Wavelets-based 6 DOF Visual Servoing.

Mouloud Ourak, Brahim Tamadazte, Olivier Lehmann, Nicolas Andreff

Wavelets-based 6 DOF Visual Servoing

Mouloud Ourak, Brahim Tamadazte, Olivier Lehmann, and Nicolas Andreff

***Abstract*— This paper presents the development of a new visual servoing control law using global information image. It consists in the use of the wavelet spatial coefficients as the visual signal to design visual controller. More precisely, we introduce the concept of multiple resolution interaction matrix that links the time-variation of wavelet coefficients to the robot spatial velocity and the associated task function controller. The proposed controller was successfully validated using a 6 degrees-of-freedom 3*PP*SR parallel kinematic manipulator in an eye-to-hand configuration. Experimental results show the accuracy, efficiency and show good behavior of such features.**

I. INTRODUCTION

Nowadays, wavelet transformation is one of the most popular mathematical tools for the time-frequency representation. It is applicable to several domains such as, edge/corners detection, filtering, pattern recognition, economic data, data compression, compressed sensing and temperature analysis [1]. The wavelet transform can be considered as a complement to the Fourier transform. The main difference consists of that the wavelet transform represents the signal (respectively image) in both time (respectively space) and frequency domains, whereas the standard Fourier transform is only described in the frequency domain.

This paper deals with a new and an original application of the multiresolution wavelet transform (MWT), it concerns the robot motion control, more precisely designing of a vision-based controller known as visual servoing (VS) [2], [3]. The latter is introduced in order to increase the adaptivity of robot systems [2]. A vision-based controller uses visual information (features) such as geometrical data (e.g., image points, lines, circles, etc.) acquired with one or several cameras to minimize an error $\mathbf{e}(t)$ between a set of current measurements $\mathbf{s}(t)$ and desired ones $\mathbf{s}^*$.

Also, in order to enhance further the flexibility and the accuracy of the robot, it is necessary to overcome one of the main visual servoing drawbacks, the visual tracking process. The latter implies features extraction and matching over time. To do this, new visual servoing approaches have been introduced in the literature which show that the design of a vision-based control law can totally remove the visual tracking process [4]. Thereby, original methods were proposed using different pure image signal: photometry [5], [6] gradient [7], entropy [8] and Fourier transform [9]. Not only do these methods allow one to completely overcome the image processing process, but they demonstrate more accuracy and robustness thanks to the redundancy of the information used to design the controller.

Unlike the previous cited methods, our approach originally uses the results returned by a MWT to build all of the 6 DOF control law elements. Actually, the half-resolution image resulting from the MWT is used as visual signal instead of the whole image. Additionally, In opposite to photometric approach we derive the spatial derivative wavelet coefficients involved in computing a so-called *multiple resolution interaction matrix* that links the time-variation of derivative wavelet coefficients to the robot spatial velocity from the detail wavelet coefficients. Certainly, one the main advantages of our method (e.g., comparing to the photometric approach) lies especially on the automatic filtering of the low and high frequencies in the image generally corresponding to the image noise. This is somehow equivalent to amounts to selecting automatically a set of noiseless and redundant visual signal for a more accurate and stable visual controller.

The proposed controller was tested on a 6 DOF robotic platform consisting of a 3*PP*SR parallel robot. The experimental validations were conducted in nominal conditions and using different coefficients resolution in order to define the optimal ones which improve the controller behaviors: convergence, robustness, and accuracy.

The manuscript is organized as follows: Section II reviews the basics of the MWT as visual signal. In Section III details the methodology followed to design the proposed 6 DOF MWT-based visual servoing controller, while the experimental results are discussed in Section IV.

II. WAVELETS MULTIPLE RESOLUTION AS VISUAL SIGNAL

A. Definition

The wavelet transform was introduced to analyze a signal simultaneously in both time and frequency domains [10]. Note that until twenty first century, only the Fourier transform was used to decompose reversibly a signal without any loss of information. Fourier transform provides a frequency analysis, but does not allow the temporal location of sharp changes of the signal. The enormous work done in the 1980s to formalize the wavelets, allowed building several frames in which a time-frequency representation of a signal became possible.

In this paper, we are interested in the frame of the MWT decomposition of an image. Consequently, for a better understanding of the controller design, it is appropriate to

This work is conducted with a financial support from the project NEMRO (ANR-14-CE17-0013-001) funded by the Agence Nationale de la Recherche, France. It is also performed in the framework of the Labex ACTION (ANR-11-LABEX-01-01)

The authors are with FEMTO-ST, AS2M, Univ. Bourgogne Franche-Comté, Univ. de Franche-Comté/CNRS/ENSMM, 25000 Besanon, France. brahim.tamadazte@femto-st.fr

remember the basics of the wavelet transform and more precisely MWT.

Consider a 2D signal $\mathcal{F}(x,y)$ ($\mathcal{F} \in \mathbb{L}^2(\mathbb{R}^2)$) and a wavelet function defined by $\mathcal{G}(x,y)$ ($\mathcal{G} \in \mathbb{L}^2(\mathbb{R}^2)$). The general wavelet transform is defined by the following inner product

$$\left\langle \mathcal{F}(x,y), \mathcal{G}(x,y) \right\rangle = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathcal{F}(x,y)\ \mathcal{G}(x,y)\ dxdy \qquad (1)$$

The MWT for its part uses two functions: scaling and mother wavelet. Furthermore, this paper uses the Daubechies [11] pair of the fourth order ($db4$) available in MATLAB. Usually, those functions are used in their frequency state represented in list of numerical parameters. More precisely, the pair of functions $db4$ used in this paper are

- *Scaling function* (ϕ) is represented as low pass filter $l_{db4}(f_x, \xi_l)$ in frequency domain with the following numerical parameters in MATLAB

$$\xi_l = \{-0.01, 0.03, 0.03, -0.18, -0.02, 0.63, 0.71, 0.23\}$$

  while, for modelling the control law in the section III we need to compute the first derivative $\frac{\partial l_{db4}}{\partial f_x}(f_x, \xi_l')$ values of $l_{db4}(f_x, \xi_l)$, which are built with the following numerical parameters

$$\xi_l' = \{0, -0.09, -0.03, 0.32, -0.34, -1.22, -0.02, 0\}$$

- *Mother wavelet* (ψ) is represented as high pass filter $h_{db4}(f_x, \xi_h)$ in frequency domain with the numerical parameters in MATLAB

$$\xi_h = \{-0.2, 0.7, -0.63, -0.02, 0.18, 0.03, -0.03, -0.01\}$$

  whereas the first derivative $\frac{\partial h_{db4}}{\partial f_x}(f_x, \xi_h')$ values of $h_{db4}(f_x, \xi_h)$ as for scaling function are

$$\xi_h' = \{0.45, -0.60, 0.27, 0.08, -0.07, -0.02, 0.01, -0.1\}$$

Moreover, the pair of filters is applied to the image based on the inner product (1). The wavelet functions associated to $\mathcal{G}(x,y)$ in (1) in the MWT are the result of 4 combinations

$$\Gamma^0(x,y) = \left(\Delta \circ \phi(x)\right) \circ \left(\Delta \circ \phi(y)\right) \qquad (2)$$

$$\Gamma^H(x,y) = \left(\Delta \circ \phi(x)\right) \circ \left(\Delta \circ \psi(y)\right) \qquad (3)$$

$$\Gamma^V(x,y) = \left(\Delta \circ \psi(x)\right) \circ \left(\Delta \circ \phi(y)\right) \qquad (4)$$

$$\Gamma^D(x,y) = \left(\Delta \circ \psi(x)\right) \circ \left(\Delta \circ \psi(y)\right) \qquad (5)$$

where Δ is the decimating operator taking one pixel out of two, and $\circ$ is the composition of functions.

In fact, the inner product (1) between the image $\mathbf{I}_{(2^{j+1})}$ at resolution 2^{j+1} with (2), yields

$$\mathbf{I}_{(2^j)} = \left\langle \mathbf{I}_{(2^{j+1})}\ ,\ \Gamma^0_{2^j}(x,y) \right\rangle \qquad (6)$$

where, the result $\mathbf{I}_{(2^j)}$ is the new image at half resolution (Fig. 1(b.1)) and used as measurement $\mathbf{s}$ in the controller.

In contrast, the inner product (1) between $\mathbf{I}_{(2^{j+1})}$ and the wavelet functions (3) to (5)

$$\mathbf{d}^k_{(2^j)} = \left\langle \mathbf{I}_{(2^{j+1})}\ ,\ \Gamma^k_{2^j}(x,y) \right\rangle, \quad \forall k = H, V, D \qquad (7)$$

give the details coefficients $\left(\mathbf{d}^k_{(2^j)}\right)_{(k=H,V,D)}$ (Fig. 1(b.(2-4))), where H is horizontal orientation (Fig. 1(b.2)), V is vertical orientation (Fig. 1(b.3)), and D is diagonal orientation (Fig. 1(b.4)).

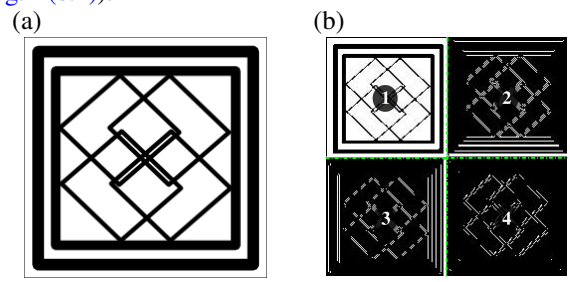

Fig. 1. (a) the original image $\mathbf{I}_{(2^{j+1})}$, (b) the MWT decomposition, (b.1) the new thresholded (for a better visualization) image at a half resolution $\mathbf{I}_{(2^j)}$, (b.2), (b.3) and (b.4) the horizontal ($\mathbf{d}^H_{(2^j)}$), vertical ($\mathbf{d}^V_{(2^j)}$) and diagonal ($\mathbf{d}^D_{(2^j)}$) thresholded wavelet coefficients, respectively.

B. Modeling

In order to estimate the motion between two images, the optical flow is generally used. Based on the assumption of brightness constancy [12], it is possible to write the optical flow constraint equation (OFCE) for small movement of the whole image $\mathbf{I}$ as

$$\nabla \mathbf{I} \dot{p} + \frac{\partial \mathbf{I}}{\partial t} = 0 \qquad (8)$$

where $\dot{p} = (\dot{x}, \dot{y})^\top$ is the velocity of a point $p = (x,y)^\top$ in the image, and $\nabla \mathbf{I}$ is the spatial image gradient of $\mathbf{I}$.

Following several mathematical manipulations explained in [13], (8) can be written based on the MWT as

$$\left\langle \mathbf{I}_{(2^{j+1})}, \frac{\partial \Gamma^H}{\partial x} \right\rangle \dot{x} + \left\langle \mathbf{I}_{(2^{j+1})}, \frac{\partial \Gamma^V}{\partial y} \right\rangle \dot{y} + \frac{\partial}{\partial t} \left\langle \mathbf{I}_{(2^{j+1})}, \Gamma^0 \right\rangle = 0 \qquad (9)$$

In order to simplify (9), we introduce the following notations [1]

$$\mathbf{g}^H_{(2^j)} \triangleq \left\langle \mathbf{I}_{(2^{j+1})}, \frac{\partial \Gamma^H}{\partial x} \right\rangle \qquad (10)$$

$$\mathbf{g}^V_{(2^j)} \triangleq \left\langle \mathbf{I}_{(2^{j+1})}, \frac{\partial \Gamma^V}{\partial y} \right\rangle \qquad (11)$$

where $\mathbf{g}^H_{(2^j)}$ and $\mathbf{g}^V_{(2^j)}$ are computed based on the first derivative filters shown in Fig. 3. Hence, replacing the parts of (9) by (6), (10) and (11) yields

$$\forall p = (x,y), \quad \mathbf{g}^H_{(x,y)}\ \dot{x} + \mathbf{g}^V_{(x,y)}\ \dot{y} + \frac{\partial}{\partial t}\ \mathbf{I}_{(x,y)} = 0 \qquad (12)$$

Choosing $\mathbf{s} = \mathbf{I}_{(2^j)}$, the letter writes (for each point $p = (x,y)$)

$$\begin{bmatrix} \mathbf{g}^H_{(x,y)} & \mathbf{g}^V_{(x,y)} \end{bmatrix} \dot{p} + \dot{\mathbf{s}}(x,y) = 0 \qquad (13)$$

[1] with a slight breach for simplicity sake.

Using the 2D point interaction matrix $\mathbf{L}_{2D}$ [3]

$$\mathbf{L}_{2D} = \begin{pmatrix} \frac{-1}{z} & 0 & \frac{x}{z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{z} & \frac{y}{z} & -(1+y^2) & -xy & -x \end{pmatrix} \quad (14)$$

linking the image point velocity $\dot{p}$ to ${}^c\mathbf{v}_c \in se(3)$ the velocity of the camera in the camera frame.

$$\dot{p} \;=\; \mathbf{L}_{2D}\; {}^c\mathbf{v}_c \quad (15)$$

we finally get

$$\dot{\mathbf{s}} \;=\; \underbrace{-\begin{bmatrix} \mathbf{g}^H_{(2^j)} & \mathbf{g}^V_{(2^j)} \end{bmatrix} \mathbf{L}_{2D}}_{\mathbf{L}_w}\; {}^c\mathbf{v}_c \quad (16)$$

where $\mathbf{L}_w$ $\left(\mathbf{L}_w \in \mathbb{R}^{k\times 6}\right)$ will be named the interaction matrix associated to the multiresolution wavelet transform coefficients.

III. CONTROL

A. Eye-to-Hand Robot Control

From (16) and the visual servoing formalism [3], the exponential convergence and stable control based on the proposed approach is written as follows

$${}^c\mathbf{v}_c \;=\; -\lambda\, \widehat{\mathbf{L}}_w^+ \left(\mathbf{s}-\mathbf{s}^*\right) \quad (17)$$

where λ is a positive gain, $\mathbf{s} = \mathbf{I}_{(2^j)}$ and $\mathbf{s}^* = \mathbf{I}^*_{(2^j)}$ are the current and desired visual signals, respectively, and $\widehat{\mathbf{L}}_w^+$ is an estimate of the pseudo-inverse of $\mathbf{L}_w$ which is computed at the desired configuration. This control is trivially *Lyapunov* stable if $\mathbf{L}_w\widehat{\mathbf{L}}_w^+$ is definite positive.

Moreover, to ensure a stable and smooth convergence, we use the *Levenberg-Marquardt* optimization method [14]. Actually, this optimization method allows the combination of gradient descent optimization for a distant pose and *Gauss-Newton* optimization for an approachable pose. The choice of *Levenberg-Marquardt* leads us to write the velocity in the camera's frame $\mathcal{R}_c$ as

$${}^c\mathbf{v}_c \;=\; -\lambda\, \left(\mu\, \mathbb{I}_{6\times 6} + \widehat{\mathbf{L}_w}^\top \widehat{\mathbf{L}_w}\right)^{-1} \widehat{\mathbf{L}_w}^+ (\mathbf{I}_{2j} - \mathbf{I}^*{}_{2j}) \quad (18)$$

where μ is constant positive scalar, and $\mathbb{I}_{6\times 6}$ identity matrix.

In an *eye-to-hand* configuration of the system the robot velocity $\dot{\mathbf{q}}$ writes as

$$\dot{\mathbf{q}} \;=\; -{}^b\mathbf{K}_e^{-1}\; {}^b\mathbf{V}_c\; {}^c\mathbf{v}_c \quad (19)$$

where ${}^b\mathbf{K}_e^{-1}$ is the inverse kinematic Jacobian matrix of the robot, and ${}^b\mathbf{V}_c$ is the transformation matrix associated to the velocity change frame from the robot base frame $\mathcal{R}_b$ to the camera one $\mathcal{R}_c$. The latter is constructed as

$${}^b\mathbf{V}_c = \begin{bmatrix} {}^b\mathbf{R}_c & \left[{}^b\mathbf{t}_c\right]_\wedge\, {}^b\mathbf{R}_c \\ 0 & {}^b\mathbf{R}_c \end{bmatrix} \quad (20)$$

where ${}^b\mathbf{R}_c$ is the 3×3 rotation matrix from $\mathcal{R}_c$ to $\mathcal{R}_b$, ${}^b\mathbf{t}_c$ is the 3×1 associated translation vector, and $[\cdot]_\wedge$ is the skew symmetric matrix associated with the vector cross-product.

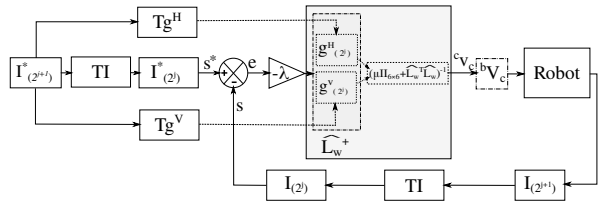

Fig. 2. Closed loop visual servoing using the half-resolution (2^j) wavelet as visual signal. **TI**, $\mathbf{Tg^H}$ and $\mathbf{Tg^V}$ are explained in Fig. 3.

B. MWT Control Approach

The control loop (see Fig. 2) used to servo the $SE(3)$ configurations of 3*PP*SR parallel robot with the camera is based on two main novel MWT parts. First, the approximation wavelet coefficients $\mathbf{I}_{(2^j)}$ resulting from the top path of Fig. 3, in order to construct the visual signal of the controller $\mathbf{s}$. Those approximation coefficients $\mathbf{I}_{(2^j)}$ are the only computed for each iteration of the control loop in order to reduce the computation time. Second, the details wavelet coefficients $\mathbf{g}^H_{(2^j)}$ and $\mathbf{g}^V_{(2^j)}$ involved in the construction of the interaction matrix associated to MWT shown in the two bottom path of Fig. 3. This interaction matrix is computed once in the initialization of the algorithm, and in the reference configuration of the robot.

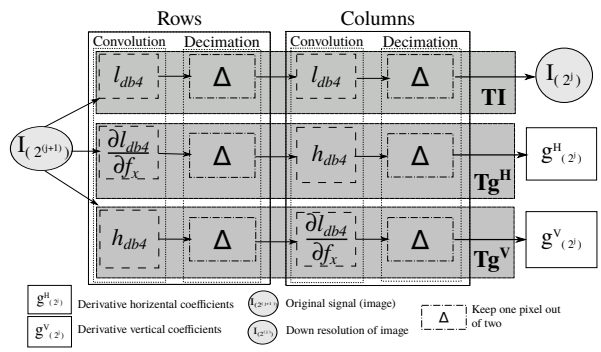

Fig. 3. Diagram representing the algorithm used to compute the derivative multiresolution wavelet coefficients.

IV. EXPERIMENTAL VALIDATION

The proposed approach was validated using the experimental set-up described below using different scenarios and various scenes. In each test, the 6 DOF of our robot are controlled. The error $\mathbf{e}$ and the camera velocity $\mathbf{v}$ evolution (in each DOF), and the cost-function value $\mathcal{C}(\mathbf{r})$ over iterations are plotted and analyzed.

A. Experimental Setup

The proposed controller was tested and validated on our lab experimental set-up. It consists of an *eye-to-hand* camera/robot system (Fig. 4) including

- a 6 DOF parallel robotic structure: 3*PP*SR robot SpaceFAB SF-3000 BS from Micos[2]. The latter is characterized with the following features: translation range

[2]http://www.pimicos.com

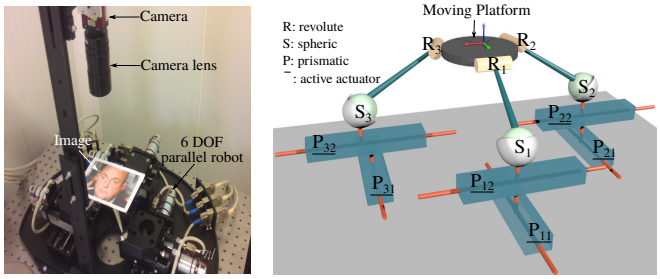

Fig. 4. Global view on the 6 DOF experimental platform.

$(t_x, t_y, t_z)^\top_{max}$ = (50, 100, 12.7)$^\top$ (mm) and rotation range $(r_x, r_y, r_z)^\top_{max}$ = (10, 10, 10)$^\top$ (°), a linear resolution of 0.2μm (repeatability of $\pm$0.5μm) and an angular resolution of 0.0005° (repeatability of $\pm$0.0011°);

- a Firewire monochrome CCD camera is attached on top of the robot platform. The camera frame rate is 25 frames-per-second (fps) for a resolution of 640 $\times$ 480 pixels;
- two computers were used: a 3.2GHz i5 core Intel CPU with a Linux distribution which is dedicated to computer vision and control and a 2.33GHz Xeon Intel CPU with a Windows distribution is used for the robot inner control (inner PID loop, static and differential kinematic models, etc.). The computers communicate using asynchronously a TCP/IP protocol.

B. Results

*1) **Test 1**:* For this experiment, a planar object (i.e., photography of *Bernard BLIER*) was used (Fig. 5). The initial Cartesian error between the desired image $\mathbf{I}^*$ and the current image $\mathbf{I}$, was $\Delta\mathbf{r}_{\mathbf{init}}$ = (12mm, 12mm, 5mm, 7.5°, 7°, 5.5°). Note that the desired pose is chosen such that the photography and the camera planes are almost parallel. Also, the experimental conditions are optimal (stable lighting source, acceptable calibration parameters (i.e., camera intrinsic parameters matrix and eye-to-hand matrix $\left({}^c\mathbf{V}_b\right)_{6\times6}$), and without significant external disturbances (e.g., occlusions, vibrations, etc.).

Fig. 5(a) and (b) show the desired and the current images, respectively, while Fig. 5(c) and (d) the corresponding half resolution wavelet. Finally, Fig. 5(e) shows the image difference between the desired and the current image at $t = 0$, i.e., $\mathbf{I}_{diff} = \frac{(\mathbf{I}-\mathbf{I}^*)+255}{2}$, whereas Fig. 5(f) represents the final image difference demonstrating convergence of the controller towards the desired position.

Fig. 6(a) presents the Cartesian error decay $\Delta\mathbf{r}_\mathbf{i}$ in each DOF as well as the decrease of the norm of the error Fig. 6(b). Thereby, it can be underlined that the different components converge, simultaneously, to their respective desired values. This demonstrates the good properties of the controller regarding the evolution of the norm of the error with a satisfying exponential decrease. The final Cartesian error, computed using high resolution robot encoders supplied by the robot software, was estimated to be $\Delta\mathbf{r}_{\mathbf{final}}$ = **(0.071mm, 0.067mm, 0.164mm, 0.099°, 0.054°, 0.051°)**.

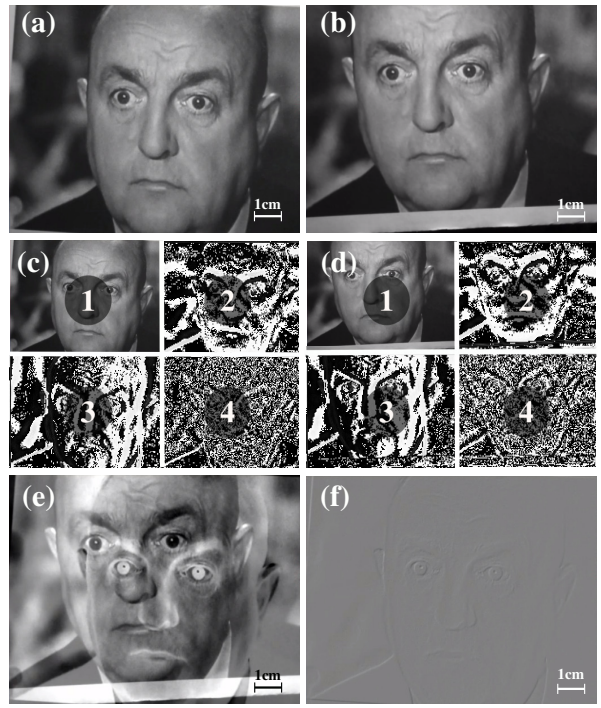

Fig. 5. [*Test 1*]: image snapshots acquired during a 3D positioning task in nominal conditions: (a) reference image $\mathbf{I}^*$, (b) current image $\mathbf{I}$, (c) and (d) the multiple resolution wavelets results corresponding to the reference and current positions for half resolution, respectively, (e) initial image difference $\mathbf{I}_{diff}$, and (f) final $\mathbf{I}_{diff}$ illustrating that the controller converges accurately towards the reference position.

The translation $(v_x, v_y, v_z)^\top$ and the rotation $(\omega_x, \omega_y, \omega_z)^\top$ velocities in each DOF *versus* number of iterations (iteration = 0.12 second) are plotted in Fig. 6(c) and (d), respectively. Fig. 6(e) depicts the robot trajectory during the positioning task recorded and expressed in the desired frame. It can be underlined that after a slight deviation, the robot converges to the desired position in an almost straight-line. However, it is important to remember that the control is done on the image; this does not generate exponential convergence of the Cartesian error unlike the 3D visual servoing methods. The convergence domain of the controller is illustrated by Fig. 6(f) showing a perfect convex shape presenting a global minimum without any local minima of the cost-function.

*2) **Test 2**:* This experimental scenario was carried out by adding an external disturbances. To do that, the reference image was taken under nominal conditions and during the positioning task, the current images were acquired by adding external object (i.e., *token*) on the scene simulating significant partial occlusions Fig. 7). An image sequence was captured during the positioning task (Fig. 7) which shows the perfect convergence of the controller as can be seen in Fig. 7(d).

The initial Cartesian error was $\Delta\mathbf{r}_{\mathbf{init}}$ = (12mm, 12mm, 5mm, 7.5°, 7°, 5.5°). Despite the unfavorable conditions, the robot converges smoothly toward the desired position, this almost unsurprisingly when analyzing the cost-function $\mathcal{C}(\mathbf{r})$ that has a perfect convex shape with a well-defined

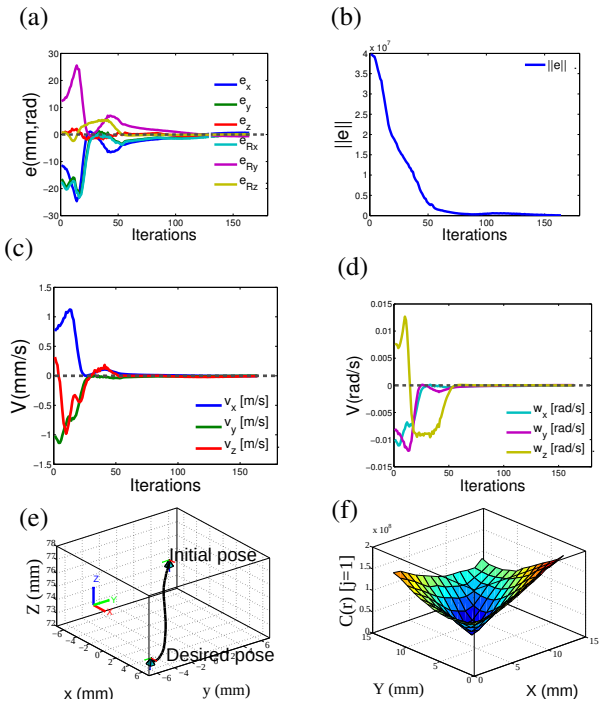

Fig. 6. [*Test 1*]: (a) Cartesian error $\Delta\mathbf{r}_i$ in each DOF, (b) decay of the norm of the error $\|\mathbf{e}\|$, (c) translation velocities $\mathbf{v}_t$ (in mm/s), (d) rotation velocities $\mathbf{v}_\omega$ (rad/s), (e) 3D robot trajectory and (f) wavelet coefficients cost-function $\mathcal{C}(\mathbf{r})$ (case of nominal conditions).

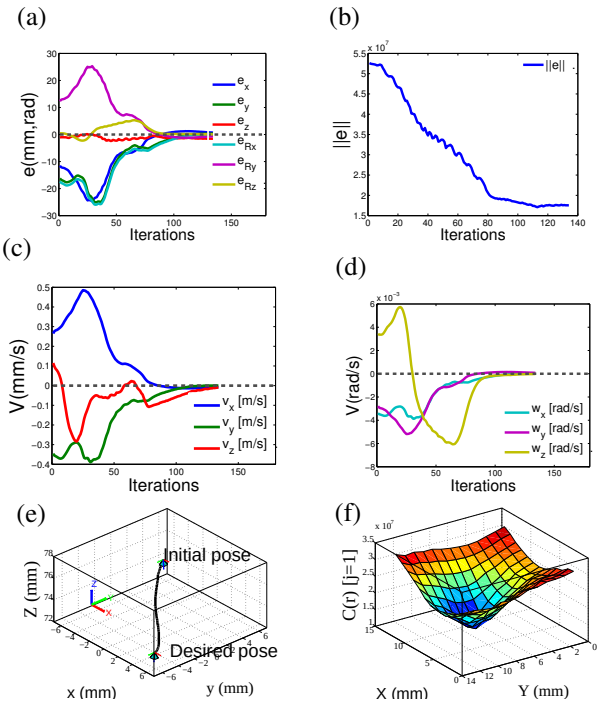

Fig. 8. [*Test 2*]: (a) Cartesian error $\Delta\mathbf{r}_i$ in each DOF, (b) decay of the norm of the error $\|\mathbf{e}\|$, (c) translation velocities $\mathbf{v}_t$ (in mm/s), (d) rotation velocities $\mathbf{v}_\omega$ (rad/s), (e) 3D robot trajectory and (f) wavelet coefficients cost-function $\mathcal{C}(\mathbf{r})$ (case of occlusion).

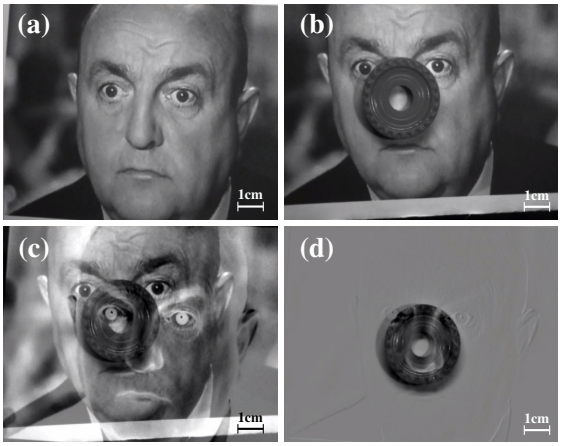

Fig. 7. [*Test 2*]: image sequence captured during the positioning task in order to judge the sensitivity of the controller under lighting changes. (a) desired image $\mathbf{I}^*$ acquired in normal illumination, (b) current image $\mathbf{I}$ captured under sensitive change of the illumination, (c) initial $\mathbf{I}_{diff}$ and (d) final $\mathbf{I}_{diff}$ showing that the controller reaches the desired position, although $\mathbf{I}_{diff}$ is not completely gray, this is due to the token add in the scene.

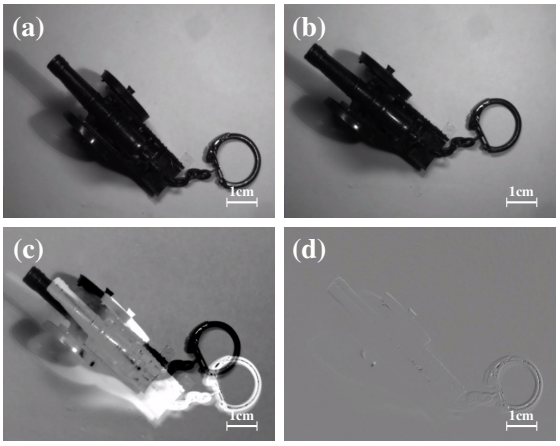

Fig. 9. [*Test 3*]: Snapshots captured during the positioning task performed using 3D scene. (a-b) desired and current images, respectively, (c) initial $\mathbf{I}_{diff}$ and (d) final $\mathbf{I}_{diff}$.

global minimum (Fig. 8 (f)). The Cartesian error in each DOF $\Delta\mathbf{r}_i$ as well as the norm of the error $\|\mathbf{r}\|$ are displayed in Fig. 8 (a) and (b). It can be highlighted that the Cartesian error remains very low despite the occlusion i.e., $\Delta\mathbf{r}_{final}$ = **(0.213mm, 0.110mm, 0.513mm, 0.188°, 0.245°, 0.161°)**.

The computed velocities and robot trajectory are still very interesting and quite similar to the the those of the case of nominal conditions (i.e., *Test 1*) despite the introduction of a significant dissimilarity between the current and desired images Fig. 7 (a) and (b).

*3) **Test 3**:* The objective of this supplementary experiment is to assess the performance of our controller *wrt* a 3D scene. Thus, the 2D scene used in the first experiments was replaced by a complex and less textured 3D scene (i.e., *a key ring as a cannon*). Note that the 3D depths of the scene were neither reconstructed nor estimated (only the interaction matrix that assumes a constant depth Z^* computed at the desired position). Despite that, the controller remains robust to depth variations as can be shown in Fig. 9.

Fig. 10(f) shows the computed cost-function $\mathcal{C}(\mathbf{r})$ in the case of 3D scene illustrating again a perfect convex shape. This allows a convergence to the desired position without

much difficulty as can be shown in the Cartesian error decay in each DOF as well as in the norm of the error (Fig. 10(a-b)). The final Cartesian error for this test was estimated to be $\Delta\mathbf{r}_{final}$ **= (0.110mm, 0.224mm, 0.128mm, 0.282°, 0.120°, 0.290°)**.

The robot joint translation and rotation velocities are plotted in Fig. 10(c-d) depicting a smooth convergence to zero. The robot trajectory is reported in Fig. 10(e) demonstrating a behavior comparable to those obtained in the 2D scenes.

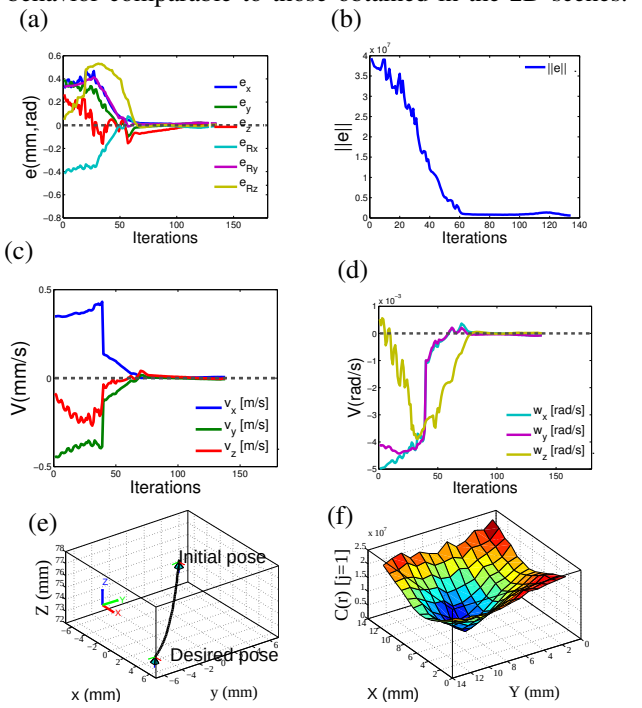

Fig. 10. [*Test 3*]: (a) Cartesian error $\Delta\mathbf{r}_i$ in each DOF, (b) decay of the norm of the error $\|\mathbf{e}\|$, (c) translation velocities $\mathbf{v}_t$ (in mm/s), (d) rotation velocities $\mathbf{v}_\omega$ (rad/s), (e) 3D robot trajectory and (f) wavelet coefficients cost-function $\mathcal{C}(\mathbf{r})$ (case of 3D scene).

*4) **Test 4**:* The objective of this experiment is to compare our method to the photometric one [5] (an example is provided in ViSP library [15]). TABLE I presents the final Cartesian error for both WMT-based visual servoing and photometric-based visual servoing acquired during a 6 DOF positioning task under the same conditions of use (current position, lighting source, calibration parameters, etc.). Thus, two comparison tests was achieved: 1) nominal conditions and 2) under partial occlusions. The obtained results demonstrate that the MWT VS is almost similar to the photometry VS (i.e., $||e||_{WMT} = \mathbf{0.23}$ and $||e||_{photometry} = \mathbf{0.22}$) while it presents a better behavior under partial occlusions (i.e., $||e||_{WMT} = \mathbf{0.6}$ and $||e||_{photometry} = \mathbf{0.84}$).

V. CONCLUSION

In this paper, it was demonstrated for the first time that it is possible to design an efficient 6 DOF visual servoing using the wavelet coefficients as visual signal. The originality of our approach is due to the direct use of the wavelets coefficients for the design of a *multi-resolution interaction matrix*. It links directly the time-variation of wavelet coefficients to the robot spatial velocity and the associated task function controller.

TABLE I

NUMERICAL VALUES (WMT *vs* PHOTOMETRY) IN NOMINAL CONDITIONS (NC) AND OCCLUSION (OCC) $\left(\triangle T_i \text{ (MM) AND } \triangle R_i \text{ (°)}\right)$

	$\triangle T_x$	$\triangle T_y$	$\triangle T_z$	$\triangle R_x$	$\triangle R_y$	$\triangle R_z$
initial error	5.000	5.000	3.000	4.000	3.000	3.000
WMT (nc)	**0.006**	**0.018**	**0.026**	**0.192**	**0.128**	**0.032**
photom. (nc)	0.011	0.009	0.031	0.178	0.098	0.085
WMT (occ)	**0.017**	**0.097**	**0.452**	**0.304**	**0.207**	**0.107**
photom. (occ)	0.086	0.035	0.461	0.410	0.120	0.562

The proposed controller was tested experimentally using a 6DOF parallel robotic structure equipped with a CCD camera which was mounted in an *eye-to-hand* configuration. The obtained results have demonstrated the efficiency of the developed controller in terms of accuracy (some tens of micrometers and few tenths of millidegree in the translation and rotation Cartesian space, respectively), convergence, and robustness (the controller has kept on working under partial occlusion).

In future work, we aim to investigate the use of the developed controller on other image modalities as medical images i.e., optical coherence tomography. Even farther, the use of wavelet transform on the designing of the controller naturally leads to the investigation of a hybrid *visual servoing-compressed sensing* control scheme.

REFERENCES

[1] S. Mallat, "A wavelet tour of signal processing: the sparse way". Academic press, 2008.

[2] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Tran. on Rob. and Auto.*, vol. 12(5), pp. 651–670, 1996.

[3] F. Chaumette and S. Hutchinson, "Visual servo control. part I. basic approaches," *IEEE Rob. & Auto. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[4] D. Koichiro, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. J. of Computer Vision*, vol. 37, no. 1, pp. 7–20, 2000.

[5] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Trans. on Rob.*, vol. 27, no. 4, pp. 828–834, 2011.

[6] B. Tamadazte, N. Piat, and E. Marchand, "A direct visual servoing scheme for automatic nanopositioning," *IEEE/ASME Trans. on Mech.*, vol. 17, no. 4, pp. 728–736, 2012.

[7] E. Marchand and C. Collewet, "Using image gradient as a visual feature for visual servoing," in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2010, pp. 5687–5692.

[8] A. Dame and E. Marchand, "Entropy-based visual servoing," in *IEEE Int. Conf. on Rob. and Auto.*, 2009, pp. 707–713.

[9] N. Marturi, B. Tamadazte, S. Dembélé, and N. Piat, "Visual servoing schemes for automatic nanopositioning under scanning electron microscope," in *IEEE Int. Conf. on Rob. and Auto.*, 2014, pp. 981–986.

[10] Y. Meyer, *Ondelettes, filtres miroirs en quadrature et traitement numerique de l'image*. Springer, 1990.

[11] I. Daubechies *et al.*, *Ten lectures on wavelets*. SIAM, 1992, vol. 61.

[12] B. K. Horn and B. G. Schunck, "Determining optical flow," in *1981 Technical symposium east*. Int. Society for Optics and Photonics, 1981, pp. 319–331.

[13] C. Bernard, "Wavelets and ill-posed problems: optic flow estimation and scattered data interpolation," *Praca doctorska, Ecole Polytechnique, Francja*, 1999.

[14] M. Lourakis, "A brief description of the levenberg-marquardt algorithm implemented by levmar," *Foundation of Research and Technology*, vol. 4, pp. 1–6, 2005.

[15] E. Marchand, F. Spindler, and F. Chaumette "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Rob. and Auto. Mag.*, vol. 12, no. 4, pp. 40–52, 2015.