RELIABLE APPLICATION SPECIFIC DETECTION OF ROAD USERS WITH VEHICLE ON-BOARD SENSORS

DOCUMENT

DELIVERABLE NUMBER	**D6.11**	DUE DATE	**30/06/2011**
ISSUED BY	**MM**	ACTUAL DATE	**31/01/2012**
CONTRIBUTING WP/TASK	**WP6 / TASK 6.4**	PAGES	**40**
CONFIDENTIALITY STATUS	**PUBLIC**	ANNEXES	**-**

PROJECT

GRANT AGREEMENT NO.	**216049**
ACRONYM	**ADOSE**
TITLE	**RELIABLE APPLICATION SPECIFIC DETECTION OF ROAD USERS WITH VEHICLE ON-BOARD SENSORS**
CALL	**FP7-ICT-2007-1**
FUNDING SCHEME	**STREP**

DELIVERABLE D6.11

ANALYSIS OF OBJECT AND SITUATION REFINEMENT

AUTHORS	**MM**	**D. FONTANA** **D. MARENCO** **P. PORTA**

APPROVAL

WORKPACKAGE LEADER	**MM**	**G. GHISIO**
PROJECT COORDINATOR	**CRF**	**N. PALLARO**

AUTHORISATION

PROJECT OFFICER	**EUROPEAN COMMISSION**	**I. HEIBER**

TABLE OF CONTENTS

1. INTRODUCTION 4
1.1 RELEVANT SOFTWARE WP6 OBJECTIVES 4
1.2 PURPOSE OF THIS DOCUMENT 5
2. ANALYSIS OF FUTURE HIGH LEVEL SOFTWARE 5
2.1 INTRODUCTION 6
2.2 OBJECT REFINEMENT 8
2.2.1 Data fusion 8
2.2.2 Object Segmentation 19
2.2.3 Object Classification 21
2.3 SITUATION REFINEMENT 26
2.3.1 Tracking 26
2.3.2 3D reconstruction 29
2.3.3 Radar Tracking 34
3. CONCLUSIONS 36
4. BYBLIOGRAPHY 37
5. LIST OF FIGURES 39
6. LIST OF ACRONYMS 39

REVISION HISTORY				
VER.	**DATE**	**PAG.**	**NOTES**	**AUTHOR**
1.0	13/10/2010	All	Initial Document, Template, Contents	D. Fontana (MM) D. Marenco (MM) P. Porta (MM)
2,0	19/10/2010	All	Analysis Of Future High Level Software	D. Marenco (MM) P. Porta (MM)
3,0	22/10/2010	All	Object refinement	D. Marenco (MM) P. Porta (MM)
4,0	10/11/2010	All	Object refinement - Classification	D. Marenco (MM) P. Porta (MM)
5,0	22/11/2010	All	Situation Refinement	D. Marenco (MM) P. Porta (MM)
6,0	12/09/2010	All	Situation Refinement – 3D reconstruction	D. Marenco (MM) P. Porta (MM)
7,0	12/09/2010	All	Update	D. Marenco (MM) P. Porta (MM)
8.0	27/12/2011	All	Radar Update	J. Tsahalis (PARAGON)
9.0	31/01/2012	37	Radar Conclusion update	J. Tsahalis (PARAGON)

1. INTRODUCTION

1.1 Relevant software WP6 objectives

In the context of ADOSE **only 'technology-dependent' pre-processing algorithms have been developed:**

a) algorithms implemented into the sensor hardware (e.g. processing pipeline in case of vision sensor like input control, noise removal, image enhancement, output control, …);
b) algorithms on raw data, coming from the sensor hardware, implemented on a PC-based processing which are strictly related to the sensing technology and its demonstration (e.g. 'hot spot' feature extraction in case of low-resolution FIR-camera).

With reference to the software architecture of a typical driver assistance system as defined in ProFusion (PReVENT), **the algorithm development in ADOSE has been only limited to the perception stage and in particular to the sensor and the object refinement** (Figure 1).

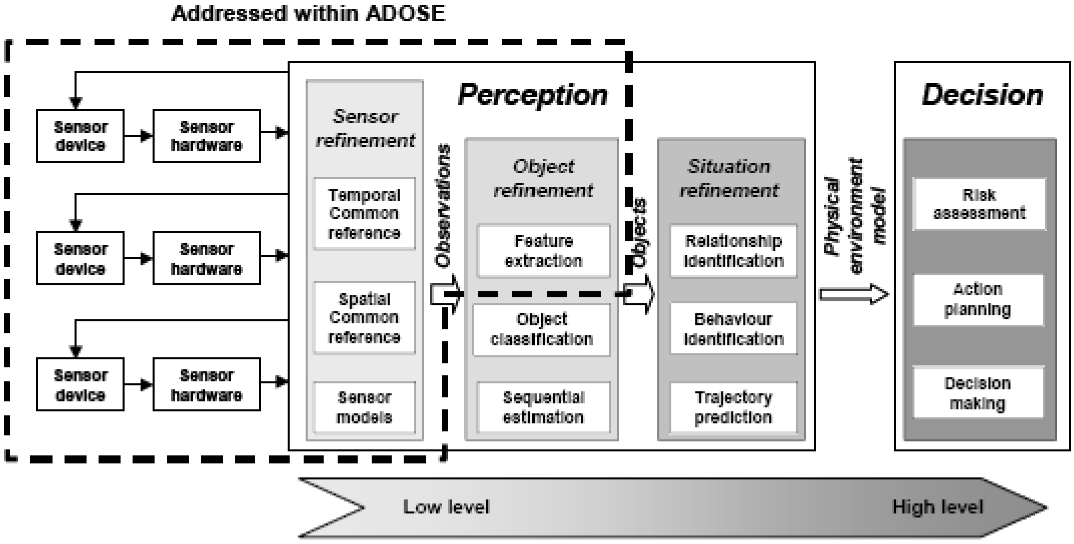


Figure 1: Data processing tasks of a typical driver assistance system

ProFusion guidelines (PReVENT) will be followed, but algorithm developments will not be extended to Sensor Data Fusion.

For the validation phase two demonstrator vehicles have been set-up with the developed sensors:

- CRF vehicle: FIR camera, multi-functional optical sensor and ranging camera.
- AIT vehicle: silicon retina stereo sensor, scanning radar for tag detection and radio communication system.

No high level software, except a medium level algorithm has been designed to implement simplified vehicle detection matching information coming from FIR and NIR cameras.

1.2 Purpose of this document

The purpose of the present document is to present a preliminarily analysis of processing algorithms for object refinement like classification and 3D reconstruction. These activities are planned in the context of Task 6.4.

D6.11 deliverable report proposes a classification process of the detected obstacles around vehicle (like vulnerable road users, other vehicles, barriers, etc.) and 3D model reconstruction of the external scenarios supported by a sensor data fusion approach.

2. ANALYSIS OF FUTURE HIGH LEVEL SOFTWARE

In this task the activity is only limited to a task analysis without real software implementation in order to avoid overlapping with ongoing EU projects.

ADOSE addresses five breakthrough sensing technologies, with the goal to improve the current state-of-the-art in terms of performance, reliability and costs. In the other hand, ADOSE proposes new technologies to prevent / reduce the number of road accidents. For this reason, the study will concern about vehicles and vulnerable road users.

The deliverable is specially focused on the refinement phase analysis, composed by:

- Object refinement with
  - Object classification;
  - Sequential estimation;

- Situation refinement with
  - Relationship Identification;
  - Behaviour Identification;
  - Trajectory prediction.

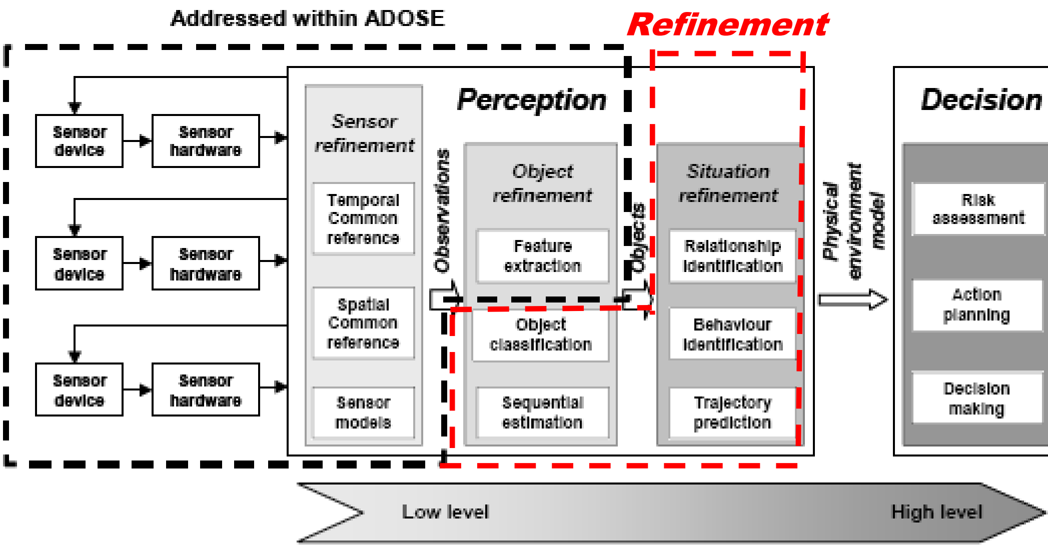


Figure 2: Refinement phase

2.1 Introduction

Object refinement and situation are obtained through an articulate process composed by different elaboration phases, made on data provided by sensor refinement. Each elaboration phase is connected to a particular subset of the others, and every connection is associated to an exchange of data so that a phase can pass information as output or can receive useful information for its own processing.
The elaborations to make on sensor refined data can be expressed as follow:

1) Feature extraction
2) Object segmentation
3) Object classification
4) Situation refinement based on relationship and behaviour identification
5) Tracking based on sequential estimation

Sensor refined data provided by ADOSE sensors are subjected to a data fusion with additional application-wise features that is done accordingly tracking information.
Note that tracking information is useful also for object segmentation process, for object classification and situation refinement.
After the data fusion process between different sensors data and the application features, an object segmentation have to be done, considering information provided by the situation refinement (which returns in output data that can be used also for the tracking). Subsequently object classification and tracking are performed.
The whole process is shown in the figure 3:

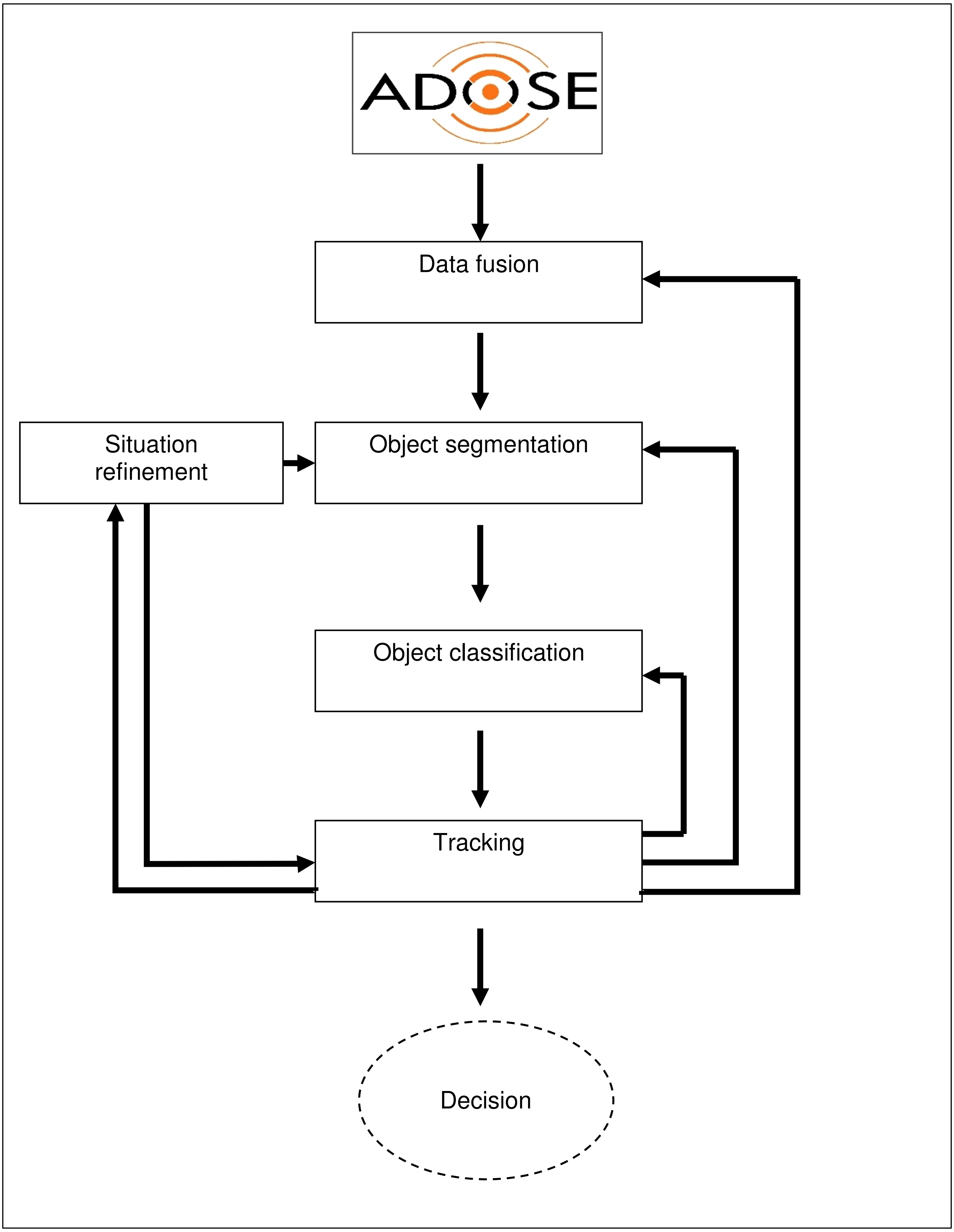


Figure 3: Software - High level

The figure 3 shows the path of information, starting from data provided by low level software applied to ADOSE sensors and ending with the final decision.

2.2 Object refinement

2.2.1 Data fusion

The whole process can be done taking advantage by the combination of different types of information provided by multiple sensors. This technique is called data fusion and it aims to achieve improved accuracies and more specific inferences than could be achieved by the use of a single sensor alone.
Data fusion concept can be understood considering the scheme in the figure 4.

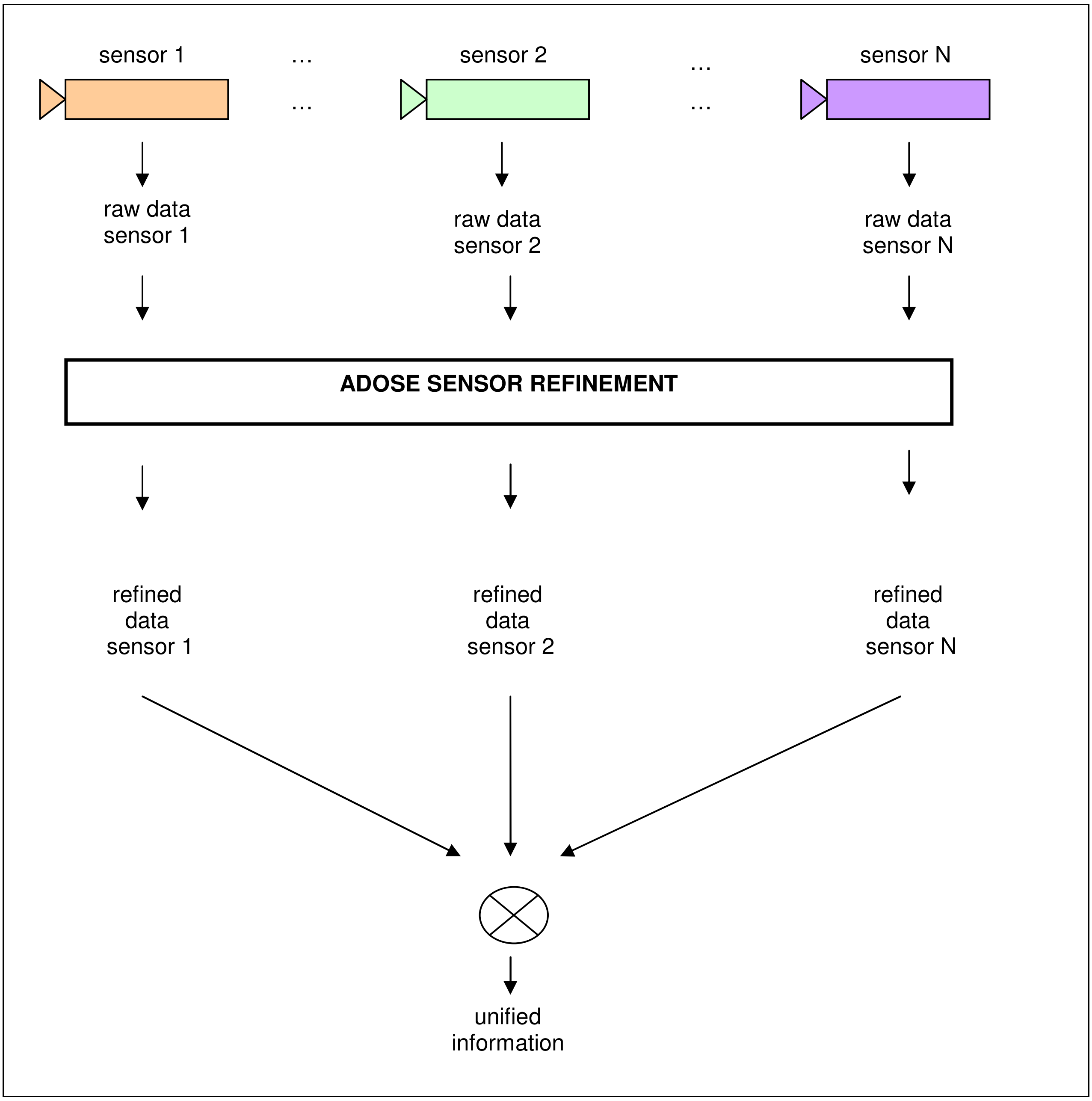


Figure 4: Data fusion concept

In ADOSE project, the low level data fusion advantages are great in feature merging:
this technique has been allowed considering both information regarding temperature in different zones of sensor images and symmetry (as explained D 6.5 Section 5.4.4).
An example of data fusion using FIR and NIR is shown in the figure 5.

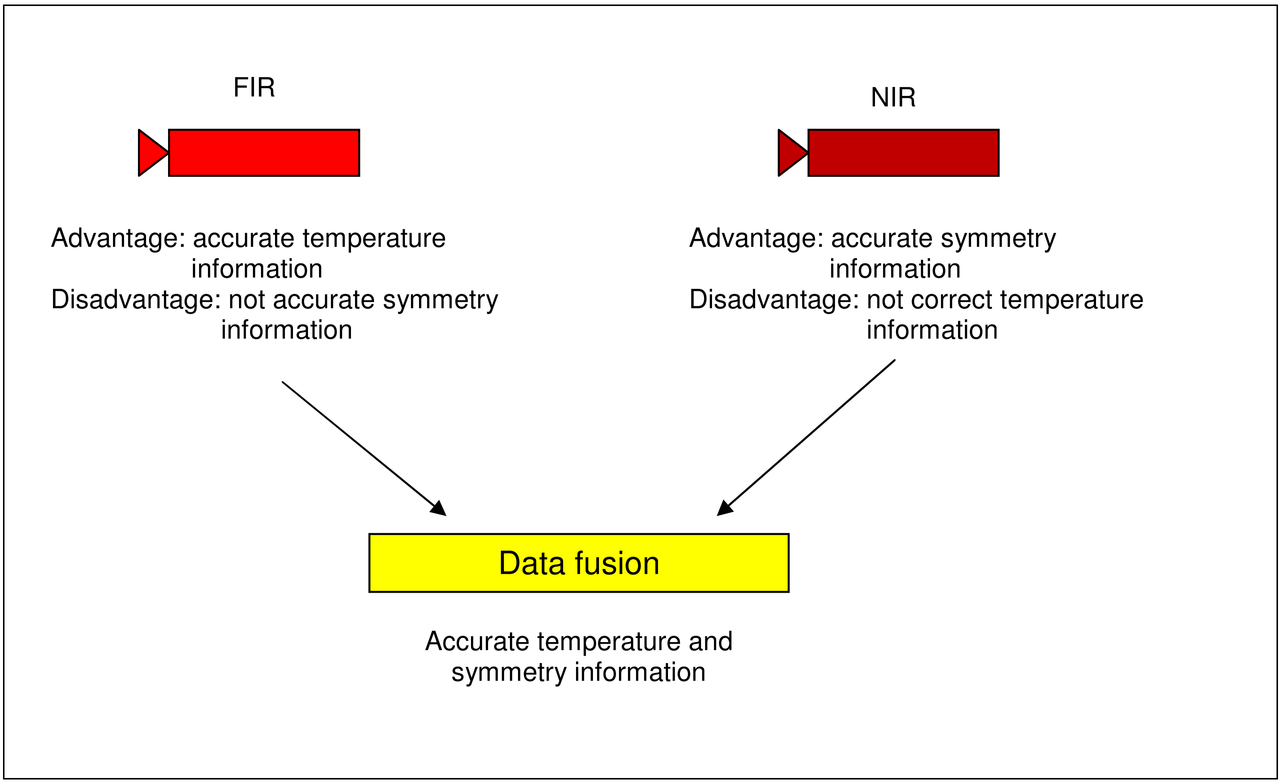


Figure 5: Data fusion example

This scheme represents a particular instance of the general model seen in the previous image. It is possible to see how data fusion allow us to get over the information extraction difficulties of the specific sensors; in fact, FIR does not give an accurate information for symmetry computation while NIR information about temperature of the objects in the image are inadequate. Data fusion is strictly connected to feature extraction; in fact, there are many applications related to different features structures.

Another case of low level data fusion in ADOSE is the combination of different radar inputs by the Radar ECU. Both conventional and harmonic returns with different properties complement to each other to improve the radar ECU output (as shown in figure 5b).

Radar low level fusion may expand to incorporate LRR & SRR sensors. Each radar sensor may have a different Field-of-View and discretization grid, but they all share the same hardware & software techniques and they provide compatible information. The Radar ECU can handle these differences and associate/merge the detected RUs between different radar sensors, creating a new FoV with the compiled information.

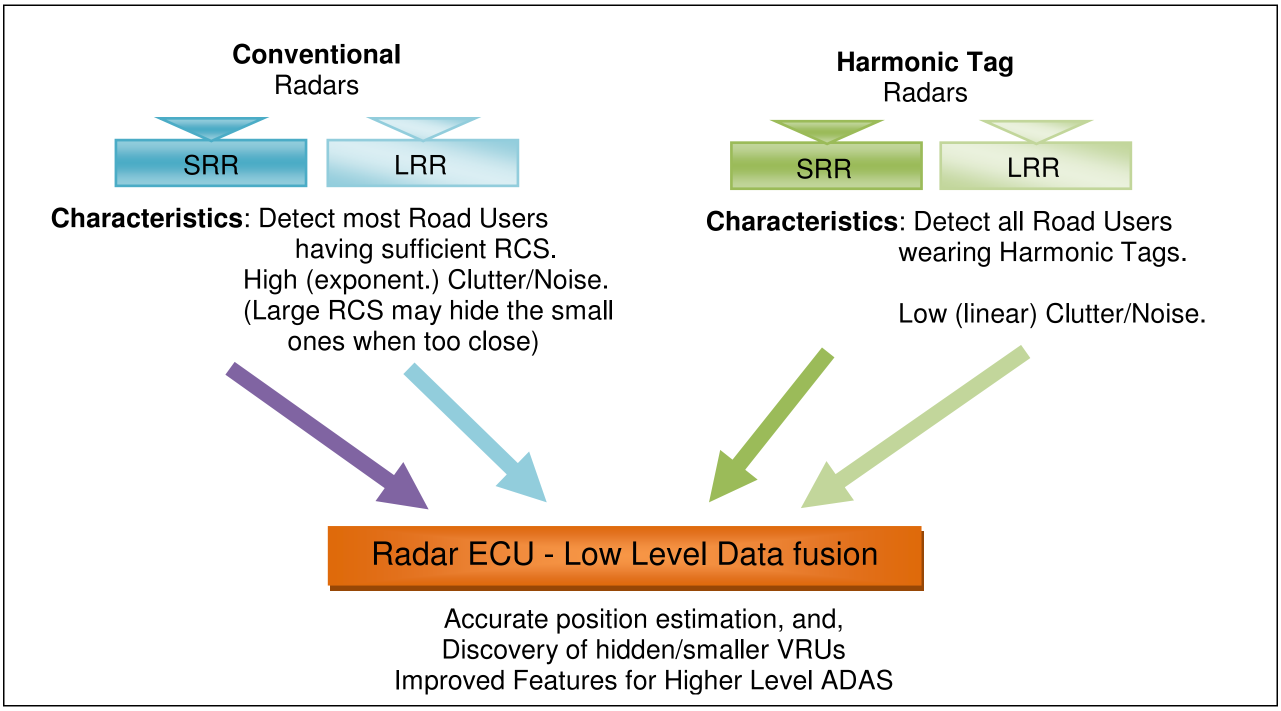


Figure 6: Radar Data fusion example

The above scheme illustrates the advantages of early low level fusion of radar data. For example a person beside a large vehicle will be transparent to the conventional radar due to its small RCS compared to the vehicle's RCS. By wearing a Tag, the person will become visible to the harmonic radar which, in its turn, ignores the presence of the vehicle. By fusing both signals we almost eliminate the probability to miss a Road User (see Figure 5c).

In the Radar ECU another low level data fusion occurs between the radar data and the ego car data. This type of fusion is required by the radar tracking mechanism in order to adapt its predictions whenever the car dynamics change. This fusion helps also to distinguish the Road Users as stationary or moving.

The detected objects by the Radar ECU with all features are then available for further, high level fusion, by the ADAS system in order to refine each object characteristics and proceed with further classification and situation refinement.

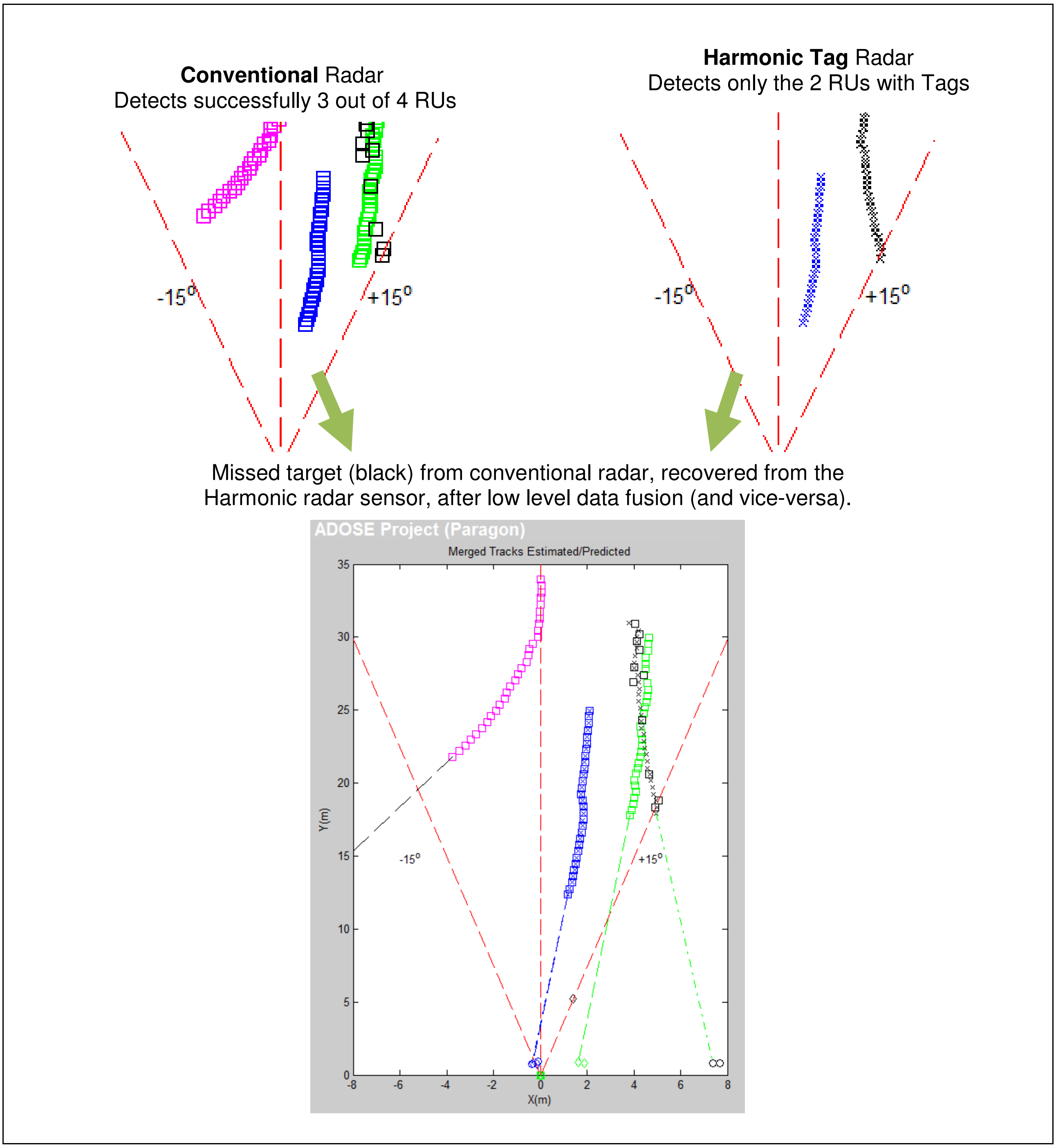


Figure 7: Radar Data fusion reduces miss probability

In addition to the general features selected in the ADOSE project, additional features have to be added to achieve to the specific application that needs to implement.
In this analysis has been presented specific features for:

1) Pedestrians and cyclists
2) Vehicles - Trucks.

Pedestrians and cyclists

Vision-based pedestrian detection is a challenging problem in real traffic scenarios; a lot of variables need to be considered: rotated position and pose of the pedestrians (note that parts or limbs can be partially occluded), different dress colors that can be similar to the background, unpredictable walking of the pedestrians and variable illumination conditions into the environment. The background itself can be various, containing buildings, moving or parked cars, cycles, street signs, signals etc.
Moreover the camera is installed on a moving vehicle and it makes the background dynamic ad pedestrians significantly vary in scale. So, under these conditions, pedestrian detection is different from that of detecting and tracking people in the context of surveillance applications, where the cameras are fixed and the background is stationary.
Note that a candidate selection mechanism is normally applied, to ease the pedestrian recognition. This selection can be implemented by performing object segmentation (see Section.2.1.2).

A large variability problem that has to be solved, for pedestrians detection, so the optimal selection of discriminant features are an issue of the greatest importance.

Two of multiple possible approaches can be distinguish:

- *Component-Based Approach;*
- *Combination of Feature Extraction Methods;*

The component-based approach, shown in the article [10], suggests the division of the candidate body into several parts over which features are computed.
A possible division of the body can be made, selecting a total of six different sub-regions for each candidate regions of the interest, rescaled to an appropriate size. This solution constitutes a tradeoff between exhaustive sub-region decomposition and the holistic approach. To individuate an efficient regions location needs several trials, and a possible solution is depicted in the figure 6 (shown also in [10]).

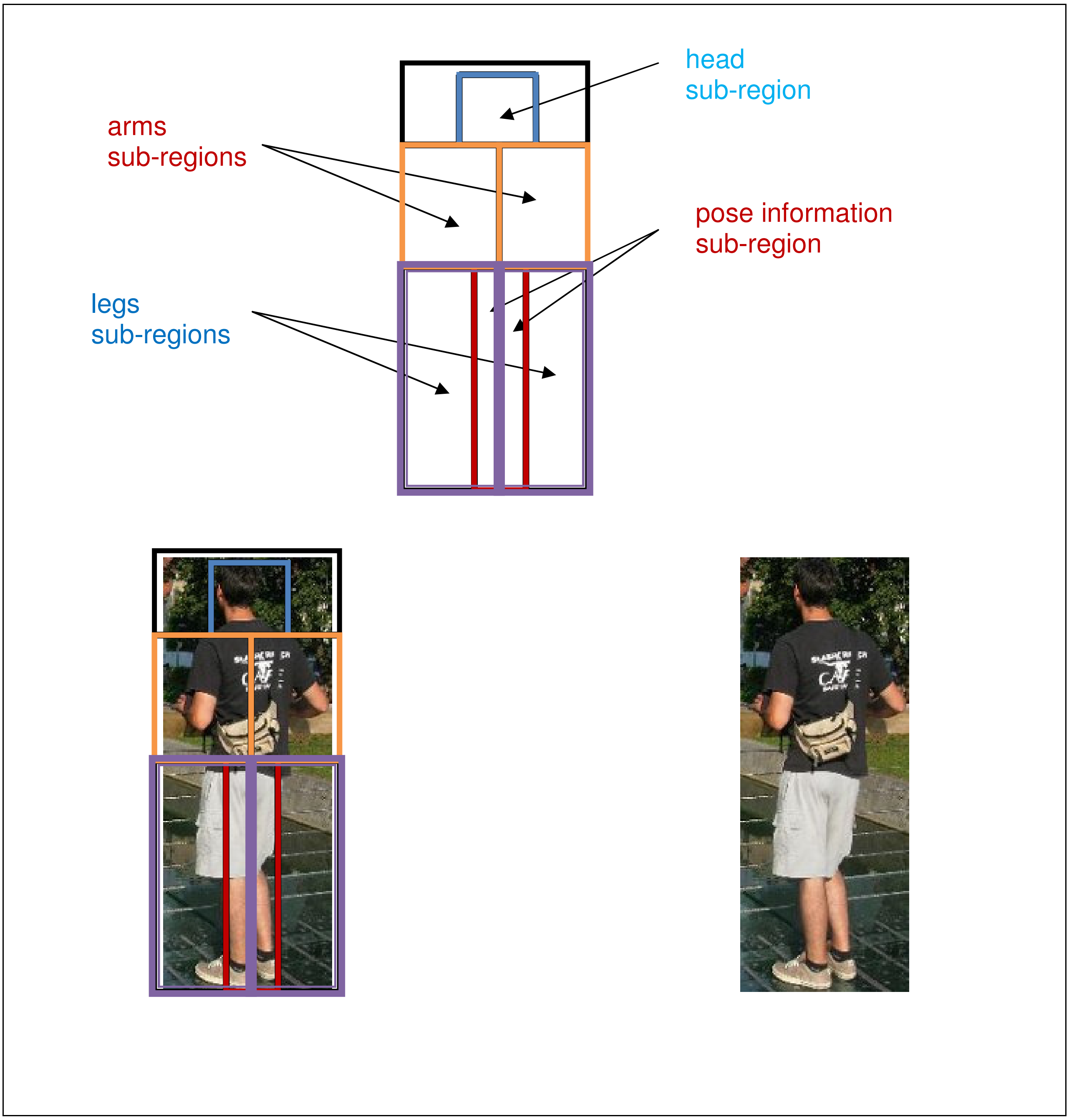


Figure 8: Feature extraction – Pedestrian case example

The first sub-region is located in the zone where the head would be. The arms and legs are covered by the second, third, fourth, and fifth regions, respectively. The sixth region is defined between the legs, which covers an area that provides relevant information about the pedestrian pose.
This sub-region is fundamental to recognize stationary pedestrians.

Recognition performances depend on features. Some features seem to be more suitable than others for representing specific parts of human body, for example legs and arm s are long elements that tend to produce straight lines in the image, while the torso and head are completely different parts, which are not so easy to recognize. This suggests the combination of several feature extraction methods:

- Canny image: the Canny edge detector [11] computes an image gradient, i.e., highlighting regions with high spatial derivatives. The edges computation reduces a lot the amount of data that needs to be managed and filter out useless information, preserving shape properties in the image. The result obtained after applying a Canny filter to the region of interest is directly applied to the input of the classifier.

- Haar wavelets, proposed for pedestrian recognition in [12]. In this case only the vertical features have been considered.

- Gradient magnitude and orientation: the magnitude of the spatial derivatives *gx* and *gy* *has to be* computed for all pixels in the image plane, then orientation as

$$\theta = \arctan(gx, gy)$$

- Cooccurrence matrix [13]: cooccurrence is specified as a matrix of relative frequencies *Pi,j* with which two neighboring pixels, which are separated by distance *d* at orientation *θ*, occur in the image: one with gray level *I* and the other with gray level *j*. The Cooccurrence matrix can be computed over the gray-level image or over the Canny image. The resulting matrices are symmetric and can be normalized by dividing each entry in a matrix by the number of neighboring pixels used in the matrix computation. In pedestrian detection the best solution as distance is one pixel and four different cooccurrence matrices for the following orientations (bins): (0°, 45°, 90° and 135°). The resulting size of the feature vector depends on whether the cooccurrence matrix is computed over the original gray-level image or over a binary one (after applying the Canny operator).

- Histogram of intensity differences: the relative frequencies of intensity differences can be computed between neighboring pixels along four orientations over a normalized image of 128 gray levels.

- Histogram of normalized gradients [14] (HON): the Gradient image is considered. Orientation can be discretized to 20 bins (corresponding to an accuracy of 18°). Only pixels in the Gradient image exhibiting a magnitude greater than some threshold (10) are considered. For those pixels, the values of gradient are accumulated in a 20-bin histogram.

  Number of texture unit (NTU) [15]: the local texture information for a pixel can be extracted from a neighborhood of 3×3 pixels, which represents the smallest complete unit of texture. The corresponding texture unit can be computed by comparing the pixel under study with its eight neighboring pixels.

In the first approach, performance comparison will be made according the next steps: F
first, each feature extractor is applied over the six candidate sub-regions and this yields a set of six feature vectors for each candidate; secondly, the obtained feature vectors are applied to the input of a classifier system that provides a single output, which represents whether the candidate is classified as pedestrian or not (see Object classification).

There are other types of features: people are characterized by low speed and stationary property that can be used. Some groups suggest some ways for motion detection. A system devised for surveillance applications by the Queen Mary and Westfield College, London [18] uses a zero-crossing detection algorithm using the convolution of a spatio-temporal Gaussian with the history over the values of a pixel in the past six frames. This gave good results, with the extension of a second-order Kalman filter that copes with occlusions. Original work by Cutler and Davis of the University of Maryland [19] uses a subtraction between an image at time and a version of the same image stabilized with respect to image at instant. This operation, followed by an appropriate thresholding, gives a map of the pixels representing moving objects. Their method is then based on a two-step approach where recognition is made through analysis of the correlation of two frames taken with a delay of. The authors report good performance both with stationary cameras and moving cameras, provided that the background is homogeneous to some extent.

In the system exploited in [22], vertical symmetries are related to pedestrian candidates both moving and stationary. Further information are made available from symmetry maps of horizontal edges and of their number per column. Then a bounding box encloses interesting regions for a separated recognition step. A more complicated system has been developed at the Ruhr-Universität Bochum [23].
The focus of attention is directed by a composition of a map of the local image entropy, a model-matching module with the shape of a representing human legs, and finally inverse perspective mapping (binocular vision) for the short distance field. This information is combined in a temporal dynamic activation field (DAF) that efficiently allocates computational resources in the following recognition and tracking step.

For the recognition phase, in recent research are pursued two main trends:

• detection of the usual periodicity of *human gait* in the movement of foreground regions;
• *shape analysis* of foreground regions.

Methods based on gait recognition are very robust, but they require the analysis of multiple frames and easily apply only to pedestrians crossing the street in the path of the vehicle, where the alternating movement of legs is evident.
An important drawback of this family of systems is their inability to correctly classify still persons as pedestrians. However shape-based approaches are more sensible to false positives and thus they need a good detection phase, but they correctly recognize even stationary people.
In addition, periodicity of the *human gait* can be recognized with traditional methods like the Fourier transform. Some systems perform a frequency analysis of the changes of candidate patterns over time and then select those that show the frequency spectrum characteristic of human gait.
In the algorithm developed by the group at Ruhr-Universität Bochum [53], the torso of a candidate pedestrian is tracked so that the lower part of the region can be analyzed to reveal the relative motion of legs.
A rough model of two legs consisting of two rod-like pieces each, jointed at the knees, is juxtaposed on the image area below the tracked torso. The periodic movement detected is then correlated to an experimental curve derived from the statistical average of human gait periods. High peaks of the correlation function indicate the presence of a person.
Basic *shape analysis* methods consist in matching a significant and simple shape onto candidate foreground regions. Some systems, like in the system exploited in [22] or the one by SRI International [51], employ a model for the head and shoulders.

A problem of this approach is that it is sensible to scale variation, and this implies that multiple models of different scales are needed.

An algorithm developed within the UTA project at DaimlerChrysler [56] presents an approach made by two-step where both phases rely on shape and pattern analysis. The detection step is based on a search of the image with a numerous set of silhouettes of the human body using a distance transform of the edge image. The silhouettes are organized hierarchically with a coarse-to-fine approach, so that generic forms are tried first, and similar and more detailed shapes afterwards.

In some works feature extractions is more connected to tracking than the others, for example in a work by the University of Maryland [27] a statistical shape model of a pedestrian is first built and then approximated by a linear point distribution model.
The tracking of this model over the image sequence is accomplished with a quasi-random sampling method, based on a zero-order motion model with large process noise high enough to account for the greatest expected change in shape and motion.

Feature extraction methods for cyclists are similar to pedestrian ones as cyclists have a lot of properties in common with pedestrians. All the previous shown methods can be applied, except the component based approach, in fact to apply it, it is necessary to modify the sub-regions structure, reducing it to 4 zones.

Analyzing the ADOSE sensors may be useful fuse data between FIR and NIR: the FIR camera allows identifying the hot spots while the NIR camera the features previous described related to pedestrians and cyclists. In particular, people in FIR images are characterized by two hot zones, corresponding respectively to the head and the pelvis.
Knowing a mean distance between the two zones, a vertical pattern matching could be used to classify the obstacle fusing information elaborated from NIR frames.

Vehicles-Trucks

How to identify pedestrians and cyclists, the vehicles-trucks identification needs additional features.

A useful feature for vehicle detection is the aspect ratio of a vehicle. In fact vehicle detection can be based on the knowledge of the size of the vehicle and the association between image coordinates and world coordinates of each point represented in the acquired image. Knowing the ratio between height and width of a vehicle we can identify the area occupied by it in the image, in correspondence of a certain distance; so we can construct a bounding box associated to every zone of the image that is candidate to contain a vehicle.

Other useful features are Gabor filters.
Gabor filters give a mechanism for obtaining some degree of invariance to intensity due to global illumination, selectivity in scale, as well as selectivity in orientation. They are orientation and scale tunable edge and line detectors. Vehicles do contain strong edges and lines at different orientation and scales, thus, the statistics of these features could be very useful for vehicle detection. So it is possible to subdivide an image into sub-windows and then extract statistics from them, instead of the whole image extraction
These features are then used to train a SVM classifier (see Object classification).

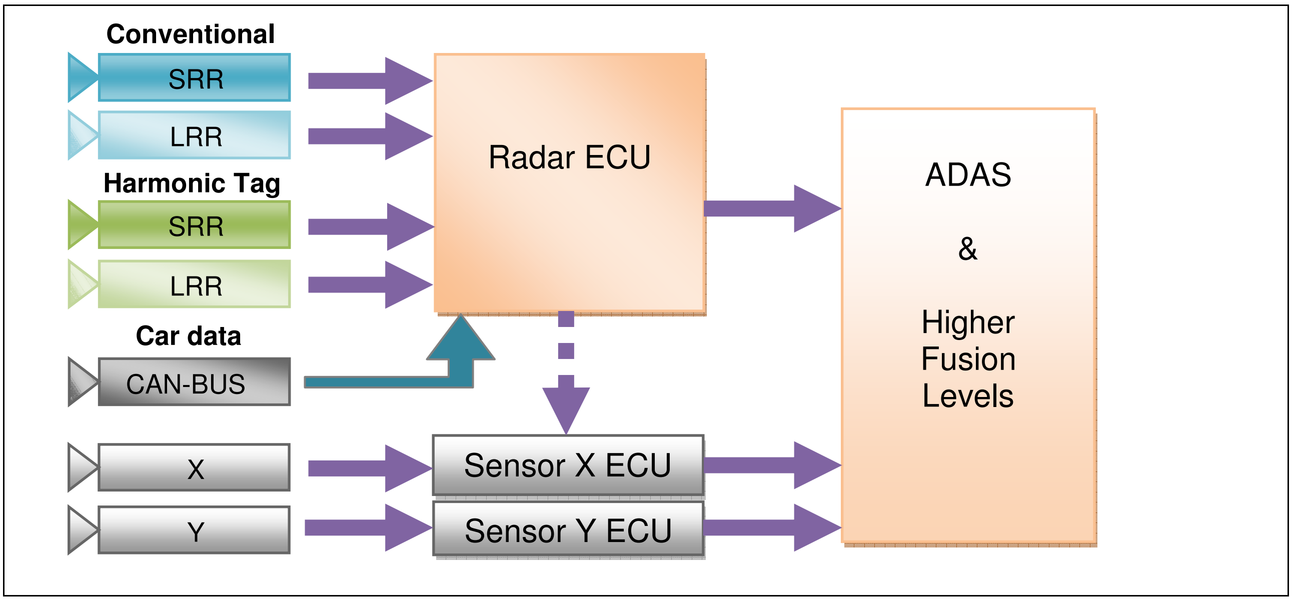
Matthews et al. [16] gave an introduction on how to use Principal Components Analysis (PCA) to extract features of regions of interest. In this method vehicle training images and non-vehicle training images are scaled to 20×20 grayscale images, each 20×20 image was then divided into 25 4×4 sub-windows, each 4×4 sub-window was extracted by PCA. Then a neural net classifies the features.

Other methods consider wavelet features because they form a compact representation, encode edge information, and capture information from multiple scales, and can be computed efficiently. Usually wavelet features used to detect object contain information on the sign of the intensity gradient, that is sensitive to the varied lighting condition and cast shadow.
This means that two attributes which belong to the same class with same absolute values but different signs denote two different features. As we only concern the object existence, this feature pattern will lead to high intra-class variability and increase the complexity of classification.

Data fusion can be very useful in the features composition,. FIR sensor is usually used to detect hot spot regions while other features are computed using NIR sensor. This method allows reducing a lot computational complexity.
In vehicle detection, it is possible to detect low beams and silencer using FIR information, while extracting other features with the NIR sensor (considering the relative positioning of low beams and silencer).
In addition features can be fused and tracked considering ego motion (see Tracking section).

Fusion of Radar ECU Output Data

Radar-based detection and pre-processing by the Radar ECU produces a set of data that are subsequently forwarded to higher fusion levels or to other ECUs.



Radar information typically contains data & parameters regarding the dynamic behavior of detected object, such as: position, velocity, direction, and acceleration.

Radars provide the most accurate information about the dynamics of the detected object. This information is invaluable for other sensor technologies where the accuracy on position or velocity is usually lower.

For example:

- By fusing the radar information about the position, velocity and direction of an object with the vision based information, new features or extra information is acquired (e.g. what we are facing, the front or the side of a vehicle).
- By fusing the combined radar-car information regarding object stationary property with the visually detected objects, any static/scenery objects can be ignored thus reducing the complexity of any subsequent calculation.
- By combining the horizontal 2-D grid of the Radar information with the vertical 2-D visual information a fused 3-D reconstruction of the scene is created, that may be further enhanced by other sensor information for a complete situation refinement.

Finally, Radar ECU tracking information about the detected objects is fused with other sensor tracking information to produce the high level object tracking. Radar tracking is a principal component for these procedures as it provides the most accurate values and predictions.

2.2.2 Object Segmentation

Object segmentation can be made through different techniques; in this analysis has been taken in considerations two classes of obstacles:

1. Pedestrians and cyclists;
2. Vehicles - Trucks.

Pedestrians and Cyclists

Pedestrians object segmentation methods can be applied also for cyclists, for the same reasons explained in the data fusion section.
A first solution for pedestrian object segmentation is the monocular one. It is a cheap solution in fact monocular systems are less demanding from the computational point of view and ease the calibration maintenance process. However, the main problem with candidate selection mechanisms in monocular systems is that, on average, they are bound to yield a large amount of candidates per frame to obtain a low false negative rate. Another problem in monocular systems is the fact that depth cues are lost unless some constraints are applied, such as the flat terrain assumption, and it is not always applicable. It is possible to overcome these problems by using stereo vision systems, but in that case other problems arise (for example the need to maintain calibration and the high computational cost required to implement dense algorithms).

Majority of pedestrian recognition vision systems uses FIR imagery, because the human body heat, so humans are associated to bright objects at these frequencies.

This eases the image segmentation process, and makes it possible to detect pedestrians also in environment without natural or artificial lighting, and at great distances, even at night. Although FIR images are less useful during summer, because high temperatures cause many objects in the environment to appear as bright as human bodies.
In this case the whole image is first scanned to analyze contrast and brightness (note that region of interest is an area containing an object that is warmer than the background). Then, a hot spot detector looks for the regions covered by the brightest pixels, or those with a lower grey-level, but that are connected to a bright region. This is necessary to extend each bright region to include all pixels that seem to belong to the same object. A low level segmentation is much more helpful than thresholding in fact pedestrians don't have the same grey-level for the whole body, and some body portions would be discarded by a simple threshold.
Then a column-wise histogram can be then computed for each region, to select only hot spots that have a significant vertical component (it happens when they contain a pedestrian).

However there are alternative segmentation techniques, involving the analysis of more than one image, such as:

- the analysis of *motion (connected to tracking)*;
- the processing of *stereo images*.

Motion heavily uses temporal information and has proved to be quite reliable if one wants only to find a moving object, while it is not precise to find velocity. Unfortunately, it does not detect standing pedestrians and needs the analysis of a sequence of a few frames before returning a response.
A few works use this kind of detection with optical flow as a means of segmentation. The basic idea is to detect blobs with a given shape or a common feature, like color that have similar values of optical flow and track their movement in subsequent frames.
A group at the University of Rochester [17] analyzes the scene using a discrete cube (that is a representation of a sequence of frames, each divided in areas in its two spatial dimensions), where they assign to each region its average optical flow.
In this system, four divisions are made in each spatial dimension, and six along the temporal dimension, resulting in a feature vector containing the average optical flow of each region. Then a Fourier analysis is employed in order to classify these values. This method has been applied to the monitoring of repetitive human activities with a stationary camera. However in optical flow methods a good cancellation of ego motion is critical in applications with a moving camera.

Vehicles-Trucks

Vehicles-trucks detection is based on the knowledge of the size of the target in exam and the association between image coordinates and world coordinates of each point represented in the acquired image.
Knowing the ratio between height and width of a vehicle it can identify the area occupied by it in the image, in correspondence of a certain distance; so it can construct a bounding box associated to every zone of the image that is candidate to contain a vehicle.

2.2.3 Object Classification

There are different types of classifiers:

- Neural nets;
- Linear classifiers (SVM, PCA, LDA,…);
- Complex classifiers (AdaBoost, Random Forest,…).

A classifier can be seen as a transferring function that makes a segmentation of a n-dimensional space. To do a classification some sets of objects are needed:

1) Training set
2) Validation set
3) Certification set

Training set is the one used for the learning while test set contains examples not included in (1), and it's used to test the correct generalization of the learning algorithm.
Certification sent is useful if the algorithm needs some parameters to be adapted to improve performances.
These relations, between learning of the classifier and sets, are shown in figure 7.

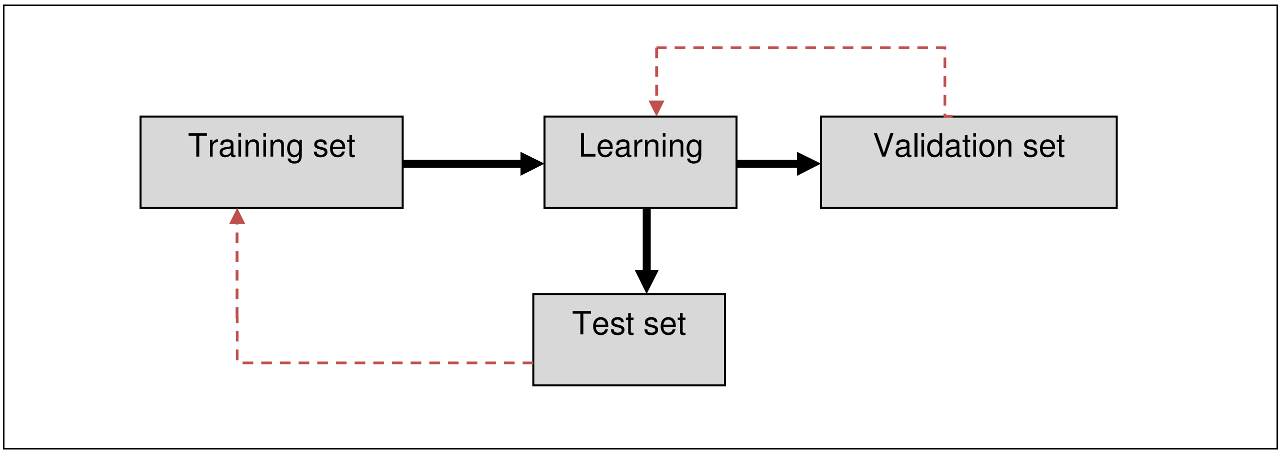


Figure 9: Learning and classifiers sets

Neural Nets

Between Neural Nets there are multi-layer architectures with an input layer, an output layer and other hide layers (see figure 8).

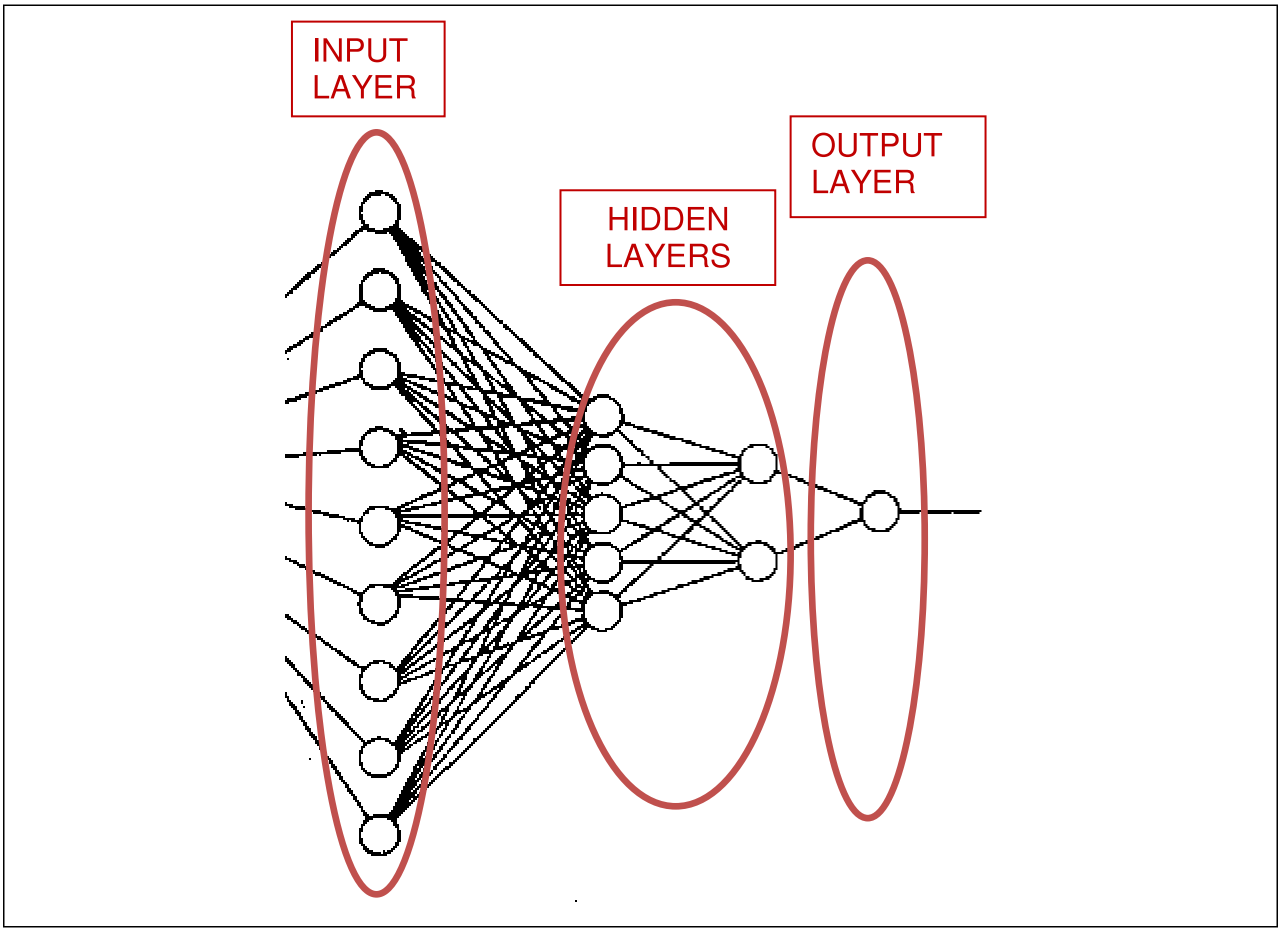


Figure 10: Neural net high-level representation

Each layer is made by neurons connected in a specific way.
A neuron is made by a linear adder the produce a net input and a non-linear activation threshold function, as shown in figure 9.

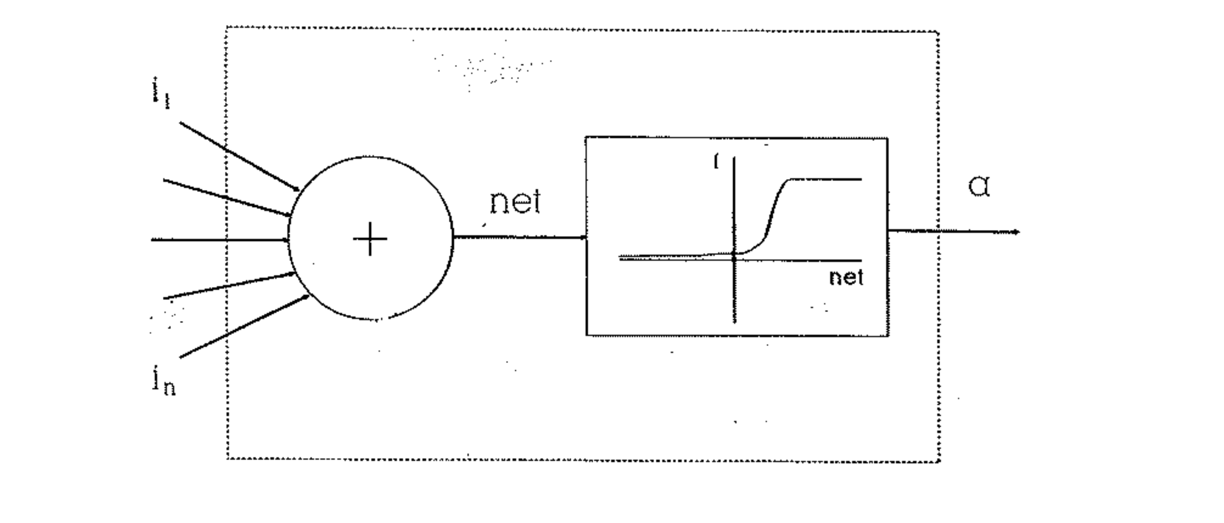


Figure 11: Neuron representation

Neurons are linked between weighted connection, and the behaviour of the net is defined by number of neuron, topology and connection weights.

Complex classifiers

Among the complex classifier, it is possible to distinguish AdaBoost (ADAptive BOOSTing) ,that is an ensemble learning classifier that uses a set of weak classifiers generated by a set of loops on an images group. It represents an efficient solution in vehicle detection context.

A weak classifier is a classifier that maps correctly at least 50%+1 input attributes. AdaBoost is a meta-algorithm that can be used in conjunction with many other learning algorithms to improve their performance. It calls a weak classifier repeatedly in a series of rounds t=1,...,T. For each call a distribution of weights y_i is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of each incorrectly classified example are increased, so that the new classifier focuses more on those examples.
AdaBoost algorithm is the following:

1. The same weight is associated to every sample x_i and a classification function h is found to maximize $\sum_i \; y_i \cdot h(x_i)$ where $h(x_i)$ is a classification hypothesis;
2. Every weight is updated: weight of incorrectly classified samples are increased
3. The successive h that maximize $\sum_i \; y_i \cdot h(x_i)$ is found. Weight of this classifier *alfa* are computed.
4. Return to point 2.

The final classifier is:

$$sgn(\sum_{i=1}^{T} y_t \cdot h_t(X))$$

In order to find weights associated to each classifier an objective function that minimizes *alfa* is needed. This should be a non-linear function, hard to compute, so the following linear function is used instead:

$$Obj = \sum_{i=1}^{n} e^{-y_i(\sum_{i=1}^{T} alfa_t \cdot h_t(X))}$$

So we can divide vehicle detection in two phases:

a. Bounding boxes identification
b. Bounding boxes classification

The first phase is made by the following steps:

1. Association between a row of the image and a particular distance.
2. An inverse perspective mapping that associates world coordinates to image coordinates is done
3. Construction of the bounding boxes associated to the row (a single bounding box is created in correspondence of each pixel of the row)
4. Change of row and return to step (1)

Note that obviously bounding boxes associated to points on the same row have the same dimension while the ones on different rows have different dimension (that increase with the distance associated to a pixel in a row).

The second phase is made by the following steps:

1. A new bounding box is selected
2. Weak classifiers make decisions about the presence of the vehicle in the selected bounding box of step (1)
3. A selection of classifiers is made according a minimization of the classification error
4. Return to step (1)

Note that in this phase it is possible to use Haar feature classifier, a class of weak classifiers based on the Haar features: a particular set of images in which white region is summed to the correspondent region of the associated image and black region is subtracted. A little set of these features are represented in the figure 10.
For a description about Haar feature classifier see *Aggregate features and AdaBoost for music classification* (1) and *Robust real-time object detection* (2).

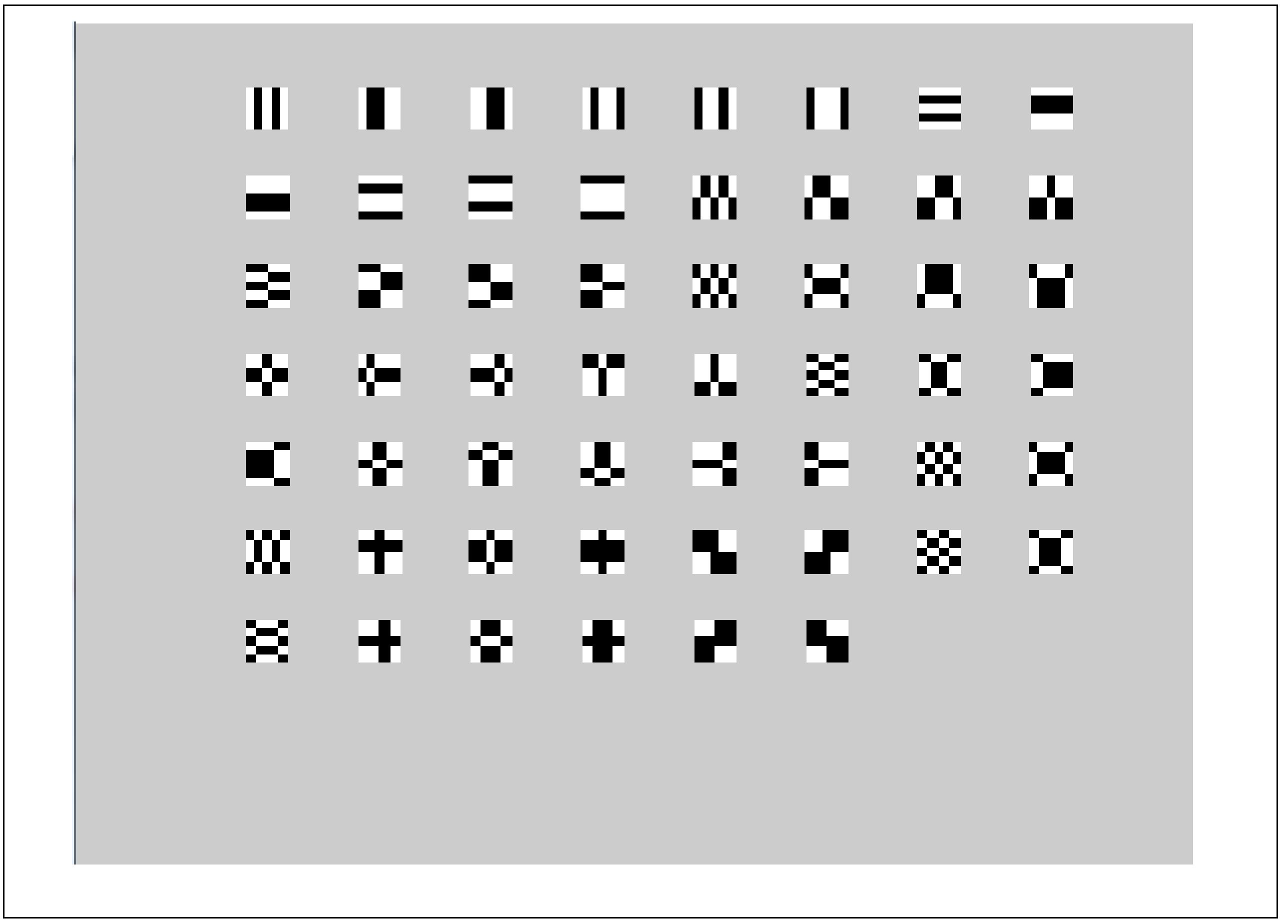


Figure 12: Haar feature

Pedestrians and Cyclists

There are some important aspects that need to be addressed when constructing a classifier: the global classification structure and the use of single or multiple cascaded classifiers.
These issues are directly connected to the way features are extracted. The first decision to make implies the development of a holistic classifier against a component-based one.

In the first option, features are extracted from the bounding box in the image plane (that describes the candidate).
Each pedestrian body part is then independently learned by a specialized classifier in the first learning stage. So each individual classifier return an output that correspond to a particular body part, and all outputs can be integrated in a second stage that provides the final classification output. The component-based approach can provide correct classification results as long as a sufficient number of body parts or limbs are visible in the image. This allows for the detection of partially occluded pedestrians whenever the contributions of the pedestrian visible parts are reliable enough to compensate for the missing ones.

More systems employ pattern recognition with classifiers to accomplish the recognition step. In some cases the original imaged is processed before the application of the classifier.
For example, Zhao and Thorpe [20] propose a three-layer feed forward network processing the intensity gradient image rather than the original image.

Vehicles-Trucks

In the deliverable D6.10 it is shown that AdaBoost represents an efficient solution in vehicle detection context but other classifiers can be used according additional features.
For example, Matthews et al. [16] use neural network for classification, while other methods that are based on Gabor filters feature use linear classifiers like SVN.

2.3 Situation refinement

The systems should be able to recognize the maneuvers performed by the driver, and the associated aims, in order to trigger alarms or preventive actions in danger contexts. This can be done by situation refinement level, which tasks can be identified in:

1) Entity (of the road environment) recognition
2) Entity relationship recognition

These two types of recognition can made possible to react efficiently to a lot of situations like lane change, overtaking maneuver, and to avoid other unwanted situations.

Lane departure and lane support systems can be developed to prevent the vehicle from changing lane unintentionally or entering a lane where other vehicles are moving very closely.
Current lateral protection systems introduced in the market focus essentially in the perception of the lateral behaviour and the detection of lane changes but more complex systems have to be developed.

Particular features can be used to detect guardrails, roads, lanes and road-signs. The mutual relations between these objects allow understanding the context, refining other object refinement processes. For example by lanes number analysis and road-signs recognition is possible to understand that the vehicle is in a motorway context, so speed constraints have to be monitored to assure security, while elaboration associated to pedestrian detection can be avoided. On the contrary, in an urban environment elaboration pedestrian detection is a more critical aspect.
Similar consideration can be done using information about vehicle position, extracted by a map and detection about day/night temporal location.

2.3.1 Tracking

Pedestrians, cyclists and vehicles-trucks detection are associated to dynamic scenes with independently multiple moving objects that are represented by features whose motion is tracked in image sequences.

Moreover the feature points may disappear temporarily leaving and entering in the field of view. In addition to these incomplete trajectories, it is possible to have a great number of feature points and different speeds of the moving objects.
The feature based tracking techniques extract features from the images and identify the corresponding features in each image of the sequence.

Tracking processing algorithm analyzes sequential video frames and outputs the movement of targets between the frames. There are a variety of algorithms every of them has certain advantages and weaknesses.

The main components of a visual tracking system are:

1) Target representation and localization

2) Filtering and data association.

Some type (1) algorithms are:

- Blob tracking: Segmentation of object interior (for example blob detection, block-based correlation or optical flow);
- Kernel-based tracking (Mean-shift tracking): An iterative localization procedure based on the maximization of a similarity measure;
- Contour tracking: Detection of object boundary (e.g. active contours or Condensation algorithm).

The filtering and data associaton component is mainly a top-down process characterized by prior information incorporation about scenes or objects, dealing with object dynamics and evaluation of hypotheses. These methods are useful to track objects along with more complex object interaction like tracking objects moving behind obstructions but the computational complexity of these algorithms is usually very high.

Some filtering algorithms are :

- Kalman filter
- Particle filter

Kalman Filter

Kalman filter is an optimal recursive Bayesian filter for linear functions subjected to Gaussian noise while particle filter is useful for sampling the underlying state-space distribution of non-linear and non-Gaussian processes.

The iterative predictor-corrector nature of the Kalman filter can be very helpful in fact, at each moment only one constraint on the state variable need be considered; this process is repeated, considering a different constraint at every time instance. All the measured data are accumulated over time and help in predicting the state.

Considering the standard Kalman filter, the state transition *x(.)* from *t* to *t+1* can be expressed with the equation

$$x(t+1) = Ax(t) + w(t) \qquad (1)$$

where *A* is the state transition matrix and w(t) is a noise term assumed independent of the state *x(t)*. To be more precise *w(t)* is a Gaussian random variable with zero mean and a covariance matrix *Q*, so its probability distribution is:

$$p(w) \sim N(0,Q) \qquad (2)$$

The covariance matrix *Q* will be referred to as the process noise covariance matrix.
It accounts for possible changes in the process between *t* and *t+1* that are not already accounted for in the state transition matrix.

But it is necessary to model also the measurement process, or the relationship between the state and the measurement. In fact it is not always possible to observe the process directly .Some of the parameters describing the state may not be observable at all, or observable with an error, so measurements might be scaled parameters, or possibly a combination of multiple parameters. Note that the whole is based on the assumption that the relationship is linear.
So the measurement *y(t)* can be expressed in terms of the state *x(t)* with

$$y(t) = Cx(t) + v_t, \qquad (3)$$

where *C* is a matrix *m x n* which relates the state to the measurement; and *y(t)* is the noise of the measurement with a normal distribution expressed by

$$p(v) \sim N(0,R), \qquad (4)$$

where *R* is the covariance matrix referred to as measurement noise covariance matrix.

Particle Filter

Particle filters using is usually to estimate Bayesian models with latent variables that are connected in a Markov chain, so this is similar to a hidden Markov model. A difference with hidden Markov Model is that usually in it the state space of the latent variables is continuous rather than discrete, and not sufficiently restricted to make exact inference tractable. Particle filter attempts to estimate a sequence of parameters x_k exploiting observed data y_k. All Bayesian estimates of x_k follow from the posterior distribution $p(x_k \mid y_0, y_1, \ldots, y_k)$.

Particle model can be shown in the following form:

a) Assume $x_0,\ldots x_n$, a first order Markov process with an initial distribution $p(x_0)$, so that each x_i depends only con x_{i-1}

b) Each y_k only depends on x_k

c) Consider noises (w_k , v_k) for parameters and observed data respectively

An example of this scenario is the following:

$$x_k=f(x_{k-1})+w_k \qquad (1)$$

$$y_k=h(x_k)+v_k \qquad (2)$$

where f(.) and h(.) are known functions and the noises are identically distributed and mutually independent.

2.3.2 3D reconstruction

3D reconstructions problem has drawn considerable attention. A lot of methods have been presented in literature, principally based on the analysis of some image points and geometric clues in the image, and on stereo vision. These methods can be classified as:

1) Model based methods
2) Constraint-based methods
3) Stereo-vision based methods

The former consists of an assemblage of primitive shapes, such as cylinders, prisms, parallelepipeds. In these methods the scene is defined as in CAD systems, for examples some buildings can be decomposed in parallelepipeds, truncated pyramids etc.
After that a fitting of the model is done with image data, and its position, orientation and dimension are determinated.
Note that these methods have the limitation that the scene must be decomposable in primitive shapes.

The second class of methods analyzes geometric properties of scene-planarity, orthogonality and other geometric items. The limitation of the first class methods does not exist, in fact any shape can be reconstructed if there are enough geometric properties to define a unique reconstruction. Geometric properties detection can be done both automatically than by user.
Constraint-based methods can be classified as:

a) Single-view methods
b) Multi-view methods

The former ones rely on the possibility of expressing geometric properties as linear constraints on the estimated quantities, and this implies that edge of interest and 3D directions orthogonal or parallel to planes should be estimated before the reconstruction. For this class of methods, 3D directions are called “dominant directions” and knowing them beforehand allows expressing planarity by a linear constraint.

Figure 11 shows that, given a dominant direction **v**, N 3D points **P1, P2,…,PN** belong to a plane normal to the direction if and only if

$$\mathbf{v}^T \mathbf{P1} = \mathbf{v}^T \mathbf{P2} = \ldots = \mathbf{v}^T \mathbf{PN}$$

In single-view methods, some dominant directions are estimated before reconstruction, using vanishing points and calibrating the camera (e.g. with the technique of Caprile and Torre [30]). So after these dominant directions are known, it is possible to express the geometric information by linear equalities. Then other linear constraints provided by the image are added and the resulting linear equation system is solved. However, to this main idea are associated many variations, for example in [31] and [32] are used linear equations in which appears the height of planes.
Methods of classes (1) and (2) can be applied to the NIR camera.

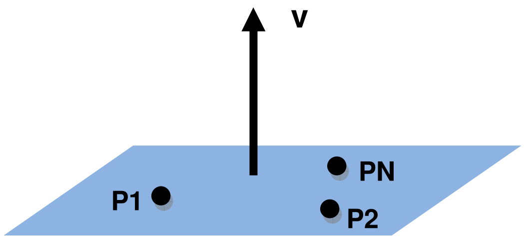


Figure 13: Dominant direction and normal plane points

Stereo-vision based methods exploit that 3D reconstruction is possible knowing:

- Calibration parameters of the acquisition system for 2 (or more) cameras;
- Localization of homologous points, i.e. projections of the same point in world coordinates, on the two cameras images.

While calibration is simple and can be precise, homologous points localization can be very complex, in fact the homologous of a point does not necessary exist, and after point choice in the first image, all the second image usually needs to be scanned.
However, homologous points detection can be done easily using some relations, for example considering the epipolar line, that is the projection of the epipolar plane (i.e. the plane containing pin holes, world point and his projections on the images of the two cameras). In fact, fixed a point on the first image, the homologous point in the second image must be placed on the epipolar line. Figure 12 shows these concepts.

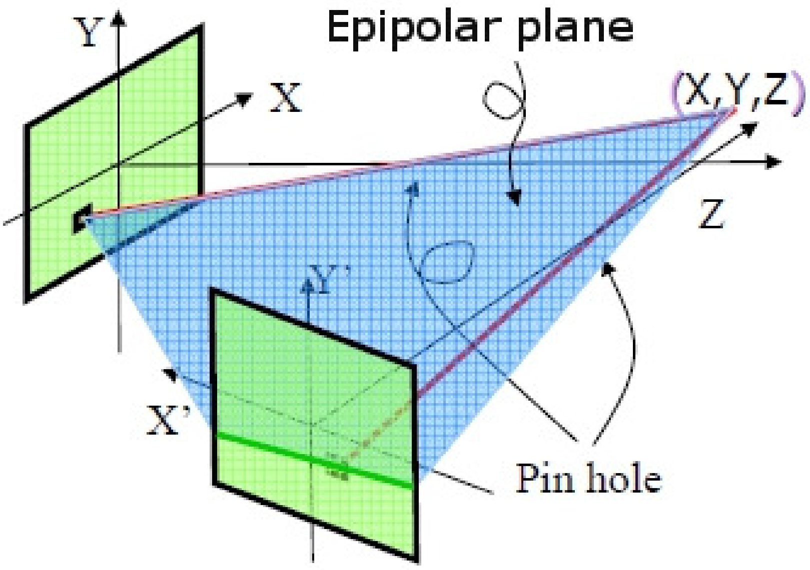


Figure 14: Images and epipolar line

So, given the projections points coordinates of all the world points, 3D reconstruction is possible, but homologous association needs that the points have to be located in zones rich of information, i.e. zones with texture. This condition is not satisfied for all the image points. An example of points located in a zone with no information is shown in Figure 12.

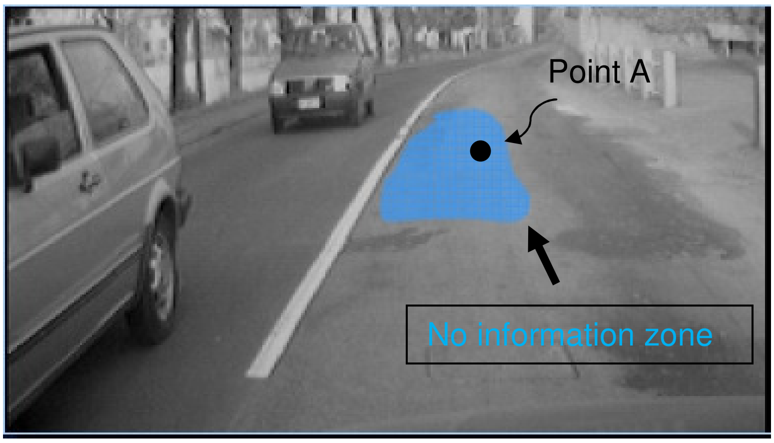


IMAGE 1 FROM FIRST CAMERA

IMAGE 2 FROM SECOND CAMERA

Figure 15: Zones without information in camera images

Moreover if epipolar lines are not perfectly horizontal (because set-up introduce a roll angle between the cameras and the ground), computation is more complex, so a rectification of the image is needed.
Figure 13 shows rectification of an image, to make epipolar lines perfectly horizontal.

IMAGE 1 FROM FIRST CAMERA

IMAGE 1 FROM FIRST CAMERA WITH EPIPOLAR LINES HIGHLINED

RECTIFIED IMAGE 1

RECTIFIED IMAGE 1 WITH EPIPOLAR LINES HIGHLINED

Figure 16: Rectification

2.3.3 Radar Tracking

A supervising ADAS system uses the Radar ECU tracking information fused with any other sensors tracking information, to produce the high level object tracking.

Radar tracking information is a principal component for these procedures as it provides more accurate values and predictions.

Radar tracking treats Road Users as points in the 2-D field-of-view and assigns properties such as position, velocity & acceleration in each dimension. Radar tracking applies various techniques (algorithms) to estimate and predict the dynamic state of an object. Some of these are:

- Simple α-β-γ tracker
- Kalman Filter (KF)
- Extended Kalman Filter (EKF)
- Many variations of EKF
- Adaptive Multi-Model Partitioning Algorithm

<u>The α-β tracker</u> is a simple computationally light tracker, suitable for constant speed trajectories. If the object velocity is not constant, the α-β-γ tracker handles acceleration too. The main advantage is that it is a simple, fast and computationally light; its drawback is that the gains are constant and does not handle uncertainties.

<u>The Kalman filter</u> is an Bayesian linear recursive estimator, able to handle Gaussian uncertainties in an optimal way. The Kalman tracker estimates not only the position or velocity of an object but also the inaccuracy or uncertainty of these values. The filter also adjusts continuously its gain according to the residuals, in order to improve the final estimations. Details on the algorithm are already presented in the previous section 2.3.1 where the filter is applied for image tracking.

<u>The Extended Kalman filter</u> is a non-linear version of the Kalman filter able to handle model nonlinearities and parametric uncertainties. Depending on the linearization level there are EKFs of different order. In addition there are <u>various EKF implementations</u> depending on the specific needs of the physical problem. Finite memory or receding horizon filter versions are often used in tracking problems. When using EKF it is essential to control its deviation due to unexpected change & model non-linearity.

<u>The Adaptive MMP Algorithm</u> is a more robust, adaptive algorithm utilizing many elemental Kalman or EKF filters in parallel. The overall estimate of the MMPA can be taken either to be the individual estimate of the elemental filter exhibiting the highest posterior probability (called “MAP –Maximum A Posteriori- estimate”) or the weighted average of the estimates of the elemental filters, where the weights are simply the posterior probabilities associated with each estimate (called “MMSE – Minimum Mean Square- estimate”). At each iteration, the algorithm selects the model that corresponds to the maximum a posteriori probability as the correct one. This probability tends (asymptotically) to one, while the remaining probabilities tend to zero. If the model structure changes, the algorithm senses the variation and increases the corresponding a posteriori probability, while decreasing the remaining ones. Thus the algorithm is adaptive in the sense of being able to track model changes in real time. This procedure incorporates the algorithm’s intelligence and provides not only a tracking filter but also a classification and decision tool that may handle complex (fused) data more efficiently.

3. CONCLUSIONS

The object refinement stage could uses data fusion between FIR and NIR sensor in both direction, so FIR sensor that detects a hot blob and FIR sensor that uses it as ROI to search for symmetry features or NIR sensor that detects symmetry feature and FIR sensor that uses it as ROI to search for hot blobs.

The object segmentation and classification could be done only by means of the NIR sensor camera with the usual method described in this document.

Regarding the situation refinement some considerations need to be done for ADOSE system, considering the sensors as they have been tested at the end of the project:

1) Unfortunately FIR resolution is too small, and the difference between FIR images and NIR ones creates problems on pixel association;

2) FIR and NIR cameras are positioned in different points, at different distances from world points and with no aligned optical axes;

3) Points matching between a NIR image and a FIR image is not simple, not only due to different resolution of the sensors but also because of their nature; in fact, a point with a value for the NIR image has a completely different value in the FIR image (associated to heat)

For all these reasons, it is possible to conclude that 3D reconstruction is not possible for NIR-FIR couple of available sensors, using stereo-vision based methods.

All ADAS applications toward road safety will benefit by the middle & high level fusion possibilities offered by the ADOSE radar ECU results. Automotive Radar data acquired using various radar technologies are processed by the radar ECU in order to deliver a very accurate set of estimations about the road users' properties and especially distance, speed, direction and tracking.

The results of the ADOSE radar ECU, can be subsequently forwarded, either to the same processing level, such as another sensor ECU, or, to any higher fusion level of an ADAS framework. By inserting these highly accurate radar ECU results to an ECU of another sensor, the dynamic parameters of a detected object are improved, thus, enhancing all other sensor specific properties of the object (shape, size, image, intensity, etc.). By fusing the radar ECU results at any higher level in ADAS, the detected objects by the radar ECU are fused with the corresponding objects from the other sensors and the result is a more accurate set of enhanced object properties and an improved situation description that leads to improved road safety.

ADOSE radar ECU is specially designed to be modular and adjustable to different radar technologies and can be applied either for an early stage radar sensor fusion or for a higher level object fusion in ADAS.

4. BYBLIOGRAPHY

[1] M. Litzenberger, J. Kogler, W. Kubinger; ADOSE Deliverable D6.1 Specification of the silicon retina stereo sensor; 2008

[2] J. Kogler, C. Sulzbachner, W. Kubinger; ADOSE Deliverable D6.3 Preliminary algorithm design of silicon retina stereo sensor; 2008

[3] C. Sulzbachner, W. Kubinger, H. Hemetsberger; A real-time framework for performing computer vision algorithms on embedded platforms; The 17th International DAAAM Symposium; 2006

[4] Q.-T. Luong and O. D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. Int. J. Comput. Vision, 22(3):261-289, 1997.

[5] D. CAPELLO, ST ITALY L. BIZZARRI , FRAUNHOFER H. OPPERMANN, FRAUNHOFER H. SCHRÖDER, FRAUNHOFER T. TEKIN; ADOSE DELIVERABLE D3.1 CONCEPT DESIGN OF THE MULTI-SPECTRAL OPTICAL SENSOR

[6] K.F. REINHART, F. SCHINDLERSAEFKOW, J. VAN NYLAN; ADOSE DELIVERABLE D2.1 Concept of low-cost FIR focal-plane-array

[7] D. San Segundo Bello; ADOSE DELIVERABLE D4.1 Concept design for 3-D ranging camera

[8] Alberto Broggi, Massimo Bertozzi, Mike Del Rose, and Mirko Felisa, A SYMMETRY-BASED VALIDATOR AND REFINEMENT SYSTEM FOR PEDESTRIAN DETECTION IN FAR INFRARED IMAGES, In Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2007, pages 155-160, Seattle, WA, USA, September 2007.

[9] Massimo Bertozzi, Alberto Broggi, Claudio Caraffi, Mike Del Rose, Mirko Felisa, and Guido Vezzoni, PEDESTRIAN DETECTION BY MEANS OF FAR-INFRARED STEREO VISION, Computer Vision and Image Understanding, 106(2):194-204, May-June 2007

[10] Ignacio Parra Alonso, David Fernández Llorca, Miguel Ángel Sotelo, *Member, IEEE*, Luis M. Bergasa, *Associate Member, IEEE*, Pedro Revenga de Toro, Jesús Nuevo, Manuel Ocaña, and Miguel Ángel García Garrido COMBINATION OF FEATURE EXTRACTION METHODS FOR SVM PEDESTRIAN DETECTION

[11] F. J. Canny, "A COMPUTATIONAL APPROACH TO EDGE DETECTION," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[12] C. Papageorgiou and T. Poggio, "A TRAINABLE SYSTEM FOR OBJECT DETECTION," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15–33, 2000.

[13] R. M. Haralick, "STATISTICAL AND STRUCTURAL APPROACHES TO TEXTURE," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[14] N. Dalai and B. Triggs, "HISTOGRAM OF ORIENTED GRADIENTS FOR HUMAN DETECTION," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recog.*, 2005, pp. 886–893.

[15] L. Wang, "TEXTURE UNIT, TEXTURE SPECTRUM AND TEXTURE ANALYSIS," *IEEE Trans. Geosci. Remote Sens.*, vol. 28, no. 4, pp. 509–512, Jul. 1990.

[16] N. D. Matthews, P. E. An, D.Charnley, C.J.Harris, "Vehicle detection and recognition in greyscale imagery", Control Eng.Practice, 1996, 4(4), pp. 473-479.

[17] R. POLANA AND R. C. NELSON, "DETECTION AND RECOGNITION OF PERIODIC, nonrigid motion," *Int. J. Comp. Vis.*, vol. 23, no. 3, pp. 261–282, 1997.

[18] S. J. McKenna and S. Gong, “Non-intrusive person authentication for access control by visual tracking and face recognition,” in *Int. Conf. Audio and Video Authentication*, 1997, pp. 177–184.

[19] R. Cutler and L. S. Davis, “Robust real-time periodic motion detection, analysis and applications,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 781–796, Aug. 2000.

[20] L. Zhao and C. Thorpe, “Stereo and neural network based pedestrian detection,” *IEEE Trans. Intell. Transport. Syst.*, vol. 1, no. 3, pp. 148–154, 1999.

[21] D. Beymer and K. Konolige, “Real-time tracking of multiple people using continuous detection,” in *Proc. Int. Conf. Comp. Vis.*, 1999.

[22] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, “Shape-based pedestrian detection,” in *Proc. IEEE IV*, 2000, pp. 215–220.

[23] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen, “Walking pedestrian recognition,” *IEEE Trans. Intell. Transport. Syst.*, vol. 1, no. 3, pp. 155–163, 2000.

[24] C. Wöhler, U. Kressler, and J. K. Anlauf, “Pedestrian recognition by classification of image sequences. Global approaches vs local spatiotemporal processing,” in *Proc. IEEE Int. Conf. Pattern Recognition*,2000.

[25] H. Fujiyoshi and A. Lipton, “Real-time human motion analysis by image skeletonization,” in *Proc. IEEE WACV’98*, 1998, pp. 15–21.

[26] D. M. Gavrila, “Pedestrian detection from a moving vehicle,” in *Proc. Eur. Conf. Comp. Vis.*, 2000, pp. 37–49.

[27] V. Philomin, R. Duraiswami, and L. Davis, “Pedestrian tracking from a moving vehicle,” in *Proc. IEEE IV*, 2000, pp. 350–355.

[28] A. Mohan, C. Papageorgiou, and T. Poggio, “Example-based object detection in images by components,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 4, pp. 349–361, 2001.

[29] C. Papageorgiou and T. Poggio, “A pattern classification approach to dynamical object detection,” in *Int. Conf. Computer Vision*, 1999, pp. 1223–1228.

[30] B. Caprile, V. Torre, Using vanishing points for camera calibration, Intl. J. Comput. Vis. 4 (1990)127–140.

[31] P. Sturm, A method for 3D reconstruction of piecewise planar objects from single panoramic images ,in : OMNIVIS—CVPR Workshop on Omnidirectional Vision, Hilton Head Island, 2000, pp. 119–126.

[32] P.F. Sturm, S.J. Maybank, A method for interactive 3D reconstruction of piecewise planar objects from single views, in: British Machine Vision Conf., 1999, pp. 265–274.

5. LIST OF FIGURES

Figure 1: Data processing tasks of a typical driver assistance system 4
Figure 2: Refinement phase 6
Figure 3: Software - High level 7
Figure 4: Data fusion concept....9
Figure 5: Data fusion example....10
Figure 6: Radar Data fusion example....11
Figure 7: Radar Data fusion reduces miss probability....12
Figure 9: Feature extraction – Pedestrian case example....14
Figure 10: Learning and classifiers sets....21
Figure 11: Neural net high-level representation 22
Figure 12: Neuron representation....23
Figure 13: Haar feature 25
Figure 14: Dominant direction and normal plane points....30
Figure 15: Images and epipolar line 31
Figure 16: Zones without information in camera images 32
Figure 17: Rectification....34

6. LIST OF ACRONYMS

3DCAM	3D Range Camera
ACC	Adaptive Cruise Control
ADOSE	Reliable Application Specific Detection of Road Users with Vehicle On-Board Sensors
AGC	Automatic Gain Control
ASIC	Application-Specific Integrated Circuit
BSD	Blind Spot Detection
CCD	Charge Coupled Device
CMOS	Complementary metal oxide semiconductor
DSP	Digital Signal Processor
ECU	Electronic Control Unit
FIR	Far InfraRed
FOV	Field of View
FPGA	Field Programmable Gate Array
fps	frames per second
GUI	Graphical user interface
HR-ATAG	Harmonic Radar with Active Tags
HR-PTAG	Harmonic Radar with Passive Tags
HW	Hardware
I2C	Inter Integrated Circuit
IMEC	Interuniversity Microelectronics Centre
ITOF	Indirect Time Of Flight
JPEG	Joint Photographic Experts Group

LED	Light Emitting Diode
LRR	Long Range Radar
LUT	Look-Up Table
LVDS	Low-voltage differential signaling
MEMS	Micro Electro-Mechanical Systems
MFOS	MultiFunctional Optical Sensor
NIR	Near InfraRed
PC	Personal Computer
PCB	Printed Circuit Board
RCS	Radar Cross Section
RFID	Radio Frequency IDentification
ROI	Region of Interest
SOI	Silicon On Insulator
SR	Silicon Retina
SRR	Short Range Radar
SRS	Silicon Retina Stereo
SW	Software
TCP	Transmission Control Protocol
TOF	Time Of Flight
UDP	User Datagram Protocol
USB	Universal Serial Bus
VIS	Visible light
VRU	Vulnerable Road Users