Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 96568, 23 pages
doi:10.1155/2007/96568

Research Article

Robust Real-Time Tracking for Visual Surveillance

David Thirde,[1] Mark Borg,[1] Josep Aguilera,[2] Horst Wildenauer,[2] James Ferryman,[1] and Martin Kampel[2]

[1] *School of Systems Engineering, Computational Vision Group, The University of Reading, Reading RG6 6AY, UK*
[2] *Computer Science Department, Pattern Recognition and Image Processing Group, Vienna University of Technology, 1040 Vienna, Austria*

Received 21 October 2005; Revised 23 March 2006; Accepted 18 May 2006

Recommended by John Maccormick

This paper describes a real-time multi-camera surveillance system that can be applied to a range of application domains. This integrated system is designed to observe crowded scenes and has mechanisms to improve tracking of objects that are in close proximity. The four component modules described in this paper are (i) motion detection using a layered background model, (ii) object tracking based on local appearance, (iii) hierarchical object recognition, and (iv) fused multisensor object tracking using multiple features and geometric constraints. This integrated approach to complex scene tracking is validated against a number of representative real-world scenarios to show that robust, real-time analysis can be performed.

1. INTRODUCTION

This paper describes work undertaken on the EU project AVITRACK. The main aim of this project is to automate the supervision of commercial aircraft servicing operations on the ground at airports (in bounded areas known as *aprons*). Figure 1 shows apron echo-40 at Toulouse Airport in France. The servicing operations are monitored from multiple cameras that are mounted on the airport building surrounding the apron area, each servicing operation is a complex 30-minute routine involving the interaction between aircraft, people, vehicles, and equipment.

The full AVITRACK system is presented in Figure 2. The focus of this paper is on the real-time tracking of the objects in scene, this tracking is performed in a decentralised multi-camera environment with overlapping fields of view between the cameras [1]. The output of this—the scene tracking module—is the predicted physical (i.e., real-world) objects in the monitored scene. These objects are subsequently passed (via a spatiotemporal coherency filter) to a scene understanding module where the activities within the scene are recognised. This result is fed—in real time—to apron managers at the airport. The modules communicate using the XML standard, which although inefficient allows the system to be efficiently integrated. It is imperative that this system must be capable of monitoring a dynamic environment over an extended period of time, and must operate in real time (defined as 12.5 FPS with resolution 720×576) on colour video streams. More details of the complete system are given in [2].

The tracking of moving objects on the apron has previously been performed using a top-down model-based approach [3] although such methods are generally computationally intensive. On a desktop computer (2×3 GHz pentium-4 processors with 2 Gb RAM running Suse Linux 9.1) we have found the model-based method to fit one model in 0.25 seconds. In the AVITRACK system there are 28 different object types which would therefore result in a frame rate of 0.14 frames per second for tracking a *single* object. An alternative approach, bottom-up scene tracking, refers to a process that comprises the two subprocesses, *motion detection* and *object tracking*; the advantage of bottom-up scene tracking is that it is more generic and computationally efficient compared to the top-down method.

Motion detection methods attempt to locate connected regions of pixels that represent the moving objects within the scene; there are many ways to achieve this including frame-to-frame differencing, background subtraction, and motion analysis (e.g., optical flow) techniques. Background subtraction methods [4–6] store an estimate of the static scene, which can be accumulated over a period of observation; this background model is subsequently applied to find foreground (i.e., moving) regions that do not match the static scene.

Image-plane-based object tracking methods take as input the result from the motion detection stage and commonly apply trajectory or appearance analysis to predict, associate,

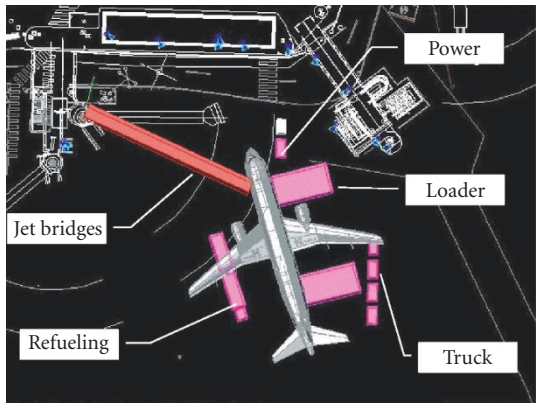


Figure 1: The distribution of equipment around an A320 aircraft on apron echo-40 at Toulouse Airport, France.

and update previously observed objects in the current time step. The tracking algorithms have to deal with motion detection errors and complex object interactions in the congested apron area, for example, merging, occlusion, fragmentation, nonrigid motion, and so forth. Apron analysis presents further challenges due to the size of the vehicles tracked; therefore, prolonged occlusions occur frequently throughout apron operations. The Kanade-Lucas-Tomasi (KLT) feature tracker [7] combines a local feature selection criterion with feature-based matching in adjacent frames; this method has the advantage that objects can be tracked through partial occlusion when only a subset of the features are visible. To improve the computational efficiency of the tracker motion segmentation is not performed globally to detect the objects. Instead, the features are used in conjunction with a rule-based approach to correspond to connected foreground regions; in this way the KLT tracker *simultaneously* solves the problems of data association and tracking without presumption of a global motion for each object.

The goal of object recognition is to identify at least the object category and at most the object category, size, and precise spatial attributes (e.g., orientation, centroid, etc.). In the latter scenario, model-based methods (e.g., [3]) can be applied to locate the objects of interest in the scene. An alternate approach is to train a classifier to distinguish the different object types (e.g., [8]); a major drawback with this approach is the scalability to classifying multiple objects from multiple cameras, especially when there are minor differences between some object types or when objects appear vastly different under perspective transformations. The challenges faced in apron monitoring are the quantity (28 categories) and similarity of objects to be classified, for example, the majority of vehicles have similar appearance and size; therefore, the simple descriptors used in many visual surveillance algorithms are likely to fail.

Data fusion combines the tracking data measured by the individual cameras to maximise the useful information content of the observed apron. The main challenge of data fusion for apron monitoring is the tracking of large objects with significant size, existing methods generally assume point sources [1] and therefore extra descriptors are required to improve the association. People entering and exiting vehicles also pose a problem in that the objects are only partially visible; therefore, they cannot be localised using the ground plane.

This paper is organised as follows. Section 2 describes the scene tracking module, this module is responsible for tracking the objects in the scene and comprises motion detection, object tracking, object recognition, and data fusion component modules. Section 3 evaluates the performance of each of these component modules over a range of representative test sequences. Where appropriate, results are presented for test sequences that are not from the AVITRACK project, to show the genericity of the proposed methods.

2. SCENE TRACKING

Figure 3 shows the scene tracking module in the AVITRACK system. This module comprises two distinct stages—per camera (2D) object tracking and centralised world (3D) object tracking. The per camera object tracking consists of motion detection (Section 2.1) to find the moving objects in the observed scene followed by object tracking in the image plane of the camera (Section 2.2). The tracked objects are subsequently classified using a hierarchical object recognition scheme (Section 2.3). The tracking results from the eight cameras are then sent to a central server where the multiple observations are fused into single estimates (Section 2.5). In this section we detail each step of the scene tracking module.

2.1. Motion detection

The output of a motion detector is connected regions of foreground pixels, which are then used to track objects of interest across multiple frames. For AVITRACK, a total of 16 motion detection algorithms were implemented and quantitatively evaluated on various apron sequences under different environmental conditions (sunny conditions, fog, etc.). The metrics adopted for AVITRACK, the evaluation process, and the results obtained are described in more detail in Section 3.1. Three algorithms (all based on the aforementioned background subtraction method) were shortlisted in the evaluation process, as they were found to have acceptable susceptibility to noise and good detection sensitivity. These are mixture of Gaussians [9], colour and edge fusion [5], and colour mean and variance [6]. After taking into account the evaluation results, the colour mean and variance method was the final choice for AVITRACK.

The colour mean and variance is a motion detection algorithm that uses the background subtraction technique to segment foreground objects from the background. A pixel-wise Gaussian distribution over the normalised RGB colour space is used for modelling the background.

In order to achieve a real-time frame rate, a coarse-to-fine quad-tree optimisation technique is used during motion detection. The image is initially divided into 9×9 pixel blocks and motion detection using the colour mean and variance algorithm performed at the corner pixels of each block. If

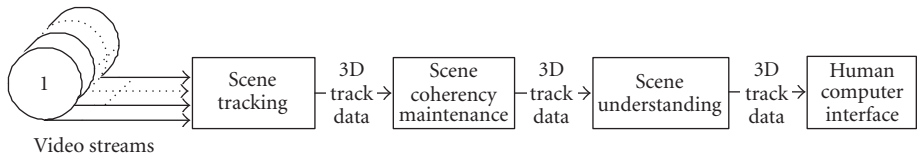


Figure 2: The AVITRACK system.

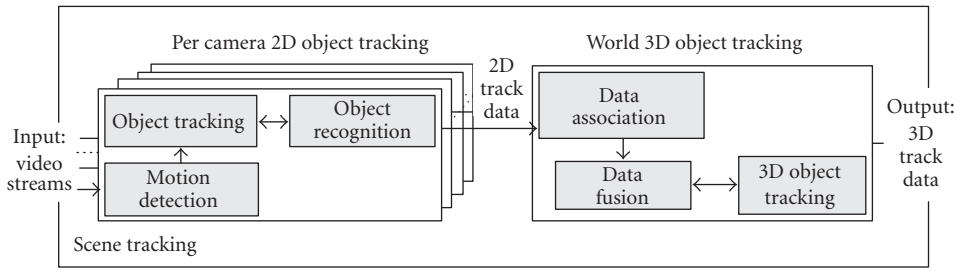


Figure 3: The scene tracking module in the AVITRACK system.

the motion labels of the corner pixels of the block are identical (either "foreground," "background," "shadow," or "highlight"), then the whole block of pixels is assigned that particular label; if any of the corner pixels has a different label from the others, then the block is subdivided into four. The process is repeated iteratively until the block size becomes 1 pixel or until a block is found which has the same motion detection labels for its corners. This optimisation technique provides a large speed-up in motion detection; the disadvantage is that moving objects with an area less than 9×9 pixels are not detected.

2.2. Object tracking

Real-time object tracking can be described as a correspondence problem, and involves finding which object in a video frame relates to which object in the next frame. Normally, the time interval between two successive frames is small; therefore, interframe changes are limited, thus allowing the use of temporal constraints and/or object features to simplify the correspondence problem.

The KLT algorithm [7] is used to track objects in the AVITRACK system, this algorithm considers features to be independent entities and tracks each of them individually. Therefore, it is incorporated into a higher-level tracking process that groups features into objects, maintains associations between them, and uses the individual feature tracking results to track objects, taking into account complex object interactions.

For each object O, a set of sparse features S is maintained. $|S|$—the number of features per object—is determined dynamically from the object's size and a configurable feature density parameter $\boldsymbol{\rho}$:

$$|S| = \frac{\text{area}(O)}{|w|^2} \times \boldsymbol{\rho}, \tag{1}$$

where $|w|$ is the size of the feature's window (9×9 pixels in our case). In experiments $\boldsymbol{\rho} = 1.0$, that is, $|S|$ is the maximal number of features that can spatially cover object O, without overlap between the local feature windows.

The KLT tracker takes as input the set of observations $\{M_j\}$ identified by the motion detector. Here, an observation M_j is a connected component of foreground pixels, with the addition of a nearest neighbour spatial filter of clustering radius r_c, that is, connected components with gaps $\leq r_c$ are considered as one observation. Given such a set of observations $\{M_j^t\}$ at time t, and the set of tracked objects $\{O_i^{t-1}\}$ at $t-1$, the tracking process is summarised as follows.

(1) Generate object predictions $\{P_i^t\}$ for time t from the set of known objects $\{O_i^{t-1}\}$ at $t-1$, with the set of features $S_{P_i^t}$ set to $S_{O_i^{t-1}}$.
(2) Run the KLT algorithm to individually track each local feature belonging to $S_{P_i^t}$ of each prediction.
(3) Given a set of observations $\{M_j^t\}$ detected by the motion detector, match predictions $\{P_i^t\}$ to observations by determining to which observation M_j^t the tracked local features of P_i^t belong.
(4) Any remaining unmatched predictions in $\{P_i^t\}$ are marked as missing observations. Any remaining unmatched observations in $\{M_j^t\}$ are considered to be potential new objects.
(5) Detect any matched predictions that have become temporarily stationary. These are integrated into the background model of the motion detector as a new background layer.
(6) Update the state of matched predictions in $\{P_i^t\}$ using a weighted sum with the associated observations and replace any lost features. The final result is a set of tracked objects $\{O_i^t\}$ at time t. Let $t = t + 1$ and repeat step (1).

In step (3), features are used in matching predictions to their corresponding observations to improve the tracking

robustness in crowded scenes. This is achieved by analysing the spatial and motion information of the features. Spatial rule-based reasoning is applied to detect the presence of merging or splitting foreground regions; in the case of merged objects the motion of the individual features are robustly fitted to (predetermined) motion models to estimate the membership of features to objects. If the motion models are not distinct or unreliable, then the local states of the features are used to update the global states of the merged objects. The spatial rule-based reasoning is described in more detail in Section 2.2.1, while the motion-based segmentation method is described in Section 2.2.2. Section 2.2.3 describes the technique in step (5), for detecting and handling moving objects that become temporarily stationary.

2.2.1. Using spatial information of features

This method is based on the idea that if a feature belongs to object O_i at time $t-1$, then the feature should remain spatially within the foreground region of O_i at time t. A match function is defined which returns the number of tracked features w of prediction P_i^t that reside in the foreground region of observation M_j^t:

$$f(P_i^t, M_j^t) = |\{w : w \in S_{P_i^t},\ w \in M_j^t\}|. \tag{2}$$

In the case of an isolated object, (2) should return a nonzero value for only one (matched) prediction and observation pairing, ideally with $f(P_i^t, M_j^t) = |S_{P_i^t}|$ (i.e., all tracked features reside in the observed foreground region). In practice the match score is rarely this high due to lost or incorrectly tracked features. A table of score values returned by (2) is constructed for all prediction and observation pairs and a rule-based approach is adopted to determine the association between the tracked features from the object predictions and the newly observed foreground regions. These three rules determine whether the object is tracked (one-to-one match between the prediction and observation), split (one-to-many match), or merged (many-to-one match). The ability to recognise these states allows the tracker to explicitly handle complex object interactions, for example, by creating new objects during a split event or predicting object locations during a merged state. The first rule determines the ideal matches in the case of spatially disjoint objects, that is, one-to-one matches between predicted objects and foreground observations:

$$\begin{gathered} f(P_i^t, M_j^t) > 0, \\ f(P_k^t, M_j^t) = 0, \quad f(P_i^t, M_l^t) = 0, \quad \forall k \neq i,\ l \neq j. \end{gathered} \tag{3}$$

The second rule determines the case when an object at time $t-1$ splits into several objects when seen at time t. This occurs when several observation regions match with a single prediction P_i^t—in other words, the set of observations is partitioned into two subsets: the subset $M1$ of observations that match only with P_i^t and the subset of those that do not match with P_i^t:

$$\begin{aligned} &f(P_i^t, M_j^t) > 0, \quad M_j^t \in M1 \subseteq M,\ |M1| > 1, \\ &f(P_k^t, M_j^t) = 0, \quad \forall M_j^t \in M1,\ k \neq i, \\ &f(P_i^t, M_l^t) = 0, \quad \forall M_l^t \notin M1. \end{aligned} \tag{4}$$

Upon recognition of this case the predicted object is split into new objects, one for each of the matched observations in $M1$. The features of the original prediction P_i are assigned to the corresponding new object depending on whether they reside within its observation region or not. In this way, features are maintained throughout an object splitting event. The resulting object with the highest match score is assigned the object ID of the original prediction.

The third matching rule determines whether multiple objects are merging into a single foreground region. This occurs when more than one predicted object matches with an observation region:

$$\begin{aligned} &f(P_i^t, M_j^t) > 0, \quad P_i^t \in P1 \subseteq P,\ |P1| > 1, \\ &f(P_i^t, M_k^t) = 0, \quad \forall P_i^t \in P1,\ k \neq j, \\ &f(P_l^t, M_j^t) = 0, \quad \forall P_l^t \notin P1. \end{aligned} \tag{5}$$

The merged object case demonstrates the benefits of using a local feature-based object tracker in that objects can be tracked during occlusion events, provided that a subset of the original features can be tracked throughout the merged state. In a merged foreground region the state of the individual objects (e.g., position and bounding box) cannot be obtained by a straightforward update from the observation's state, since only one combined (merged) observation is available from the motion detector. Instead, the known local states of the tracked features are used to update the global states of the predictions. The prediction's new centre is estimated by taking the average relative motion of its local features from the previous frame at time $t-1$ to the current one. This is based on the assumption that the average relative motion of the features is approximately equal to the object's global motion—this may not always be true for nonrigid objects undergoing large motion, and may also be affected by the aperture problem due to the small size of the feature windows. The sizes of the bounding boxes of the predictions are also updated in order to maximise the coverage of the observation region by the combined predictions' bounding boxes. This handles cases where objects are moving towards the camera while in a merged state and hence their sizes increase. If not done, the result is parts of the observation region that are not explained by any of the predictions.

2.2.2. Using motion information of features

The motion information obtained from tracking the local features of a prediction P_i is also used in the matching process of step (3). Features belonging to an object should follow approximately the same motion (assuming rigid object

motion). Motion models are fitted to each group of k neighbouring features of P_i. These motion models are then represented as points in a motion parameter space and clustering is performed in this space to find the most significant motion(s) of the object (following the technique presented in [10]). A weighted list is maintained per object of these significant motions and the list is updated over time to reflect changes in the object's motion—if a motion model gains confidence, its weight is increased; if a new motion model is detected, it is added to the list, or replaces an existing lower probable one.

The motion models are used to differentiate the features of merged objects by checking whether a tracked feature belongs to one motion model or the other. This allows tracking through merging/occlusion and the replenishment of lost features by matching them to existing motion models identified. The motion models of an object are further used to identify object splitting events. If a secondary motion becomes significant enough and is present for a long time, it is likely that there may be more than one object contained within the foreground region and splitting is performed. Although the underlying assumption is of rigid object motion, the use of a weighted list of motion models should allow for the identification of the different motions for articulated vehicles. Future work will address this issue. Figure 4 gives an example of the use of weighted motion models.

Two types of motion models have been used for AVITRACK—affine and translational models. The affine motion model is generated by solving for

$$w_t^T F w_{t-N} = 0, \tag{6}$$

where w_t and w_{t-N} are the (homogeneous) location vectors of feature w at time t, $t-N$, and F is the fundamental matrix representing the motion. For the affine case, F has the form

$$F = \begin{bmatrix} 0 & 0 & f_{13} \\ 0 & 0 & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}, \tag{7}$$

F is obtained through a minimisation process based on eigen analysis, as described in [10]. The affine motion model is then represented in terms of 5 motion parameters: $v_{\text{affine}} = \langle \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\theta} \rangle$, where

$$\boldsymbol{\alpha} = \arctan\left(\frac{-f_{13}}{f_{23}}\right), \qquad \boldsymbol{\gamma} = \arctan\left(\frac{f_{31}}{-f_{32}}\right),$$

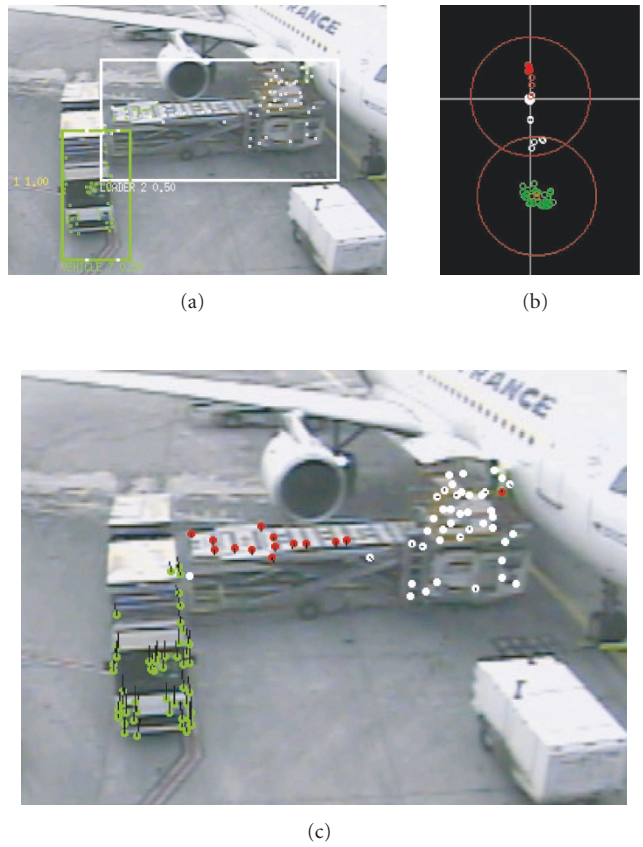
$$\boldsymbol{\rho} = \sqrt{\frac{f_{31}^2 + f_{32}^2}{f_{13}^2 + f_{23}^2}}, \qquad \boldsymbol{\lambda} = \frac{f_{33}}{\sqrt{f_{13}^2 + f_{23}^2}}, \tag{8}$$

$$\boldsymbol{\theta} = \boldsymbol{\alpha} - \boldsymbol{\gamma}.$$

Clustering is performed in the motion parameter space to get the list of most significant motion models for the object. A potential weakness with the clustering approach described in [10] is that the process fits spherical Gaussian models to the motion parameters, which have different scales per dimension. In practice the technique fits the Gaussians to the dense clusters of significant motion in the space, and the result is generally sufficient to be able to determine any significant motions present. Figure 4 shows an example of the clusters and significant motions round in the motion space. The second motion model is simply the translational motion in the image plane:

$$v_{\text{translational}} = w_t - w_{t-N}. \tag{9}$$

When tested on AVITRACK sequences, it was found that perspective and lens distortion effects cause the affine motion models to become highly dispersed in the motion parameter space and clustering performs poorly. The translational model, as can be expected, also suffers from these problems and affine motion effects, but the effect on clustering is less severe. An example is shown in Figure 5 where the two objects are extracted from the merged foreground region using motion clustering with the translational model. This motion



(a) (b) (c)

Figure 4: (a) Sample frame from Dataset S28-Camera 5 showing two merged vehicles: a transporter vehicle moving towards the camera, and a stationary loader vehicle with its platform being raised upwards. (c) Segmentation of the features of the two vehicles by fitting motion models and clustering: the features of the transporter are explained by a single motion model, while those of the loader are explained by two main motion models (a main motion model with weights 0.68 for the stationary loader and a secondary motion model with weight 0.26 for the loader's platform). (b) The motion models as plotted in the motion parameter space and after performing clustering.

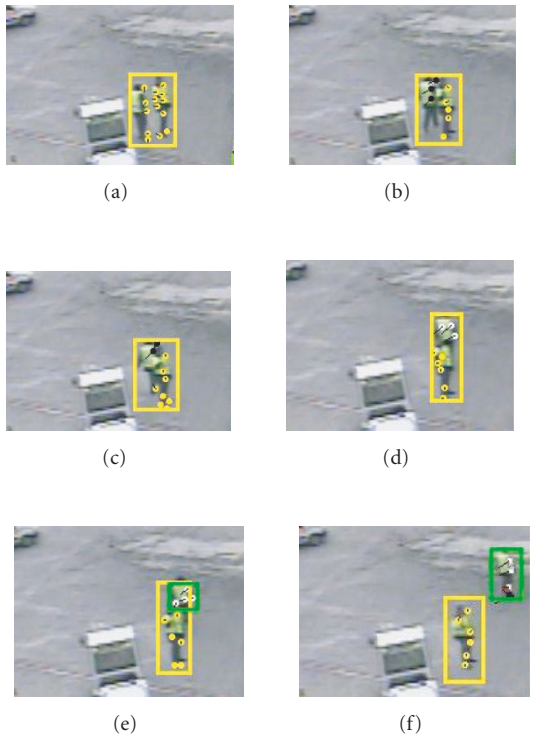


Figure 5: (a) Shows two quasi-stationary persons merged together, with their features highlighted in yellow and explained by a single motion model; (b)–(c) as the person on the left starts moving, the motion of its features (shown by black vectors) creates a secondary motion model with initially low confidence (shown by black circles); (d) confidence in the secondary motion model increases (turning to white circles), until (e) splitting occurs and a new object is created; (f) the two persons are no longer merged.

"fragmentation" for the translational model is mitigated by the use of the weighted list of motion models for each object. At present, the translational model is performing better than the affine model. Future work will look into improving the affine model and using perspective motion models.

2.2.3. Stationary objects

For the apron environment, activity tends to happen in congested areas near the aircraft with several vehicles arriving and stopping for short periods of time in the vicinity of the aircraft, creating occlusions and object merging problems. To allow objects to be differentiated and the tracking of moving objects in front of stopped objects, the motion detection process described in Section 2.1 was extended to include a multiple background layer technique built upon the work presented in [11].

The tracker identifies stopped objects by one of two methods: by analysing object's regions for connected components of foreground pixels which have been labelled as "motion" for a certain time window; or by checking the individual motion of local features of an object. The accuracy of the second method depends on the sparseness of the features, and hence on the density parameter ρ introduced in Section 2.2. Stationary objects are integrated into the motion detector's background model as different background layers.

This technique is similar in spirit to the temporal layers method described by Collins et al. [8], except that their method works on a pixelwise level, using intensity transition profiles of pixels to classify them as "stationary" or "transient." This is then combined with pixel clustering to form moving or stationary regions. This method performed poorly when applied to AVITRACK sequences, mainly due to stationary objects becoming fragmented into many layers as the duration of objects remaining stationary increases. This results in different update rates to the layers and incorrect reactivation once an object starts moving again. In the case of AVITRACK, the aircraft can remain stationary for up to half an hour—it is imperative that the object remains consistent throughout this time, its background layer gets updated uniformly, and it is reactivated as a whole. The method adopted for AVITRACK (based on [11]) works at the region level and is handled by the tracker rather than at the motion detection phase, where the motion information of the local features can provide robust information on an object's motion. This use of region-level analysis helps to reduce the creation of a large number of background layers caused by noise.

The stationary object detection method was improved to take into account cases where the majority of the object is stationary except for a subregion (e.g., a person emerges from a vehicle while it is slowing down to a stop). This relaxation of the stationary object detection criteria allows the handling of partial motion as illustrated in Figure 6.

The relaxation of the stationary object detection criteria, and the use of background layers in general, can result in ghosts (false positives) being detected when part of the background is uncovered. A method based on the movement density, that is, the average change in a region, is used to detect such ghosts. Figure 7 illustrates the use of a multilayered background model to distinguish overlapping objects. The matching of predictions to observations described in Sections 2.2.1 and 2.2.2 then takes into account the interaction that occurs between objects that become temporarily stationary and moving objects.

2.2.4. Object confidence

To improve reasoning in later modules, we introduce an observability confidence measure that the image-plane observation represents the entire object (i.e., it is unoccluded, unclipped, etc.). The object confidence is affected by the observability of the object when it is undergoing occlusion and/or clipping, clipping occurs at the image borders when objects enter/exit the scene. Objects with "low" confidence are partially visible and as such will generally have reduced localisation accuracy as well as related problems such as reduced classification accuracy, and so forth. The resulting confidence value is used to improve the robustness to "low" confidence observations in the data fusion module by reducing the influence of these relative to the more reliable observations.

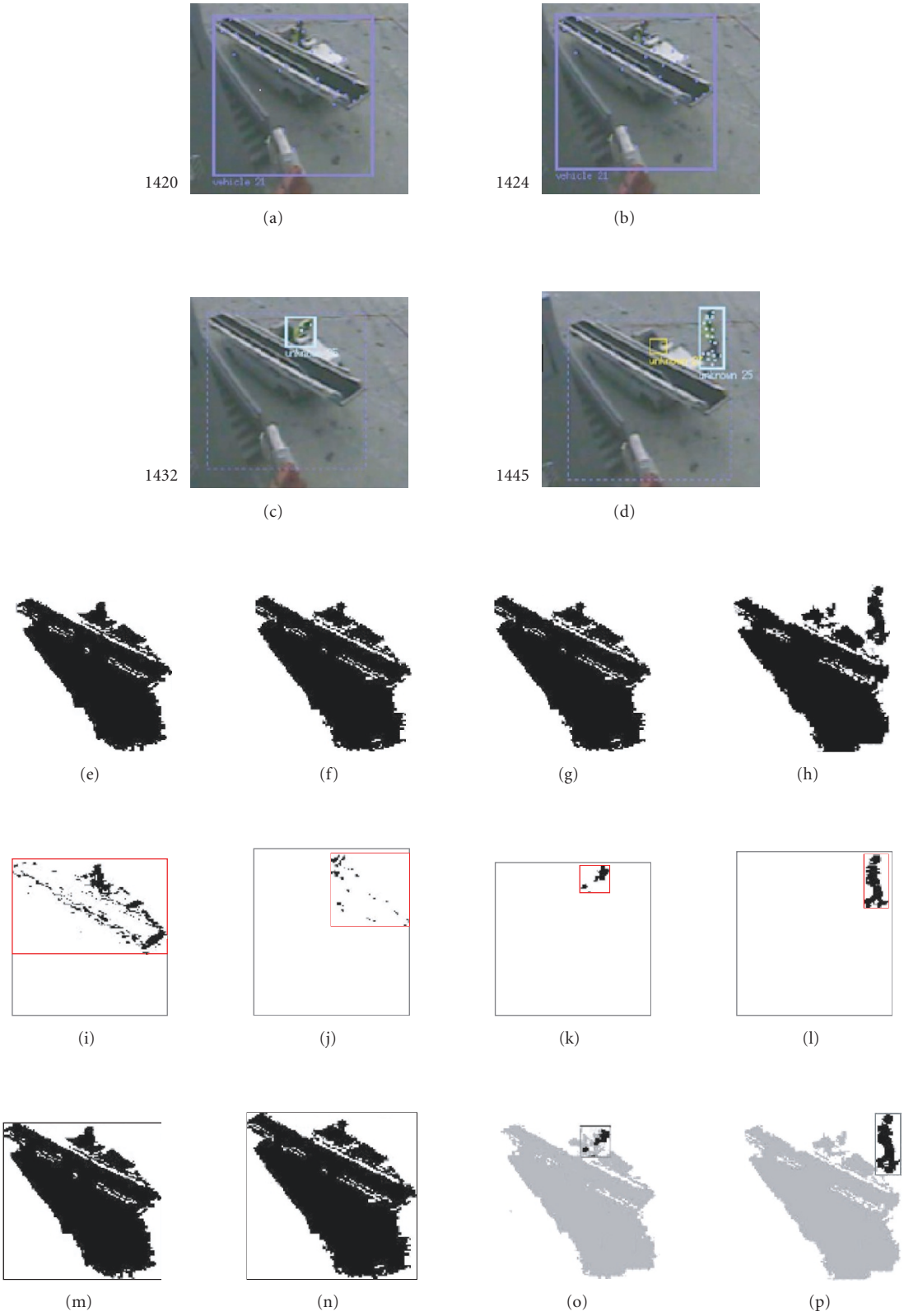
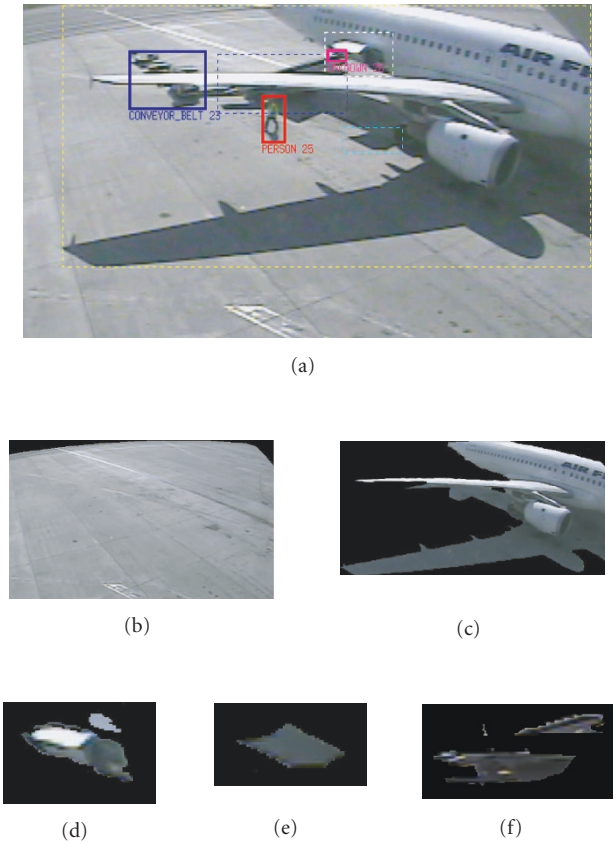


FIGURE 6: (a)–(d) Several frames showing a conveyor belt vehicle coming to rest, while its driver remains in motion and the vehicle exits; (e)–(h) show the pixels labelled as foreground by the motion detector (in black); (i)–(l) the foreground pixels detected as nonstationary are shown in black; (m)–(p) the object's part in motion is shown in black, while the stationary part of the object is shown in grey. In (o), the driver is separated from the vehicle object, but due to problems with the update of the background model, it can be seen that a temporary ghost is created in (d).



(a) (b) (c) (d) (e) (f)

Figure 7: (a) Frame 2352 of sequence S3-A320 showing overlapping stationary and moving objects. The bounding boxes of stationary objects are shown as dotted lines; for moving objects, their bounding boxes consist of solid lines. (b) The basic (full image) background layer. Other background layers (in order of creation) representing stationary objects: (c) the aircraft, (d) aircraft door, (e) aircraft door shadow, and (f) partially visible conveyor-belt vehicle.

The confidence measure for an object $\psi_{O_i} \in [0,1]$ is estimated as $\psi_{O_i} = \psi_o \psi_c$ where ψ_o is the estimated confidence that the object is unoccluded and ψ_c is the estimated confidence that the object is unclipped. If an object is occluded (i.e., in a merged state) $\psi_o = 0.5$, otherwise $\psi_o = 1.0$. ψ_c is estimated using the bounding boxes of the object and the image. If the bounding box of the object is touching or outside the border of the image, then $\psi_c = 0.0$, since the object is likely to be clipped. If the object bounding box edges are no closer than n $(= 25)$ pixels from the image border, then $\psi_c = 1.0$, since the object is likely to be unclipped. Between these two cases a linear ramp function is used to scale ψ_c between 0.0 and 1.0 based on the proximity of the object bounding box edges to the image border.

2.3. Object recognition

To efficiently recognise the people and vehicles on the apron, the key issue faced in recognising such objects is the quantity of potential categories (28) and the interclass similarity (e.g., many vehicles have similar size and appearance). Due to these issues 2D-descriptor-based bottom-up classifiers (e.g., [8]) are difficult to apply, and suffer further due to the requirement of training the classifier on all camera views. In the AVITRACK project a hierarchical classification procedure is applied to use simple 2D descriptors to categorise objects into broad higher-level categories (people, ground vehicle, aircraft, or equipment) and then to use a computationally intensive top-down model-based classifier to categorise the type of ground vehicle. The top-down classifier is only applied if the bottom-up stage recognises the object as a "ground vehicle." This hierarchical combination achieves a balance between efficiency and accuracy not available to the individual classifiers.

As stated previously, the first stage categorises the higher-level types of object that are expected to be found on the apron (people, ground vehicle, aircraft, or equipment). This is achieved using a bottom-up Gaussian mixture model classifier trained on efficient descriptors such as 3D width, 3D height, dispersedness, and aspect ratio; the separation of the higher-level classes is such that the training procedure can be performed globally over all cameras. This efficient classification stage was inspired by the work of Collins et al. [8], where simple descriptors were also applied to categorise distinct classes of object. However, this work did not attempt any finer-level classification (e.g., type of car) where the 2D descriptors would be inadequate.

To improve the finer-level classification of ground vehicles (many of which are similar in size and appearance) a sophisticated (and computationally intensive) top-down classification stage was required. This stage builds upon previous work in top-down model-based tracking [3, 12] to categorise objects by fitting textured 3D models to the detected objects in the scene.

Detailed 3D appearance models were constructed for the vehicles and encoded using the "facet model" description language introduced in [3]. The model fit at a particular world point is evaluated by back-projecting the 3D model into the image and performing normalised cross-correlation (NCC) of the facets' appearance model with the corresponding image locations. To find the best fit for a model, the SIMPLEX algorithm is used to find the pose with best score in the search space, assuming the model's movements are constrained to be on the ground plane. See Figure 8 for an example. The initial pose of the 3D model $(x_0, y_0, \boldsymbol{\theta}_0)$ used to initialise the search is estimated from the centroid of the object (projected on to the ground plane) and its direction of motion. The x, y range in the search space is estimated from the image-plane bounding box of the object when projected on to the ground plane; while the $\boldsymbol{\theta}$ search range is currently restricted to $\boldsymbol{\theta}_0 + / - 15$ degrees.

In the computation of the evaluation score for a model, weighting functions are used to combine the NCC scores $e(j)$ of each visible facet j. The first weight w_a takes into account the angle between the camera's optical axis $\overline{\mathrm{CA}}$ and the facet's normal $\overline{\mathrm{FN}}$; this smooths out any discontinuities in the evaluation surface that arise when facets suddenly come into view as the model is rotated. While the second weight w_b takes into account the facet's visible area compared to the total visible

(a) (b) (c) (d)

FIGURE 8: (a) Frame of sequence S21 showing a transporter vehicle. (b) Edge-based and appearance-based 3D model for the transporter vehicle. (c) The appearance model fitted to the vehicle, with the ground-plane (x, y) search area shown in blue. (d) x-, y-slice of the evaluation score surface in the $(x, y, \boldsymbol{\theta})$ search space.

surface area, where p is a model facet point in (10) below:

$$w_a = \left[\arcsin\left(\frac{\overline{\mathrm{FN}} \cdot \overline{\mathrm{CA}}}{|\overline{\mathrm{FN}}||\overline{\mathrm{CA}}|} \right) \frac{2}{\pi} \right]^{11}, \tag{10}$$

$$w_b = \frac{\sum_{\forall p} \text{visible}\,(p) = 1}{\sum_{\forall p} 1}. \tag{11}$$

The final 3D model evaluation score e is then given by

$$e = \frac{\sum_{\text{facet } j} [e(j) w_a(j) w_b(j)]}{\sum_{\text{facet } j} w_a(j) w_b(j)}. \tag{12}$$

While 3D model fitting performs quite well with few false matches, it is computationally intensive; this is solved by running the algorithm on a background (threaded) process to the main (bottom-up) tracking system and updating the object classification when it is available; a processing queue is used to synchronise the two methods together. For apron monitoring the subtype category only becomes important when a vehicle enters specific spatial zones near the aircraft; the time between a vehicle entering the scene and entering such a zone is generally adequate to perform model-based categorisation at least once for each object. Running the classifier as a background process means that the object location and orientation are measured for a *previous* frame, thus creating a latency in object localisation—this is a compromise required to achieve real-time performance. This problem is corrected by applying an efficient object localisation strategy described in the following section.

The computational performance of the object recognition module was profiled for the test sequence *S21-Camera 7*, containing people and vehicles interacting on the apron area. Many of the modules (including categorisation) are threaded, therefore the performance was measured at the scene tracking module level to give meaningful timing information (including colour mean and variance motion detection and KLT-based tracking). During the tests the system was configured to read and write using the hard drive (as opposed to network-based communication) and had the visualisation/GUI activated. Whilst this is not the system in the fastest configuration, relative comparison of the different classifier performance is still valid. The tests were performed using a desktop workstation with 2×3 GHz pentium-4 processor and 2 Gb RAM, running Suse Linux 9.1.

It was found that using the bottom-up classifier alone achieved a scene tracking module average frame rate of 6.88 fps ± 3.73. The top-down classifier alone achieved an average frame rate of 6.058 fps ± 3.78. Finally, the hierarchical classifier achieved an average frame rate of 6.36 fps ± 3.74. These results demonstrate that the computational performance of the hierarchical classifier lies between the faster bottom-up and slower top-down classifiers while retaining the classification ability of both classifiers. Section 3.3 presents categorisation performance results for the object recognition module.

2.4. Object localisation

The localisation of an object in the context of visual surveillance generally relates to finding a location in the world coordinates that is most representative of that object. This is commonly taken to be the centre of gravity of the object on the ground plane and it is this definition that we adopt here. With accurate classification and detection, the localisation of vehicles in the 3D world can be reduced to a 2D geometrical problem. For state-of-the-art algorithms accurate classification and detection is not reliable enough to apply such principled methods with confidence. For the AVITRACK project we therefore devised a simple, but effective, vehicle localisation strategy that gives adequate performance over a wide range of conditions. The cameras are spatially registered using coplanar calibration to define common "world" coordinates, this allows image coordinates to be mapped to ground-plane (i.e., $z = 0$) locations in world coordinates.

The first step of the strategy is to use the object recognition result to categorise the detected objects as person or nonperson. The motivation behind this is that people generally have a negligible depth compared to vehicles and hence a different strategy is required to locate each type. For the person class of objects the location is taken to be the bottom centre of the bounding box of the detected object, this location estimate for people is commonplace in visual surveillance systems.

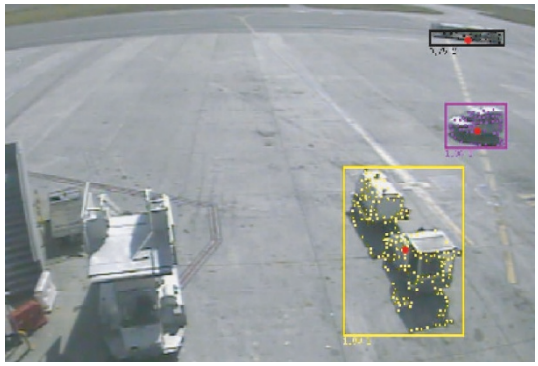


Figure 9: Detected object locations (red circles) shown for 3 vehicles in the near-, mid-, and far-field of Camera 5 for sequence S4.

For vehicles localisation many researchers arbitrarily choose the centroid of the bounding box or detected foreground pixels to locate the object in the world. This method has the drawback that for objects further away from the camera the bottom of the bounding box is a better approximation of the object location than the centroid. To alleviate this problem we compute the angle made between the camera and the object to estimate an improved location. For a camera lying on the ground plane the location of the object will be reasonably proximal to the bottom centre of the bounding box, whereas for an object viewed directly overhead the location of the object will be closer to the measured centre of the bounding box.

Using this observation we formulated a smooth function to estimate the position of the centroid using the (2D) angle to the object. Taking α to be the angle measured between the camera and the object, the proportion p of the vertical bounding box height (where $0 \leq p \leq 1/2$) was estimated as $p = 1/2(1 - \exp(-\lambda a))$, the parameter λ was determined experimentally ($\equiv \ln(2)/(0.15 \times 1/2\pi)$) to provide good performance over a range of test data. The vertical estimate of the object location was therefore taken to be $y_{lo} + (p \times h)$ where y_{lo} is the bottom edge of the bounding box and h is the height of the bounding box. The horizontal estimate of the object location was measured as the horizontal centre line of the bounding box, since this is generally a reasonable estimate. Examples of estimated vehicle centroids are shown in Figure 9, it can be seen that the estimate is closer to the actual object location than simply using the centroid of the bounding box. In practice this localisation is adequate for many vehicle types, however, for elongated (i.e., long/tall) vehicles the localisation strategy may be less accurate. The measurement noise in the data fusion module (detailed in the next section) can be increased to account for this possible inaccuracy, allowing greater uncertainty in the localisation of the objects to improve the association of such vehicles.

2.5. Data fusion

The method applied for data fusion is based on a discrete nearest neighbour Kalman filter approach [1] with a constant velocity model. The main challenge in apron monitoring relates to the matching of tracks to observations in crowded regions, which require extra descriptors to be applied to differentiate the different objects. This problem is not solved by a probabilistic filter; therefore, the simpler deterministic filter is sufficient as the basis for the proposed algorithm. The (synchronised) cameras are spatially registered using coplanar calibration to define common "world" coordinates.

The data association step associates existing track predictions with the per camera measurements. In the nearest neighbour filter the nearest match within a validation gate is determined to be the sole observation for a given camera. For multiple tracks viewed from multiple sensors the nearest neighbour filter is as follows.

(1) For each track, obtain the validated set of measurements per camera.
(2) For each track, associate the nearest neighbour per camera.
(3) Fuse associated measurements into a single measurement using intersection of the measurement uncertainties.
(4) Kalman filter update of each track state with the fused measurement.
(5) Intersensor association of remaining measurements to form candidate tracks.

The validated set of measurements are extracted using a validation gate [1]; this is applied to limit the potential matches between existing tracks and observations. In previous tracking work the gate generally represents the uncertainty in the spatial location of the object; in apron analysis this strategy often fails when large and small objects are interacting in close proximity on the congested apron, the uncertainty of the measurement is greater for larger objects; hence, using spatial proximity alone larger objects can often be misassociated with the small tracks. To circumvent this problem we have extended the validation gate to incorporate velocity and category information, allowing greater discrimination when associating tracks and observations.

The observed measurement is a 7D vector:

$$\mathbf{Z} = \left[x, y, \dot{x}, \dot{y}, P(p), P(v), P(a)\right]^T, \tag{13}$$

where $P(\cdot)$ is the probability estimate that the object is one of three main taxonomic categories (p = person, v = vehicle, a = aircraft). This extended gate allows objects to be validated based on spatial location, motion, and category, which improves the accuracy in congested apron regions. The effective volume of the gate is determined by a threshold τ on the normalised innovation squared distance between the predicted track states and the observed measurements:

$$d_t^2(i,j) = \left[\mathbf{H}\hat{\mathbf{X}}_t^-(i) - \mathbf{Z}_t(j)\right]^T \mathbf{S}_t^{-1} \left[\mathbf{H}\hat{\mathbf{X}}_t^-(i) - \mathbf{Z}_t(j)\right], \tag{14}$$

where $\mathbf{S}_t = \mathbf{H}\hat{\mathbf{P}}_t^-(i)\mathbf{H}^T + \mathbf{R}_t(j)$ is the innovation covariance between the track and the measurement; this takes the

form

$$\mathbf{S}_t = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & 0 & 0 & 0 & 0 & 0 \\ \sigma_{yx} & \sigma_y^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}}^2 & \sigma_{\dot{x}\dot{y}} & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{y}\dot{x}} & \sigma_{\dot{y}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{P(p)}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{P(v)}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{P(a)}^2 \end{bmatrix}. \quad (15)$$

For the kinematic terms the predicted state uncertainty $\widehat{\mathbf{P}}_t^-$ is taken from the Kalman filter and constant a priori estimates are used for the probability terms. Similarly, the measurement noise covariance $\mathbf{R}$ is estimated for the kinematic terms by propagating a nominal image plane uncertainty into the world coordinate system using the method presented in [13]. Measurement noise for the probability terms is determined a priori. An appropriate gate threshold can be determined from tables of the chi-square distribution [1].

Matched observations are combined to find the fused estimate of the object, this is achieved using covariance intersection. This method estimates the fused uncertainty $\mathbf{R}_f$ for N_i matched observations as a weighted summation:

$$\mathbf{R}_f = \left(w_1\mathbf{R}_1^{-1} + \cdots + w_{N_i}\mathbf{R}_{N_i}^{-1}\right)^{-1}, \quad (16)$$

where $w_m = w'_m / \sum_{n=1}^{N_i} w'_n$ and $w'_m = \psi_m^c$ are the confidence of the mth associated observation (made by camera c) estimated using the method in Section 2.2. The measurement vector $\mathbf{Z}_f$ of the fused estimate is computed as [14]

$$\mathbf{Z}_f = \mathbf{R}_f\left(w_1\mathbf{R}_1^{-1}\mathbf{Z}_1 + \cdots + w_{N_i}\mathbf{R}_{N_i}^{-1}\mathbf{Z}_{N_i}\right). \quad (17)$$

The fused estimate ground-plane location (x, y) is retained from the vector $\mathbf{Z}_f$. This location estimate is used to update the state vector (containing location and velocity) of the associated track using the Kalman filter. The overall confidence of the fused object is estimated as $(\sum_{n=1}^{N_i} w'_n)/N_{x,y}$ where $N_{x,y}$ is the number of cameras that contain the ground-plane location (x, y) in their respective image planes. In lieu of an explicit scene model this confidence is premultiplied by a scaling factor to account for the fact that not all cameras have unoccluded views of the ground-plane location.

To estimate the observed category information for each tracked object the category estimates for all associated observations are averaged, weighted by the confidence of the observation. The estimated category information for each object is filtered in an α-β IIR filter of the form $E^+ = \alpha E^- + (1 - \alpha)F$ where E^- is the previous estimate of the category vector, F is the category vector estimated from the associated observations, and E^+ is the updated (filtered) estimated category vector.

If tracks are not associated using the extended validation gate, the requirements are relaxed such that objects with inaccurate velocity or category measurements can still be associated. Remaining unassociated measurements are fused into new tracks, using a validation gate between observations to constrain the association and fusion steps. Ghost tracks without supporting observations are terminated after a predetermined period of time (during which the track state is estimated using the Kalman filter prediction). To track objects that cannot be located on the ground plane we have extended the tracker to perform epipolar data association (based on the method presented in [13]).

The data fusion module is followed in the AVITRACK system by a scene coherency module (see Figure 2). This module uses spatial and temporal reasoning to link the identities of different tracks that represent the same physical object. A physical object may be represented by more than one track due to tracking errors caused by spatial fragmentation, occlusion, or poor camera coverage, etc. The output coherent tracks from this module are subsequently used as input to the scene understanding module.

3. EXPERIMENTAL RESULTS

The evaluation methodology characterises the performance of the subcomponents of the scene tracking module. This evaluation is performed on a set of representative test data, the evaluation of the components strongly depends on the choice of the video sequences. We have chosen video datasets containing realistic conditions for an objective evaluation.

The evaluation of the scene tracking module is organised as follows: the motion detection module evaluation procedure and results are presented in Section 3.1. The object tracking module is evaluated in Section 3.2. The object recognition procedure is evaluated in Section 3.3. The accuracy of the object localisation is presented in Section 3.4, and finally the data fusion module is evaluated in Section 3.5.

3.1. Motion detection results

The evaluation of motion detection is performed using the methodology presented by Aguilera et al. [15], this is based on the object-level evaluation methodology of Correia and Pereira [16]. The quality of motion segmentation can in principle be described by two characteristics. Namely, the spatial deviation from the reference segmentation, and the fluctuation of spatial deviation over time. In this evaluation, however, we concentrate on the evaluation of spatial segmentation characteristics. That is, we will investigate the capability of the error metrics listed below to describe the spatial accuracy of motion segmentations.

(i) False detection rates

The normalised false negative rate (fnr) and false positive rate (fpr) metrics are based on pixelwise mismatches between ground truth and observations in a frame [17],

$$\text{fnr} = \frac{N_{\text{FN}}}{N_{\text{TP}} + N_{\text{FN}}}, \qquad \text{fpr} = \frac{N_{\text{FP}}}{N_{\text{FP}} + N_{\text{TN}}}, \quad (18)$$

where N_{FN} and N_{FP} denote the number of false negative and false positive pixels, respectively. N_{TN} and N_{TP} are the number of true negatives and true positives.

(ii) Misclassification penalty

The obtained segmentation is compared to the reference mask on an object-by-object basis; misclassified pixels are penalised by their distances from the reference objects border [18],

$$\mathrm{MP} = \mathrm{MP}_{\mathrm{FN}} + \mathrm{MP}_{\mathrm{FP}} \tag{19}$$

with

$$\mathrm{MP}_{\mathrm{FN}} = \frac{\sum_{j=1}^{N_{\mathrm{FN}}} d_{\mathrm{FN}}^{j}}{D}, \qquad \mathrm{MP}_{\mathrm{FP}} = \frac{\sum_{k=1}^{N_{\mathrm{FP}}} d_{\mathrm{FP}}^{k}}{D}. \tag{20}$$

Here, d_{FN}^{j} and d_{FP}^{k} stand for the distances of the jth false negative and kth false positive pixels from the contour of the reference segmentation. The normalised factor D is the sum of all pixel-to-contour distances in a frame.

(iii) Rate of misclassifications

The average normalised distance of detection errors from the contour of a reference object is calculated using [19]

$$\mathrm{RM} = \mathrm{RM}_{\mathrm{FN}} + \mathrm{RM}_{\mathrm{FP}} \tag{21}$$

with

$$\mathrm{RM}_{\mathrm{FN}} = \frac{1}{N_{\mathrm{FN}}} \sum_{j=1}^{N_{\mathrm{FN}}} \frac{d_{\mathrm{FN}}^{j}}{D_{\mathrm{diag}}}, \qquad \mathrm{RM}_{\mathrm{FP}} = \frac{1}{N_{\mathrm{FP}}} \sum_{k=1}^{N_{\mathrm{FP}}} \frac{d_{\mathrm{FP}}^{k}}{D_{\mathrm{diag}}}. \tag{22}$$

N_{FN} and N_{FP} denote the number of false negative and false positive pixels, respectively. D_{diag} is the diagonal distance within the frame.

(iv) Weighted quality measure

This measure quantifies the spatial discrepancy between estimated and reference segmentation as the sum of weighted effects of false positive and false negative pixels [20],

$$\mathrm{QMS} = \mathrm{QMS}_{\mathrm{FN}} + \mathrm{QMS}_{\mathrm{FP}} \tag{23}$$

with

$$\mathrm{QMS}_{\mathrm{FN}} = \frac{1}{N} \sum_{j=1}^{N_{\mathrm{FN}}} w_{\mathrm{FN}}(d_{\mathrm{FN}}^{j}) d_{\mathrm{FN}}^{j}, \qquad \mathrm{QMS}_{\mathrm{FP}} = \frac{1}{N} \sum_{k=1}^{N_{\mathrm{FP}}} w_{\mathrm{FP}}(d_{\mathrm{FP}}^{k}) d_{\mathrm{FP}}^{k}, \tag{24}$$

N is the area of the reference object in pixels. Following the argument that the visual importance of false positives and false negatives is not the same, and thus they should be treated differently, the weighting functions w_{FP} and w_{FN} were introduced:

$$w_{\mathrm{FP}}(d_{\mathrm{FP}}) = B_1 + \frac{B_2}{d_{\mathrm{FP}} + B_3}, \qquad w_{\mathrm{FN}}(d_{\mathrm{FN}}) = C \cdot d_{\mathrm{FN}}. \tag{25}$$

In our work, we used the parameters $B_1 = 19$, $B_2 = -178.125$, $B_3 = 9.375$, and $C = 2$, resulting in the weighting functions shown in Figure 10. One can see that missing (false negative) pixels gain more importance with increasing distance than added foreground pixels. Thus, our weighting favours algorithms which provide larger foreground estimates over more conservative ones. Naturally, the choice of weighting functions depends on the targeted application. See [21, 22] for examples.

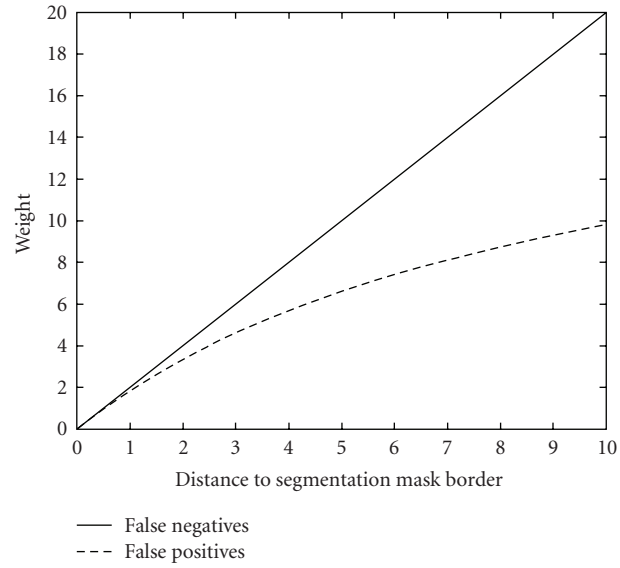


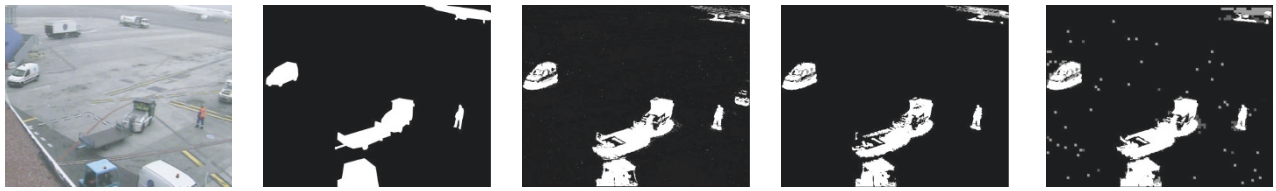
Figure 10: Weighting functions for false positives and false negatives.

We evaluate three different motion detection algorithms on airport's apron datasets using the presented methodology and metrics. The algorithms used in the evaluation are the colour and edge fusion (CEF) [5], mixture of Gaussians (MoG) [9], and colour mean and variance (CMV) [6]. All these algorithms are based on the background subtraction method for moving object detection.

Representative apron sequences acquired under a wide range of disturbing conditions have been chosen. The sequences are as follows.

S3-Camera 2

The sequence shows an aircraft parking on the apron. Moreover it contains individuals and vehicles such as conveyor belts, transporters with dollies, and a stair vehicle working on maintenance tasks. Strong shadows, occlusions, and illumination changes are presented in the scene.



(a) (b) (c) (d) (e)

Figure 11: (Clockwise from top-left) the manually created ground truth and the detection results for the CEF, MoG, and CMV algorithms for frame 5814 taken from Dataset S21-Camera 7. The apparent increased noise (grey blocks, representing highlights and/or shadow) in the CMV result is due to the multiresolution implementation of this background subtraction algorithm.

S4-Camera 5

A tanker and a service vehicle move across the apron. A ground power unit (GPU) parks in the maintenance area and a person leaves the GPU. The sequence presents strong shadows and illumination changes.

S5-Camera 5

Three individuals walk around the apron while a transporter and a GPU park in the maintenance area. The sequence contains reflections either caused by liquid on the ground or by the paint on the ground.

S8-Camera 6

A GPU enters in the scene and two individuals walk on the apron. The sequence presents in close-up a transporter with dollies in movement. As a night sequence, the vehicle lamps produce large reflections on the ground.

S21-Camera 7

The sequence contains individuals walking on the apron. Vehicles in movement such as a GPU, a tanker, a catering vehicle, and service vehicles are shown. Shadows are presented in the scene.

S26-Camera 6

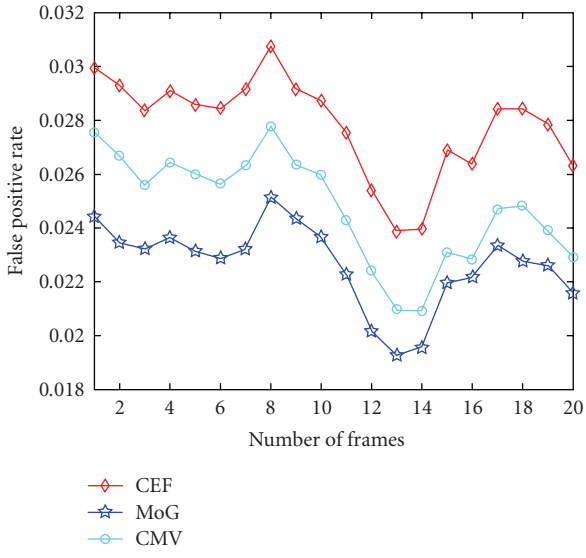
A group of individuals walking and a conveyo belt in movement are shown. An aircraft starts its departure. The scene contains shadows.

Sequences S3, S4, and S5 were acquired on a sunny day. S8 is a night sequence whereas both S21 and S26 include the presence of fog. All of the sequences are stored at a size of 720×576 pixels, and at a frame rate of 12.5. From each of the sequences has been extracted a subsequence of twenty frames in length. A total of six datasets are used in the evaluation for which ground truth motion images have been manually generated for the twenty frames.

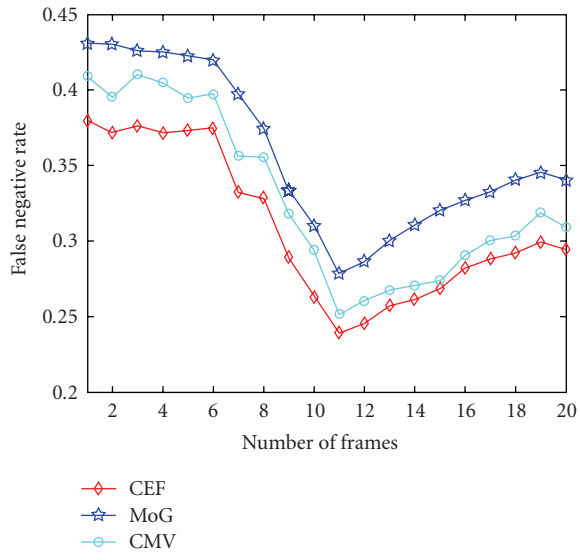
In Figure 11 a sample image from Dataset S21, showing five moving objects (aircraft, car, transporter, GPU, and a pedestrian), is given. Results of the segmentation process are shown in Figure 11. All motion detectors are robust against illumination changes. Strong shadows are detected as part of the mobile objects, and fragmentation is present in some objects (e.g., the aircraft) due to appearance similarity between background and foreground objects. It is noted that the CEF algorithm generates some false positive detections around the stationary vehicle close to the pedestrian due to sensitivity of the gradient estimatation to noise, preventing the whole vehicle being detected as a stationary object.

At first fpr and fnr (error rates, ER) were calculated for the ground truth frames in Dataset S21. It is desirable to have a false positive/negative rate approaching 0%. The results of this evaluation are given in Figure 12. The motion detectors present a false negative rate between 38% and 44%, which is higher than expected. The high false negative rate appears to be due to the similarity in appearance between the background and foreground objects (especially when the foreground objects are not in direct sunlight, which increases the similarity). Both the false positive and false negative rates appear to be in the same order of magnitude for the three algorithms, which confirms the visual similarity observed in the results in Figure 11.

In addition, the weighted quality measure QMS, the misclassification penalty MP, and the rate of misclassifications RM were computed separately for each object in Dataset S21 (see Figure 13). We computed the overall object-based segmentation quality as an average of the individual object's segmentation errors. At frame one three moving objects are in the scene (aircraft, car, and transporter). A GPU and a pedestrian enter in the scene after five and eight frames, respectively. Such objects produce lower individual QMS and MP error than the aircraft, car, or transporter segmentation errors (see Figures 13(d), 13(e), 13(f) and 13(g), 13(h), 13(i)). This is reflected in Figures 13(a) and 13(b) which show the decrease in overall QMS and MP at frames five and eight. For the selected dataset the evaluation of the rate of misclassifications RM (see Figure 13(c)) provides less stable results. This can be explained by its sensitivity with respect to certain types of segmentation errors. The RM computes the average distance of misclassified pixels from the reference object's contour. Therefore, already a small number of erroneous pixels can produce a relatively high error rate. The MP error metric generates a considerable larger segmentation error for the transporter than in the other objects (see Figures 13(g), 13(h), 13(i)). This is due to the fact that the transporter produces a large false negative/positive error compared to the



(a)



(b)

Figure 12: (a) false positive and (b) false negative error rates versus number of frames for Dataset S21.

other object's size. It can be explained by its penalisation of misclassified pixels with respect to their distances from the reference object border.

The performance results of the motion detectors on all tested datasets are presented in Table 1. False positives and negatives of QMS, MP, and RM error metrics were computed for each object and averaged per frame. Furthermore, fpr and fnr (ER) were calculated for whole frames. CMV produces the best false negative error results (for all metrics) on sequences S4-Cam5, S5-Cam5, S8-Cam6 (not for RM) and S26-Cam6 (not for fnr). All motion detectors provide similar results in front of illumination changes and shadows (see S3-Cam2 and S4-Cam5 results). A high amount of false negatives is produced by the motion detectors on the night sequence S8-Cam6. CMV and MoG produce the best false positive error results on the sequences with the presence of fog (see S21-Cam7 and S26-Cam6 results).

In conclusion, the CMV algorithm was found to give the "best" results for the selected datasets and performance metrics. The CMV motion detection algorithm is therefore selected as the input module to the object tracking component.

3.2. Object tracking results

To evaluate the performance of the local feature tracking method two apron datasets were chosen, both were acquired under a range of disturbing conditions. The first sequence is *S21-Camera 7:* (2400 frames), this was used in the motion detection evaluation and contains the presence of fog. The second sequence is the following.

S28-Camera 5

(1200 frames) a crowded scene containing many objects interacting within close proximity near the aircraft, this sequence was acquired on a sunny day.

The datasets have been manually annotated using ViPER annotation tool [4]. ViPER (video performance evaluation resource) is a semiautomatic framework designed to facilitate and accelerate the creation of ground truth image sequences and evaluate performance of algorithms. The ViPER's performance evaluation tool has been used to compare the result data of the local feature tracking method with the ground truth in order to generate data describing the success or failure of the performance analysis. At first, the evaluation tool attempts to match tracked objects (TO) to ground truth objects (GTO) counting objects as matches when the following metric distance is less than a given threshold,

$$D_i(t,g) = \frac{1 - 2\,\text{Area}\,(t_i \wedge g_i)}{\text{Area}\,(t_i) + \text{Area}\,(g_i)}, \tag{26}$$

where t_i and g_i define the bounding box of the tracked objects and ground truth objects at frame i, respectively. Once the tracked and ground truth objects have been matched true positives, false negative and false positive objects are counted and summed up over the chosen frames. The following metrics defined by Black et al. [23] were used to characterise the tracking performance:

(i) *tracker detection rate (TRDR)*: $\text{TP}_t/(\text{TP}_t + \text{FN}_t)$;
(ii) *false alarm rate (FAR)*: $\text{FP}_t/(\text{TP}_t + \text{FP}_t)$;
(iii) *track detection rate (TDR)*: $\text{TP}_o/(\text{TP}_o + \text{FN}_o)$;
(iv) *track fragmentation (TF)*: number of TO matched to GTO.

Where TP, FN, and FP are either the total number t or the number for object o of true positives, false negatives, and false positives, respectively. The TRDR and the FAR metrics characterise the performance of the tracker. The TDR metric determines the completeness of individual ground truth objects. The TF metric determines the number of object label changes. It is desiderable to have a TF value of one.

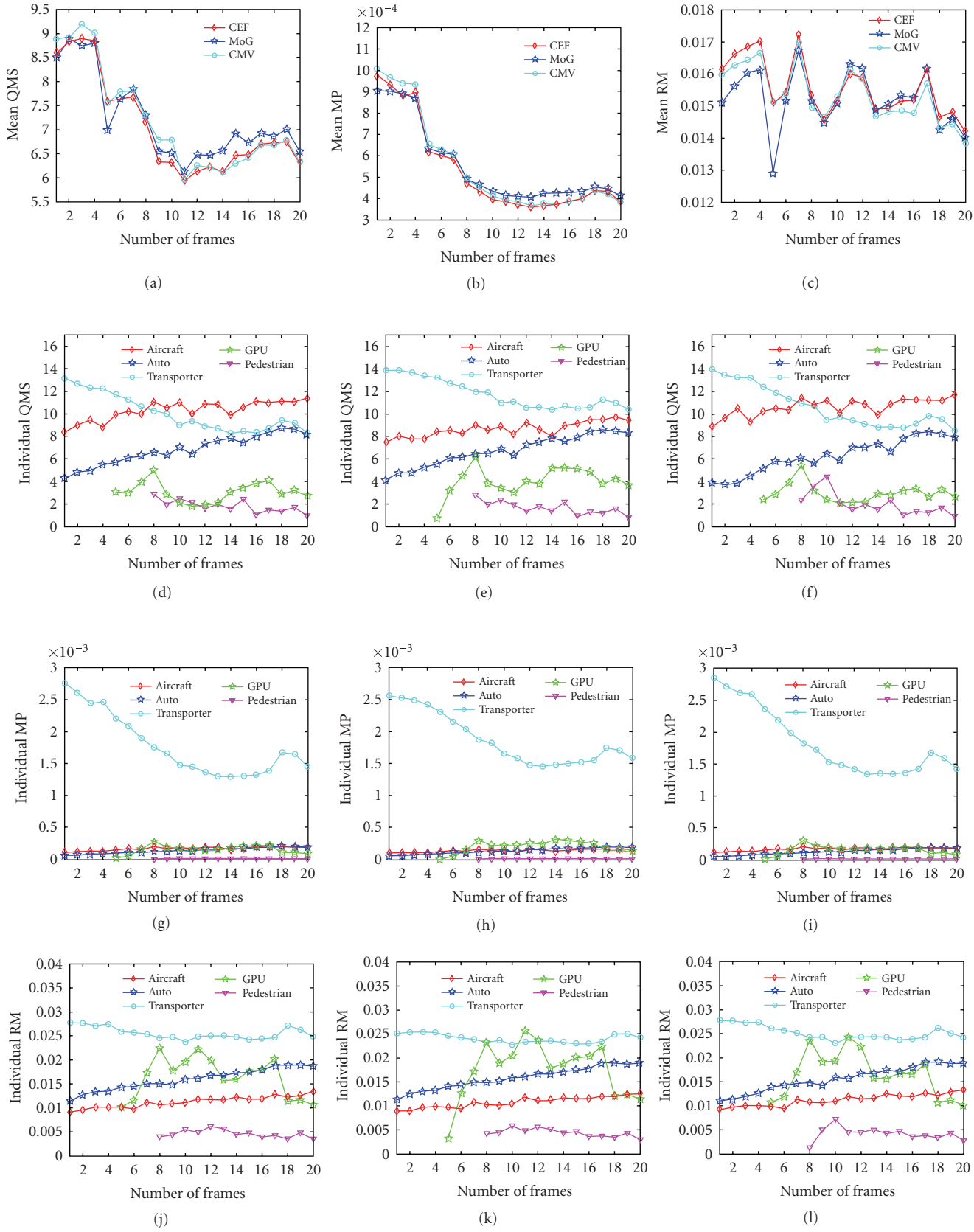


FIGURE 13: Average and individual object segmentation errors for Dataset S21; (a), (b), (c) average QMS, MP, and RM object-based segmentation errors of the motion detectors; (d), (e), (f) QMS individual object segmentation error of CEF, MoG, and CMV; (g), (h), (i) MP individual object segmentation error of CEF, MoG, and CMV; (j), (k), (l) RM individual object segmentation error of CEF, MoG, and CMV.

Table 1: Average per-frame performance results of the motion detection algorithms for all apron datasets.

	S3-Cam2		S4-Cam5		S5-Cam5		S8-Cam6		S21-Cam7		S26-Cam6	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
QMS-CEF	5.517	1.576	4.674	1.324	4.636	0.445	1.898	10.74	12.37	4.501	1.45	0.849
QMS-MoG	4.653	2.888	4.621	1.407	4.534	0.551	1.901	16.49	2.454	6.237	0.716	2.089
QMS-CMV	4.843	2.026	4.632	1.199	4.905	0.392	3.28	6.61	2.902	5.727	0.527	1.182
MP-CEF	6.3e-4	5.5e-5	4.44e-4	4.5e-5	4.4e-5	2.3e-6	1.1e-3	2.1e-3	3.7e-4	3.6e-4	4.5e-5	1.4e-5
MP-MoG	5.93e-4	9.7e-5	4.41e-4	4.8e-5	5.3e-5	2.9e-6	4.5e-4	3.7e-3	2.6e-4	3.7e-4	2.8e-5	3.3e-5
MP-CMV	5.95e-4	6.8e-5	4.43e-4	3.9e-5	5.9e-5	2.1e-6	2.1e-3	1.4e-3	3.1e-4	3.3e-4	1.6e-5	1.3e-5
RM-CEF	4.53e-3	1.06e-2	5.4e-3	1.13e-2	2.44e-3	8.1e-3	1.63e-2	0.013	8.8e-3	8.1e-3	3.7e-3	3.8e-3
RM-MoG	4.94e-3	1.11e-2	5.3e-3	1.15e-2	2.43e-3	8.2e-3	1.75e-2	0.011	8.5e-3	6.6e-3	3.6e-3	3.2e-3
RM-CMV	4.51e-3	1.09e-2	5.1e-3	1e-2	2.16e-3	8e-3	1.62e-2	0.017	8.2e-3	0.011	1.8e-3	2.7e-3
ER-CEF	1.72e-2	0.171	1.98e-2	0.131	7.2e-3	0.076	0.031	0.375	0.032	0.232	0.01	0.09
ER-MoG	1.57e-2	0.276	1.91e-2	0.139	5.7e-3	0.092	0.03	0.564	0.018	0.375	5.2e-3	0.228
ER-CMV	1.58e-2	0.213	1.93e-2	0.121	6.5e-3	0.075	0.046	0.243	0.021	0.325	2.5e-3	0.184

(a) (b)

Figure 14: The results obtained from the local feature-based tracking algorithm. Image (a) has been chosen from S21-Cam7 and image (b) from S28-Cam5.

Representative results of the local feature tracking method are presented in Figure 14. Strong shadows are detected and tracked as part of the mobile objects such as the tanker from S21-Cam7 and the transporter with containers from S28-Cam5 (see Figures 14(a), 14(b)). In Figure 14(a) a person (at the bottom on the right side) leaves the ground power unit (GPU) and in Figure 14(b) a container is unloaded from the aircraft. Both objects produce a ghost which remains behind the previous object position. If an object is stationary for an extended period of time it is deemed to be part of the static scene model and the background layer is flattened (i.e., merged) with the lowest level background layer. This operation is performed to prevent the build-up of large numbers of objects in the layered background representation over extended time periods, which would increase the likelihood of incorrect object reactivation. When such objects start to move again, ghosts are created when the background behind the moving object becomes uncovered. Furthermore, ghosts are produced when parts of the background start moving (e.g., objects in the scene when the tracking system is initialised). Objects in the scene such as the container from Figure 14(b) are partially detected due to the similarity in appearance between the background and foreground objects.

At first, the track detection rate TDR and the track fragmentation TF were computed separately for each ground truth object. The results of the performance evaluation are given in Table 2 for sequence S21-Cam7 (eighteen GTO) and in Table 3 for S28-Cam5 (eight GTO). Two ground truth objects were not matched to tracked objects (see Table 2, object 17 and 18). These two objects were partially detected due to their colour similarity with the background. Most of the objects from sequence S21-Cam7 present a track detection rate between 92% and 99%. All ground truth objects from sequence S28-Cam5 (see Table 3) have been matched to tracked objects. S28-Cam5 also contains several dynamic occlusions causing tracked object label changes, this increases

Table 2: Individual object performance results for the local feature tracking algorithm for S21-Cam7.

Object	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
TP	333	94	33	426	944	166	391	77	125	108	143	209	116	124	113	33	0	0
FN	3	5	10	19	2	6	32	6	32	7	6	4	9	7	3	10	310	65
TDR	0.99	0.95	0.83	0.96	0.99	0.95	0.92	0.93	0.80	0.94	0.96	0.98	0.93	0.95	0.97	0.77	0	0
TF	1	1	1	1	1	1	3	1	2	1	1	1	1	1	3	1	0	0

Table 3: Individual object performance results for the local feature tracking algorithm for S28-Cam5.

Object	1	2	3	4	5	6	7	8
TP	289	551	827	601	274	200	207	72
FN	0	17	10	6	54	10	11	0
TDR	1.00	0.97	0.99	0.99	0.83	0.95	0.95	1.00
TF	3	2	3	2	3	3	1	1

Table 4: Performance results of the local feature tracker.

Dataset	TP	FP	FN	TRDR	FAR
S21-Cam7	3435	275	536	0.87	0.07
S28-Cam5	3021	588	108	0.97	0.16

the track fragmentation rate TF in Table 3 since more than one tracked object is found for the ground-truth objects 1–6.

In addition, the tracker detection rate TRDR and the false alarm rate FAR were calculated for whole frames. The results of this evaluation are given in Table 4. The presence of fog in S21-Cam7 together with the similarity in object appearances cause a considerable number of false negatives provoking the decrease in TRDR (87%). S28-Cam5 contains ghosts and reflections causing the increase in FAR (16%).

3.2.1. General tracking results

To review the generality of the motion detection and tracking algorithms (Sections 2.1 and 2.2) we applied the per camera object tracker to other visual surveillance domains. The result of this can be seen in Figure 15 where representative tracking results for five test sequences are shown. The results for the standard PETS 2001 sequences (a) are encouraging, although the loss of object identity in severe occlusions is noted. The ADVISOR sequence (c)—containing poor quality underground station footage—can be compared to the results obtained by Siebel and Maybank [11]. The SAFEE sequence (d) presents a different challenge, with an object in the near-field of the camera. For this sequence the object lost identity when it became occluded behind one of the seats, also, it is possible to see the problem of reflection in the right-hand image. For this sequence contextual information is required to keep track of the object. Finally, the traffic sequence (e) applies the tracker to the problem of tracking vehicles in the mid-to far-field of the camera. In this sequence there are some false negative detections dueto the similar appearance of cars and background (in greyscale). A common problem observed is that when multiple objects enter the scene within close proximity they can remain merged; to resolve this relative feature velocities can be analysed to separate the objects. To separate merged observations into individual objects a priori models can be applied (e.g., active shape models [11] for people or textured 3D models for vehicles), although the computational burden may outweigh any benefits of such a step.

3.3. Object recognition results

The evaluation of the object categorisation module was divided into two subtasks to reflect the hierarchical method in which classification is performed: the per-frame bottom-up coarse-level classification for the main types of objects (people, vehicles, aircraft, equipment) and the detailed top-down vehicle recognition performed by 3D model fitting in a background process. These are

(i) coarse categorisation: this task decides whether the object was correctly classified in its main category or not;
(ii) recognition of the object in the category: when the object was correctly classified in its category, the object recognition task evaluates whether the category type (vehicle subtype) of the object was correctly assigned or not.

Table 5 describes the possible categories of the objects in the evaluated datasets and for each category, the related subcategories are enumerated. The subcategories are necessary in order to differentiate objects with similar size or appearance (e.g., vehicles).

For this evaluation, four sequences were considered. The first sequence is S21-Camera 7 (2012 frames), this was used in the motion detection evaluation and contains individuals walking on the apron and vehicles in movement such as GPU, a tanker, a transporter with dollies, and service vehicles. The remaining evaluation sequences are as follows.

S10-Camera 8

(180 frames) the dataset shows a tanker vehicle and a person who walks along the apron.

S22-Camera 5

(1305 frames) the scene shows a loader, a transporter, and a conveyor.

Figure 15: Results of the tracking system running on various datasets. (a) PETS 2001 Dataset 1 Camera 1 frames 875, 988, and 2470; (b) PETS 2001 Dataset 1 Camera 2 and same frames as the previous row; (c) ADVISOR sequence 38 Camera 1 frames 3, 39, and 55 (see [11]); (d) SAFEE sequence EADS Camera 3 frames 260, 330, and 630; (e) traffic sequence frames 607, 1046, and 1090.

Table 5: Category of the objects and correspondent subcategories.

Category	Subcategories
Aircraft	Aircraft
Vehicle	GPU tanker, transporter and dollies, car, loader
Person	One person, group of people
Equipment	Container
Other	Other

Table 6: Classification rates for object categorisation and object subcategorisation.

	Categorisation		Subcategorisation	
Dataset	TP	FP	TP	FP
S10-Cam8	73.77	26.23	68.89	31.11
S21-Cam7	97.86	2.14	77.38	22.62
S22-Cam5	91.03	8.97	61.31	38.69
S44-Cam4	60.13	39.87	88.93	11.07

S44-Camera 4

(1578 frames) three people walk in the apron and a GPU vehicle runs in the scene.

The evaluation procedure was performed as follows: for each sequence, the evaluation was done frame by frame, checking whether objects present in the scene were properly classified into the appropriate category or not. At the same time, the recognition of the object by its subcategory was checked. When the classification of the object corresponds with the real type of the object, a true positive is counted. When the application assign an incorrect class to an object, a false positive is counted.

Table 6 summarises the categorisation results for each evaluated sequence, in terms of coarse-level and detailed-level classification. Table 6 shows that some classification errors occur during the coarse-level classification. These errors appear especially in sequence 44 and sequence 10. The reason for this is that the bottom-up features used during the categorisation process are not properly detected, and therefore the categorisation process fails. But note the high accuracy obtained on the other two evaluated sequences. For the subtype classification, more errors occur because of the similarity of several vehicles and also caused by incorrect model fitting by the SIMPLEX search algorithm (local minimum found instead of the global one).

3.4. Object localisation results

For the evaluation of the 3D localisation module an individual person and vehicle have been considered as follows.

S27-All cameras

The dataset shows individuals walking on well-known trajectories along the grid of the apron.

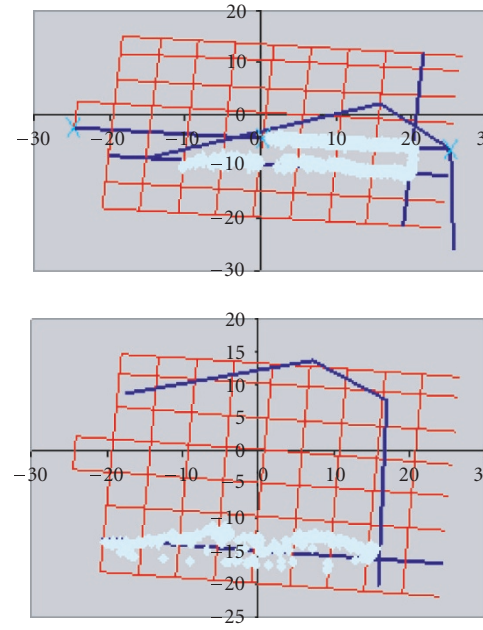


Figure 16: 2D trajectory graph for the person object (S27-Camera 2, person 8 (left) and Camera 4, person 13 (right)). The light (red) lines represent the patching lines and the light (blue) lines represent the camera field of view.

S27-Cameras 3, 4, 5, 6

The dataset contains a single service vehicle driving on the apron for which EGNOS positional measurements were recorded.[1]

To allow the comparison between the apron grid lines and the trajectories, we consider the trajectories defined by the object as paths along the apron. 3D localisation output data (e.g., Info 3D $(X, Y, Z = 0)$) has been generated for each of the test cameras installed at the airport's apron. The coordinate Z is equal to 0 because the objects are constrained to lie on the known ground plane. For each location along the individual path the shortest Euclidean distance (in metres) is computed between the point and the associated grid line. The following performance statistics metrics are applied to the results [24]: mean, standard deviation, minimum, and maximum.

For the person class, it can be seen that person (left) trajectory (see Figure 16) is broken due to occlusions. Occlusions lead to loss of 3D data information causing errors on 3D trajectory reconstruction. In Figure 16 the second person (right) walks along the $y = -15$ grid line. The accuracy of the localisation module depends on the distance between the camera and the object due to the perspective effect and the uniform quantisation of sensor pixels. Reflections of objects

[1] The EGNOS measurements were kindly provided by the ESA project GAMMA (http://www.m3systems.net/project/gamma/); the EGNOS system gives an estimated accuracy of 2-3 m for 95% of measurements.

Table 7: 3D localisation statistical results.

Metric	C1-P27	C2-P8	C2-P12	C3-P10	C4-P10	C4-P13	C5-P8	C5-P13	C6-P27	C7-P8	C7-P25	C8-P5
Frames	148	842	501	361	432	416	419	336	431	265	164	87
Mean	0.83	0.31	0.96	0.73	0.48	1.42	0.93	0.18	2.3	0.34	0.23	0.68
STD	0.48	0.2	0.66	0.52	0.55	0.8	0.74	0.13	2.85	0.59	0.36	0.7
Min	0.14	0.02	0	0	0.01	0	0.003	0	0.01	0.001	0	0.001
Max	1.8	4.4	2.25	2.29	2.13	3.3	3.6	0.62	12.6	2.92	1.93	2.37

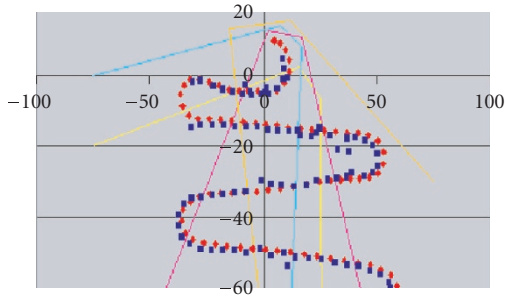


Figure 17: Vehicle 2D trajectory graph showing (red) the EGNOS trajectory and (blue) the estimated location on the apron. The scale is measured in metres and the camera fields of view are shown.

in the ground plane provoke errors on the reconstruction of 3D trajectories.

Table 7 shows the statistical results for the eight cameras; these results demonstrate that the accuracy of the person localisation is approximately 1 metre average over all cameras, this is to be expected due to detection or calibration error. Due to the general innaccuracy in the far-field of all cameras these results show that the use of multiple overlapping cameras is justified for this surveillance system to ensure that the objects are accurately located on the airport apron.

For the evaluation of the vehicle trajectory we only consider a single trajectory estimate made by the "best" camera. The reasoning for this is that the EGNOS data was captured over a large area, and several cameras can view this trajectory. Therefore, at each time step, the size of the tracked object is measured in the four cameras and the one with the largest viewable object is chosen to make the trajectory estimate. In this way we are able to compare the estimated for the entire EGNOS measurement sequence.

The results, shown in Figure 17, demonstrate that the estimated vehicle location is reasonably accurate close to the camera sensors (at the top of the figure). In the far-field the estimate diverges from the measured EGNOS signal due to the perspective effect and the uniform quantisation of the sensor pixels. The mean distance between the EGNOS signal and the estimated location was found to be 2.65 metres +/− 0.34. The minimum deviation was found to be 0.58 metres and the maximum was found to be 4.64 metres.

3.5. Data fusion results

The data fusion module is qualitatively evaluated for two representative test sequences.

S21-All cameras

(9100 frames) the sequence contains individuals walking on the apron. Vehicles in movement such as a GPU, a tanker, a catering vehicle, and service vehicles are also present.

S28-All cameras

(1200 frames) a crowded scene containing many objects interacting within close proximity near the aircraft, this sequence was acquired on a sunny day.

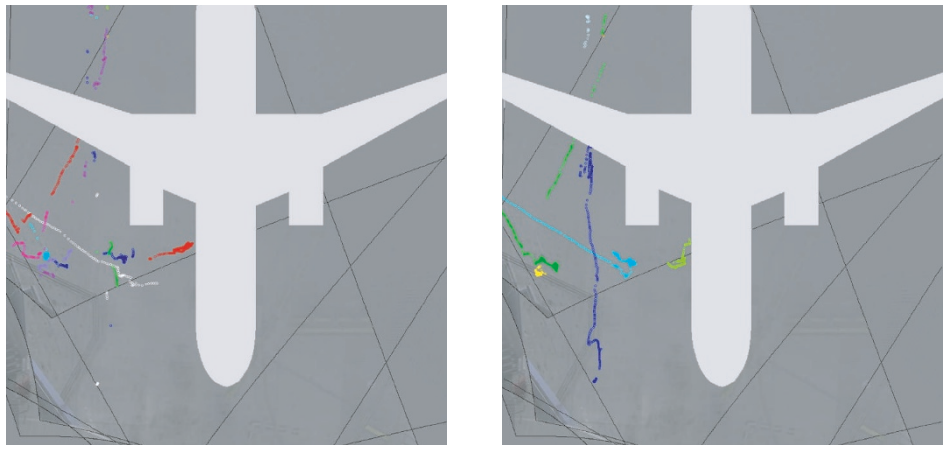
The data fusion performance is shown in Figure 18 where estimated objects on the ground plane are shown for the two test sequences. It is clear to see that by extending the validation gate to include velocity and category, as well as the use of measurement confidence in the fusion process, the extended data fusion module outperforms the standard (i.e., spatial validation and fusion) data fusion process. Many more objects estimated by the extended data fusion are contiguous, with less fragmentation and more robust matching between measurements and existing tracks. It can be seen that the data fusion process is robust against objects that are not on the ground plane (e.g., the containers on the loader in S28). This is achieved by using camera line-of-sight to determine that the container observations do not agree between the cameras and hence the estimated object is given a lower confidence.

The results are encouraging, for many scenarios the extension of the validation gate provides much greater stability, especially when objects are interacting in close proximity. It is noted that the track identity can be lost when the object motion is not well modelled by the Kalman filter or when tracks are associated with spurious measurements. The data fusion module currently has no contextual information about the 3D geometry of the scene; therefore, the camera line-of-sight cannot be accurately determined. Due to this factor, objects can have lower than expected confidence in the data fusion process since some camera measurements cannot be made due to occlusions. The addition of contextual information would also allow the tracking of large objects when they are off the ground plane (e.g., the containers in S28). For larger objects epipolar analysis is not practical; therefore, contextual information about the loader vehicle would be required to position the container objects correctly.

4. DISCUSSION AND FUTURE WORK

The results are encouraging for the components of the scene tracking module. The motion detection module (specifically,

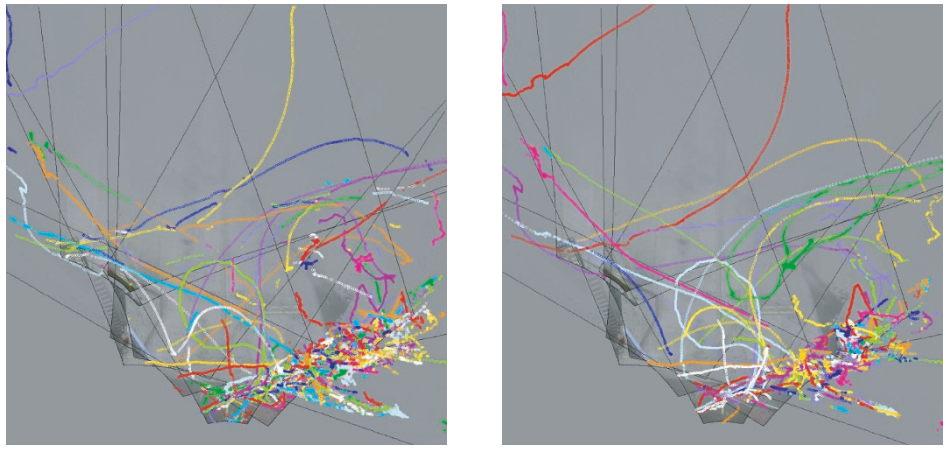
(a)



(b1) Spatial fusion (b2) Extended fusion

(b)

(c)



(d1) Spatial fusion (d2) Extended fusion

(d)

FIGURE 18: Results of the data fusion module showing tracked object locations on the ground plane for two representative datasets. The track colour is derived from the object ID, limited to eight colours for visualisation. (a) S28-all cameras frames: 0, 500, 750, 1000. (b) Objects tracked by the data fusion module with (extended fusion) and without (spatial fusion) the extended validation gate and confidence-based fusion. The aircraft is added for illustrative purposes. (c) S21-All cameras frames: 0, 6000, 7000, 9000. (d) Objects tracked by the data fusion module with (extended fusion) and without (spatial fusion) the extended validation gate and confidence-based fusion.

the colour mean and variance algorithm) showed good performance over a range of test sequences. The object tracking module was found to detect a high proportion of the objects in the scene and these objects are tracked over extended time periods. The object tracking module extended the KLT tracking algorithm to overcome some of the challenges associated with crowded scenes analysis. However, under severe partial occlusions we have found that the tracks become fragmented and lose the track ID. This observation motivates the need for the later scene coherency maintenance module (see Figure 2) that analyses and repairs spatiotemporal discontinuity or fragmentation of the tracked objects. The track localisation methodology, although simple in concept, was shown to be accurate for vehicles and people, although naturally the accuracy reduces further from camera sensor. The data fusion result is promising and improves the tracking result in the crowded scene, although further analysis is required to quantify the accuracy of this module.

Future work will look into using perspective projection motion segmentation in the per camera object tracking module. In the recognition module we will investigate constraints to improve the efficiency and also apply robust region-based descriptors for the bottom-up method to allow categorisation under partial occlusion. In addition, future work will address the classification of articulated vehicles. In the data-fusion module a particle filter based approach will be evaluated to improve performance in the presence of noise.

ACKNOWLEDGMENT

This work was supported by the European Union, Grant AVITRACK (AST3-CT-3002-502818). However, this paper does not necessarily represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

REFERENCES

[1] Y. Bar-Shalom and X. R. Li, *Multitarget Multisensor Tracking: Principles and Techniques*, YBS Publishing, Storrs, Conn, USA, 1995.

[2] D. Thirde, M. Borg, J. Ferryman, et al., "Visual surveillance for aircraft activity monitoring," in *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS '05)*, pp. 255–262, Beijing, China, October 2005.

[3] G. D. Sullivan, "Visual interpretation of known objects in constrained scenes," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 337, no. 1281, pp. 361–370, 1992.

[4] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Proceedings of the 15th International Conference on Pattern Recognition (ICPR '00)*, pp. 167–170, Barcelona, Spain, September 2000.

[5] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, "Detection and location of people in video images using adaptive fusion of color and edge information," in *Proceedings of the IEEE/IAPR 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 4, pp. 4627–4631, Barcelona, Spain, September 2000.

[6] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[7] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94)*, pp. 593–600, Seattle, Wash, USA, June 1994.

[8] R. Collins, A. Lipton, T. Kanade, et al., "A system for videosurveillance and monitoring: VSAM final report," Tech. Rep. CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa, USA, May 2000.

[9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, vol. 2, pp. 246–252, Fort Collins, Colo, USA, June 1999.

[10] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*, Kluwer Academic, Dordrecht, The Netherlands, 1996.

[11] N. T. Siebel and S. J. Maybank, "Fusion of multiple tracking algorithms for robust people tracking," in *Proceedings of the 7th European Conference on Computer Vision (ECCV '02)*, pp. 373–387, Copenhagen, Denmark, May 2002.

[12] J. Ferryman, A. D. Worrall, and S. J. Maybank, "Learning enhanced 3D models for vehicle tracking," in *Proceedings of the British Machine Vision Conference*, pp. 873–882, Southampton, UK, September 1998.

[13] J. Black and T. Ellis, "Multi camera image measurement and correspondence," *Measurement - Journal of the International Measurement Confederation*, vol. 35, no. 1, pp. 61–71, 2002.

[14] M. Xu, J. Orwell, and G. Jones, "Tracking football players with multiple cameras," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '04)*, vol. 2, pp. 2909–2912, Suntec City, Singapore, October 2004.

[15] J. Aguilera, H. Wildenauer, M. Kampel, M. Borg, D. Thirde, and J Ferryman, "Evaluation of motion segmentation quality for aircraft activity surveillance," in *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS '05)*, pp. 293–300, Beijing, China, October 2005.

[16] P. Correia and F. Pereira, "Objective evaluation of relative segmentation quality," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '00)*, vol. 1, pp. 308–311, Vancouver, British Columbia, Canada, September 2000.

[17] T. Ellis, "Performance metrics and methods for tracking in surveillance," in *Proceedings of the 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS '02)*, pp. 26–31, Copenhagen, Denmark, June 2002.

[18] C. E. Erdem and B. Sankur, "Performance evaluation metrics for object-based video segmentation," in *Proceedings of the 10th European Signal Processing Conference (EUSIPCO '00)*, pp. 917–920, Tampere, Finland, September 2000.

[19] T. Schlögl, C. Beleznai, M. Winter, and H. Bischof, "Performance evaluation metrics for motion detection and tracking," in *Proceedings of the International Conference on Pattern Recognition (ICPR '04)*, vol. 4, pp. 519–522, Cambridge, UK, August 2004.

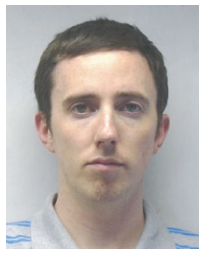
[20] P. Villegas and X. Marichal, "Perceptually-weighted evaluation criteria for segmentation masks in video sequences," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1092–1103, 2004.

[21] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Still image objective segmentation evaluation using ground truth," in *Proceedings of the 5th COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication*, pp. 9–14, Prague, Czech Republic, October 2003.

[22] A. Cavallaro, E. D. Gelasca, and T. Ebrahimi, "Objective evaluation of segmentation quality using spatio-temporal context," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 301–304, Rochester, NY, USA, September 2002.

[23] J. Black, T. Ellis, and P. Rosin, "A Novel method for video tracking performance evaluation," in *Proceedings of Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS '03)*, pp. 125–132, Nice, France, October 2003.

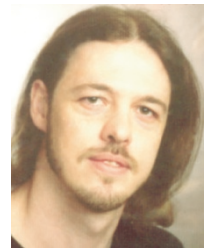
[24] C. J. Needham and R. D. Boyle, "Performance evaluation metrics and statistics for positional tracker evaluation," in *Proceedings of the 3rd International Conference on Computer Vision Systems (ICVS '03)*, pp. 278–289, Graz, Austria, April 2003.



David Thirde received his B.Eng. in electrical and electronic engineering from Loughborough University in 2000. He is undertaking a Ph.D. degree with the Digital Imaging Research Centre at Kingston University, where he was funded until 2003. From 2003 to 2004 he worked as a researcher on the EU project INMOVE. Upon completion of this project he moved to the Computational Vision Group at the University of Reading, where he has worked on the EU projects AVITRACK and ISCAPS. His research interests include high-level interpretation of video and autonomous surveillance systems.

Mark Borg received a B.S. degree in mathematics and computer science from the University of Malta in October 1995, and an M.S. degree in engineering and information sciences from the University of Reading in 2003. In 2004, Mark joined the Computational Vision Group of the University of Reading, where he worked as a research assistant in the area of automated visual surveillance and tracking, in particular participating in the EU AVITRACK project. Starting in 2006, Mark returned back to industry and is currently working in the R&D group of Crimsonwing developing new e-commerce solutions, as well as providing freelance consultancy services.

Josep Aguilera received his M.S. degree in computer science at the University Autonoma of Barcelona in 2000. He joined the Pattern Recognition and Image Processing Group at the Vienna University of Technology in 2004. He has worked on the EU project AVITRACK, which addresses the specific case of visual surveillance and monitoring of an airport's apron. His research is focused on multi-camera visual surveillance systems and performance evaluation.



Horst Wildenauer received his B.S. and M.S. (honours) degrees in computer science from the Vienna University of Technology, Austria, in 1996 and 1998, respectively. Currently he is pursuing his Ph.D. degree at the Institute of Computer Aided Automation, Vienna University of Technology. Since 1999 he is with the Pattern Recognition and Image Processing Group, working as a Research Assistant. His research interests include linear subspace-based appearance modelling for object recognition, kernel methods for pattern analysis, colour image processing, and industrial applications.

James Ferryman research interests include model-based methods for people and traffic surveillance, human-computer interaction, robotics and autonomous systems, and "smart" cameras. He was investigator on two EC Framework V proposals: ADVISOR (IST-1999-11287) on people tracking in metro stations, and ANFAS on modelling flood risks (IST-1999-11676), and the EU Framework VI Aero project AVITRACK which focused on the automated visual surveillance of airport aprons. Dr. Ferryman is cochair of the IEEE International Workshops on Performance Evaluation of Tracking and Surveillance in 2000–2004, and is a reviewer for the EU Sixth Framework IST Programme. He is currently a coinvestgator of the UK EPSRC project REASON on the robust monitoring of people in public spaces, the UK EPSRC network ViTAB (Video-based Threat Assessment and Biometrics) and is a principal investigator for the EU FP6 Aero project: SAFEE which addresses on-board security.

Martin Kampel received the B.S. degree in data technologies and computer science, the M.S. degree (Diplom Ingenieur) in computer science (computer graphics, pattern recognition, and image processing) in 1999 and the Ph.D. degree in computer science in 2003 from the Vienna University of Technology. He is an Assistant Professor (Univ. Ass.) of computer vision at the Pattern Recognition and Image Processing Group, Vienna University of Technology, engaged in research, project leading, industry consulting, and teaching. His research interests are 3D vision and cultural heritage applications, visual surveillance and image sequence analysis. He is author or co-author of more than 60 scientific publications presented at several international conferences and workshops and is a Member of the IAPR and the IEEE.