UNIVERSITÀ DEGLI STUDI DI PARMA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

PARMA (I) - PARCO AREA DELLE SCIENZE 181A

Tel. 0521 – 905800 • Fax 0521 – 905758

Dottorato di Ricerca in Tecnologie dell'Informazione
XVII ciclo

Amos Tibaldi

Image processing techniques for the perception of automotive environments with applications to pedestrian detection.

DISSERTAZIONE PRESENTATA PER IL CONSEGUIMENTO
DEL TITOLO DI DOTTORE DI RICERCA

Gennaio 2006

Ai miei

cari genitori

Acknowledgments

First of all I would like to thank my tutor Professor Broggi for having supported and motivated me in all these years. I would like to thank my dear friends Luca and Massimo who have helped me a lot too and all the people of the DII department of Parma. In particular the folks of the Vision Laboratory that have always a lot to do but have found time for wonderful moments to spend together. Finally many thanks to my parents who have made this beautiful experience possible.

Contents

List of Figures

Introduction

If you don't know
where you are going,
any road will take you there.
Lewis Carroll

THIS thesis focuses on a very challenging problem which represents an extremely hot research topic and is being studied in almost all car manufacturers R&D departments.

The perception of the environment surrounding a road vehicle is of paramount importance for the design of active and passive safety systems. An optimized triggering of systems such as pre-crash belt tensioning or airbag blowing can lead to an improvement in car occupants safety.

Many different sensors, each one using its own technology, can be used to sense the surrounding environment in automotive applications. Each sensor delivers dif-

ferent kind of information depending on its technology: radars deliver distance and speed measurements, laser-scanners provide distance and shape information, sonars provide short distance estimations, while traditional daylight cameras provide information on the scene's brightness.

Indeed, information conveyed by the use of the artificial vision (cameras) may be used to estimate other measurements in addition to the color or brightness of the scene. In fact artificial vision can be used to measure distances, shapes, speeds and the presence of specific obstacles, as indirect measurements.

In other words the processing of image sequences can be used to provide information about the whole 3D space around the vehicle by using appropriate algorithms. Furthermore the decreasing cost of cameras and the increasing computational power available at a low cost are the main reasons that justify the widespread use of this technology as the main sensing device on vehicles.

After the above analysis, the work described in this thesis addresses the problem of sensing automotive environments by using artificial vision, in particular the techniques developed in this work were applied to the detection of the lane and to the detection of pedestrians in front of the vehicle.

This work is structured as follows: chapter one provides an overall description of the framework on which this thesis is based: the GOLD system, a software framework that allows fast application prototyping. Chapter two describes an innovative and evolutionary approach to lane markings detection. Chapters three, four, five and six focus on the detection of pedestrians: in particular chapter three presents techniques and results using a stereoscopic approach, chapter four describes a shape based

approach, and chapter five integrates the two previous approaches and extends them with tracking.

Finally chapter six describes the way in which quantitative performance have been determined: the tool developed and described in this last chapter refers to the collection of ground truth and its comparison to the results obtained by the pedestrian detection algorithms.

Each one of the following chapters is structured for an easy standalone reading, namely each of the chapters includes a brief introduction, the description of the core system and a final section with the discussion on results and performance. In other words the thesis conclusion is distributed throughout the different chapters.

Chapter 1

Experiments in Robotics for Intelligent Road Vehicles

Contenuto capitolo

This chapter presents the experience of the ARGO Project. It started in 1996 at the University of Parma, based on the previous experience within the European PROMETHEUS Project. In 1997 the ARGO prototype vehicle was set up with sensors and actuators, and the first version of the GOLD software system –able to locate one lane marking and generic obstacles on the vehicle's path– was installed. In June 1998 the vehicle underwent a major test (the *MilleMiglia in Automatico*, a 2000 km tour on Italian highways) in order to test the complete equipment. The analysis of this test allowed to improve the system. This chapter presents the current implementation of the GOLD system, featured by enhanced Lane Detection abilities and extended Obstacle Detection abilities, such as the detection of leading vehicles and pedestrians. Moreover it is described how this technology was transferred to the automatic driving of snowcats in extreme environments.

1.1 Introduction

The main target of the ARGO Project is the development of an active safety system with the ability to act also as an automatic pilot for a standard road vehicle.

In order to achieve autonomous driving capabilities on the existing road network with no need for specific infrastructures, a robust perception of the environment is essential. Although very efficient in some fields of application, active sensors –besides polluting the environment– feature some specific problems in automotive applications due to inter-vehicle interference amongst the same type of sensors, and due to the wide variation in reflection ratios caused by many different reasons, such as obstacles' shape or material. Moreover, the maximum signal level must comply with safety rules and must be lower than a safety threshold. For this reason in the implementation of the ARGO vehicle only the use of passive sensors, namely *cameras*, has been considered.

A second design choice was to keep the system costs low. These costs include both production costs (which must be minimized to allow a widespread use of these devices) and operative costs, which must not exceed a certain threshold in order not to interfere with the vehicle performance. Therefore low cost devices have been preferred, both for the image acquisition and the processing: the prototype installed on ARGO is based on *cheap cameras* and a *commercial PC.*

The following section present the main functionalities integrated on the ARGO vehicle:

- Lane Detection and Tracking

- Obstacle Detection
- Vehicle Detection and Tracking
- Pedestrian Detection.

1.2 The GOLD System

GOLD is the acronym used to refer to the software that provides ARGO with autonomous capabilities. It stands for Generic Obstacles and Lane Detection since these were the two functionalities originally developed. Currently it integrates two other functionalities: Vehicle Detection and Pedestrian Detection.

1.2.1 The Inverse Perspective Mapping

The Lane Detection and Obstacle detection functionalities share the same underlying approach: the removal of the perspective effect obtained through the Inverse Perspective Mapping (IPM) [1, 12].

The IPM is a well-established technique that allows to remove the perspective effect when the acquisition parameters (camera position, orientation, optics,...) are completely known and when a knowledge about the road is given, such as a *flat road hypothesis*. The procedure aimed at removing the perspective effect resamples the incoming image, remapping each pixel toward a different position and producing a new 2-dimensional array of pixels. The so-obtained *remapped image* represents a top view of the road region in front of the vehicle, as it were observed from a significant

height. Figures 1.1.a and 1.1.b show an image acquired by ARGO's vision system and the corresponding remapped image.

1.2.2 Lane Detection

Lane Detection functionality is divided in two parts: a lower level part, which, starting from iconic representations of the incoming images produces new transformed representations using the same data structure (array of pixels), and a higher level one, which analyzes the outcome of the preceding step and produces a symbolic representation of the scene.

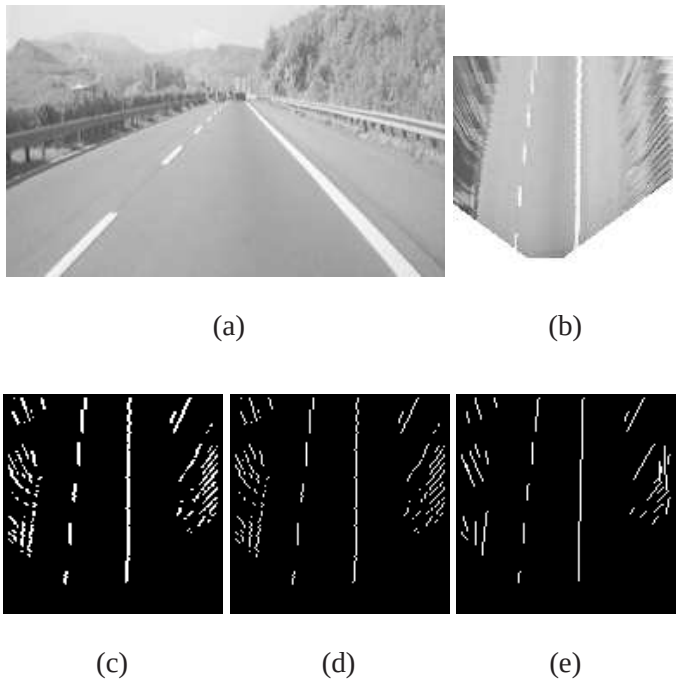
Low- and Medium-level Processing for Lane Detection

Lane Detection is performed assuming that a road marking in the remapped image is represented by a quasi-vertical bright line of constant width on a darker background (the road). Thus, pixels belonging to a road marking feature a higher brightness value than their left and right neighbors.

The first phase of road markings detection is therefore based on a filter able to detect dark-bright-dark transitions.

The brightness value of a generic pixel belonging to the remapped image is compared to the two horizontal left and right neighbors at a given distance. A new image (shown in figure 1.1.c), whose values encode the presence of a road marking, is computed assigning:

1. zero to the pixels whose one or both of the two neighbors have a higher bright-



(a) (b)

(c) (d) (e)

Figure 1.1: The sequence of images produced by the low-level Lane Detection phase: (*a*) original; (*b*) remapped; (*c*) enhanced; (*d*) binarized; (*e*) polylines.

ness value, or

2. the absolute difference between the pixel's brightness and their neighbors' ones to the pixels whose brightness is higher than the ones of the two neighbors.

Due to different light conditions (e.g. in presence of shadows), pixels representing road markings may have different brightness, yet maintaining their superiority relationship with their horizontal neighbors. Therefore, since a simple threshold seldom gives a satisfactory binarization, the image is enhanced exploiting its vertical correlation. Finally, the binarization is performed by means of an adaptive thresh-

old. [1]; the result is presented in figure 1.1.c. The binary image is scanned row by row in order to build chains of 8-connected non-zero pixels (see figure 1.1.d).

Subsequently, each chain is approximated with a *polyline* composed by one or few segments, by means of an iterative process. Initially, a single segment that joins the two extrema of the chain is considered. The horizontal distance between segment's mid point and the chain is used to determine the quality of the approximation. In case it is larger than a threshold, two segments sharing an extremum are considered for the approximation of the chain. Their common extremum is the intersection between the chain and the horizontal line that passes through the segment's mid point. The process is iterated until a satisfactory approximation has been reached (see figure 1.1.e).

High-level Processing for Lane Detection

in the high-level processing, the list of polylines is processed in order to semantically group homologous features and to produce a high level description of the scene.

Each polyline is compared against the result of the previous frame, since continuity constraints provide a strong and robust selection procedure. The distance between the previous result and each extremum of the considered polyline is computed: if all the polyline extrema lay within a stripe centered onto the previous result then the polyline is marked as useful for the following process. This process is repeated for both left and right lane markings.

Once the polylines have been selected, all the possibilities are checked for their joining. In order to be joined, two polylines must have similar direction; must not be

too distant; their projections on the vertical axis must not overlap; the higher polyline in the image must have its starting point within an elliptical portion of the image; in case the gap is large also the direction of the connecting segment is checked for uniform behavior. Figure 1.2 shows that polyline A cannot be connected to: B due to high difference of orientation; C due to high distance (does not lay within the ellipse); D due to the overlapping of their vertical projections; E since their connecting segment would have a strongly mismatching orientation. It can only be connected to F.

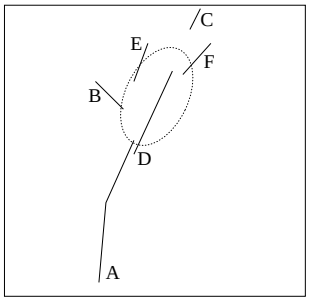


Figure 1.2: Joining of similar polylines.

All the new polylines, formed by concatenations of the original ones, are then evaluated. In case the polyline does not cover the whole image, a penalty is given. Then, the polyline length is computed and a proportional penalty is given to short ones, as well as to polylines with extremely varying angular coefficients. Finally, the polyline with the highest score is selected as the best representative of the lane marking.

The polyline that has been selected at the previous step may not be long enough

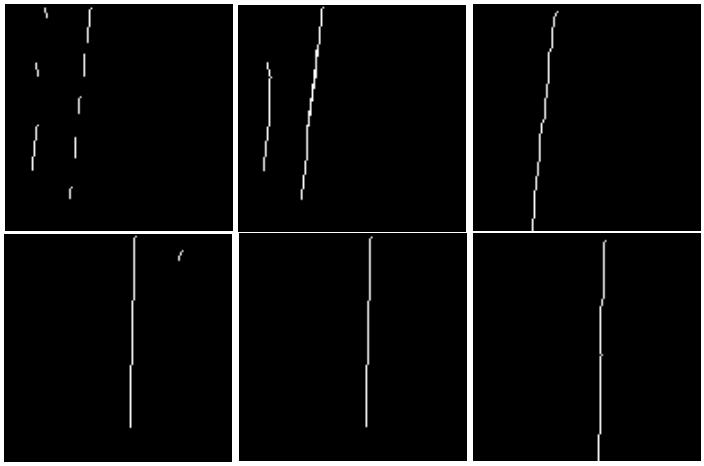


Figure 1.3: Filtered polylines, joined polylines, and model fitting for the left (upper row) and right (bottom row) lane markings.

to cover the whole image; therefore a further step is necessary to extend the polyline. In order to take into account road curves, a parabolic model has been selected to be used in the prolongation of the polyline in the area far from the vehicle. In the nearby area, a linear approximation suffices.

The two reconstructed polylines (one representing the left and one the right lane markings) are now matched against a model that encodes some more knowledge about the absolute and relative positions of both lane markings on a standard road.

The model is kept for reference: the two resulting polylines are fitted to this model and the final result is obtained as follows. First the two polylines are checked for non-parallel behavior; a small deviation is allowed since it may derive from vehicle movements or deviations from the flat road assumption, that cause the calibration to be temporarily incorrect (diverging of converging lane markings). Then the quality

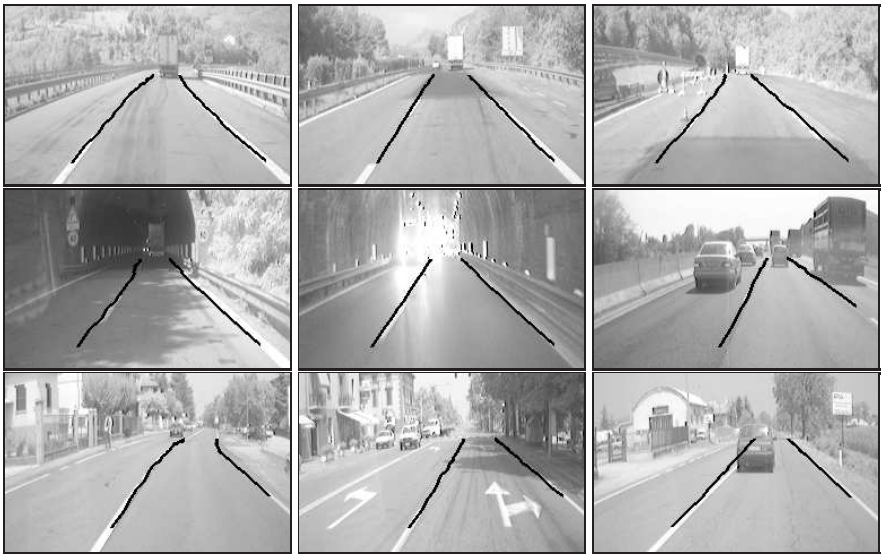


Figure 1.4: Some results of Lane Detection in different conditions.

of the two polylines, as computed in the previous steps, is matched: the final result will be attracted toward the polyline with the highest quality with a higher strength. In this way, polylines with equal or similar quality will equally contribute to the final result; on the other hand, in case one polyline has been heavily reconstructed, or is far from the original model, or is even missing, the other polyline will be used to generate the final result.

Finally, figure 1.3 presents the resulting images referring to the example presented in figure 1.1. It shows the results of the selection, joining, and matching phases for the left (upper row) and for the right (bottom row) lane markings.

Results of Lane Detection

This subsection presents a few results of lane detection in different conditions (see figure 1.4) ranging from ideal situations to road works, patches of non-painted roads, the entry and exit from a tunnel. Both highway and extra-urban scenes are provided for comparison; the systems proves to be robust with respect to different illumination situations, missing road signs, and overtaking vehicles which occlude the visibility of the left lane marking. In case two lines are present –a dashed and a continuous one–, the system selects the continuous one.

1.2.3 Obstacle Detection

The Obstacle Detection functionality is aimed at the *localization* of generic objects that can obstruct the vehicle's path, without their complete *identification* or *recognition*. For this purpose a complete 3D reconstruction is not required and a matching with a given model is sufficient: the model represents the environment without obstacles, and any deviation from the model detects a potential obstacle. In this case the application of IPM to stereo images [3], in conjunction with a-priori knowledge on the road shape, plays a strategic role.

Low-level Processing for Obstacle Detection

Assuming a *flat road* hypothesis, IPM is performed on both stereo images. The flat road model is checked computing a pixel-wise difference between the two remapped images. In correspondence to anything rising up from the road surface, the result

features sufficiently large clusters of non-zero pixels. Due to the stereo cameras' different angles of view, an ideal homogeneous square obstacle produces two clusters of pixels with a triangular shape in the difference image, in correspondence to its vertical edges [12].

Due to the texture, irregular shape, and non-homogeneous brightness of real obstacles, the detection of the triangles becomes difficult. Nevertheless, in the difference image some clusters of pixels with a quasi-triangular shape are anyway recognizable, even if they are not clearly disjointed. Moreover, in case two or more obstacles are present in the scene at the same time, more than two triangles appear in the difference image. A further problem is caused by partially visible obstacles which produce a single triangle. The low-level portion of the process, detailed in figure 1.5, is consequently reduced to the computation of difference between the two remapped images, a threshold, and a morphological opening aimed at removing small-sized details in the thresholded image.

Medium- and High-level Processing for Obstacle Detection

The following process is based on the localization of pairs of triangles in the difference image by means of a quantitative measurement of their shape and position [23].

A *polar histogram* is used for the detection of triangles: it is computed scanning the difference image with respect to a point called *focus* and counting the number of overthreshold pixels for every straight line originating from the focus. A low-pass filter is applied in order to decrease the influence of noise (see figure 1.5.f and 1.5.g). When the focus is placed in the middle point between the projection of the

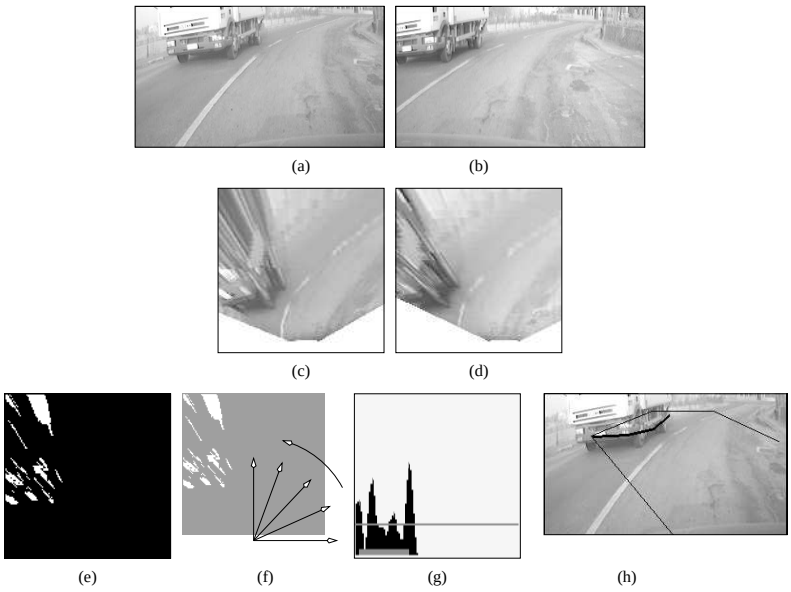


Figure 1.5: Obstacle Detection: (*a*) left and *(b)* right stereo images, (*c*) and *(d)* the remapped images, (*e*) the difference image, (*f*) the angles of view overlapped with the difference image, (*g*) the polar histogram, and (*h*) the result of Obstacle Detection using a black marking superimposed on the acquired left image; the thin black line highlights the road region visible from both cameras.

two cameras onto the road plane, the polar histogram presents an appreciable peak corresponding to each triangle [12]. Since the presence of an obstacle produces two disjointed triangles (corresponding to its edges) in the difference image, Obstacle Detection is limited to the search for pairs of adjacent peaks. The position of a peak in fact determines the angle of view under which the obstacle edge is seen (figure 1.6).

Peaks may have different characteristics, such as amplitude, sharpness, or width. This depends on the obstacle distance, angle of view, and difference of brightness and

Figure 1.6: Correspondence between triangles and directions pointed out by peaks detected in the polar histogram.

texture between the background and the obstacle itself. Two or more peaks can be joined according to different criteria, such as similar amplitude, closeness, or sharpness. The analysis of a large number of different situations made possible the determination of a weight function embedding all of the above quantities. According to the notations of figure 1.7, R is defined as the ratio between areas A_1 and A_2. If R is greater than a threshold, two adjacent peaks are considered as generated by the same obstacle, and then joined; otherwise, when the two peaks are far apart or the valley is too deep they are left alone (not joined). Figure 1.8 shows some examples of peak joining. Obviously, a partially visible obstacle produces a single peak that cannot be joined to any other. The amplitude and width of peaks, as well as the interval between joined peaks, are used to determine the angle of view under which the whole obstacle is seen.

The difference image is also used to estimate the obstacle distance. For each peak of the polar histogram a *radial histogram* is computed scanning a specific sector of the difference image. The width α_i of the sector is determined as the width of the polar histogram peak in correspondence to 80% of the peak maximum amplitude h_i.

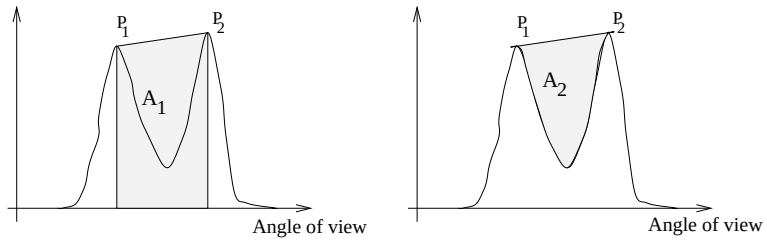


Figure 1.7: If the ratio between areas A_1 and A_2 is greater than a threshold, the two peaks are joined.

The number of overthreshold pixels is computed and the result is normalized. The radial histogram is analyzed to detect the corners of triangles, which represent the contact points between obstacles and road plane, therefore allowing the determination of the obstacle distance through a simple threshold.

Results of Obstacle Detection

Figure 1.10 shows the results obtained in a number of different situations. The result is displayed with black markings superimposed on a brighter version of the left image; they encode both the obstacles' distance and width.

1.2.4 Vehicle Detection

The Platooning task is based on the detection of the distance, speed, and heading of the preceding vehicle. Since Obstacle Detection does not generate sufficiently reliable results –in particular regarding obstacle distance–, a new functionality, Vehicle Detection, has been considered; the vehicle is localized and tracked using a single monocular image sequence.

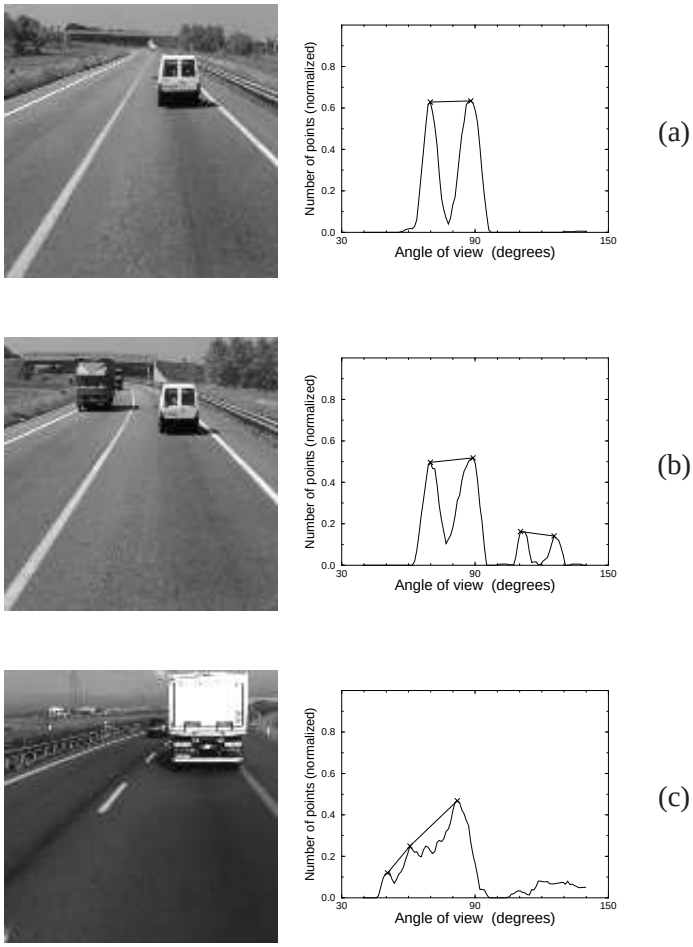


Figure 1.8: Some examples of peaks join: (*a*) one obstacle, (*b*) two obstacles and *(*c) a large obstacle.

The Vehicle Detection algorithm is based on the following considerations: a vehicle is generally symmetric, characterized by a rectangular bounding box which satisfies specific aspect ratio constraints, and placed in a specific region of the image. These features are used to identify vehicles in the image in the following way: first an

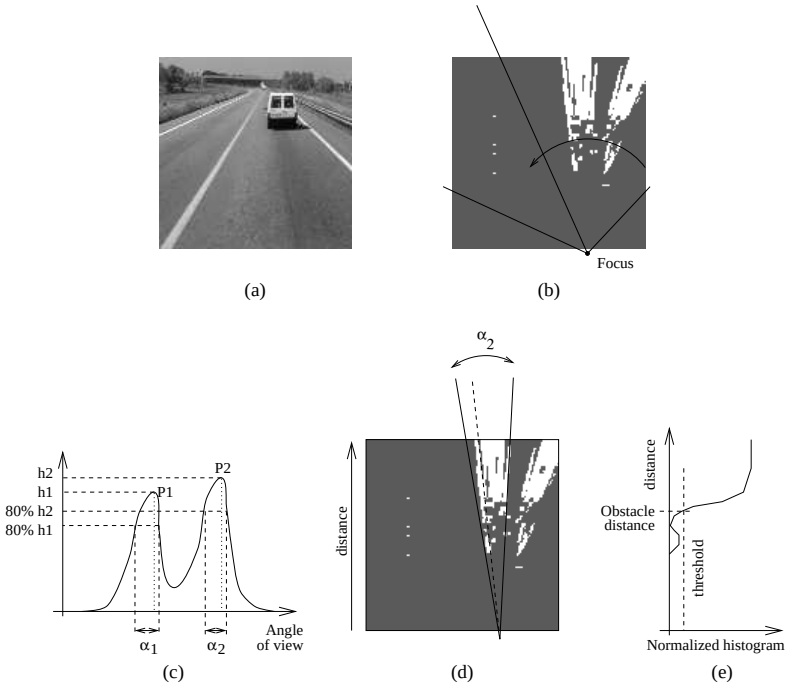


Figure 1.9: Steps involved during the computation of radial histogram for peak P_2: (*a*) original image; (*b*) binary difference image; (*c*) polar histogram; (*d*) sector used for the computation of the radial histogram; (*e*) radial histogram.

area of interest is identified on the basis of road position and perspective constraints. This area is searched for possible vertical symmetries; not only gray level symmetries are considered, but vertical and horizontal edges symmetries as well, in order to increase the detection robustness. Once the symmetry position and width has been detected, a new search begins, which is aimed at the detection of the two bottom corners of a rectangular bounding box. Finally, the top horizontal limit of the vehicle is searched for, and the preceding vehicle localized.

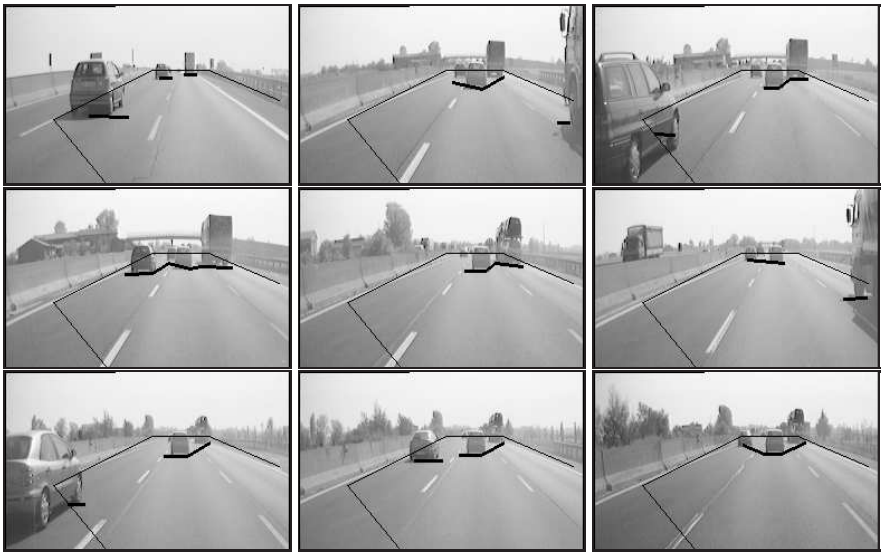


Figure 1.10: Obstacle Detection: the result is shown with a black marking superimposed onto a brighter version of the image captured by the left camera; a black thin line limits the portion of the road seen by both cameras.

The tracking phase is performed through the maximization of the correlation between the portion of the image contained into the bounding box of the previous frame (partially stretched and reduced to take into account small size variations due to the increment and reduction of the relative distance) and the new frame.

Symmetry detection

In order to search for symmetrical features, the analysis of gray level images is not sufficient. Strong reflections cause irregularities in vehicle symmetry, while uniform areas and background patterns present highly correlated symmetries. In order to get

rid of these problems, also symmetries in other domains are computed. In fact, to get rid of reflections and uniform areas, edges are extracted and thresholded, and symmetries are computed into this domain as well. Similarly, the analysis of symmetries of horizontal and vertical edges produces other symmetry maps, which –with specific coefficients detected experimentally– can be combined with the previous ones to form a single symmetry map. Figure 1.11 shows all symmetry maps and the final one, that allows to detect the vehicle. For each image, the search area is shown in dark gray and the resulting vertical axis is superimposed. For each image its symmetry map is also depicted. Bright points in the map encode the presence of high symmetries. The 2D symmetry maps are computed by varying the axis' horizontal position within the grey area (shown in the original image) and the symmetry horizontal size. The lower triangular shape is due to the limitation in scanning large horizontal windows for peripheral vertical axes.

Bounding box detection

After the localization of the symmetry, the width of the symmetrical region is checked for the presence of two corners representing the bottom of the bounding box around the vehicle. Perspective constraints as well as size constraints are used to reduce the search. Figure 1.12 presents the results of the lower corners detection. This process is followed by the detection of the top part of the bounding box, which is looked for in a specific region whose location is again determined by perspective and size constraints.

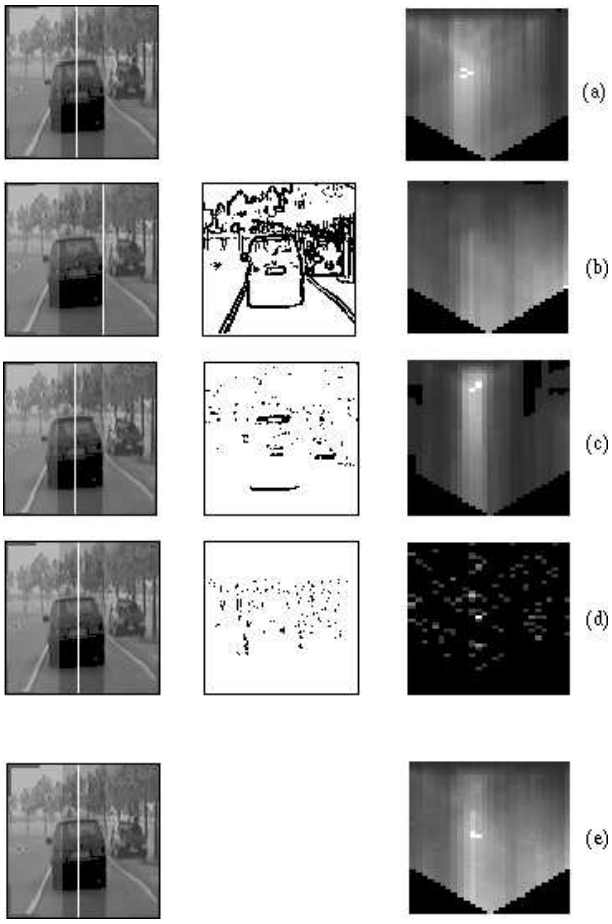


Figure 1.11: Computing the resulting symmetry: (*a*) grey-level symmetry; (*b*) edge symmetry; (*c*) horizontal edges symmetry; (*d*) vertical edges symmetry; (*e*) total symmetry. For each row the resulting symmetry axis is superimposed onto the leftmost original image.

Backtracking

Sometimes it may happen that in correspondence to the symmetry maximum no correct bounding boxes exist. Therefore, a backtracking approach is used: the symmetry

(a) (b) (c)

Figure 1.12: Detection of the lower part of the bounding box: (*a*) original image with superimposed results; (*b*) edges; (*c*) localization of the two lower corners.

map is again scanned for the next local maximum and a new search for a bounding box is performed.

Results of Vehicle Detection

Figure 1.13 shows some results of vehicle detection in different situations.

1.2.5 Pedestrian Detection

The latest functionality integrated in the ARGO prototype vehicle is aimed at detecting pedestrians in road environments. The system is able to localize pedestrians in various poses, positions and clothing, and is not limited to moving people.

The processing is divided in two different stages. Initially, attentive vision techniques relying on the search for specific characteristics of pedestrians such as vertical symmetry and strong presence of edges, allow to select interesting regions likely to contain pedestrians. Then, such candidates areas are validated verifying the actual presence of pedestrians by means of an shape detection technique based on the application of autonomous agents.

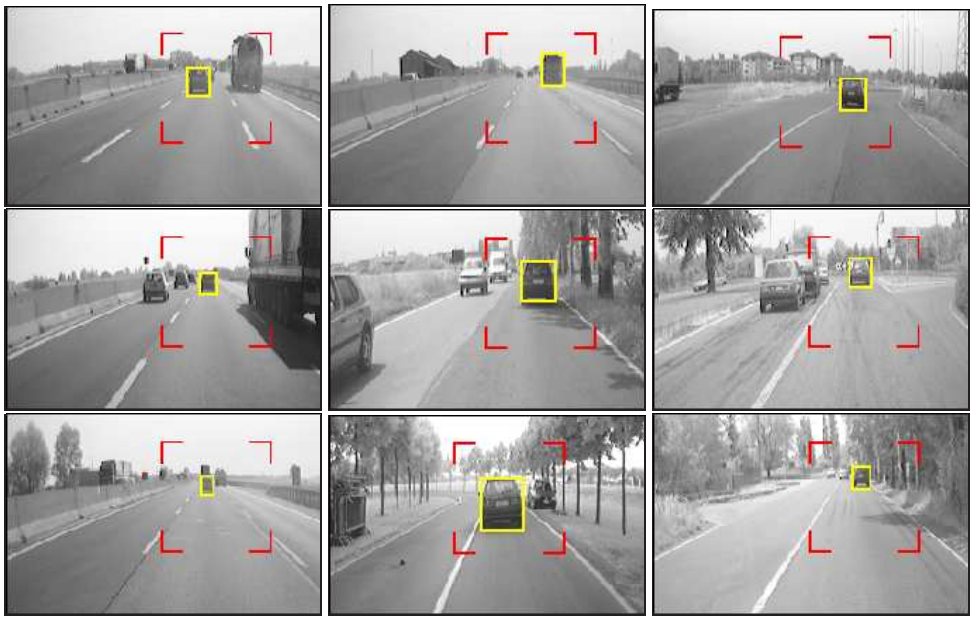


Figure 1.13: Results of Vehicle Detection in different road scenes.

Attentive vision

The areas considered as candidate in the first step are rectangular bounding boxes which:

- have a size in pixels deriving from the knowledge of the intrinsic parameters of the vision system;
- enclose a portion of the image which exhibits a strong vertical symmetry and a high density of vertical edges.

The search for candidates would require an exhaustive search in the whole image. However, the knowledge of the system's extrinsic parameters, together with a flat

scene assumption, is exploited to limit the analysis to a stripe of the image. The displacement of this stripe depends on the pedestrian's distance, while its height is related to the pedestrian's height. Indeed, the analysis cannot be limited to a fixed size and distance of the target and a given range for each parameter is in fact explored.

A pre-attentive filter is applied, aimed at the selection of the areas with a high density of edges. Then, for each vertical symmetry axis lying in these areas the best candidate area is selected among the bounding boxes which share that symmetry axis, while having different position (base) and size (height and width). Vertical symmetry has been chosen as a main distinctive feature for pedestrians. Alternatively, two different symmetry measures are performed: one on the gray-level values and one on the gradient values, considering only edges with a vertical direction The selection of the best bounding box is based on maximizing a linear combination of the two symmetry measures, masked by the density of edges in the box. Figure 1.14 shows the original input image, the result of a clustering operation used to improve the detection of edges, a binary image containing the vertical edges, and a number of histograms representing the maximum (i) symmetry of gray-levels, (ii) symmetry of vertical edges, and (iii) density of vertical edges among the bounding boxes examined for each axis. The histogram in figure 1.14.g represents the linear combination of all the above. It is evident that, using the density of vertical edges as a mask, interesting areas present high values for both the symmetry of gray-levels and symmetry of vertical edges. The resulting histogram is therefore thresholded and its overthreshold peaks are selected as representing candidate bounding boxes.

Shape detection using autonomous agents

The outcome of the low-level processing is a list of candidate bounding boxes which is fed to the following stage, whose task is their validation as pedestrians, based on higher-level characteristics. Different edges are selected and connected, where possible, in order to form a contour. Essentially, the process consists in adapting a deformable coarse model to the bounding box. Thanks to its roughness the model is sufficiently general and can be adapted to a variety of postures. Anyway, it is limited to standing pedestrians.

The model adjustment is done through an evolutionary approach with a number of independent agents acting as edge trackers. The agents explore a feature map displaying the edges contained in a given bounding box and stochastically build hypotheses of a feasible contour of a human. The idea is taken from the Ant Colony Optimization (ACO) metaheuristic originally inspired by the communication behavior of real ants [20].

This model can be applied to the analysis of an image by creating a colony of artificial ants that looks for an optimal combination of edge pixels that maximizes the coherency of their position according to a given model (see figure 1.15). Each ant in turn traces a solution in a solution space made up of all the possible paths connecting two pixels in a matrix. The decisional basis for each step of an ant is provided by two factors: one is a local heuristic that quantifies the attractiveness of pixel for its intrinsic characteristics; the second is the information on that pixel made available by previous attempts of other ants, in the form of a quantity of pheromone.

The world is visited by a number of ants in parallel, and the process is repeated for several cycles. At the end of each cycle, new pheromone is deposed on the trails pursued by the ants, and some of that accumulated evaporates. In this way, solutions built several cycles before, progressively loose their importance. On the other hand, pheromone on pixels that compose the path of frequently selected solutions grows. and eventually this information surpasses that given by the heuristic. Finally, the output is the path of the ant of the highest rank in the last cycle.

Results of Pedestrian Detection

This algorithm suits a medium distance search area. In fact, large bounding boxes may contain a too detailed shape, showing many disturbing small details that would certainly make their detection extremely difficult. On the other hand, very small bounding boxes enclosing far away pedestrians feature a very low information content. In these situations it is easy to obtain false positives, since many road participants (other than pedestrians), other objects, and even road infrastructures may present morphological characteristics similar to a human shape. With the current setup the search area ranges from 10 to 30 m.

The candidate selection procedure based on vertical symmetry and edge density proved to be a robust technique for focusing the attention on interesting regions. As an example, figure 1.16 shows the result of the selection of candidate bounding boxes in three different situations. Some general considerations can be drawn. In situations in which pedestrians are sufficiently contrasted with respect to the background and completely visible the localization of candidates proves to be robust. Thanks to the

use of vertical edges the width of the bounding boxes enclosing pedestrians is generally determined with a good precision. On the other hand, a lower accuracy is obtained for the localization of the top and bottom of the bounding box. A refinement of the bounding box height is under development. Symmetrical objects other than pedestrians may happen to be detected as well. In order to get rid of such false positives a number of filters have been devised which rely on the analysis of the distribution of edges within the bounding box. These filters, which are still under evaluation, show promising results regarding the elimination of both artifacts (such as poles, road signs, buildings, and other road infrastructures) and symmetrical areas given by a uniform portion of the background between two foreground objects with similar lateral borders (see figure 1.16.c).

From the first preliminary results, the ant-based processing appears to be a promising method for detecting the contour of a human shape. To extend the detection to a larger set of pedestrian postures, other models are currently under development.

1.3 The ARGO Prototype Vehicle

ARGO, shown in figure 1.17, is an experimental autonomous vehicle equipped with vision systems and an automatic steering capability.

It is able to determine its position with respect to the lane, to compute the road geometry, to detect generic obstacles on the path, and to localize a leading vehicle and pedestrians. The images acquired by a stereo rig placed inside the cabin are analyzed in real-time by a computing system located into the boot. The results of the

processing are used to drive an actuator mounted onto the steering wheel and other assistance devices.

The system was initially conceived as a safety enhancement unit: in particular it is able to supervise the driver behavior and issue both optic and acoustic warnings or even take control of the vehicle when dangerous situations are detected. Further developments have extended the system functionalities to fully automatic driving capabilities.

Thanks to a control panel the driver can select the level of system intervention. The following three driving modes are integrated.

- **Manual Driving:** the system simply monitors and logs the driver's activity.
- **Supervised Driving:** in case of danger, the system warns the driver with acoustic and optical signals.
- **Automatic Driving:** the system maintains the full control of the vehicle's trajectory, and the two following functionalities can be selected: *Road Following:* consisting of the automatic movement of the vehicle inside the lane; or *Platooning:* namely the automatic following of the preceding vehicle.

1.4 The *MilleMiglia in Automatico* Test

In order to extensively test the vehicle under different traffic situations, road environments, and weather conditions, a 2000 km journey was carried out in June 1998. Other prototypes were tested on public roads with long journeys (CMU's Navlab *No*

Hands Across America, and a tour from Munich to Odense organized by the Universität der Bundeswehr, Germany) whose the main

differences were that the former was relaying also on non-visual information (therefore handling occlusions in a different way) and that the latter was equipped with complex computing engines.

The *MilleMiglia in Automatico* test was carried out about 2 years ago, and the system was much more primitive than it is currently. Only Lane Detection and Obstacle Detection were tested: Lane Detection was based on the localization of a single line, while the detection of the preceding vehicle was performed by the Obstacle Detection module; no tracking was done and only the Road Following functionality was available.

1.5 Discussion and Technology Transfer

The functionalities, the algorithms, and –more generally– the experience developed within the ARGO project were transferred to different domains. One of them is the automatic driving of a snowcat in extreme environments. In this project, founded by ENEA, visual information acquired from the driving cabin of a snowcat are used to localize the tracks of preceding vehicles, with the aim of following them as precisely as possible. The reason is that cracks in the ice can put in serious danger both the driver and the snowcat itself. Therefore it is imperative that the vehicle follows the same precise path defined by preceding vehicles.

Due to the extreme conditions of the working environment –where temperatures

can reach even -80 degrees Celsius, the terrain is completely covered by snow or ice, strong sun lighting and reflections may be present, and no specific ground references are available nor assumptions can be made on the terrain slope– this application is extremely challenging and presents many additional problems with respect to the driving of unmanned vehicles on traditional (un)structured roads.

Figure 5.9 shows some results of snowcat track detection in different conditions. The algorithm [14], not discussed in this paper, is able to successfully detect the tracks even in noisy or critical conditions such as shadows, sun reflections, unknown terrain slope, and when dark objects are present as well.

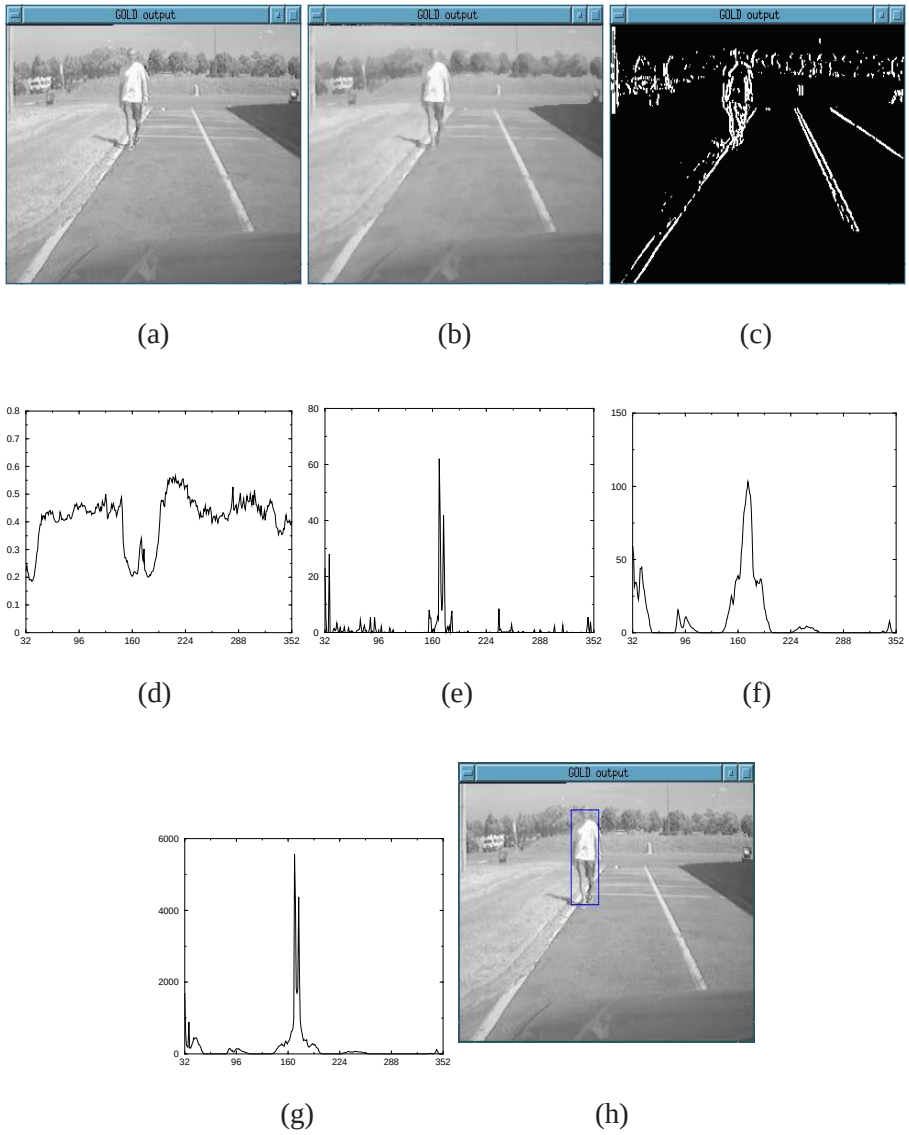


Figure 1.14: Intermediate results leading to the localization of bounding boxes: *(a)* original image; *(b)* clusterized image; *(c)* vertical edges; *(d)* histogram representing grey level symmetries; *(e)* histogram representing vertical edges symmetries; *(f)* histogram representing vertical edges density; *(g)* histogram representing the overall symmetry S for the best bounding box for each column; *(h)* the resulting bounding box.

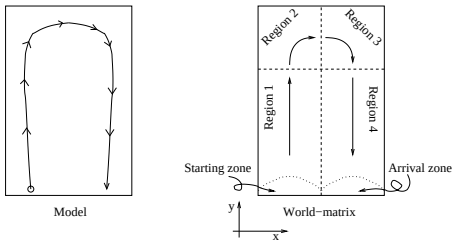


Figure 1.15: Artificial ants move through the world-matrix starting from the left half of the lower border, and moving through regions 1, 2, 3 and 4 until they reach the arrival line.

(a) (b) (c)

Figure 1.16: Result of low-level processing in different situations: *(a)* a correct detection of two pedestrians *(b)* a complex scenario in which only the central pedestrian is detected; the left one is confused with the background, the right one is only partially visible, while the high symmetry of a tree has been detected as well; *(c)* two crossing pedestrians have been localized, but other symmetrical areas are highlighted as well.

Figure 1.17: The ARGO prototype vehicle.

Figure 1.18: The prototype vehicle during a test in the Italian test site.

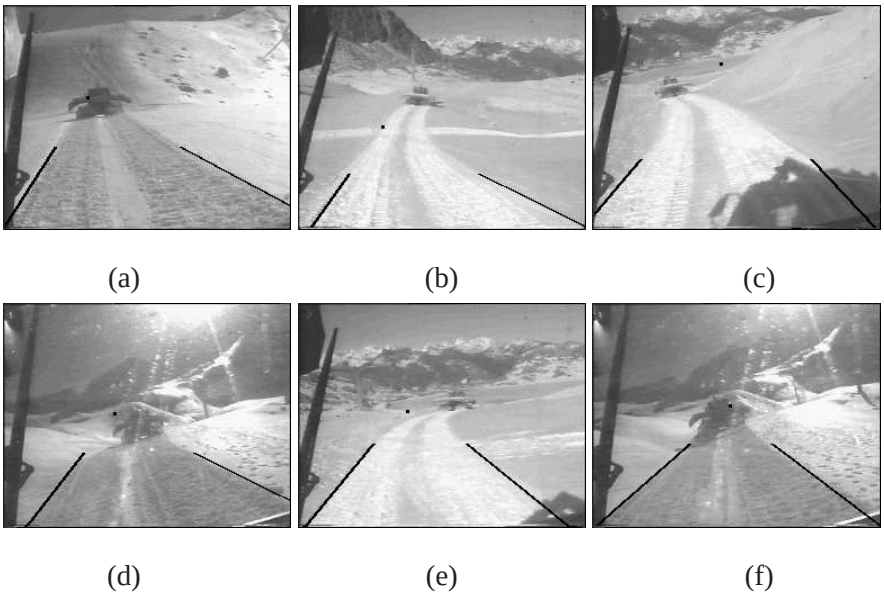


Figure 1.19: Results of snowcat track detection in different conditions.

Chapter 2

An Evolutionary Approach to Lane Markings Detection in Road Environments

Contenuto capitolo

This chapter presents the application of an evolutionary technique to lane markings detection in road environments. The aim is the localization of the path in images acquired by a vision system installed on-board of a vehicle for driving assistance or automation purposes. The first step of the procedure is the removal of the perspective effect from the images. The resulting bird's eye view image is analyzed by means of *ant* agents able to locate the lane markings. Results are compared against the ones obtained thanks to a deterministic approach.

2.1 Introduction

An autonomous intelligent vehicle has to perform a number of functionalities. Among them *Lane Detection* plays a basic role. A number of research groups has developed Lane Detection systems using artificial vision [16, 30, 37, 41].

The visual perception of the road environment is a challenging task: the knowledge of the lane position has to be extracted from visual patterns detected in the images. In the localization of specific features such as road markings painted on the road surface, basic problems have to be faced:

- shadows (projected by trees, buildings, bridges, or other vehicles) may produce artifacts onto the road surface, and thus alter the road texture;
- the system has to be robust enough to cope with situations where lane markings are worn and partly missing;
- the system should be enough flexible to adapt to different road environments.

This chapter presents an approach aimed at the identification of lane markings in road images by means of *collaborative autonomous agents*.

The action paradigm of the agents has been conceived on the basis of the behavior of real ants that seek for food: each ant has the task of exploring the world, locating food, and signalling to other ants the path toward food. Ant leave the nest and explore the world in a stochastic way. When they find a place with food, they mark the path from the nest to that place with pheromone. Pheromone attracts other ants, swiftly leading them to food. In this way paths leading to regions rich of food will

attract more and more ants. At the same time, pheromone evaporates as time passes, avoiding to mislead ants to old places where food is already exhausted or toward not enough profitable paths [5, 20, 21].

In this approach, autonomous agents (the *ants*) explore the image looking for lane markings (the food). Digital pheromone is used to mark best paths.

This chapter is organized as follows: section 2.2 illustrates principles and data representation of the agent paradigm. Section 2.3 introduces the model involved with agents' supervision, while section 2.4 presents significant results and touches possible improvements of the system.

2.2 Lane detection algorithm

A camera installed onto the ARGO prototype vehicle [12] is used to obtain images of the road (see figure 2.1.a).

2.2.1 Pre-processing phase

The initial processing step is the removal of the perspective effect [3]. Thanks to the knowledge of the camera calibration and to the assumption of a flat road in front of the vehicle, pixel are remapped onto a new domain. Resulting images represent a bird's eye view of the road (see figure 2.1.b). In these images lane markings are nearly-vertical bright lines surrounded by a darker background. Hence, a specific adaptive filtering is used to extract quasi-vertical bright lines (see figure 2.1.c) [4, 11, 12].

The result is a binary image where overtrheshold pixels represent lane markings.

2.2.2 The motion domain

An evolutionary approach with a number of independent agents acting as lane markings trackers is used for detecting lane markings. Agents explore the resulting binary image and stochastically detect the markings. The idea derives from the *Ant Colony Optimization* meta-heuristic, devised to solve hard combinatorial optimization problems, originally inspired by the communication behavior of real ants [20].

The ants' motion domain $\mathcal{D}$ is the binary image obtained from the pre-processing phase: overthreshold pixels represent ant's food, namely a *food-field* mapping defined as $\mathcal{F} : \mathcal{D} \mapsto \{f_{\text{yes}}; f_{\text{no}}\}$. Initially, the pheromone level $\mathcal{P} : \mathcal{D} \mapsto [0..\mathbb{R}^+]$ is 0 for each element in $\mathcal{D}$. Anyway, $\mathcal{P}$ is continuously updated by ants, while $\mathcal{F}$ is constant during the processing.

The domain $\mathcal{D}$ is recursively explored by subsequent batches of ants. The first ant of the batch enters $\mathcal{D}$ from the bottom in a random position ($\mathbf{a} = (x_{\text{ant}}, y_{\text{ant}})$ with $y_{\text{ant}} = 0$). Since lane markings are nearly vertical lines, at each step, the ant performs a single pixel movement along the vertical axis toward the top end of the image. Thus vertical movements are fully deterministic, while horizontal movements are stochastically computed according to the rules described in the following paragraph.

2.2.3 Evolutional algorithm

The horizontal position of an ant (x_{ant}) is modified according to values of $\mathcal{F}$ and $\mathcal{P}$ into sub-domain $\mathcal{N} \stackrel{\bullet}{=} \{\mathbf{a} \in \mathcal{D} \mid \quad y = 1 + y_{\text{ant}}, x_{\text{ant}} - \rho < x < x_{\text{ant}} + \rho\}$, where ρ represents the lateral field of view of each agent. In the current implementation of the

(a) (b) (c)

Figure 2.1: Initial steps of the processing: (*a*) original image, (*b*) removal of the perspective effect and (*c*) binary result.

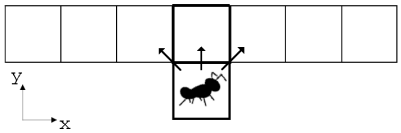
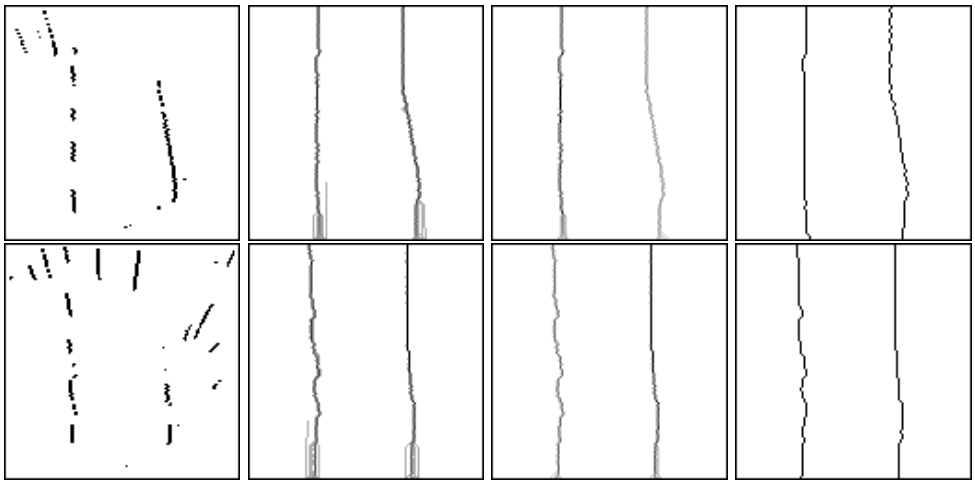


Figure 2.2: The $\mathcal{N}$ domain and possible ant's moves.

algorithm $\rho = 3$ as shown in figure 2.2.

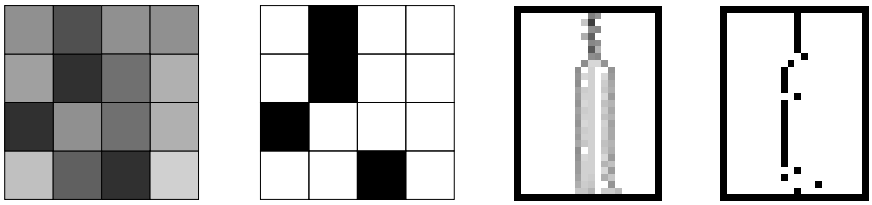
For each pixel $\mathbf{n} \in \mathcal{N}$ a quality parameter $w_{\mathcal{N}}(\mathcal{F}(\mathbf{n}), \mathcal{P}(\mathbf{n}))$ is computed as:

$$w_{\mathcal{N}} = \begin{cases} \alpha \times \mathcal{P}(\mathbf{n}) + \beta \times \mathcal{F}(\mathbf{n}) + \gamma \times (x_{\mathbf{n}} - x_{\text{ant}}) \times \mathcal{P}(\mathbf{n}) & \text{where} \quad \mathcal{F}(\mathbf{n}) = f_{\text{yes}} \wedge \mathcal{P}(\mathbf{n}) \neq 0 \\ 2 \times \alpha - 3 \times \gamma \times (x_{\mathbf{n}} - x_{\text{ant}}) & \text{where} \quad \mathcal{F}(\mathbf{n}) = f_{\text{yes}} \wedge \mathcal{P}(\mathbf{n}) = 0 \\ \alpha \times \mathcal{P}(\mathbf{n}) + \gamma \times (x_{\mathbf{n}} - x_{\text{ant}}) \times \mathcal{P}(\mathbf{n}) & \text{where} \quad \mathcal{F}(\mathbf{n}) = f_{\text{no}} \wedge \mathcal{P}(\mathbf{n}) \neq 0 \\ \begin{cases} 0 & \text{where} \quad |x_{\mathbf{n}} - x_{\text{ant}}| > 1 \\ 2 - |x_{\mathbf{n}} - x_{\text{ant}}| & \text{elsewhere} \end{cases} & \text{where} \quad \mathcal{F}(\mathbf{n}) = f_{\text{no}} \wedge \mathcal{P}(\mathbf{n}) = 0 \end{cases} \tag{2.1}$$



(a) (b) (c) (d)

Figure 2.3: Two examples of ants paths: (*a*) motion domain, (*b*) ant's presence level, (*c*) pheromone map $\mathcal{P}(\mathcal{D})$, and (*d*) detected markings.



(a) (b) (c) (d)

Figure 2.4: Marking selection: (*a*) synthetic example of pheromone distribution, the darker the pixel the higher the pheromone value, (*b*) selected markings, namely pheromone maxima for (*a*), (*c*) real example of pheromone distribution and (*d*) selected markings for (*c*).

being α, β and γ empirically computed values.

The higher $w_{\mathcal{N}}$, the higher the pixel's *attraction* for ants. Therefore, an ant tends

```
foreach x ∈ D
    update F(x)
    reset P(x)
endfor
repeat following n̄ times
    foreach track in {left; right}
        update_starting_point(track)
        with every ants.v
            food_places_visited ← 0
            consecutive_empty ← 0
            path_length ← W
            k ← 0
            cycle row through {0..W}
                ant_step(a^k, F, P, track)
                path[k] ← a^k
                if F(a^k) = F_yes
                    increase empty_place_visited
                fi
                if on_lateral_border(a^k)
                    path_length ← row
                    next cycle break
                fi
                kill_path_if(consecutive_empty, food_places_visited)
                k ← k + 1
            endcycle
            max_food_places_visited max= food_places_visited
        endwith
        with every ants.v
            if food_places_visited = max_food_places_visited
                raise( P(path[*]) )
            fi
    endforeach
```

Figure 2.5: Main processing flow illustrated using a pseudo-language.

to *move left*, *stay in center*, or *move right* according to $w_{\mathcal{N}}$ belonging to pixels on the left side, in front, or on the right side respectively. More precisely, an attraction index (τ) for the three possible horizontal moves is computed as:

$$\begin{aligned} \tau_{\text{left}} &= \sum_{x_{\mathbf{n}}<x_{\text{ant}}} w_{\mathcal{N}}(\mathbf{n}) \\ \tau_{\text{center}} &= \sum_{x_{\mathbf{n}}=x_{\text{ant}}} w_{\mathcal{N}}(\mathbf{n}) \\ \tau_{\text{right}} &= \sum_{x_{\mathbf{n}}>x_{\text{ant}}} w_{\mathcal{N}}(\mathbf{n}) \end{aligned} \tag{2.2}$$

A truly random parameter u where $0 \leq \mathsf{u} \leq \tau_{\text{left}} + \tau_{\text{center}} + \tau_{\text{right}}$ is used to introduce a stochastic behavior in horizontal movements. When $u < \tau_{\text{left}}$ a left movement is chosen, when $\tau_{\text{left}} \leq u \leq \tau_{\text{left}} + \tau_{\text{center}}$ the central pixel is chosen, otherwise a right move is performed.

In order to speed up computation, invalid or scarce-food paths are immediately discarded: when an ant reaches lateral borders of $\mathcal{D}$ or when too many consecutive path's positions have $\mathcal{F} = f_{\text{no}}$, the ant is eliminated. This aspect is clearly evident in figure 2.3.b where ant's presence shows a number of ending paths.

The ant action performed during the step k has been implemented into the routine `ant_step(...)`, summarized in figure 2.5.

2.2.4 Batch processing

When an ant reaches the top end of the image, another ant of the batch enters $\mathcal{D}$. At the end of the batch, the path of the ant that ran across the highest number of lane markings pixels is chosen as the best, and the pheromone level $\mathcal{P}$ of all of the pixels belonging to this path is increased by a unit.

Thanks to the iteration of this procedure on a number of batches, the attraction of the best paths becomes greater and greater. The number of ants that stepped over a pixel of $\mathcal{D}$ is shown in figure 2.3.b. Figure 2.3.c depicts the pheromone deposed at

the end of the recognition phase.

As shown in figure 2.4, at the end of the image processing, the pixel representing the marking is selected for each row as the pixel that features the maximum of $\mathcal{P}(x)$, with x belonging to the row. No thinning procedure is needed, since for each row there is only a single maximum; on the other side, this approach does not guarantee that detected markings would be continuous (see figure 2.4.d).

Since the final target is to find the lane, thus both left and right markings, two different processings are performed for locating the left and the right lane markings, the only difference being the initial position of ants used for crossing $\mathcal{D}$. The assumption that the initial position of right and left markings is in the right and left half portion of $\mathcal{D}$ respectively is used. Ants used for detecting the left marking enter $\mathcal{D}$ in a random positions within an interval $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ in the left portion of the image bottom. Analogously, the detection of the right marking starts from the right side of the bottom of $\mathcal{D}$ within the $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ interval. The distance between left and right markings prevents the ants to reach the other marking allowing a reliable separate detection.

2.3 Lane tracking

At the beginning of the processing, the vision system is assumed to be centered inside the lane. Therefore entrance intervals $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ and $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ of ants batches are centered in given positions symmetrical with respect to the center of the image (see figure 2.6.a).

A simple tracking is performed in order to cope with lateral vehicle movements inside the lane and to take advantage of the high temporal correlation amongst subsequent frames. The assumption that the car is always oriented along the road direction

is used and only slow lateral shifts and lane changes are considered. Strong correlation is then expected between position of markings in subsequent images.

$\mathcal{E}_{\mathcal{D}}^{\text{left}}$ and $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ are updated according to the result of the processing. The average value of left and right lane markings abscissa within a bottom band of $\mathcal{D}$ is computed (see figure 2.6.b). Resulting values for left and right markings are used to move $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ and $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ where next frame markings are supposed to be found (see figure 2.6.c). This mechanism allows to also cope with varying width lanes.

A more complex strategy is used when the driving system is executing a lane change. In such a case $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ or $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ move outside $\mathcal{D}$. When this event occurs, namely when the left or the right marking leaves the field of view, a new marking is supposed to enter the image from the opposite side. The new marking is searched for assuming the same width for different lanes. At each step the lane width W_l is computed as the average distance between $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ and $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ in the previous 5 frames. For example, when executing a left lane change, $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ and $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ move rightward until $\mathcal{E}_{\mathcal{D}}^{\text{right}}$ exits $\mathcal{D}$. Then $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ is assumed as the new $\mathcal{E}_{\mathcal{D}}^{\text{right}}$, while a new $\mathcal{E}_{\mathcal{D}}^{\text{left}}$ is placed at

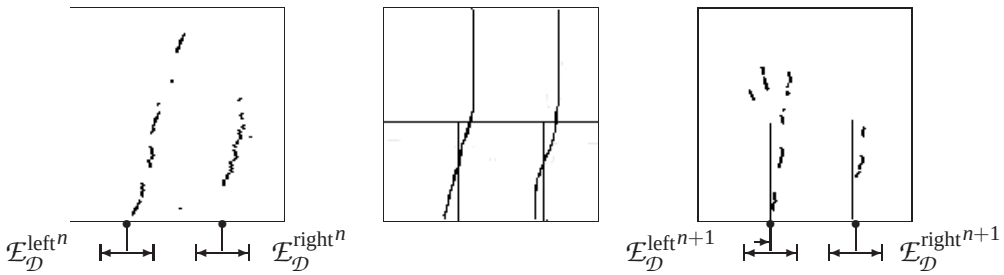


Figure 2.6: Update of entrance intervals: (*a*) initial position of intervals, (*b*) computation band used for determining the new position and (*c*) new intervals positions.

distance W_l on the left.

2.4 Discussion

The system has been implemented and tested on an Athlon 1.3 GHz architecture using Linux. The whole processing can be carried out in 8.6 ms, namely at a 116 Hz rate without considering image acquisition.

The algorithm has been tested and proven to be robust in different conditions: without obstacles, on straight and curved roads, in different illumination conditions. Figure 2.7 shows a number of partial and final results of the processing. Figures 2.7.a and 2.7.b show the original image before and after the removal of the perspective effect, while figures 2.7.c and 2.7.d present the final result superimposed on 2.7.a and 2.7.b.

The most critical behavior corresponds to the absence of markings in presence of very strong curves and to the presence of vehicles that occlude markings. Figure 2.8 shows such a situation: the border of a marking-occluding vehicle is wrongly detected as the lane marking. Nevertheless, the misdetection only affects the portion of the image where no marking is detectable, while the visible portion of the marking is always correctly detected.

Figure 2.9 shows a comparison between the results obtained through this approach and the ones obtained by a previously developed fully-deterministic algorithm [4]. In particular, the most critical situation for the deterministic approach is the presence of obstacles occluding lane markings that could lead to the misdetection of the whole marking as shown in the first row of figure 2.9.a. Conversely, the new

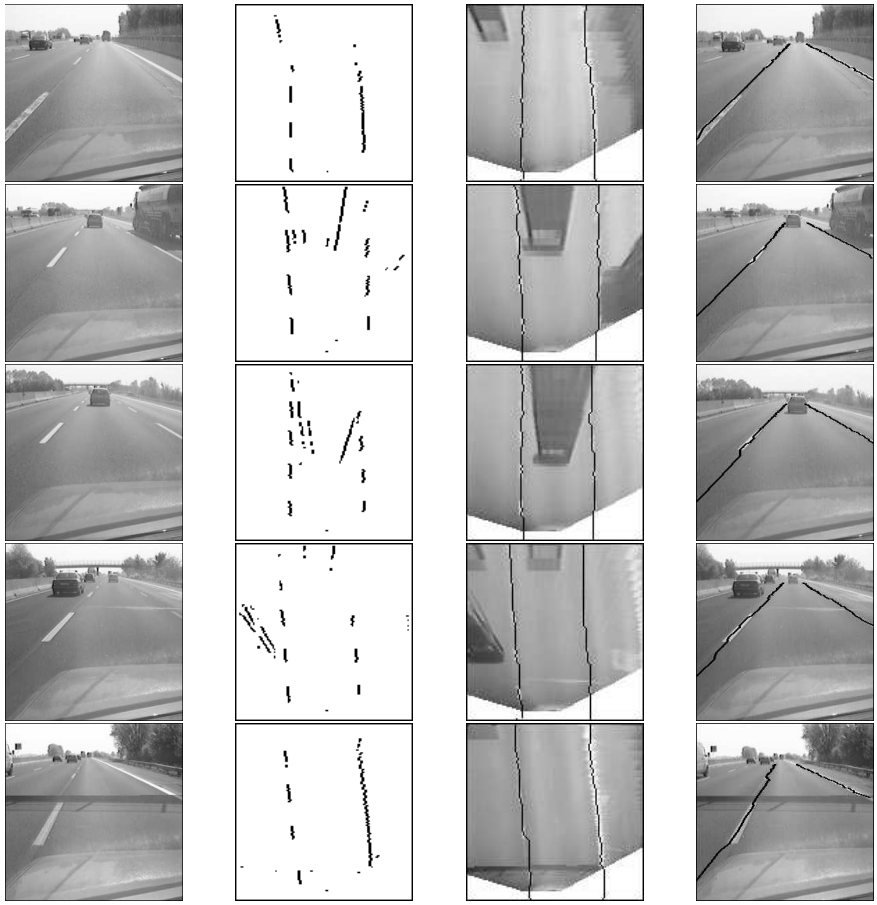
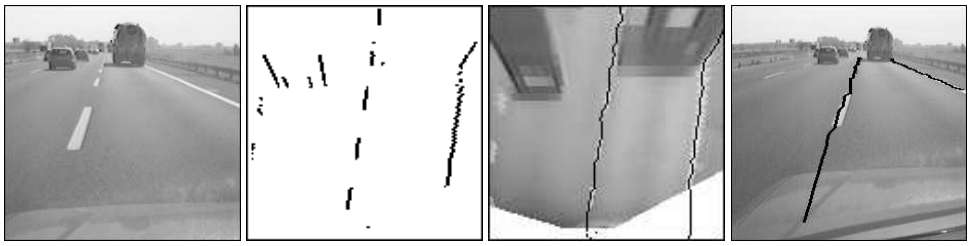
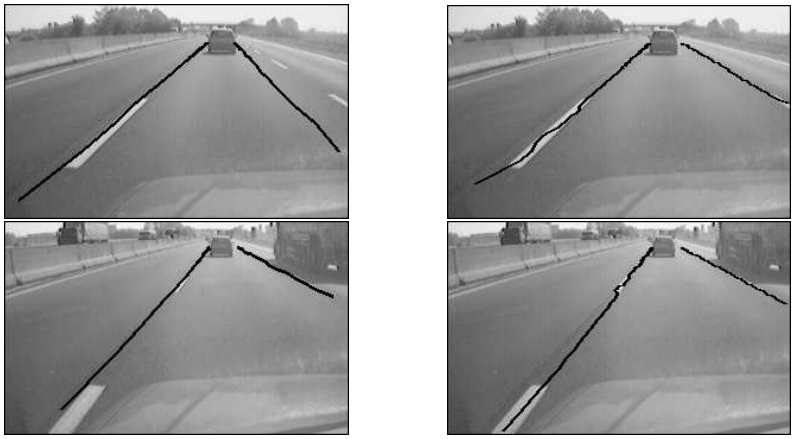


Figure 2.7: Overview of the processing: (*a*) original image, (*b*) binarization after the removal of the perspective effect, (*c*) result superimposed on the image without the perspective effect and (*d*) results superimposed onto original image.



(a) (b) (c) (d)

Figure 2.8: Example of critical situation (occluding obstacle): (*a*) original image, (*b*) binarized, (*c*) result superimposed onto the image obtained by the removal of the perspective effect and (*d*) result superimposed onto original image.



(a) (b)

Figure 2.9: Different approaches comparison: (*a*) results of the deterministic approach [4] and (*b*) results of the stochastic technique.

stochastic approach has evidenced not only a faster execution time in all conditions, but also a better precision and robustness even with differently positioned vehicles or obstacles. On the other side, ant's paths are less smooth than the ones deterministically computed.

Chapter 3

Stereo-based Preprocessing for Human Shape Localization in Unstructured Environments

Contenuto capitolo

This chapter describes the research activities for the localization of human shapes using visual information in the frame of a common project with the TACOM Department of U. S. Army. The chapter proposes the application of a stereoscopic technique as a preprocessing for the localization of humans in generic unstructured environments. Each row of the left image is matched with the epipolar row of the right image. This creates a map of each object in the scene as well as the slope of the road. Preliminary results have proved to be promising.

3.1 Introduction

Auonomous navigation will be a vital part in the near future for the U. S. Army. The Vetronics Technology Area, a division within the Army's TACOM Research, Development, and Engineering Center (TARDEC), and the University of Parma are working towards detect human shapes from a moving vehicle. Programs within Vetronics have a need for autonomous navigation, the Robotic Follow program [10], and semi-autonomous navigation, the Crew integration and Automation Test bed program [9]. Human shapes detection is a vital piece to make these programs within the Vetronics Technology Area succeed.

The Robotic Follower is a robotic vehicle that is used to follow behind a person or another vehicle to carry supplies to and from areas [10]. The path the robotic follower takes is based on electronic breadcrumbs that are left behind by the lead person or vehicle. The greater the distance between the leader and the follower, the more chance of people interrupting the breadcrumb trail. The Robotic Follower needs the ability to detect people in and around it so that it can take the necessary precautions to avoid them.

The Crew integration and Automation Test bed program is designed to incorporate several driver aided packages [9], one of which is detecting humans. As the driver traverses through an area, the system will detect people, and highlight them, so that the driver is more aware of their presence.

However, the need for automatic human detection goes beyond the Vetronics Technology Area. Within the Army, and throughout the commercial community,

the need for detecting people is great. Driver awareness systems, security systems, traffic/pedestrian control systems, and automatic switching systems are just a few areas that would benefit greatly from this technology.

There are many approaches to pedestrian detection. Some use learning machines like neural networks [47] or support vector machines [32], some use motion to detect pedestrians [40, 43]. Throughout this study, motion was not used. The authors felt that approaching the problem of solving the cases for individual frames will prove to be more beneficial than using continuous frames. Tracking can be added later to reduce the number of false positives. The specific application of the method shown in this chapter is unique to the stereo vision community. Each row of the left image is matched with the corresponding row of the right image. This creates a map of each object in the scene as well as the slope of the road. Both information can be used in the human shape localization algorithm presented in [5]. Preliminary results have proved to be promising.

This chapter is organized as follows: section 3.2 introduces the vision-based system for detecting pedestrians in road environments developed in the last years by the University of Parma in collaboration with TACOM. Section 3.3 presents a stereo based technique for the extraction of features of interest. The results of this approach are shown in section 3.4, while its application and advantages are discussed in section 3.5.

3.2 A stereo-based approach in structured environments

In the last years the University of Parma and the TACOM Department of U. S. Army developed a vision-based system for detecting pedestrians in road environments [5,6]. The system is aimed at the localization of pedestrians in various poses, positions and clothing, and is not limited to moving people.

Attentive vision techniques relying on the search for specific characteristics of pedestrians, such as vertical symmetry and strong presence of edges, are used to select interesting regions likely to contain pedestrians. More precisely, the acquired image is scanned and symmetries and edges are extracted; since a human shape is characterized by a strong vertical symmetry, symmetrical areas with a specific aspect ratio identify possible candidates. Thanks to some a-priori knowledge on the environment (the slope is known since the road is assumed flat), size and perspective constraints are also adopted to ease and speed up the search.

Specific filters are then used to remove evident detection errors and false positives.

Subsequently, the remaining candidate areas are validated verifying the actual presence of pedestrians by means of shape-based techniques. A method based on the application of autonomous agents has been investigated [5], and other approaches are under study.

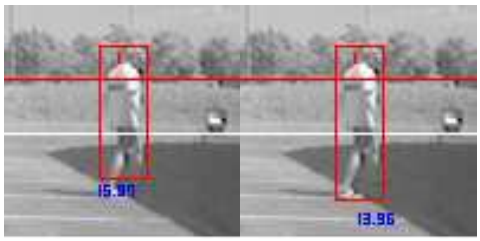
This system, completely based on monocular techniques, has subsequently been enhanced thanks to a stereo-based refinement. In fact, some errors may arise in the first phase due to an incorrect localization of candidates. In other words, a human

body may present a sufficiently high symmetry to be detected, but the detected area may not be precise. This generally happens to the legs, which can be in different positions. In these cases, a bounding box enclosing the human body is drawn around the detected shape, but it may cut out a part of the body –generally the legs.–

An incorrect localization of the bounding box may be critical for the following shape detection process aimed at its validation. Moreover, this error affects distance estimation in monocular images. The stereo refinement is targeted to fix this problem in the assumption of a flat road.

First, the left image is searched for symmetries, bounding boxes corresponding to candidates are generated, and a set of filters are applied to remove obvious errors in the detection. Then, for each surviving bounding box the right image is searched for areas which exhibit a content similar to the one included in the bounding box (a correlation measure is performed). Once the correspondence between the bounding box located in the left image and its counterpart in the right image has been found, stereoscopy can be used to determine the distance to the vision system. This step requires the correct calibration of cameras parameters and orientations.

Once the correct distance estimation for each bounding box has been provided to the system by means of stereoscopy, a refinement of the bounding box base can take place, based on calibration and perspective constraints. More precisely, the knowledge of the camera orientation with respect to the ground and the road slope can provide information about the position of the point of contact of the human shape with the ground. This knowledge is used to stretch the bottom of the bounding box till it reaches the ground and frames the entire shape of the pedestrian, thus easing



(a) (b)

Figure 3.1: Distance refinement: (*a*) result before refinement: a potential pedestrian is detected but the bounding box cuts the legs thus affecting the distance estimation; (*b*) result after stereo refinement: the bounding box has been stretched till the ground; the distance estimation is now correct.

the following shape-based validation. Figure 3.1 shows the result of this stereo refinement.

3.3 A stereo technique for feature extraction

The current research addresses the problem of human shape localization in generic environments, including urban, country, and desolate.

The previously discussed approach could be easily generalized to any scenarios (including non flat ones) removing the assumption on the knowledge of the road slope. In this case, however, size and perspective constraints are to be dropped, and an exhaustive search has to be performed in the candidates generation phase. This entails both a higher computational complexity and a more complex selection of the interesting areas since a high number of candidates must be considered and compared.

Moreover, the stereo refinement of the bounding boxes, as defined previously, is

not possible if the scene slope is unknown, and possible errors in the bounding box localization are to be tackled in the following shape-based validation step.

Following these considerations, a new stereoscopic approach has been developed to deal with generic environments where the scene slope is unknown, featuring low computational complexity. This method is based on a row-wise comparison of the two stereo images, assuming the two optical axes lie on the same plane and both cameras have a null roll angle.

This approach has been tested on both synthetic and real images, see figure 3.3. The left and right images (figures 3.3.a and 3.3.b) are processed with the following steps: an edge extraction, followed by a binarization and a morphological horizontal expansion are performed. The results are pixel-ORed with the original images. In this way, the original grey-level values are only preserved in correspondence to areas with a relevant information content (i.e. edge points). Figures 3.3.c show the result of the processing of figures 3.3.a (left images); the same processing is applied to the right images. The resulting images will be referred to in the following as *feature images*.

For each line the correlation between the left and right epipolar lines of the feature images is computed for different offsets. Figures 3.3.d show the value of the correlation of each image line for different offsets: these *correlation images* display the offset on the horizontal axis and the image line number on the vertical axis, encoding high correlation values with bright pixels.

Perspective considerations allow to discriminate different components (modeled in figure 3.2) in this correlation image:

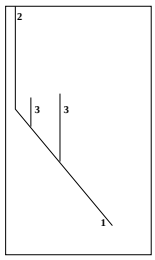


Figure 3.2: Different components of correlation: (*1*) ground slope, (*2*) background, and (*3*) obstacles.

1. a slanted line encodes the ground slope (in this cases, the road),

2. a vertical line on the top left of the image encodes the background (above the horizon), and

3. other vertical segments originating upward from the slanted line represent potential obstacles (in this case, pedestrians).

These components are hardly discernible as shown in figures 3.3.d, because they mask each other; moreover, noise affects the correlation measure.

The following procedure is aimed at extracting them one by one.

In fact, the strongest component of the correlation encodes the longitudinal slope of the scene, provided that the transversal slope of the scene is neglectable. For example, in case of a flat scene without obstacles the offset yielding the maximum correlation decreases with the distance from the vision system according to a known function (component 1), and becomes constant in correspondence to the horizon and upper (component 2) [34]. This behavior is due to the fact that the difference of displacement in left and right images for 3D points lying close to the camera is larger

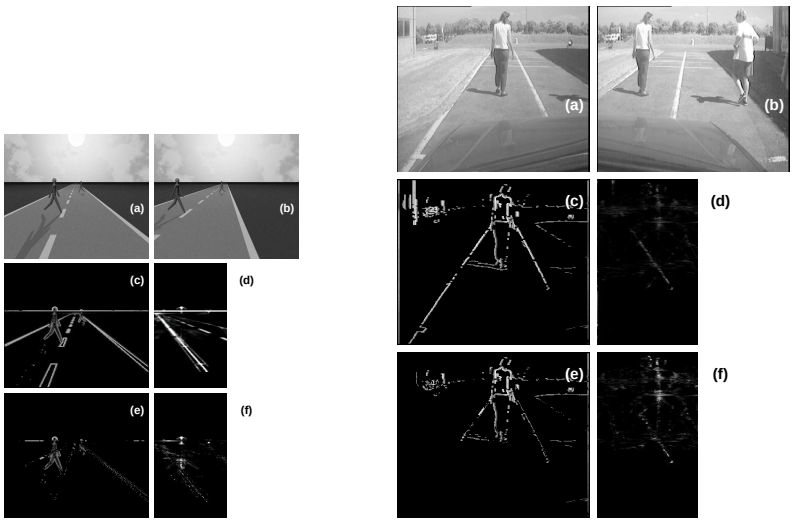


Figure 3.3: Obstacle detection in a synthetic and a real situation: (*a*) left and (*b*) right images, (*c*) relevant features computed for the left images, (*d*) line-wise correlation values between left and right features images for different offsets, (*e*) left features image after the removal of background, (*f*) line-wise correlation values computed after the removal of background.

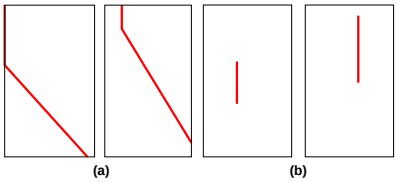


Figure 3.4: Reconstruction of correlation components extracted from figures 3.3.d: (*a*) road and background components, (*b*) components given by the closest pedestrians.

than for 3D points lying far away from the vision system. On the other hand, 3D points at infinite distance are imaged in the same position in the left and right images when the two optical axes are parallel, or at a constant offset in case the axes are convergent or divergent.

Components 1 and 2 are evident when the ground surface and background present an appreciable texture When present, they prevail and partially mask the other obsta-

cles' components. Each obstacle contributes to a vertical segment in the correlation image correspondent to a constant offset. However, components 1 and 2 are stronger than the obstacle's one. This effect can be exploited to identify and remove the ground and background features leaving only the features belonging to the obstacles. The correlation image is thus analyzed to detect its components. More specifically, the slanted line corresponding to the road slope and the vertical line representing the background offset are first extracted by using a Hough transform on the correlation image. Figures 3.4.a show a reconstruction of components 1 and 2 extracted from figures 3.3.d in the synthetic and real case.

The set of offsets encoded in this polyline are then applied to match the epipolar lines of the left and right feature images. This comparison is used to remove matching features (i. e. ground texture and background objects), thus leaving non-matching areas (i. e. foreground obstacles). Figures 3.3.e show the left features images after the removal of ground and background features; it can be noticed that obstacles are more evident since many disturbing features have been filtered out.

The computation of the correlation image is then repeated, starting from the feature images with ground and background removed (see figures 3.3.f). Now, the obstacle components prevail in the correlation image and can be extracted thanks to a vertical histogram. Figures 3.4.b show a synthetic reconstruction of the correlation values for the closest obstacles.

Once its offset is known, the area of each obstacle can be further analyzed to derive the cluster of features belonging to it. The right feature image is shifted with the offset corresponding to the obstacle and compared to the left, and their matching

features are examined. A vertical histogram allows the identification of the position of the lateral borders of the obstacle. A horizontal histogram computed in the vertical stripe the object belongs to gives hints on the bottom and top limits of the object.

In presence of multiple objects lying at different distances from the vision system, the localization of individual objects is simplified if the previously segmented objects are in turn eliminated from the feature image and the correlation function is every time recomputed. In this manner, the strong contribution to the correlation given by an evident object does not mask weaker contributions given by other objects, and objects can be extracted in subsequent iterations of the processing. At each stage the features belonging to a different obstacle can be clustered and labeled. Figures 3.5.c-g show how the features belonging to the ground and background and three different obstacles are identified and removed at subsequent steps of the processing. In figure 3.5.h the detected clusters of features have been labeled with different colors.

3.4 Results

Figures 3.6 and 3.7 show examples of extraction of obstacles' features from stereo images in unstructured environments. The original left image is displayed together with a copy with obstacles' edges highlighted with different colors.

Figure 3.6 presents three examples of correct detection, detailed in the following. Figure 3.6.a shows a parking lot with four pedestrians and vehicles on the sides: edges of pedestrians are correctly localized. Figure 3.6.b displays descending ramp: four objects are localized (three pedestrians and a short wall on the right), the wall

on the left is also detected despite its weak texture. Figure 3.6.c side view of a steep descending ramp with a group of children: humans are localized. Note that the high transversal slope causes an incorrect detection of the top left group of children (shadows are misinterpreted as belonging to obstacles).

Figure 3.7 shows some problems of the current version of the algorithm. In particular, figure 3.7.a presents an off-road country environment: the very weak texture of the ground does not allow the determination of its slope and the resulting cluster of pixels include also features of the ground under the obstacle. Figure 3.7.b shows a descending ramp with trees' shadows: one pedestrian is not detected due to low contrast and the problems in the detection of the ground slope generate a too large cluster for the pedestrian on the right. Figure 3.7.c refers to uphill driving: one pedestrian is not detected due to low contrast and the long wall on the right is correctly detected as an obstacle but sliced in three parts due to the large range of distances (offsets) covered.

3.5 Discussion

The stereo technique discussed in section 3.3 provides clusters of features that can be fed into the original monocular processing described in section 3.2 aimed at distinguishing human shapes from other obstacles. The inclusion of this preprocessing allows to limit the computation of symmetries to the detected area of interest only, improving computational time.

Moreover, the preprocessing ability to determine the ground slope permits both the application of perspective considerations and the correct detection of the objects'

point of contact with the ground.

This approach has the advantage to adapt the original method to generic scenarios. Furthermore, since the scene slope can be obtained from the stereo row-wise correlation, different approaches could be developed for flat and non-flat scenarios.

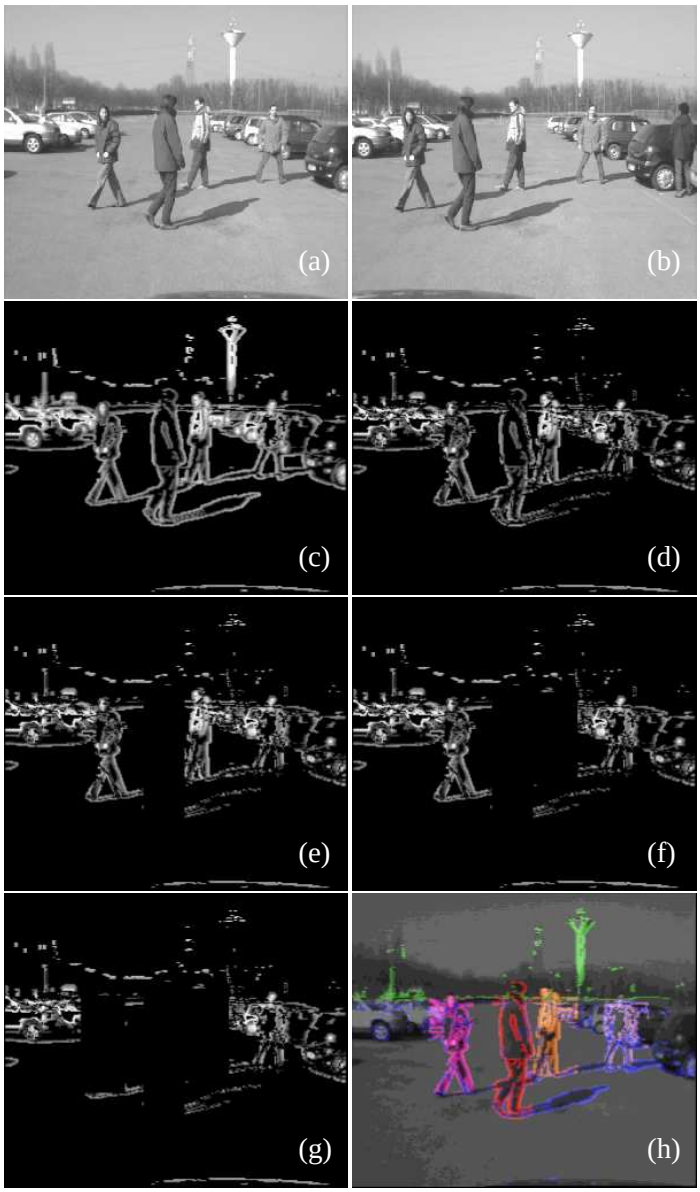
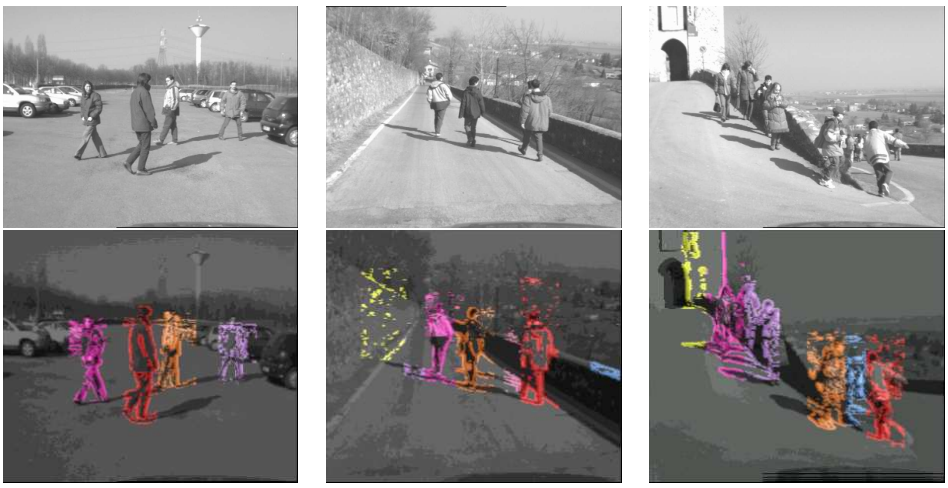


Figure 3.5: (*a*) and (*b*) Left and right images, (*c*) relevant features computed for the left images, (*d*) removal of ground texture and background, (*e*) removal of first object, (*f*) removal of second object, (*g*) removal of third object, (*h*) the clusters of features labeled with different colors; the ground features are shown in blue, while background objects are highlighted in green.



(a) (a) (a)

Figure 3.6: Results in different situations.

(a) (a) (a)

Figure 3.7: Situations in which the features extraction experience problems.

Chapter 4

Shape-based pedestrian detection and localization

Contenuto capitolo

This chapter presents a vision-based system for detecting and localizing pedestrians in road environments by means of a statistical technique.
Initially, attentive vision techniques relying on the search for specific characteristics of pedestrians such as vertical symmetry and strong presence of edges, allow to select interesting regions likely to contain pedestrians. These regions are then used to estimate the localization of pedestrians using a Kalman filter estimator.

4.1 Introduction

The pedestrians detection is an essential functionality for intelligent vehicles, since avoiding crashes with pedestrians is a requisite for aiding the driver in urban environments.

Vision-based pedestrian detection in outdoor scenes is a challenging task even in the case of a stationary camera. In fact, pedestrians usually wear different clothes with various colors that, sometimes, are barely distinguishable from the background (this is particularly true when processing grey-level images). Moreover, pedestrians can wear or carry items like hats, bags, umbrellas, and many others, which give a broad variability to their shape.

When the vision system is installed on-board of a moving vehicle additional problems must be faced, since the observer's ego-motion entails additional motion in the background and changes in the illumination conditions. In addition, since Pedestrian Detection is more likely to be of use in a urban environment, also the presence of a complex background (including buildings, moving or parked cars, cycles, road signs, signals...) must be taken into account.

Widely used approaches for addressing vision-based Pedestrian Detection are: the search of specific patterns or textures [17], stereo vision [24,32,47], shape detection [26–28], motion detection [18,35,36], neural networks [45,46]. The great part of the research groups use a combination of two or more of these approaches [17,40,47]. Anyway, only a few of these systems have already proved their efficacy in applications for intelligent vehicles.

This chapter presents the first results of a new localization and association rule specifically designed to follow the detection process previously developed [5].

In this work the strong vertical symmetry of the human shape is exploited to determine specific regions of interest which are likely to contain pedestrians. This method allows the identification of pedestrians in various poses, positions and clothing, and is not limited to moving people. In order to improve the reliability of the system and as preliminary work for pedestrian tracking, a pedestrian localization step has been added. Pedestrian localization iteratively computes the position of pedestrians in the 3D world. It has been conceived to be used for a tracking system.

This chapter is organized as follows. Section 2 introduces the structure of the algorithm. Section 3 describes the detection module, section 4 presents the localization procedure. Section 5 ends with some final remarks.

4.2 Algorithm structure

Figure 5.1 shows the algorithm structure. As a first processing step, attentive vision techniques are applied to concentrate the analysis on specific regions of interest only. In fact, the aim of the low-level part of the processing is the focusing on potential candidate areas to be further examined at a higher-level stage in a following phase.

The areas considered as candidate are rectangular bounding boxes which:

- have a size in pixels falling in a specific range. This range is computed from the knowledge of the intrinsic parameters of the vision system (angular aperture and resolution) and from allowed size and distance of pedestrians;

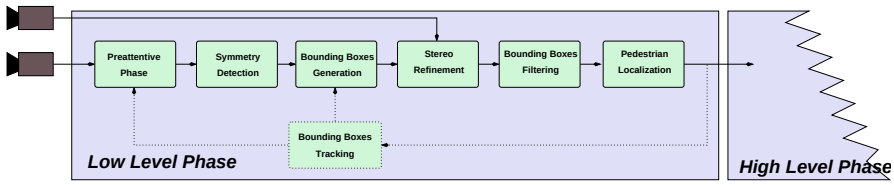


Figure 4.1: The algorithm architecture.

- enclose a portion of the image which exhibits the low-level features that characterize the presence of a pedestrian, i. e. a strong vertical symmetry and a high density of vertical edges.

A stereo refinement is used to refine the computed bounding boxes. The other image is searched for the same detected object and a triangulation is used to determine the distance.

Moreover, since other objects than pedestrians feature high symmetrical content, a set of filters is used to remove objects like poles, trees. . . .

The forward loop process ends by estimating the pedestrian position in the road scene. This stage uses an internal model (the *scene bounding box*) that allows to take into account the possible bad fitting of the detected bounding box with respect to the real pedestrian shape.

Beside the obvious usefulness of a pedestrian localization functionality for a driver assistance system (for example, in order to know which pedestrian is the more dangerous and to focus the perception on him); in addition, as shown on figure 5.1 (dotted lines) localization can also be use to foresee the future position of the bounding boxes (i.e. to track the pedestrians). The tracking can be used to directly act onto

the detection stages in order to improve the reliability of the system. Currently, the loop has not been yet closed. The full tracking system is under development.

4.3 Pedestrian detection

4.3.1 Search area

In the first phase a search for pedestrians candidates is performed. Thanks to the knowledge of the system's extrinsic parameters together with a flat scene assumption, this search is limited to a reduced portion of the image (see figure 4.2). The displacement of this area depends on the pedestrian's distance, while its height is computed as a function of the pedestrian's maximum height. Besides the obvious advantage of avoiding false detections in wrong areas, the processing of a reduced search area only, reduces the computational time. The analysis is not limited to a target featuring a fixed size or a given distance, but a range for each parameter is in fact considered. The introduction of these ranges generates two further degrees of freedom in the size and position of the bounding boxes. In other words, the search area is enlarged to accommodate all possible combinations of height, width, and distance for pedestrians.

4.3.2 Symmetry detection

The analysis proceeds in this way: the columns of the image are considered as possible symmetry axes for bounding boxes. For each symmetry axis different bounding boxes are evaluated scanning a specific range of distances from the camera (the distance determines the position of the bounding box base) and a reasonable range of

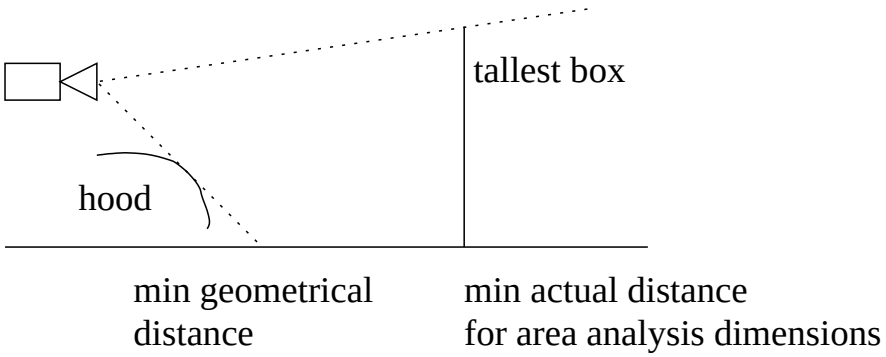


Figure 4.2: Undersampling constraints.

heights and widths for a pedestrian (the corresponding bounding box size can be computed through the calibration).

However, not all the possible symmetry axes are considered: since edges are chosen as discriminant in most of the following analysis, a pre-attentive filter is applied, aimed at the selection of the areas with a high density of edges. Axes centered on regions which contain a number of edges lower than the average value are dropped.

For each of the remaining axes the best candidate area is selected among the bounding boxes which share that symmetry axis, while having different position (base) and size (height and width). Vertical symmetry has been chosen as a main distinctive feature for pedestrians. Symmetry edge maps, e. g. the Generalized Symmetry Transform (GST) [42], have already been proposed as methods to locate interest points in the image prior to any segmentation or extraction of context-dependent information. Unfortunately, these methods are generally computationally expensive. Alternatively, two different symmetry measures are performed: one on the gray-level values and one on the horizontal gradient values. The selection of the best bounding

box is based on maximizing a linear combination of the two symmetry measures, masked by the density of edges in the box.

4.3.3 Bounding boxes generation

An adjustment of the bounding boxes' size is yet needed. In fact, when comparing the gray-level symmetry of different bounding boxes centered on the same axis, larger boxes tend to overcome smaller ones since pedestrians are generally surrounded by homogeneous areas such as concrete underneath or the sky above. Therefore, the bounding box which presents the maximum symmetry tends to be larger than the object it contains because it includes uniform regions. For this reason, for each selected symmetry axis, the exact height and width of the best bounding box are actually taken as those possessed by the box which maximizes a new function among the ones having the same axis. This function is computed as the product of the symmetry of vertical edges and density of vertical edges only. Figure 4.3 summarizes the overall candidate generation process.

The result of this step is a first list of candidate bounding boxes that contains potential pedestrians.

4.3.4 Stereo refinement

The distance of the potential pedestrians can be computed using the knowledge of the camera calibration and the assumption of a flat scene. Unfortunately, the computed values are greatly affected by a wrong detection of the lower part of pedestrians. In order to refine this measurement, which is of importance for discriminating amongst

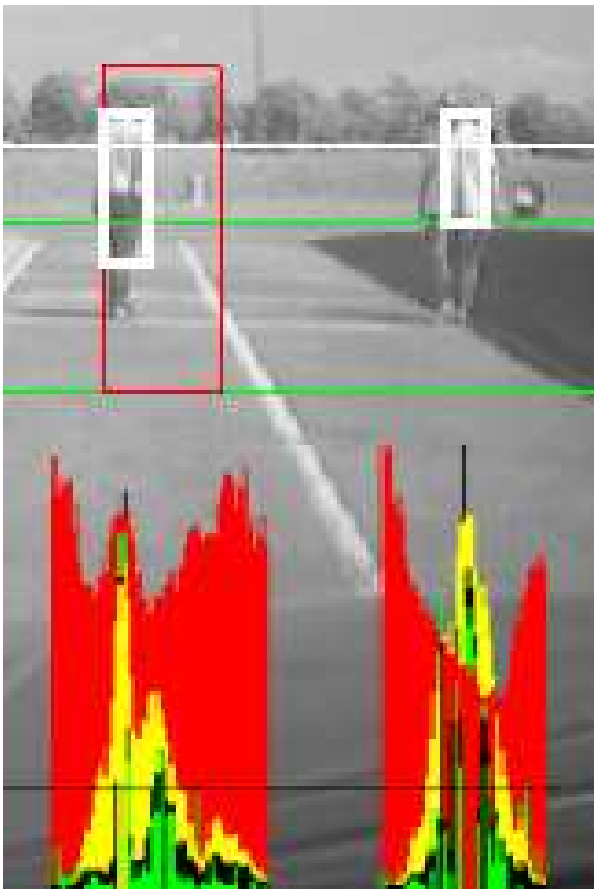


Figure 4.3: Bounding boxes generation phase: the two horizontal green lines represent the search area; the symmetry histograms for grey-levels (red), vertical edges (green), the vertical edges density (yellow), and their combination (black) are shown in the bottom part of the image.

obstacles and actual pedestrians, a refinement phase is mandatory.

A simple stereo technique is used: for each bounding box in this list, starting from a rough estimation of the distance, a portion of the other image is searched for areas which exhibit a content similar to the one included in the bounding box by means of

a correlation measure. The correlation formula used for matching left a_i and right b_i pixels is:

$$\chi = \frac{(\sum a_i b_i)^2}{(\sum {a_i}^2)(\sum {b_i}^2)}$$

Once the correspondence between the bounding box located in the left image and its counterpart in the right image has been found, a triangulation is be used to determine the distance to the vision system. Therefore, a refinement of the bounding box base can take place, based on calibration and perspective constraints. More precisely, the knowledge of the camera orientation with respect to the ground and the road slope can provide information about the position of the point of contact of the human shape with the ground. This knowledge is used to stretch the bottom of the bounding box till it reaches the ground and frames the entire shape of the pedestrian and the technique is robust in the sense that even if the background is different from one image to another, the distance is correctly evaluated and the base exactly refined for all observed cases (see figure 4.4).

4.3.5 Bounding boxes filtering

Symmetrical objects other than pedestrians may happen to be detected as well. In order to get rid of such false positives a number of filters have been devised which rely on the analysis of the distribution of edges within the bounding box and on segmentation and classification of the box region. These filters, which are still under development, show promising results regarding the elimination of both artifacts (such as poles, road signs, buildings, and other road infrastructures) and symmetrical areas

Figure 4.4: Stereo refinement: the yellow bounding box is generated during the symmetry detection, the red line represents the stereo refinement of the box's bottoms.

given by a uniform portion of the background between two foreground objects with similar lateral borders.

4.4 Discussion

A new localization technique has been presented. This technique exploits scene localization of pedestrians by means of iterative image coordinates modeling. The reprojection onto the source images shows a correct spatial positioning. The localization is not affected by target's movements.

The system has been tested in different situations. Currently, the result is not exploited by the preattentive phase, but results obtained by the localization phase

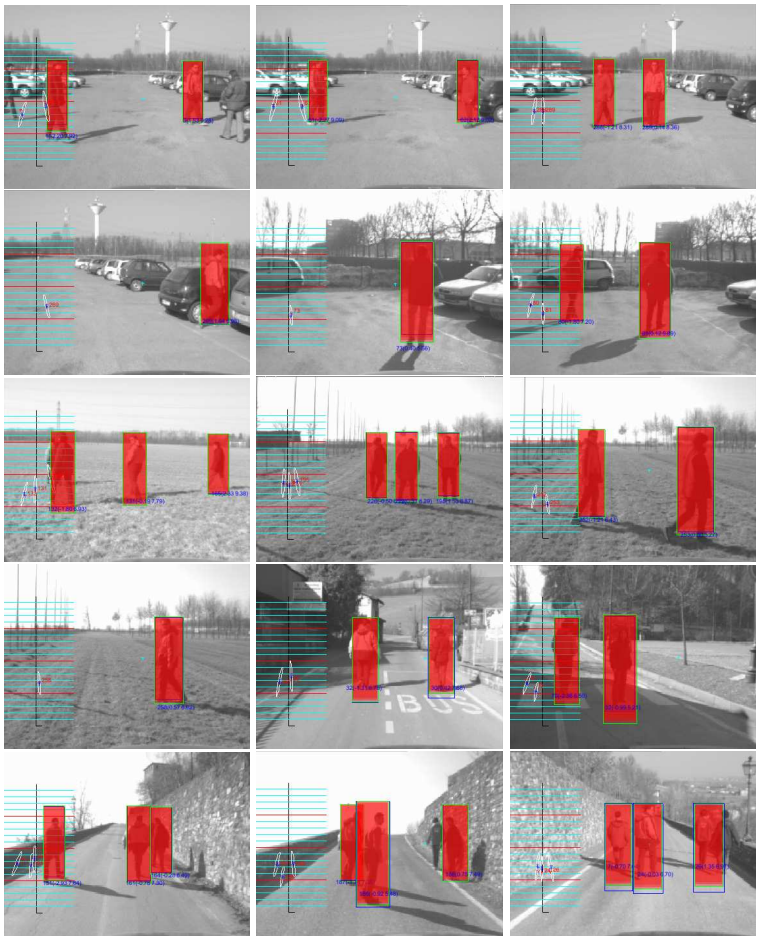


Figure 4.5: Localization results: the localization area is superimposed on original images in red, the three numbers below these areas represent the localization ID and its coordinates in the world (meters). On the left side of the image a top view of the scene is sketched; each horizontal line represents 1 meter. For each pedestrian an error ellipsoid is given with its ID as well.

promise to improve the reliability and efficiency of the preattentive stage.

A full tracking system that exploits the pedestrians localization function is described.

Chapter 5

Pedestrian Localization and Tracking with Kalman Filtering

Contenuto capitolo

This chapter presents an implementation of a vision-based system for recognizing pedestrians in different environments and precisely localizing them with the use of a Kalman filter estimator configured as a tracker. Pedestrians, in various poses and with different kinds of clothing, are first recognized by the vision subsystem through the use of algorithms based on edge density and symmetry maps. The information produced in this way is then passed on to the tracker module which reconstructs an interpretation of the pedestrians positions in the scene. An appropriately configured indoor system setup with an accurate measurement of the imposed human trajectory has been realized. This setup has permitted an accurate evaluation of the accuracy of the results, when the new auxiliary tracker is activated.

5.1 Introduction

Widely used approaches for addressing vision-based pedestrian detection are: the search of specific patterns or textures [17], shape detection [13, 15, 22, 25, 27, 39] and neural nets-based methods [38].

This chapter presents the system introduced in [2] that is aimed at the localization of pedestrians by means of vision. This system has been designed to be installed on board of moving vehicles in order to provide the driver with warning signals. In particular, the implementation of a new tracking layer based on Kalman filtering [31, 33] for this system is examined and the article mainly deals with performance measurements of the system activity.

This chapter is composed of the following sections:

section 5.2 presents the system scheme,

section 5.3 introduces the new tracking functionality of the system,

section 5.4 summarizes the most valuable numerical results obtained,

section 5.5 discusses the results and outlines the possibilities for future improvements of the system.

5.2 System structure

In this section the components of the pedestrian localization system are briefly explained. Fig. 5.1 depicts the relationships between the system components that perform the following tasks:

- “Preattentive Phase” - low level vision elaboration,

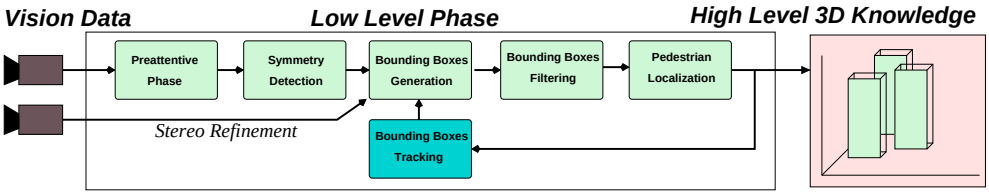


Figure 5.1: The system architecture.

- "Symmetry Detection" - symmetry maps evaluation,
- "Bounding Boxes Generation" - pedestrians outlining,
- "Bounding Boxes Filtering" - pedestrian boxes selection,
- "Pedestrian Localization" - spatial position estimation for pedestrian boxes,
- "Bounding Boxes Tracking" - state variables and associated accuracy evaluation.

5.2.1 Preattentive phase

The knowledge of the vision system's extrinsic parameters and the flat scene assumption allows to reduce the search for candidates to one limited part of the image, and reasonable ranges and steps are considered for dealing with different pedestrian dimensions.

Fig. 5.2 (*a*) presents the image clustering and edge extraction performed at this stage of the processing. Besides the obvious advantage of avoiding false detections in wrong areas, this technique, combined with an undersampling procedure, strongly

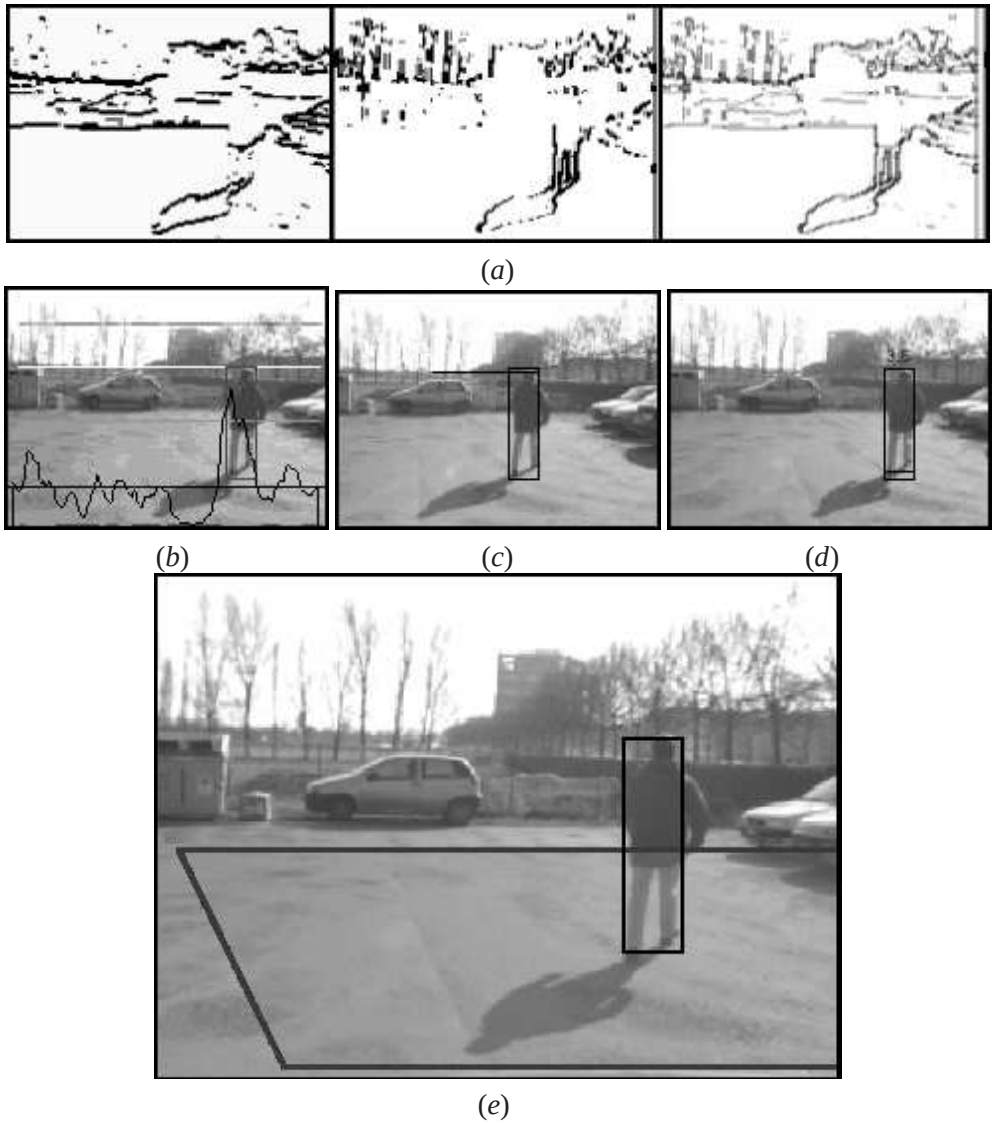


Figure 5.2: The vision algorithm processing stages for an example outdoor image acquired from a moving vehicle: (*a*) low level horizontal, vertical and combined edges; (*b*) preattentive filtering; (*c*) search range for the homologous box; (*d*) stereo refinement for the base of the box; (*e*) result (the stereo search area is surrounded with a border).

reduces the computational time needed for frame elaboration and shows excellent temporal results.

5.2.2 Symmetry detection

After the low level preprocessing and the analysis of vertical symmetry maps derived from gray-level and horizontal gradient image values, the identification of regions that can be characterized as human shapes takes place. Since pedestrians evidence an high symmetry, especially vertical, image columns can be considered as possible symmetry axes and edges can be used as a discriminant in a pre-attentive *filtering* stage (see fig. 5.2 (*b*)).

Approaches based onto this kind of maps have already been illustrated with the Generalized Symmetry Transform (GST) [42].

5.2.3 Bounding boxes generation

The axis-based approach is followed by maps analysis for the extraction of the boxes and after this a particular *stereo refinement* technique is used to improve the accuracy of the identification of the boxes' bases (see fig. 5.2 (*c*,*d*)). Fig. 5.2 (*e*) presents the box generation result for an image acquired from a moving vehicle.

This processing level produces boxes with an high probability to fit one pedestrian, and candidates are characterized by problem specific dimensions in pixels and symmetry axes placed nearby the peaks of relative maximums in the axial weighted symmetry sum.

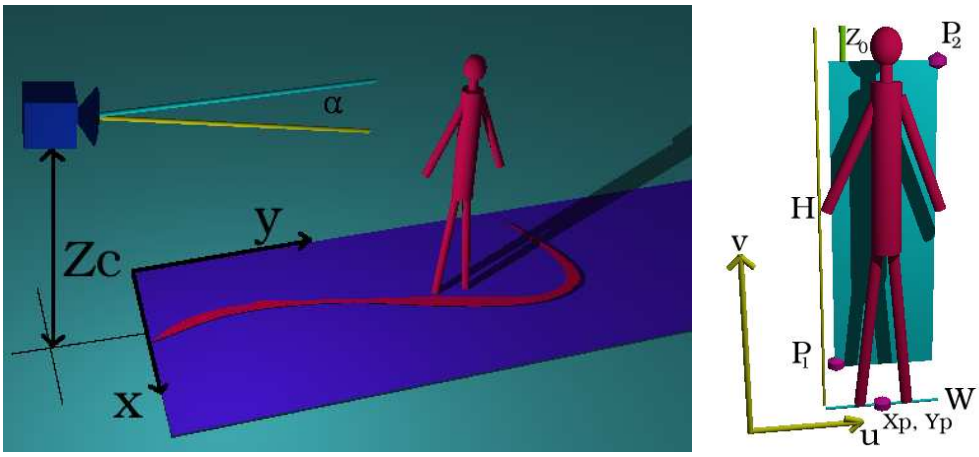
Figure 5.3: Bounding boxes filtering: the discarded pedestrian candidates are marked with a black “x”, each example shows the original and the edges inside the candidate bounding box.

5.2.4 Bounding box filtering

Unfortunately symmetrical objects other than pedestrians may happen to be detected as well. In order to get rid of such false positives a number of filters based on regionalization have been devised and are still under development. Fig. 5.3 illustrates some examples of how the filters check the eligible candidates and eliminate some of them that do not actually represent a human shape. These filters evidence promising results with artifacts such poles, road infrastructures, traffic signs and buildings that cause the box generation to fail.

5.2.5 Pedestrian localization

This module estimates the position of the pedestrians in the scene in the chosen coordinate reference system. The contact point (X_p, Y_p) of each pedestrian vertical axis with the ground assumed flat is associated with opportune state variables for this purpose. The height from the ground Z_c, the tilt angle α of the camera observing the



(*a*) (*b*)

Figure 5.4: Setup scheme and problem variables: (*a*) world coordinate reference system; (*b*) image coordinates of the scene bounding box.

scene and a set of intrinsic calibration parameters represented by e_u and e_v must be known. Fig. 5.4 (*a*) shows the coordinate system in which the contact point is defined according to the road plane and also the position of the camera.

The estimation uses an original modeling that takes explicitely into account the unavoidable difference of the vision-detected bounding box of a pedestrian with the real ideal one, defined by a height H and by a width W with fixed average and standard deviation values, realistic enough to represent a human shape ($H = 1.65\ m$, $\sigma_H = 0.1\ m$, $W = kH$ with $k = 0.3$ a realistic width/height ratio, $\sigma_W = 0.1\ m$). Z_0 represents half of the difference between the scene bounding box height and the real pedestrian height, with average value zero and standard deviation $\sigma_{Z_0} = 0.1\ m$.

Considering a perspective projection of the scene onto the image, the relationship between the coordinates of the corners $P_1 = (X_1, Y_1, Z_1)^t$ and $P_2 = (X_2, Y_2, Z_2)^t$ of a

pedestrian bounding box in the camera coordinate system can be linked in a linear way, thanks to small angle approximation for α, to the planar position coordinates X_p and Y_p (5.1). The correspondent image coordinates $p_1 = (u_1, v_1),\ p_2 = (u_2, v_2)$ and the observation system $\mathbf{Y} = H \cdot \mathbf{X} + \mathbf{v}$ (5.2) are then easily deduced.

$$\begin{cases} X_1 = X_P - \frac{W}{2} \\ Y_1 = Y_P - \alpha(Z_0 - Z_c) \\ Z_1 = \alpha Y_P + (Z_0 - Z_c) \end{cases} \text{ and } \begin{cases} X_2 = X_P + \frac{W}{2} \\ Y_2 = Y_P - \alpha(H - Z_0 - Z_c) \\ Z_2 = \alpha Y_P + (H - Z_0 - Z_c) \end{cases} \tag{5.1}$$

$$\begin{pmatrix} -e_v(Z_0 - Z_c) \\ e_u \frac{W}{2} \\ -e_v(H - Z_0 - Z_c) \\ -e_u \frac{W}{2} \end{pmatrix} = \begin{pmatrix} 0 & -v_1 \\ e_u & -u_1 \\ 0 & -v_2 \\ e_u & -u_2 \end{pmatrix} \begin{pmatrix} X_P \\ Y_P \end{pmatrix} + \underline{v} \tag{5.2}$$

The contribution of all the parameters subject to error is also taken into account with the use of the covariance matrix of the noise vector $\underline{v}$, in order to improve the estimation of the positions of the pedestrians, concretely realized with a Kalman filter.

More details on how this modeling deals with the pedestrian spatial positioning are available in [2]. A new bounding box tracking stage now completes the approach.

5.3 Bounding Boxes Tracking

In this section the implementation of the tracker is explained in terms of design choices, box management queues, tracking politics and Kalman filtering integration.

Each new pedestrian identified by the localization is provided with an unique `id`. This is used to drop the box if the timeout for joining with an appropriately new

detected pedestrian expires, to log the history of the pedestrian path, to differentiate it from the others and also to render clearly all the graphical information (see fig. 5.6).

The tracker presents a flexible politic for data logging, box processing, matrixes allocation and an efficient method encapsulation for complex procedural sections; the subsystem presents two possible working modalities: *single tracking* and *multi tracking* mode; these discriminates the way in which the rejoining of lost traces is managed.

The main tasks of the tracker are: the *merge* of visual localizations with the internal state representation, the calculations relating to the evolution of the state of each pedestrian (through Kalman filtering), and the prediction projection for the triggering of the elaboration. Input and output buffering queues are used for filtering purposes too, in order to implement insertion and removal politics that enhance the reliability of noisy sensor data.

Graphics are used to illustrate the state variables history of the pedestrian boxes. This is done for sake of an efficient and constant system check by the human supervisor, both in the perspective image (fig. 5.6 (*a*)) and in the road *top view* plane (fig. 5.6 (*b*)). The current box position and the position prediction, in the form of probability-blended image projection areas, are drawn in the perspective representation. Moreover the error ellipsis for each box is represented on the experimental road plane image.

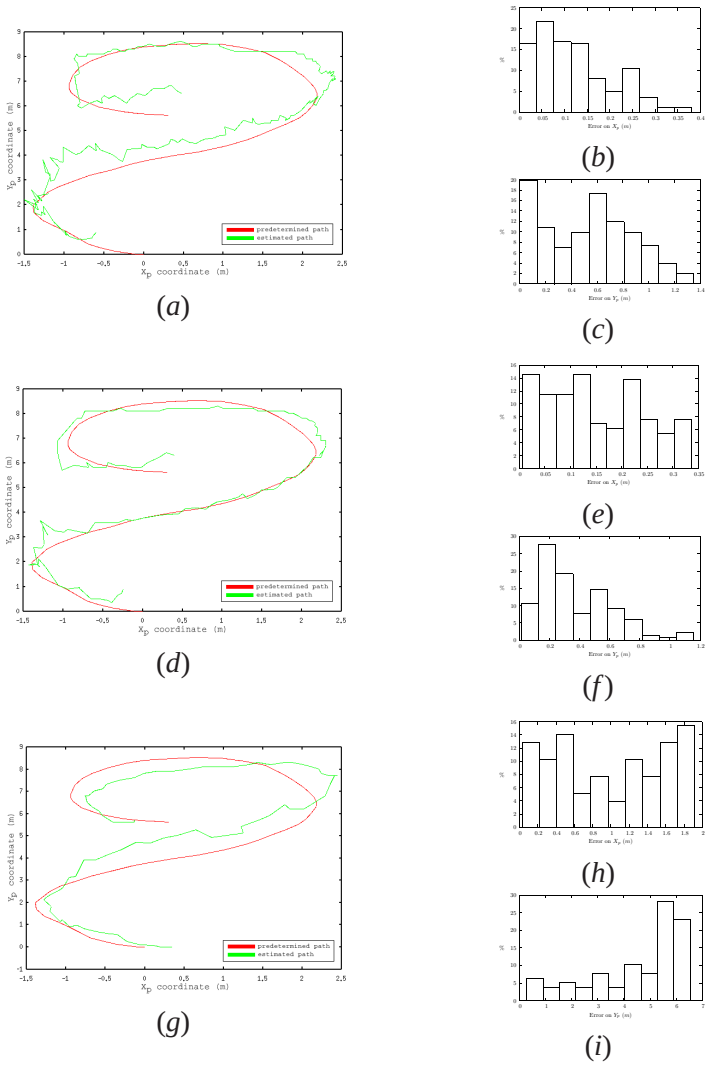
The *merge function* performs one feedback task related to the association of newly detected pedestrians with the set of spatially localized and tracked ones. This approach solves problems related to wrong estimations and temporal mismatches. It

is based on box areal overlapping and Mahalanobis distance estimation, and is responsible for updating the tracked set of pedestrians. The overlapping criterion is based on probability image areas after Kalman prediction and the metric criterion exploits the state of each tracked pedestrian. Instead of a Mahalanobis distance classification *tout-court*, the product $H \cdot \mathbf{X}$ is used as observations for the evaluation of the metric r.

The most effective formula for the extraction of r at the iteration κ for the vision observation n and the consequent classification has been found to be the match criterion in (5.3),

$$\begin{aligned} & r_n^{(k,)}(i) = \Delta^\top \cdot {C^{(k-1,i)}}^{-1} \cdot \Delta \\ & \Delta \triangleq [H_n^{(k,)} \cdot \mathbf{X}_\mathbf{n}^{(\mathbf{k},)} - H^{(k-1,i)} \cdot \mathbf{X}^{(\mathbf{k-1},\mathbf{i})}] \\ & \texttt{match} \coloneqq i \mid \min\{r_n^{(k,)}(i)\} \leq t^* \end{aligned} \tag{5.3}$$

where a generic matrix denoted as $A^{(h,p)}$ refers to the tracked pedestrian p at the h-th iteration of the tracker and t^* is an opportunely chosen threshold.



(*a*) (*b*) (*c*) (*d*) (*e*) (*f*) (*g*) (*h*) (*i*)

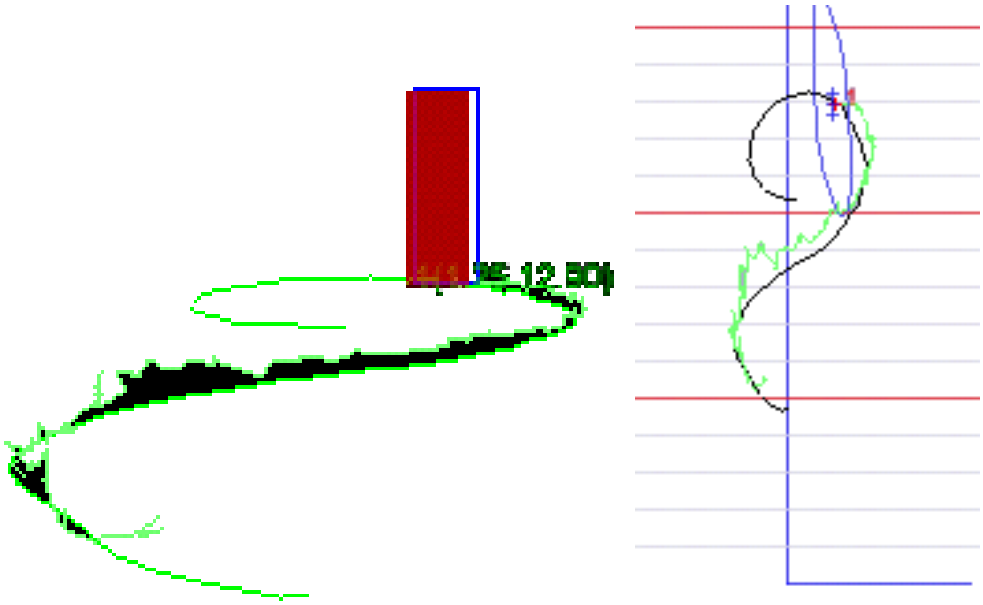
Figure 5.5: Comparison between the estimated paths and the trajectory: (*a*) Planar estimation for the slowly forward walking experiment; (*b*) normalized histogram of the error in the X_p coordinate for the slowly forward walking experiment; (*c*) histogram of the error in the Y_p coordinate; (*d*,*e*,*f*) analog data representation for the regular speed forward walking experiment; (*g*,*h*,*i*) data for the backwards running experiment.

5.4 Precision evaluation

Indoor experiments have been realized in order to verify the correctness of the estimated coordinates of the pedestrian path with the use of the tracker.
A reference trajectory has been set up simply by defining a set of way-points on the ground sufficiently close one to the next. The X and Y coordinates of all these points have been measured with classical measurement instruments in the chosen reference system. A calibrated camera has been positioned to look at this trajectory, with height and orientation as if it was installed inside a car. The position of the camera observing the reference points in the coordinate reference axes has been determined. Fig. 5.6 (*a*) shows a camera view of the pedestrian trajectory including perspective and fig. 5.6 (*b*) shows the associated bird-eye view.

The test includes the movement of a pedestrian along the predefined trajectory and the acquisition of the corresponding images in order to post-process them with the vision algorithm. An example of image acquired in this way is provided in fig. 5.7. For convenience the experiments have been realized indoor; due to this the images presented many vertical edges that lead to the generation of additional noise caused by the indoor structure. To solve this problem, a simple background subtraction has been applied in the middle of the processing; of course this is not needed in outdoor scenes: it is only used to make the localization verification possible.

Fig. 5.6 (*b*) shows a superposition example of the reference trajectory and of the trajectory estimated by the system. The current pedestrian position and its covariance ellipsis are also drawn together with the estimated trajectory.



(*a*) (*b*)

Figure 5.6: Pedestrian path and reference trajectory for the indoor acquisition as reported by the tracker when the *single-tracking* mode is selected: (*a*) perspective projection; (*b*) top view of the ground plane, the predefined trajectory is shown in black and the measured trajectory is shown in green.

Thanks to the use of a digital camera, the experiments have been characterized by a known intraframe temporal gap, so that the temporal synchronization of the estimated and of the reference trajectories has been made possible through a parametrization. Since the estimator provides the values of X_p and Y_p separately, it has been possible to compare the X and the Y coordinates independently. The maximal and average errors measured for the X_p and Y_p planar pedestrian coordinates of the experiments are reported in table 5.1; one time plot example of the euclidean error is in fig. 5.8. Fig. 5.9 shows plots and error composition histograms regarding the result-

Figure 5.7: Indoor test setup.

ing estimated paths for various ways of covering the trajectory. Considering that the Y coordinate is the one related to the depth of the scene relatively to the camera, the fact that the error on Y_p is greater than the one on X_p is not surprising and is a rather obvious conclusion in the field of computer vision; however, the overall precision is remarkable.

Another significant result obtained with these experiments is that the average errors on X_p and Y_p obtained by measurements coincide with the *a priori* error estimated at the output of the Kalman filter. This has the important meaning that the

Table 5.1: Mean and maximum coordinate errors (m)

Sequence	$\overline{e}_{X_p}$	Max e_{X_p}	$\overline{e}_{Y_p}$	Max e_{Y_p}
forward slowly	0.12	0.38	0.53	1.35
forward regular	0.15	0.34	0.37	1.16
forward running	0.47	1.48	0.47	2.04
forward natural	0.17	0.44	0.39	1.98
backwards slowly	0.66	1.54	4.07	6.89
backwards regular	0.65	1.38	4.19	6.98
backwards running	0.98	1.87	4.47	6.46
backwards natural	0.76	1.67	2.66	5.38

errors provided by the estimator can be considered reliable.

5.5 Discussion

The high-level tracking module for *environmental understanding* has evidenced with its filtering capabilities a good accuracy in the spatial localization of a walking pedestrian. The maximum errors from the measured path and the maximum variances along the axes that have been observed during the system activity on the indoor prerecorded image sequences, have proved a high reliability of the new approach. It has been possible therefore to adopt the new tracker module for the outdoor vehicular system activity and the multi trace results so obtained are illustrated in fig. 5.10. Integration of observations obtained from other different types of sensors can be easily achieved with the current system structure and can lead to more significant results in the form of the *data fusion* paradigm.

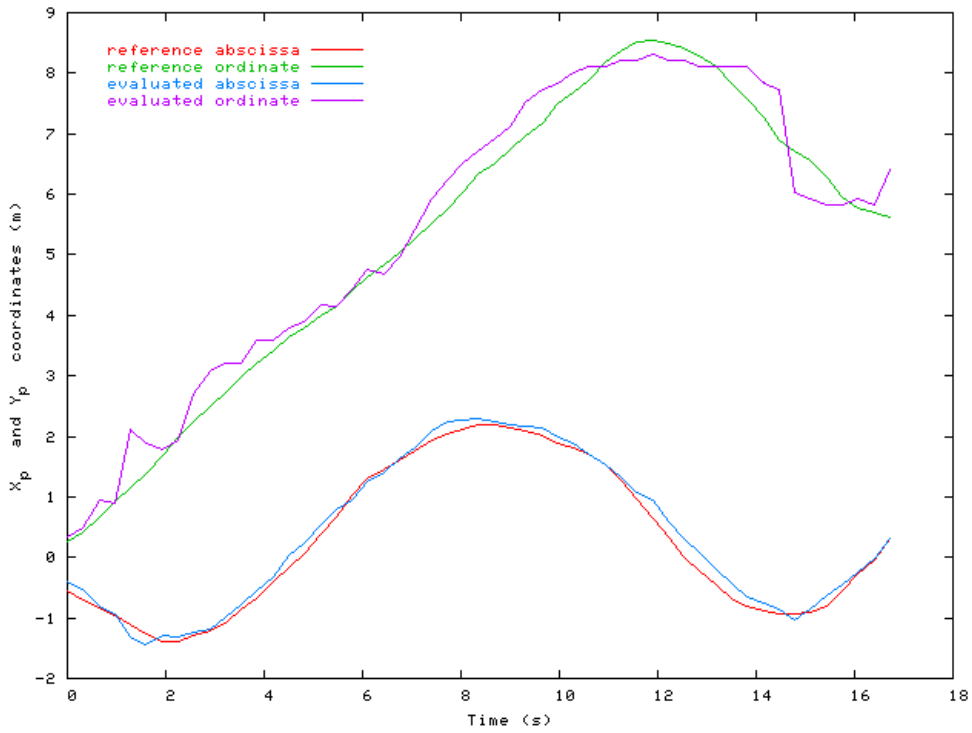


Figure 5.8: Temporal comparison example of ground plane coordinates between the imposed trajectory and the evaluated pedestrian path.

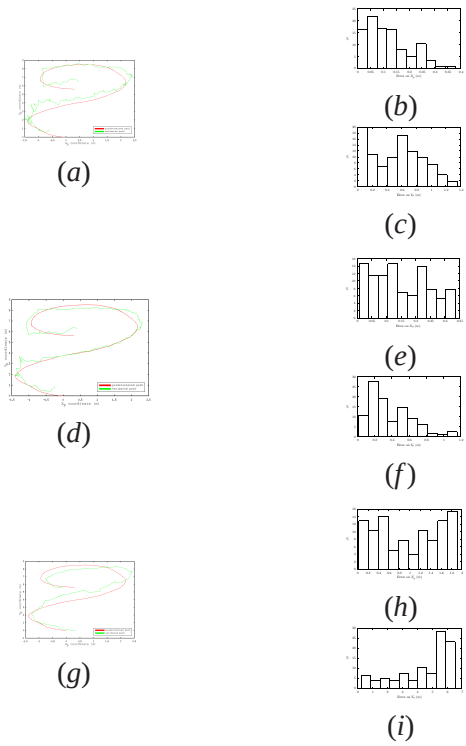
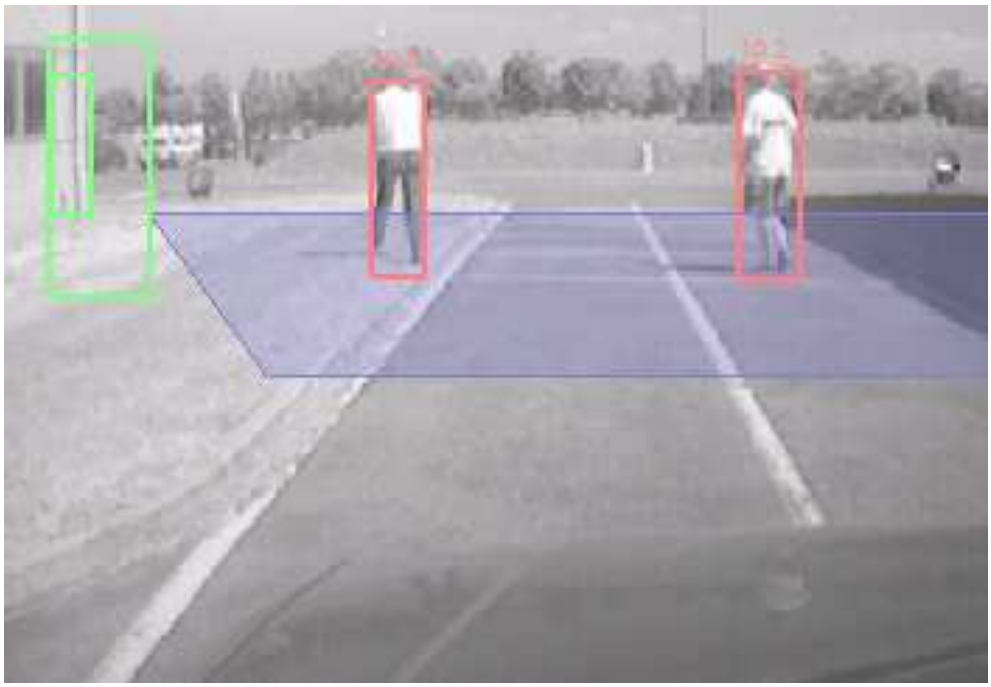
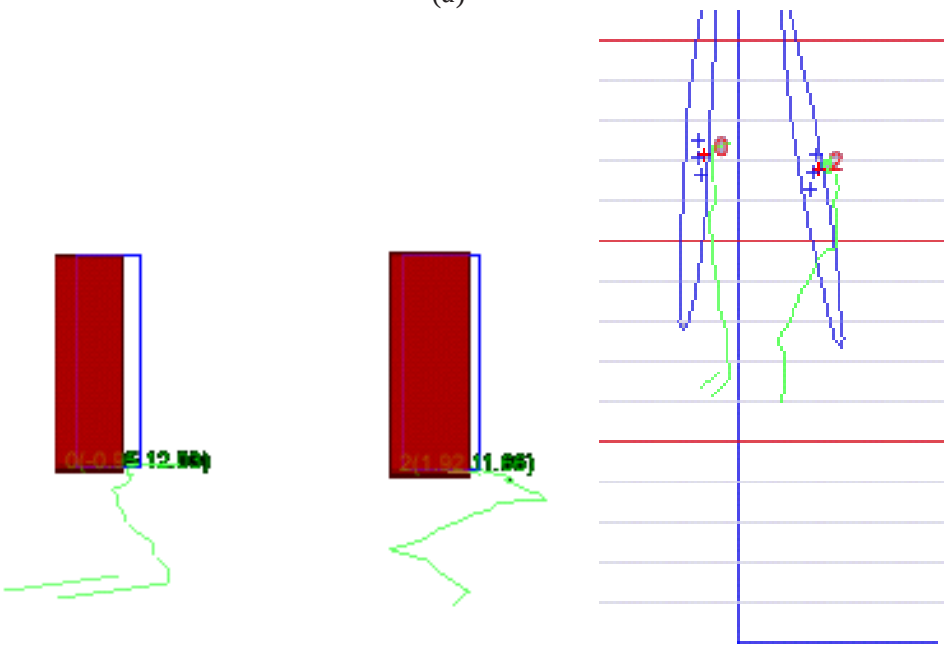


Figure 5.9: Comparison between the estimated paths and the trajectory: (*a*) Planar estimation for the slowly forward walking experiment; (*b*) normalized histogram of the error the X_p coordinate for the slowly forward walking experiment; (*c*) histogram of the error on the Y_p coordinate; (*d*,*e*,*f*) analog data representation for the regular speed forward walking experiment; (*g*,*h*,*i*) data for the backwards running experiment.



(*a*)



(*b*) (*c*)

Figure 5.10: Outdoor vehicular stereo results in *multi tracking* mode: (*a*) the vision algorithm recognizes the pedestrians (the stereo localization area is shown in transparent blue on the ground); (*b*) perspective view of the results, trajectories provided by the tracker are also shown; (*c*) pedestrian trajectories and error ellipsis of the current estimated pedestrian positions are represented on the road plane and marked with the corresponding pedestrian `id` (there is no correspondence between the reference grid of the graphical representation that represents the camera reference system and the grid painted on the asphalt).

Chapter 6

A tool for vision based pedestrian detection performance evaluation

Contenuto capitolo

This chapter describes a system for evaluating pedestrian detection algorithm results.
The developed tool allows a human operator to annotate on a file all pedestrians in a previously acquired video sequence. A similar file is produced by the algorithm being tested using the same annotation engine. A matching rule has been established to validate the association between items of the two files. For each frame a statistical analyzer extracts the number of mis-detections, both positive and negative, and correct detections. Using these data, statistics about the algorithm behavior are computed with the aim of tuning parameters and pointing out recognition weaknesses in particular situations.

6.1 Introduction

The detection of human shapes is one of the most active research objective in the field of artificial vision. Various approaches have recently been proposed (many applications rely on such detectors, like automotive precrash, security and surveillance systems) [2, 7, 22, 27]. An important issue at the basis of the design of a human shape detector is the availability of a tool for performance evaluation. Working on real images, because of the intrinsic problem complexity, some kind of external information is necessary in order to validate the algorithm results.

A system for performance evaluation needs to know the ”ground truth”, this can be obtained using two different approaches: recording additional data together with processed images data and using the annotation based approach, presented in this chapter. The former collects information about the pedestrians position using sensors different from vision, such as radio transmitters. The correctness of the algorithm can be evaluated in realtime but some problems may occur in cases, for example, where a pedestrian is partially occluded but the radio transmitter (or other) is anyway sensed by the detector. The other approach, the one dealt with in this chapter, relies on a frame by frame manual annotation, by a human operator, of all pedestrians appearing in each frame of a video sequence. This is a post processing operation, thus images must be acquired and saved on a storage device. Subsequently, the images are annotated in laboratory: a human operator, using a GUI, defines the position and size of pedestrians in each frame and produces a file containing the description of all pedestrian in the image sequence. A similar file of the same format is created by the

pedestrian detection algorithm which is under test. Finally the two files are compared and statistics are extracted. Parameters and thresholds can be adjusted and their effect on the algorithm behavior highlighted.

The outline of this chapter is as follows. Section 2 briefly reviews the state of the art in performance evaluation tools for vision algorithms. Section 3 describes the annotation tool composed by: engine, GUI, and performance analyzer. In section 4, an evaluation method for algorithms is proposed along with a case study. Finally, the chapter is concluded with a discussion describing results about the optimization of the case study.

6.2 Performance evaluation

This section describes the state of the art in performance evaluation for vision algorithms and in particular for pedestrian detection.

Vision applications proved their efficiency and usefulness in many fields but current research practices, and in particular system-building techniques, are inadequate especially for fine tuning and filter combination testing. One key aspect of this problem is the inability to conduct adequate performance characterization of new technologies (like pedestrian detectors). Reasons of this fact are due to the complexity of real scenes, sometimes pedestrians are occluded by other obstacles, sometimes parts of framed obstacles looks like pedestrians (even for humans observers).

The main purpose of a general approach to Performance Characterization of Computer Vision Systems [44] is the statistical testing, tuning, algorithmic combination and algorithmic re-use in order to improve algorithms reliability and robustness.

The work presented in [8] uses ground truth automatically extracted from pseudo synthetic video to perform evaluation of a pedestrian tracker on typical surveillance images taken from a fixed camera. However when dealing with real images taken from moving a cameras installed on a vehicle, the manual ground-truth generation approach seems to be more robust.

ViPER (Video Performance Evaluation Resource) described in [29] and [19] is a Java integrated tool for authoring ground truth meta-data in image sequences and evaluate performance of algorithms.

A similar system is proposed in this chapter. A key advantage of this tool is it's integration in a complete environment for the development of vision algorithms. This simplify design and tuning of parameters allowing to directly check the impact, of their variation, on performances.

The study presented in [27] points out the importance of a good performance evaluation method for the actual deployment of systems on board of vehicles. This study was based on a large number of tests. Results were analyzed using ROC curves to highlight the impact of parameters variations on the algorithm performance in terms of detection rate and false positive rate.

6.3 The annotation tool

6.3.1 Description

Performance evaluation using the annotation tool takes place in three steps: supervised sequence annotation by a human operator, automatic sequence annotation by the algorithm being tested, annotations comparison and analysis. Thanks to a com-

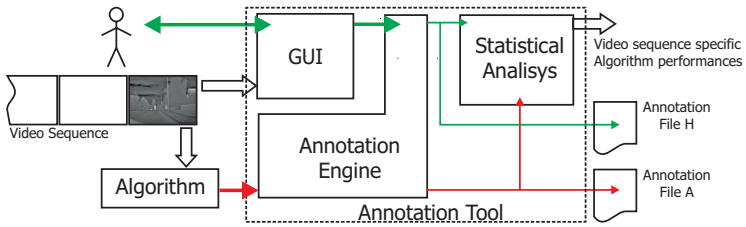


Figure 6.1: Block diagram of the algorithm: human and algorithm process the same video sequence producing annotation files H and A. The two files are compared producing statistics about algorithm performance

mon annotation engine, the tool allows the human operator and the algorithm to extract, into separate files, pedestrian information relative to the same image sequence. Pedestrians are described by means of the bounding boxes (BBs) framing their shape. The two files are compared and statistical information about the algorithm behavior are extracted. Each step is described below in detail. The performance evaluation tool structure is shown in figure 6.1.

Supervised sequence annotation by a human operator

During this step a human operator analyzes every frame of a pre-recorded video sequence. For each frame in the sequence the operator manually draws the BB around the pedestrian using a graphical user interface described in section 6.3.2. For each BB, the operator also locates the region containing the pedestrian's head. The head is an important human shape feature that can easily be found. The existence of informations describing heads allows to profile recognition performances of the algorithms that looks for this feature.

It is also possible to classify the pedestrian as completely visible or partially occluded by an obstacle. This description of the sequence ground truth is stored in a file named H shown in figure 6.1. An example reporting a frame during the annotation process is reported in figure 6.2.a.

Automatic sequence annotation by the algorithm under test

The output of a pedestrian detection algorithm can be described in terms of a list of BBs for each frame. Optionally, the algorithm can also produce information regarding the position of the head or some other interesting feature related to the pedestrian.

If needed an additional block can be added to the algorithm output stage in order to translate its results in a format compatible with the annotation engine input. For example if an algorithm extracts the human shapes from source images its output can be converted in a list of BBs each one defining the pixel area in the source image occupied by the pedestrian's shape. The description of the sequence produced by the algorithm is saved in a file named A also shown in figure 6.1. An example of BB generated by the algorithm under test is reported in figure 6.2.b.

Annotations Comparison and Analysis

This is the last step of the performance evaluation process. It takes as input the two previously created files and compares them frame by frame, extracting statistical information about the algorithm behavior. Three values are calculated for each frame: false positives (FP), false negatives (FN), and correct detections (CD).

In order to distinguish if a BB generated by the algorithm represents a correct

detection (CD) it is necessary to match it to all the BBs annotated by the human operator.

Two BBs, p and q, of area Z_p and Z_q respectively, are defined as *matching* if $Z_{pq} \doteq W_{pq}^2 / Z_p Z_q$ is greater than Z_{Th}, were W_{pq} is the overlapped area between p and q and Z_{Th} is a threshold adjustable by the user (a good value may be $Z_{Th} = 0.7$).

This relation embodies the following property: well overlapped BBs generate high values of Z_{pq}, but as the overlapping area decreases linearly, the value of Z_{pq} decreases at higher rate (square).

Every frame of the sequence analyzed through a particular algorithm can be modeled with the following two sets:

$H_n \doteq \{\text{BBs annotated by a human operator}\}$

$A_n \doteq \{\text{BBs annotated by the examined algorithm}\}$

where n is the frame number of a specific video sequence N frames long.

Let the symbol $|X|$ represents the cardinality of X, namely the number of elements in the set.

It is possible to define the matching operator $\oslash$ between a BB annotated by the operator and one annotated by the algorithm under testing in this way:

let $a_i \in A_n$ and $h_j \in H_n$:

$$a_i \oslash h_j \doteq \begin{cases} 1 & Z_{a_i h_j} = max\{Z_{a_i h_k} | Z_{a_i h_k} > Z_{Th}\} \\ 0 & otherwise \end{cases}$$

Based on this definition $\forall a_i \in A_n$ exists at most one j such that $a_i \oslash h_j = 1$. In fact, given a_i, $Z_{a_i h_j} > Z_{th}$ for different values of j. This ambiguity is resolved by the $max()$ function.

(a) (b) (c)

Figure 6.2: (*a*) Annotation window during the human supervised annotation process: the currently selected BB is cyan filled (it can be resized or moved), non-filled green BBs are non selected BB, the yellow one is marked as occluded, the green-filled one is currently being drawn by the operator. Each BB is composed of two rectangles framing the pedestrian shape and head. (*b*) BBs generated by the algorithm. The red numbers indicate the distance of the pedestrian, the violet area represents the 3D space where stereo vision can be applied. (*c*) Matching phase result: BBs found by the pedestrian detector are presented in red, annotated BBs are in blue and yellow (if occluded), matched BBs are in green.

A graphic representation of the matching process result is reported in figure 6.2.c.

Using these definitions, three different values are defined:

$$\begin{aligned} CD_n &= \sum_{i=0}^{|A_n|} \sum_{j=i}^{|H_n|} a_i \oslash h_j \\ FP_n &= |A_n| - CD_n \\ FN_n &= |H_n| - CD_n \end{aligned}$$

These values represent respectively the number of correct detections, false positives, and false negatives for the *n*-th frame of the sequence. In this way it is possible

to identify frames in which the algorithm works fine and situations related to algorithm weaknesses.

Now a set of global values is defined referred to the whole sequence in order to compare different algorithms working on that sequence.

$$CDR = \frac{\sum_{n=0}^{N-1} CD_n}{|H_n|} \tag{6.1}$$

$$FPR = \frac{\sum_{n=0}^{N-1} FP_n}{N} \tag{6.2}$$

These values are sequence specific and measure respectively: the correct detection rate and the false positive rate. FPR cannot be normalized because false positives have no upper limit.

6.3.2 User Interface

The input interface has been designed with the objective of reducing, as much as possible, the workload for frame annotation. An example of the annotation window during the drawing process of a new pedestrian is presented in figures 6.2.a. In figure 6.6.b is reported the annotation panel during the annotation process. The key points for reducing the annotation time are the following:

Similarity between consecutive frames. Usually a frame in a real-time sequence contains little differences from the previous one. For this reason it can be assumed that a BB containing a pedestrian will have a similar position and size in the subse-

quent frame, possibly with some little corrections. To remove the need for a complete redrawing of BBs on every frame of a sequence, the GUI copies all BBs of a certain frame to the following one, leaving to the operator the task of adjusting size and position, as well as adding and deleting new and disappeared BBs.

Easy input method. The interface has been studied keeping ergonomics in mind: the operator uses one hand to command the mouse and the other one for the keyboard. In this way all important commands such as tracing, resizing and repositioning of BBs are directly available to the user. Moreover using the mouse wheel the operator can select the target BB to modify. Some additional keyboard commands allow to speed up common operations such as deleting all BBs in the frame.

It has been proven that this kind of interface is user friendly. This GUI allows a human operator to annotate about 100 frames/h. This number is the average speed obtained from 10 different users who never used the tool before, each annotating 200 frames. However it is reasonable to assume this speed will increases with experience. A more accurate drawing of BBs can be obtained magnifying the tracing area.

6.4 Algorithms Evaluation Space

This section contains some considerations regarding algorithms evaluation.

The values *CDR* and FPR introduced in the previous section were referred to a single sequence. Indeed in order to have a more general and robust statistical description of the algorithm these values must be computed on a sufficiently large sequence including a wide variety of different scenarios.

The 2D space $< FPR, CDR >$ is defined as shown in figure 6.3; the optimal

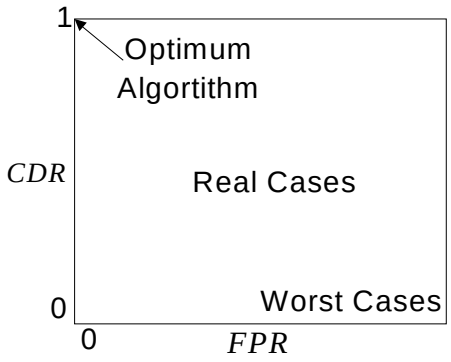


Figure 6.3: Vision based pedestrian detection normalized evaluation space.

algorithm is placed in the point $(0,1)$. Namely, all algorithms that do not give any false positive should stay in the segment $[0,a]$ with $a \in [0,1]$. Algorithms with bad performance fall in the right bottom part of the space while real cases fall inside the central area.

This kind of evaluation allows to determine if an algorithm modification improves (even slightly) the recognition performance. For example this system is useful to fine tune parameters and thresholds. It is possible to evaluate the impact that a parameter modification has on the algorithm performance observing the movement of the point representing the algorithm in the evaluation space. Moreover the inclusion of new filters can be evaluated measuring performance variations. Indeed the time consuming annotation process that requires the human supervision is performed only once for every sequence.

It is also possible to define a metric (for example the euclidean distance from the [0,1] point) to asses improvements. It is necessary to underline that the specific optimality criterion is strictly dependent on the application. In some applications,

such as quality control for example, it may be desirable to avoid false positives and disregard false negatives. In other applications, such as automotive precrash systems, some false positives may be acceptable, even if an excessive number of FP reduces the user confidence in the recognition system. In these two cases the distance from the optimum point should be defined in different ways.

6.5 Discussion

The performance evaluation tool described in this paper has been used to evaluate pedestrian detection algorithms. In particular, the algorithm presented in [13] has been chosen as a case study. A number of sequences for nearly 1500 images have been manually annotated. The sequences were taken in different scenarios (parking lot, open field, and downtown), under different illumination and weather conditions, and framed different subjects at different distances. The aim of this step was to create a test set describing many of the the possible cases that the algorithm can deal with. In the test sequence, composed of 1500 images, 1897 human shapes were annotated as completely visible while 361 were marked as partially occluded. Figure 6.4 shows a number of different situations for the test sequence.

The overall system performance shows that the correct detection rate is about 83% (1572/1897=82.9%). The high sensitivity of the algorithm (83%) comes along with an appreciable false positives rate 0.46 FP per frame. Indeed, a set of higher thresholds in the algorithm would decrease the number of false positives, but, at the same time would reduce the correct detection rate. The false negatives rate (426/1897=26.1%) summed up with the correct detection rate exceeds 100% (82.9%+26.1%=109%)

since the latter also includes occasional detections of occluded human shapes.

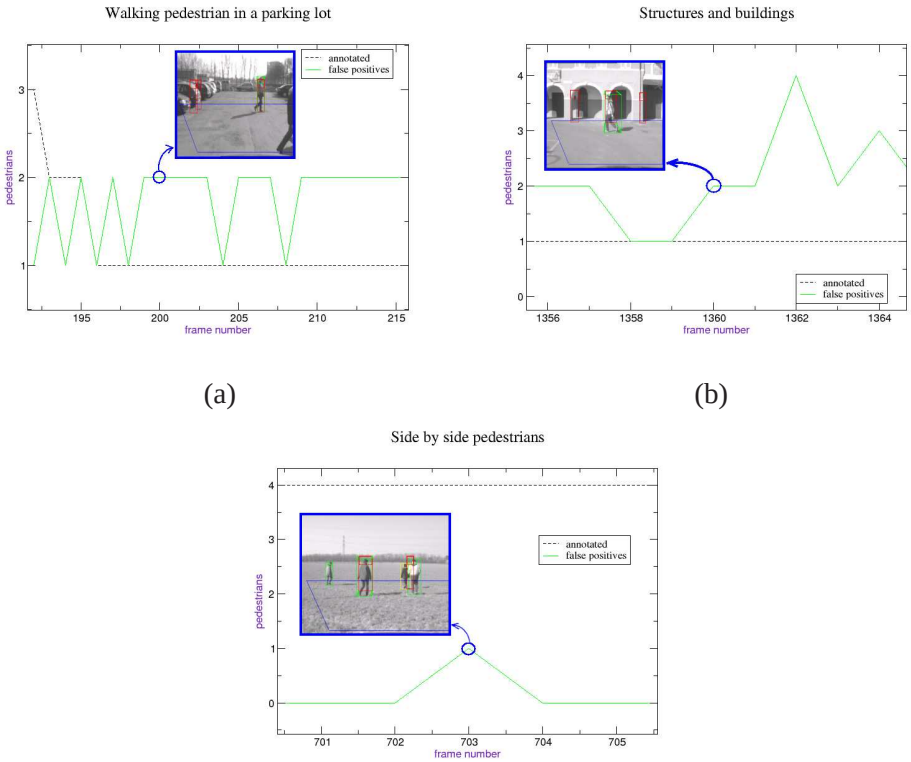


Figure 6.4: Examples of situations in which the human shape detection algorithms partially fails: (*a*) background noise generated by parked vehicles introduces false positives; (*b*) columns generate false positives due to their symmetry; (*c*) two pedestrians walking side by side mislead symmetry evaluation.

The main result of these tests were the statistical characterization of the detector behaviour and the precise identification of particularly challenging segments of a large sequence. The program output while extracting statistics from the algorithm is shown in figures 6.5 and 6.6.b.

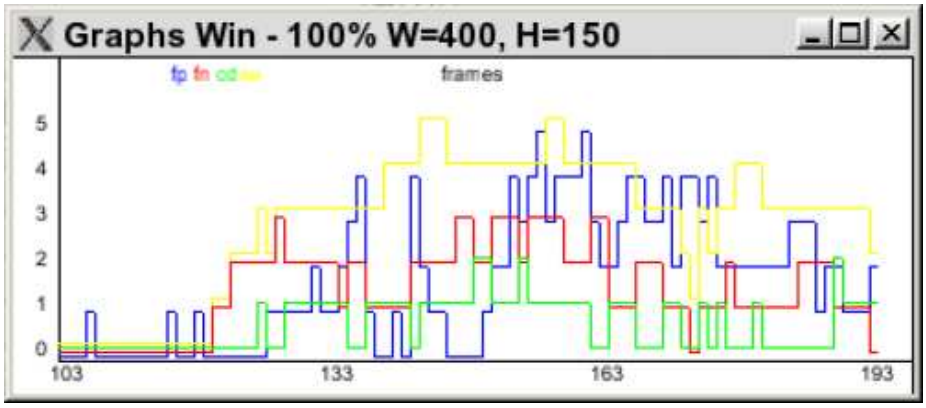
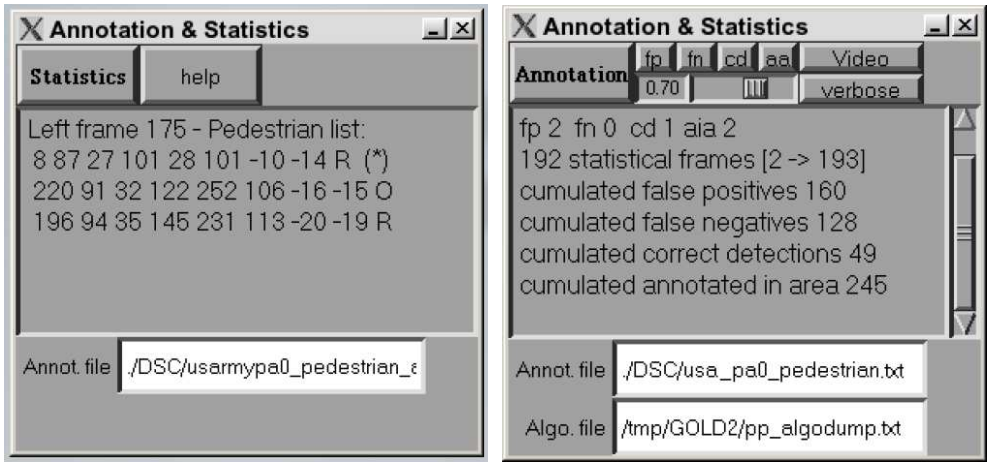


Figure 6.5: Statistics extraction screen-shot: for each frame values of CD, FP, FN and human annotated (in yellow) are computed.



(a) (b)

Figure 6.6: (*a*) Statistics extraction screen-shot: for each frame values of CD, FP, FN are computed and cumulated up to the current frame. (*b*) Annotation panel during the human supervised annotation process: information about current operation are displayed

The presented performance evaluation tool has been proven to be effective though it requires a very expensive annotation process. The time required to annotate one

frame is a value than can be reduced only by either modifying the input method (finding more efficient shortcuts for frequent operations) or trying to detect off-line the BB modifications. Thus, improvements should be in GUI refining and in automatic resize/reposition of the bounding boxes using motion detection techniques whenever possible. GUI refinement can be done following impressions of user that performs longs annotations. Motion detection techniques such as correlation analysis between frames and optical flow can be used to determine the new coarse position and size of BBs in new frame starting from those in the previous frames. It is necessary to consider that the time spent to perform such detection can't be too hight in order to maintain the number of annotated frames/h comparable with the human one. Relying on an accurate prediction of the new BBs position, operator's task would be reduced to a mere supervision activity reducing the time spent to annotate each single frame.

This study shows a tool to detect positive aspects and weakness points of a pedestrian detector working on a given video sequence. The tool can also serve as a performance comparison method between different algorithms.

Bibliography

[1] Massimo Bertozzi and Alberto Broggi. GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Trans. on Image Processing*, 7(1):62–81, January 1998.

[2] Massimo Bertozzi, Alberto Broggi, Roland Chapuis, Frédéric Chausse, Alessandra Fascioli, and Amos Tibaldi. Shape-based pedestrian detection and localization. In *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2003*, pages 328–333, Shangai, China, October 2003.

[3] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. Stereo Inverse Perspective Mapping: Theory and Applications. *Image and Vision Computing Journal*, 8(16):585–590, 1998.

[4] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. Visual Perception and Learning in Road Environments. In *Procs. 6th Intl. Conf. on Intelligent Autonomous Systems, IAS-6*, pages 885–892, Venice, Italy, July 2000.

[5] Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli, and Paolo Lombardi. Vision-based Pedestrian Detection: will Ants Help? In *Procs. IEEE Intelligent Vehicles Symposium 2002*, volume 1, pages 1–7, Paris, France, June 2002.

[6] Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli, and Massimiliano Sechi. Shape-based Pedestrian Detection. In *Procs. IEEE Intelligent Vehicles Symposium 2000*, pages 215–220, Detroit, USA, October 2000.

[7] Massimo Bertozzi, Alberto Broggi, Thorsten Graf, Paolo Grisleri, and Marc-Michael Meinecke. Pedestrian Detection in Infrared Images. In *Procs. IEEE Intelligent Vehicles Symposium 2003*, pages 662–667, Columbus, USA, June 2003.

[8] James Black, Tim Ellis, and Paul Rosin. A Novel Method for Video Tracking Performance Evaluation. In *Procs. Joint IEEE Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France, October 2003.

[9] B. Brendle. Cockpit Development in the Crew Integration and Automation Testbed advanced Technology Development Program. In *Procs. SPIE - AeroSense Conference 2003*, volume 5080, Orlando, FL, April 2003.

[10] B. Brendle and J. Jaczkowski. Robotic Follower: Near-Term Autonomy for Future Combat Systems. In *Procs. SPIE - AeroSense Conference 2002*, volume 4715, Orlando, FL, April 2002.

[11] Alberto Broggi, Massimo Bertozzi, Gianni Conte, and Alessandra Fascioli. ARGO Prototype Vehicle. In Ljubisa Vlacic, Fumio Harashima, and Michel Parent, editors, *Intelligent Vehicle Technologies*, chapter 14, pages 445–493. Butterworth–Heinemann, London, UK, June 2001. ISBN 0750650931.

[12] Alberto Broggi, Massimo Bertozzi, Alessandra Fascioli, and Gianni Conte. *Automatic Vehicle Guidance: the Experience of the ARGO Vehicle*. World Scientific, Singapore, April 1999. ISBN 9810237200.

[13] Alberto Broggi, Michael Del Rose, Alessandra Fascioli, Isabella Fedriga, and Amos Tibaldi. Stereo-based Preprocessing for Human Shape Localization in Unstructured Environments. In *Procs. IEEE Intelligent Vehicles Symposium 2003*, pages 410–415, Columbus, USA, June 2003.

[14] Alberto Broggi and Alessandra Fascioli. Artificial Vision in Extreme Environments for Snowcat Tracks Detection. *IEEE Trans. on Intelligent Transportation Systems*, 3(3):162–172, September 2002.

[15] Alberto Broggi, Alessandra Fascioli, Marcello Carletti, Thorsten Graf, and Marc-Michael Meinecke. A Multi-resolution Approach for Infrared Vision-based Pedestrian Detection. In *Procs. IEEE Intelligent Vehicles Symposium 2004*, pages 7–12, Parma, Italy, June 2004.

[16] Jill D. Crisman and Charles E. Thorpe. UNSCARF, A Color Vision System for the Detection of Unstructured Roads. In *Procs. IEEE Intl. Conf. on Robotics and Automation*, pages 2496–2501, Sacramento, CA, April 1991.

[17] Cristobal Curio, Johannes Edelbrunner, Thomas Kalinke, Christos Tzomakas, and Werner von Seelen. Walking Pedestrian Recognition. *IEEE Trans. on Intelligent Transportation Systems*, 1(3):155–163, September 2000.

[18] Ross Cutler and Larry S. Davis. Robust Real-time Periodic Motion Detection, Analysis and Applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):781–796, August 2000.

[19] David Doermann and David Mihalcik. Tools and Techniques for Video Performance Evaluation. In *Procs. IEEE Intl. Conf. on Pattern Recognition*, volume 4, pages 167–170, Barcelona, Spain, September 2000.

[20] Marco Dorigo and Gianni Di Caro. The ant colony optimization meta-heuristic. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw-Hill, London, UK, 1999.

[21] Marco Dorigo and Luca Maria Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Tran. on Evolutionary Computation*, 1(1):53–66, April 1997.

[22] Hadi Elzein, Sridhar Lakshmanan, and Paul Watta. A Motion and Shape-Based Pedestrian Detection Algorithm. In *Procs. IEEE Intelligent Vehicles Symposium 2003*, pages 500–504, Columbus, USA, June 2003.

[23] Alessandra Fascioli. *Vision-based Automatic Vehicle Guidance: Development and Test of a Prototype*. PhD thesis, Dipartimento di Ingegneria dell'Informazione, Università di Parma, Italy, January 2000.

[24] K. Fujimoto, H. Muro, N. Shimomura, T. Oki, Y. Kishi K. Maeda, and M. Hagino. A Study on Pedestrian Detection Technology using Stereo Images. *JSAE Review*, 23(3):383–385, August 2002.

[25] Dariu M. Gavrila. Pedestrian Detection from a Moving Vehicle. In *Procs. of European Conference on Computer Vision*, volume 2, pages 37–49, June–July 2000.

[26] Dariu M. Gavrila. Sensor-based Pedestrian Protection. *IEEE Intelligent Systems*, 16(6):77–81, November–December 2001.

[27] Dariu M. Gavrila and J. Geibel. Shape-Based Pedestrian Detection and Tracking. In *Procs. IEEE Intelligent Vehicles Symposium 2002*, Paris, France, June 2002.

[28] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who, When, Where, What: A Real Time System for Detecting and Tracking People. *Image and Vision Computing Journal*, 17(1), January 1999.

[29] Crystopher Jaynes, Stephen Weeb, R. Matt Steele, and Quanren Xiong. Development Environment for Evaluation of Video Surveillance Systems. In *Procs. IEEE Intl. Workshop on Performance Analysis of Video Surveillance and Tracking*, Copenhagen, Denmark, June 2002.

[30] Todd M. Jochem, Dean A. Pomerleau, and Charles E. Thorpe. MANIAC: A Next Generation Neurally Based Autonomous Road Follower. In *Procs. 3rd Intl. Conf. on Intelligent Autonomous Systems*, Pittsburgh, USA, February 1993.

[31] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME Journal of Basic Engineering*, 82(1):35–45, March 1960.

[32] Seonghoon Kang, Hyeran Byun, and Seong-Whan Lee. Real-Time Pedestrian Detection Using Support Vector Machines. *Lecture Notes in Computer Science*, 2388:268, February 2002.

[33] Dieter Koller, Joseph Weber, T. Huang, Jitendra Malik, G. G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Procs. IAPR Conf. on Computer Vision & Image Processing*, pages 126–131, October 1994.

[34] Raphaël Labayrade, Didier Aubert, and Jean-Phillipe Tarel. Real Time Obstacle Detection in Stereo Vision on non Flat Road Geometry through "V-Disparity" Representation. In *Procs. IEEE Intelligent Vehicles Symposium 2002*, Paris, France, June 2002.

[35] D. Makris and T. Ellis. Path Detection in Video Surveillance. *Image and Vision Computing Journal*, 20(12):895–903, October 2002.

[36] Stephen J. McKenna and Shaogang Gong. Non-intrusive Person Authentication for Access Control by Visual Tracking and Face Recognition. *Lecture Notes in Computer Science*, 1206:177–184, March 1997.

[37] Sridhar Lakshmanan Michael Beuvais, Chris Kreucher. Building World Model for Mobile Platforms using Heterogeneous Sensors Fusion and Temporal Anal-

ysis. In *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems'97*, page 101, Boston, USA, November 1997.

[38] Harsh Nanda, Chiraz Benabdelkedar, and Larry Davis. Modelling Pedestrian Shapes for Outlier Detection: a Neural Net based Approach. In *Procs. IEEE Intelligent Vehicles Symposium 2003*, pages 428–433, Columbus, USA, June 2003.

[39] Constantine Papageorgiou, Theodoros Evgeniou, and Tomaso Poggio. A Trainable Pedestrian Detection System. In *Procs. IEEE Intelligent Vehicles Symposium'98*, pages 241–246, Stuttgart, Germany, October 1998.

[40] V. Philomin, R. Duraiswami, and L. Davis. Pedestrian Tracking from a Moving Vehicle. In *Procs. IEEE Intelligent Vehicles Symposium 2000*, pages 350–355, Detroit, USA, October 2000.

[41] Dean A. Pomerleau and Todd Jochem. Rapidly Adapting Machine Vision for Automated Vehicle Steering. *IEEE Expert*, 11(2):19–27, April 1996.

[42] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context Free Attentional Operators: the Generalized Symmetry Transform. *Intl. Journal of Computer Vision, Special Issue on Qualitative Vision*, 14:119–130, 1994.

[43] Y. Song, X. Feng, and P. Perona. Towards detection of humans. In *Procs. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 810–817, South Carolina, USA, June 2000.

[44] Neil Thacker. Performance characterization in computer vision. Technical report, PCCV project of the EU-IST programme, July 2003.

[45] C. Wöhler, U. Kreßel, and J. K. Anlauf. Pedestrian Recognition by Classification of Image Sequences – Global Approaches vs. Local Spatio-Temporal Processing. In *Procs. IEEE Intl. Conf. on Pattern Recognition*, Barcelona, Spain, September 2000.

[46] Christian Wöhler, Joachim K. Aulaf, Till Pörtner, and Uwe Franke. A Time Delay Neural Network Algorithm for Real-time Pedestrian Detection. In *Procs. IEEE Intelligent Vehicles Symposium'98*, pages 247–251, Stuttgart, Germany, October 1998.

[47] Liang Zhao and Charles Thorpe. Stereo and neural network-based pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 1(3):148–154, September 2000.