# ASSET-2:
# Real-Time Motion Segmentation and Object Tracking

Technical Report TR95SMS2b
(Shorter versions of this have now been published in PAMI [30] and ICCV95 [28])

1995

S.M. Smith
Oxford Centre for Functional Magnetic Resonance Imaging of the Brain (FMRIB),
Department of Clinical Neurology, Oxford University, Oxford, UK
(Previously in Computer Vision and Image Processing Group, DRA Chertsey, DERA, UK)
steve@fmrib.ox.ac.uk
www.fmrib.ox.ac.uk/~steve

**Abstract**

This paper describes how image sequences taken by a moving video camera may be processed to detect and track moving objects against a moving background in real-time. The motion segmentation and shape tracking system is known as ASSET-2 – A Scene Segmenter Establishing Tracking, Version 2. Motion is found by tracking image features, and segmentation is based on first-order (i.e., six parameter) flow fields. Shape tracking is performed using two dimensional radial map representations. The system runs in real-time, and is accurate and reliable. It requires no camera calibration and no knowledge of the camera's motion.

**Category:** Active and real-time vision, motion analysis

# 1 Introduction

This paper describes how image sequences taken by a moving video camera may be processed to detect and track moving objects against a moving background in real-time. The motion segmentation and shape tracking system is known as ASSET-2 – A Scene Segmenter Establishing Tracking, Version 2.

The work has been conducted using as a typical real life situation, a vehicle travelling along a road, taking video pictures of what is in front of it. If other vehicles can be seen, they must be "segmented" from the background of the rest of the scene so that their motion can be estimated and the necessary evasive action taken. This situation provides a testbed for the developed program which is quite general, and ASSET-2 has been tested on other types of sequence containing independent motion.

This paper gives a brief review of current image sequence segmentation methods followed by an overview of the ASSET-2 system. Next, short descriptions of each of the sections of ASSET-2 are given. Finally, results of using the ASSET-2 system are shown and discussed.

ASSET-2 supersedes ASSET [29], which matched features only from one frame to the next (as opposed to tracking them over many frames) and segmented the resulting flow vectors into clusters assuming constant flow within clusters (now improved with the allowance of linear flow variations).

# 2 Review of Past Research

A summary of previous work on segmentation of the image flow field is now given. Smith [26] gives a more complete review.

Spoerri and Ullman [31] and Waxman and Duncan [37] segment the image flow field into independently moving objects using local flow discontinuities, giving fairly inaccurate boundaries. Meygret and Thonnat [21], Thompson [32], Thompson and Pong [33], McLauchlan *et al.* [19] and Nelson [23] segment on the basis of simple flow clustering or inconsistency with the background flow. Adiv [1], Debrunner and Ahuja [11], Diehl [12], Burt *et al.* [10], Bergen *et al.* [4] and Torr and Murray [34] segment using analytic image transformations; fits are found within the flow field of analytic functions with a number of parameters. Murray and Buxton [22], Bouthemy and Lalande [8] and Black [5] use statistical regularization to perform globally optimal segmentation, utilizing various locally defined cost functionals. Verri *et al.* [35] and Rognone *et al.* [24] perform segmentation on the basis of global motion clustering. Previous work on segmentation has usually involved making restrictive assumptions about the world, the motion of the camera etc. ASSET-2 does not depend on this sort of assumption; it is shown later that good results can be obtained without doing so.

Very little work has been undertaken which involves the temporal integration, or tracking, of motion segmentation results. Irani *et al.* [16] use the segmentation methods of Burt, Bergen *et al.* with temporal integration of the segmentation results to improve performance. However, this "temporal integration" does *not* involve any sort of shape tracking or modelling, so that information about scene events is not readily available. Meyer and Bouthemy

[20] use the approach of Bouthemy and François [7] to achieve segmentation, and then track moving objects' outlines over time. The objects are matched over time using a polygonal representation. The motion estimation and segmentation stages are quite separate from the shape tracking stage, leading to a rather unintegrated approach. Results are only presented for a static camera.

The research described in this paper covers this area as well as the segmentation, that is, a temporally coherent list of segmented objects is maintained as time proceeds, and objects move about in the image.

# 3 ASSET-2 – Overview

The ASSET-2 system is built around feature-based image motion estimation. The features used primarily are two dimensional (often referred to as "corners"), and edges can be used to refine the results obtained by using two dimensional feature motion. The main part of ASSET-2 has solely two dimensional calculations and not three. This means that feature matching is performed in the image plane, as is object segmentation and tracking. The only area in the main body of ASSET-2 where three dimensional behaviour is acknowledged is in predicting which one of two touching objects will occlude which. The reason for staying in the image plane is that no more information about the world processes can be obtained by working in three dimensions than by staying in the original two. Unless interaction with outside processes and measurements is needed at an early stage in a vision system, there is no need to move to three dimensions until the final interpretation of the processed data is made.

A brief summary of ASSET-2 is now given, and can be seen graphically in Figure 1.

ASSET-2 is fed a stream of digitized video pictures, usually taken from a moving platform. For an example sequence see Figure 2, taken from ROVA (ROad Vehicle Autonomous, the autonomous vehicle at DRA Chertsey). In this sequence ROVA is travelling forwards and turning slowly to the left with the curve of the road. The ambulance is overtaking ROVA and the Landrover is travelling with almost identical speed to ROVA. At about frame 103 the ambulance passes behind the Landrover, and becomes occluded.

Each frame taken by the video camera is initially processed to find two dimensional features and edges. The two dimensional feature list is passed to a feature tracker which uses a two dimensional motion model to match and track features over as many frames as possible. A two dimensional vector field can then be created by taking either feature velocities or displacements over a fixed number of frames. The resulting vector field is passed to a flow segmenter which splits the list of flow vectors into clusters which have similar flow within them and are different to each other, using the assumption of first order flow variation.

Next, this cluster list is compared with a temporally filtered list of clusters, and the filtered list is updated using the newly found clusters. This gives good results as it stands, but the boundaries of the clusters in the filtered list can be further refined by using image edges, if available.

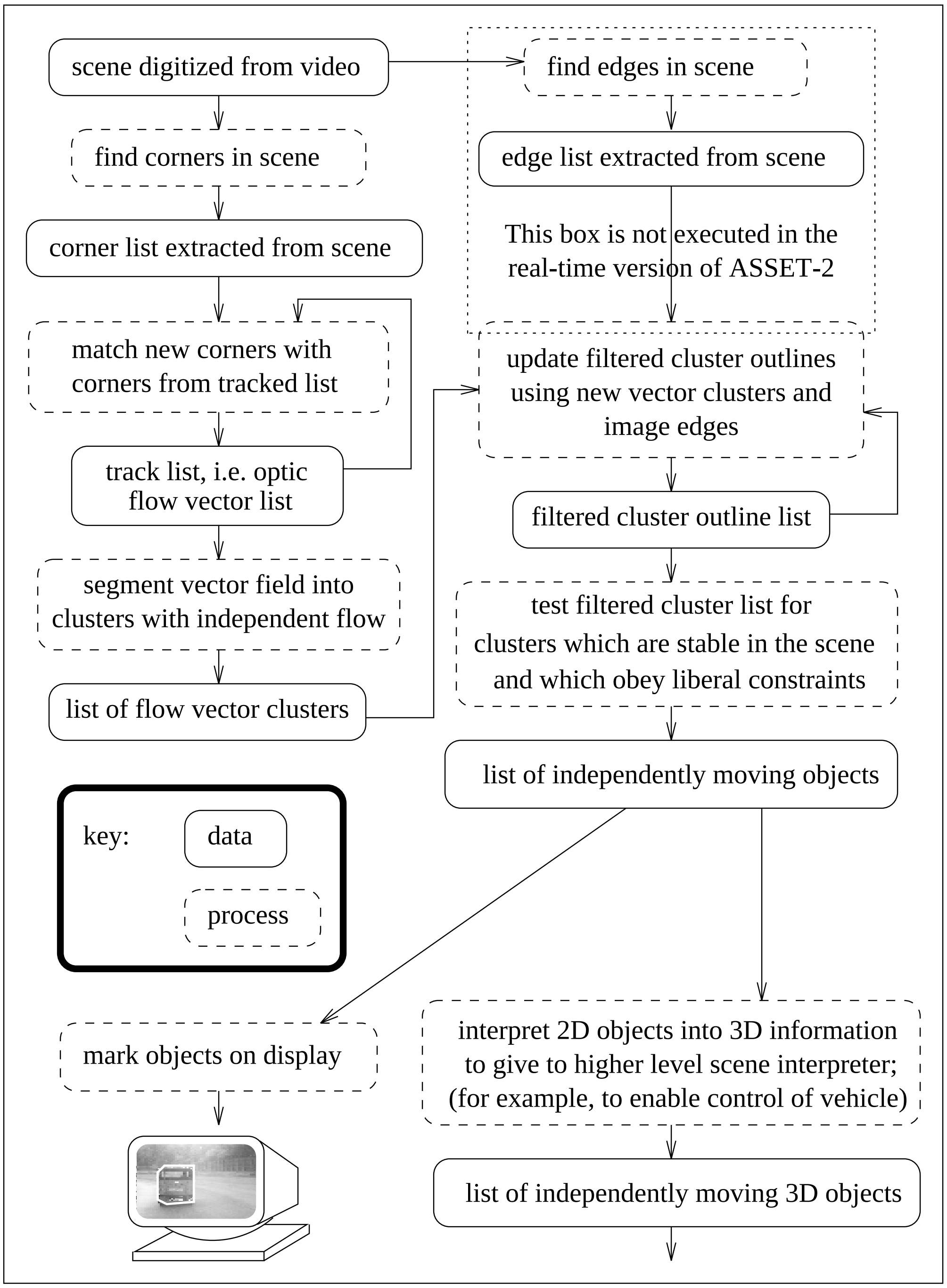


Figure 1: Overview of the ASSET-2 system.

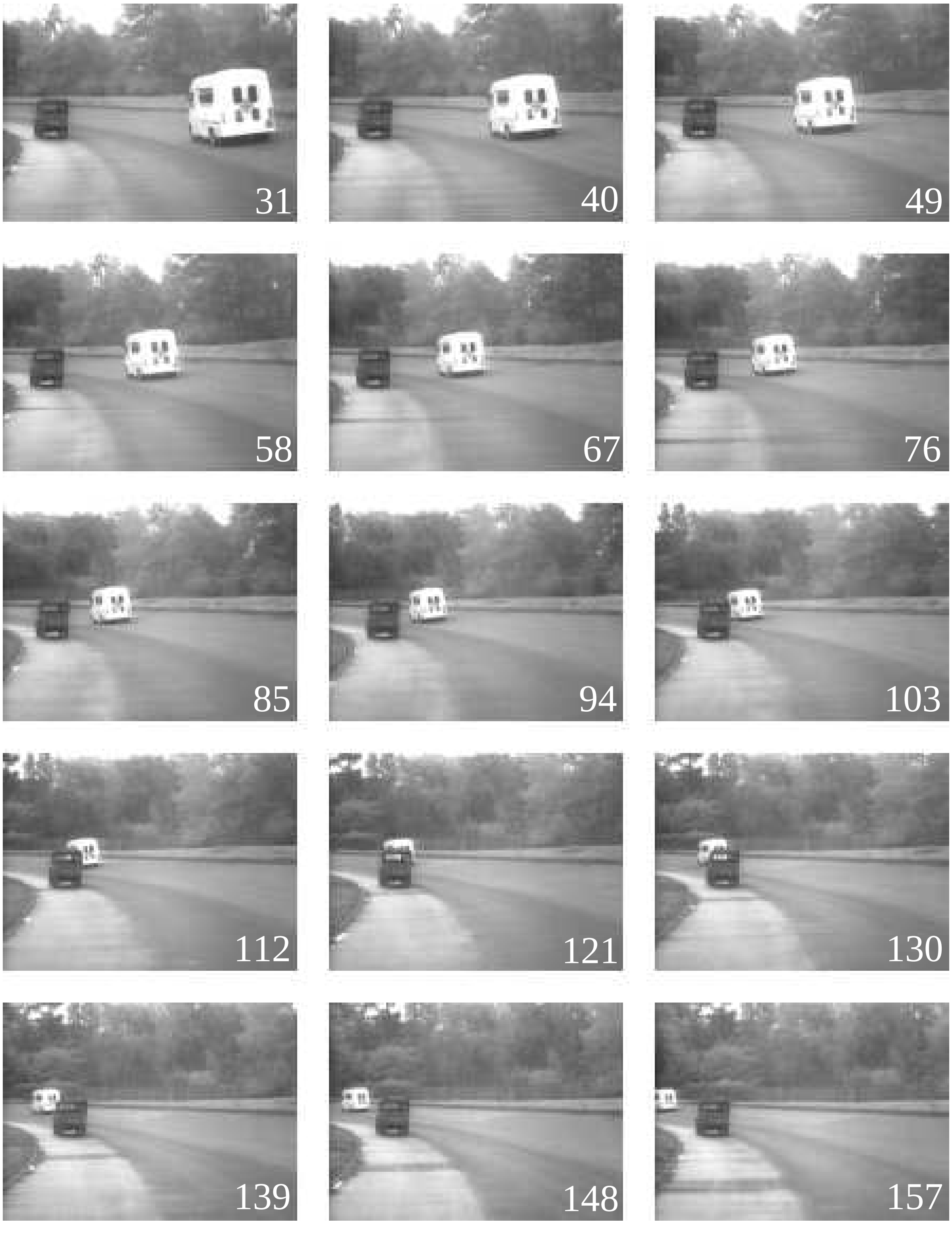


Figure 2: 15 frames from an example video sequence which ASSET-2 is expected to process. Every ninth frame is shown.

This list of clusters which are spatially and temporally significant is finally used to provide information about the motions of objects viewed by the video camera. The three dimensional positions and motions of these objects can be estimated by making simple assumptions about the world. In the case of an autonomous vehicle, this information is of practical value; for example it can enable the avoidance or the following of other vehicles.

# 4 Feature-Based Optic Flow Estimation

The decision to use image features to find optic flow was made for several reasons. Firstly, optic flow found using two dimensional features should contain as much information about the scene motion as is available, at the places in the image where the process of flow recovery is most well conditioned and where the information is most relevant. Directly related to this, flow discontinuities are not smoothed in the proposed method, as they are with most others, and no constraints are necessary in order to find full flow. The use of features to find optic flow naturally leads on to a sensible and simple representation of object shape, as discussed in Section 6. Finally, the process of finding image flow (to the accuracy required here) is relatively computationally expensive using other methods of motion estimation. (The advantages of using two dimensional features are discussed further elsewhere [2], [9].)

The two dimensional features are found using either the SUSAN corner detector [27], [26] or the Harris corner detector [15].

Feature tracking is performed using simple two dimensional motion models. Either constant velocity or constant acceleration models are used, depending on the application. (Few applications benefit from using the higher order constant acceleration model.)

The first stage in tracking features is to instantiate a motion model for each new feature, by matching features from the first two frames. If image motion is smaller than the distance between image features then there is no difficulty in finding a list of correct matches, and the problem is trivial. However, even in the case of stereo matching, where epipolar geometry constrains the possible disparity vectors, ambiguous matching possibilities usually arise. In monocular motion estimation this occurs even more frequently. An intelligent algorithm to disambiguate different possible matches is therefore necessary. (The following is described in more detail in [29].)

The matching done is based on local operations. Each feature in the first image of an image pair is looked at in turn, and an attempt is made to find for each a matching feature in the second image. The criteria which have been used to determine the quality of a match vary greatly in their complexity. Many non-real-time programs (both stereo and motion) have used correlation of small patches to give a measure of the quality of the match; see [3] (and the discussion here on different methods used), [18], [36] and [25]. However, the theoretical justification for using the correlation-based matching method is weak if the features are formed from the physical corners of objects, where the moving background will make up more than half of the surrounding patch. Clearly if the background has local structure then the idea of using small patch correlation is not a good one as the correlation for a correct match may be very poor. This will also be the case if the motion is strongly non-translational. Correlation is also in general much more computationally intensive than the method described below.

Another method for matching is to use a smaller amount of information describing the structure of the image at the feature to perform matching. Possible functions suitable for matching include image brightness and local image spatial derivatives. The image functions chosen are typically combined into a vector which is compared with the corresponding vector in the second image. The vector created from $n$ functions will be $n$ dimensional, and these functions need to be weighted so that they affect the vector to an extent roughly equivalent to their reliability as matching components.

This method of finding feature matches was chosen because it is much faster than a correlation-based method but is not much less reliable. Correlation-based matching is approximately 20 times slower than matching using a small number (three, for example) of feature attributes. The two methods typically attain matching success rates of approximately 95% and 85% respectively, where this "percentage" is given by;

$$\text{matching percentage} = \frac{(\text{number of correct matches} - \text{number of incorrect matches})}{\text{maximum possible number of correct matches}}. \quad (1)$$

(An interactive graphics program was written to allow matching of features to be done by eye so that the match lists which were produced automatically could be analyzed.)

It was found that the best attributes to use were the image brightness (not smoothed) and the $x$ and $y$ components of the position of the USAN centre of gravity, when using the SUSAN corner detector. When using the Harris detector, the attributes are found in the hardware and are fixed to be the smoothed image brightness and the $x$ and $y$ image derivatives.[1]

Obviously, once a corner has been matched once, the velocity estimate thus obtained allows a large reduction in the necessary search space for future matches. The motion model update filter used is a simplified two dimensional Kalman filter in which the search space is reduced over time and the model estimates are given increasing importance over time.

Simple logic is used to cope with the standard issues of temporarily unmatched features, new features, the purging of "bad" features and the designation of "high quality" features. Typically, two thirds of the features found by the feature detector are tracked successfully enough to be reported to the following stage as "high quality".

Figure 3 shows the set of velocity vectors found by the feature tracking stage of ASSET-2 at frame 44 of the sequence described and shown earlier. The Landrover has very little motion, the background is generally moving to the right, and the ambulance is moving to the left and away from the observer. Note the negative divergence in the flow of the ambulance.

[1] ASSET-2 has been written and tested using both the SUSAN two dimensional feature detector and the Harris "corner" detector. This is because the Harris detector is available in real-time hardware, and it is desirable that ASSET-2 should be able to work well with either these features or the SUSAN ones. Only SUSAN edges have been used.

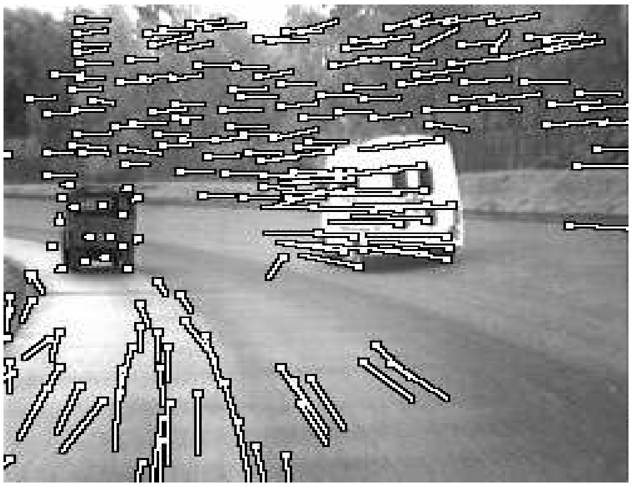

Figure 3: An example set of flow vectors found by the feature tracking stage of ASSET-2. The vectors point in the direction of motion and have magnitudes equal to twice the inter-frame image motion.

# 5 Flow Segmentation

The next step after flow estimation is to segment the flow list into clusters which represent different objects in the world.[2] The flow within each cluster is fit to a linear space dependent (i.e., affine) model, giving a total of six parameters for each cluster, using a least-squares fit.

The vector values used for the segmentation process may either be the current velocity estimate for each feature, or the feature's displacement over the past $N$ frames, where $N$ is constant, and typically set to 10. The former, more traditional, approach gives results which are more recently accurate, i.e., more "up-to-date", whilst the latter is more robust, and usually gives greater discrimination between objects with only slightly differing motion. It is the latter which is usually used within ASSET-2 to create the motion vector field.

The list of flow vectors is processed to find separate clusters using simple segmentation similar to the concept of the minimum spanning tree [13]. Each new candidate vector is compared with the value expected by the running flow model. When a new cluster is started, the initial flow model is that of constant flow, i.e., two parameters; this is replaced with the affine model once enough vectors have been included into the cluster. The distance function used for comparing a candidate flow vector $\vec{u}$ with the vector $\vec{u_m}$ estimated by the current model at the candidate flow vector's position allows small vectors to be matched to each other without bias by including a noise term into the fractional vector error;

$$D = \frac{|\vec{u} - \vec{u_m}|}{\frac{|\vec{u}|+|\vec{u_m}|}{2} + \sigma}, \qquad (2)$$

[2]Obviously, the objects so found may be physically connected but have a spatial discontinuity between them, resulting in separate motion models. It is not currently within the scope of this project to attempt to "link" such objects together.

where $\sigma$ is of the order of the error estimate of the flow vector. The "distance" thus calculated between a new flow vector and $\vec{u_m}$ must be less than a threshold $D_{\max}$ for the new vector to be included in the current cluster. The clustering process terminates when no more "linkable" neighbours can be found for any member of the cluster. A new cluster is then started. There is no question at this stage of ensuring that clusters do not overlap; the clusters are found individually, and once a flow vector has been used it is removed from the list.

Finally, once the list of independent clusters of flow vectors has been found, each cluster has its bounding box, centroid and motion model calculated, and added to the cluster description. (The centroid calculated is the mean position of the flow vectors which make up the cluster. This has more meaning than some "centre of shape" centroid, as outlying vectors – vectors incorrectly joined to the cluster – will not have so great an effect on the centroid.) This information is then passed on to the part of ASSET-2 which tracks clusters over time.

Figure 4 shows a typical set of clusters found by the flow segmentation stage of ASSET-

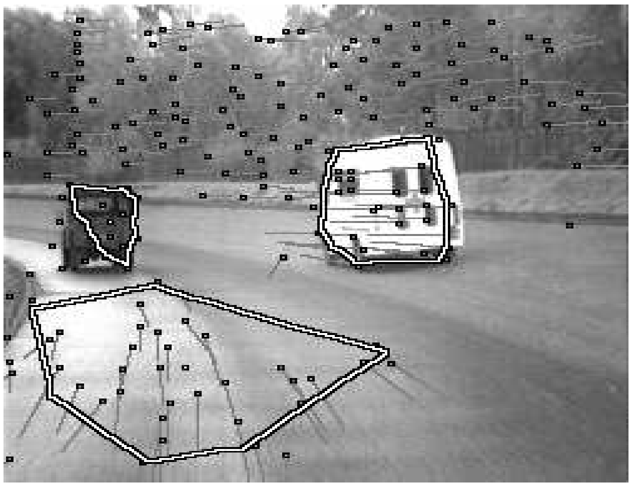

Figure 4: An example set of clusters found by the flow segmentation stage of ASSET-2, superimposed over the flow vectors.

2. No cluster has been formed where a flow vector has no neighbours to form one. Note the small errors in the two important clusters, particularly in the Landrover cluster, which should be filtered out over time. The cluster in the foreground will not appear in the filtered cluster list as a "good" cluster (see the following section) because it will not be consistent over time. The background cluster is ignored here by the display part of ASSET-2.

Figure 5 shows the set of clusters found near the ambulance when a variety of values for $D_{\max}$ is used. It is usually left at 0.15. The results show in this case that variations in $D_{\max}$ of one order of magnitude in either direction are acceptable, which is an encouraging result for a threshold.

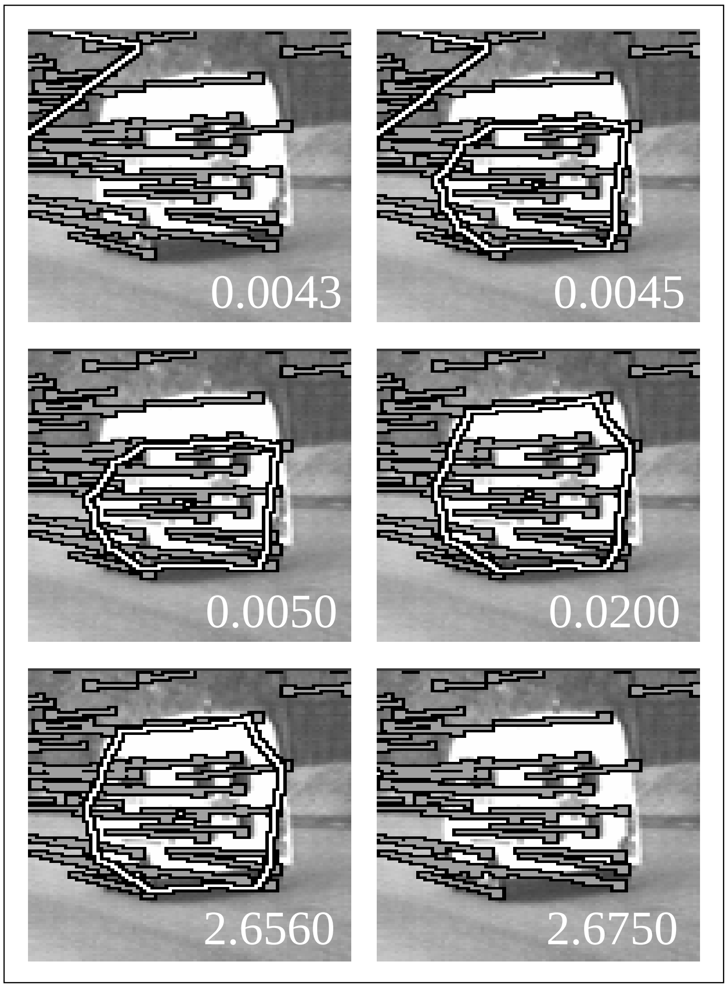


Figure 5: The clusters found near the ambulance when the segmenting threshold is varied over a wide range. The value of the threshold is shown for each case in the bottom right hand corner of the image.

## 6 Cluster Tracking and Filtering

Newly found clusters are now matched with the clusters in the filtered list, using time-symmetric matching [29]; both elements of a potential matched pair must prefer their proposed partner more than any other in the partner's list for the matched pair to be valid. The attributes used for matching are the cluster's motion model and the shape of the cluster. The reverse matching check is applied as before, except that the best backwards match (within a maximum search space and within a maximum allowed difference in matching attributes) is chosen to be the oldest cluster. This is so that a cluster will be consistently tracked over many frames even if it is occasionally incorrectly split into two clusters at the segmentation stage.

Each cluster's motion model (including modelling of acceleration of the whole cluster if appropriate) is updated in the same way that individual features' models are updated. To update the shape of each cluster requires more complicated processing, as it is clearly not possible to take an "average" of two shapes which are defined by lists of two dimensional points. The method used is based around a radial map shape representation. This is an array of distances from the centroid of the cluster to the "boundary", calculated at equal angle increments; each radius is initially set to the smallest value which just includes all of the features found between the neighbouring radial divisions – see Figure 6. The map is then forced to be convex, a heuristic which is appropriate for most applications and which is desirable given the sparse nature of the feature-based optic flow field. The radial map clearly does allow for non-convexities, and these may develop when image edges are used to aid boundary estimation. Notice that the radial map approximation to the convex hull gets worse at vertices as the number of radii is decreased. Typically, 32 radii are used, which is easily enough to match the accuracy of the rest of the system.

The radial map representation is not very different in its temporally consistent actions and scene reactions from some applications of active contours (that is, "snakes" [17], [6]). In particular, the boundary enhancement described later has similar properties to the edge finding functionals used by Kass *et al.* [17]; the effect is similar, but in ASSET-2 the solution is found by allowing image forces to act directly rather than finding local minima in functionals (that is, potentials, in the "physical forces" analogy).

The radial map is now a useful representation of cluster shape. It allows two shapes to be combined very easily, using for each updated radial estimate a similar method to that described for the cluster centroid and motion, i.e., a simplified Kalman filter, taking the previous best estimate and the new estimate as input.

This overall approach to shape tracking seems much more flexible and integrated than that described by Meyer and Bouthemy [20] and discussed in Section 2.

Apart from cluster tracking and filtering, this stage of ASSET-2 also performs a two-stage test for cluster occlusion; the first stage tests for occlusion of each tracked moving object by any other tracked moving object, and the second stage estimates when a tracked moving object has become occluded by a part of the world (i.e., the "background").

The first part of the test is simply for two dimensional overlap calculated using the

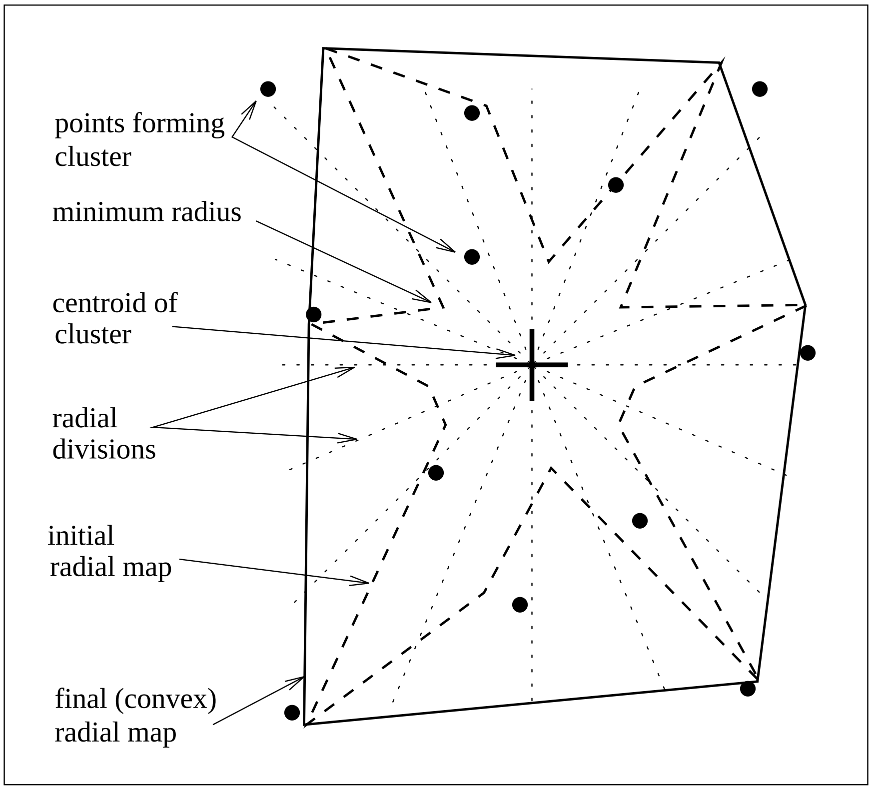


Figure 6: Converting a list of image points into a convex radial map.

bounding boxes of the clusters. If overlap is found, the cluster which will be occluded is predicted using the safe assumption that this cluster will have a higher image position for its lowest point than will the other cluster. This will hold for a ground-based situation, and could be replaced by more detailed analysis of cluster shape change in a more general situation. Once a cluster has been identified as being occluded its shape is kept constant as it moves behind the object in front of it. If it is at all visible then the part which can be seen is used to update the position and velocity of the cluster; if the object is completely hidden, estimates of its position are made. Once the object appears from behind the object in front then it is tracked as before.

The second part of the occlusion test tries to estimate when a tracked cluster is moving behind a static[3] part of the scene. There are several ways that this could be done, including detailed shape-change analysis, flow-boundary analysis and consistency between the motion of the centre-of-gravity and the integrated mean optic flow of the cluster. However, the simplest, most robust and most well-conditioned method is as follows. The number of features within a cluster would normally be expected to vary smoothly and slowly. By comparing the current number of features within a cluster with a heavily filtered version of

[3]Note; in this context static does not necessarily mean static in the image, but static with respect to the world, which the camera may be moving through.

this number, any rapid change in feature number can be found. Typically, the filter will have an update fraction of $\frac{1}{20}$, and the ratio of the current number of features within a cluster to the filtered number is compared with $\frac{1}{2}$ to determine whether or not the number of features is rapidly decreasing. When this type of occlusion is predicted, a cluster is marked as occluded and processed in exactly the same way as it is if occlusion by another *moving* object occurs. Thus the cluster's shape is frozen, and its motion model is updated using as much available information as possible. If the cluster becomes unoccluded (tested for using the same criteria) then normal tracking is resumed.

The filtered list of clusters can be passed to a higher level stage for scene interpretation. Not all clusters in the filtered list are output as reliable objects. All the clusters found in each image are passed to the filtered list, and until they can be discarded as being spurious they remain in the list. Therefore only clusters which have been seen for longer than a minimum lifetime and which have been seen for more than a minimum percentage of this lifetime are counted as reliable objects. Also, in order to reject spurious objects which have some kind of consistency but which are physically untenable (for example, on the ground immediately in front of the camera), the change in position of the centre of gravity of each cluster over some time interval is compared with an integrated measure of its velocity. This determines whether the object is self-consistent. Examples of the graphical output are given in Section 9.

# 7 Boundary Enhancement Using Edges

Boundaries of objects in the real world almost always give rise to image edges. Therefore a section has been built into ASSET-2 which uses image edges found by the SUSAN edge detector [26] to "fine-tune" the cluster boundaries so that nearby edges can pull the boundaries into them. (Related here is the interesting work by Etoh, e.g., in [14], which also uses "static" image cues to aid motion segmentation.) This will of course not always be successful, as background edges may confuse the issue. However, this part of ASSET-2 has usually been a valuable addition to the core processes. In a situation where edge information is not relevant ASSET-2 can run without using the "edge enhancement" stage.

Edge correction of a cluster's boundary only takes place after it has been tracked for a certain number of frames. This is partly to prevent the wastage of computational effort on spurious clusters, but, more importantly, because, until the estimate of the cluster boundary is close to the actual object boundary, edge correction could degrade the object boundary instead of improving it.

This part of ASSET-2 works by taking the list of SUSAN edges and creating a vector field from it which guides the object boundaries towards image edges. A vector field is initialized to $(0,0)^T$ at every point. Next each edge point is used to add to the existing field a line of vectors pointing towards that edge point – see Figure 7. This line of influence extends away from the edge in both directions perpendicular to it. The magnitudes of the vectors decrease in proportion to their distance from the original edge point.

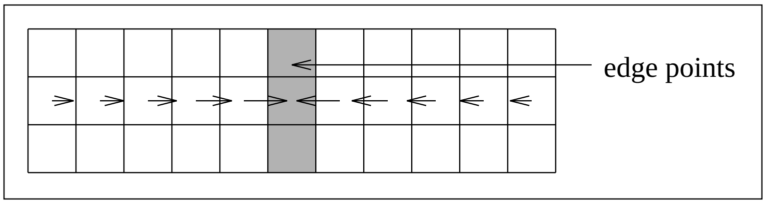


Figure 7: The addition to the edge vector field caused by a single edge point.

Next the radial map describing the cluster outline is allowed to "follow the forces" which the vector field defines in the following way. The map boundary is split up into sections, centred at the end of one radius (say radius $i$) and running halfway towards the adjacent radii ends (radii $i-1$ and $i+1$). Thus each section will be made up of two line segments. Each section of the map boundary has the radial force on it calculated in the following way;

$$f(i) = \vec{n}(i). \sum_{x,y} \vec{e}(x,y), \tag{3}$$

where $f(i)$ is the "force" on $r(i)$, the $i$th radius, $\vec{e}(x,y)$ is the edge vector field, the sum is taken over each pixel in the two line segments making up the boundary section, and $\vec{n}(i)$ is the unit vector in the direction of the $i$th radius.

The dot product finds the component of the sum of the "forces" on the boundary section in the direction of the radius. The force $f(i)$ is calculated for all $i$ and then each $f(i)$ is smoothed with its neighbours. Next each radius is allowed to change its length by a maximum of one pixel (in either direction), depending on the force on it. This process is repeated several (typically ten) times. This method was chosen instead of allowing a small number of larger changes to the radii for the following reasons. Following this procedure the shape is gradually allowed to change and the radii to affect each other to find a best global fit to the edge influences. It also solves the problem of the fact that the closer the section is to an image edge the greater the force on it, which would give unstable (or at least unpredictable) oscillations about the correct position if the correction to the radius were made proportional to the force on it. The method used gives a stable and accurate solution.

Finally, the corrected cluster shape is used to improve the original filtered radial map. To provide good temporal stability a weighted mean of the input to the edge correction algorithm and its output is taken.

This section of ASSET-2 is not implemented in the real-time version, as image data is not available to the main (PowerPC) processors with the current system architecture. However, this will soon be rectified with the release of the Parsytec TIP-Bus PowerPC boards.

# 8 Real-Time Implementation of ASSET-2

The ASSET-2 system has been implemented at DRA Chertsey to run live on the ROVA PowerPC-based image processing system (see Figure 8). Feature detection is performed by

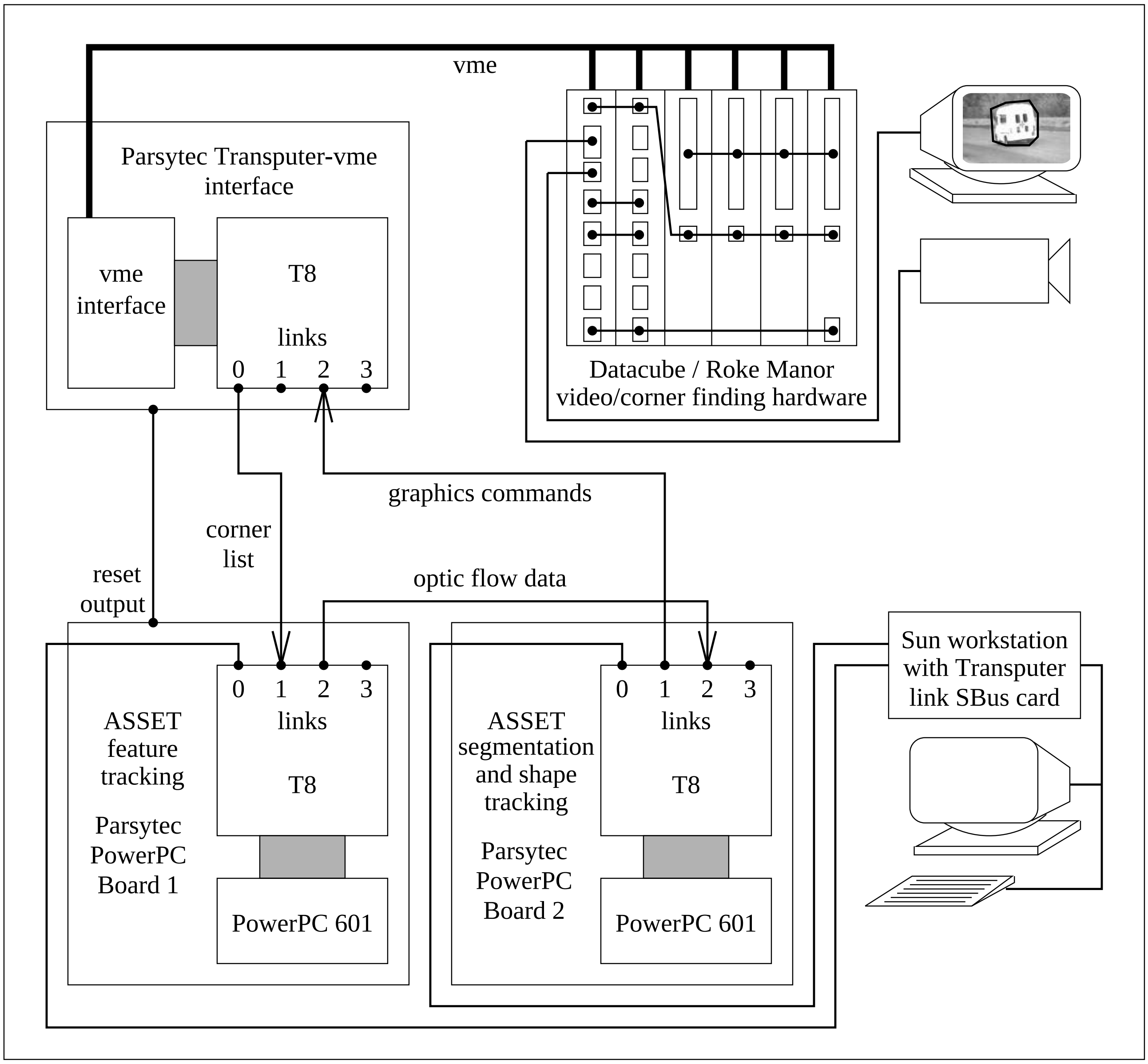


Figure 8: The real-time implementation of ASSET-2 on the ROVA PowerPC-based image processing system.

custom hardware supplied by Roke Manor Research Limited, which provides a list of features at full frame rate (25 Hz). This custom hardware comprises four VME boards with Datacube Maxbus digital image bus connections on the front panels. A Datacube Digimax board digitizes each image and stores this digital image in a Datacube Framestore. The digital image is then sent from the Framestore down the Maxbus link to the Roke Manor boards which carry out the different stages of the Harris corner finding algorithm [15]. A Parsytec Transputer link to VME interface board takes the resulting list of image corners from VME space and sends it down a Transputer link to the first of two Parsytec Transputer/PowerPC processing boards, straight into PowerPC memory.

The Parsytec PowerPC boards run PowerPC 601 processors at 80 MHz; two such boards are used here for real-time running of ASSET-2. The feature tracking stage of ASSET-2 runs on the first PowerPC board at full frame rate. This is possible due to the efficient design of all algorithms which make up ASSET-2. The resulting vector flow field is sent down a Transputer link to the second PowerPC board, where the remainder of ASSET-2 is run, also at full frame rate.

Results are graphically overlaid onto the images stored in the Datacube Framestore using the Parsytec VME interface board again. Images are not stored for longer than one frame period in this double-buffered Framestore, so that graphics are not overlaid on exactly the image which they correspond to. However, the incorrect positioning is barely visible by eye, and errors are reduced by "predicting ahead" the movement of all graphics, as image velocities are available to make this possible.

# 9 Results

In this section the results of testing ASSET-2 are presented. The image sequences were produced by DRA except where indicated otherwise.

Figure 9 shows the tracking of the Landrover shown earlier, from the first tracked frame. On the left, the edge correction stage has not been used to "fine-tune" the cluster boundaries. On the right, image edges have been used; they are first used in frame 8. In both cases ASSET-2 is working stably, and the improvement over time of the estimated boundary is evident. By frame 10 the advantage of using edges is clear. The central rectangle identifies the tracked centre of gravity of the cluster, with a vector coming out from it showing the tracked image motion of the cluster, not visible here due to very low image velocity. (Edge correction was also used in the tests shown in Figures 10 to 14, but not in the remaining tests.)

Figure 10 shows frame 44 of the ambulance sequence. The boundaries are being accurately and stably tracked. No spurious objects are present.

Figure 11 shows sections of frames 100 to 166 of the ambulance sequence. Both vehicles are still being accurately tracked, with the exception of the minor error in the top edge of the Landrover. There is no single isolated image edge corresponding to the top of the Landrover, so there is a slight inaccuracy in the estimation of its boundary here. In frame 106

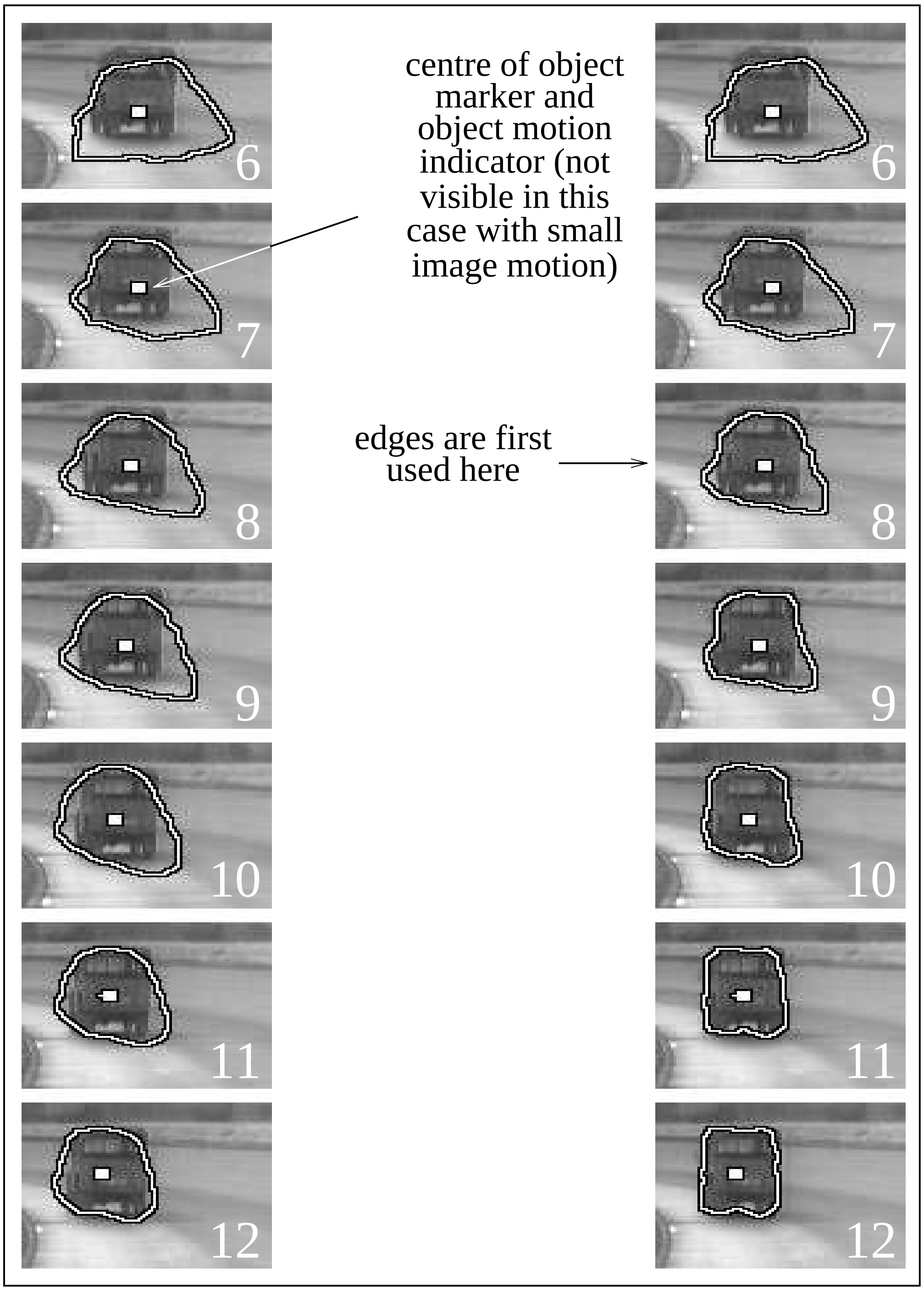


Figure 9: Seven frames showing the increasing accuracy of the estimated object boundary as time commences from initialization of a filtered cluster. On the left, the edge correction stage has not been used to "fine-tune" the cluster boundaries. On the right, image edges have been used; they are first used in frame 8.

the ambulance is partially occluded by the Landrover. ASSET-2 shows that it has recognized this occurrence by marking the ambulance's boundary with a darker line. The tracking of the two vehicles is good, with the "unocclusion" of the ambulance being recognized in frame 148.

Figure 12 shows ASSET-2 tracking a radio-controlled truck in a situation where occlusion by a static obstacle takes place. As the truck passes behind the static obstacle, occlusion is automatically registered and shown by changing the brightness of the superimposed outline. The shape is frozen but the position is updated using the existing 2D motion model. Without acceleration modelling, when the vehicle emerges from behind the obstacle, it is a long way ahead of the expected position. However, using acceleration modelling (as shown here), the prediction of the object's position is accurate enough to allow the old cluster to rejoin the truck once it reappears (shown by continuity of cluster label - here the number 3).

Figure 13 shows a Ford vehicle travelling away from the camera. In this case, the camera is static. This shows ASSET-2 functioning accurately in a situation where there is no background flow, and the tracking continues until the motion of the vehicle is less than one pixel per frame.

Figure 14 shows an image sequence taken from a moving vehicle at night with an infra-red camera. The quality of the infra-red images is not as good as images taken with normal video cameras; the resolution is not as high, and there are horizontal dark and light stripes superimposed on the image. (Both of these degradations are just visible on the printed picture shown here.) However, ASSET-2 still functions; the overtaking vehicle is adequately tracked. The sequence was kindly provided by Pilkington Plc.

Figure 15 shows a frame from a sequence in which a Landrover fills up a large portion of the image, whilst being followed by the vehicle carrying the video camera. Even this large object is tracked as a single cluster successfully.

Figure 16 shows ASSET-2 tracking a moving aircraft. The original video was taken with a hand-held long focal length camcorder; the resulting image sequences are very shaky, but a suitable object motion model (no acceleration modelling, low velocity updates and high position updates) allows successful tracking of various aircraft seen at the airshow, including independent tracking of several helicopters simultaneously. (The helicopters are small and the output is only meaningful if viewed dynamically, hence it is not shown here.)

Figure 17 shows ASSET-2 tracking several independently moving vehicles at a roundabout. The video sequence is taken by a traffic monitoring camera which, though mounted in a stationary position, moves around as its platform sways. The number of radii in the shape model was reduced to 8 (hence the more "quantized" outlines) due to the limited graphics speed of the Framestore. The sequence was kindly provided by Roke Manor Research Limited.

Figure 18 shows ASSET-2 tracking several independently moving vehicles on a motorway. Simple geometry calculations and calibrations are used to convert image speed and position to vehicle speeds, which are superimposed over the vehicles.

## 10 Conclusions and Future Work

In this paper ASSET-2, a complete real-time vision system for segmentation and tracking of independently moving objects, has been described. ASSET-2 requires no camera calibration or motion registration, and performs all calculations in the image plane. Information gained from the motion of two dimensional features is integrated with one dimensional boundaries. ASSET-2 achieves automatic instantiation of tracking for each new object detected, and any number of objects can be tracked. Occlusion of one object by another or by a "static" part of the scene is correctly handled. The temporal integration of segmentation information in such a robust and flexible way is a step forward in the area of image sequence understanding. The results presented show ASSET-2 performing reliable and accurate object tracking for several image sequences.

In the near future ASSET-2 will be extended to incorporate knowledge of the world so that simple estimates of the three dimensional positions and motions of objects in its field of view can be made (this has already been done for the simple case of motorway traffic monitoring). The ASSET-2 system will be integrated with a structure-from-motion system which recovers world structure in a *static* environment. ASSET-2 will be used to segment out and track moving vehicles, and the static part of the scene will then be tracked by the structure-from-motion system.

## References


[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.

[2] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):2–14, 1986.

[3] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(4):333–340, 1980.

[4] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 27–32, 1990.

[5] M.J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *Proc. 2nd European Conf. on Computer Vision*, pages 485–493. Springer-Verlag, 1992.

[6] A. Blake and R. Cipolla. Robust estimation of surface curvature from deformation of apparent contours. *Image and Vision Computing*, 9(2):107–112, 1991.

[7] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. Journal of Computer Vision*, 10(2):157–182, 1993.

[8] P. Bouthemy and P. Lalande. Detection and tracking of moving objects based on a statistical regularization method in space and time. In O. Faugeras, editor, *Proc. 1st European Conf. on Computer Vision*, pages 307–311. Springer-Verlag, 1990.

[9] J.M. Brady. Seeds of perception. In *Proc. Alvey Vision Conference*, pages 259–265, 1987.

[10] P.J. Burt, J.R. Bergen, R. Hingorani, R.J. Kolczynski, W.A. Lee, A. Leung, J. Lubin, and J. Shvaytser. Object tracking with a moving camera; an application of dynamic motion analysis. In *IEEE Workshop on Visual Motion*, Irvine, CA., 1989.

[11] C. Debrunner and N. Ahuja. Motion and structure factorization and segmentation of long multiple motion image sequences. In *Proc. 2nd European Conf. on Computer Vision*, pages 217–221. Springer-Verlag, 1992.

[12] N. Diehl. Segmentation and motion estimation in image sequences. *SPIE 1260; Sensing and Reconstruction of Three-Dimensional Objects and Scenes*, pages 250–261, 1990.

[13] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[14] M. Etoh and Y. Shirai. Segmentation and 2D motion estimation by region fragments. In *Proc. 4th Int. Conf. on Computer Vision*, pages 192–199, 1993.

[15] C.G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[16] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proc. 2nd European Conf. on Computer Vision*, pages 282–287. Springer-Verlag, 1992.

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

[18] D.T. Lawton. Processing translational motion sequences. *Computer Vision, Graphics and Image Processing*, 22:116–144, 1983.

[19] P.F. McLauchlan, I. Reid, and D.W. Murray. Coarse image motion for saccade control. In *Proc. 3rd British Machine Vision Conference*, 1992.

[20] F.G. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *Computer Vision, Graphics and Image Processing: Image Understanding*, 60(2):119–140, September 1994.

[21] A. Meygret and M. Thonnat. Segmentation of optical flow and 3D data for the interpretation of mobile objects. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 238–245, 1990.

[22] D.W. Murray and B.F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2):220–228, March 1987.

[23] R.C. Nelson. Qualitative detection of motion by a moving observer. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1991.

[24] A. Rognone, M. Campani, and A. Verri. Identifying multiple motions from optical flow. In *Proc. 2nd European Conf. on Computer Vision*, pages 258–266. Springer-Verlag, 1992.

[25] L.S. Shapiro, H. Wang, and J.M. Brady. A matching and tracking strategy for independently-moving, non-rigid objects. In *Proc. 3rd British Machine Vision Conference*, 1992.

[26] S.M. Smith. *Feature Based Image Sequence Understanding*. D.Phil. thesis, Robotics Research Group, Department of Engineering Science, Oxford University, 1992.

[27] S.M. Smith. A new class of corner finder. In *Proc. 3rd British Machine Vision Conference*, pages 139–148, 1992.

[28] S.M. Smith. ASSET-2: Real-time motion segmentation and shape tracking. In *Proc. 5th Int. Conf. on Computer Vision*, pages 237–244, 1995.

[29] S.M. Smith and J.M. Brady. A scene segmenter; visual tracking of moving vehicles. *Engineering Applications of Artificial Intelligence*, 7(2):191–204, April 1994.

[30] S.M. Smith and J.M. Brady. ASSET-2: Real-time motion segmentation and shape tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):814–820, 1995.

[31] A. Spoerri and S. Ullman. The early detection of motion boundaries. In *Proc. 1st Int. Conf. on Computer Vision*, pages 209–218, 1987.

[32] W.B. Thompson. Combining motion and contrast for segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(6):543–549, 1980.

[33] W.B. Thompson and T.-C. Pong. Detecting moving objects. In *Proc. 1st Int. Conf. on Computer Vision*, pages 201–208, 1987.

[34] P.H.S. Torr and D.W. Murray. Statistical detection of independent movement from a moving camera. In *Proc. 3rd British Machine Vision Conference*, 1992.

[35] A. Verri, S. Uras, and E. De Micheli. Motion segmentation from optical flow. In *Proc. 5th Alvey Vision Conference*, pages 209–214, 1989.

[36] H. Wang, M. Brady, and I. Page. A fast algorithm for computing optic flow and its implementation on a transputer array. In *Proc. 1st British Machine Vision Conference*, pages 175–180, 1990.

[37] A.M. Waxman and J.H. Duncan. Binocular image flows : Steps toward stereo-motion fusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):715–729, 1986.

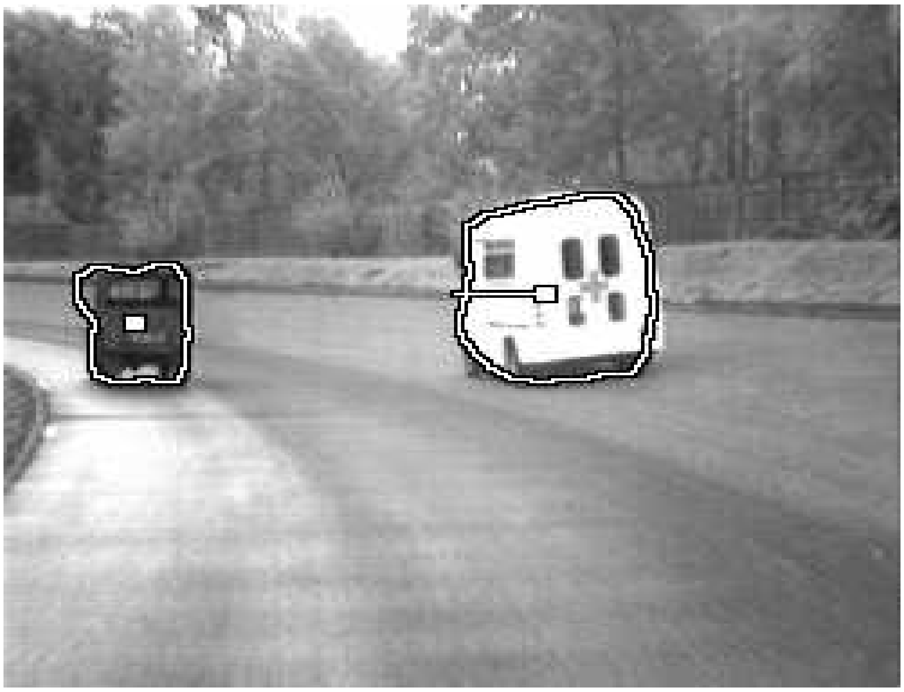

Figure 10: The output of ASSET-2 at frame 44 of the ambulance sequence.

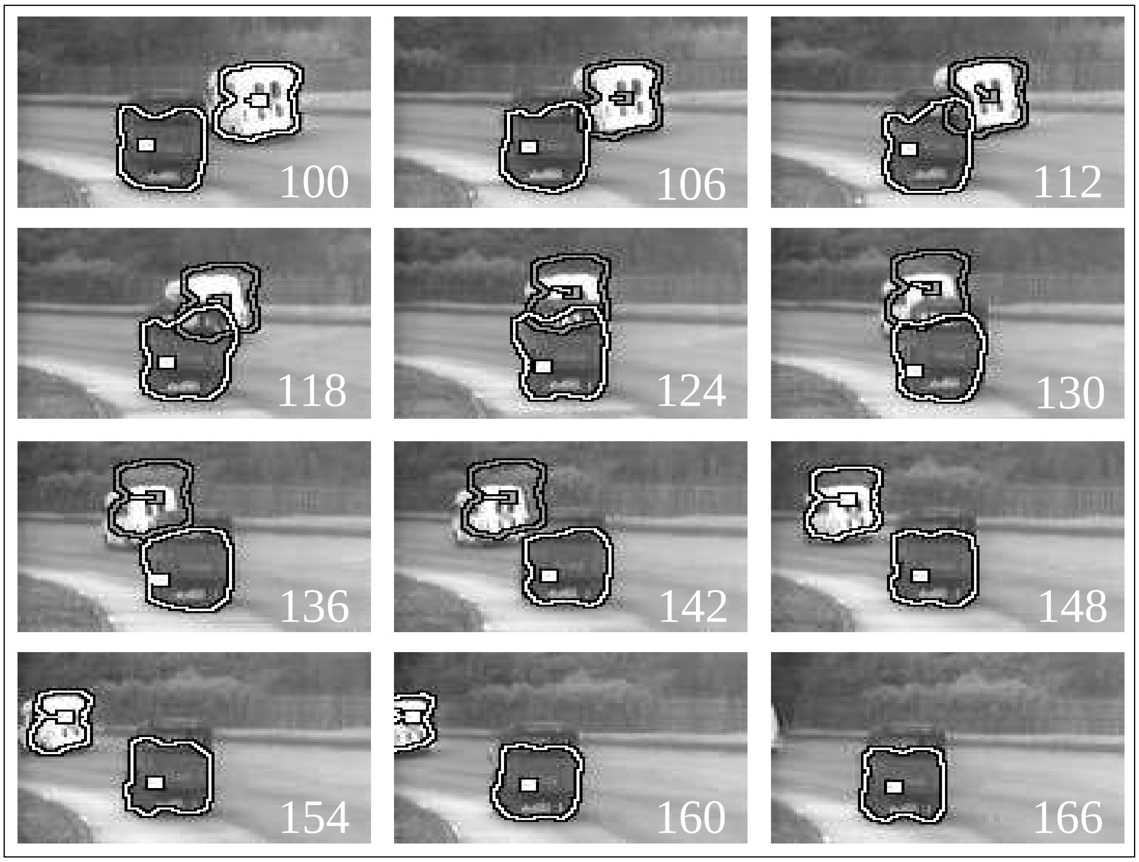


Figure 11: Twelve frames showing how ASSET-2 tracks objects before, during and after occlusion, using frames 100 to 166 of the ambulance sequence.

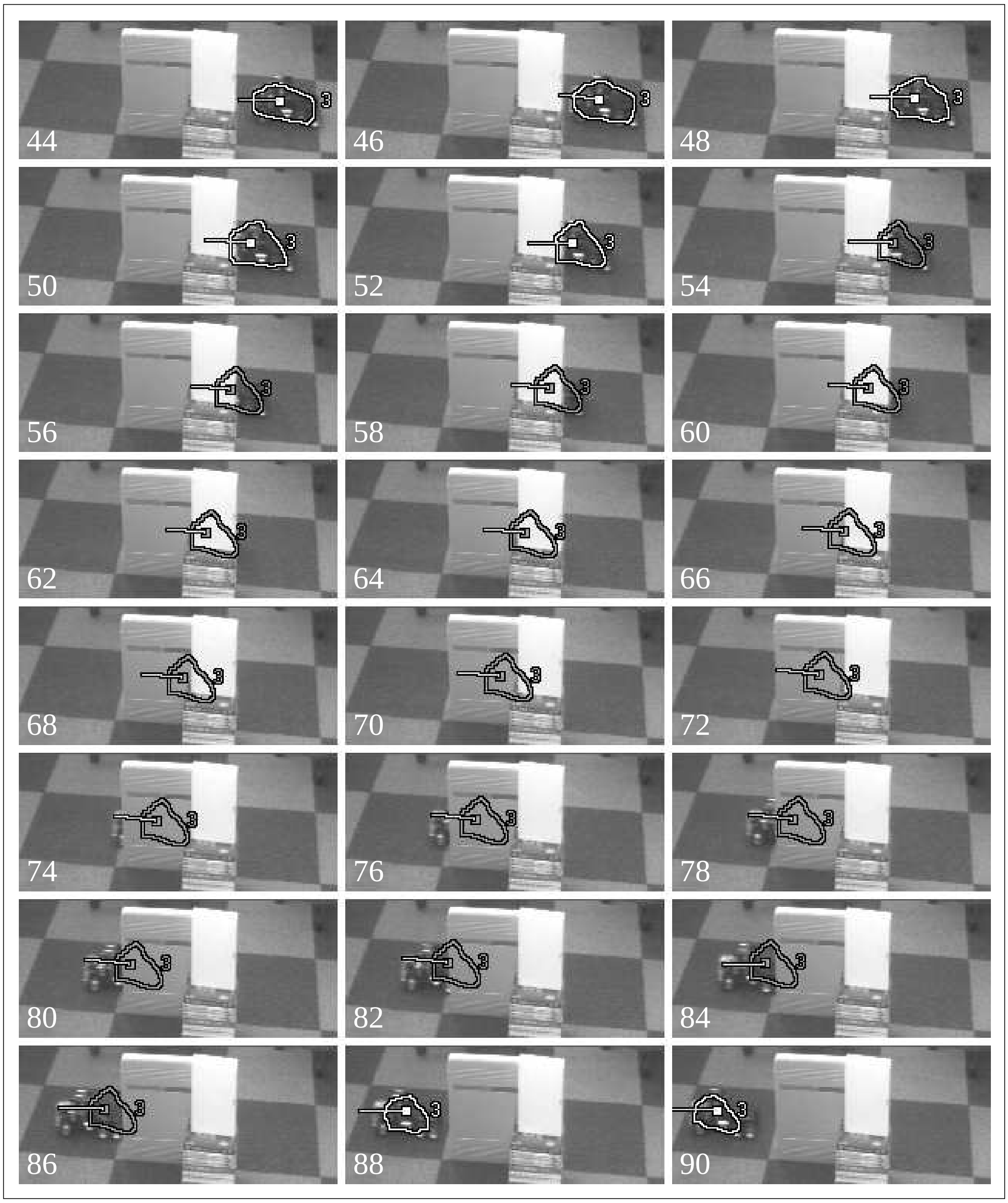


Figure 12: Example output from ASSET-2; a moving object is occluded by a static obstacle.

Figure 13: The output of ASSET-2 at frame 9 of the Ford sequence.

Figure 14: The output of ASSET-2 at frame 8 of a sequence taken by an infra-red camera at night.

Figure 15: The output of ASSET-2 at frame 7 of a sequence taken following a Landrover at close range.

Figure 16: The output of ASSET-2 when given a sequence from an airshow.


Figure 17: The output of ASSET-2 when given a sequence from a traffic monitoring video camera.

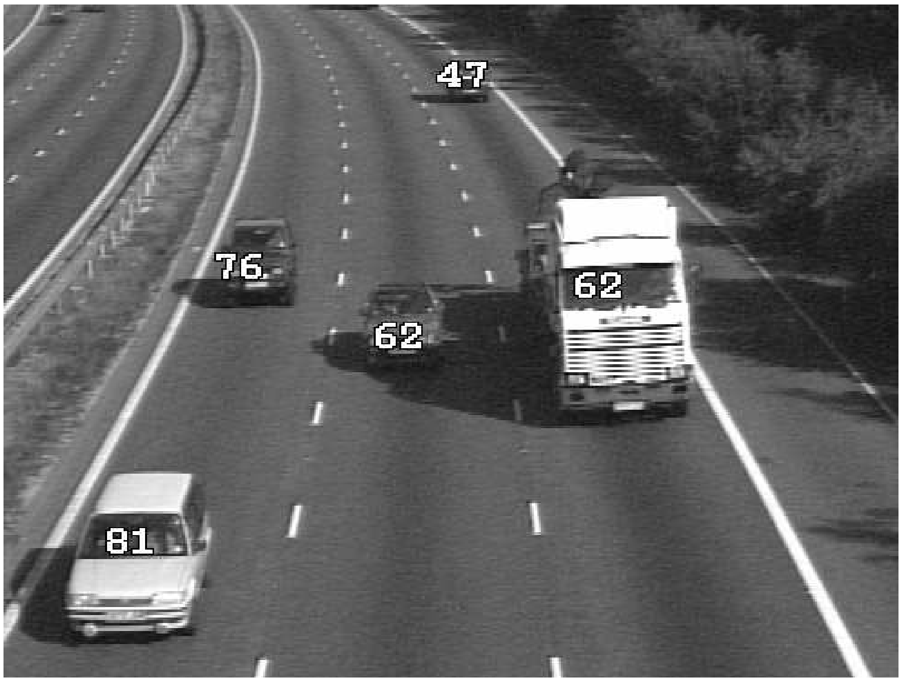


Figure 18: The output of ASSET-2 when given a sequence from a traffic monitoring video camera over a motorway, with real speeds calculated.