Model-based vehicle detection and classification using orthographic approximations*

G D Sullivan, K D Baker, A D Worrall, C I Attwood and P R Remagnino
Department of Computer Science,
The University of Reading, Reading, UK. RG6 6AY
g.sullivan@reading.ac.uk

Abstract

This paper reports the current state of work to simplify our previous model-based methods for visual tracking of vehicles for use in a real-time system intended to provide continuous monitoring and classification of traffic from a fixed camera on a busy multi-lane motorway. The main constraints of the system design were:

(i) all low level processing to be carried out by low-cost auxiliary hardware,
(ii) all 3-D reasoning to be carried out automatically off-line, at set-up time.

The system developed uses three main stages: (i) pose and model hypothesis using 1-D templates, (ii) hypothesis tracking, and (iii) hypothesis verification, using 2-D templates. Stages (i) & (iii) have radically different computing performance and computational costs, and need to be carefully balanced for efficiency. Together, they provide an effective way to locate, track and classify vehicles.

1 Introduction

Over many years we have been developing model-based methods for detecting, locating, tracking and classifying vehicles confined to move on a ground plane (of known position with respect to the camera). Our efforts have mainly been directed at developing a flexible system able to deal with arbitrary views of vehicles which are then tracked through long sequences of images [1][2]. This has proved to be a challenging task for the study of 3-D vision [8].

The work reported here takes a different tack. Many practical applications of vision for monitoring traffic need far less sophisticated analysis. Typical motorway surveillance systems generate images (e.g. Figure 1) from static cameras looking at highly predictable traffic flow. A major objective is to develop automatic means to detect, determine the speed, and classify vehicles as they pass a predetermined point. Most approaches to vehicle tracking (e.g. [3][4][5][6]) take a strictly 2-D approach in which

* The work reported here was carried out under an EPSRC & Dept of Transport LINK project (TIO-68; GR/H/90278) awarded to the University of Reading and Computer Recognition Systems Ltd. (1993-6).
some image attribute, such as the convex hull of regions of change, or the inter-image correlation of the region itself, is tracked. These techniques cope poorly with multiple overlapping vehicles, or with shadows and rain, since the image attributes used fail to distinguish a vehicle’s structure from other signals. This problem can be greatly reduced by using 3-D model-based methods, but at a far higher cost.

We report here work carried out to simplify our 3-D algorithms to suit constrained view applications. In brief, we make use of 3-D geometrical knowledge encapsulated in models to condition a series of algorithms which operate in small regions of the image, under the assumption of locally orthographic projection.

One algorithm pre-computes sparse 2-D templates according to the expected pose of each of the models known to the system. The template is used to predict edge evidence (essentially as in our previous methods, see [7][8] for overviews), and the template in best agreement with the image data is deemed to identify the class of vehicle. Each template typically contains about 1,000 points and needs to be evaluated at all locations covering some image region. The approach is linear on the number of models, with at least 20 models needed in practice. At frame rate (25 Hz), this represents a computational load that is not yet feasible using low-cost hardware (see Section 6).

A second algorithm has therefore been developed where the per-model cost is much lower, and which is able to work efficiently over fairly wide areas of the image. This algorithm searches for simultaneous maxima in a set of 3 separate 1-D correlations. Since it takes no notice of 2-D structure it is less discriminating than the 2-D template method. However, its merit is that it allows initial hypotheses about likely vehicle interpretations (occurring over a fairly large region of the image) to be generated relatively cheaply. These hypotheses are subsequently tracked over several frames to detect consistent runs
of high scores. A successful track can then accurately predict the frame in which the hypothesised vehicle will pass a “trap” already prepared for use by the 2-D sparse template-matching algorithm.

The initial hypothesis generation stage greatly reduces the cost of the 2-D algorithm, since the likely class of the vehicle has been identified and there is relatively little uncertainty about its location in the image (although there may be several such hypotheses to consider in one frame). Yet further saving in real-time computing requirements can also be made by exploiting the fact that, when predicted by the 1-D algorithms, frames only rarely need to be considered for 2-D analysis. Thus by buffering the selected region (and moment) of interest, the more expensive 2-D matching stage need not be carried out synchronously in real-time at frame rate.

Both algorithms are heavily based on work that we have published previously (see [7][9][10]), but with many modifications, as explained below. We take this opportunity to describe the overall system, and to report experiments which demonstrate the benefits of co-operation between the 1-D & 2-D model-matching algorithms in long sequences of video film of a busy motorway scene.

2 Static camera assumptions

The traffic analysis algorithms reported here are specifically tailored for use with static cameras, calibrated with respect to the ground plane in advance, operating in scenes in which vehicles are expected to follow fixed routes. These preconditions are true in most existing applications of motorway surveillance systems, although there is an increasing use of variable pan-tilt-zoom cameras. Methods for automatically updating the camera calibration parameters can easily be envisaged, but are not addressed here. In the experiments reported below, the camera was calibrated directly from the video tape sequences, using the Interactive Calibration Tool described in [9]. This provides sufficient accuracy for our purposes without requiring a survey of the site, or any other information about the camera and viewing conditions.

Traffic is therefore expected to flow (approximately) along fixed directions in the image, which map in a known way onto the 3-D ground plane. The central novelty of the algorithms is that we will make use of (now, conventional) 3-D model based methods to compute the expected appearance of different vehicles, in order to compile patterns of lower dimension for matching with image data. In fact for each model, and for each traffic lane, we compute appearance in two different positions, as illustrated in Figure 1. The first position (solid box, further from the camera, for approaching traffic) is used by the hypothesis generation algorithm; this is fixed in position. The second (dashed box) is used by the verification algorithm, and can move laterally. The two algorithms assume (different) locally orthographic projections.

Both algorithms are designed to use image gradient data as their basic evidence, which is estimated using vertical and horizontal 3*3 Sobel filters. In practice this is done in an image preprocessing stage in real-time using dedicated hardware. Therefore the cost of computing the gradient does not need to be considered when comparing the performance of the two algorithms.
3 Hypothesis generation

The purpose of the hypothesis generation stage is to search close to the first trap position for likely vehicle hypotheses (i.e. class and approximate pose) in individual video frames. The main design goals are: (i) as wide as possible coverage in the image, (ii) minimal per-model cost, (iii) useful discrimination performance, and (iv) low miss rates. The algorithm works as follows.

Each model is represented in an object-centred coordinate system, in which \((x,y,z)\) correspond to the sideward, forward, and upward directions respectively. For each model we construct a wire-frame representation (with hidden line removal) of the model positioned on the ground plane at the centre of the trap (Figure 1, solid box). The object coordinate frame defines three directions (in the image) which correspond to directions in which edges are most common. [This is particularly true in the \(x\) & \(y\) directions, due to the strong rectilinear structures of vehicles parallel to the ground, but less so in \(z\), since upright features of vehicles are more variable.] We construct three lines orthogonal to these image directions and project the lines of the instantiated model onto these, to give the 1-D model templates labelled \(m\) in Figure 2. All this is carried out off-line, once the camera calibration is established.

A region of interest (ROI) is identified in the image (thick solid box, Figure 1) which contains the entire projected model when it is centred anywhere in the trap position (thin solid box). At run time, we first form the dot products of the gradient image in the ROI with the directions of the object axes, and collapse them (as histograms) onto the

\[m_x(u)\]
\[m_y(u)\]
\[m_z(u)\]
\[s_x(u)\]
\[s_y(u)\]
\[s_z(u)\]
\[c_x(u)\]
\[c_y(u)\]
\[c_z(u)\]

Figure 2 Hypothesis generation, using 1-D templates projected along the three vehicle-centred axes. The triples of distributions show: model projections \((m)\), data profiles \((s)\) and their correlations \((c)\). Black lines show the main peaks of \(c\) back-projected along the axes.

projection lines, giving three 1-D data profiles labelled s in Figure 2. Note that since all models share the same expected orientation, this stage is done only once per frame.

Model matching is carried out by correlating the data profiles with each set of model templates, to create distributions such as those shown as c in Figure 2; this process is greatly speeded up by considering only the most significant points in each 1-D template. These correlation distributions tend to show marked peaks. For each model, the best peak is identified in each 1-D correlation, together with all other peaks within 60% of its value. These are all back-projected across the ROI (dark lines in Figure 2, centre). This determines a set of combinations of 1-D peaks, each forming a triangle in the image. We accept for further consideration only those combinations where the triangle is small (the radius of the inscribed circle is less than 1.8 pixels) and then identify the combination for which the product of the three 1-D correlations is greatest. If this correlation product exceeds a fixed threshold, then a hypothesis is generated.

Figure 2 (centre) shows the selected intersection (heavy black lines). The intersection point determines the position of the object coordinate system on the ground plane, and it is then easy to instantiate the model at that position (white overlay).

Figure 3  Strongest hypotheses (for one model), shown as position in the image in the direction of travel (ordinate), against frame number (abscissa). Note how “aliases” have short tracks, and become superseded by the “true” hypotheses (grey bar) - see text.

![Graph showing spatio-temporal conflict](image.png)
Each model therefore generates up to one hypothesis per frame (per ROI). The purpose of the hypothesis tracking stage is to identify sequences of hypotheses that are consistent with one vehicle moving (approximately) in a straight line at constant velocity. The main problem to be overcome is illustrated in Figure 3. As a vehicle begins to intrude into the ROI, the best current hypothesis usually represents an alias, where (e.g.) the windscreen of the model becomes mismatched with the bonnet of the car (Figure 3(a)). This alias may be successfully tracked for a few frames as more of the car enters the ROI, until another alias is detected, having a better correlation value (Figure 3(b)). This may happen several times before the true match is made, which (with luck) will continue to obtain the highest correlation score, all the way up to the point when the car begins to leave the ROI (Figure 3(c)). Thereafter a series of trailing aliases (Figure 3(d)) are likely to provide the highest correlation scores.

The task is to identify the long sequences which distinguish the true matches from the aliases. The task is made harder by occasional failures (misses, or false positives) in the hypothesis generation stage, possibly due to low contrast vehicles, or the presence of multiple vehicles following each other very closely. The latter mainly occurs at low traffic velocities, where the number of frames contributing to a track increases substantially; fortunately, the increased observations allow more stringent testing.
The tracking algorithm we have developed is essentially a 3-state automaton. For each model the system maintains one of three states: (A) no current track; (B) a track is evolving; (C) a track has been accepted. Transitions between states occur as illustrated (in a somewhat simplified form) in Figure 4. The track quality score (q) is based on a combination of the length of a track, and the average strength of the correlation scores obtained; the failure count (n) allows isolated failures to be tolerated.

Any track which meets the criterion of acceptance (q>thd) represents a candidate for a hypothesis. The track history is then used to estimate the position, velocity, and acceleration (in the image) of the vehicle. A further stage of track conflict resolution then occurs, based on spatio-temporal overlap between hypothesised vehicles (see Figure 3). Each tracked hypothesis (grey trace) identifies a period of time in which no other hypothesis can exist at that position in space (thin lines parallel to the grey trace). If such a conflict is detected, then the track with the highest quality score wins.

Surviving tracks can then easily be propagated forward to determine: (a) the future frame in which the vehicle will pass the second trap, and (b) where (laterally) in the image it will then appear. These form explicit predictions, which are posted in a queue (in frame order) to be managed by the Hypothesis Verification stage.

5 Hypothesis verification

When the frame corresponding to a prediction is encountered, the system buffers the gradient data for a region of interest around the predicted position (outer dashed box, Figure 2). This is then matched against a 2-D sparse template for the particular model of the hypothesis.

The 2-D templates are precomputed, off-line. Given the position of the trap point, each model is instantiated into the discrete image in the form of a wire-frame model (with hidden line removal). Each visible “wire” defines a set of image addresses which are associated with the direction (in the image) perpendicular to the wire. The value associated with a vehicle hypothesis is computed by averaging at all selected points the gradient vectors projected in the direction of the appropriate perpendicular. To decrease the sensitivity of this measure to the exact fit between (orthographic) model and (perspective) image data, each sampled point is replicated on each side of its wire by two further points which contribute weights of 0.5 of the central point - i.e., conceptually, each wire is blurred to form a 3 pixel wide triangular taper.

Hypotheses are evaluated in this way at each location within a small 2-D region (inner dashed box, Fig 2) defined by the tracking process, simply by off-setting the list of pixel addresses (the orthogonal projection assumption). All interpretations predicted to apply to any given frame are considered (in practice this is rarely more than a few) and if the maximum score obtained exceeds a threshold, then that hypothesis is accepted.

6 Comparison of computational costs

The algorithms described in sections 3 and 5 have very different computational costs and benefits. In order to compare their performance, we consider a “typical” traffic monitoring task, obtained from a stretch of the M25 and illustrated in Figure 1. The purpose is to detect and classify all vehicles in each of the three lanes, as they leave the
We consider each lane separately, and avoid inter-field shatter (from the already installed interlaced camera) by sub-sampling the even fields only, to a resolution of 384*288 pixels.

Towards the front of the image, where the resolution is best, and the separation between vehicles in the image is good, one lane subtends approximately 100 (subsampled) pixels. All such positions must be covered by the algorithm. Vehicles travel at any speed up to (approximately) 160 k/h., i.e. about 1.8 m/frame. Near the front of the image 1.8m (longitudinally) represents about 40 pixels. Full coverage of one lane can therefore be expected by considering models centred in an image region of about 40*100 pixels (inner solid box, Figure 1).

The 2-D template algorithm (section 5) requires about 1,000 primitive operations (vector product and add) per model, per location in the image. The computational cost of covering the lane in this way by 20 different models would therefore be $8 \times 10^7$ operations, repeated at 25 Hz, i.e. approximately $2 \times 10^9$ operations per second. This is not feasible with low cost hardware.

The 1-D templates used in the hypothesis generation algorithm typically require about 10 non-zero points per template (see Section 3). Each 1-D template is correlated with a 1-D data profile comprised (in this case, where the traffic is viewed more-or-less head on) of approximately 100 (x & z) and 40 (y) points. The total cost per model is therefore $(100+100+40) \times 10 = 2.4 \times 10^3$, giving a cost for 20 models of $4.8 \times 10^4$ multiply and add operations. [To this must be added an overhead which is independent of the number of models, for assembling the data profiles over the whole ROI, roughly equivalent to $7,000 \times 3$ vector projections, and the small cost of searching for correlation peaks etc. In this application, these approximately double to cost to $9.6 \times 10^3$.]

In this illustrative application, the 1-D algorithm is therefore able to run on all frames (at 25 Hz) on hardware able to deliver about $2.4 \times 10^6$ multiply and add operations per second, three orders of magnitude lower that the 2-D algorithm. This IS feasible with

![Figure 5](image)  
**Figure 5**  
*Top:* Simple cases: Examples of recovered poses from each lane.  
*Bottom:* Awkward cases: Car classified despite canoe on roof rack, van wrongly classified as car, poorly fitting van, car recognised, despite heavy shadow.
modern DSP circuitry. The predictions generated by the 1-D algorithms greatly reduce the search required by the 2-D algorithm, typically to a few models only, over a very small region of image (inner dotted box, Figure 1), in occasional frames.

7 Performance

An early version of the system* has been demonstrated on a 1 min. video of motorway traffic (Figure 1), using three independent systems to cover each of the three lanes. In all some 46 cars (of many different styles) and 7 light vans appeared in the scene. A selection of poses recovered at the hypothesis generation stage is shown in Figure 5; note that both failures (middle two images, bottom row) were corrected during the hypothesis verification stage. The Table summarises the results obtained.

<table>
<thead>
<tr>
<th></th>
<th>Lane 1</th>
<th>Lane 2</th>
<th>Lane 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cars</td>
<td>Vans</td>
<td>Cars</td>
</tr>
<tr>
<td>Number Observed</td>
<td>20</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>Classified at Hyp Gen</td>
<td>20</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>Classified at Hyp Ver</td>
<td>20</td>
<td>2</td>
<td>19</td>
</tr>
</tbody>
</table>

Performance seems very good. It should be noted that the second row (“Detected at Hypothesis Generation Stage”) shows the number of time the correct class model gave the best quality score (of all the classes recognised and tracked); there are occasions when other classes prove to have the best response at the Hypothesis Verification Stage (third row). In all, only one vehicle was mis-classified - a van in the slow lane (lane 3). This failure was probably due to the limited number of models in use.

8 Conclusions

We have demonstrated a set of algorithms for detecting and classifying vehicles passing before a fixed camera in busy motorway scenes, which reap most of the benefits of 3-D model-based processing at low computational cost. The algorithms use precompiled low dimensional patterns which are matched to image data under the assumption of locally orthographic projection.

The system comprises three parts:

   (i) Hypothesis generation, using three 1-D templates to identify likely interpretations; this has low-order complexity in the number of models and the 2-D extent of the search; this algorithm is applied to each video frame.

   (ii) Hypothesis tracking, which filters the results of (i) to identify sequences of video frames consistent with a single vehicle class at a (near) constant speed.

   * The current system has a number of deficiencies: the set of models is incomplete (only 9 models are used); the hypothesis verification stage is not limited to the specific model class(es) identified in the tracking stage; the trap sizes and locations have not been arranged for optimal efficiency in the trade off between the two algorithms.
Hypothesis verification, which uses the tracks from (ii) to anticipate the frame number, and the approximate position in the frame, when the hypothesis is closest to a predetermined point. The local optimal response of a sparse 2-D, boundary based, evaluation function is then used to determine the exact position and to accept or reject the hypothesis. By buffering the local ROI, this stage need not run at frame rate.

In addition, we have:

(a) Shown that the system works well with video data collected under normal operating conditions.
(b) Analysed the computational complexity of the algorithms in (i) & (iii) to illustrate the major advantages of the multiple-stage process.

This work brings the possibility of model-based vehicle surveillance in real-time far closer to commercial practice.

9 References


