ViCoMo : visual context modeling for scene understanding in video surveillance

Published in:
Journal of Electronic Imaging

DOI:
10.1117/1.JEI.22.4.041117

Published: 01/01/2013

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 07. Dec. 2018
ViCoMo: visual context modeling for scene understanding in video surveillance

Abstract. The use of contextual information can significantly aid scene understanding of surveillance video. Just detecting people and tracking them does not provide sufficient information to detect situations that require operator attention. We propose a proof-of-concept system that uses several sources of contextual information to improve scene understanding in surveillance video. The focus is on two scenarios that represent common video surveillance situations, parking
1 Introduction

Automatic understanding of human behavior in surveillance video is one of the ultimate goals for computer vision research, in order to selectively direct the attention of human operators to potentially suspicious activities. More specifically, the importance of video monitoring of crowds has increased recently in response to heightened security concerns. To cut costs, human operators often observe many video feeds simultaneously, a job that is both error prone and tedious. Also, the chance of missing an important event in one of the many surveillance videos increases. One of the challenging aspects of automatically understanding human behavior, which is not often considered in research, is the presence of contextual information. Just detecting and tracking a human or group of humans does not give sufficient information to automatically decide if the group behavior is normal or abnormal. In many cases, more contextual information is required from the scene in order to interpret human actions in more detail. Contextual information can contain the locations of roads, zebra crossings, cars, or (traffic) signs. For example, a car detected on a parking place is a normal situation whereas a stationary car detected on tramway rails is a reason to raise an alarm. One of the central goals of the European visual context modeling (ViCoMo) project as part of the ITEA2 research program has been using contextual information to improve scene understanding, and this article describes the result from the collaboration of several ViCoMo partners.

The work has been previously published on various related topics. One of these is the field of action recognition. In this field, the goal is to classify the actions of humans in videos into several categories (i.e., running, falling, walking, etc.). Examples of publications are the work by Efros et al.,1 Minnen et al.,2 and Parameswaran and Chellappa.3 This work is not suitable for our application because it does not take any contextual information into account. While recognizing actions in video is certainly interesting, without using contextual information it can be difficult or even impossible to interpret if a situation needs operator attention. Another interesting research topic is that of multitarget or group tracking. The work of both Lanz4 and Bazzani et al.5 shows methods of tracking groups of people in surveillance applications. Again, we can comment that without contextual information it is difficult to automatically recognize potentially dangerous or suspicious activities. In the field of scene classification, images or frames from a video are categorized as belonging to a category, i.e., tall buildings, forest, office, etc. Relevant work by Battiatto et al.6 describes a bag-of-words (BoW)-based method using Texton distributions within a spatial hierarchy. Further, development by Farinella et al.7 showed a method of performing this categorization in the frequency domain, matching performance of other state-of-the-art solutions. Next, Farinella and Battiatto8 improved on this previous result by making the algorithm work directly on DCT coefficients, which are available in images coded in JPEG format. This led to a computationally very efficient algorithm that was implemented by Battiatto et al.9 on a mobile platform, the Nokia N900 smartphone. Van de Sande et al.10 published work on the topic of scene recognition. They use color descriptors to improve performance on a video concept classification dataset, and show that it increases performance by 7% on this task. Parizi et al.11 describe another approach to scene recognition, their algorithm represents a scene as a collection of parts, and they use a latent variable classifier to partition the scene into a set of predefined regions. Although certainly interesting, this research has a different goals rather than scene recognition. Our goal is scene understanding, more specifically understanding the actions of the persons in the scene by using contextual clues. There is some available literature on the topic of scene understanding. For example, the research work of both Li et al.12 and Malisiewicz and Efros13 shows methods that can recognize multiple objects in images as well as provide a segmentation of those objects. However, these works are aimed at single images and our application is video surveillance. Their approaches do not seem feasible for real-time video processing. Additionally, these approaches lack decision making to decide when to raise operator alarms, and have not been shown to work on videos of crowds, which is considered to be challenging footage in surveillance literature. Finally, the work of Marques et al.14 introduces a method that uses contextual information to improve the detection and understanding of the rest of the scene. While certainly interesting, the application is again somewhat different from ours, where we aim at exploiting contextual information to interpret human behavior, and not primarily to improve the detection and recognition of the rest of the scene.

In this article, we develop and integrate several computer vision algorithms to automatically detect some of these contextual clues, and afterwards, use these clues to interpret human actions. To test this approach, two scenarios are proposed. The first scenario encompasses cars that park in parking places or parking places for the disabled, in which drivers are automatically tracked with a pan–tilt–zoom (PTZ) camera after leaving their vehicle. The second scenario contains groups of humans that perform various activities such as crossing roads and zebra crossings, gathering near a bus stop, fighting, and panicking. In the first scenario, we introduce a PTZ tracking algorithm for an unconstrained outdoor environment. We apply this algorithm to track persons who illegally park in a parking spot for the disabled. To analyze the behavior of groups of people in the second scenario, we present a novel group detection algorithm15 which detects split-and-merges of groups, panic situations, and group inactivity.

In order to detect the contextual information, we introduce the following approaches. We present our novel traffic sign detection and classification algorithm16 for detecting and classifying various traffic signs that occur in the scenes, in addition to license plates on cars. Additionally, we develop a fast region labeling algorithm17 to classify pixel regions of
the video into broad categories such as sky, vegetation, water, road, construction, and zebra. Finally, a decision engine is designed to interpret the results from all the components and decide if the scene contains suspicious activity. For example, it will be shown that the people gathering and their behavior can be successfully classified in various scenarios, depending on their mutual interaction and the surrounding context information. This introduces and enables the automated analysis of video surveillance events that are much more complex than possible with the conventional object detection and tracking techniques.

The remainder of the article is structured as follows. Section 2 describes the parking lot surveillance scenario, with the PTZ human tracker in Sec. 2.2 and the traffic sign recognition system in Sec. 2.1. Section 3 introduces the group behavior scenario, with the human group analysis system in Sec. 3.2 and the region labeling system in Sec. 3.3. The results of the proof-of-concept systems and their components can be found in Sec. 4. The parking lot surveillance system in Sec. 3.2 and the region labeling system in Sec. 3.3. The classification of the group behavior of the second scenario and corresponding results of the subsystems (group behavior analysis and region labeling) can be found in Sec. 4.2. Finally, the conclusions and future work are located in Sec. 6.

2 Scenario 1: Context-Based Object Detection and Tracking

Many surveillance cameras are deployed for parking lot observation. The typical actors in parking lots are humans and cars, which, therefore, need to be reliably detected and tracked. Instead of using a conventional static camera, we propose to use a PTZ camera for this scenario. This has the advantage of being able to survey a large area while still having the ability to zoom-in on persons or cars to see more details of the objects of interest. The manual operation of such a system is labor intensive because humans constantly have to adjust the camera direction and zoom level and watch for activity. This limits their ability to simultaneously keep track of multiple cameras.

This scenario can be analyzed and followed with an automatic surveillance system. Our automated PTZ tracker embedded in the PTZ camera has an overview preset in which it zooms out completely, so that it has a good overview of the entire parking lot. When a car is detected (by the license plate detector), the camera zooms in and monitors the car in more detail to enable eventual license plate recognition. If the car then proceeds to park at a parking space for the disabled (as indicated by a disabled parking sign), this is detected and the operator is notified if this license plate is not registered for disabled parking. After parking the car, the car driver can be tracked and even zoomed-in upon to identify the person. The system implementing the fundamentals of this scenario contains two components: (1) the PTZ-tracker that can track a moving object and control the camera to follow this object and (2) the traffic sign and license plate detector, which supplies valuable contextual information about the scene. In addition to these two components, an event-level decision algorithm is used to interpret the results from these subsystems and control the PTZ tracker as appropriate and raise alarms to the operator if necessary. The context information about parking place for the disabled and the involved traffic signs supply the context information for the car and person-based surveillance and decision making. In this section, we describe a prototype for an automatic surveillance monitoring system for this scenario (Fig. 1).

2.1 Traffic Sign and License Plate Detection and Classification

Traffic sign detection and recognition using panoramic images (see Fig. 2) have become important for the making of a sign inventory to support road maintenance and safety. Similar developments can be seen for computer vision applications in the car industry and the exploitation of pictorial city databases on the Internet. An example is the commercial in-car traffic sign detection system by Mobileye that can detect various common traffic signs in real time while driving. All these applications are based on using photographic collections to survey real-world objects such as traffic signs to design efficient working processes that are partially automated by computer vision algorithms. The resulting traffic sign information can be used for maintenance purposes, road safety analysis, change detection (when performed yearly) and navigation applications (e.g., locations/changes in one way streets, speed limits, etc.). de la Escalera et al. present a traffic sign detection and recognition system for intelligent vehicle applications. Colors are first thresholded into specified traffic sign colors (i.e., red, white, blue, etc.), after which a genetic algorithm is employed to detect traffic sign shapes in the thresholded colors. Sign recognition is performed based on neural networks and a custom color representation. Ruta et al. introduce a traffic sign recognition system that works in real time on video. The detection stage is again based on color thresholding followed by equiangular polygon detection. Sign recognition is based on finding class-specific discriminative image regions using feature selection techniques. These approaches are both interesting and suitable for real-time applications, however, our experience with color thresholding is that it is challenging to make this stage robust to the widely varying lighting conditions and sign conditions that occur in large-scale datasets, and our previous experiments indicate that it can lead to reduced performance.

First, the traffic signs are detected in the photographs, then detections from neighboring panoramas are triangulated to determine which one corresponds to the same physical sign, see Fig. 3. Since the triangulation stage is not relevant to the work in this article, it is not explained in more details (see Ref. 16). Finally, the signs are classified to determine the exact type of traffic sign. Over 100 different traffic signs can be recognized with reasonably good accuracy (about 95% to 97%). Since the datasets for this application are very large (hundreds of thousands to millions of high-resolution panoramic images), efficient detection algorithms have been
developed for this purpose. When this detection algorithm is applied to regular full-HD resolution surveillance video, near real-time performance (5 fps) is obtained, which is more than sufficient for objects that remain mostly stationary.

The detection algorithm finds the presence of traffic signs in the image. Multiple detectors are used to find broad classes of traffic signs. For example, all red circular signs are detected using a single detector, but a different detector is used for red triangular signs. The detection algorithm is based on the popular histogram of oriented gradients (HOG) by Dalal and Triggs. In this algorithm, first the local image gradients are obtained by a simple \[-1, 0, 1\] filter. These values are then converted to polar coordinates. The features are extracted as a histogram in the two spatial dimensions and the orientation dimension. The orientation contribution of each pixel is spread over a total of eight bins, where for each of the two closest orientation bins, the four nearest spatial bins are increased by trilinear interpolation, depending on the distances to the bin centers and the magnitude of the gradient. Finally, local normalization is performed to make the features more invariant to local contrast changes in the image due to different lighting conditions. The classification is performed with a sliding window using a linear support vector machine (SVM) classifier. This corresponds to a convolution of the features with the SVM kernel coefficients.

We have incorporated color information in the HOG features, since this significantly improves the performance for traffic sign detection applications. Furthermore, we have developed a custom color transformation to further improve the detection performance as described in Ref. 23. For each pixel, this method calculates the distance in color space to a set of reference colors, as specified by

\[ p_t = \| p - p_r \|. \]  

(1)

Here, the transformed pixel \( p_t \) is calculated from the input pixel \( p \) using a reference color \( p_r \). The reference colors are chosen to be the set of colors that commonly occur in traffic signs: red, blue, yellow, white, and black. By applying this transformation, the gradients at the edges of the object are always in a uniform direction, independent of the background of the traffic sign. This has shown to further improve the detection performance.

The traffic sign classification algorithm is performed using the signs that are found by the detector and for which a three-dimensional (3-D) coordinate has been automatically triangulated. Since the number of detected signs is much lower than the number of windows that the detection algorithm needs to process, a more time-consuming algorithm can be used for this purpose. The goal is to determine exactly what type of traffic sign is detected, given the broad classification obtained from the detection algorithm. For this algorithm, the popular scale invariant feature transform (SIFT) descriptor by Lowe is used. The SIFT descriptor contains very similar information to the previously described HOG features, as it consists of orientation histograms extracted from pixel regions around the point of interest. The main difference is that SIFT features are generally normalized with respect to orientation, while HOG features are not. The descriptors are extracted on a densely sampled grid.

These descriptors are subsequently used for a BoW classification scheme, which is a technique that originates from the field of natural language processing and was pioneered in the computer vision field by Csurka et al., Sivic et al., and Sudderth et al. In a BoW classification system, first a visual dictionary is created, consisting of descriptors of small patches from the training images. Typically, these patches are clustered using \( K \)-means clustering to create a visual dictionary. The descriptors from the dense grid are compared to the descriptors in the visual dictionary, and a histogram is made to describe the image patch. One important problem that occurs when applying this approach on highly unbalanced datasets, such as those available for traffic sign classification, is that classes with many samples become overrepresented in the visual dictionary. Therefore, we propose a modification to this approach, which aims at separate visual dictionaries of a fixed size for each positive class, and concatenates these dictionaries into one large visual dictionary. Additionally, one (larger) dictionary is created from nontraffic-sign images. We have shown that this approach leads to a significant performance enhancement for our application.

In addition to the BoW features, the SIFT descriptors themselves are also appended to the feature vector, as this was found to improve performance even further. Finally, a one-versus-all linear SVM is used as a classifier for the traffic signs. The above traffic sign detection and recognition approach are instrumental in our first surveillance scenario for the monitoring of parking lot scenes where both static objects such as traffic signs and moving objects such as cars with their license plates play an important role.
2.2 PTZ Human Tracking

This scenario involves the detection and tracking of objects in an unconstrained outdoor setting. Whether certain objects need to be tracked or ignored is based on the context, and based on cues from the traffic sign and license plate detection as discussed above and in more detail in Sec. 2.1. This section will specifically focus on the tracking of objects using commercially available PTZ cameras to support the scenario analysis.

There are several studies regarding automatic PTZ control, in which the authors construct a background image mosaic and utilize background subtraction to detect objects. For our unconstrained outdoor application, a static background mosaic is not optimal, as the video will include moving background such as trees and vehicles passing by. Comaniciu and Ramesh have proposed a system for face tracking that provides continuous tracking without the need of a background model. However, the proposed system is tuned for faces only, uses color information, and only works when the person is directly facing the camera. Xiang has worked on a system based on a mean shift tracker using color histograms in combination with a Kalman filter and a proportional-integral-derivative (PID) controller that can track objects in real time. These two proposals cannot be expanded to our application due to the lack of specific color information.

We therefore aim at combining a high quality, nonreal-time robust object detector with a feature point tracker with a low computational complexity. This enables the system to detect and track humans of a large variety in an outdoor setting at large distances. Furthermore, we improve the tracking system by a special motion estimation model for the feature points that also incorporates zooming of the camera.

2.2.1 PTZ system overview and technical challenges

Figure 4 gives a graphical representation of the human tracking system. A professional outdoor PTZ camera is connected to a video encoder that communicates with the LAN/WAN network. The used PTZ camera provides 4CIF analog video (equivalent with 704 × 576 pixels) and is controlled by an RS422 interface. The tracking algorithm processes the video from the remote camera, and sends back commands to control the pan, tilt, and zoom speeds in order to track a particular person. The annotated video and the estimated position are supplied to the context-based decision system.

There are several bottlenecks in this system, which make smooth and reliable tracking difficult. Time delays are introduced at various places in the system, such as the transmission of video data and the control delay of the PTZ camera. Furthermore, PTZ cameras are originally not designed to be controlled by an automated algorithm, and for the utilized camera, it takes 20 ms to send and acknowledge a command over an RS422 connection. These commands are also blocking; e.g., when requesting the position of the PTZ, no other commands are executed for 160 ms, which forms a significant problem for real-time tracking. Last, the PTZ camera utilizes a fixed zooming speed, and cannot be controlled proportionally like the pan and tilt operations. This causes many problems as the aggressive zooming may lead to the disappearance of objects from the view during the time delays present in the control loop.

2.2.2 Robust human tracking using active cameras

The complete processing chain of the proposed human tracking algorithm is shown in Fig. 5(a), with the gray blocks denoting the main contributions of this article, and Fig. 5(b) showing more detailed functions of the PTZ control box. Tracking is performed by using a combination of a robust but complex object detector, with a fast and simple feature point tracker. For robust tracking, we have found that it is crucial to validate the feature points’ correctness and periodically refresh them. Using the stable feature points, we estimate the translation and zoom of the tracked object, and calculate the PTZ control signals. Each of these processing steps are discussed in detail in the following subsections.

2.2.3 Combined object detection and tracking

As mentioned previously, we propose to use a combination of a robust but complex object detector, and a fast and simple feature point tracker. The object detector is used to initialize the feature point tracker, and to occasionally perform a re-detection in order to re-initialize the set of feature points. We have chosen to utilize the object detection proposed by Wijnhovenet al. based on multiscale HOG, but trained it to track humans from various viewpoints.

The feature point tracker is based on a hierarchical Kanade–Lucas–Tomasi (KLT) feature point tracker, where the small features of 3 × 3 pixels can be found significantly faster than performing a complete object detection. Once an object has been detected, the feature point tracker is initialized, using a fixed number of feature points automatically chosen from within the object region. Points are chosen that lie within the object, based on the object mask, to reduce the probability of points moving along with the edge of the object, or avoid locking to the background.

2.2.4 Feature point validation and estimation of translation and zoom

During tracking, feature points can disconnect from the object when the object passes over various types of background or when the viewpoint changes significantly. It is difficult to determine the outliers when using an active camera, because feature points will also significantly move around the image and resize drastically due to camera movement and especially zooming. This is aggravated by the fact that the utilized PTZ cameras can only zoom at a fixed speed. As a first step, all feature points are validated, and
all feature points that are inherently unstable or have a large tracking error, are rejected. By calculating the median of the position and the displacement of each of the feature points, we acquire an initial estimation of the location and speed of the group of feature points. Then, all points with a distance of more than three times the standard deviation from the group center are rejected, and the remaining feature points are considered stable.

Translational movement is calculated from the average displacement of the stable points, and zoom is estimated by the change in standard deviation of the distances from the group center. These estimates are used to determine a new bounding box, which serves as the input for the PTZ control algorithm of Fig. 5(b). To avoid drift, the estimated motion and zoom values are calculated with subpixel precision and rounding remainders are accumulated internally. In order to adapt to changes in the size and shape of the tracked object, the estimated bounding box is updated recursively, using the actual positions of the stable feature points.

The feature point tracker is re-initialized periodically, by performing a computationally expensive object detection cycle using the HOG detector. This interval is dynamic, and reduces in latency when fewer feature points are being tracked, thereby balancing complexity and the quality of the tracked feature points. Found objects are given a confidence score, based on the HOG detection value, corrected with the distance to the currently tracked object. This prevents the tracker jumping from one object to another, even when they are passing each other closely.

2.2.5 PTZ camera motion and zoom control

Figure 5(b) shows the PTZ control system. Translational motion (pan and tilt) is controlled by two PID controllers with the horizontal and vertical error as input. Because different control speeds are required at different zoom levels, the PID output is compensated based on the current zoom level, using the known field-of-view (FOV) at different zoom levels. Alternatively, FOV-dependent control in the PTZ camera can be enabled if supported. The latter option allows for more accurate tracking, as the current zoom level is always known by the camera, and pan–tilt control can use the full resolution of speed control, reducing quantization effects. Pan and tilt control behavior varies with different PTZ cameras, and a look-up table is utilized to linearize pan and tilt rotational speeds to deg/s.

Zoom is controlled based on the size difference of the measured and desired boxes. Only when the object is at the center of the image is zoom-in initiated, to avoid shifting the object outside the FOV. Zoom-out is always allowed and is also initiated if the object approaches the image boundaries and assists the maximum pan and tilt speeds, so that fast moving objects are not lost from the view. In the utilized camera, zoom occurs at a fixed speed, and even at lowest settings, it is often too fast and causes object tracking failure. We have implemented a pulse width modulation system to limit the zoom speed when necessary by only enabling the zooming control during a part of the duty cycle. This effectively implements a simple method of proportional control with nonlinear means.

3 Scenario 2: Group Behavior Analysis Based on Scene Context

3.1 System Overview

The second scenario in video surveillance involves the monitoring of large groups and crowds of people and associated behavioral analysis. Many algorithms exist to detect and track single persons, but analyzing group behavior is a relatively unexplored area of research. Several factors can affect the interpretation of the actions of groups of people, such as contextual region-labeling results (e.g., what parts of the image are sky, vegetation, buildings, etc.) and locations of traffic signs. We propose a proof-of-concept system for automatically analyzing group behavior using these contextual clues to interpret and classify the actions of the group of people. Evidently, this list of contextual clues is not exhaustive but for this proof-of-concept system we limit ourself to these aspects. Let us describe some examples of situations in which traffic signs and image regions can influence the interpretation of group behavior. A group of people that remains stationary for a long period of time can indicate loitering, fighting, injury, or other potentially suspicious behavior, while a group of nonmoving people close to a bus stop indicates a perfectly normal situation. When persons cross a road, the location of a zebra crossing indicates a regular road crossing, while a dangerous road crossing may require...
attention from an operator. Signs can also indicate areas that are forbidden for pedestrians, or many other situations. Semantic region labeling can indicate roads, zebra crossings, and other important areas. Additionally, region labeling helps to filter and decrease false group or traffic sign detections (e.g., vegetation region (trees) should not contain any group detections). Our proof-of-concept system contains three components: (1) a group detection algorithm, to track the movement and any splits/merges of groups of people, (2) a traffic sign detector, and (3) a region labeling algorithm. Finally, an event-level decision engine interprets the output of these algorithms and decides when an alarm needs to be initiated (see Fig. 6 for an overview of the system).

### 3.2 Human Group Analysis

Locating humans and recognizing their actions have been important corner stones of surveillance. Experience learns that mature single-human algorithms do not scale well to densely crowded scenes, mostly due to dynamically arising occlusions and heterogeneity of the tracked persons. Consequently, new approaches are needed for applications in crowded settings. Our crowd analysis rests on a motion based, rather than object detection based, framework that localizes groups. Our approach also has the ability to detect pixel/motion-based group events, such as merges and splits of groups of people.

Our group detector extracts and monitors the transformation of motion blobs in video sequences. Motion is sampled both spatially, where every regular cell is overlaying the spatial grid of the frame, and temporally, over a frame interval in the video sequence. In order to classify group events, the consistency of behavior is assessed by finding spatial blob matches between frames and checking whether the matching record is continuous over a substantial number of frame intervals. Where heterogeneity of motion appears and persists within the blob, the motion blob is later given a higher semantical meaning. Heterogeneity of motion occurs when a part of the blob’s optical field deviates from (or is incoherent to) the motion of the rest of the blob. Such content is referred to as incoherences or incoherent blobs. The diagram in Fig. 7 gives an overview of the entire processing chain of our system from motion blob formation to event classification.

#### 3.2.1 Motion extraction and characterization

For motion extraction, the video frame is spatially overlaid by a grid of square cells, \(10 \times 10\) pixels each. The center of each grid cell (referred to as a grid point) becomes the sample point for motion estimation. Motion estimation is performed by the OpenCV implementation of the Lucas–Kanade optical flow algorithm. Optical flow is estimated between two frames temporally separated by an interval of 2 to 5 frames. While every grid point results in a displacement vector, most of those lie outside the range of relevant magnitudes. Grid cells exhibiting irrelevant displacements (<2 or >30 pixels) are set to 0. These displacements are (nearly) stationary points, mistakes of the optical flow algorithm, or noise. The displacements are smoothened by a wide (11 × 11 grid cells) Gaussian kernel, see Fig. 8(b). Smoothing of the optical field to extract motion trends is a common technique from the research on fluid dynamics modeling. Based on spatial locality, the remaining grid cells are formed into motion clusters, also referred to as
parent (motion) blobs. This is done by extracting the outer contours of the active grid cells, see Fig. 8(c).

3.2.2 Group behavior classification

A directional analysis of the grid cells in each parent blob is performed in the separate coherence analysis stage. The majority of the grid cells will have coherent displacement vectors, since they all follow the common drift direction of the moving body they belong to. Since, our moving body is a group of individuals and hence nonrigid, thresholds of motion coherence here cannot be as strict as for a solid object. If the parent blob undergoes a transformation, such as splitting or merging, groups of incoherently moving grid cells appear within the blob, referred to as incoherent blobs. A recursive algorithm is used to separate one or more incoherent blobs from the coherent content of the parent blob (the details of this recursive algorithm can be found in Ref. 15).

To remove stray, nonevent-related incoherences, a temporal consistency analysis is performed. These incoherences in the crowd setting are most often attributed to human limb movements and inner group dynamics. The true incoherences culminating in an event have an observable lifespan, i.e., they can be matched between a number of consecutive frame intervals. This temporal consistency of true incoherences has proven to be a defining characteristic on which crowd motion pattern analysis can be based. If a blob is found with a long enough lifespan (three or more consecutive matches in our application), the blob is passed on to the event classification stage to determine if a merge, split or slide-by event has occurred.

To perform this behavioral classification, two key classification features are used, the fraction of the incoherent’s border grid cells pushing out of their parent blob and the corresponding fraction pushing into their parent blob. The two fractions are computed at every frame interval, and the maximum value observed during the lifespan of the incoherent blob is used to make the classification. Our approach allows us to distinguish such group events as merging, splitting, and sliding-by.

To realize a higher degree of accuracy in the classification between merge and slide-by events, two additional features are introduced, a global drift-correlation feature and the incoherent blob’s submergence into the parent. A heuristic classifier distinguishes between merge, split, and slide-by events, based on these four features.

One remaining problem with the previously described approach is that stationary groups cannot be tracked, since the framework is purely motion-based. To solve this problem, texture-rich grid cells that were part of a motion blob are frozen in status if no more motion is detected. Additionally, a HOG (see Sec. 2.1) based on human detector is used to confirm these cells and inject more active cells into the motion blob, if necessary.

3.3 Region Labeling

Context-based region labeling in a video will contribute to accurate semantic understanding of the video, e.g., a better understanding of the group behavior in a scene. In this section, we present our fast region labeling approach as a supporting technology to provide context for the event in a scene. We propose a region labeling system based on the observation that each region is more likely to be found at a specific vertical position. A region corresponding to a specific semantic meaning normally covers a certain part of the color space and has a distinct texture. We aim at classifying six types of regions: sky, vegetation, road, water, construction, and zebra crossing. Our algorithm contains three stages, as depicted in Fig. 9, and discussed briefly below.

- Stage 1: Context/region segmentation. In our system, the basic idea is to combine the image segmentation with the region classification technique. Instead of labeling each region directly at the pixel level, we first divide the image into several regions with a uniform color (or texture), which sufficiently considers additional assumptions on the color continuity discussed later.

Fig. 9 Block diagram of our region labeling approach.
• Stage 2: Context/region analysis. In this step, global and local features of each segmented region are extracted.
• Stage 3: Context/object classification. This classification is divided into two aspects as follows.

• MultiSVM. For each class of regions, we use an off-line separately trained SVM.
• Statistical model process. As postprocessing, we compare the percentage of labeled pixels with a predetermined threshold $T$ for each region. We assign a label to a region for which the percentage of positively classified pixels is $\geq T$.

### 3.3.1 Context/region segmentation

We employ an efficient graph-based segmentation as preprocessing in our region labeling, to achieve two objectives: (a) distinguish each region from other objects while preserving the overall characterization of the region itself and (b) perform fast segmentation for real-time application in our region labeling, to achieve two objectives:

We employ an efficient graph-based segmentation as prepro-

### 3.3.2 Context/region analysis: feature extraction

Prior to training a reliable and robust SVM classifier, it can be sufficient to use only local features such as color and texture. However, when classes have similar characteristics (overlapping classes), complications arise. These complications can be solved by using spatial context (SC) as an additional feature. In our case, this context involves the vertical position of the regions in the image, e.g., the sky tends to be at the top of the image and the water at the bottom. Summarizing, we combine the locally calculated pixel-based features and the region-based features to achieve a more reliable region labeling approach. This section presents all features used for analyzing the images.

#### Pixel-based features

For estimating local region properties, we consider the following features.

- **Color** may be one of the most straight-forward features utilized by humans for visual recognition and discrimination. We have performed an experiment to determine the most appropriate color space. Three different color spaces are tested: RGB, HSV, and CIELUV (proposed by Benedek and Szirmi as the most efficient color space). See Table 2 in Sec. 4.2.2 for the results.
- **Texture** takes into account the local neighborhood variation and a better classification is achieved when texture information is included in the analysis. Gabor-transformed features are widely applied to the computer vision and image analysis. In addition to make time-frequency location more accurate, they also provide robustness against varying brightness and contrast of images. Based on these properties, we apply here a group of Gabor filters with three scales and six orientations. The two-dimensional Gabor filter, as described by Kamarainen et al. is specified by the following equation:

$$
\psi(x, y) = \frac{a_0}{\pi} e^{-\left(a^2 x^2 + \beta^2 y^2\right)} e^{2 \pi x f_0},
$$

$$
x' = x \cos \theta + y \sin \theta,
$$

$$
y' = -x \sin \theta + y \cos \theta,
$$

Here $f_0$ defines the central frequency (scale) of the sinusoidal plan wave, $\theta$ is the anti-clockwise rotation of the Gaussian and the plane wave, and $a$ and $\beta$ are the sharpness values of the major and minor axes of the elliptic Gaussian. To build our feature vector, we filter the gray level image with the Gabor filter bank. We use six orientations and three central frequencies (scales). The filter parameters are as follows:

$$
f_0 = \frac{\sqrt{2}}{2\sqrt{2}}, \quad a = \frac{\sqrt{2}}{\sqrt{\log 2}} \pi f_0, \quad \beta = \frac{\sqrt{2}}{\sqrt{\log 2}} \pi f_0 \tan \frac{\pi}{16};$$

$$
p = 3, 4, 6; \quad \theta = 0, 3\pi 4\pi 5\pi 7\pi 8 \frac{8}{8} \frac{8}{8} \frac{8}{8} \frac{8}{8} \frac{8}{8}$$

---

**Region-based features.** We propose two different region-based features based on the vertical position of each region. To compose the feature vector, we combine the above pixel-based features with one of the following two region-based features:

- **SC** helps to perform accurate region labeling. For each pixel $(i, j)$, we calculate its normalized vertical position $SC_{ij} = i/n$, where $i$ is the row number in the image and $j$ is the column number. Each region consists of $n$ rows. We call this method a gravity model.
- **Global region statistics**, such as the commonly known variance and mean, are computed as follows. Let us assume that we have $M$ regions of a particular type (e.g., sky) in the training set of images. For each region, we calculate mean values $\mu_k (k = 1, \ldots, M)$ of the vertical positions of its pixels. We also calculate the standard deviation $\sigma_k$ of the vertical pixel positions for each region. Then, we take minimum and maximum values for all means and standard deviations for this region type: $\mu_{\text{min}} = \min(\mu_1, \ldots, \mu_M)$, $\mu_{\text{max}} = \max(\mu_1, \ldots, \mu_M)$, $\sigma_{\text{min}} = \min(\sigma_1, \ldots, \sigma_M)$, and $\sigma_{\text{max}} = \max(\sigma_1, \ldots, \sigma_M)$. In this way, we obtain intervals for mean and variance for the region. We assume that the mean value of vertical pixel positions lies in the interval $(\mu_{\text{min}}, \mu_{\text{max}})$ and standard deviation in $(\sigma_{\text{min}}, \sigma_{\text{max}})$. We find these intervals for each of the six region types described in this article. We call this method a statistics-based model.

### 3.3.3 Context/object classification approaches

After analyzing and segmenting the image, we proceed to obtain the labeling results. The labeling is performed by a classification system based on an off-line trained SVM. Here, we present two approaches for region classification.
Fast classification using the Gravity model. In the first approach, color, texture, and SC are used to train the SVM for each class of regions separately, i.e., an individual SVM is trained for each type of region. Our fast unitary-category classification is described in the inner part of Algorithm 1.

For multiclass labeling, we assign to each segment one of the six labels: sky, vegetation, construction, road, water, and zebra crossing. To this end, we classify each segment by six SVMs using our unitary classification Algorithm 1 and obtain six classification values, indicating the percentages of positive pixels for each SVM. Finally, a segmented region is assigned to a particular class if its percentage is higher than the corresponding empirically determined threshold. Algorithm 1 illustrates our multiclassification algorithm with the unitary algorithm embedded into it. The empirical threshold \( T_e \) for each region has a value typically in the interval \((0.1,0.8)\).

Classification using the Statistics-based model. We propose a second approach to classify images based on the position of each region using statistical information from the statistics model defined in the previous subsection. First, we apply the SVM with color and texture while using the same steps as in the first approach. Then, for assigning a label to a region, we check that both the following conditions are satisfied: (1) the percentage of positively classified pixels exceeds the threshold \( T_e \); (2) the mean and variance of the vertical positions of the pixels lie in the intervals defined in the previous subsection.

**Algorithm 1** Our fast multiclassification with embedded unitary classification in pseudo-language.

```plaintext
for 6 classes do
    Define the next class type;
    Set corresponding threshold \( T_e \);
    for a segmented region do
        Randomly choose 100 samples in this region and use the SVM classifier to label the samples;
        Calculate the percentage of positive samples in this region;
        if the percentage of positive samples exceeds the possibility threshold \( p \) then
            Set this region is positive;
        end
        Compare results to class threshold \( T_e \);
        Label the current region;
    end
end
```

4 Experimental Results: Use Cases

4.1 Scenario 1: Parking Lot Surveillance

For the parking lot surveillance scenario, we first discuss the performance of the individual analysis components, such as traffic sign detection and classification and the PTZ tracker. Second, we briefly discuss the detection performance of the complete scenario, where the car and person are tracked after detection.

Figure 10 portrays a functional diagram of our parking lot scenario in Scenario 1. At the left, a number of detectors are active simultaneously to find the correct context information in parallel with the analysis involved with the foreground objects of interest. Contextual information in this scenario is provided by the traffic sign detection and classification and the license plate detector that defines the car identity. The person detector and tracker are providing the primary information about the object of interest and its behavior. In this scenario, further intelligent tracking is realized by the included PTZ control which enables a larger visibility and behavior analysis of the person of interest. At the right side of the figure, the event detection and classification unit combines all the contextual and foreground person information in order to create a detailed event analysis and classification.

4.1.1 Traffic sign detection and classification

Dataset. We have found panoramic images to be a challenging dataset to work with, as was noted by Frome et al. Our findings are that the performance of state-of-the-art object detection algorithms, such as unmodified HOG, is insufficient for this seemingly simple problem. Even though traffic signs themselves seem objects that are easy to detect (they were designed for that purpose), the large variety of recording conditions, sign appearances, and large amounts of background clutter to make the problem challenging. More specifically, the panoramic images are taken from a moving vehicle. This results in motion blur, significantly varying lighting conditions, various objects that can partially occlude the traffic signs, and the signs themselves become damaged by collisions, dirt, and fade with age. In addition to these problems, there is a wide variety of background objects with similar characteristics to traffic signs, such as colored advertisements.

The dataset used in this article consists of 300,000 panoramic photographs taken at 5-m intervals in a rural region. The set contains approximately 13,000 traffic signs of various classes, and each traffic sign is visible within several different panoramas. The position of each panorama is known with an accuracy of approximately 10 cm. In this article, we only consider the red triangular, yield (give way), and blue circular traffic signs. Performance of most other traffic signs has been found to be similar. There are 20 different red

![Fig. 10 Schematic of the parking lot surveillance system.](image_url)
triangular signs that appear sufficiently often in our dataset, so the remaining type is not considered in our classification experiments.

**Experiments.** The results of applying the detection algorithm to our dataset are visualized in Fig. 11. The performance of the blue circular traffic sign detector is somewhat lower than the other sign detectors, due to the occurrence of blue circular shapes in the sky, leading to false positives.

The results of applying our classification algorithm on crops of three classes of traffic signs are shown in Table 1. These crops are taken from a set of approximately 500,000 panoramas. It can be seen that the standard method of codebook creation does not work well on the unbalanced data in our dataset. Concatenating individual codebooks for each class greatly increases the performance. It can be seen in the table that by classifying ambiguous samples as unreliable, the most difficult samples can be manually checked. This class of unreliable signs is used for after checking, whereas the reliable sign class is immediately accepted to save efforts. This reliable class is fully automatically classified and has a high accuracy.

![Traffic sign detection performance for various traffic signs](image)

**Fig. 11** The results of applying the traffic sign detection algorithm on a dataset of 300,000 panoramic street-level photographs. Results are shown for red triangular, blue circular, and yield (give way) signs.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification accuracy</td>
<td>67.2%</td>
<td>95.0%</td>
</tr>
<tr>
<td>TP rate</td>
<td>0.944</td>
<td>0.995</td>
</tr>
<tr>
<td># unreliable samples</td>
<td>2103</td>
<td>327</td>
</tr>
<tr>
<td>Unreliable rate</td>
<td>0.288</td>
<td>0.045</td>
</tr>
<tr>
<td># false classifications</td>
<td>291</td>
<td>36</td>
</tr>
<tr>
<td>False. class. rate</td>
<td>0.040</td>
<td>0.005</td>
</tr>
</tbody>
</table>

**Table 1** Traffic sign classification performance on 7289 samples for two experiments: (1) unmodified BoW and (2) concatenated per-class codebooks.

4.1.2 PTZ human tracking

To fully test the human tracking subsystem, several indoor and outdoor tests have been executed to push the system to its performance limits. Figure 12 shows periodical screenshots of such a test, where the detector has been trained to detect faces. All aspects are tested in this sequence, as it includes various lighting conditions, a cluttered background, and a moving subject at various distances. At close range, maximum rotational speeds of the PTZ are required to follow the person when squatting down (at time 00:00:09, not clearly visible in these screenshots). At larger distances, the motion of the camera is smaller and more delicate, because zoom is utilized. The fact that the camera is zooming can be seen from the difference in the size of the painting on the wall at times 00:00:05 (left in the background) and 00:00:34 (behind the subject). At the last part of the sequence, at 00:00:37, it can even be seen that the subject is running and motion blur occurs in the background.

4.1.3 Illegal parking detection

To test our proof-of-concept of an illegal parking detection system, we have created several videos with a regular parking place and a parking place for the disabled. In the video, an actor parks at the regular parking place as well as at the parking place for the disabled. To correctly match the detected cars to detected signs, a limited amount of 3-D information is needed, such as camera height and angle, and the angle of the parking lots to the camera which is entered during the setup of the system. The traffic signs are mostly detected and classified correctly (some misclassifications occur due to low resolution and camera interlacing artefacts). The license plate of the car is detected and as soon as a person exits the vehicle, the PTZ camera starts tracking the person until he leaves the FOV of the camera. Some screenshots of the processed video can be seen in Fig. 13. Additionally, the video is included as a multimedia appendix to the article.

4.2 Scenario 2: Outdoor Group Behavior Analysis

This scenario involves the behavioral classification of a group using the context information of the scene and a careful motion analysis of the group members. Again, we start with the performance of the primary individual components.

Figure 14 exhibits the functional diagram of Scenario 2 featuring outdoor group behavior analysis. At the bottom of the figure, the group behavior analysis is built-up from the detailed motion analysis of the group members and their mutual proximity to each other. The classification of that behavior with respect to safety is helped by the contextual information created in the upper three branches of the diagram. This contextual information involves vegetation detection to limit the group’s motion trajectories, and traffic control signals in the form of traffic signs and zebra crossings to identify regions of safe road crossing. Besides road crossing, the traffic sign detector is also used for localizing bus stops so that group behavior around such bus stops can be monitored as well. As in Scenario 1, we employ the combination of context information and foreground object information to provide more detail and a more robust scenario analysis.
Fig. 12 Periodic screenshots of PTZ tracking system test.

Fig. 13 Screenshots of a video demonstrating the proof-of-concept illegal parking detection system (Video 1, MPEG, 24.7 MB) [URL: http://dx.doi.org/10.1117/1.JEI.22.4.041117.1.]
The quality of the group tracking can be measured by the tightness of the computed contour around the target compared to manually annotated frames. We quantify the quality-of-fit of the motion blob’s contour by two measures: the percentage of the blob that does not overlap with the ground truth (excess measure) and the percentage of the ground-truth blob that is missing (lack measure). Our tests on two representative frames from our dataset have revealed excess measures of 37% and 31% and lack measures of 3% and 0.8%. These values are representative for the system, as it has a tendency to include shadows within the blob, while humans are rarely missed.

4.2.2 Region labeling

We start with our labeling experiments. We have constructed a broad dataset that consists of images from multiple Internet datasets and a personal archive. It contains our earlier defined six classes plus and one class “unknown,” using 234 images, of which 101 images are for training and 113 images are for testing. For the segmentation, there are three parameters to be defined: the standard deviation $\sigma$ of the Gaussian filter, threshold $\kappa$, and the minimum region size. When the minimum region size is small, there will be more regions after segmentation, and it will bring an extra burden for the following classification process. For real-time applications, $\sigma = 1.4$ and $\kappa = 300$ are a good choice for complex images. The means and variances in the statistics-based model are calculated based on 20 images from the training set. Let us now present the experimental results on region labeling. Example labeling results for two different images are shown in this section. Figure 16(a) shows the original image of our dataset, Fig. 16(b) visualizes the ground truth for region labeling of that image, which is achieved by manually segmenting the original image. Figure 16(c) shows the result of the gravity model. Figure 17 illustrates the result of our gravity model to detect a zebra crossing for our surveillance application. Figure 18 illustrates a challenging image along with the results of three different region labeling approaches to highlight the differences between the labeling algorithms. This image is challenging for labeling and contains several regions of interest and simultaneously, the color information is quite poor.
with only small color differences between neighboring regions. It can be observed that the gravity model achieves better results, while considerably improving the false detection/rejection rates.

To evaluate the performance of the region labeling algorithm, we use the coverability rate (CR), which measures how much of the true region is detected by the algorithm. This rate is computed by \( CR(O, GT) = \frac{|O \cap GT|}{|GT|} \), where we use the manually annotated ground-truth area (GT) and O is the automatically detected area.\(^{50}\) In order to analyze the performance of our region labeling algorithm, we have compared our results with the method of Bao et al.\(^ {40}\) Table 2 shows the results of applying the gravity model and statistics model approaches compared to Bao’s algorithm for our dataset. We can observe that the gravity model results in a higher CR than the other approaches. Furthermore, the results show that the best performance is obtained in the HSV color space. In the literature, it has been shown that the unitary-category classification of Bao et al.\(^ {40}\) demonstrates a much better performance compared to another state-of-the-art approach.\(^ {51}\) We have extended the unitary-category classification of Ref. 40 into multicategory classification and applied contextual information as a special feature. With our contribution, we have shown that the gravity model for region labeling substantially outperforms the results achieved by Bao et al.\(^ {40}\)

4.2.3 Abnormal group behavior

The proof-of-concept group behavior analysis system consists of the human group analysis system for detecting the groups and motion-level group events. The context information is supplied by the traffic sign detection and classification system and the region labeling algorithm. The outputs of
these algorithms are analyzed by a decision engine that can analyze the group behavior at a scenario level. The system is tested with the recorded video in an outdoor setting, containing several acted scenarios. In several scenes, groups of people cross a road (the road in the video is not busy but could be) either at the zebra crossing or at other dangerous places. The zebra crossing is found by means of the traffic signs that are detected at each side as well as by the region labeling algorithm. In another scene, a fight breaks out and a group of people gathers around it. A contrasting scene shows a group of people gathering at a bus stop, where the bus-stop sign is recognized by the traffic sign detection and classification system. A different set of scenes shows groups of people that suddenly disperse in panic, and this situation is detected by analyzing the sudden high-speed dispersion of the group. Several screen shots of recorded scenes are shown in Fig. 19, and a video containing several scenes is included as multimedia appendix to the article.

4.3 Performance Evaluation on Public Datasets
In order to enable a fair comparison of our work to already published research, we have evaluated the performance of our system using publicly available datasets. Since these datasets are not fully suitable to test our system, we can only report the performance of a subset of the complete functionality of our system. We focus on motion-based group event detection, tracking groups and detecting splits, merges and slide-bys (occluding groups passing each other without merging) events. For this evaluation, we have used the publicly available PETS 2004 and PETS 2009 datasets. Because these datasets only contain a limited amount of relevant group events, we have opted to add some of our own test sequences to the dataset to increase the statistical significance of the comparison experiment.

We report two precision-recall scores, one for the detection of the basic crowd behavior events and another for the detection combined with a correct classification of the event type (split, merge, or slide-by). We consider repeated detections of the same event to be false positives. The dataset contains 89 events with 33 merges, 31 splits, and 25 slide-by events, from 33 video sequences. The system exhibits a detection recall of 83% with a corresponding precision of 37%. The scores for combined detection and subsequent correct classification are 71% and 32% for recall and precision, respectively. The low value for precision can be largely explained by the repeated detections, shadows, motion of limbs within the group, and stationary objects that appear in the foreground (between group and camera).

In the literature, there are a few systems that also report performance figures on PETS datasets. Garate et al.52 have tested their crowd event recognition system on the PETS 2009 dataset, and have reported a recall of 79% for splitting and 60% for merging events. Chan et al.53 report error rates of 23% and 32% for classifying splitting and merging events, respectively. These scores are all relatively close to our results indicating that our performance is comparable to the state-of-the-art systems discussed in the recent literature.

5 Discussion
The common element in this building of Scenarios 1 and 2 are the simultaneous analysis of contextual information and...
foreground objects of interest. This concept has been visualized in Fig. 20, where in the general case, a number of context detectors and foreground object detectors are depicted. Conceptually, the context detectors provide information about the near surroundings of the objects of interest and therefore they may define safety regions to the foreground objects. At the bottom of the diagram, a group detector and possibly other moving objects are analyzed with respect to the motion and behavior and for safety, proximity criteria between persons and other moving objects are likely to be required to define safety in a more clear detail. It is evident that such criteria and region definitions fuel the correctness required to define safety in a more clear detail. It is evident between persons and other moving objects are likely to be the motion and behavior and for safety, proximity criteria objects. At the bottom of the diagram, a group detector and therefore they may define safety regions to the foreground objects of interest. This concept has been visualized in Fig. 20, where in the general case, a number of context detectors and foreground object detectors are depicted. Conceptually, the context detectors provide information about the near surroundings of the objects of interest and therefore they may define safety regions to the foreground objects. At the bottom of the diagram, a group detector and possibly other moving objects are analyzed with respect to the motion and behavior and for safety, proximity criteria between persons and other moving objects are likely to be required to define safety in a more clear detail. It is evident that such criteria and region definitions fuel the correctness of the event detection and classification function which is depicted at the right side of the diagram. In an alternative arrangement, the safety region and proximity criteria blocks are integrated in the event detection and classification function.

There is a clear issue emerging with such detailed scenario event and analysis systems as described in this article. This issue is involved with the determination of the performance of such context-supported scenario analysis. The difficulty in the performance analysis appears to have both practical and theoretical aspects.

A practical consideration for the broad testing and evaluation is the lack of availability of test material that allows such context supported event analysis. Ideally, many sequences of recorded video from real surveillance cameras are required for a proper evaluation, including many that contain the events of interest. At present, this is difficult if not impossible to obtain, due to the rareness of the events of interest and due to the trouble with regards to privacy of recording and storing large amounts of security camera footage. In the case of context-based event analysis systems, it is inevitable that the system consists of several components.

This leads the discussion to the second, more theoretical issue regarding the performance analysis of the scenario analysis systems. As previously mentioned, several components are employed to detect different contextual clues, and one decision engine will interpret the detected contextual clues and decide if an alarm needs to be raised. The context and foreground object analysis subsystems are typically optimized for their primary purpose, and such subsystems can be well analyzed with respect to the performance because more video data is available for those subsystems analysis problems. One interesting question that arises from this is to what degree the performance of the overall system can be predicted based on the well-known performance of the subsystems. Intuitively, it seems natural to assume that if the subsystems can each be shown to work well (as also shown in this article for both scenarios), the system as a whole should also execute with adequate performance. However, it is interesting to predict this performance when the performance figures are well defined/measured. This question is a new and interesting topic for further research. The concept of predicting main system performance based on the performance of contributing subsystem components is more common in other branches of research. For example, in the field of predicting software run-time performance, Becker et al.54 and Chen et al.55 describe methodologies to predict the run-time performance of component assemblies based on performance measurements of components. In our case of video system analysis, such literature does not exist. Interesting issues for the overall performance are knowledge about the correlation versus independent behavior of failures (false positive detections or missed detections) in individual components and their influence on the overall performance. An example of a case where failures may be correlated are unusual weather conditions such as snow or fog. Uncorrelated failures can occur due to many causes such as (partial) occlusions. Additionally, the system may benefit from the context detectors to gain a higher robustness against missed detections by detecting context in different ways. For example, in our zebra crossing scenario, the zebra can be detected by the region labeling algorithm as well as by the traffic sign recognition subsystem, which can find the signs located at each side of the zebra. This redundancy means that a missed detection in one of these subsystems does not necessarily lead to a reduction in overall system performance, as long as the errors do not occur in a correlated fashion (this is much more unlikely in any case). In order to say something about the prediction of the performance of the total system we assume a simple model that is typical for a composed system. For the case that a few context detectors operate independently and in parallel to the foreground analysis, it can be shown with this model that the performance indeed increases in terms of robustness. For example, let us assume that the failure probability of the $i$th detector is $p_i$ and that $n$ components operate in parallel.

The correct operation of the total system is then equal to $1 - \Pi_{i=1}^{n} p_i$. In our case, the individual success probabilities for each subsystem detector are in the order of 90% to 98%, so that the probability of failure is indeed a product of a number of small probabilities which is then indeed a small probability and the robustness of the total system is increased. In object detection, there is a difference between failure probability and the occurrence of false positives and negatives. This relation should be established first. Second, the system with parallel detectors increases the robustness when those detectors aim at finding the same type of information and still work independently. This is not always the case. In our Scenario 2, we indeed have two parallel detectors for finding a zebra crossing, i.e., by either the traffic sign or the painted road markings. However, in that scenario, the performance of the group behavior detector is clearly more critical in terms of performance than the context detectors so that we consider that detector to be decisive for the overall performance of the system. The above discussion clearly points out that further exploration for this kind of composed systems is required.

6 Conclusions and Future Work

A key objective of the ViCoMo project is to evaluate the use of contextual information to improve scene understanding along with the use of conventional object detection and tracking systems. This article has reported on the findings of this project and the development and testing of important...
components. We have reported on two complex scenarios where context information is essential for decision making about the semantic understanding in surveillance video. Finally, a proof-of-concept system has validated these scenarios.

In the parking lot surveillance scenario, an automatically controlled PTZ camera surveys a parking lot with a regular parking space and a parking space for the disabled. Persons are detected and tracked by the automatic PTZ tracking system. Context is provided by the traffic sign detector to detect the signs indicating both parking places. In this scenario, the traffic sign detector is also used to detect the license plate on the car. When a car approaches, the license plate is detected and when a person leaves the vehicle, the person is automatically tracked by the PTZ camera until leaving the scene. If the person illegally parks at the parking place for the disabled, an operator alert is raised to monitor the situation.

In the group behavior scenario, groups of people performed various actions that are better understood by the system when using contextual clues. The groups of people and motion-level group events are detected with a human group analysis system. The contextual information is provided by a traffic sign detection and classification system, and a region labeling algorithm. Using the contextual clues, the system is able to distinguish between people crossing a street at a zebra crossing and people crossing the street at a potentially dangerous location. In other scenes, the system is able to distinguish a group gathering with a potentially suspicious behavior from a group of people regularly gathering at a bus stop. Finally, the system is able to detect a panic situation, in which the group rapidly disperses because of an approaching car. We have created and reported on two proof-of-concept systems to illustrate examples of how contextual information can be used to improve scene understanding in common surveillance applications. The subsystems were individually tested extensively using various datasets.

Although the individual components of our complex scenarios are broadly tested for multiple datasets, we are aware that the number of experiments at the scenario level is limited. This is mainly due to the lack of test material and broad availability of that material. Our scenarios are complex so that our dataset had to be partially captured by the project members. Given this constraint, we have found that our system evaluation proves to be well working for the dataset at hand. This positive result is due to the considerable testing of the individual components and the relative simplicity of the decision engines for the scenario content. Another conclusion is that context information is not very complicated to add a surveillance video, while it adds significant value to the automated surveillance analysis. We are particularly pleased that we can report on automated scenario analysis that was not earlier possible. This brings surveillance analysis to a higher level, while improving the reliability of the video analysis at the scenario level.

In future work, the system can be extended to recognize many other types of contextual information such as cars, doors, escalators, elevators, and traffic lights. Additionally, a learning-based reasoning engine may be developed that can recognize unusual events based on statistics of previously observed events. Moreover, the development of performance metrics for such complex scenarios are an essential part of the future research.

Acknowledgments
This work has been performed in the framework of the ITEA2 ViCoMo Project, cofunded by grants from The Netherlands, Finland, France, Spain, and Turkey.

References

Ivo M. Creusen obtained his BSc and MSc in electrical engineering at Eindhoven University of Technology (TU/e) in 2005 and 2008, respectively. Currently, he is a researcher at CycloMedia Technology B.V. where he works on object recognition from street-level panoramic images. His main research interests are the application of computer vision techniques in real-world systems and work processes, and robust and efficient large-scale implementations.

Solmaz Javanbakhshi received her MSc degree in computer engineering from Shahid Beheshti University, Tehran, Iran, in 2010. She did her master’s thesis on segmentation and quantifying of “MR images” at the Biomedical Engineering Department in Eindhoven University of Technology (TU/e), The Netherlands. Since 2010, Solmaz has been a PhD student in Video Coding and Architecture Group at the Department of Electrical Engineering in TU/e. Currently, she is working on video surveillance applications including video context analyzing and region labeling for semantic understanding of the video.

Marjolijn J. H. Loomans received his MSc in electrical engineering from Eindhoven University of Technology (TU/e) and his master of technological design in embedded systems from the National University of Singapore (NUS) in 2005 and 2004, respectively. Currently, he is pursuing his PhD at TU/e, where he is researching scalable video coding for embedded surveillance systems, and in 2011, he received the first place ICCE best paper award. Alongside his PhD, he participated in the ViCoMo and WaterVisie projects, where he investigated real-time tracking of moving objects using active cameras. His research interests include video surveillance, scalable video coding, real-time systems, and embedded-system implementations.

Lykele B. Hazelhoff received his BSc and MSc degrees in electrical engineering from Eindhoven University of Technology (TU/e) in 2006 and 2008, respectively. Currently, he is working at the research department of CycloMedia Technology B.V., where he is working toward a system for recognizing traffic signs from street-level panoramic images. His main interests are the application of computer vision techniques in supervised systems, their robustness against real-world situations and their corresponding efficiency improvements.

Nadejda Roubtsova received both her BSc and MSc degrees in electrical engineering from Eindhoven University of Technology (TU/e) in 2009 and 2011, respectively. Nadejda was a temporary researcher involved in the ViCoMo project at the Video Coding and Architectures (VCA) Group. Currently, he is pursuing a PhD degree in the field of computer vision at the University of Surrey.
Svitlana Zinger received the MSc degree in computer science in 2000 from the Radiophysics Faculty of the Dnepropetrovsk State University, Ukraine. She received the PhD degree in 2004 from the Ecole Nationale Superieure des Telecommunications, France. Her PhD thesis was on interpolation and resampling of three-dimensional (3-D) data. In 2005, she was a postdoctoral fellow in the Multimedia and Multilingual Knowledge Engineering Laboratory of the French Atomic Agency, France, where she worked on creation of a large-scale image ontology for content based image retrieval. In 2006 to 2008, she was a postdoctoral researcher at the Center for Language and Cognition Groningen and an associated researcher at the Artificial Intelligence Department in the University of Groningen, The Netherlands, working on information retrieval from handwritten documents. Currently, she is an assistant professor at the Video Coding and Architectures Research Group in the Eindhoven University of Technology.

Peter H. N. de With graduated in electrical engineering (MSc., ir.) from Eindhoven University of Technology and received his PhD degree from University of Technology Delft, The Netherlands. From 1984 to 1997, he worked for Philips Research Eindhoven, where he worked on video compression and chaired a cluster for programmable TV architectures as senior TV Systems Architect. From 1997 to 2000, he was full professor at the University of Mannheim, Germany, Computer Engineering, and chair of Digital Circuitry and Simulation. From 2000 to 2007, he was with LogicaCMG in Eindhoven as a principal consultant Technical SW and distinguished business consultant. He was also part-time professor at the University of Technology Eindhoven, heading the chair on Video Coding and Architectures. In the period 2008 to 2010, he was VP Video (Analysis) Technology at CycloMedia Technology. Since 2011, he has been an assigned full professor at Eindhoven University of Technology and appointed scientific director Care and Cure Technology and theme leader Smart Diagnosis in the University Health program. He is a national and international expert in video surveillance for safety and security and has been involved in multiple EU projects on video analysis, featuring object and behavior recognition, and also surveillance projects with the Harbor of Rotterdam, Dutch Defense, Bosch Security Systems, TKH-Security, etc. He is board member of DITSS and R&D advisor to multiple companies. He is fellow of the IEEE, has (co-) authored over 300 papers on video coding, analysis, architectures, 3D processing and their realization. He is the (co-) recipient of multiple papers awards of the IEEE CES, VCIP and Transactions papers.