Automated Scene Understanding for Airport Aprons

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

[1] Computational Vision Group, The University of Reading, UK,
{J.Ferryman,M.Borg,D.J.Thirde}@reading.ac.uk
[2] ORION Team, INRIA Sophia-Antipolis, France,
{Florent.Fusier,Valery.Valentin,
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 complete visual surveillance system for automatic scene interpretation of airport aprons. The system comprises two main modules — Scene Tracking and Scene Understanding. The Scene Tracking module is responsible for detecting, tracking and classifying the semantic objects within the scene using computer vision. The Scene Understanding module performs high level interpretation of the observed objects by detecting video events using cognitive vision techniques based on spatio-temporal reasoning. The performance of the system is evaluated for a series of pre-defined video events specified using a video event ontology.

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*). A combination of visual surveillance and video event recognition algorithms are applied in a decentralised multi-camera environment with overlapping fields of view (FOV) 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 data is transmitted to a central server where fused object tracking is performed. This tracking result is 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 extended periods of time, operating in real-time (12.5 FPS, 720×576 resolution) on colour video streams.

The tracking of moving objects on the apron has previously been performed using a top-down model based approach [10] 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 [9, 7, 13] store an estimate of the static scene, learnt from an initial period of observation, which 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 and update previously observed objects in the current time step. One such method, the Kanade-Lucas-Tomasi (KLT) feature tracker [8] 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 sub-set of the features are visible. 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 frequently 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* [14] 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. The approach adopted for AVITRACK [12] addresses these problems using cognitive vision techniques based on spatio-temporal reasoning, *a priori knowledge* of the observed scene and a set of predefined video events corresponding to the servicing operations to recognise. Previous work was performed on primitive video events; here the focus is on more complex video events corresponding to servicing operations on apron area.

Section 2 details the Scene Tracking module comprising per-camera motion detection, bottom-up feature-based object tracking and finally fused object tracking using the combined object tracking results from the camera agents. Section 3 describes the Scene Understanding 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 per-camera detection and tracking of moving objects, transforming the image positions into 3D world co-ordinates, and fusing the multiple camera observations of each object into single world measurements.

2.1 Motion Detection

For detecting connected regions of foreground pixels, 16 motion detection algorithms were implemented for AVITRACK and evaluated quantitively on various apron sequences under different environmental conditions (sunny conditions, fog, etc.). The evaluation process is described in more detail in [1]. Of these algorithms, the colour mean and variance method was selected [13], after taking into account processing efficiency and sensitivity. This motion detector 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* [6], 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 of finding which object in a video frame relates to which object in the next frame. As the time interval between two frames is small, inter-frame changes are limited, allowing the use of temporal constraints and object features to simplify the correspondence problem.

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 $\boldsymbol{\rho}$. The KLT tracker takes as input the set of observations $\{M_j\}$ identified by the motion detector, where M_j is a connected set of foreground pixels, with the addition of a nearest neighbour spatial filter of clustering radius r_c, i.e., connected components with gaps $\leq r_c$. A prediction P_i^t is then associated with one or more observations, through a matching process that uses the individual tracking results of its features S and their spatial and/or motion information, in a rule-based approach.

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

The use of motion information in the matching process, is based on the idea that features belonging to an object should follow approximately the same motion (assuming rigid object motion). Affine motion models (solving for $w_t^T F w_{t-N} = 0$ [15]) are fitted to each group of k neighbouring features of P_i; then represented as points in a motion parameter space and clustering is performed to find the most significant motion(s) of the object. These motions are subsequently filtered temporally and matched per frame to allow tracking through merging/occlusion and identify splitting events.

2.3 Data Fusion

The data fusion module combines the tracking data seen by the individual cameras to maximise the useful information content of the scene being observed and hence achieve enhanced occlusion reasoning, a larger visible area and improved 3D localisation. Spatial registration of the cameras is performed using per camera coplanar calibration and the camera streams are synchronised temporally across the network.

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

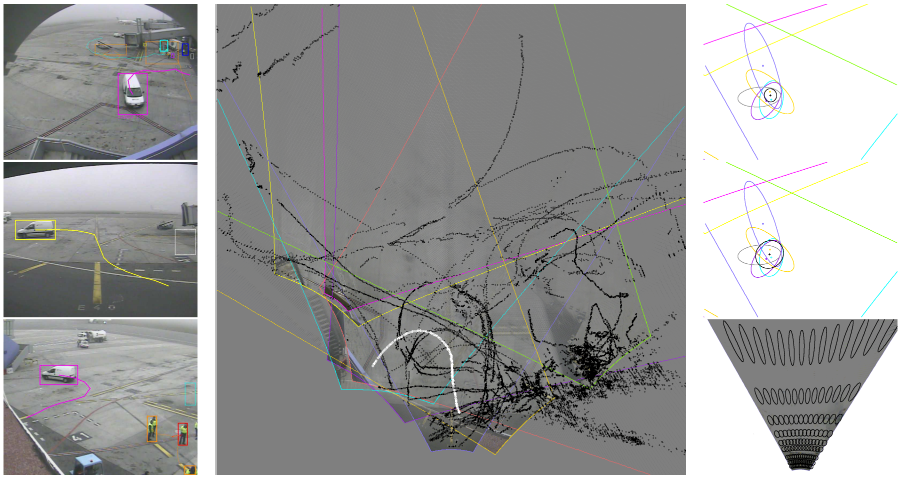


Fig. 1. (Left) Tracking results for 3 cameras for frame 9126 of sequence 21. (Middle) shows data fusion results on the ground-plane for the sequence (9600 frames) with the vehicle track shown in white. (Top-right) the fused observation (in black) for the vehicle (frame 9126) using the covariance accumulation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

estimated by propagating a nominal image plane uncertainty $\boldsymbol{\Lambda}$ such that the measurement uncertainty in the world co-ordinate system is given by [4] i.e. $\mathbf{R}\left(x_w, y_w, z_w\right) = \mathbf{J}\left(x_c, y_c\right)\boldsymbol{\Lambda}\mathbf{J}\left(x_c, y_c\right)^T$ 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: $\mathbf{R}_{fused} = \left(w_1\mathbf{R}_1^{-1} + \ldots + w_N\mathbf{R}_N^{-1}\right)^{-1}$, 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 demonstrated in Figure 1; the covariance accumulation method results in a more localised estimate of the fused measurement than the covariance intersection approach. Remaining 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 events that have been recognised.

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 [5].

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 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 mobile objects can be of different subtypes to represent different vehicles (GPU, Loader, Tanker, Transporter).
- 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 the detection of 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 2, is a change of primitive state values and a composite event is a combination of states and/or events.

```
CompositeState(Vehicle_Stopped_Inside_Zone,
PhysicalObjects((v1 : Vehicle), (z1 : Zone))
Components( (c1 : PrimitiveState Inside_Zone(v1, z1))
            (c2 : PrimitiveState Vehicle_Stopped(v1)))
Constraints(   (c2 during c1)))
```

```
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)))
```

Fig. 2. (Left) The model of the composite state for detecting when a vehicle stops inside a zone of interest. (Right) The model of the primitive event when a vehicle enters a zone of interest.

3.2 Video Event Recognition

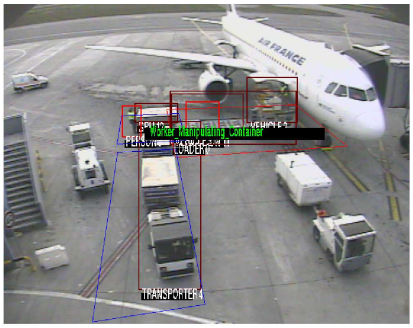
The video event recognition algorithm recognises which events are occurring in a stream of mobile objects tracked by the vision module. The algorithm to recognise a primitive state consists of two operations in a loop: (1) selection of a set of physical objects; then (2) verification of the corresponding 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* [12].

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



```
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)))
```

Fig. 3. (Left) Two dynamic zones (in blue) linked with the Loader and the Transporter vehicles involved in the event "Worker_Manipulating_Container" (event 26) detected. (Right) The Unloading operation involving 8 physical objects and 3 composite components with 2 constraints on the vehicle subtypes, 4 constraints on the zones of interest and 2 temporal constraints.

of the scene. For example, access zones to aircraft can be at different positions according to the aircraft type. In some cases, one needs 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 are 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 Scene 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 This is illustrated in Figure 3). This notion of dynamic context allows more complex scenarios to be defined in which mobile objects can directly interact with each other.

3.4 Predefined Video Events

Currently a set of 21 basic video events has been defined, including 10 primitive states, 5 composite states and 6 primitive events; these are used in the definition of video events representing the handling operations. The primitive states correspond to spatio-temporal properties related to persons and vehicles involved in the scene. Some examples include: a person is located inside a zone of interest, a person is close to a vehicle, a person has stopped, a vehicle is located inside a zone of interest, a vehicle is located outside a zone of interest, a vehicle is close to another vehicle, a vehicle has stopped, a vehicle is moving at a slow pace, and a vehicle is moving at a normal speed.

Using these primitive states, the following composite states have been modelled, such as: a person stays inside a zone of interest, a vehicle has arrived in a zone of interest, a vehicle has stopped in a zone of interest (as shown in Figure 2), a vehicle stays inside a zone of interest, and a vehicle is exceeding the speed limitation. The composite states have in turn been used to model primitive events, such as: a person enters a zone of interest, a person changes from a zone of interest to another, a person leaves a zone of interest, a vehicle enters a zone of interest (as shown in Figure 2), a vehicle change from a zone of interest to another, and a vehicle leaves a zone of interest. These states and events are used in the definition of the composite events (modelling behaviours) representing the apron operations.

Current work has been performed 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 operation and (3) the Loader and Transporter vehicles which are involved in the baggages loading/unloading operations.To recognise these operations 28 composite video events were defined, 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); then it enters (event 3) and stops (event 4) in the "GPU Access Area", the driver gets out of the vehicle (event 5) and deposits the chocks and stud at the location where the plane will stop (events 6 and 7). This operation, and another modelled one, the refuelling operation, are considered to be basic operations because they involve only one person and one vehicle.

The baggage unloading operation is more complex. It 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 the following steps: first, the Loader vehicle arrives in the ERA zone (event 17), enters its restricted area (event 18) and then stops in this zone (event 19); a dynamic zone is automatically added, at the rear of the Loader's 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. The back of the Loader is then elevated (event 24) and the baggage containers are unloaded from the aircraft by the Loader conductor (event 25) one by one. The conductor unloads these containers into the dynamic zone of the Transporter where a worker arrives (event 26) and directs the containers (event 27) on to the Transporter.

4 Results

The Scene Tracking evaluation assesses the performance of the three core components (motion detection, object tracking and data fusion) on representative test data. The performance evaluation of the different motion detector algorithms for AVITRACK is described in more detail in [1]. It is noted that some objects are partially detected due to the achromaticity of the scene and the presence of fog causes a relatively high number of foreground pixels to be misclassified as highlighted background pixels resulting in a decrease in accuracy. Strong shadows also cause problems, often detected as part of the mobile objects. The performance evaluation of the tracking algorithm (representative results shown in Figure 1), is described in more detail in [11]. In is noted that some objects can produce a ghost which remains behind the previous object position. An object is integrated into the background when becomes stationary for an extended time period. In these cases, ghosts are created when stationary objects start to move again. Partial detection of objects can result in fragmentation in tracked objects with similar colour as the background. The Data Fusion module performs adequately given correctly detected objects 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 data fusion can be traced back to motion detection errors i.e. shadow, reflections etc.

The Scene Understanding evaluation has been performed on sequences for which the Scene Tracking module gives good results. Video event recognition has been tested on sequences involving the GPU (aircraft arrival preparation operation), the Tanker (refuelling operation) and the Loader and the Transporter vehicles (baggage unloading). Video events 1 to 4, involving a GPU, have been tested on a dataset of 4 scenes corresponding to 2x4 video sequences (containing from 1899 to 3774 frames and including one night sequence). These events are detected with a perfect True Positive rate. Video events 4 to 8, also involving 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 extending the refuelling pipe to the aircraft (events 14 to 16). The "Unloading Baggage operation" involving the Loader (events 17 to 20, 24 and 25) and the Transporter (events 21 to 23) have been tested on one scene where the point of view allows full observation of the vehicle movements and interactions between the vehicles and people. Currently, the Scene Understanding 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.

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 1. Performance results of the Scene Understanding module for apron monitoring. TP = "Event exists in the real world and is well recognised", FN = "Event exists in the real world but is not recognised", FP = "Event does not exist in the real world but is recognised".

5 Discussion and Future Work

The results are encouraging for both the Scene Tracking and Scene 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, more interactions between people and vehicles. Another issue is incorporating 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).[4]

References

1. J. Aguilera, H. Wildernauer, M. Kampel, M. Borg, D. Thirde, and J. Ferryman. Evaluation of motion segmentation quality for aircraft activity surveillance. 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*, volume 26 num 11, pages 823–843, Nov 1983.
3. Y. Bar-Shalom and X.R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.
4. J. Black and T.J. Ellis. Multi Camera Image Measurement and Correspondence. In *Measurement - Journal of the International Measurement Confederation*, volume 35 num 1, pages 61–71, 2002.
5. M. Thonnat F. Brémond, N. Maillot and V. Vu. Ontologies for video events. In *Research report number 51895*, Nov 2003.
6. T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV’99 FRAME-RATE Workshop*, 1999.
7. 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 Internation Conference on Pattern Recognition*, pages 4627–4631, 2000.
8. J. Shi and C. Tomasi. Good features to track. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
9. C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. International Conference on Pattern Recognition*, pages 246–252, 1999.
10. G. D. Sullivan. Visual interpretation of known objects in constrained scenes. In *Phil. Trans. R. Soc. Lon.*, volume B, 337, pages 361–370, 1992.
11. D. Thirde, M. Borg, V. Valentin, F. Fusier, J.Aguilera, J. Ferryman, F. Brémond, M. Thonnat, and M.Kampel. Visual surveillance for aircraft activity monitoring. In *Proc. Joint IEEE Int. Workshop on VS-PETS*, Beijing, Oct 2005.
12. V. Vu, F. Brémond, and M. Thonnat. Automatic video interpretation: A novel algorithm for temporal event recognition. In *IJCAI’03, Acapulco, Mexico*, Aug 2003.
13. C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Transactions on PAMI*, volume 19 num 7, pages 780–785, 1997.
14. T. Xiang and S. Gong. On the structure of dynamic bayesian networks for complex scene modelling. In *Proc. Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 17–22, Oct 2003.
15. G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*. Kluwer Academic Publ., 1996.

[4] 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.