the link between two tracks if there is enough relative motion between the two.

In the normal evolution of the graph, vehicles are overgrouped near the detection region since the graph is liberally connected at first. But as vehicles move down the road, they are segmented as they perform a distinguishing motion such as lane drift or an acceleration. When the last

Figure 6: Example tracks of corner features.

Figure 7: Example groups of corner features.

track of a connected component enters the exit region, a new vehicle hypothesis is generated and the component is removed from the grouping graph.

Fig. 7 shows the final groups computed for the vehicles in the tracking region (which is the middle part of the image). Corner features are indicated by circles, and there is an edge drawn between grouped corners.

4.3 Preliminary analysis of parameter measurement errors

The tracking process provides, at each processing cycle, the instantaneous positions and velocities of all tracked vehicles. As indicated in the block diagram of Figure 4 under "coarse traffic parameters", this information can be used in a straightforward way to compute the local traffic parameters listed in Table 1. The following provides a brief, preliminary analysis of the theoretical error in parameter measurement for the vision-based tracking system (required accuracies are enclosed in square brackets). Actual error rates are discussed in Section 9.

1. *Vehicle flow rate:* [± 2.5% @ 500 vehicles/lane/hour] The number of tracks initiated and confirmed per lane is integrated over a finite window. The accuracy here depends largely on the track initialization error rate, discussed above.

2. *Average vehicle speed:* [±1 mph] Estimated via a Kalman filter for each vehicle. Accuracy improves with the number of frames in which the vehicle is visible. Accuracy for a single vehicle is limited by pixel localization. If a vehicle track is followed for 100 pixels, naive methods have an average error of 1 pixel. Standard sub-pixel localization methods can reduce this to 0.2 pixels, so that 1% error is well within reach. Accuracy for the whole scene can be improved by averaging over n vehicles. This would improve the accuracy by a factor of $\sqrt{n}$, since pixel errors are independent for each vehicle.

3. *Vehicle classification:* [± 5%] This is done using the technique explained in Section 5.

4. *Lane Changes:* [± 5%] Can be estimated from current lane position. Errors should only arise from failure to detect and track vehicles. To date, our system has achieved 100% accuracy in lane change detection for those vehicles actually detected and tracked.

5. *Queue length:* [± 5%] Assuming track information is accurate, queue length is easily derived. Errors here are anticipated to arise from uncertainty as to the "end" of the queue and queue membership.

6. *Spatial headway:* [± 5%] Also easily derivable from track information. Due to the nature of Kalman filtering, the accuracy of these parameters is a function of time.

7. *Vehicle acceleration:* [± 5%] Acceleration is derived from velocity comparisons over time. A rough estimate of the accuracy for one vehicle can be obtained as follows: Break the track into two equal time intervals. For each the speed can be estimated to about 2% accuracy ($\leq$1 pixel error over a track of 50 pixels length), giving us an accuracy of $2\sqrt{2} \approx 2.8\%$ for average acceleration measurement.

5 Vehicle Classification

The vehicle classifier takes measurements of significant attributes for each vehicle's feature group and attempts to determine the class to which that vehicle belongs.

class	description
1	Motorcycles
2	Cars, Vans, and Pickup Trucks
3	Buses
4	Trucks and Recreational Vehicles
5	Large Trucks (over 40 ft)

Table 2: Vehicle Class Definitions.

5.1 Vehicle Class Definitions

The five defined vehicle classes are shown in Table 2.

5.2 Significant Attribute Definitions

Significant attributes of a vehicle are those attributes which can reasonably be expected to help the classifier discriminate among the defined classes. We chose the following five measured attributes to train our vehicle classifier: height, length, width, downstream velocity, and lateral position. Height was chosen because it separates vehicles between the first two and the last three classes; length separates class 4 from class 5 vehicles; width separates class 1 from class 2; downstream velocity separates slower moving trucks from other vehicles; and lateral position separates buses from trucks, as trucks are often prohibited from traveling in the left-most lane while buses are not (especially if the left-most lane is a carpool lane, in which case buses are encouraged to travel there).

5.3 Measurement of Significant Attributes

The grouper takes a set of feature tracks and uses common motion and other criteria to join those tracks together in a vehicle group. Each track carries with it a history of the track's position and velocity for each video frame in which that track is active. The grouping process links common tracks together but does not destroy individual track histories; these track histories are used to estimate measurements of the above significant attributes.

5.3.1 Downstream Velocity and Lateral Position

The history of each track in a given group is checked, and the common start frame (first frame in which all tracks are simultaneously active) for that group is determined. The time index for this frame is stored, and the downstream velocities and lateral positions (obtained from each track's Kalman Filter estimate) for all tracks are averaged to form an estimate of the vehicle's velocity and position. No attempt is made at this point to correct for the non-zero heights of features (see below). This causes the velocity estimates to be reported to the classifier with a slight positive bias. The overall effect of this bias will be to cause an arbitrary shift in both the classifier's decision boundaries and the data it is classifying, therefore the bias causes no appreciable increase in classification error.

5.3.2 Vehicle Height

The common start frame and common end frame (last frame in which all tracks are simultaneously active) is determined for the feature group of the vehicle being measured. For each track

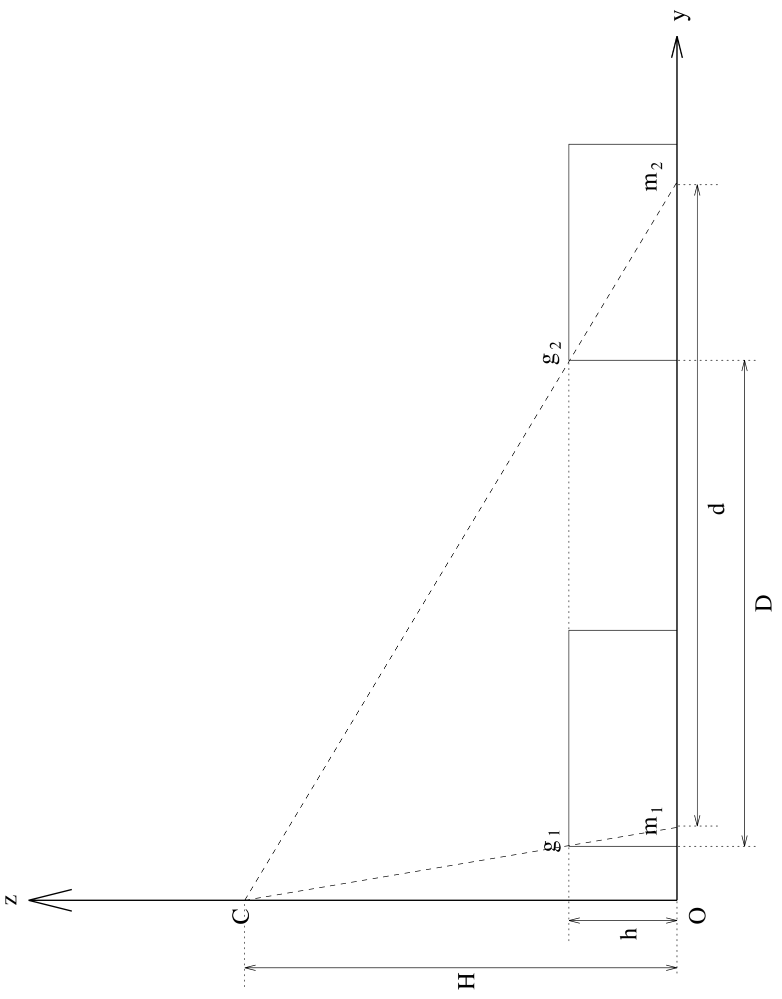


Figure 8: Calculation of Height

in that group, the apparent downstream displacement (d) is measured. Note that the apparent displacement is different from the actual displacement (D) for features with non-zero height. To calculate the height of each feature, we exploit this difference in displacements.

Figure 8 shows a typical situation. A feature (in this case, the top back of a vehicle) has y-position g_1 at time t_1, but because this feature has non-zero height h a camera mounted at point C projects this feature's y-position to m_1 on the ground plane. Similarly, at time t_2, the actual y-position of the feature is g_2 but the camera projects the y-position to m_2 on the ground plane. Thus, the actual y-displacement D is given by:

$$D = g_2 - g_1 \tag{1}$$

while the measured y-displacement d is given by:

$$d = m_2 - m_1 \tag{2}$$

Thus, if the camera height H and the actual y-displacement D are known, we can use similar triangles[1] $\triangle C g_1 g_2$ and $\triangle C m_1 m_2$ to solve for h:

[1] We are abusing notation here by using the symbols $m_1, m_2, g_1, andg_2$ as both displacements and point labels.

$$h = H(1 - \frac{D}{d}) \tag{3}$$

Of course, to use Equation 3 we must know the actual displacement D. If there were a feature of zero height (say, a corner made between a tire and the road surface), its actual and measured y-displacements would be equal. Moreover, the measured y-displacement of the feature nearest the ground plane will be the smallest in any group. Thus, as an approximation to D we use d_{min}, that is, the shortest y-displacement measured for a given feature group for the common frame interval (the number of frames between common frame start and common frame end).

Notice that the calculation of vehicle height is independent of the choice of origin on the ground plane. If the origin is not mounted directly over the origin then each term in both Equations 1 and 2 will have an offset term added. As can be readily seen from these two equations, this offset term will be subtracted out, so the calculation of d and D will be unaffected, which means the calculation of h will also be unaffected.

This approach suffers from one obvious source of error: if features of extreme height (either zero or maximum) are not tracked from the entry to exit zones, the estimate of height will be erroneous. However, if the error is such that the height of each vehicle is underestimated, but that the estimate is a monotonic function of the vehicle's actual height, the classifier can still distinguish between vehicles by compressing its decision boundaries in the height dimension during its training phase.

5.3.3 Vehicle Length and Width, Camera Position Known

We present here a method for estimating vehicle length. The procedure for calculating vehicle width is analogous, substituting y-displacements and origin offsets with x-displacements and offsets.

Using Equation 3 above, and the approximation $D \simeq d_{min}$ we can obtain an estimate of the height of each feature. To estimate the vehicle's length, we make a few simplifying assumptions: first, that the y-axis in the world coordinate system is parallel to the (downstream) direction of travel; second, that the vehicle we are trying to measure is aligned with this direction; and third, that the y-displacement between a point on the ground plane directly beneath the camera and the origin in the world coordinate system is known[2].

Figure 9 shows the situation. Two feature points of height h_1 and h_2 from the same vehicle group are shown as arrows pointing up at y-displacements g_1 and g_2 respectively. The world origin O is shown displaced down the y-axis a distance K_y from the point on the ground plane directly beneath the camera (which is mounted at point C). Again, it is assumed that this displacement K_y and the camera height H are known. Also known are the camera's projections of the feature points onto the ground plane, measured with respect to point O at y-displacements m_1 and m_2 respectively. Using similar triangles, we can show for any feature i:

$$g_i = m_i(1 - \frac{h_i}{H}) - K_y(\frac{h_i}{H}) \tag{4}$$

Our length estimation algorithm makes use of Equation 4 by calculating the true y-displacement g_i from the measured displacement m_i at the common start frame, then by choosing the maximum (g_{max}) and minimum (g_{min}) from among these points. The estimate of vehicle length is then simply $g_{max} - g_{min}$. The common start frame is chosen because (assuming departing traffic) the quantization error in the image translates to the smallest error in world coordinates when the features are closest to the camera. In addition to this error and the errors introduced by using calculated height (discussed above), this approach also suffers if features are not tracked at

We hope the meaning of these symbols will be clear from their context.

[2]The origin is normally chosen to be a point in the camera's field of view. Unless the camera is pointed straight down, the point on the ground plane directly beneath the camera is not visible, so these two points are almost always different from each other.

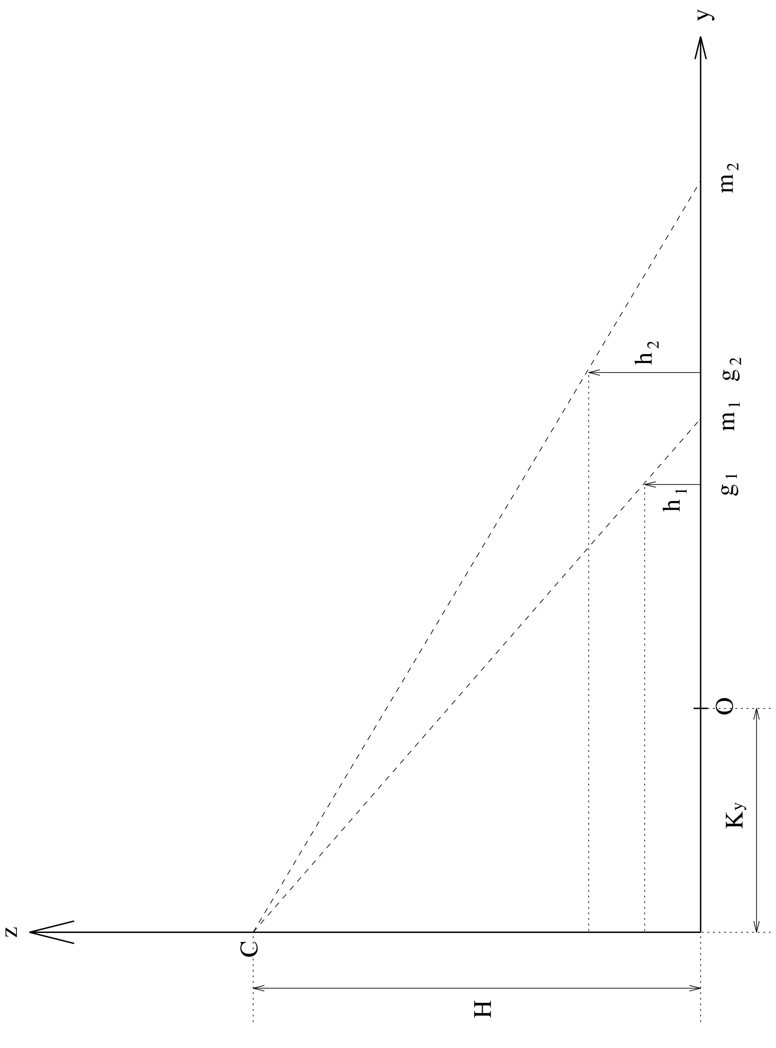


Figure 9: Correcting Downstream position using Height

extreme y-displacements, or if the vehicle being measured is sufficiently turned from the world y-axis (in which case the projection of the vehicle's length onto the world y-axis is estimated). Again, if the errors are systematic and sufficiently well-behaved, the classifier can compensate by adjusting its decision boundaries during its training phase.

5.3.4 Vehicle Length and Width, Camera Position Unknown

If the displacement K_y (and its counterpart, K_x) are not known, then we can still estimate vehicle length (and width) using pairs of features whose heights are approximately equal. Referring again to Figure 9, if we wish to know the true distance $g_2 - g_1$ we can solve for this quantity using Equation 4 twice:

$$g_2 - g_1 = m_2(1 - \frac{h_2}{H}) - m_1(1 - \frac{h_1}{H}) - K_y(\frac{h_1 - h_2}{H}) \tag{5}$$

Now, if $h_1 \simeq h_2 \simeq h$, where h is usually chosen to be the arithmetic average of h_1 and h_2, then for sufficiently small values of K_y, the last term in Equation 5 is neglible and can be discarded. Setting the h_i's in Equation 5 to h and factoring common terms, we obtain the following approximation:

$$g_2 - g_1 \simeq (m_2 - m_1)(1 - \frac{h}{H}) \tag{6}$$

Note that it is not necessary to know H (camera height) explicitly to use Equation 6; it suffices to know the ratio $\frac{h}{H}$, which can be estimated directly using measured track lengths. Combining Equations 6 and 3 with the approximation $D \simeq d_{min}$ and rearranging terms, we obtain an alternative form of the length estimate in terms of measurable parameters:

$$g_2 - g_1 \simeq (m_2 - m_1)(\frac{d_{min}}{d}) \tag{7}$$

If we wish to be consistent with our choice of h as the arithmetic average of h_1 and h_2, then it can be shown that we should choose d to be the harmonic average of d_1 and d_2.

This approach suffers from the same types of errors as the approach using known camera position, with the additional caveat that if the extreme feature points of length (or width) are detected but occur at substantially different heights, the length between them is never calculated because their heights are not close enough to each other. A human can adjust the height-difference threshold by balancing the tradeoff between the above problem (seen when the threshold is set too low), and the inaccuracies introduced when features at very different heights are compared (seen when the threshold is set too high).

5.4 Classification Using Measured Data

The classifier takes a set of training examples which have been labeled with ground truth by a human and searches for patterns that would help it classify new examples. After some experimentation we chose to use normal density estimation [3]. The procedure begins with the assumption that, given a class of vehicles (say, trucks) the distribution of measured attributes will be approximately normal. The (unbiased) mean and variance are then calculated conditioned on that class, so that given enough examples, we obtain an estimate of the mean length, width, height, velocity, and lateral position of each vehicle class, along with the covariance matrix for these classes. It is then possible to guess which class a new instance belongs to by measuring the distance from the new instance to the mean of each class. Here the distance is not the usual Euclidean distance; it is one which takes into account the spread of values for each class (using the covariance matrices) as well as the prior probabilities for each class, which can be set by the user. The results are given in Section 9.2.5.

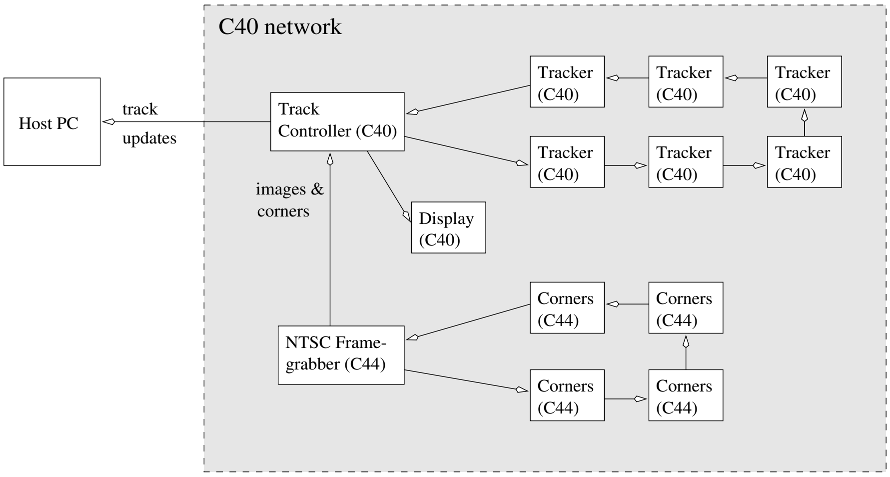


Figure 10: The C40 network used for feature tracking.

6 Hardware Port

We have implemented the tracker on a network of 13 Texas Instruments C40 DSPs obtained from Traquair Data Systems. The decision was made early on to run the grouper on the host PC, running Linux. This was made for two reasons. Firstly the grouper has to store the complete trajectories of all current features, and our C40 modules have limited memory capacity, so are not suitable for storing large and especially varying amounts of memory. The tracker only has to store the *latest state* of each feature track. Secondly, the host PC is a quite powerful 150MHz Pentium (equivalent to 3 or 4 C40s) and would otherwise not be utilized except to load the executable code onto the C40 network.

The C40 processor is well suited for the convolution/correlation operations common in low-level vision. It has two memory buses, and parallel read/write operations may be performed in parallel on alternate busses. The computationally heavy operations in the tracking algorithm are convolution in the feature detector and correlation in feature tracker, so we are able to take advantage of the C40's performance in this area. The six 20MByte/sec communication links allow each processor to communicate with its neighbours and/or the external host.

6.1 Overall Architecture

The network connectivity is shown in Figure 10. The network consists of:-

- An NCSC frame-grabber C44 module (C44s are cut-down C40s with only four communication links).
- A quad C44 module (four processors) for corner feature detection.
- A large memory module (8M RAM) for maintaining the state of current feature tracks (track controller).
- Six fast SRAM modules for feature tracking.
- A VGA graphics module for displaying results.

The processors are arranged in two loops, each of which is operated as a pipeline feeding back to its source. The two pipelines are controlled by the frame grabber and the track controller; they compute the corner features and the track updates respectively. Four C44 processors are assigned to corner detection. They each process one quarter of the user-defined detection region. The corners are fed back to the frame-grabber, which passes them along with the original image to the track controller. A simple efficiciency gain is achieved by sending the image first, since the track controller can then update the existing tracks while the corners are computed.

6.2 Updating tracks

The job of the track controller is to maintain the state of the complete list of current tracks. It does this by receiving updates for existing tracks from its pipeline of six C40s, and creating new tracks at positions indicated by the corner detector. The main problem with parallelizing this task is load balancing, i.e. keeping each pipeline processor occupied for as much time as possible. In contrast to the corner detector, which is a single large task which must be split up into chunks, the task of feature tracking presents us with a large number of distinct and independent track updates, where the goal is the take the state of a track at the previous image together with the new image data, and perform the correlation, sub-pixel localization and Kalman filter update. The most obvious architecture for achieving this would be the processor farm, in which each processor would have a separate link to the track controller, and upon completing the update of a given track, would immediately be sent another to update.

This architecture achieves automatic load balancing, but the disadvantage is that it requires a separate duplex link for each tracker processor. This is impossible because three of the six track controller links are taken by the connections to the frame-grabber, the display and the host PC. To reduce the link requirement we again arrange the tracker processors as a pipeline (Figure 10). The tracker C40s each update one sixth of the tracks. Since track updates are fairly homogeneous tasks, this achieves reasonably good load balancing. The steps involved in updating tracks given a new image are as follows:-

1. The track controller passes the number of tracks around the pipeline.
2. Each track in turn is passed into the pipeline.
3. Each tracker C40 keeps one sixth of the tracks and passes the rest on down the pipeline.
4. The tracks are updated (correlation, Kalman filter cycle).
5. Updated tracks are passed out to track controller.

The track controller also eliminates tracks marked by trackers as completed, initialize new tracks around corner features, and passes all the new track information to the grouper process running on the host PC via a socket interface.

6.3 Software development

A major problem with specialized real-time hardware is inflexible software. The typical development model is to implement algorithms first on standard workstation platforms, and then to transfer the implementation to the hardware, introducing whatever hardware-specific modifications that are necessary. This leads to problems when further development of the system is necessary, because the software development tools for transputer systems are usually quite rudimentary. Our solution to this problem has been to constantly maintain identical implementations on both the C40 system and workstations. Improvements are debugged first on the workstation implementation and then transferred to C40s. Software compatibility between the two implementations is maintained at the highest possible level, by which we mean the level above which the C40 links are involved. Thus the source code for the update of a single feature track, or corner detection in an image window, is almost identical, with only a few modifications necessary to take advantage of the peculiar characteristics on the C40 processor.

About 90% of the C source code is held in the vision library “Horatio”, which we have written for the purpose of developing real-time vision applications on transputer networks. Only the remaining 10% of source code is application specific. A on-line documentation system for Horatio is in progress.

The performance of the tracker is 7.5Hz in uncongested traffic, dropping to 3Hz in congested traffic, where many more tracks are in progress at any given time. This reduction in speed does not of itself lead to a reduction in performace of the tracker, since vehicle speeds in congested traffic are reduced, and so the requirement for tracking rate is naturally reduced. Problems may occur in situations where one or more lanes are congested while other lanes are free-flowing, but this situation is relatively rare.

6.4 Field Deployable system

The computational system is placed in a weatherproof housing certified to operate within the environmental guidelines. The camera can be placed remotely in its own environmental casing, or existing surveillance cameras already in operation can be used. The system to be placed in the field is diagramed in figure 11.

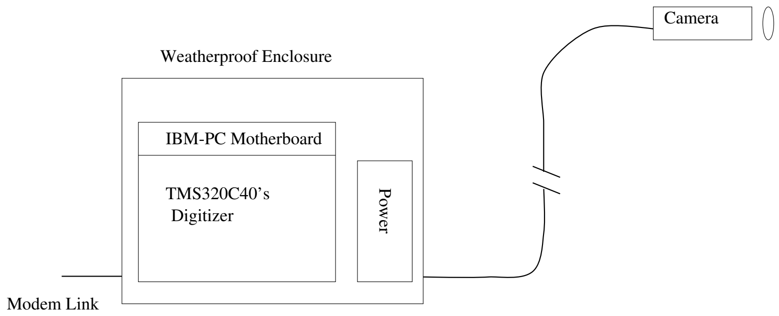


Figure 11: Schematic of field deployed system. The camera and processing are separated in order to place computing hardware in a more accessible and/or protected location.

7 Parameters computed at the TMC

Link travel times and origin-destination counts cannot be locally determined with data from only one camera. Instead, one or more computers, typically located at the traffic management center, calculate these parameters based on vehicle reports from multiple cameras. We will call this centralized system the "TMC." The TMC is also responsible for real-time collation and presentation of local and global traffic parameters.

7.1 Overall design

The TMC computes travel times and O/D counts by matching individual vehicles appearing at two or more camera sites. As a vehicle is reidentified at a sequence of camera sites, the TMC builds up a "path," which records the route taken by the vehicle through the freeway network. The complete set of recorded paths can be queried to establish any desired O/D count. Figure 12 summarizes the overall processing scheme of the implemented system. Figure 13(a) shows an example screen from the TMC display. The displayed information is computed in real time from incoming vehicle reports and uses a customizable window size for averaging and a customizable update frequency.

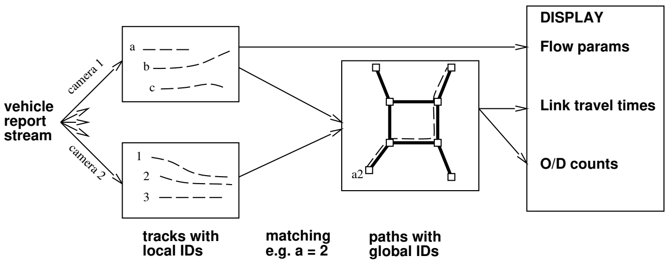


Figure 12: Overall design of the TMC subsystem.

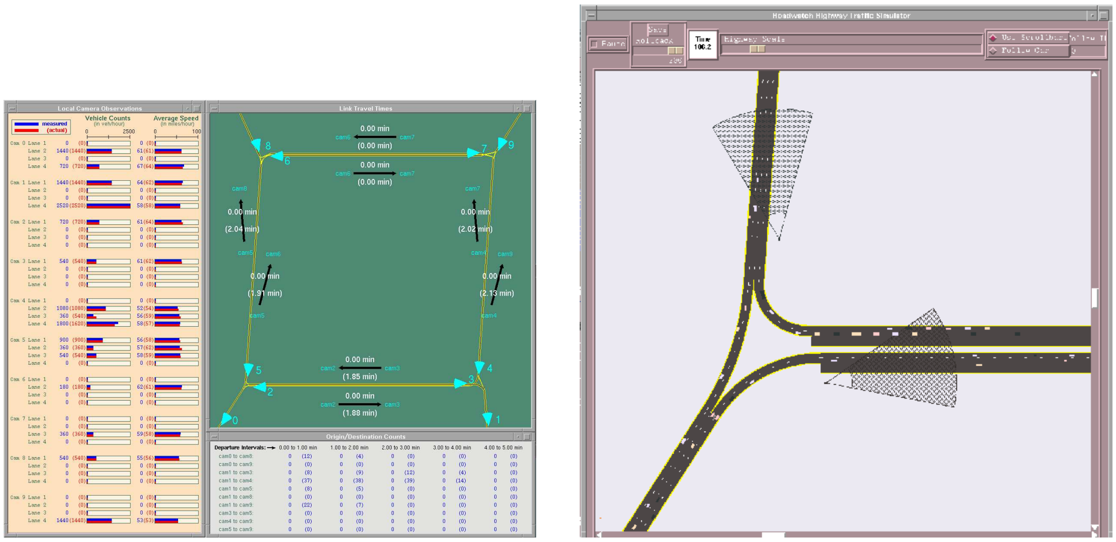
In addition to the TMC system, we also implemented a complete microscopic freeway simulator capable of simulating several hundred vehicles in realistic traffic patterns. The simulator includes virtual cameras that can be placed anywhere on the freeway network and that transmit real-time streams of vehicle reports to the TMC (see Figure 13(b)). The reported data can be corrupted by any desired level of noise. We have found the simulator to be an invaluable tool for designing and debugging the TMC algorithms, allowing development of the TMC in parallel with the development of the vehicle tracking systems.

The data structures and algorithms implemented for collation and presentation of traffic parameters are not of technical interest and will not be discussed further in this report. Computing reliable matches is the central task and presents a number of difficult technical problems. The approach described below achieves good performance even under difficult conditions.

7.2 Vehicle matching

The key task of the TMC is to identify matching vehicles based on vehicle reports from upstream and downstream camera pairs.[3] This is closely related to the traditional "data association"

[3] In the case of freeway splits and merges, any given "downstream" camera may have more than one "upstream" camera, and vice versa. In this case, we merge the vehicle report sequences as appropriate.



(a) (b)

Figure 13: (a) TMC real-time display showing lane-by-lane speeds and flows at 10 sites together with link travel times and O/D counts. (b) Screen dump showing part of the simulated freeway system with camera fields of view indicated.

problem from the tracking literature, in which new "observations" (from the downstream camera) must be associated with already-established "tracks" (from the upstream camera). There are several issues specific to vehicle matching that must be resolved:

1. Sensor noise and bias are large, unknown, time-varying, site-dependent, and camera-dependent.
2. Successive observations of a vehicle may be widely separated in time and space.
3. Vehicle trajectories are highly unpredictable and highly correlated.
4. The usefulness of various features (e.g., color and speed) for matching varies enormously depending on conditions (e.g., night-time and congested traffic).
5. Vehicles may appear downstream via unmonitored onramps, may disappear via unmonitored offramps, or may simply fail to be detected at one or both cameras.
6. The number of tracks and observations is potentially very large, leading to problems of computational complexity.

To address these issues, we have undertaken a first-principles analysis to compute the probability of candidate matches, taking all relevant sensor information into account. We use online re-estimation of our probabilistic sensor models and motion models, and efficient algorithms to compute most likely matches even in the presence of new and missing vehicles. An additional benefit of the probabilistic analysis is that the system can compute the reliability of each match and select those exceeding a threshold related to the performance target.

7.2.1 Computing most likely matches

Our system receives as input a continuous stream of chronologically ordered vehicle reports from the trackers that operate at each camera, and it outputs proposed matches that extend the paths for vehicles observed at one camera to observations at adjacent downstream cameras.

Each vehicle report corresponds to a vehicle observed by one camera and consists of measurements of various **features** such as vehicle lane, size, and color. The matching algorithm is designed to be independent of the specific features used; new features of arbitrary complexity, informativeness, and noise level can be added without changing the algorithm. Adding informative features will typically improve performance.

We have divided our vehicle matching system into three major components. The first component takes the stream of vehicle reports and computes for each **candidate match**, i.e., each possible pairing of observed vehicles from adjacent cameras, the probability that the features recorded at the downstream camera would appear given the features recorded at the upstream camera and given that both vehicle reports were generated by the same vehicle. This probability is called the **appearance probability** and is computed based on empirically estimated models for predicted arrival time and for other measured features such as lane, size, and vehicle color.

Formally, if f_i^u is the set of features observed by upstream camera **u** of vehicle v_i, and if f_j^d is the set of features observed by downstream camera **d** of vehicle v_j, then the appearance probability for that candidate match is:

$$P(f_j^d | f_i^u, v_i = v_j)$$

which we will write as $P(f^d|f^u)$ where no confusion is possible. It is important to note that the appearance probability is *not* the probability that the candidate match of i and j is correct. The latter probability can only be computed by taking into account all other vehicle observations, as described in the following paragraphs.

Given the appearance probabilities for each candidate match, the second component of our system computes the best assignment set, in which each observation at the upstream camera is either assigned to an observation at the downstream camera or marked as having exited the freeway, and each observation at the downstream camera is either assigned to an observation at the upstream camera or marked as having newly entered the freeway.

The problem of determining the best assignment set, where some assignments are regarded as better than others, is known in graph theory as the weighted bipartite matching problem. By organizing the appearance probability information into an "association matrix" in which the rows correspond to the observations at the upstream camera, the columns correspond to the observations at the downstream cameras, and the entries correspond to the negative logarithm of each pairing's appearance probability, the system can utilize any of several well-known weighted bipartite matching algorithms to obtain a minimum weight assignment set, which corresponds to the most likely set of candidate matches. (Additional columns and rows are added before matching to handle the cases where vehicles entered or exited the highway in between the two cameras.)

The third component of our system decides which matches to accept from the best assignment by comparing individual matches with a designated reliability threshold. This is done by "forbidding," in turn, each match contained in the best assignment set and then measuring the reduction in likelihood for the new best assignment set under this restriction. Matches whose forbidding results in significant reduction are deemed reliable, since this corresponds to a situation where there appears to be no other reasonable assignment for the upstream vehicle in question.

7.2.2 Measured vehicle features

Currently, the vehicle reports include the following features:

t, time of observation

x, lateral position across highway (reduced to lane, i.e., 1, 2, 3, etc.)
y, forward position in lane
$\dot{x}$, lateral velocity
$\dot{y}$, forward velocity
w, vehicle width
l, sum of vehicle length and height
h, mean vehicle color hue
s, mean vehicle color saturation
v, mean vehicle color value
C, histogram of color distribution over vehicle pixels

7.2.3 Sensor and motion models

The appearance probability is currently computed as the product of separate models as follows:

- lane (x): discrete distribution $P(x^d|x^u)$;
- size (w, l): multivariate Gaussian

$$P(w^d, l^d | w^u, l^u) = N_{\mu_{w,l}, \Sigma_{w,l}}(w^d - w^u, l^d - l^u)$$

- color (h, s, v): multivariate Gaussian

$$P(h^d, s^d, v^d | h^u, s^u, v^u) = N_{\mu_{h,s,v}, \Sigma_{h,s,v}}(h^d - h^u, s^d - s^u, v^d - v^u)$$

- arrival time (t): univariate Gaussians conditioned on upstream and downstream lane

$$P(t^d | t^u, x^d, x^u) = N_{\mu_t^{x^d,x^u}, \sigma_t^{x^d,x^u}}(t^d - t^u)$$

The arrival time model is particularly important, since it drastically reduces the number of vehicle pairs that are considered to be plausible matches. The parameters $\mu_t^{x^d,x^u}$ and $\sigma_t^{x^d,x^u}$ represent the mean and standard deviation of the predicted link travel time for cars that start upstream in lane x^u and end up downstream in lane x^d. This allows the system to accurately model, for example, the fact that the link travel time for cars in the HOV lane is usually substantially less than the link travel time for cars in other lanes.

Because traffic and lighting conditions change throughout the day, our system uses online (recursive) estimation for the appearance probability model parameters. As new matches are identified by the vehicle matcher, the parameters are updated based on the observed feature values at the upstream and downstream sites. Figure 14 shows a sample set of x values for matched vehicles, from which $P(x^d|x^u)$ can be estimated. In order to track changing conditions, we use online exponential forgetting. For example, if a new match is found for a vehicle in lane x^u upstream and lane x^d downstream with link travel time t, then the mean is updated as follows:

$$\mu_t^{x^d,x^u} \leftarrow \gamma \mu_t^{x^d,x^u} + (1 - \gamma)t$$

The γ parameter, which ranges from 0.0 to 1.0, controls the effective "window size" over which previous readings are given significant weight. It can also be adjusted according to the probability that the new observed match is in fact correct.

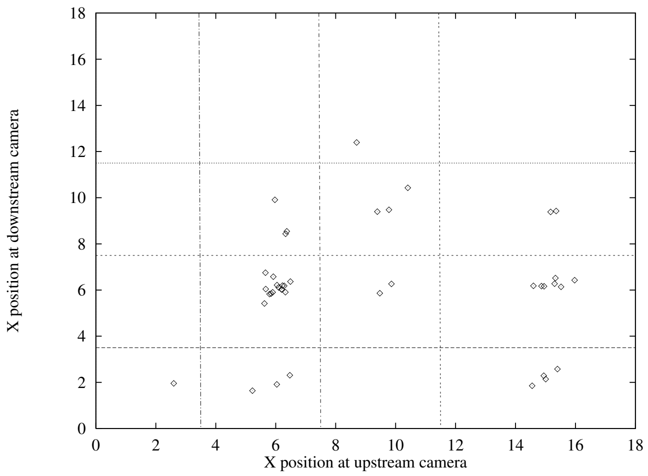


Figure 14: Diagram showing observed upstream and downstream x-position data for a sample of 41 matched vehicles from the Mack Rd. and Florin Rd. cameras. The horizontal axis corresponds to upstream x-position and the vertical axis corresponds to downstream x-position. Each marked point corresponds to a single matched vehicle. Lane dividers are shown as horizontal and vertical lines. For example, 13 vehicles are observed in lane 4 (onramp, highest x values) upstream, of which 7 end up in lane 2 (middle lane) downstream, indicating that $P(x^d{=}2|x^u{=}4) \approx 0.54$.

7.3 Computation of travel times and O/D counts

Link travel times between each camera pair are currently calculated by averaging the observed travel times for matched vehicles. Averaging is done over all matched vehicles falling within the data window specified by the TMC display manager.

O/D counts are extracted from the stored path information for matched vehicles simply by adding the appropriate entries corresponding to all possible routes from origin to destination and dividing by the fraction of vehicles matched on those routes. This procedure assumes that the matched vehicles are representative of the actual population.

Results are given in Section 9.3. Preliminary indications are that link travel times and O/D counts for camera pairs can be estimated well within the required error bounds. O/D counts for paths involving three or more sites will be inaccurate at current detection and matching rates, as explained in Section 9.3.

We are continuing to improve the effectiveness of the TMC. The main priorities include better handling of vehicles that enter and exit the freeway between cameras, faster operation of the overall system, and more extensive testing on longer data sequences. In addition, we expect that more accurate size measurement and the addition of more advanced features will enhance matching performance considerably.

8 Testing methodology

8.1 Introduction

Our testing plan used both comparative tests against existing detection methods and human verification to validate the vehicle tracker. For human verification, we used the largest possible observation area, A, as outlined in Appendix D. Specifically, L was set to the maximum possible detection area. For the automated comparison tests against loops, the observation area, A, is defined by the pre-existing detector. In other words, we set L equal to the loop detector or speed trap length. We are aware of the problems associated with loop detectors, but feel the large quantities of available data make this an attractive starting point. We checked all anomalies between the loop and tracker data and used a random sample to verify both measurement techniques.

When testing against existing methods, we must verify that the tracker is not over- or under-sampling vehicles and quantify any problems with the existing detector. To this end, we hand calibrated several test sets under various conditions for high accuracy testing. Then, we ran large batch studies for a large sample size and investigated any anomalies. Note that many of the manual tests also serve to verify vehicle detection.

We used the following procedures in development as well as testing. Note that the testing video tapes are separate from the development tapes.

8.2 Individual parameter testing

8.2.1 Vehicle flow

To test vehicle flow, we collected concurrent video and loop data from five sites (see appendices). We hand calibrated several test sets to verify the accuracy of the tracker and the loop detectors. Next, we compared the tracker output against the loop data over extended periods and investigated any discrepancy. Thus, we were able to verify the tracker over a large data set with our available manpower. The steps for our primary testing were as follows:

1. Collect video of vehicles passing over a loop detector and the corresponding loop data at the highest possible sampling rate (1/60th of a second).
2. Calibrate loop and video data clocks.
3. Manually calibrate several test segments using a spatially long detection region (i.e., L set to maximum possible for the viewing angle).
4. Create a virtual loop for the tracker over the actual loop detector (a fixed rectangle in the time space plane of length L equal to the real loop detector length). Measure Edie's generalized flow for the real and virtual loop detector data.
5. Compare the results (error) and investigate any discrepancy.
6. Compare virtual loop detection (short L) against our normal methods (longest possible L) and investigate any discrepancy.

Note that the primary issue when testing flow is if the tracker is over/under sampling vehicles. Although it is not explicitly stated, the other manual calibration tests provide further verification of sampling by the tracker.

8.2.2 Average speed

Following the same logic as the vehicle flow testing, we examined concurrent video and loop data to test average speed measurements. The testing steps were as follows:

1. Collect video of vehicles passing over a two loop speed trap and the corresponding loop data at the highest possible sampling rate (1/60th of a second).
2. Calibrate loop and video data clocks.
3. Manually calibrate several test segments using a spatially long detection region (i.e., L set to maximum possible for the viewing angle).
4. Create a virtual loop for the tracker over the actual speed trap (a fixed rectangle in the time space plane of width L equal to the length of the real speed trap). Measure Edie's generalized velocity for the real and virtual loop detector data.
5. Compare the results (error) and investigate any discrepancy.
6. Compare virtual loop detection (short L) against our normal methods (longest possible L) and investigate any discrepancy.

8.2.3 Link travel time

See O/D tracking below.

8.2.4 Vehicle classification

To test vehicle classification, we relied on human judgment to verify tracker performance. The testing steps were as follows:

1. Human user classifies all vehicles in a given sequence; the results are stored to disk.
2. Using a subset of the human-classified data, train the classifier.
3. Using the remainder of the human-classified data, test the output of the classifier against the human-derived classification.

8.2.5 Lane changes

As with vehicle classification, we relied on human judgment to answer simple questions about the tracker performance detecting lane changes. Specifically, "Does the detected lane change correspond to an actual lane change?" "Are there any lane changes that are undetected?" To prevent viewer overload, these parameters can be measured independently on separate runs. The testing steps were as follows:

1. Film a location with a high density of lane changes (e.g., a weaving section)
2. Select the interface between two lanes and a specific vehicle to indicate the start of the observation period.
3. A human observer watched the interface and pressed a key for each maneuver while the computer records the time of each key press. Repeat for all other lane to lane interfaces in the detection region. In this fashion, we were able prevent user overload and focus on one lane at a time.

4. Using automated procedures, compare human results to tracker results.

8.2.6 Queue length (distance)

Because of the variable location of the end of queue, it would be prohibitively expensive to automate this test. To simplify the observer's task, we calibrated the video image before testing and superimposed a distance metric for validation. The observer verified that the reported end of queue corresponded to the correct distance mark. The testing steps were as follows:

1. Manually calibrate the distance from the stop bar in 10 ft increments.
2. Superimpose distance marks on video and have the tracker continually display its estimated queue length. Repeat for each lane.
3. Observer presses a key corresponding to the calibration mark closest to the end of queue whenever the tracker is in error.

8.2.7 Queue length (number of vehicles)

Again, this parameter required manual calibration. In this case, the observer simply kept track of the number of vehicles in the queue and compared it to the tracker estimate.

8.2.8 Spatial headway (inverse of time average density)

Following the same logic as the vehicle flow and average speed testing, we examined concurrent video and loop data. The testing steps were as follows:

1. Collect video of vehicles passing over a two loop speed trap and the corresponding loop data at the highest possible sampling rate (1/60th of a second).
2. Calibrate loop and video data clocks.
3. Manually calibrate several test segments using a spatially long detection region (i.e., L set to maximum possible for the viewing angle).
4. Create a virtual loop for the tracker over the actual speed trap (a fixed rectangle in the time space plane of width L equal to the length of the real speed trap). Measure Edie's generalized density for the real and virtual loop detector data.
5. Compare the results (error) and investigate any discrepancy.
6. Compare virtual loop detection (short L) against our normal methods (longest possible L) and investigate any discrepancy.

8.2.9 Acceleration

As a coarse test of acceleration, using the derivatives of flow and density, we observed wave speeds as disturbances pass over the surveillance region. We then verified that the measured accelerations were compatible with the disturbance speeds. In other words, we checked the microscopic measurements against the macroscopic features.

8.2.10 O/D tracking

We hand-calibrated vehicle matches for several sample periods using the human observer and frame grabbing for verification. Link travel times were computed by averaging over all matched vehicles. O/D counts could only be obtained for the two consecutive sites, as we were unable to film simultaneously at more than two wites in sufficiently close proximity.

8.3 Test conditions

We have tested the widest range of environmental conditions available, including: day, night, twilight, congested, uncongested, and some adverse weather.

8.4 Primary test sites

For our study, we used six primary test sites. The particular features for each site are outlined below:

8.4.1 Sacramento Test Site, Northbound

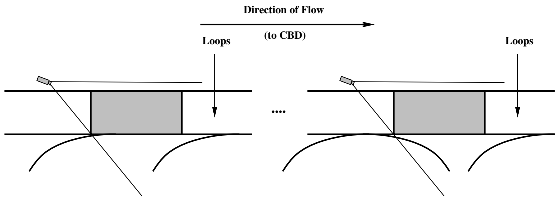


Figure 15: Schematic diagram of the test site.

(a)

(b)

Figure 16: (a) Typical image from Mack Rd. camera. (b) Typical image from Florin Rd. camera.

Location: Highway 99 northbound between Mack Rd and Florin Rd. Two camera locations approximately two miles apart.
Features:

1. Color cameras, with similar angles relative to the traffic.
2. Approximately two miles between the cameras, with no turns or other ramps not shown in the figure.
3. Approximately even with the upstream camera, an HOV lane forms (main line goes from two lanes to three), so there is significant mixing between the two locations, but little congestion.
4. There are two on ramps at the upstream location, one ramp enters within the surveillance area, the second ramp is visible within the surveillance area but enters downstream. The poor angle and short travel distance precludes tracking on the second ramp.
5. The downstream site has two off ramps and one on ramp outside of the field of view.
6. The upstream loops are just beyond the surveillance area while the downstream loops are within the surveillance area.

8.4.2 Sacramento Test Site, Southbound

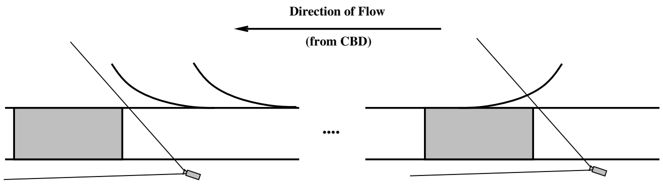


Figure 17: Schematic diagram of the test site.

Location: Highway 99 southbound between Mack Rd and Florin Rd. Two camera locations approximately two miles apart.
Features:

1. Color cameras, with similar angles relative to the traffic.
2. Approximately two miles between the cameras, with no turns or other ramps not shown in the figure.
3. Approximately even with the downstream camera, an HOV lane ends (main line goes from three lanes to two), so there is significant CONGESTION; however, most of it is outside the field of view (i.e., between the cameras).
4. There is one on ramp at the upstream location, I believe it enters within the surveillance area.
5. The downstream site has two off ramps outside of the field of view.
6. No loop detector data for this segment.
7. Upstream site has three lanes while the downstream site only has two.

8.4.3 Walnut Creek Test Site 1, I-680 Southbound

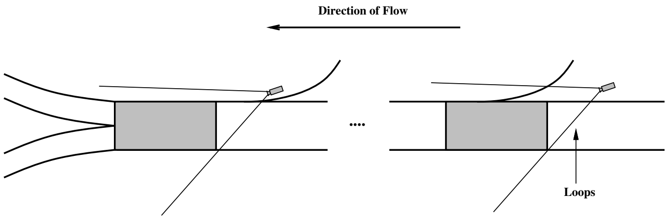


Figure 18: Schematic diagram of the test site.

(a) (b)

Figure 19: (a) Typical image from upstream camera. (b) Typical image from downstream camera.

Location: Just north of the I-680/Hwy 24 interchange. Two camera locations approximately one half mile apart.
Features:

1. black and white cameras.

2. Approximately one half mile between the cameras, a gentle curve between the two locations, no off ramps and the on ramps enter within the field of view.

3. Only have loop data for just upstream of the upstream location.

4. four lanes at both sites with a diverge downstream of the second camera. In the morning, we see congestion first in the two left lanes and then a shift to congestion in the two right lanes. During the period of surveillance, the morning peak coincided with the sun rise, so we have light/dark transition data from these locations.

8.4.4 Walnut Creek Test Site 2, Hwy 24 Westbound

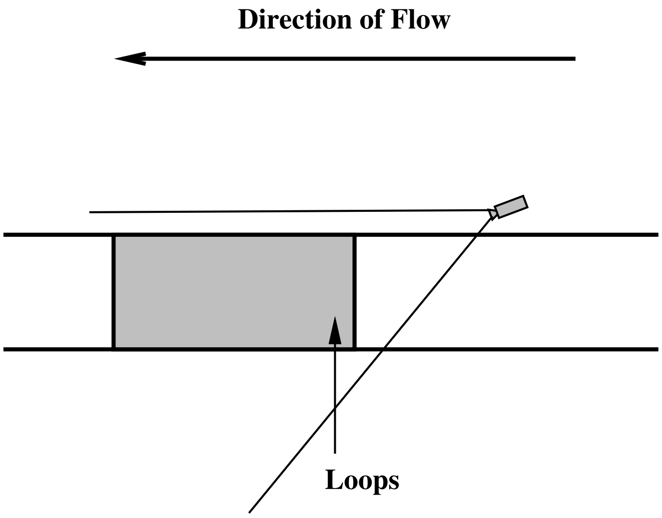


Figure 20: Schematic diagram of the test site.

Figure 21: Typical camera image.

Location: Just west of the I-680/Hwy 24 interchange.
Features:

1. black and white camera on an overpass, directly over the outside lane.
2. No ramps at this location, however, it is at the crest of a hill, so there is no congestion here. This site was selected for the good camera angle and loop placement.

8.4.5 San Jose Intersection Test Sites

Figure 22: Typical camera image.

Location: Several intersections in and around the San Jose CBD.
Features:

1. Several sites with color cameras and wide range of views, including approaching and departing traffic.

2. No secondary detector data (e.g., loop data).

Sequence	Description	Length	N	G
Highway 55	heavy congestion	2:46	238	203
Florin Road	mix of free flow and congestion	2:46	244	260
Mac Road	mix of free flow and congestion	1:00	69	68
Florin Road	night	1:20	58	64
San Jose	urban intersection	1:45	34	34

Table 3: Video sequences for laboratory testing. Length is in min:sec, N is the number of actual vehicles (counted by hand) and G is the number of reported vehicle groups.

(a)

(b)

Figure 23: (a) Typical image from Highway 55 camera. (b) Typical night image from Florin Rd. camera.

9 Results

Testing and analysis has proceeded on three levels: vehicle level, single site level and multiple site level. At the vehicle level, we have analyzed the underlying vehicle detection algorithm and the results are reported in terms of vehicles. At the single site level, we have generated and tested the traffic flow parameters that can be observed from a single camera (flow, velocity, etc.) and are reported by lane. Finally, at the multiple site level we test origin/destination matching and travel time estimation.

9.1 Vehicle level results

Our tracking and grouping system has been laboratory tested on a set of 5 videotapes covering a range of scene conditions, including congestion, free-flow, night, and an urban intersection (see Table 3 and Figure 23). Vehicle ground truth was manually defined for each sequence. For a particular vehicle, ground truth is a binary mask outlining the vehicle, and it is usually defined in a single "representative" frame for the vehicle. The number of ground truths is denoted as N in Table 3, and the number of reported groups is G.

Table 4 shows the performance of our system, as well as the distribution of errors. A separate

Sequence	true match	false neg.	over-seg	false pos.	over-group
Highway 55	75.2%	18.5%	2.9%	2.9%	1.9%
Florin Road	89.7%	1.6%	5.7%	1.5%	1.5%
Mac Road	95.6%	1.5%	0.0%	1.4%	1.4%
Florin Road (night)	91.4%	6.9%	1.7%	14.0%	0.0%
San Jose	85.3%	2.9%	5.9%	0.0%	2.9%

Table 4: Performance of the tracking/grouping system on the test sequences.

automatic evaluation program compares the vehicle ground truths against the groups reported by the tracker/grouper and tallies the following events:

1. *True match.* A one-to-one matching between a ground truth and a group. Rate measured as #matches/N.
2. *False negative.* An unmatched ground truth. Rate measured as #false neg/N.
3. *Oversegmentation.* A ground truth that matches more than one group. Rate measured as #oversegs/N.
4. *False positive.* An unmatched group. Rate measured as #false pos/G.
5. *Overgrouping.* A group that matches more than one ground truth. Rate measured as #overgroups/G.

In analyzing the results, it should be said that the Highway 55 sequence is a difficult one because of an unusually poor camera position (Figure 23(a)) and a number of large trucks that sometimes completely occlude automobiles. In terms of trading off the different error conditions, we have noticed that oversegmentation and overgrouping can be traded off by adjusting the grouping thresholds. One slight warning about these results is that we have not used a uniform set of grouping thresholds for all sequences. We are currently investigating how to automatically chose the grouper thresholds.

As the first three sequences have long shadows, the experimental results show that the system can handle shadows – shadow sub-features tend to be unstable over time, especially in congestion.

9.2 Single site level results

This section presents the testing results for the parameters that can be measured with a single camera. Specifically, flow, velocity, lane changes, queue length, spatial headway (or density), classification, and acceleration. The analysis follows the testing methodology outlined above. The bulk of this section is devoted to the core traffic parameters: flow, density, spacing and velocity. Since these parameters are inter-related, they are addressed together. Unless otherwise noted, the use of "parameters" alone refers to the core traffic parameters. Finally, we address the secondary traffic parameters of acceleration, lane changes, classification, and queue length detection.

9.2.1 Results for parameters with automated calibration

Aggregate traffic parameters are calculated after the tracker and grouper have identified vehicles in the traffic stream. At the parameter level, our testing plan calls for comparative tests against

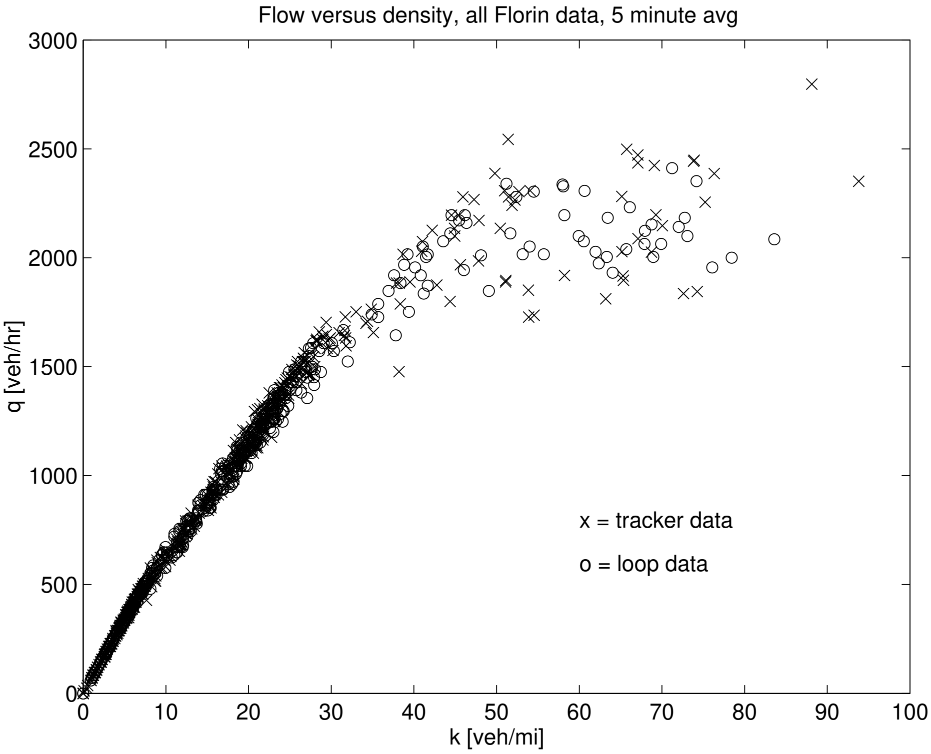


Figure 24: Scatter plot of flow and density.

existing detection methods and human verification to validate the vehicle tracker. By using existing detectors, we are able to process large quantities of data in automated tests. The trade off for automation is the fact that the tracking region is limited to the detection region of the existing loop detectors (approximately 20 ft (6.1m)). At the moment, there are no reliable area-wide vehicle detectors to test the tracker against. To test the vehicle tracker under normal operating conditions, it is necessary to manually calibrate video sequences. As a side benefit, manual tests also serve to verify the performance of the loop detectors used in the automated tests.

Our test set consists of 64 lane-hours of concurrent video and loop data. The concurrent data came from two locations, each with one camera angle. The primary site, northbound Florin Rd. & Hwy 99 in Sacramento, CA, had three lanes inbound to the CBD. Lane 1, on the left, is a carpool or HOV lane; it exhibited little if any congestion. Lane 3, on the right, exhibited some degree of congestion for approximately 20% of the samples. Finally, the loops in lane 2 were bad and it was excluded from the final analysis. Figure 24 shows the observed traffic states on the flow density plane. The secondary site on Hwy 24 in Walnut Creek, CA, had five lanes with ideal camera placement for lanes 3-5. Unfortunately, the site did not exhibit any congestion during the study period (see Appendix E for more information on the test sites).

To complement the automated tests, over two lane-hours of video were manually calibrated. Vehicle trajectories were extracted with one second resolution. After correlating the data with the tracker output, ground truth parameter measurement proceeded according to Edie's definitions. The data came from three cameras and five camera angles: Mack Rd northbound, Florin Rd two northbound views and one southbound view, and I-680 downstream. Approximately half of the manual data was collected in the field at Florin Rd. from real time video using post calibration.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.86	0.18	0.19	0.19
5%	0.95	0.31	0.33	0.34
10%	1.00	0.60	0.59	0.60
15%	1.00	0.79	0.79	0.81
20%	1.00	0.91	0.90	0.89
25%	1.00	0.96	0.96	0.94

Table 5: Error distribution for the aggregate traffic parameters for 44 lane-hours of data from Florin Road. The total for both lanes together is shown, comprising 514 5-minute samples.

Aggregate traffic parameters Traditionally, traffic engineers have defined flow as the number of vehicles passing a point in space over a fixed interval of time. This approach is appropriate for existing loop detectors that have a negligible spatial detection region. The Roadwatch vehicle detector, however, has an effective tracking region in excess of 200 ft (60 m) and is a true area wide detector.

Following the method first proposed by Edie [4], we extend the definition of flow, density (vehicles per unit distance) and average velocity to utilize the increased detection region and extract all of the information available in the vehicle trajectories. To summarized Edie's method over a given region A of the time space plane (i.e., within a fixed distance along a single lane, observed for a fixed interval of time), flow (q), density (k) and velocity (v) are defined as follows:

$$q(A) = d(A)/A, \quad k(A) = t(A)/A, \quad v(A) = d(A)/t(A) \tag{8}$$

Where d(A) is the sum of the distance traveled by all vehicles in the region A and t(A) is the sum of the time spent by all vehicles in A. The three parameters obey the following relationship:-

$$q = kv \tag{9}$$

By applying equation 8 to each lane independently, our parameter measurements are robust to lane changes. Furthermore, the fundamental equation of traffic engineering, equation 9, is valid for each lane (See Appendix ZZZ for a more detailed discussion).

Net results at northbound Florin Rd. At the aggregate parameter level, we tested q, k, v and average spacing ($1/k$). Figures 25, 26, 27 and 28 compare 44 lane-hours of measured parameter data versus loop data. Each of these figures have 514 samples from a total of 40,000 vehicles. The error distribution for all four traffic parameters are summarized in table 5.

As one would expect from a feature based tracker, the measured velocity is very accurate. Even if the tracker over-groups or over-segments vehicles, the erroneous blobs still move at the prevailing speed. The errors in flow, density and spacing are due to missed or over counted vehicles. Often, an error of two or three vehicles in one sample can be very significant. For example, one missed vehicle in a five minute sample at q=1,000 veh/hr results in a 2% error. At the mean flow for the data, 910 veh/hr, the error per missed vehicle is slightly higher, at 2.2%.

Out of these data, we can extract all of the samples with average velocity below 40 mph as representative of congested operation. This thresholding yields 28 samples from lane 3 with 4890 vehicles. We summarize the error distribution in table 6.

The tracker performance appears to have degraded slightly from the total data set. This phenomenon may, in part, be due to the small sample size.

Individual sequence results at northbound Florin Rd. The preceding data included all operating conditions: day, night, twilight, long shadows and rain; congestion and free flow.

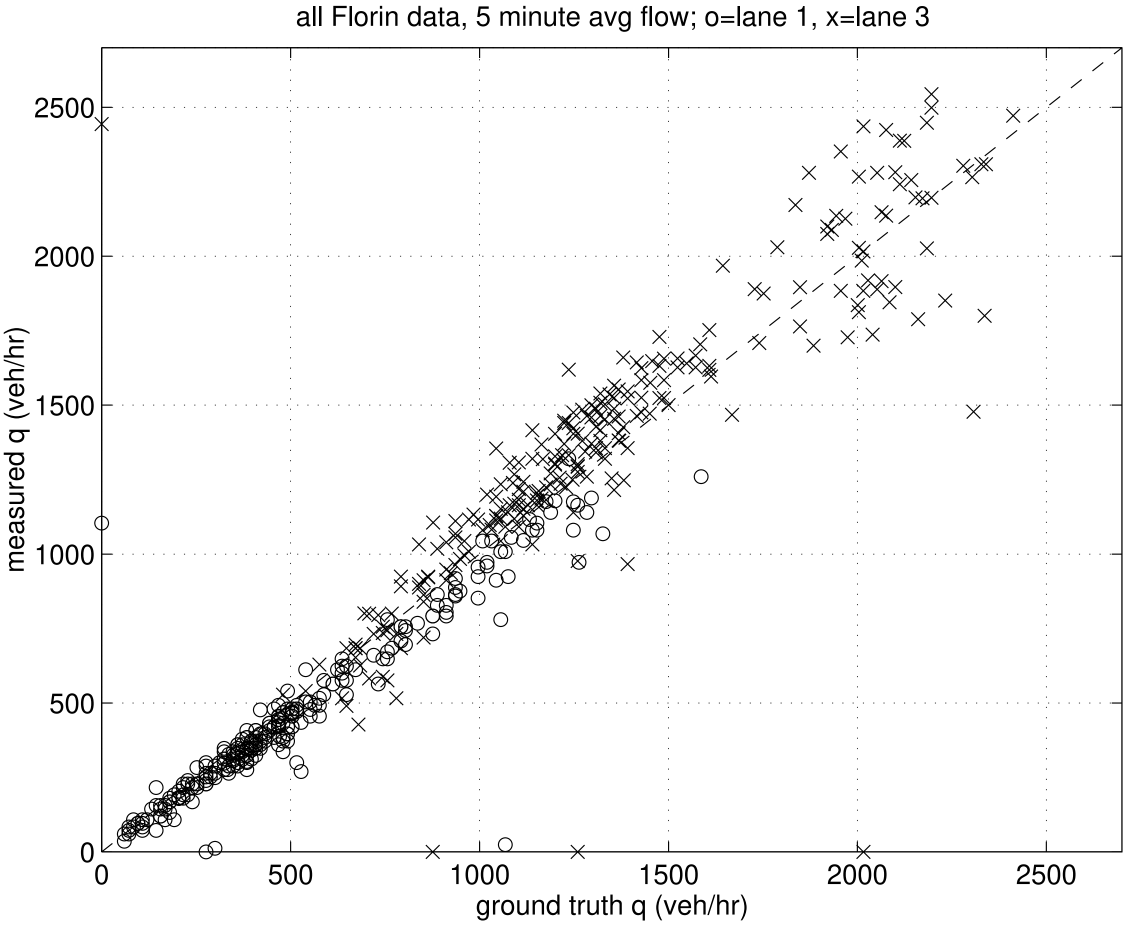


Figure 25: Scatter plot of flow.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.81	0.11	0.04	0.04
5%	0.96	0.19	0.22	0.22
10%	1.00	0.56	0.52	0.52
15%	1.00	0.78	0.74	0.74
20%	1.00	0.89	0.89	0.89
25%	1.00	0.96	0.96	0.96

Table 6: Error distribution for the aggregate traffic parameters for congestion from Florin Rd. tape #17, comprising 28 samples.

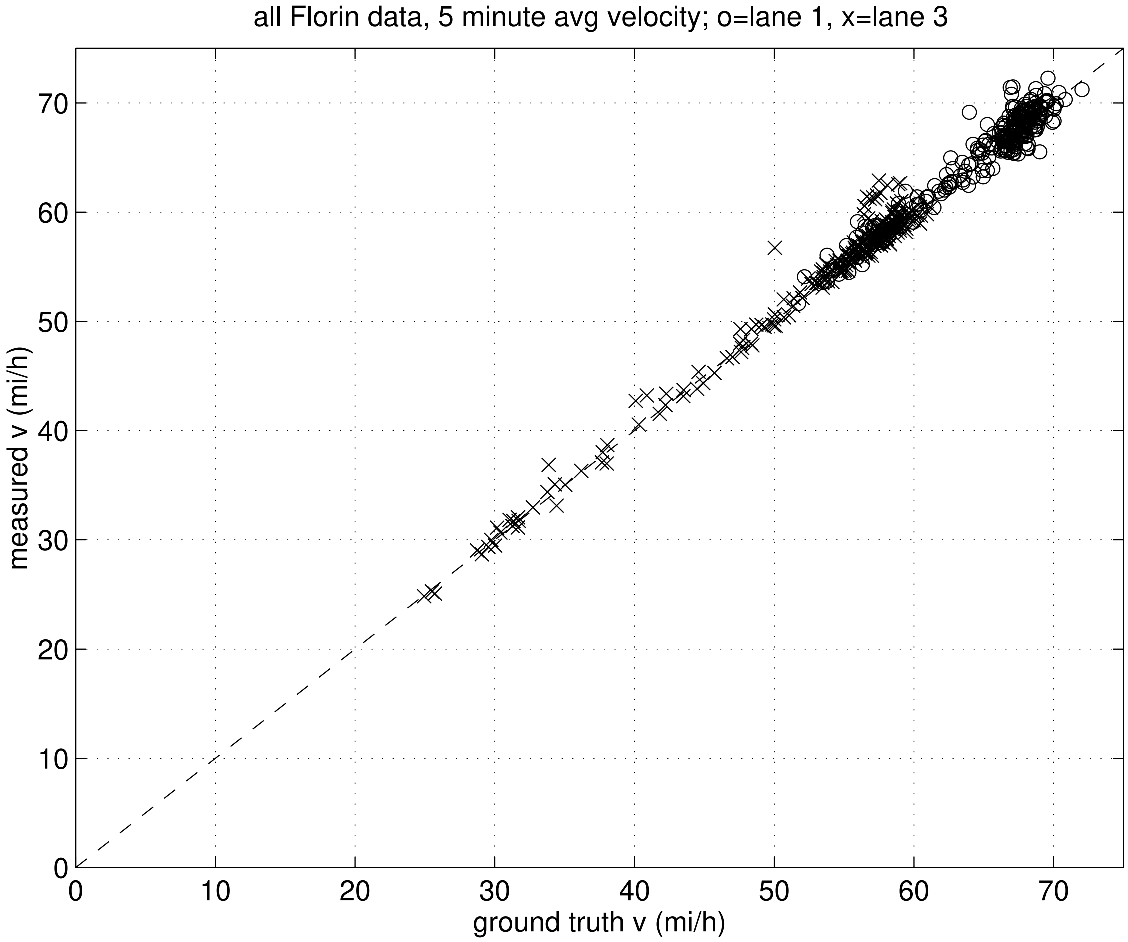


Figure 26: Scatter plot of velocity.

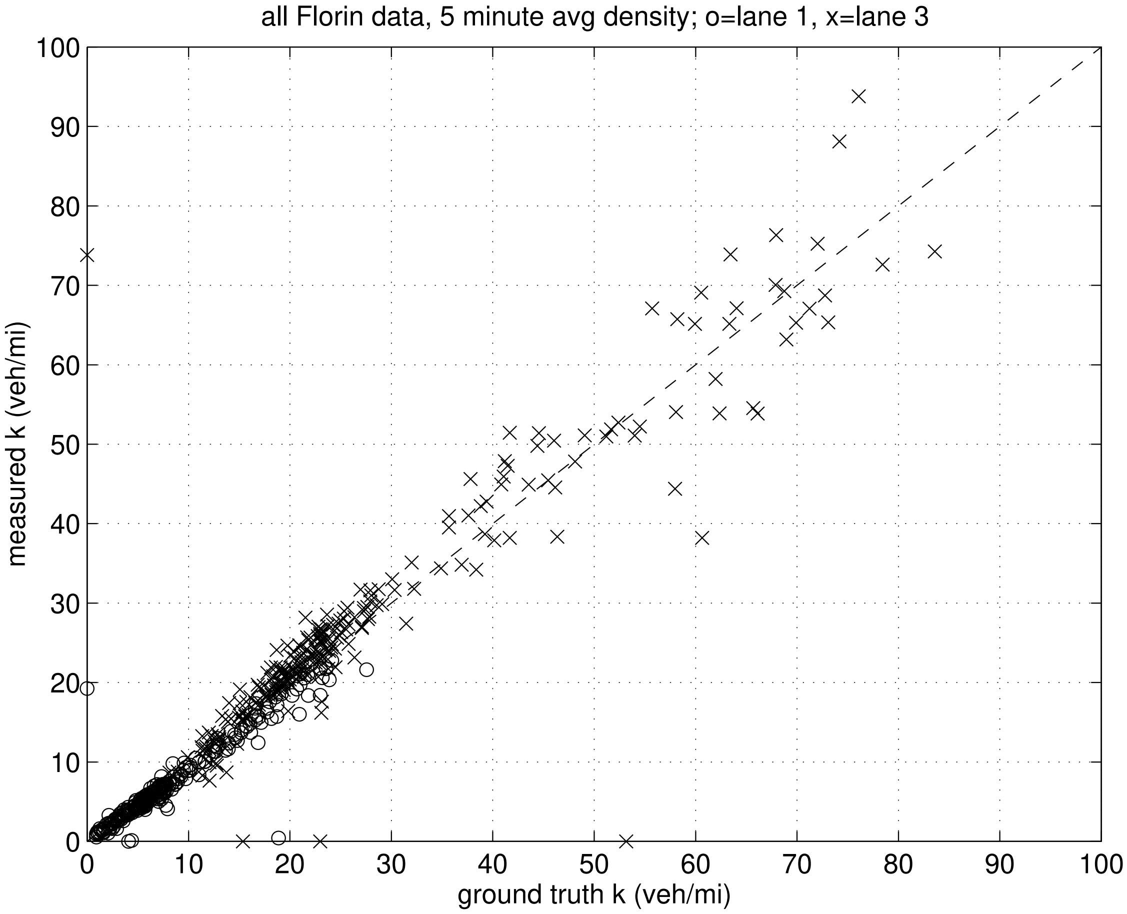


Figure 27: Scatter plot of density.

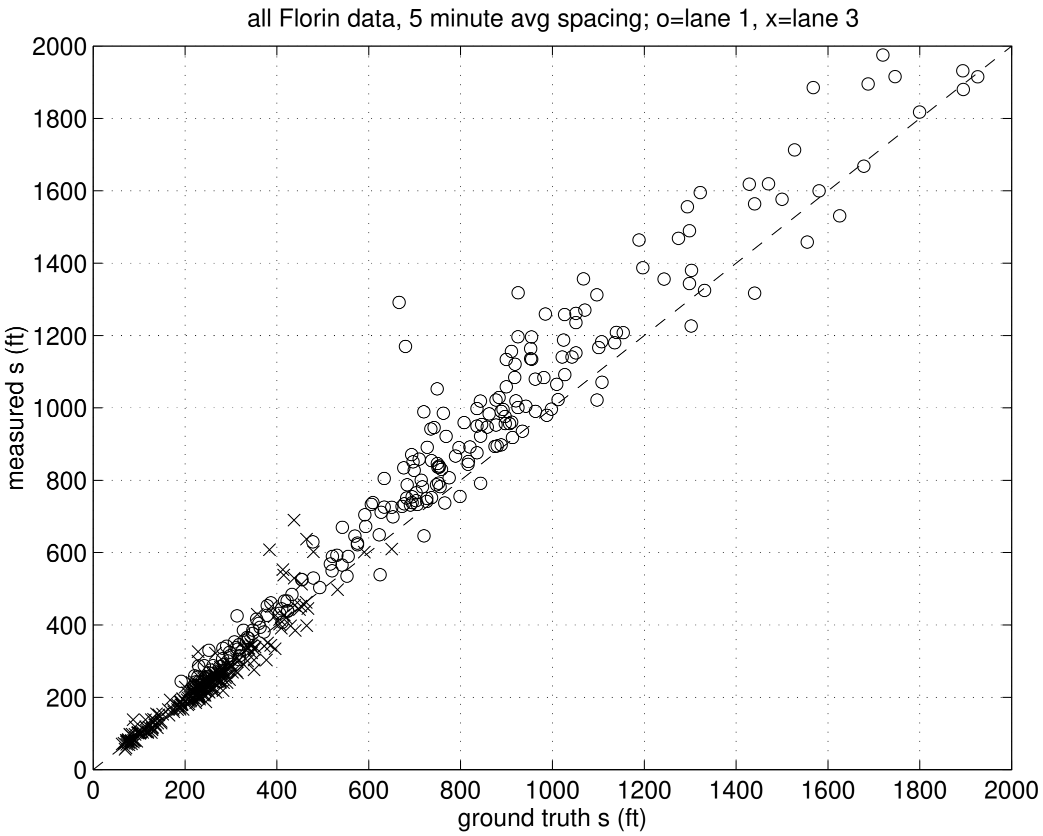


Figure 28: Scatter plot of spacing.

Figure 29: Two images from the start and end of a two hour morning run of the real time tracker (Florin Rd. tape #17).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.83	0.13	0.17	0.17
5%	0.96	0.25	0.29	0.25
10%	1.00	0.67	0.67	0.63
15%	1.00	0.88	0.83	0.83
20%	1.00	1.00	1.00	0.92
25%	1.00	1.00	1.00	1.00

Table 7: Error distribution for the aggregate traffic parameters for lane 1 of Florin Rd. tape #17 data.

It obscures the true power of the feature based tracker under difficult conditions, specifically: dramatic lighting changes, long shadows and partial occlusion. To illustrate this power we will examine two hours from the morning of 4/10/96. The sequence starts with night time conditions, progresses through sunrise and long shadows, ending with day time conditions. Figures 29a and 29b, taken from the start and end of the sequence, illustrate the dramatic change. Note that the morning peak starts during the sequence and approximately 30 minutes of data from lane three are under light congestion. Figure 30 shows the time series velocity and flow measured by the tracker and loop detectors. There are 48 samples with a total of 4,600 vehicles. Because the fundamental equation (equation 9) is preserved with our parameter measurement, we only need to present two parameters graphically, q and v, without a loss of information. The error distribution for all four traffic parameters are summarized in tables 7 and 8. We then repeat this exercise for the ten remaining sequences used in the net parameter measurement, and also for two tapes from the Highway 24 site.

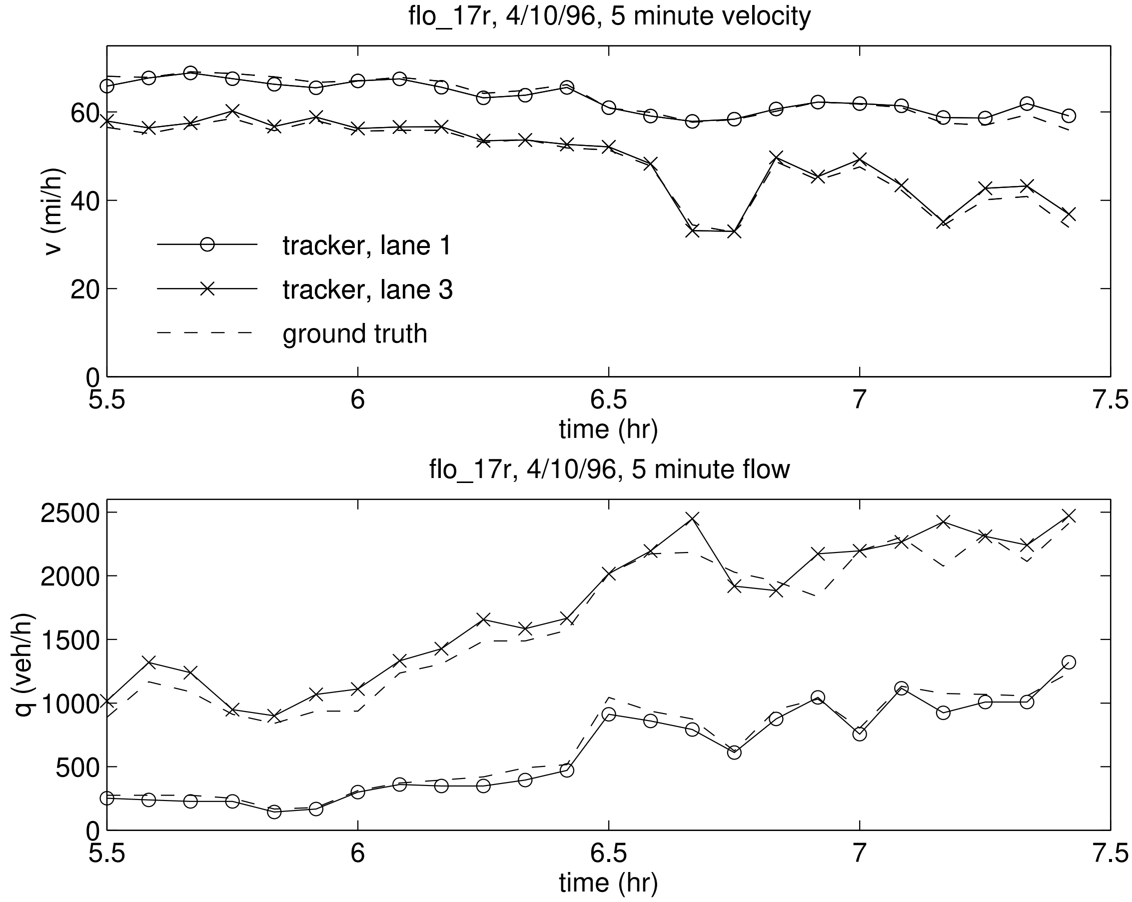


Figure 30: Florin Rd. tape #17 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.71	0.25	0.17	0.17
5%	0.88	0.33	0.29	0.29
10%	1.00	0.65	0.63	0.71
15%	1.00	0.88	0.88	1.00
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 8: Error distribution for the aggregate traffic parameters for lane 3 of Florin Rd. tape #17 data.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	1.00	0.10	0.10	0.10
5%	1.00	0.20	0.25	0.25
10%	1.00	0.50	0.50	0.40
15%	1.00	0.65	0.60	0.55
20%	1.00	0.85	0.80	0.70
25%	1.00	0.90	0.90	0.80

Table 9: Error distribution table for Florin Rd. tape #15, lane 1. Date: 4/9/96. Time: 14:30-16:10. Conditions: mid day, free flow. Total # of vehicles: 850.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.85	0.10	0.15	0.15
5%	0.95	0.20	0.30	0.30
10%	0.95	0.55	0.60	0.60
15%	1.00	0.75	0.75	0.80
20%	1.00	0.85	0.85	0.90
25%	1.00	0.90	0.90	0.95

Table 10: Error distribution table for Florin Rd. tape #15, lane 3. Total # of vehicles: 2238.

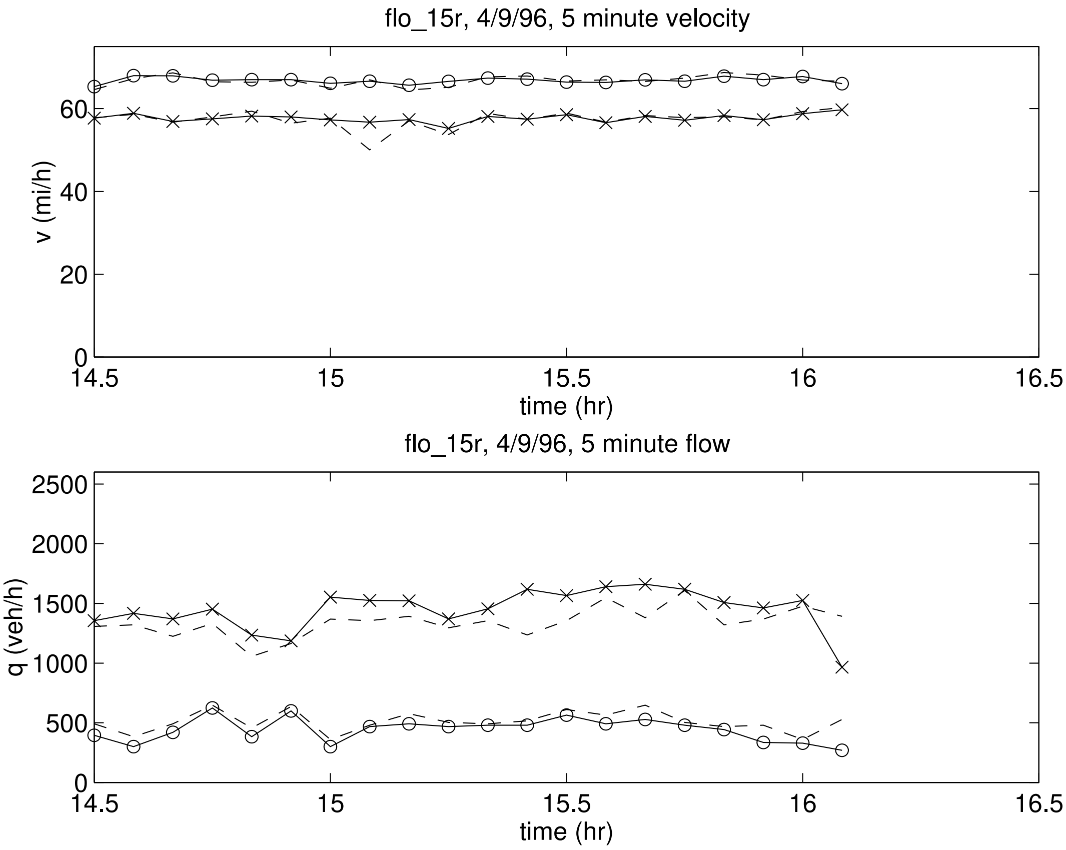


Figure 31: Florin Rd. tape #15 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.52	0.09	0.17	0.17
5%	0.83	0.30	0.22	0.22
10%	1.00	0.61	0.61	0.61
15%	1.00	0.91	0.78	0.70
20%	1.00	0.96	0.96	0.87
25%	1.00	1.00	1.00	0.96

Table 11: Error distribution table for Florin Rd. tape #18, lane 1. Date: 4/10/96. Time: 7:30-8:00, 9:00-9:30, 10:30-11:00, 12:00-12:30. Conditions: congestion in first segment, midday free flow in remaining segments. Total # of vehicles: 1072.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.83	0.26	0.22	0.22
5%	1.00	0.43	0.43	0.43
10%	1.00	0.70	0.65	0.65
15%	1.00	0.87	0.91	0.87
20%	1.00	1.00	1.00	0.91
25%	1.00	1.00	1.00	1.00

Table 12: Error distribution table for Florin Rd. tape #18, lane 3. Total # of vehicles: 2681.

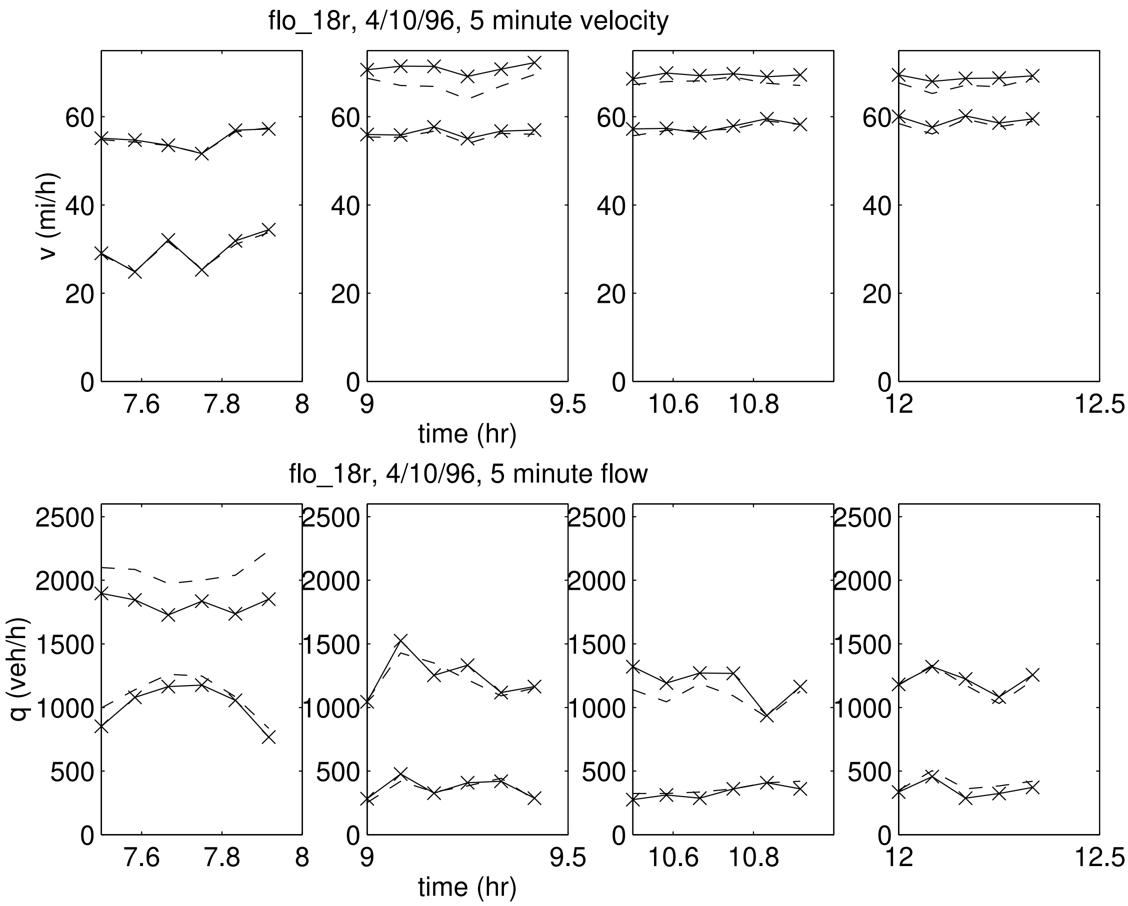


Figure 32: Florin Rd. tape #18 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.91	0.09	0.13	0.13
5%	1.00	0.26	0.22	0.22
10%	1.00	0.61	0.61	0.57
15%	1.00	0.87	0.83	0.78
20%	1.00	0.96	0.96	0.91
25%	1.00	0.96	0.96	1.00

Table 13: Error distribution table for Florin Rd. tape #19, lane 1. Date: 4/11/96. Time: 5:30-7:30. Conditions: night to day transition, ends at morning peak. Total # of vehicles: 1206.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	1.00	0.13	0.13	0.13
5%	1.00	0.26	0.22	0.26
10%	1.00	0.61	0.52	0.61
15%	1.00	0.91	0.91	0.96
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 14: Error distribution table for Florin Rd. tape #19, lane 3. Total # of vehicles: 3080.

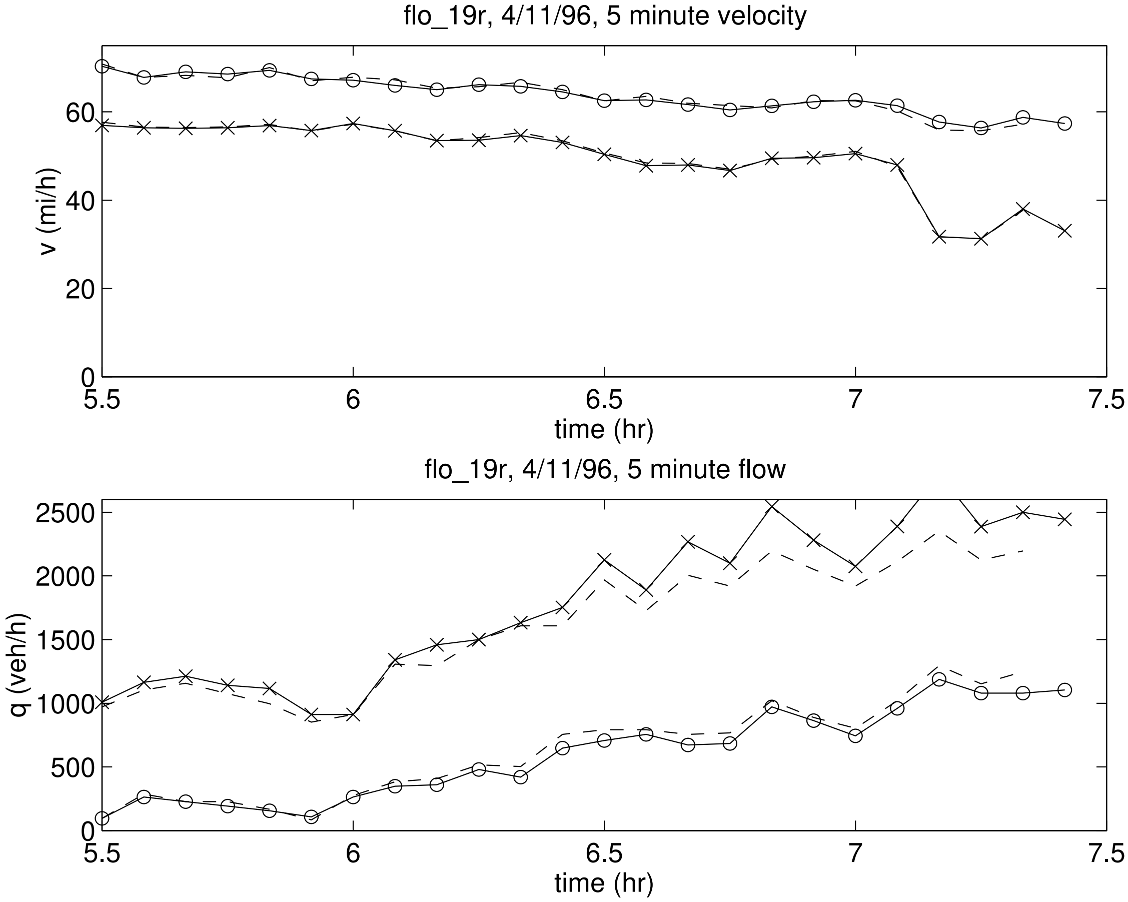


Figure 33: Florin Rd. tape #19 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.87	0.22	0.13	0.09
5%	1.00	0.35	0.35	0.35
10%	1.00	0.61	0.57	0.57
15%	1.00	0.74	0.65	0.70
20%	1.00	0.87	0.83	0.78
25%	1.00	0.96	0.91	0.83

Table 15: Error distribution table for Florin Rd. tape #20, lane 1. Date: 4/11/96. Time: 7:30-8:00, 8:30-9:00, 10:00-10:30, 11:30-12:00. Conditions: congestion in first segment, midday free flow in remaining segments. Total # of vehicles: 1210.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.91	0.30	0.30	0.30
5%	1.00	0.48	0.48	0.48
10%	1.00	0.87	0.83	0.83
15%	1.00	1.00	1.00	1.00
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 16: Error distribution table for Florin Rd. tape #20, lane 3. Total # of vehicles: 2802.

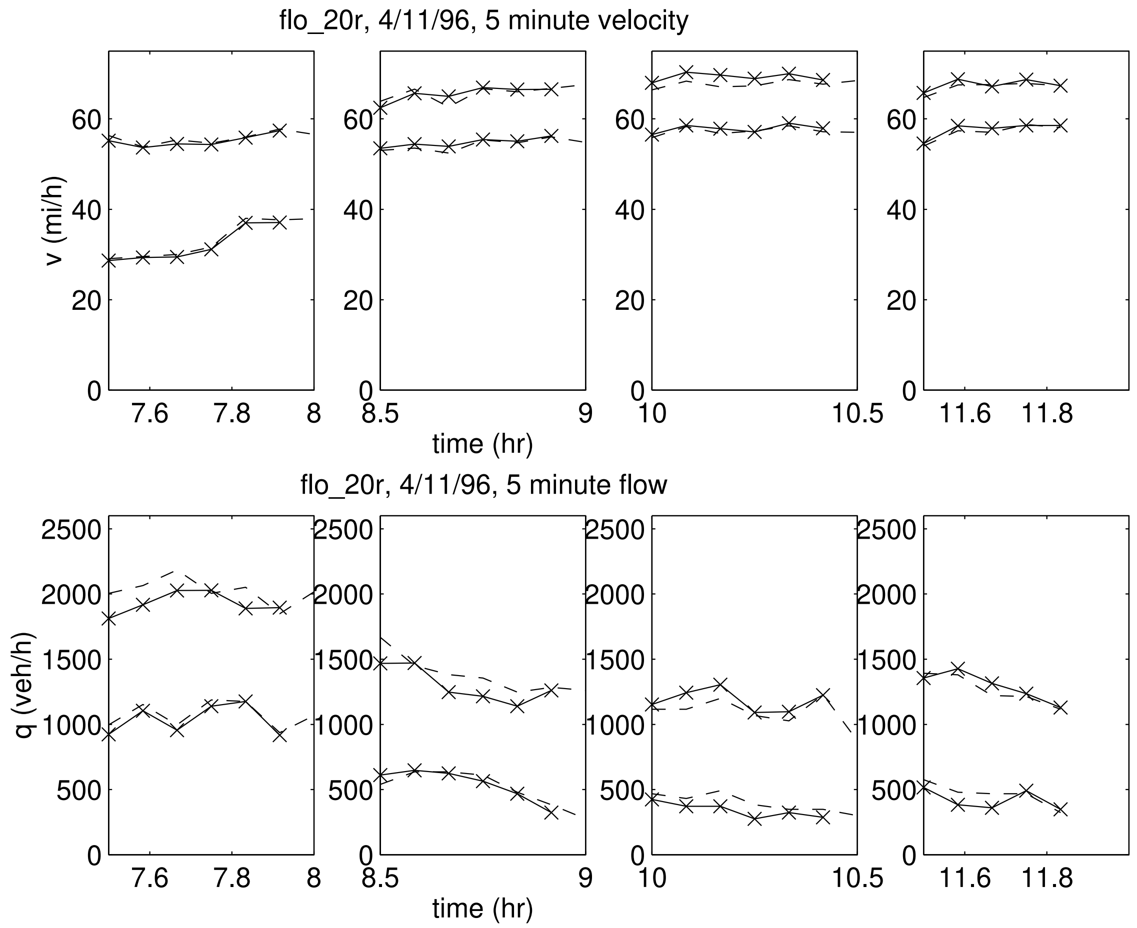


Figure 34: Florin Rd. tape #20 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	1.00	0.04	0.04	0.04
5%	1.00	0.13	0.04	0.04
10%	1.00	0.65	0.52	0.48
15%	1.00	0.83	0.83	0.83
20%	1.00	0.83	0.83	0.83
25%	1.00	0.96	0.96	0.83

Table 17: Error distribution table for Florin Rd. tape #23, lane 1. Date: 4/16/96. Time: 15:00-17:00. Conditions: mid day free flow with rain at end. Total # of vehicles: 860.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	1.00	0.26	0.26	0.26
5%	1.00	0.57	0.57	0.57
10%	1.00	0.74	0.74	0.74
15%	1.00	0.83	0.91	0.96
20%	1.00	0.96	0.96	0.96
25%	1.00	1.00	1.00	0.96

Table 18: Error distribution table for Florin Rd. tape #23, lane 3. Total # of vehicles: 2476.

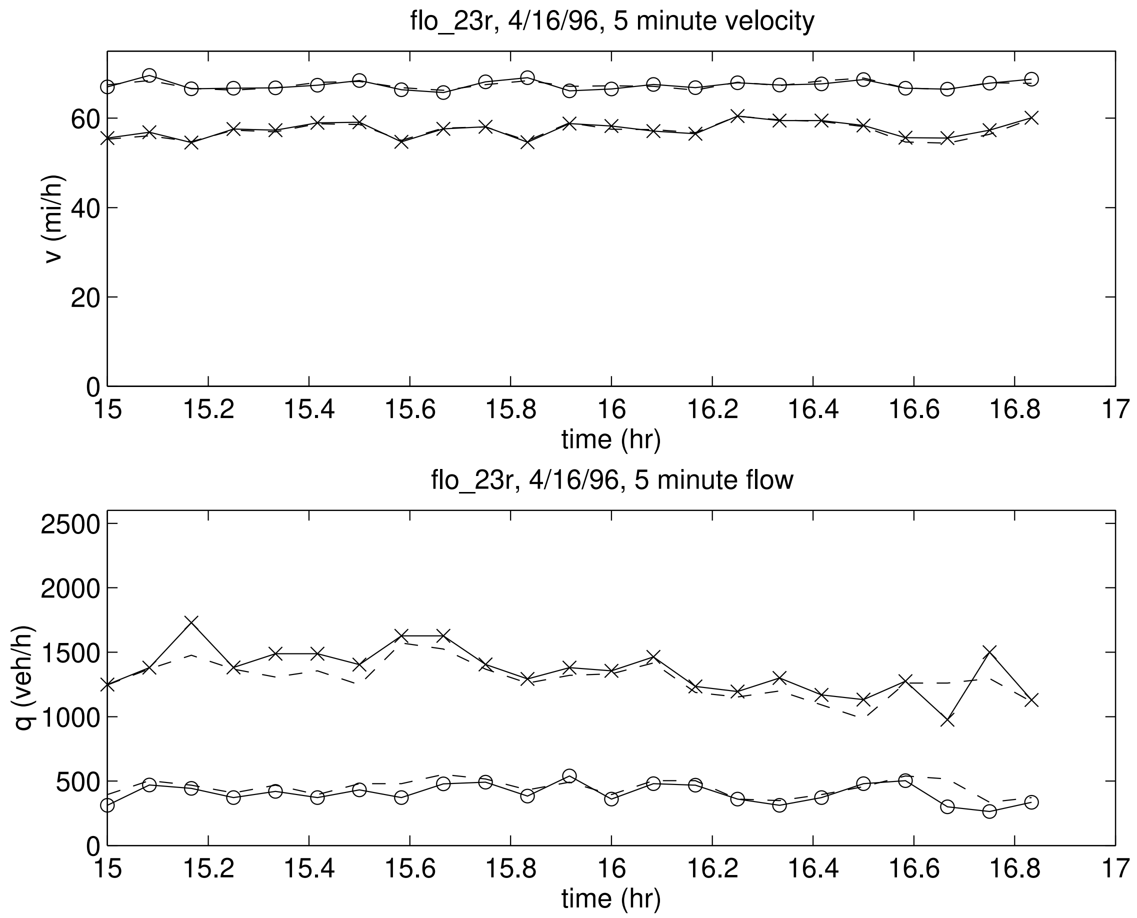


Figure 35: Florin Rd. tape #23 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.96	0.17	0.17	0.17
5%	1.00	0.17	0.22	0.22
10%	1.00	0.48	0.48	0.43
15%	1.00	0.65	0.70	0.61
20%	1.00	0.70	0.70	0.70
25%	1.00	0.78	0.78	0.70

Table 19: Error distribution table for Florin Rd. tape #24, lane 1. Date: 4/16/96. Time: 19:00-21:00. Conditions: evening light to dark transition, free flow, a little rain. Total # of vehicles: 447.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.83	0.13	0.13	0.13
5%	1.00	0.22	0.26	0.26
10%	1.00	0.43	0.39	0.39
15%	1.00	0.57	0.52	0.61
20%	1.00	0.70	0.70	0.70
25%	1.00	0.87	0.78	0.78

Table 20: Error distribution table for Florin Rd. tape #24, lane 3. Total # of vehicles: 1458.

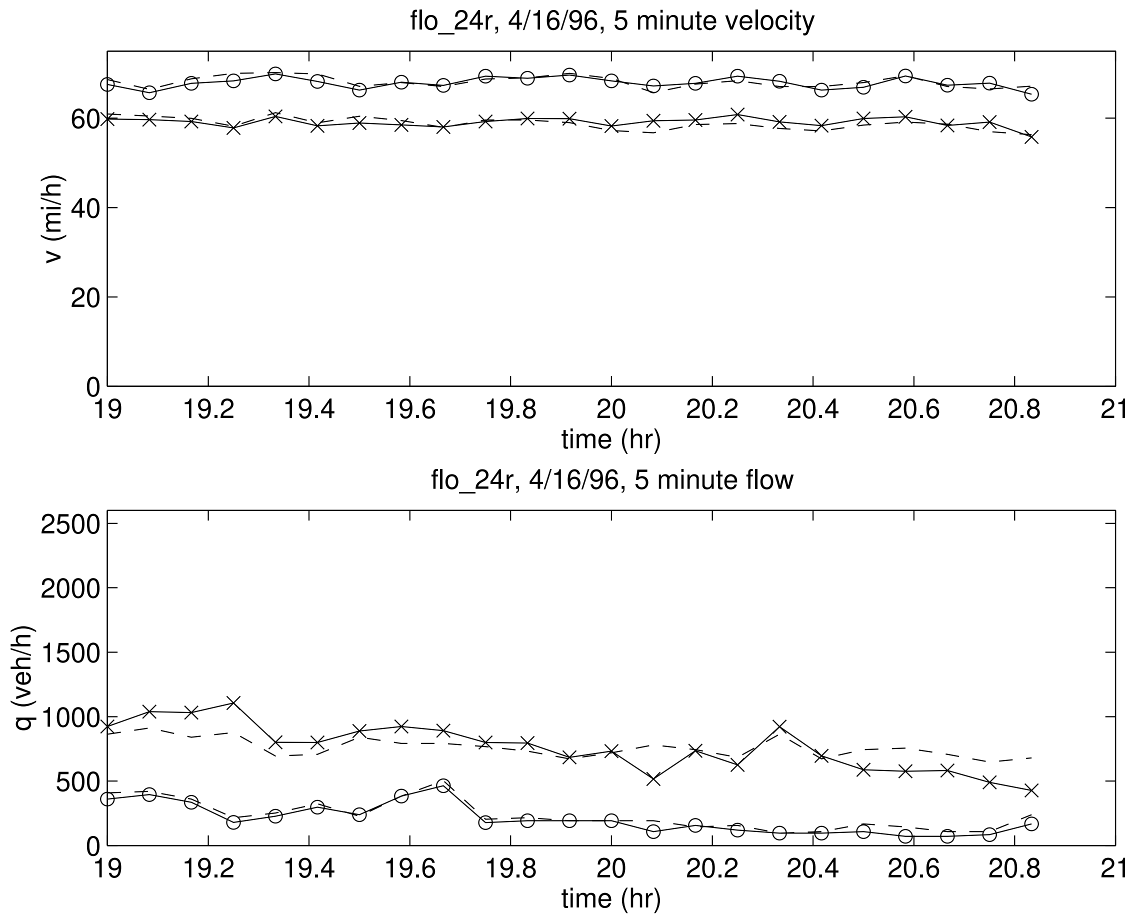


Figure 36: Florin Rd. tape #24 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.91	0.13	0.09	0.09
5%	1.00	0.17	0.22	0.17
10%	1.00	0.39	0.35	0.26
15%	1.00	0.70	0.65	0.52
20%	1.00	0.83	0.83	0.74
25%	1.00	0.91	0.91	0.83

Table 21: Error distribution table for Florin Rd. tape #25, lane 1. Date: 4/17/96. Time: 5:30-7:30. Conditions: night to day transition, ends at morning peak. Total # of vehicles: 1210.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	1.00	0.39	0.39	0.39
5%	1.00	0.52	0.48	0.52
10%	1.00	0.74	0.70	0.74
15%	1.00	0.83	0.83	0.87
20%	1.00	0.91	0.91	0.87
25%	1.00	0.96	0.96	0.91

Table 22: Error distribution table for Florin Rd. tape #25, lane 3. Total # of vehicles: 3065.

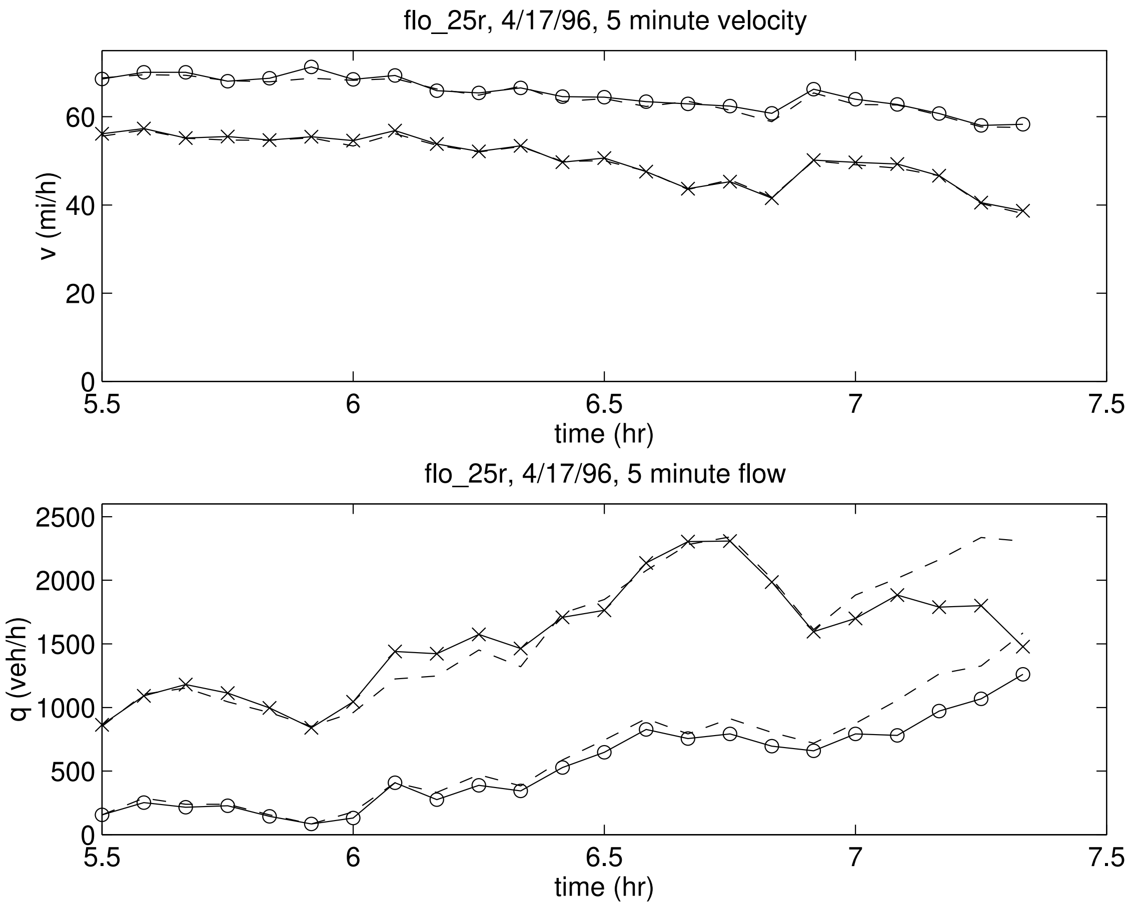


Figure 37: Florin Rd. tape #25 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.78	0.17	0.17	0.17
5%	1.00	0.30	0.26	0.26
10%	1.00	0.57	0.57	0.48
15%	1.00	0.74	0.70	0.61
20%	1.00	0.96	0.96	0.83
25%	1.00	1.00	1.00	0.96

Table 23: Error distribution table for Florin Rd. tape #27, lane 1. Date: 4/18/96. Time: 5:30-7:30. Conditions: night to day transition, ends at morning peak. Total # of vehicles: 1193.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.91	0.04	0.09	0.09
5%	1.00	0.17	0.26	0.26
10%	1.00	0.61	0.61	0.65
15%	1.00	0.74	0.78	0.78
20%	1.00	0.87	0.83	1.00
25%	1.00	1.00	1.00	1.00

Table 24: Error distribution table for Florin Rd. tape #27, lane 3. Date: 4/18/96. Time: 5:30-7:30. Conditions: night to day transition, ends at morning peak. Total # of vehicles: 2988.

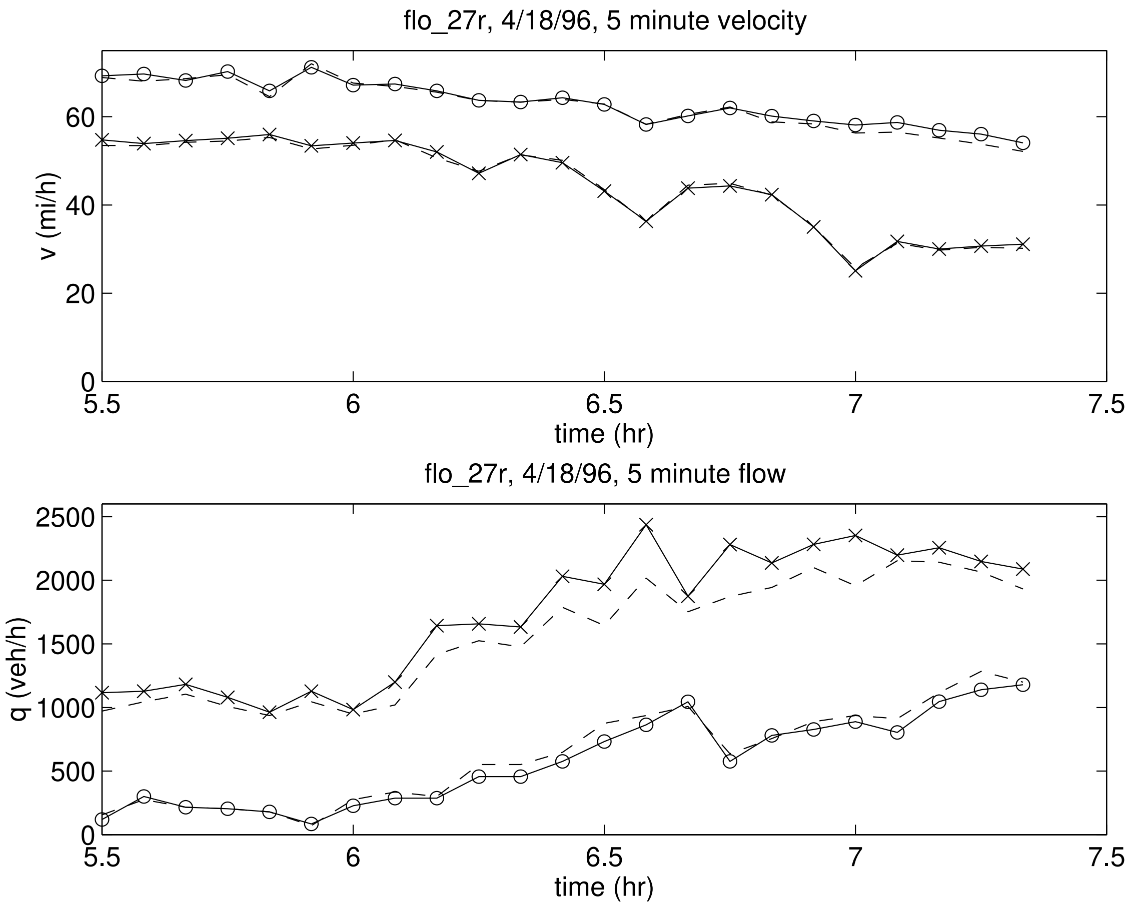


Figure 38: Florin Rd. tape #27 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	1.00	0.09	0.09	0.09
5%	1.00	0.26	0.22	0.22
10%	1.00	0.65	0.52	0.48
15%	1.00	0.70	0.70	0.65
20%	1.00	0.83	0.83	0.74
25%	1.00	1.00	1.00	0.83

Table 25: Error distribution table for Florin Rd. tape #28, lane 1. Date: 4/18/96. Time: 15:00-17:00. Conditions: midday free flow. Total # of vehicles: 903.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	1.00	0.04	0.00	0.00
5%	1.00	0.09	0.04	0.13
10%	1.00	0.17	0.17	0.26
15%	1.00	0.57	0.48	0.74
20%	1.00	0.87	0.87	0.96
25%	1.00	0.96	0.96	1.00

Table 26: Error distribution table for Florin Rd. tape #28, lane 3. Date: 4/18/96. Time: 15:00-17:00. Conditions: midday free flow. Total # of vehicles: 2478.

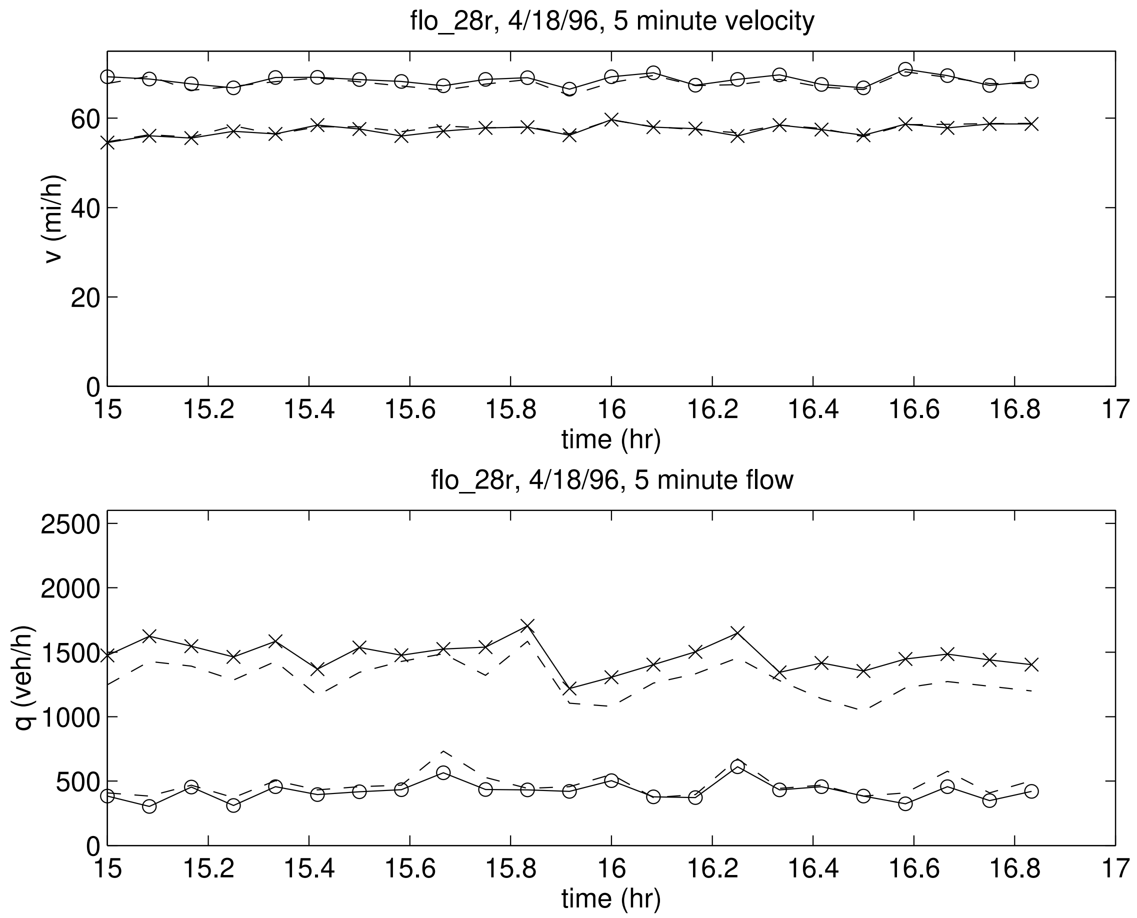


Figure 39: Florin Rd. tape #28 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.56	0.20	0.04	0.04
5%	0.96	0.32	0.24	0.24
10%	1.00	0.52	0.48	0.48
15%	1.00	0.84	0.80	0.64
20%	1.00	0.92	0.92	0.88
25%	1.00	0.92	0.92	0.92

Table 27: Error distribution table for Florin Rd. tape #29, lane 1. Date: 4/18/96. Time: 13:00-14:00, 21:00-21:30, 22:00-22:30. Conditions: midday free flow first segment, night free flow remaining segments. Total # of vehicles: 507.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.48	0.20	0.48	0.48
5%	0.52	0.36	0.64	0.68
10%	1.00	0.72	0.84	0.84
15%	1.00	0.88	0.88	0.84
20%	1.00	1.00	0.96	0.88
25%	1.00	1.00	1.00	0.96

Table 28: Error distribution table for Florin Rd. tape #29, lane 3. Date: 4/18/96. Time: 13:00-14:00, 21:00-21:30, 22:00-22:30. Conditions: midday free flow first segment, night free flow remaining segments. Total # of vehicles: 1872.

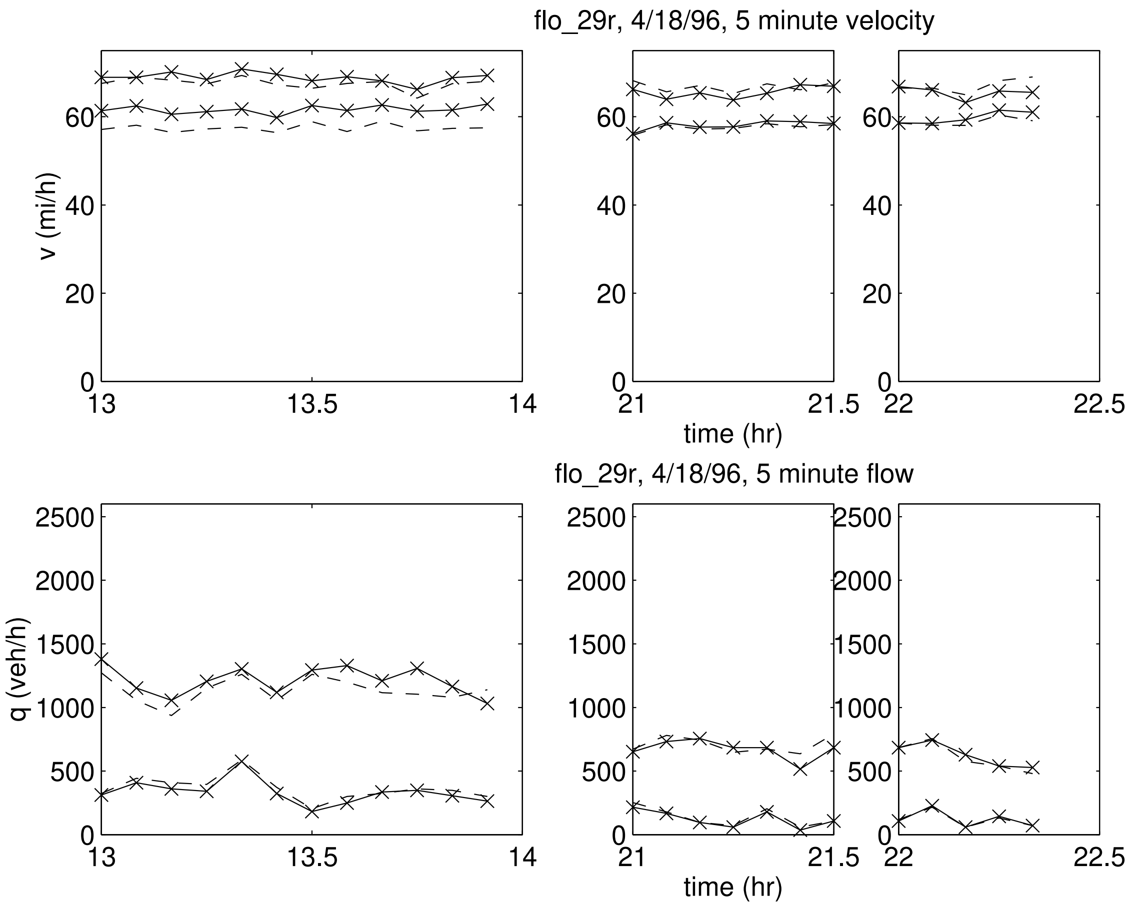


Figure 40: Florin Rd. tape #29 time series flow and velocity. Solid lines: tracker (with circles for lane 1, crosses for lane 3). Dashed lines: ground truth (loop data).

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.53	0.35	0.24	0.24
5%	0.59	0.35	0.29	0.24
10%	0.94	0.35	0.35	0.41
15%	1.00	0.35	0.41	0.41
20%	1.00	0.47	0.47	0.59
25%	1.00	0.47	0.59	0.65

Table 29: Error distribution table for H'way 24 tape #6, lane 1. Date: 4/29/96. Time: 22:00-23:45. Conditions: night, free flow, low flow. Total # of vehicles: 50.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.00	0.41	0.14	0.14
5%	0.14	0.55	0.41	0.36
10%	1.00	0.91	0.73	0.68
15%	1.00	1.00	1.00	0.86
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 30: Error distribution table for H'way 24 tape #6, lane 2. Total # of vehicles: 568.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.18	0.41	0.14	0.14
5%	0.82	0.64	0.41	0.41
10%	1.00	1.00	0.82	0.77
15%	1.00	1.00	1.00	1.00
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 31: Error distribution table for H'way 24 tape #6, lane 3. Total # of vehicles: 740.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.82	0.41	0.41	0.36
5%	1.00	0.68	0.50	0.50
10%	1.00	0.91	0.91	0.91
15%	1.00	0.95	0.95	0.95
20%	1.00	0.95	0.95	0.95
25%	1.00	0.95	0.95	0.95

Table 32: Error distribution table for H'way 24 tape #6, lane 4. Total # of vehicles: 598.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.43	0.76	0.24	0.24
5%	0.90	0.76	0.67	0.67
10%	1.00	0.81	0.76	0.76
15%	1.00	0.90	0.86	0.81
20%	1.00	0.90	0.95	0.90
25%	1.00	0.95	0.95	0.95

Table 33: Error distribution table for H'way 24 tape #6, lane 5. Total # of vehicles: 123.

9.2.2 Results for parameters with manual calibration

The automated tests against loop detectors allowed us to verify the tracker performance over an extended period. To fully test the tracker, it was necessary to go beyond the scope of the automated tests for three reasons: the automated tests were limited to the 20' detection region of the two loop speed trap, we did not observe heavy congestion at the sites equipped with loop detectors and finally to verify the field testing. In the automated tests, we selected the largest possible tracking region, in the range of 60-100 m. Vehicle trajectories were extracted by hand with one second resolution. After correlating the data with the tracker output, ground truth parameter measurement proceeded according to Edie's definitions. Several stop waves from I-680 were manually reduced as well as 75 lane minutes from the field tests.

Due to the labor intensive nature of ground truthing, the manually calibrated sequences tended to be short duration. Many either 10 or 15 minutes long while the stop waves tended to be even shorter, on the order of one minute. With the short ground truth segments, it is unreasonable to calculate 5 minute averages. We would have two or three observations per segment. Instead, we calculated the 30 second averages for a larger sample size. In doing so, we trade off sensitivity to errors in flow and density. Missing a single car when the flow is at 2000 vehicles per hour results in a 6% error for that sample. Instead of presenting the error distribution, we present the measured parameter values using the tracker and manual calculations.

Shock waves To test the limits of the vehicle tracker, several stop waves were manually calibrated. In most cases the traffic comes to a complete stop as a backwards moving disturbance passes the tracking region. Figures 41, 42, 43, 44, 45 and 46 show vehicle position along the lane as a function of time. These figures illustrate the ability of the tracker to detect individual vehicles and follow them even over a wide range of speeds. In several of the sequences, vehicle speeds go from 40 mph down to 0 and back to 40 mph in less than a minute. More importantly, vehicles are tracked even when they stop for 10 seconds or longer (represented by horizontal trajectories).

Pretaped sequences In the case of the pretaped sequences, the tracker was run for the entire sequence length, not just the calibrated segment. Tables 39-41 and figures 47-49 present 30 lane minutes of free flow data during the night to day transition at the northbound Florin Rd. test site. Figures 47, 48 and 49 plot the measured 30 second average versus the ground truth value.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.75	0.19	0.25	0.25
5%	1.00	0.38	0.50	0.50
10%	1.00	0.56	0.56	0.56
15%	1.00	0.56	0.56	0.63
20%	1.00	0.69	0.69	0.69
25%	1.00	0.69	0.75	0.88

Table 34: Error distribution table for H'way 24 tape #7, lane 1. Date: 4/30/96. Time: 15:15-16:45. Conditions: midday free flow. Total # of vehicles: 293.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.88	0.06	0.00	0.00
5%	1.00	0.06	0.06	0.06
10%	1.00	0.63	0.31	0.19
15%	1.00	1.00	1.00	0.81
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 35: Error distribution table for H'way 24 tape #7, lane 2. Total # of vehicles: 1901.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.94	0.13	0.19	0.19
5%	0.94	0.38	0.44	0.44
10%	1.00	0.69	0.69	0.69
15%	1.00	1.00	0.94	0.81
20%	1.00	1.00	1.00	0.94
25%	1.00	1.00	1.00	1.00

Table 36: Error distribution table for H'way 24 tape #7, lane 3. Total # of vehicles: 1823.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.19	0.56	0.31	0.31
5%	1.00	0.88	0.81	0.81
10%	1.00	0.94	0.94	0.94
15%	1.00	1.00	1.00	1.00
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 37: Error distribution table for H'way 24 tape #7, lane 4. Total # of vehicles: 1763.

% error less than	% vel samples	% flo samples	% dens samples	% space samples
2.5%	0.81	0.25	0.19	0.19
5%	1.00	0.56	0.44	0.44
10%	1.00	0.81	0.81	0.81
15%	1.00	0.94	1.00	1.00
20%	1.00	1.00	1.00	1.00
25%	1.00	1.00	1.00	1.00

Table 38: Error distribution table for H'way 24 tape #7, lane 5. Total # of vehicles: 695.

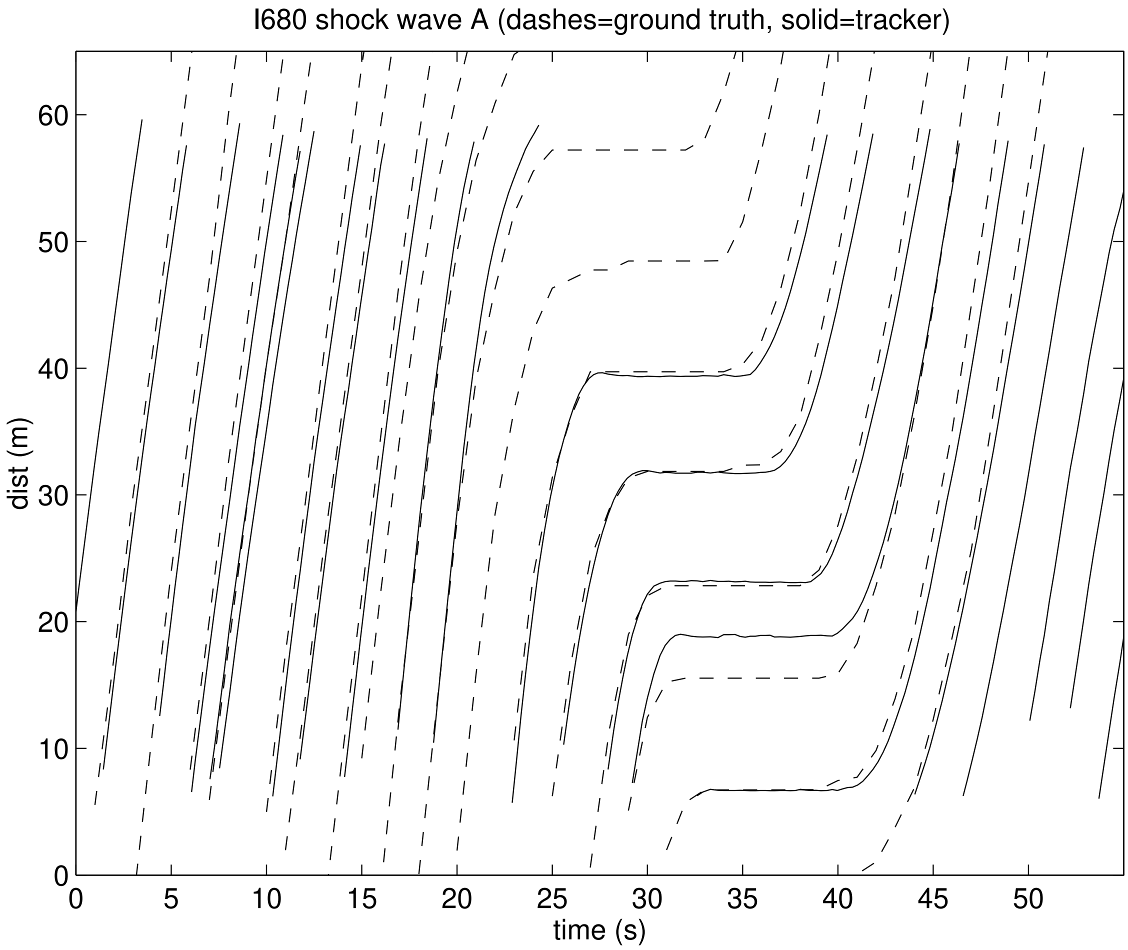


Figure 41: Shock wave example.

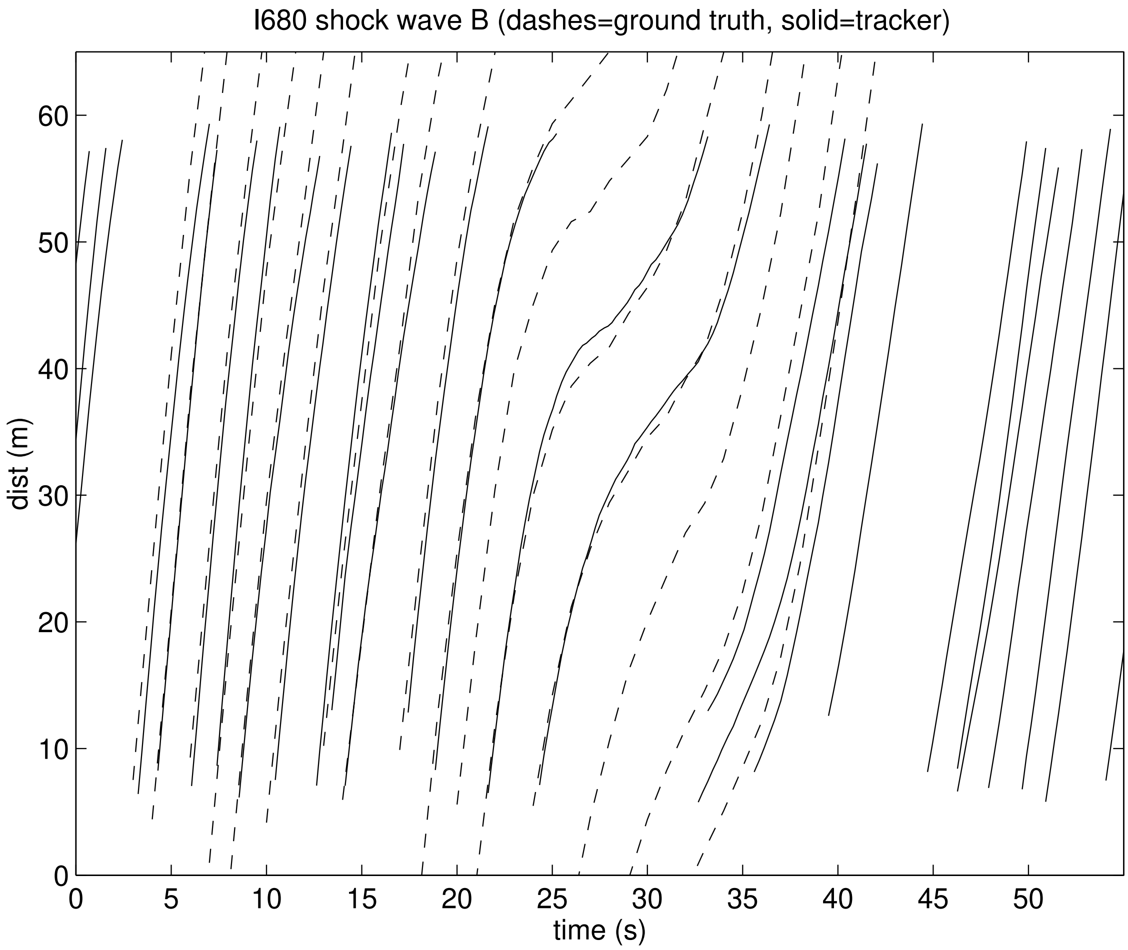


Figure 42: Shock wave example.

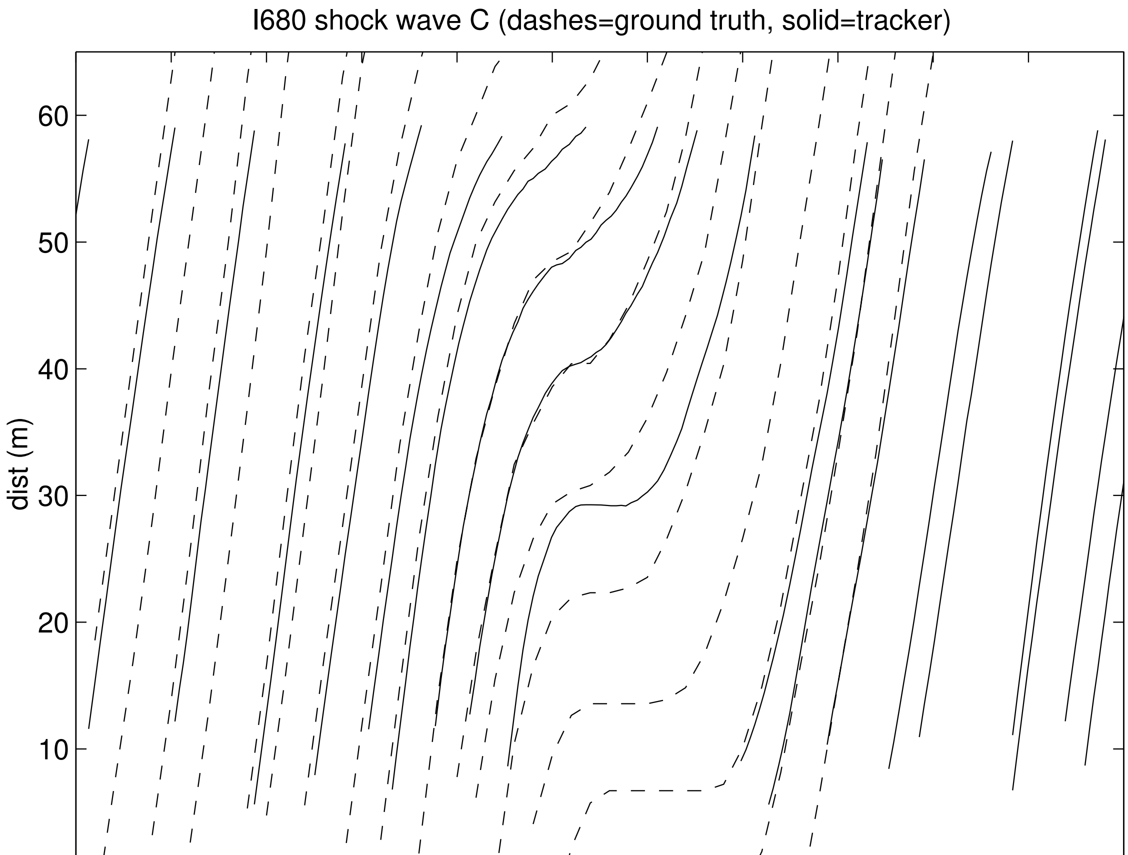


Figure 43: Shock wave example.

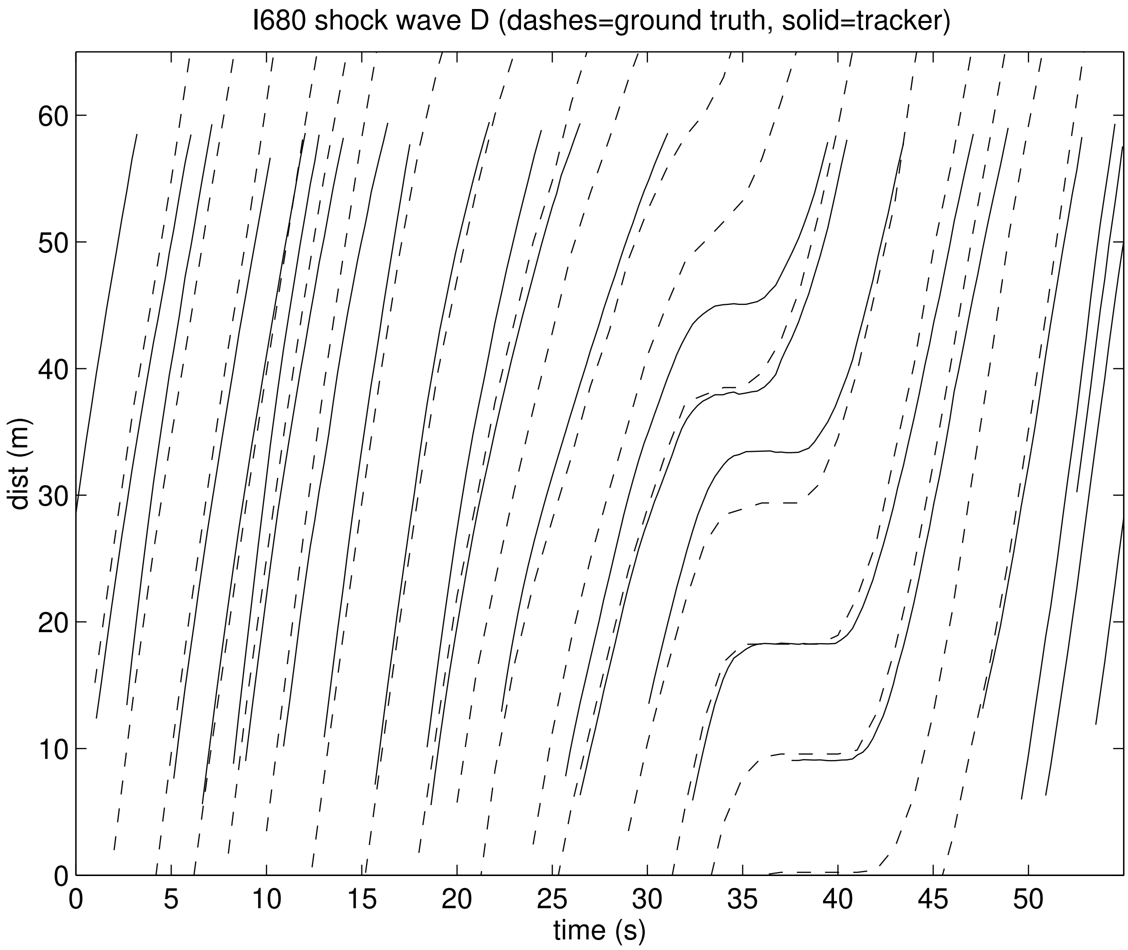


Figure 44: Shock wave example.

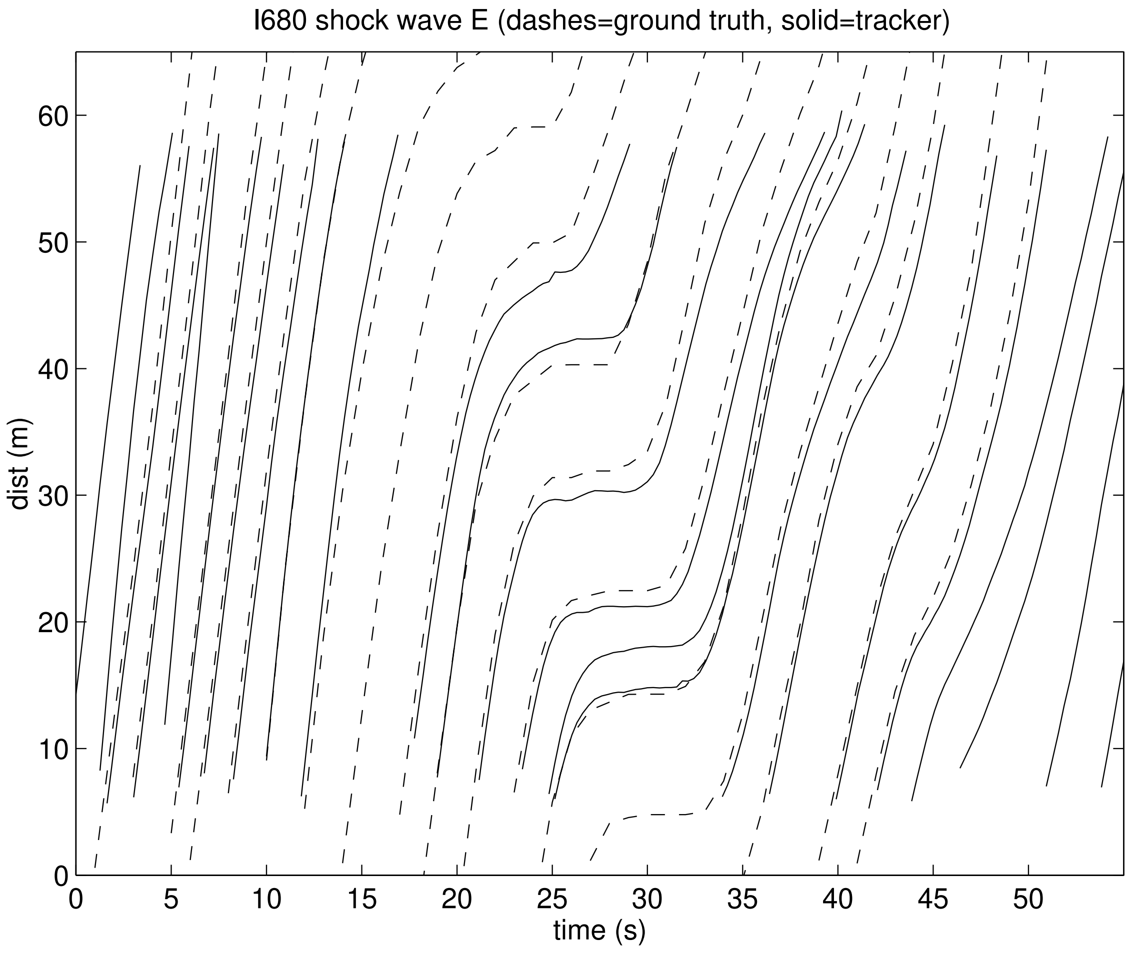


Figure 45: Shock wave example.

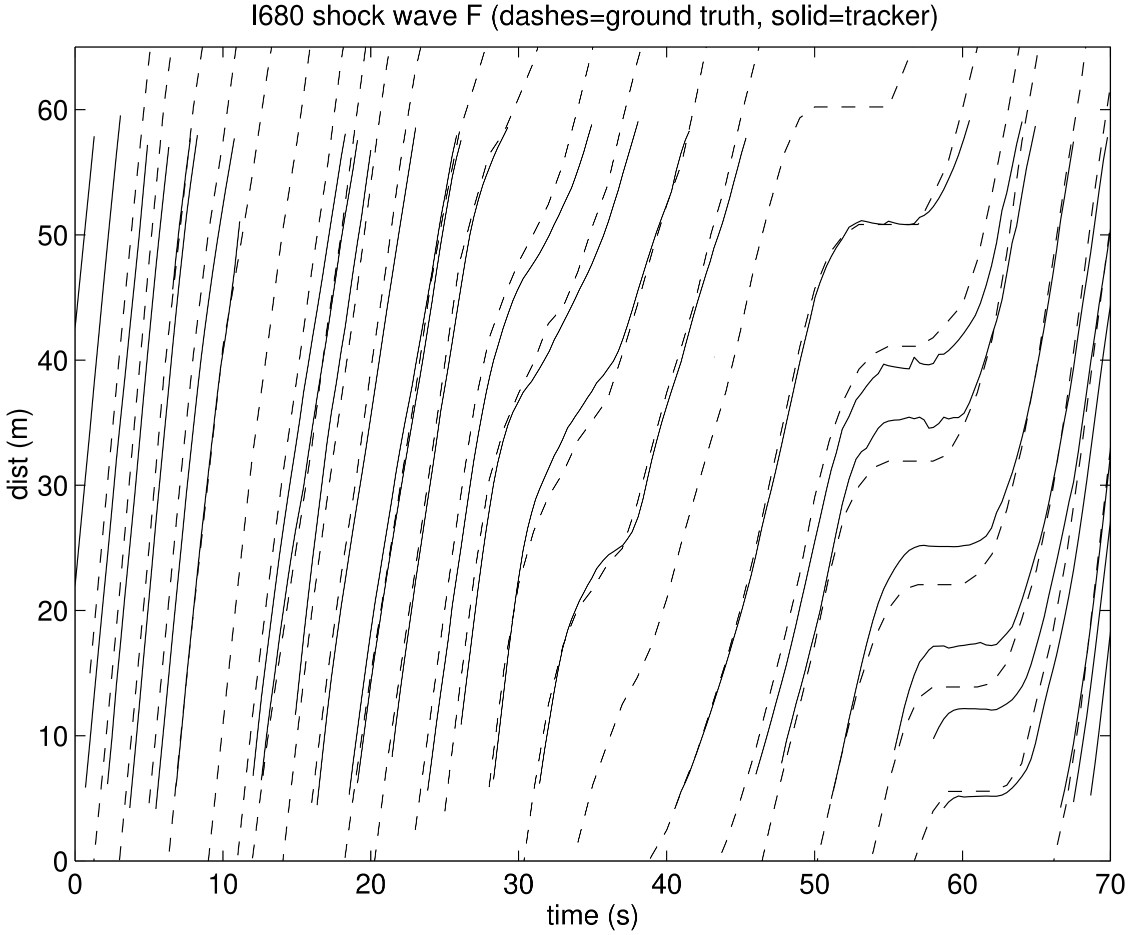


Figure 46: Shock wave example.

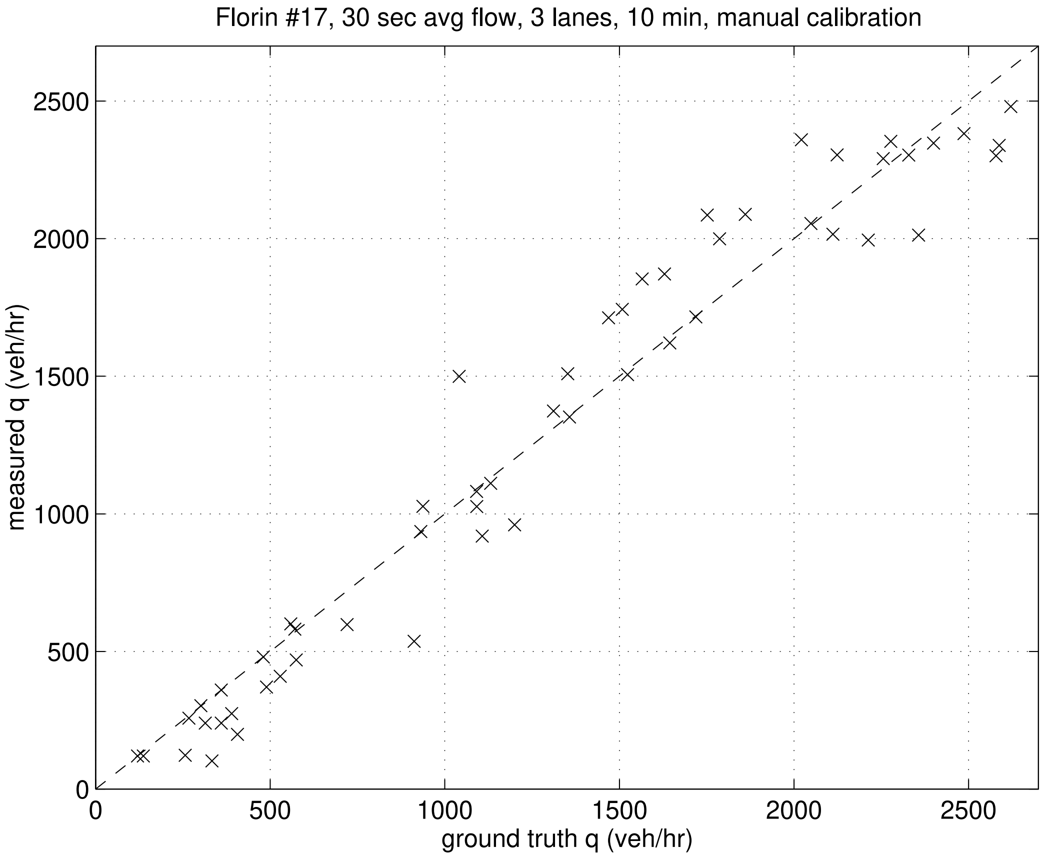


Figure 47: Scatter plot of vision and manually computed flow. Tape: Florin Rd, #17. Date: 4/10/96. Time: 6:15-6:25. Conditions: night to day transition. Manually calibrated 30 sec samples.

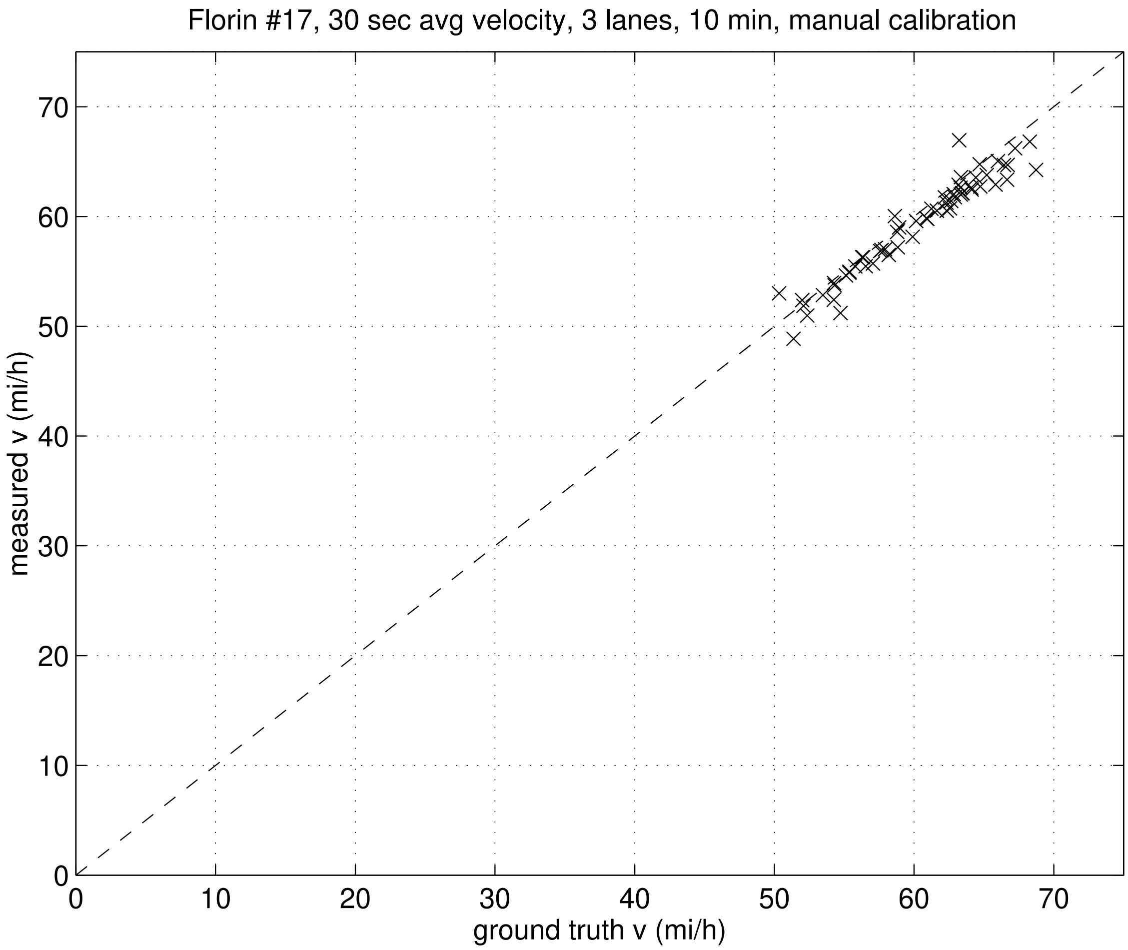


Figure 48: Scatter plot of vision and manually computed velocity. Tape: Florin Rd, #17. Date: 4/10/96. Time: 6:15-6:25. Conditions: night to day transition. Manually calibrated 30 sec samples.

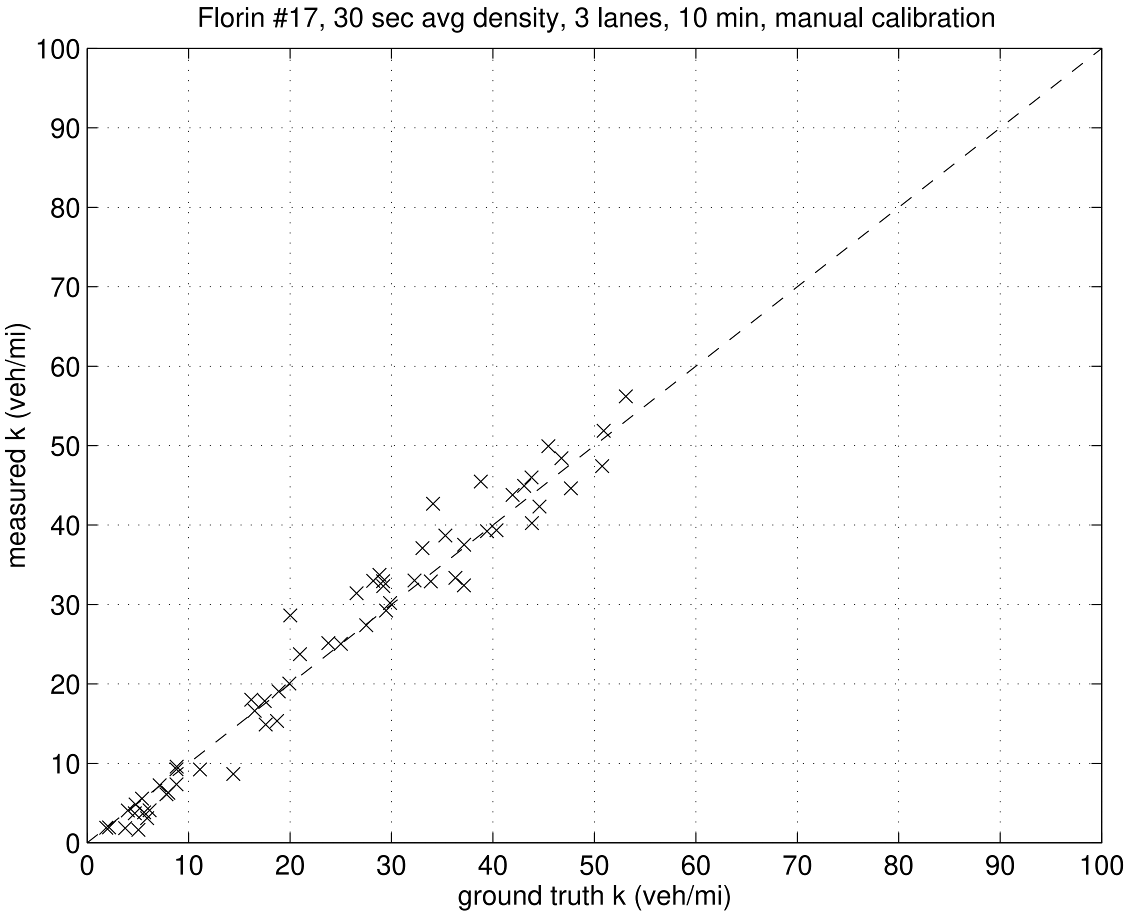


Figure 49: Scatter plot of vision and manually computed density. Tape: Florin Rd, #17. Date: 4/10/96. Time: 6:15-6:25. Conditions: night to day transition. Manually calibrated 30 sec samples.

lane 1 flow		lane 2 flow		lane 3 flow	
true	measured	true	measured	true	measured
4.6567	5.0000	22.2098	24.9844	7.7601	7.7991
2.2250	2.1492	20.7251	19.8498	12.2399	14.2708
2.7750	0.8508	19.4038	19.1999	12.5692	14.5245
4.0000	4.0000	17.5962	16.8001	13.5747	15.5946
1.0000	1.0000	18.4397	16.6238	14.3221	14.2951
7.6028	4.4728	21.8394	20.6701	18.9777	19.6141
2.5133	2.5272	19.9976	19.5593	8.6772	12.4967
3.2452	2.2870	24.1316	22.0185	16.8433	19.6602
4.7548	4.8416	21.5642	19.4920	11.3087	11.2641
2.6157	2.0000	17.0695	17.1267	9.4278	9.2591
3.3843	1.6552	11.2648	12.5742	9.2249	7.6594
6.0000	4.9795	22.7129	23.6597	12.6925	12.5503
4.0769	3.0883	19.6387	16.7737	13.0445	15.4497
2.1364	1.0218	9.0920	9.0107	10.9271	11.4453
4.7867	3.9103	14.8898	16.6628	9.0940	8.5547
1.1373	1.0000	22.6702	22.7885	13.7044	13.5087
10.0000	8.0000	21.4896	19.1780	15.5051	17.4020
3.0000	2.0000	24.4929	25.7045	18.7941	19.0893
3.0000	3.0000	24.4721	24.6114	14.5958	17.3821
4.4037	3.4137	17.6957	19.2046	7.8122	8.5626

Table 39: Flow, reported in number of vehicles per 30 sec sample: Florin Rd. tape #17, comparison of vision data with manually calibrated data.

lane 1 dens		lane 2 dens		lane 3 dens	
true	measured	true	measured	true	measured
8.8236	9.5818	45.4622	49.9369	16.5221	16.6296
4.0056	4.0710	40.3500	39.3483	29.1828	32.3162
5.0588	1.6232	37.1660	37.5122	28.2058	32.9896
7.1390	7.2501	33.8815	32.9105	28.8140	33.7341
1.8941	1.8874	36.3098	33.3632	29.8549	30.1764
14.4051	8.6566	44.5819	42.3365	41.9362	43.8078
4.7731	4.8221	39.4171	39.2247	20.0312	28.6312
6.1592	4.0996	50.7593	47.4221	38.8083	45.4826
8.8126	9.2615	47.6944	44.6207	25.0089	25.0562
4.7057	3.7089	32.2725	33.0173	18.8877	19.0529
5.9086	3.0919	20.9827	23.7412	17.5930	14.8855
11.1281	9.2277	43.8101	45.9895	27.5070	27.3962
7.8194	6.1014	37.1449	32.4055	26.5651	31.4246
3.7547	1.8350	17.5040	17.8620	23.7848	25.1379
8.8092	7.3572	29.1756	32.9377	19.9404	20.0466
2.1718	1.9343	46.7506	48.4078	29.4760	29.2384
18.7233	15.3307	43.8348	40.2411	33.0531	37.1049
5.6145	3.8433	53.0891	56.2086	43.0747	44.9425
5.4195	5.5649	50.9207	51.8663	34.0956	42.6947
8.0073	6.2987	35.3076	38.6780	16.1866	18.0199

Table 40: Density, reported in vehicles per mile. Florin Rd. tape #17, comparison of vision data with manually calibrated data.

lane 1 vel		lane 2 vel		lane 3 vel	
true	measured	true	measured	true	measured
63.3304	62.6188	58.6241	60.0384	56.3617	56.2790
66.6551	63.3507	61.6359	60.5356	50.3304	52.9920
65.8272	62.8995	62.6501	61.4195	53.4749	52.8329
67.2361	66.2062	62.3215	61.2575	56.5337	55.4734
63.3533	63.5800	60.9412	59.7922	57.5669	56.8462
63.3346	62.0025	58.7846	58.5881	54.3046	53.7277
63.1868	62.8911	60.8799	59.8378	51.9822	52.3765
63.2264	66.9425	57.0496	55.7171	52.0815	51.8709
64.7462	62.7326	54.2558	52.4205	54.2625	53.9463
66.7030	64.7086	63.4702	62.1279	59.8983	58.1504
68.7321	64.2413	64.4234	63.5563	62.9220	61.7467
64.7011	64.7550	62.2129	61.7350	55.3715	54.9727
62.5662	60.7406	63.4444	62.1143	58.9246	58.9971
68.2798	66.8233	62.3309	60.5355	55.1297	54.6361
65.2043	63.7796	61.2421	60.7067	54.7273	51.2089
62.8392	62.0390	58.1900	56.4913	55.7919	55.4423
64.0911	62.6192	58.8288	57.1895	56.2915	56.2793
64.1193	62.4459	55.3624	54.8766	52.3578	50.9699
66.4266	64.6912	57.6712	56.9419	51.3700	48.8550
65.9956	65.0363	60.1426	59.5829	57.9162	57.0210

Table 41: Velocity, reported in miles per hour. Florin Rd. tape #17, comparison of vision data with manually calibrated data.

Field testing On the morning of October 23, 1996, the vehicle tracker was installed at the Florin CCTV camera. After slightly more than an hour of calibration, the tracker was run for approximately five hours with real time video. Two views were used, similar to the northbound and southbound views described in the appendix. Two hours of midday, free flow, northbound traffic were recorded and fifteen minutes were manually calibrated from the tape. Figures 50, 51 and 52 plot the measured 30 second average versus the ground truth value. The raw data are summarized in tables 42, 43 and 44:

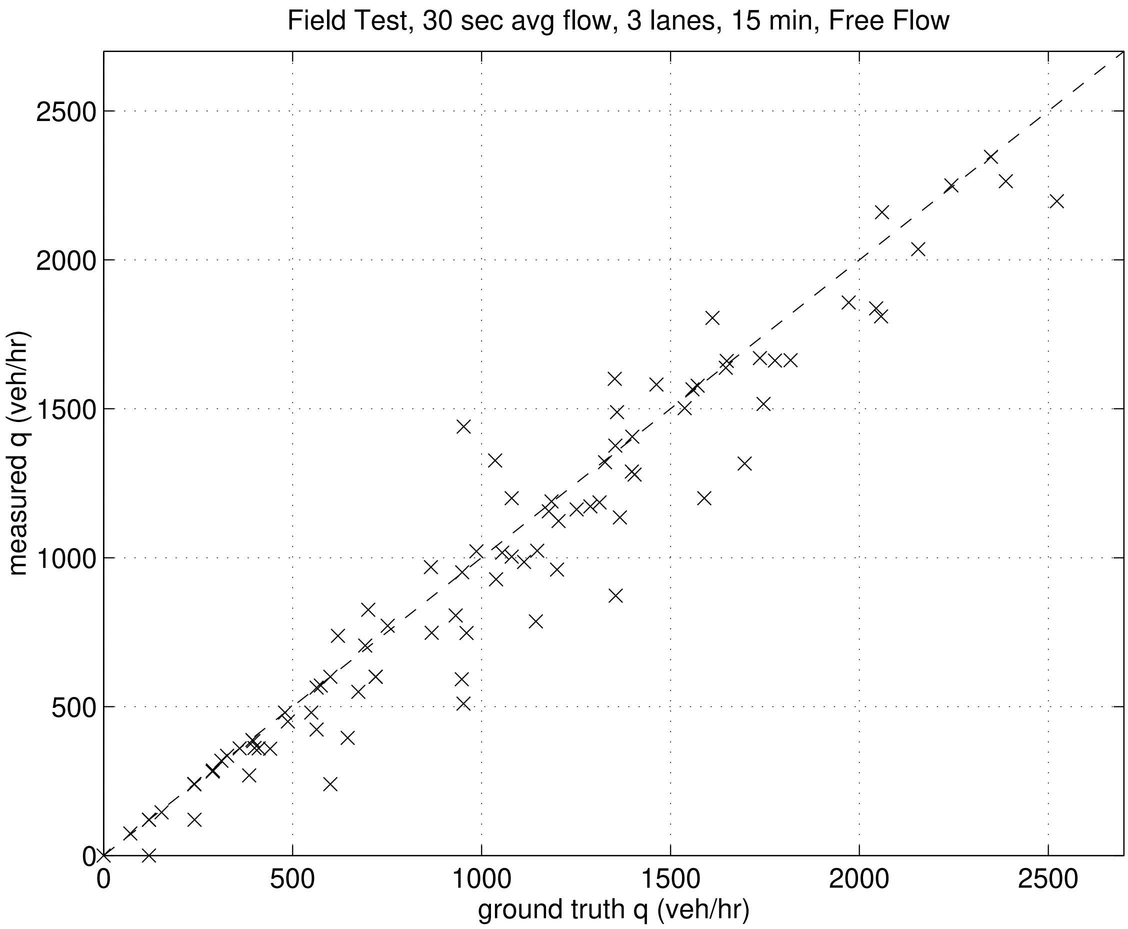


Figure 50: Scatter plot of vision and manually computed flow. Tape: Northbound Florin Rd, real time. Date: 10/23/96. Time: 13:00-13:10. Conditions: midday, free flow. Manually calibrated 30 sec samples.

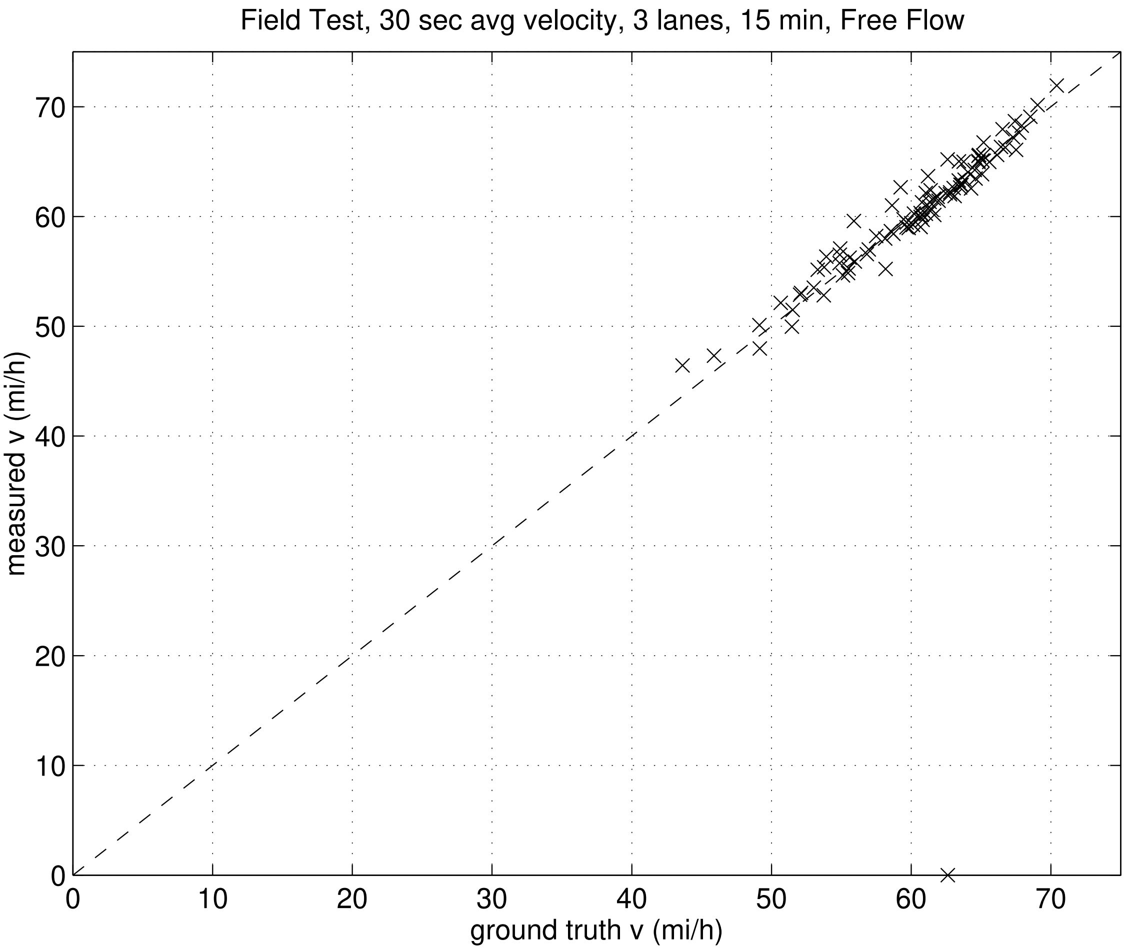


Figure 51: Scatter plot of vision and manually computed velocity. Tape: Northbound Florin Rd, real time. Date: 10/23/96. Time: 13:00-13:10. Conditions: midday, free flow. Manually calibrated 30 sec samples.

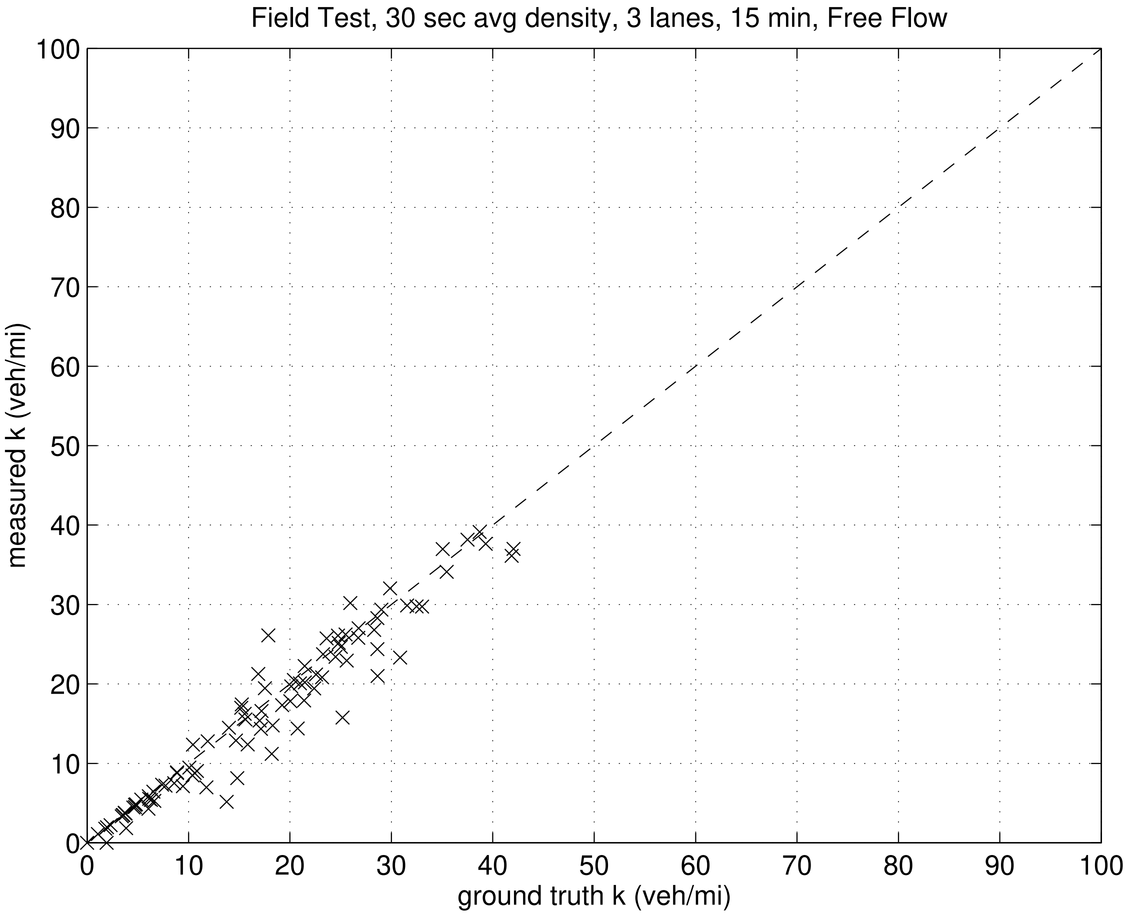


Figure 52: Scatter plot of vision and manually computed density. Tape: Northbound Florin Rd, real time. Date: 10/23/96. Time: 13:00-13:10. Conditions: midday, free flow. Manually calibrated 30 sec samples.

lane 1 flow		lane 2 flow		lane 3 flow	
true	measured	true	measured	true	measured
7.9360	4.2516	18.6935	18.7521	17.1424	15.0875
4.0640	3.7484	13.7202	13.6462	10.9353	9.8791
2.0000	2.0000	11.6479	10.7444	7.9059	7.9265
2.0000	2.0000	17.9636	16.9661	11.3218	12.4085
5.0000	5.0000	21.0225	18.3080	13.4280	15.0421
0	0	5.6156	4.5813	4.6970	4.6974
4.0000	4.0000	11.3844	9.4582	9.5621	8.5296
4.7901	4.7587	14.5539	12.6355	9.2726	8.2116
3.2099	2.2413	14.8042	13.8493	11.2911	7.2733
4.5804	4.0000	8.2219	8.5144	10.4329	9.6838
3.4196	3.0000	14.4742	13.9196	11.0571	11.0034
2.0031	1.0000	12.8108	12.5171	9.0000	8.3618
2.0000	2.0000	3.2803	3.2323	7.2349	6.2323
6.0000	5.0000	19.5697	19.5496	10.7333	9.7677
1.0000	0	15.1500	13.8606	10.0318	9.3575
3.6735	2.9881	19.8961	18.8687	13.2463	10.0000
3.3265	3.0119	17.0355	15.3087	7.9405	12.0000
2.0000	2.0000	11.6572	11.7203	9.5336	6.5503
2.0000	2.0000	8.7871	8.4756	6.2621	6.4275
2.4123	2.3831	9.8752	9.9081	5.8339	6.8746
0.5877	0.6169	7.7641	6.7186	5.1682	6.1476
1.0000	1.0000	8.0035	6.2283	9.8158	9.6297
2.4032	2.3467	17.1630	18.0000	7.8982	4.9320
2.5968	2.6533	9.0000	10.0000	13.7393	13.8404
3.0000	3.0000	12.1901	13.1788	5.3803	3.2907
2.0000	2.0000	13.0988	13.1450	11.7092	10.6601
2.7223	2.7892	14.1376	10.9690	11.2709	13.3399
1.2777	1.2108	12.9870	13.0434	5.7688	5.8779
6.0000	5.0000	10.0000	8.0000	8.6317	11.0535
2.0000	2.0000	11.2856	11.4705	4.6969	3.5251
1.0000	1.0000	8.6525	7.7304	5.0000	2.0000
3.3040	3.2015	16.4299	15.4722	7.2166	8.0694

Table 42: Flow, reported in number of vehicles per 30 sec sample. Northbound Florin Rd, comparison of vision data with manually calibrated data.

lane 1 dens		lane 2 dens		lane 3 dens	
true	measured	true	measured	true	measured
14.8122	8.1554	37.5002	38.1730	41.8628	36.1300
7.7357	7.1860	26.7799	27.0111	22.3806	19.4316
3.7643	3.6930	21.4785	20.1920	15.6111	15.5195
3.7179	3.7357	35.4473	34.1044	24.7378	26.0843
8.8592	8.8722	42.0423	36.9929	29.8817	32.0415
0	0	10.3963	8.4636	10.0821	9.4591
7.4006	7.3158	21.4028	17.9140	19.2325	17.3405
8.8706	8.7786	28.6150	24.3963	20.0414	17.8362
6.0557	4.2775	28.3002	26.7971	25.1952	15.7609
8.5804	7.4969	15.5309	16.2457	22.5686	21.1846
6.2955	5.3942	28.6289	28.2938	25.0234	24.6736
3.8394	1.8407	26.7365	25.8028	20.9867	20.0793
3.7146	3.7816	6.5391	6.4355	17.1382	14.3483
10.8374	9.0441	38.7048	39.1302	23.1679	20.8365
1.9161	0	32.4840	29.7564	24.4874	23.4061
6.6248	5.2762	39.2982	37.6351	30.8538	23.3077
6.1314	5.5512	33.0287	29.7297	17.8735	26.1116
3.5594	3.4949	23.2669	23.7411	20.7586	14.3850
3.7844	3.6915	17.1967	16.5984	13.9804	14.4949
4.7290	4.4913	20.3817	20.5056	15.2575	17.4334
1.0569	1.1169	14.6710	12.8699	10.4302	12.3478
1.7836	1.7853	15.8276	12.3916	20.1204	19.6949
4.5478	4.4500	35.0699	36.9720	18.2104	11.1912
4.8077	4.8598	17.5237	19.4718	29.0110	29.3603
5.3322	5.4493	23.6099	25.7411	11.7626	6.9873
3.5018	3.4740	25.4933	26.2294	25.6066	22.9410
4.7308	4.7711	28.6316	21.0096	25.9531	30.1873
2.3176	2.2146	24.8209	25.1650	11.9017	12.7742
10.6012	8.7889	18.2937	14.7754	16.8732	21.2563
3.4079	3.3368	21.4593	22.2490	9.4448	7.1159
1.9579	1.9716	17.0081	15.3984	13.7524	5.1697
6.0848	5.8610	31.5618	29.8881	15.2035	16.9917

Table 43: Density, reported in vehicles per mile. Northbound Florin Rd, comparison of vision data with manually calibrated data.

lane 1 vel		lane 2 vel		lane 3 vel	
true	measured	true	measured	true	measured
64.2930	62.5587	59.8190	58.9488	49.1388	50.1106
63.0421	62.5956	61.4798	60.6250	58.6325	61.0083
63.7569	64.9887	65.0765	63.8535	60.7715	61.2890
64.5520	64.2456	60.8123	59.6972	54.9207	57.0850
67.7259	67.6271	60.0039	59.3886	53.9247	56.3346
0	0	64.8180	64.9562	55.9058	59.5923
64.8598	65.6115	63.8296	63.3571	59.6621	59.0266
64.7992	65.0491	61.0333	62.1513	55.5207	55.2468
63.6075	62.8782	62.7737	62.0185	53.7773	55.3776
64.0583	64.0264	63.5266	62.8917	55.4729	54.8537
65.1825	66.7384	60.6695	59.0360	53.0244	53.5152
62.6080	65.1935	57.4982	58.2126	51.4611	49.9723
64.6103	63.4645	60.1975	60.2703	50.6583	52.1228
66.4365	66.3413	60.6737	59.9526	55.5939	56.2533
62.6282	0	55.9660	55.8963	49.1605	47.9746
66.5417	67.9594	60.7544	60.1630	51.5191	51.4851
65.1031	65.1092	61.8934	61.7913	53.3115	55.1478
67.4270	68.6710	60.1226	59.2405	55.1113	54.6423
63.4181	65.0151	61.3176	61.3203	53.7500	52.8164
61.2121	63.6739	58.1417	57.9829	45.8837	47.3202
66.7325	66.2745	63.5055	62.6450	59.4607	59.7449
67.2810	67.2160	60.6804	60.3145	58.5424	58.6734
63.4119	63.2829	58.7272	58.4226	52.0459	52.8847
64.8157	65.5152	61.6309	61.6277	56.8307	56.5680
67.5137	66.0641	61.9577	61.4368	54.8885	56.5153
68.5371	69.0842	61.6578	60.1384	54.8728	55.7612
69.0539	70.1515	59.2531	62.6515	52.1134	53.0284
66.1553	65.6100	62.7875	62.1978	58.1648	55.2160
67.9169	68.2680	65.5965	64.9730	61.3877	62.4010
70.4249	71.9254	63.1085	61.8660	59.6757	59.4451
61.2903	60.8644	61.0478	60.2431	43.6288	46.4242
65.1585	65.5492	62.4674	62.1207	56.9602	56.9887

Table 44: Velocity, reported in miles per hour. Northbound Florin Rd, comparison of vision data with manually calibrated data.

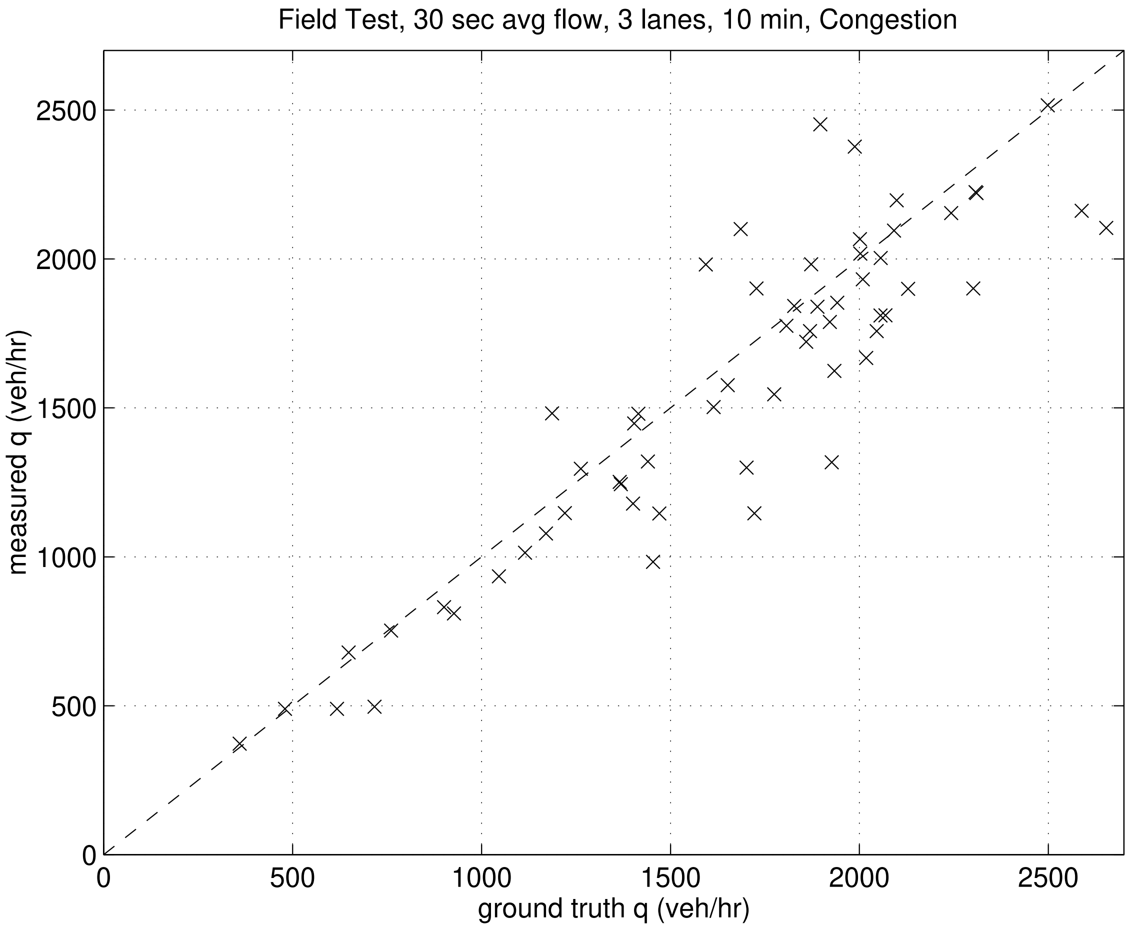


Figure 53: Scatter plot of vision and manually computed flow. Tape: Southbound Florin Rd, real time. Date: 10/23/96. Time: 4:50-5:00. Conditions: evening rush hour, long shadows. Manually calibrated 30 sec samples.

Two hours of evening peak, southbound traffic were recorded and ten minutes were manually calibrated from the tape, including some congestion. Figures 53, 54 and 55 plot the measured 30 second average versus the ground truth value. The raw data are summarized in tables 45, 46 and 47:

9.2.3 Acceleration detection

The primary goal of the acceleration detector is to find rapid decelerations. Our basic approach is to sub-sample average velocity over short periods and measure the difference between samples. Unfortunately, there is a lot of noise between successive short samples, in this case 10 second samples as shown by the points in the top of Figure 56. We use a 300 second moving average of all non-zero samples to smooth the data, plotted as a solid line on the same graph. This induces a short time delay, but it also significantly reduces the probability of a false alarm. Next, we take the difference between the current smoothed sample and one 60 seconds earlier. This is shown as a solid line in the middle plot of Figure 56. Next, we threshold this curve to find periods of rapid deceleration, as indicated by "X"'s. Finally, we count the number of samples below the threshold out of the previous 30 samples, as shown as "o"'s in the bottom of the figure. This

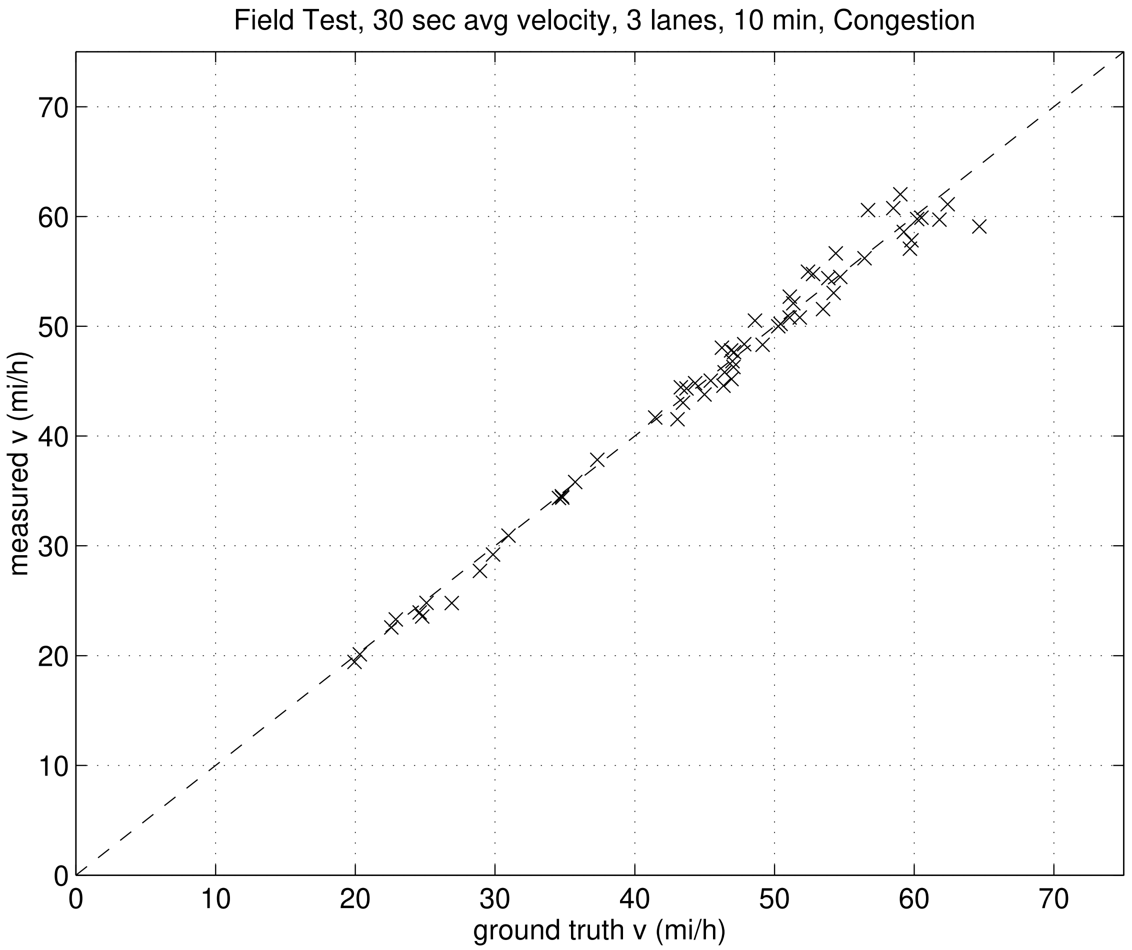


Figure 54: Scatter plot of vision and manually computed velocity. Tape: Southbound Florin Rd, real time. Date: 10/23/96. Time: 4:50-5:00. Conditions: evening rush hour, long shadows. Manually calibrated 30 sec samples.

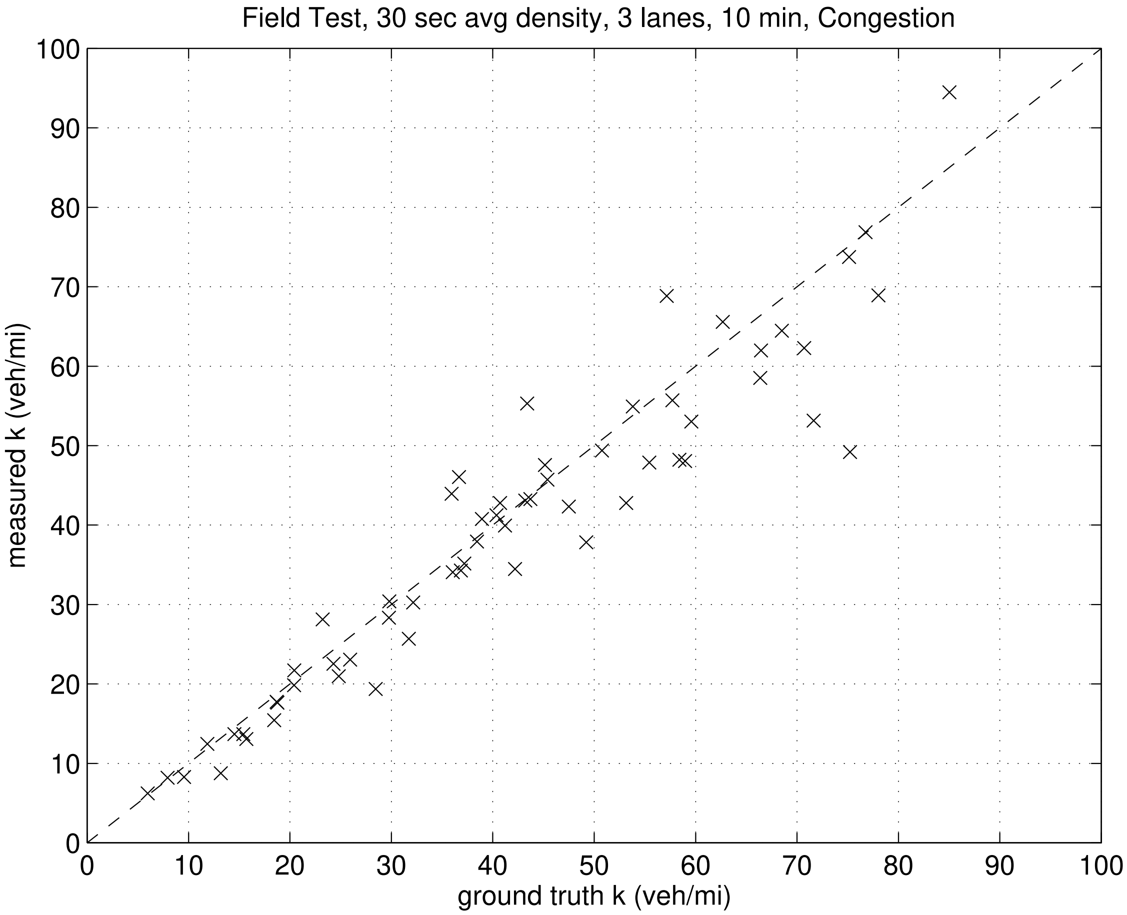


Figure 55: Scatter plot of vision and manually computed density. Tape: Southbound Florin Rd, real time. Date: 10/23/96. Time: 4:50-5:00. Conditions: evening rush hour, long shadows. Manually calibrated 30 sec samples.

lane 1 flow		lane 2 flow		lane 3 flow	
true	measured	true	measured	true	measured
9.8896	12.3484	14.3508	9.5482	11.3821	10.4310
6.3383	6.2671	14.3964	15.8402	11.7901	12.3365
5.4044	5.6577	16.1148	13.5353	15.7397	15.3295
13.7621	13.1344	18.6906	17.9502	14.7886	12.8784
9.2916	8.4457	14.1764	10.8290	16.0554	10.9805
11.4061	10.3662	16.7420	16.0993	16.0133	14.9038
7.5075	6.9267	16.5626	19.8096	17.1292	15.0863
12.1141	8.1940	17.2404	15.0927	17.7416	15.8285
10.1688	9.5567	15.2264	15.3505	15.6012	16.5198
8.7172	7.7878	16.6759	17.2207	15.8052	20.4310
4.0000	4.0828	16.6877	16.8160	14.0465	17.5037
3.0000	3.1038	19.2497	18.5100	15.0542	14.7995
12.0000	11.0000	17.4299	17.4611	13.4524	12.5218
5.1471	4.0810	15.5769	14.6502	15.4914	14.3493
11.6733	9.8234	17.1373	16.6947	11.7008	12.0653
16.1797	15.4439	19.2286	18.5367	17.0473	14.6475
5.9750	4.1382	20.8204	20.9665	16.8154	13.8961
10.5219	10.7994	17.4888	18.3093	13.2787	16.5132
9.7539	8.9888	22.1074	17.5342	12.2537	9.5449
7.7243	6.7500	21.5677	18.0138	19.1785	15.8409

Table 45: Flow, reported in number of vehicles per 30 sec sample. Southbound Florin Rd, comparison of vision data with manually calibrated data.

lane 1 dens		lane 2 dens		lane 3 dens	
true	measured	true	measured	true	measured
23.2255	28.1165	75.2193	49.1983	68.5057	64.4588
14.5171	13.6806	85.0186	94.4939	62.6735	65.5805
11.8534	12.4585	78.0275	68.9114	76.7504	76.8635
32.1571	30.2546	75.1242	73.7432	70.7024	62.2816
18.6802	17.7607	49.2096	37.8210	71.6342	53.1592
25.9406	23.0591	57.7201	55.6992	66.4549	61.9691
15.3976	13.6790	57.1472	68.8563	66.3797	58.5264
28.4537	19.3567	55.4416	47.8616	59.5791	53.0378
20.4047	19.8278	38.9344	40.7630	45.1512	47.5565
18.4485	15.4270	40.7158	42.7777	43.3921	55.3063
7.9314	8.1805	41.2041	39.9420	35.9458	43.9436
5.9783	6.2320	43.1981	43.0816	38.4441	37.9258
24.2978	22.5287	40.3699	41.2515	29.7635	28.3286
9.5523	8.2884	37.1857	35.1642	36.8531	34.2869
24.8136	20.9737	43.7002	43.2857	29.8039	30.3918
36.0511	34.0729	50.7747	49.3767	47.5060	42.3199
13.1827	8.7666	53.8124	54.9210	42.1893	34.4790
20.4300	21.7008	45.3879	45.7254	36.6731	46.0360
18.7575	17.6487	58.9631	48.0472	31.7285	25.6878

Table 46: Density, reported in vehicles per mile. Southbound Florin Rd, comparison of vision data with manually calibrated data.

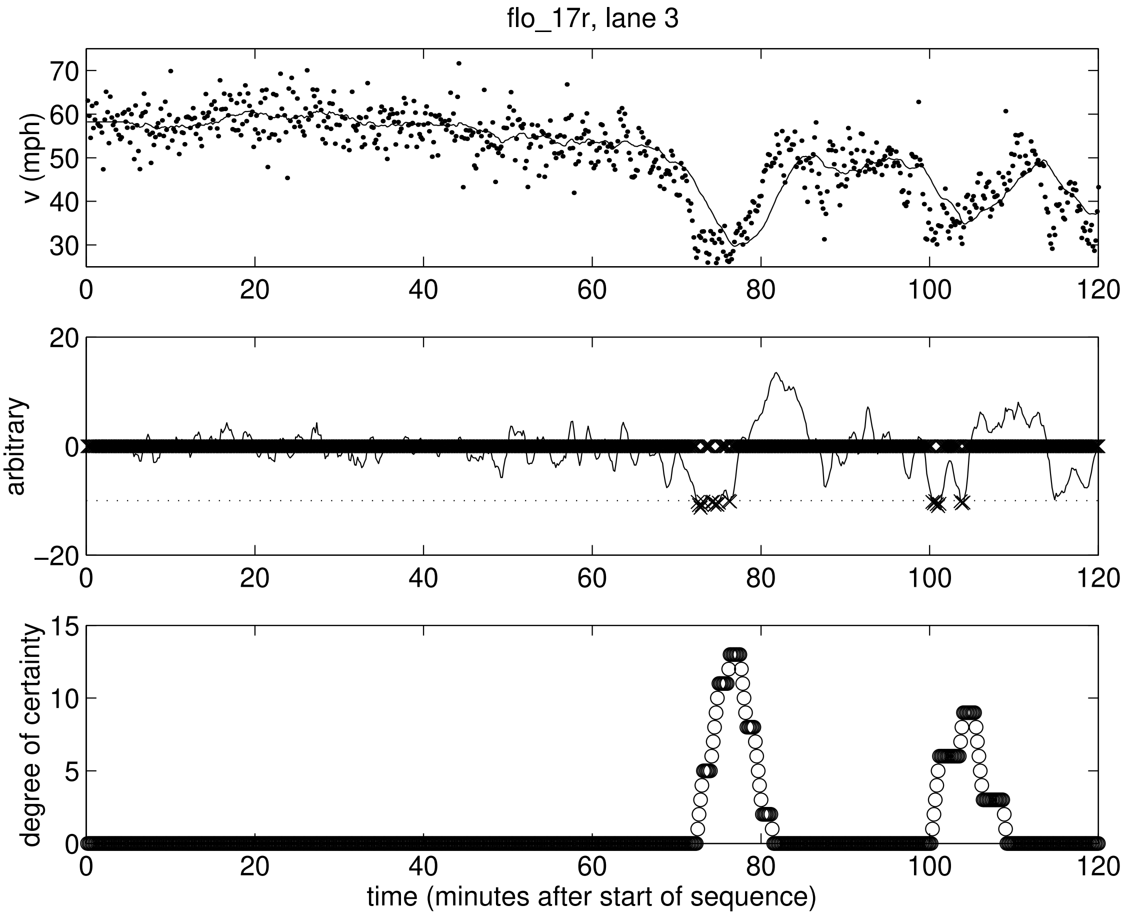


Figure 56: Accelation detection for Florin Rd. tape #17, lane 3. **Top graph:** 10 second sampled velocity estimates from tracker (points) with 300 second moving average (line). **Centre graph;** Acceleration plot with threshold and marked rapid deceleration points. **Bottom graph:** Accumulated deceleration counts.

lane 1 vel		lane 2 vel		lane 3 vel	
true	measured	true	measured	true	measured
51.0972	52.7022	22.8944	23.2890	19.9378	19.4190
52.3933	54.9721	20.3199	20.1159	22.5742	22.5735
54.7124	54.4948	24.7832	23.5699	24.6091	23.9326
51.3559	52.0953	29.8555	29.2097	25.0999	24.8133
59.6884	57.0635	34.5698	34.3587	26.8957	24.7870
52.7641	54.7555	34.8065	34.3931	28.9158	27.7207
58.5092	60.7649	34.7789	34.5233	30.9659	30.9324
51.0895	50.7977	37.3157	37.8410	35.7338	35.8126
59.8025	57.8381	46.9294	45.1895	41.4639	41.6847
56.7016	60.5781	49.1482	48.3075	43.7091	44.3299
60.5193	59.8907	48.6003	50.5213	46.8923	47.7986
60.2183	59.7647	53.4737	51.5579	46.9904	46.8265
59.2647	58.5920	51.8105	50.7941	54.2370	53.0424
64.6597	59.0847	50.2674	49.9946	50.4427	50.2209
56.4526	56.2038	47.0588	46.2822	47.1113	47.6393
53.8557	54.3914	45.4445	45.0497	43.0615	41.5336
54.3893	56.6456	46.4288	45.8110	47.8283	48.3636
61.8023	59.7181	46.2383	48.0501	43.4500	43.0443
62.3998	61.1179	44.9924	43.7923	46.3446	44.5888
59.0063	62.0280	44.3208	44.8202	43.2982	44.4355

Table 47: Velocity, reported in miles per hour. Southbound Florin Rd, comparison of vision data with manually calibrated data.

final measurement provides a rapid response to decelerations and increases in magnitude relative to the degree of deceleration. Thus, after an immediate indication of congestion, we have a slower measure of severity or, "degree of certainty", that takes an additional minute or two to calculate, but provides the user with important information. In sum, we have a rapid indication of deceleration followed by a slower measure of magnitude. The measurement could easily be extended to test multiple thresholds simultaneously and/or to use multiple sample periods.

Figure 57 is an example of the deceleration detector working properly when there are no decelerations. Figure 58 shows the detection of two shock waves, the first with a 40 second response time and the second with a 70 second response time. Figure 59 shows the detection of one shock wave with a 60 second delay. Figure 60 shows two shock waves that are just below the threshold detection. This figure shows the trade off between detection sensitivity and risking false positives. If the traffic engineer were interested in the disturbances shown, a lower threshold or multiple thresholds could be used to detect them. Figure 61 shows the detection of two shock waves, the first with a 30 second response time and the second with a 50 second response time.

9.2.4 Lane change detection

Mack Rd sequence #18 was selected to test lane change detection due to the heavy congestion (including several stop waves) and high frequency of lane changes.

	#observed by human	#not detected by tracker	#false positive by tracker
heavy congestion:	20	10	2
light congestion:	27	6	1

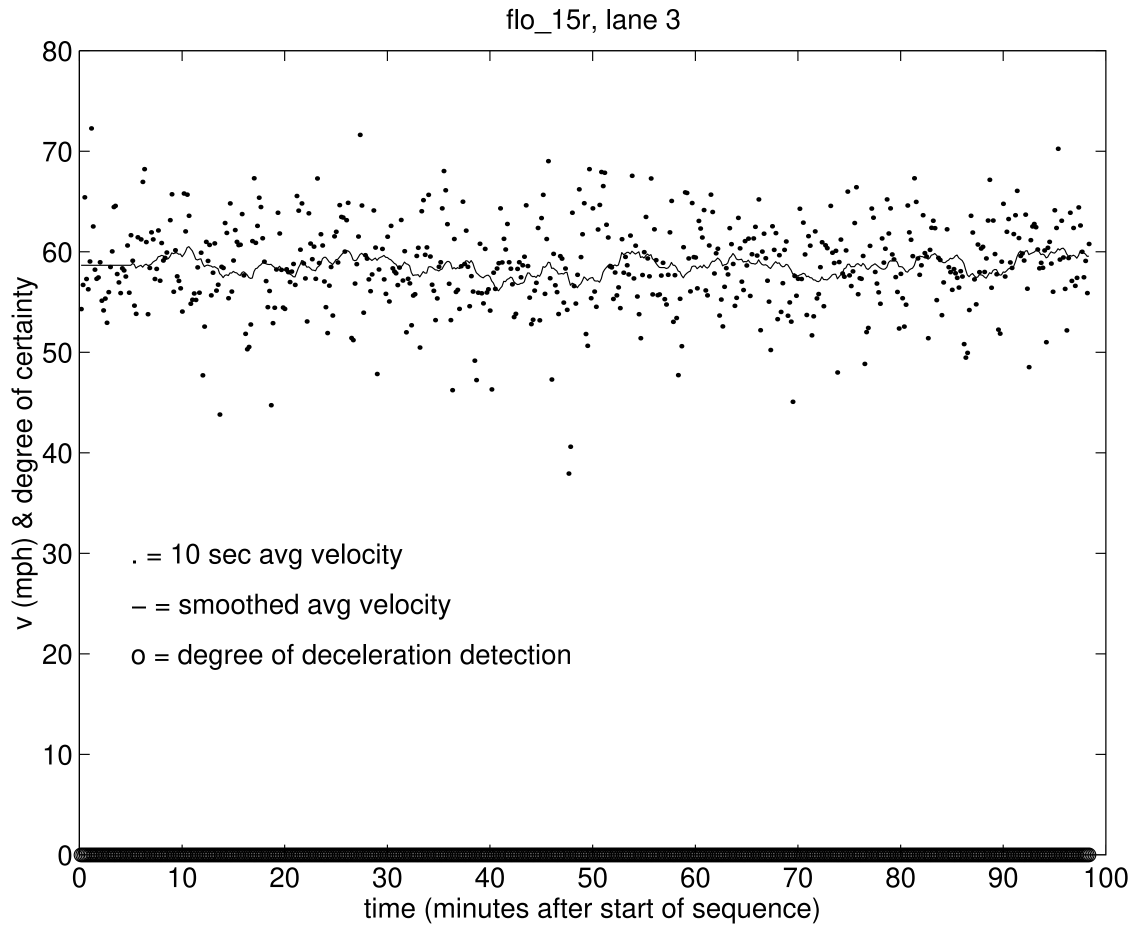


Figure 57: Accelation detection for Florin Rd. tape #15. 10 second sampled velocity estimates from tracker are shown as points, with a 300 second moving average plotted as a thin line. No decelerations are detected, so the deceleration count plot (thick line) is zero everywhere.

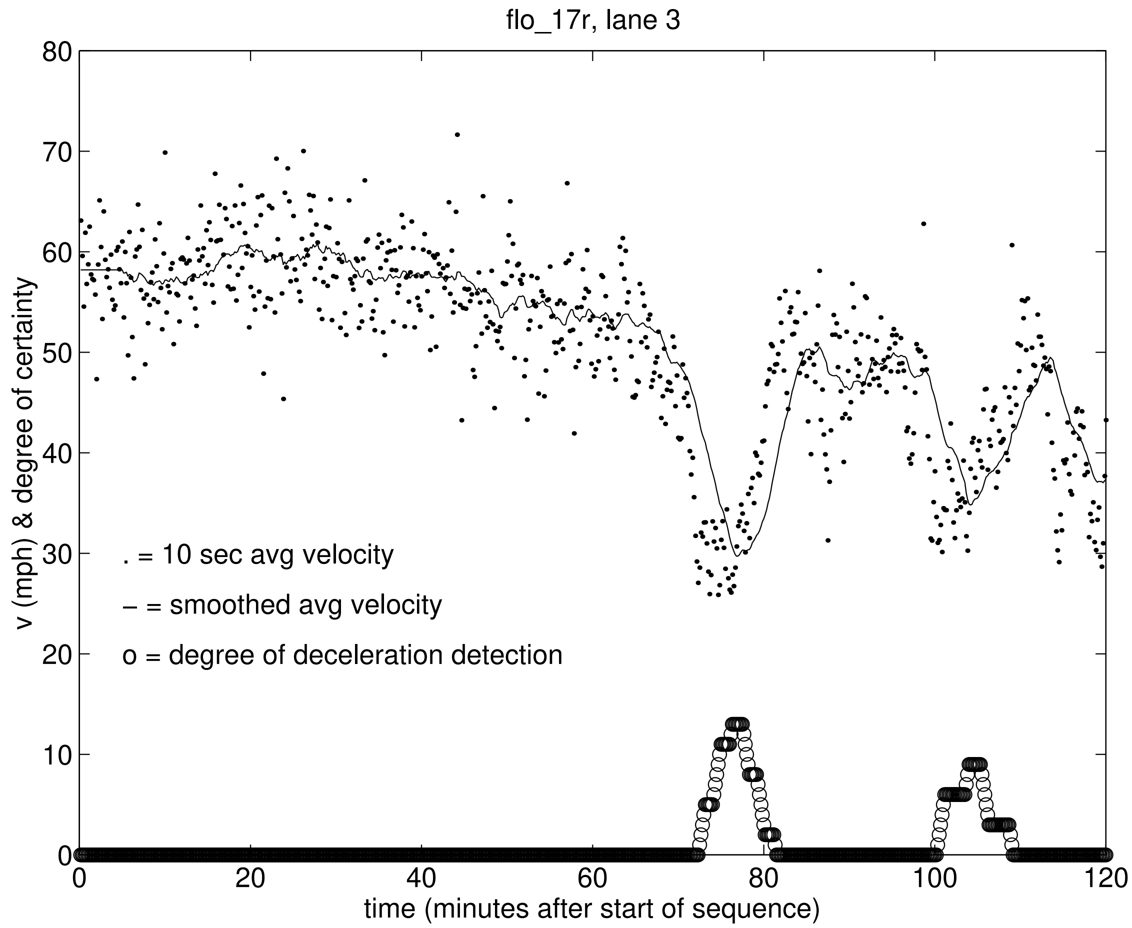


Figure 58: Accelation detection for Florin Rd. tape #17, lane 3. 10 second sampled velocity estimates from tracker are shown as points, with a 300 second moving average plotted as a thin line. Two shock waves are detected in the deceleration plot (thick line).

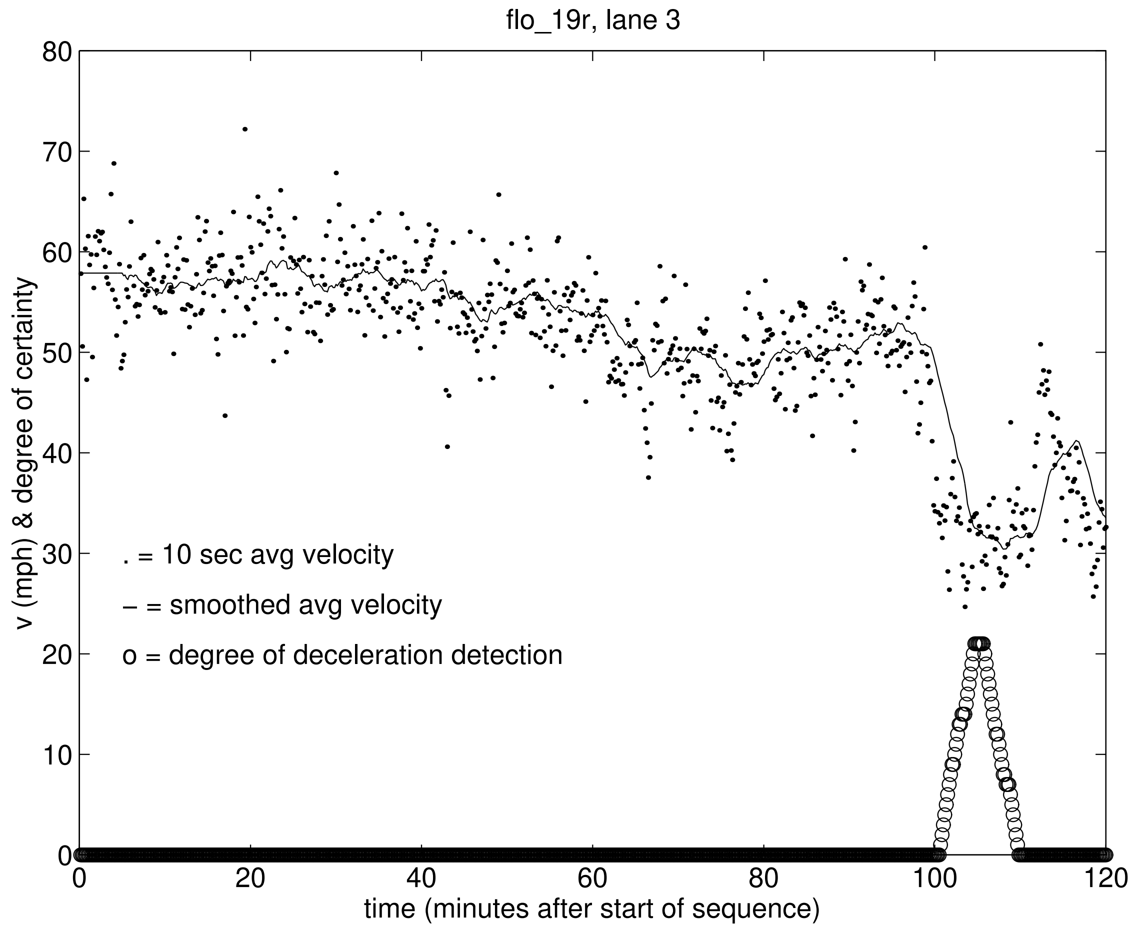


Figure 59: Accelation detection for Florin Rd. tape #19, lane 3. 10 second sampled velocity estimates from tracker are shown as points, with a 300 second moving average plotted as a thin line. One shock wave is detected in the deceleration plot (thick line).

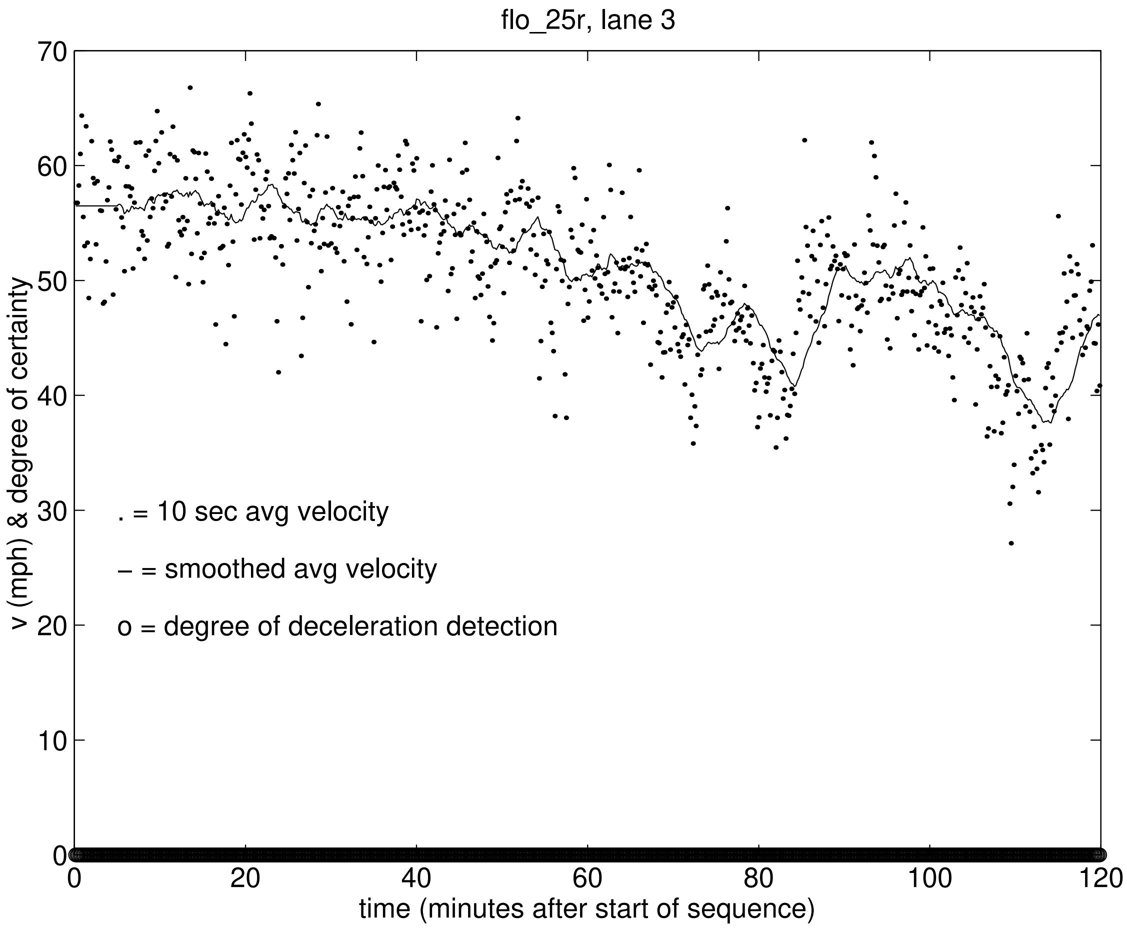


Figure 60: Accelation detection for Florin Rd. tape #25, lane 3. 10 second sampled velocity estimates from tracker are shown as points, with a 300 second moving average plotted as a thin line. No shock waves are detected in the deceleration plot (thick line).

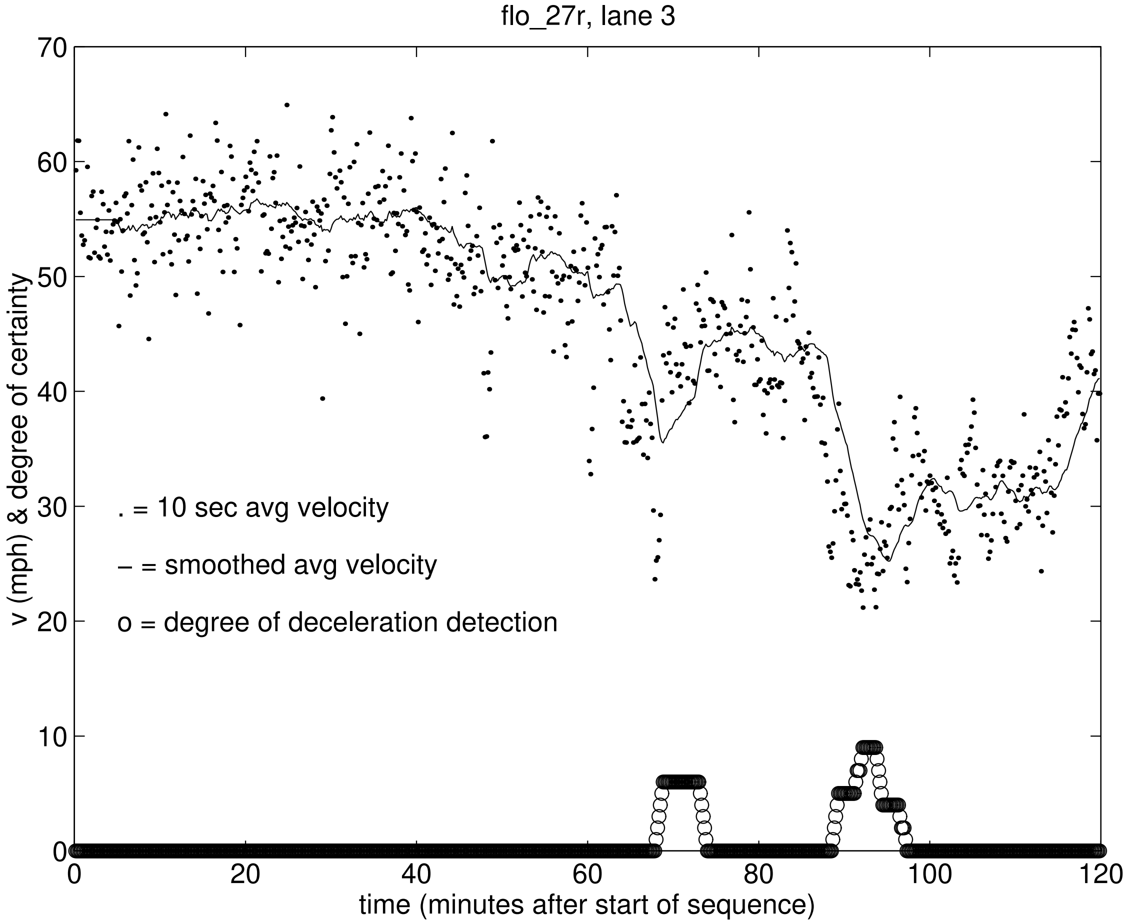


Figure 61: Accelation detection for Florin Rd. tape #27, lane 3. 10 second sampled velocity estimates from tracker are shown as points, with a 300 second moving average plotted as a thin line. Two shock waves are detected in the deceleration plot (thick line).

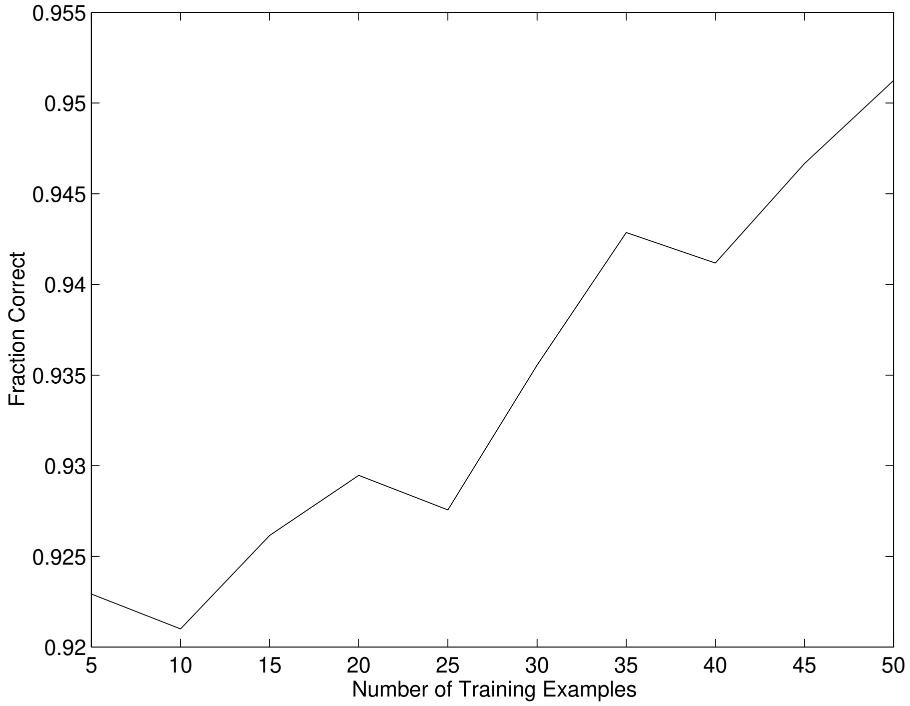


Figure 62: Learning Curve for the Classifier.

The Mack Rd camera angle proved to be non-ideal and the tracker lost several vehicles under congestion. All of the undetected lane changes were due to tracks being dropped before reaching the exit region, while the false positives were due to over-segmentation. Thus, all of the errors came before the lane change detector.

9.2.5 Classification results

The effectiveness of the size-based approach was evaluated using learning curves. A set of 210 vehicles was classified by a human. A supervisor routine then randomly chose a subset of these vehicles and provided their measurements to the classifier's training routine with the vehicle classes attached. The training routine then used these measurements to learn the means and covariances for each class. Next, the supervisor routine submitted measurements without vehicle classes to the classifier's test routine and asked which class each vehicle belonged to, making sure that the sets of test and training vehicles were disjoint. This process was repeated for different size training sets and the results for each size were averaged over several iterations to reduce noise.

Figure 62 shows the results of running the learning curve evaluation algorithm on the set of 210 vehicle instances. The evaluation was repeated 5 times for each training set size and the results were averaged. Note that there are slight fluctuations in the learning curve caused by the random choice of training sets, but that the fraction correct increases with training set size as expected. Appendix F shows the confusion matrix for the trained system.

A number of problems remain to be resolved in our classifier. Estimates of camera position were based on guesses provided by the person who collected the test videos. It is apparently difficult to obtain accurate measurements of distances from site drawings since those drawings often focus on features not visible to the camera, such as underground pipes and electrical conduits. We are attempting to find drawings of the test sites which show measurements between visible surface landmarks.

The classifier and color histogram generator currently works only with the off-line version of the tracker because we have not yet solved the logistic problems associated with storing past video frames in the on-line DSP network. It is necessary to have some subset of past frames stored so we can obtain color information from the largest picture of a vehicle available.

The data sets we tested with did not contain enough examples of buses or motorcycles to calculate an unbiased estimate of the covariances for these classes. When a class is underrepresented like this the classifier is programmed to reject that class as a possible hypothesis for new data until it sees enough training data from that class to from the unbiased estimate.

9.2.6 Queue length detection

At the conclusion of the project, the queue length detector was still too noisy to test. Empirically, for about two thirds of the time the detector is within 10' of the correct end of queue location. The small error is due to the fact the tracker may only detect a portion of a car, rather than the entire vehicle. The remaining portion of the time, either the queue length detector has erroneously selected a noise feature upstream of the queue or the last vehicle in the queue is undetected. The noise features seem to be due to the frame grabber and most are filtered out by Kalman filter. However, the queue detector and Kalman filter are operating in parallel on the features and any noise features are potential candidates for the end of queue until they are eliminated. One possible solution for this error is to add a short time delay, say five seconds, to the queue detector. The viability of this option depends on the application.

9.3 Multiple site level results

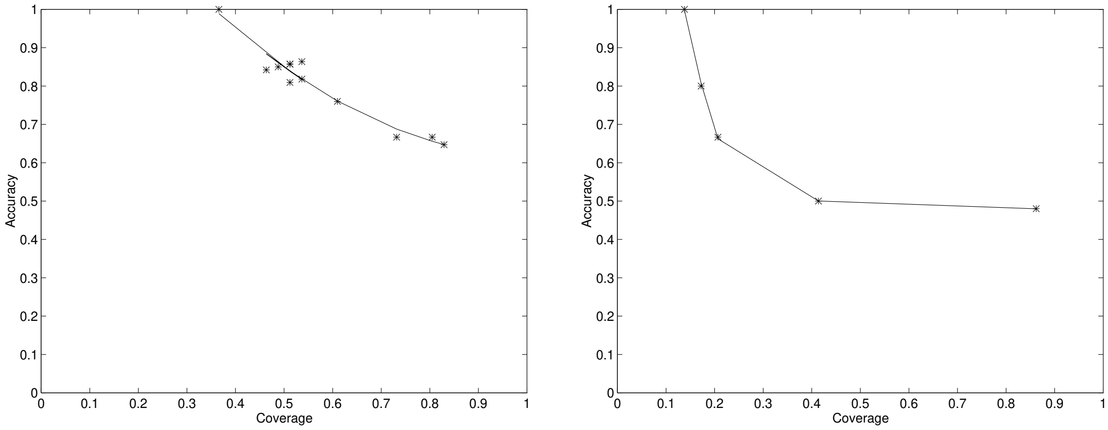
We tested the vehicle matcher with data generated by the region-based tracker on video sequences from adjacent cameras at Mack Road (upstream) and Florin Road (downstream) in Sacramento.

On any given run, the number of matches proposed by the vehicle matcher depends on the reliability threshold selected for that run. In the results discussed below, **coverage** refers to the percent of vehicles that actually appear at both cameras for which matches were proposed, and **accuracy** refers to the percent of proposed matches that were in fact correct. In general, the coverage goes down as the reliability threshold is increased, but the accuracy goes up. In other words, as the reliability threshold increases, the matcher returns only the vehicle matches for which it has the most confidence. The RFP requires achieving 90% accuracy with 10% coverage. Our results below show that when the reliability threshold is set so as to achieve at least 90% accuracy, the coverage is well above 10%.

To verify the accuracy of the matcher, the correct, or **ground-truth**, matches were determined by a human viewing the videotaped sequences themselves and the digitized sequences with the aid of a frame-based movie viewer. Since this method required about 3 hours of viewing to match each minute's worth of video, it was used only during the early stages of testing. In subsequent testing, we first ran the matcher on the vehicle report data and then used the frame-based movie viewer to verify whether or not the matches suggested by the vehicle matcher were in fact correct.

The first test results were carried out on a specially prepared sample in order to eliminate extraneous sources of error. From a pair of 60-second sequences from Mack Rd and Florin Rd, we extracted just those vehicles (41 in all) that appeared and were detected at both cameras, discarding undetected vehicles and vehicles that exited or entered the freeway between the two cameras. Ground truth matches were used to estimate the required probability models. Figure 63(a) shows the resulting accuracy/coverage curve.

The second set of results was obtained under more realistic conditions. We used 60-second sequences at the same locations, containing 29 vehicles upstream and with no preprocessing to remove undetected or entering/exiting vehicles. Furthermore, the probability models were estimated online from the system's own proposed matches. Figure 63(b) shows the resulting accuracy/coverage curve. The system achieved 100% accuracy with a coverage of 15%, and 50% accuracy with a coverage of 80%.



(a) (b)

Figure 63: (a) Results for "cleaned" sequences of 41 vehicles, showing accuracy achieved at different levels of coverage. (b) Results for later, raw sequences with 29 upstream vehicles.

It should be noted that due to the excessive costs of ground truth acquisition, our sample sizes are small. It should also be noted that the current level of vehicle matching performance is probably not sufficient to sustain long-distance O/D counts where vehicles must be tracked over several camera sites; this capability would also require somewhat higher vehicle detection rates. We can performa crude analysis as follows: if the coverage for the vehicle matcher is c, and the matching accuracy is a, and the vehicle detection rate is p, then the probability that a vehicle is correctly tracked across n sites is $p^n a^{n-1} c^{n-1}$. Suppose now that $n = 10$. To achieve 90% accuracy in O/D counts, we need $a^9 = 0.9$ or $a \approx 0.988$ as well as a sufficiently high number of tracked vehicles in order to keep sampling error low. The required percentage of vehicles to be tracked across the 10 sites will depend on flow rates and the length of the reporting period. To track, say, 10% of vehicles across 10 sites we need $p^{10} c^9 = 0.1$. Given $p = 0.95$, this means we need $c \approx 0.82$. Currently, simultaneous achievement of 98.8% accuracy and 82% coverage is not feasible. However, we anticipate that improved measurement of features such as width and height would provide dramatic improvement in coverage and accuracy. This as might be obtained by using edge tracking in addition to corner tracking. Other possibilities include selecting a subset of pixels from the rear plane of each vehicle to be used as a match feature.

Link travel times computed from the matched pairs were accurate in both cases to within 1% over a distance of 2 miles, over a wide range of coverage/accuracy tradeoff points. This accuracy is maintained even when the matched pairs include some incorrect matches and despite the fact that only a fraction of all vehicles were matched. This suggests that matched vehicles are representative of the traffic flow.

References

[1] K.D. Baker and G.D. Sullivan. Performance assessment of model-based tracking. In *Proc. of the IEEE Workshop on Applications of Computer Vision*, pages 28-35, Palm Springs, CA, 1992.

[2] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association.* Academic Press, New York, 1988.

[3] Chris M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[4] Edie, L.C. (1963) "Discussion of Traffic Stream Measurements and Definitions", Proc. Second International Symposium on the Theory of Traffic Flow, OECD, Paris, France, pp 139-154.

[5] Arthur Gelb, editor. *Applied Optimal Estimation.* The MIT Press, Cambridge, MA., 1974.

[6] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J.Weber. Automatic symbolic traffic scene analysis using belief networks. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 966–972, Seattle, WA, July 31–Aug. 4, 1994.

[7] Klaus-Peter Karmann and Achim von Brandt. Moving object recognition using an adaptive background memory. In V Cappellini, editor, *Time-Varying Image Processing and Moving Object Recognition, 2*. Elsevier, Amsterdam, The Netherlands, 1990.

[8] Michael Kilger. A shadow handler in a video-based real-time traffic monitoring system. In *IEEE Workshop on Applications of Computer Vision*, pages 1060–1066, Palm Springs, CA, 1992.

[9] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based Object Tracking in Monocular Image Sequences of Road Traffic Scenes. *International Journal of Computer Vision*, 10: 257-281,1993.

[10] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proc. Third European Conference on Computer Vision*, pages 189–196, Stockholm, Sweden, May 2-6, 1994, J.-O. Eklundh (ed.),Lecture Notes in Computer Science **800-801**, Springer-Verlag, Berlin, Heidelberg, New York, 1994.

[11] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proceedings of the International Conference on Pattern Recognition*, Israel, November 1994.

[12] H.A. Mallot, H.H. Bulthoff, J.J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991.

[13] P. Michalopoulos. Vehicle detection video through image processing: the Autoscope system. *IEEE Trans. on Vehicular Technology*, 40:21-29, 1991.

[14] G.D.Sullivan. Visual interpretation of known objects in constrained scenes. In *Phil. Trans. Roy. Soc (B)*, 337: 361-370, 1992.

Appendix A: Camera Calibration

Camera calibration is required in order to extract real world parameters from image coordinates. Our system uses the projective transform which exists between the road plane and the image plane. This projection is a simple transformation between image features and locations on the road. A point on the road located physically at a distance (x, y) meters from the camera will be projected onto a point (i, j) in the image via the relation in homogeneous coordinates:

$$(i, j, 1) \propto A(x, y, 1). \tag{10}$$

The inverse projection is found by using the inverse of the matrix A.

The 3×3 matrix A can be determined from a simple calibration procedure. It is found by placing four lines on the image corresponding to four known lines on the road. Using the fact that equation 10 implies a relationship between lines in the image and road planes as well as points, the four line correspondences can be used to compute the elements of the matrix A. Four points in the image could have been used to compute the calibration instead of four lines, but lines are easier to register with one another than points and thus are less prone to calibration error.

While the ideal system would be able to automatically calibrate itself using the lane markers or special calibration markers, a fallback position is to do the calibration manually. In the automatic case, the system automatically detects line or point features and computes the matrix A. In the manual case, a sample image is transmitted to the TMC and the operator manually labels the line features. In an example of the manual case, Figure 64 shows a possible set of the lines needed for calibration. In this example, the lines are the shoulder boundaries on the inner and outer most lanes plus lines which cross all visible lanes at two fixed distances. These lines are drawn on the screen using a mouse or light pen. The user then enters the known distances of these lines and calibration parameters are then computed and sent back to the remote sensor.

Figure 64: Example lines needed for calibration of the system. These correspond to known lines on the road plane. Any four non-parallel lines can be used.

If the camera has a fixed mount, calibration is performed only once and can be done under good lighting conditions. If the camera is on a pan/tilt mount, then the calibration needs to be recomputed every time the camera is moved and one wants to run the tracker. In this case, the calibration procedure will have to be more robust to poor lighting conditions since there is no knowing when a TMC operator may wish to move a camera.

The visual line calibration method results in small calibration error which can be estimated given knowledge of the camera installation (height and inclination angle). This error results in a quantifiable positional bias in the estimates of positions and velocities of vehicles. This effect is expected to be much smaller than the other sources of position and velocity error.

Appendix B: Interfaces

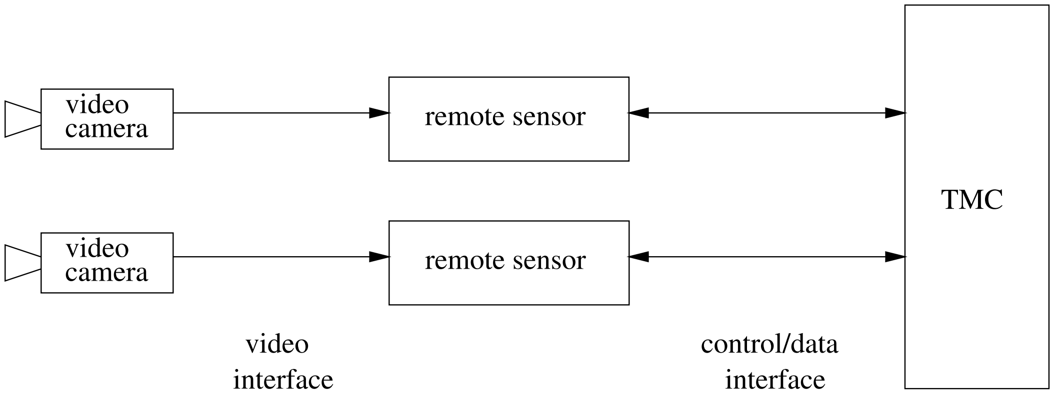


Figure 65: Interfaces between the major components.

The main components in our prototype system are videocameras, remote sensors, and the TMC, and these components are connected as shown in Figure 65. In this section we describe the two types of interfaces, the video interface between the cameras and remote sensors and the control/data interface between remote sensors and the TMC. For each interface we will list its main function, its physical characteristics and protocols, and the data format of information passing through the interface.

Video interface

The video interface connects a surveillance videocamera to a remote sensor. In normal operation, the remote sensor would be located in a cabinet located next to the camera or in a more accessible location on the side of the road. As mentioned previously, for demonstrating the prototype, the remote sensor may actually be located at the TMC. In this scenario, we would use a preexisting camera and video feed provided by our testing partner, Caltrans. Both scenarios for the location of the remote sensor will be discussed.

The function of the video interface is to transmit videocamera images to the remote sensor for processing. In describing the physical characteristics and data format of the interface, consider the two scenarios:

1. *Remote sensor at remote site.* Since the camera and remote sensor are near to one another, they can be connected by a coaxial cable. The data format will be a simple analog NTSC signal.

2. *Remote sensor at TMC.* In this case, the connection between the videocamera and remote sensor will use a a preexisting communications channel established by Caltrans. Possible example connections include fiber optic cable or a microwave link. The signal could potentially be compressed, but we plan to seek an uncompressed video feed. The data format will be determined by Caltrans, but it will likely be NTSC or JPEG or a similar format.

Control/Data interface

The control/data interface connects a remote sensor to the TMC computer. The function of this communications channel is to allow the TMC to issue commands to the remote sensor, and for the remote sensor to report on traffic scene summaries, tracked vehicles, etc.

	TMC command	remote sensor reply
tracker messages	start tracker	scene report vehicle reports
	stop tracker	acknowledge
	set reporting rates	acknowledge
calibration	calibrate	acknowledge
image messages	send image	compressed image
diagnostic	self test	diagnostic report
	future expansion	
camera position messages	query position set position	current position acknowledge
interrupts		traffic alert internal condition
operator messages	login reboot	 acknowledge

Table 48: List of messages passed between the TMC and remote sensor.

As for the video interface, the physical characteristics of the control/data interface depends on the placement of the remote sensor. Again, consider the following scenarios:

1. *Remote sensor at remote site.* Communications will be handled using a modem link over a standard telephone line, using a standard modem protocol for 28.8 Kbps.

2. *Remote sensor at TMC.* In this simplified scenario, the remote sensor and TMC computers will be sitting in the same room at the TMC. A simple connection strategy – and one that is not too different from the modem connection – is to use a serial link between the serial ports of the two machines. The protocol will be the RS 232 standard, and the serial line will be configured to 28.8 Kbps.

The data format of the control/data interface will consist of messages sent back and forth between the remote sensor and TMC. Arranging the communications link around message passing will facilitate making our protocol comply with NTCIP in the future, which itself is object oriented and message-based. Most of our messages will fall into one of two broad message classes: **control** messages issued by the TMC to the remote sensor, and **data** messages that relay scene and vehicle information back to the TMC. In Table 48, which lists the messages, one can see that most of the messages are paired into TMC commands and remote sensor replies. In the remainder of this section we describe the messages from Table 48.

start tracker The remote sensor begins tracking vehicles and returns scene and vehicle reports at regular intervals. Parameters included in this command message are the reporting rates for the scene and vehicle reports.

scene report A summary set of traffic statistics that are averaged over a user-definable time period. The statistics include:

1. *Camera ID.* The unique ID number of the remote sensor and camera.
2. *Timestamp.* The time at which the measurements were taken.
3. *Vehicle flow rate.* The number of vehicles that have passed a detection zone in the averaging period. This is measured separately for each lane in terms of vehicles/hour/lane.

4. *Average speed.* The speed, measured at a given point on the roadway for all vehicles. It is calculated separately for each lane and is averaged over the averaging period.
5. *Classification counts.* The number of vehicles of each class passing through the detection zone for the averaging period. The classes are: 1=motorcycle, 2=automobile/pickup truck, 3=buses, 4=trucks up to 40 feet in length, 5=trucks greater than 40 feet in length.
6. *Lane changes.* This is the number of vehicles changing lanes over the averaging period, measured separately for vehicles switching to the right or left for each lane. For intersections the lane changes reflects the right and left turning vehicle counts.
7. *Queue length.* For intersections, this parameter refers to the number of vehicles awaiting to be cleared by a green cycle, and is also known as Approach Queue Length. For freeways, this parameter measures the ramp queue lengths.
8. *Spatial headway.* This is the distance between identical points on consecutive vehicles in a given time period and is averaged over the vehicles. It is equal to the sum of the vehicle length and the gap between the vehicles.
9. *Average acceleration.* This is the rate of change of vehicle speed and is averaged over all vehicles in the averaging period.

vehicle reports Vehicle reports describe an individual vehicle tracked by the remote sensor. There are two types of vehicle reports:

location report Report on the vehicle's current position and speed. It contains the following parameters:

1. *Camera ID.* The unique ID number of the remote sensor and camera.
2. *Vehicle ID.* The unique ID number of the tracked vehicle.
3. *Timestamp.* The time of the position/speed measurement.
4. *Position.* The position (x, y) of the vehicle, measured in feet, where x measures lateral position within the road lanes, and y is arclength along the road lanes. A local coordinate system will be used, where measurements are made relative to an origin placed near the camera.
5. *Velocity.* The velocity $(\dot{x}, \dot{y})$ of the vehicle, measured in miles/hour.

description report Report on static, descriptive vehicle parameters that can be used for matching vehicles between cameras.

1. *Camera ID.* The unique ID number of the remote sensor and camera.
2. *Vehicle ID.* The unique ID number of the tracked vehicle.
3. *Timestamp.* The time of the parameter measurements.
4. *Color.* The color of the vehicle. Format to be determined.
5. *Appearance.* Appearance parameters of the vehicle. Format to be determined.
6. *Shape.* The estimated width, height, and length of the vehicle, measured in feet.
7. *Classification.* Vehicle class, a number from 1-5. (See classification counts above.)

stop tracker The remote sensor halts the tracker, which results in a cessation of scene and vehicle reports.

set reporting rates Change the reporting rates for the scene and vehicle location reports. The parameters for this command message are the new reporting rates, where the unit is the time period in seconds between reports.

calibrate The TMC instructs the remote sensor to perform the calibration procedure. If the calibration procedure is completely automatic at the remote sensor, then this message has no arguments. In calibration is manually performed, then the message has as arguments the calibration parameters extracted/computed at the TMC.

send image Request that a compressed version of the current camera image be sent to the TMC computer. If the tracker is presently running, this may result in some scene and vehicle reports being dropped in order to devote bandwidth to image transmission.

compressed image A compressed version of the current camera image that is sent from the remote sensor to the TMC. The format will be a JPEG encoded image.

self test Instruct the remote sensor to perform a self diagnostic check and report the results.

diagnostic report A report from the remote sensor to the TMC relaying the results of a self test. Format to be determined.

The following messages are to be considered for future expansion. For instance, if the video-camera is on a pan/tilt mount, then additional commands are required to handle these rotation angles. In addition, it may be useful to allow a remote sensor to interrupt the TMC to inform it of a rapid deterioration in traffic conditions or of an internal malfunction.

query position Command message from the TMC to remote sensor to return the current camera parameters, such as pan, tilt, and zoom.

current position Report message from the remote sensor to the TMC on the current camera parameters.

set position Command message to set the current camera parameters. The parameters to this message include things like pan, tilt, and zoom.

traffic alert The remote sensor interrupts the TMC to make a special traffic report (e.g. rapid slowdown in traffic flow). Format of this message to be determined.

internal condition The remote sensor interrupts the TMC to report an internal malfunction that has been detected by the remote sensor. Format to be determined.

login The TMC issues this command when it wants to change the mode of communication from normal message passing to a Unix login shell. This allows operators at the TMC to log into the computer at the remote sensor. This may be useful for performing a thorough debugging job or downloading new remote sensor code.

reboot A command message that allows the TMC operators to reboot the PC at the remote sensor.

Appendix C: Computation of 3-D Shape

Determining the three dimensional shape of an object from image information is difficult because of the projection from three to two dimensions which occurs during imaging. In fact, without model constraints the task is impossible from a single view because of the projection. In the surveillance domain we can use constraints from the geometry of the system to obtain three dimensional shape information from planar images. We make use of the fact that vehicles are constrained to lie on the road surface. In addition, we assume a simple rectangular model of the vehicle. This model consists of the size parameters we are interested in: the length l, width w and height h of the vehicle. Using these two constraints there exists a linear mapping between image coordinates and the real world values. The mapping is a function of camera location which is assumed known.

Figure 66 shows the generic vehicle model used. From the locations of the points defining the bounding box (points A, B, C, D in the image), the size of the vehicle can be obtained. Since we know the camera calibration matrix A, a linear relationship between image points (X_i, Y_i) and size parameters (W, L, H) exists:

$$(X_A, Y_A, X_B, Y_B, X_C, Y_C)^T = M \begin{pmatrix} \mathrm{W} \\ \mathrm{L} \\ \mathrm{H} \end{pmatrix} \tag{11}$$

From the pseudo-inverse of the matrix M, a least squares solution for the size parameters (W, L, H) can be found. The matrix M is a function of camera calibration parameters and the location of the vehicle in the image.

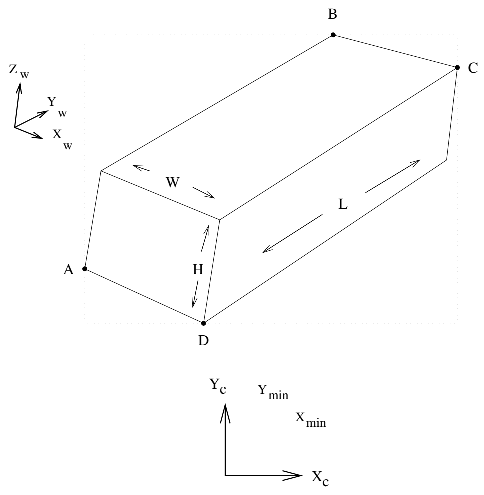


Figure 66: Simple vehicle model used for obtaining vehicle size from image measurements.

Appendix D: Parameter Measurement

Introduction

This section outlines the parameter measurement strategies. In particular, we provide a brief review of Edie's traffic stream measurement definitions. Edie's definitions for measuring flow, density and velocity are ideally suited for the area wide detection of the vehicle tracker. Next, the working definitions for: link travel time, vehicle classification, lane changes, queue length, acceleration, and O/D tracking are presented.

Parameter Definitions

Flow, Density (Spatial Headway) and Velocity

Traditionally, traffic engineers have defined flow as the number of vehicles passing a point in space over a fixed interval of time (Figure 67).

Following the method first proposed in [4], the loop detector occupies a small finite length of roadway, *dl*. We extend the definition of flow to account for this distance (Figure 68).

Finally, we can generalize the definition to an arbitrary region in the time-space plane by integrating several horizontal slices (Figure 69).

If we define $t(A)$ as the time spent by all vehicles in region A, we can start with a vertical line (i.e., trajectories over space at an instant in time) and repeat the analysis for density. The resulting equations are:

$$density = k = \frac{n}{L} \Rightarrow k(A) = \frac{n \cdot dt}{L \cdot dt} = \frac{t(A)}{A} \Rightarrow k(A) = \frac{1}{T}\int_A \frac{n}{L} dt = \frac{t(A)}{A}$$

Furthermore, for a given region in the time space plane,

$$average\ velocity = v(A) = \frac{q(A)}{k(A)} \Rightarrow v(A) = \frac{d(A)}{t(A)}$$

To calculate the traffic state in a given region, A, of the time-space plane, we simply need to calculate the initial coordinates, (t_{s_j}, x_{s_j}), when the j-th vehicle enters A and the final coordinates, (t_{f_j}, x_{f_j}), when j-th vehicle leaves A. This data will yield individual vehicle time and distance within A, which can be summed to generate d(A) and t(A). In practice, we will treat each lane individually, set equal to the maximum range of tracking and allow the user specify the sampling period, . Note that this methodology is robust for lane changes, e.g., the final coordinates in the originating lane for a vehicle making a lane change maneuver correspond to the initial coordinates in the destination lane.

Link Travel Time

We calculate individual link travel times for as many vehicles as possible between two consecutive detection locations. We report the average link travel time during the observation period across all lanes and by individual lanes (as detected at downstream location). We also report the sample size for each lane.

Vehicle Classification

For a given observation period, the total number of vehicles in each class is reported by lane. Classification relies on vehicle features and will utilize artificial intelligence.

Lane Changes

For a given observation period, the total number of lane change maneuvers between adjacent lanes is reported individually (e.g., from lane 1 to 2, lane 2 to 1, lane 2 to 3, etc.).

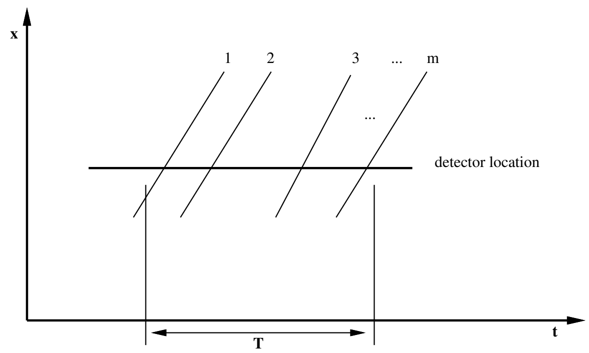


Figure 67: Time-space diagram showing m vehicles crossing detector in sample period T. Flow is given by $q = m/T$.

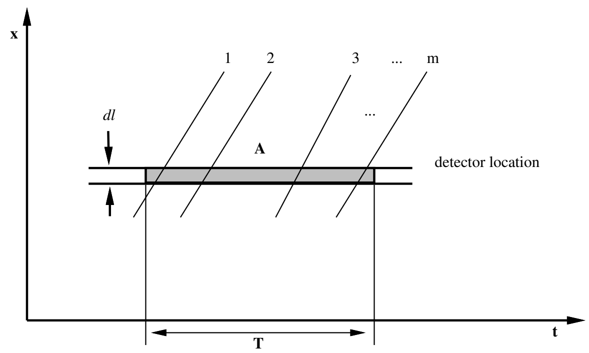


Figure 68: Time-space diagram with extended detector region of length dl. Let A be the time-space area sampled by the detector in the sample period. Let $d(A)$ be the distance traveled in A by all vehicles in the sample period. Then $q(A) = m/T = (m \cdot dl)/(T \cdot dl) = d(A)/A$.

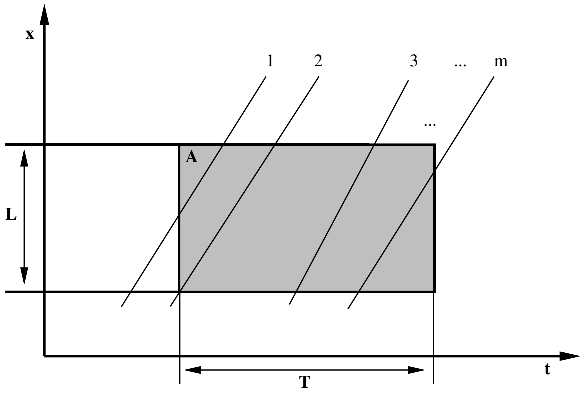


Figure 69: Time-space diagram with arbitrary detection region of length L. Let A be the time-space area sampled by the detector in the sample period. Let $d(A)$ be the distance traveled in A by all vehicles in the sample period. Then $q(A) = \frac{1}{L} \int_A \frac{m}{T} dl = d(A)/A$.

Queue Length

At the end of a reporting cycle, we report both the current and last known queue length for each lane. These parameters will be provided in feet (or meters) and number of vehicles between the stop bar and the end of queue. To avoid problems when the queue starts to dissipate, the end of the queue will be defined as the largest of the following: (a) the last vehicle that stopped (b) the last vehicle ever considered in the queue, even if the vehicle is moving, thus, making our measurements robust to any gaps that may form in a discharging queue (c) the last vehicle before density drops below a user specified threshold (i.e., a minimum gap flag), thus, allowing us to detect queued vehicles that slow, but do not stop or have stopped beyond the end of the detection region. In the event a queue overruns the detection region, we will report the queue distance, "in excess of maximum detectable" while returning the actual number vehicles in the detection region as the number of vehicles in the queue.

Acceleration

Acceleration is defined as the change in average speed over a short, "sub-sample" time period. The user will specify the observation period and number of sub-sample time periods (minimum of one). We will report the average and maximum acceleration for a given observation period by lane.

In particular, this definition includes rapid decreases in speed that may be useful for detecting incidents. As such, it would be a simple extension to monitor the acceleration / deceleration rates and trigger an incident alarm when a pre-specified threshold is exceeded. Although other researchers have devoted considerable efforts to incident detection, it is still unclear what an appropriate threshold should be. Because of this fact, we will not attempt to specify the threshold level, we will only report the acceleration rate.

Origin/Destination Tracking

We calculate individual origin destination pairs for as many vehicles as possible. These data could be used to construct a net O/D matrix for a large network,. Our studies, however, are limited to two nodes.

While vehicles are between detectors, it will be impossible to tell if a vehicle has exited the network or is in transit until the link travel time has elapsed. To this end, we will maintain two O/D tables. One that only includes final information after vehicles have definitely left the network. The other O/D table will include vehicles that may still be in the network and changes dynamically as new information is collected.

Appendix E: Data Sources

We have utilized six data sources for video and ground truth data. For each source, we list the location, our contact person, their facilities, highlight any important features, outline the parameters tested at the site and the format of the collected data.

San Jose Department of Streets and Traffic

Contact person: Charles Felix (408) 277-3070
Facilities: 17 color CCTV installations at urban intersections on line at TMC
Status:
Parameters to test: flow, travel time, queue length, acceleration; possibly: velocity, classification.
Data format: video tapes, live video
Testing plan: collect recorded data periodically throughout the end of the study. Field test at TMC in July.
Notes:

Caltrans District 3 - Sacramento - Hwy 99

Contact person: Joe Palen (916) 654-8420
Facilities: 11 color CCTV installations on line, two sites used for six different views (see appendix B). 18 loop detector speed traps on line, 1/60th of a second sampling periods yielding vehicle arrival and departure times at the upstream and downstream loops 20 feet apart.
Status:
Parameters to test: flow, density, average speed, lane changes, travel time, O/D
Data format: video tapes, loop data
Testing plan: Collect recorded data periodically throughout the end of the study.
Notes: All three speed traps at Mack Rd and one at Florin Rd have noisy falling edges (i.e., +/- 1/12 seconds), one detector at Florin has noisy rising edges as well, one detector at Florin appears to have good rising and falling edges. We were able to apply Edie's principles using just the rising edges, and work around the loop detector problems.

Caltrans District 4 - Walnut Creek - Hwy 24/I680

Contact person: Michael Lee (510) 286-6142
Facilities: 20 b/w CCTV installations, three sites used (see appendix B); several loop detector speed traps on line, 1/60th of a second sampling periods yielding vehicle arrival and departure times at the upstream and downstream loops 20 feet apart.
Status: confirmed
Parameters to test: flow, density, average speed, lane changes
Data format: video tapes, loop data
Testing plan: Collect recorded data periodically throughout the end of the study.
Notes: All nine speed traps at the two loop detector stations have noisy falling edges (i.e., +/- 1/12 seconds). We were able to apply Edie's principles using just the rising edges, and work around the loop detector problems.

Caltrans H.Q.

Contact person: Joe Palen (916) 654-8420
Facilities: 1 color CCTV instillation on line at TMC
Status: in negotiation
Parameters to test: flow, average speed, vehicle classification, lane changes
Data format: video tapes, live video

Testing plan: if confirmed, field test at TMC.
Notes:

Roadwatch tape library

Contact person: Ben Coifman (510) 848-5121
Facilities: a large selection of tapes collected over the past four years to test and develop the tracker
Status: confirmed
Parameters to test: vehicle classification, lane changes, queue length, acceleration
Data format: video tapes
Testing plan: test throughout the end of the study.
Notes:

Institute of Transportation Studies tape library

Contact person: Michael Cassidy (510) 642-7702
Facilities: a large selection of tapes collected for various freeway flow and merging studies over the past several years. Some ground truth data available.
Status: confirmed
Parameters to test: vehicle classification, lane changes
Data format: video tapes
Testing plan: test throughout the end of the study.
Notes:

Appendix F: Classification confusion matrix

For convenience, the classification results can be broken down and displayed as a confusion matrix showing the probability that a vehicle of each type is classified as the same or another type. For our test sample of 210 vehicles, we tested two classifiers: one using five variables (height, width, length, x position, and speed; results shown in Table 49), and the other using height only (results shown in Table 50). The 210 vehicles consisted of 1 Motorcycle, 192 Cars, 0 Buses, 8 Small Trucks, and 9 Big Trucks. Considering all variables gives a lower overall correct rate (197/210) but shows separation between small and big trucks. Considering height only gives perfect classification of cars and big trucks but fails on other categories.

	Mtrcyc	Car	Bus	S.Truck	L.Truck
Mtrcyc	0.000000	1.000000	0.000000	0.000000	0.000000
Car	0.000000	0.994792	0.000000	0.005208	0.000000
Bus	0.000000	0.000000	0.000000	0.000000	0.000000
S.Truck	0.000000	0.625000	0.000000	0.375000	0.000000
L.Truck	0.000000	0.444444	0.000000	0.222222	0.333333

Table 49: Confusion matrix showing the probability that a vehicle of each type is classified as the same or another type. Results for classification using height, length, width, x-position, speed.

	Mtrcyc	Car	Bus	S.Truck	L.Truck
Mtrcyc	0.000000	1.000000	0.000000	0.000000	0.000000
Car	0.000000	1.000000	0.000000	0.000000	0.000000
Bus	0.000000	0.000000	0.000000	0.000000	0.000000
S.Truck	0.000000	0.750000	0.000000	0.000000	0.250000
L.Truck	0.000000	0.000000	0.000000	0.000000	1.000000

Table 50: Confusion matrix showing the probability that a vehicle of each type is classified as the same or another type. Results for classification using height only.

Appendix G: Acknowledgements

We would like to thank JPL, and particularly Rondle Nelson and Frank Condos, for their admirable contract management and innumerable suggestions. Additional funding was provided by CalTrans via PATH project MOU 152 and 214.

Principal researchers on the project included David Beymer, Phil McLauchlan, Joe Weber, Benn Coifman, Tim Huang, and Dan Lyddy.

Brian Simi, Michael Lee, Sean Coughlin and Joe Palen of Caltrans, and Charlie Felix of the City of San Jose, were extremely helpful in the data acquisition process and in many other ways.

We gratefully acknowledge additional research contributions by Chris Bregler, Chad Carson, and Daniela Musto.