Real-Time Traffic Monitoring

N. J. Ferrier S. M. Rowe A. Blake

Robotics Research Group
Department of Engineering Science
University of Oxford, U.K. OX1 3PJ

Abstract

Traffic statistics desired by road engineers and planners, and "traffic warning" systems demand real-time performance which precludes the use of batch processing. We apply recent real-time tracking techniques along with scene specific tuning of the dynamics to enable the tracker to accurately predict target location and thus reduce the amount of search and/or image processing required. The benefit of learning dynamics and accurate prediction is speed – our tracker operates at frame rate.

Initial calibration of the projective relationship between the image and ground planes enables metric information to be derived from the image positions and velocities without full camera calibration.

Results are presented on real-world traffic scenes showing the tracker to be both fast and robust to vibrations, which are inevitable in traffic locations.

1 Introduction and Related Work

Automatic surveillance of traffic has been of interest for many years. As early as 1980 road engineers were investigating methods for automatic surveillance of traffic [16, 24, 25]. The purpose of such systems is to obtain information on road usage in order to determine areas in need of expansion or requiring alteration of existing traffic patterns. More recently interest has focussed on "early warning" systems which would alert drivers to heavy congestion and/or accidents further ahead on the road. Such *on-line* systems would require *real-time* performance, and hence preclude the use of batch processing. Another requirement of a system designed to operate in a real world traffic environment is that the the vision/tracking system must be robust to vibration of the camera. Typically, a system will be placed either on a bridge or a large pole above the traffic area to be monitored. Both locations are subject to large amounts of vibration (from traffic and wind). In this paper we demonstrate a system that satisfies these two requirements: it achieves real-time performance and is robust to vibration.

An early tracking system of Hogg [7] used extensive search to locate features within the image. Although this was designed to track people, not cars, the tracker demonstrates one extreme: locating features in the entire image and searching through all possible combinations in order to track quite complicated objects (people) in image sequences. The computation required is intensive, however an instance of an object even in unexpected poses will be found. With a static camera, viewing traffic, there are many constraints on the possible locations of vehicles and on the possible motions between frames. Traffic monitoring systems have incorporated these constraints with varying degrees of success.

Much previous work on traffic monitoring has considered traffic monitoring as a model-based tracking problem and performed a full 3D analysis to extract vehicle information. Sullivan [20] and Tan *et al* [22, 23, 21] use a wireframe model of a vehicle. Based on an estimated pose (position and orientation), and full calibration information, the model is back-projected into the image and then the edges of the model are matched to lines in the image. The system requires full calibration of both intrinsic and extrinsic camera parameters, a task sensitive to noise. This system will not be immune to vibrations of the camera as the extrinsic calibration parameters vary as the camera vibrates. In order to estimate the pose, this system uses the ground plane constraint and then separates the three motion degrees of freedom on the plane (two translation components and one rotation component) in order to search individually in each dimension for the best pose estimate. This heuristic approach to find the best motion parameters requires the motion parameter space to be separable, which it is not. The implicit dependence on the precise knowledge of the

ground plane will leave the system sensitive to noise. Kalman filtering has be used to smooth the trajectory obtained [26] by this system.

Koller *et al* [10, 11] have also built a 3D model based system to monitor junctions. An elaborate bootstrapping procedure is used to initialise the tracking: first the image is segmented using optical flow to determine possible vehicle locations, then a pre-defined wireframe model is fitted the image edges. Once tracking has begun, Kalman filtering is used to improve performance. As tracking proceeds, the wireframe model is back projected into the image based on the current estimate of pose and this estimated position is used to search for the new position. Their model of the vehicle dynamics is questionable: they assume constant velocity, an assumption which will be violated at corners and intersections. This system also requires a model of the lighting (the sun position) which will change during the day. The results obtained are good, however, the system is computationally expensive and does not run in real-time.

Detecting and tracking cars based on symmetry has also been proposed [19]. The symmetrical view of a vehicle within an image is only possible from particular vantage points (on the ground, viewing a car while following it). Although this is certainly the case for cameras mounted in other vehicles, it will not be the best viewing position for monitoring traffic.

Other work [9, 12] simplifies the tracking by using the outline of the moving vehicle. Image differencing is used to initialise and run the tracker. In [12] the boundary of each vehicle is obtained as the convex hull of points obtained via the image differencing scheme. In [9] the connected components of the blobs detected by the image differencing scheme are computed then segmented into possible vehicles using size to determine single vs. multiple vehicle blobs. Both systems also deal with tracking in the presence of occlusion. From a vantage point above the motorway, the relative depth of tracked objects is encoded in the y component of the image. The computation of the relative depths of objects is used to resolve decisions on occlusions. Unfortunately, simple image differencing schemes will be sensitive to vibration of the camera and computation of the convex hull is computationally expensive. The template mechanism which we use in this paper eliminates some of the difficulties encountered with occlusions.

In a more 'AI' approach to the problem of monitoring vehicles, Mohnhaupt and Neumann proposed a traffic reasoning system [14, 15] which uses data provided from a (supposed) computer vision system to "reason" about traffic. The system attaches symbolic descriptions to the motion of vehicles. The assumed input is a set of motion parameters for each vehicle, the position and velocity of the vehicle on the ground plane, and a model or map of the ground plane. Given that input, this system can provide an account of the traffic, in symbolic form. (example: "car drove down street X, turned left at avenue Y"). This system is a demonstration that reasonable analysis can be performed using *only* the motion parameters of the vehicle on the ground plane.

In agreement with the systems of Mohnhaupt, Kilger [9] and Koller [12], we argue that, full 3D reconstruction is not necessary to obtain the desired traffic statistics. Model based systems running as batch processes require the storage of large streams of traffic data to obtain statistics and such systems will not be useful for on-road monitoring such as 'early warning' systems described earlier. Instead, we analyse which statistics are required and employ recent real-time tracking techniques [2, 3]. We incorporate the ground plane constraint to simplify the problem and by extracting motion parameters in the image along with information on projectivity between the ground plane and the image plane, we show that metric information can be obtained. We obtain the desired statistics *on-line*. We show that by tracking the occluding contour of the vehicle, we can obtain real-time performance using intensity/motion information.

The layout of this paper is as follows. In section 2, we explore which statistics are desired by traffic engineers. In section 3 we present the tracking system, which is an implementation of the tracker presented in [2, 3]. In section 4 we show how we compute metric information and in section 5 extraction of qualitative information is discussed. In section 6 we present results on tracking real vehicles on a motorway.

2 Traffic Statistics

A traffic monitoring system should provide the information required to alert drivers to problems, and/or, the statistics required by road engineers [16, 24]. A list of its features may include the following:

- vehicle speed, count, and lane occupancy (at a particular line across the road)
- surveillance over a length of road (tracking vehicles and lane-changing manœuvres)
- monitoring movement through a junction

- incident detection (speed changes on a section of highway)
- classification (register vehicle shapes and match against templates to fit the vehicle to a classification code)

Except for classification, all of the above do not require precise 3D modelling of the vehicles and hence previous model based traffic surveillance methods discussed are 'overkill' for any of the other items in the above list. The main objective is to extract, distinguish and classify trajectories of interest. We now investigate the extraction from image sequences of the first four "target" statistics listed above.

3 Real Time tracking

In this section we present the method used to track cars and extract motion parameters. Blake, *et. al.* presented a real-time tracker in [2, 3], and an extended version of this in [4]. We use a version of this tracker on a SUN IPX with a Datacell s2200 image capture board. The tracker is based on work originally done by Kass, *et. al.* [8] and uses a B-spline representation of a curve. By using a dynamical model, and a restricted search space, real-time performance has been demonstrated. The tracker details are summarised here for completeness.

3.1 The Tracker

The outline of the car is represented as a closed, quadratic B-spline curve $(x(s), y(s))^T$ with

$$x(s) = \mathbf{B}^T(s)\mathbf{X} \quad \text{and} \quad y(s) = \mathbf{B}^T(s)\mathbf{Y}, \quad 0 \le s \le N$$

where $\mathbf{B}$ is a vector of B-spline basis functions[1], and $\mathbf{X} = (X_1, ..., X_N)^T$, $\mathbf{Y} = (Y_1, ..., Y_N)^T$, are stacked vectors of the X and Y coordinates of the control points. For stability, the tracker requires a template, some average shape of the curve to be tracked. The template curve is also described by a B-spline with control points $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})^T$.

An underlying assumption is made that the tracked outline of the car is a roughly planar shape (or equivalently we have weak perspective viewing conditions), and that the curve $(x(s), y(s))$ transforms in an affine manner. Thus there are 6 degrees of freedom which describe the motion of the curve. In the tracker these are represented in a 6-dimensional sub-vector $\mathbf{Q}$ of the full configuration $(\mathbf{X}, \mathbf{Y})$, defined by matrices M, W:

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = W\mathbf{Q} \quad \text{and} \quad \mathbf{Q} = M \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}.$$

where M, W are defined in terms of the shape template $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$:

$$W = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \bar{\mathbf{X}} & \mathbf{0} & \mathbf{0} & \bar{\mathbf{Y}} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \bar{\mathbf{Y}} & \bar{\mathbf{X}} & \mathbf{0} \end{bmatrix}$$

and

$$M = \left[W^T \mathcal{H} W\right]^{-1} W^T \mathcal{H}$$

where $\mathcal{H}$ is the metric matrix converting measurements in control point space to euclidean measurements (see [3]), $\mathbf{0}$ and $\mathbf{1}$ are N-vectors of zeroes and ones, respectively – details are given in [3, 4] (where extensions of the motion model to non-affine and non-rigid motion are also given). In this notation we also have $W\bar{\mathbf{Q}} = (\bar{\mathbf{X}}, \bar{\mathbf{Y}})^T$ describing the template, which represents the template as $\bar{\mathbf{Q}} = (0, 0, 1, 1, 0, 0)^T$. The affine constraint is satisfied locally and does not restrict tracking. As shown later (figure 7), the scaling component of the tracker allows tracking of various sized vehicles, including trucks.

3.1.1 Tracker Dynamics

The dynamics of the tracker specify the change in the tracker relative to the template shape. The motion model employed is a second order dynamics model in each of the six degrees of freedom of planar motion. If we write the position and velocity for the curve in the state vector

$$\mathcal{X} = \begin{bmatrix} \mathbf{Q} \\ \dot{\mathbf{Q}} \end{bmatrix}$$

then the tracker models the dynamics of the object by the following discrete equation based on data sampled at video rate:

$$\mathcal{X}_{n+1} - \bar{\mathcal{X}} = A(\mathcal{X}_n - \bar{\mathcal{X}}) + \begin{bmatrix} \mathbf{0} \\ B\mathbf{w}_n \end{bmatrix}$$

where $\bar{\mathcal{X}} = (\bar{\mathbf{Q}}, \mathbf{0})^T$ acts as a driving term to stabilise the second order system around this curve shape (i.e. the template). The time interval between the states is video rate ($\Delta = 0.02$ seconds) and the system matrix:

$$A = \begin{bmatrix} I_6 & I_6\Delta \\ A_0 & A_1 \end{bmatrix},$$

defines the deterministic part of the dynamics (modes, natural frequencies and damping constants). Matrix B specifies the coupling of the standard normal random variables $\mathbf{w}_n$ into the dynamics and I_6 is the 6x6 identity matrix.

When locked to a feature, the tracking algorithm uses a steady state approximation to a Kalman filter based on the above dynamic system equation.

In order to track rapidly moving objects, such as cars, the system model (ie. the matrix, A) must accurately model the expected motions of objects to be tracked. This system model can be defined *a priori* or learned [4]. Once tracking, the extracted data can be used to tune the system model (using system identification techniques [4]). This produces an accurate model of the dynamics for a particular set of motions. As we are observing traffic, moving basically in the same pattern, the tuned system model yields accurate estimates of object motion between frames.

In this paper, the B-spline used to track cars has four control points, each with multiplicity two (C^0 continuity only).

3.1.2 Feature Measurement

The measurement process consists of casting rays simultaneously along several normals $\mathbf{n}(s)$ to the estimated curve as described. Along each normal, grey-level values in the video-input frame store are read to measure the relative position of target features. In [2, 3], areas of high contrast were used as features. In our implementation, we use a more predictive scheme[18]. This uses a 1D edge detector to highlight potential edges, and then weights these by a gaussian centred on the predicted edge location. The strongest weighted edge is then chosen as the feature. The search scale along each normal can be varied. We used a scale of 10 pixels in the experiments presented here.

3.1.3 Automatic Tracker Operation

In order to automatically initialise the tracker, a default position is associated with the template (this is usually in the centre of one of the lanes of the road). Before lock, the template is held stationary at the default position, and a local image differencing technique used to spot a moving object (a car) and provide an estimate of its initial position. Once the moving target is fully within the local acquisition region, the full dynamic tracker described above is started. Figure 1 demonstrates a typical sequence.

The region technique described later in section 6 is also used to detect when the target has left the region where it can be usefully tracked (i.e. it is approaching the horizon) and then trigger a re-initialisation of the tracker ready to capture another target.

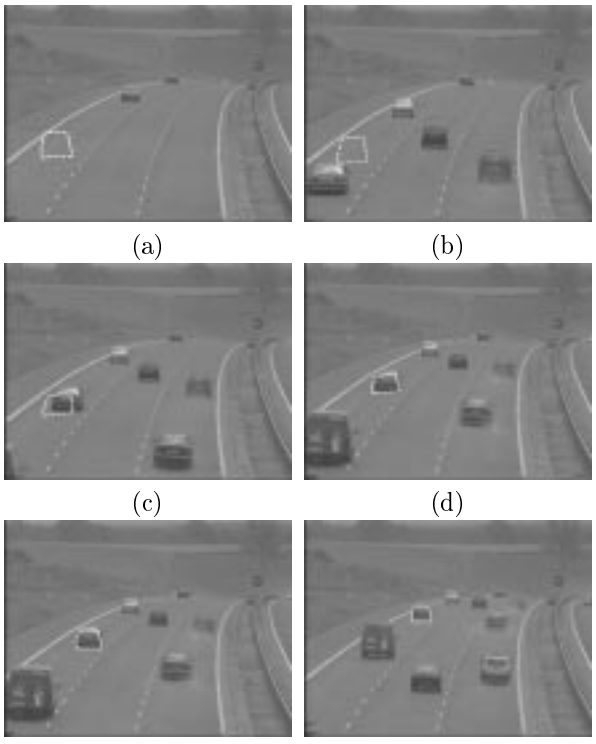
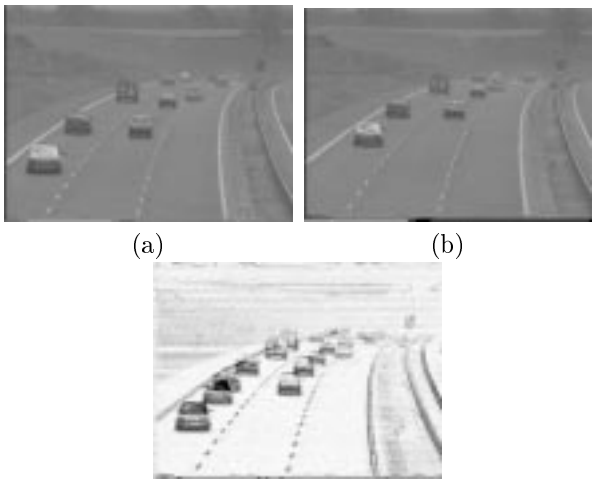


Figure 1: (a) The template waits for motion to trigger tracking. (b) As a vehicle enters the tracker locale, motion will be detected. (c) Detected motion initialises tracking, and (d) the tracker follows the vehicle, (e) settling on the vehicle boundary. (f) The tracker is locked through the calibrated region where speed is computed

3.2 Using motion information in the Tracker

Previous work [9, 12] uses motion information (under the static camera assumption) to locate and track vehicles where image differencing (or background subtraction) is used to segment the vehicle locations from the background. Our experience has lead us to believe that a camera fixed to an over-pass or pole will be subject to vibration (mostly due to the heavy traffic and wind). While devices could be designed to damp this vibration, the portability and economics of the system may be compromised. In figure 2(a-b) we show two images in a sequence taken from an over-pass. There was no traffic on the over-pass during this filming. In figure 2(c) we show result of image differencing. Observe that much of the background is highlighted (in addition to the vehicle locations). Hence image differencing or background subtraction schemes are going

to have noise caused by the vibration of camera. Our tracker is resilient against such vibrations because (1) the template mechanism [3] along with local search does not measure motion in other (non-local) regions of the image (2) the dynamic model absorbs the vibration into the B term as noise.



(a) (b)

(c)

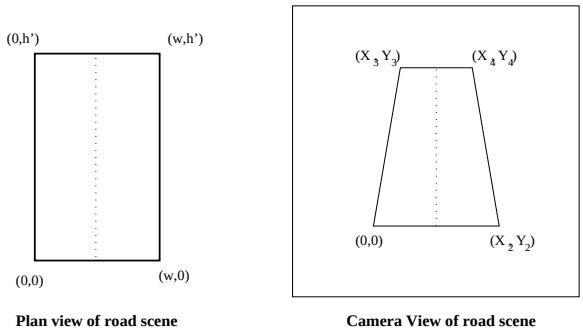
Figure 2: (a,b) Two views of a roadway taken two seconds apart. (c)Image difference between the two views. Notice that vibration of the camera does not allow the background to be completely subtracted.

4 Obtaining Metric Information

The tracker (specifically control points of the B-spline) provide the image position and velocity of the car. However, we seek metric information in the "world", such as vehicle speeds and changes in speed with respect to a fixed position. The vehicle is assumed to move on the ground plane. In order to obtain metric information, some sort of calibration must be performed. As we are dealing with both motion on a plane and sensing on a plane, we exploit the projective relationship between planes in lieu of complex calibration procedures [6]. A similar calibration has been used in [17].

4.1 Camera to Road Coordinate transformation

We have world points P_1, P_2, P_3, P_4 with coordinates $(0,0)$, $(w,0)$, $(0,h')$ and (w,h') respectively. These correspond to image points $(0,0)$, (X_2,Y_2), (X_3,Y_3) and (X_4,Y_4) as depicted in figure 3. (where, without loss of generality, we can chose the origin of the image coordinate system, here we chose it to coincide with a corner of the road 'box').



(a)

(b)

Figure 3: Plan view of the road scene showing world location of calibration points and their projections onto the image plane. Road scene and points used for calibration procedure. In this instance we have the image points in pixels P_1 = (50,293), P_2 = (569, 320), P_3 = (222, 228), P_4 = (586, 250), and the world lengths $(w,h') = (16,100)$ (yards) which gives the parameters for the projectivity $(a,b,d,e,g,h) = (34.43, 2.45, -1.79, 0.92, 0.0038, 0.0042)$.

The image and world points are related by an equation of the form:

$$\begin{bmatrix} X_n \\ Y_n \\ 1 \end{bmatrix} \sim \begin{bmatrix} a & b & 0 \\ d & e & 0 \\ g & h & 1 \end{bmatrix} \cdot \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \qquad (1)$$

where $\sim$ means *equals* in a homogeneous sense. Using the four calibration points the plane to plane correspondence between the road and the image plane can be determined. The camera to world transform is the inverse of this matrix. We can obtain the image of such a 'box' by using markers along the motorway at known distances (as exist along British motorways)

and knowledge of the road width. In figure 3 we show the image of a road scene with the extracted calibration points.

Once the calibration has been performed, the transformation allows us to compute the average velocity of a vehicle using image displacement and frame count. In figure 4 we see two instances in a sequence of a vehicle being tracked. The initial frame occurred at 790.98 seconds in the sequence and the end frame occurred at 794.82 seconds. The distance travelled during this time was 163.57 yards, hence the vehicle is moving with the approximate speed of 87.13 miles/hour. Without ground truth information (car speedometers are notoriously inaccurate and the police were unwilling to lend us a radar gun), we cannot confirm our measurements although based on human observation, they are reasonable and the average speeds for vehicles on numerous trials are repeatable.

(a) (b)

Figure 4: (a) vehicle at time 790.98 seconds; (b)vehicle at time 794.82 seconds The computed speed is approximately 87 mph.

5 Qualitative Information

The position/velocity of the tracker converts to metric information about the vehicle. In addition, the projective mapping can be used to locate regions in the image corresponding to regions in the world, say traffic lanes. We can infer traffic maneuvers from position information. In the results presented in the following section, position information is used to determine which lane the tracked vehicle is travelling in. Regions 1, 2 and 3 corresponds to the left hand lane, the center lane and the right hand lane, respectively. In the UK we expect speed to be *lane dependent* and hence the outside lane (3) to be faster.

Incident detection can also be obtained by observing changes in vehicle speeds. As an example, we have a sequence featuring a police car. In figure 5 we see two frames from this sequence. We observed that the speeds of vehicles are slower, presumably as the drivers notice the presence of the police car, than previously measured within this sequence. The car tracking in figure 5(b) entered the tracking region at 249.9s, left at 254.62s, travelled approximately 163 yards and hence was travelling at 71 mph, slower than vehicles tracked later in lane number 1 (see table 1).

(a) (b)

Figure 5: Incident detection? The computed speeds of cars is noticably slower in the presence of the police car (outlined in (a)).

6 Experiments

The tracker has been tested extensively. We filmed various roadways around Oxford from over-passes and used the video as input to the image capture board (still working at frame rate). Each new view of a traffic scene (a new camera position) required tuning the tracker dynamics and determining the new projective relationship. As described in [4], the system matrix, A (refer to section 3.1.1) can be determined using system identification techniques. This means for each stretch of road observed, the system dynamics can be tuned to accomodate typical traffic motion patterns. Regions (quadrilaterals) within each view are defined corresponding to areas of interest for the surveillance. In our experiments the regions correspond to lanes of the motorway. Thus tracker position gives "lane" information. The average speed of the vehicle while moving through the area of interest is computed using the calibration information described in section 4.

The template mechanism offers robustness to occlusion. Objects far away present difficulty as the search scale is large enough that other features 'distract' the tracker. We employ a constant scale in all tracking but for views of traffic such as the ones presented here, dynamic adjustment of the scale proportional to the area of the template could reduce this problem. Figure 6 shows the tracker operating successfully with occlusions.

Figure 6: (a)Tracking with partial occlusion is handled, even with a fair amount of overlap (b). (c-d) With a fixed search scale, smaller objects create problems and tracking fails.

Sample vehicle speeds obtained during a typical run are given in table 1. Although we did not have ground truth speeds, the results obtained fit with the apparent speeds of the vehicles.

The different sizes of vehicles can be accomodated due to the affine variability permitted by the tracker. Small trucks are simply a 'scaled' version of the template. Larger trucks require a separate template.

Figure 7: Tracking trucks: the affine variability allows the tracker to adjust for larger vehicles

7 Discussion and Extensions

The strength of the proposed method first and foremost is speed. We have demonstrated that the extraction of traffic statistics in real-time is possible with quite modest hardware. The system is robust to small vibrations of the camera as it does not require precise calibration and/or image differencing (except to initialise the tracker in a well localised area). At present, we can compute the speed of a vehicle, and its lane. From this other inferences can be made. Information on maneuvering and other vehicle activity can be extracted from this information using a system such as [14, 15] (see also [13]) and is the subject of further work.

region (lane)	start (seconds)	exit (seconds)	travelled (yards)	speed (mph)	av spd (mph)
1	269.28	273.96	132	58	
1	275.92	279.72	127	68	
1	297.86	301.56	129	72	
1	303.96	308.40	130	60	68
1	314.12	317.24	133	87	
1	321.76	325.24	126	74	
1	330.20	334.04	132	70	
1	343.16	347.58	123	57	
2	687.38	692.18	158	67	
2	708.46	712.36	164	86	
2	727.26	731.20	155	80	76
2	733.12	737.72	164	73	
2	749.12	753.64	169	77	
3	506.78	510.66	156	83	79
3	513.04	517.04	148	75	

Table 1: Statistics summary

An interesting aspect of this tracking system is that it combines projective distortions at the global scale (to compute speeds and locate regions) and affine distortions at the local scale.

The method presented makes a few assumptions and obviously the system will fail when those assumptions are violated. The weak-perspective viewing conditions are easily satisfied if the camera is far enough above the road being viewed. The "calibration" used exploited the existence of road markers every 100 yards. For locations without existing markers, markers can easily be placed in order to calibrate the system.

At present the tracker tracks only individual vehicles. The tracker must be able to run in parallel (for more than one vehicle on the road). Work in progress [5] has already achieved success at running multiple trackers simultaneously which we hope to incorporate in this project. Preliminary experiments indicate that the current tracker could support the tracking of at least 6 independent objects at 25 Hz.

Another possible extension would include the use of either thermal cameras or IR cameras for viewing in unfavourable conditions (such as at night). The CCD camera we employ is sensitive enough to cope with a hazy, grey English day. A superior camera would overcome viewing difficulties at night. Car lights are the most visable features of cars at night, and hence a different system could be designed to work using lights rather than occluding contours to track vehicles at night.

Acknowledgements

We thank D. Reynard and I. Reid for helpful discussions. The first author acknowledges the financial support of the Lady Wolfson Research Fellowship from St. Hugh's College, Oxford. The second author acknowledges the financial support of the EPSRC.

References

[1] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.

[2] A. Blake, R. Curwen, and A. Zisserman. Affine invariant contour tracking with automatic control of spatil-temporal scale. In *Proc. 4th Int. Conf. on Computer Vision*, 1993.

[3] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. Journal of Computer Vision*, 1993.

[4] A. Blake and M. Isard. 3D position, attitude and shape input using video tracking of hands and lips. In *Siggraph*, 1994.

[5] A. Cox. Multiple dynamic contours. Master's thesis, University of Oxford, 1994. (to appear).

[6] O. Faugeras. *Three Dimensional Vision*. MIT Press, Cambridge, MA, 1994.

[7] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[8] M. Kass, A. Witkin, and D. Terzopoulos. Snakes:active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pages 259–268, 1987.

[9] M. Kilger. Video-based traffic monitoring. In *IEE 4th International Conference on Image Processing and its applications*, 1992.

[10] D. Koller, K. Daniilidis, T. Thòrhallson, and H.-H. Nagel. Model-based object tracking in traffic scenes. In *Proceedings of the European Conference on Computer Vision*, pages 437–452, Santa Margherita, Italy, May 1992. Springer-Verlag.

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

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

[13] H. Kollnig, H-H. Nagel, and M. Otte. Association of motion verbs with vehicle movements extracted from dense optical flow fields. In *Proc. European Conf. on Computer Vision*, pages 338–347, Stockholm, May 1994.

[14] M. Mohnhaupt and B. Neumann. On the use of motion concepts for top-down control in traffic scenes. In *BMVC*, pages 598–600, Berlin, 1990. Springer-Verlag.

[15] M. Mohnhaupt and B. Neumann. Understanding object motion: Recognition, learning and spatiotemporal reasoning. *Robotics and Autonomous Systems*, 8:65–91, 1991.

[16] L.J. Mountain and J. B. Garner. Application of photography to traffic surveys. In *The Journal of the Institution of Highway Engineers*, pages 12–19, November 1980.

[17] I. D. Reid, D. W. Murray, and K. J. Bradshaw. Towards active exploration of static and dynamic scene geometry. In *IEEE Int'l Conference on Robotics and Automation*, San Diego CA, 1994.

[18] S. Rowe. *People tracking using Adaptive and Dynamic Contours*. PhD thesis, University of Oxford, 1995. (to appear).

[19] M. Schwarzinger, T. Zielke, D. Noll, M. Brauckmann, and W. von Seelen. Vision-based car-following: Detection, tracking and identification. In *Proceedings of the Intelligent Vehicles Conference*, Detroit, 1992.

[20] G.D. Sullivan. Visual interpretation of known objects in constrained scenes. *Phil. Trans. R. Soc. Lond. B.*, B(337):109–118, 1992.

[21] T.N. Tan, G. D. Sullivan, and K.D. Baker. Structure from constrained motion using point correspondences. In *Proc. BMVC*, pages 301–309, University of Glasgow, September 1991. Springer-Verlag.

[22] T.N. Tan, G. D. Sullivan, and K.D. Baker. Structure from motion using ground plane constraint. In G. Sandini, editor, *ECCV*, pages 277–281, Santa Margherita, Italy, May 1992. Springer-Verlag.

[23] T.N. Tan, G. D. Sullivan, and K.D. Baker. Structure from motion using ground plane constraint. In *BMVC*, pages 69–78, Leeds, September 1992. Springer-Verlag.

[24] R.C. Waterfall and K. W. Dickinson. Image processing applied to traffic. In *Traffic Engineering and Control*, pages 60–67, February 1984.

[25] H.J. Wootton and R. J. Potter. Video recorders, micro computers and new survey techniques. In *Traffic Engineering and Control*, .

[26] A.D. Worrall, R.F. Marslin, G.D. Sullivan, and K.D. Baker. Model-based tracking. In *Proc. BMVC*, pages 310–318, University of Glasgow, September 1991. Springer-Verlag.

A Vibrational Noise

In this section we justify claims from section 3.2 that the vibrational noise is absorbed in the B matrix in the dynamic model (introduced in section 3.1.1). We have amplified the magnitude of the vibration to demonstrate the effect. The data presented are in the affine sub-space and so have 6 components, the first two of which are the translational components.

The B matrix with *no noise*, $\mathbf{B}_{NoNoise} =$

$$\begin{bmatrix} 0.051 & -0.029 & 0.058 & -0.099 & 0.067 & 0.004 \\ -0.029 & 0.069 & 0.032 & 0.328 & -0.292 & -0.029 \\ 0.058 & 0.032 & 17.500 & 3.253 & -1.494 & 3.720 \\ -0.099 & 0.328 & 3.253 & 25.605 & -1.497 & -1.835 \\ 0.067 & -0.292 & -1.494 & -1.497 & 10.713 & 1.476 \\ 0.004 & -0.029 & 3.720 & -1.835 & 1.476 & 5.216 \end{bmatrix}$$

the B matrix with *noise in X and Y*, $\mathbf{B}_{Noise_{xy}} =$

$$\begin{bmatrix} 4.6428 & -0.8585 & -0.6151 & -0.6334 & 0.2725 & 0.6408 \\ -0.8585 & 4.0344 & 0.8840 & 0.8876 & -0.07340 & -0.2046 \\ -0.6151 & 0.8840 & 17.9260 & 3.4113 & -1.3759 & 3.6253 \\ -0.6334 & 0.8876 & 3.4113 & 25.7838 & -1.2327 & -2.0171 \\ 0.2725 & -0.0734 & -1.3759 & -1.2327 & 10.9340 & 1.4505 \\ 0.6408 & -0.2046 & 3.6253 & -2.0171 & 1.4505 & 5.2479 \end{bmatrix}$$

and the B matrix with *noise in Y*, $\mathbf{B}_{Noise_y} =$

$$\begin{bmatrix} 0.0525 & -0.0240 & 0.0222 & -0.1181 & 0.0505 & -0.0005 \\ -0.0240 & 3.8598 & 0.3542 & 0.4779 & 0.1876 & 0.0969 \\ 0.0222 & 0.3542 & 18.0891 & 3.4682 & -1.3862 & 3.7881 \\ -0.1181 & 0.4779 & 3.4682 & 25.9320 & -1.2859 & -1.9113 \\ 0.0505 & 0.1876 & -1.3862 & -1.2859 & 10.7930 & 1.4034 \\ -0.0005 & 0.0969 & 3.7881 & -1.9113 & 1.4034 & 5.2536 \end{bmatrix}$$

To show how large the changes are, the B matrix generated from the 'noise free' data was subtracted from the noisy B matrices, the results of this subtraction were then normalised by dividing each element by the corresponding element in the noise free B matrix. We obtain the normalised noise in both the X and Y components: $\mathbf{abs}\left([\mathbf{B}_{Noise_{xy}} - \mathbf{B}_{NoNoise}]/[\mathbf{B}_{NoNoise}]\right) =$

$$\begin{bmatrix} \mathbf{90.1436} & \mathbf{29.0733} & \mathbf{11.6751} & \mathbf{5.4273} & \mathbf{3.0907} & \mathbf{174.3460} \\ \mathbf{29.0733} & \mathbf{57.7481} & \mathbf{26.7331} & \mathbf{1.7029} & \mathbf{0.7482} & \mathbf{6.1448} \\ \mathbf{11.6751} & \mathbf{26.7331} & 0.0243 & 0.0488 & 0.0788 & 0.0253 \\ \mathbf{5.4273} & \mathbf{1.7029} & 0.0488 & 0.0069 & 0.1767 & 0.0990 \\ \mathbf{3.0907} & \mathbf{0.7482} & 0.0788 & 0.1767 & 0.0205 & 0.0174 \\ \mathbf{174.3460} & \mathbf{6.1448} & 0.0253 & 0.0990 & 0.0174 & 0.0061 \end{bmatrix}$$

and the normalised noise just in the Y component: $\mathbf{abs}\left([\mathbf{B}_{Noise_y} - \mathbf{B}_{NoNoise}]/[\mathbf{B}_{NoNoise}]\right) =$

$$\begin{bmatrix} 0.0315 & 0.1591 & 0.6134 & 0.1987 & 0.2419 & \mathbf{1.1319} \\ 0.1591 & \mathbf{55.2051} & \mathbf{10.1123} & 0.4554 & \mathbf{1.6434} & \mathbf{4.3840} \\ 0.6134 & \mathbf{10.1123} & 0.0336 & 0.0663 & 0.0719 & 0.0184 \\ 0.1987 & 0.4554 & 0.0663 & 0.0127 & 0.1412 & 0.0413 \\ 0.2419 & \mathbf{1.6434} & 0.0719 & 0.1412 & 0.00745 & 0.0493 \\ \mathbf{1.1319} & \mathbf{4.3840} & 0.0184 & 0.0413 & 0.0493 & 0.0071 \end{bmatrix}$$

Note that the significantly large elements in these matrices correspond to rows/columns to do with the appropriate translational degrees of freedom to which the noise has been added.