Direct 3D servoing using dense depth maps

Céline Teulière, Eric Marchand

***Abstract*— This paper proposes a novel 3D servoing approach using dense depth maps to achieve robotic tasks. With respect to position-based approaches, our method does not require the estimation of the 3D pose (direct), nor the extraction and matching of 3D features (dense) and only requires dense depth maps provided by 3D sensors. Our approach has been validated in servoing experiments using the depth information from a low cost RGB-D sensor. Positioning tasks are properly achieved despite the noisy measurements, even when partial occlusions or scene modifications occur.**

I. INTRODUCTION

Most of the robotic positioning tasks are achieved by estimating first the relative pose between the robot and the scene or the object of interest, and then using a position-based control scheme [21]. However, the pose estimation problem itself is complex in its general formulation. Also known as the *3D localization problem*, this problem has been widely investigated by the computer vision community [6] [13] but remains non-trivial for unknown environments. Using range data, a range flow formulation has been proposed [10][8] to estimate the 3D pose of a mobile robot. Alternatively, the alignement of successive 3D point clouds using ICP [1] [3] has become a very popular method. Many variants have been proposed in the litterature [18] and the development of the so-called RGB-D cameras attracted lot of attention on these methods [20] [16] [9] [17] in the recent years.

In this paper, we propose to perform robotic tasks without reconstructing the full 3D pose between the robot and its environment, but using a sensor-based servoing scheme, the considered data being directly the depth map obtained from a range sensor. Our approach is thus related to other sensor-based methods, such as image-based visual servoing (IBVS) [2], where a robotic task is expressed directly as the regulation of a visual error. In IBVS, the visual error is usually defined as the difference between a current and a desired set of geometric features (points, straigth lines, etc.) selected from the image, to control the desired degrees of freedom. Therefore, IBVS schemes usually require the extraction of visual features from image measurements, and their matching in successive frames. However, those steps, based on image processing techniques, are often considered as the bottleneck of visual servoing methods.

Recently, some work proposed to use all the image directly, without any extraction or matching step, by minimizing the difference between the current image and a reference image. This approach is referenced as *photometric visual servoing* [4]. However, since it is based on the luminance consistancy assumption, it is sensitive to illumination changes. In our work we propose to use the depth map obtained from a range sensor as a visual feature, without any feature extraction or matching step, and to control a robot with this feature directly. Our approach is thus both direct (without any 3D pose estimation) and dense (without feature extraction). To our knowledge this is the first time dense depth information is used in a direct servoing approach.

The remainder of this paper is organized as follows: our dense depth map servoing framework is described in section II. In section III we propose simple solutions to increase its robustness to some practical issues such as incomplete measurements or occlusions. Positioning experiments have been conducted to validate the approach. The results are given and discussed in section IV.

II. DIRECT DENSE DEPTH MAP SERVOING

This section presents the heart of our approach, i.e. how to control a robot using dense depth maps. We first introduce what we call a depth map and what it means to use it as a feature to regulate (section II-A). Then we derive the fondamental equations necessary to compute our control law (section II-B and II-C). In section II-D we underline the main differences between our approach and sparse 3D approaches.

A. Depth map sensing

There are multiple technologies of sensors capable of providing depth information, (or range). Most range sensors without contact are active, and based on the time of flight (ToF) principle: the idea is to send waves of known velocity and measure the time it takes them to go from the sensor and come back after reflection on the scene. This can be achieved by sending light pulses. Another approach would consist in using a modulated signal and measuring the phase shift. In each case, knowing the velocity of the sent signal, the depth information is derived (eg: Laser scans, sonars, radars, ToF or RGB-D cameras).

Another existing technology for active range sensing is based on structured light: known patterns (stripes, dots, ...) are projected onto the scene and the depth information is deduced from their deformation. This technology is used for instance in the recent Microsoft Kinect or Asus Xtion pro devices, based on PrimeSense technology [7].

Depth can also be measured with passive sensors such as cameras: by matching image features in two different views of a calibrated stereo rig, depth can be computed from geometry. The depth information is sparse when a finite set

C. Teulière is with INRIA, Lagadic Project, Rennes, France
E. Marchand is with Université de Rennes 1, IRISA, INRIA, Lagadic Project, Rennes, France firstname.name@irisa.fr

of features are matched, but dense depth maps can also be obtained [19].

In the following, we will assume that we have a range sensor capable of providing dense range measurements. Without loss of generality, we consider that those measurements are converted to depth maps expressed in sensor centered cartesian coordinates. Formally, we denote by $Z(x, y, t)$ the depth of the 3D point of coordinates (X, Y, Z) in the sensor frame, with $X = xZ$ and $Y = yZ$, at time t (see Figure 1).

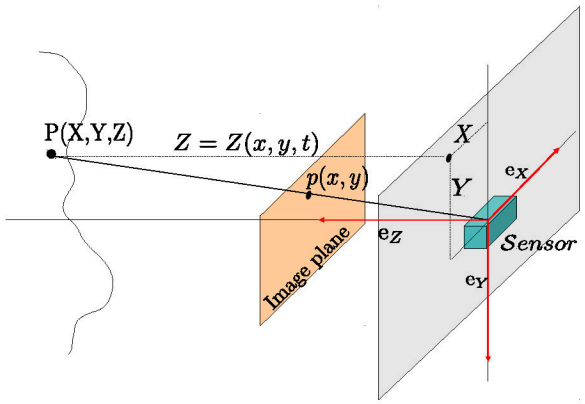

Fig. 1. Sensor frame representation.

The data $(x, y, Z(x, y, t))$ are dense metric depth maps, which can be represented as images, with (x, y) the metric image coordinates. Note that the sensor is assumed to be calibrated so that pixel coordinates can be mapped to metric image coordinates. Figure 2 gives an example of depth map obtained from Microsoft Kinect RGB-D sensor. The depth values have been scaled to greyscale levels. White pixels correspond to unavailable depth values, i.e. pixels where the sensor could not compute any depth information. Note also that for better visualisation purpose, we used histogram equalization on the depth maps we show, throughout the paper.

(a) (b)

Fig. 2. Example of static scene (a) and corresponding depth map representation (b) acquired from Microsoft Kinect sensor. The darkest pixels correspond to the smallest depths. White pixels correspond to unavailable data.

The next section shows how such dense depth maps can be used to control a robot.

B. Control law

Let's consider that a robot end effector is equipped with a range sensor (Figure 3).

Fig. 3. ADEPT Viper robotic system equipped with a Microsoft Kinect sensor.

We express a positioning task as the regulation of the feature $\mathbf{Z}$ to a desired value $\mathbf{Z}^*$. Here, $\mathbf{Z} = (Z_1, ..., Z_N)$ is a vector containing the N depth values corresponding to the current dense depth map. The desired value $\mathbf{Z}^*$ thus corresponds to a reference depth map acquired at the desired robot position.

Therefore, the control law to design aims at regulating the following error to zero:

$$\mathbf{e} = \mathbf{Z} - \mathbf{Z}^* = \begin{pmatrix} \vdots \\ Z_i - Z_i^* \\ \vdots \end{pmatrix} \quad (1)$$

An illustration of such error is given in Figure 4.

In analogy with the visual servoing framework [2] we denote by $\mathbf{L_Z}$ the interaction matrix associated to the feature $\mathbf{Z}$, and characterized by the relation:

$$\frac{\partial \mathbf{Z}}{\partial t} = \mathbf{L_Z}\mathbf{v} \quad (2)$$

where $\frac{\partial \mathbf{Z}}{\partial t}$ is the temporal variation of the depth and $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ is the sensor instantaneous velocity screw.

Assuming that we want an exponential decrease of the error (1), i.e. $\dot{\mathbf{e}} = -\lambda \mathbf{e}$, we then define a proportional control law:

$$\mathbf{v} = -\lambda \mathbf{L_Z^+} \mathbf{e} \quad (3)$$

where $\mathbf{L_Z^+}$ denotes the pseudo-inverse of $\mathbf{L_Z}$:

$$\mathbf{L_Z^+} = (\mathbf{L_Z}^\top \mathbf{L_Z})^{-1} \mathbf{L_Z}^\top. \quad (4)$$

The computation of the matrix $\mathbf{L_Z}$ required in the control law (3) is described in section II-C.

C. Interaction matrix computation

In this section, we derive the expression of the interaction matrix $\mathbf{L_Z}$ which characterizes the relation between the temporal variation of the depth and the sensor's instantaneous velocity screw (see equation (2)). In that purpose, we consider the continuous formulation of the depth map as a surface $Z(x, y, t)$. Assuming that the scene is static and the surface $Z(x, y, t)$ is smooth, taking its full derivative yields to:

$$\frac{dZ}{dt} = \frac{\partial Z}{\partial x}\dot{x} + \frac{\partial Z}{\partial y}\dot{y} + \frac{\partial Z}{\partial t}, \quad (5)$$

where $(\dot{x}, \dot{y})$ is the 2D velocity of the image point (x, y). Equation (5) is known as the *range flow constraint equation*

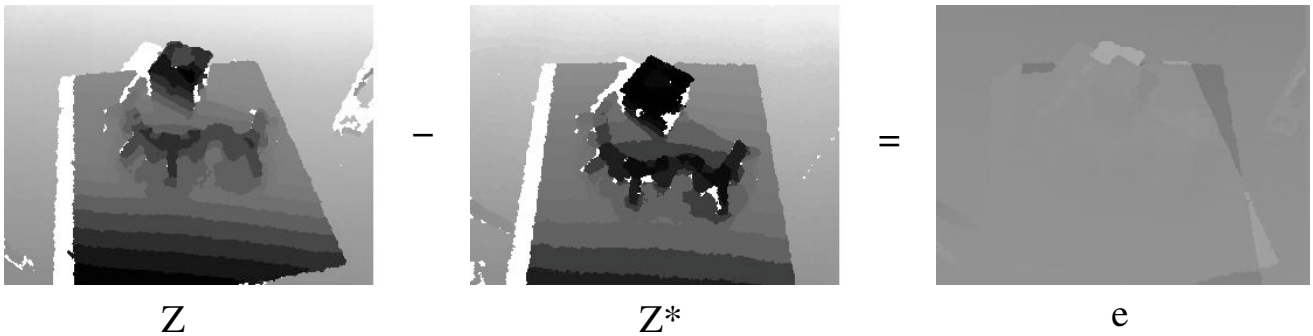

Fig. 4. The task error is the difference of depth maps $\mathbf{Z} - \mathbf{Z}^*$.

[22] or *elevation rate constraint equation* [10]. It is analogous to the *brightness change constraint equation* that is used in the computation of optical flow [11] and used in direct photometric visual servoing methods [4].

From equation (5) we know that the temporal variation of the depth is:

$$\frac{\partial Z}{\partial t} = \dot{Z} - \frac{\partial Z}{\partial x}\dot{x} - \frac{\partial Z}{\partial y}\dot{y}, \tag{6}$$

where $\dot{Z} = \frac{dZ}{dt}$ denotes the Z velocity expressed in the sensor frame.

Therefore, the interaction matrix $\mathbf{L}_Z$ related to one depth value is expressed by:

$$\mathbf{L}_Z = \mathbf{L}_{P_Z} - \frac{\partial Z}{\partial x}\mathbf{L}_x - \frac{\partial Z}{\partial y}\mathbf{L}_y. \tag{7}$$

The matrices $\mathbf{L}_x$, $\mathbf{L}_y$ defined such that $\dot{x} = \mathbf{L}_x\mathbf{v}$ and $\dot{y} = \mathbf{L}_y\mathbf{v}$ are the well-known interaction matrices of image point coordinates, given by:

$$\mathbf{L}_x = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \end{bmatrix} \tag{8}$$

$$\mathbf{L}_y = \begin{bmatrix} 0 & \frac{-1}{Z} & \frac{y}{Z} & -(1+y^2) & -xy & -x \end{bmatrix}, \tag{9}$$

and $\mathbf{L}_{P_Z}$ is the interaction matrix related to the coordinate Z of a 3D point, such that $\dot{Z} = \mathbf{L}_{P_Z}\mathbf{v}$. It is given by:

$$\mathbf{L}_{P_Z} = \begin{bmatrix} 0 & 0 & -1 & -yZ & xZ & 0 \end{bmatrix}. \tag{10}$$

More details on the derivation of those interaction matrices can be found in [2].

The full interaction matrix $\mathbf{L_Z}$ of size $N \times 6$ corresponding to the entire depth map is thus the stack of the 1×6 matrices $\mathbf{L}_{Z_i}$:

$$\mathbf{L_Z} = \begin{bmatrix} \mathbf{L}_{Z_1} \\ \vdots \\ \mathbf{L}_{Z_N} \end{bmatrix}. \tag{11}$$

D. Dense vs sparse 3D servoing

Depth information has already been used in position-based visual servoing. For example, [15] proposed to use the 3D coordinates (X, Y, Z) of a set of 3D points as features to be regulated in a proportional control law. In other words, the positioning task was expressed as the regulation of the feature $\mathbf{P} = (X_1, Y_1, Z_1, ..., X_N, Y_N, Z_N)$ to a reference feature $\mathbf{P}^* = (X_1^*, Y_1^*, Z_1^*, ..., X_N^*, Y_N^*, Z_N^*)$ corresponding to the 3D coordinates of the set of points at the desired robot position. The interaction matrix related to a single 3D point is then given by [2]:

$$\mathbf{L}_P = \begin{bmatrix} -1 & 0 & 0 & 0 & -Z & Y \\ 0 & -1 & 0 & Z & 0 & -X \\ 0 & 0 & -1 & -Y & X & 0 \end{bmatrix}. \tag{12}$$

At first sight, the formulation of this kind of 3D feature $(X_1, Y_1, Z_1, ..., X_N, Y_N, Z_N)$ can seem very close to the vector formulation $\mathbf{Z} = (Z_1, ..., Z_N)$ that we defined in section II-B. However, a key difference with respect to our approach is that [15] uses a sparse set of 3D features. Consequently, in [15] a matching step is required to determine the feature values through the sequence, and the range flow equation (5), based on a smoothness assumption, does not hold in the sparse case. On the contrary, one of the key advantages of the method we propose, is that it does not require any feature extraction or matching step and uses directly the dense depth information from the range sensor thanks to the range flow equation.

III. Practical issues and robustness improvements

In the previous section, we presented our depth map based servoing method. When testing it, we found that this method was efficient in simulation sequences, with perfect data, but we had to face some practical issues in real conditions, in particular, in our case, using a Kinect sensor. This section presents the modifications we had to undertake in order to improve the robustness of the servoing task with respect to noisy and incomplete measurements (section III-A) and to scene perturbations and occlusions (section III-B).

A. Noisy and incomplete measurements

As illustrated in Figure 2-b the depth map acquired by a Kinect sensor is noisy and incomplete. In practice, we only considered the pixels for which a depth value was available both in the reference $\mathbf{Z}^*$ and the current $\mathbf{Z}$ depth maps. This means that the number N of depth values in $\mathbf{Z}$ and (11), is inferior to the size of the depth map (320×240). In the experiments presented in this paper, about 80% of the total number of pixels could typically be used.

In addition, we reduced the noise by applying a simple 3×3 Gaussian filter on the depth maps, the convolution being computed only with the valid neighbors.

Similarly, the spatial gradient was computed using a simple 3×3 derivative kernel taking into account the valid neighbors only.

B. Occlusions and scene modifications

Another issue to take into account is the possibility of partial occlusions or scene modifications during the servoing process. To reduce the effect of such events on the task achievement, we use robust M-estimation [12][5]. We thus introduce a modification of our task objective (1) allowing uncertain measures to be less likely considered or in some cases completely rejected. The new task error is given by:

$$\mathbf{e} = \mathbf{D}(\mathbf{Z} - \mathbf{Z}^*) \tag{13}$$

where $\mathbf{D}$ is diagonal weighting matrix given by: $\mathbf{D} = diag\,(w_1, ..., w_N))$, the weights w_i depending on their distance to the median of the error vector $\mathbf{e}$ according to a robust function [12]. Different functions are possible for the robust estimation. In practice, we used Tukey's estimator to completely reject the least likely values.

Using (13), the new control law becomes [5]:

$$\mathbf{v} = -\lambda(\mathbf{D}\mathbf{L}_\mathbf{Z})^+\mathbf{D}(\mathbf{Z} - \mathbf{Z}^*). \tag{14}$$

Experimental results using this control scheme are presented in the next section.

IV. Experimental results

In this section we present the results of our approach for positioning tasks. A Kinect sensor has been mounted on a ADEPT Viper robot (see Figure 3). In each experiment, the robot first acquires the reference depth map at the desired position. It is then moved to an initial position from which the control scheme is launched, aiming at going back to the desired one. A fixed gain $\lambda = 2.5$ is used in these experiments.

The first experiment illustrates the behavior of our system in a nominal case, namely with a static scene and no occlusions. The depth maps are acquired using the LibFreenect[1] driver through the ViSP library [14], with a resolution of 320×240. The scene is composed of various objects of different shapes and materials (Figure 5 1-a).

The initial and final states are illustrated in Figure 5. The first row shows the RGB views provided by the Kinect for the initial (1-a) and final (1-b) positions. Those images are never used in the control scheme but are useful for a better understanding of the setup. The depth maps are shown in the second row, and the last row gives the corresponding error, i.e. the difference between the desired and the current depth maps, unavailable data being discarded as explained in III-A. The difference images are scaled so that a plain grey frame (3-b) corresponds to a null error, and thus to the good achievement of the task. In Figure 5 (3-a) we see that the initial error was significant.

[1]http://openkinect.org/

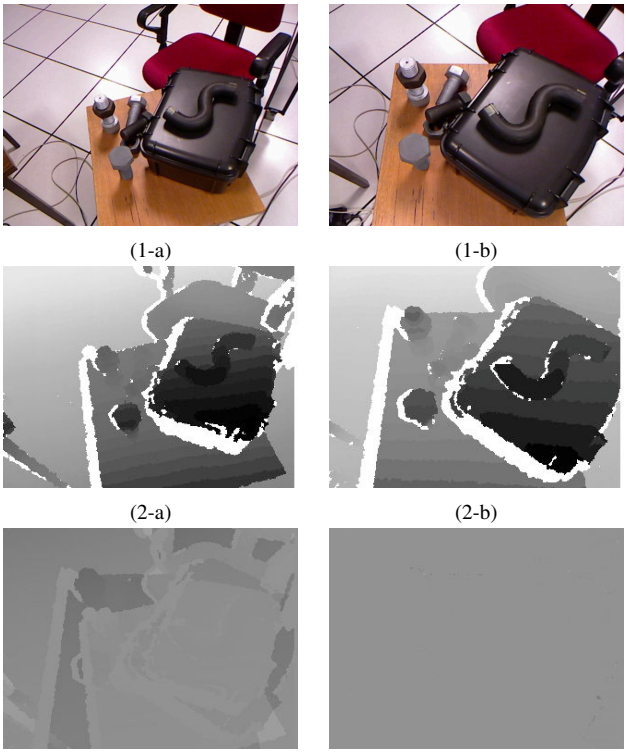

Fig. 5. First experiment. First column corresponds to the initial position. The RGB view from the Kinect (1-a) is not used in the algorithm. (2-a) Initial depth map, where white parts correspond to unavailable data. (3-a) Difference between the initial and desired depth maps. Second column corresponds to the end of the motion. The final depth map (2-b) corresponds to the desired one, since their difference (3-b) is a uniform grey.

The corresponding quantitative values for the task error, the 3D positioning errors and the velocities are given in Figure 6. Figures (b) and (c) show that with an initial error of 5 to 15cm in translation and 5 to 22deg in rotation, the positioning task is achieved with a remaining error of less than 3mm in translation and 0.4deg in rotation. Given the low depth resolution of the sensor and its noisy measurements, this corresponds to a good achievement of the task.

Note that in this scene the smoothness assumption was not verified since large depth discontinuities exist at the border of the objects, for example between the table and the floor. This experiment thus shows that the method is successful beyond its initial assumption.

In the second experiment, we evaluate the robustness of our approach with respect to partial occlusions or modifications of the observed scene. The initial scene is illustrated in Figure 8 (1-a). During the task achievement, someone entered the sensor field, removed an object and put it back several times. Some selected frames of this sequence are shown in Figure 8. The initial and final positions are illustrated in the first and last columns, while columns (b) and (c) show examples of occlusions. Note that at the end of the sequence the white bear has been completely removed from the scene, and the final depth map (Figure 8 (2-d)) is thus different from the desired one (Figure 7 (b)). This difference

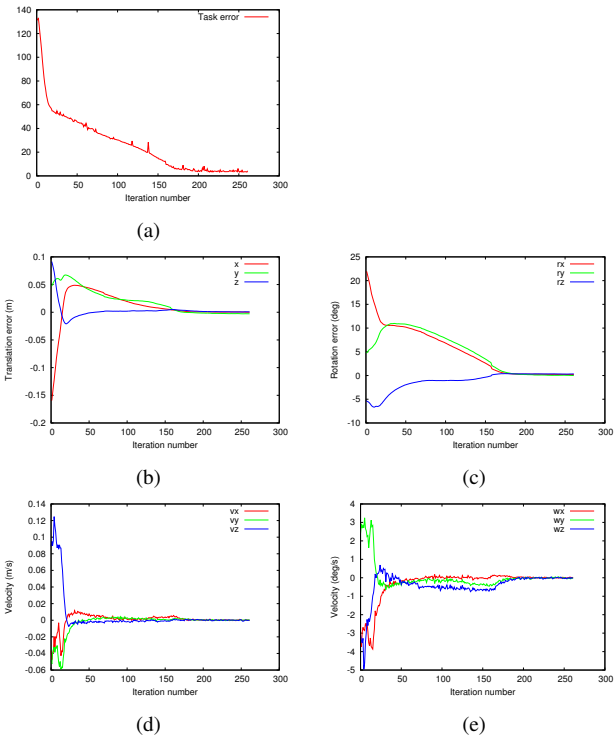

Fig. 6. First experiment. (a) Task error, (b) translational part of positioning error, (c) rotational part of positioning error, (d) translational velocities, (e) rotational velocities.

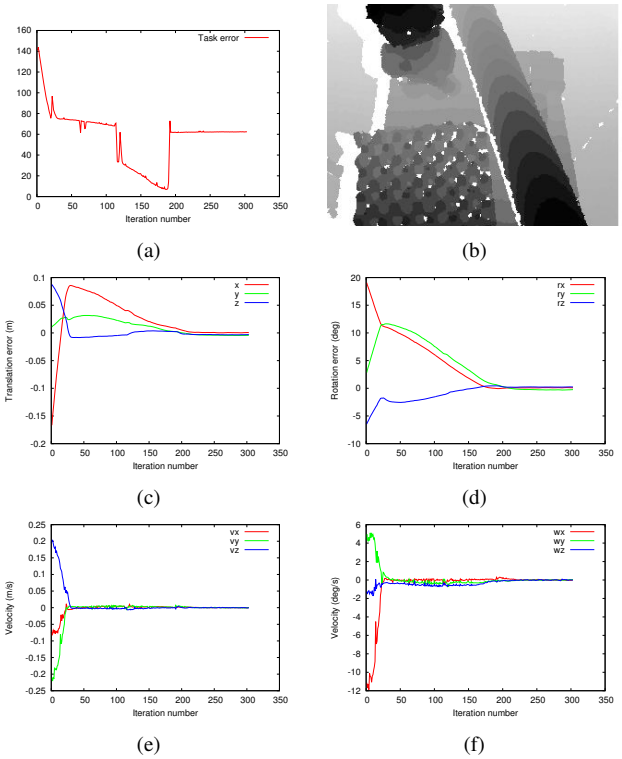

Fig. 7. Second experiment. (a) Task error, (b) desired depth map, (c) translational part of positioning error, (d) rotational part of positioning error, (e) translational velocities, (f) rotational velocities.

appears in the final difference image (Figure 8 (4-d)) and the task error function (Figure 7 (a)). However, despite the scene modifications and occlusions, the positioning task is successfully achieved, as shown by the convergence of the positioning errors in Figure 7 (b) and (c). The robustness of our control scheme to perturbations is the result of the use of M-estimation (see III-B). The effect of M-estimation is illustrated on the third row of Figure 8, where we represented the relative weights of each data in equation (13). Black pixels correspond to rejected values and brightest ones to inliers. Figure 8 (3-b), (3-c), and (3-d) show that the perturbations are correctly detected since the corresponding pixels are given a smaller weight. the positioning accuracy for this experiment is similar to the first one. The videos of these experiments are provided with this paper.

V. CONCLUSIONS

We have demonstrated that it is possible to use a dense depth map directly to achieve a robotic task. The main advantage of this approach is that it does not require any pose estimation, feature extraction or matching step. Moreover, when the depth map is obtained from an active sensor, the resulting approach is not sensitive to illumination changes as photometric approaches can be. Some limitations can appear with the use of active sensors such as Kinect RGB-D camera, in particular the noise and the absence of some measurements. We shaw however that those issues can be overcome thanks to the use of M-estimators and basic image pre-processing.

REFERENCES

[1] P.J. Besl and H.D. McKay. A method for registration of 3-D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992.

[2] F. Chaumette and S. Hutchinson. Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, December 2006.

[3] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729, 1991.

[4] C. Collewet and E. Marchand. Photometric visual servoing. *IEEE Transactions on Robotics*, (99):1–7, 2011.

[5] A.I. Comport, E. Marchand, and F. Chaumette. Statistically robust 2-D visual servoing. *IEEE Transactions on Robotics*, 22(2):415–420, 2006.

[6] D.F. DeMenthon and L.S. Davis. Model-based object pose in 25 lines of code. *Int. Journal of Computer Vision*, 15(1):123–141, 1995.

[7] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth mapping using projected patterns, May 2010. Patent US 20100118123.

[8] H. Gharavi and S. Gao. 3-D Motion Estimation Using Range Data. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):133–143, March 2007.

[9] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D Mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Int. Symposium on Experimental Robotics (ISER)*, 2010.

[10] B.K.P. Horn and J.G. Harris. Rigid body motion from range image sequences. *CVGIP: Image Understanding*, 53(1):1–13, January 1991.

[11] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, August 1981.

[12] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

[13] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, 31(3):355–395, 1987.

[14] E. Marchand, F. Spindler, and F. Chaumette. ViSP for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4), December 2005.

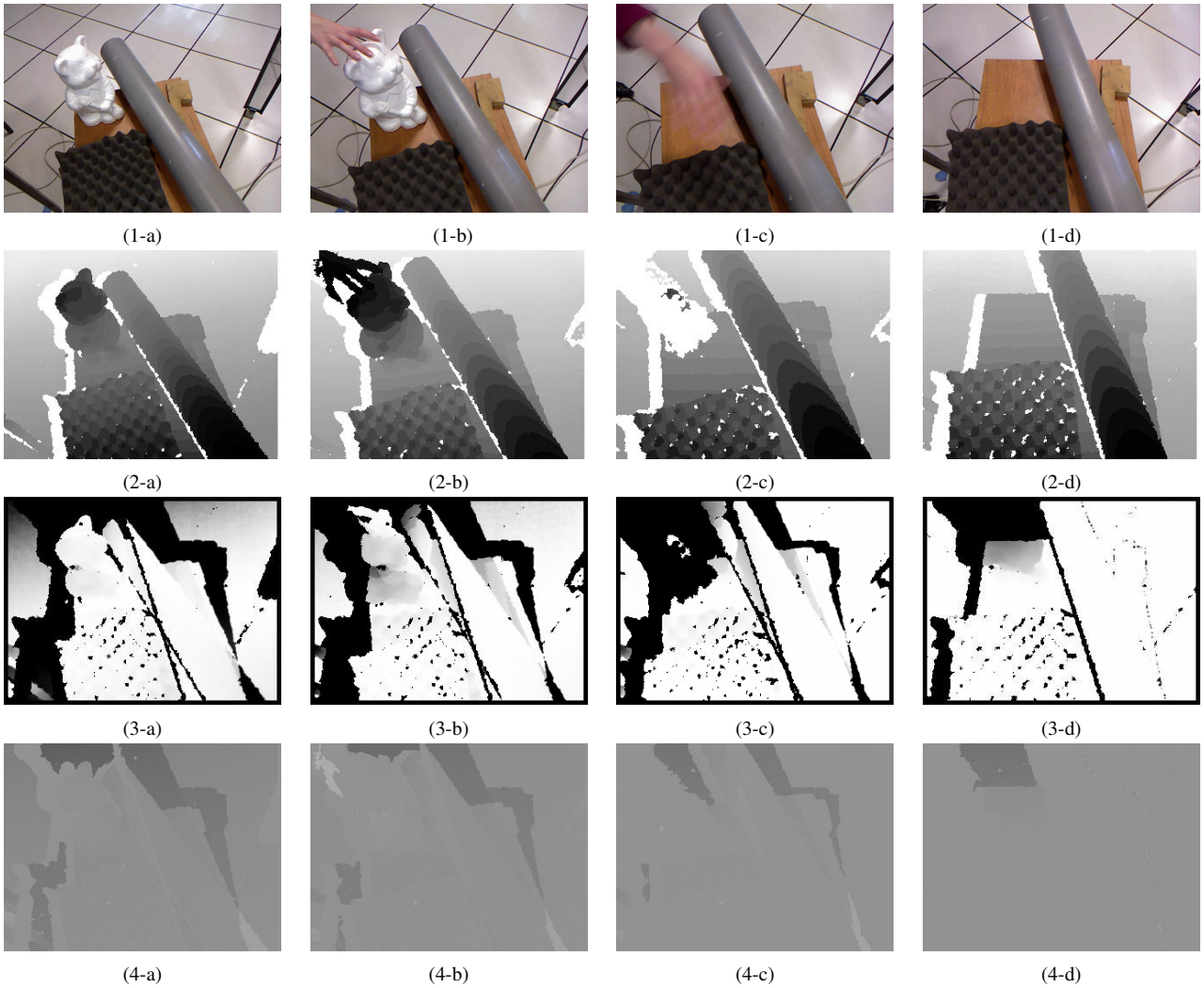

Fig. 8. Selected frames of the second experiment. Columns correspond to frames 1, 15, 69 and final frame respectively. Those frames illustrate occlusions and object removal (1-b) (1-c) (1-d). The first row gives the RGB view from the Kinect, which is not used in the algorithm but shows the setup. The depth maps are represented in the second row. The images of the third row represent the weights of each pixel in the M-estimation. Black pixels are discarded. Frames (3-b) (3-c) (3-d) show that occluded areas are given a very low weight. Fourth row: difference between the initial and desired depth maps.

[15] P. Martinet, J. Gallice, and D. Khadraoui. Vision based control law using 3D visual features. In *World Automation Congress, Robotics and Manufacturing systems*, volume 96, pages 497–502, 1996.

[16] S. May, D. Droeschel, D. Holz, S. Fuchs, E. Malis, A. Nuchter, and J. Hertzberg. Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics*, 26(11-12):934–965, 2009.

[17] RA. Newcombe, S. Izadi, and O. Hilliges. KinectFusion: Real-time dense surface mapping and tracking. In *Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[18] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Int. Conf. on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[19] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, (47):7–42, 2002.

[20] A. Swadzba, B. Liu, and J. Penne. A comprehensive system for 3D modeling from range images acquired from a 3D ToF sensor. In *Int Conf. on Computer Vision Systems (ICVS)*, 2007.

[21] WJ. Wilson and CC W. Hulls. Relative End-Effector Control Using Cartesian Position Based Visual Servoing. *IEEE Transactions on Robotics and Automation*, 12(5), 1996.

[22] M. Yamamoto, P. Boulanger, J-A. Beraldin, and M. Rioux. Direct estimation of range flow on deformable shape from a video rate range camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 15(1):82–89, 1993.