ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

CONTEXT AND KNOWLEDGE MANAGEMENT COMPONENT STATE OF THE ART AND FEASIBILITY ANALYSIS

DELIVERABLE D4.1.1

by
VTT

Due date of deliverable : t0+6

Actual submission date: t0+6

Version : 0.4	*State :* Proposal	*Dissemination level : PU/RE/CO*

DOCUMENT HISTORY

Version	Date	Comments	Author
0.1	06/04/11	Created as deliverable D4.1.1. Contains the Table of Contents and general document structure.	VTT Mikko Nieminen
0.2	13/05/11	First draft. State of the Art contributions included from ASELSAN, ENSTA, IEF and VTT. Initial feasibility analysis process defined along with IEF feasibility analysis results.	VTT Mikko Nieminen
0.3	26/05/11	Updated draft. State of the Art contributions from CogVis and C2Tech added.	VTT Mikko Nieminen
0.4	22/06/11	Proposal version. State of the Art contributions from EOLANE, Multitel and ACIC added, ENSTA contribution updated. Feasibility analysis results and conclusions added. Text for general sections (Scope, Abbreviations etc) updated. References combined.	VTT Mikko Nieminen
1.0	17/02/12	Cleaned up the finalized document and updated the version number into 1.0	VTT Mikko Nieminen

	Name and function	Date	Signature
Prepared by	Mikko Nieminen	22/06/11	
Reviewed by			
Approved by			
Authorized by			

CONTENTS

SUMMARY

This document contains the state of the art on context and knowledge management technology regarding modern networked surveillance systems. Also included are feasibility study results on the current state of the art, compared to the goals for context and knowledge management in the ITEA2 SPY project.

For the state of the art study regarding video and image analysis, we have looked at technologies and algorithms related to object detection and localization, feature extraction, motion detection, tracking and event detection. Furthermore, issues related to embedded video analysis have taken into consideration.

The state of the art is also studied from the perspective of context analysis, event recognition and decision making in distributed multi-sensor systems. Studied aspects consist of exploitation of multiple multi-modal sensors, information adaption, data fusion, distributed intelligence and the use of positioning in mobile surveillance.

In the feasibility analysis, we compare the studied state of the art into the requirements and software architecture plans for the SPY framework, taking in consideration technological, economical, operational and schedule constraints.

Feasibility analysis of different context and knowledge management aspects identifies some potential challenges, but also methods to avoid and overcome them. Overall, no serious feasibility issues have been discovered.

1. SCOPE

The SPY project aims at reaching advancements in developing a flexible and generic fusion and reasoning framework for supporting a scalable network of multiple sensors and sensor modalities. Further goals consist of building data fusing support for new event types generated by new and improved event detection algorithms, providing an open framework enabling integration and exploitation of data from legacy systems and detecting anomalies automatically and reliably in the stream of surveillance data.

The objective of this deliverable is to identify technology limitations and evaluate how to overcome these limitations to fulfill the SPY system requirements defined in WP3. Considering the system specification, feasibility of the technological, economical, operational and schedule constrains will also be studied. The state of the art will focus on video and surveillance applications.

2. ASSOCIATED DOCUMENTS

2.1 APPLICABLE DOCUMENTS

A1	Project Full Proposal. SPY
A2	SPY WP3 System Specification D3.1

2.2 REFERENCE DOCUMENTS

See Chapter 6 for references.

3. TERMINOLOGY

3.1 ABBREVIATIONS

COTS	Commercial Off The Shelf
EM	Expectation Minimization
EMC	Electromagnetic compatibility
ESD	Electrostatic Discharge
GPS	Global Positioning System
HoG	Histogram of Oriented Gradients
IIR	Infinite Impulse Response
ISODATA	Iterative Self-Organizing Data Analysis
JDL	Joint Directors of Laboratories
LBP	Local Binary Patterns
MID	Mobile Internet Device
MRF	Markov random Fields
MSER	Maximally Stable Extremal Regions
N/A	Non Applicable
OGC	Open Geospatial Consortium
PCA	Principal Component Analysis
PTZ	Pan Tilt Zoom
RAG	Region Adjacency Graph
RAM	Random Access Memory
RANSAC	RANdom SAmple Consensus
SDK	Software Development Kit
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
SUSAN	Smallest Univalue Segment Assimilating Nucleus
TBC	To Be Completed

3.2 DEFINITIONS

N/A

4. CONTEXT AND KNOWLEDGE MANAGEMENT STATE OF THE ART

This chapter details the State of the Art of context and knowledge management in modern distributed surveillance systems. Regarding the exploitation of video in the surveillance context, we present the state of the art on video analysis, computer vision and image processing. Furthermore, issues regarding automated and distributed intelligence in networked surveillance systems, including the exploitation of multiple and multi-modal sensors, are explored.

4.1 OBJECT DETECTION AND LOCALISATION BY VIDEO ANALYSIS (ENSTA)

4.1.1 Scope and General Architecture

4.1.1.1 Introduction and Scope

This chapter is dedicated to the problem of finding objects of interest in a video. “Object” is understood in its familiar (i.e. semantic) sense: e.g. car, tree, human, road… and the system is supposed to automatically find the location of such objects in the captured video. To be consistent with the project technological level, we shall exclude the “developmental” approaches, where the system does not know the objects in advance, and constructs incrementally its own internal representation. We then suppose that the system operates with a provided representation of the objects and its environment that has been constructed (learned) off-line, and that may evolve on-line. Such representation includes a set of object classes that the system is then expected to recognize and localise in every image, either by attributing accordingly a label to every location in the image (task referred to as “semantic segmentation”), or by localising – more or less precisely – instances of each class in the video and tagging every image accordingly (referred to as “semantic indexing”).

In this chapter we present a state-of-the-art of the video analysis methods for object and environment modelling and semantic indexing or segmentation with respect to the corresponding model. Being one of the Grail quests of computer vision for a long time, object detection has motivated a huge literature, and this state-of-the-art is by no mean exhaustive; our objective is rather to construct a representative survey of applicable methods, according to the following arguments:

- Every chosen technique should be known to be successful enough and well referenced.
- The presented techniques should differ fundamentally enough to cover the largest range of methodologies.
- The technique should be applicable in the context of outdoor sequences acquired from in-car camera in urban or peri-urban scenarios.
- The technique should be reasonably adaptable to an embedded implementation, or at least provide hints for the reduction of the computational cost.

We first present the general architecture of video based object detection systems, identifying the fundamental tasks or parts that are present in most systems, or should be present in our particular context. Then, in the following sections, we present and discuss some significant instances of each one of these parts that can be found in the literature.

4.1.1.2 General Architecture

Figure 1 shows the generic software architecture of a vision based object detection and localization system. According to the previously stated restrictions, the system is separated in a "learning" phase (dashed-line box), which is previously computed off-line, and whose function is to construct the “world (objects, background, context...) model” from a series of training examples, and in an "operating" phase (plain line box), which embeds the world model and performs on-line the task of object detection and localization. The description in the figure emphasises some fundamental parts which are all present in most techniques given in reference, but under many different forms:

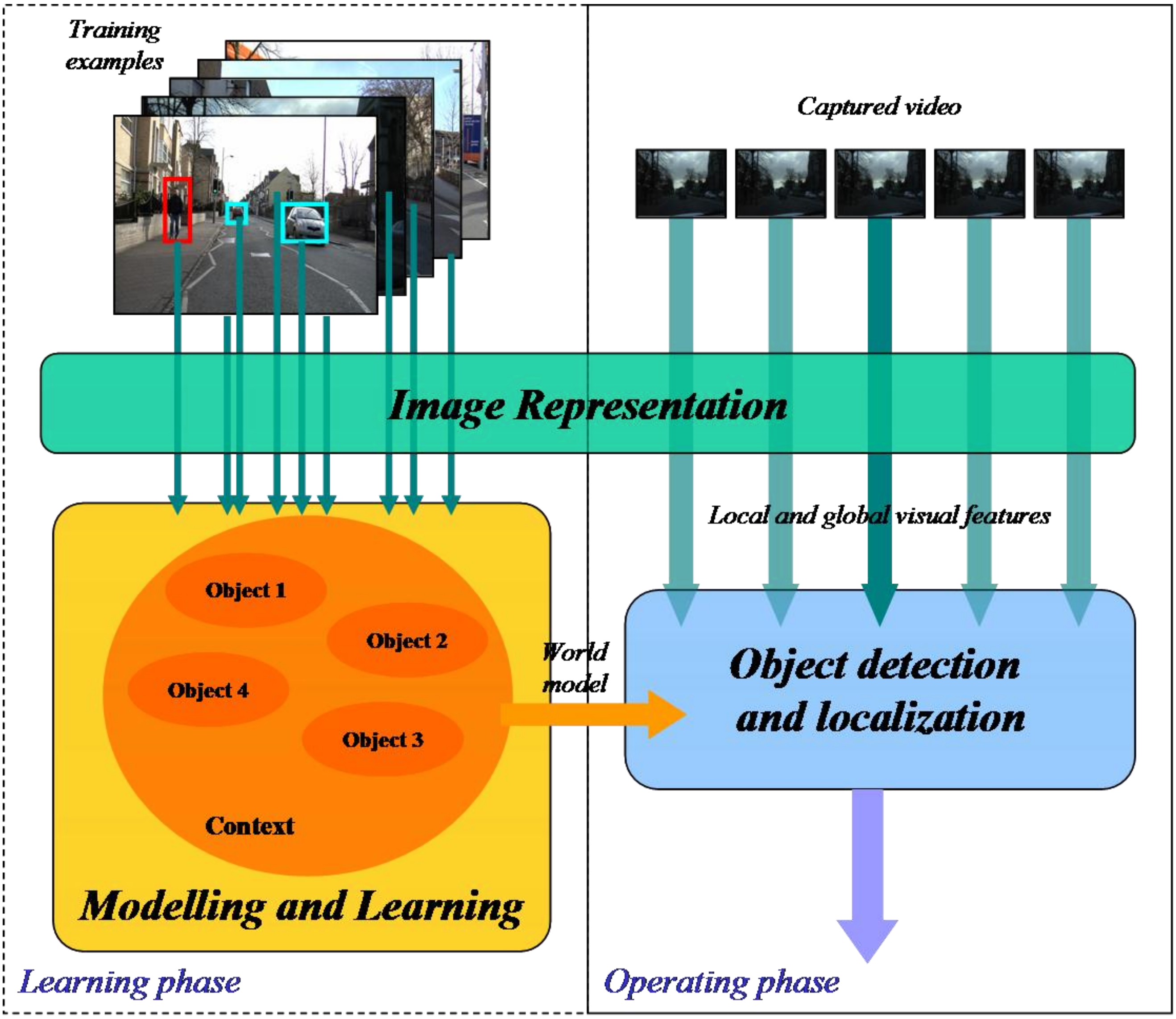


Figure 1: Object detection methods general overview

- **Visual representation**: this part refers to the extraction of visual information from images. Some filters are applied to select local structures (colour, region, direction, frequency...) in images. The produced information can be summarised using statistics and clustering, and/or used to reduce the computation domain to a small number of significant points. The corresponding tasks are applied both during the learning and the operating phases, but not as intensively.
- **Modelling and Learning**: this part is the creation of the world model from the series of training examples. It corresponds to the process of selecting the most relevant visual features and/or automatically finding the parameters of the classification mechanism that will be applied in the operating phase.
- **Context representation and modelling**: this part is not a module in itself, as it is usually performed by tasks from the two previous items. It refers to mechanisms using global

description of the image or the video to improve the detection of objects by contextual considerations (e.g. a car is more likely to appear on the road than in the sky).

- **Object detection and localisation**: this distinctive part of the operating phase refers to the task of finding instances of objects from known classes in every image of the video. As this will be the real-time part of the system, we will particularly examine the data exploration and prediction strategies able to reduce the computational cost.

The following sections now present the most significant existing work according to the previous organization.

4.1.2 Object Visual Representation

In this section we examine the image processing operations performed to extract meaningful information from the images, and the way this information is reduced or coded to provide a useful representation of the objects and their environment.

4.1.2.1 Filter Banks and Local Structures

The initial information available in every pixel, say colour or grey level, is very sensitive to small changes or distortions, and then unreliable for direct representation purposes. Thus, the first level of processing is enriching the local information by computing measures relative to the local appearance of pixels.

Those measures are generally multiple and obtained through a bank of filters, usually a set of convolution kernels whose aim is to quantify the local geometry of pixels, regarding: orientation, curvature, scale and frequency.

The local jet, defined as the set of partial derivatives calculated at every image location, is a fundamental feature space [1], that can be used to construct many geometrical invariants [2]. In the scale-space framework, the partial derivatives are estimated at a given scale, which is done by convolving the image with the corresponding derivative of the 2d Gaussian function G_σ, whose standard deviation σ corresponds to the estimation scale:

$$I_{x^i y^j} = I * \frac{\partial^{i+j} G_\sigma}{\partial x^i \partial y^j}$$

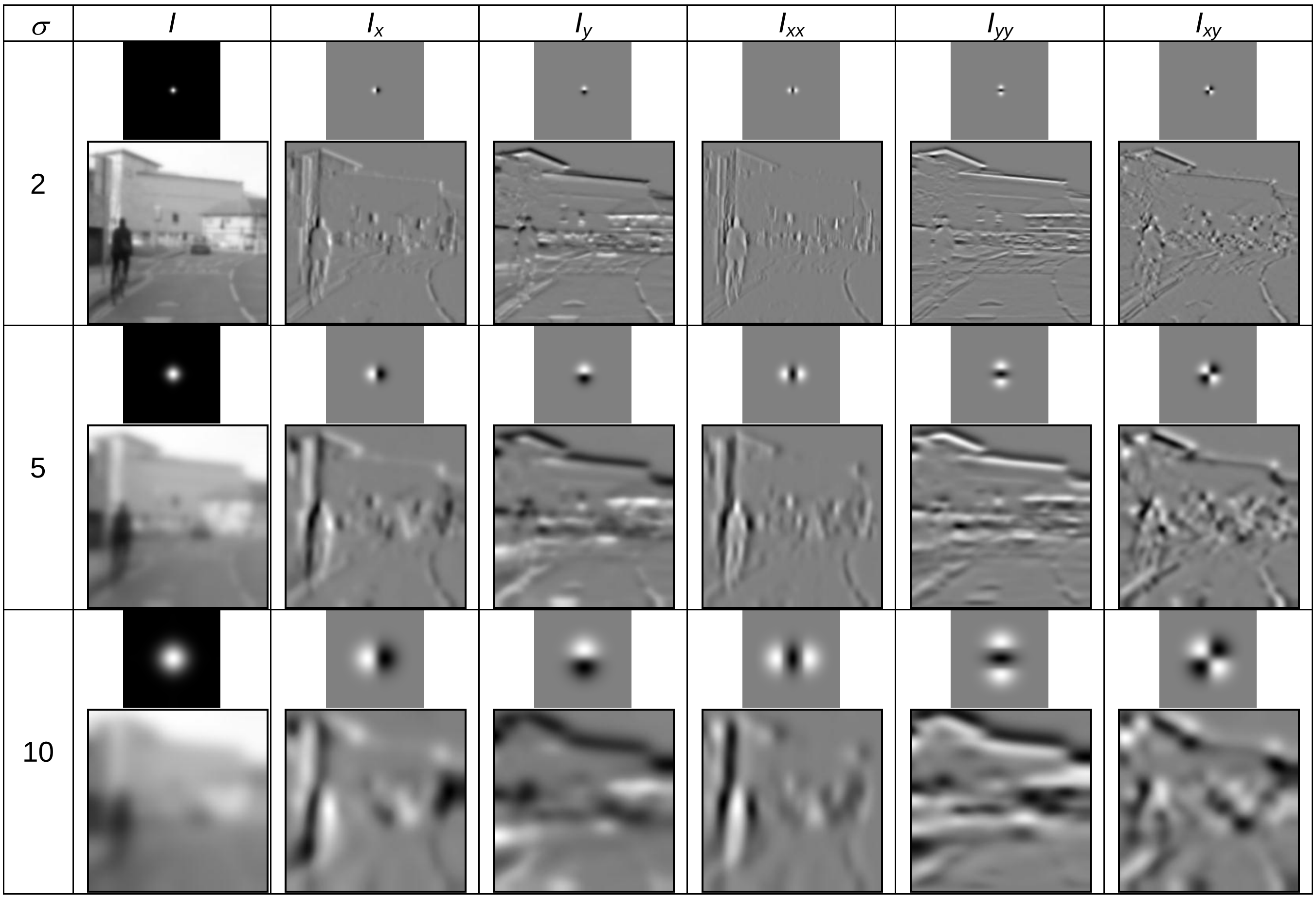


Figure 2: Three scale (normalised) local jet of order 3. The derivative estimation is obtained by convolution with the corresponding kernel appearing on the top.

Figure 2 shows an example of the multiscale local jet and the corresponding filter bank of Gaussian derivatives.

The local jet is one of the most generic and versatile local description spaces. It includes - or can be reduced to - many useful invariant features, such that the orientation of the gradient or orientation of the principal curvature, and it is used to compute another fundamental invariant: the image contours, usually defined as the local maxima of the gradient intensity in the gradient direction.

Figure 3 shows examples of such invariant features. The orientation of the gradients [3] and the contours [4] have been used in several real-time object detection methods.

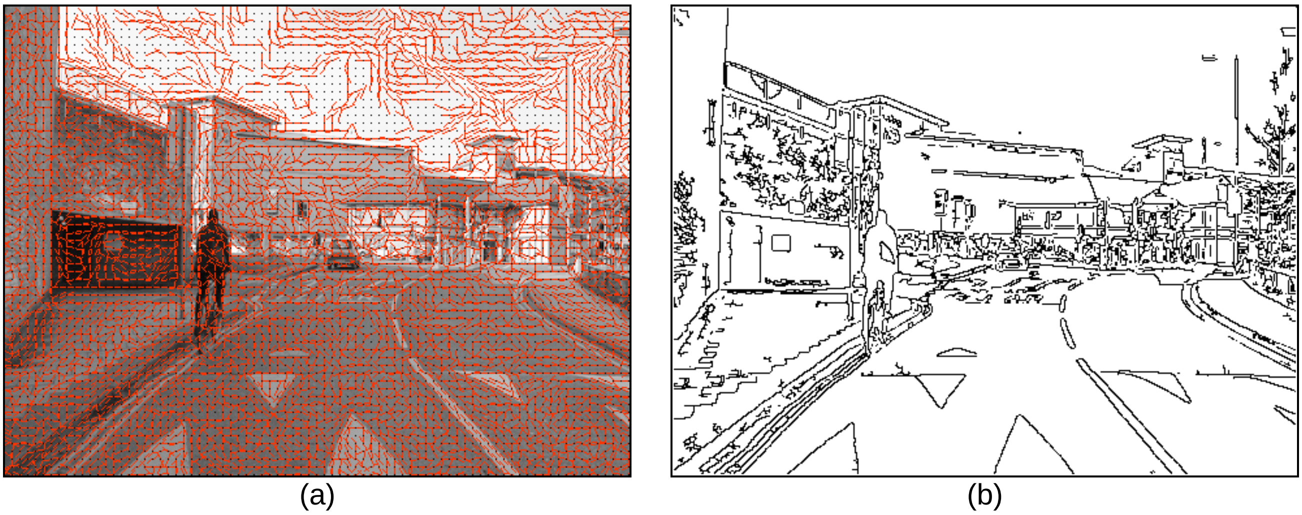


Figure 3: Two examples of contrast invariant features (a) Direction of the isophote (orthogonal to gradient), and (b) Contours defined as the local maxima of the gradient intensity in the gradient direction.

Another important filter bank is the Gabor filter collection, whose simplified real expression can be defined as:

$$H_{(\sigma,\theta,\omega)} = G_\sigma \cos(2\pi\omega(x\cos\theta + y\sin\theta))$$

Where σ corresponds to the spatial extent, θ the orientation angle, and ω the frequency of the filter. In practice the spatial extent is often coupled with the frequency, and then a Gabor filter finally detects the presence of rectilinear periodic structures at a certain frequency and orientation. Gabor filters are acknowledged as a good model of one fundamental early visual function of the mammalians and then has been used in many object modelling and detection systems [5]. Figure 4 shows an example of direction/frequency decomposition using a bank of Gabor filters.

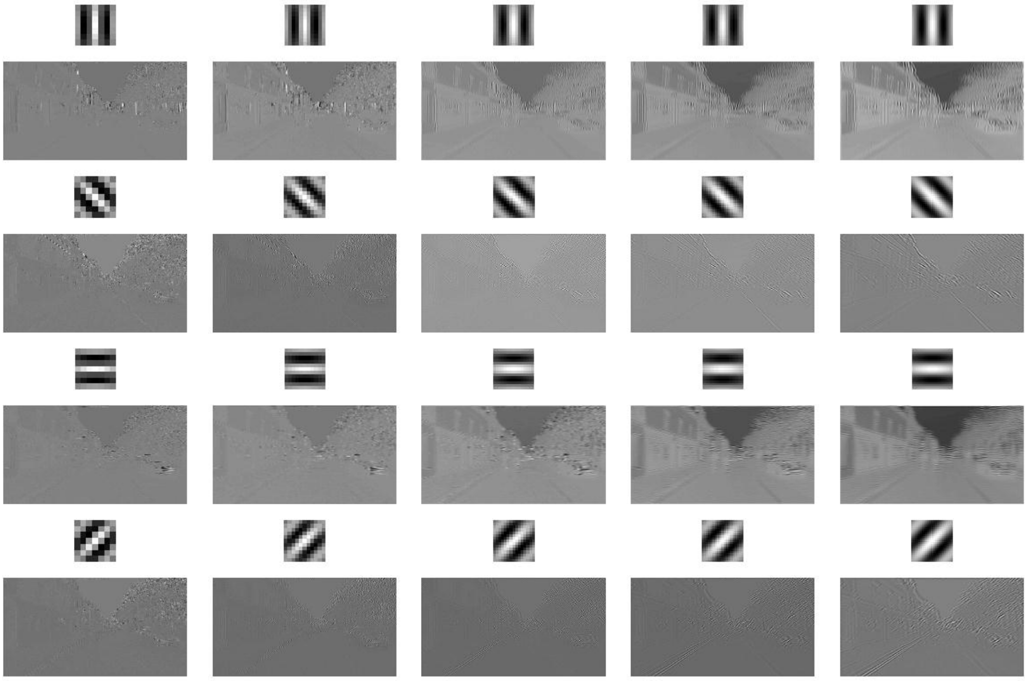


Figure 4: Local response to five scales and four orientations using a bank of Gabor filters.

Many real-time methods are based on collections of Haar filters [6], sort of approximations of derivative convolution kernels, which are particularly attractive for being computed very fast thanks using integral images. A Haar filter can be defined as a convolution kernel with rectangular support, with values only equal to -1 or +1. The number of operations needed to compute these filters do not depend on the size of the support, but on the number of rectangles with the same value inside the support. Figure 5 shows a few examples of Haar filters and their corresponding output.

Some methods radically differ from the previously cited ones in the sense that they begin by aggregating pixels in small homogeneous regions called “superpixels”, that will be used later as more reliable (and less numerous) individuals than pixels to extract relevant descriptors from images [7], [8]. In this case, the lowest level operator is a segmentation algorithm, like the morphological watershed, which is fast enough and allows easy tuning of the size and relative contrast of the superpixels. Figure 6 shows examples of multi-level watershed superpixel segmentation. Like the contours, the superpixel approach can be better adapted than filter banks methods to the case of poorly textured objects.

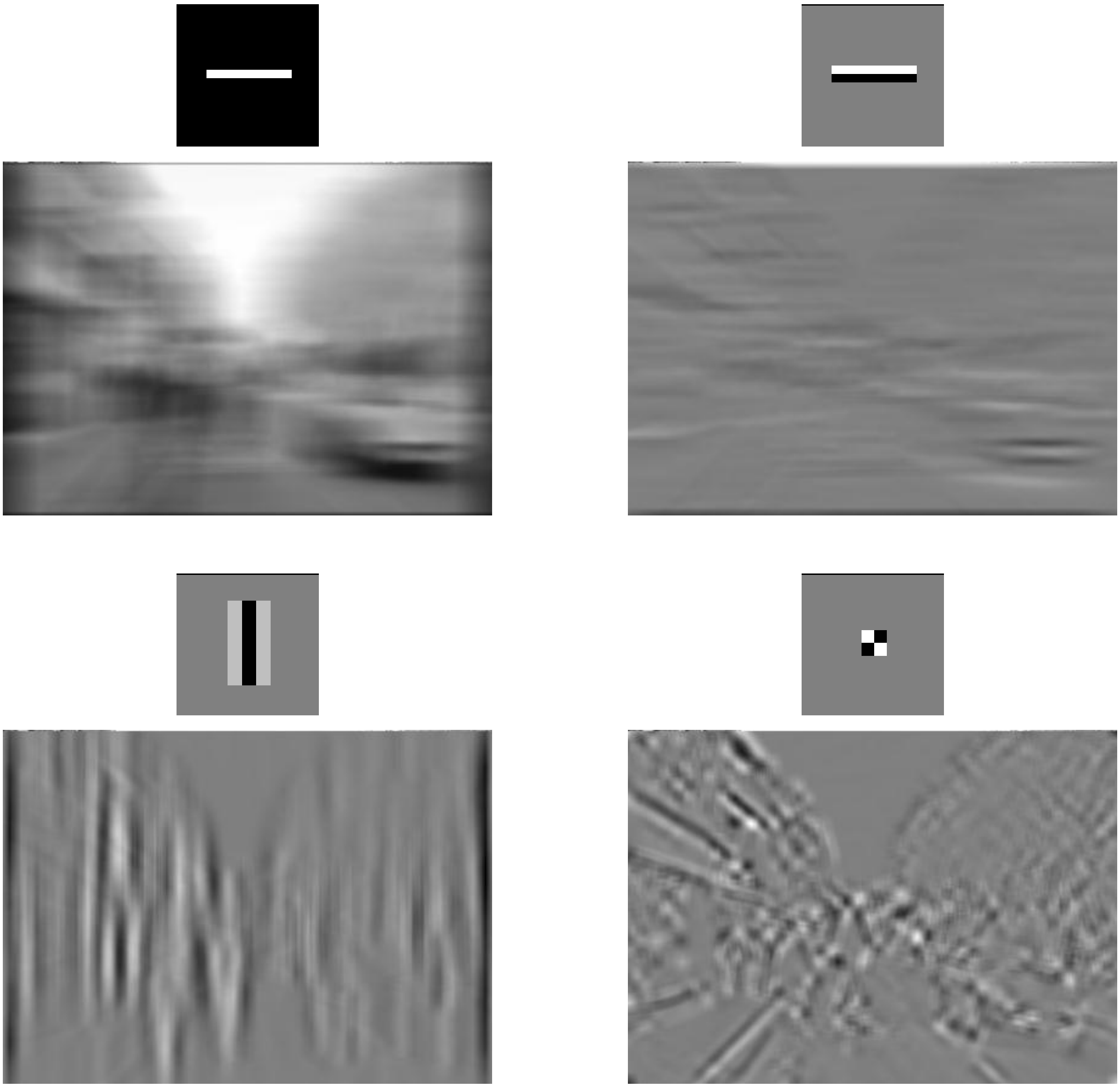


Figure 5: Four examples of Haar filters approximating multiscale derivatives

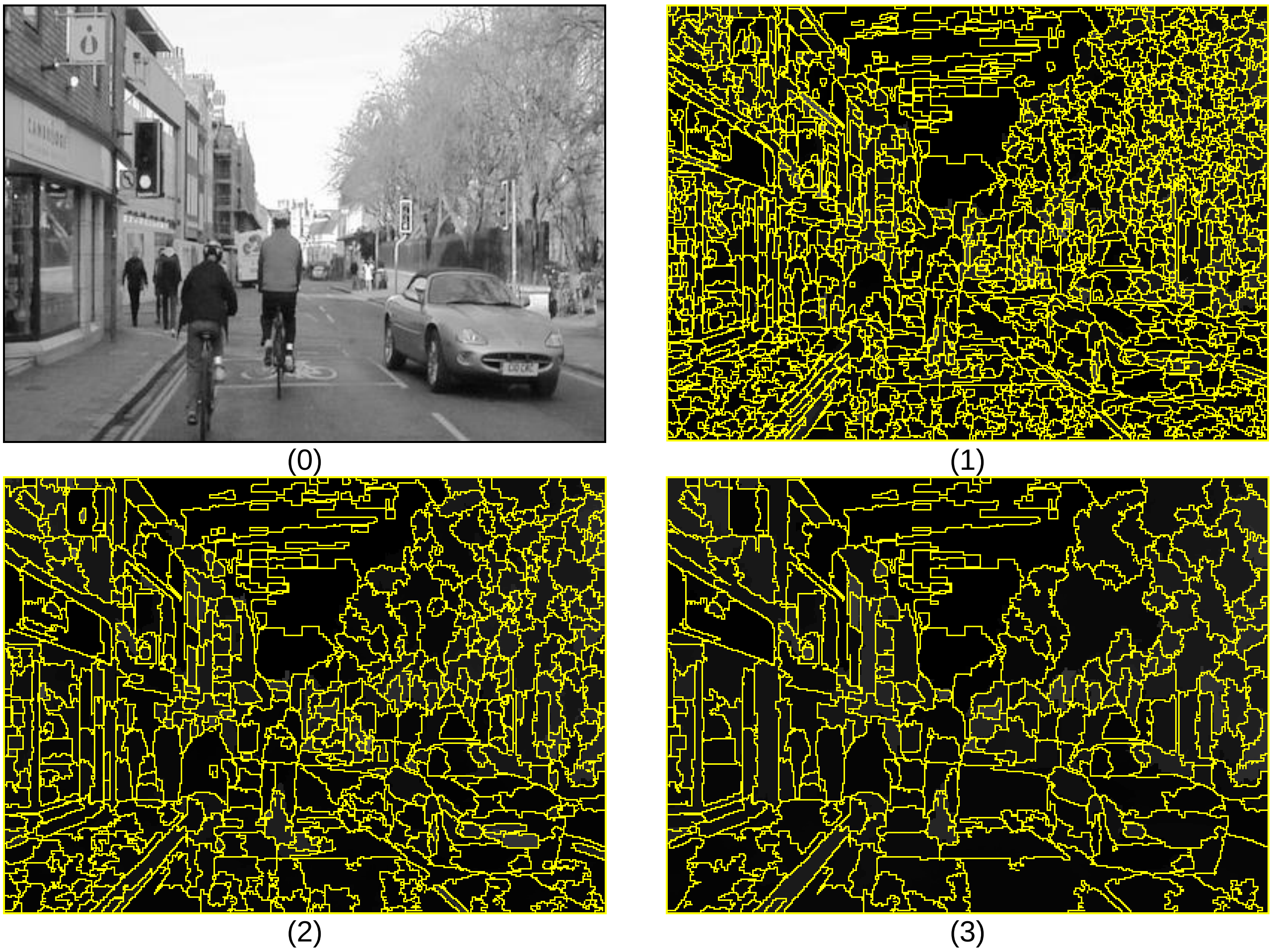


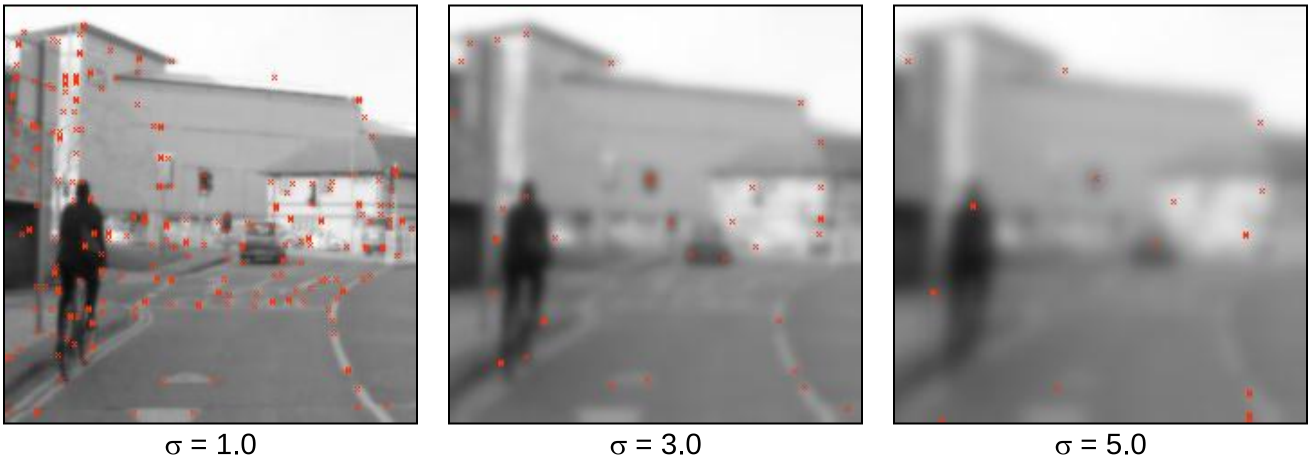
Figure 6: An image (0) decomposed in superpixels by watershed (over)-segmentation of the morphological gradient image, with area closing of different sizes (1-3).

4.1.2.2 Salient Point and Regions

One important question rising in any method is whether the visual descriptors may be extracted from everywhere in the image or only from a few points or regions previously selected in the image. This can be interesting both to reduce the data flow and to improve the robustness (by selecting the most significant and stable structures), but it needs extra processing to perform the detection of salient structures. This detection generally uses a combination of filters and local detectors presented in the previous sub-section. A brief presentation of some significant detectors follows:

- The multiscale Harris detector [9] outputs the local maxima of an interest function computed from the autocorrelation matrix, estimated at various scales. It corresponds to corner points and it is rotation invariant (see Figure 7).
- The SIFT points [10] are the local extrema, both in space and scale, of differences of Gaussian filters. They correspond to peak and valleys disappearing during a progressive smoothing of the image. Every point is associated to a particular scale and orientation (see Figure 8)

- The SURF detector [11] is a much faster multiscale detector which uses certain collection of Haar filters to approximate the multiscale second derivatives. The SURF salient points are then defined as the local maxima of the determinant of the Hessian matrix.
- The MSER detector [12] uses a segmentation approach and selects superpixels with certain invariance and stability properties, corresponding to regional extrema of certain size, which are stable to perspective transformations.



σ = 1.0

σ = 3.0

σ = 5.0

Figure 7: The Harris salient points (red crosses), calculated at three different scales.

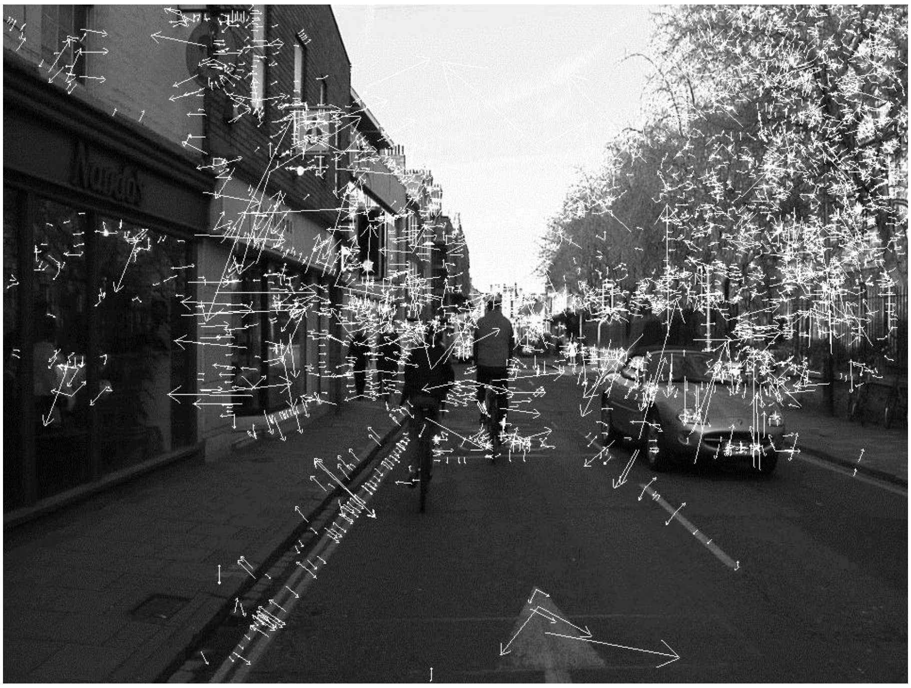


Figure 8: The SIFT salient points. The salient point is located at the origin of the arrow; the length of the arrow represents the scale, its direction the argument of the gradient.

4.1.2.3 Feature Descriptors and Statistics

In this sub-section, we discuss the last part of visual representation, i.e. how the visual information is finally represented in the world model. The challenge of designing good descriptors is to find both concise and rich models, to be compared with the unknown objects from the video in an efficient and relevant way. The descriptor must capture essential structure, discard irrelevant details and lend itself to efficient metrics for comparison purposes.

When the image support is reduced to salient structures, the representation can be simply made by the collection of feature vectors corresponding to local structures calculated at every interest point location. For example, in [2], the descriptor attached to every Harris point (which is attached to a specific scale) is a vector made of Hilbert invariants (combinations of the local jet components with rotation invariance), calculated at the corresponding scale. This local approach has several advantages, like some robustness to deformation and occlusion, but it is very sensitive to the quantity and quality of detected salient points.

One common problem in designing descriptors is to find a good trade-off between local and global representation. Many approaches address this problem by computing spatial statistics or estimating regional tendency of a local measure. One of the most successful examples is the Histogram of Gradient orientations (HoG) and its variants. The descriptors usually attached to the SIFT [10] or SURF [11] salient points belong to this category. Every SIFT or SURF point is attached to the specific scale where it has been detected. It also comes with a specific orientation corresponding with the argument of the gradient calculated at the location point and the selected scale. Now the descriptors principle is to calculate the orientation of the gradients for every pixel around the salient point and to calculate the histogram of these orientations for one (or more) windows surrounding the salient point. The number of orientations is quantized to reduce the size of the descriptor, and the occurrence of every orientation is weighted in the histogram by (1) the distance of the pixel w.r.t. the salient point, and (2) the intensity of the gradient. Figure 9 and Figure 10 illustrate the actual descriptors proposed in the original articles.

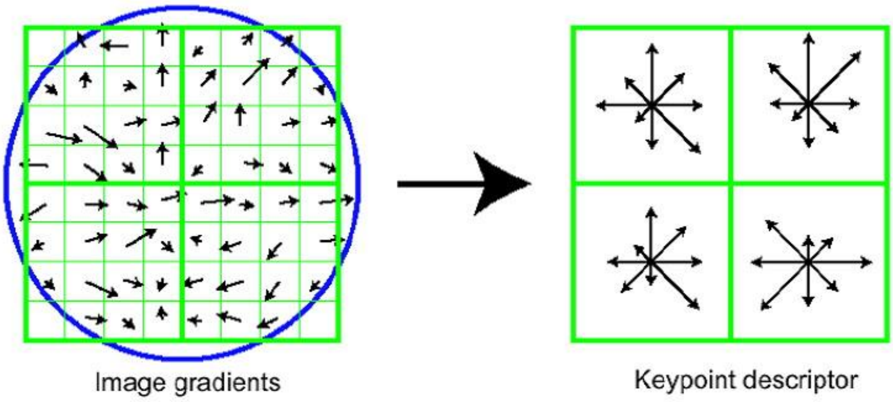


Figure 9: SIFT descriptor, from [10]. The image is split in small blocks around the salient point, and the histogram of weighted and quantized orientation is calculated for each block. Dominant orientations correspond to the arrows of highest lengths in the descriptor.

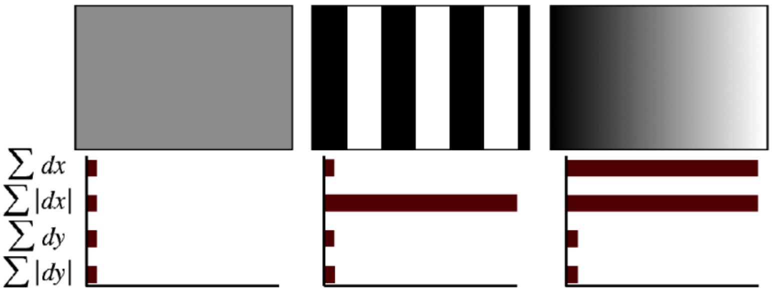


Figure 10: One example of SURF descriptor, from [11]. The sum of some (absolute or signed) partial derivatives is recorded for small sub-regions around the salient point in the image.

Histograms of orientations can also be used more densely, i.e. not only around salient points but in the whole image by dividing the image in (overlapping) blocks. In this case it is essential to use a dedicated orientation bin as “void” to code the homogeneous area without significant gradient. This is done in particular by [3], who adopt a more radical approach (motivated by getting a very small code for real-time purposes), consisting in retaining only the dominant orientations present in a given block instead of a full histogram, turning the visual model into a simple binary code (see Figure 11).

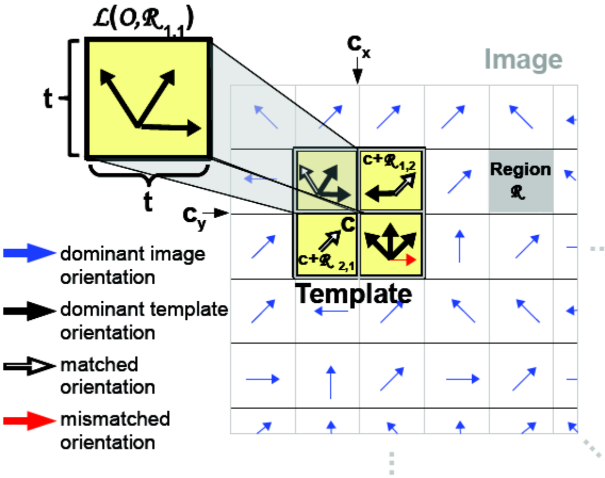


Figure 61: Dominant orientations descriptor from [3]. The model records a subset of the (quantized) dominant orientations present in a small image block. To measure the (binary) matching with the model (Figure), only one dominant orientation is calculated by block (blue arrows).

One very popular class of methods in object modelling is the Bag-of-Features approach, which generalizes the concept of texton that has been used in texture recognition. The principle is based on the quantization of the n-dimensional descriptor space, which is then reduced to a codebook of N visual words (so called in analogy with the bag-of-words classification methods in linguistics). A visual class, object or image can then be represented by a histogram of visual words. A fundamental characteristic of these approaches is to mostly ignore the geometry of objects by considering only the presence of visual structures and not the relations between them. However, taking into account more geometry can be done by considering higher order statistics, like co-occurrence between visual words.

Bag-of-features (first order) classical approaches are known to work well for global image classification, e.g. place recognition, example based image retrieval, image categorization. But they are not designed to perform localization of the object in the image. To address this issue, several authors have proposed visual codebooks including information of relative localization. For example, [13] use a codebook made of triplets (filter, patch, position). Every triplet is associated to a selected (learned) point and includes: (1) a filter from a filter bank, (2) the resulting patch containing the local output of the filter applied at the point location, and (3) the relative location of the point with respect to the object (See Figure 12). In [14], every code word or texton *t* is paired with a rectangular mask *R* made of an origin point and a rectangle. The matching measure at location *x* with the corresponding feature is obtained by counting the number of pixels of texton *t* within rectangle *R* when its origin is in x (See Figure 13).

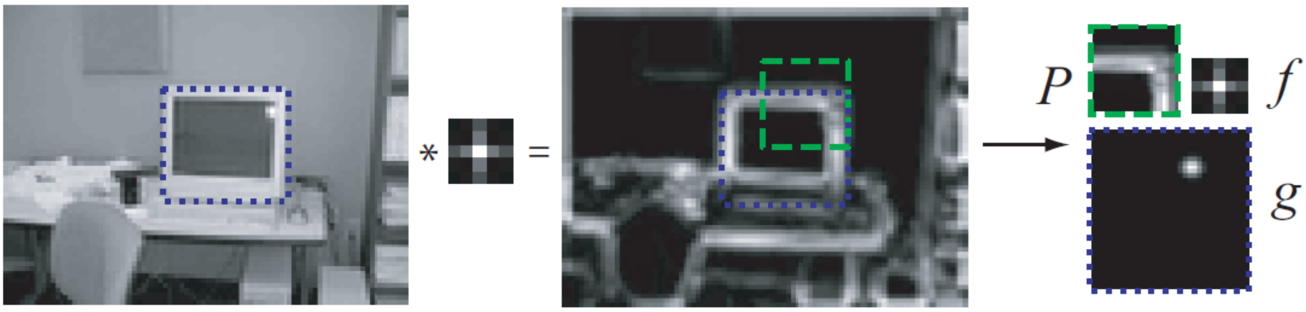
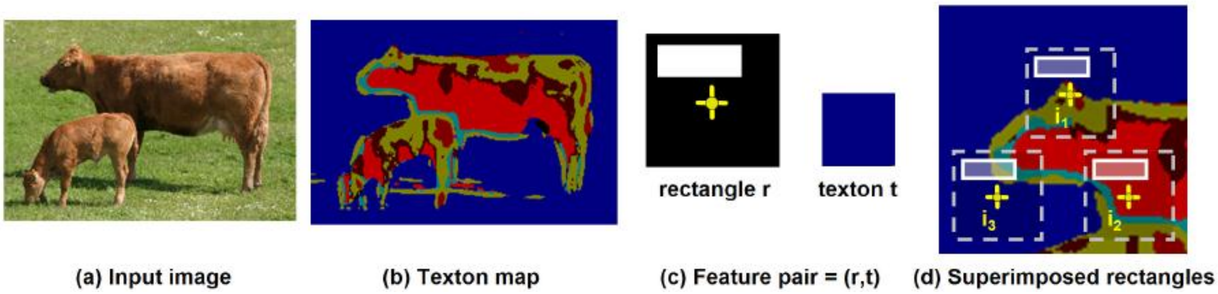


Figure 12: Feature triplet from [13], combining a filter *f*, a response patch *P*, and a relative localisation (blurred Dirac) *g*.



(a) Input image (b) Texton map (c) Feature pair = (r,t) (d) Superimposed rectangles

Figure 13: Texture shape by pairing texton index and relative location mask, according to [14]

4.1.3 Learning the Object Model

In this section we describe the techniques used to construct the world model from instances of objects captured off-line.

4.1.3.1 Selection of the Object Prototypes

In the methods based on codebooks, object prototypes are constructed within a process of statistical reduction of the representation space. In this case the learning algorithm consists in summarizing the set of descriptors extracted from all the instances of objects captured in the learning base, to a (much smaller) set of significant vectors. The selection is usually performed by a vector clustering method such as K-means, but is sometimes done at random. Figure 14 shows some examples of the selected triplet prototypes used in [13] to represent an object class.

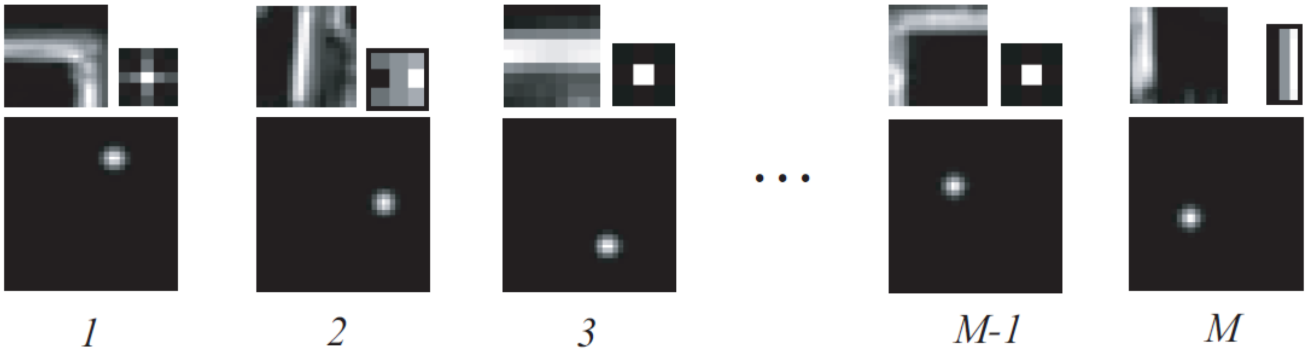


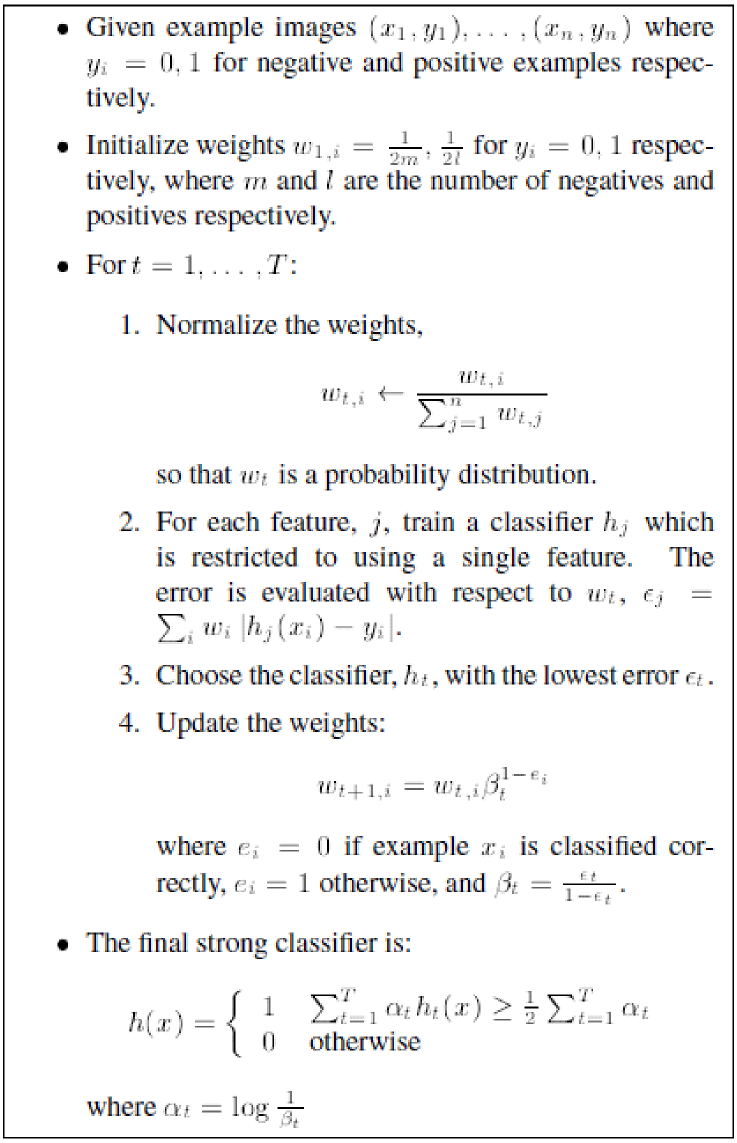
Figure 14: Examples of the *M* triplet prototypes selected by vector quantization to represent the "Screen" object class in [13] (see also Figure).

In some cases the data reduction is done by reducing the dimensionality of the descriptor space, e.g. using principal component analysis or non-negative matrix factorization. In this case the object is not represented by prototypes, but by a small number of vectors from a new algebraic base representing the main directions of variation of the descriptor space.

4.1.3.2 Feature Selection and Learning

As previously explained, several techniques are based on the calculation of a large – potentially huge – number of operators from a bank of filters. However, only a small proportion of them are really significant for the detection of a certain class of objects. Automatically selecting the basic operators, and/or combining them to construct more sophisticated local detectors, is at the core of various object learning approaches.

One particularly successful technique is the Adaboost method applied for the selection and combination of the Haar filters [6]. Although extremely fast to compute, the collection of all Haar filters computable within a region of reasonable size is much too high to be computable at detection time. The Adaboost algorithm learns the detection operators associated to a given class using a set of positive and negative example images. Every example is attributed a weight defining its influence in the learning process, the weights being uniformly distributed at the beginning. Then, the algorithm iteratively selects the best "weak classifier" defined as the single Haar filter that best separate the positive and negative examples by simple threshold of its output. The best weak classifier is the one that minimizes the error measure obtained by summing all the weights of the failed examples. The weight of every example is then updated in such a way that the influence of the badly classified example increases in the selection of the next weak classifier, and so on. Finally a "strong classifier" is constructed using a linear combination of the weak classifiers, each weak classifier being weighted according to its single performance. See Figure 15 for the detailed algorithm.



- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,

     $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

     so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:

     $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

Figure 15: Adaboost algorithm, taken from [6], for training one strong classifier using *T* weak classifiers. Every weak classifier is of the form $h_t(x) = 1 \iff \pm f_t(x) < \pm\theta_t$, where f_t is the output of one single Haar filter.

It is worth noting that Adaboost is an instance of more general boosting meta-algorithms, whose purpose is to automatically construct sophisticated classifiers from a set of simple (weak) binary classifiers. Boosting algorithms have been frequently used under different forms in object modelling and detection.

4.1.3.3 Template and Shape Hierarchies

As seen before, the most computationally efficient methods generally rely on rough descriptors (contours, dominant orientation), which are seriously affected by the shape variability in terms of size and point of view that can occur in the applications. One common way to address this problem is to construct, during the training phase, a higher level representation of the object combining different instances of descriptors of the same object appearing at different views. Those different instances should be organized in such a way to allow efficient matching during the detection phase. It generally corresponds to a hierarchical representation.

For example, in [3], the authors starts with a collection of template vectors corresponding to block-wise dominant orientations captured from different views of a learned new object. The templates are grouped in clusters of similar templates, and every cluster can be assigned a generic descriptor vector (by simple OR operations), so that the template matching can be done efficiently using branch and bound in the detection phase.

In [4], a coarse-to-fine contour hierarchy is constructed as a tree, every level of the tree being associated to a given resolution (the root represents the coarser resolution). At each level, the set of template contours at a given resolution is grouped in a few clusters; every cluster is associated to a node, and represented by a prototype, which is (ideally) the median contour of the cluster, i.e. the template contour minimizing its mean distance to other templates in the cluster. Clustering and choosing the prototypes is performed by minimizing an objective function using simulated annealing.

4.1.3.4 Training Markov Superpixel Fields

In the case of semantic segmentation, labelling is densely performed at the superpixel (region) level, and then the knowledge acquired from the training examples must be integrated in the priors of the superpixel classification method. Those methods are naturally well adapted to Markov Random Fields (MRF) based classification, where the topology of the MRF is given by the Region Adjacency Graph (RAG) of the segmentation, and potential functions defined on nodes and edges of the RAG are used as probabilistic modelling of the labelling decision, ruling (among others): the relation between a superpixel descriptor and its label, or the dependence between the labels of adjacent superpixels.

In this case, the learning phase corresponds to the training of the MRF and the construction of the potential functions. For example, in [8], the potentials attached to higher order cliques are designed from the co-occurrence statistics of labels appearing in adjacent superpixels, obtained from the hand segmented training examples.

4.1.4 Context Representation

In many applications of object detection, taking the context into account in the modelling is very valuable, both in terms of robustness and computational efficiency. This is particularly true in the SPY project, where the camera will be embedded in a car, probably with a constant field of view. We are then dealing with street or road scenes, whose variability in terms of illumination and presence of objects can be large, but whose geometry and expected background (i.e. road, sky, building…) can be roughly predicted in a certain measure.

Such context representation involves global features of the image characterising the scene as a whole, but also some relations between the individual object models, in terms of temporal co-occurrence or spatial organization…

4.1.4.1 Context Modelling in Semantic Segmentation

The semantic segmentation methods necessarily include context modelling, because their purpose is to label every location (pixel or superpixel) of the scene, whether it belongs to an interest object or not. It is then natural to differentiate various background labels: sky, vegetation, road, etc. that turn out useful as context hints for the detection of the interest objects. As said earlier, those methods [7], [8] use MRF formulations modelling conditional probabilities linking the values of labelling function $\lambda : S \longrightarrow C$, where *S* is the set of superpixels, and *C* the set of label classes. Let $V \subset S \times S$ be the set of adjacent pairs of superpixels, such that (*S,V*) is the graph of adjacency of the segmentation. The semantic segmentation principle is to minimise an energy function *E* linking λ, *S* and *V*, for example:

$$E(\lambda) = \alpha \sum_{s \in S} -\log(P(s^{app} / \lambda(s))) + \beta \sum_{(s,s') \in V} -\log(P(\lambda(s), \lambda(s'))) + \gamma \sum_{s \in S} -\log(P(s^{loc} / \lambda(s)))$$

The first term relates to the visual appearance of superpixel *s*: s^{app} is the value of the visual descriptor calculated at s. The second term relates to the probability of co-occurrence of a couple of labels on adjacent pixels. This term can model the regularity (by attributing a higher probability to pairs with the same label) but also more contextual relationships like the likeliness of two labels to appear side by side, that can be learned from co-occurrence statistics (see Sec. 4.1.3.4). Finally the third term refers to the global context of the scene: s^{loc} is the location (coordinates) of the superpixel, and the likeliness of a given label to appear at a given location in the image can also be learned from the training images. Figure 16 shows an example of the conditional probability fields $P(s^{loc} / \lambda(s))$ learned from image bases for different labels. It is clear that those measures make sense and can be used in other framework than MRFs.

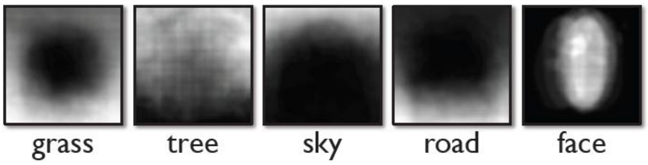


Figure 16: Location conditional probability fields associated to different labels, used as contextual hints, taken from [7]

4.1.4.2 Scene Descriptors

Sometimes the mere location hint discussed in the previous paragraph is not usable because the scene presents too much variability. This can occur in our application if the visual environment of the vehicle changes significantly, for example from a commercial urban street to a bare countryside road. In these cases, the “local” background must be estimated in order to exploit contextual information. This can be done by using global scene descriptors, calculated at run-time to improve the object detection and localisation.

Such descriptor is used by [13], thanks to a global feature called “gist”. The gist of an image is obtained by (1) applying a bank of filters (e.g. Gabor) on the image to compute local responses in scale and orientation, (2) cutting the image in small blocks and calculating the average response for each scale and orientation and for every block, thus obtaining a vector of dimension $n \times m \times p$, where *n* is the number of scales, *m* the number of orientations and *p* the number of blocks, (3) finally the dimension of the descriptor is reduced using Principal Component Analysis (PCA). Figure 17 illustrates the way the gist captures the global organisation of the textural features for two different scenes.

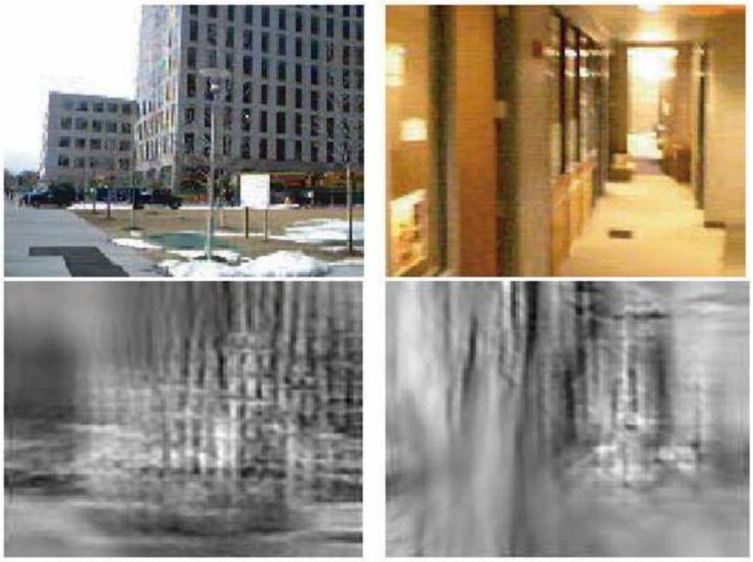


Figure 17: Illustration of the gist calculated for two images (top), taken from [13]. The bottom line shows two synthesis images with the same gist as the image above (obtained by iteratively modifying a random image)

The gist is then used as a location prior in a similar manner as the previous subsection, except that the conditional probability for a label does not depend only of the coordinates of the pixel (or superpixel), but on the value of the gist at this location. Figure 18 shows an example where the value of the pixel is multiplied by the conditional probability for four different object labels. Now the conditional density of every label with respect to the gist needs to be learned with the other parameters of the world model, which is done in [13] using multinomial regression and expectation minimisation (EM) algorithm.

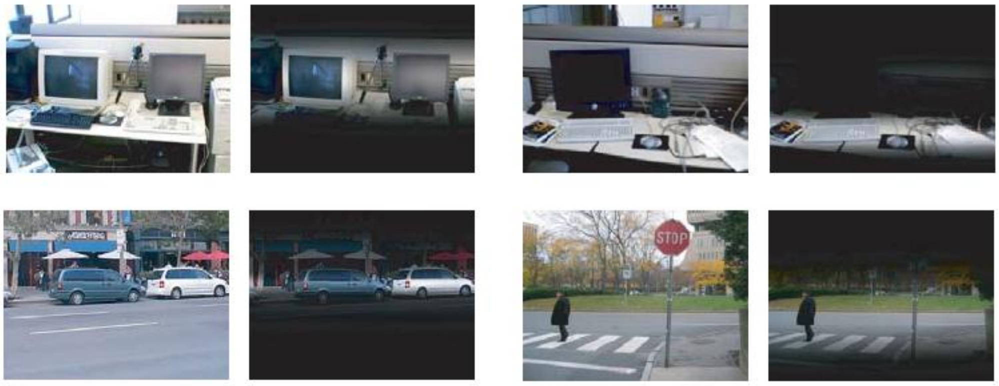


Figure 18: Using the gist for location priming of four different object classes (screen, keyboard, car, pedestrian). Taken from [13].

4.1.5 Detection and Localisation

This section is dedicated to the online part of object detection systems, related to the tasks of detecting the presence of a known object and localise it in the image at run time. The evaluation of the computation cost is critical at this level, so we will particularly pay attention to the strategies that have been proposed to make this task as efficient as possible.

4.1.5.1 Model Matching Methods and Measures

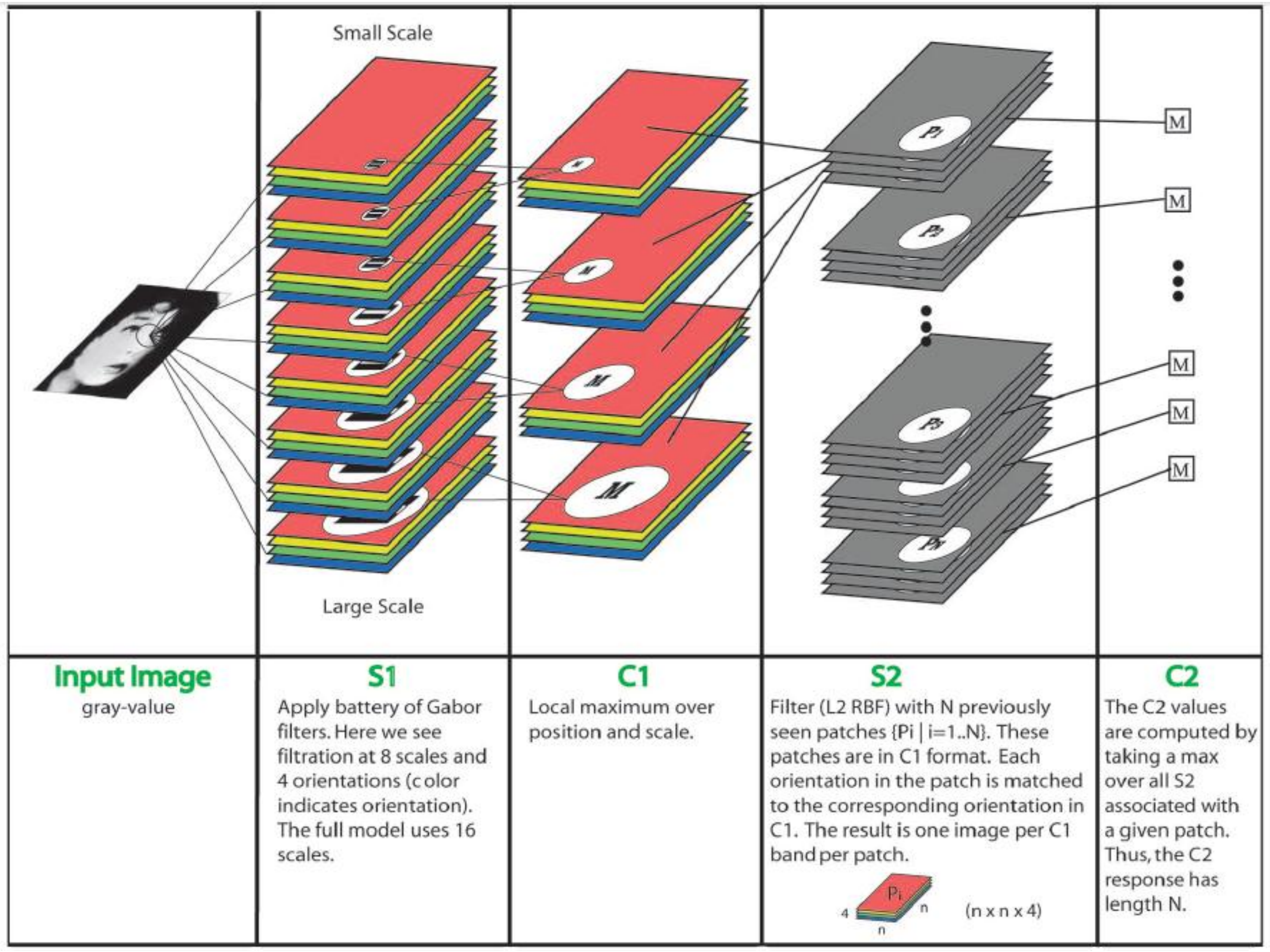
Generally the first part of the detection task consists in applying to the image the same operators as those used in the object descriptors of the world model. Thus, in techniques where features have been selected (e.g. by boosting as in [6]), the detection consists in applying the selected filters and thresholds.

In semantic segmentation methods [8], the detection/localisation task corresponds to the optimisation part of the MRF using for example Gibbs sampling, to calculate the label fields of minimal energy.

When the object model is made of a collection of local descriptors [2] [10] [11], the detection is performed by calculating the corresponding descriptor, either densely or only at salient point locations. The matching measure is then obtained by computing the distance between the descriptor vector v_x calculated at pixel x, and every model descriptor vector v_m, which can be simple Euclidean distance $d_E(v_x,v_m) = \left((v_x - v_m)^t(v_x - v_m)\right)^{1/2}$ or Mahalanobis distance $d_M(v_x,v_m) = \left((v_x - v_m)^t \mathrm{K}^{-1}(v_x - v_m)\right)^{1/2}$, where K is the covariance matrix of the descriptor distribution in the world model. When the learning is made off-line, K^{-1} is pre-calculated then the extra cost is negligible. Note that diagonalising the covariance matrix is equivalent to performing a PCA (keeping all the dimensions). Once computed, point-wise classification can be done by nearest neighbour rule, while object localisation can be made by distance threshold followed by centroid calculation.

In some cases, the descriptor, or part of it, represents a distribution (e.g. histograms of colour), for which the direct bin-to-bin distance may perform poorly for comparison purposes, because of distortions due to quantization, light changes, or deformation, that can induce important shifts in the histogram domain. Many specific histogram distances have been proposed to address this problem; see for example [15].

In bio-mimetic detection systems, every pixel undergoes a sequence of local processing, grouping and matching inspired by the architecture of animal vision. Thus, in [5], the detection task is performed in a feed-forward manner using two layers, each layer being the sequence of the application of simple cells (S) corresponding to local processing (filtering and matching), and complex cells (C) corresponding to max-pooling mechanisms. See Figure 19 for a graphical representation of the two-layer-four-stage systems.



Input Image
gray-value

S1
Apply battery of Gabor filters. Here we see filtration at 8 scales and 4 orientations (color indicates orientation). The full model uses 16 scales.

C1
Local maximum over position and scale.

S2
Filter (L2 RBF) with N previously seen patches {Pi | i=1..N}. These patches are in C1 format. Each orientation in the patch is matched to the corresponding orientation in C1. The result is one image per C1 band per patch.

C2
The C2 values are computed by taking a max over all S2 associated with a given patch. Thus, the C2 response has length N.

Figure 19: Filtering layer (S1,C1) and matching layer (S2,C2) in the cortex-like object recognition system, taken from [5].

In the first (filtering) layer, the S1 stage corresponds to the application of the bank of filters (here Gabor filters). The C1 stage locally aggregates the output of filters by replacing every pixel value by the maximal value of the corresponding output over the neighbouring pixels and adjacent scales. The resolution at this stage may be reduced, and the scales regrouped in bands. This layer normally ignores the world model, unless feature selection is performed in the learning phase. In the second (matching) layer, the S2 stage corresponds to locally calculating difference between the output of C1 and a collection of template patches recorded in the learning phase. The C2 stage finally calculates the maximum of all S2 values for each template patch. Pixel level classification is obtained using the output of C2 through a simple linear classifier trained in the learning phase.

As reported in Section 4.1.2, some methods put particular efforts in the conciseness of the descriptor, for real-time purposes. *A fortiori,* special attention is dedicated to the efficiency of the runtime model matching tasks corresponding to their descriptors. In [3], the image is cut in small blocks and dominant orientation is calculated for every block like in the model, except that one single dominant orientation (or none) is allowed for each block. If there is *n* possible orientations, every template block from the model (resp. every image block at the runtime detection), is coded with an *n* length binary word, whose possibly many (resp. only one or zero) bits have value 1. The template matching is then calculated by a simple binary AND between the model and the image binary descriptor (See Figure 11 for an illustration of dominant orientation template matching). In the case of [4], the object model is made of a hierarchy of template contours (see Section 4.1.3.4) whose best instances need to be found in the contour map of the current image at runtime. It is then necessary to compute very rapidly a matching measure between (small) contour prototypes and the current contour map at a particular location. This can be done very efficiently by computing distance transform of the current contour map, i.e. the function attributing to each pixel its distance to a contour pixel (see Figure 20 for an example), which are calculated very rapidly using constant time scanning procedures. The matching measure for a template contour at a given location *x* is then simply given by the sum of the distance transform along the template contour translated at position *x*.

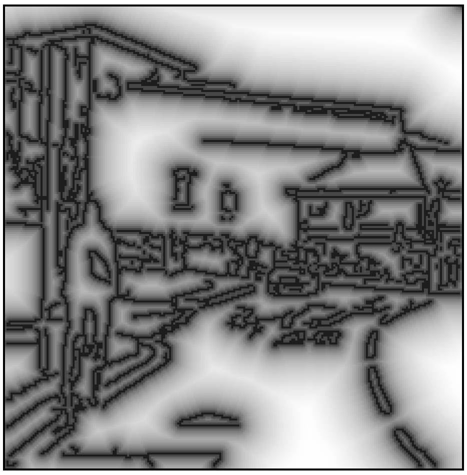


Figure 20: Distance transform of the contour map of the current image for fast calculation of the matching measure with a template contour.

4.1.5.2 Image Exploration Strategies

To make the runtime detection/localisation task computationally tractable, it is necessary to reduce the number of operations performed on the current image whenever possible, by ignoring irrelevant parts of the image and concentrating on most promising regions. Mechanisms to rapidly decide whether a zone needs to be further explored or not are then highly desirable.

The first way of optimisation is to exploit the time redundancy and the motion coherence from one image to the other. In our context the scene geometry and the motion of the vehicle can be well modelled and estimated, so using the localisation information from the last frame is clearly valuable to improve the detection in the current frame. Generally those techniques relate to visual tracking, which is detailed in another chapter. The time redundancy can also be addressed more specifically depending on the detection technique. For example, in semantic segmentation methods using MRF framework, overlapping superpixels from two consecutive frames can be linked by a temporal edge and attributed a particular energy term, e.g. $\delta \sum_{(s,s')\in V^{time}} -\log(P(\lambda(s),\lambda(s')))$ (See Section 4.1.4.1).

Another important way to lower the computational cost is to perform partial work on a region in order to decide whether this region must be discarded or further investigated. This is frequently done in the object detection methods using cascaded computations. For example, [6] apply a cascade of strong classifiers, constructed as follows. As seen in Section 4.1.3.2, Adaboost learning algorithm can be used to construct from a series of training examples a strong classifier, which is a combination of several (possibly many) weak classifiers (i.e. one Haar filter followed by a threshold). As a consequence, the best strong classifiers are very long to compute if they are calculated everywhere. The idea is then, instead of learning one single complex strong classifier, to learn first a very simple strong classifier (i.e. made of one or two weak classifiers), in such a way that the resulting classifier has maximal detection rate (and also high false detection rate), and to train a second – more complicated – strong classifier, using the false alarms of the previous one as the negative examples, and so on, until the desired performance is achieved. At the runtime level, the resulting sequence of classifiers is applied so that every classifier is applied on the image regions (windows) selected as possible object locations by the previous classifier. At its turn it rejects some regions and selects some other ones for further processing (See Figure 21).

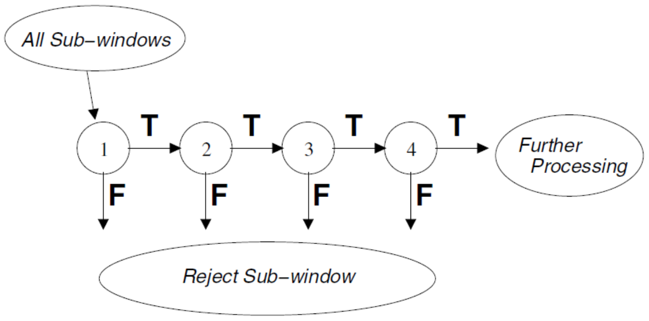


Figure 21: Runtime application of a cascade of strong classifiers, according to [6].

Those motion based and content based image exploration mechanisms are highly desirable in a real-time context and should both be considered in our application.

4.2 FEATURE EXTRACTION

This chapter details the state of the art regarding Feature Extraction from video, specifically e.g. geometric and photometric invariants.

4.2.1 Feature Extraction Introduction (ASELSAN)

Finding salient points on an image, namely feature extraction, is the basic task for most of the image processing and computer vision applications. For instance, reconstruction, detection, recognition and pose estimation necessarily require this task. Therefore, feature extraction is considered as a performance-critical factor for these applications.

Features are investigated in two main categories: local and global. If the extracted feature represents a key point or a local region over the image (or an N-D signal in general), it is simply a local feature. On the other hand, if it is calculated using some statistics obtained from the whole signal, the feature is global. Recent studies show that due to their performance in complex scenes, local features are more robust, hence, usually chosen over the global ones.

For images, the concept of cornerness comes into attention as a principle salient point. Simple calculus tells us that, any function can be represented via its derivatives at any point. This means that, the behaviour of the signal depends on the behaviour of its derivatives, which is the change in that signal. The signal with no change does not carry any information. Thus, for images, the regions with high derivatives (i.e. corners and edges) carry the valuable information. Studies [Biedermann 1987] on the subject show that human visual cognition finds the corners obtained from the silhouettes of objects more descriptive than edges.

The basic deliverable of a feature is a key point on the signal. However advanced features provide more descriptive information, namely descriptors. Descriptors are the mathematical representations of the gray level information (or some other statistical data) of a designated region around the extracted key point. Either a texture filter, or a histogram; any type of descriptor necessarily requires the position of and the effective region around the key point to be correctly estimated; which generally are the basic two deliverables of any feature extraction method.

Studies [Tuytelaars & Mikolajczyk 2007] categorize the performance criteria for feature extraction methods in six principle groups: repeatability (or stability), distinctiveness, locality, quantity, accuracy and efficiency. Different criteria become important for different applications. For instance, accuracy is important for registration tasks, whereas distinctiveness is more important for tasks requiring recognition. In addition these criteria show trade-offs between each other. Features with high repeatability are expected to show lower locality, where features with high distinctiveness are expected to show less efficiency. Thus there’s no perfect feature. The suitable feature is decided in accordance with the needs of the application and the hardware.

For certain applications, there is an important and particular relation between the repeatability and distinctiveness criteria. Many applications (e.g. recognition) require high level of distinctiveness. However, since this distinctiveness is defined over a local region around the key point; it is important to extract this effect region invariant to transformations such as rotation, scaling, etc. These types of transform invariances are repeatability attributes. The more a feature is invariant to transformations, the more repeatable it is. Furthermore, if a feature with no distinctiveness property, can provide this so-called effect region invariant to transformations; an external descriptor, which is not a part of the feature vector, can be used to increase distinctiveness. In other words after the key point, and the effect region around it are designated; using a descriptor a distinctive mathematical representation may be appended to the feature vector.

Another particular issue about feature repeatability is scale invariance. Scale invariance property of a feature or descriptor is its ability to obtain the same mathematical representation from the scaled and/or sampled (i.e. image with different resolution) version of the image. For certain applications, scale invariance is of most importance, such as recognition in real scenes. For scale invariance, a scale-space representation of the signal is needed.

One of the famous key point detectors, which attracted great attention in both academic and industrial society, is Harris corner detector. Harris is a highly repeatable accurate, and efficient, local salient point detector, which provides lots of corners. It has both affine, and scale invariant versions. It is mainly preferred for applications which require lots of repeatable corners to be matched to each other, such as registration, pose estimation etc. Harris lacks distinctive power, thus it is not generally preferred for recognition and detection applications. Since it is very efficient, its FPGA designs are very common and simple.

Another famous and relatively newer point detector is the SIFT (scale invariant feature transform). SIFT is scale invariant and highly repeatable as well. Since it provides a powerful descriptor, it outperforms Harris in terms of distinctiveness. However distinctiveness necessitates computation power. The trade-off between distinctive power and efficiency can be clearly seen between SIFT and Harris.

There are many other point detectors such as SUSAN, SURF, etc. Not only point detectors, different types of features can also be used for different applications. Some studies show that, for infrared images, wavelet based features give better results; because there is very low textural information in IR. As mentioned above, the type of feature should be decided depending on the application, the image and the hardware.

4.2.2 Feature Points, Geometric Invariants (IEF)

Interest points or corners have undoubtedly proved to be powerful features which can be easily extracted in every kind of scenes, whether they are structured or textured.

The most relevant methods among the seminal works are the Harris [38] and the KLT detectors (Kanade, Lucas and Tomasi [39][40]) which are based on the analysis of Hessian matrices of the image in a small neighbourhood around the feature points.

The current trend is to use geometric invariants such as SIFT (Symetric Invariant Feature Transform) and its speeded up version SURF (Speeded Up Robust Features). Initially designed for content indexing, the improvement of computer capabilities now allow their use for temporal real-time matching, by RANSAC or Kalman Filtering or in the videosurveillance domain [41].

To some extent, these features are invariant to image scale and rotation, affine distortion, change in 3D viewpoint, addition of noise, and change in illumination.

Due to the informative property of these descriptors, they can also be used for image recognition, via machine learning techniques (Adaboost, SVM classifiers).

4.2.3 Region Extraction (IEF)

Segmenting an image or an image sequence consists in partitioning the image into homogeneous regions in terms of color (or luminance), texture, depth or motion. The segmentation can also be achieved by combining several cues: in video-surveillance applications, the spatio-temporal segmentation is the most logical.

Generally speaking, region segmentation methods can be classified into two main classes:

- Global techniques: the features (color, texture, motion) are first classified (for example by K-means, ISODATA, Mean-shift [42])
- Local techniques. The regions are directly extracted in the image domain for example by region growing, morphological operators (watershed [43]), cooperation between regions and edges [44], or color and texture [45].

4.2.4 Color Representation and Color Invariance (IEF)

The feature points or regions can be helped with the use of an appropriate color representation, in particular color invariants [46] which are robust against shadows and lighting intensity changes.

These colors invariants have been approved on feature points detection and tracking [47], on segmentation, shadow removing [48].

An appropriate choice of the colorspace can be useful to better segment skin [49].

4.3 MOTION DETECTION, TRACKING AND EVENT DETECTION

This chapter details the state of the art regarding Motion Detection from video, specifically the following aspects:

- Image stabilization
- Detection of 3D planar surfaces
- Person tracking
- Shadow detection
- Event Detection/Action Detection/Behavior Recognition

4.3.1 Motion Detection (ASELSAN)

Motion detection is the segmentation of regions containing moving objects in video sequences. It is usually the first step of visual surveillance systems aimed at classifying and tracking motion of people, vehicles, etc. The success of subsequent processes depends on the success of this detection process. Most visual surveillance systems use stationary cameras to monitor indoor and outdoor environments. Outdoor environments tend to be more dynamic than indoor ones with distracting motion such as clouds, swaying tree branches, and water ripple. They suffer from gradual illumination changes due to the motion of the sun along with small camera displacements due to wind [20]. If the surveillance system uses panning cameras, motion detection algorithms need to compensate for camera motion.

With the advances in infrared (IR) camera technology, some motion detection systems started using IR or IR fused with visible imagery for detection. Namely, IR is considered for an application that needs to detect motion in low-light conditions such as night-time driving. IR images are of a much lower contrast than visual images due to the much narrower range of emissivity differences. They generally have lower spatial resolution, and sensitivity and image intensities of the same object are not uniform [34]. They have few distinguishable feature points and limited texture information [33].

Motion detection methods use spatial, temporal or spatio-temporal information to extract regions of interest. Some of the most conventional approaches are outlined as follows.

4.3.2 Temporal Differencing (ASELSAN)

Temporal differencing uses pixel-wise differences between several consecutive frames to detect motion. Temporal differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels e.g., there may be holes left inside moving entities [22]. Connected component analysis could be used to cluster detected areas into motion regions.

Frame differencing is the simplest method that could be used in detection of moving objects. In this method, background model is just the previous frame. Moving (running) average [16] is another method through that the current background of the video is recursively estimated from past image frames using recursive first order Infinite Impulse Response (IIR) filters acting on each pixel of the video in a parallel manner.

Yoo and Park [35] use temporal difference as a feature to detect regions of motion. Their novelty is using signed differences between frames to match covered and uncovered regions that appear when an object moves. They claim that matching these regions can reject false motion due to illumination changes since such changes do not produce a pair of signed differences. They use a similarity measure to match the regions. The algorithm operates in real-time and it can detect moving objects without prior training even when lighting conditions change. However, the authors mention that water ripples are detected as moving objects.

4.3.3 Background Subtraction (ASELSAN)

This is the most popular method for motion detection. Background regions are defined as static regions with no information of interest and foreground regions are those with motion. Foreground regions could be segmented to extract objects, people, etc. Most background subtraction techniques use statistical models and only differ in how these models are chosen. These models could be parametric or nonparametric. Foreground pixels/regions are those that are different than the background scene determined through pixel-wise subtraction or thresholding operations. Foreground detection could include steps after background detection such as noise removal, morphological filtering, etc. Background subtraction techniques include steps related to background modeling, model initialization and maintenance.

Pfinder [18] uses a multi-class statistical model for the foreground objects, but the background model is a single Gaussian per pixel. The method is designed for indoor scenes. After an initialization period during which the room is empty, the system reports good results. There have been no reports on the success of this method in outdoor scenes.

Stauffer et al. [21] propose a method in which background is modeled as a mixture of Gaussians for each pixel and the model is updated in an iterative manner. In this approach, for each pixel, a mixture of K (generally between 3 and 5) Gaussians is assigned. The pixel values that do not match these background components are considered as foreground pixels.The parameters of these Gaussians, namely mean and variance, and their contribution to the mixture are updated similar to updating in [16]. The learning rate can be taken as constant but there are some works that aims at selecting it through a formulation properly [29]. This method can compensate some natural movements, such as swaying tree branches, waves on the sea surface. To sum up, mixture of Gaussians approach is one of the most robust and computationally inexpensive methods in the literature.

Elgammal, et al. [20] propose another statistical background subtraction technique that models the background through nonparametric kernel density estimation. Whereas [18] and [21] assume a statistical distribution function with parameters that were obtained during the background modeling stage, this approach has no underlying assumptions about the data distribution and estimates the density function to build the background model. The probability of the current pixel belonging to the foreground is estimated with the model constructed using the most recent N samples. The detected foreground regions are segmented to detect/track people using a foreground model that is also determined through kernel density estimation.

Haritaoğlu proposes a complete real time visual surveillance system for detecting and tracking multiple people [19]. It uses statistical background modeling to distinguish people and other moving objects. It constructs appearance models for people and can track them after occlusions. It also detects and tracks body parts to determine if people are carrying objects or not. The background model is initialized using a median filter over 20-40 seconds of video to determine stationary pixels. The stationary pixels are then modeled using their minimum and maximum intensity values and the maximum intensity difference over N consecutive frames. Foreground pixels are those whose intensity differs by more than the maximum frame-to-frame variation over the minimum and the maximum values determined in the model. It keeps track of the number of times a pixel is classified as background, foreground and the elapsed time since the last time it was foreground to determine updates to the model. Foreground pixels go through noise cleaning, morphological filtering and connected component analysis to get cleaner foreground regions that form objects/people.

In the eigenbackground subtraction method [31], an eigenspace that models the background is built. This eigenspace model describes the range of appearances (e.g., lighting variations over the day, weather variations, etc.) that have been observed. The main idea of this method can be described as: since moving objects do not appear in the same location in the sample N images, they do not have significant contributions to this model. Consequently, the portions of an image containing a moving object cannot be well-described by this eigenspace model, whereas the static portions of the image can be accurately described as a sum of the various eigenbasis vectors. That is, the eigenspace provides a robust model of the probability distribution function of the background, but not for the moving objects. This method has reliable results if there are no significant changes in the background, for the reason that the method has no adaptation step. Therefore, this method is not suitable for outdoor environments.

Vaswani, et al. [37] uses spatio-temporal variance of pixels to detect foreground regions. The background model is based on the maximum spatial intensity differences in the first frame. The variances are calculated with a temporal window for each spatial location and compared to the variance in the previous frame and to the background model. Those pixels that are above these thresholds are considered to be foreground. The algorithm operates in real-time and claims high detection rates with low false alarm rates.

Unlike the methods mentioned above [32] is a feature based algorithm that uses IR imagery. The novelty of the method is the aggregation of the selected features by fuzzy integrals to detect foreground pixels. Their algorithm is independent of the background model chosen. The features that are aggravated are the IR intensity and texture similarity measures. IR intensity similarity measure is the ratio of the pixel intensity to the value in the background model. Texture similarity measure is the ratio between the Local Binary Patterns (LBP) of the current pixel and the background. LBP is a texture measure that is robust against illumination changes. It thresholds the eight neighbors of pixel using the central pixel value and weighs them. Fuzzy measures are used when the information about a source is insufficient to determine which class of measures should be used to classify it. The fuzzy integral here aggregates the two sources of information and determines if the pixel is a foreground pixel. The method is a real-time method that has comparable results to the well-known Mixture of Gaussians method.

4.3.4 Optical Flow (ASELSAN)

Optical flow is a dense field of displacement vectors which defines the translation of each pixel in a region. It is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames [30]. Popular techniques for computing dense optical flow include methods by Horn and Schunck [30], Lucas and Kanade [31].

Optical flow based motion detection uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence. For example, Meyer et al. [26] computes the displacement vector field for the extraction of articulated objects. The results are used for gait analysis. Optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real time without specialized hardware. More detailed discussion of optical flow can be found in Barron's work [28].

4.3.5 Obstacle Detection: Binocular Versus Monocular Approaches (IEF)

Vision-based autonomous vehicles must face numerous challenges in order to be effective in practical areas. Among these lies the detection and localization of independent moving objects, so as to track or avoid them. Various approaches have been proposed using various embedded sensors to increase the certainty (exteroceptive sensors such as radar and Lidar and proprioceptive such as odometers, accelerometers or gyros). However, most of them may provide imprecise measurements. Moreover, an occasional failure could lead to missing data. Finally data availability at time t could vary from one sensor to another. Then, a usual idea is to secure cooperation between sensors to robustify decisions. That makes sensor fusion a highly active research topic [50, [51]. Besides, many studies advocate for rather making the most of "vision" before any sensor fusion. Then one can basically distinguish between binocular and monocular approaches. Stereovision based methods provide, through calibration, an absolute measurement of a 3D space. Disparity information can be used in order to detect, without any other input, potential obstacles [52], [53]. But even if stereovision appears widely preferred in this context [54], it is considered as restrictive because of camera calibration or/and rectification step(s). Monocular vision is preferred for its several advantages including its cost, both economic and energetic, and the wealth of information extracted from monocular image sequences like, among others, obstacle motion. In some studies, rather than sensor fusion, image processing modules are cooperating.

Recent years have seen a profusion of work on 3D motion, egomotion or structure from motion estimation using a moving camera. It was followed by numerous classifications of existing methods based on various criteria. A classification commonly accepted groups existing techniques into three main categories: discrete, continuous and direct approaches.

- Discrete approaches [55], [56] are based on matching and tracking primitives that are extracted from every image in sequences (point, contour lines, corners, etc.). They are usually very effective. However, they suffer from a lack of truly reliable and stable features, e.g. time and viewpoint invariant. Moreover, in applications where the camera is mounted on a moving vehicle, homogeneous zones or linear marking on the ground hamper the extraction of reliable primitives.
- Continuous approaches [57], [58] exploit optical flow. The relationship between the computed optical flow and real theoretical 3D motion allows -- through optimization techniques -- to estimate the motion parameters and depth at each point. Results are dependent on the quality of the computed optical flow.
- In direct approaches [59], [60] motion is determined directly from the brightness invariance constraint without having to calculate explicitly an optical flow. Motion parameters are then deduced by conventional optimization approaches.
- Independent of the classification above, a large group of approaches [59] -- indifferently discrete, continuous or direct -- exploits the parallax generated by motion (motion parallax, affine motion parallax, plane+parallax). These methods are based on the fact that depth discontinuities make it achievable to separate camera rotation from translation. For instance, in "Plane+parallax" approaches, knowing the 2D motion of an image region where variations in depth are not significant permits to eliminate the camera rotation. Using the obtained residual motion parallax, translation can be exhibited easily.

Because on the one hand monocular methods lack the exact knowledge of objects depth and can only determine the exact position of a given object up to a scale factor and on the other hand the information provided by both cues (binocular and monocular) is complementary, then a current trend is to make those collaborate, in order to exploit motion analysis and scene structure. For instance, the past decade has seen many attempts to achieve a useful collaboration in the domain of obstacle detection [61], [62] or in the field of ego-motion recovery (odometry) and pathfinding [63], [64]. Some authors, such as [65] have tried to estimate the egomotion of a stereo-rig and then compute a 3D-displacement field due to this ego-motion, in order to identify dynamic objects. They use the predicted displacement field only to discriminate between static and dynamic objects; stereo-vision is then used to extract the different targets. In [66], the consistency of every feature point with the extracted ego-motion is checked through the use of a robust correlation technique. Information from stereo and motion is used to extract the egomotion of the vehicle. Known defects of this estimation are exploited to detect independent-moving obstacles. This method allows an early and reliable detection, even for objects partially occluded.

4.3.6 Detection of 3D Planar Surfaces (IEF)

Planes are important geometric features and can be used in a wide range of vision tasks like scene reconstruction, path planning and robot navigation. Homographies arise from perspective images because either the camera motion is restricted to pure rotation around the camera center, or the motion may be arbitrary, but the scene structure is restricted to a 3D plane. Both situations have been extensively exploited in computer vision. Homographies induced by the camera motion have for example been used for mosaïcing and super-resolution. The strong constraint imposed by scene planes has been used mainly for structure and/or motion recovery. Homographies allow to linearize the motion recovery and to perform measurements on scene planes in spite of perspective distortion. They also allow reconstruction of non-planar scenes, which can be described by a collection of planes together with the deviations from these planes (‘plane-to-parallax’ approaches). If the camera motion has a translational component, detecting homographies between images is equivalent to detecting scene planes, and, conversely, known scene planes enable the transfer of features from one image to the other by applying the corresponding homography. Several approaches have been proposed for detecting 3D planes including segmentation of point clouds based on fuzzy clustering methods [67] or Hough Transform in 3D point clouds [68]. In [69], a set of match pairs of interest points is obtained. An algorithm was developed to cluster interest points belonging to the same plane based on the reprojection error of the affine homography. From the calculated homographies, the planar flow is computed for each image pixel. To detect planes with an arbitrary position and orientation, in [70], a method which does establish dense correspondences is presented. A Ransac-scheme is applied to instantiate homographies and detects planar regions in the set of corresponding points, which are delineated by region-growing. Finally, in [71], a Hough-like frame called “c-velocity” supports a surface detection from an image sequence, without calibration. A vehicle’s environment considered as a set of 3D planes can then be reconstructed exploiting iso-velocity curves bound to an estimated optical flow.

4.3.7 Object and Feature Tracking (IEF)

Tracking objects or features encounters various difficulties, such as the clutter of the environment, the non-rigid motion, the photometric and geometric variations, the partial the imperfections in the camera, etc. We distinguish three main types of methods: detection-prediction, local feature points tracking, global tracking.

Detection –prediction. In most video-surveillance systems, the camera is motionless and perfectly calibrated w.r.t the observed scene. In that context, the classical approach is to detect the moving objects by background subtraction, the trajectories of these objects are then constructed using Kalman Filtering [72] or particle filters.

To some extent, such approaches can be extended to a mobile camera, by egomotion compensation. In that case, the objects are detected if and only if their motion if different enough from the egomotion. When available, stereovision can provide detections whatever the motion of the objects or the camera.

When the object detection can not be performed constantly in each frame (either because of the time and resources costs of the detector or because the object has a same motion as the dominant motion) then a temporal tracking or matching strategy is required.

Feature points tracking. The object can be seen as a collection of feature points (SIFT, SURF, KLT). Therefore, tracking the object consists for example in a block matching method, a Kalman filtering or a SSD tracking depending on the nature of the features. Such local methods take comprehensively the spatial information into account. Although time-effective, they usually fail when non-rigid objects are considered.

Global tracking. The global approaches, mean-shift [73, 74] to begin with, represent the target with a global statistical representation, mainly based on color or texture. A large number of extensions has been proposed, they differ mainly by the statistical distribution and on the similarity function. Some authors have improved the procedure either by introducing an objet/background classification [75] or by combining mean-shift with local approaches [76] or with Kalman Filtering or particle filtering, in order to deal with severe occlusions.

Unfortunately, classical histograms are not always discriminative, since they do not preserve spatial information.

Several propositions have been made to address that issue by proposing the spatiogram [76] and the correlogram [77]. In the former method, each bin of the histogram is weighed by the mean and covariance of the locations of the corresponding pixels. In the latter, color correlations are considered for several directions. In [78], the Color Connectedness Degree has also improved the tracking performances.

The Covariance trackers have recently emerged [79]. They use the covariance matrix of features as a compact and discriminant spatio-colorimetric representation of the target.

4.3.8 Shadow Detection (CogVis)

Current State-of-the-Art people trackers often use the tracking-by-detection approach, meaning that the steps of object detection and tracking are not separated, but combined. This provides the possibility to make use of further information provided by object detection algorithm, e.g. the authors of [80] does not only rely on the result of the object detection algorithm, but make use of the underlying probability map to enhance the robustness of their algorithm. For people detection, the Histograms of oriented gradients [81] and cluster boosted trees [3] are widely used (e.g. [80], [83], [86]). Furthermore, appearance models are also widely used (e.g. [84], [85], [86], [87]) to describe the persons' appearance. This enhances the robustness as it is possible to correctly identify people after full occlusions.

Shadow detection is an issue in many computer vision applications. Detecting and object in a scene is often established by shape matching or blob comparison. When a detected blob represents the desired object and its corresponding shadow and the whole blob is then used for finding the best finding match e.g. within a database, the system will return wrong or inaccurate results. In [88] a survey on problems related to shadow detection is presented as well as solutions for some of them.

Additionally, an approach for a robust but user guided shadow detection is shown. In [89] a pixel-based statistical shadow detection approach is presented to model moving cast shadows of nonuniform and varying intensity. The used models are learned using a Gaussian Mixture Model, which can also deal with complex scenes (e.g. varying lighting conditions) and prevent shadow detections in regions, where shadows cannot occur. An online learning mechanism for detecting the shadow within an image is shown in [90]. It combines a learning and a training phase. After the determination of the moving object a shadow likelihood is calculated and updates after each frame.

The purpose of event detection is to be able to detect various events in image sequences such as a robbery in bank foyers or other conspicuous situations. The input of event detection algorithms is a database of events and the output is the classification of the current situation in the image sequence. In [91] dynamic events are classified with the help of trajectories from a vision based tracking algorithm. Event detection in crowds is a challenging task due to overlaps of persons. Shandong et.al. present a method for anomaly detection, where people in crowds are tracked using crowd flow modeling [92]. In crowds, particle tracking is used to find trajectories of people which is later used in order to detect events.

4.3.9 Event Detection and Behavior Recognition (Multitel)

Practically speaking, very few scientific literature exist on the specific task of behavior recognition and event detection from embedded camera, and available ones are dedicated to police applications and very recent. For example, [93, 94] propose a method for automatic detection of specific abnormal events during police traffic stop, like opening door, person running out or officer falling down. This method use the recording cameras already installed in many police cars. Algorithms developed for surveillance systems [95] could also be applied here to detect abnormal events. This kind of system could be used to automatically detect critical events, like officer aggression, and perform appropriate action, for example notify headquarters for help.

The traffic stop scenario presents several advantages : it is a critical time during which many abnormal events can occur, the camera is static and close to the observed scene, and the system can be manually triggered when the officer leave the vehicle to avoid to many false alarms. Extending the system to other scenarios, and in particular to the case of moving vehicle, is probably unattainable at the current state of the art.

4.4 VEHICLE EMBEDDED VIDEO ANALYSIS

This chapter details the state of the art regarding Vehicle Embedded Video Analysis, specifically the following aspects:

- Academic/commercial applications for embedded video analysis (from active/passive driver assistance systems to video surveillance)
- Interesting/used technologies in this context (laser, stereo, etc...)
- Main initiatives/projects currently ongoing (DARPA, google driverless vehicle...)
- Specific applications related to police forces (LPR, etc.)
- Hardware constraints on image analysis algorithms

4.4.1 Academic and Commercial Applications for Embedded Video Analysis

4.4.1.1 Pedestrian Detection (Multitel)

Automatic pedestrian detection in images and videos is a very active research topic in computer vision. Accurate pedestrian detection is key to a number of important applications, such as automotive safety, robotics, video surveillance and human machine interfaces, among others. Despite many improves have been made in this field over the last years, it still remains a challenging problem. The main difficulty is the huge variability of pedestrian appearance arising from changing pose, clothing, lighting and point of view. Inter-person occlusions, high background variability and/or small resolution images make the problem even harder in many real world scenarios.

Some approaches based on background subtraction have been proposed (see [96]), but these methods suffer from a high sensitivity to environment, are limited to low density of persons and are not usable with moving camera. The vast majority of recent works use learning approaches with corresponding large training sets to build a model of pedestrian appearance. This model distinguish whether an image region contains a pedestrian or not and is scanned on the images to detect people regardless of their position and size.

These techniques have two main processing steps: feature extraction and classification. The feature extraction task extract image features from the visual content, and the classification task use the obtained features in a classification framework to detect the pedestrians. The aim of the feature extraction step is to extract higher-level information from raw pixel data to facilitate the task of the classifier. Two different types of classifiers can be used: discriminative classifiers and generative classifiers.

Generative classifiers build a model of appearance of the object to be detected, constructed from examples of that object. The feature used is generally ([97, 98]) the global shape, extracted by an edge detector, which is relatively insensitive to illumination variations and clothing. The variability due to the pose and orientation of people is modeled by the generative classifier. Some other works use an Implicit Shape Model([99, 100]), which is based on a codebook of characteristic local appearances and a model of their spatial distribution. Generative approaches, however, are less common than discriminative approaches.

Discriminative classifiers are trained using both positive and negative examples to determine the best boundary between these two sets. The classification methods used are mainly:

Cascade of Adaboost classifiers Adaboost is the most widely used method in this field [101-105]. The principle of this method is to build a strong classifier from a set of weak classifiers, which are typically a decision-tree formed from one or more feature. Adaboost is often used as a cascade of classifiers, a technique first proposed by [101] that permits to achieve a computation time compatible with real-time applications.

This approach is based on the finding that a large majority of tested windows are negative. Instead of applying a monolithic classifier on all the windows, one constructs a sequence of classifiers of increasing complexity that eliminate gradually the negative windows while retaining almost all the positive windows. In this way, the full classifier is applied only to a small minority of windows.

Support Vector Machine (SVM) SVMs are also widely used for pedestrian detection [106, 107]. This studies use linear SVM ; indeed, although some work [106, 108] suggest a slight improvement in performance can be achieved with more complex kernel functions, it comes at the cost of a significant increase in computing time.

Neural network Some works [109-112] have proposed the use of neural networks for classification task. In these works, no explicit feature extraction step is involved : the special structure of the neural networks ([113]) allow to use pixel values as input. The feature extraction is done inside the network, and is thus tuned during the training to be most discriminative.

Methods based on a discriminative classifier are numerous and use a wide variety of features. The most common features are:

Haar Filter These filters have been used by [101], which was the first detection person method compatible with real-time. Of very simple shape, they have the advantage of being extremely fast to compute using integral images. They have been improved by [104], which uses combinations of these filters on multiple frames to describe the movement.

Histograms of oriented gradient (HOG) [106] show very good performances of this type of feature for person detection. These descriptors represent the intensity distribution of the gradient depending on the orientation. They are therefore well suited to describe the shape of people.

Local Binary Pattern (LBP) Originally intended for the description of texture, these features were used by [114], [107] and [115] for person detection. LBP mainly describe the textures of the object, and are therefore complementary with features based on shape, as Haar or HOG.

Covariance Described in [103], the feature consists of the covariance matrix of several low-level descriptors, mainly related to the gradient. Covariance matrices not belonging to a classical vector space, the classification method must be adapted to work in a Riemannian space. [116] reduces the computation time by several optimizations and manages a real-time system with good performance, combined with background subtraction algorithm.

Color The majority of current features do not use color, which is not considered strong enough, because of the great variability due to clothing. However, some studies show that adding it to other features can improve performances. [117] adds information to the HOG, and shows that it provides a performance gain. [105] also tests its descriptors on the LUV color space, and shows that it provides much more information than grayscale only.

Motion A first approach to take into account the movement is background extraction, only applicable in case of fixed cameras. [116] uses probability map and the foreground as a template to further increase performance and reduce the computing time of his method. Another approach is to exploit directly the specific movements of people. [118] uses a frequency analysis to detect periodic motions of human walking. [104] apply combinations of Haar descriptors on several consecutive images to model the movement and reduce the false alarm rate of the detector. [106] proposes to use the optical flow, and develop a version of HOG based on it.

A combination of different types of descriptors can also be used [107, 119–121], and brings significant performance improvement.

When available, depth is an important cue for person detection that can greatly improve performances of the detector. It can be used in a preprocessing step to rapidly discard irrelevant regions of the image [109, 111, 122], or in the detector as an additional feature [121, 123]. The use of depth require of course additional sensor (generally stereo camera), and processing power to compute dense depth map from stereo images.

Several studies have been made to standardize procedures to assess performances and compare these methods with each other [119, 124–126]. Some extensive surveys [96, 127, 128] are also available. These works show the overall prevalence of methods based on the HOG feature and the value of combining different features and cues (intensity, motion and depth) to improve performances. They show that if the best methods happen to good performance on high-quality images, they fail in the case of more realistic scenarios, with occlusions, variable quality images and highly variable background.

In addition to the detector itself, [128] also review the problems of candidate generation and tracking. These two steps have shown to be indispensable to achieve good performances in real systems. Candidate generation extract the regions of interest from the image to be sent to the detector, avoiding as many background regions as possible, instead of the classical exhaustive scanning approach. This approach is used especially by [111] and [129], which use stereo information to discard irrelevant regions from the image. The tracking step follow detected pedestrians over time and is mainly used to reduce false detection rate [111, 130].

Finally, some studies have tried to setup a complete system in the context of Pedestrian Protection System. The aim of these systems is to predict possible collision with a pedestrian, to warn the driver, and to brake automatically in urgent situation to avoid collision. This is of course a critical application, which requires very high robustness and low reaction time.

The European Commission-funded research project PROTECTOR (2000-2003) [110, 111] was one of the first attempts. The system developed is based on a stereo-camera, and consists of a cascade of five modules: a stereo-based candidate generation module, a shape-based generative detector, a neural-network detector, a stereo-based verification module and a tracking module. In real urban traffic conditions, the overall system reaches a detection rate of 62–100% at the cost of 0.3–5 false classifications per minute (depending on the metric considered).

Another system has been developed by Mobileye [130]. It use a single camera, and consist of a cascade of four modules: an attention-based candidate generation module, a multi-part discriminative detector, a multi-frame approval module that track the pedestrians detected and analyses further their trajectory, and a range measurement module. Reported performance is, for good conditions, a detection rate of 90% for less than 2 false alarms per minute. The system has been integrated in some new Volvo cars, in combination with radar, in the first automatic pedestrian avoidance system.

However, all these systems still suffer from important limitations: they are limited to daytime, good weather conditions, and detect accurately only close and non-occluded pedestrians.

4.4.1.2 Vehicle Detection and Traffic Classification (Multitel/ACIC)

On-road vehicle detection is an active field of research, because of its important applications in automotive safety and driverless vehicles. A survey of vision-based methods can be found in [131].Most of these works report good performances, but are limited to specific conditions (rear or front view, highway, ...), and the lack of realistic test data and benchmark makes the evaluation of their performance in real conditions difficult.

A method based on stereo camera [132] was used in driverless vehicle ARGO to detect and track the preceding vehicle. A vehicle detection algorithm is implemented in MobilEye products, for their forward collision warning system, and is integrated in combination with a radar system in some new Volvo cars. However, most of the current vehicle detection systems used in automotive security or in driverless vehicle rely on radar or laser sensor, for robustness reasons.

Last, the relative speed of surrounding vehicles can also be measured using a camera fixed on the front of a bus operating on its reserved lane, and the traffic conditions can then be classified in real-time into different categories (“fluid”, “congestion”…) using an odometer and/or GPS data [133].

Overview of techniques and trends for on-board detection of surrounding vehicles and obstacles

This overview is inspired by the paper of Sun et al. [143], which we believe is a complete and clear overview of the state-of-the-art in the field. In particular, we re-use some of their proposed terminology. The interested reader may refer to that paper for more details.

The two steps of vehicle detection

Since processing the full images would be too expensive in terms of computing time and prevents from achieving real-time performances, two basic steps are often covered by today’s methods:

1. Hypothesis generation, i.e. the detection of candidates regions in the images where vehicles are potentially present
2. Hypothesis verification, i.e. verifying that the pre-detected regions effectively include vehicles.

There might off course be strong overlap of these two steps that can even be merged in some cases.

Hypothesis Generation

Knowledge-based methods

First way to generate hypothetical vehicle regions exploits a priori knowledge on the vehicles and the visual scene. Typical examples of a priori knowledge include symmetry (man-made objects present symmetry), colour (road and lanes have typical colours so that vehicles can be segmented from the background), shadow (that is often to be observed under the vehicles), corners (a car observed from rear or front present 4 typical corners), vertical and horizontal edges (cars present particular constellations of edges), texture patterns and vehicle lights (for night detection). All these features have advantages and disadvantages but the edges have proven to be the most promising cue for hypothesizing vehicle presence and methods exist to speed up their detection and interpretation, including multi-scale approaches [144]. The main drawback of edges-based approach relies in the number of parameters that could affect the system performance and robustness.

Stereo-vision-based methods

There are two ways to exploit stereo images, i.e. acquired with two close and horizontally aligned cameras. First, the map of the pixel difference between the left and right images, named the disparity map, can be constructed as far as the calibration parameters of the stereo rig are known. The disparity map once computed can be transposed into a 3D map of the visual scene so that obstacles within a depth of interest can be detected. The main disadvantage of this approach relies in the complexity for computing the correspondence between left and right images' pixels. Solutions have been proposed for this method, such as associating a local feature extractor, but this remain a real issue in the presence of vibrations and irregular movement of the host vehicle.

Another way to exploit stereo images is to use the Inverse Perspective mapping (IPM) [145]. This refers to a projection of the image points toward the horizontal plane through a centre of projection located between the image and the horizontal planes. The IPM results in a new image including all the projections on the horizontal plane. Assuming a horizontal flat road this transform would project the road onto itself, while objects above the ground would be distorted on the resulting image. The difference between the IPM transform of the right and left images should ideally present triangles corresponding to the borders of the objects. The detection then consists in identifying these distorted triangles, which is not a trivial task.

Rigs of more than two cameras have also been proposed, providing a richer image information with the drawback of higher costs and more complex pixel correspondence computation.

Motion-based methods

While the two knowledge-based and stereo-based methods exploit features discriminating the vehicles from the background, motion-based methods analyse the relative motion to be observed in the visual scene. Typically, approaching or overtaking vehicles should have a motion pattern different from the background. The relative motion can be obtained by calculation of the optical flow, i.e. the computation of a motion field from the intensity field through temporal and spatial derivatives. However, a reliable estimation of the optical flow with a moving-camera is not an easy task and pretty the consuming.

"Sparse optical flow" can improve the performances. It consists in employing additional image features information (corners, colour, local minima and maxima, etc). Though this provide a sparse motion information, it is sufficient to detect potential presence of vehicles. Moreover, the fusion of the information makes this approach more robust to noise.

From a general point of view, motion-based methods are definitely promising. However, they suffer from several factors, the displacement between consecutive frames (too low frame rate may affect the computation of motion if the host vehicle does moves too fast), the lack of textures (too homogeneous regions, e.g. the road, can affect the derivative computation for motion calculation) and the shocks and vibrations of the vehicle introducing noise. Image stabilization techniques have been proposed to address that issue [146].

Hypothesis verification

Template-based methods

The idea of template-based methods is to assess the presence of vehicles by estimating the correlation between the hypothetical vehicles and templates. Templates include for instance a "U shape' expected for rear/front silhouette of vehicles, pronounced symmetry, pairs of headlights or other a priori knowledge about the vehicles shapes and appearance. The weakness of template-based methods is obviously due to the extreme variability in vehicle appearance consequent to the various angles of views, illumination changes and types of vehicles.

Appearance-based methods

Appearance-based methods consists a two class classification task to conclude whether the candidate region correspond to a vehicle or not. Starting from a training set, a classifier (such as based on Neural Networks, Support Vectors Machines or Bayesian) is trained and tested on the candidate regions. The classifier relies on a description of the regions based on features that might be either local (to a region) of global. Classical features include for instance Haar-types functions. Traditional feature selection techniques have been used for the classification training, such as Principal Component Analysis (PCA) and others.

The key challenge in this classification task is the constitution of a representative training set that must support the extreme in-class variability of the vehicle class.

Tracking

Exploiting the temporal coherence between the video frames can help predicting the position of vehicles in the frames [146]. This is why tracking is ever more used for detecting vehicles and obstacles. The majority of approaches use a detect-then-track approach, i.e. detect vehicles with traditional techniques and then follow them over the next images. The advantage is obviously an improved robustness with the drawback of higher complexity. However, we expect tracking to get more and more importance in tomorrow's approaches for on-board obstacles detection.

Expected advances

In [143], Sun et al. also discuss possible advances toward improvement of today's vehicle detection techniques. Among them, progressed in the customisation of the algorithms in view of fulfilling the requirements of each specific functionality, the combination of multiple cues and more advanced feature selection techniques in the classification tasks.

Expected progresses also covers the improvement of sensors, in terms of dynamic range and resolution, the fusion of multiple sensors, such as acoustic + video or exploiting infrared, software and hardware improvements and the ability for the processing unit to autonomously detects its own failure (e.g. excessive false alarm rate) preventing the system to become inconvenient for the driver.

4.4.1.3 Automatic License Plate Recognition (Multitel)

Automatic License Plate Recognition is now a relatively well solved problem in computer vision, and is used in many commercial products [MU 134–137]. A survey of methods can be found in [138]. These methods impose some constrains on image capture in resolution, angle of view and contrast, but these constrains can be reasonably satisfied with carefully designed system. Under these constrains, the reported performances are quite good, and sufficient for daily use in police car. The commercial products use generally infrared camera and illuminator to avoid illumination problems and operate at any time of the day.

Mobile ALPR systems have the ability to quickly scan a large number of license plates that can be automatically compared with police databases of stolen vehicles, prohibited or uninsured drivers, etc. They are now widely used in some country and have been shown to improve police efficiency [139].

4.4.1.4 Other applications (ACIC)

Image processing activities for automotive safety or intelligent vehicles cover other. Among them are:

- Lane detection for lane departure and drowsy driver warning systems [140].
- Car tracking or following with extraction of measures [141].
- Bicycle detection
- Obstacle detection other than pedestrians, vehicles…
- Automatic parking
- Adaptive cruise control based on vehicles/obstacle detections.
- Night vision
- Traffic sign recognition
- Intelligent headlight control.

We have to keep in mind that most commercial applications use several sensors at once rather than only video processing. Most often, the sensors are video sensors in the visual domain, lasers and infrared sensors.

While some of these activities are clearly not relevant for SPY, some may be of interest for the safety of Police patrols (e.g. night vision and obstacle detection in poor lightning conditions).

4.4.1.5 Police Specific Applications (ACIC)

- Automatic number plate recognition (ANPR). This system uses cameras to observe the number plates of all vehicles passing or being passed by the police car, and alerts the driver or user to any cars which are on a 'watch list' as being stolen, used in crime, or having not paid vehicle duty.
- Speed recognition device. Some police cars are fitted with devices to measure the speed of vehicles being followed, such as ProViDa, usually through a system of following the vehicle between two points a set distance apart. This is separate to any radar gun device which is likely to be handheld, and not attached to the vehicle.
- Car following. This system allows to get an optimal video of a vehicle that is purchased by a police car.
- Surveillance. Those systems typically just record videos. However, more intelligent systems could be used for tracking criminals activities with unmarked vehicles. There is a very high demand for such systems being able to optimize special forces human resources with state of

the art video analytics systems but with with the constraint of being easy to deploy by non-experts.

As an exemple, the ProVida system provides ANPR, speed recognition and surveillance functionalities [142].

4.4.2 Hardware Constraints on Image Analysis Algorithms (EOLANE)

Electronic systems in general and video systems in particular have to fit its deployment environment. A system is designed and packaged with very different ways depending on whether it is located in labs, in a dedicated room with available air-conditioned, in an industrial plant or if it is operating in a vehicle such as car, train or drone.

4.4.2.1 Environmental Constraints

When a video system is embedded in a vehicle, the available volume space is usually low because most of the place is reserved to the core system, such as the Control and Command System of a train for example. Standard solutions based on COTS devices are rarely compliant with the allocated space and mechanical engineers have to do utmost to define packages ready to be installed where it is planned.

The low volume does not make easy the air-cooling because little air can be fanned and draining ways are much reduced. Conduction-cooling can be an expensive but possible solution, but fans induce noise whose level should be controlled, especially when passengers are on board. Due to this, the choice of the components is falling on ones with low energy consumption to reduce heat emanation. Another mechanical topic, other than the packaging, is related to the resilience to shocks and vibrations. Again in this case, solutions based on COTS devices are rarely compliant to these requirements and specific adaptations are needed. Moreover, the remaining available volume is generally airless and bad located, near heat sources like engines, or under the roof.

The source of energy is an additional constraint. Sometimes the device must be autonomous; it is then powered by wind or solar energy. In other case, the available power is limited. For example, more and more electronic devices are installed inside police cars and batteries are not enough strong to power these devices and automotive devices: when lights are put on, the battery can be down! Embedded systems must therefore be careful with the energy: the choice of processor depends also on these criteria.

Finally, the list of environmental constraints would not be full whether EMC and ESD are not mentioned, which imply specific protections having consequences upon the size and the weight of the device.

4.4.2.2 Microprocessor Market

On one hand, the environmental constraints listed above imply that the processor used in an embedded video system shall be very careful of energy and thermal characteristics. On an other hand, the integration of a widely use microprocessor may have advantageous effect on the price of the chip. The correlation of these two criteria leads to the Mobile Internet Devices (MID) market, such as mobile phone and netbooks. These devices are based on two different microprocessor architectures: the ARM Cortex-A8 series and the Intel Atom N330.

Some studies have been done to compare these two microprocessors family from an architectural and benchmarking point of view [147]. The basic benchmarking results show that the Cortex-A8 provides much more significant power savings than the Intel Atom‘s. On the other hand, if the Cortex-A8 increases its clock speed to 1.5GHz, it can achieve similar performance to the Intel Atom as both chips provide approximately the same performance per MHz.

This type of microprocessor matches with the requirements of embedded video system and delivers more than enough power for a basic IP camera. When introducing additional features, such as video processing, algorithms will not have available a process power equivalent to a quad core i7 from Intel or any processor of this type, and have therefore to be careful with the necessary power.

4.5 DECISION MAKING IN DISTRIBUTED MULTI-SENSOR SURVEILLANCE SYSTEMS

In this chapter the State of the Art regarding automated decision making in networked multi-sensor surveillance systems is detailed, also regarding the use of positioning.

In modern surveillance systems for public safety, real-time distributed architecture is required to transmit sensor data immediately for deduction. Awareness and intelligence is applied to address the automatic deduction. Video surveillance is thoroughly used in public safety and the usage of wireless networks in the field is growing. Surveillance personnel often patrol in surveyed areas and their precise location must be known to exploit their benefit to the fullest [148].

4.5.1 Multi-Sensor Surveillance (VTT)

In current surveillance systems, data are collected by distributed sources such as sensors, and typically transmitted to a remote control center. Multisensor systems can capitalize from processing either the same type or different type of information collected by sensors, e.g., video cameras and microphones, of the same monitored area. Appropriate processing techniques and new sensors offering real-time information associated to different scene characteristics can assist both to improve the size of monitored environments and to enhance performances of alarm detection in regions monitored by multiple sensors [149].

As an example, improving smart cameras with additional sensors could transform them into a high-performance multisensor system. By combining visual, acoustic, tactile, or location-based information, the smart cameras become more sensitive and can transmit results that are more precise. This makes the results more applicable widely [150].

In its current generation, the technology revolves around wide-area surveillance systems. This results in the advantages of the collection of more accurate information by combing different types of sensors and in the distribution of the information. Difficulties lie in the efficient integration and communication of information, establishment of design methodologies, plus moving and multisensor platforms [150].

4.5.2 Information Adaption, Data Fusion and Information Fusion (VTT)

In contemporary surveillance systems, in which multiple asynchronous and miscellaneous sensors are used, the adaption of the information acquired from them to derive the events from the environment is an important and challenging research problem. Information adaption refers to the process of combining the sensor and nonsensor information using the context and past experience. The issue of information adaption is vital, because when information is acquired from multiple sources, adapted information offers more precise inferences of the environment than individual sources [161].

Data fusion techniques can be used to enhance the estimation of performance and system robustness by exploiting the redundancy offered by multiple sensors observing the same scene. With recent advancements in camera and processing technology, data fusion is being considered for video-based systems. Intelligent sensors, which are equipped with microprocessors to execute distributed data processing and computation, are available and can decrease the computational burden of a central processing node [160].

Blasch and Plano [162] state that “data fusion” is a term used to refer to the bottom-level, data-driven fusion. “Information fusion” refers to processing of already-fused data, such as from primary sensors or sources, into meaningful and preferably relevant information to another part of the system, human or not [162].

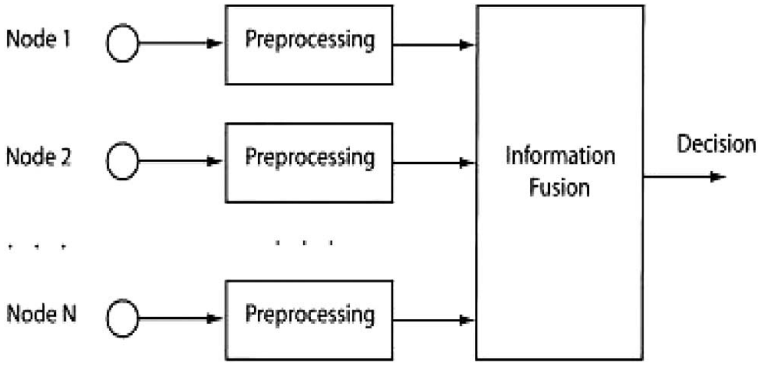


Figure 22: Simple example of a basic architecture [153].

Figure 22 illustrates a simple architecture for information fusion. The nodes scan the environment periodically and transmit a signal. The received signal is first processed by a preprocessor to extract significant characteristics from the environment. The preprocessors are responsible for quantifying how much the environment is different from the steady state. The information fusion function then deducts if there is an intruder present or not [153].

Data fusion from multiple cameras involving the same objects is a main challenge in multicamera surveillance systems and influences the optimal data combination of different sources. It is required to estimate the reliability of the available sensors and processes to combine complementary information in regions where there are multiple views to solve dilemmas of specific sensors, such as occlusions, overlaps, and shadows. Some traditional benefits, in addition to extended spatial coverage, are the enhancements in accuracy with the combination of covariance reduction, improved robustness by the identification of malfunctioning sensors, and enhanced continuity with complementary detections [163].

Typically, surveillance systems are composed of numerous sensors to acquire data from each target in the environment. These systems encounter two types of dilemmas, which are 1) the fusion of data which addresses the combination of data from distinct sources in an optimal manner, and 2) the management of multiple sensors, which addresses the optimization of the global management of the system through the application of individual operations in every sensor [164].

Information adaption is a challenging task, because of 1) the diversity and asynchrony of sensors, 2) the disagreement or agreement of media streams, and 3) the confidence regarding the media streams. There is an issue on how to fuse individual information to establish comprehensive information. These are items of importance and essential challenges [165].

4.5.3 Information-Decision Fusion Engine (C2Tech)

In modern surveillance systems, information coming from several sensors has to be fused in order to overcome the uncertainty in the observed area. The main purpose of fusion is to provide an overall picture of the information collected by different platforms to classify/identify the targets and to show the locations and movements of all entities. Multisensor data fusion is an evolving technology, concerning the problem of how to fuse data from multiple sensors in order to make a more accurate estimation of the environment.

The Multisensor Information Fusion engines, take feeds of data from the sensors, run pre-existing fusion algorithms to mine the data and analyse the input to generate an operational picture. The engine must be capable to assign default labels applied to sensors supplying real time data feeds and must be combine evidence to determine platform’s position, velocity, direction, and identity parameters.

A rule based engine, which is defined during run-time, is feasible. The user should apply any rule defined through the user interface, should define new rules using the existing predicates on its library and should add the new statements to library.

The execution process on the system should be customized as listed below.

- Rule execution order can be changed.
- New rules can be added.
- New predicates can be added.
- Required predicate queries can be defined.
- New database can be used.

In the case where multiple sensors are used, information fusion engine should correlate the data about the same real world object from these different sensors according to the pre-defined common data fields.

Network-centric fusion requires that information is pre-aligned to a uniform model (or can be rapidly aligned) so that all data sets are self-synchronized. For this purpose information fusion engine should wait for the inputs in a standardized way. Although the system should be capable of parsing different filetypes and querying different database management systems, data should be pre-aligned and indexed. Each data field should be clearly defined with its structure, data type, metrics, etc...

Open Geospatial Consortium (OGC) is an international not for profit voluntary industry consensus standards organization that provides a forum and proven processes for the collaborative development of free and publicly available interface specifications (open standards).

OGC provides standards for data fusion as well. One can consider Information Fusion as a decision fusion process. According to OGC, decision fusion focuses on client environments for analysts and decision makers to visualize, analyse, and edit data into fusion products for an understanding of a situation in context.

Decision fusion includes the ability to fuse derived data and information with processes, policies, and constraints. Collaboration with other analysts is done using social networking services and collaboration tools that are location enabled.

The objectives for fusion in this category include:

1. Discovery of data (static and dynamic) resources that meet a user’s immediate requirements and to make those resources part of a fusion process under the control of the decision maker or analyst.
2. Retrieval of real-time or time-series data in standard encodings that provide the ability to fuse the data into useable information based upon the users uncertainty of the measurement and parameters needed to process the data.
3. Determination of the quality and validity of the data and fusion products produced from the data.
4. Ability to fuse derived data and information with processes, policies, and constraint information as set by the data/information owners (i.e., Concept of operations) and decision services processing nodes.
5. Ability to present the derived information in a spatial client application (e.g., SLD, SE, W3D) including portrayal of maps and 3D visualization.
6. Ability to collaborate with other decision makers and analysts using social networking services and collaboration tools that are location enabled.
7. Documents that capture an analysis result and allows for distribution to others for viewing the same context.

According to OGC, fusion can be categorized as sensor, object/feature and decision fusion.

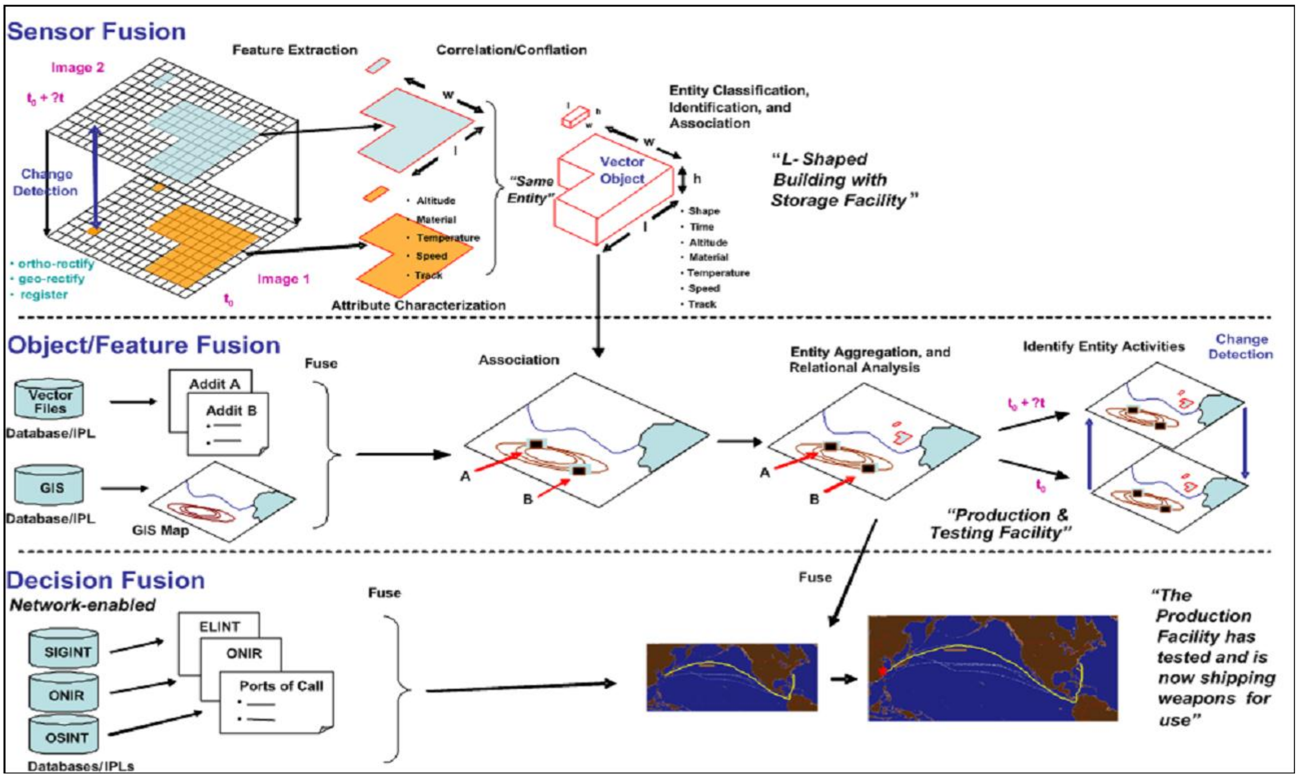


Figure 23 : Categories of Fusion

Sensor Fusion: ranging from sensor measurements of various observable properties to well characterized observations including uncertainties. Fusion processes involve merging of multiple sensor measurements of the same phenomena (i.e., events of feature of interest) into a combined observation; and analysis of the measurement signature.

Object/Feature Fusion: includes processing of observations into higher order semantic features and feature processing. Object/feature fusion improves understanding of the operational situation and assessment of potential threats and impacts to identify, classify, associate and aggregate entities of interest. Object/feature fusion processes include generalization and conflation of features.

Decision Fusion: focuses on client environments for analysts and decision makers to visualize, analyze, and edit data into fusion products for an understanding of a situation in context. Decision fusion includes the ability to fuse derived data and information with processes, policies, and constraints. Collaboration with other analysts is done using social networking services and collaboration tools that are location enabled.

Decision fusion actually compounds other two fusion categories. It is client based and more convenient for visualization. So, decision fusion would be the best alternative for the fusion process.

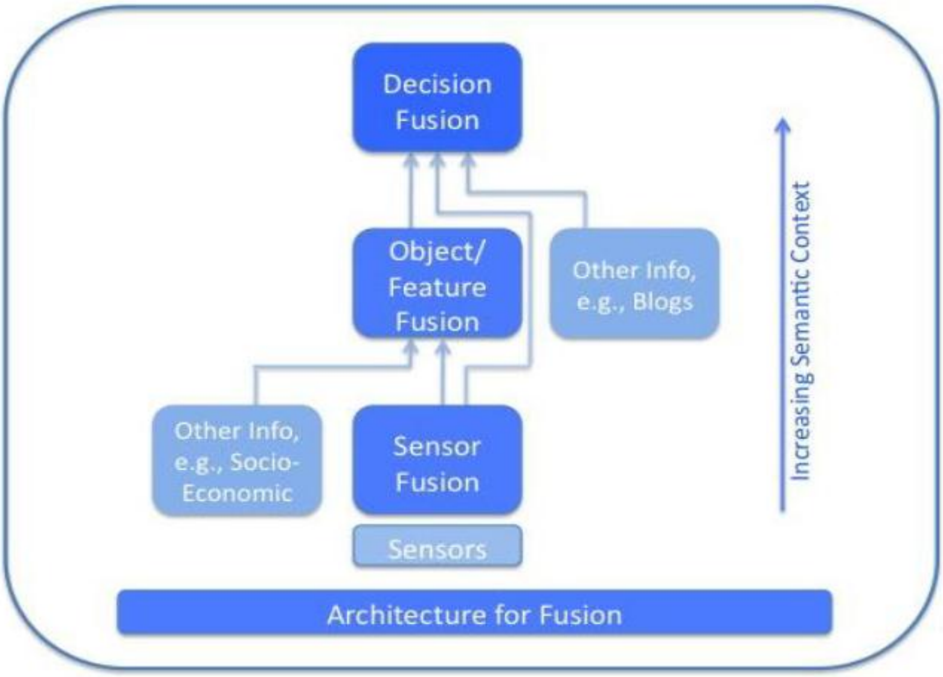


Figure 24 : ER Model of Fusion Categories

Different approaches were offered on how to model a fusion process. Some of them are stated below:

JDL DATA FUSION MODEL:

One of the most widely used frameworks is the JDL Data Fusion Framework. The Joint Directors of Laboratories (JDL) data fusion sub-panel within the US Department of Defence originally defined this system in the early years of data fusion. This framework was developed to aid the developments in military applications. Here are the levels at which data fusion could be undertaken:

- Level 1, object refinement, attempts to locate and identify objects. For this purpose a global picture of the situation is reported by fusing the attributes of an object from multiple sources. The steps included at this stage are: Data alignment, prediction of entity's attributes (i.e. position, speed, type of damage, alert status, etc.), association of data to entities, and refinement of entity's identity.
- Level 2, situation assessment, attempts to construct a picture from incomplete information provided by level 1, that is, to relate the reconstructed entity with an observed event (e.g. aircraft flying over hostile territory).
- Level 3, threat assessment, interprets the results from level 2 in terms of the possible opportunities for operation. It analyses the advantages and disadvantages of taking one course of action over another.

THOMOPOULOS ARCHITECTURE

Thomopoulos posed architecture for data fusion consisting of three modules, each integrating data at different levels or modules to integrate the data, namely:

- Signal level fusion, where data correlation takes place through learning due to the lack of a mathematical model describing the phenomenon being measured.
- Evidence level fusion, where data is combined at different levels of inference based on a statistical model and the assessment required by the user (e.g. decision making or hypothesis testing).
- Dynamics level fusion, where the fusion of data is done with the aid of an existing mathematical model.

BEHAVIOURAL KNOWLEDGE BASED DATA FUSION MODEL

A feature vector is first extracted from the raw data. This vector is then aligned and associated to defined features. Fusion is then undertaken at the sensor attribute and data analysis levels. The final step is composed of a set of behavioural rules, which can be extracted in terms of the final representation of the fused output. Rather than assuming the blackboard architecture typically found in knowledge-based systems, this process model uses a hierarchical approach containing three levels of representation:

- The lowest level contains, for each sensor, a vector space with coordinate dimensions and measured parameters.
- The next level extracts relevant features from these vectors, and attaches labels to them.
- The third level contains a set of formalisms about the world model that relate feature vectors to events.

Today the most used model in security applications is JDL processing model. It is more convenient to decision fusion since it interprets the results and directly helps to make decisions.

Possible technologies to use with decision fusion are stated in table below:

Web Services	Means to connect producers and consumers of resources (data and services), e.g., SOAP and REST
Security	Means to enable authentication, authorization, confidentiality, and integrity of resources and interconnections
Workflow	Standardized means for automation of business processes and event processing
Grid computing	High performance distributed computing and very large datasets
Cloud computing	Software as a Service (SaaS) and Infrastructure as a Service (IaaS)
Metadata	ISO19115, UncertML
Discovery	CSW, ebRIM, SOA
Portrayal	ISO19117, Styled Layer Descriptor (SLD)

	and Symbol Encoding (SE)
Application schema	GML, profiles, and subsetting tools
Data quality / uncertainty modeling and representations:	UncertML, SensorML, O&M, ISO 19115 and 19115 part 2. ISO 19113, 19114, 19138 provide quality requirements
Data integration/conflation:	Conflation rules, WCPS, WPS, WFS-G, OLS Geocoder,
Spatial-Temporal-Semantic analytics:	O&M, SensorML, UncertML, Event-PatternML, OWL, WPS
Visualizing, linking, organizing, sharing:	GML, CityGML, X3D ISO/IEC 19775, VRML, GeoRSS, KML, LOF, OWS, etc
Automation:	WPS, WCPS, WfCS, Wf-XML, XPDL, BPEL
Grid and Cloud computing	Open Grid Forum and cloud standards by other organization

To enable fusion, one should have certain essential infrastructure capabilities such as:

- Scalable to massive data volumes and complex processing
- Streaming and caching
- Managed and hosted (distributed, off-premise)
- Automated and manage processing and workflows
- Reliable and available
- Security in distributed information systems
- Distributed, virtualized nodes made accessible and interconnected via open Web services and standards-based grid and cloud-computing infrastructures
- Scalable, reliable, cost-effective storage, network and computing capabilities for enabling fusion.

References: [166, 167]

4.5.4 Distributed Intelligence (VTT)

The current generation surveillance systems use distributed intelligence functionality. An important design issue is to determine the granularity at which the tasks can be distributed based on available computational resources, network bandwidth, and task requirements. The distribution of intelligence can be achieved by the dynamic partition of all the logical processing tasks, including event recognition and communications. The dynamic task allocation dilemma is studied through the usage of a computational complexity model for representation and communication tasks [149].

A surveillance task can be separated into four phases, which are 1) event detection, 2) event representation, 3) event recognition, and 4) event query. The detection phase addresses multisource spatiotemporal data fusion for efficient and reliable extraction of motion trajectories from videos. The representation phase revises raw trajectory data to construct hierarchical, invariant, and adequate representations of the motion events. The recognition phase handles event recognition and classification. The query component indexes and retrieves videos that match some query criteria [151].

The key to security is situation awareness. Awareness requires information, which spans across multiple scales of time and space. A security analyst must keep track of “who are the people and vehicles in a space” (identity tracking), “where are the people in a space” (location tracking), and “what are the people/vehicles/objects in a space doing” (activity tracking). The analyst must use historical content to interpret this data. Smart video surveillance systems are capable of enhancing situational awareness over multiple scales of time and space. Currently, the component technologies are evolving in isolation. For instance, face recognition technology handles the identity-tracking challenge, while restricting the subject to be in front of the camera, and intelligent video surveillance technologies offer activity detection capabilities to video streams while disregarding the identity tracking challenge. To offer comprehensive, nonintrusive situation awareness, it is crucial to address the challenge of multiscale, spatiotemporal tracking [152].

The automatic capability to learn and adjust to altering scene conditions and the learning of statistical models of normal event patterns are growing issues in surveillance systems. The learning system offers a mechanism to flag potentially anomalous events through the discovery of the normal patterns of activity and flagging the least probable ones [149].

Due to the availability of more advanced and powerful communications, sensors, and processing units, the architectural choice in the current generation surveillance systems can potentially become extremely variable and flexibly customized to acquire a desired performance level. The system architecture represents a key factor. For instance, different levels of distributed intelligence can result in preattentive detection methods either closer to the sensors or deployed at different levels in a computational processing hierarchy. Another source of variability results from the usage of heterogeneous networks, either wireless or wired, and transmission modalities both in means of source and channel coding and in means of multiuser access techniques. Temporal and spatial coding scalability can be extremely productive for reducing the quantity of information to be transmitted by every camera depending on the intelligence level of the camera itself. Multiple access techniques are a fundamental tool to allow a significant amount of sensors to share a communication channel in the most efficient and robust way [149].

4.5.5 Mobile Surveillance and Positioning (VTT, Roger-GPS)

Currently, the development of an automated surveillance system based on mobile multifunctional robots is an active research area. Mobility and multifunctionality are generically adopted to reduce the amount of sensors required to cover a given region. Mobile surveillance units such as mobile robots can be organized in teams, which results in intelligent distributed surveillance over considerable areas. Several worldwide projects attempt to develop completely or semiautonomous mobile security systems [154].

Examples of mobile surveillance unit solutions include the iBot platform, which uses camera equipped mobile robots that move on a planned path. The robots can be remotely controlled by a centralized controller server in response to live video images they capture [155]. In another example Liu et al. present an unmanned water vehicle (UWV), which performs automatic maritime visual surveillance. The UWV mobile platform is equipped with a Global Positioning System (GPS) device and a high resolution 360 degree omnicamera. Targets are detected with a saliency-based model and adaptively tracked with through selective features. Each target is geo-registered to a longitude and latitude coordinate. The target geo-location and appearance information is then transmitted to the fusion sensor, where the target location and image is displayed on a map [156].

Despite GPS being a sophisticated solution to the location discovery process, it has multiple network dilemmas. First, GPS is expensive both in terms of hardware and power requirements. Second, GPS requires line-of-sight between the receiver and the satellites. It does not function well when obstructions, such as buildings, block the direct “view” of the satellites. Locations can be calculated by trilateration. For a trilateration to be successful, a node needs to have at least three neighbors who already are aware of their positions [157].

The analysis and fusion of different sensor information requires mapping observations to a common coordinate system to achieve situational awareness and scene comprehension. Availability of mapping capabilities enables critical operational tasks, such as the fusion of multiple target measurements across the network, deduction of the relative size and speed of the target, and the assignment of tasks to Pan, Tilt, Zoom (PTZ) and mobile sensors. This presents the need for automated and efficient geo-registration mechanism for all sensors. For instance, target observations from multiple sensors may be mapped to a geodetic coordinate system and then displayed on a map-based interface [158].

5. CONTEXT AND KNOWLEDGE MANAGEMENT FEASIBILITY ANALYSIS

In this chapter, the feasibility of the planned SPY context and knowledge management features and design are analyzed against the current state of the art.

5.1 FEASIBILITY ANALYSIS PROCESS

Considering the SPY framework system specification, the feasibility of the SPY context and knowledge management system should be investigated. The level of feasibility is determined by analyzing the state of the art data presented in chapter 4 against the planned SPY framework design, as presented in e.g. the WP3 system specification output.

The analysis will consider the technical aspects presented in chapter 4: different techniques and features of video and image processing related to surveillance, as well as decision making in a distributed surveillance system including exploitation of multiple and multi-modal sensors.

The following viewpoints should be considered in the analysis, regarding all technical sub-sections and aspects of the component design:

- Technological feasibility
- Economical feasibility
- Operational feasibility
- Schedule feasibility

5.2 FEASIBILITY ANALYSIS RESULTS

This chapter details the detailed conclusions of feasibility regarding the proposed context and knowledge management component design and features, versus the state of the art data.

5.2.1 Feasibility in Image Processing (IEF)

The IEF was an actor of the LOVe project (Logiciel d'Observation des Vulnérables 2006-2009) supported by the "Pôle de Compétitivité" System@tic Paris Région. The aim of the project was to develop an embedded in-vehicle system for real-time detection and tracking of pedestrians and obstacles. Although largely perfectible, the proposed algorithm based on stereovision has provided quite satisfying results. The method relies on: the computation of the depth map, the extraction of regions of uniform depth (by the UV disparity method), the association of the detections to define their trajectories. A pedestrian recognition method has been proposed by an other partner (based on AdaBoost). Unfortunately, the whole application does not perform in real-time on a single embedded architecture. However, in the context of the SPY project, it could be considered that some of the algorithms are executed from a central server in the supervisor office.

While some of the algorithms proposed by IEF either have to be extended or require development, some other algorithms have been applied and tested in other applicative contexts. Therefore their performances in the context of SPY have to be studied. The feasibility depends mainly on:

- the quality of the sensors which could affect object tracking and feature matching. For example, the kernel-based methods usually perform exclusively on color images sequences since the object is represented by a global histogram, which is not always discriminant enough in grey-level sequences. The performances of the feature point matching is sensitive to the local acquisition noise.
- the opportunity to use a binocular sensor. Indeed, when the vehicle is moving, the use of stereovision can hugely facilitate the obstacles detection as well as the motion analysis (for stabilization for example).
- the complexity of the scenarios finally chosen. As an example, IEF has to study the feasibility of its tracking algorithm when a group of people is considered and not only one person or object.

5.2.2 Feasibility in Object Detection and Localization (ENSTA)

In Section 4.1, we presented a partial state-of-the-art of the object visual modelling, learning, and localizing methods based on video analysis. The selection of the presented works was made on feasibility criteria with respect to (1) the mobile context: the camera is moving and embedded in a vehicle, and (2) the real-time constraint: the run-time part of the methods, i.e. detection and localization, has to be done in real-time in the embedded system. The feasibility analysis in these topics is then essentially part of Section 4.1. We simply recall here feasibility criteria and constraints that we considered or shall consider in the continuation of the project.

- Real-time detection and localization. The most computationally intensive part of detection generally lies in the application of a collection of image filters, which is at the basis of most visual representations. Thanks to its high level of regularity, however, this process can be much accelerated, using fine grained data parallelism (GPU or vector parallelism extensions), or recursion and decomposition adapted to coarse grained parallelism (multi-core CPU). A reduction of the number of filters can be done, at the cost of degrading the detection performance, generally regarding scale invariance. As emphasized in Section 4.1, another important way of improving the real-time performance is to reduce the data support by discarding rapidly irrelevant zones, using cascade (Sec. 4.1.5.2) or context hints (Sec. 4.1.4).
- Video quality. The quality of video images is related to the camera resolution and acquisition rate. But a higher quality implies a higher processing power. The embedding within a rapidly moving vehicle is another difficulty: motion blurring and illumination variations, among others, are expected. These problems may not be redhibitory for object detection, but obviously, have to be considered early by (1) using as representative videos as possible in the design of the algorithms and (2) using image data from the final system camera in the (off-line) modelling and learning phase.

5.2.3 Hardware Platform for IP Camera (EOLANE)

The platform supporting the IP camera should take into account the environmental constraints described in section 4.4.2.

The iMX51 / 53 processor family designed by Freescale is based on the ARM Cortex-A8 core processor and is generally used in MIDs. This family processor is well known from EOLANE and designing the SPY camera with this processor allows to take advantage of EOLANE experience on it and to limit the risks from a delay point of view.

The SPY camera is running under Linux. Development is done under Open Embedded, with source code in C written, therefore no SDK is required.

The iMX51 / 53 processor is dedicated to the multimedia market and includes video, and image resources. For this reason, it includes the Neon extension. The Video Processing Unit performs the video encoding and decoding by hardware, therefore, emerging compression algorithms cannot be integrated. Supported video formats are MPEG-4, H264 and MJPEG. The same video native flow CCIR656 (or BT.656) can be encoded simultaneously in multiple formats. That means that it could be H264 encoded for recording and MJPEG encoded for analysis for example. A library delivers the encoded flow in a standard form. It is reasonable to think that it is enough standard to allow each partner to develop its analysis software on its own platform and then perform the integration on the EOLANE platform without too much work and source modification.

The minimum configuration is based on iMX51 processor with a capacity of 512Mo for RAM, 512Mo for Flash and a processor clock of 800 Mhz. A few sensors (at least 2) can be supported. If required, an additional processor board could be added to the system to increase the processing power.

5.2.4 Feasibility in Distributed Multi-Sensor Decision Making (VTT)

For efficient exlpoitation of multiple sensors and data fusion in a multi-modal sensor environment, sensor diversity poses a challenge. The openness and adaptability of the context and knowledge management component, especially regarding its interfaces, must be given high priority in the architectural specification and design phases. This helps ensure the integration and addition of different types of sensors is as easy as possible. Sensor integration and information adaptation can be further eased by careful definition of common meta-data types for different sensor input. Asynchronicity of the sensors and their activity must also be taken into consideration, by finding the best solutions for scheduling and parallel processing for processing multiple incoming data types.

A crucial aspect in defining an architecture for the context and knowledge management component is the distribution of processing and analysis functionality. Optimal distribution of context management capabilities needs to be determined based on the computational power of the mobile surveillance units, estimated network bandwidth and other requirements from the SPY use cases. As the SPY framework is composed around a wireless network of a potentially large number of mobile sensors, network issues are especially important to consider. Performing at least a basic level of context analysis and event recognation from raw sensor data already within the mobile unit, instead of transmitting all raw data on the network, will help avoid network cognestion issues.

Having embedded context analysis and decision making on the mobile surveillance units also makes it possible to address the issue of situation awareness by further processing input from different detection algorithms usually working in isolation, as described in section 4.5.4.

The applicability of GPS technology for positioning has been demonstrated in the context of modern surveillance systems. The technology does however possess technological challenges which need to be taken into consideration in the context and knowledge analysis process. Effort should be made to maximize the capabilities of positioning and tracking when faced with areas containing potential obstructions. A possible way for this would be to enable the position-aware context software to take use of e.g. GPS repeaters improving coverage in difficult spaces.

On the other hand, hardware and power requirements of GPS should not be a major issue in the SPY project context. As vehicles such as police cars are considered the primary mobile surveillance units, ensuring that sufficient processing capabilities and power for tracking are included should be fairly easy compared to smaller and more resource-limited mobile units.

5.3 FEASIBILITY ANALYSIS CONCLUSIONS

This chapter gives an overall conclusion on the feasibility analysis results, stating the overall perceived feasibility of the context and knowledge management component against the current surveillance system state of the art.

Performance of complex processing on an embedded system is a potential challenge in several aspects of analysis, from image processing algorithms to multi-sensor data fusion. However, the networked SPY architecture provides the possibility to distribute processing and to execute certain features on the centralized supervisor side instead of embedded sensors on mobile surveillance units. Hardware upgrades to the mobile IP camera platform are also possible if processing power poses a serious issue.

Distribution of processing over the SPY network may also help increase the feasibility of several other aspects such as complex event processing, although network bandwidth and performance also provide some limitations. Care should be taken in designing the management components to avoid neither embedded processing nor network issues becoming serious bottlenecks.

To ensure the feasibility of video and image algorithms, visual data as representative to the final SPY platform as possible should be used already in the design phase. To overcome potential problems with hardware constraints, it should be ensured during integration planning that different analysis software, algorithms and sensor types are able to operate together on the demonstrator platform. Furthermore, several technologies and algorithms have proved to be working on other environments, but their feasibility on the SPY platform and scenarios remains to be studied. Factors effecting the final feasibility consist of e.g. sensor quality and sensor features.

Feasibility analysis of different context and knowledge management aspects has identified some potential challenges, but also methods to avoid and overcome them. Overall, no serious feasibility issues have been discovered.

6. REFERENCES

[1] Koenderink, J., & Van Doorn, A. (1987). Representation of local geometry in the visual system. Biological Cybernetics, 55, 367-375.

[2] Schmid, C., & Mohr, R. (1997). Local grayvalue invariants for image retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(5), 530-534.

[3] Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., & Navab, N. (2010). Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects. CVPR. San Francisco, California (USA): IEEE.

[4] Gavrila, D., & Philomin, V. (1999). Real-Time Object Detection for "Smart" Vehicles. ICCV (pp. 87-93). Kerkyra - Greece: IEEE.

[5] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., & Poggio, T. (2007). Robust Object Recognition with Coretx-Like Mechanisms. IEEE Trans. on Pattern Analysis and Machine Intelligence, 411-426.

[6] Viola, P., & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. CVPR (pp. 511-518). IEEE.

[7] Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2007). TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. Int. Journal of Computer Vision, 81(1), 2-23.

[8] Micusik, B., & Kosecka, J. (2009). Semantic Segmentation of Street Scenes by Superpixel Co-Occurrence and 3D Geometry. ICCV Workshop on Video-Oriented Object and Event Classification. IEEE.

[9] Mikolajczyk, K., & Schmid, C. (2004). Scale and Affine invariant interest point detectors. Int. Journal of Computer Vision, 60(1), 63-86.

[10] Lowe, D. (2004). Distinctive image features from scale-invariant key- points. Int. Journal of Computer Vision, 60(2), 91-110.

[11] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF : Speeded Up Robust Features. European Conference on Computer Vision.

[12] Matas, J., Chum, O., Urban, M., & Pajdla, T. (2002). Robust wide-baseline stereo from maximally stable extremal regions. British Machine Vision Conference, (pp. 384-393). Cardiff.

[13] Murphy, K., Torralba, A., Eaton, D., & Freeman, W. (2006). Object detection and localization using local and global features. In Toward Category-Level Object Recognition (pp. 382-400). LNCS.

[14] Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. European Conference on Computer Vision (ECCV), (pp. 1-15).

[15] Pele, O., & Werman, M. (2010). The Quadratic-Chi Histogram Distance Family. ECCV.

[16] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L.Wixson, “A system for video surveillance and monitoring,” Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep., CMU-RI-TR-00-12, 2000.

[17] AVITRACK (Aircraft Surroundings, Categorised Vehicles & Individuals Tracking for apRon's Activity Model Interpretation & ChecK) Project, www.avitrack.net

[18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder:Real-time Tracking of the Human Body,” IEEE Trans. on Patt. Anal. and Machine Intell., vol. 19, no. 7, pp. 780-785, 1997.

[19] I. Haritaoglu, D. Harwood and L. S. Davis. “W4: Real - Time Surveillance of People and Their Activities”, IEEE Trans. on Patt. Anal. and Machine Intell, vol. 22, no. 8, pp. 809–830, Aug 2000.

[20] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis, “Background and foreground modeling using non-parametric kernel density estimation for visual surveillance”, Proc. of the IEEE, vol. 90, pp. 1151-1163, 2002.

[21] C. Stauffer and W. E. L. Grimson. “Learning patterns of activity using real-time tracking”, IEEE Trans. on Patt. Anal. and Machine Intell, vol. 22, no. 8, pp. 747–757, Aug 2000.

[22] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 34, no. 3, pp. 334-352, 2004

[23] A. J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In Proc. of Workshop Applications of Computer Vision, pages 129–136, 1998.

[24] L. Wixson and A. Selinger. Classifying moving objects as rigid or non-rigid. In Proc. DARPA Image Understanding Workshop, 1998

[25] C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In Proc. of IEEE Int. Conf. on Intelligent Vehicles, pages 241–246, Germany, October 1998.

[26] T. Brodsky et al. Visual Surveillance in Retail Stores and in the Home, chapter 4, pages 51–61. Video-Based Surveillance Systems. Kluwer Academic Publishers, Boston, 2002.

[27] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis and applications. In IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 8, pages 781–796, 2000.

[28] A. J. Lipton, “Local application of optic flow to analyze rigid versus nonrigid motion,” in Proc. Int. Conf. Computer Vision Workshop Frame-Rate Vision, Corfu, Greece, 1999.

[29] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection” IEEE Trans. Image Process., vol. 13, no. 11, pp. 1459–1472, Nov. 2004

[30] B. Horn and B. Schunk. “Determining optical flow”. Artific. Intell. 17, 185–203, 1981.

[31] B. Lucas, T. Kanade, “An iterative image registration technique with an application to stereo vision”, Proc. Image Understanding Workshop, 1981

[32] F. El Baf, T. Bouwans, B. Vachon “Fuzzy foreground detection for infrared videos”, IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), 2008

[33] Y. Fang, K. Yamada,Y. Ninomiya,B. Horn,I. Masaki, “A shape independent-method for pedestrian detection with far-infrared images”, IEEE Transactions on Vehicular Technology, 2004

[34] E. Goubet, J. Katz, F. Porikli, “Pedestrian tracking using thermal infrared imaging”, Proc. SPIE, Vol. 6206,2006

[35] Y. Yoo, T.S. Park “A moving object detection algorithm for smart cameras”, IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), 2008

[36] Y.L. Tian, A. Hampapur, “Robust salient motion detection with complex background for real-time video surveillance”, Proc. IEEE Workshop on Motion and Video Computing (WACV/MOTION’05), Vol. 2, 2005

[37] N. Vaswani, A.K. Agrawal, Q. Zheng, R. Chellappa, “Moving object detection and compression in IR sequences”, www.cfar.umd.edu/~namrata/Chapter5.pdf

[38] Chris Harris and Mike Stephens. \A Combined Corner and Edge Detector". Proc. of The Fourth Alvey Vision Conference, Manchester, pp. 147-151. 1988

[39], B.D. Lucas and T. Kanade. An iterative image registration technique. 1981. IJCAI'81, pp 674—679

[40] C. Tomasi and T. Kanade. Detection and tracking of point features. Carnegie Mellon University", 1991.Technical Report CMU-CS-91-132.

[41] Lamberto Ballan, Marco Bertini, Alberto Del Bimbo, Lorenzo Seidenari and Giuseppe Serra Recognizing Human Actions by fusing spatio-temporal appearance and motion descriptors. IEEE ICIP International Conference on Image Processing 2009

[42] D. Comaniciu and P. Meer. Mean Shift Analysis and Applications. Proc_ IEEE ICCV 1999.

[43] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In International workshop on image processing, real-time edge and motion detection (1979)

[44] D. Mumford and J. Shah. Optimal Approximations by Piecewise Smooth Variational Functions and Associated Communications on Pure and Applied Problems 685, 1989.

[45] Y.Deng and B.S. Manjunath Unsupervised Segmentation of Color-Texture Regions in Images and Video. IEEE Trans. on PAMI, vol 22, 2001, p 800-810.

[46] T. Gevers and A.W.M. Smeulders, “Color-based object recognition,” Pattern Recognition, vol. 32, no. 1999, pp. 453–464, 1999

[47] M.Gouiffès. Tracking by combining photometric normalization and color invariants according to their relevance. IEEE International Conference on Image processing 2007.

[48] Kais Siala, Moez Chakchouk, Faten Chaieb, Olfa Besbes, "Moving Shadow Detection with Support Vector Domain Description in the Color Ratios Space," Pattern Recognition, International Conference on, pp. 384-387, 17th International Conference on Pattern Recognition (ICPR'04) - Volume 4, 2004.

[49] L. Sigal and Stan Sclaroff and V. Athitsos. Skin Color-Based Video Segmentation under Time-Varying Illumination. IEEE PAMI 2003.

[50] J. Laneurit, C. Blanc, R. Chapuis, L. Trassoudaine. ”Multisensorial data fusion for global vehicle and obstacles absolute positioning”, in it IEEE Intelligent Vehicles Symposium, Columbus, OH, USA, June 9-11, 2003.

[51] A. I. Comport, E. Malis and P. Rives, ”Real-Time Quadrifocal Visual Odometry”, International Journal of Robotic Research, Special Issue on Robotic Vision, pp. 486-492, 2010.

[52] T.A. Williamson, A High Performance Stereo Vision System For Obstacle Detection, PhD Dissertation, Robotics Institute Carnegie Mellon University, Pittsburg, 1998

[53] N. Hautiere, R. Labayrade, M. Perrollaz and D. Aubert, Road Scene Analysis by Stereovision a Robust and Quasi-Dense Approach, International Conference on Control, Automation, Robotics and Vision, pp 1-6, 2006

[54] K. Berndt, A. Geiger and H. Lateggahn, ”Visual Odometry Based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme”, IEEE Intelligent Vehicle Symposium, pp 486-492, 2010.

[55] R. Hartley. In defense of the 8-point algorithm. In IEEE International Conference on Computer Vision, 1995, pp 1064–1070.

[56] Q.T. Luong, O.D. Faugeras. “Camera calibration, scene motion and structure recovery from point correspondences and fundamental matrices”. International Journal of Computer Vision 22(3), 1997, 261–289.

[57] E.C. Hildreth. “Recovering heading for visually-guided navigation”. Vision Research 32(6), 1992, 1177–1192.

[58] W. MacLean, A. Jepson, R. Frecker. „Recovery of egomotion and segmentation of independant object motion using the em algorithm”. In: British Machine Vision Conference, 1994, pp 13–16

[59] M. Irani, B. Rousso, S. Peleg. “Recovery of egomotion using region alignment”. IEEE Transactions on Pattern Analysis Machine Intelligence 19(3), 1997, 268–272

[60] G.P. Stein, O. Mano, A. Shashua. “A robust method for computing vehicle egomotion”. In: IEEE Intelligent Vehicles Symposium, 2000, pp 362–368

[61] S. Heinrich, Fast Obstacle Detection Using Flow/Depth Constraint, IEEE Intelligent Vehicles Symposium, pp 658-665, vol. 2, June 2002

[62] U. Franke, C. Rabe, H. Badino and S. Gehrig, 6D-Vision : Fusion of Stereo And Motion For Robust Environment Perception, Lecture Notes in Computer Science - Pattern Recognition, pp 216-223, vol.3663, 2005

[63] D. Demirdjian and T. Darrell, Motion Estimation From Disparity Images, IEEE International Conference on Computer Vision, pp 213-218, vol. 1, July 2001

[64] H. Badino, A Robust Approach For Ego-Motion Estimation Using A Mobile Stereo Platform, Lecture Notes in Computer Science -Complex Motion, pp 198-208, vol. 3417, 2007

[65] A. Taludker and L. Matthies, Real-Time Detection of Moving Objects from Moving Vehicles Using Dense Stereo and Optical Flow, IEEE International Conference on Intelligent Robots and Systems, pp 315-320, Sendai, Japan, Sept. 2004

[66] A. Bak, S. Bouchafa, D. Aubert. "Detection of independent-moving objects through stereo-vision and ego-motion extract". IEEE Intelligent Vehicules Symposium (IV), pp. 863_870, San Diego CA, june 21-24, 2010.

[67] J. Biosca, J. Lerma. “Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods”. ISPRS Journal of Photogrammetry and Remote Sensing 63 (1), 1008, 84-98.

[68] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter. “The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design”. 3D research, 2011, vol. 2(2), pp.1-13.

[69] K. R. Teixeira Aires, Kelson Rômulo Teixeira Aires. « A Plane Segmentation System Based on Affine Homography and Optical Flow”. 23rd SIBGRAPI Conference on Graphics, Patterns and Images. Gramado, 2010, Pp. 346 – 352.

[70] K. Schindler. “Generalized use of homographies for piecewise planar reconstruction”. In: Proceedings of the 13th Scandinavian Conference on Image Analysis, Gothenburg, Sweden, 2003, pp. 470–476.

[71] S. Bouchafa and B. Zavidovique. “C-Velocity: A Cumulative Frame to Segment Objects From Ego-Motion”. Pattern Recognition and Image Analysis, vol 19, 2009, 583-590.
Object and Feature Tracking

[72] Kalman, R.E., “A New Approach to Linear Prediction Problems”, Transactions of the ASME--Journal of Basic Engineering, pp. 35-45, March 1960.

[73] D. Comaniciu, V. Ramesh, P. Meer, “Kernel-based object tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5), pp564-575, 2003.

[74] Benjamin Gorry, Zezhi Chen, Kevin Hammond, Andy Wallace, and Greg Michaelson Using Mean-Shift Tracking Algorithms for Real-Time Tracking of Moving Images on an Autonomous Vehicle Testbed Platform. World Academy of Science, Engineering and Technology 34 2007

[75] S. Rastegar, M. Bandarabadi, Y. Toopchi, and S. Ghoreishi, “Kernel based object tracking using metric distance transform and svm classifier,” Aus. Jour. of Basic and Applied Science, vol. 3, no. 3, pp. 2778–2790, 2009.

[76] S.T. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking.,” in Computer Vision and Pattern Recognition, 2005, pp. 1158–1163.

[77] Q. Zhao and H. tao, “A motion observable representation using color correlogram and its application to tracking,” Computer Vision and Image Understanding, vol. 113, pp. 273–290, 2009.

[78] M. Gouiffès and F. Laguzet and L. Lacassagne. Connectedness degree for Mean-shift tracking. ICPR 2010.

[79] F. Porikli, O.Tuzel, and P. Meer, “Covariance tracking using model update based on lie algebra,” in IEEE Computer Vision and Pattern Recognition, 2006, pp. 728–735.

[80] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, “Online Multi-Person Tracking-by-Detection from a Single, Uncalibrated Camera,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99, Dec. 2010.

[81] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE, 2005, pp. 886-893.

[82] B. Wu and R. Nevatia, “Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Object Detection,” Computer Vision, 2007. ICCV 2007, pp. 1-8.

[83] P.K. Sharma, C. Huang, and R. Nevatia, “Evaluation of People Tracking , Counting and Density Estimation in Crowded Environments,” Proc. of the 11th IEEE International Workshop on PETS, Miami: 2009, pp. 39-46.

[84] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, “PROST: Parallel robust online simple tracking,” Conference on Computer Vision and Pattern Recognition, San Francisco: IEEE, 2010, pp. 723-730.

[85] J. Yang, P. Vela, Z. Shi, and J. Teizer, “Probabilistic multiple people tracking through complex situations,” Proc. of the 11th IEEE International Workshop on PETS, Miami: 2009, pp. 79-86.

[86] B. Babenko, M.-H. Yang, and S. Belongie, “Visual Tracking with Online Multiple Instance Learning,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, Dec. 2010.

[87] C.-H. Kuo, C. Huang, and R. Nevatia, “Multi-target tracking by on-line learned discriminative appearance models,” CVPR 2010, pp. 685-692.

[88] E. Arbel, H. Hel-Or. Shadow Removal Using Intensity Surfaces and Texture Anchor Points. IEEE TPAMI. 33(6): 1202-1216, 2011.

[89] N. Martel-Brisson and A. Zaccarin. Learning and Removing Cast Shadows through a Multidistribution Approach. IEEE Trans. PAMI. 29(7): 1133–1146, 2007.

[90] C.H. Huang and R.C. Wu. An Online Learning Method for Shadow Detection. In Proc. Of PSIVT'10 pp 145 - 150.

[91] H. Veeraraghavan, N.P. Papanikolopoulos, and P. Schrater. Learning dynamic event descriptions in image sequences. In Conference on Computer Vision and Pattern Recognition, pages 1–6, 2007.

[92] S. Wu, B.E. Moore, M. Shah. Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. CVPR 2010, pp.2054-2060, 13-18 June 2010

[93] Indiana Forensic Institute. Advanced in-car video system. 2011.

[94] A. Jazayeri, H. Cai, M. Tuceryan, and J.Y. Zheng. Smart video systems in police cars. In Proceedings of the international conference on Multimedia, pages 807–810. ACM, 2010.

[95] W. Zajdel, JD Krijnders, T. Andringa, and DM Gavrila. Cassandra: audio-video sensor fusion for aggression detection. In Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on, pages 200–205. IEEE, 2007.

[96] Neeti A. Ogale. A survey of techniques for human detection from video. Master's thesis, University of Maryland, 2006.

[97] Y. Wu, T. Yu, and G. Hua. A statistical field model for pedestrian detection. 2005.

[98] D.M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1408–1421, 2007.

[99] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. 2005.

[100] E. Seemann, M. Fritz, and B. Schiele. Towards robust pedestrian detection in crowded image sequences. In IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07, pages 1–8, 2007.

[101] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. Proc.
CVPR, 1:511–518, 2001.

[102] Q. Zhu, S. Avidan, M.C. Yeh, and K.T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In CVPR, volume 1, page 4. Citeseer, 2006.

[103] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 1:1713–1727, 2008.

[104] M. J. Jones and D. Snow. Pedestrian detection using boosted features over many frames. In ICPR, pages 1–4, 2008.

[105] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. BMVC, 2009.

[106] N. Dalal. Finding people in images and videos. Doktorarbeit, Institut National Polytechnique

de Grenoble, 2006.

[107] X. Wang, T.X. Han, and S. Yan. An HOG-LBP Human Detector with Partial Occlusion Handling. In IEEE International Conference on Computer Vision, 2009.

[108] I.P. Alonso, D.F. Llorca, MA Sotelo, L.M. Bergasa, P.R. de Toro, J. Nuevo, M. Ocana, and M.A.G. Garrido. Combination of feature extraction methods for SVM pedestrian detection, IEEE Transactions on Intelligent Transportation Systems, 8(2):292–307, 2007.

[109] L. Zhao and C.E. Thorpe. Stereo-and neural network-based pedestrian detection. IEEE Transactions on Intelligent Transportation Systems, 1(3):148–154, 2000.

[110] DM Gavrila, J. Giebel, and S. Munder. Vision-based pedestrian detection: The protector system. In Intelligent Vehicles Symposium, 2004 IEEE, pages 13–18. IEEE, 2004.

[111] D.M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. International Journal of Computer Vision, 73(1):41–59, 2007.

[112] M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata. Pedestrian detection with convolutional
neural networks. In IEEE Intelligent Vehicles Symposium, 2005. Proceedings, pages 224–229, 2005.

[113] K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition.
Neural networks, 1(2):119–130, 1988.

[114] Y. Mu, S. Yan, Y. Liu, T. Huang, and B. Zhou. Discriminative local binary patterns for human detection in personal album. CVPR 2008, 1:1–8, 2008.

[115] X. He, J. Li, Y. Chen, Q.Wu, andW. Jia. Local binary patterns for human detection on hexagonal
structure. In Proceedings of the Ninth IEEE International Symposium on Multimedia, pages 65–71. IEEE Computer Society, 2007.

[116] J. Yao and J.M. Odobez. Fast human detection from videos using covariance features. 1, 2008.

[117] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis. Human Detection Using Partial Least Squares Analysis. In IEEE International Conference on Computer Vision, 2009.

[118] R. Cutler and L. Davis. Real-time periodic motion detection, analysis, and applications. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, pages 326–332, 1999.

[119] C.Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. Lecture Notes in Computer Science, 5096:82–91, 2008.

[120] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. 2009.

[121] M. Enzweiler and D. Gavrila. A multi-level mixture-of-experts framework for pedestrian classification. IEEE transactions on image processing: a publication of the IEEE Signal Processing

Society, 2011.

[122] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.

[123] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. 2010.

[124] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 28(11):1863–1868, 2006.

[125] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2009.

[126] M. Enzweiler and D.M. Gavrila. Monocular pedestrian detection: Survey and experiments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(12):2179–2195, 2009.

[127] T. Gandhi and M.M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. Intelligent Transportation Systems, IEEE Transactions on, 8(3):413–430, 2007.

[128] D. Geronimo, A.M. Lopez, A.D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(7):1239–1258, 2010.

[129] D. Geronimo, A. Sappa, A. Lopez, and D. Ponsa. Adaptive image sampling and windows classification for on-board pedestrian detection. In Proceedings of the International Conference on Computer Vision Systems, Bielefeld, Germany, 2007.

[130] A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In Intelligent Vehicles Symposium, 2004 IEEE, pages 1–6. IEEE, 2004.

[131] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 694–711, 2006.

[132] A. Bensrhair, A. Bertozzi, A. Broggi, A. Fascioli, S. Mousset, and G. Toulminet. Stereo visionbased feature extraction for vehicle detection. In Intelligent Vehicle Symposium, 2002. IEEE, volume 2, pages 465–470. IEEE, 2002.

[133] C. Parisot, J. Meessen, C. Carincotte and X. Desurmont. Real-time road traffic classification using on-board bus video camera. 11th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC), p. 189-196. Beijing, China. October 2008

[134] http://ndi-rs.com/us/talon.

[135] www.fedsig.com/solutions/alpr-mobile.

[136] www.citysynctech.com/.

[137] www.mobile-vision.com/products/alertvu.html.

[138] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos, and E. Kayafas. License plate recognition from still images and video sequences: A survey. Intelligent Transportation Systems, IEEE Transactions on, 9(3):377–391, 2008.

[139] Irwin M. Cohen, Darryl Plecas, and Amanda V. McCormick. A report on the utility of the automated licence plate recognition system in british columbia. Technical report, Scholl of Criminology and Criminal Justice, 2007.

[140] Walter Ziegler, U. Franke, G. Renner, A. Kühnle, Computer Vision on the Road: A Lane Departure and Drowsy Driver Warning System, 1995

[141] Zielke, T., Brauchkmann, M., CARTRACK: computer vision-based car following, ., IEEE Workshop on Applications of Computer Vision, Proceedings, 1992.

[142] ProVida Police car system: http://www.petards.com/emergency_services/provida_2000.aspx

[143] Zehang Sun, George Bebis and Ronald Miller, “On-Road Vehicle Detection: A Review,” IEEE trans. on Pattern Analysis and Machine Intelligence, vol.28(5), May 2006

[144] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, “A Real-Time Precrash Vehicle Detection System,” Proc. IEEE International Workshop Application of Computer Vision, Dec. 2002.

[145] H. Mallot, H. Bulthoff, J. Little, and S. Bohrer, “Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection,” Biological Cybernetics, vol. 64, no. 3, pp. 177-185, 1991.

[146] Jia, Zhenh , “Recent Developments in Vision Based Target Tracking for Autonomous Vehicles Navigation,” Proc. of 9th International IEEE conference on Intelligent transportation sytesm, Toronto, September 17-20, 2006.

[147] Katie Roberts-Hoffman, Pawankumar Hegde, "ARM Cortex-A8 vs. Intel Atom: Architectural and Benchmark Comparisons" University of Texas at Dallas EE6304 Computer Architecture Course Project – Fall 2009

[148] T. Räty, “Survey on Contemporary Remote Surveillance Systems for Public Safety”, IEEE Trans. Systems, Man and Cybernetics, vol. 40 issue 5, pp. 493-515, September 2010, doi: 10.1109/TSMCC.2010.2042446.

[149] C. S. Regazzoni, V. Ramesh, and G. L. Foresti, “Scanning the issue/technology special issue on video communications, processing, and understanding for third generation surveillance systems,” Proc. IEEE, vol. 89, no. 10, pp. 1355–1367, Oct. 2001, doi: 10.1109/5.959335.

[150] M. Valera and S. A. Velastin, “Intelligent distributed surveillance systems: A review,” IEE Proc.-Vis. Image Signal Process., vol. 152, no. 2, pp. 192–204, Apr. 2005, doi: 10.1049/ip-vis: 20041147.

[151] Z. Dimitrijevic, G. Wu, and E. Y. Chang, “SFINX: A multi-sensor fusion and mining system,” in Proc. 2003 Joint Conf. Fourth Int. Conf. Inf., Commun. Signal Process., Dec., vol. 2, pp. 1128–1132, doi: 10.1109/ICICS.2003.1292636.

[152] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, S. Pankanti, A. Senior, C. F. Shu, and Y. L. Tian, “Smart video surveillance: Exploring the concept of multiscale spatiotemporal tracking,” IEEE Signal Process. Mag., vol. 22, no. 2, pp. 38–51, Mar. 2005, doi: 10.1109/MSP.20005.1406476.

[153] J. Chen, Z. Safar, and J. A. Sorensen, “Multimodal wireless networks: Communication and surveillance on the same infrastructure,” IEEE Trans. Inf. Forensics Secur., vol. 2, no. 3, pp. 468-484, Sep. 2007, doi: 10.1109/TIFS.2007.904944.

[154] D. Di Paola, D. Naso, A. Milella, G. Cicirelli, and A. Distante, “Multisensor surveillance of indoor environments by an autonomous mobile robot,” in Proc. 15th Int. Conf.Mechatronics Mach. Vis. Pract. (M2VIP), Dec. 2008, pp. 23–28, doi: 10.1109/MMVIP.2008.474501.

[155] J. N. K. Liu, M. Wang, and B. Feng, “iBotGuard: An internet-based intelligent robot security system using invariant face recognition against intruder,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 35, no. 1, pp. 97–105, Feb. 2005, doi:10.1109/TSMCC.2004.840051.

[156] H. Liu, O. Javed, G. Taylor, X. Cao, and N. Haering, “Omni-directional surveillance for unmanned water vehicles,” presented at the 8th Int. Workshop Vis. Surveill., Marseilles, France, Oct. 2008.

[157] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Worst and best-case coverage in sensor networks,” IEEE Trans. Mobile Comput., vol. 4, no. 1, pp. 84–92, Jan./Feb. 2005, doi: 10.1109/TMC.2005.15(410)4.

[158] K. Shafique, F. Guo, G. Aggarwal, Z. Rasheed, X. Cao, and N. Haering, “Automatic geo-registration and inter-sensor calibration in large sensor networks,” in Smart Cameras. New York: Springer-Verlag, 2009, pp. 245–257.

[159] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, “Distributed embedded smart cameras for surveillance applications,” Computer, vol. 39, no. 2, pp. 68–75, Feb. 2006, doi: 10.1109/MC.2006.55.

[160] L. Snidaro, R. Niu, G. L. Foresti, and P. K. Varshney, “Quality-based fusions of multiple video sensors for video surveillance,” IEEE Trans. Syst., Man, Cybern. – Part B: Cybern., vol. 37, no. 4, pp. 1044–1051, Aug. 2007, doi: 10.1109/TSMCB.2007.895331.

[161] P. K. Atrey, M. S. Kankanhalli, and R. Jain, “Timeline-based information assimilation in multimedia surveillance and monitoring systems,” in Proc. 3rd Int. Workshop Video Surveill. Sens. Netw. (VSSN), Nov. 2005, pp. 103–112.

[162] E. Blasch and S. Plano, “Proactive decision fusion for site security,” in Proc. 8th Int. Conf. Inf. Fusion, Jul. 2005, pp. 1584–1591, doi: 10.1109/ICIF.2005.1592044.

[163] F. Castanedo, M. A. Patricio, J. Garcia, and J. M. Molina, “Robust data fusion in a visual sensor multi-agent architecture,” in Proc. 10th Int. Conf. Inf. Fusion, Jul. 2007, pp. 1–7, doi: 10.1109/ICIF.2007.4408121.

[164] F. Castanedo, M. A. Patricio, J. Garcia, and J. M. Molina, “Extending surveillance systems capabilities using BDI cooperative sensor agents,” in Proc. 4th Int.Workshop Video Surveill. Sens. Netw. (VSSN), Oct. 2006, pp. 131–138.

[165] P. K. Atrey, M. S. Kankanhalli, and R. Jain, “Timeline-based information assimilation in multimedia surveillance and monitoring systems,” in Proc. 3rd Int. Workshop Video Surveill. Sens. Netw. (VSSN), Nov. 2005, pp. 103–112.

[166] Fusion_Standards_Study_Engineering_Report - Open Geospatial Consortium - 2010-03-21 - OGC 09-138

[167] Review of Data Fusion models and Architectures: Towards Engineering Guidelines, Jaime Esteban1, Andrew Starr1, Robert Willetts1, Paul Hannah1, Peter Bryanston-Cross2 1The University of Manchester School of Mechanical, Aerospace and Civil Engineering Sackville Street, Manchester, M60 1QD, UK