Video Surveillance for Aircraft Activity Monitoring

Mark Borg, David Thirde,
James Ferryman
Computational Vision Group
The University of Reading
Whiteknights, Reading, RG6 6AY, UK
{M.Borg, D.J.Thirde,
J.M.Ferryman}@reading.ac.uk

Florent Fusier, Valéry Valentin,
François Brémond, Monique Thonnat
ORION Team
INRIA Sophia-Antipolis,
2004 Route des Lucioles BP 93
06902 SOPHIA-ANTIPOLIS, FRANCE
{Valery.Valentin, Florent.Fusier,
Francois.Bremond,
Monique.Thonnat}@sophia.inria.fr

Abstract— **This paper presents a complete 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.**

I. 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 (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 data association and 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 over an extended period of time, and must operate in real-time (12.5 FPS on 720×576 colour video streams).

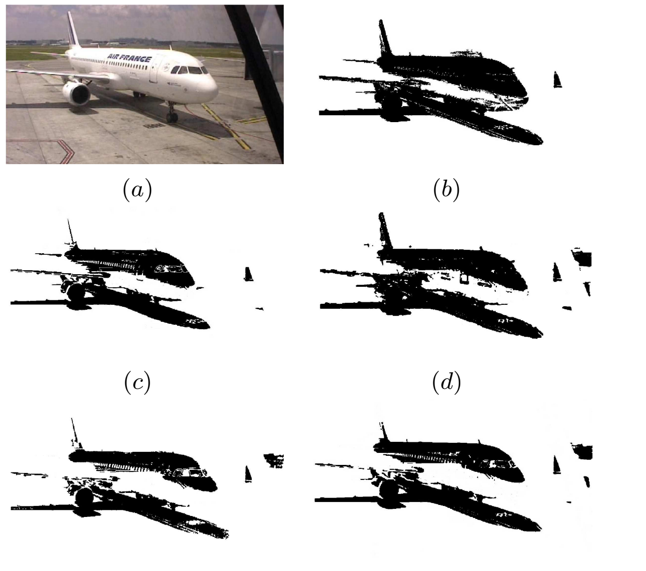
Prior work in scene tracking on the apron used a top-down model based approach [1]; such methods are generally computationally expensive and not generic enough for real-time applications. An alternative approach is bottom-up scene tracking, which refers to a process that comprises the two sub-processes *motion detection* and *object tracking*.

Motion detection methods attempt to locate connected regions of pixels that represent the moving objects within the scene; different approaches include frame to frame differencing, background subtraction and motion analysis (e.g. optical flow) techniques. Background subtraction methods [2], [3] store an estimate of the static scene, accumulated over a period of observation; this background model is used to find foreground (i.e. moving) regions that do not match the static scene. The airport apron, being an outdoor environment, presents 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 predominantly achromatic.

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 [4] combines a local feature selection criterion with feature-based matching in adjacent frames. The CamShift algorithm [5] uses appearance based (colour histogram) representation of objects to perform tracking using the mean-shift algorithm. The work of Pece [6] 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. merging objects, occlusion, fragmentation, 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) causing prolonged occlusions throughout apron operations, and the apron can be congested with objects.

Video event recognition algorithms analyse tracking results spatially and temporally to automatically recognise the high-level activities occuring in the scene; for aircraft servicing analysis such activities occur simultaneously over extended time periods in apron areas. Recent work by Xiang *et al* [7] 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 II details the Scene Tracking and Data Fusion modules. Section III describes the Scene Understanding, representation and recognition of video events, and in Section IV, its application to apron monitoring. Sections V and VI present results, discussion and future work.



(a) (b)

(c) (d)

(e) (f)

Fig. 1. (a) Observed video frame and (b-f) the motion detection results for frame 1500 taken from sequence 'Airport03062004', Camera 3. The motion detection algorithms used are: (b) linear prediction, (c) Gaussian mixture model, (d) colour & edge fusion, (e) kernel density estimation, and (f) colour mean and variance.

II. SCENE TRACKING

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

A. Motion Detection

A motion detector segments an image into connected regions of foreground pixels, which are then used to track objects of interest across multiple frames. 16 motion detectors were implemented and evaluated for AVITRACK. Of these, 5 algorithms were found to perform adequately on a range of test sequences: linear prediction-based method [8], mixture of Gaussians [2], colour and edge fusion method [3], kernel density estimation [9], and colour mean and variance. A representative output of each algorithm is shown in Figure 1. By taking into account processing efficiency and sensitivity, the colour mean and variance method was selected.

B. 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, thus the inter-frame changes are limited, allowing the use of temporal constraints and/or object features to simplify the correspondence problem. The 3 object tracking approaches mentioned in Section I (the KLT local feature tracker [4], the CamShift-based tracker [5], and difference image cluster tracker [6]), have been evaluated in the AVITRACK project. As described in Section V, the KLT local feature tracker was found to given 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 mantained.

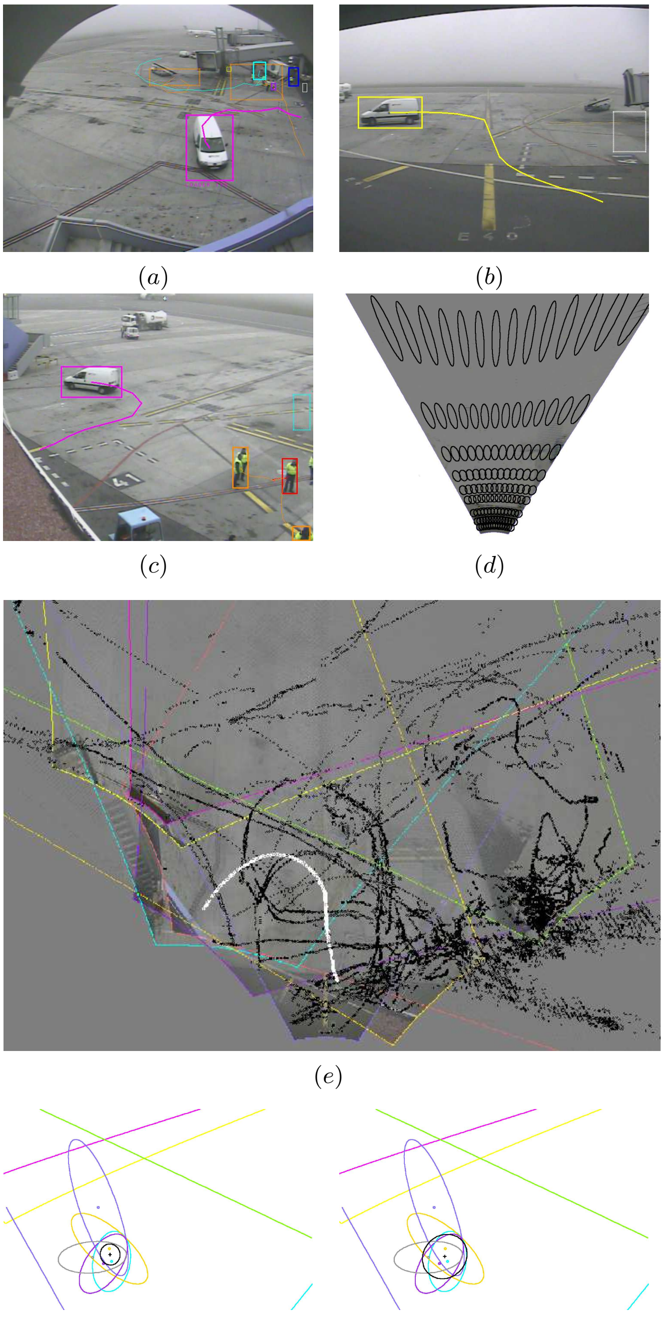
Given a set of tracked objects $\left\{O_i^{t-1}\right\}$ at time $t-1$, and a set of observations $\left\{M_j^t\right\}$ at time t obtained from the motion detector, i.e. connected components of foreground pixels, the tracking process is summarised as follows:

1) Generate object predictions $\{P_i^t\}$ for time t from the list 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) Given a set of observations $\left\{M_j^t\right\}$ detected by the motion detector, match predictions $\{P_i^t\}$ to observations.
4) Any remaining unmatched predictions in $\{P_i^t\}$ are marked as missing observations. Any remaining unmatched observations in $\left\{M_j^t\right\}$ are considered to be 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.

A match function is defined (used for the matching in step 3 above) which returns the number of features W of prediction P_i^t that reside in the foreground region of observation M_j^t, i.e., $f\left(P_i^t, M_j^t\right) = \left|\left\{W : W \in S_{P_i^t}, W \in M_j^t\right\}\right|$ For a non-interacting object, this returns a non-zero value for only one prediction and observation. To handle complex object interactactions, a rule-based approach is adopted. One such rule handles the object splitting case, i.e. several observations at time t match with a single prediction P_i^t. This prediction is split into new objects, one for each of the matched observations, and the features are assigned to the corresponding new object in which they reside in. In this way, features are maintained through an object splitting event. Another rule handles merging objects, where the known local states of the tracked features are used to update the individual predictions whose state cannot be otherwise determined from the 'combined' observation.

C. Data Fusion

The Data Fusion module combines the tracking data obtained from each of the individual cameras to maximise the useful information content of the observed scene and hence achieve enhanced occlusion reasoning, a larger visible area and improved 3D localisation compared to single-camera systems. Spatial registration of the cameras is performed using per camera coplanar calibration and the camera streams are synchronised temporally across the network by the central server to prevent temporal drift between image frames



(a) (b)

(c) (d)

(e)

(f) (g)

Fig. 2. (a-c) The tracking results for 3 cameras for frame 9126 of sequence 21 are shown; (d) shows the sensory uncertainty field measured for camera 6, and plotted on to the ground-plane; (e) shows the data fusion results on the ground-plane for the sequence (9600 frames), with the services vehicle track highlighted in white; (f) shows the fused observation (in black) for the services vehicle using the covariance accumulation method (g) shows the result for the covariance intersection method.

acquired by each camera, which affects the fusion accuracy.

The method for data fusion is based on a nearest neighbour Kalman filter approach [10]. The measurement noise covariance ($\mathbf{R}$) is estimated by propagating a (predefined) image plane covariance to the world co-ordinate system at a given value of height for the world location. This estimation of measurement uncertainty allows formal methods to be used to determine the association of observations originating from the same measurement, as well as providing mechanisms for fusing the observations into a single, estimated, measurement. For the measurement covariance $\mathbf{\Lambda}$ at location (x, y) in the image plane of camera c, the measurement uncertainty $\mathbf{R}\left(x_w, y_w, z_w\right)$ at a given height $z_w = 0$ (i.e. the ground plane) in the world co-ordinate system is given by [11] i.e. $\mathbf{R}\left(x_w, y_w, z_w\right) = \mathbf{J}\left(x_c, y_c\right) \mathbf{\Lambda} \mathbf{J}\left(x_c, y_c\right)^{\mathrm{T}}$ 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 demonstrated in Figure 2 for camera 6.For each detected object in the image plane, the measurement location (x_c, y_c) and associated uncertainty $\mathbf{\Lambda}$ is also dependent on the dimensions of the object and a bias can be introduced for larger objects (i.e. vehicles) to increase the uncertainty in the ground plane location of the object; this is currently achieved using a heuristic strategy that incorporates the angle of the camera to the ground plane and the vehicle size. In the data association stage, the per-camera world co-ordinate observations are matched to existing tracks using the nearest neighbour search strategy.

A validation gate is applied to limit the potential matches in the association step. The gate is determined by a threshold τ on the normalised innovation squared distance between the predicted track states and the observed measurements:

$$d^2 = \left(\mathbf{H}\widehat{\mathbf{X}}_k^- - \mathbf{Z}_k\right)^{\mathrm{T}} \mathbf{S}_k^{-1} \left(\mathbf{H}\widehat{\mathbf{X}}_k^- - \mathbf{Z}_k\right) \tag{1}$$

where $\mathbf{S}_k = \mathbf{H}\widehat{\mathbf{P}}_k^- \mathbf{H}^{\mathrm{T}} + \mathbf{R}_k$ is the innovation covariance and $\mathbf{Z}_k$ is the observed measurement at time k. An appropriate threshold can be determined from standard tables of the chi-square distribution with the degrees of freedom determined by the dimensionality of the measurement vectors. For matched observations, the location and uncertainties are combined to find the optimal *fused* estimate of the object's location and uncertainty. This is achieved using two methods — *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 except that the observation uncertainty covariances are weighted in the summation:

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

$$\text{where } w_i = \frac{w'_i}{\sum_{i=1}^{N} w'_i} \; \textit{and} \; w'_i = \frac{1}{\mathrm{Tr}\left(\mathbf{R}_i^c\right)}$$

and $\mathbf{R}_i^c$ 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 2 for the "Services Vehicle" object; the covariance accumulation method gives a more localised estimate of the fused measurement than the covariance intersection approach. Any remaining unassociated measurements are clustered into new tracks, using a validation gate between observations to constrain the association and fusion steps.

III. Scene Understanding

To detect high-level activities, cognitive vision techniques based on spatio-temporal reasoning, *a priori* knowledge of the observed environment and a set of predefined event models are used. The Scene Understanding task is based on a Video Event Recognition module that takes the tracked mobile objects from the previously described modules as input, and outputs events that have been recognised.

A. A Priori Knowledge

The *a priori* knowledge of the environment corresponds to a 3D empty scene model of the observed environment and a set of video events models. The empty scene model contains static information about the contextual objects (e.g. equipments, zones of interest, airport walls, jet-bridge) characterised by their 3D geometry (approximative shape) and semantics (e.g. how they interact with personnel). The video event models are predefined by domain experts (e.g. managers of handling companies) using a video event description language described in [12].

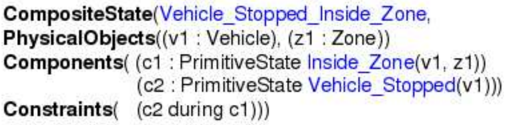
B. Video Event Representation

The aim of video event representation is to formalise the knowledge for the system to be able to detect video events. The description of this knowledge has to be declarative and intuitive (in natural terms), so that the experts in the aircraft activity monitoring can easily define and modify it. A video event model E is composed of five components:

- a set of Physical Object variables corresponding to the physical objects involved in E (any contextual objects including static objects (equipment, zones of interest) and mobile objects (people, vehicles, 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 the detection of E
- a set of constraints (symbolic, logical, spatial and temporal constraints including Allen's interval algebra operators [13]) involving these variables
- a set of decisions corresponding to the tasks predefined by experts that are needed to be executed when E has been detected (e.g. to launch an alarm or to display a message on a window)

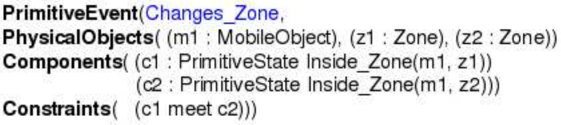
There are four types of video events: primitive states, composite states, primitive events and composite events. A state describes a situation characterising one or several physical objects defined at time t or a stable situation defined over a time interval. A primitive state (e.g. a person is located inside a zone) corresponds to a visual property directly computed by the Scene Tracking module. A composite state as shown in Figure 3 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 enters a zone of interest: it is outside the zone and then inside the zone). A primitive event (as shown in Figure 4) 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)))
```

Fig. 3. The model of the composite state "Vehicle_Stopped_Inside_Zone" is composed of two components and one constraint.



```
PrimitiveEvent(Changes_Zone,
PhysicalObjects( (m1 : MobileObject), (z1 : Zone), (z2 : Zone))
Components( (c1 : PrimitiveState Inside_Zone(m1, z1))
            (c2 : PrimitiveState Inside_Zone(m1, z2)))
Constraints(  (c1 meet c2)))
```

Fig. 4. The model of the primitive event "Changes_Zone" is composed of two components and one constraint.

C. Video Event Recognition

The video event recognition algorithm recognises which events are occurring in a stream of mobile objects tracked by the Scene Tracking module. The algorithm to recognise a primitive state consists of 2 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 states that match 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 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 that of primitive events. The video event recognition algorithm is detailed in [14].

IV. SCENE UNDERSTANDING FOR APRON MONITORING

A. Dynamic contextual information

In the Apron Monitoring application, contextual information of the empty scene model containing the contextual objects (e.g. equipment, zones of interest, for example, access or stopping zones) is required – this is static and fixed contextual information. Dynamic contextual information about vehicles (e.g. Ground Power Unit (GPU), Loader, Tanker) and aircraft is also required and is defined in the local coordinate system of the vehicles and aircraft. To compute this dynamic contextual information in the global coordinate system, the context manager module computes a transformation matrix using the orientation and the position of the tracked mobile objects. Then, for parked vehicles and aircraft, the dynamic zones in the global coordinate system are added into the static context (empty scene model) to recognise people and vehicle interactions. This allows, for example, the recognition of a person exiting a vehicle (e.g. the person appears in the dynamic zone corresponding to the vehicle door which has been added to the empty scene model when the vehicle is parked – this is shown in Figure 5.)

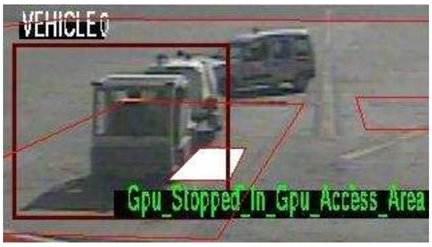
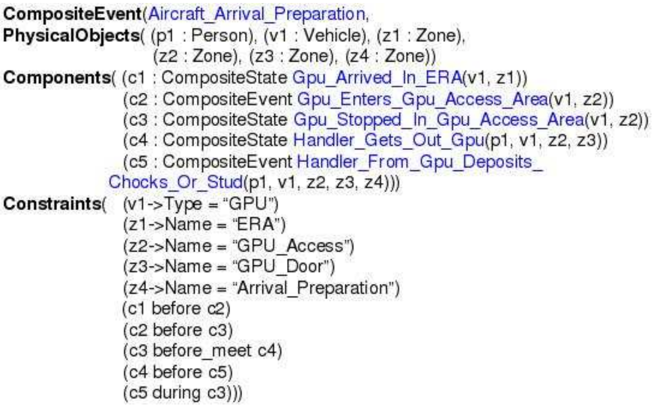


Fig. 5. Dynamic zone representing the vehicle door added in the empty scene model after the vehicle becomes stationary.

B. Predefined Video Events

Presently, a generic set is defined of 5 primitive states, 4 composite states and 3 primitive events, needed for the recognition of handling operations. In addition to these generic video events, the current focus is on handling operations involving only a vehicle and/or a person. The main test was performed for the "Aircraft Arrival Preparation" event. This operation involves a vehicle (GPU), its driver (Handler) and four zones of interest. The system recognises that the GPU arrives and stops in the access area and then the driver gets out from the GPU and deposits the chocks and the stud at the location where the plane will stop. This operation (shown in Figure 6) is recognised when the 5 video events involving the GPU have been detected and the constraints verified.

The video event corresponding to the "Refueling Operation" has also been modelled. The system has to recognise that the Tanker arrives and stops in the "Refueling Area" and then the driver exits from the vehicle and refuels the aircraft. The video event recognition module has been tested on a first stage of the "Refueling Operation" corresponding to the part when the Tanker is getting ready to refuel the aircraft. The model of the composite event "Tanker_Arrival" contains 3 physical objects, 3 components and 5 constraints.



```
CompositeEvent(Aircraft_Arrival_Preparation,
PhysicalObjects( (p1 : Person), (v1 : Vehicle), (z1 : Zone),
                 (z2 : Zone), (z3 : Zone), (z4 : Zone))
Components( (c1 : CompositeState Gpu_Arrived_In_ERA(v1, z1))
            (c2 : CompositeEvent Gpu_Enters_Gpu_Access_Area(v1, z2))
            (c3 : CompositeState Gpu_Stopped_In_Gpu_Access_Area(v1, z2))
            (c4 : CompositeState Handler_Gets_Out_Gpu(p1, v1, z2, z3))
            (c5 : CompositeEvent Handler_From_Gpu_Deposits_
         Chocks_Or_Stud(p1, v1, z2, z3, z4)))
Constraints(  (v1->Type = "GPU")
              (z1->Name = "ERA")
              (z2->Name = "GPU_Access")
              (z3->Name = "GPU_Door")
              (z4->Name = "Arrival_Preparation")
              (c1 before c2)
              (c2 before c3)
              (c3 before_meet c4)
              (c4 before c5)
              (c5 during c3)))
```

Fig. 6. The model of the composite event "Aircraft_Arrival_Preparation" contains 6 physical objects, 5 components and 10 constraints.

V. RESULTS

This section describes evaluation of the Scene Tracking and the Scene Understanding modules.

For motion detection, the 5 algorithms listed in Section II-A were evaluated. All these methods had acceptable susceptibility to noise, although detection noise was encountered on thin object components (e.g. the aircraft wing edge) due to aliasing and JPEG artifacts. The algorithms were reasonably robust to illumination changes. The colour and edge fusion technique was the most sensitive for detection in low contrast regions; linear prediction also had a good detection sensitivity, often finding moving objects undetected by the other techniques. The most computationally efficient algorithms were the colour mean and variance and the Gaussian mixture model.

Figure 1 demonstrates the common failing of the detection algorithms tested; with false negatives due to the similar appearance (at pixel level) of the aircraft body and background, and false positives detected in shadowed background regions. The false negatives generally have negligible effect on the estimated bounding box dimensions, since edges of moving object regions are mostly detected. False positives (caused by strong shadow) present a much greater challenge since existing shadow/highlight detection methods generally rely on colour information. AVITRACK datasets contain predominantly achromatic regions (moving and stationary) causing such methods to fail. By taking into account processing efficiency as well as sensitivity, the colour mean and variance method was selected for the AVITRACK project.

The local feature tracking method gives the best results, and representative results are shown in Figure 2. It can suffer from loss of object identity if the local features are lost during merged or occluded states and cannot be replenished fast enough. It also requires objects to be textured in order for good features to be selected. The results of the difference image clusters method are promising, but it suffers from tracking failures when clusters merge together.The colour tracking method performs quite badly because of the achromatic nature of the scene.

The Data Fusion module performs adequately given iso-

TABLE I

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"

Event	Sequences	TP	FP	FN
GPU				
Event 1	4 scenes * 2 cam.	8	0	0
Event 2	4 scenes * 2 cam.	8	0	0
Event 3	4 scenes. * 2 cam.	8	0	0
Event 4	4 scenes. * 2 cam.	8	0	0
Event 5	2 scenes * 1 cam.	2	3	0
Event 6	2 scenes * 1 cam.	2	0	0
Event 7	2 scenes * 1 cam.	2	0	0
Event 8	2 scenes * 1 cam	2	0	0
Tanker				
Event 9	2 scenes * 1 cam.	2	0	0
Event 10	2 scenes * 1 cam.	2	0	0
Event 11	2 scenes * 1 cam.	2	0	0
Event 12	2 scenes * 1 cam.	2	0	0
Event 13	2 scenes * 1 cam.	2	0	0

lated targets correctly detected in the frame tracker (representative results shown in Figure 2). Uncertainty information is incorporated 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 data fusion can be traced back to motion detection errors i.e. shadow, reflections etc.

The evaluation performed on the Scene Understanding module, was undertaken on the sequences on which the Scene Tracking module gives good results. The video events involving a GPU have been tested on a dataset of 4 scenes corresponding to 8 video sequences (containing from 1899 to 3774 frames and including one night sequence) showing the "Aircraft Arrival Preparation" and 2 scenes showing the "Tanker Arrival" (see Table I). The Scene Understanding module has been tested on the two best camera viewpoints from where the GPU can be observed and on the only viewpoint from where the Tanker can be observed.

The current evaluation is mainly qualitative and performed manually with no ground truth. The aim is to provide an idea of the performance of the Scene Understanding and to anticipate potential problems in event detection for apron monitoring. The result of the performance evaluation shows that all video events are recognised correctly (45 TPs) with very few false alarms (3 FPs) and no miss detection (0 FNs). The results are very encouraging but it must be considered that situations in which the Scene Tracking misdetects or overdetects mobile objects are not yet addressed. The events 5, 6, 7 and 8 are only detected on one of the two tested cameras because this part of the scene is only observed by one camera. The FPs of event 5 are due to too vague a model of event 5. This event should be detected when the handler (driver of the GPU vehicle) exits from the vehicle in a predefined zone near the door of the vehicle (called "Gpu_Door"). FPs occur when a person walks in this zone and is analysed by the system as exiting the GPU.

VI. Conclusions 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. The video event recognition results prove that Scene Understanding can be applied to apron monitoring. The main difficulty in using the video event recognition module for apron monitoring is to model the handling operations using expert knowledge and to attempt to recognise them all in parallel and in real time.

Future work in Scene Tracking will address more effective shadow supression, explicit occlusion analysis, and improved data fusion. Future work in Scene Understanding will consider also the recognition of more complex operations (e.g. "baggage loading/unloading") involving more people and vehicles. Further planned work includes modelling uncertainty to enable recognition of events even when the Scene Tracking module loses physical objects or gives unreliable output.

VII. Acknowledgments

This work was supported by the European Union, grant AVITRACK (AST3-CT-3002-502818)[1]. The AVITRACK website is located at www.avitrack.net.

References

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

[2] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking." *In Proc. International Conference on Pattern Recognition*, pp 246–252, 1999.

[3] 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. International Conference on Pattern Recognition*, pp 4627–4631, 2000.

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

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

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

[7] T. Xiang and S. Gong, "On the Structure of Dynamic Bayesian Networks for Complex Scene Modelling." *In Proc. Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pp 17–22, Nice, France, October, 2003.

[8] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance." *In Proc. International Conference on Computer Vision*, pp 255–261, 1999.

[9] A. Elgammal, D. Harwood and L. Davis, "Non-parametric model for background subtraction." *In IEEE ICCV'99 FRAME-RATE WORKSHOP*, 1999.

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

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

[12] F. Brémond, N. Maillot, M. Thonnat and V. Vu. "Ontologies for Video Events." *Research report number 51895*, INRIA Sophia-Antipolis, Nov 2003.

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

[14] V. Vu, F. Brémond, M. Thonnat, "Automatic Video Interpretation: A Novel Algorithm for Temporal Event Recognition." *IJCAI'03*, Acapulco, Mexico, 9-15 Aug 2003.

[1] 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.