Visual Surveillance for Aircraft Activity Monitoring

David Thirde[1], Mark Borg[1], Valéry Valentin[2], Florent Fusier[2], Josep Aguilera[3]
James Ferryman[1], François Brémond[2], Monique Thonnat[2], Martin Kampel[3]

[1] Computational Vision Group, The University of Reading, UK
{D.J.Thirde, M.Borg, J.Ferryman}@rdg.ac.uk

[2] ORION Team, INRIA Sophia-Antipolis, France
{Valery.Valentin,Florent.Fusier,Francois.Bremond,Monique.Thonnat}@sophia.inria.fr

[3] Pattern Recognition and Image Processing Group, Vienna University of Technology, Austria
{agu,kampel}@prip.tuwien.ac.at

Abstract

This paper presents a visual surveillance system for the automatic scene interpretation of airport aprons. The system comprises two modules — Scene Tracking and Scene Understanding. The Scene Tracking module, comprising a bottom-up methodology, and the Scene Understanding module, comprising a video event representation and recognition scheme, have been demonstated to be a valid approach for apron monitoring.

1. Introduction

This paper describes work undertaken on the EU project AVITRACK. The 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*). A combination of visual surveillance algorithms are applied in a decentralised multi-camera environment with overlapping fields of view [3] to track objects and recognise activities predefined by a set of servicing operations. Each camera agent performs per frame detection and tracking of scene objects, and the output is transmitted to a central server where data association and fused object tracking is performed. This result is subsequently fed to a video event recognition module where spatial and temporal events relating to the servicing of the aircraft are detected and analysed. The system must be capable of monitoring and recognising the activities and interaction of numerous vehicles and personnel in 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.

The tracking of moving objects on the apron has previously been performed using a top-down model based approach [14] although such methods are generally computationally expensive when applied to real time tracking. An alternative approach, bottom-up scene tracking, refers to a process that comprises the two sub-processes *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 [13, 10] 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. The airport apron, being an outdoor environment, provides several challenges to motion detection. It must handle a wide range of environmental conditions, weather, and illumination changes, which can be long-term changes (diurnal cycle) or short-term (cloud movements, reflections, etc). The AVITRACK test sequences, like many CCTV applications, also suffer from chrominance and luminance sensitivity and have significant JPEG artifacts; the moving objects and apron are also of an achromatic nature with low contrast between the observed foreground and background.

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 and update previously observed objects in the current time step. The Kanade-Lucas-Tomasi (KLT) feature tracker [12] combines a local feature selection criterion with feature-based matching in adjacent frames. The CamShift algorithm [6] uses appearance based (colour histogram) representation of objects to perform tracking using the mean-shift algorithm. The work of Pece [11] filters the spatial location and extent of objects using a recursive probabilistic generative model. Tracking algorithms have to deal with motion detection errors and complex object interactions; e.g. objects appear to merge together, occlude each other, fragment, undergo non-rigid motion, etc. Apron analysis presents further challenges due to the size of the vehicles tracked (e.g. the aircraft size is $34 \times 38 \times 12$ metres), therefore prolonged occlusions occur throughout apron operations. The apron can also be congested with objects; this enhances the difficulty of associating objects with regions.

Video event recognition algorithms analyse tracking results spatially and temporally to automatically recognise the high-level activities occurring in the scene; for aircraft servicing analysis such activities occur simultaneously over extended time periods in apron areas. Recent work by Xiang *et al* [16] applied a hierarchical dynamic Bayesian network to recognise scene events; however, such models are incapable of recognising simultaneous complex scene activities in real-time over extended time periods.

Section 2 details the scene tracking module comprising per-camera motion detection, bottom-up feature-based per-camera object tracking and finally fused object tracking using the combined object tracking results from the camera agents. Section 3 describes the video event recognition module including both the representation of video events and the video event recognition algorithm itself applied to apron monitoring. Section 4 presents the results, while Section 5 contains the discussion and lists future work.

2. Scene Tracking

The scene tracking module is responsible for the detection and tracking of moving objects from individual cameras; the tracked object locations are subsequently transformed into 3D world co-ordinates. The data fusion algorithm determines single world measurements for each object from the multiple camera observations.

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. 16 motion detection algorithms were implemented and evaluated. Of these, the following three algorithms (all based on the aforementioned background subtraction method) have acceptable susceptibility to noise and good detection sensitivity: mixture of Gaussians [13], colour and edge fusion [10] and colour mean and variance. By taking into account processing efficiency as well as sensitivity, the colour mean and variance method was selected. The colour mean and variance algorithm has a background model represented by a pixel-wise Gaussian distribution $N(\mu, \sigma^2)$ over the normalised RGB colour space. In addition, a shadow/highlight detection component based on the work of Horprasert *et al* [9], is used to handle illumination variability. The algorithm also employs a multiple background layer technique to allow the temporary inclusion into the background model of objects that become stationary for a short period of time.

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 inter-frame changes are limited, thus allowing the use of temporal constraints and/or object features to simplify the correspondence problem. Three approaches to object tracking were applied: based on the tracking of local features [12], colour information [6] and difference image clusters [11]. The local feature tracking method (KLT) was found to give the most reliable tracking result in evaluation and was chosen for use in this project. The KLT 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, maintain 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, with the number of features determined dynamically from the object size and a configurable feature density parameter $\boldsymbol{\rho}$.

Given a set of tracked objects $\left\{O_i^{t-1}\right\}$ at time $t-1$, and a set of observations $\left\{M_j^t\right\}$ at time t obtained from the motion detector, i.e. connected components of foreground pixels, the tracking process is summarised as:

1. Generate object predictions $\{P_i^t\}$ for time t from the set of known objects $\left\{O_i^{t-1}\right\}$ at $t-1$.
2. Run the KLT algorithm to track each local feature from the set of features $S_{P_i^t}$ of prediction $\{P_i^t\}$.
3. For an observation set $\left\{M_j^t\right\}$ detected by the motion detector, match predictions $\{P_i^t\}$ by determining to which M_j^t the tracked features of P_i^t belong to.
4. Unmatched predictions in $\{P_i^t\}$ are marked missing observations. Unmatched observations in $\left\{M_j^t\right\}$ are considered potential new objects.
5. Update the state of those predictions in $\{P_i^t\}$ that were matched to observations and replace 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.

For step 3 above, 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:

$$f\left(P_i^t, M_j^t\right) = \left|\left\{W : W \in S_{P_i^t}, W \in M_j^t\right\}\right| \qquad (1)$$

For a non-interacting object, (1) returns a non-zero value for only one prediction-observation pair. But to handle complex object interactactions, a rule-based approach is adopted. The rule for object splitting takes care of assigning the local features to the corresponding new object, so that features are maintained through the splitting event. In the case of merging objects, the known local states of the tracked features are used to update the 'unknown' global states of the predictions. Other rules are used to handle interactions of moving objects with objects that have become stationary (integrated into the background model), and stationary object re-activation once they start moving again.

2.3. Data Fusion

The data fusion module combines the tracking data seen by each of the individual cameras to maximise the useful information content of the scene being observed and hence achieves enhanced occlusion reasoning, a larger visible area and improved 3D localisation. Data fusion also minimises the volume of data generated by the many cameras and reduces the bandwidth needed to send information to later modules. Spatial registration of the cameras is performed using per camera coplanar calibration (with radial lens distortion coefficient) and the camera streams are synchronised temporally across the network by the central server.

The method for Data Fusion is based on a nearest neighbour Kalman filter approach [3] with a constant velocity model. The measurement noise covariance $\mathbf{R}$ is estimated by propagating a nominal image plane uncertainty $\mathbf{\Lambda}$ such that the measurement uncertainty in the world co-ordinate system is given by [4] i.e. $\mathbf{R}\left(x_w, y_w, z_w\right) = \mathbf{J}\left(x_c, y_c\right)\mathbf{\Lambda}\mathbf{J}\left(x_c, y_c\right)^T$ where $\mathbf{J}$ is the Jacobian matrix found by taking the derivatives of the two mapping functions between the image and world co-ordinate systems. The measurement uncertainty field is shown in Figure 1 for camera 6; this estimate of uncertainty allows formal methods to be used to associate observations originating from the same measurement, as well as providing mechanisms for fusing observations into a single estimated measurement. For each object the measurement location and associated uncertainty is also dependent on the object dimensions; a bias is incorporated in the estimate using a heuristic method that includes the camera angle to the ground plane, object category and the measured object size.

In the association step a validation gate [3] is applied to limit the potential matches between existing tracks and observations. Matched observations are combined to find the fused estimate of the location and uncertainty of the object, this is achieved using *covariance accumulation* and *covariance intersection*. Covariance accumulation estimates the fused uncertainty $\mathbf{R}_{fused}$ for N matched observations as $\mathbf{R}_{fused} = \left(\mathbf{R}_1^{-1} + \ldots + \mathbf{R}_N^{-1}\right)^{-1}$. The covariance intersection method is conceptually similar to the accumulation except that the observation uncertainty covariances are weighted in the summation:

$$\mathbf{R}_{fused} = \left(w_1\mathbf{R}_1^{-1} + \ldots + w_N\mathbf{R}_N^{-1}\right)^{-1} \qquad (2)$$

where $w_i = w_i' / \sum_{j=1}^{N} w_j'$ and $w_i' = 1/\mathrm{Tr}\left(\mathbf{R}_i^c\right)$. $\mathbf{R}_i^c$ is the measurement uncertainty of the i'th associated observation (made by camera c); Covariance intersection therefore weights in favour of the sensors that have more certain measurements. The resulting fused observations are shown in Figure 1; the covariance accumulation method results in a more localised estimate of the fused measurement than the covariance intersection approach. Unassociated measurements are fused into new tracks, using a validation gate between observations to constrain the association and fusion steps. The track category is estimated as a weighted average over the fused observations; with each class probability modelled using a supervised 2-D Gaussian Mixture Model, representing object width and height in world co-ordinates.

3. Scene Understanding

The scene understanding module is responsible for the recognition of video events in the scene observed through video sequences. This module performs a high-level interpretation of the scene by detecting video events occurring in it. The method to detect video events uses cognitive vision techniques based on spatio-temporal reasoning, *a priori* knowledge of the observed environment and a set of predefined event models. A Video Event Recognition module takes the tracked mobile objects from the previously described modules as input, and outputs the recognised events.

The *a priori* knowledge is the knowledge about the observed empty scene. This includes the camera information, the vehicle models, the expected moving objects and the empty scene model (also called the static environment observed by the cameras) containing the contextual objects (e.g. equipment, zones of interest, walls, doors). Contextual objects are characterised by their 3D geometry (to provide an approximative shape) and by their semantics (to describe how they interact with mobile objects like persons or vehicles). The *a priori* knowledge also includes the set of event models defined by the domain experts using a video event description language described in [7].

3.1. Video Event Representation

The video event representation corresponds to the specification of all the knowledge used by the system to detect video events occurring in the scene. To allow experts in the aircraft activity monitoring to easily define and modify the video event models, the description of the knowledge is declarative and intuitive (in natural terms). Thus, the

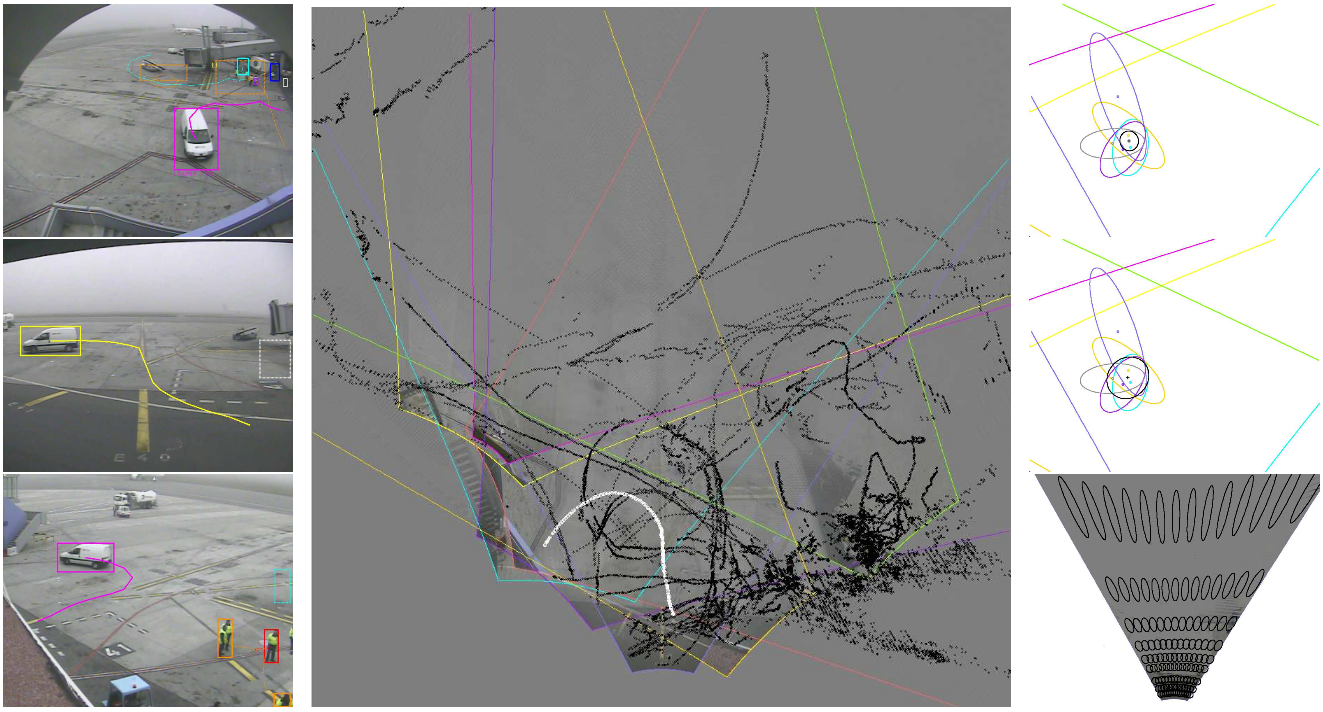
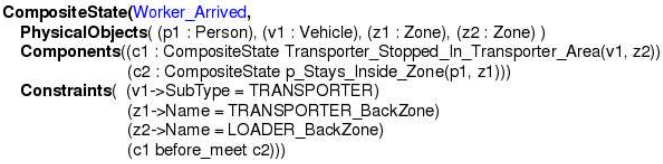


Figure 1: (Left) Tracking results for 3 cameras for frame 9126 of sequence 21. (Middle) shows data fusion results on the ground-plane for the sequence (9600 frames) with the vehicle track shown in white. (Top-right) the fused observation (in black) for the vehicle (frame 9126) using the covariance accumulation method, (Middle-right) shows the result for covariance intersection. (Bottom-right) shows the sensory uncertainty field measured for camera 6.

video event recognition uses the knowledge represented by experts through event models. The proposed model of a video event E is composed of five parts:

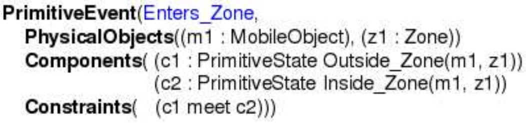
- a set of Physical Object variables corresponding to the physical objects involved in E : any contextual object including static object (equipment, zone of interest) and mobile object (person, vehicle, aircraft...). The vehicle type can be of different subtypes to represent different vehicles types (GPU, Loader, Tanker etc.)
- a set of temporal variables corresponding to the components (sub-events) of E
- a set of forbidden variables corresponding to the components that are not allowed to occur during E
- a set of constraints (symbolic, logical, spatial and temporal constraints including Allen's interval algebra operators [2]) involving these variables
- a set of decisions corresponding to the tasks predefined by experts that need to be executed when E is detected (e.g. activating an alarm or displaying a message)

There are four types of video events: primitive state, composite state, primitive event and composite event. A state describes a situation characterising one or several physical objects defined at time t or a stable situation defined over a time interval. A primitive state (e.g. a person is inside a zone) corresponds to a vision property directly computed by the vision module. A composite state, as shown in Figure 2, corresponds to a combination of primitive states. An event is an activity containing at least a change of state values between two consecutive times (e.g. a vehicle leaves a zone of interest : it is inside the zone and then it is outside). A primitive event, as shown in Figure 3, is a change of primitive state values and a composite event is a combination of states and/or events.



```
CompositeState(Worker_Arrived,
  PhysicalObjects( (p1 : Person), (v1 : Vehicle), (z1 : Zone), (z2 : Zone) )
  Components((c1 : CompositeState Transporter_Stopped_In_Transporter_Area(v1, z2))
             (c2 : CompositeState p_Stays_Inside_Zone(p1, z1)))
  Constraints(  (v1->SubType = TRANSPORTER)
             (z1->Name = TRANSPORTER_BackZone)
             (z2->Name = LOADER_BackZone)
             (c1 before_meet c2)))
```

Figure 2: The model of the composite state "Worker_Arrived" (event 25) : the Transporter vehicle is stopped (c1) in its area (z2) and a worker stays inside (c2) the container zone (z1) waiting for container to be unloaded and to be attached to the Transporter.



```
PrimitiveEvent(Enters_Zone,
   PhysicalObjects((m1 : MobileObject), (z1 : Zone))
   Components( (c1 : PrimitiveState Outside_Zone(m1, z1))
               (c2 : PrimitiveState Inside_Zone(m1, z1))
   Constraints(  (c1 meet c2)))
```

Figure 3: The model of the primitive event "Enters_Zone" : a vehicle enters a zone.

3.2. Video Event Recognition

The video event recognition algorithm recognises which events are occurring in a stream of mobile objects tracked by the vision module. The algorithm to recognise a primitive state consists of two operations in a loop: (1) selection of a set of physical objects; then (2) verification of the corresponding atemporal constraints until all combinations of physical objects have been tested. Once a set of physical objects satisfies all atemporal constraints, the primitive state is said to be recognised. In order to facilitate primitive event recognition, event templates are generated for each primitive event, the last component of which corresponds to this recognised primitive state. The event template contains the list of physical objects involved in the primitive state. These physical objects partially instantiate the event template.

To recognise a primitive event, given the event template partially instantiated, the recognition algorithm selects (if needed) a set of physical objects matching the remaining physical object variables of the event model. It then looks back in the past for any previously recognised primitive state that matches the first component of the event model. If these two recognised components verify the event model constraints, the primitive event is said to be recognised. In order to facilitate composite event recognition, after each primitive event recognition, event templates are generated for all composite events, the last component of which corresponds to this recognised primitive event.

The recognition of composite states and events usually requires a search in a large space composed of all the possible combinations of components and objects. To avoid this combinatorial explosion, all composite states and events are simplified into states and events composed of at most 2 components through a stage of compilation in a preprocessing phase. Then the recognition of composite states and events is performed in a similar way to the recognition of primitive events. The video event recognition algorithm is based on the method of Vu *et al* [15].

3.3. Video Event Recognition for Apron Monitoring

In the Video Event Recognition module, *a priori* knowledge corresponds to apron zones of interest (access zones, stopping zones), aircraft and vehicle (e.g. GPU, Loader, Tanker and Transporter) models. Even if the handling operations on the apron are codified and controlled, some problems may occur while trying to build an accurate context of the scene. For example, access zones to aircraft can be at different positions according to the aircraft type. In some cases, we need to detect a person getting out of a parked vehicle which does not always stop exactly at the same place. To solve these problems, dynamic properties have been added to the *a priori* knowledge, by defining dynamic zones in the local coordinate system of vehicles. In order to effectively use dynamic context, accurate information is needed from the tracking modules for the orientation when a vehicle is parked. A transformation matrix is computed from local to global scene coordinate system and then dynamic zones are added to the context. Figure 4 illustrates the use of dynamic context. This notion of dynamic context allows more complex scenarios to be defined in which mobile objects can directly interact with each other.

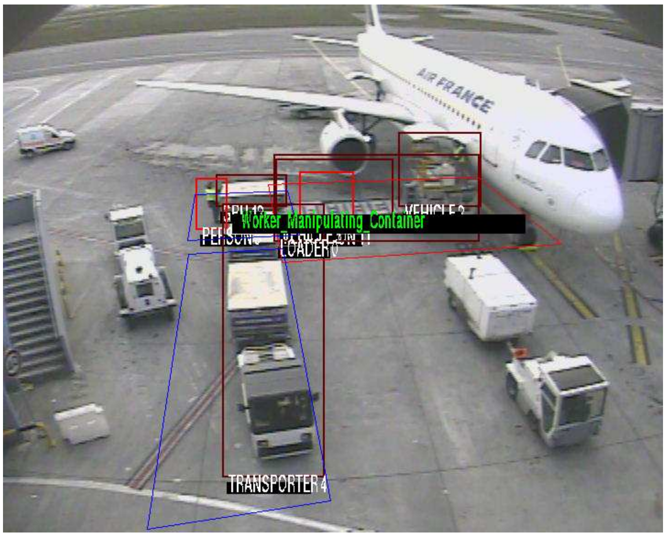
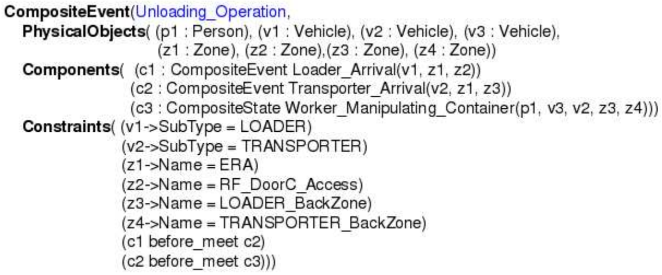


Figure 4: Two dynamic zones (in blue) linked with the Loader and the Transporter vehicles involved in the event "Worker_Manipulating_Container" (event 26) detected.

3.4. Predefined Video Events

At the moment, we have defined a set of 21 basic video events (with one person and one vehicle) including 10 primitive states (e.g. a person is located inside a zone), 5 composite states (e.g. a vehicle is stopped inside a zone) and 6 primitive events (e.g. a vehicle changes zone). These basic video events are used in the definition of video events representing the handling operations. We have worked on video events involving (1) the GPU (Ground Power Unit) vehicle which operates in the aircraft arrival preparation operation, (2) the Tanker vehicle which operates in the refuelling oper-



```
CompositeEvent(Unloading_Operation,
   PhysicalObjects( (p1 : Person), (v1 : Vehicle), (v2 : Vehicle), (v3 : Vehicle),
                    (z1 : Zone), (z2 : Zone),(z3 : Zone), (z4 : Zone))
   Components(  (c1 : CompositeEvent Loader_Arrival(v1, z1, z2))
                (c2 : CompositeEvent Transporter_Arrival(v2, z1, z3))
                (c3 : CompositeState Worker_Manipulating_Container(p1, v3, v2, z3, z4)))
   Constraints( (v1->SubType = LOADER)
                (v2->SubType = TRANSPORTER)
                (z1->Name = ERA)
                (z2->Name = RF_DoorC_Access)
                (z3->Name = LOADER_BackZone)
                (z4->Name = TRANSPORTER_BackZone)
                (c1 before_meet c2)
                (c2 before_meet c3)))
```

Figure 5: The Unloading operation involves 8 physical objects (1 person, 3 vehicles and 4 zones of interest). It is composed of 3 composite components which are recognised when the Loader vehicle arrives (event 20 is recognised when events 17, 18, 19 have been recognised), when the Transporter arrives (event 23 is recognised when events 21, 22 have been recognised), and the worker is manipulating an unloaded container (event 26) to attach it to the Transporter. There are 2 constraints on the vehicle subtypes, 4 constraints on the zones of interest and 2 temporal constraints.

ation and (3) the Loader and Transporter vehicles which are involved in the baggages loading/unloading operations. To recognise these operations we have defined 28 composite video events including 8 video events for the aircraft arrival preparation operation, 8 video events for the refuelling operation, and 12 video events for the unloading operation.

The aircraft arrival preparation operation (event 8) involves the GPU, its driver and 4 zones of interest. The system recognises that the GPU vehicle arrives in the ERA Zone (event 1), respecting the speed limit (event 2) and then it enters (event 3) and stops (event 4) in the "GPU Access Area" and then the driver gets out from the vehicle (event 5) and deposits the chocks and stud at the location where the plane will stop (events 6 and 7).

The refuelling operation involves the Tanker which arrives in ERA zone (event 9), respecting the speed limit (event 10) and then enters (event 11) and stops (event 12) in the "Refuelling Area". This sequence corresponds to the Tanker arrival (event 13). Then the driver gets out from the Tanker and gets inside the Tanker platform zone (event 14), activates the Tanker platform (event 15), branches the fuel pipe (event 16) and refuels the aircraft.

The operation of baggage unloading is more complex. This operation involves both a Loader and a Transporter vehicle, the conductor of the Loader, and a person working in the area. This operation is composed of successives steps. First the Loader vehicle arrives in the ERA zone (event 17) and enters in its restricted area (event 18), then it stops in this zone (event 19) and automatically a dynamic zone is loaded in the back of the Loader stop position ("Loader Arrival", event 20), where the Transporter will enter and stop. When the Transporter enters (event 21) and stops (event 22) in this zone ("Transporter Arrival", event 23), another dynamic zone is automatically added to the context. Then the back of the Loader is elevated (event 24) and receives the baggage containers which are unloaded from the aircraft by the Loader conductor (event 25) one by one. Then the conductor unloads these containers into the dynamic zone of the Transporter where a worker arrives (event 26) and takes the containers (event 27) to attach them to the Transporter. The Unloading operation (as shown in Figure 5) is detected when the events 17, 18, 19, 20, 21, 22, 23, 26 and 27 have been recognised by the system.

4. Results

4.1. Scene Tracking

The Scene Tracking evaluation accesses the performance of the three core components (motion detection, object tracking and data fusion) on representative test data. More detail on the Scene Tracking evaluation work is given in [1] and [5]. To evaluate the performance of the colour mean and variance motion detector three apron datasets were chosen. Dataset 1 (9148 frames) contains the presence of fog whereas Datasets 2 and 3 (6023 frames) are acquired on a sunny day. Fifteen reference frames were chosen from each dataset for which ground truth motion images were manually generated. These segmented objects were compared with the foreground objects detected by the motion detector and false positive, false negative, true positive and true negative object pixels were counted and summed up over the chosen frames. The following metrics defined by Ellis [8] were used to evaluate the performance of the algorithm:

- Detection rate: $TP/(TP+FN)$
- Accuracy: $(TN+TP)/N$
- False negative rate: $FN/(TP+FN)$
- False positive rate: $FP/(FP+TN)$

where N is the total number of pixels, TP number of true positives, TN number of true negatives, FN and FP number of false negatives and positives respectively.

Figure 6 demonstrates the robustness of the motion detector against illumination changes and weather conditions. Objects in the scene such as the aircraft from Dataset 1 are partially detected due to the achromaticity of the scene. Strong shadows are detected as part of the mobile objects, and holes and fragmentation are presented in objects with the same colour as background. The results of the performance evaluation can be seen in Table 1. It is desirable to have a detection rate and accuracy approaching 100% and a false positive/negative rate approaching 0%. For Datasets 1 and 3, the motion detector provides a detection rate of 77% and a false negative rate of 23%. In Dataset 2 the detection

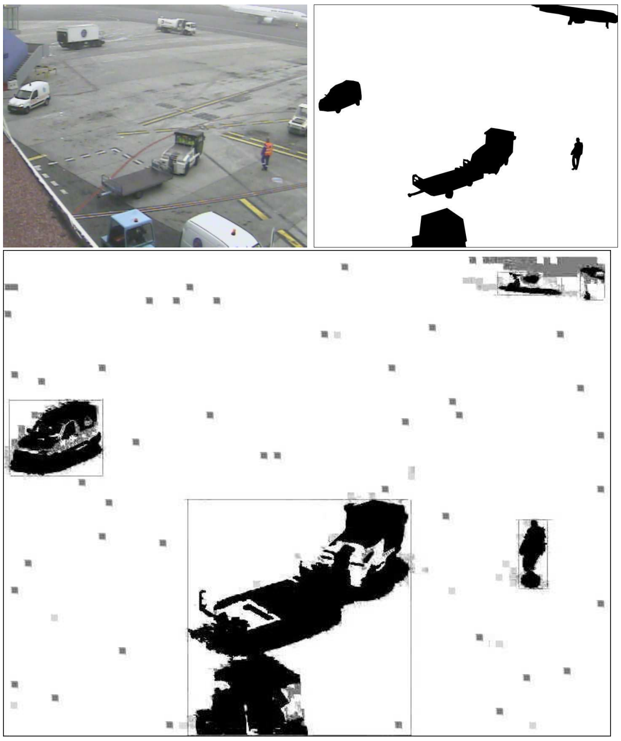


Figure 6: Representative motion detection result from Dataset 1 showing (Top-Left) reference image, (Top-Right) ground truth and (Bottom) detection result.

Dataset	detect. rate	accur.	false pos. rate	false neg. rate
1	0.7688	0.9351	0.0736	0.2310
2	0.6052	0.9737	0.0134	0.3946
3	0.7664	0.9779	0.0181	0.2334

Table 1: Performance results of the colour mean & variance motion detector in the apron datasets.

rate decreases to 60%. The achromatic nature of the scene generates a considerable number of false negatives causing the decrease in detection rate and the increase in false negative rate. The fog in Dataset 1 causes a high number of foreground pixels to be misclassified as highlighted background pixels resulting in a decrease in accuracy (93%).

Representative results of the local feature tracking method are presented in Figure 1. The local feature tracking method gives the best results so far although it suffers from loss of object identity if the local features are lost during merged or occluded states and cannot be replenished. Objects must also be textured in order for good features to be selected. The difference image clusters approach looks promising, but suffers from object identity loss when clusters merge and requires higher-level processing to alleviate this. The colour tracking method performs quite badly due to the achromatic nature of the scene.

The data fusion module performs adequately given isolated targets correctly detected in the frame tracker (a representative result is shown in Figure 1). The data fusion module incorporates uncertainty information in the location estimate of the observation and it is often an inaccurate location estimate that results in the failure of the data association step; a significant proportion of the localisation problems that occur in the data fusion module can be traced back to motion detection errors i.e. shadow, reflections etc.

4.2. Scene Understanding

The Scene Understanding evaluation have been performed on sequences for which the Scene Tracking module gives good results. We have tested the video event recognition on sequences involving the GPU (aircraft arrival preparation), the Tanker (refuelling) and the Loader and the Transporter vehicles (baggage unloading).

The video events 1 to 4 involving a GPU have been tested on a dataset of 4 scenes corresponding to 2*4 video sequences (containing from 1899 to 3774 frames and including one night sequence). These events are detected with a perfect True Positive rate. The video events 4 to 8 involving also a GPU have been tested on 2 scenes corresponding to 2 video sequences because only one camera is available to observe these events. The video events involving the Tanker have been tested on one scene (more than 15000 frames corresponding to about 30 minutes) showing the "Tanker Arrival" (event 13) and the driver of the Tanker branching the refuelling pipe to the aircraft (events 14, 15, 16).
The "Unloading Baggage operation" involving the Loader (events 17 to 20, event 24 and event 25) and the Transporter (events 21 to 23) have been tested on one scene where the point of view allows to fully observe the vehicle movements and interactions between vehicles and people. Eight cameras observe the same scene with different fields of view. The video event recognition module has been tested on the two best points of view from where the GPU can be ob-

Vehicle type	Sequence	TP	FP	FN
GPU				
Events 1 to 4	4 scenes * 2 cam.	32	0	0
Events 4 to 8	2 scenes * 1 cam.	8	0	0
Tanker				
Events 9 to 13	2 scenes * 1 cam.	10	0	0
Events 14 to 16	1 scene * 1 cam.	3	0	0
Loader-Transporter				
Events 17 to 28	1 scenes * 1 cam.	12	0	0

Table 2: Performance results of the Scene Understanding module for apron monitoring. TP = "Real world event is recognised", FN = "Real world event is not recognised", FP = "Spurious event is recognised".

served and on the only point of view from where the Tanker can be observed, and from the best point of view from where the Loader and the Transporter can be observed.

Currently, this evaluation is mainly qualitative and performed manually, the results of the evaluation are shown in Table 2. The goal is to give an idea of the performance of the Scene Understanding and to anticipate potential problems in event detection for apron monitoring. All video events are recognised correctly (49 TPs) without false alarms (0 FPs) and misdetection (0 FNs). These results are very encouraging but one has to keep in mind that situations where the vision module misdetects or overdetects mobile objects were not addressed.

5. Discussion and Future Work

The results are encouraging for both the Scene Tracking and Understanding modules. The performance of multi-view object tracking provides adequate results; however, tracking is sensitive to significant dynamic and static object occlusion within the scene. Future work will address shadow supression, and explicit occlusion analysis.

The Scene Understanding results show that the proposed approach is adapted to apron monitoring and can be applied to complex activity recognition. The main difficulty for apron monitoring is to model operations using *a priori* expert knowledge (49 video events already defined) and to recognise them all in parallel. The recognition of complex operations (e.g. "baggage unloading") involving people and vehicles gives good results and encourages us to continue with more complex operations with more interactions between people and vehicles. Another issue is to incorporate uncertainty to enable recognition of events even when the Scene Tracking module gives unreliable output.

Acknowledgements

This work is supported by the EU, grant AVITRACK (AST3-CT-3002-502818).[1]

References

[1] J.Aguilera, H. Wildernauer, M.Kampel M. Borg, D. Thirde and J. Ferryman. "Evaluation of Motion Segmentation Quality for Aircraft Activity Surveillances." *In Proc. Joint IEEE Int. Workshop on VS-PETS*,Beijing, Oct 2005.

[2] J. F. Allen. "Maintaining Knowledge about Temporal Intervals." *In Communications of the ACM*, 26(11) pp 823–843, 1983.

[3] Y. Bar-Shalom and X.R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS, 1995.

[4] J. Black and T.J. Ellis, "Multi Camera Image Measurement and Correspondence." *Measurement - Journal of the International Measurement Confederation*, 35(1), pp 61–71, 2002.

[5] D. Thirde, M. Borg, J.Aguilera, J. Ferryman, K.Baker and M.Kampel. "Evaluation of Object Tracking for Aircraft Activity Surveillance." *In Proc. Joint IEEE Int. Workshop on VS-PETS*, Beijing, Oct 2005.

[6] G. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface." *Intel Technology Journal*, Q2, 1998.

[7] F. Bremond, N. Maillot, M. Thonnat and V. Vu. "Ontologies for Video Events." *Technical Report 51895*, INRIA Sophia-Antipolis, Nov 2003.

[8] T. Ellis, "Performance Metrics and Methods for Tracking in Surveillance" *In IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, pp 26–31, Copenhagen, June 2002.

[9] T. Horprasert, D. Harwood and L.S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection." *In IEEE ICCV99 FRAME-RATE Workshop*, Kerkyra, Sept 1999.

[10] 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 Proc. IAPR Int. Conf. on Pattern Recognition*, pp 4627–4631, Barcelona, Sept 2000.

[11] E.C. Pece, "From Cluster Tracking to People Counting." *In IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, pp 9–17, Copenhagen, June 2002.

[12] J. Shi and C. Tomasi, "Good Features to Track." *In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp 593–600, Washington, June 1994.

[13] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking." *In Proc. IAPR Int. Conf. on Computer Vision and Pattern Recognition*, pp 246–252, Fort Collins, June 1999.

[14] G. D. Sullivan, "Visual Interpretation of known objects in constrained scenes" *Phil. Trans. R. Soc. Lon.* , B, 337, pp 361–370, 1992.

[15] V. Vu, F. Bremond and M. Thonnat, "Automatic Video Interpretation: A Novel Algorithm for Temporal Event Recognition." *In Proc. Int. Joint Conf. on Artificial Intelligence*, Acapulco, Aug 2003.

[16] T. Xiang and S. Gong, "On the Structure of Dynamic Bayesian Networks for Complex Scene Modelling." *In Proc. Joint IEEE Int. Workshop on VS-PETS*, pp 17–22, Nice, Oct 2003.

[1] However, this paper does not necessarily represent the opinion of the EU, and the EU is not responsible for any use which may be made of its contents.