Adding context information to video analysis for surveillance applications

S. Javanbakhti[1], X. Bao[1], I. Creusen[1,2], L. Hazelhoff [1,2], W.P. Sanberg[1], D.W.J.M. van de Wouw[1], G. Dubbelman[1], S. Zinger[1], P.H.N. de With[1,2]
1. Eindhoven University of Technology, Eindhoven, The Netherlands
2. Cyclomedia Technology B.V., Zaltbommel, The Netherlands

ABSTRACT

Smart surveillance systems must be able to detect and track moving objects, classify these objects and detect the activities. Automatic object detection is currently being embedded in smart surveillance systems. To achieve a higher level of semantic scene understanding, the objects and their actions have to be interpreted in the given context. One of the challenging aspects of automatically understanding a scene, which is still a subject of active research, is the presence of contextual information. In many cases, contextual information is required in order to correctly interpret the actions. This chapter explores the extraction of contextual information in several aspects such as spatial, motion, depth and co-occurrence, depending on applications. In this chapter, it is shown that using contextual information enables the automated analysis of complicated scenarios that was previously not possible using conventional object classification techniques.

Keywords: smart surveillance, video understanding, context, spatial context, motion, depth information, classification.

INTRODUCTION

Automatic surveillance video understanding is one of the ultimate application fields of computer vision research. The objective of visual surveillance systems is not only to use cameras instead of human eyes, but also to perform surveillance automatically using video analysis. One of the key objectives for automatic surveillance is to selectively guide the attention of human operators to potentially suspicious activities. The important arguments for doing so are twofold. First, automation reduces labor costs so less human operators have to observe many video feeds simultaneously, a job that is both error-prone and tedious. Second, with the huge amount of information contained in parallel viewing of video channels, the chance of missing an important event in one of the many surveillance videos is high. Smart surveillance systems should be able to at least detect and track moving objects, classify these objects and interpret their activities. A large number of surveillance systems have been proposed in recent years. These systems still need improvement in terms of reliability and robustness with respect to event interpretation and a real semantic understanding of scenes. These improvement points can be realized by adding additional information about those objects and/or scenes, so that a better classification and understanding is achieved. This extra information is typically the context of the behavior of objects or of the scene. This chapter aims at exploiting contextual information in two ways: (1) to help to better interpret events based on object behavior with higher reliability and robustness, (2) obtaining a higher semantic level of scene understanding by adding contextual information about the scene itself.

Although it is present in scenes, the automated interpretation of the events and associated object detection in a monitored space is typically completely based on object detection and recognition, while the contextual information e.g. about the surroundings of the objects is overlooked. For example, a car detected on a parking place is a normal situation, whereas a car standing on tramway rails is a reason to raise an alarm. In this example, the rails are static object information from the surroundings acting as contextual information. The benefit is that a higher level of understanding about the parked car is obtained. In this case, a better object detection is achieved leading to an improved event understanding. In general, context can be applied at different levels: the involved features of an object usually at the pixel level, information about the object itself, and at the level of scene understanding (e.g. event classification and event detection). Challenges here include several aspects. First, what kinds of algorithms are needed for extracting the additional information from a surveillance video in order to contribute to the semantic meaning. Second, the ways for including the context information at the various levels is another important challenge. For example, scenes can contain static or moving regions and objects. The object recognition research is mostly based on considering objects in isolation from the surrounding scene. However, this chapter considers that object detection and eventually event detection should not happen in isolation, i.e., the process of recognizing one object in a scene can be influenced by the presence of additional information such as motion- or depth-based features, presence of other objects, as well as by the semantic context of the scene.

This chapter intends to show that using contextual information enables the automated analysis of complicated traffic surveillance scenarios that was previously not possible using conventional object classification techniques. Moreover, it is illustrated that using contextual information increases the reliability of the object detection tasks which helps to achieve high level of surveillance scene understanding. Although there is no standard view on how context information should be classified, it is evident that such information can contribute to various analysis levels for a surveillance system. This chapter explores the following aspects of contextual information:

a. Feature information, such as color, texture, shape, depth, motion, etc,
b. Spatial region properties, such as semantic region labeling,
c. Semantically meaningful information - specific objects or behavior, etc.

The intention is to use the above examples of context information to improve and assist in a better semantic interpretation of events occurring in the monitored space. The chapter contribution consists of exploiting additional information from various features, image regions, etc to augment the performance towards a better object detection due to additional context, or an improved event understanding resulting from new contextual information. The chapter validates these ideas by presenting a number of use cases.

The chapter has the following structure. The next section presents the background on context-based reasoning for surveillance video analysis. Then, the algorithms for context extraction are presented, where feature, spatial- and semantic-based contextual information is explored. In the section on use cases for scene understanding, several surveillance video analysis systems are illustrated which profit from the contextual information. Finally, the conclusions and future work are discussed.

BACKGROUND ON CONTEXT-BASED REASONING FOR SURVEILLANCE VIDEO ANALYSIS

In recent years, the demand for context modeling in computer vision applications has considerably increased. A primary reason for exploiting the context is to enhance the performance of existing image/video analysis techniques by incorporating additional information obtained from the surroundings of the object of interest. The object recognition research is mostly based on considering objects in isolation from its surroundings in order to be not distracted from other textures or structures. However, in most real-world object recognition tasks, the context provides a rich source of information that can help to improve the performance of the task (Marques *et al*., 2011).

There is no common understanding on the correct classification of different types of context into meaningful groups and categories. Therefore, this section continues with the initial structure provided in the previous section. It is important to acknowledge that context information can be useful at different levels of a surveillance analysis system and consider it to be applicable to (1) *feature* information (color, texture, depth, motion), (2) *spatial region* properties (semantic region labeling,…),(3) *semantically meaningful* information (specific objects or behavior,…), up to (4) the *scene level* (landscape elements, city view, traffic flow, …). Another way of classifying context information is to consider not natural features but mathematical features. In a 3D spatio-temporal space, context refers to any *spatio-temporal* information (Sun *et al.,* 2011). In (Galleguillos *et al*., 2010)., three main types of contextual information that can be exploited in computer vision solutions are proposed. First, *probability (semantic)* information refers to the likelihood of an object being found in some scenes but not in others. Second, *size (scale)* exploits the fact that objects have a predetermined size in relation with other objects in the scene. Third, the *position (spatial)* corresponds to the likelihood of finding an object at specific positions in the scene with respect to other objects. This section reviews the latest research developments concerning the contextual information extraction which are applied in surveillance video analysis. Since there is no clear preference for a certain information structure to use context, the chapter will follow the ordering as mentioned above, ranging from features to scene level elements. In particular, the state-of-the-art literature is discussed at the feature level with respect to color, texture, position, motion and depth, and at the scene level, semantic region labeling and the use of traffic guiding elements are addressed.

(a) Features as context

Color and *Texture* (orientation) are a set of primitive visual features concerning the fundamental input to the human visual system (Marques *et al.,* 2011). Color is one of the most straightforward features utilized by humans for visual recognition and discrimination (Smith *et al.,* 1995). Different color spaces for feature extraction will generate different results (Yu *et al.,* 2002). This chapter explores the most appropriate color spaces for video analysis, such as HSV (Bazzani *et al.,* 2011) and RGB. Texture analysis methods have been published numerously, it would be hard and fastidious to make an exhaustive review of them. Among the most popular methods, one can find those based on co-occurrence matrices to extract characteristic features of the observed texture (Rellier *et al.,* 2004). These approaches give good results for mono-spectral images, but the adaptation to hyper-spectral images is computationally expensive. Besides accurate time-frequency location, Gabor features provide also robustness against varying brightness and contrast of images (Kong *et al., 2003*), occurring frequently in surveillance video.

In this chapter, in a study, a group of Gabor filters (Kamarainen *et al.*, 2002) is applied to extract texture features.

Motion is another category of feature-based contextual information. Motion features are very important in surveillance systems. Motion-based segmentation helps detecting regions corresponding with moving objects such as vehicles and humans. Detecting moving regions provides a focus of attention for performing tracking and behavior analysis because only these regions need to be considered. Conventional approaches for motion segmentation include block-matching, image differencing, background subtraction and optical flow (Hu *et al.,* 2004). The block-matching algorithm (BMA) is a well-developed technique for motion estimation in video sequences. It is widely employed in video compression and coding (Huang *et al.*, 2006). However, its performance is largely affected by the choice of the block size, which makes BMA rarely used in motion segmentation. Image differencing or background subtraction techniques are often used to find the moving blobs in consecutive frames (Arshad *et al.,* 2010; Broggi *et al.,* 2088; Tamersoy *et al.,* 2009; wang *et al.,* 2005). The drawback of these methods is that they are sensitive to sudden changes in the background, e.g. lighting changes. Optical flow based methods can detect moving objects even in the presence of camera motion (Hu *et al.*, 2004). Therefore, it is widely used in the surveillance research domain for extracting moving objects (Aslani *et al.,* 2013; Zhang *et al.,* 2010). However, the above-discussed techniques employ motion features and extract binary "blobs`` at a very early stage, so that the merging and separation of blobs become complicated steps, due to the lack of features. In this chapter, motion features are well explored at the pixel, segment and region levels and in different stages. This enables the modeling of motion context into two aspects: motion *similarity* and motion *saliency*. Based on these two aspects, the explored motion context can provide meaningful information for typical tasks in surveillance such as tracking and behavior analysis.

Depth information forms an alternative category of contextual information which refers to multimodal signal analysis. This chapter presents a segmentation method that combines luminance and depth values, thereby improving the segmentation. For this purpose, a flexible segmentation approach is needed which can easily include texture and shape information as well. Segmentation graphs meet these demands. Graphs can be defined over multiple scales, or provide hierarchical segmentation (Sanberg *et al.*, 2013). This makes graphs flexible to incorporate information from multiple signal modi. Moreover, graph structures can be implemented and segmented efficiently (Felzenszwalb and Huttenlocher 2004). In the survey of Peng *et al.,* (2013), several graph-based texture segmentation methods are discussed and compared. From their quantitative analysis, the authors conclude that the method of Felzenszwalb and Huttenlocher (2004) (Local Variance Segmentation, LVS) performs best in extracting the key structures in an image, especially when there are many objects present, and has the best average segmentation quality. Furthermore, the complexity of LVS is low, making it an attractive method to build upon. In this chapter, the LVS method is extended. Some ongoing research is also addressed in the area of intelligent transportation systems, where depth information is exploited in a collision avoidance system. This is will be further addressed in one of the use cases.

(b) Objects/regions as context

Automatic region labeling with semantically meaningful labels, such as road, water, etc. is used to extract spatial contextual information. The annotation of regions is helpful for improving the object-of-interest detection because the object position in the scene can then be exploited. For example, in port surveillance, the detection of the water region can help detect ships, because ships can only travel within the water region. Existing approaches in region labeling (Carson *et al.*, 2002; Li *et al.*, 2003) often produce a set of textual semantic labels as “words”, describing the image content without linking these words to particular segments of the image. The annotation of regions ignoring their context, and focusing only on the information within the object boundaries (such as color and texture information) is often an impossible task (Boix *et al.,* 2012). Kluckner *et al.,* (2009) propose a labeling approach by integrating additional contextual constraints such as class co-occurrences into a randomized forest classification framework. Ladicky *et al.,* (2010) incorporate object co-occurrence in Conditional Random Fields (CRF). The co-occurrence model tends to require large numbers of training samples to estimate the correct probability (Jeon *et al.,* 2003). In this chapter, a general framework is developed as a case study for performing automatic semantic labeling of video scenes based on the observation that each region is more likely to be found at a specific vertical position.

Traffic guidance elements are basically objects along roads or painted onto roads which refer to semantic context. They can include lane marking, traffic lights, traffic signs, etc. Automatic traffic sign detection is helpful for improving the traffic scene understanding because it supplies important contextual information. Traffic signs can be found along various kinds of roads, making their recognition an important task for traffic analysis in both urban and countryside areas. In traffic surveillance, the interpretation of traffic signs can help detect cases for law enforcement, e.g. traffic participants not following the rules, for example illegal parking in specific areas, pedestrians crossing the road in illegal/dangerous locations, etc. Recognition of particular signs allows for automated decision making, without the need of manually setting up rules for each camera. For in-car applications, there are some existing systems for traffic sign recognition. Escalera *et al.,* (2003) introduce a traffic sign detection and recognition system for intelligent vehicle applications. Their approach is to classify RGB values 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 color information after thresholding. Sign recognition is performed based on neural networks and a custom color representation. Ruta *et al.,* (2010) describe a traffic sign recognition system that works in real-time on video. The detection stage is again based on color classification followed by equiangular polygon detection. These approaches are both interesting and suitable for real-time applications, however, using only color information is challenging to make this stage robust to the widely varying lighting conditions and sign conditions that occur in large-scale datasets and it may lead to a reduced performance (Creusen *et al.,* 2010). In addition, in-car systems often only support a small subset of traffic signs, which are also relatively easy to detect. This chapter will discuss a case study where traffic signs are used as context information for classifying scene traffic events.

The sequel of this chapter will discuss a number of techniques for extracting and classifying contextual information in scenes. The presented approaches include semantic region labeling, motion estimation, traffic sign detection and depth-based scene segmentation.

CONTEXTUAL INFORMATION EXTRACTION APPROACHES

This section presents theoretical concepts and fundamental techniques of context extraction approaches. The natural division into levels ranging from features to scene understanding will not be strictly followed here, since some approaches build further on the techniques of an earlier approach. This section explores several contextual information extraction approaches at different levels, such as semantic region labeling, motion estimation, traffic sign detection and depth-based scene segmentation.

This section commences with considering spatial context because semantic annotation of regions is helpful to interpret scenes and detect object of interest in the scene.

(a) Semantic region labeling

Automatic natural scene understanding and labeling regions with semantically meaningful labels (e.g. road, sky, etc.) have increasingly attracted attention, since they are key aspects in supporting image and video understanding.

For a reliable model of a scene and associated context information, the labeling task involves image feature analysis at global and local scene levels. Although local features such as color and texture per pixel or region are instrumental for understanding, they are typically not uniquely determining the semantic meaning of such a region (e.g. sky and water). In general, local features can be influenced by the presence of other objects as well as by the overall context of the scene. A region corresponding to a specific semantic meaning normally covers a certain part of the color space and has a distinct texture and it is more likely to be found at a specific vertical position. In this section, 6 semantic labels are considered: sky, vegetation, construction, road, water and zebra crossing, plus the class “unknown”. The algorithm contains the following three stages.

Stage 1: *Uniform regions*. The image is divided into several regions with uniform color using graph-based segmentation.

Stage 2: *Feature extraction*. The region-based feature (vertical position) and pixel-based features (HSV color space and a group of Gabor features) of each segmented region are extracted.

Stage 3: *Classification*. The proposed algorithm employs two concepts in a sequential order.

1. Multiple-SVM (one vs. all). For each region class, an off-line separately trained SVM (Support Vector Machine) is used to classify pixels in that region.
2. Assigning labels. For each class, the percentage of pixels classified as belonging to this class in a given region, is measured. A specific label is assigned to a region when the percentage of positively classified pixels in this region is above a threshold.

Figures 1 and 2 depict the two instantiations of the proposed region labeling approach. The first approach adds *spatial context* in the form of the gravity-based model to the feature extraction stage. The second approach operates without gravity-based model in the feature extraction, but uses Global Region Statistics (GRS) in the classification stage. Both models are discussed while providing more details on each labeling stage below.

Figure 1. The gravity-based region labeling approach

Figure 2. The GRS-based region labeling approach

Stage 1: Uniform regions. An efficient graph-based segmentation from (Felzenswalb *et al.,* 2004) is adopted as pre-processing in the proposed region labeling, to achieve two objectives: (a) distinguish each region from other objects while preserving the overall characterization of the region itself, (b) perform fast segmentation to support a real-time application in surveillance systems (Bao *et al.*, 2013a).

Stage 2: Feature extraction. To train 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, complications arise, which can be solved by adding spatial context. It 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, the locally calculated pixel-based features and the region-based features are combined to achieve a more reliable region labeling approach.

a. Color: the HSV color space is used (Creusen *et al.,* 2013a).
b. Texture: a group of Gabor filters are applied.
c. Spatial Context (SC). This information becomes specific for the region when a vertical position is used. This builds a gravity-based model and helps to overcome the ambiguities of using only color and texture (Shotton *et al.,* 2009). For each pixel $(\boldsymbol{i}, \boldsymbol{j})$, its normalized vertical position is calculated $\boldsymbol{SC_{i,j} = i/n}$, where $\boldsymbol{i}$ is the row (line) number in the image and $\boldsymbol{j}$ the column number (horizontal pixel coordinate). Each region consists of $\boldsymbol{n}$ rows. This method is called a Gravity model.

Regarding global feature extraction, two methods are proposed: (1) spatial context in which the normalized vertical position for each pixel is calculated; (2) Global Region Statistics (GRS) in which intervals for mean and standard deviation of vertical positions for each specific region are obtained.

Stage 3: Classification Approaches. After segmenting the image and extracting the features, the labeling results can be obtained. The labeling is performed by a classification system based on an off-line trained SVM. Here, two approaches for region classification are presented, as depicted in Figures 1 and 2.

a. Fast classification using the gravity model. In this approach, color, texture and spatial context are used to train the SVM for each region class individually, to achieve unitary-category classification (i.e. an individual SVM is trained for each region type). Later, 100 pixels are randomly sampled from a segmented region. The previously trained SVM for the considered class assigns labels to each pixel as positive or negative, depending on the classification results. The percentage of positive samples is calculated in that region. Then, the region is labeled as belonging to the considered class (e.g. the segment depicted by sky), if this percentage of positive samples is higher than an empirically defined threshold. For multi-category labeling, each segment obtains one of the following labels: sky, vegetation, construction, road, water, plus the class “unknown”. To this end, each segment is classified by five SVMs using the unitary classification to obtain five numbers, 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 empirical threshold, which from now on it is called $\boldsymbol{T}_{\mathrm{e}}$. The empirical threshold $\boldsymbol{T}_{\mathrm{e}}$ for each region is set to 0.5.
b. Classification based on the GRS-based model. The GRS is defined as the standard deviation and mean of the region position. It is assumed that there are $\boldsymbol{M}$ regions of a particular type, for example sky, in the training set of images. For each region, mean values $\boldsymbol{\mu_k}(\boldsymbol{k = 1, \dots, M})$ of the vertical positions of its pixels are calculated. The standard deviations $\boldsymbol{\sigma_k}$ of the vertical pixel

positions for each region are also calculated. Then minimum and maximum values for all means and standard deviations for this region type are taken: $\boldsymbol{\mu_{min}} = \mathbf{min}(\boldsymbol{\mu_1}, \dots, \boldsymbol{\mu_M})$, $\boldsymbol{\mu_{max}} = \mathbf{max}(\boldsymbol{\mu_1}, \dots, \boldsymbol{\mu_M})$, $\boldsymbol{\sigma_{min}} = \mathbf{min}(\boldsymbol{\sigma_1}, \dots, \boldsymbol{\sigma_M})$ and $\boldsymbol{\sigma_{max}} = \mathbf{max}\,(\boldsymbol{\sigma_1}, \dots, \boldsymbol{\sigma_M},)$. In this way, intervals for mean and standard deviations for the region position are obtained. It is assumed that the mean value of vertical pixel positions lies in the interval $(\boldsymbol{\mu_{min}}, \boldsymbol{\mu_{max}})$ and standard deviation – within $(\boldsymbol{\sigma_{min}}, \boldsymbol{\sigma_{max}})$. For a correctly labeled region, the region borders are in the typical interval values for mean and standard deviation of the vertical positions. Therefore, for assigning a label to a region, it is checked that both following conditions are satisfied: (1) the percentage of positively classified pixels exceeds the threshold $\boldsymbol{T_e}$; (2) the mean and standard deviation of the vertical positions of the pixels lie in the intervals as discussed above.

The above models for spatial context and classification in this section can be exploited for an improved region labeling approach, featuring besides color and texture also the vertical position as part of a gravity-based model. The models provide a general framework for based on spatial context (in this case vertical position information) for labeling each region. For local feature extraction, a group of Gabor filters is selected combined with the color features. For fast classification, it is possible to apply random sampling for each segment and the subsequent multiple-SVM classification is based on a probability model of the segment to be classified as a specific region type. As an alternative, a system without the gravity-based approach may be pursued as well, but based on the Global Region Statistics model. This model involves the computation of mean and standard deviation of the vertical region positions. Both systems propose algorithms with relatively high accuracy and low computational complexity, so that they are suitable for real-time implementation in embedded video surveillance.

It should be noted that motion also plays an important role in analyzing a scene in a surveillance video. In the following section, motion-based features are presented and incorporated in the semantic region labeling to improve moving object detection.

(b) Motion estimation

Motion, or temporal, features are very important context in smart surveillance systems. More precisely, the detection of moving regions enables further local and more detailed processing, such as moving object detection, tracking and behavior analysis because only these regions need to be considered. Temporal features are widely explored in moving regions/object detection (Wei *et al.*, 2009). However, a temporal feature approach either relies on multi-source images or a complete tracking system, which both have a high complexity for real-time applications. In order to create an approach suited for real-time processing, a motion saliency analysis is performed. The context of the scene is modeled by first segmenting the video frame and semantically labeling the segments, such as water, vegetation, etc. Then, based on the assumption that each object has its own motion, labeled segments are merged into individual semantic regions even when occlusions occur. For example, ships and harbor infrastructures can have similar appearances, however, ships are always moving in surveillance videos, whereas infrastructures are typically stationary. These features can be explored to distinguish moving objects from the background or from other objects. Therefore, motion features are based on analyzing the motion pattern of the moving objects and modeled as motion context. In particular, the analysis consists of two aspects: motion similarity and motion saliency. In this approach, motion features are explored at the following three levels.

1. *Pixel-based motion* vectors are computed as basic motion features.
2. *Segment-level motion* is analyzed to group labeled segments into semantic regions based on motion similarity.
3. *Region-level motion* is explored to distinguish ships from other unknown objects based on motion saliency. Because of the detailed motion analysis, a higher robustness is established for occlusions, as the algorithm can distinguish between different moving objects.

In motion similarity analysis, the motion context is modeled based on the assumption that each semantic region has a particular region-level motion which distinguishes it from the others. Firstly, the above-discussed region labeling is applied to the video frame which produces labeled segments. Then, the pixel-wise motion is calculated using optical flow (Liu *et al.,* 2009). Then a modified Statistical Region Merging (SRM) (Bao *et al.,* 2013a) is employed to group the labeled segments. A merging predicate $\boldsymbol{P(C_i, C_j)}$ is defined to determine whether two regions $\boldsymbol{C_i}$and $\boldsymbol{C_j}$ are from the same statistical region ($\boldsymbol{i \neq j}$). Instead of using color features as in (Dalal & Triggs, 2005), the criterion for motion features is added as an additional constraint on the segments being of the same region type (same label). An average flow vector is explored, which is the average of flow vectors of all pixels in a segment, to represent the motion of a segment. Based on the over-segmentation results, a motion map is created by calculating the values of magnitude ***MAG*** and angel ***ANG*** for each average flow vector. In the motion map, the meaningful regions representing objects with inertia should have a common homogeneity property in three ways: (1) in a certain statistical region, each statistical segment has the same expectation in both ***MAG*** and ***ANG***; (2) for two adjacent statistical regions, the expectations different from each other in either ***MAG*** or ***ANG*** values; (3) each statistical region should contain segments with the same label. By exploring the motion context in similarity analysis, the labeled segments are grouped into individual semantic regions with a particular regional motion.

As for motion saliency analysis, the motion context is modeled based on the fact that moving object should have its own motion pattern and its motion is more significant than the motion of its surroundings (local background). Motion saliency is determined at the region level to avoid expensive pixel-based saliency checking as described below.

a. *ROI (Region-Of-Interest) extraction*. The ROI is extracted including the outer part of a candidate object $\mathbf{C_{obj}}$ and the local background $\mathbf{C_{bg}}$ around it. $\mathbf{C_{bg}}$ is defined as in (Bao *et al.,* 2013a). Only the outer part of an object is considered because the inner parts of objects are normally in same color which tends to bring errors in motion estimation. Morphological operations are used to obtain a set of ROI, each of them containing a candidate object.
b. *Motion calculation at the region level*. The region-level motion of $\mathbf{C_{obj}}$as $\mathbf{v_{obj}}$ and $\mathbf{C_{bg}}$ as $\mathbf{v_{bg}}$ are calculated.
c. *Motion saliency criterion*. Two criteria are incorporated to model the motion saliency: $\frac{|\mathbf{v_{obj}} - \mathbf{v_{bg}}|}{|\mathbf{v_{obj}}|} > \mathbf{T_1}$ and $|\mathbf{v_{obj}} - \mathbf{v_{bg}}| > \mathbf{T_2}$. In the equations, $\mathbf{T_1}$ and $\mathbf{T_2}$ are set based on experiments with real-world data.

The above framework presents an approach for automatic detection of moving objects in surveillance videos with robustness for occlusions. In this approach, important elements from the visual, spatial and temporal features of the scene are used to create a model of the contextual information and perform a motion saliency analysis. The context of the scene is modeled by first segmenting the video frame and then contextual labeling of the found segments, such as water, vegetation, etc. Then, based on the assumption that each object has its own motion, labeled segments are merged into individual semantic regions even when occlusions occur. The context is finally modeled to help locating a candidate moving object. Additionally, it is assumed that the objects should move with a significant speed compared to its surroundings. As a result, objects are detected by checking motion saliency according to the pre-defined criteria. This motion context extraction is very useful for e.g. traffic video analysis because it allows the detection of various traffic participants (vehicles, people, etc.), independently from their appearance, size and color.

Besides motion and regions, specific objects in a scene may occur that have an important semantic meaning for understanding of a scene. For example, in traffic surveillance, such important semantic objects are traffic guidance elements, since they add not only to the scene understanding, but also lead to prediction of the behavior of the traffic. Traffic guidance elements or objects can include lane markings, traffic lights, traffic signs, etc. It is clear that automatic traffic sign detection is helpful for improving the traffic scene understanding. Next section presents an approach for automatic traffic sign detection as a generic case for key object detection and classification.

(c) Automatic traffic sign detection

Traffic signs are a class of objects with high visual importance, appearing often in traffic surveillance video sequences. Automatic traffic sign detection is important for improving the car and person-based understanding in surveillance video and the associated decision making, as it provides semantic information about the scene. For example, in traffic situations, the interpretation of traffic signs can help detect traffic participants which do not follow the traffic rules, like e.g. parking in no-parking areas, pedestrians crossing the road in illegal/dangerous locations, etc. Recognition of these signs allows for automated decision processes, without the need of manually setting up rules for each camera.

It has been established and validated that the performance of state-of-the-art object detection algorithms, such as unmodified HOG, is insufficient for detecting traffic signs. The large variety of recording conditions, sign appearances, and large amounts of background clutter make the problem challenging. It has also been found that panoramic images, taken from a moving vehicle, are a challenging dataset to work with. The quality of such images can suffer from motion blur, significantly varying lighting conditions, various objects that can partially occlude the traffic signs, signs damaged by collisions, dirt, and color fading 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.

A.Traffic sign detection. The proposed traffic sign detector is summarized as follows. First, traffic signs are detected in the images using independent detectors for all sign appearance classes (i.e. red circular signs), using a multi-scale sliding window approach. This leads to a detection bounding box and a sign class. Next, the detected signs are classified to determine the exact type of traffic sign, using the extracted bounding boxes. Here, algorithmic efficiency is important for facilitating embedded surveillance applications and (semi-)automated car guidance. Therefore, such algorithms are typically implemented with high efficiency, leading to real-time performance on HD video data.

The detection algorithm locates traffic signs in the images. 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 Histogram of Oriented Gradients (HOG) by Dalal and Triggs (2005). This algorithm first obtains the local image gradients by applying a simple [-1,0,1] Sobel filter in both $\boldsymbol{x}$ and $\boldsymbol{y}$ directions. The obtained 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, the two closest orientation bins and the four nearest spatial bins (two in the $\boldsymbol{x}$ direction and two in the $\boldsymbol{y}$ direction) 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 sign/no sign classification is performed with a sliding window using a linear SVM classifier. This corresponds to a convolution of the features with the SVM kernel coefficients. This step is one of the most computationally intensive parts of the detector, and is therefore implemented with a fast algorithm in the frequency domain.

Implementation of fast algorithm in the frequency domain. The left image of Figure 3 shows a computational overview of a regular sliding window detector. The different feature maps represent the features extracted for each of the orientation bins, as well as for each of the color channels. Each of these feature maps can be interpreted as a 2D image, containing information about gradients in one particular orientation and color channel. Similarly, the SVM kernel can be split into separate 2D kernels for each orientation bin and color channel. In a sliding window detector with a linear SVM, each of these feature maps should then be convolved with its corresponding SVM kernel, and the results should be accumulated to obtain the final detection output. This process is repeated for each of the different sign appearance classes (i.e. red circular signs) that the proposed detector supports. By transforming the feature maps to the frequency domain, these convolutions can be performed as point-wise multiplications in the frequency domain. By storing the kernel maps in their frequency-domain representation, this transform does not have to be repeated each time. Additionally, the summation of the result maps from the different orientations and color channels can be performed in the frequency domain, which significantly decreases the number of inverse transforms that need to be performed. One problem that arises in this scheme is that the kernel maps use a lot of memory in their frequency domain representation. This is because the kernels, which are typically very small, need to be zero padded up to the same size as the input images. This means that the memory usage is determined by the image resolution, number of supported sign categories, number of orientation bins and the number of color channels. To avoid the dependence on image resolution, the Overlap-Add method is proposed, to split the convolution into multiple smaller parts. This way, the memory usage becomes small enough such that all the kernel maps can remain in memory, and there is no more dependence on image resolution (Creusen *et al.,* 2013b).

Figure 3. Side-by-side computational overview of a regular sliding window detector (left), and the improved frequency domain implementation (right)

Because color is an important visual marker for traffic signs, color information is included in the HOG features. This significantly improves the performance for traffic sign detection applications (Creusen *et al.,* 2010). Furthermore, a custom color transformation is proposed to further improve the detection performance (Creusen *et al.*, 2013b). For each pixel, this method calculates the distance in color space to a set of reference colors, as specified by:

$$\mathbf{p_t} = |\mathbf{p} - \mathbf{p_r}| \quad (5)$$

The transformed pixel $\mathbf{p_t}$ is calculated from the input pixel $\mathbf{p}$ using a reference color $\mathbf{p_r}$. 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 improves the detection performance.

After detection of the road signs, each detection is subject to classification to retrieve the specific sign category (e.g. speed limit of 50 km/hr), given the sign appearance class (red circular sign). Below, the classification component of the road sign recognition system (Hazelhoff *et al.*, 2012; Creusen *et al.*, 2014) is briefly discussed; a more extensive description is provided in (Hazelhoff *et al.*, 2014).

B. The classification component is based on a combination of Bag of Words (BOW), which aims at identifying objects based on the occurrence of characteristic key features of the object categories of interest, and structural information, which exploits the spatial distribution of the key features. The system overview is shown in Figure 4 and consists of two parts: the online classification stage, and the offline codebook generation and training stage. Below, the 6 stages of the online classification stage are briefly described, followed by a description of the offline codebook generation and training stage in (Hazelhoff *et al.*, 2014).

Figure 4. System overview of the road sign classification system

The online classification stage contains the following steps.

1. Image resampling: each input sample is resized to a fixed standard size of 100x100 pixels and converted to grayscale, as different road sign types do typically not contain color differences.
2. Descriptor extraction: the resized gray-scale image is resampled to five different sizes, and for each resampled image, SIFT descriptors (Lowe *et al.*, 2004) are extracted. These SIFT descriptors are known to offer very robust features when considering small image deformations, image rotations, blur and illumination changes, as reported in (Mikolajczyk *et al.*, 2005)).
3. Dimensionality reduction: The extracted descriptors are 128-dimensional vectors. Since the required memory for storing the visual dictionary and the matching time are linearly dependent on the descriptor dimensionality (without implementation optimizations), a reduced descriptor dimensionality lowers memory usage and improves execution speed. Therefore, the dimensionality of the descriptors is reduced using Principal Component Analysis (PCA). This method linearly transforms the descriptor to a vector of lower or equal dimensionality with orthogonal dimensions. These dimensions are ordered by the contained variance, where the dimension containing the largest variance is placed first. As will follow, especially this property contributes to higher matching speeds.
4. Structural feature vector construction: the structural feature vector consists of the concatenation of the descriptors extracted from the one-but-highest scale. This vector preserves the spatial information, as each part of the feature vector corresponds to a spatial image region.

5. Codebook quantization: Each described image region is matched against a pre-defined codebook, the visual dictionary. This dictionary contains descriptors representing characteristic image regions, referred to as visual words. For each descriptor, the closest visual word is searched and a word histogram is constructed, counting the frequency of each visual word being the closest. When using the Euclidian distance, the entry of the word histogram *W* can be calculated as:

   $$W\left(cw\right) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1 & \text{if } cw = \arg\min\limits_{c\in VD}\left(\left\|d_i - c\right\|_2\right)\\ 0 & \text{otherwise}\end{cases}, \tag{6}$$

   where *cw* denotes the specific codeword, *VD* denotes the visual dictionary and *N* denotes the number of descriptors.

6. Classification: the extracted word histogram and structural feature vector are both independently normalized and concatenated afterwards. The resulting vector is subject to classification, based on a one-versus-all classification scheme with linear SVM. In this scheme, each category is represented by a single classifier, and the classifier returning the highest score determines the output sign type.

Now the offline codebook generation and training stage are discussed. The visual dictionary used above consists of key features representative for the sign categories of interest. The standard BOW approach (Csurka *et al.,* 2004) utilizes a visual dictionary that contains key patterns that occur frequently over all training samples. Such a dictionary can e.g. be constructed by clustering the extracted descriptors using K-means clustering. However, this approach may lead to the merging of close, but distinct, patterns, as certain sign types only differ in very minor details, while others differ more broadly. Therefore, independent visual dictionaries are generated per sign category, and their combination is used as visual dictionary. These category-specific visual dictionaries are constructed using a fixed number of codewords per sign type, and contain patterns occurring frequently for the respective sign type. This approach has several advantages. First, larger training sets can be used, as the memory requirements are lower. Second, it enables training with a different number of training samples per sign category, which is beneficial for the application, as certain sign categories occur frequently, while others occur seldom. Third, the resulting codebooks are slightly more discriminative, but the aforementioned practical aspects are more important.

The individual visual dictionaries are generates as follows. For each sign category, every training sample is processed according to Steps 1-3 described above, and the resulting descriptors are stored in a matrix. After processing all samples, these stored descriptors are clustered, where K-means++ (Arthur & Vassilvitskii, 2007) clustering is employed. This process deviates from standard K-means in the initialization phase, which results in a better distribution of the clusters, and thereby leads to more characteristic visual words. After generation of all individual visual dictionaries, they are concatenated to a modular codebook, as visualized in Figure 5.

Figure 5. Illustration of our modular codebook, which consists of the concatenation of codebooks generated independently per sign category

After generation of the modular codebook, the classifiers used in step 6 of the online classification stage are trained. For this, Steps 1-5 are carried out for each training sample, and the resulting feature vectors are used for training of the linear SVM.

The above algorithm description and its details form a generic framework for automatic traffic sign detection in surveillance videos or similar types of meaningful objects with rich details and color, which provide informative information for traffic scene understanding. The traffic sign detector is based on the HOG algorithm, and is implemented in the frequency domain. The performance of the detection is improved by applying a color transformation. The traffic sign is classified by combining BoW and structural features, both based on SIFT descriptors. Each type of traffic sign should be equally represented in the dictionary, therefore a modular codebook is introduced. The experimental results on large and challenging datasets show that the both components perform well.

Another important dimension for context information in surveillance video understanding is depth information which refers to the geometry information of objects and/or parts of the scene. The next section illustrates the approach for depth-based scene segmentation.

(d) Depth information

The last type of contextual information which is presented in this chapter is depth information. Depth information usually involves multimodal signal sensing and processing of the scene, where visual sensing with surveillance cameras is combined with depth-based sensors such as used with radar signals (e.g. LIDAR) or the projection of a regular pattern of invisible infrared light (Kinect). Besides these active methods, also multiple cameras capturing the same scene from different angles allow 3D reconstruction when carefully tuned. Recently, these sensors have been broadly applied in multimodal visual sensor systems, providing information on the geometry of a scene.

Let us start with a brief introduction into depth sensing. The most commonly used depth sensing techniques will be briefly addressed: passive stereo cameras, active stereo cameras and Time-of-Flight (ToF) cameras. A passive stereo camera consists of two regular cameras that are placed in parallel. Depth information is inferred from the difference in image location of an object seen by the left and right camera. Assuming that the cameras are placed side by side and are coplanar (i.e. the optical axes of both cameras are parallel), this difference will be strictly horizontal. Hence, two corresponding points $(\boldsymbol{x_1}, \boldsymbol{y_1})$ and $(\boldsymbol{x_2}, \boldsymbol{y_2})$ in the left and right image respectively have the same $\boldsymbol{y}$ coordinate $\boldsymbol{y_1} = \boldsymbol{y_2}$. The horizontal displacement $\boldsymbol{d} = (\boldsymbol{x_1} - \boldsymbol{x_2})$ is referred to as the disparity. This coplanar setup allows for efficient stereo matching, where a disparity value is computed for each point in the image by searching along a single scanline in the other image. Assuming a pinhole camera model, the depth can then be computed as $\boldsymbol{Z} = \boldsymbol{fB}/\boldsymbol{d}$, where $\boldsymbol{Z}$ represents the distance, $\boldsymbol{f}$ – the focal length of the camera, $\boldsymbol{B}$ – the baseline (horizontal displacement between the two cameras) and $\boldsymbol{d}$ – the disparity. In practice, it is very hard to perfectly align two cameras to be coplanar. Fortunately, misalignment can be compensated by rectifying the images through epipolar geometry (Hartley & Zisserman, 2003). Active stereo cameras employ light, such as laser or structured light, to simplify the stereo matching problem. In the case of structured light, a known pattern is projected onto the scene, e.g. a grid of points or lines, and the deformation of this pattern when striking a surface can be used to calculate the depth and shape of the surface. It is also possible to add artificial texture to a scene by projecting a random or known pattern onto it (Kang *et al.,* 1995; Scharstein & Szeliski, 2003). The latter allows the aforementioned techniques for passive stereo cameras to be applied with higher robustness. Time-of-flight cameras and laser systems both emit modulated (infrared) light and measure the time of flight between the camera and the scene. Depth is then computed from the time of flight, which is proportional to the phase delay of the detected light signal. Typically, laser scanners 'scan' the scene point-by-point, while time-of-flight cameras measure the entire scene with each light pulse, allowing for a higher frame rate.

Passive stereo systems have difficulties with texture-less scenes and scenes with very repetitive texture, since it may no longer be possible to uniquely match point correspondences along each scanline. Active stereo cameras can overcome this issue by projecting a pattern on the scene. However, the latter has poor results when used in an outdoor environment, as the sunlight interferes with the used light frequencies. Although ToF sensors can be used outdoors and have a high frame rate, they are still sensitive to ambient light and have low resolution in general.

In this chapter, depth information is exploited in two different applications. In the first application, a segmentation method is developed which combines luminance and depth values using Multimodal Local Variance Segmentation (MLVS). In the second application - ongoing research in the area of intelligent transportation systems - depth information is exploited in a collision avoidance system.

The LVS approach is a graph-based segmentation method. In the original LVS approach, which is called uni-model LVS, a graph ***G(V,E)*** consists of vertices $\boldsymbol{v} \in \boldsymbol{V}$ that are connected by edges ($\boldsymbol{e} = \left(\boldsymbol{v_i}, \boldsymbol{v_j}\right) \in$ $\boldsymbol{E}$) **,** that all have a weight $\boldsymbol{w}(\boldsymbol{e})$ **.** Segmenting a graph is the problem of finding disjoint subsets $\boldsymbol{S_i}$ such that $\boldsymbol{U_i S_i} = \boldsymbol{V}$.

Felzenszwalb and Huttenlocher (2004) designed their uni-modal algorithm to result in a segmentation that is neither too coarse nor too fine, using the following definitions: (1) a segmentation is too fine when there are neighboring segments without evidence of a border between them; (2) a segmentation is too coarse if there is evidence of a boundary in at least one $\boldsymbol{S_i}$. To generate this segmentation, Felzenszwalb and Huttenlocher define the difference between segments, $\boldsymbol{Ext}(\boldsymbol{S_i}, \boldsymbol{S_j})$, as the minimal edge weight connecting the segments. Furthermore, they define the variation within a segment, $\boldsymbol{Int}(\boldsymbol{S_i})$, as the maximum edge weight in the Minimal Spanning Tree (MST) of the segment.

Now the principal steps of the segmentation algorithm are described. First, the graph is initialized with one vertex per pixel, where each vertex has an edge to each of its 8 neighbors. The edges are weighted with the intensity difference of the connected pixels. Furthermore, the segmentation is initialized with one segment per vertex, all having an initial threshold of $\boldsymbol{K}$, which is a predefined parameter. In the next, the seeding, step, the edges are sorted by their weight $\boldsymbol{w}$. Sorting is a crucial step, since it guarantees two requirements for proper execution. First, the edge under evaluation is always the connection with the lowest weight between two segments, which is equal to $\boldsymbol{Ext}(\boldsymbol{S_i}, \boldsymbol{S_j})$. Second, the edge under evaluation is always the connection with the highest weight within the new segment, which is equal to $\boldsymbol{Int}(\boldsymbol{S_i})$. The last step, i.e. the merging step, executes a boundary check for all edges in order of increasing weight. The boundary check $\boldsymbol{B}\left(\boldsymbol{S_i}, \boldsymbol{S_j}\right)$ is false when the following inequality holds:

$$\boldsymbol{Ext}\left(\boldsymbol{S_i}, \boldsymbol{S_j}\right) \leq \mathbf{min}\left(\boldsymbol{Int}\ (\boldsymbol{s}) + \frac{\boldsymbol{k}}{|\boldsymbol{s_i}|}, \boldsymbol{Int}\left(\boldsymbol{s_j}\right) + \frac{\boldsymbol{k}}{|\boldsymbol{s_j}|}\right), \qquad (9)$$

where $|\boldsymbol{S_i}|$denotes the size of a segment measured in pixels. If the boundary check is false, the segments are merged and the threshold of the new segment is set accordingly. From the equation above, it can be seen that the parameter $\boldsymbol{K}$ enables small segments to grow. Using an appropriate $\boldsymbol{K}$, regions with high level of detail can be segmented in small segments, and regions with few detail can be segmented in large segments.

In the proposed Multi-model LVS (MLVS), the information from multiple signal modalities is integrated into the uni-model LVS segmentation algorithm. Since in multi-modal systems multiple signals need to be processed simultaneously, it is necessary to adapt the initialization, seeding and merging and labeling steps accordingly.

First of all, in the initialization stage of MLVS, each vertex $\boldsymbol{v} \in \boldsymbol{V}$of graph $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$ is assigned an extra value per modality in addition to its color image pixel intensity. As a consequence, each edge obtains one extra weight per modality. Then, for the seeding and merging steps of multi-modal signals, a boundary function is defined based on the combined weights of several signal modalities.

To incorporate multiple weights in the analysis, they are combined into a single weight $\boldsymbol{W_c}$ by

$$\boldsymbol{W_c = \sum_{m=1}^{M} \alpha_m \frac{W_m}{A_m}} \ , \qquad (10)$$

where $\boldsymbol{M}$ is the number of modalities and $\boldsymbol{A}$ is a factor to normalize the weights of a modality to unity. For example, ***A=255*** for the luminance weight. The factor $\boldsymbol{\alpha}$ expresses the importance of a modality. In the experiments of this chapter, $\boldsymbol{\alpha_m = 1}$ is adopted for all $\boldsymbol{m}$.

In the seeding step, the edges are sorted to the value of $\boldsymbol{W_C}$. In the merging step, a boundary check analogous to the uni-modal variant is applied. After merging two segments, the internal differences of the new segment are updated using the weights of the edge under evaluation.

An advantage of this approach is that adding additional signals only requires normalizing signal values and a difference metric that matches the nature of the signal. In the experimental results of this chapter, good results are obtained when the absolute distance is used for scalar values and the Euclidian distance for vector values.

As an alternative strategy, a partial boundary function can be defined per modality and then, for instance, a logical OR function can be applied to decide on the merging of segments. This offers the flexibility of defining boundary functions that are adapted to the nature of the signal modality. For example, the depth value of a single object may vary, in contrast to its color value. In this case, a better alternative is to define a depth-boundary check based on a constant distance threshold, instead of an adaptively growing approach. Both methods are evaluated in this chapter in the next section.

In this chapter, the Multi-modal Local Variance Segmentation algorithm (MLVS) is developed, to successfully exploit multiple signal modalities. MLVS can incorporate any number of signal modalities in an efficient and flexible way. With a combination of optimized preprocessing and by combining Luminance and Depth modalities, MLVS improves the segmentation score in comparison to the unimodal (texture-based) baseline. It will be also shown how to exploit depth information in a collision warning system for intelligent transportation systems.

The next section discusses a number of use cases to illustrate that context information can be successfully exploited for enhanced object and scene understanding. Several case studies are presented to demonstrate the context information inclusion in video surveillance analysis applications. The case studies are also instrumental for evaluating the performance of the video surveillance analysis and for considering the efficiency of applying proposed contextual information. The presented use cases include moving ship detection in port surveillance, traffic action recognition and automatic collision avoidance for intelligent vehicles.

USE CASES FOR SCENE UNDERSTANDING

In this section, the efficiency of applying the proposed context extraction approaches is explored for several real-world use cases. Several traffic surveillance use cases are considered, such as moving ship detection in port surveillance, traffic action recognition and automatic collision avoidance for intelligent vehicles. In the first scenario, which refers to automatic moving ship detection in port surveillance, the semantic region labeling is combined with motion information. In the traffic action recognition, a framework based on semantic region labeling and automatic traffic sign information is proposed. For the last use case, depth information is explored in a segmentation approach which is further discussed in the collision avoidance system use case. In this section, the presented frameworks are mostly generic and do not depend on the type of scenes, while the proposed fast algorithms allow real-time execution.

Use case 1: Motion context and region labeling for moving ship detection in port surveillance

In port areas, various hazardous scenarios occur caused by heavy traffic conditions and the mixing of large sea ships with local smaller vessels. In particular, dangerous situations can occur when small ships travel in the radar ‘shadow’ of large ships, so that they become invisible for the radar system and the harbor management. Evidently, supplementary visual surveillance is a possibility but because of the large diversity of ship functionalities and shapes, human visual inspection is highly laborious and error-prone. Automatic ship detection is an attractive research topic in the field of port surveillance, which can nurture various applications, such as vessel traffic monitoring, ship identity management and smuggling prevention.

Although video-based techniques are broadly explored for vehicle detection along roads, video analysis for ship detection still remains a domain of active research. Ship traveling in highly dynamic water regions and complex surroundings largely limits the usage of conventional background modeling, which is typically applied in relevant research. Furthermore, highly variable ship appearances intrinsically bring in difficulties for constructing robust matching templates.

For the current application, a robust approach is proposed to detect moving ships by jointly using the region labeling, semantic and motion context. The flowchart of the moving ship detection involves a sequence of processing steps, as depicted in Figure 6. The spatial and semantic context is extracted as the basis and followed by the motion context modeling through motion similarity and saliency analysis. First, a graph-based segmentation (Felzenszwalb & Huttenlocher, 2004) is employed to divide a video frame into segments. The object-centric region labeling is then employed to classify those segments into three classes: water, vegetation and “unknown”. The experimental results on region labeling have been presented in the previous section. The labeled segments are then used to analyze motion similarity. Adjacent segments with the same labels and statistically similar motion are merged into semantic regions, through which occluded regions are also separated from each other. These regions are analyzed based on semantic, spatial and scale constraints to provide knowledge of locations of candidate ships. Based on the common understanding that ships should have significant motion, the regions with salient motion are detected as moving ships. In this chapter, salient motion is defined based on a set of criteria to distinguish it from other types of motion, such as the scintillation/ripples of the water surface and the wind-based motion of vegetation. This use-case scenario commences with the performance evaluation of the region labeling approach.

Figure 6. Flowchart of the moving ship detection using context

A. Region Labeling Results. To evaluate the semantic region labeling results a broad dataset has been constructed which consists of images from multiple Internet datasets and a personal archive. The images contain six classes of regions (sky, vegetation, road, water, construction and zebra-crossing) plus the class “unknown”. The dataset consists of 255 images: 121 images for training, and 134 images for testing. For the segmentation, parameters are set according to (Bao *et al.,* 2013a). The means and variances in the GRS-based model are calculated based on 51 images from the training set. To evaluate the performance of the region labeling algorithm, the Coverability Rate (CR) is used, which measures how much of the true region is detected by the algorithm (Bao *et al.,* 2013a).

The gravity model is tested on 30 images of the dataset in three different color spaces: CIE L*u*v*, RGB and HSV. Experimental results show that the gravity model in HSV color space improves the results by 3%. Therefore, the HSV color space is chosen for the classification.

In order to benchmark the proposed region labeling approaches, they are compared with two state-of-the-art approaches: Bao *et al.,* (2012a) and Millet *et al.,* (2005). Here, the unitary-category classification of Bao *et al.,* (2012a) is extended into multi-category classification and contextual information is applied as an additional feature. The rule-based approach proposed by Millet *et al.,* (2005) relies on pre-knowledge on the relative spatial positions between regions. Table 1 shows the results of applying the gravity-based model and GRS-based model approaches, compared to Bao’s and Millet’s algorithms on 112 images of our dataset. It can be observed that the gravity-model approach results in a higher CR. The gravity-based approach outperforms the recently published algorithm of Bao *et al.,* with approximately 2%. The gravity-based model also surpasses Millet *et al.,* (2005), while preventing any preset rules which reduce flexibility of the method. Unlike Millet, the gravity-based approach does not need to be rebuilt if a new region is added. Figure 7 illustrates a challenging image containing several regions of interest where the color information is quite poor with only small color differences between neighboring regions. It can be observed that the gravity-based model achieves results that better correspond to the ground truth.

The proposed algorithm is tested on the LabelMe (Russell *et al.,* 2008) and WaterVisie (Bao *et al.,* 2013a) datasets. From LabelMe, 142 images are randomly selected and divided into 102 training and 40 test images. The gravity-based model is also trained on 111 frames and tested on 16 videos of WaterVisie dataset. Figure 8 shows the original images of the datasets and the corresponding results of the gravity-based model.

Table 1: Coverability Rate (%) comparison for several semantic labeling algorithms

Figure 7. (a) Image from the dataset, (b) The gravity-based region labeling, (c) Region labeling from Bao et al., (2012a), (d) Region labeling from Millet et al., (2005), (e) ground truth of (a)

Figure 8. (a) Image from WaterVisie (Bao et al., 2013a), (b) The gravity-based region labeling of (a), (c) Image from LabelMe (Russell *et al.*, 2008), *(d) The gravity-based region labeling of (c), (e) Image from our dataset, (f) The gravity-based region labeling of (e)*

The average of CR over six regions for the gravity-based labeling approach is 93% for a private dataset, 94% for LabelMe and 96% for WaterVisie. This shows a significant improvement on LabelMe dataset compared to the 59% reported by Jain *et al.*, (2010). It should be noted though that Jain *et al.*, (2010) aimed at a clearly higher number of semantic region types, which is more difficult.

B. Ship detection results. The ship detection approach is tested using 16 different video sequences. Those sequences contain a significant amount of visual variation and categorized into three scenarios: (S1) single/multiple ship without occlusion; (S2) ships present with occlusions between different ships and/or clutter caused by vegetation; (S3) ships during sunrise or sunset moment (highly flickering water). Since the ship detection system is based on a pan-tilt-zoom camera, it is hard to benchmark this system, as the existing systems are mainly based on a static camera or an un-tethered camera. Furthermore, there is no benchmark dataset to evaluate ship detection systems in general. Therefore, the performances of different ship detection techniques are difficult to compare. To analyze the proposed approach, it is compared with the existing algorithms "Existing" (Bao *et al.*, 2013b) and "Cabin detector" (Wijnhoven *et al.*, 2010; Bao *et al.*, 2102b).

Table 2 shows the detection results of the three detection algorithms. In this evaluation, only the "miss or hit" is considered, which means that the detection is successful even if the detected ship contains a certain portion of non-ship objects. In Scenario 1, the ship detection approach using context information ("improved method") successfully detects 1,413 ships out of 1,593 ships, with a total precision of 94.5% and a recall of 88.7%. It gains around 2% in both precision and recall compared to the "Existing" method, directly benefiting from the more advanced context model. For "Cabin detector", the system obtains similar recall value at the cost of a low precision value. This is caused by the fact that the developed appearance model is simplified, but not distinctive enough for other textured objects. Therefore, it tends to generate false detections in vegetation or redundant detections along long vessels. In Scenario 3, the numerical results show that the "Improved method" approach outperforms the "Cabin detector" when a flickering background affects the ship appearance severely (e.g. sunrise in Figure 9(b) and Figure 9(j)). Since the "Improved" algorithm avoids using the detector which is trained for finding ship appearances ("Cabin detector"), it still performs well when the target ship differs from the training samples. However, the "Cabin detector" relies on frame-based features, so that the performance significantly deteriorates becoming poor, which is caused by the water flickering. Comparing between the "Improved method" and the "Existing" method, the higher values in both precision and recall demonstrate the advantage of the context-based approach.

Table 2. Ship detection results. TP+FN = manually marked ships, TP+FP = detected ships, TP = correctly detected ships.

A visual comparison is made between the three approaches is shown in Figure 9. For all typical frames, the "Improved" approach can successfully find the whole ship with a bounding box indicating the delineation of the ship's body. However, the "Cabin detector" can only mark the cabin parts of the ship

(Figure 9(j)) or it generates several detections along the ship body (Figure 9(i)). For small ships moving in the flickering water region, the "Cabin detector" misses the target (Figure 9(l)), while the "Improved method" can still find the ship with a boundary indication (Figure 9(d)). As for the "Existing" method, Figure 9(f) gives an example when a ship is cluttered by vegetation; the approach detects the ship and vegetation as one object because the ROI extraction only considers spatial adjacency of non-water segments. In Figure 9(g), the detection fails because the two ships traveling in two opposite direction are regarded as one ship, which makes the motion of the object not salient compared to the surroundings.

Figure 9. Visual comparison among the ship detection, "Existing" method and "Cabin detector": the first row shows the results for the ship detection approach; the second and third rows are the corresponding results of the "Existing" method and "Cabin detector". The 4 typical frames demonstrate the categorized 3 scenarios from left to right: a long vessel without occlusion (S1); a ship cluttered by vegetation (S2); two ships occlude each other (S2); a sailing ship during sunrise moment (S3)

In this section, an automatic ship detection system is presented for camera-based port surveillance, featuring the analysis of context information for improving the reliability. The information processing is mapped onto a sequential processing architecture, where the derived context information feeds the subsequent ship detection with specific information about the typical ship locations and candidates in the image. It simplifies the ship detection and makes it more robust and suitable for real-time implementation. The proposed algorithms are not limited to static cameras, and also enable the use on moving cameras.

The main advantages of using context information within the ship detection are as follows.

1. The system requires no prior knowledge of ship appearances and yet it works successfully for various types of moving ships (container ships, speed boats, tanker ships, finishing boats and sailing boats).
2. The system is able to handle occlusions between different ships and is robust to clutter caused by vegetation
3. All the proposed algorithmic subsystems are designed to operate on videos obtained by moving cameras.

In traffic surveillance video scenes, not only harbor monitoring, but also street is of high importance. In this case, a traffic sign should be detected to provide semantic information and semantic image regions of interest are road and zebra crossing. The next section shows how reliable contextual aspects, such as automatic region labeling and traffic sign context, lead to semantic understanding of a scene.

Use case 2: Automatic traffic sign detection and region labeling for traffic action recognition

This use-case scenario involves traffic action recognition for safety, using the context information of the scene. It commences with the performance evaluation of the primary individual components, such as region labeling and traffic sign detection. The region labeling results are presented in the Use case 1.

A. The traffic sign detector is evaluated on two criteria: execution speed and detection performance. To evaluate the gain in execution speed due to the frequency-domain implementation, it is compared to a regular pixel-domain convolution as implemented in OpenCV.

Figure 10 shows an overview of the execution time experiments. Obviously, the proposed frequency-domain approach leads to a significant reduction in processing time. For typical configuration of 20 classes, 48 feature maps and 8x8 kernels, the gain in processing time is a factor of 5.3.

Figure 10. Processing time of the sliding window stage of the traffic sign detector. In the above, the effect of the number of classes on execution time is plotted, while on the bottom, the effect of the number of feature maps is shown

In addition to processing time, also the performance of the detector is validated. For this experiment, a manually annotated test set containing 45,250 panoramic images is used. For training, 402 and 274 cropped traffic sign images are used for the Red/Blue circular signs and Blue parking signs respectively. In addition, 700 images without traffic signs are used as negative training samples. Figure 11 shows that the proposed transform improves the detection performance for these two classes.

Figure 11. Influence of the proposed color transformation on the detection performance for two sign appearance classes

Let us now provide the classification performance overview. The traffic sign classification system is extensively evaluated (Hazelhoff *et al.,* 2014) for two different sign appearance classes: red circular restriction signs and blue rectangular information signs. This work investigates the effects of the different system parameters on both performance and computational efficiency, resulting in an optimal parameter combination per sign appearance class. Below, the dataset and classification results are briefly summarized, the complete description of results is provided in (Hazelhoff *et al.,* 2014).

The traffic sign classifier is evaluated on two large datasets, representing both sign appearance classes. Each of the sets is constructed by manually labeling the detection output of the traffic sign recognition system (Hazelhoff *et al.,* 2012; Creusen *et al.,* 2014), which contains independent detectors per sign appearance class. These datasets are constructed from the detection output on about 100,000 street-level panoramic images, which are captured within different geographical regions, and recorded during different weather and lighting conditions, and with different capturing cars. The pixel resolution of the included detections varies between 40x40 and 240x240 pixels. Each dataset also contains a category representing false detections returned by the road sign detection algorithm. The dataset characteristics are summarized in Table 3. Figure 12 displays a selection of representative examples.

Table 3. Characteristics of the used datasets

Figure 12. Examples of samples present in the datasets used for evaluation of the road sign classification component. Figures (m)-(n) and (u) display falsely identified road signs present in our datasets

The traffic sign classification system is evaluated on the above described datasets using 10-fold cross-validation, as this approaches the systems performance when all samples were subject to training (Kohavi 1995). To numerically assess the classification accuracy, the balanced Classification Error (CE_{bal}) is computed by

$$CE_{bal} = 1 - \frac{1}{|T|}\sum_{t=1}^{|T|}\frac{1}{|S_T|}\sum_{s_t=1}^{|S_T|}\begin{cases}1 & class\ .outp\ . = t\\ 0 & \text{otherwise}\end{cases}, \tag{11}$$

where $\boldsymbol{t}$ denotes the category (out of $|\boldsymbol{T}|$categories), s_t represents a sample of category $\boldsymbol{t}$ (with $|\boldsymbol{S_t}|$ samples in total) and “class. outp.” is the classification output for sample $\boldsymbol{S_t}$. Additionally, the evaluation times were measured for each parameter setup. These evaluation times are measured in a Core i7 3930K, operating at 3.2 GHz, using single-threaded implementations.

Table 4 shows the balanced Classification Error and execution times obtained for both datasets, using the best parameter setup as found in [41]. As follows, the single-image classification accuracy exceeds 96%, even for the blue rectangular sign appearance class, which consists of many very similar sign categories (as shown e.g. in Figure 12 (p) - Figure 12 (q)) and categories with varying custom text, such as street name and place name signs.

Table 4. Overview of the classification results

B. Higher level of Scene Understanding. Now an example case is presented to demonstrate a situation where traffic signs help in the understanding of surveillance footage. Here, the security camera footage shows several scenes of actors in three scenarios, and the actors and traffic signs in the scene are automatically detected. The detected actors and signs are analyzed by a decision engine, which decides if situation requires operator attention. In the first scenario, the presence of a zebra crossing is detected because of the traffic signs, and this information is used to identify people who cross the road in illegal locations, which may lead to a dangerous traffic situation. In the second scenario, two contrasting scenes are shown, first a crowd gathers near a bus stop and waits for a bus to arrive, which is a perfectly normal situation and can be identified as such through the bus-stop sign. In the second scene, a crowd gathers to watch a fight, which is not a normal situation and requires security operator attention. Screenshots from the first two scenarios can be seen in Figure 13. In the third scenario, a pan-tilt-zoom camera is used to monitor parking spots. The signs indicating a parking spot for handicapped people are detected automatically. This information allows the operator to detect people who use the parking space for handicapped people unnecessarily. Screenshots from this scenario can be seen in Figure 14. These scenarios are described in more detail in (Creusen *et al.*, 2013a).

Figure 13. Screenshots from the zebra-crossing scenario are shown in (a) and (b). Figures (c) and (d) show screenshots of the bus-stop/fight scenario

Figure 14. Screenshots from the PTZ parking lot demo

The presented framework is capable to recognize actions in traffic surveillance video. It has been clarified that traffic signs provide contextual information for surveillance applications and help to interpret the actions of traffic participants, as well as to detect illegal or dangerous traffic situations automatically. The traffic sign detector is based on the HOG algorithm, and is implemented in the frequency domain. A color transformation is introduced to improve the detection performance. The traffic sign classification system is a combination of BoW and structural features, both based on SIFT descriptors. A modular codebook is introduced, to ensure that each type of traffic signs is equally represented in the dictionary. Finally, the performance of both components is evaluated on large and challenging datasets, and both are shown to perform well. The presented traffic sign detection is constructed from generic techniques and can be re-used and learned for the detection of other type of objects, depending on the studied use case or scenario.

In the traffic surveillance domain one of the interesting challenges is a collision avoiding system which allows preventive analysis of the traffic behavior. In the next section, some ongoing research is presented on exploiting depth information in a collision avoidance system.

Use case 3: Using depth information for automatic collision avoidance in intelligent vehicles

In this section, depth information is exploited in two different applications. In the first application, a segmentation method is developed which combines luminance and depth values. In the second application depth information is used in a collision avoidance system. First, the experimental results on the proposed segmentation approach are described. Then, the collision avoidance system is further discussed.

To train the color pre-processing settings, the challenging Berkeley Segmentation Data Set 500 (BSDS), presented in (Arbeláez *et al.*, 2011) is used. The setting for $\boldsymbol{\theta}_{\mathbf{Prec}}$ is selected such that it promotes an increase in precision for the uni-modal algorithm to ensure the detection of strong color boundaries. The proposed algorithm aims at increasing the recall by including detected boundaries from other modalities.

A: Depth-enhanced segmentation. Next, the proposed two multi-modal segmentation methods are executed on the NYU test set. The NYU Depth dataset V2 (NYU), presented in (Silberman *et al.*, 2012), contains aligned color/depth/normal-frames from a variety of indoor scenes (kitchens, bedrooms, office spaces, etc.), captured with a Kinect camera. For this experiment, $\boldsymbol{\theta}_{\mathbf{Prec}}$ is used on the Luminance signal Y and a number of different combined weights, additionally exploiting depth (D), normal (N) and angle (A) modalities. To evaluate the performance, recall and precision scores on edge pixels are measured, where recall is the ratio of true boundary pixels that are detected by the segmentation and precision is the ratio of detected boundary pixels that match true boundary pixels. As a final metric, the F-score is calculated, which is the harmonic mean of recall and precision (Figure 15).

Figure 15. Boundary detection results on the NYU test set, using the combined weight with different signal modalities (Luminance (Y), Angle (A), Depth (D), Normal (N)). The baseline method is LVS on the luminance signal

The leftmost image of Figure 16 shows an example for which a good increase in both the F-score and precision is achieved, by using the combined weight of the luminance and angle signals. The highest increase in recall is obtained with the second image, using partial boundary functions on the color and depth signals. The third image shows a good performance of the most stable multimodal approach (using a combined weight of luminance and depth values). The multi-modal segmentation scores are worst on the fourth image, mainly due to the normal signal that causes false boundaries on the wrinkled blanket. More details on the method, more experiments and additional qualitative and quantitative results can be found in (Sanberg *et al.*, 2013).

Figure 16. Images of the NYU test set. For each image, we show the original image with ground truth (top), our color segmentation result (middle) and a multi-modal result of interest (bottom); Applied methods from left to right: w_C(Y,A), $B(Y,D)$, w_C(Y,D), $B(Y,D,N)$. Here, w_C stands for the combined weight approach, B for the partial-boundary function approach. The used signal modalities are Luminance (Y), Angle (A), Depth (D), Normal (N)

To conclude, the proposed MLVS algorithm extends a uni-modal segmentation algorithm to perform multi-modal analysis, by introducing a method that can process any number of signal modalities that are available in a flexible way. Based on the quantitative analysis on the NYU dataset, the use of a combined weight of luminance and depth values improves the harmonic mean of recall and precision from 0.480 to 0.512, an increase of 6.7%.

B. Collision avoidance using depth. As mentioned above, the second application in this use case exploits depth information for a collision avoidance system, which is still under development. This automatic collision assessment system is embedded in intelligent vehicles in crowded urban environments. Throughout this section, the focus lies on collision assessment for trams, although the system can be applied to other vehicle types as well. Such a system consists of several blocks as depicted in Figure 17. First, the ground plane is detected. Next, potential obstacles are segmented directly in the disparity signal of the non-ground areas. Third, a zone is derived in which objects may collide with the intelligent vehicle, i.e. the tram. This is accomplished by detecting the track and/or lane and defining a region around it.

Figure 17. Overview of the collision avoidance system

The intelligent vehicle, in this case a tram, that operates in a crowded urban environment, is equipped with a passive stereo camera. Active stereo and Time-of-Flight (ToF) cameras are less robust in such outdoor lighting conditions, and therefore not considered here. Moreover, ToF cameras have a lower resolution and require additional processing to be applied from a moving vehicle.

As indicated earlier, it is impractical to perfectly align two cameras to be coplanar. Therefore, the stereo camera is first calibrated. Once the camera is calibrated, the (later) acquired stereo images can be rectified to ensure that point correspondences lie on the same scanline as is the case in a coplanar setup. This rectification process effectively reduces the search space to find the disparity of each point from 2D to 1D space, resulting in a significant speedup. Moreover, the robustness is improved by restricting the search space.

After calibration, the disparity map is computed. This map contains the displacement for each pixel in the left image to its corresponding point in the right image. Although many algorithms exist that compute a disparity map, this system applies a Semi-Global Block Matcher (Hirschmuller 2008). Figure 18 shows a disparity map obtained from a tram's viewpoint.

The depth is inversely proportional to disparity. Hence, the disparity map can be utilized to efficiently compute a 3D point-cloud representing the scene. The ground plane can then be described by a plane fitting through (part of) the point cloud. Although this plane is commonly assumed to be linear, this does not hold in general, since the slope of the road changes, e.g. elevated crossroads, viaducts, bridges, etc. In these situations, a linear ground plane assumption does not lead to a proper road model, because some obstacles are missed or false obstacles are detected. Therefore, a non-linear ground plane is obtained by applying a penalized regression spline to more accurately fit the real ground plane. To cope with outliers and points with high leverage, i.e. points which are far from the main body of points, an iteratively re-weighted penalized regression spline is applied using bi-square weighting and approximated Studentized residuals (Garcia 2010). Points with residuals above a certain cutoff, i.e. outliers, are set to zero and are effectively suppressed. Points with residuals within the cutoff value are assigned weights as bi-square weights of the residuals. The complexity of fitting a spline over the 3D point cloud is further reduced by first creating a ***y-z*** histogram, where it is assumed that the ground plane's slope only alters in the ***z*** direction. This reduces the re-weighted penalized regression problem to a 1D regression problem. The number of outliers is further reduced by applying a slope filter to the disparity map prior to the ground plane fitting, removing those disparity values that do not correspond to a sloped (road) surface. Although the iterative re-weighted regression spline can approximate the road surface well, the first iteration may sometimes fit to the incorrect plane in the scene. To overcome this problem, first a linear ground plane is estimated using the Hough transform, after which the initial residuals are computed with respect to this linear ground plane. The iterative re-weighting and spline fitting will then converge to the correct non-linear ground plane. Once the ground plane is found, the creation of the obstacle mask is straightforward. For each 3D point ***(x,y,z)***, the height is compared to the height of the ground plane at that position ***y = p(z).***

As can be seen in Figure 18 (b), there are many detected obstacles in the scene, of which there are only a few that may actually collide with the vehicle. For this reason, the tram track is detected and a collision zone is defined around this track as shown in Figure 19. The tram track detection in this system uses only a single monocular camera and its implementation falls outside of the scope of this text. In short, the system incorporates inverse perspective mapping and a-priory geometrical knowledge of the rails to find possible track segments. These are combined into a track using graph theory, which is solved as a max-cost arborescence graph.

Figure 18. Example disparity (a) and obstacle (b) map generated from the pair of stereo images on the left. The color of each point in the disparity map is inversely proportional to the distance of the point w.r.t. the vehicle. Objects that rise above the groundplane are marked by the color representing their distance to the vehicle

Figure 19. Example of the unsupervised tram track detection. A region in front of the track can be defined as collision zone, meaning the driver will be warned if an obstacle is detected within this region

In this section, depth data are used in two different applications. In the first application, a segmentation method is illustrated which is based on combining luminance and depth values. This improves the segmentation score by 6.8% for a large and challenging dataset of indoor scenes with bad illumination and high clutter. In the second application, in the area of intelligent transportation systems, depth information is exploited in a collision avoidance system.

The use cases presented in this section consider realistic traffic surveillance video analysis challenges, such as moving ship detection in port surveillance, traffic action recognition and automatic collision avoidance for intelligent vehicles. The contextual information has proved to be very useful for reliable scene understanding and behavior analysis. The presented frameworks are constructed with mostly generic components and do not depend on the type of scenes, therefore, they are suitable for real-time execution.

FUTURE RESEARCH DIRECTIONS

Multimodal sensing technologies will gradually substitute the currently dominating surveillance cameras, based exclusively on SD/HD videos. Dependent on the type of the area to be monitored, different combinations of sensing devices are possible. For example, infrared sensing, audio signals, depth information can be combined together for better context extraction and content analysis. Multimodal signal processing provides complimentary information to the scene understanding, which enhances robustness and reliability of the analysis. Optimally combining the information from the sensors will contribute to result enhancement in all stages/levels of video analysis: feature extraction, object detection, context extraction, scene interpretation and event detection. Furthermore, the proposed object detector systems can be extended to recognize many other types of objects that can serve as 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. Furthermore, the ship detection approach cannot handle the detection of temporarily static ships or long vessels across the whole frame with little visual changes. To deal with the limitation, a ship tracking algorithm can be developed for combined detection-tracking strategies to further improve consistency and robustness. Another promising research direction for smart surveillance development is distributed networks of sensors for monitoring entire regions and public areas. Such networks and the involved sensors will become intelligent in certain levels of understanding, so that e.g. moving objects can be tracked from one camera to another and their trajectory can then be reconstructed. Also, the sensors will share the event detection information and direct their attention in the direction where the danger occurs. For example, pan-tilt-zoon cameras can turn to observe a situation detected as potentially dangerous, based on video content analysis. In this case, the scene monitoring will be much better, compared to observing it with only one camera.

With respect to depth processing, the results of LVS have been improved with pre- and post-processing steps, but it is considered that the seeding stage allows further optimization, such that all edges are evaluated with a parallel, more global approach to avoid border isolation. In addition, the multi-modal algorithm should better exploit the different characteristics of signal modi. For example, the depth signal should primarily be used for separating foreground from background objects, and normals should mainly be applied to detect object surface boundaries. To this end, the boundary check or mode weight can be enhanced with e.g. the local confidence in a mode. Moreover, the developments of performance metrics for such complex depth-based scenarios are an essential part of the future research.

CONCLUSIONS

For smart surveillance systems, it is important to achieve a reliable scene understanding and, based on that, event detection and interpretation. This high level of understanding is enabled by the incorporating context information of the scene. The key objectives of this chapter are on explaining techniques for the extraction of contextual information and applying these techniques for a higher level of scene understanding of better object detection in video-based surveillance system. To this end, several contextual information extraction approaches have been presented, such as semantic region labeling, motion, traffic sign detection and depth information.

For automatic semantic region labeling, this chapter has discussed and illustrated contextual information based on spatial image features which corresponds to the likelihood of finding an object at specific positions in the scene, with respect to other objects. Here, a fast region labeling algorithm is presented to classify pixel regions of the video into broad categories such as sky, vegetation, water, road, and construction.

To provide motion information, a system is presented based on the practical assumptions that a moving object should have its own motion pattern and its motion is more significant than the motion of its surroundings (local background). Motion saliency is determined at the region level to avoid expensive pixel-based saliency checking.

Automatic traffic sign detection is presented as an example case of object detection for traffic guidance, and this is divided in two steps. First, independent detectors are applied for all sign appearance classes (i.e. red circular signs) by using a multi-scale sliding window, which leads to a detection bounding box and a sign class. Next, the detected signs are classified to determine the exact type of traffic sign, using the extracted bounding boxes.

Finally, the Multi-modal Local Variance Segmentation algorithm (MLVS) is used to utilize depth information, which refers to multiple signal modalities or multi-camera setups. It is shown that with a combination of optimized preprocessing and combining Luminance and Depth modalities, MLVS improves the segmentation in comparison to the unimodal (texture-based) baseline.

The contextual information extraction components are broadly tested for multiple datasets. Experimental results show that all components perform well for datasets. This chapter demonstrates that context information is essential for decision making about the semantic understanding in surveillance video. For this purpose, several use cases are presented, such as moving ship detection in port surveillance, traffic action recognition and automatic collision avoidance for intelligent vehicles.

In the ship detection scenario, an automatic ship detection system is presented for camera-based port surveillance. The contextual information is provided by motion information, and a region labeling algorithm. Knowing the candidate ships in the labeled region and the contextual ship area, motion saliency detection is the core function in the ship detection. The motion saliency is defined with two criteria that remove non-ship objects with small relative motion and static non-ship objects surrounded by small distracting motion. The ship detection system requires no prior knowledge of ship appearances and yet it works successfully for various types of motion. Large advantages are that it detects the entire ship instead of only a part of the ship, and it also produces a full pixel-true segmentation between the ships and their surroundings with a corresponding bounding box and an indication of the centroid and bottom line. Another conclusion for the ship detection approach is that it is able to handle occlusions between different ships and is robust to clutter caused by vegetation and all the proposed algorithmic subsystems are designed to operate on videos obtained by moving cameras. The system is compared to two recent ship detection algorithms and shows robustness and good accuracy for real-life surveillance videos.

In the traffic action recognition scenario, various traffic actions are better understood by the system when using contextual clues. The contextual information is provided by 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. The traffic action recognition scenarios are complex so that the dataset had to be partially captured by a group of volunteers. Given this constraint, it has been found that the proposed traffic action recognition system evaluations prove 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.

In the collision avoidance scenario, a stereo camera is used to provide a depth signal. This simplifies the analysis of complex traffic scenes, since the geometry of a scene can be analyzed directly, next to the information provided in the color appearance signal. To this end, the ground plane is extracted from the depth signal. Later, it is assumed that all other elements in the scene are obstacles. Potential collision is assessed by analyzing obstacles inside or nearing the tram way. This is detected in the color signal. The added value of such multi modal strategies is quantified by evaluating the Multi Modal Local Variance Segmentation algorithm on a large indoor data set.

The discussed algorithms and techniques are not limited to static cameras, and enable the use on moving cameras. Although the individual components of the above complex scenarios are broadly tested for multiple datasets, but the number of experiments at the scenario level is limited. This is mainly due to the lack of test material. Another conclusion is that context information is not complicated to extract from a surveillance video, while it adds significant value to the automated surveillance analysis.

The presented concepts of using context information at different levels for better automated scenario analysis are mostly novel, since they offer a higher level of understanding or a better robustness. This brings surveillance analysis generally at a higher level of quality, while improving the reliability of the video analysis at the scenario level. The frameworks are generic and do not depend on the type of scene, while the fast algorithms allow real-time execution. From the presented approaches and use-case studies, it can be concluded that context information is an important source for improving automated video surveillance analysis, as it not only improves the reliability of moving object detection (e.g. ships), but also enables scene understanding that is far beyond object understanding (e.g. traffic scenarios). This chapter contents proves this by demonstrating several surveillance scenarios where the semantic meaning of the events can only be detected due to the availability of the context. This brings video surveillance analysis to a higher quality level, while improving the reliability of the semantic interpretation.

REFERENCES

Arbeláez, P., Maire, M., Fowlkes, C. 7 Malik, J. (2011). Contour detection and hierarchical image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 33(5), 898–916.

Arshad, N., Moon,K. & Kim, J. (2010). Multiple ship detection and tracking using background registration and morphological operations. In Signal Processing and Multimedia, 123, 121–126.

Arthur, D. & Vassilvitskii, s. (2007). k-means++: the advantages of careful seeding. In Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms, 1027–1035.

Aslani, S. & Mahdavi-Nasab, H. (2013). Optical Flow Based Moving Object Detection and Tracking for Traffic Surveillance. International Journal of Electrical, Robotics, Electronics and Communications Engineering, 7(9). 772-776.

Bao, X., Zinger, S., De With, P. H. N. & Wijnhoven, R. (2012a). Water region supporting ship identification in port surveillance," in [Advanced Concepts for Intelligent Vision Systems, 444-454.

Bao, X., Zinger, S., de With, P. H. N., Wijnhoven, R. & Han, J. (2012b). Water region and multiple ship detection for port surveillance. In Proceedings of the 33rd WIC Symposium on Information Theory in the Benelux, 20-27.

Bao, X., Javanbakhti, S., Zinger, S., Wijnhoven, R. & De With, P. H. N. (2013a). Context modeling combined with motion analysis for moving ship detection in port surveillance. Journal of Electronic Imaging, 22, 041114.

Bao, X., Zinger, S., Wijnhoven, R. & De With, P. H. N. (2013b). Ship detection in port surveillance based on context and motion saliency analysis. In Proceedings of the SPIE 8663, Video Surveillance and Transportation Imaging Applications, 86630D.

Bazzani, L., Cristani, M., Perina, A. & Murino, V. (2011). Multiple-shot person reidentification by chromatic and epitomic analyses. Pattern Recognition Letters, 33(7), 898.903.

Boix, X., Gonfaus, J. M., Weijer, J. van de, Bagdanov, A. D., Serrat, J. & Gonz´alez J. (2012). Harmony Potentials Fusing Global and Local Scale for Semantic Image Segmentation. International Journal of Computer Vision, 96(1), 83–102.

Broggi, A., Cappalunga, A., Cattani, S. & Zani, P. (2008). Lateral vehicles detection using monocular high resolution cameras on terramax. In Intelligent Vehicles Symposium, IEEE, 1143–1148.

Carson, C., Belongie, S., Greenspan, H. & Malik, J. (2002). Blobworld: Image Segmentation using Expectation-Maximization and Its Application to Image Querying. IEEE Transaction on PAMI, 24(8), 1026–1038.

Creusen, I.M., Wijnhoven, R.G.J., Herbschleb, E., De With, P. H. N. (2010). Color exploitation in hog-based traffic sign detection. in Proc. of the IEEE Int. Conf. on Image Processing, 2669–2672.

Creusen, I. M., Hazelhoff, L. B. & De With, P. H. N. (2012). Color transformation for improved traffic sign detection", IEEE International Conference on Image Processing , 461-464.

Creusen, I.M., Javanbakhti, S., Loomans, M. J. H., Hazelhoff, L. B., Roubtsova, N., Zinger, S. & De With, P. H. N. (2013a). ViCoMo: visual context modeling for scene understanding in video surveillance. Journal of Electronic Imaging, 22(4), 041117-1-1/19..

Creusen, I. M., Hazelhoff, L. B. & De With, P. H. N. (2013b). A Frequency-Domain Implementation of a Sliding-Window Traffic Sign Detector for Large Scale Panoramic Datasets. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 1(2), 7-11.

Creusen, I. M., Hazelhoff, L. B. & De With, P. H. N. (2014). Exploiting street-level panoramic images for large-scale automated surveying of traffic signs. Accepted in: Machine Vision and Applications.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J. & Bray, C. (2004). Visual categorization with bags of keypoints. Workshop on statistical learning in computer vision, 1(1-22), 1-2.

Dalal, N. & Triggs, B. (2005). Histogram of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 886–893.

Escalera, A. De la, Armingol, J. M. & Mata, M. (2003). Traffic sign recognition and analysis for intelligent vehicles. Image Vision Comput., 21(3), 247–258.

Felzenszwalb, P. F. & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. International Journal of Computer Vision, 59, 167-181.

Galleguillos, C. & Belongie, S. (2010). Context based object categorization: a critical survey. Computer Vision and Image Understanding, 114(6), 712-722.

Garcia D. (2010). Robust smoothing of gridded data in one and higher dimensions with missing values. Comput. Stat. Data Anal. 54, 1167–1178.

Hartley, R. & Zisserman, A. (2003). Multiple View Geometry in Computer Vision (2 Ed.). Cambridge University Press.

Hazelhoff, L. B., Creusen, I. M. & De With, P. H. N. (2012). Robust detection, classification and positioning of traffic signs from street-level panoramic images for inventory purposes. Workshop on Applications of Computer Vision, 313 –320.

Hazelhoff, L. B., Creusen, I. M. & De With, P. H. N. (2014). Optimal performance-efficiency trade-off for Bag of Words classification of road signs. In proceedings of the international conference on pattern recognition.

Hu, W., Tan, Ti., Wang, Li. & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 34(3), 334-352.

Hirschmuller, H. (2008). Stereo Processing by Semiglobal Matching and Mutual Information, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30(2), 328-341.

Huang, Y., Chen, C., Tsai, C., Shen, C. & Chen, L. (2006). Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results. Journal of VLSI signal processing systems for signal, image and video technology 42(3), 297-320.

Jain, A., Gupta, A. & Davis, L. S. (2010). Learning what and how of contextual models for scene labeling, The 11th European Conference on Computer Vision, 199-212.

Jeon, J., Lavrenko, V. & Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 119-126.

Kamarainen, J., Kyrki, V. & Kalvianen, H. (2002). Fundamental frequency Gabor filters for object recognition. Inernational Conference on Pattern Recognition, 1, 628-631.

Kang, S. B., Webb, J., Zitnick, L. & Kanade, T. (1995). A multi- baseline stereo system with active illumination and realtime image acquisition. In Computer Vision, 1995. Proceedings., Fifth International Conference on, 88–93.

Kluckner, S., Mauthner, T., Roth, P. M. & Bischof, H. (2009). Semantic image classification using consistent regions and individual context. British Machine Vision Conference.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. in IJCAI, 14(2), 1137-1145.

Kong, W. K.., Zhang, D. & Li, W. (2003). Palmprint feature xtraction using 2-D Gabor filters. The Journal of the Pattern Recognition, 36, 2339-2347.

Ladicky, L., Russell, C., Kohli, P. & Torr, P. (2010). Graph cut based inference with co-occurence statistics. In Computer Vision–ECCV, 239-253.

Li, J. & Wang, J. Z. (2003). Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Transaction on PAMI, 25(9), 1075–1088.

Liu, C. (2009). Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. PhD thesis, Massachusetts Institute of Technology.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision , 60(2), 91-110.

Marques, O., Barenholtz, E., & Charvillat, V. (2011). Context modeling in computer vision: techniques, implications, and applications. Multimed Tools and Applications, 51(1), 303–339.

Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence., 27, 1615–1630.

Millet, C., Bloch, I., Hède, P. & Moëllic, P. A. (2005). Using relative spatial relationships to improve individual region recognition, The 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technologies, 119-126.

Peng, B., Zhang, L. & Zhang, D. (2013). A survey of graph theoretical approaches to image segmentation. Pattern Recognition, 46(3), 1020–1038.

Rellier, G., Descombes, X., Falzon, F. & Zerubia, J. (2004). Texture feature analysis using a Gauss-Markov model in hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 42, 1543-1551.

Russell, B. C., Torralba, A., Murphy, K. P. & Freeman, W. T. (2008). LabelMe: a database and web-based tool for image annotation, International Journal of Computer Vision, 77, 157-173.

Ruta, A., Li, Y. & Liu, X. (2010). Real-time traffic sign recognition from video by class-specific discriminative features. Pattern Recognition, 43(1), 416-430.

Sanberg, W.P., Do, L.Q., De With, P. H. N. (2013). Flexible multi-modal graph-based segmentation. Lecture Notes in Computer Science in Advanced Concepts for Intelligent Vision Systems , 8192, 492—503.

Scharstein, D. & Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In Proceedings of the Conf. on Computer Vision and Pattern Recognition, 1, I-195.

Shotton, J., Winn, J., Rother, C. & Criminisi, A. (2009a). TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. International Journal of Computer Vision, 81(1), 2-23.

Silberman, N., Hoiem, D., Kohli, P. & Fergus, R. (2012). Indoor segmentation and support inference from RGBD images. In Computer Vision–ECCV, 746-760.

Smith, J. R & Chang, S. (1995). Single color extraction and image query. IEEE International Conference on Image Processing, 3, 528–531.

Sun, J., Wu, X., Yan, S., Cheong, L. F., Chua, T.S. & Li, J. (2009). Hierarchical Spatio Temporal Context Modeling for Action Recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2004-2011.

Tamersoy, B. & Aggarwal, J. K. (2009). Robust vehicle detection for tracking in highway surveillance videos using unsupervised learning. In IEEE International Conference on Advanced Video and Signal Based Surveillance, 529–534.

Wang, Y. K. & Chen, S. (2005). A robust vehicle detection approach. In IEEE Conference on Advanced Video and Signal Based Surveillance,117–122.

Wei, H., Nguyen, H., Ramu, P., Raju, C., Liu, X. & Yadegar, J. (2009). Automated intelligent video surveillance system for ships. Proceedings of the Journal of Electronic Imaging, 7306, 73061N.

Wijnhoven, R., van Rens, K., Jaspers, E. G. T. & De With, P. H. N. (May 2010). Online learning for ship detection in maritime surveillance. Iin Proceedings of the 32rd WIC Symposium on Information Theory in the Benelux, 73-80.

Yu, H., Li, M., Zhang, H. J. & Feng. J. (2002). Color texture moments for content-based image retrieval. International Conference on Image Processing, 3, 929-932.

Zhang, H. (2010). Multiple moving objects detection and tracking based on optical flow in polar-log images. International Conference on Machine Learning and Cybernetics, 3, 1577-1582.

KEY TERMS AND DEFINITIONS

Image segmentation: is the process of partitioning a digital image into multiple segments

HSV color space: HSV is a transformation of an RGB color space, and its components and colorimetry are relative to the RGB colors pace from which it was derived.

Spatial information: Spatial Information describes the physical location of objects

Image Texture: gives us information about the spatial arrangement of color or intensities in an image or selected region of an image.

Contextual information: The information which is implicitly present in an input.

Salient region: Most noticeable or important region.

Co-occurrence: coincidence of two objects/regions in a scene.

Table 1

Region	Gravity-based model	GRS-based model	Bao *et al.*, (2012a)	Millet *et al.*, (2005)
Sky	96	96	95	94
Construction	88	88	87	84
Water	93	91	89	89
Road	92	92	92	89
Vegetation	90	85	85	87
Unknown	95	96	97	99
Average	93	91	91	90

Table 2

Test Videos	**Methods**	**TP+FN**	**TP+FP**	**TP**	**Precision (%)**	**Recall (%)**
S1	Improved method	1593	1496	1413	94.5	88.7
	Existing	1593	1491	1374	92.1	86.3
	Cabin detector	1593	2135	1389	65.1	87.2
S2	Improved method	455	433	422	97.5	92.7
	Existing	455	325	320	98.5	70.3
	Cabin detector	455	207	190	91.8	41.8
S3	Improved method	173	135	130	96.3	75.0
	Existing	173	130	122	93.8	71.8
	Cabin detector	173	115	97	84.3	56.1

Table 3

	# included categories	# included samples
Red circular restriction signs	34	30,756
Blue rectangular information signs	63	49,996

Table 4

	CE_{bal}	Processing time
Red circular restriction signs	2.00%	1.12 s
Blue rectangular information signs	3.75%	1.47 s