1

3D Visual Information for Dynamic Objects Detection and Tracking During Mobile Robot Navigation

D.-L. Almanza-Ojeda* and M.-A. Ibarra-Manzano
Digital Signal Processing Laboratory, Electronics Department; DICIS,
University of Guanajuato, Salamanca, Guanajuato
Mexico

1. Introduction

An autonomous mobile robot that navigates in outdoor environments requires functional and decisional routines enabling it to supervise the estimation and the performance of all its movements for carrying out an envisaged trajectory. At this end, a robot is usually equipped with several high-performance sensors. However, we are often interested in less complex and low-cost sensors that could provide enough information to detect in real-time when the trajectory is free of dynamic obstacles. In this context, our strategy was focused on visual sensors, particulary on stereo vision since this provides the depth coordinate for allowing a better perception of the environment. Visual perception for robot mobile navigation is a complex function that requires the presence of "salience" or "evident" patrons to identify something that "breaks" the continuous tendency of data. Usually, interesting points or segments are used for evaluating patrons in position, velocity, appearance or other characteristics that allows us forming groups (Lookingbill et al., 2007), (Talukder & Matthies, 2004). Whereas complete feature vectors are more expressive for explaining objects, here we use 3D feature points for proposing a strategy computationally less demanding conserving the main objective of the work: detect and track moving objects in real time.

This chapter presents a strategy for detecting and tracking dynamic objects using a stereo-vision system mounted on a mobile robot. First, a set of interesting points are extracted from the left image. A disparity map, provided by a real-time stereo vision algorithm implemented on FPGA, gives the 3D position of each point. In addition, velocity magnitude and orientation are obtained to characterize the set of points on the space R^6. Groups of dynamic 2D points are formed using the *a contrario* clustering technique in the 4D space and then evaluated on their depth value yielding groups of dynamic 3D-points. Each one of these groups is initialized by a convex contour with the velocity and orientation of the points given a first estimation of the dynamic object position and velocity. Then an active contour defines a more detailed silhouette of the object based on the intensity and depth value inside of the contour. It is well known that active contour techniques require a highly dense computations. Therefore, in order to reduce the time of processing a fixed number of iterations is used at each frame, so the convergence of the object real limits will be incrementally achieved along

*Part of this work was developed when authors were with LAAS-CNRS, Toulouse, France several frames. A simple and predefined knowledge about the most usual dynamic objects found in urban environments are used to label the growing regions as a rigid or non-rigid object, essentially cars and people. Experiments on detection and tracking of vehicles and people, as well as during occlusion situations with a mobile robot in real world scenarios are presented and discussed.

2. Related works

The issue of moving object detection has been largely studied by the robotic and computer vision community. Proposed strategies use mainly a combination of active and passive sensors mounted on the mobile robot like laser (Vu & Aycard, 2009) with cameras (Katz et al., 2008), or infrared cameras (Matthies et al., 1998), just to name a few. However, a multi-sensor system requires to solve the problem of fusing the data from different sources which often requires more complex estimation cases. Indeed, more information could be acquired using several sensors but these systems are expensive and complex. To overcome those constraints, the proposed solution consist in using one or more cameras as the only source of information in the system (Talukder & Matthies, 2004), (Williamson, 1998). Essentially vision sensors provide enough information for localization and mapping (Sola et al., 2007) or for describing static and moving objects on the environment (Klappstein et al., 2008).
The stereo-vision is one of the most used techniques for reconstructing the 3D (depth) information of a scene from two images, called left and right. This information is acquired from two cameras separated by a previously established distance. The disparity map is a representation that contains the depth information of the scene. It is well known that dense stereo-vision delivers more complete information than sparse stereo-vision but this is a high-processing cost technique which enables to perform in real time using and ordinary computer system. We use a stereo-vision technique in order to detect moving objects but implemented on a re-configurable architecture that maximizes the efficiency of the system. In the last decade, several works have proposed the development of high-performance architectures to solve the stereo-vision problem i.e. digital signal processing (DSP), field programmable gate arrays (FPGA) or application-specific integrated circuits (ASIC). The ASIC devices are one of the most complicated and expensive solutions, however they afford the best conditions for developing a final commercial system (Woodfill et al., 2006). On the other hand, FPGA have allowed the creation of hardware designs in standard, high-volume parts, thereby amortizing the cost of mask sets and significantly reducing time-to-market for hardware solutions. However, engineering cost and design time for FPGA-based solutions still remain significantly higher than software-based solutions. Designers must frequently iterate the design process in order to achieve system performance requirements and simultaneously minimize the required size of the FPGA. Each iteration of this process takes hours or days to be completed (Schmit et al., 2000). Even if designing with FPGAs is faster than designing ASICs, it has a finite resource capacity which demands clever strategies for adapting versatile real-time systems (Masrani & MacLean, 2006).

2.1 Overall strategy for 3D dynamic object detection from a mobile robot

A seminal work of this strategy was presented in (Almanza-Ojeda et al., 2010) and (Almanza-Ojeda et al., 2011). Whereas the former proposes a monocamera strategy (Almanza-Ojeda et al., 2010), and the latter the fusion with the information provided by inertial (IMU) (Almanza-Ojeda et al., 2011), here we propose an extension of this strategy to a stereo vision images provided by a bank stereo mounted on the robot. The stereo images are

processed by a stereo vision algorithm designed on FPGA that calculates the disparity map which provides the depth at each point of the input image. Thus, we use a combination of the 3D and 2D representation of feature points for grouping them according to similar distribution function in position and velocity. Moreover, these groups of points permit us to initialize an active contour for obtaining the object boundary. Therefore, the contour initialization of the detected points is fundamental in order to properly perform the shape recovering.
A block diagram of the proposed generalized algorithm is depicted in figure 1. A stereo vision module for FPGA based on Census Transform provides the disparity map using left and right images. At the same time, a selection-tracking process of feature points is carried out on a loop. However, rather than considering both stereo vision images during interest point selection, we will consider only left image for obtaining these 2D feature points. Further, each feature point will be associated with a depth value given by the disparity map. At this point it is necessary to point out that clustering task is performed until a small number of images have been processed. According to this, clustering module will receive a spatial-temporal set of points since each feature location and velocity have been accumulated through the time. The clustering method that we use is the *a contrario* method proposed by Veit et. al. (Veit et al., 2007). This clustering method consist in grouping the interesting points which have a "coherent" movement along a short number of consecutive images. Here the term coherent refers to movements that follow a similar and constant magnitude and direction described by the probability density function of the points under evaluation.

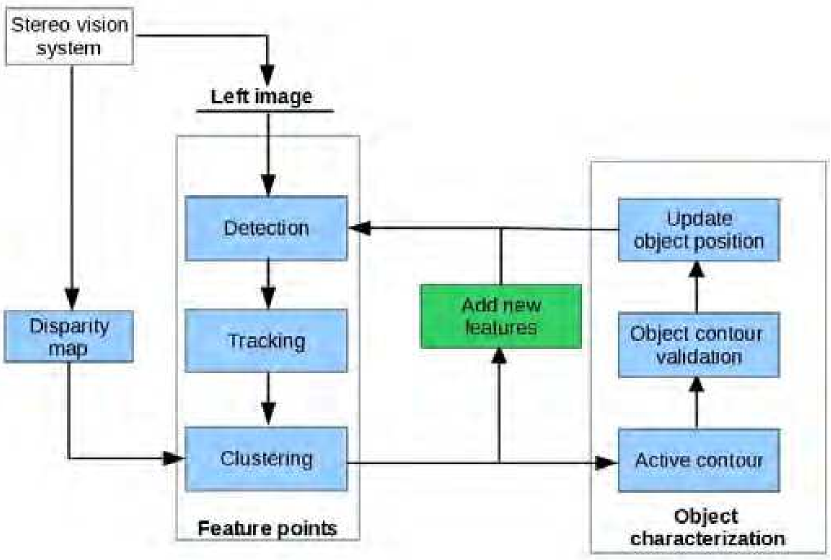


Fig. 1. Global strategy of object detection and identification using a stereo vision system mounted on a mobile robot

Once dynamic group of points have been detected by the clustering process, from this group of points it is possible to delimit a region and an irregular contour in the image with the aim of characterizing a probable dynamic object. Thus, by doing an analysis of intensity inside of this contour, we propose to match the provided region along the image sequence in order to follow the non-rigid objects that have more complicated movement, like pedestrians.
This work provides two distinct contributions: (1) the addition of depth information for performing the clustering of 3D set of dynamic points instead of only 2D as in the *a contrario* clustering technique (Veit et al., 2007), (2) the use of dynamic groups for initializing

an irregular contour active that temporally recovers actual object boundary. The second contribution was preliminary discussed in (Almanza-Ojeda et al., 2011) where the same active contour was initialized for representing object region, however that only uses a snake based on contour information, so nothing is doing with features inside of the region.

The structure of the chapter is as follows. In section 3, we describe the stereo vision module and its performance. Then section 4.1 details interest 2D point selection and how the 3D points are obtained using the disparity map. The grouping technique of the interest points based on the *a contrario* clustering is explained in section 4.2. We present in section 5 object shape recovering using active contours. Section 6 contains our experimental results from indoor an outdoor environments. Finally, we end up with conclusions in section 7.

3. Stereo vision module

We use a disparity map calculated using the Census Transform algorithm (Zabih & Woodfill, 1994) implemented on a programmable device, in this case a FPGA (Field Programmable Gate Array). The architecture of the Census transform algorithm was developed by (Ibarra-Manzano, Devy, Boizard, Lacroix & Fourniols, 2009) in which left and right images acquired from the stereo vision bank are processed for generating up to 325 dense disparity maps of 640 × 480 pixels per second. It is important to point out that most of the vision-based systems do not require high video-frame rates because usually they are implemented on computers or embedded platforms which are not FPGA-based. As this is also our case, we have adapted the disparity map generation to the real-time application required by our system by tuning some configuration parameters in the architecture. A block diagram of the stereo vision algorithm is shown in figure 2. In the following, we will describe in general the calculation of the disparity map based on the Census transform. However, the architectural implementation on the FPGA is a problem that has not dealt with in this work, all these details will be found in (Ibarra-Manzano, Devy, Boizard, Lacroix & Fourniols, 2009), (Ibarra-Manzano & Almanza-Ojeda, 2011).

In the stereo vision algorithm each of the images (right and left) are processed independently in parallel. The process begins with the rectification and correction of the distortion for each image in order to decrease the size of the search of points to a single dimension during disparity calculation. This strategy is known as epipolar restriction in which, once the main axes of the cameras have been aligned in parallel, founding the displacement of the position between the two pixels (one per camera) is reduced to search in each aligned line. That is, if any pair of pixels is visible in both cameras and assuming they are the projection of a single point in the scene, then both pixels must be aligned on the same epipolar line (Ibarra-Manzano, Almanza-Ojeda, Devy, Boizard & Fourniols, 2009). Therefore under this condition, an object location in the scene is reduced to a horizontal translation. Furthermore, the use of the epipolar restriction allows to reduce the complexity and the size of the final architecture.

Next, rectified and corrected images are filtered using an arithmetic mean filter. Once input images have been filtered, they are used to calculate the Census Transform as depicted in the figure 2. This transform is a non-parametric measure used during the matching process for measuring similarities and obtaining the correspondence between the points into the left and right images. A neighborhood of pixels is used for establishing the relationships among them. From the Census Transform, two images are obtained referred to as I_{Cl} and I_{Cr} which represent the left and right Census images. Two pixels extracted from the Census images (one for each image) are compared using the Hamming distance. This comparison

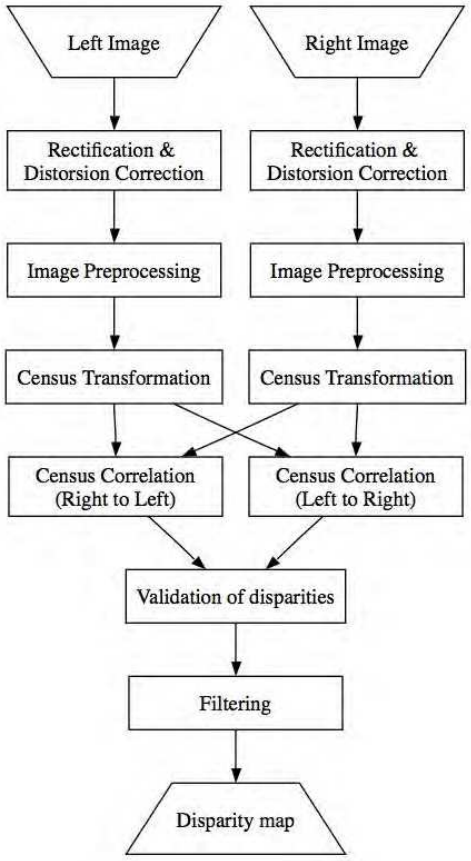

Left Image
Right Image
Rectification & Distorsion Correction
Rectification & Distorsion Correction
Image Preprocessing
Image Preprocessing
Census Transformation
Census Transformation
Census Correlation (Right to Left)
Census Correlation (Left to Right)
Validation of disparities
Filtering
Disparity map

Fig. 2. Block diagram of the stereo vision algorithm (Ibarra-Manzano & Almanza-Ojeda, 2011)

which is called the correlation process allows us to obtain a disparity measure. The disparity measure comes from the similarity maximization function in the same epipolar line for the two Census images. The similarity evaluation is based on the binary comparison between two bit chains calculated by the Census Transform. The correlation process is carried out two times, (left to right then right to left) with the aim of reducing the disparity error and thus complementing the process. Once both disparity measures have been obtained, the disparity measure validation (right to left and left to right) consists of comparing both disparity values and obtaining the absolute difference between them. Before delivering a final disparity image, a novel filtering process is needed for improving its quality. In this final stage, the use of a median spatial filter was found more convenient.

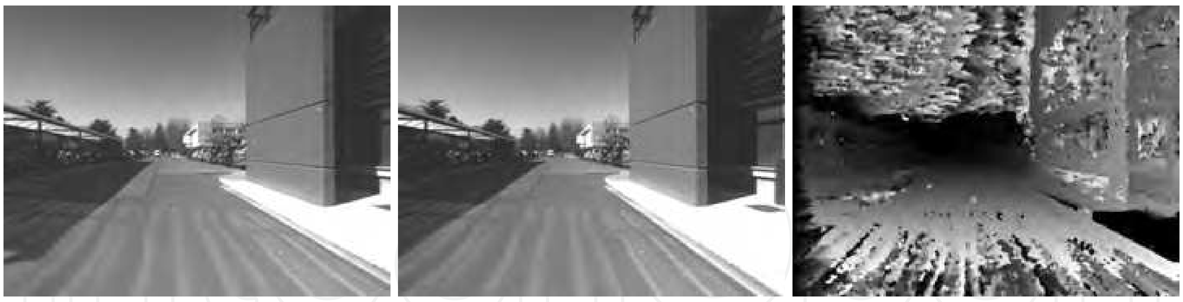
3.1 Disparity map acquired in real time

The final architecture for executing the stereo vision algorithm based on the Census Transform was developed using the level design flow RTL (Ibarra-Manzano, Devy, Boizard, Lacroix & Fourniols, 2009). The architecture was codified in VHDL language using Quartus II workspace and ModelSim. Finally, it was synthesized for an EP2C35F672C6 device contained in the Cyclone IV family of Altera. Table 1 lays out some of the configuration parameters used during the disparity map computation exploited during the dynamic object detection and tracking approach. We would like to highlight that disparity image is computed through the time with a high performance of 30 image per second although detection and tracking approach does not reach this performance. More technical details about the implementation are discussed in section 6.

Parameter	Value
Image size	640 × 480
Window size	7 × 7
Disparity max	64
Performance	40
Latency (μs)	206
Area	6, 977
Memory size	109 Kb

Table 1. Configuration parameters of the Stereo vision architecture.

The architecture was tested for different navigational scenes using a stereo vision bank, first mounted in a mobile robot and then in a vehicle. In the following, we will describe the obtained results for two operational environments. Figure 3 shows the left and right images acquired from the bank stereo and the associated disparity image delivered by the stereo vision algorithm. Dense disparity image depicts the disparity value in gray color levels in figure 3 (c). By examining this last image, we can determine that if the object is close to the stereo vision bank that means a big disparity value, so it corresponds to a light gray level. Otherwise, if the object is far from the stereo vision bank, the disparity value is low, which corresponds to a dark gray level. In this way, we observe that the gray color which represents the road in the resulting images gradually changes from the light to dark gray level. We point out the right side of the image, where we can see the different tones of gray level corresponding to the building. Since large of the building is located at different depths with respect to the stereo vision bank, in the disparity map corresponding gray color value is assigned from lighter to darker gray tones.



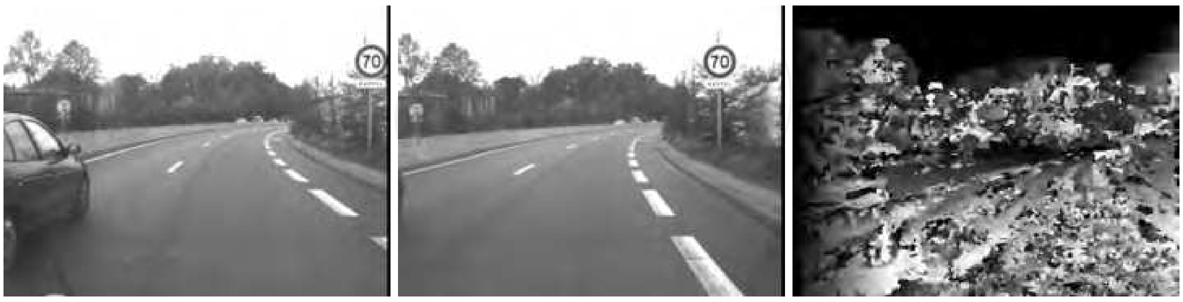
(a) Left image

(b) Right image

(c) Result from FPGA implementation

Fig. 3. Stereo images acquired from a mobile robot during outdoor navigation: a) left image b) right image and c) the disparity map.

In the second test (see figure 4), the stereo vision bank is mounted on a vehicle that is driven on a highway. This experimental test results in a difficult situation because the vehicle is driven at high-speed during the test. Furthermore, this urban condition requires a big distance between both cameras in the stereo vision bank with the aim of augmenting the field of view. A consequence of having a big distance between the two cameras is represented in figure 4 (a) and (b) in which, while the left image show a car that overtakes our vehicle, this car is out of sight in the right camera. Therefore, the dense disparity map shown in figure 4 (c) does not display a cohesive depth value of the vehicle due to mismatches in the information between both images. Nevertheless, we highlight all the different depths represented by the gray color value in the highway that gradually turns darker until the black color which represents an infinity depth.



(a) Left image

(b) Right image

(c) Result from FPGA implementation

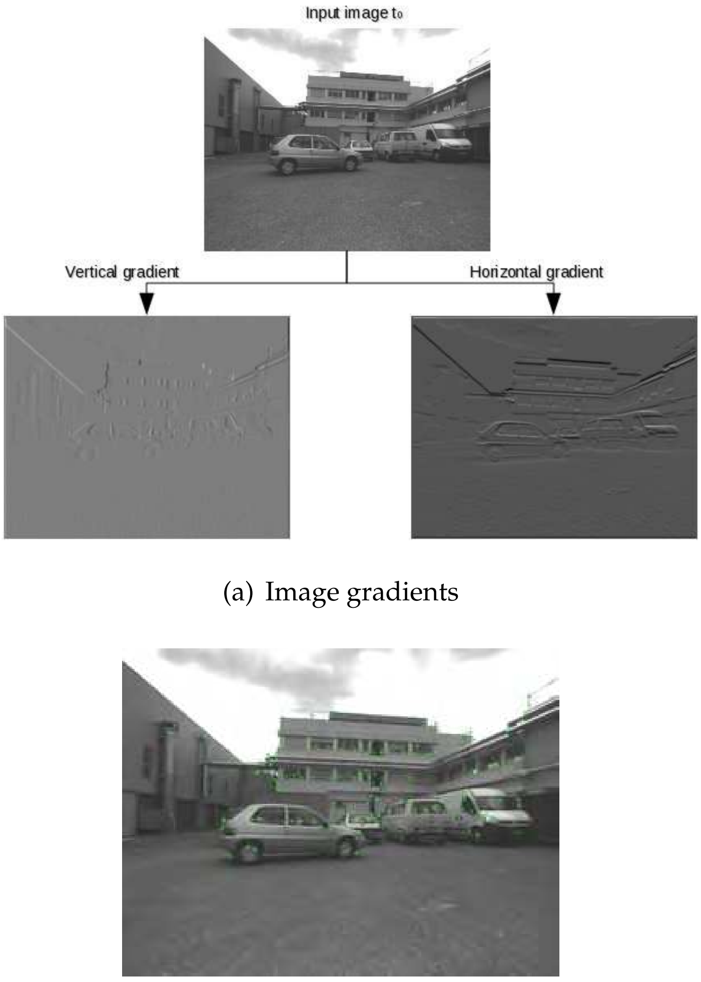
Fig. 4. Stereo images acquired from a vehicle in the highway: a) left image b) right image and c) the disparity map.

4. Moving features perception and grouping

4.1 Active 3D point selection

When dealing with vision-based approaches, the problem of processing a large quantity of information requires that the system resources be sophisticated and expensive if we want to get a real time performance. As we work with left and right images from the stereo vision bank, processing both images yields to more information about the scene but has also more computation requirements. To overcome this problem, we consider to deal with a sparse set of features resulted by the analysis of the left image that represents the most significant and salience feature points in all the image. We use this sparse proposition because it is important to distribute time of processing among some others essential tasks of the strategy, therefore feature selection process must expend minimal time. For the feature point selection, we use the Shi-Tomasi approach (Shi & Tomasi, 1994). This approach consists in providing the most representative points based on image gradient analysis, i.e. corners, borders, and all the regions with high contrast change. Figure 5 depicts both image gradients in horizontal and vertical directions of an input image in gray color level, from which N best features are selected for describing image content. In particular, this input image has high content of information, therefore, a larger number of points have to be selected in accordance with image-density information. Furthermore, the number N is restricted by the time of processing required for reaching real-time performance. According to this, the number of points does not have to exceed $N = 180$. Once 2D's interesting points have been selected on the left image and the disparity map for the stereo-images computed, obtaining the 3D characterization of the points is straightforward. Each 2D point is associated with a corresponding depth value provided by the disparity image to conform the 3D point representation.

Until now, we have obtain a set of 2D features on the left image at time t and their corresponding 3D characterization. For each feature, displacement vectors are computed through the time by using the Kanade-Lucas-Tomasi tracker, referred to as the KLT



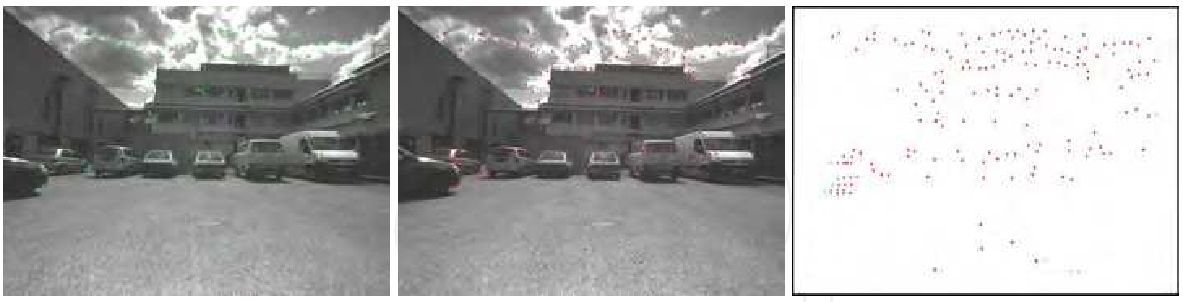
(a) Image gradients

(b) N best features selected

Fig. 5. Image processing for the best interest point selection. a) Gradients in vertical and horizontal direction of the above input image. b) Green points represent the *N* best interesting features resulted by the gradient image analysis.

technique (Shi & Tomasi, 1994), (Lucas & Kanade, 1981). These displacement vectors are used to calculate feature velocities. We are interested in the accumulation of previous position and velocity of the points in order to establish a trail of motion. An example of the accumulation point positions for *N* initial features detected appears in figure 6. Note that, while most of the points are distributed in all the image, a small set of points horizontally aligned can be remarked on the left side of the figure c). These points represent the front part of the vehicle that enters into the field of view of our robot. Once a short number of images have been processed, the accumulated vector displacements and positions of the feature points are evaluated in order to find significant patterns of motion that possibly represent dynamic objects in the scene. A new feature selection task is carried out, as indicated in the green block of figure 1. Further, in the case that any dynamic group of points is found, this information will initialize in the second stage of our strategy, that is the object characterization (right rectangular box in figure 1).

In the following, we explain the a *contrario* method used for clustering position and velocity of the points that possibly describe a mobile object.



(a) First image of the tracking (b) Last image of the tracking (c) Accumulate position through 4 images

Fig. 6. N feature points initially detected at image a) are tracked through 4 images. Image b) displays the last image during the tracking task. Image c) depicts all the accumulate positions for the N initial points detected calculated by the tracking process.

4.2 Clustering of 3D points cloud

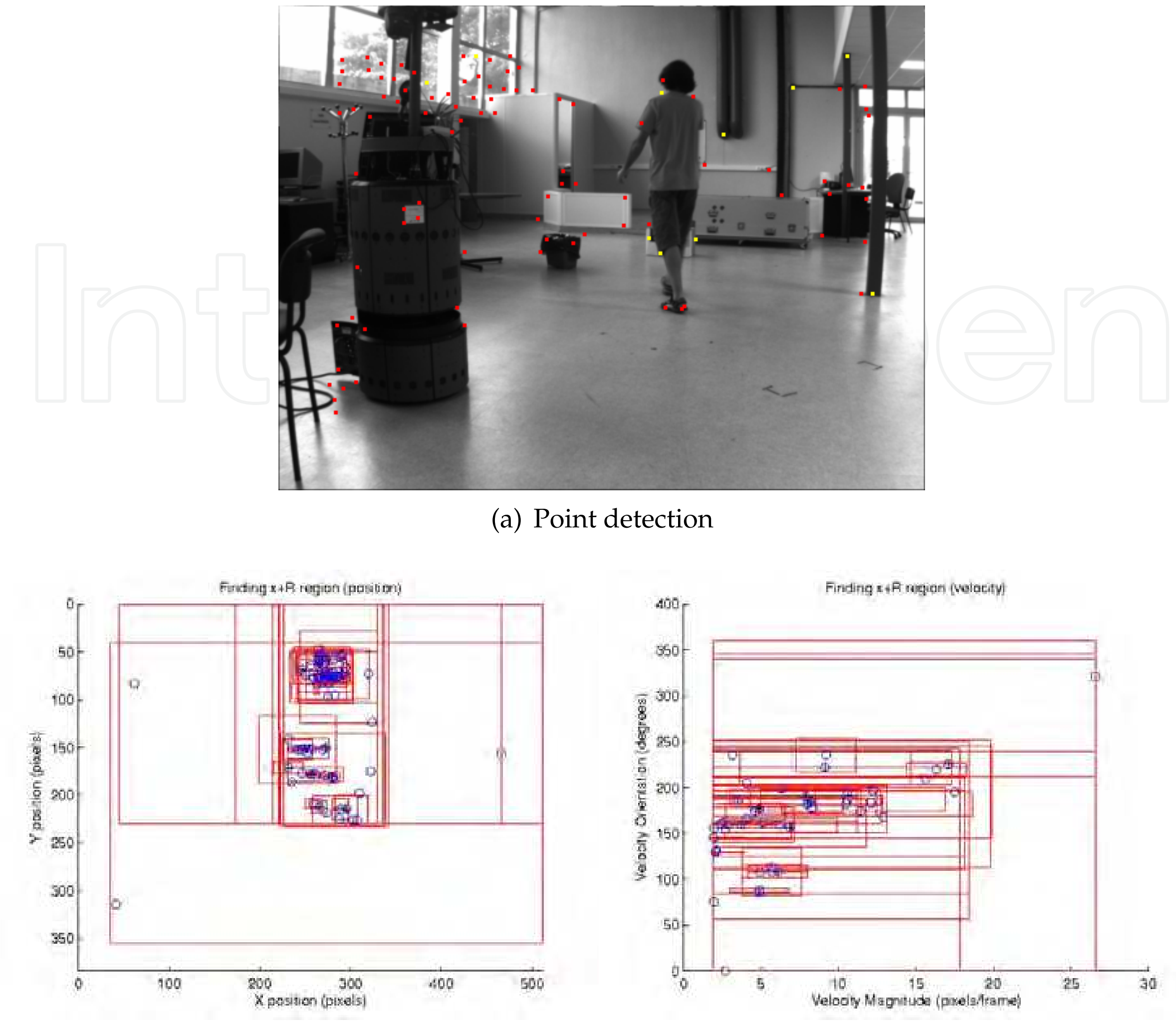
At this point, we have a distributed set of 2D feature points characterized by their position and velocity that have been tracked through a short number of consecutive images. Alternatively, we associated each feature with their corresponding depth position which allows us to manage with a 3D-data set. Following with the global diagram in figure 1, clustering of feature points is the next task to carry out. In order to do that, we use the *a contrario* clustering method proposed in (Veit et al., 2007). This algorithm is based on the Gestalt theory that establishes which groups could be formed based on one or several common characteristics of their elements. In accord to this statement, the *a contrario* clustering technique identifies one group as meaningful if all their elements show a different distribution than an established background random model. Contrary to most clustering techniques, neither initial number of clusters is required nor parameter have to be tuned. These characteristics result very favorable in an unknown environment context where the number of resulted clusters have not been predefined.

In this section we summarize some important concepts of the *a contrario* clustering method, used to group feature points. A detailed description of the method derivation is available in (Desolneux et al., 2008), (Desolneux et al., 2003), (Cao et al., 2007).

4.3 *A contrario* algorithm description

As we mentioned above, a distribution model for the background has to be defined for comparing with the associated distribution of the set of points, here referred to as $V(x, y, v, \theta)$ in R^4. In this work, we use the background model proposed in (Veit et al., 2007) which establishes a random organization of the observations. Therefore, background model elements are independent identically distributed (*iid*) and follow a distribution p. The *iid* nature of random model components proposes an organization with non coherent motion present.

Next, given the input vector $V(x, y, v, \theta)$ from the KLT process (section 4.1), the first objective is to evaluate which elements in V shows a particular distribution contrary to the established distribution p of the background model (that explains "*a contrario*" name). To overcome



(a) Point detection

(b) Position graph

(c) Polar velocity graph

Fig. 7. Best fit region search. An initial set of points is selected and tracked in an indoor environment displayed in image a). The accumulated locations and velocities of the points are analyzed in order to find the region of points with $NFA\,(G) \leq 1$.

the problem of point by point evaluation, $V(x, y, v, \theta)$ is divided in testing groups with different size of elements using a single linkage method. This method constructs a binary tree where each node represents a candidate group G. Once a different group of points have been established, these will be evaluate using a set of given regions represented by $\mathcal{H}$. This set of regions is formed by a different size of hyper-rectangles that will be used to test the distribution of each data group in G. An example of groups distribution evaluation is depicted in figure 7. Each region $H \in \mathcal{H}$ is centered at each element $X \in G$ to find the region H_X that contains all the elements in G and at the same time this region has to minimize the probability of the background model distribution. This procedure requires that sizes of hyper-rectangles be in function of data range, in our experiments we use 20 different sizes by dimension. The measure of meaningfulness (called Number of False Alarms NFA in referenced work) is given by eq 1.

$$NFA\left(G\right) = N^2 \cdot |\mathcal{H}| \min_{\substack{X \in G, \\ H \in \mathcal{H}, \\ G \subset H_X}} B\left(N-1, n-1, p\left(H_X\right)\right) \tag{1}$$

In this equation N represents the number of elements in vector V, $|\mathcal{H}|$ is the cardinality of regions and n is the elements in group test G. The term which appears in the minimum function is the accumulated binomial law, this represents the probability that at least n points including X are inside the region test centered in X (H_X). Distribution p consist of four independent distributions, one for each dimension data. Point positions and velocity orientation follow a uniform distribution because object moving position and direction is arbitrary. On the other hand, velocity magnitude distribution is obtained directly of the empirically histogram of the observed data. So that, joint distribution p will be the product of these four distributions. A group G is said to be meaningful if $NFA\left(G\right) \leq 1$.

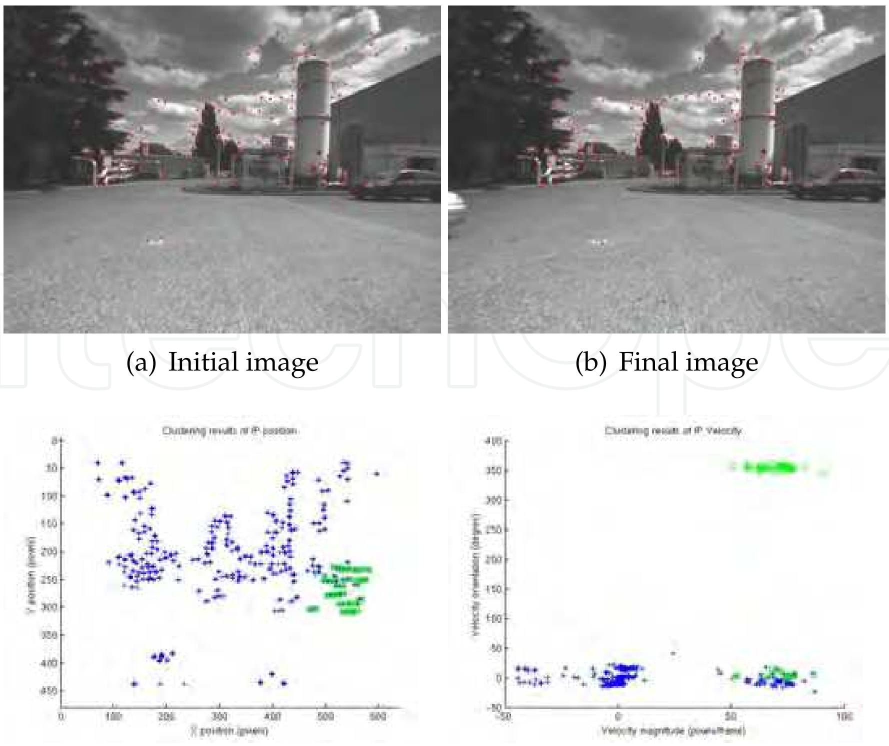
Furthermore two sibling meaningful groups in the binary tree could be belong to the same moving object, then a second evaluation for all the meaningful groups is calculated by Eq. 2.To obtain this new measure, we reuse region group information (dimensions and probability) and just a new region that contains both test groups G_1 and G_2 is calculated. New terms are $N' = N-2$, number of elements in G_1 and G_2, respectively $n'_2 = n_1 - 1$ and $n'_2 = n_2 - 1$, and term $\mathcal{T}$ which represents the accumulated trinomial law.

$$NFA_G\left(G_1, G_2\right) = N^4 \cdot |\mathcal{H}|^2 \, \mathcal{T}\left(N', n'_1, n'_2, p_1, p_2\right) \tag{2}$$

Both mesures 1 and 2 represent the significance of groups in binary tree. Final clusters are found by exploring all the binary tree and comparing to see if it is more significant to have two moving objects G_1 and G_2 or to fusion it in a group G. Mathematically, $NFA(G) < NFA_G(G_1, G_2)$ where $G_1 \cup G_2 \subset G$. A descriptive result is provided in figure 8. Here blue points correspond to those that could be dynamic but without a well defined motion, so they are associated with the background. On the other hand, green points represent the group in G which shows a different distribution than the random background model. Notice that graph c) displays polar velocity considering magnitude in X-axis and orientation in Y-axis, from there, green point positions are not together because they have orientations among 5° and 355° Further, these points correspond to the vehicle entering on the right side of the image b).

4.4 Depth validation

As previously mentioned, the 3D characterization of points is achieved using the depth value described in the disparity map for each two dimensional feature point. From previous section, a set of dynamic points have been obtained using the clustering process, however this process was performed considering uniquely 2D points. Thanks to the disparity map, we know the depth of each feature in dynamic group, so it is possible to perform a second evaluation looking for similarity in their depths. This depth evaluation is computed in a formally analogous way to that of the *a contrario* clustering. However in this case, the number of regions for testing is considerably reduced since the group of points is already detected in a defined region, so it is only need the evaluation of similar depths of the points around that region avoiding the best region search. Additionally, there is not an associated velocity in Z-axis direction which reduces from hyper-rectangle regions to 3D boxes. Namely x and y denote the point location in the image and z denotes the disparity value.



(a) Initial image

(b) Final image

(c) Position points

(d) polar-velocity points

Fig. 8. Clustering results after processing 5 consecutive images. a)First image used for feature selection, b) Last image used for obtaining the trails of the points. A unique dynamic group of points is detected (green points) for whose graph c) depicts their position and graph d) their velocity and orientation.

In the next section, we explain how an irregular contour is initialized using the groups of points calculated by the *a contrario* clustering, therefore used for passing from a dynamic set of points to actual object boundary.

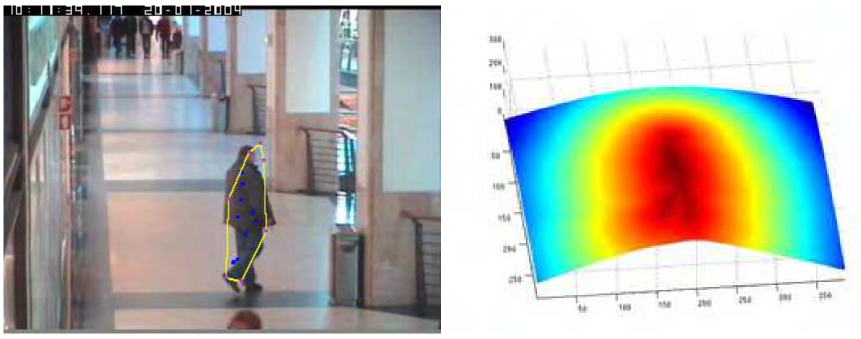
5. Object characterization by an active contour

Active models have been widely used in image processing applications, in particular for recovering shapes and the tracking of moving objects (Li et al., 2010), (Paragios & Deriche, 2005). An active contour or snake is a curve which minimizes energy from restrictive external and internal forces in the image, typically calculated from edges, gradient, among others. Essentially, a snake is not thought of to solve the problem of automatic search for prominent contours of the image, but rather for recovering the contour of a form, from an initial position proposed by other mechanisms. That is to say, if the initial contour is relatively close to the solution (for example a contour defined manually by an operator or obtained through any other method), the contour evolves up until minimizing the function of energy defined from the internal and external forces.

One of the main objectives of this work is to track all mobile objects detected without any prior knowledge about object type that one follows. In a real and dynamic context, we expect to find rigid and non rigid mobile objects ie. vehicles, persons. We have shown in section 4.2 the good performance of the *a contrario* method for finding initial sets of dynamic points which correspond to the mobile objects in the scene. However, whereas clustering process deals with feature points, here we describe a strategy for recovering the deformable shape of the objects through the time by considering others features like intensity image gradient inside of contour. Therefore, through this section we will describe fundamental details of the active contours theory, and the object tracking procedure by means of the active contours.

5.1 Active contour initialization

The results obtained in section 4.2 allow the definition of an initial irregular contour that contains totally or partially our interest object. To this end, we take an outer location of points in the detected group for initializing an active contour which delimits the object on the image. An example is displayed in figure 9 [1]: in image (a) a set of points dynamic (in blue) is detected on which correspond to the person in motion on this sequence. From this set of points we have selected those illustrated in color magenta to describe the object contour depicted in yellow. Due to the fact these points are the most farthest from the center then they are the most representative and closest to the object frontiers. For the results show in image 9(a), we have 8 points on the curve and we use for each 4 control points. The value of 4 control points is fixed for each point on the frontier in order to introduce the corners or discontinuities in the curve (Marin-Hernandez, 2004). Initial contour will allow us to obtain a deformable model by a bounded potential shown in figure 9(b). The zone occupied by the object (represented in red color) is separated from the background. The most intens tones inside of the object are used to best adapt initial contour to the actual object silhouette. Therefore, a more detailed object shape is achieved by analyzing internal and external energy in a bounding box that contains the initial contour.



(a) Initial contour of a non-rigid object

(b) energy functional

Fig. 9. Test performed with real images acquired by a fixed camera in an indoor environment. a)Initial contour derived from points further away from the center b) Energy functional that concentrates internal and external energy considering the initial contour of a).

5.1.1 Parametric active contours.

At this point, it is necessary to represent the initial contour in 9 by a parametric curve $u(\tau) = (x(\tau), y(\tau))$, $\tau \in [0,1]$, with $u(0) = u(1)$. this contour is deformed through the time domain to minimize the energy expressed by:

[1] This image sequence was downloaded from the web site (Fisher, 2011) provided by EC Funded CAVIAR project/IST 2001 37540

$$E_{snake} = \int_0^1 \left[E_{int}(u(\tau)) + E_{ext}(u(\tau)) \right] d\tau \tag{3}$$

where E_{int} is expressed by two main terms, the first one refers to the elasticity and the second the flexibility, given:

$$E_{int} = \alpha \int_0^1 |u_\tau(\tau)|^2 d\tau + \beta \int_0^1 |u_{\tau\tau}(\tau)|^2 d\tau \tag{4}$$

τ and $\tau\tau$ indexes in the term $u(\tau)$ implies respectively first and second order of derivation. By returning to the equation 3 for defining the term E_{ext} or the energy of the image (Sekhar et al., 2008), as the field of potential P:

$$E_{ext} = \int_0^1 P(u(\tau)) d\tau \tag{5}$$

The potential includes different terms defined from image proprieties like edges, lines, etc. Edges energy is obtained by computing the magnitude of the gradient intensity $|\nabla I|$. Without a good initialization, the energy of the edges will not be enough to locate the objects on noisy or low contrast images. Therefore an additional potential of the regions is added to the edge energy. Generally, the potential of the regions is defined by the mean (μ) and the variance (σ^2) of the pixels intensity in the region. However, other constraints could be added like the object velocities, or other statistics derived from region characteristics (Brox et al., 2010). Because of the real-time constraints, we calculate only some statistics in the region that describe the object, such as the main properties to its implementation in correspondence to the next image. The following section describes the proposed strategy about shape recovering and object tracking using active contours.

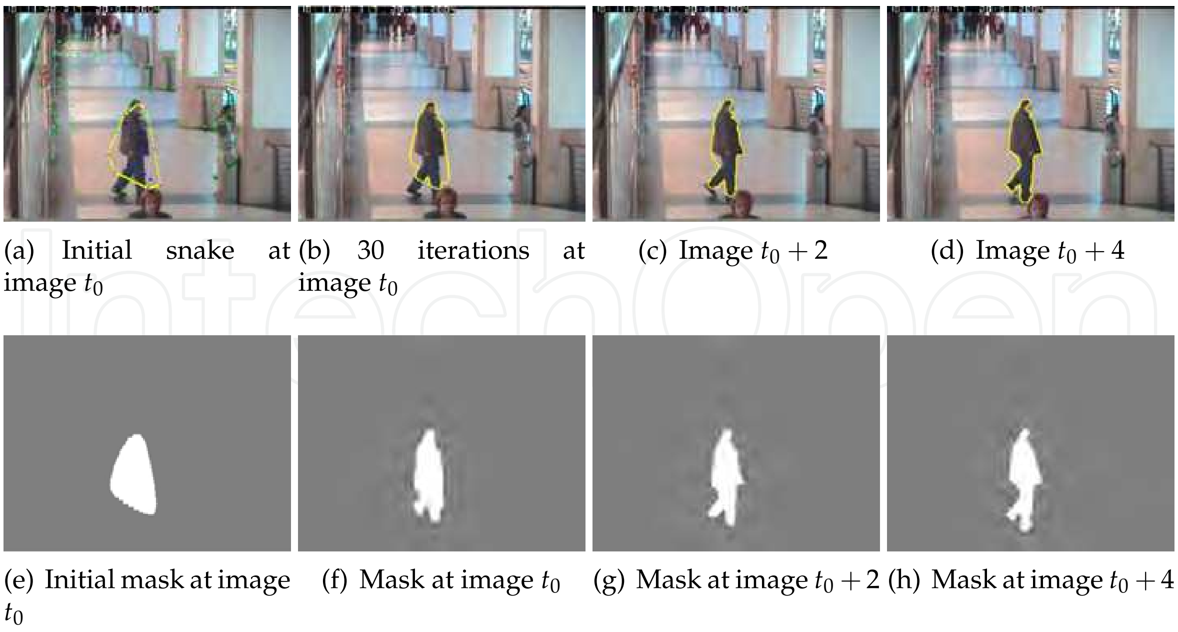
5.2 Incremental silhouette definition

We have tested a method based on the work of Chan and Vese (Chan & Vese, 2001) in order to find the silhouette of the object. In this work, the authors give a region of initialization which may contain total or partially the object. The analysis of this initial contour will allow evaluation of the conditions of the energy minimization inside and outside of the contour. An example of a shape recovering of a person in an indoor environment using the Chan and Vese method is shown in figure 10. First row of images display the contour evolution. In the second row the region inside of the contour for each corresponding above image is illustrated. These images are given as input to the process for recovering real object contour. Furthermore, white regions on these images are labeled as occupied locations which avoid the detection of new interesting points inside of it through the time. According to this, new feature selection (detailed in section 4.1) will look for unoccupied locations in the image allowing the detection of incoming objects.

Once a partial contour at image t in figure 10 has been obtained by minimizing eq. 3, we estimate its position at image $t+1$ for starting a new convergence process. The prediction of the region on the next image is always given by the Kalman filter. The vector state of our object is expressed as:

$$\mathbf{x}_0 = \left[\bar{x}, \bar{y}, \bar{v}_x, \bar{v}_y \right]^T \tag{6}$$

Namely, $\bar{x}$, $\bar{y}$ denote the barycenter location and $\bar{v}_x$, $\bar{v}_y$ the means of velocity vector in X and Y direction respectively. Figure 11 illustrates an example of the filter prediction. The barycenter position and the partial converged contour are located in accordance with Kalman



(a) Initial snake at image t_0 (b) 30 iterations at image t_0 (c) Image $t_0 + 2$ (d) Image $t_0 + 4$

(e) Initial mask at image t_0 (f) Mask at image t_0 (g) Mask at image $t_0 + 2$ (h) Mask at image $t_0 + 4$

Fig. 10. Results of the detection and tracking of a non-rigid dynamic object, in this case a pedestrian on the same sequence of figure 9. Here, we have processed 5 consecutive images, executing 30 iterations per image in order to find the actual object shape.

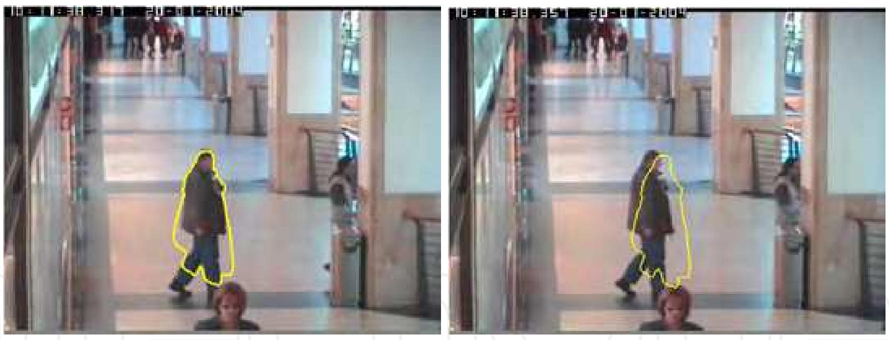
filter prediction. We always consider a velocity constant model and the vector state is obtained from model object displacement. However, in some cases it is first necessary to tune the initial parameters of the filter because there may exist different types of object movements producing an undesired acceleration of the predicted position.

5.3 Object contour validation

The difficulties rising from our proposed strategy points out the sensitivity of the active contours to small discontinuities on the object edges with low-contrast because the energy is configured over the entire image. To overcome these difficulties we evaluate the disparity inside of the region corresponding with the binary mask (that will be presented in the next section figure 15). This evaluation consists in ordering in an ascendant way all the disparity values inside the region designed by the binary mask then we uniquely consider the median of the values. In particular, we consider a valid point of the object region if its disparity value is located up to the 4th percentil in the ordered list (here referred to as $4p$). It follows that our statistical validation for object contour refining can be written as:

$$M = \begin{cases} 1 & depth \geq 4p \\ 0 & otherwise \end{cases} \tag{7}$$

where M represents the binary mask. It is important to remark that the depth value is the inverse value of the disparity therefore in this case the fact of rejecting points located on the first four percentiles represents that these values have a lower disparity value, that is, disparity values of the background points are expecting to be low. This constrain allow us to develop a last validity evaluation for obtaining a most detailed and accurated representation of the object shape. The next section describes our experimental results and presents how the disparity map plays a fundamental role in increasing the efficiency and improve the obtained results.



(a) Object silhouette (b) Estimation for the next image

Fig. 11. Motion estimation using Kalman filter. a) The barycenter location at image *t* is used for predict its position at the next image. b) The irregular contour is centered at the predicted barycenter position, this will be used as initialization region in incoming images.

6. Experimental results

We evaluate the performance of the proposed strategy for detecting and tracking dynamic objects by carrying out experiments using both secure and mounted cameras on autonomous vehicles. Since, using one secure camera, vibrations, egomotion, among other typical problems of mounted cameras are neglected, first we do an experiment under this controlled condition to verify that our algorithm detects moving objects in real images. Moreover, we have observed that the total convergence of energy equation is computationally demanding. To overcome this constraint, it was necessary to propose that convergence task works uniquely a small number of iterations and re-starting the process at next image from previous results. In our case, we see a significant improvement in the efficiency of computation by using this, even when the most similar contour to actual object shape is found after some iterations.

6.1 Fixed camera case

Figure 12 shows 10 consecutive images in which a person appears in the field of view of one security camera on a Commercial Center. These resulting images refer to the case where the number of iterations for converging the active contour at each image is set to 30. The initial contour obtained from the cluster of dynamic points was displayed in figure 9. The effect of the Kalman filter estimation permit us to achieve a more detailed shape than the initial contour frame by frame. We remark that even if initial contour does not contain all the object, the actual shape is achieved after processing 5 images. It is important to point out that resulting object bounds are almost the same as the real thanks to the high-contrast generated between the person and the background.

6.2 Experiments during mobile robot navigation: rigid objects detection

A second experiment was performed in an outdoor environment where the robot accomplishes a linear trajectory of navigation at low-speed (about 6 m/s). In our experiments, we only use the stereo-vision bank mounted on the mobile robot. The robot cameras provide images with resolution of 640 × 480 pixels. The egomotion derived from the robot displacements is neglected by verifying similar depth values inside of the region containing the dynamic group of points. Moreover, dynamic groups detected on the road shows different depth values, so they are rejected to define an initial contour. Figure 13 illustrates the detection and tracking of a white vehicle. Note that this vehicle comes from the imaginary epipolar

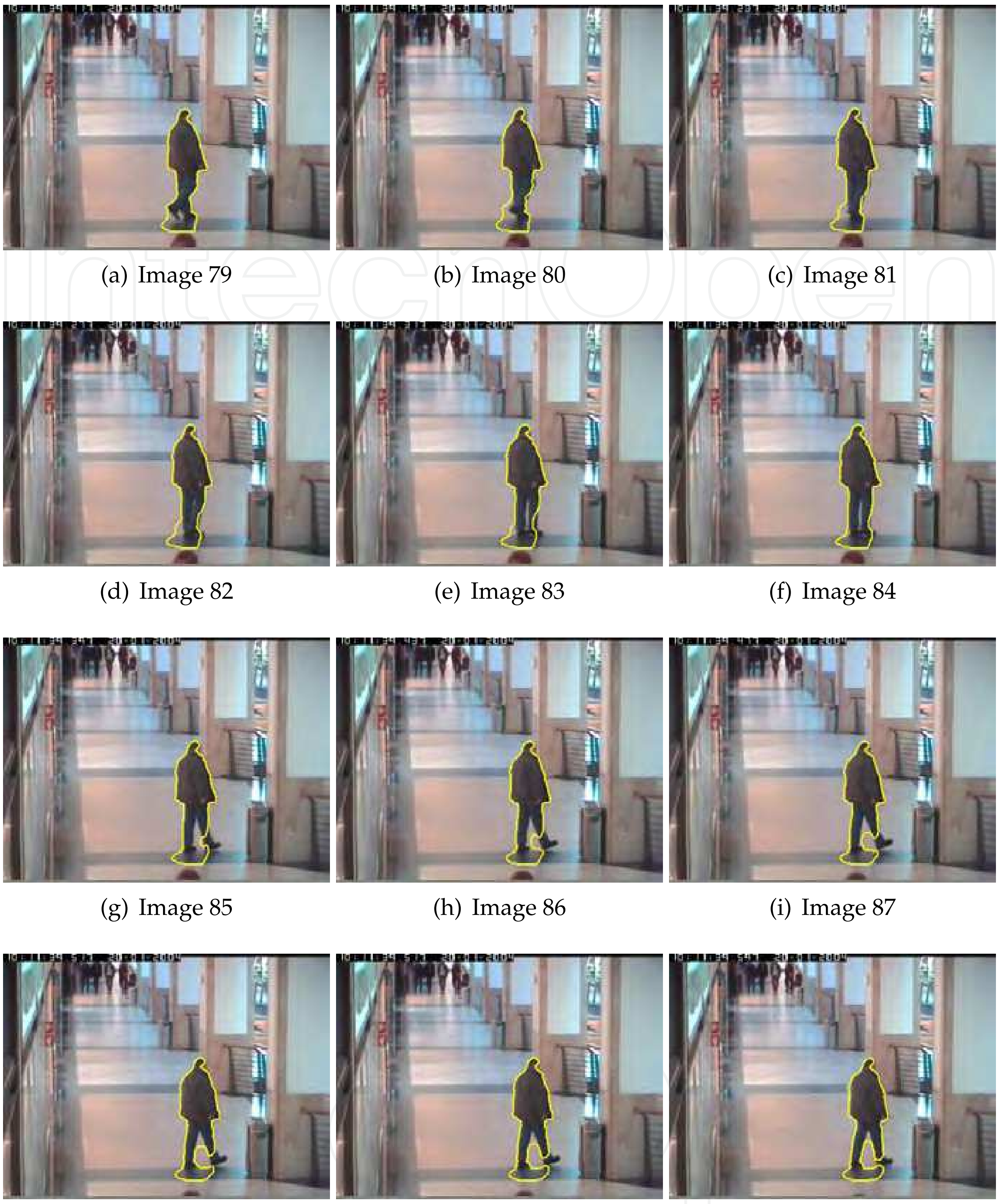
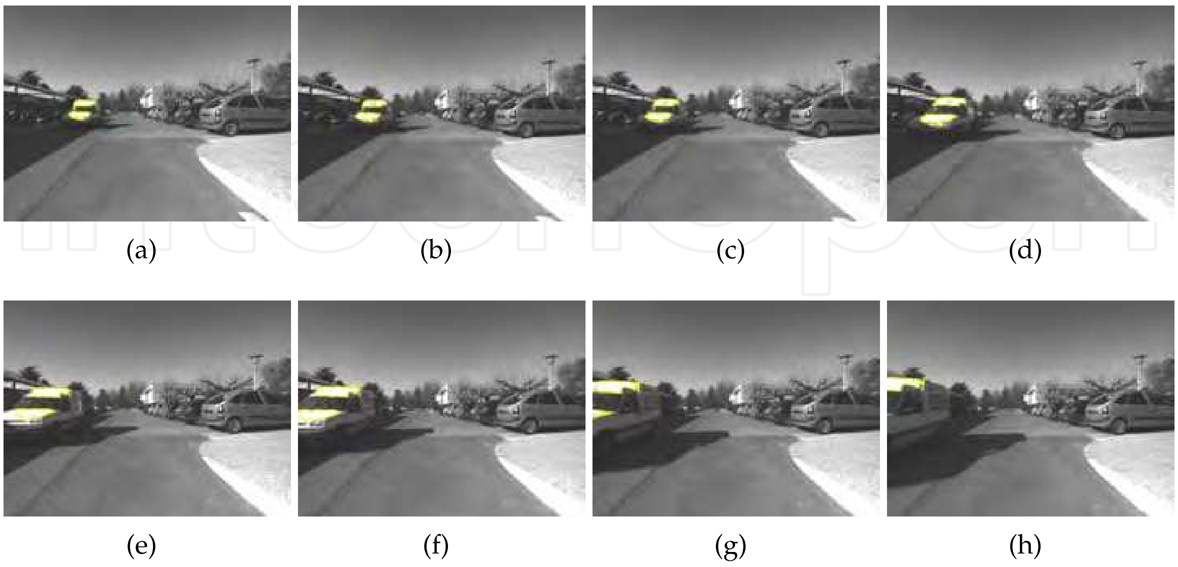

(a) Image 79 (b) Image 80 (c) Image 81
(d) Image 82 (e) Image 83 (f) Image 84
(g) Image 85 (h) Image 86 (i) Image 87
(j) Image 88 (k) Image 89 (l) Image 90

Fig. 12. Detection and tracking of a non-rigid dynamic object along 12 consecutive images. In each image, 30 iterations are used in order to find the object silhouette.

point of image, however it can only be perceived by our strategy at the moment of it is closer to our robot position. Whereas the first experiment was performed under ideal conditions of controlled illumination provided by the indoor environment here this conditions do not hold. This fact notably avoids the computation of a cohesive representation of the energy functional as the one illustrated in figure 9(b). As a consequence, there is only one object at each image of figure 13 but its contour is represented by two separated regions that have similar energy level inside. Furthermore, we analyze the disparity map for obtaining the statistics of disparity values in both regions as mentioned in section 5.3


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

(e) (f) (g) (h)

Fig. 13. Experimental results in an outdoor environment during robot navigation. Images show the detection and tracking of a rigid object.

6.3 Experiments during mobile robot navigation: non rigid objects detection

In the third experiment, we considered a mobile robot moving in an outdoor environment again but in this case the robot finds a dynamic non-rigid object during its trajectory. The left image of figure 14 displays the frame in which the dynamic object was detected and initialized as a irregular contour, middle image shows the disparity map and right image the respective initial mask. In practice detecting non-rigid objects is more complicated than the previous experiment because a person walking has different motions in his legs than his shoulder or his head. Because of improvements, we use in this experiment 40 iterations per image for converging the active contour. Figure 15 illustrates some resulted images of the tracking performed by our proposed strategy. In this experiment, we found that almost all the person could be covered with the active contour. However by examining the left column

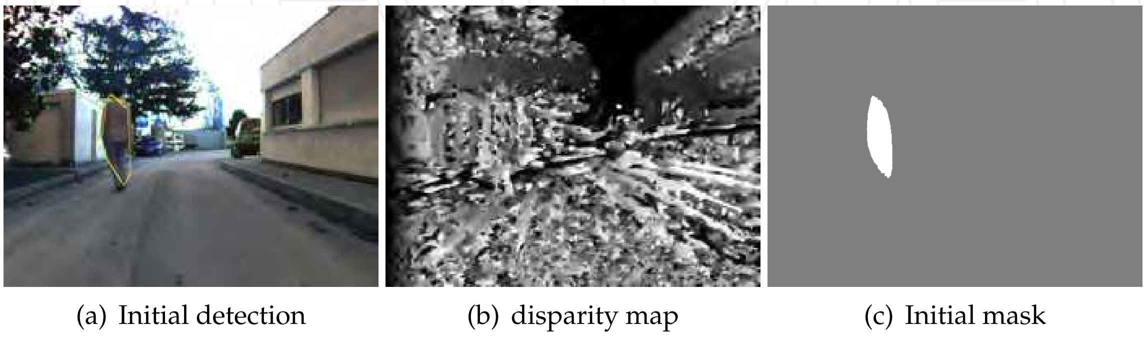

(a) Initial detection (b) disparity map (c) Initial mask

Fig. 14. a) Initial contour derived from the dynamic group of points. b) The corresponding disparity map of the image, c) the initial mask used by the active contour.

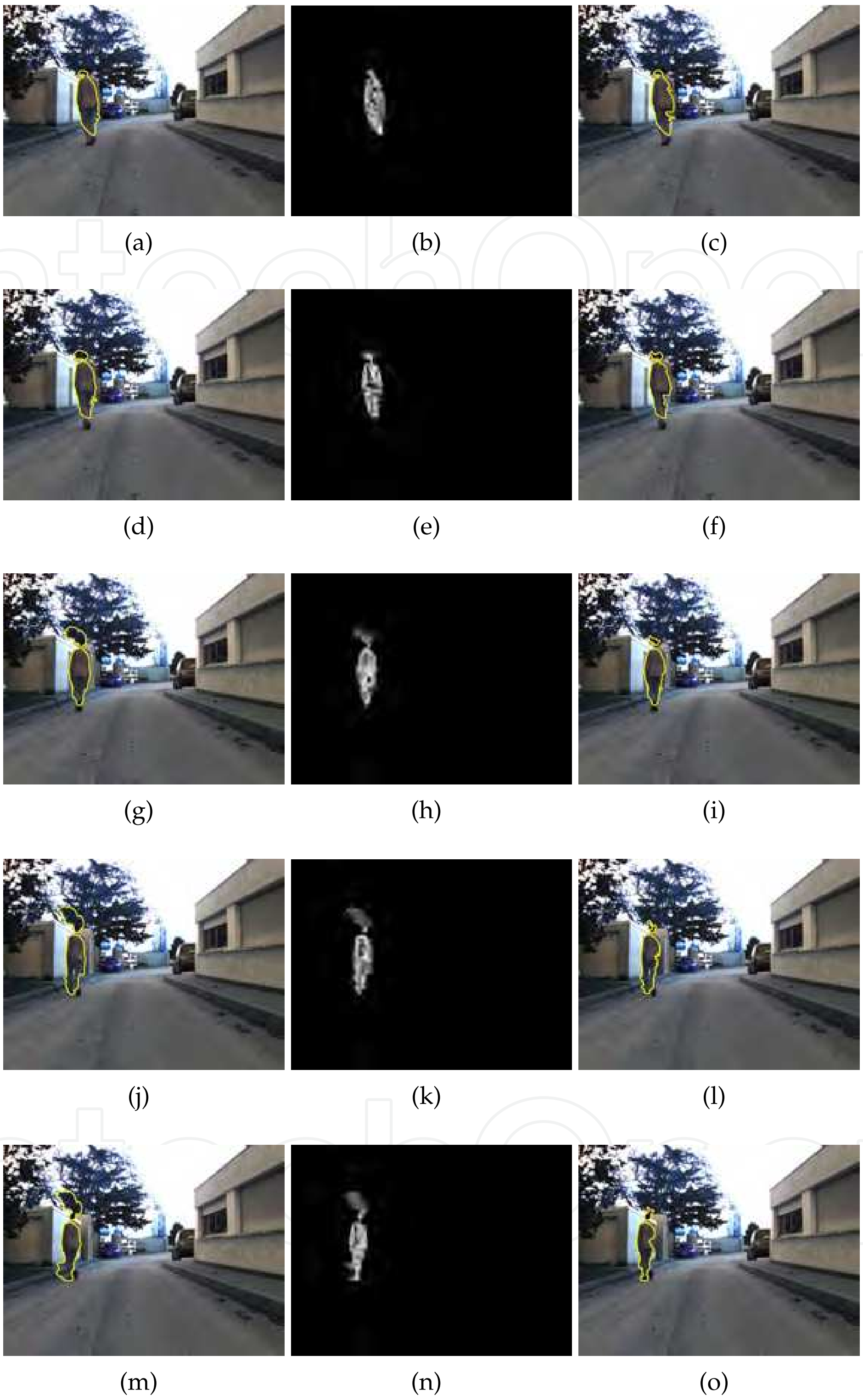


Fig. 15. Images on the left column show the contour obtained by only considering the intensity inside of the region. The middle column isolates the object contour representation and its corresponding disparity values. Images on the right evaluate the depth of the target to boil down extra regions included in the object contour.

of figure 15, we can figure out that part of the background was also included in our object region. To overcome this problem we performed the same statistical calculation of disparity values inside of the region-object as described in section 5.3. By examining the middle column of figure 15, we can figure out that similar depth values are concentrated in actual object contour. We can see that additional zones to this contour that represents the tree and other locations of the background will be rejected using the disparity map since they provide lower values of disparity, that is they are farther away than our interesting moving object.

The proposed algorithm was coded on C/C++ and TCL, however disparity maps are computed for the FPGA by means of an architecture codified in VHDL. The Cyclone IV card calculates 30 disparity maps per second. After several tests, we measure that our algorithm runs around 1 or 1.5 Hz depending of the nature of the environment. From this, note that the disparity is available at higher frequency than our algorithm performance, however we comment that until now the goal of the experiment was to provide an algorithm for detecting and tracking moving objects.

7. Conclusions

In this project, we considered the problem of dynamic object detection from a mobile robot in indoor/outdoor environments of navigation. Specially, the proposed strategy uses only visual-information provided by a stereo-vision bank mounted on the mobile robot. The speed of the robot during navigation is established low to avoid disturbance on the velocity data due to robot ego-motion. Experimental results allow us to realize that the proposed strategy performs a correct and total detection of the rigid and non-rigid objects and it is able to tracking them along the image sequence. Motivated by these results future contributions to this project consist in decreasing the time of computation. Nevertheless, we make some assumptions by avoiding excessive or unnecessary computations ie. the number of selected feature points, number of iterations during the active contour processing, our global algorithm is not able to perform in real time (at least 10 Hz). Significant improvements could be obtained by an emigration of all our algorithm design to embedded architectures like GPU or FPGA devices. Furthermore these kinds of devices provide a high portability towards robotics or autonomous vehicle platforms.

We also comment the difficulties rising from the disparity map constructed by the stereo vision module in which a cohesive and accurate representation of the actual scene have to be improved. To this end, future works consider the addition of a strategy for rejecting "spikes" in the disparity map cased by stereo mismatches.

8. Acknowledgments

This work was partially funded by the CONACyT with the project entitled “Diseno y optimizacion de una arquitectura para la clasificacion de objetos en tiempo real por color y textura basada en FPGA”. Authors would like to thank Cyril Roussillon for help provided with the robot.

9. References

Almanza-Ojeda, D., Devy, M. & Herbulot, A. (2010). Visual-based detection and tracking of dynamic obstacles from a mobile robot, *In Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2010)*, Madeira, Portugal.

Almanza-Ojeda, D. L., Devy, M. & Herbulot, A. (2011). Active method for mobile object detection from an embedded camera, based on a contrario clustering, *in* J. A. Cetto, J.-L. Ferrier & J. Filipe (eds), *Informatics in Control, Automation and Robotics*, Vol. 89 of *Lecture Notes in Electrical Engineering*, Springer Berlin Heidelberg, pp. 267–280. 10.1007/978-3-642-19539-6_18.
URL: *http://dx.doi.org/10.1007/978-3-642-19539-6_18*

Brox, T., Rousson, M., Deriche, R. & Weickert, J. (2010). Colour, texture, and motion in level set based segmentation and tracking, *Image Vision Comput.* 28(3): 376–390.

Cao, F., Delon, J., Desolneux, A., Musé, P. & Sur, F. (2007). A unified framework for detecting groups and application to shape recognition, *Journal of Mathematical Imaging and Vision* 27(2): 91–119.

Chan, T. & Vese, L. (2001). Active contours without edges, *Transactions on Image Processing, IEEE* 10(2): 266–277.

Desolneux, A., Moisan, L. & Morel, J.-M. (2003). A grouping principle and four applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(4): 508–513.

Desolneux, A., Moisan, L. & Morel, J.-M. (2008). *From Gestalt Theory to Image Analysis A Probabilistic Approach*, Vol. 34, Springer Berlin / Heidelberg.

Fisher, R. (2011).
URL: *http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/*

Ibarra-Manzano, M.-A. & Almanza-Ojeda, D.-L. (2011). *Advances in Stereo Vision*, InTech, chapter High-Speed Architecture Based on FPGA for a Stereo-Vision Algorithm, pp. 71-88.

Ibarra-Manzano, M., Almanza-Ojeda, D.-L., Devy, M., Boizard, J.-L. & Fourniols, J.-Y. (2009). Stereo vision algorithm implementation in fpga using census transform for effective resource optimization, *Digital System Design, Architectures, Methods and Tools, 2009. 12th Euromicro Conference on*, pp. 799 –805.

Ibarra-Manzano, M., Devy, M., Boizard, J.-L., Lacroix, P. & Fourniols, J.-Y. (2009). An efficient reconfigurable architecture to implement dense stereo vision algorithm using high-level synthesis, *2009 International Conference on Field Programmable Logic and Applications*, Prague, Czech Republic, pp. 444–447.

Katz, R., Douillard, B., Nieto, J. & Nebot, E. (2008). A self-supervised architecture for moving obstacles classification, *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008*, pp. 155–160.

Klappstein, J., Vaudrey, T., Rabe, C., Wedel, A. & Klette, R. (2008). Moving object segmentation using optical flow and depth information, *PSIVT '09: Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology*, Springer-Verlag, Berlin, Heidelberg, pp. 611–623.

Li, C., Xu, C., Gui, C. & Fox, M. D. (2010). Distance regularized level set evolution and its application to image segmentation, *IEEE Trans. Image Process.* 19(12): 3243–3254.

Lookingbill, A., Lieb, D. & Thrun, S. (2007). *Autonomous Navigation in Dynamic Environments*, Vol. 35 of *Springer Tracts in Advanced Robotics*, Springer Berlin / Heidelberg, pp. 29–44.

Lucas, B. D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision, *Proceedings of DARPA Image Understanding Workshop*, pp. 121–130.

Marin-Hernandez, A. (2004). *Vision dynamique pour la navigation d'un robot mobile.*, PhD thesis, INPT-LAAS-CNRS.

Masrani, D. & MacLean, W. (2006). A Real-Time large disparity range Stereo-System using FPGAs, *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on*, p. 13.

Matthies, L., Litwin, T., Owens, K., Rankin, A., Murphy, K., Coombs, D., Gilsinn, J., Hong, T., Legowik, S., Nashman, M. & Yoshimi, B. (1998). Performance evaluation of ugv obstacle detection with ccd/flir stereo vision and ladar, *ISIC/CIRA/ISAS Joint Conference* pp. 658–670.

Paragios, N. & Deriche, R. (2005). Geodesic active regions and level set methods for motion estimation and tracking, *Computer Vision and Image Understanding* 97(3): 259 – 282.
URL: *http://www.sciencedirect.com/science/article/pii/S1077314204001213*

Schmit, H. H., Cadambi, S., Moe, M. & Goldstein, S. C. (2000). Pipeline reconfigurable fpgas, *Journal of VLSI Signal Processing Systems* 24(2-3): 129–146.

Sekhar, S. C., Aguet, F., Romain, S., Thévenaz, P. & Unser, M. (2008). Parametric b-spline snakes on distance maps—application to segmentation of histology images, *Proceedings of the 16th European Signal Processing Conference, (EUSIPCO2008)* .

Shi, J. & Tomasi, C. (1994). Good features to track, *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.

Sola, J., Monin, A. & Devy, M. (2007). BiCamSLAM: two times mono is more than stereo, *IEEE International Conference on Robotics Automation (ICRA2007), Rome, Italy*, pp. 4795–4800.

Talukder, A. & Matthies, L. (2004). Real-time detection of moving objects from moving vehicles using dense stereo and optical flow, *proceedings of the International Conference on Intelligent Robots and Systems (IROS2004)*, pp. 3718–3725.

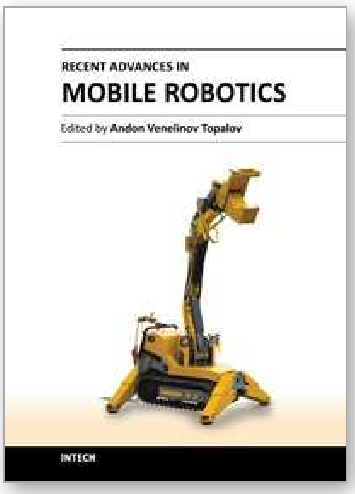
Veit, T., Cao, F. & Bouthemy, P. (2007). Space-time a contrario clustering for detecting coherent motion, *IEEE International Conference on Robotics and Automation, (ICRA07)*, Roma, Italy, pp. 33–39.

Vu, T. & Aycard, O. (2009). Laser-based detection and tracking moving objects using data-driven markov chain monte carlo, *IEEE International Conference on Robotics Automation (ICRA2009), Kobe, Japan*.

Williamson, T. (1998). *A High-Performance Stereo Vision System for Obstacle Detection*, PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Woodfill, J., Gordon, G., Jurasek, D., Brown, T. & Buck, R. (2006). The tyzx DeepSea g2 vision system, ATaskable, embedded stereo camera, *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, p. 126.

Zabih, R. & Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence, *ECCV '94: Proceedings of the Third European Conference on Computer Vision*, Vol. II, Springer-Verlag New York, Inc., Secaucus, NJ, USA, pp. 151–158.



Recent Advances in Mobile Robotics
Edited by Dr. Andon Topalov

ISBN 978-953-307-909-7
Hard cover, 452 pages
Publisher InTech
Published online 14, December, 2011
Published in print edition December, 2011

Mobile robots are the focus of a great deal of current research in robotics. Mobile robotics is a young, multidisciplinary field involving knowledge from many areas, including electrical, electronic and mechanical engineering, computer, cognitive and social sciences. Being engaged in the design of automated systems, it lies at the intersection of artificial intelligence, computational vision, and robotics. Thanks to the numerous researchers sharing their goals, visions and results within the community, mobile robotics is becoming a very rich and stimulating area. The book Recent Advances in Mobile Robotics addresses the topic by integrating contributions from many researchers around the globe. It emphasizes the computational methods of programming mobile robots, rather than the methods of constructing the hardware. Its content reflects different complementary aspects of theory and practice, which have recently taken place. We believe that it will serve as a valuable handbook to those who work in research and development of mobile robots.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821