Towards Feature-Based Situation Assessment for Airport Apron Video Surveillance

Ralf Dragon[1], Michele Fenzi[1], Wolf Siberski[2],
Bodo Rosenhahn[1], and Jörn Ostermann[1]

[1] Institut für Informationsverarbeitung
{dragon,fenzi,rosenhahn,ostermann}@tnt.uni-hannover.de
[2] L3S
siberski@l3s.uni-hannover.de
Leibniz Universität Hannover, Germany

Abstract. We present a feature-based surveillance pipeline which, in contrast to traditional image-based methods, allows to learn a detailed description of the observed background as well as of foreground objects. The pipeline consists of motion segmentation of feature trajectories and subsequent tracking-by-recognition with updates. Furthermore, 3D object representations are learned in order to extract the 3D object pose of a later object recognition. Finally, we show how such sufficiently reliable information is inputted into a reasoning system comparing actual and nominal condition of an airport apron. By this, automatic situation assessment becomes possible in a manageable and reliable way.

1 Introduction

Video surveillance is a field in which manual interpretation of camera images dominates. Although it is known that the human assessment of video material is a fatiguing task with a short attention span of approximately 20 minutes [19], computer assistance for operators is still at a very basic level: The usual assistance is activity detection and convenient access to video material of multiple cameras and time instances. Even though there are continuous advances in this field, most approaches still suffer from high false positive rates or they are very specific to certain setups, e.g. abandoned bag detection [2] or traffic analysis [6]. Furthermore, recent advances in object tracking, crowd analysis, face recognition, and unusual event detection are not integrated into commercial systems since they are too complex to handle or compute, or since their output is too noisy for an automatic situation assessment.

Summing up, computer vision approaches in surveillance allow remarkably-well results in certain disciplines, but the high-level classification "is everything alright?" has not been tackled yet. Besides the problem of imprecise knowledge about the actual condition of the scene, the nominal condition (the background knowledge) is also not present. This is crucial for detecting unusual events and surveillance in general, since critical events cannot be trained by example.

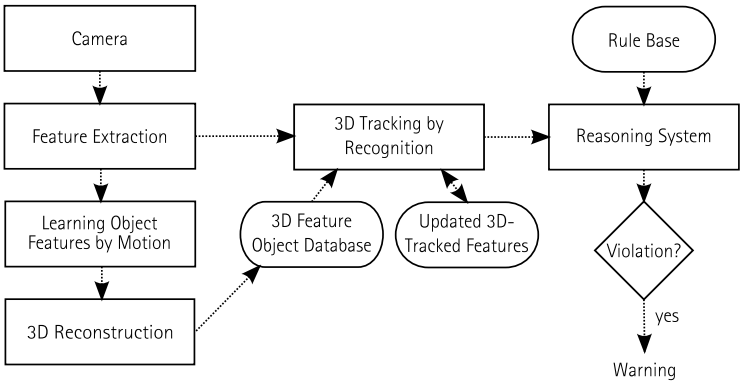


Fig. 1: System overview for a feature-based situation analysis system. For simplicity, the 3D tracking-by-recognition pipeline of only one camera is displayed. In a complete system, multiple tracking-by-recognition systems from different cameras are connected to one inference system.

In this paper we propose a solution to both problems. As displayed in Figure 1, we extract scene knowledge by the *tracking by recognition* approach. Since such a feature-based surveillance system is sufficiently reliable, we can use its output as facts in a reasoning system. Here, the actual condition is compared with a nominal condition. If both states differ in a critical way, a warning is generated in order to steer the operator's attention. Such reasoning systems are widely used in medical applications and are thus manageable and reliable at the same time.

Our approach targets the automatic situation assessment for event-based video surveillance (ASEV) on airport aprons. However, since the methods used are quite general, they can be transferred easily to other scenarios. In the apron scenario, the conflict between privacy and safety is very high since ramp staff is monitored all the time and safety concerns are big. Since the whole approach can be applied without knowing the original images, we believe that privacy as well as safety can be enhanced.

This paper is organized as follows: In Section 2, we show how the image-based pipeline in object tracking can be replaced by a feature-based which enables learning object features by motion segmentation. By this, tracking by recognition can be used which is very robust and allows to deal with long-time occlusions. In Section 3, we show that on top of this, the 3D feature point cloud of an object can be learned, which is used for 3D tracking by recognition. In Section 4, we describe our reasoning system. In Section 5, we demonstrate how to create background masks for privacy protection and to direct the attention of the operators. Finally, a conclusion is given in Section 6.

2 Feature-based vs. Image-based Surveillance

2.1 Image-based Surveillance

Traditional image-based approaches reason on a stream of camera images $I_t(\boldsymbol{x})$, where $\boldsymbol{x}$ is a coordinate of a pixel in image I_t which may contain intensity, color, and depth information. To detect and track objects in a scene, the change in multiple images I_t is analyzed. A comprehensive survey can be found in [31]. In surveillance scenarios, two approaches are commonly used: background removal and optical flow.

In background removal, a binary foreground mask $F(\boldsymbol{x})$ is defined for every $\boldsymbol{x}$ by learning the probabilistic distribution p_b of the background

$$F(\boldsymbol{x}) = \begin{cases} 1 & p_b(I(\boldsymbol{x})|\boldsymbol{x}) < \tau_i \\ 0 & \text{otherwise}\,. \end{cases} \tag{1}$$

In early approaches [41], p_b was modeled by a pixel-wise Gaussian $\mathcal{N}(\mu(\boldsymbol{x}), C(\boldsymbol{x}))$ containing the two parameters standard deviation $\mu(\boldsymbol{x})$, which denotes an average background image, and variance, or covariance matrix respectively, $C(\boldsymbol{x})$, which describes the variability of pixel $\boldsymbol{x}$ over time. More recent approaches use multivariate Gaussian distributions [39], non-linear colorspaces [34], and hierarchical modeling instead of pixel-wise [10]. In a post-processing step, obvious errors in F like very small or very elongated objects are deleted. Furthermore, special methods for shadow and reflection handling like [37] are applied. Since the background model itself is learned and updated using F, drifting occurs if the background is hidden by foreground objects for a long time or if foreground is falsely classified as background[3]. An example for this is if a foreground object is looking similar to the background (cf. Figure 2). On the other side, if the background is classified as foreground, it is not updated and the modeling becomes worse (cf. Figure 3). The main idea to circumvent this is to learn the background model from long time spans (one day or more) which has a very high computational complexity and which is not responsive if the background changes. In terms of artificial intelligence, the approaches suffer from the adaptivity vs. plasticity dilemma [8].

Optical flow approaches analyze the spatial difference between two consecutive images I_{t-1} and I_t, to find the discrete displacement field $\boldsymbol{D}(\boldsymbol{x})$ by

$$\arg\min_{\boldsymbol{D}} \|I_{t-1}(\boldsymbol{x}) - I_t(\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{x}))\|_2\,. \tag{2}$$

[3] These approaches are *recursive* as the learning is performed on previous classifications. There also exist *non-recursive* background models which estimate p_b on the basis of N_t previous images, e.g. by computing the pixel-wise median [7]. Such approaches are not taken into account since N_t must be much larger than the amount of frames a foreground object may rest still. By this, the computational complexity becomes too high and the model looses responsiveness since an update would take N_t frames.

Fig. 2: Background removal using a multivariate background probability. Displayed are input image I_t, the average background μ, and the foreground mask F. It can be observed, that the shirt of the left person is only partially detected as foreground since it looks similar to the background. Furthermore, shadows on the floor are detected as foreground.

(a) Original view at time t_0. (b) Illumination changes and reflections at $t_0 + 10$s. (c) Foreground from the background model of (b).

Fig. 3: Diverged Gaussian mixture background model caused by illumination changing faster than the model adapted.

By assuming a static camera, the foreground F can be found from $\boldsymbol{D}$ by

$$F(\boldsymbol{x}) = \begin{cases} 1 & \|\boldsymbol{D}(\boldsymbol{x})\| > \tau_f \\ 0 & \text{otherwise}\,. \end{cases} \tag{3}$$

However, determining the optical flow in (2) is complex and since foreground objects do not necessarily move (e.g. a car waiting before traffic lights), this method can only be used as a prior for (1).

To recapitulate: In image-based methods only the background is described, as, in contrast to the foreground, it can be learned over a long time. However, the performance of the various approaches is still not sufficient for many real-world applications and high gains are still to be achieved [30]. A stable prior for motion segmentation is the optical flow which is very complex even for two consecutive images, but for reliable detection motion has to be analyzed over longer time spans. In the following section we propose to adapt the methods of image-based surveillance to feature-based. This has the advantage, that background as well as foreground can be learned and that motion can be analyzed over longer time spans.

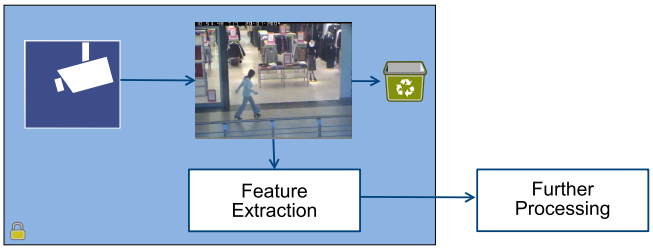


Fig. 4: A feature-camera (blue frame) captures images, computes local features and only exports the features.

2.2 Tracking in Feature-based Surveillance

In the past decade, the combination of interest point detectors like SIFT [24] and Harris-affine [26] with local descriptor like SIFT, GLOH [27] and MSER [25] has been successfully applied in a high number of computer vision problems. The main reason for that is the fact, that establishing local image correspondences, which is one of the main computer vision problems, can be solved by inexpensive descriptor matching. Since local image descriptors are used to establish correspondences, such feature-based approaches are able to cope with partial occlusions and clutter. The descriptors are intentionally built such that changes in illumination as well as scaling and rotation of the image plane leaves them mostly unchanged. Thus, the main problem in modeling, namely that the background changes due to illumination, is suppressed up to a high degree when using local image descriptors. This can be observed in Figure 5, in which the illumination changes, which caused a background model to diverge in Figure 3, are still acceptable in order to establish correspondences between the two images. This gives hope that we can describe the background by means of features.

In the case of video surveillance, privacy protection plays an important role. Since for feature-based methods no original image data is needed, a *feature camera* could be used. As depicted in Figure 4, the features are extracted inside the camera such that no image data leaves the camera. By this, unauthorized access to the camera images becomes by far harder[4].

2.3 Learning Object Features by Motion Segmentation

In this section we demonstrate that by using local features, the background can be described even if the camera moves. Furthermore, we can also describe foreground objects and by this learn their local features. In contrast to image-based modeling, our feature-based approach distinguishes objects by their motions, not their appearance. Thus, we extract the foreground by motion segmentation instead of building a pixel-wise foreground mask.

[4] Recently, methods to reconstruct images from local image features were proposed [40]. By this, the global scene layout could be recovered remarkably well. However, details cannot be not recovered with this method since they are mainly hallucinated.

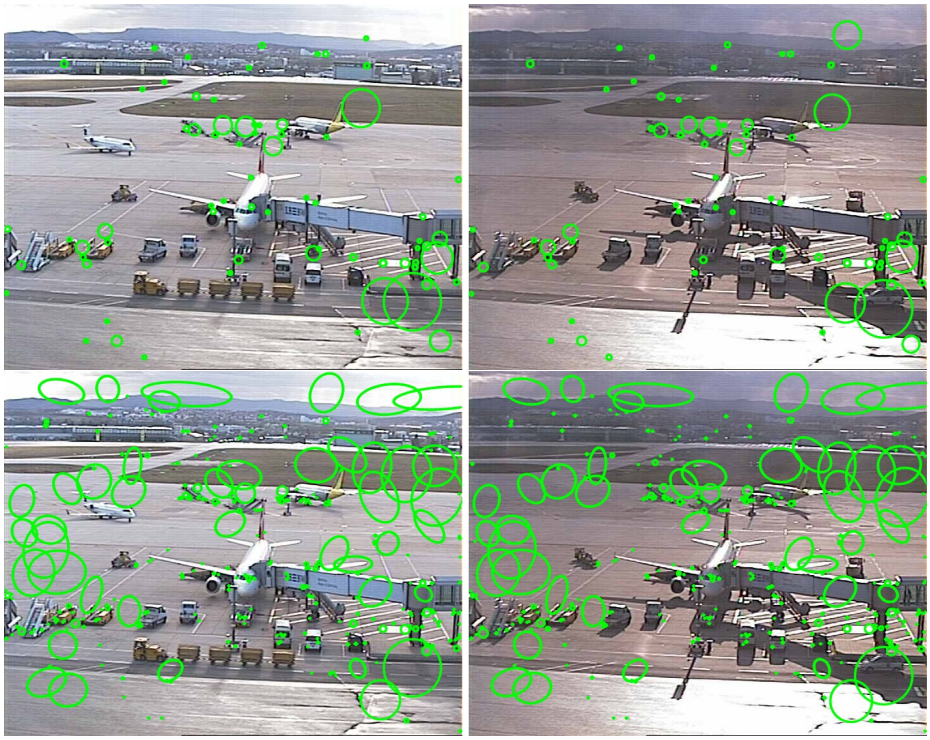
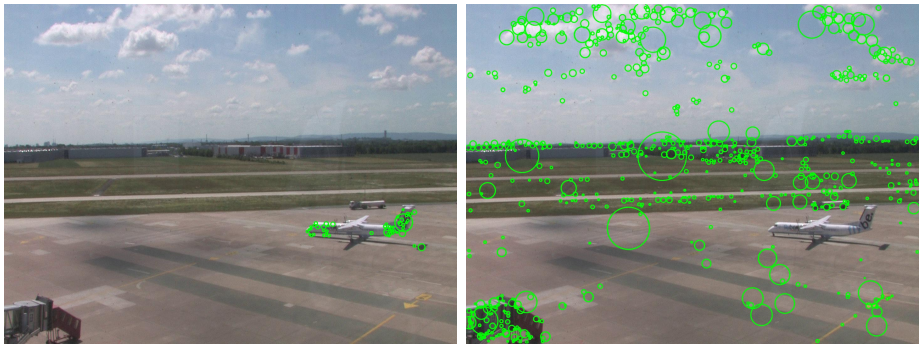


Fig. 5: SIFT correspondences (top), and complementary NF feature [11] correspondences (bottom) between the views in Figures 3(a) and (b) which caused the image-based background model to diverge. From the point-of-view of illumination invariant features, the images are similar since the correspondences cover wide areas.

In the field of motion segmentation, feature trajectories T_i are clustered into groups of common motion. In the surveillance context, the camera is far from the object. By this, motion groups correspond to objects with different motions and motion segmentation becomes equivalent to object segmentation (cf. Figure 6). Motion segmentation approaches can be differentiated into subspace- and affinity-based approaches. Subspace-based approaches like [9, 12, 13, 42] assume complete trajectories T_i which are inserted into a data matrix W. Since rigid object trajectories form linear subspaces in W, different object motions can be segmented by analysis of these subspaces. However, since we cannot provide complete data, we use an affinity-based approach like [5, 17]. Here, the square affinity matrix A is computed which consists of pair-wise affinity measures $a_{i,j}$ between trajectories T_i and T_j. In these measures, the spatial distance between T_i and T_j as well as their similarity in motion is included. In a final spectral clustering [28] step, the association of the trajectories to motion clusters is found.



(a) Foreground object (b) Background

Fig. 6: Motion segmentation of SIFT features. Although the airplane performs a turning operation in which perspective effects are non-negligible, *motion* segmentation corresponds to *object* segmentation.

Motion segmentation is applied to trajectories found from subsequent feature correspondences. In order to achieve longer trajectories, we apply the trajectory repair idea from [38]. The problem is computationally tractable since only a window over a time range sufficient for motion segmentation needs to be analyzed, here 5 s. Furthermore, the here-used independently-detected features allow satisfactory results when matching over a time span of 0.5 s, which is much longer than tracked features like KLT [36]. Thus, we analyze windows of $N = 10$ frames taken at 2 Hz. By this, we can reliably segment motion if it is noticeably fast during the given window size and if the motion consists of enough features. The first constraint could be weakened by enlarging the window size. Regarding the second constraint, we deal with this by using high image resolution (up to 1.5 Mpel) or by using pan-tilt-zoom (PTZ) cameras scanning the scene with high zoom until they detect motion. To our experience, objects need to own approximately 10 to 15 detected features in order to reliably get detected (cf. Figure 7).

Motion segmentation extracts sets of local features $\mathcal{M}_i(t)$, corresponding to different motions i at frame t. We store these sets in the object feature database $O = \{\mathcal{M}_1(t_1), M_2(t_1), \ldots, \mathcal{M}_1(t_2), \ldots\}$ in order to compare the features with later input data. In contrast to image-based approaches, the background is treated as a regular object. As demonstrated in Figure 8, this allows using non-static cameras like PTZ cameras performing camera motion in the analyzed frames. Since the objects are described by their features and their geometric alignment, illumination changes as well as shadows and reflections do not pose major problems.

Fig. 7: Motion segmentation of small objects. Compared to the image resolution of $1440 \times 1080\,\mathrm{pel}^2$, the objects are quite small with approximately $110 \times 70\,\mathrm{pel}^2$ (left) and $100 \times 100\,\mathrm{pel}^2$ (right). Similarly, the number of features (18 and 15, respectively) is only a fraction of the global scene (966 and 891, respectively).

Fig. 8: Motion segmentation under panning and tilting. The four images are taken during a time span of 2 s. As it can be observed, that although the segmented airplane is moving slowly compared to the panning, it is segmented correctly.

Fig. 9: Four views automatically learned from motion-segmented feature trajectories of the foreground object in Figure 6(a).

3 3D Object Learning and Recognition

The results of motion segmentation as described in Section 2.3 could be used for multiple instance learning (cf. Figure 9). However for critical applications such as airport surveillance, 2D object recognition is usually not sufficient for a fully-informed operation as it merely permits to find the 2D camera coordinates of the object location. In contrast, 3D tracking allows to recover complete 3D information, such as object location, pose and motion direction, expressing them with respect to a world coordinate frame. In these coordinates, safety rules of an airport apron can easily be expressed (cf. Section 4), e.g. the rule "only the scheduled airplanes may enter the taxi ways".

One of the possible choices for the detection and tracking framework is the model-based approach pioneered in [18, 32], where SIFT features [24] are used to reconstruct in an off-line fashion a 3D point cloud representing the target object. Once the model database has been assembled, on-line recognition and tracking can be performed by establishing putative correspondences between 3D model and current frame features and then estimating the 3D object pose.

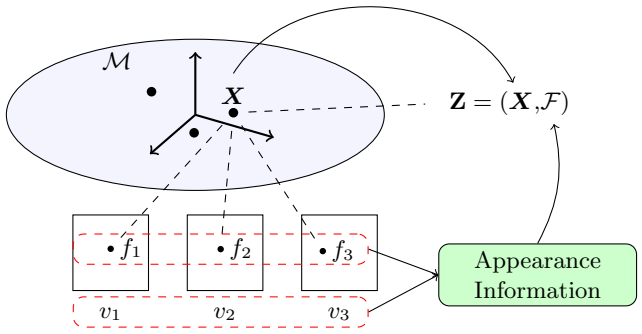


Fig. 10: 3D feature $\mathbf{Z}$ built from views v_1, v_2, and v_3. The 3D descriptor $\mathcal{F}$ is established from the corresponding 2D descriptors f_1, f_2, and f_3, respectively.

3.1 3D Object Learning

The tracking-by-recognition approach requires first to build a set of 3D object models in an off-line stage. Here we start from a set of training views which are automatically found from motion-segmented trajectories $T_i = (\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n})$ (cf. Section 2.3 and Figure 6). So like in similar approaches [3, 20, 21, 23, 29], we detect and track SIFT features over visually close training views. The trajectories are input to a Structure from Motion (SfM) algorithm that outputs a 3D point cloud that represents the object structure. The feature descriptors $f_{i,j}$, observed at the respective 2D positions $\boldsymbol{x}_{i,j}$ to which a corresponding 3D point X_i projects, are provided together with the 3D point coordinates. By doing so, it is possible to implement a 3D-2D feature matching at the recognition and tracking stage.

Since the set of descriptors can be highly redundant, particularly in case of long tracks, many of the above methods employ a feature quantization step. In [20], a hierarchical quantization is used for preserving matching ambiguities until the pose estimation step, where incoherent matches are dropped. Feature quantization can also be motivated by dimensionality reduction, as the 3D model size is usually too large to keep the system operating in real-time. This approach is shared by [3] and [21], where feature quantization is applied for outdoor scene reconstruction and image registration, respectively.

In order to form a 3D feature $\mathbf{Z}$, for each 3D point $\boldsymbol{X}$ we compute a 3D feature descriptor $\mathcal{F}$ containing appearance information from multiple views by applying a high-dimensional mean-shift clustering to the set of corresponding features:

$$\mathbf{Z} = (\boldsymbol{X}, \mathcal{F}). \tag{4}$$

In Figure 10, an example of building of a 3D feature $\mathbf{Z}$ is shown. In this case, $\mathcal{F}$ contains the matching 2D descriptors f_1,f_2,f_3 from views v_1,v_2,v_3.

3.2 3D Object Recognition

Once the model database has been assembled, the general on-line operation envisages the creation of 2D-3D correspondences between frame features and model features, and the estimation of the pose by solving the projection problem.

A set of features is extracted in each frame and it is matched against the model feature set by using one of the following matching strategies. The most straightforward approach is to match the entire set of detected features against the model feature set by using a matching strategy based on the second-nearest-neighbor (2nn) distance ratio, as proposed by [24]. That is, a match (f,f_{nn}) between a feature f and its nearest-neighbor feature f_{nn} is considered to be correct if

$$d(f,f_{nn}) < d(f,f_{2nn}) \cdot \tau, \tag{5}$$

where $d(\cdot,\cdot)$ is an appropriate distance metric, f_{2nn} is the second nearest neighbor for f in the $d(\cdot,\cdot)$ metric, and τ is a threshold, given as 0.7 in the original paper [24]. Since the 2nn distance ratio strategy was conceived in order to reject false matches due to background clutter, it may remove many true positives if repetitive patterns or texture symmetries occur on the object surface. In [20] countermeasures are proposed based on dropping the 2nn strategy and employing hierarchical feature quantization and pose estimation constraints. Matches are created by thresholding their normalized cross correlation and stored along with their 3D location. Potentially spatial incoherent matches are kept until a pose estimation step, where geometric constraints will single out the true matches and discard the others. On the contrary, the 2nn approach can be maintained if difficult feature arrangements are handled by spatially clustering the original image feature set, e.g., by using mean shift clustering. Since features tend to cluster over the object surface, individual objects can be isolated before the matching step and thus, ambiguities can be avoided.[5] A visual example of the usefulness of spatial feature clustering is given in Figure 11.

Once feature clusters are established, object recognition and pose estimation is performed on the clusters, as represented in the block diagram given in Figure 12. Attention has to be payed as clustering may split or merge objects, visible in Figure 11.

After the matching, putative correspondences are established. Given a set of N 2D-3D correspondences $(\boldsymbol{x}_i,\boldsymbol{X}_i)$, a projection matrix $\mathbf{P}$ is to be computed such that

$$\mathbf{P} = \arg\min_{\tilde{\mathbf{P}}} \sum_{i=1}^{N} D(\boldsymbol{x}_i, \tilde{\mathbf{P}}\boldsymbol{X}_i)^2 \,. \tag{6}$$

Thus, $\mathbf{P}$ minimizes the sum of the squared re-projection errors D over all correspondences. Since the putative matches set contains outliers, a statistically robust approach is typically used in order to estimate the mathematical model

[5] Of course the motion segmentation methods from Section 2.3 could be applied here, too. However this clustering method only works if an object is currently moving. Thus, the tracking-by-recognition paradigm would be dismissed.

Fig. 11: By spatial feature clustering, the objects in the scene are segmented into five different clusters. This increases the inlier-outlier ratio for each cluster, and permits to avoid mismatches due to objects having a similar appearance, as in the case of the three airplanes.

underlying the samples. One of the most used algorithms is Random Sample Consensus (RANSAC) [14], in which a minimal subset of samples is iteratively used to estimate the model parameters, and the rest of the samples ranks the model consensus and can eventually be used to refine the parameters themselves.

If the minimal subset is created by randomly selecting the samples, no additional information regarding the importance of each sample and the relations among the samples themselves is used. Several approaches have been proposed in the literature in order to guide the RANSAC sampling by exploiting properties or constraints among the samples. E.g., in [20], a geometrical constraint based on the co-visibility of the 3D points in the sample set is used. After the first sample has been chosen, the part of the remaining samples that do not share any common view in the training images is discarded. This concept can be easily extended by giving the samples a weight-based priority computed from additional 3D information. For each feature, we compute weights on the basis on their frequency in the training images and of their co-visibility with other samples. These weights guide RANSAC towards a better selection, thus improving the final robustness and accuracy of the estimated pose. [23] only exploits feature priority in the matching step for the purpose of speeding up the process. Instead of using all model features for matching, they propose to use a subset

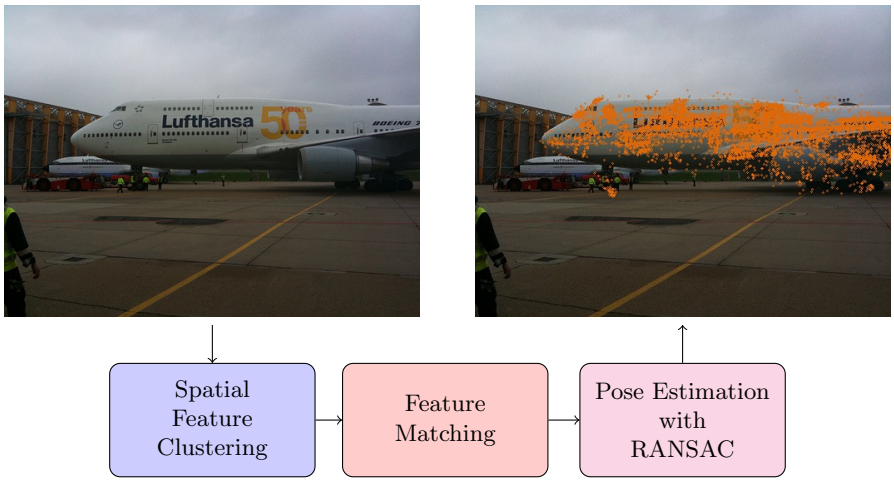


Fig. 12: On-line stage. SIFT features are detected and spatially clustered. Each cluster is matched against a database object and its pose is estimated using RANSAC. The model cloud is reprojected in orange to show the precision of the estimated pose.

of features selected on the basis of priorities representing feature frequency and co-frequency.

In Table 1, an overview of the contribution of the guided sampling in terms of average iteration count for different inlier ratios is given until at least 75% of the inliers are found. The results are averaged over 1000 runs per frame for a short video sequence. It can be observed that our method is highly beneficial in real-time applications where the permitted number of iterations is small.

Table 1: Mean and standard deviation of the number of iterations for different inlier ratios.

Inlier ratio	No weight	Guided Sampling
60%	39.9 ± 40.6	$\mathbf{5.8 \pm 6.2}$
50%	110.5 ± 113.3	$\mathbf{9.4 \pm 12.6}$
40%	309.0 ± 286.7	$\mathbf{17.4 \pm 19.9}$
30%	627.4 ± 515.0	$\mathbf{19.0 \pm 27.6}$
20%	1428.5 ± 1294.6	$\mathbf{29.1 \pm 56.1}$

After the minimal subset is determined, a method for estimating the pose $\mathbf{P}$ given in Eq. (6) that best fits the 2D-3D point pairs is used. This is called the Perspective-n-Point (PnP) problem. The algorithms for estimating the pose presented in the literature are countless, e.g., DLT, clamped DLT, POSIT, P4P,

etc., and therefore the choice depends mainly on the complexity and time constraints given by the application considered. In case of real-time applications, like the one at hand, the Enhanced PnP (EPnP) method is a very common choice. It guarantees speed, as its complexity is only $O(n)$, and accuracy at the same time, as shown in detail in [22]. Finally, the estimated pose that holds the maximum consensus among the entire set of correspondences is returned and thus, the object is considered detected.

3.3 3D Object Tracking

Regarding object tracking, different strategies are typically presented in the literature, as, e.g., tracking-by-recognition. Advantages of the later method are the absence of error drift and the fact that tracking failures do not affect successive frames as each frame is treated separately.

However, the appearance between the object model and its current projection on the image plane may vary too much. This can be due to the fact that the model was created off-line from a finite and small number of views and that SIFT features do not offer enough invariance. Therefore, it proves to be a hard problem if the object pose is far from the training images. In order to cope with this, [20] and [21] have proposed adding synthetic features created by deforming the training images in an affine way and extracting the features out of them. A clear disadvantage of this approach is the increase in size of the model, which can enlarge by more than one order of magnitude. Further, the distinctivity is lowered. A possible alternative is to use a model updating stage, where the model description is augmented after it has been detected. As a matter of fact, a matching image descriptor provides a reasonable approximation of the appearance of the same 3D point in the following frame. By this, the detection rate is boosted without a loss in precision. Some further images showing the tracking performance of our system are given in Figure 13.

4 Reasoning on Streams of Object Recognitions and Detections

The aim of the ASEV (automatic situation assessment for event-based video surveillance) system is to detect potentially safety-critical situations based on the image analysis results. To achieve this goal, the detected status is continually checked against safety rules, and violations are displayed as warnings to video surveillance operators. This section explains the challenges involved into this task, and how they were solved.

The reasoning component uses Semantic Web standards to represent the relevant expert knowledge. The airport domain is modeled using the Web Ontology Language (OWL Lite, [1]), including types and properties of objects found on the airport ramp, in particular the different vehicle types (cf. Figure 14). Safety rules cannot be expressed in OWL Lite, therefore this knowledge is captured as classical logical rules, represented in the Rule Interchange Format (RIF, [4]). Figure 15

Fig. 13: Example of the 3D tracking of a toy plane. Different situations are presented: blank background, clutter and a combination of clutter and occlusion. The model cloud is reprojected in orange to show the preciseness of the estimated pose.

shows as example a distance rule between moving planes and any other vehicle. These static expert knowledge is taken from official safety procedures, e.g. [15], from airport-internal guidelines, from work plans and flight schedules. During runtime, the facts describing the current situation are added to the knowledge base. These facts are generated by the object recognition algorithms described in Section 3.

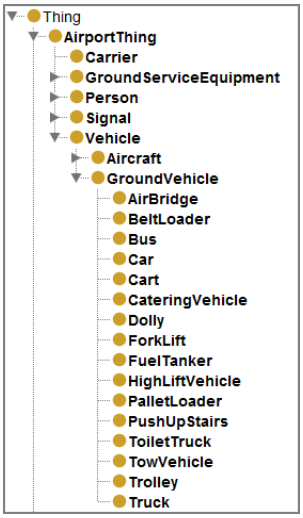


Fig. 14: Part of the airport ontology.

Usual reasoning systems rely on the assumption that the knowledge based is rather static, while users pose a variety of queries over time. These systems are optimized to index and preprocess the facts and rules such that any arbitrary query can be processed efficiently. However, an update of the knowledge base invalidates intermediate results and requires a complete recomputation [33]. For the airport surveillance context, this assumption does not hold. New facts arrive every second, while only one query is ever posed to the system, i.e., "is there a safety critical situation?" In addition, the majority of incoming facts are not new, but fact updates concerning the position, orientation, and speed of planes and vehicles on the ramp. Therefore, existing reasoning engines could not be used to process the incoming object detection event stream efficiently.

Instead, we implemented a novel reasoning system, based on the Rete algorithm [16]. This algorithm works as follows: In an offline step, the domain knowledge captured in rules is converted into a directed graph, consisting of two types of nodes, α- and β-nodes. α-nodes represent conditions expressed in one of the rules, and β-nodes join these conditions. The leafs of this graph are *productions* which generate additional facts derived through the rule network. Figure 16 shows a part of the Rete network for the distance rules from Figure 15. Arriving

```
# Rule1: any vehicle with speed ¿ 0 is moving
If And( rdf:type(?v asev:Vehicle) asev:speed(?v ?s) numeric-greater-than(?s, 0.0) )
Then Assert( asev:moving(?v) )

# Rule2: any aircraft with active anti-collision beacon is moving
If And( rdf:type(?a asev:Aircraft) asev:hasBeacon(?a ?acb) asev:active(?acb "true") )
Then Assert( asev:moving(?a) )

# Rule3: create warning if vehicle distance to moving aircraft is too low
If And(
    rdf:type(?a asev:Aircraft) asev:moving(?a)
    rdf:type(?b asev:Vehicle) asev:distance(?d ?a ?b)
    numeric-less-than(?d, asev:MinDistanceMoving) )
Then
    Assert(rdf:type(?w asev:DistanceWarning))
    Assert(rdfs:member(asev:warnings ?w))
    Assert(asev:participant(?w ?a))
    Assert(asev:participant(?w ?b))
```

Fig. 15: Distance rule between moving aircraft and vehicles, modeled in RIF.

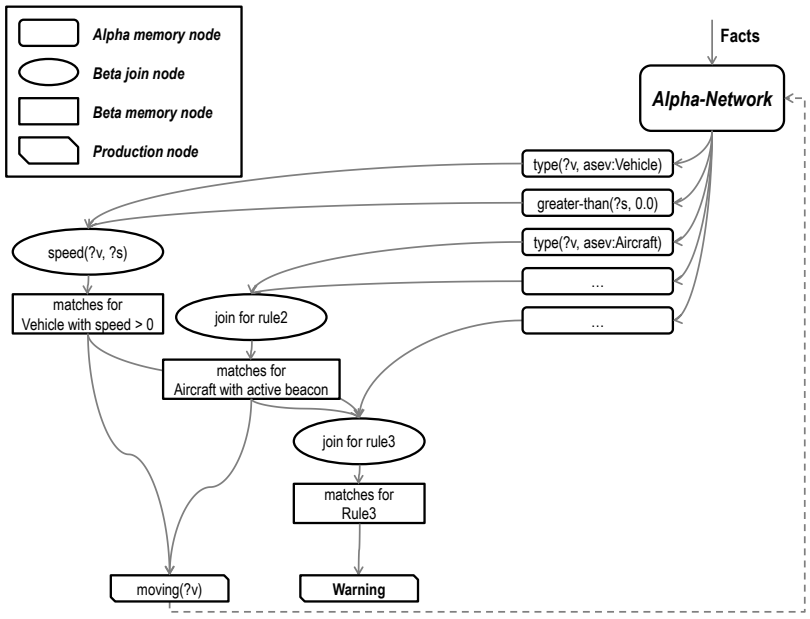


Fig. 16: Sample Rete network for distance rules from Figure 15.

facts are forwarded to all α-nodes, which act as filters (shown on the top right). Matching facts are stored in the corresponding α memory nodes. For example, the top α memory node maintains a list of all objects of type *asev:Vehicle*. If a fact satisfies the constraint represented by an α-node, it is forwarded to all β-nodes which rely on it. These nodes now perform a look-up in their β memory to check if there is a join possible. For example, the leftmost β join node matches objects which are vehicles and have a speed greater than 0. If a match could be found, the result is forwarded to its successors. A successful join at a leaf node triggers a production, which creates a derived fact. These facts are fed back into the Rete-network to possibly derive further facts.

In the original Rete algorithm, nodes hold references to related facts. To optimize this approach for fact updates, we introduce an additional reverse index, which allows a lookup of all α- and β-nodes maintaining this fact in their memory. When an update for a fact arrives, this enables very efficient updating of the respective node memories, to take the new value into account.

The reasoning engine is connected to the video analysis component via an XML event stream. The tracking-by-recognition component sends high-level object attributes such as type, position, speed, etc., and updates their values based on the analysis of each frame. If a safety-critical situation is detected, the reasoning engine sends a warning message to the video operator application.

5 Logging and Controlling Access to Surveillance Data

Let us imagine a feature-based system like the one presented here reports a critical event. An operator would then like to have a view on the scene before he takes further steps. If pure feature-cameras (cf. Figure 4) are used, this is not possible since no image signal leaves the camera. However, by introducing a system which controls access to images and logs this access, the use of surveillance image data becomes transparent. The following access rules are self-explaining:

- Since the scene content is known, it is logged which operator observes which object. Thus, mis-use by stalking is documented. Furthermore, regions with irrelevant information can masked out (cf. Figure 17). In order to provide context to an operator, such regions may instead be faded out or blurred.
- An operator is allowed to get access to image data only if a critical event is detected. Overriding this is possible, but it is logged.

In order to quickly mask parts of the image, we use a method similar to the feature-based background removal from [35]. Given a set of features $\mathcal{X}_+$ and $\mathcal{X}_-$ which should be visible or not, respectively, we search for a binary segmentation $s(\boldsymbol{x})$, which is determined using spatial background and foreground probabilities p:

$$s(\boldsymbol{x}) = \begin{cases} 1 & p(\boldsymbol{x}|\mathcal{X}_+) > p(\boldsymbol{x}|\mathcal{X}_-) \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Fig. 17: Dense segmentation from SIFT feature density of the moving airplane and the resting fueling vehicle from the sequence displayed in Figure 9.

The spatial probabilities are estimated from kernel density estimation of $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ as

$$p(\boldsymbol{x}|\mathcal{X}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_i, C_i) \tag{8}$$

using the normal kernel $\mathcal{N}$ with adaptive bandwidth C_i, estimated from the covariance of the nearest 10% neighbors of $\boldsymbol{x}_i$.

By this, the attention of operators can be directed to important objects and unimportant image parts can be masked out. Furthermore, security and privacy is enhanced at the same time.

6 Conclusion

In this paper, we have shown the concepts of the feature-based surveillance system ASEV (automatic situation assessment for event-driven video surveillance). It consists of a 3D tracking-by-detection system which inputs 3D information about visible objects into a reasoning system. By this, the current condition can be compared with a nominal condition which is specified by a rule set. Furthermore we have shown how to learn 3D models from motion and how foreground masks can be created from sets of foreground and background features.

Compared to traditional image-based surveillance, feature-based surveillance has the advantage that it is much more robust towards changes in illumination or background motion. Furthermore, our ASEV system allows to describe the foreground as well, which in turn enables tracking through long-time occlusions. The reasoning system facilitates comprehensive and reliable output event messages to operators. Since the scene interpretation can be used to mask out non-related scene content, the attention of surveillance operators is directed and privacy is enhanced at the same time.

Acknowledgement: Research was conducted inside the BMBF-funded project ASEV.

References

1. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: Owl web ontology language reference. Tech. rep., World Wide Web Consortium (2009)
2. Bhargava, M., Chen, C.C., Ryoo, M.S., Aggarwal, J.K., Aggarwal, J.K.: Detection of object abandonment using temporal logic. Mach. Vis. Appl. pp. 271–281 (2009)
3. Bhat, S., Berger, M.O., Sur, F.: Visual words for 3D Reconstruction and Pose Computation. In: The First Joint 3DIM/3DPVT Conference (Mar 2011)
4. Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A., Reynolds, D.: Rif core dialect. Tech. rep., World Wide Web Consortium (2010)
5. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: Proc. ECCV. pp. 282–295 (2010)
6. Buch, N., Velastin, S., Orwell, J.: A review of computer vision techniques for the analysis of urban traffic. Intelligent Transportation Systems, IEEE Transactions on 12(3), 920 –939 (Sep 2011)
7. Calderara, S., Melli, R., Prati, A., Cucchiara, R.: Reliable background suppression for complex scenes. In: Proc. ACM Video Surveillance and Sensor Networks (VSSN). pp. 211–214 (2006)
8. Carpenter, G.A., Grossberg, S.: The art of adaptive pattern recognition by a self-organizing neural network. Computer 21(3), 77–88 (Mar 1988)
9. Chen, G., Lerman, G.: Motion segmentation by scc on the hopkins 155 database. Proc. ICCV Workshop on Dynamical Vision (2009)
10. Chen, Y.T., Chen, C.S., Huang, C.R., Hung, Y.P.: Efficient hierarchical method for background subtraction. Pattern Recogn. 40, 2706–2715 (Oct 2007)
11. Dragon, R., Shoaib, M., Rosenhahn, B., Ostermann, J.: Nf-features - no-feature-features for representing non-textured regions. In: Proc. ECCV (Sep 2010)
12. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Proc. CVPR. pp. 2790–2797 (2009)
13. Favaro, P., Vidal, R., Ravichandran, A.: A closed form solution to robust subspace estimation and clustering. In: Proc. CVPR. pp. 1801–1807 (2011)
14. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24(6), 381–395 (1981)
15. Flight Safety Foundation: Ground accident prevention ramp operational safety procedures. `http://flightsafety.org/archives-and-resources/ground-accident-prevention-gap`
16. Forgy, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence 19(1), 17 – 37 (1982)
17. Fradet, M., Robert, P., Perez, P.: Clustering point trajectories with various life-spans. Proc. CVMP pp. 7–14 (2009)
18. Gordon, I., Lowe, D.: What and Where: 3D Object Recognition with Accurate Pose. In: Toward Category-Level Object Recognition. pp. 67–82 (2006)
19. Green, M.W.: The appropriate and effective use of security technologies in u.s. schools. Tech. rep., Sandia National Laboratories (Sep 1999)
20. Hsiao, E., Collet, A., Hebert, M.: Making Specific Features Less Discriminative to Improve Point-Based 3D Object Recognition. In: CVPR. pp. 2653–2660 (2010)
21. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From Structure-from-Motion Point Clouds to Fast Location Recognition. In: CVPR. pp. 2599–2606. IEEE (2009)

22. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An Accurate O(n) Solution to the PnP Problem. IJCV 81, 155–166 (Feb 2009)
23. Li, Y., Snavely, N., Huttenlocher, D.P.: Location Recognition Using Prioritized Feature Matching. In: ECCV. pp. 791–804 (2010)
24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (Nov 2004)
25. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing (2004)
26. Mikolajcyk, K., Schmid, C.: An affine invariant interest point detector. In: ICCV (2002)
27. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
28. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Proc. NIPS. pp. 849–856 (2001)
29. Park, Y., Lepetit, V., Woo, W.: Multiple 3D Object tracking for augmented reality. In: ISMAR. pp. 117 –120 (Sep 2008)
30. Parks, D.H., Fel, S.S.: Evaluation of background subtraction algorithms with post-processing. In: AVSS. pp. 192 – 199 (2008)
31. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: a systematic survey. IEEE Transactions on Image Processing 14(3), 294–307 (2005)
32. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. IJCV 66(3), 231–259 (2006)
33. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, chap. 9: Inference in First-Order Logic. Prentice Hall (2009)
34. Setiawan, N., Seok-Ju, H., Jang-Woon, K., Chil-Woo, L.: Gaussian mixture model in improved hls color space for human silhouette extraction. In: Advances in Artificial Reality and Tele-Existence, vol. 4282, pp. 732–741. Springer Berlin / Heidelberg (2006)
35. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: Proc. ICCV. pp. 1219–1225 (2009)
36. Shi, J., Tomasi, C.: Good features to track. In: CVPR (jun 1994)
37. Shoaib, M., Dragon, R., Ostermann, J.: View-invariant fall detection for elderly in real home environment. In: PSIVT (Nov 2010)
38. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. IJCV 67(2), 189–210 (2006)
39. Stauffer, C., Grimson, W.E.L., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR. pp. 2246–2252 (1999)
40. Weinzaepfel, P., Jégou, H., Pérez, P.: Reconstructing an image from its local descriptors. In: CVPR (2011)
41. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 780–785 (1997)
42. Yu, J., Chin, T.J., Suter, D.: A global optimization approach to robust multi-model fitting. In: Proc. CVPR (2011)