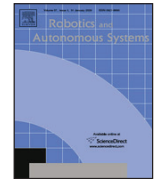




Contents lists available at SciVerse ScienceDirect

## Robotics and Autonomous Systems

journal homepage: [www.elsevier.com/locate/robot](http://www.elsevier.com/locate/robot)Robust camera pose and scene structure analysis for service robotics<sup>☆</sup>Sorin M. Grigorescu<sup>\*</sup>, Gigel Macesanu, Tiberiu T. Cocias, Dan Puiu, Florin Moldoveanu

Department of Automation, Transilvania University of Brasov, Mihai Viteazu 5, 500174, Brasov, Romania

## ARTICLE INFO

## Article history:

Received 24 March 2011

Received in revised form

28 June 2011

Accepted 4 July 2011

Available online 27 July 2011

## Keywords:

Robot vision systems

Feedback control

Stereo vision

Robustness

3D Reconstruction

## ABSTRACT

Successful path planning and object manipulation in service robotics applications rely both on a good estimation of the robot's *position and orientation* (pose) in the environment, as well as on a reliable understanding of the visualized scene. In this paper a robust real-time camera pose and a scene structure estimation system is proposed. First, the pose of the camera is estimated through the analysis of the so-called *tracks*. The tracks include key features from the imaged scene and geometric constraints which are used to solve the pose estimation problem. Second, based on the calculated pose of the camera, i.e. robot, the scene is analyzed via a robust depth segmentation and object classification approach. In order to reliably segment the object's depth, a feedback control technique at an image processing level has been used with the purpose of improving the robustness of the robotic vision system with respect to external influences, such as cluttered scenes and variable illumination conditions. The control strategy detailed in this paper is based on the traditional open-loop mathematical model of the depth estimation process. In order to control a robotic system, the obtained visual information is classified into objects of interest and obstacles. The proposed scene analysis architecture is evaluated through experimental results within a robotic collision avoidance system.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Integrating visual perceptual capabilities into the control architecture of a robot is not a trivial task, especially for the case of service robots which have to work in unstructured environments with variable illumination conditions [1]. As a result of progress in the research on robot vision and technology development, the use of vision as a primary perception sensor for providing information for controlling autonomous systems, such as mobile robots and redundant manipulators, has grown significantly in recent years [1,2]. A crucial requirement for a robot vision system is the achievement of a human-like robustness against the complexity of the robot's environment in order to provide reliable visual information for autonomous functioning. A robot vision system is used to robustly analyze images acquired from complex scenes where objects to be recognized are surrounded by a variety of other objects. As well as being robust against cluttered scenes, a robot vision system has to be robust against unpredictability in the appearance of objects due to different external influences such as variable illumination. The main requirement for such a visual system is to reliably map the

imaged scene into a virtual 3D environment that can be used to infer the future and immediate actions of the robot.

The most common approach to 3D perception, or depth sensation, is through stereo vision. Basically, stereo vision exploits the geometry between two perspective cameras imaging a scene. By analyzing the perspective views between the acquired images, 3D visual information can be extracted. Traditionally, stereo vision is implemented using a pair of calibrated cameras with a known baseline between their optical points. Having in mind that the geometrical relations between the two cameras are known, by calculating the relative perspective projection of object points in both images, their 3D world coordinates can be reconstructed until a certain accuracy [3]. The problem of stereo camera *position and orientation* (pose) estimation is illustrated in Fig. 1. As a camera  $C$  moves through the Euclidean space, the distances to the imaged objects, as well as the translation  $t_i$  and rotation  $R_i$  of  $C_i$  with respect to its previous poses should be determined from a set of observed feature points  $P_j$ . The pose of the robotic system is inherently obtained once the pose of the camera has been calculated.

## 1.1. Related work

In this paper, the authors propose a feedback control approach of a depth estimation system, aiming at compensating the problem of using constant image processing parameters in complex environments. The objectives of the proposed architecture are to reliably extract the objects of interest together with the camera–objects distances, as well as the pose of the camera while

<sup>☆</sup> The source code for the proposed scene perception system can be found at the SVN repository <http://rovis.unitbv.ro/rovis>.

<sup>\*</sup> Corresponding author. Tel.: +40 268 418 836; fax: +40 268 418 836.

E-mail addresses: [s.grigorescu@unitbv.ro](mailto:s.grigorescu@unitbv.ro) (S.M. Grigorescu), [gigel.macesanu@unitbv.ro](mailto:gigel.macesanu@unitbv.ro) (G. Macesanu), [tiberiu.cocias@unitbv.ro](mailto:tiberiu.cocias@unitbv.ro) (T.T. Cocias), [puiudan@unitbv.ro](mailto:puiudan@unitbv.ro) (D. Puiu), [moldof@unitbv.ro](mailto:moldof@unitbv.ro) (F. Moldoveanu).

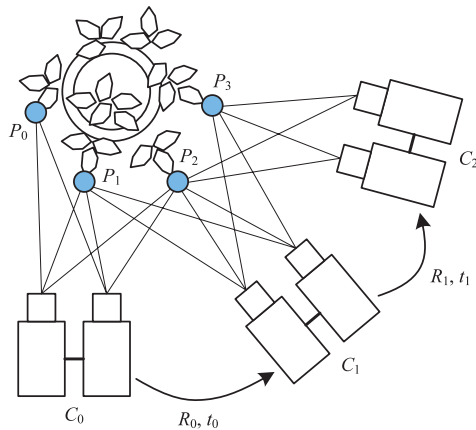


Fig. 1. The camera pose (i.e. robot pose) and scene reconstruction problem.

it moves through space. When feedback control techniques are discussed in connection to robot vision, they are usually put in the context of controlling a certain system using visual information. Such devices are typically named *Active Vision* [4] or *Visual Servoing Systems* [5]. There are relatively few publications dealing with control techniques applied directly on the image processing chain.

The idea of feedback image processing has been tackled previously in the computer vision community in papers such as [6] or [7]. One of the first comprehensive papers on the usage of feedback information at image processing level can be found in [8], where reinforcement learning was used as a way to map input images to corresponding optimal segmentation parameters. In [6], a hypothesis generation and verification method was developed in order to calculate interest operators which can be used to locate target objects, such as bridges, in noisy data. Also, in [7], a feedback strategy was employed in the self-adaptation of a learning-based object recognition system that has to perform in variable illumination conditions. In [9], dynamic closed-loop systems were used to automatically adapt camera parameters at the image acquisition stage. In the area of stereo vision, probabilistic methods for the robust analysis of depth estimation were adopted in [10].

Although the mentioned literature is focused on closed-loop processing, it does not provide a suitable control framework from both the image, as well as from the control point of view. Techniques for image processing inspired from control engineering were used in [11] for adapting a character recognition system, as well as for a quality control one. In the field of robot vision, the authors successfully used feedback control concepts to tune region [12] and boundary [13] based segmentation operations in order to improve the visual perceptual capabilities of a service robot [14]. In this paper, feedback machine vision is further investigated by proposing a closed-loop model of depth sensing based on the extremum seeking control paradigm set forth in [15]. Its use is demonstrated in the vision system proposed in this paper.

Camera motion estimation, or *egomotion*, has been studied within the *Simultaneous Localization and Mapping* (SLAM) context [16]. Using detected visual information, motion estimation techniques can provide a very precise egomotion of the robot. In SLAM also, the basic sensor used is the stereo camera [17,18]. The main operation involved in stereo motion estimation is the computation of so-called correspondence points used for calculating the 3D pose of the robot's camera. Based on the extracted features, the robot's motion can be calculated with the help of estimators such as the *Kalman* [19] or *Particle Filter* [20]. Although for the visual control of a robot the estimation of its motion is crucial, it is not treated in this paper since the objective here is to obtain a robust visual perception of the imaged environment. Nevertheless, in order

to determine the positions and orientations of the visualized objects with respect to the robot, its pose has been obtained via correspondence points matching, thus neglecting dynamic measures such as the robot's velocities and accelerations. Also, the robot's pose is needed for fusing the obtained disparity maps in order to construct a virtual 3D model of the scene.

## 1.2. Structure and main contributions

The main contributions of the presented paper may be summarized as follows:

1. improvement of the depth estimation process through the inclusion of a feedback control technique at the depth map computation level;
2. fusion of closed-loop depth computation with camera pose estimation.

This paper is organized as follows. In Section 2, the proposed theory behind feedback modeling of image processing systems is presented, followed in Section 3 by a description of the visual architecture and information flow within the vision system. In Section 4, the estimation of the imaged scene geometry using the novel closed-loop depth sensing algorithm is detailed, along with the fusion of the calculated depth maps. In Section 5, the recognition of objects of interest and obstacles, based on the obtained depth information and 2D feature extraction, is presented. Finally, before conclusions and outlook, performance evaluation is given through experimental results.

## 2. Feedback control in image processing

In a robotic application, the purpose of the image processing system is to understand the surrounding environment of the robot through visual information. Usually, an object recognition and 3D reconstruction chain for robot vision consists of *low* and *high* levels of processing operations. Low level image processing deals with pixel wise operations aiming to improve the input images and also separate objects of interest from background. Both the inputs and outputs of the low level processing blocks are images. The second type of modules, which deal with high level visual information, are connected to low level operations through a feature extraction component which converts the input images to abstract data describing the imaged objects. The importance of the quality of results coming from low level stages is related to the requirements of high level image processing. Namely, in order to obtain a proper 3D virtual reconstruction of the imaged environment at a high level stage, the inputs coming from low level have to be reliable.

Traditionally, vision systems are open-loop sequential operations, which function with constant predefined parameters and have no interconnections between them. This approach has impact on the final 3D reconstruction result, since each operation in the chain is applied sequentially, with no information between the different levels of processing. In other words, low level image processing is performed regardless of the requirements of high level processing. In such a system, for example, if the segmentation module fails to provide a good output, all the subsequent steps will fail.

The basic diagram from which feedback mechanisms for machine vision are derived can be seen in Fig. 2. In such a control system, the control signal  $u$ , or *actuator variable*, is a parameter of an image processing operation, whereas the *controlled*, or *state*, variable  $y$  is a measure of processing quality.

The design and implementation of feedback structures in machine vision is significantly different from conventional industrial control applications, especially in the selection of the pair *actuator variable–controlled/state variable*. The choice of this pair has to be

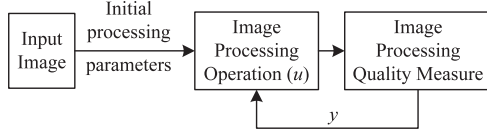


Fig. 2. Feedback control of an image processing operation.

appropriate from the control, as well as from the image processing point of view.

In order to derive a control strategy for a machine vision system, the following discrete nonlinear state-space representation model of the vision apparatus is suggested:

$$\begin{cases} \dot{\mathbf{x}}(k) = f[\mathbf{x}(k), \mathbf{u}(k)], \\ \mathbf{y}(k) = g[\mathbf{x}(k)], \end{cases} \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the state vector,  $\mathbf{u} \in \mathbb{R}$  is the actuator (input),  $\mathbf{y} \in \mathbb{R}$  is the output vector,  $f: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$  is the state transition function and  $g: \mathbb{R}^n \rightarrow \mathbb{R}$  is the output function.  $k$  represents the discrete time. Suppose that we have a control law:

$$\mathbf{u}(k) = \alpha[\mathbf{x}(k), \theta], \quad (2)$$

the control problem is to find the optimal parameter  $\theta^*$  which provides an output of desired, or reference, quality. Following the above reasoning, the closed-loop system:

$$\dot{\mathbf{x}} = f[\mathbf{x}, \alpha(\mathbf{x}, \theta)] \quad (3)$$

has its equilibrium point parameterized by  $\theta$ . Having in mind the high non-linearity of an image processing system, a control strategy based on extremum seeking [15] is suggested. Thus, the goal of the feedback control system is to determine the optimal parameter  $\theta^*$  as the minimum, or maximum, value of the state vector  $\mathbf{x}$ :

$$\theta^* = \arg \min \mathbf{x}(k) \quad \text{or} \quad \theta^* = \arg \max \mathbf{x}(k). \quad (4)$$

The choice of this particular type of control method lies in the fact that, taking into account the non-linearity of an image processing system, it is difficult to determine reference values that could be applied to classical feedback structures. Hence, in the image processing control approach, the desired state of a vision system is given by the extremal values of the state vector. In the following, the proposed model is applied to the depth estimation processed detailed in the next section.

### 3. Visual architecture overview

The objective of the proposed visual understanding system is to robustly estimate, in real-time, the structure of the imaged scene with respect to the pose of the camera in space. Having in mind the large amount of information that has to be processed, the first step in the development of the scene understanding system

is to model the flow of information into a visual architecture. In Fig. 3, the block diagram of the visual understanding system can be seen. Basically, the overall architecture has been divided into two main components, that is, the *scene geometry estimation* and the *scene understanding* modules, both of them explained in the next sections.

The goal of the *scene geometry estimation* component is to determine the pose of the camera while it moves. As shown in Fig. 3, this procedure is performed using information extracted from stereo images and grouped into so-called *tracks*. A track represents the visual data calculated from a pair of stereo images, as well as the camera's a-priori known geometry (e.g. baseline between the two optical sensors) and internal parameters (e.g. focal length, optical center, etc.). In the presented system, a track contains the following information:

- input stereo images  $I_L(x, y)$  and  $I_R(x, y)$ ;
- set of 2D correspondence points  $p_j$  between  $I_L(x, y)$  and  $I_R(x, y)$ ;
- set of 3D correspondence points  $P_j$  calculated from  $p_j$ ;
- stereo camera pose  $C$ ;
- depth map  $I_d(x, y)$ ;
- set of objects of interest  $O_{int}$ ;
- set of obstacles  $O_o$ .

In robotic pose estimation, or more particularly camera pose estimation, the main variables that have to be determined are the 3D positions of the corresponding points  $P_j$  and the position and orientation of the camera  $C_i$ .

$$P_j = [x_j \quad y_j \quad z_j]^T, \quad (5)$$

$$C_i = [x_i \quad y_i \quad z_i \quad \phi_i \quad \psi_i \quad \theta_i]^T, \quad (6)$$

where points  $P_j$  are matched 3D points in the left image of two consecutive stereo images and  $T$  represents the transpose.

Once the 3D positions of the correspondence feature points have been determined in two adjacent stereo images, the camera pose can be reconstructed using triangulation, as will be explained in the next section. In real world applications the computation of 3D feature points from their 2D correspondence is subjected to measurement noise, thus the obtained camera pose may contain errors with respect to its position and orientation. To cope with these errors, a *sparse bundle adjustment* technique was adopted [21]. The goal of bundle adjustment is to recalculate the pose of the camera based on a minimization algorithm which takes into account the 3D correspondence points and their 2D reprojection error over a sequence of tracks. After the camera's pose has been determined, the relative distances to the objects and obstacles can be calculated through computation of so-called *depth maps* [22]. This step is a crucial one in the visual system, since its result influences the behavior of the system with respect to the scene geometry. The depth map calculation process, together with its closed-loop enhancement, will be described in Section 4.

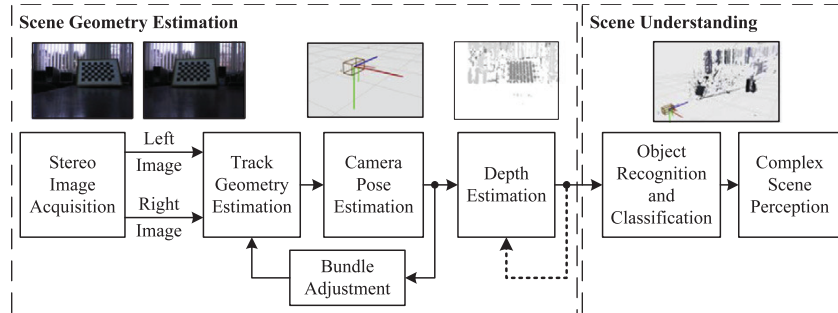


Fig. 3. Block diagram of the proposed 3D visual perception system.

In Fig. 3, the feedback control improvement of the depth sensing component is represented by the feedback loop illustrated with dashed line.

Having obtained the pose of the camera and the raw camera–objects/obstacles distances, the imaged scene can be analyzed with the purpose of detecting the objects of interest and obstacles relative to the pose of the camera, or robot. This is accomplished by the second main module of the proposed vision system, that is the *scene understanding* one. The construction of the module is relatively straightforward, namely, after an object recognition and classification procedure, the semantics of the environment are determined based on the context of the scene, as described in Section 5.

#### 4. Scene geometry estimation

One important problem to be solved in robot localization and scene understanding is the estimation of the camera's pose in the Cartesian space, along with the geometry of the imaged environment, that is, the camera–objects distances. In order to solve these problems, two types of depth calculation approached have been developed. The first one aims at determining correspondence points between two consecutive stereo images pairs with the purpose of obtaining the relative poses of the cameras with respect to each stereo pair. The second depth calculation method has as goal the estimation of the scene structure with respect to the already calculated poses of the camera. The two approaches have been named *sparse* and *dense* depth estimation, respectively.

##### 4.1. Camera and measurement models

The model of the stereo camera used in sensing the robot's environment is illustrated in Fig. 4. A real world point represented in homogeneous coordinates  $P = [X \ Y \ Z \ 1]^T$  is projected onto the image planes of a stereo camera as the homogeneous 2D image points:

$$\begin{cases} p_L = [x_L \ y_L \ 1]^T, \\ p_R = [x_R \ y_R \ 1]^T, \end{cases} \quad (7)$$

where  $p_L$  and  $p_R$  have the 2D coordinates  $(x_L, y_L)$  and  $(x_R, y_R)$  projected onto the left  $I_L$  and right  $I_R$  images, respectively. The  $p_L$  and  $p_R$  2D image positions are given by the intersection with the image plane of the line connecting point  $P$  in world coordinates with the optical centers  $O_L$  and  $O_R$  of both cameras, as shown in Fig. 4. The image, or *principal plane*, is located at a distance  $f$  from the optical center of a camera.  $f$  is commonly known as the *focal length*. The  $z$  axis of the coordinate system attached to the optical center is referred to as the *principal ray*, or *optical axis*. The principal ray intersects the image plane at image center  $(c_x, c_y)$ , also known as the *principal point*. The origin of the image coordinate system is defined as the image top-left corner  $(x_0, y_0)$ .

Knowing  $p_L, p_R$  and the distance  $T$  between the optical centers of the two cameras, the distance, or depth,  $Z$  from the stereo camera to point  $P$  can be calculated, thus obtaining the 3D position of  $P$  with respect to the camera. Having in mind the perspective projection of  $P$  onto the image planes, given by  $p_L$  and  $p_R$ , the 3D position of  $P$  is determined using the next three formulas:

$$X = x_L \cdot \frac{T}{d}, \quad (8)$$

$$Y = y_L \cdot \frac{T}{d}, \quad (9)$$

$$Z = f \cdot \frac{T}{d}, \quad (10)$$

where  $d$  is the disparity of the projected point  $P$ :

$$d = x_L - x_R. \quad (11)$$

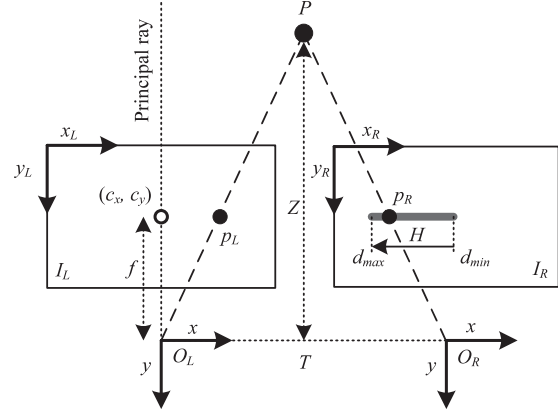


Fig. 4. Principle of depth estimation of a point  $P$  on a pair of rectified and undistorted stereo images.

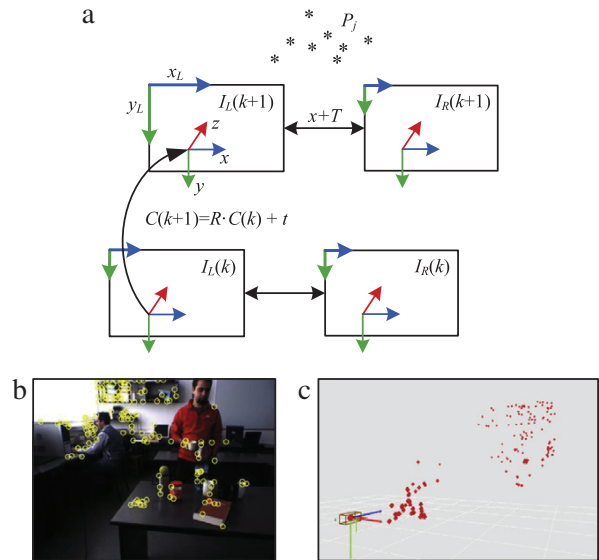


Fig. 5. Camera pose estimation through correspondence calculation. (a) Consecutive pose estimation principle (b) 2D feature points calculation. (c) 3D feature points and camera pose estimation.

From Eq. (10) it can be observed that the distance is inversely proportional to the disparity. Since we have considered rectified images as inputs, that is, images with parallel rows, the disparity  $d$  is given only by the difference between the point coordinates on the  $x$  image axis.

##### 4.2. Sparse feature points depth estimation

The goal of 3D depth estimation for feature points in consecutive stereo images is to obtain a relationship between the poses of the stereo camera. This relation is established based on the principle illustrated in Fig. 5(a). Namely, the new camera pose  $C(k+1)$  is determined with respect to the previous one  $C(k)$  by evaluating the correspondence points between the left and right stereo images and between the left images corresponding to  $C(k)$  and  $C(k+1)$ . The pose of the right image sensor differs from the left one only along the  $x$  position of the Cartesian space. This difference is represented by the baseline  $T$  between the two sensors of the stereo camera.

First, for camera pose estimation, the 2D correspondence points  $p$  between the left and right stereo images are calculated. 2D feature points have been extracted via the Harris corner detector [23], followed by a correspondence matching using a traditional cross-correlation similarity measure [24]. Their corresponding 3D positions are obtained using Eqs. (8)–(10). Second, a matching is performed between the 2D feature points in consecutive stereo images, that is between images acquired under camera poses  $C(k)$  and  $C(k + 1)$ . As convention, these matches are calculated for the left camera only. Knowing the 3D positions of the 2D points matched between adjacent images, the pose of the camera can be calculated through a *Perspective-N-Point* (PNP) algorithm [3]. By solving the PNP problem, the rotation  $R$  and translation  $t$  matrices that relate the camera's poses are obtained:

$$C(k + 1) = R \cdot C(k) + t. \quad (12)$$

In order to solve the PNP problem, a minimum number of 7 correspondence points between  $C(k)$  and  $C(k + 1)$  have to be calculated. In Fig. 5(b), an example of 2D feature points extraction and their calculated 3D positions is illustrated. Using the feature points and the above explained principle, the pose of the camera can be estimated, as seen in Fig. 5(c). As will be explained in Section 4.4, the obtained poses will be used for the fusion of the calculated depth maps.

#### 4.3. Dense closed-loop depth estimation

Once the pose of the camera has been determined, the 3D structure of the imaged scene can be reconstructed with respect to the position and orientation of the camera, i.e. with respect to the robot's pose.

In order to properly compute the camera–object distance  $Z$ , it is needed to establish the location of each 2D point, or pixel,  $p(x, y)$  in each stereo camera image, namely, the 2D image points  $p_L$  and  $p_R$ . The correspondence problem is currently one of the most investigated issues in stereo vision. In literature, there are a number of dense correspondence calculation methods, a comprehensive classification being available in [22]. In this paper, we have chosen to control the so-called *Block Matching* (BM) algorithm with the goal to obtain reliable 3D scene information.

BM is one of the most popular correspondence matching algorithm used in robotics, its main advantage being the fast computation rate, in comparison to more advanced techniques such as *Graph-Cuts* (GC) [22] or *Semi-Global-Matching* (SGM) [25]. Although BM has certain sensitivity to illumination conditions, its computation property makes the method a good candidate for real-time autonomous systems. In this paper, points are matched by calculating a *Sum of Absolute Differences* (SAD) over small sliding windows. The BM method is commonly performed in three steps. In our implementation we have considered the following operations:

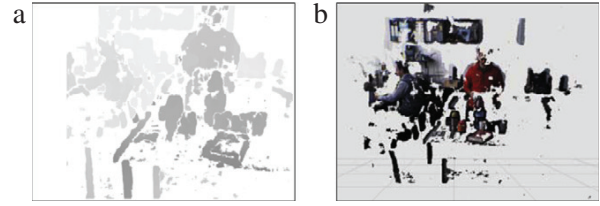
- *Pre-filter* the input images with a  $7 \times 7$  sliding window, containing a moving average filter, to reduce lighting differences and enhance texture.
- *Compute SAD* over a sliding window.
- *Eliminate bad correspondence matches through post-filtering.*

As shown in Fig. 4, the SAD values are calculated using a window shifted in the right images along the interval:

$$H = [d_{min}, d_{max}], \quad (13)$$

where  $H$  is referred to as the *horopter*, defined as the 3D volume covered by the search range of BM. The goal of computing SAD is to find the best matching candidate of point  $p_L$  in the right image, that is  $p_R$ , as:

$$m = \sum_{x,y} [I_L(x, y) - I_R(x + d, y)], \quad (14)$$



**Fig. 6.** Depth estimation via block matching. (a) Disparity map obtained with  $q_r = 16$ . (b) 3D reprojected disparity map.

where  $m$  is the SAD, or match, value and  $d \in H$ . By calculating SAD over  $H$ , we obtain a characteristic in which its maximum represents the best match candidate of  $p_L$  in the right image. Because of the linearity of the equation, SAD is a faster computational approach to BM, as opposed to other metrics such as the *Zero Mean Normalized Cross-Correlation* (ZNCC), or the *Sum of Squared Differences* (SSD) [22].

Post-filtering aims at preventing false matches, hence false disparity maps. For filtering bad matches, a uniqueness ratio function is used, defined as:

$$q_r = \frac{(m - m_{min})}{m_{min}}, \quad (15)$$

where  $m_{min}$  is the minimum SAD, or match, value. A feature is considered a match if:

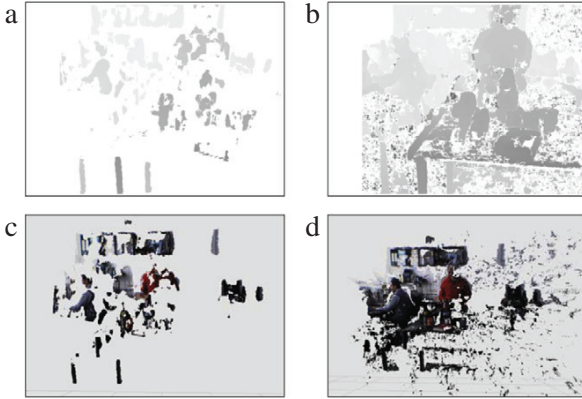
$$q_r > T_q, \quad (16)$$

where  $T_q$  is a predefined uniqueness threshold value. In [26], the value of the uniqueness threshold is suggested to be  $T_q = 12$ . As it will be shown latter in this section, a predefined constant value of  $T_q$  poses problems in 3D reconstruction since, depending on the imaged scene, it can introduce a large number of outliers in the reconstructed 3D model, or a too few number of voxels. To overcome this problem, we propose a feedback control method for the uniqueness threshold  $T_q$ . The output of BM is a gray level image  $I_d(x, y)$ , also referred to as the disparity map, where the levels of gray represent different distances. In Fig. 6(a), the disparity map calculated for the typical cluttered service robotics scene from Fig. 5(b) is presented. The pixels from Fig. 6(a) with a higher brightness are considered to be closer to the camera. Also, the pixels for which no correspondence could be calculated are represented as white. The 3D reprojected scene is illustrated in Fig. 6(b).

The main problem with the open-loop depth estimation system described previously is its low performance with respect to variations in the scene structure, such as variable illumination conditions or clutter. An example of using constant parameters of depth estimation is illustrated in Figs. 6 and 7. As said before, one of the main factors that influences the depth estimation process is the threshold value  $T_q$  of the uniqueness ratio  $q_r$ . If  $T_q$  has an optimal predetermined value, as in Fig. 6(a), the reconstruction from Fig. 6(b) is fairly reliable, having in mind that we operate only with a pair of images. On the other hand, if the scene parameters change, or  $T_q$  has a suboptimal value, 3D reconstruction might fail, as shown in Fig. 7. For a large value of  $T_q$ , as in Fig. 7(a,c), the 3D results have a low number of object voxels, whereas for a high  $T_q$  the number of obtained outliers is too large, as in Fig. 7(b,d).

Although there are a number of parameters that could be controlled, we have considered the depth estimation process, for simplicity, as a *Single Input Single Output* (SISO) model.

The depth sensing process has been modeled as the nonlinear system from Eq. (1). For the sake of clarity, the state vector  $\mathbf{x}$  is considered to have only one element which describes the behavior of the modeled process. Since, depending on the chosen



**Fig. 7.** 3D reconstruction results from suboptimal values of  $T_q$ . (a)  $T_q = 68$ . (b)  $T_q = 4$ . (c, d) 3D reprojected scenes.



**Fig. 8.** Depth segmentation using  $H = [0.7 \text{ m}, 1.5 \text{ m}]$  and different uniqueness thresholds. (a) Optimal  $q_r = 16$ . (b) Over-segmented  $q_r = 68$ . (c) Under-segmented  $q_r = 4$ .

uniqueness threshold  $T_q$ , we obtain a different disparity map  $I_d$ , as shown in Fig. 7, a straightforward way to derive a state variable for the system is to quantify  $I_d$ . In this paper, we suggest the quantification of  $I_d$  through *distance*, or *depth*, *segmentation*. A segmented distance is represented by the region thresholded image  $I_{th}(x, y)$  obtained from the segmentation of the disparity map  $I_d$ .

Depth segmentation can be implemented by specifying a range interval of interest  $H$ , also entitled *horopter*, where the desired objects reside. Using Eqs. (8)–(10), the horopter can be translated from real world metric units to pixel values that map depth in the disparity image  $I_d$ . In Fig. 8, three segmentation examples using a horopter  $H = [0.7 \text{ m}, 1.5 \text{ m}]$  and different uniqueness thresholds are illustrated. As can be seen, only the segmentation result from Fig. 8(a) corresponds to optimal segmentation, the other two being either over- or under-segmented.

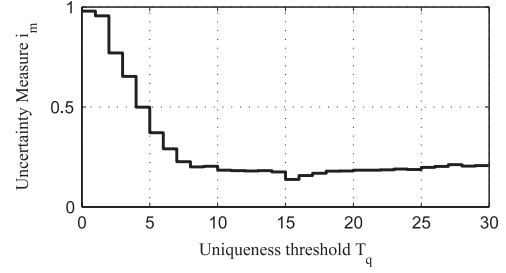
Using the above described depth segmentation principle based on region segmentation, the problem of controlling the quality of the disparity map  $I_d$  is converted into the problem of controlling the quality of the segmented image  $I_{th}$ . A region segmented image is said to be of good quality if it contains all pixels of the objects of interest forming a “full” (unbroken) and well shaped segmented object region. Bearing in mind the qualitative definition of a segmented image of good quality, the quantitative measure of segmented quality in Eq. (17) has been used:

$$i_m = -\log_2 p_8, \quad i_m(0) = 0 \quad (17)$$

where  $p_8$  is the relative frequency, that is, the estimate of the probability of a segmented pixel to be surrounded with 8 segmented pixels in its 8-pixel neighborhood:

$$p_8 = \frac{\text{no. of seg. px. surrounded with 8 seg. px.}}{\text{total no. of seg. px. in the image}}. \quad (18)$$

Keeping in mind that a well segmented image contains a “full” (without holes) segmented object region, it is evident from Eq. (18) that a small probability  $p_8$  corresponds to a large disorder in a binary segmented image. In this case, a large uncertainty  $i_m$  is



**Fig. 9.** The uncertainty measure  $i_m$  of segmented pixels vs. uniqueness threshold  $T_q$ .

assigned to the segmented image. Therefore, the goal is to achieve a binary image having an uncertainty measure  $i_m$  as small as possible in order to get a reliable depth segmentation result.

The depth estimation system was modeled according to Eq. (1), where the involved variables are:

$$\mathbf{x} = [i_m \quad q_r]^T, \quad (19)$$

$$\mathbf{y} = I_d(x, y), \quad (20)$$

$$\mathbf{u} = q_r(i_m, T_q). \quad (21)$$

In Fig. 9, the input–output (I/O) relation between the state variable  $i_m$  and the actuator parameter  $T_q$  is displayed for the case of the scene from Fig. 6. The goal of the proposed extremum seeking control system is to determine the optimal value  $T_q^*$  which corresponds to the minimum of the curve in Fig. 9.  $T_q^*$  represents the desired value of the uniqueness threshold. The shape of the obtained I/O curves, as can also be seen from Fig. 9, preserve the controllability of the system, since the value of the actuator converges to the global minimum representing the equilibrium set-point of the considered system.

Following the above presented discussion, the block diagram of the proposed depth sensing system is illustrated in Fig. 10. First, left and right images are processed in order to establish an initial depth map. The core of the method is represented by the state feedback loop which is used to automatically adapt the actuator parameter  $T_q$  in order to obtain consistent depth estimation. Once the equilibrium set-point has been achieved, the calculated  $I_d$  is used to reconstruct the viewed scene in a 3D environment by reprojecting the voxels using Eqs. (8)–(10).

Although the proposed feedback depth estimation method has been developed around the SAD approach, its adaptation to other metrics, such as ZNCC or SSD, is direct since only the matching cost function is changed, the other BM parameters remaining the same. The feedback improvement of more advanced correspondence matching algorithms, such as GC or SGM, should be investigated, their research being out of the scope of this paper.

#### 4.4. Depth maps fusion

Having obtained both the location of the robotic system through camera pose estimation and the optimal depth maps corresponding to each camera pose, a virtual 3D model of the imaged environment can be reconstructed by fusing these two pieces of information together [27].

Basically, the concept for this fusion problem is to project each calculated depth map in a virtual 3D space where each voxel's 3D location is related to its corresponding estimated camera pose. The advantage of using the closed-loop calculated depth images is that the annotated 3D model contains less noise, since the noisy voxels are filtered out by the feedback adaptation algorithm.

One major problem that has to be solved in depth map fusion is the redundant information coming from overlapping projected

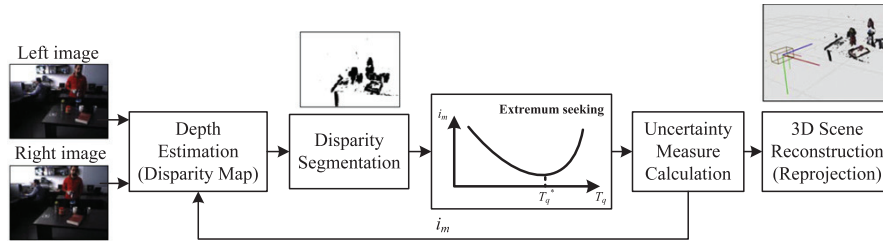


Fig. 10. Block diagram of the proposed feedback control system for robust depth estimation.

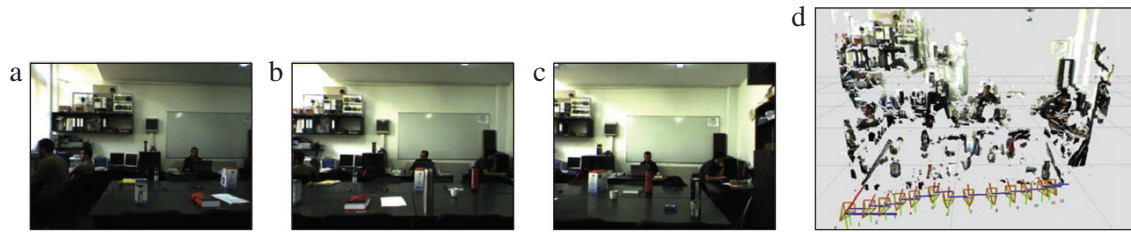


Fig. 11. Depth maps fusion example. (a–c) Snapshots of the imaged environment. (d) Annotated 3D model.

disparity images. In order to save computation time, we have considered as valid voxels those ones visible in the newest images acquired from the stereo camera. In case newer voxels overlay voxels from previous camera poses, the older ones are discarded from the 3D model. Although this approach may seem as a brute force one, it provides enough accurate collision detection results for an autonomous service robot. An example of depth maps fusion within a robotic scene is given in Fig. 11.

The obtained annotated 3D model contains crucial information for autonomous service robots which have to navigate or manipulate objects in complex and uncertain environments. As will be shown in Section 6.3, the 3D virtual scene can be used to detect obstacles and plan the movements of a redundant autonomous manipulator.

## 5. Scene understanding

The final stage in the visual architecture proposed in this paper is the recognition and classification of the visualized objects of interest and obstacles. For this purpose, a classical *Minimum Distance Classifier* [24] has been used. The features used to build the feature vectors are composed of the distance obtained from the method described in Section 4, the color of the objects extracted from the HSV (*Hue, Value, Saturation*) plane of the acquired images and the invariant moments of the color segmented objects.

The 2D segmentation of the objects has been performed on the left image of the acquired stereo image pair using the robust color segmentation algorithm described in [12] for the case of uniform colored objects. Also, for textured objects, such as book, the boundary detection method from [13] has been used. The main objective of the segmentation procedure is to determine the class of objects and their key points of interest. Depending on the obtained class, it can be inferred whether or not the object can be manipulated (e.g. bottle, glass, book, etc.). Once this manipulative feature is computed, key points are calculated with the goal to be used for the visual guidance of a robotic system. As will be described in the next section, such a system can be a redundant manipulator which has the task of object grasping.

In Fig. 12, an example of object segmentation, classification and key feature points extraction can be seen. The key points are represented in Fig. 12(b) by the bold blue and green points. As can be seen, depending on the object class, either region or boundary based segmentation has been used for its detection.

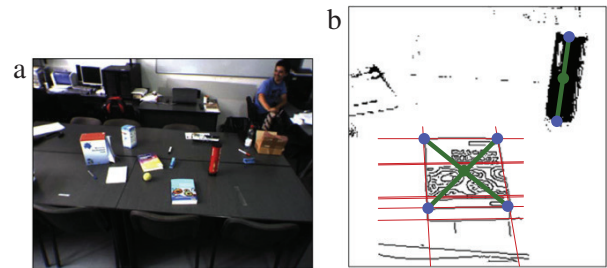


Fig. 12. Region and boundary based object recognition. (a) Input image. (b) Recognized bottle and book objects. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

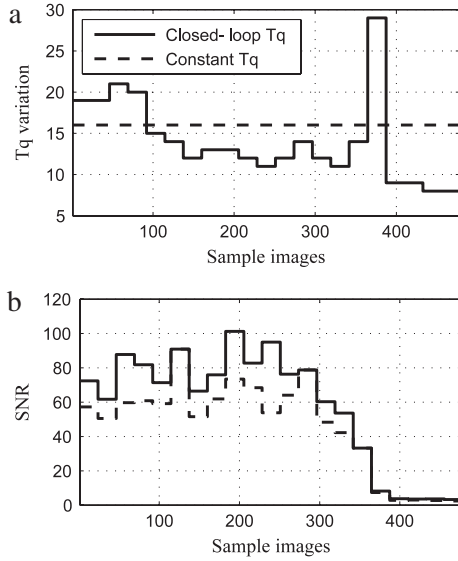
## 6. Performance evaluation

In order to test the capabilities of the proposed visual system, the performance evaluation procedure has been divided into three parts. First, a comparison between the proposed closed-loop depth estimation algorithm and an open-loop counterpart is given. Second, the overall 3D machine vision architecture is evaluated with respect to object pose detection. Finally, the advantage of using the vision platform for the visual control of a redundant manipulator arm is presented in the context of obstacle avoidance path planning.

The evaluation procedure involved a number of 500 images containing 35 objects of interest, such as bottles, glasses, or books, acquired using a Bumblebee® pre-calibrated stereo camera working at a rate of 16 *Frames Per Second* (FPS). An example of such a test scene is illustrated in Fig. 5(b). The illumination used ranged in the interval [15, 1200 lx]. This range of illumination corresponds to a variation of the light intensity from a dark room lighted with candles (15 lx) to the lighting level of an office and above. According to the European law UNI EN 12464, the optimal lighting level of an office has a value of 500 lx.

### 6.1. Closed-loop vs. open-loop depth estimation

In order to give a qualitative evaluation of the proposed closed-loop depth computation algorithm, the method has been compared with the traditional approach which uses constant parameters for



**Fig. 13.** Closed-loop vs. open-loop disparity computation. (a) Variation of the uniqueness threshold  $T_q$ . (b) Signal-to-Noise-Ratio of disparity areas.

the calculation of the disparity map. As explained before, a good depth map is one that contains dense, or “full”, disparity areas with a minimum amount of noise, as in Fig. 6. In this work, the noise is represented by small disparity areas which perturb the 3D model, as in the example from Fig. 7(b). On the other hand, a depth map should contain as many “full” areas as possible, that is, to maximize the amount of processed 3D visual information.

Mathematically, the depth map quality index can be expressed as a *Signal-to-Noise-Ratio* (SNR) between the total sum of pixels in disparity areas  $A_{total}$  and the sum of pixels in noisy areas  $A_{noisy}$ :

$$SNR = \frac{A_{total}}{A_{noisy}}, \quad (22)$$

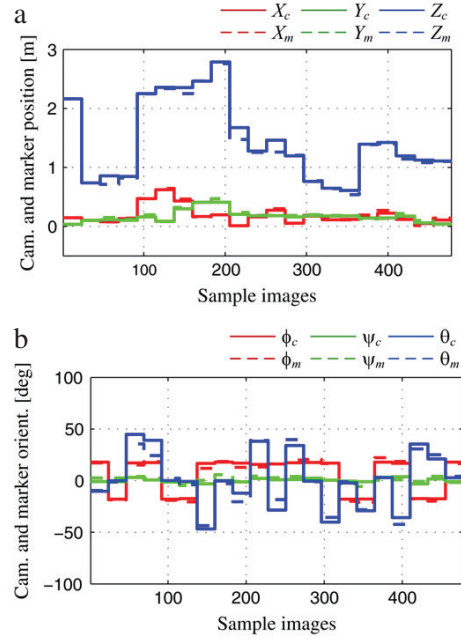
where an area is considered to be noisy if it has a value lower than a specific threshold. The threshold has been set to the heuristically determined value of 100 px. The extraction of the areas was performed via a pixels connectivity evaluation [24].

The closed-loop variation of the uniqueness threshold  $T_q$  is represented in Fig. 13(a), opposed to a manually chosen constant value of  $T_q = 16$ . The corresponding SNR diagram is illustrated in Fig. 13(b). As can be seen from the diagram, the SNR index has a larger value for the case of closed-loop depth estimation, as for the constant value of  $T_q$ . The improvement of the 3D model is also visible through the mean value of the SNR, which for closed-loop has a value of 59.2105 in comparison to 41.6336, representing to open-loop situation.

In a robotic application, the main advantage of the closed-loop depth estimation system is the noise reduced annotated 3D model which can be used for on-line obstacle avoidance. Such a path planning example will be given in Section 6.3 for the case of a redundant manipulator arm.

## 6.2. Evaluation of the overall vision architecture

The evaluation of the overall machine visual system has been performed with respect to the real 3D poses of the objects of interest. The real 3D positions and orientations of the objects of interest were manually determined using the following setup. On the imaged scene, a visual marker, considered to be the *ground truth* information, was installed in such a way that the poses of the



**Fig. 14.** Estimated positions (a) and orientations (b) of the stereo camera.

objects could be easily measured with respect to the marker. The 3D pose of the marker was detected using the ARToolKit library which provides subpixel accuracy estimation of the marker's location with an average error of  $\approx 5$  mm [28]. By calculating the marker's 3D pose, a ground truth reference value for camera position and orientation estimation could be obtained using the inverse of the marker's pose matrix. Further, the positions of the imaged objects, as well as the camera pose, were calculated using the proposed system which includes the feedback mechanisms for depth estimation. Both results, that is camera and objects poses, were compared to the ground truth data provided by the ARToolKit marker.

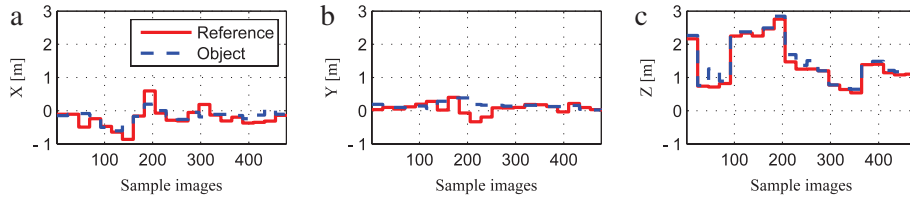
In Fig. 14, camera pose estimation results obtained using the two methods, that is through the proposed method and via the ARToolKit marker approach, are presented. As can be seen from both diagrams, the marker-less pose estimation algorithm described in this paper delivered a camera position ( $X_c, Y_c, Z_c$ ) closely related to the ground truth values ( $X_m, Y_m, Z_m$ ) obtained from the marker. Also, the calculated orientations ( $\phi_c, \psi_c, \theta_c$ ) followed the reference ones ( $\phi_m, \psi_m, \theta_m$ ). This correlation can be easily observed when analyzing the statistical error results, given in Table 1, between the two approaches. Namely, for both the position and orientation, the errors are small enough to ensure a good spatial localization of the robot and also to provide reliable depth maps fusion.

In order for a robotic system to manipulate objects of interest, their 3D pose has to be precisely determined. As for the case of camera position and orientation evaluation, the pose of the objects has been also determined with respect to the ground truth marker. In Fig. 15 the estimated positions of the imaged objects of interest are shown together with their real positions measured with respect to the marker. Since only the position has been varied, the orientation remaining constant, the diagrams in Fig. 15 illustrate only different objects positions with respect to the pose of the camera system. Statistical results of 3D position estimation errors are given in Table 2.

As can be seen from Table 2, the mean 3D position error has a value of (0.0443, 0.0264, 0.0059 m) which is in many cases

**Table 1**  
Statistical results of errors between proposed and marker based 3D camera pose estimation.

	$X_e$ (m)	$Y_e$ (m)	$Z_e$ (m)	$\phi_m$ (deg)	$\psi_m$ (deg)	$\theta_m$ (deg)
Max error	0.0497	0.0595	0.1015	4.2348	5.6915	10.1187
Mean	0.0135	0.0140	0.0429	0.7887	0.7247	0.6765
Std. dev.	0.0211	0.0207	0.0642	2.3029	2.6090	5.5249



**Fig. 15.** Real (reference) and estimated positions of objects of interest along the three Cartesian axes ( $X$ ,  $Y$ ,  $Z$ ).

**Table 2**  
Statistical results of 3D position estimation errors for the considered objects of interest.

	$X_e$ (m)	$Y_e$ (m)	$Z_e$ (m)
Max error	0.3006	0.2227	0.1074
Mean	0.0443	0.0264	0.0059
Std. dev.	0.1190	0.1082	0.1198

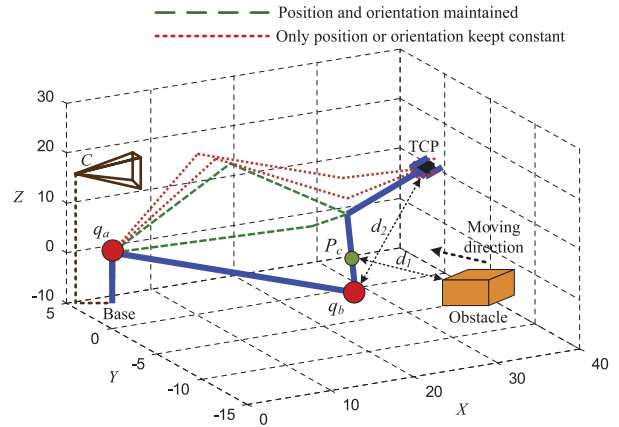
tolerable for object manipulation. Nevertheless, the maximum achieved error, at image sample 204 has a high value of (0.3006, 0.2227, 0.1074 m). Although the pose of the camera at that particular sample has a small 3D error, the position of the object detected at that instance is unreliable for robotic manipulation. It is interesting to notice that the object position error is correlated to the camera–object distance along the  $Z$  Cartesian axis. As it can be seen from Fig. 15(a), the  $Z$  position of the camera at sample 204 is at its highest value of 2.78 m from the considered object. The high object position error comes in this case from the object triangulation algorithm. Namely, a small error in manipulative key points calculation, explained in Section 5, exponentially increases the final 3D object position error when the camera–object distance increases. Since in a robotic system the camera is attached to the robotic platform, the objects of interest are usually imaged in a close range for the purpose of object manipulation.

### 6.3. Visual control of a redundant manipulator

The visual architecture proposed in this paper was also successfully used within a robotic collision avoidance system. In order to avoid collisions in a dynamic scene, a robotic system has to reliably detect the poses of objects and obstacles such that it corrects its movement in real-time [29]. For demonstrating the collision avoidance capabilities, we have chosen to treat all detected objects as obstacles. Thus, the goal of the robot is to maintain a certain kinematic configuration in the Cartesian space based on acquired visual information.

The robotic system has been modeled using a 7 Degrees-of-Freedom (DoF) manipulator arm in a Denavit–Hartenberg configuration, as shown in Fig. 16. The arm used for experiments is a Robotnik® manipulator which is a part of the RESCUER® robotic platform. The camera system is located at a fixed pose  $C$  with respect to the base of the manipulator arm.

In our implementation, the collision avoidance system has been configured as two algorithms. The first one keeps both the position and the orientation of the Tool Center Point (TCP) constant, while the second one maintains constant only the position or the



**Fig. 16.** Experimental setup for real-time collision avoidance.

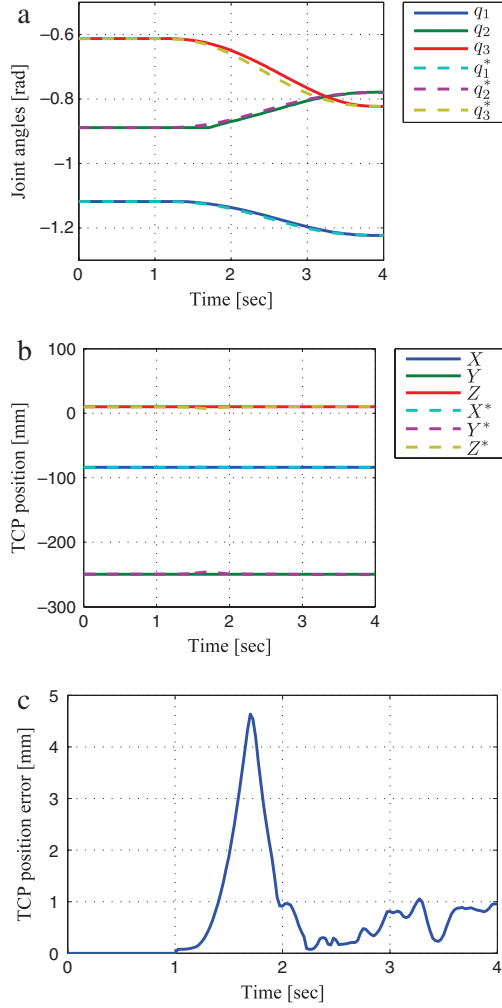
orientation. This second case is represented by such collisions that cannot be avoided if the arm's pose is kept constant. Both approaches are illustrated in Fig. 16, where the objective of the robot control system is to hold the pose of the grasped object while an obstacle moves in the direction of the arm.

In order to determine a collision free configuration for the robotic arm, the control method calculates the distance  $d_1$  between the surface normal of the detected object to the so-called collision point  $P_c$ .  $P_c$  is represented by the point on the manipulator arm nearest to the object. Further, we have considered the variable  $d_2$  represented by the distance between the TCP and the joint closest to  $P_c$ , starting from the base of the robotic arm. The collision avoidance implementation is based on the following two constraints:

1. maximize the distance  $d_1$ ;
2. maintain a constant distance  $d_2$ .

During on-line operation, the system determines the two joints  $q_a$  and  $q_b$  that must be controlled in order to maximize  $d_1$  and keep  $d_2$  constant. These joints are obtained by evaluating their impact, if actuated, on the variation of  $d_1$  and  $d_2$ . Mathematically, the variation is expressed as the derivative of the two considered distances,  $d_1$  and  $d_2$ . Hence, in order to determine  $q_a$  and  $q_b$ , the collision avoidance algorithm optimizes the following criteria:

$$\begin{bmatrix} q_a \\ q_b \end{bmatrix} = \begin{cases} \arg \max d_1, \\ \arg \min d_2. \end{cases} \quad (23)$$



**Fig. 17.** Performance evaluation results for the proposed collision avoidance system, in a 7-DoF manipulator arm. (a) First three reference  $q_i^*$  and real  $q_i$  joint angles. (b) Reference  $(X, Y, Z)$  and real  $(X^*, Y^*, Z^*)$  position of the TCP. (c) Cumulated position error of the TCP.

The desired positions of the joints between the base of the robot and the collision point are determined by solving the system:

$$\begin{cases} d_1(\mathbf{q}) = d_1^*, \\ d_2(\mathbf{q}) = d_2^*, \end{cases} \quad (24)$$

where  $d_1^*$  is the reference safe distance from the detected object to the robot arm and  $d_2^*$  the initial distance between the TCP and the joint nearest to  $P_c$ , starting from the base of the manipulator. The vector  $\mathbf{q}$  is defined as:

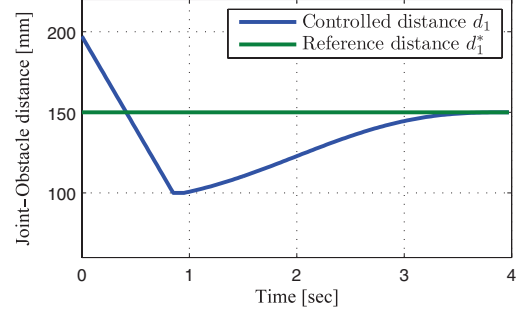
$$\mathbf{q} = [q_a^i q_b^i]^T, \quad (25)$$

where  $i$  is the number of joints of the manipulator arm. Further, the desired positions of the other robotic joints are calculated using the considered inverse kinematic model. The solution of Eq. (24) is determined through the following recursive equation:

$$\mathbf{q}^{i+1} = \mathbf{q}^i + \mathbf{J}^{-1}(\mathbf{q}^i) \cdot \delta \mathbf{F}(\mathbf{q}^i), \quad (26)$$

where:

$$\delta \mathbf{F}(\mathbf{q}^i) = \mathbf{F}(\mathbf{q}^i) - \mathbf{F}^*, \quad (27)$$



**Fig. 18.** Robot-obstacle distance behavior during experimental evaluation.

$$\mathbf{F}(\mathbf{q}^i) = [d_1(\mathbf{q}^i) \quad d_2(\mathbf{q}^i)]^T, \quad (28)$$

$$\mathbf{F}^* = [d_1^* \quad d_2^*]^T, \quad (29)$$

$$\mathbf{J}(\mathbf{q}^i) = \begin{bmatrix} \frac{\partial d_1}{\partial q_a} & \frac{\partial d_1}{\partial q_b} \\ \frac{\partial d_2}{\partial q_a} & \frac{\partial d_2}{\partial q_b} \end{bmatrix}, \quad (30)$$

where  $\mathbf{F}$  is the system matrix of Eq. (24),  $\mathbf{F}^*$  is the solution of  $\mathbf{F}$  and  $\mathbf{J}$  represents the Jacobian of  $\mathbf{F}$ .

In Fig. 17, an example of collision avoidance using the proposed method is illustrated. In the presented example, at time  $t = 0.9$  s, an obstacle is detected. The control method then triggers the system to recalculate the desired joints angles, depending on the direction of the obstacle, as described above. Since the TCP's pose must be maintained, only the first three joints of the manipulator are controlled, the other four being feed with a constant reference angle. As can be seen, the real joints values  $q_i$  follow the desired reference values  $q_i^*$ . Also, as shown in Fig. 17(b), the real position  $(X, Y, Z)$  of the TCP is also consistent with their desired values  $(X^*, Y^*, Z^*)$ . Finally, it can be seen from Fig. 17(c) that the cumulated TCP position error is low, with a maximum value below 5 mm.

The behavior of the robot with respect to the obstacle distance  $d_1$  is illustrated in Fig. 18. In order to maintain a safe distance between the robot and the detected object, or obstacle, the value of  $d_1$  should be kept above a predefined safety distance  $d_1^*$ . As can be seen from Fig. 18, once the probability of a collision has been determined at  $t = 0.9$  s, the collision avoidance system starts controlling the manipulator arm with the goal to increase  $d_1$  above the value of  $d_1^*$ . In our experiments, we have considered the minimal safety distance to be  $d_1^* = 150$  mm.

## 7. Conclusions and outlook

In this paper, a robust visual perception system for service robotics applications has been proposed. The goal of the architecture is the reliable extraction of pose and depth information that can be used in autonomous systems, such as mobile manipulators. The main feature of the visual platform is the closed-loop improvement of the depth sensing system. The robust depth estimation approach plays a crucial role in scene reconstruction tasks where the correct detection of objects of interest and obstacles is needed. The proposed system has been successfully tested within a collision avoidance architecture for a 7-DoF robotic manipulator. As future work, the authors consider the extension of the theoretical aspects of closed-loop image processing to other visual tasks, such as robust object recognition and classification. Also, the aspects of camera motion dynamics and visual information fusion will be further investigated for the improvement of the proposed visual perception system.

## Acknowledgments

This paper is supported by the Sectoral Operational Program Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the projects POSDRU/89 /1.5/S/59323, POSDRU/88 /1.5/S/59321, POSDRU/107 /1.5/S/76945 and POSDRU/6/ 1.5/S /6.

## References

- [1] D. Kragic, H.I. Christensen, Advances in robot vision, *Robotics and Autonomous Systems* 52 (2005) 1–3.
- [2] D. Kim, R. Lovelett, A. Behal, An Empirical Study with Simulated ADL Tasks using Vision-Guided Assistive Robot Arm, in: *Proc. of the IEEE 11th Int. Conf. on Rehabilitation Robotics ICORR 2009*, Kyoto, Japan, 2009.
- [3] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.
- [4] R. Bajcsy, Active perception, *Proceedings of the IEEE* 76 (8) (1988) 996–1005.
- [5] F. Chaumette, S. Hutchinson, Visual servo control part I: basic approaches, *IEEE Robotics and Automation Magazine* 13 (4) (2006) 82–90.
- [6] M. Mirmehdi, P. Palmer, J. Kittler, H. Dabis, Feedback Control Strategies for Object Recognition, *IEEE Transactions on Image Processing* 8 (4) (1999) 1084–1101.
- [7] Q. Zhou, L. Ma, D. Chelberg, Adaptive object detection and recognition based on a feedback strategy, *Image and Vision Computing* 24 (2006) 80–93.
- [8] J. Peng, B. Bahnu, Closed-loop object recognition using reinforcement learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (2) (1998) 139–154.
- [9] J. Marchant, C. Onyango, Model-based control of image acquisition, *Image and Vision Computing* 21 (2003) 161–170.
- [10] S. Gutierrez, J. Marroquin, Robust approach for disparity estimation in stereo vision, *Image and Vision Computing* 22 (2004) 183–195.
- [11] D. Ristic, *Feedback Control in Image Processing*, Shaker Verlag, Aachen, Germany, 2007.
- [12] S.M. Grigorescu, D. Ristic-Durrant, A. Graeser, ROVIS: ROBust machine Vision for Service robotic system FRIEND, in: *Proceedings of the 2009 Int. Conf. on Intelligent Robots and Systems*, St. Louis, USA, 2009.
- [13] S. Grigorescu, S. Natarajan, D. Mronga, A. Graeser, Robust Feature Extraction for 3D Reconstruction of Boundary Segmented Objects in a Robotic Library Scenario, in: *Proc. of the 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.
- [14] S.M. Grigorescu, Robust Machine Vision for Service Robotics, Ph.D. Thesis, Bremen University, Institute of Automation, Bremen, Germany Jun. 2010.
- [15] K. Ariyur, M. Krstic, *Real-Time Optimization by Extremum Seeking Control*, John Wiley and Sons, New York, USA, 2003.
- [16] J. Neira, A. Davison, J. Leonard, Visual SLAM, *IEEE Transactions on Robotics* 24 (5) (2008) 929–931 (special issue).
- [17] L. Paz, P. Pinies, J. Tardos, J. Neira, Large-scale 6-DOF SLAM with stereo-in-hand, *IEEE Transactions on Robotics* 24 (5) (2008) 946–957.
- [18] T. Lemaire, C. Berger, I. Jung, S. Lacroix, Vision-based SLAM: stereo and monocular approaches, *International Journal of Computer Vision* 74 (3) (2007) 343–364.
- [19] G. Welch, G. Bishop, *An Introduction to the Kalman Filter*, UNC-Chapel Hill, 2006.
- [20] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking, *IEEE Transactions on Signal Processing* 50 (2) (2002) 174–188.
- [21] M.A. Lourakis, A. Argyros, SBA: a software package for generic sparse bundle adjustment, *ACM Transactions on Mathematical Software* 36 (1) (2009) 1–30.
- [22] M. Brown, D. Burschka, G. Hager, Advances in computational stereo, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (8) (2003) 993–1008.
- [23] C. Harris, M. Stephens, A combined corner and edge detection, *Proceedings of the Fourth Alvey Vision Conference* (1988) 147–151.
- [24] R. Gonzalez, R. Woods, *Digital Image Processing*, Pearson Education, 3rd edition, 2007.
- [25] H. Hirschmuller, Stereo processing by semi-global matching and mutual information, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2) (2008) 328–341.
- [26] G. Bradski, A. Kaehler, *Learning OpenCV, Computer Vision with the OpenCV Library*, O'Reilly Media, USA, 2008.
- [27] R. Rusu, Z. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3D point cloud based object maps for household environments, *Robotics and Autonomous Systems* 56 (11) (2008) 927–941.
- [28] P. Malbezin, W. Piekarski, B. Thomas, Measuring ARToolKit Accuracy in Long Distance Tracking Experiments, in: *1st Int. Augmented Reality Toolkit Workshop*, Darmstadt, Germany, 2002.

- [29] N. Ogren, I. Egerstedt, X. Hu, Reactive Mobile Manipulation Using Dynamic Trajectory Tracking, in: *IEEE Inter. Conf. on Robotics and Automation ICRA 2000*, San Francisco CA, USA, pp. 3473–3478.



**Sorin M. Grigorescu** received the Dipl.-Eng. Degree in Control Engineering and Computer Science from the University Transilvania of Brasov, Romania, in 2006, and the Ph.D. degree in Process Control from the University of Bremen, Germany, in 2010. Between 2006 and 2010 he was a member of the Institute of Automation, University of Bremen. From 2009 to 2010 he coordinated the FRIEND service robotics project. Since June 2010 he is affiliated with the Department of Automation at UTBv, where he leads the Robust Vision and Control Laboratory. Sorin M. Grigorescu was an exchange researcher at several institutes, such as the Korea Advanced Institute of Science and Technology (KAIST) or the Robotic Intelligence Lab in University Jaume I. He is a member of the IEEE Robotics and Automation Society (RAS) and of the Romanian Society for Automatic Control and Applied Informatics (SRAIT). His research interests include Robust Vision, Feedback Control in Image Processing and Service Robotics.



**Gigel Macesanu** was born in Romania in 1985. He received the Bachelor degree in automation and computer science from Transilvania University of Brasov, Romania, in 2009. Since 2009, he has been with the Department of Process Control Systems, Transilvania University of Brasov, where he is currently working toward his Ph.D. His main research interests are image processing, active vision and robotics. Mr. Macesanu is a member of the Romanian Society for Automatic Control and Applied Informatics (SRAIT).



**Tiberiu T. Cocias** was born in Brasov, Romania in 1985. He received the Bachelor degree in automation and computer science from Transilvania University of Brasov, Romania, in 2009. Since 2009, he has been with the Department of Process Control Systems, Transilvania University of Brasov, where he is currently working toward his Ph.D. His main research interests are image processing and 3D reconstruction. Mr. Macesanu is a member of the Romanian Society for Automatic Control and Applied Informatics (SRAIT).



**Dan Puiu** was born in Brasov, Romania, in 1984. He received the B.S. and M.S. degrees in control engineering from Transilvania University of Brasov, Romania in 2008 and 2009, respectively. Currently, he is a Ph.D. Student at Transilvania University of Brasov, within the Process Control Systems research department. At the end of 2010, he was an exchange researcher at the Robotic Intelligence Lab from University Jaume I, Castellon de la Plana, Spain. His research interests include real-time motion planning, distributed control systems, industrial networks and electrical drives. He is a member of the Romanian Society of Automation and Technical Information (SRAIT).



**Florin Moldoveanu** received a Dipl.-Eng. Degree in Electrical Engineering from "Politechnica" Institute of Brasov, Romania in 1975, and a Ph.D. degree in Electrical Engineering from UTBv, Romania in 1998, with a thesis on Control Engineering. He joined in 1990 the Department of Automation at UTBv where he is currently a Professor. He is the author or co-author of 7 technical books and has published more than 100 papers in scientific journals and proceedings of national and international conferences. Florin Moldoveanu participated in several national and international RFD projects. He is a member of the IEEE Industrial Electronics Society (IES), head of the Brasov branch of the Romanian Society for Automatic Control and Applied Informatics (SRAIT), member of the National Society for Medical Engineering and Biological Technology (SNMITB) and member in the International Association of Online Engineering (IAOE). His main research interests include control engineering, distributed control systems and computer vision.