Robust model-based tracking with multiple cameras for spatial applications

Fabien Dionnet, Eric Marchand
INRIA, IRISA, projet Lagadic, F-35000 Rennes, France ;
Email {Fabien.Dionnet,Eric.Marchand}@irisa.fr

Abstract

This paper proposes a real-time, robust and efficient 3D model-based tracking algorithm for visual servoing. A virtual visual servoing approach is used for 3D tracking. This method is similar to more classical non-linear pose computation techniques. Robustness is obtained by integrating an M-estimator into the virtual visual control law via an iteratively re-weighted least squares implementation. The presented approach is also extended to the use of multiple cameras. Results show the method to be robust to occlusion, changes in illumination and miss-tracking.

1 INTRODUCTION

This paper presents a vision-based tracker for visual servoing applications. This study focuses on the registration techniques that allow alignment of real and virtual worlds using images acquired in real-time by moving cameras. In the related computer vision literature geometric primitives considered for the estimation are often points [7, 3, 11], contours or points on the contours [10, 2, 5], segments, straight lines, conics, cylindrical objects, or a combination of these different features [13]. Another important issue is the registration problem. *Purely geometric* (eg, [4]), or *numerical and iterative* [3] approaches may be considered. *Linear approaches* use a least-squares method to estimate the pose. *Full-scale non-linear optimisation techniques* (e.g., [7, 10, 5, 2, 14]) consist of minimising the error between the observation and the forward-projection of the model. In this case, minimisation is handled using numerical iterative algorithms such as Newton-Raphson or Levenberg-Marquardt. The main advantage of these approaches are their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence.

In this paper, pose computation is formulated in terms of a full scale non-linear optimisation: *Virtual Visual Servoing* (VVS). In this way the pose computation problem is considered as similar to 2D visual servoing as proposed in [15, 13, 2]. Assuming that the low level data extracted from the image are likely to be corrupted, we use a statistically robust camera pose estimation process (based on the widely accepted statistical techniques of robust M-estimation [8]). This M-estimation is directly introduced in the control law to address [2]. This framework is used to create an image feature based system which is capable of treating complex scenes in real-time. Among other advantages demonstrated in previous work [2] (notably the accuracy, efficiency, stability, and robustness) the framework scales to the use of multiple cameras with small or wide baselines. Previous work has been done to consider pose computation with stereo systems [14]. Although the goal is very similar, the modeling of the cost function, the visual feature considered and then the Jacobian, as well as the minimization issue that does not integrate robust estimation are different from the method presented in this paper.

The context of this work is the development of robust and fast 3D tracking algorithms for visual servoing application is spatial context. Indeed the goal is to develop a demonstrator of a robot arm in space environment able to grasp objects by visual servoing. The considered robot is the ESA Eurobot that should be on the International Space Station in 2010. The configuration of this robot (3 arms) allows to consider multiple cameras (with wide baseline) in order to allow eye-in-hand or eye-to-hand control [9].

2 MULTI-CAMERAS ROBUST VISUAL TRACKING

2.1 Overview and Motivation

As already stated, the fundamental principle of the proposed approach is to define the pose computation problem as the dual problem of 2D visual servoing [6, 9]. In visual servoing, the goal is to move a camera in order to observe an object

at a given position in the image. An explanation is now be given as to why the pose computation problem is very similar.

2.1.1 Case of monocular system

To illustrate the principle, consider the case of an object with various 3D features ${}^{o}\mathbf{P}$ (for instance, ${}^{o}\mathbf{P}$ are the 3D coordinates of these features in the object frame). A virtual camera is defined whose position in the object frame is defined by the homogeneous matrix ${}^{c}\mathbf{M}_{o}$. The approach consists of estimating the real pose by minimising the error Δ between the observed data $\mathbf{s}^{*}$ (usually the position of a set of features in the image) and the position $\mathbf{s}$ of the same features computed by forward-projection according to the current pose,

$$\Delta = \sum_{i=1}^{k} \left(\mathrm{pr}_{\xi}({}^{c}\mathbf{M}_{o}, {}^{o}\mathbf{P}_{i}) - s_{i}^{*} \right)^{2}, \tag{1}$$

where $\mathrm{pr}_{\xi}()$ is the projection model according to the intrinsic parameters ξ and where k is the number of considered features. It is supposed here that intrinsic parameters ξ are available but it is possible, using the same approach, to also estimate these parameters.

In this formulation, a virtual camera initially at ${}^{c_0}\mathbf{M}_{o}$ is moved using a visual servoing control law in order to minimise the error Δ. At convergence, the virtual camera reaches the pose ${}^{c^*}\mathbf{M}_{o}$ which minimises the error and is considered as the real camera's pose).

2.1.2 Case of stereo system

Now consider a more general system with two cameras. We do not assume a rigid system but we consider that their relative positions with respect to each other are known.

$$\Delta = \sum_{i=1}^{k_1} \left(\mathrm{pr}_{\xi_1} ({}^{c_1}\mathbf{M}_{o}, {}^{o}\mathbf{P}_{i}) - {}^{c_1} s_{i}^{*} \right)^{2} + \sum_{j=1}^{k_2} \left(\mathrm{pr}_{\xi_2} ({}^{c_2}\mathbf{M}_{o}, {}^{o}\mathbf{P}_{j}) - {}^{c_2} s_{j}^{*} \right)^{2}, \tag{2}$$

where subscripted c_1 and c_2 refers to observations in images 1 and 2.

Solving for ${}^{c_1}\mathbf{M}_{o}$ and ${}^{c_2}\mathbf{M}_{o}$ is equivalent to consider two independent systems and is of no interest here. Since the calibration of the stereo system ${}^{c_2}\mathbf{M}_{c_1}$ is assumed to be known, equation (2) is equivalent to

$$\Delta = \sum_{i=1}^{k_1} \left(\mathrm{pr}_{\xi_1} ({}^{c_1}\mathbf{M}_{o}, {}^{o}\mathbf{P}_{i}) - {}^{c_1} s_{i}^{*} \right)^{2} + \sum_{j=1}^{k_2} \left(\mathrm{pr}_{\xi_2} ({}^{c_2}\mathbf{M}_{c_1} {}^{c_1}\mathbf{M}_{o}, {}^{o}\mathbf{P}_{j}) - {}^{c_2} s_{j}^{*} \right)^{2}, \tag{3}$$

so that only 6 parameters have to be estimated, as for the pose estimation problem. In any case, assuming that $\mathbf{r}$ is a vector representation of the pose (${}^{c}\mathbf{M}_{o}$ in (1) or ${}^{c_1}\mathbf{M}_{o}$ in (3)), this remains to minimise a residual Δ defined as

$$\Delta = \sum_{i=1}^{k} \left(s_{i}(\mathbf{r}) - s_{i}^{*} \right)^{2} = \| \mathbf{s}(\mathbf{r}) - \mathbf{s}^{*} \|^{2}. \tag{4}$$

2.1.3 Outliers rejection

An important assumption is to consider that $\mathbf{s}^{*}$ is computed from the image with sufficient precision. In visual servoing, the control law that performs the minimisation of Δ is usually handled using a least squares approach [6][9]. However, when outliers are present in the measures, a robust estimation is required. M-estimators can be considered as a more general form of maximum likelihood estimators [8]. They are more general because they permit the use of different minimisation functions not necessarily corresponding to normally distributed data. Many functions have been proposed in the literature which allow uncertain measures to be less likely considered and in some cases completely rejected. In other words, the objective function is modified to reduce the sensitivity to outliers. The robust optimisation problem is then given by

$$\Delta_{\mathcal{R}} = \sum_{i=1}^{k} \rho\Big(s_{i}(\mathbf{r}) - s_{i}^{*} \Big), \tag{5}$$

where $\rho(u)$ is a robust function [8] that grows sub-quadratically and is monotonically nondecreasing with increasing $|u|$. Iteratively Re-weighted Least Squares (IRLS) is a common method of applying the M-estimator. It converts the M-estimation problem into an equivalent weighted least-squares problem.

This objective function may be minimized using a virtual visual servoing scheme [15, 13, 2]. A control law that is robust to outlier has to be built in order to minimize equation (5). The duality between visual servoing and non-linear pose estimation is used to compute the current position of the multi-cameras system.

2.2 Robust Minimization

The objective of the control scheme is to minimise the objective function given in equation (5). Thus, the error to be regulated to 0 is defined as

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \tag{6}$$

where $\mathbf{D}$ is a diagonal weighting matrix given by $\mathbf{D} = \text{diag}(w_1, \ldots, w_k)$. Each element of $\mathbf{D}$ is a weight which is given to specify the confidence in each feature location. The computation of weights w_i is described in [2].

A simple control law that allows to move a virtual camera can be designed to try and ensure an exponential decoupled decrease of $\mathbf{e}$ around the desired position $\mathbf{s}^*$. It is given by:

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}\mathbf{L}_\mathbf{s}})^+\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big), \tag{7}$$

where $\mathbf{v}$ is the virtual camera velocity, $\mathbf{L_s}$ is called the interaction matrix and links the motion of the feature in the image to the camera velocity ($\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}$) and λ is a gain that tune the convergence rate. More details about the interaction matrix or image Jacobian is given in section 2.4. Let us point out that it is necessary to ensure that a sufficient number of features will not be rejected so that $\mathbf{DL_s}$ is always of full rank (6 to estimate the pose).

2.3 Considering Multiple Cameras

Considering the minimisation of equation (2) with two independent cameras leads to:

$$\begin{bmatrix}\dot{\mathbf{s}}_1\\ \dot{\mathbf{s}}_2\end{bmatrix} = \begin{bmatrix}\mathbf{L}_1 & 0\\ 0 & \mathbf{L}_2\end{bmatrix}\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix}. \tag{8}$$

Nevertheless, in the case of a calibrated multiple cameras system, if ${}^{c_2}\mathbf{M}_{c_1}$ is known and it is then possible to express ${}^1\mathbf{v}$ wrt. ${}^2\mathbf{v}$,

$${}^2\mathbf{v} = {}^{c_1}\mathbf{V}_{c_2}\,{}^1\mathbf{v} \quad \text{with} \quad {}^{c_1}\mathbf{V}_{c_2} = \begin{bmatrix}{}^{c_1}\mathbf{R}_{c_2} & [{}^{c_1}\mathbf{t}_{c_2}]_\times\\ \mathbf{0} & {}^{c_1}\mathbf{R}_{c_2}\end{bmatrix}, \tag{9}$$

where ${}^{c_1}\mathbf{V}_{c_2}$ is the twist transformation matrix. The feature velocity in image 2 can then be related to the motion of camera 1 by

$$\dot{\mathbf{s}}_2 = \mathbf{L}_2\mathbf{v}_2 = \mathbf{L}_2{}^{c_2}\mathbf{V}_{c_1}\,{}^1\mathbf{v} \quad \text{and} \quad \begin{bmatrix}\dot{\mathbf{s}}_1\\ \dot{\mathbf{s}}_2\end{bmatrix} = \begin{bmatrix}\mathbf{L}_1\\ \mathbf{L}_2{}^{c_2}\mathbf{V}_{c_1}\end{bmatrix}{}^1\mathbf{v}. \tag{10}$$

Finally we get the following control law, with only only 6 parameters to estimate,

$${}^1\mathbf{v} = -\lambda\begin{bmatrix}\widehat{\mathbf{D}_1\mathbf{L}_{\mathbf{s}1}}\\ \widehat{\mathbf{D}_2\mathbf{L}_{\mathbf{s}2}}{}^{c_2}\mathbf{V}_{c_1}\end{bmatrix}^+\begin{bmatrix}\mathbf{D}_1\\ \mathbf{D}_2\end{bmatrix}\begin{bmatrix}\mathbf{s_1}(\mathbf{r_1}) - \mathbf{s_1^*}\\ \mathbf{s_2}(\mathbf{r_2}) - \mathbf{s_2^*}\end{bmatrix}. \tag{11}$$

Let us note that in equation (11), two diagonal matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ have to be computed (see [2]) from residuals $\mathbf{s_1}(\mathbf{r_1}) - \mathbf{s_1^*}$ $\mathbf{s_2}(\mathbf{r_2}) - \mathbf{s_2^*}$ computed from each images. Since the position of the two camera wrt. to the object may be very different, the two residual vectors are also different and the mediane of each residual that is mainly considered in the computation of $\mathbf{D}_1$ and $\mathbf{D}_2$ has to be computed according to each data set. The pose ${}^{c_1}\mathbf{M}_o$ is then updated using the exponential map of $se(3)$ (see [12] p.33 for details) while the pose of the other camera is updated using the system parameters ${}^{c_1}\mathbf{M}_{c_2}$: ${}^{c_2}\mathbf{M}_o = {}^{c_2}\mathbf{M}_{c_1}{}^{c_1}\mathbf{M}_o$ and can then be used in equation (11) to compute $\mathbf{s_2}(\mathbf{r_2})$.

2.4 Visual Feature and Interaction Matrices

Any kind of geometrical feature can be considered within the proposed control law as soon as it is possible to compute its corresponding interaction matrix $\mathbf{L}$. In [6], a general framework to compute $\mathbf{L}$ is proposed. Indeed, it is possible to compute the pose from a large set of image information (points, lines, circles, quadratics, distances, etc...) within the same framework. The combination of different features is achieved by adding features to vector $\mathbf{s}$ and by stacking each feature's corresponding interaction matrix into a large interaction matrix of size $nd \times 6$ where n corresponds to the number of features and d their dimension,

$$\begin{bmatrix} \dot{\mathbf{s}}_1 \\ \vdots \\ \dot{\mathbf{s}}_n \end{bmatrix} = \begin{bmatrix} \mathbf{L}_1 \\ \vdots \\ \mathbf{L}_n \end{bmatrix} \mathbf{v}. \tag{12}$$

The redundancy yields a more accurate result with the computation of the pseudo-inverse of $\mathbf{L}$ as given in equation (7). Furthermore if the number or the nature of visual features is modified over time, the interaction matrix $\mathbf{L}$ and the vector error $\mathbf{s}$ is easily modified consequently. In this work, a set of distances between local point features obtained from a fast image processing step and the contours of a global 3D model are considered [2]. In this case the desired value of the distance is equal to zero. The derivation of the interaction matrix related to the distance between a fixed point and a moving straight line or moving cylinder to the virtual camera motion is given in [2]. Let us note that in [14] a distance between a point projected on the normal of the contour is considered as in [5]. This leads to a very different Jacobian. Difference between the two approach from a theoretical point of view is given in [1]

3 EXPERIMENTAL RESULTS

3.1 Experimental context: spatial robotics

As already mentioned, this research has been carried out for a project supported by European Space Agency (ESA). The goal of the VIMANCO project is to achieve grasping and maintenance tasks on the International Space Station (ISS). The solution proposed by the VIMANCO consortium is to achieve these tasks using visual servoing techniques.

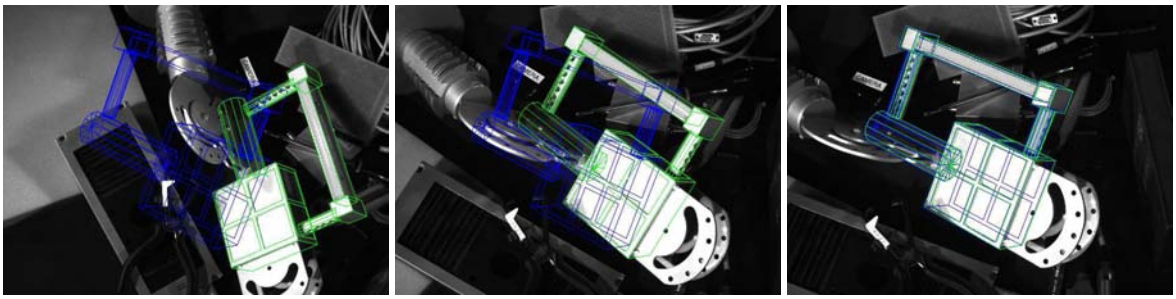
As this point the tracking and visual servoing capabilities have been tested at IRISA-INRIA Rennes using a classical 6-axis robot. Further tests using the Eurobot would be done within few weeks at ESA-ESTEC in the Netherlands on the ESA's ISS testbed. Within this paper, we consider an object named *Articulated Portable Foot Restraint* (APFR).

The following experiments consist in positioning tasks of the robot end effector with respect to the APFR by using a 3D visual servoing control laws and considering CCD camera setups:

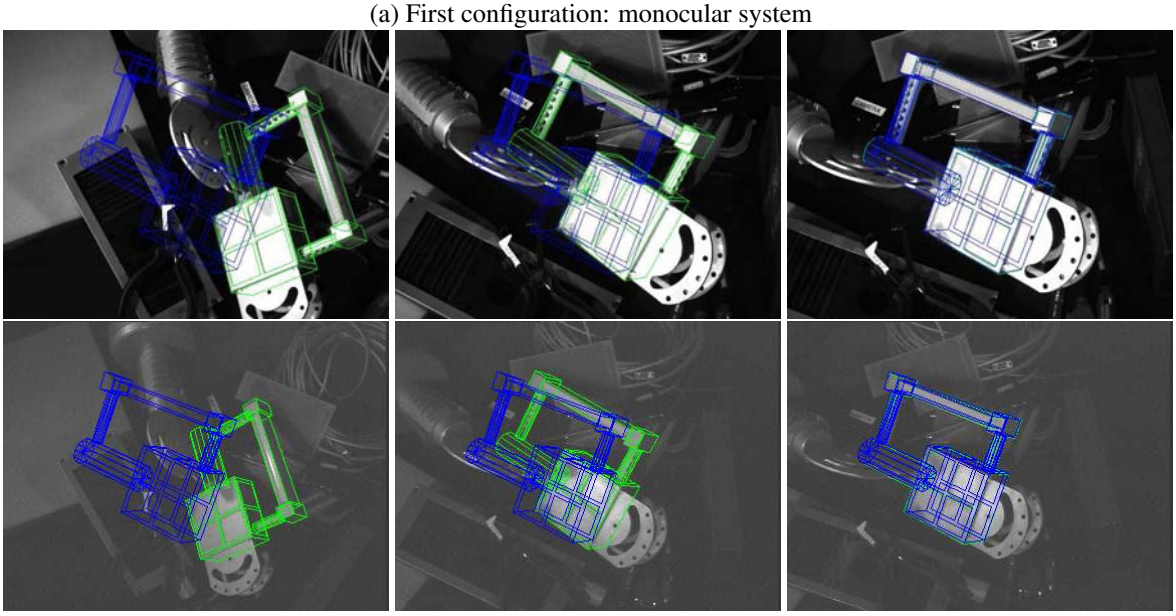
- *Monocular system*: The first experiment (see Figure 1a) was carried out by using a single camera mounted on the robot end effector. Results are shown by Figure 2.
- *Small base-line stereoscopic system*: The second experiment (see Figure 1b) was carried out by using two cameras mounted on the end effector. Results are shown by Figure 3.
- *Wide base-line stereoscopic system*: The third experiment (see Figure 1c) was carried out by using one camera mounted on the robot end effector and one fixed deported camera resulting in a wide baseline stereo system. Results are shown by Figure 4.

The robust model-based tracking method described in this paper is used to compute the current pose ${}^o\mathbf{M}_{c_1}$ with respect to object frame of the camera 1 mounted on the end effector, the goal being to move this camera to a desired pose ${}^o\mathbf{M}_{c_1^*}$. In each experiment, the initialisation phase consists in defining the desired ${}^o\mathbf{M}_{c_1^*}$ and initial ${}^o\mathbf{M}_{c_{c_1}^0}$ object poses with respect to the main camera frame.

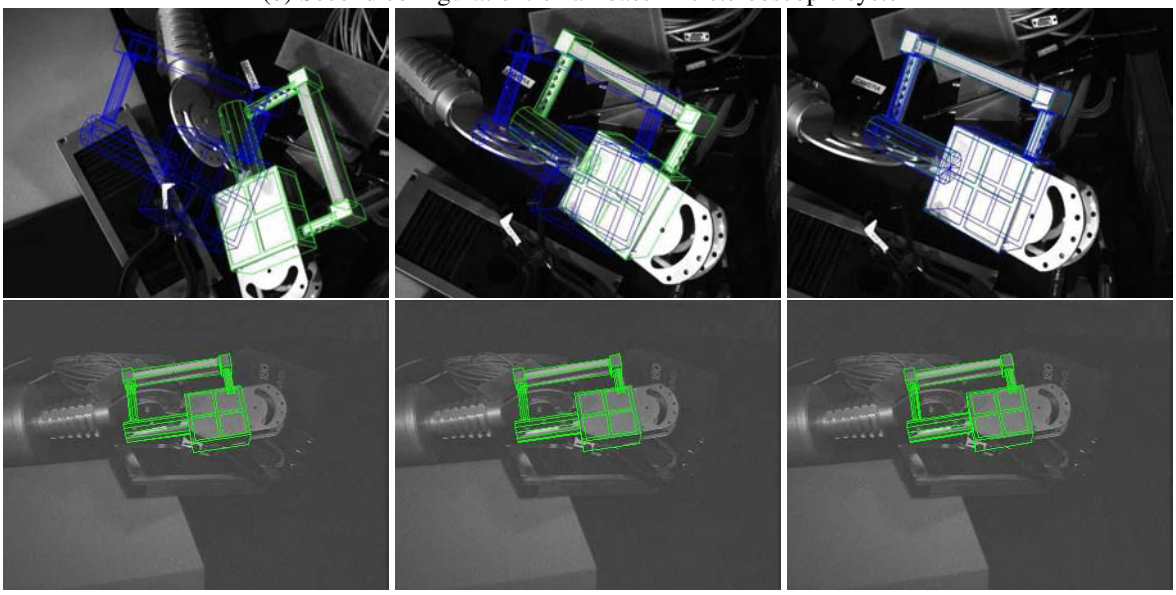
To validate the robustness of the proposed algorithm, the APFR was placed in a textured environment as shown in Figure 1. Moreover, partial auto-occlusions were caused due to the complexity of the geometry of this object. Indeed, due to computational cost, the considered CAD model is only partial and quite simplified. Shadow projections and reflexion artifacts were also appearing. In spite of all these sources of perturbation, tracking and positioning tasks were successfully achieved for each camera. configuration.



(a) First configuration: monocular system



(b) Second configuration: small base-line stereoscopic system

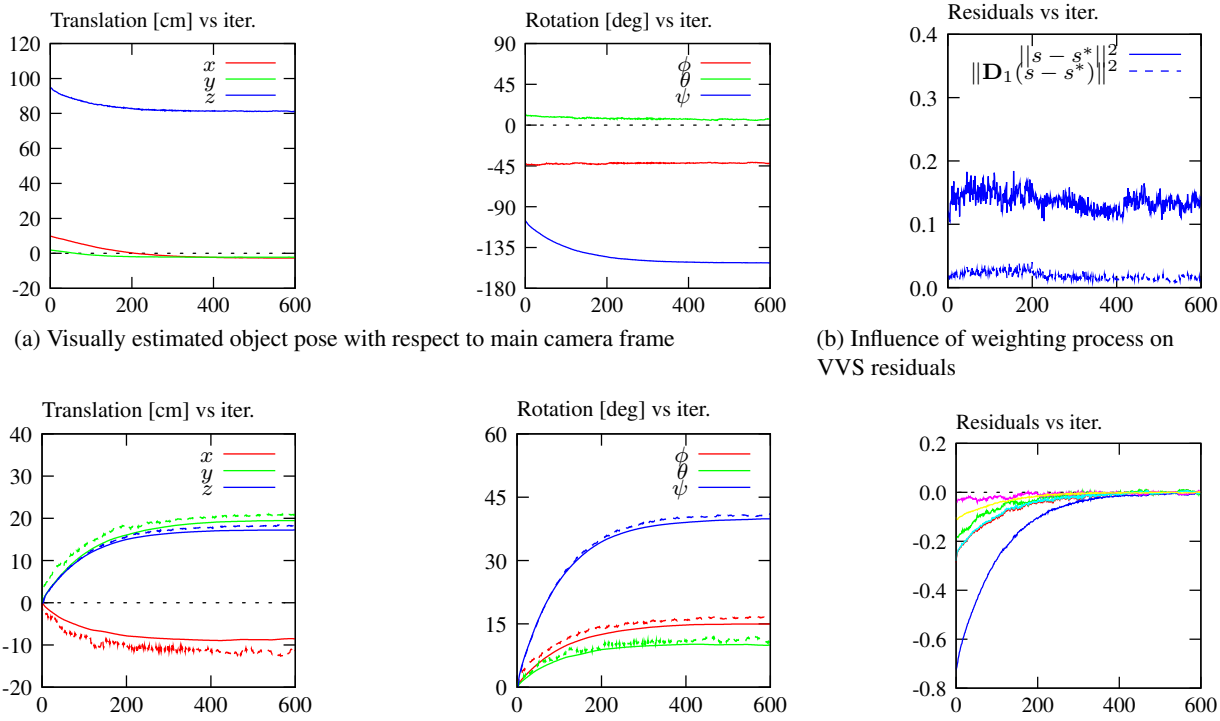


(c) Third configuration: wide base-line stereoscopic system

Figure 1: Snapshots extracted from experimental results (*green*: forward projected CAD model after pose calculation, *blue*: user defined desired position)

3.2 Results and discussion

Plotting results of the three previously described experiments are presented respectively in Figures 2, 3 and 4 which respect the following organisation.



(a) Visually estimated object pose with respect to main camera frame

(b) Influence of weighting process on VVS residuals

(c) Camera trajectory according to odometry (*solid lines*) and tracking (*dashed lines*)

(d) Convergence of the 3D visual servoing residuals

Figure 2: First experiment results: 3D visual servoing using robust model-based tracking for a monocular system setup

Plots (a) show the variation of the object pose with respect to the main camera (rotations are expressed thanks to Euler's angles). This is the direct output of the robust tracking algorithm and the input of the visual servoing control law used to controlled the manipulator. As expected from the real-time graphical display including the forward model projection, the tracking is smooth and consequently suitable for visual servoing applications.

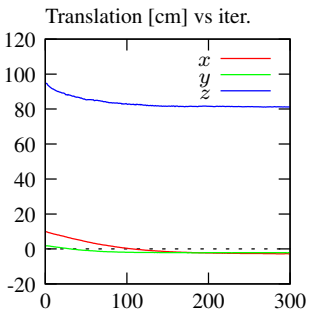
The residuals of the pose computation are shown by plots (b). This plots underline the interest of considering robust M-estimation within the minimization process. The lower level of the weighted residuals shows the efficiency of the convergence of the virtual visual servoing. The higher level of unweighted residuals shows that the pose would not be as accurate if a classical control law were used instead of a robust one. Moreover, unweighted residuals are computed at each iteration from the previous estimated pose which is robustly obtained. Without robust tracking we may observe a divergence and consequently the failure of the 3D visual servoing.

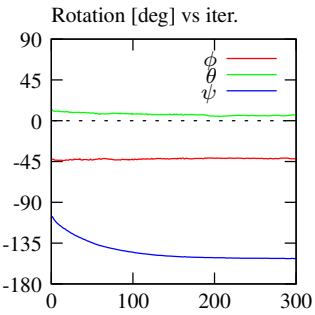
The efficiency of the robust tracking algorithm can also be analysed by comparing the trajectory of the camera during the positioning task computed from tracking data or from odometry data as done in plots (c). Indeed, there are two ways of estimating the matrix ${}^{c_1^{t_0}}\mathbf{M}_{c_1^t}$ giving the pose of the camera 1 with respect to its initial pose,

$$ {}^{c_1^{t_0}}\mathbf{M}_{c_1^t} = \begin{cases} {}^{c_1^{t_0}}\mathbf{M}_o \, {}^{o}\mathbf{M}_{c_1^t}, & \text{according to tracking,} \\ {}^{c_1^{t_0}}\mathbf{M}_{\mathcal{F}} \, {}^{\mathcal{F}}\mathbf{M}_{c_1^t}, & \text{according to odometry,} \end{cases} \tag{13} $$

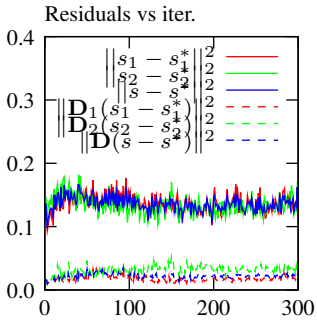
where subscripted $\mathcal{F}$ denotes the robot reference frame. The differences observed between the two measures can be explained by camera calibration errors. Indeed the current system is only roughly calibrated. Finally, plots (d) show the success of the global positioning task through the convergence of the 6 residuals of the 3D visual servoing control scheme.

In terms of time consumption (on a 2.4Gz pentium 4), it is obvious that the algorithm in configurations with two cameras is slower due to the fact that two images have to be processed simultaneously. Assuming simple objects, the proposed algorithm can easily acquire and process one image at the video rate of 50 Hz. In the case of the APFR, the algorithm needs to track a quite high number of sample points (around 250 in each image), the processing rate is then 20 Hz for a monocular system and 10 Hz in the stereovision case. Nevertheless, this is not really a strong limitation in the context described in this paper since slow motions are absolutely required in on-board or extra-vehicular space operations (for safety issues).

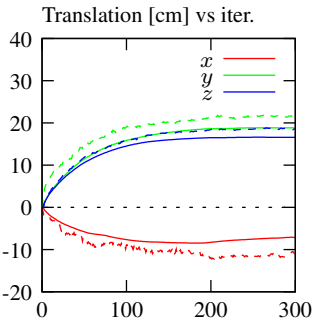


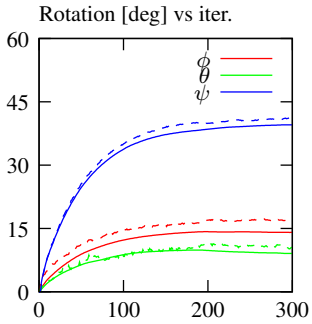


(a) Visually estimated object pose with respect to main camera frame

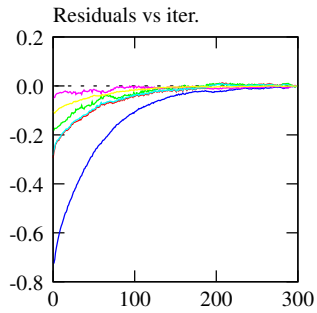


(b) Influence of weighting process on VVS residuals



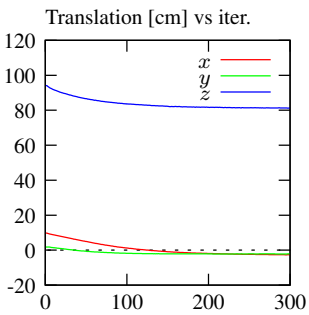


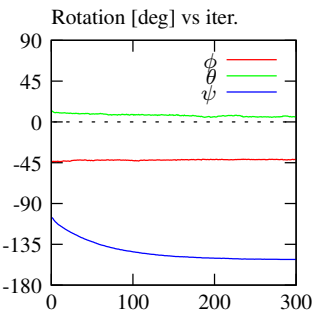
(c) Camera trajectory according to odometry (*solid lines*) and tracking (*dashed lines*)



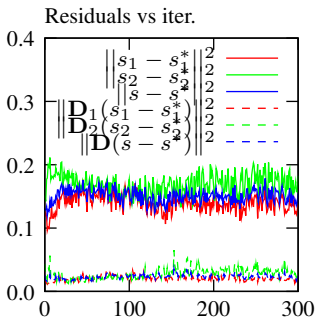
(d) Convergence of the 3D visual servoing residuals

Figure 3: Second experiment results: 3D visual servoing using robust model-based tracking for a small base-line system setup

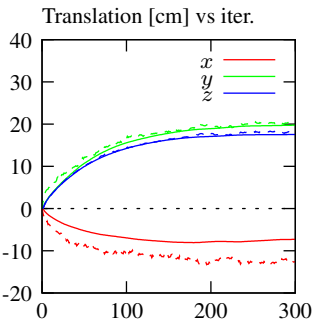


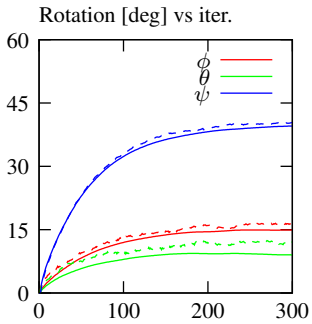


(a) Visually estimated object pose with respect to main camera frame

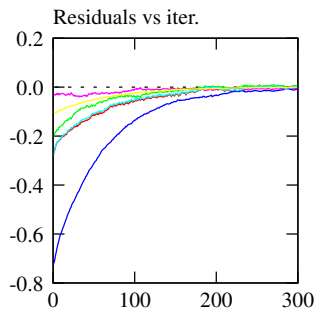


(b) Influence of weighting process on VVS residuals





(c) Camera trajectory according to odometry (*solid lines*) and tracking (*dashed lines*)



(d) Convergence of the 3D visual servoing residuals

Figure 4: Third experiment results: 3D visual servoing using robust model-based tracking for a wide base-line system setup

Videos Videos are available in the demo section of the Lagadic www site (`http://www.irisa.fr/lagadic`)

Acknowlegment

This work is supported by the European Space Agency through the VIMANCO ITT project. Author whish to thank ESA who provides the APFR.

References

[1] A.I. Comport, D. Kragic, E. Marchand, and F. Chaumette. Robust real-time visual tracking: Comparison, theoretical analysis and performance evaluation. In *IEEE ICRA'05*, pages 2852–2857, Barcelona, Spain, April 2005.

[2] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE T. on Visualization and Computer Graphics*, 12(4):615–628, July 2006.

[3] D. Dementhon and L. Davis. Model-based object pose in 25 lines of codes. *IJCV*, 15(1-2):123–141, 1995.

[4] M. Dhome, M. Richetin, J.-T. Lapresté, and G. Rives. Determination of the attitude of 3D objects from a single perspective view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.

[5] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *PAMI*, 24(7):932–946, July 2002.

[6] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[7] R. Haralick, H. Joo, C. Lee, X. Zhuang, V Vaidya, and M. Kim. Pose estimation from corresponding point data. *IEEE Trans on Systems, Man and Cybernetics*, 19(6):1426–1445, November 1989.

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

[9] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[10] D.G. Lowe. Fitting parameterized three-dimensional models to images. *PAMI*, 13(5):441–450, May 1991.

[11] C.P. Lu, G.D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *PAMI*, 22(6):610–622, June 2000.

[12] Y. Ma, S. Soatto, J. Košecká, and S. Sastry. *An invitation to 3-D vision*. Springer, 2004.

[13] E. Marchand and F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS'02 Conf. Proceeding*, pages 289–298, Saarebrücken, Germany, September 2002.

[14] F. Martin and R. Horaud. Multiple camera tracking of rigid objects. *IJRR*, 21(2):97–113, February 2002.

[15] V. Sundareswaran and R. Behringer. Visual servoing-based augmented reality. In *IEEE Int. Workshop on Augmented Reality*, San Francisco, November 1998.