Robot depth estimation inspired by fixational movements

Angel J. Duran
Robotic Intelligence Lab
Universitat Jaume I Castellon, Spain
abosch@uji.es

Angel P. del Pobil
Robotic Intelligence Lab
Universitat Jaume I Castellon, Spain
Department of Interaction Science,
Sungkyunkwan University, Seoul, Korea
pobil@uji.es

***Abstract*—Distance estimation is a challenge for robots, human beings and other animals in their adaptation to changing environments. Different approaches have been proposed to tackle this problem based on classical vision algorithms or, more recently, deep learning. We present a novel approach inspired by mechanisms involved in fixational movements to estimate a depth image with a monocular camera. An algorithm based on microsaccades and head movements during visual fixation is presented. It combines the images generated by these micromovements with the ego-motion signal, to compute the depth map. Systematic experiments using the Baxter robot in the Gazebo/ROS simulator are described to test the approach in two different scenarios, and evaluate the influence of its parameters and its robustness in the presence of noise.**

***Index Terms*—Robot prototyping of human and animal skills, Development of skills in biological systems and robots, Sensorimotor development, Cognitive vision, Monocular depth estimation**

I. Introduction

The human visual-oculomotor system is a source of inspiration for solving problems pertaining to visual perception in robotics. Many species exhibit behaviours that require accurate depth estimation in their environments. In particular, primates solve this problem by the concurrent use of multiple estimators deriving from different visual cues [5]. Even so, the most popular sensors used in robotics nowadays to obtain this information are arguably RGB-D sensors, such as Microsoft Kinect [13] and any of its variations [19]. They are typically based on the projection of a known infrared pattern and, depending on the objects in front of the sensor, the deformation of this pattern is used to estimate the depth. In computer vision, a number of methods and algorithms have been established for determining the depth of a scene using a single camera or image. For instance, by applying patches to determine the pose of planes in a single image it is possible to generate the depth map with a single image [30]. Also, from a stream of images, the depth map can be deduced if the velocity of the camera is known [22]. Recent results about obtaining structure from motion with a monocular camera are based on feature tracking and triangulation methods [24], [31]. In addition to these methods, novel deep learning approaches use complex neural network architectures to learn the correlation between an RGB image and its equivalent RGB-D in an unsupervised way [25], [8]. All of these procedures have in common that they only consider the visual cues as inputs, ignoring the motion of the camera, and sometimes even computing it from the images. RGB-D sensors and deep learning techniques have certain drawbacks. In the case of the former, objects with absorption in the infrared range are not detected, and these sensors have also problems outdoors and with reflective and transparent objects. From a more practical point of view, robotic manipulation in a confined space requires a streamlined design with a sensor in hand and, even though state-of-the-art RGB-D cameras are more compact, they are not an option compared with the fully integrated built-in eye-in-hand camera we use in our experiments. In the case of deep learning, long training processes along with large and pertinent datasets are necessary; moreover, a number of specific problems arise when this technique is applied in robotics [9].

In humans, it has been suggested that the retinal image motion is not enough to determine the depth sign in reference to the fixation plane, and the direction of the image movement relative to the observer motion is decisive to obtain this depth sign [23]. A number of species use eye movements in coordination with small displacements of the head during the process of visual fixation to obtain depth information of the gazed scene [4]. The 'fixation' process is anything but fix, since during maintained fixation tiny intersaccadic eye movements around the gazed location are produced. A large fraction of these movements is smooth, but seemingly random changes in eye position occur: so-called ocular drift and ocular tremor, respectively [17]. Moreover, during maintained fixation, very small saccades (microsaccades) are generated with variable frequency and amplitude. Even though microsaccades and saccades exhibit similar motor characteristics and share a common neural substrate [16], there has been a long controversy over the visual functions of these movements. Recent studies show that these microsaccades are precisely directed and play a fundamental role in enhancing visual acuity [14].

These ideas from biology inspired earlier work in robotics for distance estimation based on the parallax produced by camera rotations [29] and compensatory head/eye movements [18]. In later works the concept is extended to depth estimation [1]. Although Antonelli and co-workers based their work on the coordination of the neck and the oculomotor system to maintain the fixation point, they did not consider microsaccadic movements [2], [3].

In this paper, we build on our earlier work [7] on monocular depth estimation taking inspiration from mechanisms involved in fixational movements in humans and primates, namely, micro-displacements of the head and microsaccadic movements. The key idea is to consider the images after micro-movements as perturbations of the initial fixation image and use them, in combination with the ego-motion signal, to generate the depth map. Our preliminary results suggested that the approach was able to satisfactorily estimate the depth in a scene, and that microsaccades play an essential role in this process [7]. Here, we consolidate our procedures and expand our results. First, the mathematical model is presented in detail in section II including its algorithmic implementation. Then, in section III, a set of systematic experiments using the Baxter robot in the Gazebo/ROS simulator are thoroughly described, with the purpose of evaluating the approach in two different scenarios, and studying the influence of its parameters and its robustness in the presence of noise. The results are evaluated in comparison with the depth provided by the simulator as ground truth, and also with the detector algorithm for the Aruco visual markers located in the scenario. Finally, these results are discussed in section IV.

II. Model

A. Model hypothesis

The human fixation mechanisms are the source of inspiration to develop the proposed model which is sustained by these hypotheses: i) During the fixation process, head-eye movements can be considered as perturbations around an initial pose. A complex set of coordinated movements implicating the head and the oculomotor system are generated in the fixation process [4]. The aim of these movements is to maintain the gaze point in spite of random displacements of head and eyes during fixation. ii) So-called visual suppression occurs during microsaccadic movements [12] to the effect that only in the intersaccadic gaps is visual information accessible. In consequence, the fixation process can be regarded as a spatial image sampling. iii) The main cue for the estimation of depth and 3D perception is the optical flow produced by the observer. When it is not the result of external movements, optic flow and motion parallax are consistent when other depth cues are not available [11]. iv) The contribution generated by ego-motion signal makes it possible to clarify the inherent ambiguity associated with the optical flow [15].

B. Mathematical model

In the beginning, there is no information about the depth of the scene. We consider an initial gazed point. Then, the visual system moves to the initial fixation pose and the image that is received by the visual system is taken as reference. After that, microsaccades and head movements start. The source of movements is only produced by the visual system, and the scene is considered static during this fixation process. Due to perception suppression, images are not considered during saccades. When the saccadic movement has just finished the image received by the visual system is compared with the reference image and depth perception is updated with this new information. To simplify the problem, we consider a range of distances in the scene defined by a near plane Z_n and a far plane Z_f which are perpendicular to the Z visual system axis. Therefore, depth perception takes place within this range. A schema of this behaviour from a robotic point of view is shown in Fig.2. When t=0 the camera has a pose to look at the gazed point. This is the starting point of the fixation process with initial image of reference I_0. Given a point of the scene (P_0) that generates an intensity value in that image and projecting it onto the image plane, the pixel $\{x_i^0, y_i^0\}$ is obtained. The Z axis of the visual system is aligned with the gaze point and Z_0 is the value of the depth in P_0, understanding depth here as the distance from the camera frame of reference $\{C_0\}$ to the perpendicular plane to the camera Z-axis containing P_0.

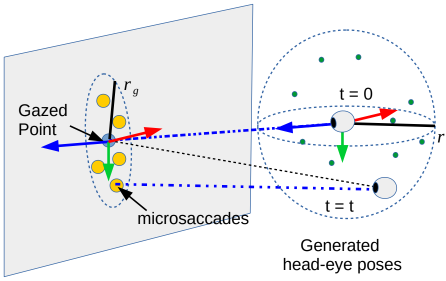


Fig. 1: Simplified scheme to show the sphere of eye movements. These poses result from head-eye movements. The figure is not scaled and it just illustrates that there are two spaces: the space of eye-head movements (a sphere with radius r_m) and the microsaccade space composed by the projections of the end point of a microsaccade around the initial gazed point (radius r_g).

After a head movement and a microsaccade (t=t), the gazed point has been displaced, and the new camera pose is aligned with this new gazed point. The depth with respect to the new camera frame $\{C_t\}$ has changed. After the microsaccade, a new image I_t is obtained. The original P_0 is now P_t with respect to the new camera frame and its projection on the image plane corresponds to a new pixel position in the image $\{x_i^t, y_i^t\}$. An optical displacement has taken place in the image plane $O_f = \{Sx, Sy\}$. This value can be estimated by computing the optical flow between both images.

Our aim is to determine the value of Z_0 that corresponds to depth sensation in the fixation point. In order to reach this goal, we define several matrices and vectors in homogeneous coordinates. The pixel coordinates in I_t and I_0 are defined by vectors $m_t = [u_t, v_t, 1, 1]^T$ and $m_0 = [u_0, v_0, 1, 1]^T$, where u and v are expressed in the centred image coordinates system. We define a projection matrix $\mathbf{K}$ that is a function of the camera parameters, mainly of the focal lengths. To simplify the model $\mathbf{K} = \{k_{i,j}, \forall i, j \in \{1, \ldots, 4\}, i \neq j \implies$

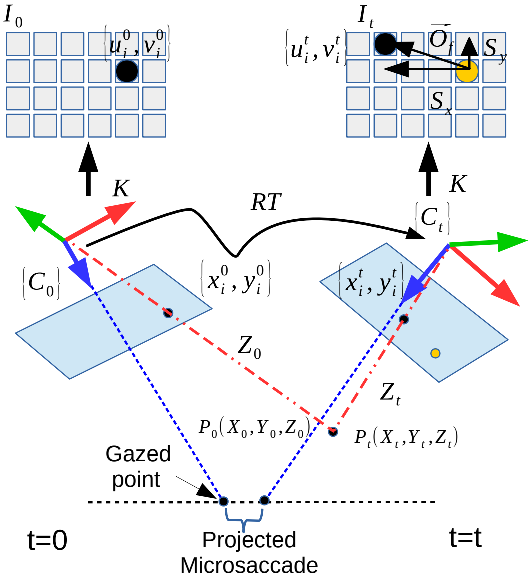


Fig. 2: Schema of the considered camera movements. Initially, the camera is represented by the $\{C_0\}$ frame of reference. A point P_0, with coordinates in $\{C_0\}$ given by $\{X_0, Y_0, Z_0\}$, projects onto the image plane with coordinates $\{x_i^0, y_i^0\}$ and pixel coordinates $\{u_i^0, v_i^0\}$, which are computed using the projection matrix K. A roto-translation (RT) of $\{C_0\}$ results in a new frame $\{C_t\}$ and the projection of that point changes to $\{x_i^t, y_i^t\}$ and $\{u_i^t, v_i^t\}$. Its apparent displacement on the image is given by $O_f = \{S_x, S_y\}$

$k_{i,j} = 0 \wedge diag(\mathbf{K}) = \{f, f, 1, 1\}\}$ where f is the focal length of the camera. To work in homogeneous coordinates, we define two matrices that depend on the depth value: $\mathbf{H}(Z) = \{h_{i,j}, \forall i, j \in \{1, \dots, 4\}, i \neq j \implies h_{i,j} = 0 \wedge diag(\mathbf{H}) = \{1/Z, 1/Z, 1/Z, 1\}\}\}$. Thus, there are two such matrices, one for the initial camera pose $\mathbf{H}(Z_0)$ and the other one for the other pose $\mathbf{H}(Z_t)$. Finally, regarding roto-translation matrix between the frames, we consider that the angular variation is small enough to approximate the rotation by using the skew matrix $\mathbf{M}$; in addition, the translation matrix $\mathbf{T}$ is given by the Cartesian difference between $\{C_0\}$ and $\{C_t\}$. These matrices are defined in (1).

$$\mathbf{M} = \begin{pmatrix} 0 & -\Delta W_z & \Delta W_y \\ \Delta W_z & 0 & -\Delta W_x \\ -\Delta W_y & \Delta W_x & 0 \end{pmatrix}; \mathbf{T} = \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} \tag{1}$$

Where $\Delta W_{(x,y,z)}$ is the angular variation in each axis. The roto-translation matrix $\mathbf{RT}$ is defined as a composition in (2).

$$\mathbf{RT} = \begin{pmatrix} 1 + \mathbf{M} & \mathbf{T} \\ 0 & 1 \end{pmatrix} \tag{2}$$

If the ego-motion signal is known by means of T and M, the new pixel position in the image plane m_t can be computed by using expression (3) from the reference image pixel position.

$$m_t = \mathbf{H}(Z_t) \cdot \mathbf{K} \cdot \mathbf{RT} \cdot \mathbf{K}^{-1} \cdot \mathbf{H}(Z_0)^{-1} \cdot m_0 \tag{3}$$

The value of Z_t can be obtained from the expression: $P_t = \mathbf{RT} \cdot P_0$, and taking into account that $P_0 = \{u_0 \cdot Z_0/f, v_0 \cdot Z_0/f, Z_0\}$, the value of Z_t can be calculated with equation (4).

$$Z_t = Z_0 + \Delta Z - X_0 \Delta W_y + Y_0 \Delta W_x \tag{4}$$

From equations (3) and (4), it can be concluded that m_t is only a function of the camera parameters (f), the ego-motion components ($\{\Delta X, \Delta Y, \Delta Z, \Delta W_x \Delta W_y, \Delta W_z\}$ and the initial depth Z_0. When the scene is considered static, the apparent displacement produced in the image of pixel m_0 is only originated by ego-motion, therefore equation (5) must be satisfied.

$$m_0 = m_t - O_f \tag{5}$$

However, given that both the ego-motion and the optic flow (O_f) can have an error in their estimations, we can rewrite equation (5) as (6).

$$m_0 = \hat{m}_t - \hat{O}_f + \epsilon \longrightarrow \epsilon = m_0 - \hat{m}_t + \hat{O}_f \tag{6}$$

Where ϵ represents the accumulative error resulting from computing m_t using expression (3), and also includes the optic flow estimation error. m_0 is known, since it is the initial pixel position in the reference image, whereas m_t can be calculated from expressions (3) and (4). ϵ is a vectorial magnitude and thus, a cost function based on its module can be defined as (7).

$$\begin{aligned} L = & \quad \tfrac{1}{2} \parallel \epsilon \parallel^2 = \tfrac{1}{2} \left(\epsilon_u^2 + \epsilon_v^2 \right) \\ = & \quad \tfrac{1}{2} \left((u_0 - \hat{u}_t + S_x)^2 + (v_0 - \hat{v}_t + S_y)^2 \right) \end{aligned} \tag{7}$$

If we assume that the value of Z_0 is not correct and the errors corresponding to optic flow components $\{S_x, S_y\}$ and ego-motion estimation are approximately constant, the greatest contribution to the value of ϵ is the undetermined knowledge about Z_0. If Z_0 were the optimum value for the cost function defined in (7), it could be computed using expression (8).

$$\begin{aligned} Z_0^* &= \arg\min_{Z_0^* \in [Z_n, Z_f]} L = \quad \{Z_0^* \mid \forall \alpha \in [Z_n, Z_f] : \\ & \qquad\qquad\qquad\qquad L(\alpha) \geq= L(Z_0^*)\} \end{aligned} \tag{8}$$

Deriving (7) with respect to Z_0, we obtain (9)

$$\frac{\partial L}{\partial Z_0} = -[m_0 - \hat{m}_t + \hat{O}_f]^T \cdot \frac{\partial \hat{m}_t}{\partial Z_0} \tag{9}$$

It is useful to define these expressions to implement (9):

$$\begin{aligned} f_u &= u_o/f; & f_v &= v_o/f \\ V_z &= (1 - \Delta W_y f_u + \Delta W_x f_v)\,; & A_z &= \Delta Z + Z_0 V_z \\ V_y &= (\Delta W_x - f_v - \Delta W_z f_u)\,; & A_y &= \Delta Y - Z_0 V_y \\ V_x &= (\Delta W_y + f_u - \Delta W_z f_v)\,; & A_x &= \Delta X + Z_0 V_x \end{aligned} \tag{10}$$

Then, $\hat{m}_t$, m_0 and $\hat{O}_f$ in (9) can be expressed as:

$$\begin{aligned} \hat{m}_t &= \left[f \frac{A_x}{A_z}, f \frac{A_y}{A_z}, 1, 1 \right]^T; \\ m_0 &= [u_0, v_0, 1, 1]^T; \qquad \hat{O}_f = [S_x, S_y, 0, 0]^T \end{aligned} \tag{11}$$

The derivative of $\hat{m}_t$ with respect to Z_0 can be written as:

$$\frac{\partial \hat{m}_t}{\partial Z_0} = [M_x, M_y, 0, 0]^T \tag{12}$$

where

$$M_x = \frac{fV_x}{A_z} - fV_z \frac{A_x}{A_z^2}; \qquad M_y = \frac{fV_y}{A_z} - fV_z \frac{A_y}{A_z^2} \tag{13}$$

From the above equations it can be concluded that the value of the derivative of the cost function depends only on the initial point coordinates (u_0, v_0), the variation of the camera pose $(\Delta X, \Delta Y, \Delta Z, \Delta W_x, \Delta W_y, \Delta W_z)$ and the measured optical flow (S_x, S_y) in the initial image pixel.

Under these conditions, depth estimation has been converted into many independent optimisation problems (one for each image pixel). This fact conditions the method of optimisation to use. Even though simple stochastic gradient descent (SGD) could solve it, it would be necessary to define a different learning rate for each optimisation problem since each pixel from the initial image is independent of the rest. Probably this learning rate could depend on the real Z value corresponding to each pixel. Consequently, gradient-based methods that work at a constant learning rate are discarded. The learning ratio must be adapted in each iteration for each pixel. Another aspect to consider is the noise in the signals for the gradient calculation. Due to the estimation method, the optical flow has an inherent variability especially in areas where there is an absence of texture. In addition, the position increase is estimated from self-perception data which may also present some noise.

A gradient descent method that can deal with these two issues to successfully compute Z_0^*, is the ADADELTA method [32]. This algorithm is based on SGD with and adaptive filter, but it also introduces several filters in the estimation of the gradient and second derivatives. These filters can reduce the noise influence.

C. Depth estimation algorithm

Algorithm 1 implements the above mathematical formulations inspired by the fixation process. The starting point is the reference image (I_0) and camera pose (C_0) captured at the time of the initial fixation process. Initially, no depth information is available, therefore all pixels in the image are assigned the same value Z_n. When the fixation process has begun, the movements of the head and the oculomotor system generate displacements in the image (I_t) and in the camera pose (C_t). That is, the microsaccades used to carry out the sampling. The initial image I_0 is correlated with each new obtained image I_t using the Lucas-Kanade method [21]. From here, the algorithm iterates for each image pixel, updating the gradient descent computation with the ADADELTA equations.

As the algorithm advances, the received information increases the sense of depth in the image that corresponds to the initial fixation point. Ultimately, this increase in information is represented in the algorithm by the term $\Delta G_t(i, j)$, which in turn depends on the cost function according to (9). Thus, if there is no optical shift between the current image and the reference image ($I = I_0$), there is no improvement in depth estimation knowledge.

Algorithm 1 Depth estimation

```
Require: $I_0$,$Z_n$, $\rho$, $\sigma$, $C_0$
$Z_0^{(0)} \leftarrow Z_n$, $t \leftarrow 0$, $h, w \leftarrow size(I_0)$;
$G_0 \leftarrow zeros(h, w)$; $\Delta G_0 \leftarrow zeros(h, w)$
loop
    $C \leftarrow HeadEyeMovement()$
    $I \leftarrow getNewImage()$
    $OF \leftarrow OpticFlow(I_0, I)$
    $S, T \leftarrow getEgomotion(C_0, C)$
    for $i = 1$ to $h$ do
        for $j = 1$ to $w$ do
            $v \leftarrow i - h/2$; $u \leftarrow j - w/2$
            $g_t(u, v, S, T, OF(i, j)) \leftarrow \frac{\partial L}{\partial Z_0}$        // Eq. 9
            $G_t(i, j) \leftarrow \rho G(i, j)_{t-1} + (1 - \rho) g_t^2$
            $\tau_t \leftarrow \sqrt{\Delta G(i, j)_{t-1} + \sigma} / \sqrt{G(i, j)_t + \sigma}$
            $\Delta G(i, j)_t = \rho \Delta G(i, j)_{t-1} + (1 - \rho) \tau_t^2$
            $Z_0^{(t)}(i, j) = Z_0^{(t-1)}(i, j) + \Delta G_t(i, j)$
            $t \leftarrow t + 1$
        end for
    end for
end loop
```

From a computational complexity point of view, each pixel is visited once in each iteration, as shown in algorithm 1, and the computations made on each pixel only depend on the state of that pixel in the previous step, the optical flux estimated on this point and the camera displacement. Therefore, the temporary asymptotic cost in this part of the algorithm is $\mathcal{O}(\mathcal{N})$, where $\mathcal{N}$ is the total number of pixels in the image. Regarding the asymptotic spatial cost, for the complete algorithm it is necessary to store the resulting depth image, the optical flux components in each iteration, the initial image and the current image. Therefore, the spatial cost has a magnitude of $\Theta(5\mathcal{N})$. This algorithm is amenable to parallel computing, since each pixel is independent of the previous and current states of the rest of the pixels. This allows it to be implemented using parallel computing techniques on either GPUs or CPUs.

D. Algorithm parameters

As it can be seen in the description of Algorithm 1, it is necessary to set a number of parameters for its proper execution: Z_0 is the initial distance for all image pixels; ρ acts as the coefficient of a low pass filter for the gradient adaptation and its derivative, and σ regulates the gain of the gradient variation in each step. Due to the fact that gradient descent techniques do not differentiate between local and global minima, the selection of these parameters is important to obtain good quality results.

In addition, if the span of the work area is known, the limits of the search can be defined a priori; if the sought minimum lies outside these limits, the algorithm will not converge. Also, the noise factor affects its performance to the effect that the values it generates may be outside these limits. In such cases, it is necessary to define an action policy for the pixels in which this phenomenon occurs.

III. Experiments

A. Experimental setup

Evaluation tests are carried out with the Baxter robot in the Gazebo/ROS simulator. Given the degrees of freedom of the Baxter's head it is not possible to replicate with it the movements of the primate's oculomotor system. Instead, we use the 7-DOF arm of this robot with an eye-in-hand camera.

Although some robotic systems described in the literature could perform this task correctly [18], [28], the design of the experiments based on this specific platform was developed in the context of the RoboPicker [6] project for which a low-cost robot is called for and manipulation takes place in a confined space; this will also allow for future experiments with this real system. The basic function of the arm in our experiments is to move the camera in such a way that it maintains orientation and it positions itself in the same way that a human eye would perform fixational movements.

Baxter's wrist camera can be configured in several ways. Of all the possible ones, we chose a resolution of 900x600 pixels and a focal length of 405.7. We set up the same parameters for the camera simulation. In addition, we added white gaussian noise to the image in order to introduce uncertainty in the optical flow computation. This value is common to all performed experiments and is equal to 0.01 pixels.

The space of movements for the camera is specified as a sphere defined by two parameters: the central point and the radius of movements r_m (see Fig.1). r_m is considered constant in order to reduce the number of experiments, with a value of 0.015 m according to the order of magnitude in the experiments of Aytekin and Rucci [4] (these authors suggest r_m is not uniform and its value depends on the distance to the fixation point). A controversial point in the literature is the maximum radius of a microsaccade. Some studies set this value between 1^o and 2^o. However, most microsaccades have a magnitude smaller than 0.5^o for many tasks [27]. The parameter r_g is defined by the microsaccade amplitude as shown in Fig.1. Taking an amplitude of 0.5^o, r_g varies with the fixation point distance as:

$$r_g \approx 0.0088 \cdot d \tag{14}$$

B. Experimental procedure

Based on the fixation process, we define a procedure that is used in all experiments:

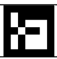
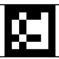
- An artificial scenario is placed in front of the wrist camera of the Baxter robot. (Fig.3a) and a starting point of the camera for the fixation process is selected.
- In order to simplify, three distances are selected for the fixation point in the scenario, all of them on the same axis Z from the camera. In addition, the microsaccade radius r_g is computed as a function of that distance (14).
- The initial image I_0 and pose C_0 are saved.
- The camera starts to move randomly within a sphere of radius r_m maintaining the fixation point projected onto the image plane within the circle of radius r_g.
- The successive images (I_t) and poses (C_t) are compared with the initial image and pose by applying Algorithm 1.

C. Evaluation methods

We consider two kind of scenarios to test Algorithm 1. The first scenario is used to evaluate the accuracy of depth estimation and the influence of the algorithm parameters. The performance of the algorithm is evaluated using two kinds of backgrounds. The first one uses the depth image generated by the simulator (Fig.3c), as the most accurate depth estimation ground truth. The second background utilises 6 squared plates placed in the simulated scenario on which 6 Aruco Markers are printed [10]. The type of markers and their relative position with respect to the initial camera location are shown in table I and Fig.3b. The error is estimated from the standard deviation after 30 measures for each marker position using the Aruco markers detector algorithm. These error values provide information about the repeatability of the measurement, not its accuracy with respect to the background. Using the Aruco markers, we can estimate the depth of each marker plane. In addition, each marker encloses an image area where the depth should be approximately the same. Therefore, applying this mask to the obtained depth image and computing the mean and standard deviation for each marker area, the result must be comparable to the distance estimated by the Aruco detector.

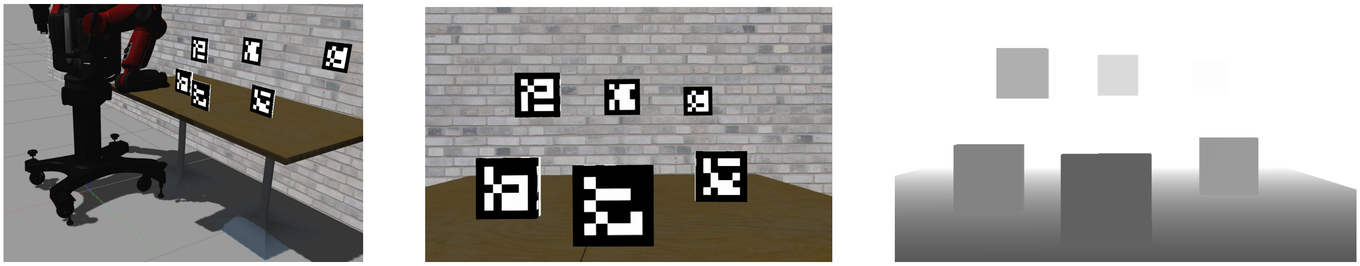
The second evaluation scenario is composed of a number of simulated objects with different shapes and textures in the same setup. Then, the obtained depth image is compared in each iteration with the real one using the mean square error between them. Several Aruco markers are also introduced in this scenario to be used as control points.

TABLE I: Aruco markers used and estimations of the depth from the camera with the Aruco detector.

Marker 101 	Marker 201	Marker 301
$(0.509 \pm 0.006)m$	$(0.386 \pm 0.005)m$	$(0.605 \pm 0.004)m$
Marker 401	Marker 501	Marker 601 
$(0.694 \pm 0.003)m$	$(0.866 \pm 0.002)m$	$(1.071 \pm 0.009)m$

D. Experimental tests

We present several experiments that have as primary objective the evaluation of the proposed algorithm. As additional secondary goals we intend: i) to study the influence of the choice of parameters on the performance and results of the adaptive process. ii) To evaluate the effect of a plausible Gaussian error in the inputs of the algorithm. iii) To validate the algorithm in an environment with ordinary objects.To avoid shifts in the image due to changes in perspective and


(a)

(b)

(c)

Fig. 3: Experimental setup. a) Relative position between Baxter robot and scenario. b) Layout of markers in the test scenario. c) Depth image generated by the simulator, using a virtual depth camera with the same intrinsic parameters than the RGB camera, and placed in the same position.

to keep the set of control markers within the scene in all images, three virtual fixation points were selected at different distances from the initial position of the camera –which is the same for all experiments– $d = \{0.3, 0.6, 0.9\}(m)$. We try to avoid any interference produced by the choice of fixation points within the environment. In this way, it can be assured that all Aruco markers will appear in almost all images and therefore it is possible to track and compare with them in each iteration. Considering that the final objective is to obtain a depth estimation as similar as possible to the image generated by the simulation of the depth camera, two of the criteria used to evaluate the results are the structural similarity index (SSIM) [33] and the global mean square error (MSE), along with the standard deviation (STD) between the depth images in each iteration. To check whether there exist differences in the performance of the algorithm depending on the depth, the comparison between the estimation of the distance in the planes defined by the aruco markers and the one estimated by the algorithm in each iteration is used. Moreover, since the exact position of each plane corresponding to each marker is known, this value is compared to the estimation of the markers and the results of the algorithm.

In addition, we defined a policy regarding how to proceed when the estimated value of the distance lies outside the defined limits of the work area. This can occur when there is an error in the optical flow estimation or in the position variation. One option was to reset its value to the initial distance or, alternatively, to decide not to adapt the value of Z^*_t. After several tentative tests, this second policy was implemented.

1) Influence of the choice of parameters: It can be observed from the adaptive part of the proposed algorithm that ρ acts as the smoothing coefficient in an exponential mean filter, both for the gradient square and $\Delta G(i,j)_t$ adaptation. The choice of ρ must be modulated by the possible noise that the estimation of the gradient and its derivative may present. It can be assumed that this noise has a similar effect on all depth image pixels, therefore the value of ρ is taken as the same for all of them.

The σ parameter –as defined by Zeiler [32]– has a regularisation function to prevent a zero value for the denominator of the τ_t estimate. Its importance changes depending on the relative value of the estimation of the square of the gradient with respect to the σ value.

To study the influence of both parameters, we fix the rest of system variables. Thus, the fixation point is placed at 0.6 m; 0.1 m is assigned as the initial value of Z for all pixels in the depth image; and, finally, the displacements of the camera and RGB images are the same for all variations of the studied parameters. Under these conditions, we fix the value of ρ and vary σ and vice versa. The considered parameter values are $\sigma = \{0.001, 0.005, 0.01, 0.05\}$ and $\rho = \{0.4, 0.5, 0.7, 0.9, 0.99\}$. An example of the obtained results for these tests is shown in Fig.4 and 5. From the analysis of these plots, it is apparent that when σ is kept constant and the value of ρ is changed, the final MSE and STD is similar for all cases (in Fig.4, the mean of the last 50 iterations is 0.0169 ± 0.0787m). Also, these results suggest a

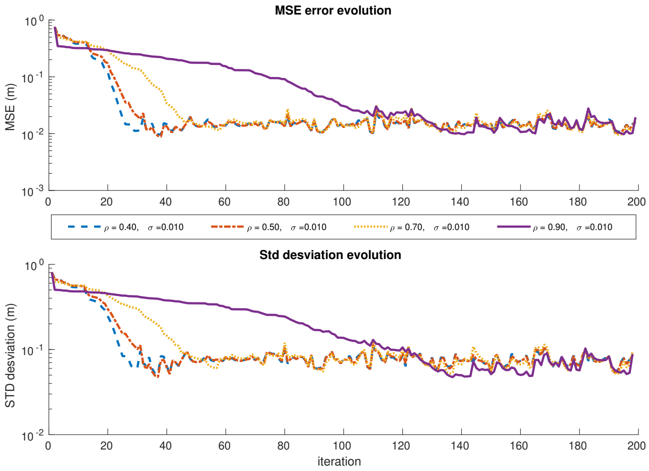


Fig. 4: MSE and STD deviation between the background depth image and the estimation made by the proposed algorithm for 60 cm fixation point and different values of ρ for a constant value of $\sigma = 0.01$

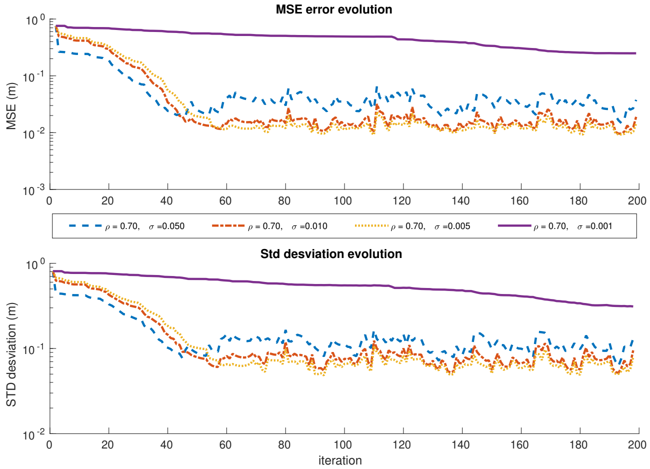


Fig. 5: MSE and STD deviation between the background depth image and the estimation made by the proposed algorithm for 60 cm fixation point and different values of σ for a constant value of $\rho = 0.70$

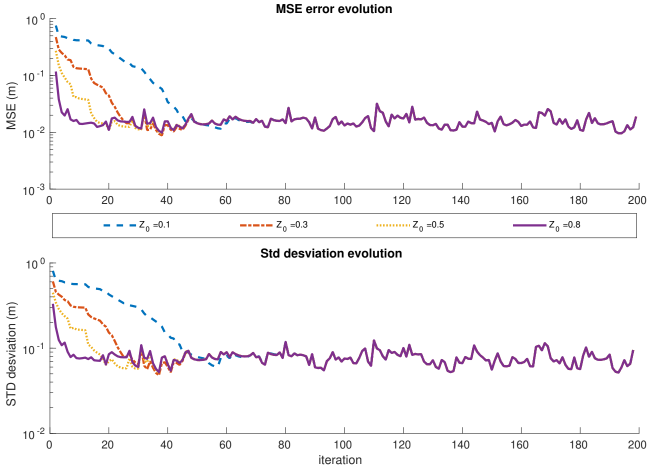


Fig. 6: MSE and STD deviation between the background depth image and the estimation made by the proposed algorithm for 60 cm fixing point and different values of Z_0 for constant values of $\rho = 0.7$ and $\sigma = 0.01$

behaviour for the influence of ρ for a constant σ, in the sense that the lower ρ is, the faster the algorithm converges (around 30 iterations for $\rho = 0.4$).In principle it seems that the lower σ is, the better the obtained results are (Fig.5). This trend, however, has a limit and for a very low sigma, the results are poor.

In addition, we studied the influence of the initial value of Z on the algorithm results. Thus, we fixed the rest of parameters and varied the value of Z_0. The obtained results are shown in Fig.6. It can be checked that the convergence to the final result seems faster the higher the value of Z_0.

2) Influence of noisy input signals: Even though the gradient descent algorithm takes advantage of parameters ρ and σ to filter in some way the noise in the input signals, still the estimation of the gradient and its derivative is severely affected by this noise. To asses the impact of this issue, we introduced in our experiments a gaussian error in the image that directly affects the precision in obtaining the optical flow. This error acts on each pixel individually, whereas the error in the estimation of the displacement of the camera affects the calculation of depth in all pixels.

From this point of view, we used the same experimental conditions, that is, radius of movements ($r_m = 0.015$m); fixation point at distance 0.6 m; same captured RGB images and camera displacements. However, in each iteration we disturb the camera displacement computations with white gaussian noise affecting its rotational and translational components, and characterised by standard deviations ϕ_r and ϕ_t. The chosen values for ϕ_t are $\phi_t = \{0.0001; 0.00050; 0.0010\}$ m that represent $\{1.2\%, 6.6\%, 13.2\%\}$ of the maximum possible displacement respectively. Moreover, the selected values for ϕ_r are $\phi_r = \{0.005°, 0.1°, 0.3°\}$. After the execution of the algorithm, the obtained results are shown in Fig.7. Gray-scale representations of the final depth images for the best and worst cases are shown in Fig.8.

3) Aruco marker comparison: The purpose of using Aruco markers in the simulation is twofold. First, to create surfaces where the distance to the camera is known, and second, to build a scenario that is easy to test when moving from simulation to reality. In simulation it is straightforward to compare the results since the distances from the markers to the camera frame are perfectly known. It is also possible to check the predictions that the Aruco algorithm makes for these known distances. In order to test these errors we applied our algorithm in the same scenario but only varying the distance of the fixation point, and keeping the rest of the parameters constant for all the tests. In table II, the obtained results are shown. The first column lists the relative errors between the estimation of the algorithm and the distance predicted from Aruco markers. The second column compares the relative errors between the estimation of the algorithm and the distance given by the simulator. A graphical example is shown in Fig.9.

4) Environment with ordinary objects: So far, only a simplified scenario has been considered, which has made it possible to evaluate the accuracy of depth estimation as well as the influence of the algorithm parameters and noise. All the surfaces involved were planes perpendicular to the camera. In order to test the effectiveness of the algorithm in other more complex environments, models of several objects have been chosen [26] and arranged in front of the camera in the same way as the markers. The RGB image for this scenario is shown in Fig.10a. There are various types of objects in terms of shape, texture and transparency. the corresponding depth image as generated by the simulator is shown in Fig.10b; it will be used as reference ground truth image.

To evaluate the results we used MSE as we did before. Due to the fact that MSE can yield misleading results in certain circumstances, SSIM was also used. SSIM provides us with

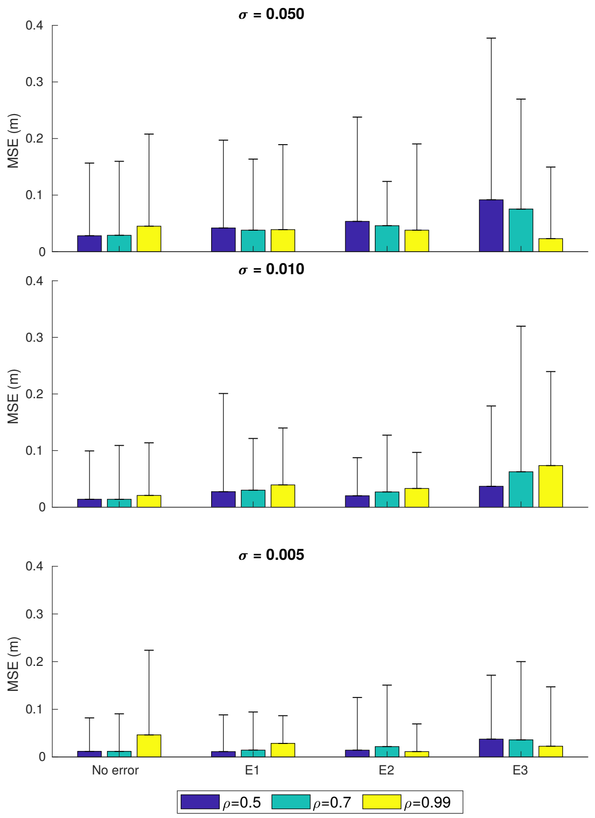


Fig. 7: MSE with error bars representing STD of the last 20 iterations. The bars are grouped by the added white noise. $E_i = \{\phi_t(i), \phi_r(i)\}$.

information about the structural similarity between the depth image generated by the simulator and that estimated by the algorithm. For these tests, the parameters were given the values for which the best results in Fig.8 were obtained without white noise error. Namely, $\rho = 0.5$ and $\sigma = 0.005$. The evolution of MSE and STD are shown in Fig.11b with the expected behaviour. Fig. 11a illustrates the evolution of SSIM index along the whole adaptive process. The range of possible values for SSIM extends from 0 to 1, being more similar the closer it is to 1. In this case, the variability of that index oscillates between 0.75 and 0.85 at the end of the algorithm iteration for all selected fixation points, comparing with the ideal depth image represented in Fig.10b.

IV. Discussion

The results of the above experiments allow us to asses the robustness of the algorithm in relation to the addition of noise, as well as the influence of the parameters on its performance. In contrast to [7], here we evaluate the algorithm more thoroughly. For this reason, MSE and STD are used to measure the average error over the whole image. Using the simulator makes possible to have a perfect background in order to compute MSE and STD. We can also compare the results of our algorithm with those generated by the Aruco Markers in the simulation.

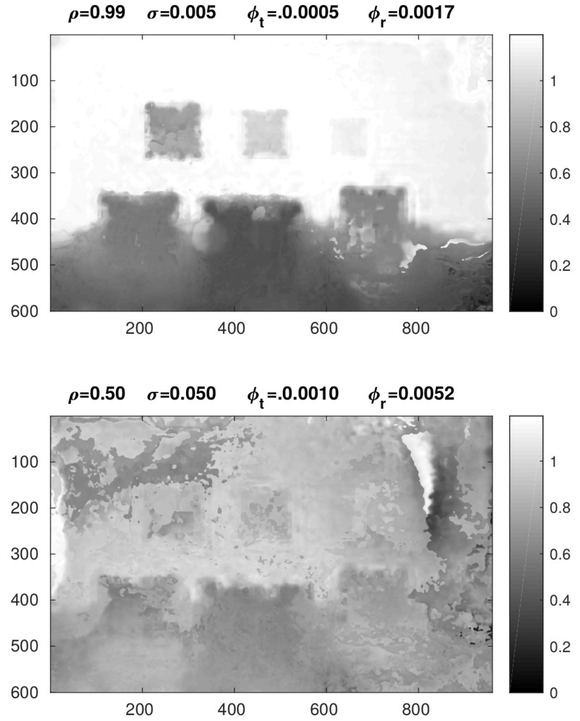


Fig. 8: Image depth, showing the distance for each pixel scaled in 0-255 range. The upper image corresponds to the best obtained MSE and on the bottom the worst MSE for all experiments where white noise error was added to the camera displacement.ϕ_r is expressed in radians and ϕ_t in meters

1) Parameter selection: In view of the results shown in Fig.4 and Fig.5, it is apparent that, as expected, ρ parameter acts as a filter causing the stabilisation of the final results in exchange for the number of iterations to reach them. On the other hand, the behaviour of σ is more complex. As it can be seen in Fig.5, for a given ρ the lower the value of σ the better the overall result. However, if σ becomes too small, the algorithm gets *frozen* (purple line in Fig.5). This is also the case for too high values of ρ (green line in Fig.4). In any case the choice of σ and ρ should be made jointly since the closer ρ is to 1 –and, therefore, filters more– the higher the value of σ should be. Finally, the initial value Z_0 only conditions the moment of reaching a more or less stable result, as shown in Fig.6, but it does not seem to have an effect on the final depth image.

2) Noise addition: By adding a Gaussian error to the estimation of the camera position, that affects equally each pixel of the final depth image, we are pushing the application of the algorithm to the limit. Notwithstanding, it is in this case when the effects of σ and ρ become more apparent.

TABLE II: Relative errors between depth estimated by the algorithm and the distance predicted from Aruco markers, and between the estimation of the algorithm and the distance given by the simulator, for the six Aruco markers and three fixation distances.

	d=30 cm	
	Aruco difference	Simulator difference
M.201	$(0.57 \pm 0.60))$ %	$(0.83 \pm 0.68))$ %
M.101	$(0.57 \pm 0.42))$ %	$(1.64 \pm 0.42))$ %
M.301	$(0.35 \pm 0.07))$ %	$(0.33 \pm 0.07))$ %
M.401	$(0.97 \pm 0.10))$ %	$(0.12 \pm 0.09))$ %
M.501	$(0.61 \pm 0.07))$ %	$(1.04 \pm 0.07))$ %
M.601	$(2.79 \pm 0.07))$ %	$(1.41 \pm 0.08))$ %
	d=60 cm	
	Aruco difference	Simulator difference
M.201	$(2.30 \pm 0.24))$ %	$(0.52 \pm 0.22))$ %
M.101	$(0.46 \pm 0.17))$ %	$(1.16 \pm 0.17))$ %
M.301	$(0.62 \pm 0.09))$ %	$(0.38 \pm 0.09))$ %
M.401	$(1.05 \pm 0.04))$ %	$(0.28 \pm 0.04))$ %
M.501	$(1.82 \pm 0.03))$ %	$(0.24 \pm 0.03))$ %
M.601	$(5.10 \pm 0.13))$ %	$(2.98 \pm 0.14))$ %
	d=90 cm	
	Aruco difference	Simulator difference
M.201	$(1.75 \pm 1.61))$ %	$(1.70 \pm 1.54))$ %
M.101	$(0.77 \pm 0.21))$ %	$(0.75 \pm 0.21))$ %
M.301	$(0.59 \pm 0.59))$ %	$(0.59 \pm 0.59))$ %
M.401	$(0.56 \pm 0.09))$ %	$(0.09 \pm 0.04))$ %
M.501	$(0.48 \pm 0.17))$ %	$(0.95 \pm 0.17))$ %
M.601	$(2.36 \pm 0.11))$ %	$(0.10 \pm 0.08))$ %

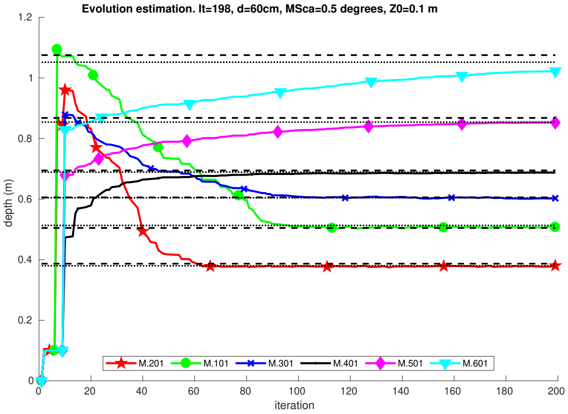


Fig. 9: Example of the evolution of depth estimate mean for the Aruco regions only, wit a fixation point at 60 cm. The dotted lines are the distances of the markers given by the simulator, and the dashed lines are the distances estimated by the Aruco algorithm.

These experiments serve also the purpose of establishing the error limits when applying the algorithm to a real robot. Our results raise some points for discussion. i) The lower σ is, it seems that the more robust the performance is in all cases; ii) increasing ρ tends to stabilise the algorithm results in some cases, depending on the value of σ, and with a limit (as in Fig.4 where for a value of $\rho = 0.99$ the algorithm hardly progresses); iii) the uncertainty when the added noise error is too high generates non-valid results. As it can be seen in Fig.8 in a qualitative way, the added error has a manifest influence in the quality of the results.

3) Aruco markers comparison: As suggested in [20], the accuracy of the Aruco Markers decays with distance. Table II shows that the greater the distance from a given marker, the results generated by the algorithm are closer to the simulator ground truth than to the values provided by the markers. This suggests that for this particular case the proposed algorithm is less sensitive to error variation with distance than the Aruco markers.

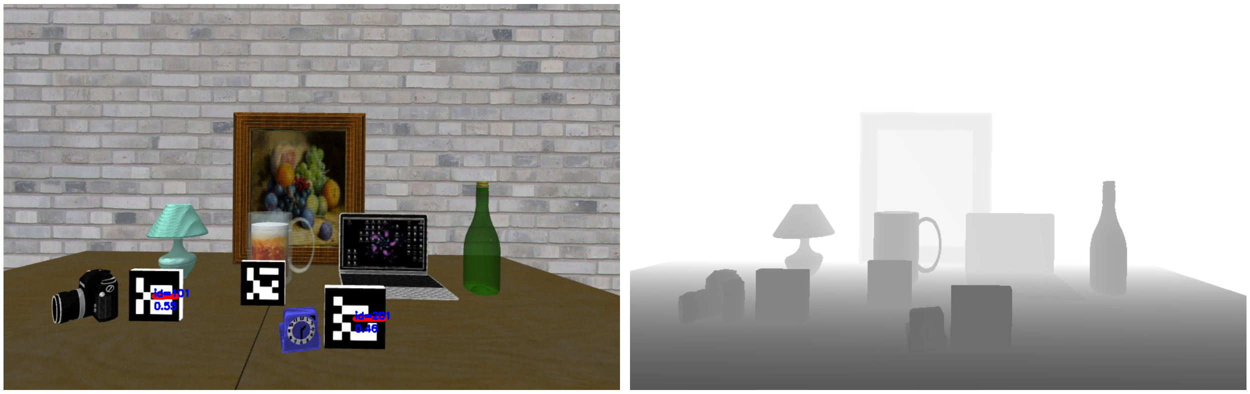
4) Real object simulation: Both the numerical results obtained from the analysis of Fig.11a and Fig.11b as well as the qualitative results derived from Fig.12 show that the proposed algorithm is able to determine the depth image of a more complex scenario. Thus, the evolution of SSIM and MSE is analogous and reaches the convergence value at iteration 40 for all cases. It is remarkable that SSIM reaches a value of about 0.80 which indicates a very high structural similarity with the reference image. It is also important to highlight the behaviour of the algorithm with respect to non-textured objects such as the night lamp or the camera for which the determination of the optic flow involves more difficulties. It is also noteworthy the behaviour for semi-transparent objects –such as the wine bottle or the beer mug handle, where the algorithm gives good results which could not be obtained, for instance, with standard depth sensors.

V. Conclusion and Future Work

In this work, we have stated several hypotheses based on monocular human visual fixation. We have proposed a model that from an initial image and camera pose is able to estimate its depth map by considering the optical displacement in the images induced by the different poses of the camera as a consequence of eye-head movements inspired by those involved in human fixation, namely, microdisplacements of the head and microsaccadic movements. It is important to highlight the fact that our algorithm is agnostic to the specific robot hardware as long as it is able to replicate the described 3D camera fixation movements. In consequence, our conclusions can be extended to other robotic platforms since it is only necessary to know the pose of the camera at each instant from the robot proprioception, and compare it with the initial pose. We have studied the behaviour of the proposed algorithm in several scenarios in order to quantify its stability with respect to noisy input signals and how its parameters influence its performance, both qualitatively and quantitatively. Our good results pave the way for the implementation in a real robot.

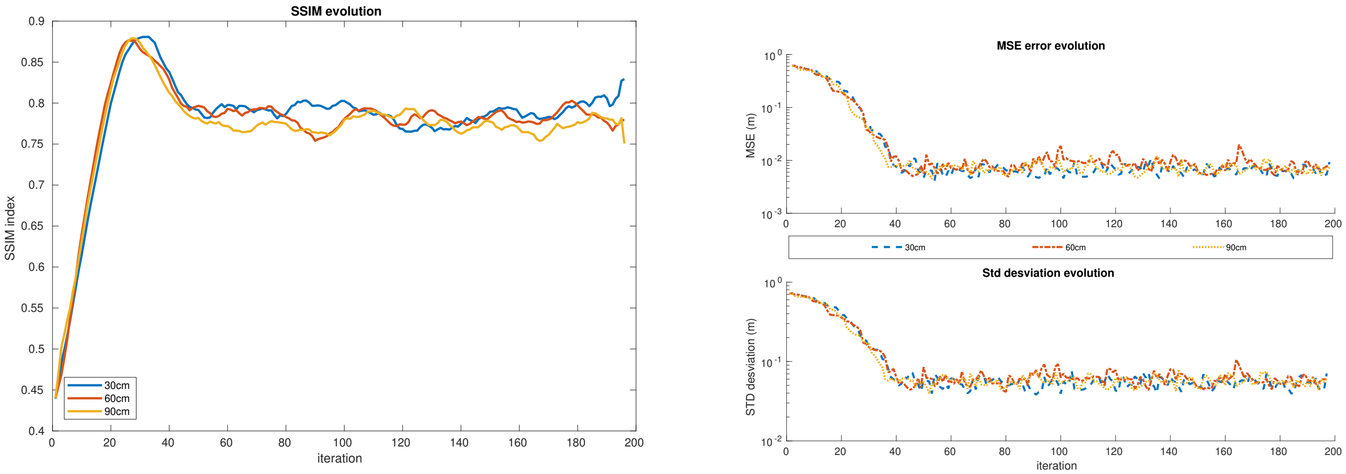
Acknowledgement

This paper describes research done at UJI Robotic Intelligence Laboratory. Support for this laboratory is provided in part by Ministerio de Economia y Competitividad (DPI2015-69041-R) and by Universitat Jaume I (UJI-B2018-74).



(a) (b)

Fig. 10: Layout for the experiments with ordinary objects. a) RGB image of the scenario. b) Ground truth depth image



(a) (b)

Fig. 11: Evolution of the algorithm results for fixation points at 30, 60 and 90 cm. a) SSIM evolution. b) MSE and STD evolution

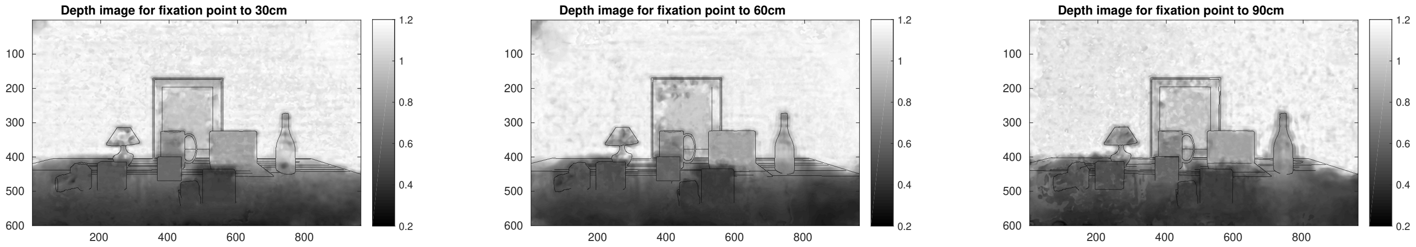


Fig. 12: Gray-scale experimental results for the scenario with ordinary objects and fixation points at 30, 60 and 90 cm.

References

[1] M. Antonelli, A. P. del Pobil, and M. Rucci. Depth estimation during fixational head movements in a humanoid robot. In *International Conference on Computer Vision Systems*, pages 264–273. Springer, 2013.

[2] M. Antonelli, A. P. del Pobil, and M. Rucci. Bayesian multimodal integration in a robot replicating human head and eye movements. In

2014 IEEE International Conference on Robotics and Automation, pages 2868–2873, May 2014.

[3] M. Antonelli, M. Rucci, and B. Shi. Unsupervised learning of depth during coordinated head/eye movements. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5199–5204, Oct 2016.

[4] M. Aytekin and M. Rucci. Motion parallax from microscopic head movements during visual fixation. *Vision Research*, 70(617):7–17, 2012.

[5] E. Chinellato, B.J. Grzyb, and A.P. del Pobil. Pose estimation through cue integration: a neuroscience-inspired approach. *IEEE Transactions on Systems, Man and Cybernetics -Part B: Cybernetics*, 424(2):530–538, 2012.

[6] A.P. del Pobil et al. Uji robinlab's approach to the amazon robotics challenge 2017. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 318–323. MFI, 2017.

[7] A. J. Duran and A. P. del Pobil. Improving robot visual skills by means of a bio-inspired model. In *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 25–30, Aug 2019.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[9] N. Sünderhauf et al. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

[10] S. Garrido-Jurado et al. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280 – 2292, 2014.

[11] C. Fantoni, C. Caudek, and F. Domini. Systematic distortions of perceived planar surface motion in active vision. *Journal of Vision*, 10(5):12–12, 2010.

[12] Ziad M Hafed and Richard J Krauzlis. Microsaccadic suppression of visual bursts in the primate superior colliculus. *The Journal of neuroscience*, 30(28):9542–9547, jul 2010.

[13] Jungong Han and et al. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[14] Janis Intoy and Michele Rucci. Finely tuned eye movements enhance visual acuity. *Nature Communications*, 11(1):1–11, 2020.

[15] A. Jain and B. T. Backus. Experience affects the use of ego-motion signals during 3D shape perception. *Journal of vision*, 10(14):30, dec 2010.

[16] Hee Kyoung Ko, Martina Poletti, and Michele Rucci. Microsaccades precisely relocate gaze in a high visual acuity task. *Nature Neuroscience*, 13(12):1549–1554, 2010.

[17] Richard J. Krauzlis and et al. Neuronal control of fixation and fixational eye movements. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1718), 2017.

[18] Xutao Kuang, Mark Gibson, Bertram E. Shi, and Michele Rucci. Active vision during coordinated head/eye movements in a humanoid Robot. *IEEE Transactions on Robotics*, 28(6):1423–1430, 2012.

[19] Bangli Liu, Haibin Cai, Zhaojie Ju, and Honghai Liu. RGB-D sensing based human action and interaction analysis: A survey. *Pattern Recognition*, 94:1–12, 2019.

[20] Alberto López-Cerón and José M Canas. Accuracy analysis of marker-based 3d visual localization. In *XXXVII Jornadas de Automatica Workshop*, 2016.

[21] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[22] Larry Matthies, Richard Szeliski, and Takeo Kanade. Kalman Filter-based Algorithms for Estimating Depth from Image Sequences. *Multisensor Fusion for Computer Vision*, 236:87–130, 1993.

[23] Jacob W Nadler, Dora E Angelaki, and Gregory C DeAngelis. A neural representation of depth from motion parallax in macaque visual cortex. *Nature*, 452(7187):642, 2008.

[24] Klaus Haming Petersl and Gabriele. the Structure-From-Motion Reconstruction Pipeline – a Survey With Focus on Short Image Sequences. *Kybernetika*, 46(5):926–937, 2010.

[25] Matteo Poggi and et. al. Towards real-time unsupervised monocular depth estimation on cpu. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5848–5854. IEEE, 2018.

[26] Amir Rasouli and John K Tsotsos. The effect of color space selection on detectability and discriminability of colored objects. *arXiv preprint arXiv:1702.05421*, 2017.

[27] Martin Rolfs. Microsaccades: Small steps on a long way. *Vision Research*, 49(20):2415–2441, 2009.

[28] Fabrizio Santini, Rohit Nambisan, and Michele Rucci. Active 3D vision through gaze relocation in a humanoid robot. *International Journal of Humanoid Robotics*, 6(3):481–503, 2009.

[29] Fabrizio Santini and Michele Rucci. Active estimation of distance in a robotic system that replicates human eye movement. *Robot. Auton. Syst.*, 55(2):107–121, 2007.

[30] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009.

[31] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-Motion Revisited. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[32] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[33] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

Angel J. Duran was born in Castellon, Spain in 1974. Received the B.Sc. in Chemistry (1997,Spain) and achieved the Degree in Technical Engineering in Information Management ,Computer Science with honors(2011,Spain) by Universitat Jaume I. He achieved the M.Sc. in Intelligence Systems in Computer Science (2012, Spain).In December 2010,he started to collaborate with the UJI Robotic Intelligence Lab. Since then, he has participated in several research projects like: Group Of Un-manned Assistant Robots Deployed In Aggregative Navigation Supported By Scent Detection (GUARDIANS) and Heterogeneous 3D Perception Across Visual Fragments(EYESHOTS). Nowadays, he is pursuing his Ph.D.at the Robotic Intelligence Lab.His research interests include, machine learning, transformations and robotic system software architectures.

Angel P. del Pobil Received the B.Sc. degree in physics and the Ph.D. degree in engineering from the University of Navarra, Pamplona, Spain, in 1986 and 1991, respectively. He is currently a Professor with Jaume I University (UJI), Castellon de la Plana, Spain, where he is the founding Director of the UJI Robotic Intelligence Laboratory. He is a Visiting Professor at Sungkyunkwan University, Seoul, Korea. He is co-Chair of the IEEE RAS Technical Committee on Performance Evaluation & Benchmarking of Robotic Systems, and a member of the Governing Board of the Intelligent Autonomous Systems (IAS) Society and EURON (European Robotics Research Network of Excellence, 2001–2009). He has over 250 publications, including four authored and nine edited books. Prof. del Pobil was co-organizer of some 50 workshops and tutorials at ICRA, IROS, RSS, ROMAN, IJCNN and HRI. He has been Program or General Chair of international conferences such as Adaptive Behaviour (SAB 2014), or Artificial Intelligence and Soft Computing. He serves regularly as Associate Editor for ICRA and IROS, and on the program committee of over 150 international conferences, such as IJCAI, ICPR, IAS, SAB, ICDL-EPIROB, ROMAN, etc. He has been involved in intelligent robotics research for the last 30 years. Professor del Pobil has been invited speaker of 63 tutorials, plenary talks, and seminars in 15 countries. He serves as associate or guest editor for 12 journals, and has supervised 16 Ph.D. thesis, including winner and finalists of the Georges Giralt PhD Award and the Robotdalen Scientific Award. He has been Principal Investigator of 30 research projects.