Stereo Tracking and Servoing for Space Applications

Fabien Dionnet Eric Marchand

INRIA Rennes-Bretagne Atlantique, IRISA, Lagadic, F-35000 Rennes, France ;

Email: `Eric.Marchand@irisa.fr`

published in " Advanced Robotics, 23(5):579-599, Avril 2009."

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 [8, 4, 12], contours or points on the contours [11, 3, 6], segments, straight lines, conics, cylindrical objects, or a combination of these different features [14]. Another important issue is the registration problem. *Purely geometric* (eg, [5]), or *numerical and iterative* [4] approaches may be considered. *Linear approaches* use a least-squares method to estimate the pose. *Full-scale non-linear optimisation techniques* (e.g., [8, 11, 6, 3, 16]) 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 [17, 14, 3]. 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 [9]). This outlier rejection process is directly introduced in the control law leading to an iterated reweighted least squares problem [3]. 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 [3] (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 [16]. 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, in [16], 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 applications in a space context. The goal is to develop a robot demonstrator able to grasp complex objects by visual servoing in space environment. The considered robot is the ESA's three-armed Eurobot (see Figure 1) whose purpose is to prepare and assist extra vehicular activities on the International Space Station (ISS). The fact that this robot is equipped with one camera mounted on each arm end effector and one stereovision system mounted on its base allows to consider tracking and visual servoing tasks using various camera configuration: one, two or more cameras, with short or wide baseline, in eye-in-hand or eye-to-hand control schemes [10].

We also want to insist on the fact that the presented tracking algorithm has to tackle several space specific problems. In particular we tried to simulate in experiments the severe and abruptly changing lighting conditions due to direct sunlight and celestial mechanics that make objects appear very bright while important cast shadows are moving. The algorithm have therefore to be highly robust in spite of another major space problem which is the lack of computing power. Indeed, resource (i.e. energy, volume, mass) and environmental (i.e. thermal dissipation, radiation compatibility) constraints limit performance of computers that may be used in space.

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 [7, 10]. 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 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 ${}^o\mathbf{S}$ (for instance, ${}^o\mathbf{S}$ 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 of the object 3D model in the image plane according to the current pose,

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

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.

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

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

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

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

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

Dealing with the specific system presented in this paper the definitions of $\mathbf{s}(\mathbf{r})$ and $\mathbf{s}^*$ are given in section 2.4.

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 [7][10]. 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 [9]. 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 [9] that grows sub-quadratically and is monotonically non decreasing 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 [17, 14, 3]. 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 a measure of the confidence that a point is an inlier. The computation of weights w_i is described in appendix A and in [3].

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

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

where $\mathbf{v}$ is the virtual camera velocity, $\mathbf{L_s}$ is called the interaction matrix (or image Jacobian) 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 tunes the convergence rate[1]. More details about the interaction matrix 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 $\mathbf{D}\mathbf{L_s}$ is always of full rank (6 to estimate the pose). Let us note that when no robust minimization is considered (i.e. $\mathbf{D} = \mathbf{I}$) this process is similar to a Gauss-Newton minimization approach.

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 ${}^1\mathbf{v}$ with respect to ${}^2\mathbf{v}$,

$${}^2\mathbf{v} = {}^{c_1}\mathbf{V}_{c_2}\,{}^1\mathbf{v} \tag{9}$$

with

$${}^{c_1}\mathbf{V}_{c_2} = \begin{bmatrix} {}^{c_1}\mathbf{R}_{c_2} & [{}^{c_1}\mathbf{t}_{c_2}]_\times \\ \mathbf{0} & {}^{c_1}\mathbf{R}_{c_2} \end{bmatrix}, \tag{10}$$

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

$$\dot{\mathbf{s}}_2 = \mathbf{L}_2\mathbf{v}_2 = \mathbf{L}_2\,{}^{c_2}\mathbf{V}_{c_1}\,{}^1\mathbf{v} \tag{11}$$

[1] $\mathbf{A}^+$ denotes the pseudo-inverse of the matrix $\mathbf{A}$

and

$$\begin{bmatrix} \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 {}^{c_2}\mathbf{V}_{c_1} \end{bmatrix} {}^1\mathbf{v}. \tag{12}$$

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

$${}^1\mathbf{v} = -\lambda \begin{bmatrix} \mathbf{D}_1 \mathbf{L}_{\mathbf{s}1} \\ \mathbf{D}_2 \mathbf{L}_{\mathbf{s}2} {}^{c_2}\mathbf{V}_{c_1} \end{bmatrix}^+ \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{s_1}(\mathbf{r_1}) - \mathbf{s_1^*} \\ \mathbf{s_2}(\mathbf{r_2}) - \mathbf{s_2^*} \end{bmatrix}. \tag{13}$$

Let us note that in equation (13), two diagonal matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ have to be computed (see [3]) from residuals $\mathbf{s_1}(\mathbf{r_1}) - \mathbf{s_1^*}$ and $\mathbf{s_2}(\mathbf{r_2}) - \mathbf{s_2^*}$ computed from each images. Since the position of the two cameras with respect to the object may be very different, the two residual vectors are also different and the median value 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 [13] p.33 for details)

$${}^{c_1}\mathbf{M}_o^{t+1} = {}^{c_1}\mathbf{M}_o^t e^{[{}^1\mathbf{v}]} \tag{14}$$

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 (13) to compute $\mathbf{s_2}(\mathbf{r_2})$.

2.4 Visual feature and interaction matrices

Any kind of geometrical features can be considered within the proposed control law as soon as it is possible to compute its corresponding interaction matrix $\mathbf{L}$. In [7], 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,

$$\begin{bmatrix} \dot{\mathbf{s}}_1 \\ \vdots \\ \dot{\mathbf{s}}_n \end{bmatrix} = \begin{bmatrix} \mathbf{L}_1 \\ \vdots \\ \mathbf{L}_n \end{bmatrix} \mathbf{v}. \tag{15}$$

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, we consider a set of distances between local point features obtained from a fast image processing step and the contours of a global 3D model. In this case the desired value of the distance is equal to zero. In Figure 2, $\mathbf{p}$ is the tracked point feature position and $\mathbf{l}(\mathbf{r})$ is the projection of a 3D model line in the image plane accordint to pose $\mathbf{r}$.

From a low level image processing point of view, normal displacements are evaluated along the projection of the object model contours using the spatio-temporal Moving Edges algorithm (ME) [1]. It consists in finding the nearest intensity discontinuity along the edge normal using a pre-computed mask function of the orientation of the contour. 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 [3].

Let us note that [16, 6] the tracked point $\mathbf{p}$ are projected in the direction of the contour normal so as to represent different rigid motion parameters as components of a distance [6]. This leads to a very different Jacobian. Difference between the two approaches from a theoretical point of view is given in [2]

3 Experimental results

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.

Any visual servoing control law can be implemented using the presented tracker (image-based, position-based or hybrid scheme) because all 3D pose information has been estimated. In the following experiments the classical 3D visual servoing approach is used [18]. Knowing current object pose ${}^{c}\mathbf{M}_{o}$ and desired one ${}^{c^*}\mathbf{M}_{o}$ the goal of this approach is then to minimize the error ${}^{c}\mathbf{M}_{c^*}$. The complete control law is implemented using the ViSP (Visual servoing platform) software [15].

The presented approach is different from a classical look and move method where the object is localized in the image and where the robot is moved using only this information. Such approach is not robust to calibration errors or to modification of the environment. In visual servoing since visual features and then the control law are computed for each new image acquired by the camera, the object is not necessarily motionless and the robot may overcome partial modification of the environment during the execution of the task. This is not the case with a look and move approach. A coarse camera and eye-hand calibration is sufficient in the case where the task is specified as a particular position of the object in the image. In practice, this is obtained using an off-line teaching by showing step where the end-effector is moved once at its desired position with respect to the object and the corresponding image is stored. In that case, the data extracted from the vision sensor will be biased due to the calibration errors, but the robustness of the visual servoing with respect to calibration errors will allow to move accurately the arm so that the final image corresponds to the desired one, ensuring a correct realization of the task.

3.1 Results obtained at IRISA and discussion

The tracking and visual servoing capabilities have been tested at IRISA-INRIA Rennes using a classical 6-axis robot. Within this paper, we consider first an object called *Articulated Portable Foot Restraint* (APFR). This is a quite complex non-polyhedric object as can be seen in Figure 3.

The first experiments consists in positioning tasks of the robot end effector with respect to the APFR by using a 3D visual servoing control law [18].

The robust model-based tracking method described in this paper is used to compute the current pose ${}^{o}\mathbf{M}_{c_1}$ of the camera 1 mounted on the end effector with respect to object frame, 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

${}^{o}\mathbf{M}_{c_1^0}$ poses.

To validate the robustness of the proposed algorithm, the APFR was placed in a textured environment as shown in Figures 4, 5 6. Moreover, partial auto-occlusions were caused due to complex 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.

The following experiments consist in a positioning task of the robot end effector with respect to the APFR by using a 3D visual servoing for different CCD camera setups:

- *Monocular system*: The first experiment (see Figure 4a) was carried out by using a single camera mounted on the robot end effector. Results are shown by Figure 7.
- *Small base-line stereoscopic system*: The second experiment (see Figure 5b) was carried out by using two cameras mounted on the end effector. Results are shown by Figure 8.
- *Wide base-line stereoscopic system*: The third experiment (see Figure 6c) 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 9.

Plotting results of the three previously described experiments are presented respectively in Figures 7, 8 and 9 which respect the following organisation.

Figures 6-7-8(a) show the variation of the object pose with respect to the main camera (rotations are expressed using Euler's angles). This is the direct output of the robust tracking algorithm and the input of the visual servoing control law used to control 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 Figures 6-7-8(b) which underlines the interest of considering robust M-estimation within the minimization process. The low 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 robot odometry (position calculated using the encoders in the joints) as done in Figures 6-7-8(c). In both case the camera displacement is computed in the same frame that corresponds to the initial camera location. Since the odometry of this particular robot is very precise we can consider it as a ground truth.

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,

$$ {}^{c_1^{t_0}}\mathbf{M}_{c_1^t} = \begin{cases} {}^{c_1^{t_0}}\mathbf{M}_o \, {}^{o}\mathbf{M}_{c_1^t}, & \text{according to tracking,} \\ {}^{c_1^{t_0}}\mathbf{M}_{\mathcal{F}} \, {}^{\mathcal{F}}\mathbf{M}_{c_1^t}, & \text{according to odometry,} \end{cases} \tag{16} $$

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. As already pointed although the pose is biased this as no effect on the positioning task since the final and desired image corresponds.

Finally, Figure (d) shows the success of the global positioning task through the convergence of the 6 residuals of the 3D visual servoing control scheme.

Main advantage of the stereo tracker is that it is more robust to partial occlusion and more reliable in an operational context (especially with a large baseline since different aspect of the target are observed). In terms of time consumption (on a 2.4 GHz 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 (0.5cm/s in translation) are absolutely required in on-board or extra-vehicular space operations (for safety issues).

Another considered object is a handrail. Handrails are located all around the Columbus laboratory as shown in Figure 11. A similar experiment has been done using the handrail and is reported in Figure 10. Here we wanted to test the robustness wrt. illumination changes and occlusions. The sequences feature cast shadows, severe lighting variations, modification of the position of the lights, saturation, various occlusions, etc.

3.2 Results on the Eurobot Testbeds

Further tests using the Eurobot have be done at ESA-ESTEC (in Noordwijck, the Netherlands) on the ISS testbed composed by the Columbus laboratory 1:1 mockup. As can be seen on Figure 11 the Eurobot prototype made by Galileo Avionica (It.) while the integration of our tracking and visual servoing software on Eurobot has been done by Trasys (Be.). The robot itself is made of three mistubishi PA-10. A complete set of experiment has been done on this testbed (see Figure 12).

A second prototype Eurobot, the Eurobot wetmodel (see figure 13), is located at (and has been built by) Thales Alenia Space in Torino (TAS-I). Experiments have been carried out on this robot (see figure 14) and extensive tests (reported in this section) have been done on the Wet Model.

The experiments at TAS-I using the Eurobot wetmodel demonstrated the application of Visual Servoing using VIMANCO. Two experiments have reported here, In the first experiment, at the desired position, the arm is "parallel" to the handrail (see Figure 2 3b) and therefore only the top face is visible. The initial position is the one illustrated in Figure 14a. In the second experiment the final position is not parallel to the handrail (see desired position in Figure 14b and Figure 14c). The main difference is that two more faces of the handrail are viewed from the camera.

The Visual Servoing experiments are performed as follows:

- The robot arm is driven to a selected position close to the handrail.

- Based on the Eurobot wetmodel controller (GNC) information the position of the end-effector w.r.t. the world frame is computed (this position is referred as desired position).
- At the desired position an image is acquired. Using this image, the position of the handrail wrt the camera is computed (final position) in an interactive way.
- The arm is moved using the GNC at an arbitrary initial position. At this position the Visual Servoing is applied in order to move the camera at the previously defined final position wrt the handrail.
- At the final position the GNC provides the end-effector position wrt the World frame. This position is indicated as test reached position. A set of positioning tasks have been carried out toward the same desired position. Each positioning task has been repeated a certain number of times in order to evaluate visual servoing accuracy and repeatability.
- The accuracy of the positioning task using visual servoing is evaluated by the mean absolute difference between the desired position and the test result position.
- The repeatability is evaluated by computing the standard deviation of the test result position of consecutive tests towards the same desired position.

Let us finally note that the a very poor camera and hand-eye calibration has been considered.

Tables 1 and 4 presents the desired and reached position for the two experiment ; Tables 2 and 5 presents presents the absolute difference between each test result and the desired position ; Tables 3 and 6 presents the mean absolute error per direction and the standard deviation between the test results.

	Rx(rad)	Ry(rad)	Rz(rad)	Tx (m)	Ty(m)	Tz(m)
Desired position	-0,2662	-1,4468	-0,2061	-5,127	0,0476	-0,3026
Test 1 reached position	-0,2718	-1,4557	-0,1715	-5,1257	0,0404	-0,3075
Test 2 reached position	-0,269	-1,4527	-0,2136	-5,1254	0,0502	-0,3034
Test 3 reached position	-0,2675	-1,448	-0,2119	-5,1268	0,0498	-0,3027
Test 4 reached position	-0,2697	-1,453	-0,2105	-5,1254	0,0498	-0,3032

Table 1: Experiment 1: raw results

	Rx(rad)	Ry(rad)	Rz(rad)	Tx (m)	Ty(m)	Tz(m)
Test 1	0,0056	0,0089	0,0346	0,0013	0,0072	0,0049
Test 2	0,0028	0,0059	0,0075	0,0016	0,0026	0,0008
Test 3	0,0013	0,0012	0,0058	0,0002	0,0022	0,0001
Test 4	0,0035	0,0062	0,0044	0,0016	0,0022	0,0006

Table 2: Experiment 1: Absolute error per test

From the experimental results we can draw the following conclusions:

	Rx(rad)	Ry(rad)	Rz(rad)	Tx (m)	Ty(m)	Tz(m)
Mean error	0,0033	0,00555	0,013075	0,001175	0,00355	0,0016
Standard deviation	0,001787	0,003198	0,014406	0,000665	0,002441	0,00222

Table 3: Experiment 1: mean error and standard deviation per direction

	Rx(rad)	Ry(rad)	Rz(rad)	Tx (m)	Ty(m)	Tz(m)
Desired position	-0,0069	-1,5661	-0,223	-5,1024	-0,0037	-0,3361
Test 1 reached position	-0,0079	-1,5612	-0,2153	-5,104	-0,0052	-0,3351
Test 2 reached position	-0,012	-1,5877	-0,2149	-5,0962	-0,0051 0,338	
Test 3 reached position	-0,0107	-1,5851	-0,2119	-5,0971	-0,0061	-0,338
Test 4 reached position	-0,0122	-1,5892	-0,226	-5,0959	-0,0023	-0,3386
Test 5 reached position	-0,0017	-1,5531	-0,1819	-5,1058	-0,0146	-0,3361
Test 6 reached position	-0,0009	-1,5531	-0,1655	-5,1065	-0,019	-0,3373

Table 4: Experiment 2: raw results

	Rx(rad)	Ry(rad)	Rz(rad)	Tx (m)	Ty(m)	Tz(m)
Test 1	0,001	0,0049	0,0077	0,0016	0,0015	0,001
Test 2	0,0051	0,0216	0,0081	0,0062	0,0014	0,0019
Test 3	0,0038	0,019	0,0111	0,0053	0,0024	0,0019
Test 4	0,0053	0,0231	0,003	0,0065	0,0014	0,0025
Test 5	0,0052	0,013	0,0411	0,0034	0,0109	0
Test 6	0,006	0,013	0,0575	0,0041	0,0153	0,0012

Table 5: Experiment 2: absolute error per test

	Rx(rad)	Ry(rad)	Rz(rad)	Tx (m)	Ty(m)	Tz(m)
Mean error	0, 0044	0,015767	0,021417	0,004517	0,005483	0,001417
Standard deviation	0,001812	0,006807	0,022363	0,001861	0,006073	0,00088

Table 6: Experiment 2: mean error and standard deviation per direction

- Visual Servoing using the eye-in-hand configuration (the camera attached on the end-effector can be applied with a very poor camera and hand-eye calibration.
- The accuracy of the positioning tasks has been identified to be in the expected ranges (wrt to European Space Agency requirements) for the particular object. Typically, mean accuracy error is around 5 millimetres. Worst values have been identified to be 1 centimeter wrt a particular direction in 3 measured tests in a sequence of 17 measured tests. Indeed, the shape of the object is such that a small rotation around the x axes (camera frame) compensated by a translation along the y axes does not modify significantly the position of the object in the image. But the computed 3D position of the camera will be different. In

addition, the rotation around the longitudinal axis of the handrail is difficult to estimate contributing therefore to the positioning errors. Possible approaches that could be considered to handle this particularity of the handrail are to take into account the a priori knowledge of the orientation around this axis or to use additional exteroceptive information.

- The repeatability of the positioning task using Visual Servoing is very good considering the nature of the object. For example, in the first experiment, the standard deviation is 0.655, 2.441 and 2.222 millimetres for the translation.

- For the particular object, the final desired position of the camera wrt the object is important since a different number of faces of the object can be seen from different directions. Typical example is top view of the handrail where few 3D visual information is available.

4 Conclusion

We have presented a robust model-based tracking algorithm able to consider information provided by multiple cameras. The efficiency of the approach has been demonstrated by the integration of the proposed tracker in a visual servoing system. The presented method allows fast and accurate positioning of a eye-in-hand robot with respect to real objects (without any landmarks) in complex situations. The algorithm has been tested in the space context on various real visual servoing scenarios demonstrating a real usability of this approach under nominal and extreme lighting conditions.

A Weights computation for robust estimation

The weights w_i, which represent the different elements of the $\mathbf{D}$ matrix and reflect the confidence of each feature, are given by [9]:

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}, \tag{17}$$

where $\psi\Big(u\Big) = \frac{\partial\rho\big(u\big)}{\partial u}$ is the influence function and δ_i is the normalized residue given by $\delta_i = \Delta_i - Med(\Delta)$ (where $Med(\Delta)$ is the median operator) and σ is the standard deviation of the inliers data computing using the Median Absolute Deviation [9].

Of the various loss and corresponding influence functions that exist in the literature, Tukey's hard re-descending function has been chosen. Tukey's function completely rejects outliers and gives them a zero weight. This is of interest in tracking applications so that a detected outlier has no effect on the virtual camera motion. This influence function is given by:

$$\psi(u) = \begin{cases} u(C^2 - u^2)^2 & \text{, if } |u| \leq C \\ 0 & \text{, else,} \end{cases} \tag{18}$$

where the proportionality factor for Tukey's function is $C = 4.6851$ and represents 95% efficiency in the case of Gaussian noise.

Acknowledgment

This work is supported by the European Space Agency through the VIMANCO ITT project. The authors wish to thank ESA for providing the APFR and the handrail. Trasys (especially Konstantinos Kapellos) have also to be thanked for the integration of the tracking and visual servoing software on the Eurobot prototype.

Video

Video could be find at the following url:

`http://www.irisa.fr/lagadic/demo/demo-vimanco/demo-vimanco.html`

REFERENCES

[1] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(5):499–511, May 1989.

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

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

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

[5] M. Dhome, M. Richetin, J.-T. Laprest, 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.

[6] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[7] 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.

[8] 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.

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

[10] 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.

[11] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[12] C.P. Lu, G.D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(6):610–622, June 2000.

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

[14] E. Marchand and F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In G. Drettakis and H.-P. Seidel, editors, *EUROGRAPHICS'02 Conf. Proceeding*, volume 21(3) of *Computer Graphics Forum*, pages 289–298, Saarebrücken, Germany, September 2002.

[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 Robotics and Automation Magazine*, 12(4):40–52, December 2005. Special Issue on "Software Packages for Vision-Based Control of Motion", P. Oh, D. Burschka (Eds.).

[16] F. Martin and R. Horaud. Multiple camera tracking of rigid objects. *Int. Journal of Robotics Research*, 21(2):97–113, February 2002.

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

[18] W. Wilson, C. Hulls, and G. Bell. Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696, October 1996.

Fabien Dionnet was born and educated in France. In 2001 he received an engineer's degree in automation from the École nationale supérieure de physique de Strasbourg (ENSPS) and a master's degree in control, vision & robotics from the Université Louis Pasteur in Strasbourg. In 2005 he received a doctoral degree in robotics from the Université Pierre et Marie Curie in Paris.

Fabien Dionnet prepared from 2001 to 2005 a doctoral thesis on microrobotics & telemicromanipulation in the former Laboratoire de robotique de Paris (LRP), now part of the Institut des systmes intelligents et de robotique (ISIR), Paris, France. During the same period, he was also teaching assistant at the University of Versailles in the department of telecommunications. From 2005 to 2007 he was expert engineer at INRIA Rennes-Bretagne Atlantique in the Lagadic team, working on a space robotic vision project for the European space agency (ESA). Since early 2008, he have been working as a postdoctoral researcher in the Robotics, brain & cognitive sciences department of the Italian institute of technology (IIT), Genoa, Italy.

Eric Marchand received the PhD degree and "habilitation diriger des recherches" in computer science from the Université de Rennes 1 in 1996 and 2004, respectively. He spent one year as a postdoctoral associate in the AI lab of the Department of Computer Science at Yale University. Since 1997, he has been an INRIA research scientist ("chargé de recherche") at INRIA Rennes-Bretagne Atlantique in the Lagadic project. His research interests include robotics, perception strategies, visual servoing and real-time object tracking. He is also interested in the software engineering aspects of robot programming. More recently, he studies new application fields for visual servoing such as augmented reality and computer animation.

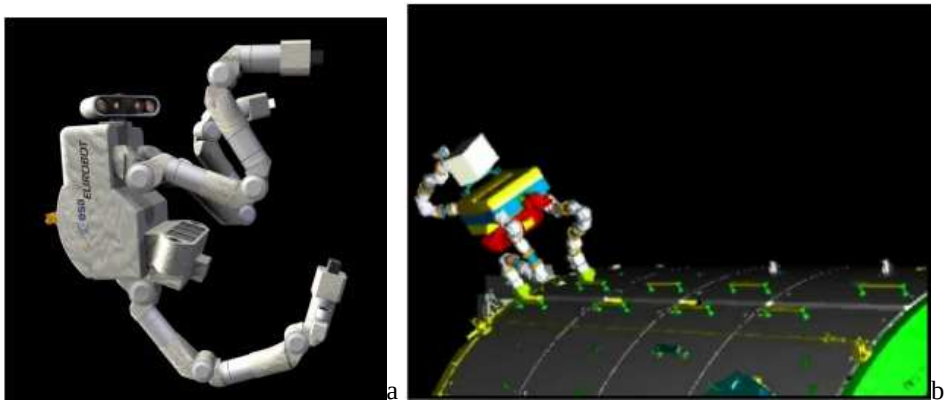


Figure 1: ESA's three-armed Eurobot (a) artist view (b) eurobot walking on the ISS (image courtesy of ESA).

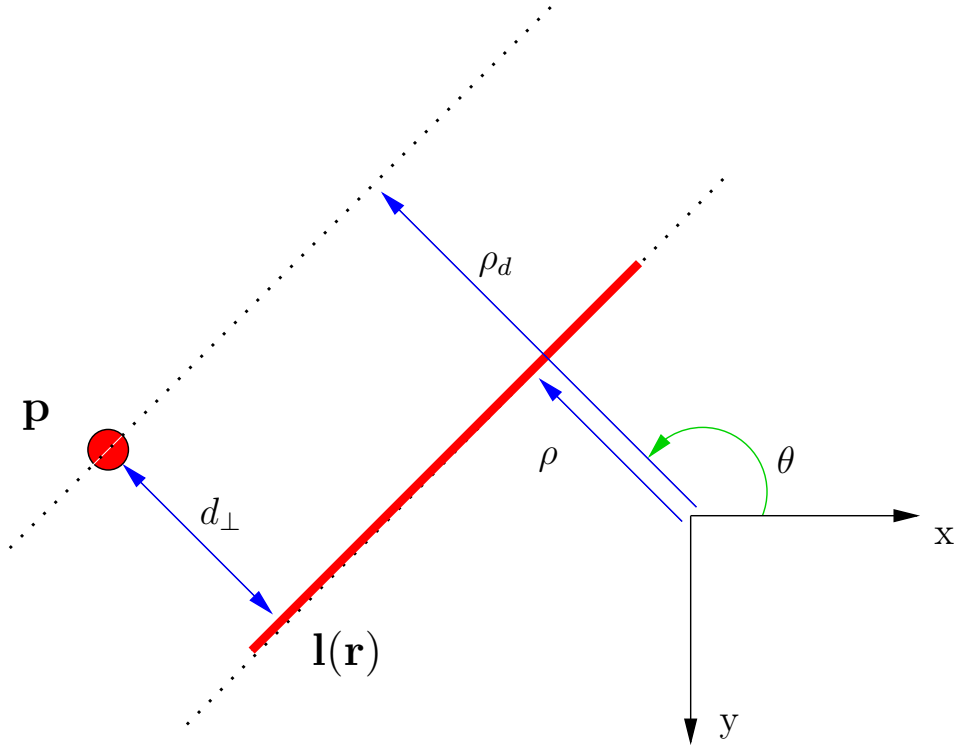


Figure 2: Distance of a point to a line

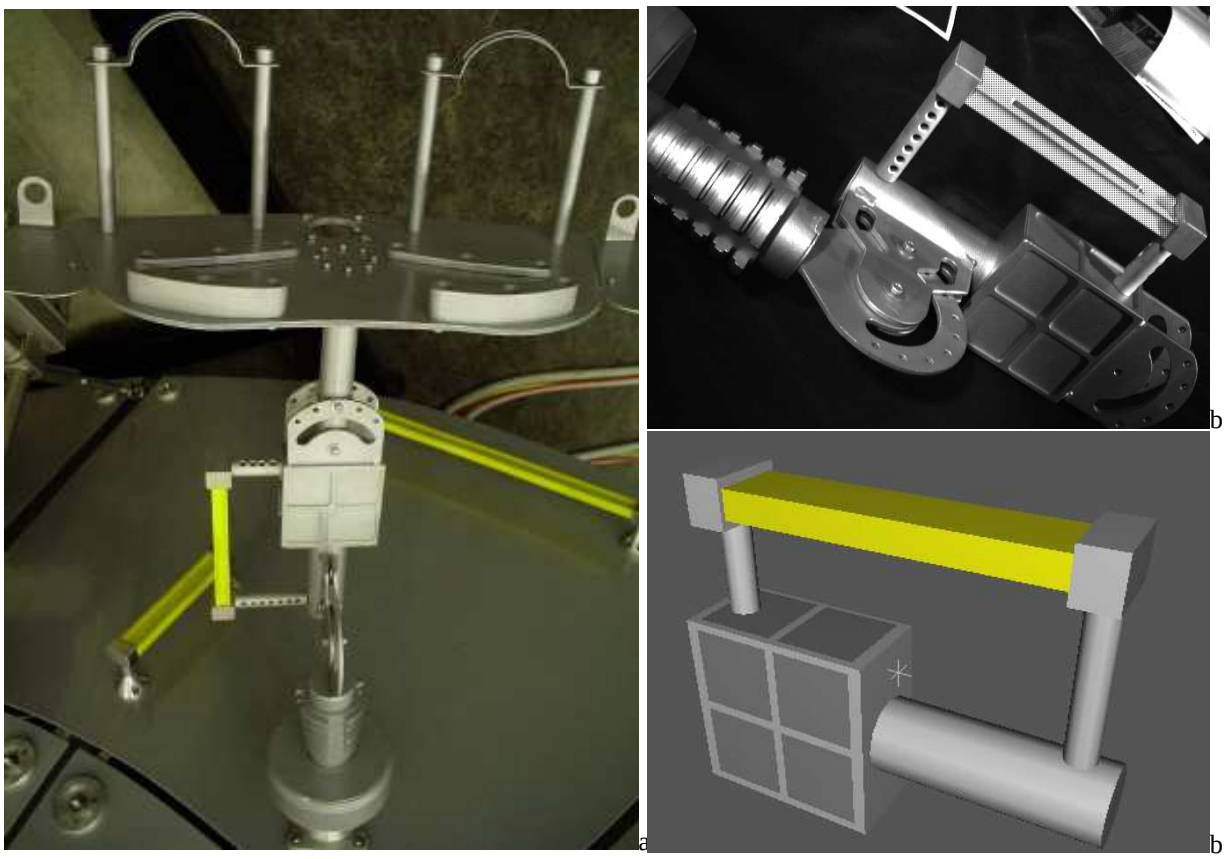


Figure 3: Articulated Portable Foot Restraint (APFR) used on the International Space Station by ESA astronauts: (a-b) real view of the object, (c) CAD model used for tracking (image (a) courtesy of ESA)

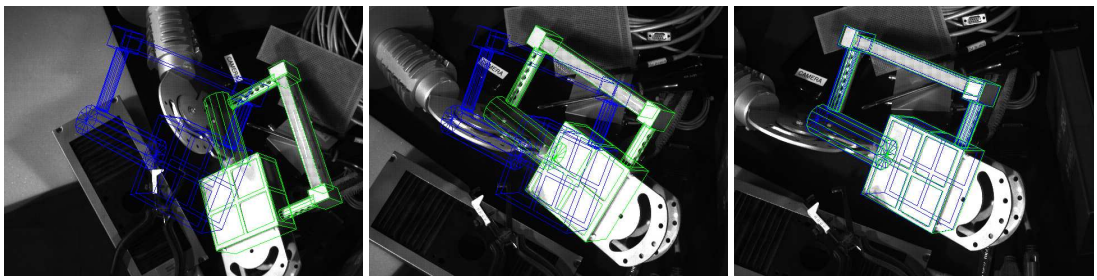
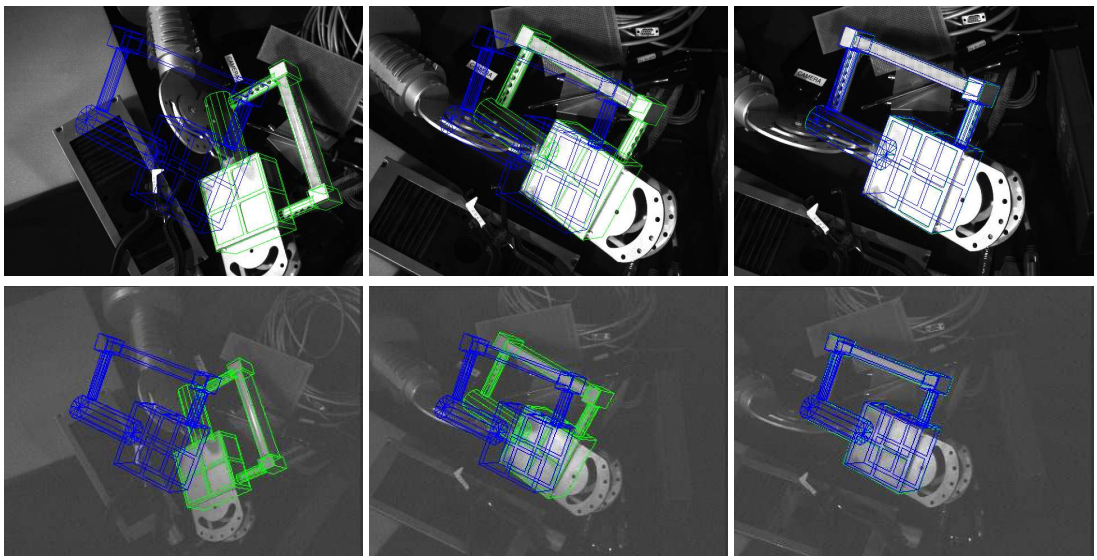


Figure 4: First configuration: monocular system. Snapshots extracted from experimental results (*green*: forward projected CAD model after pose calculation, *blue*: user defined desired position)



Second configuration: small base-line stereoscopic system

Figure 5: Second configuration: small base-line stereoscopic system. Snapshots extracted from experimental results (*green*: forward projected CAD model after pose calculation, *blue*: user defined desired position). Each row shows images acquired by the two cameras.

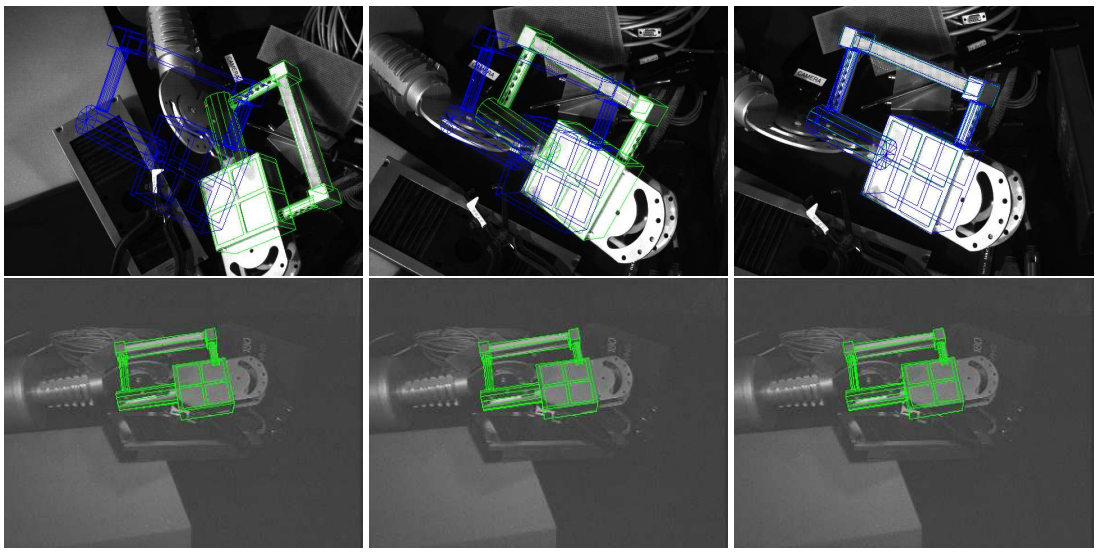
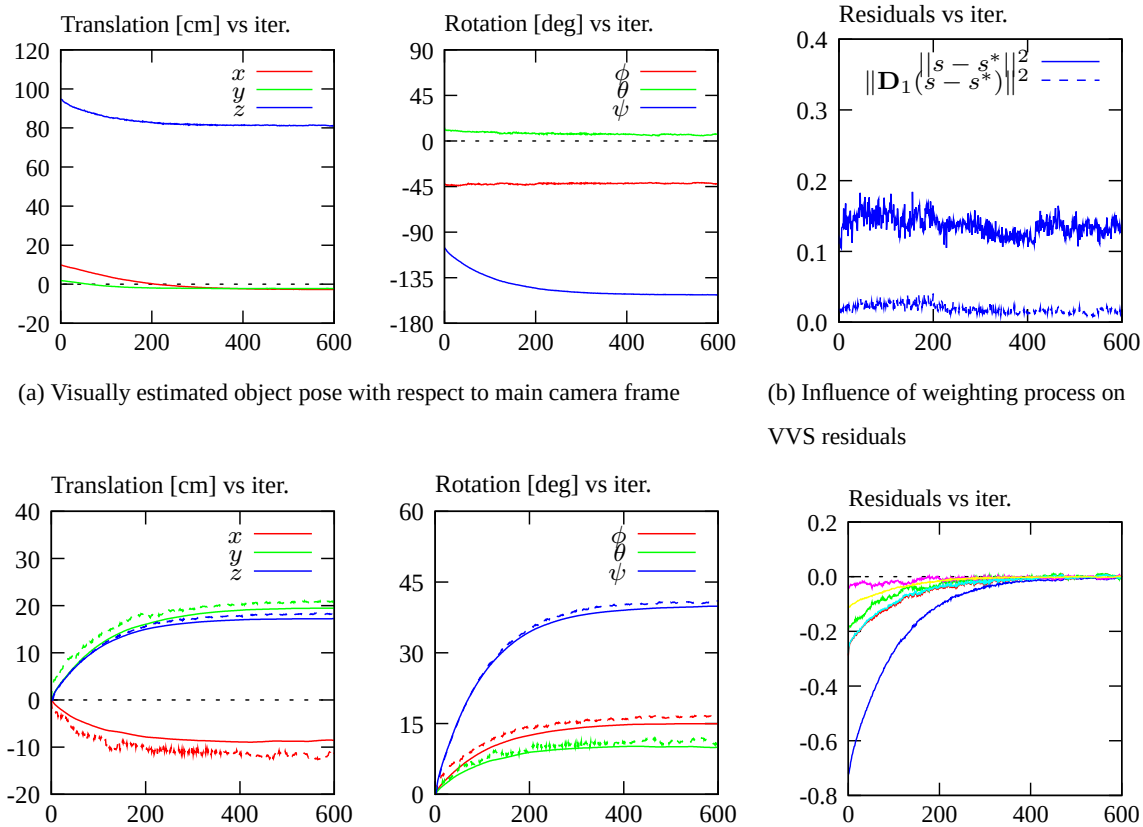


Figure 6: Third configuration: wide base-line stereoscopic system. Snapshots extracted from experimental results (*green*: forward projected CAD model after pose calculation, *blue*: user defined desired position). Each row shows images acquired by the two cameras.



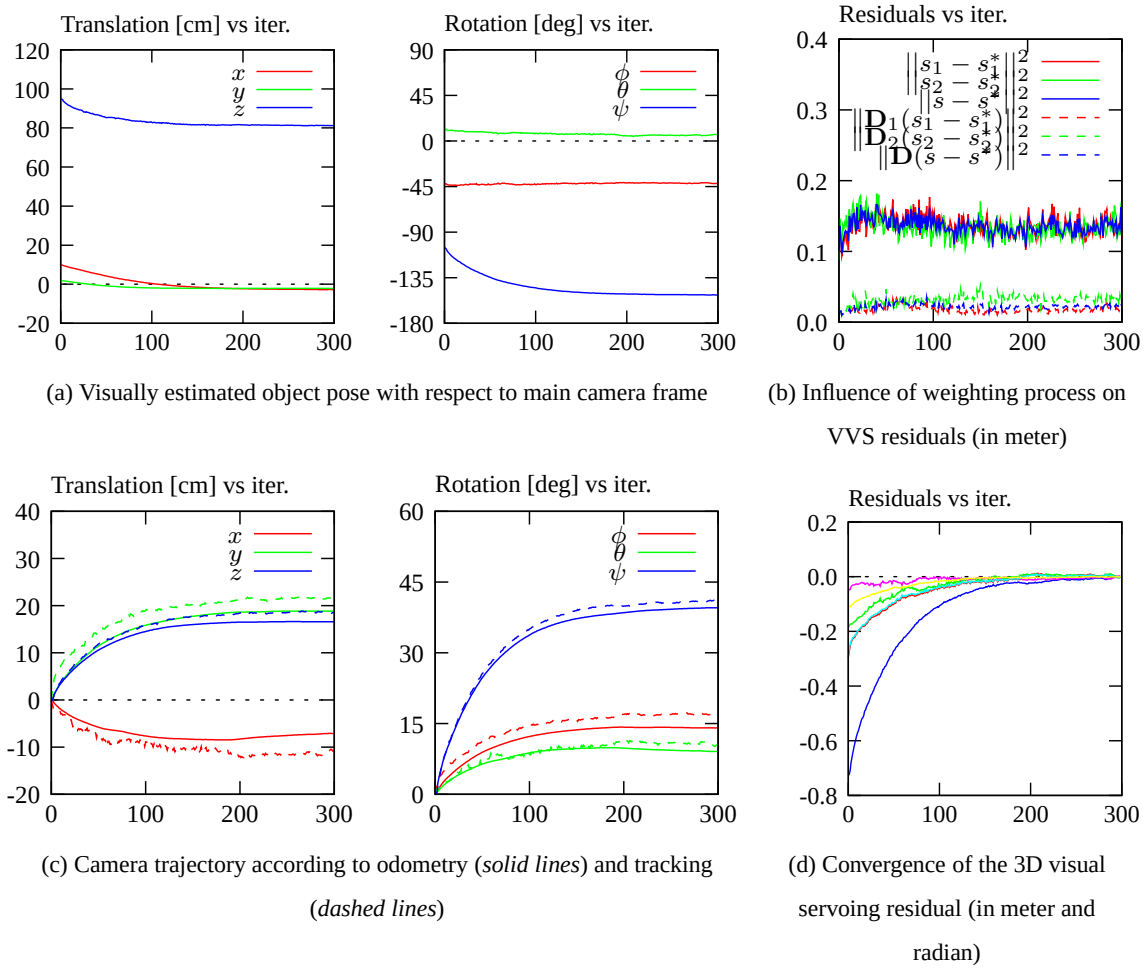
(a) Visually estimated object pose with respect to main camera frame

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



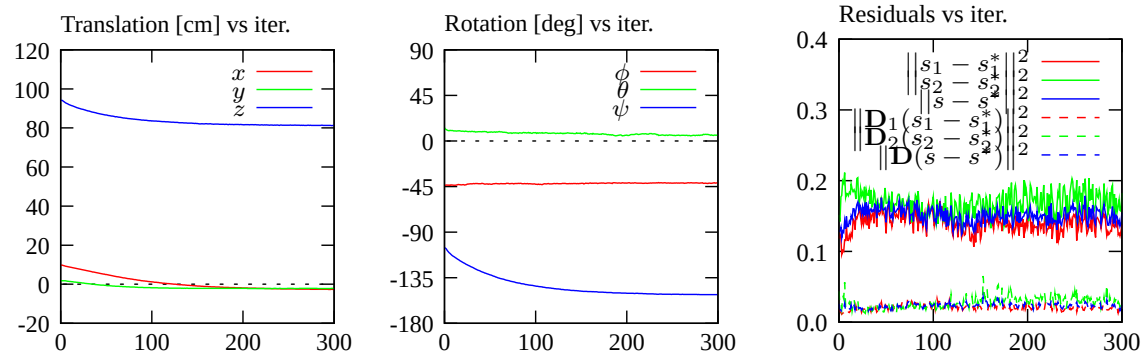
(a) Visually estimated object pose with respect to main camera frame

(b) Influence of weighting process on VVS residuals (in meter)

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

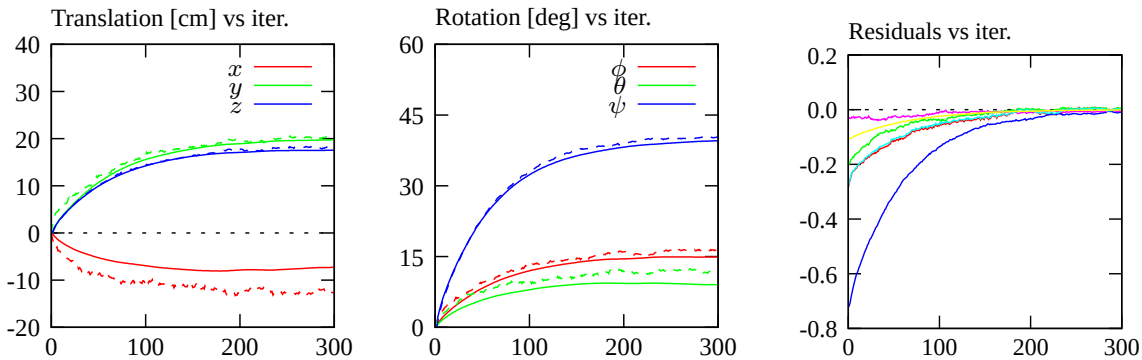
(d) Convergence of the 3D visual servoing residual (in meter and radian)

Figure 8: 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 9: Third experiment results: 3D visual servoing using robust model-based tracking for a wide base-line system setup

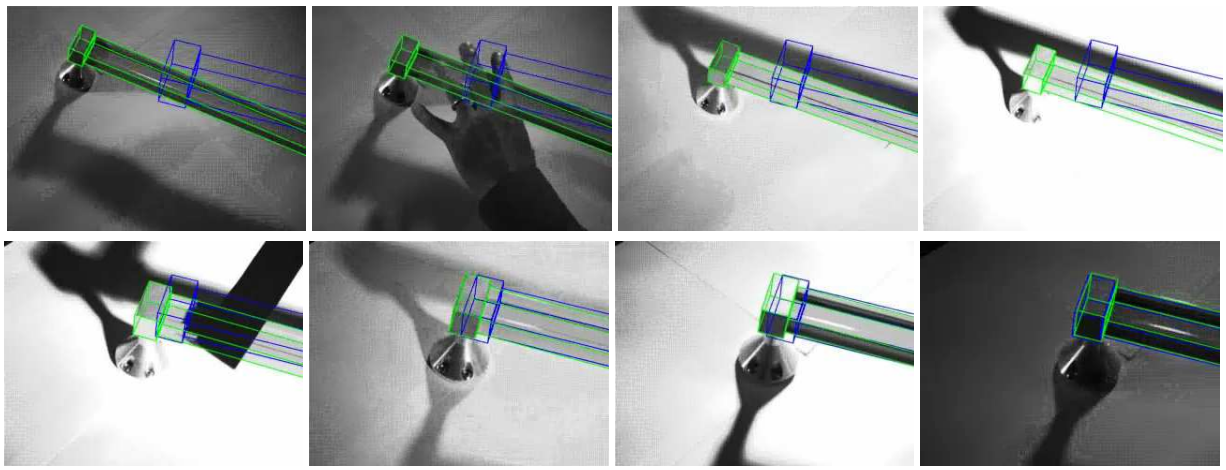


Figure 10: Tracking the handrail. As can be noted the sequence features large occlusions, important lighting variation, modification of the position of the lights,... (*green*: forward projected CAD model after pose calculation, *blue*: user defined desired position)

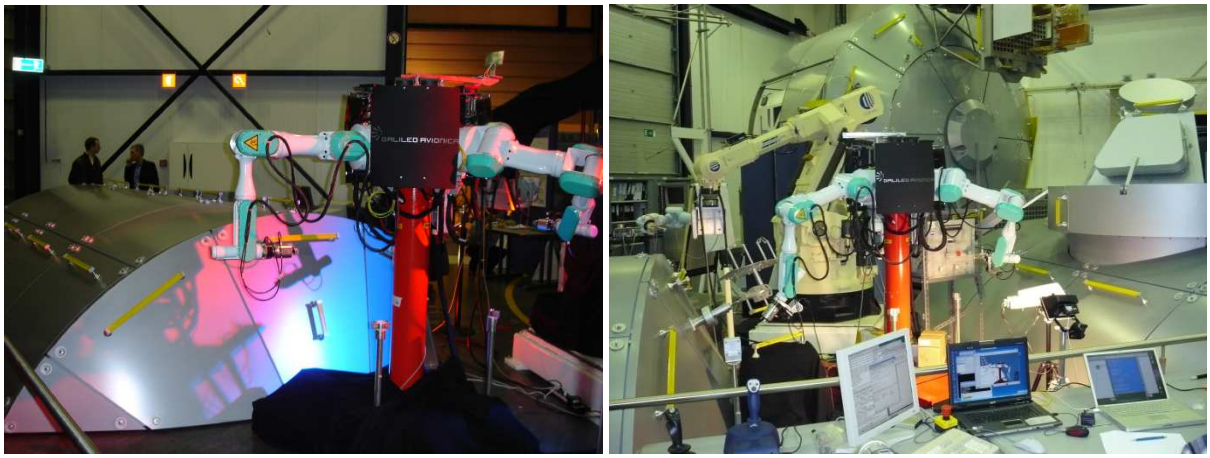


Figure 11: Eurobot prototype at ESA/ESTEC with the ESA Columbus 1:1 mockup.

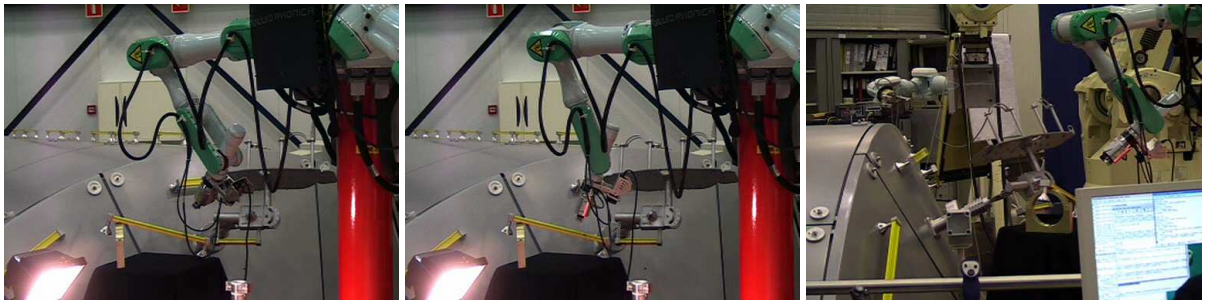


Figure 12: External view of a visual servoing experiment on an handrail at ESA/ESTEC using the Eurobot prototype. The APFR used in the other experiment is also attached to the mockup behind the robot. First image shows the robot at its initial position. In the second image the robot is parallel to the handrail. In the last image the robot is in grasping position.

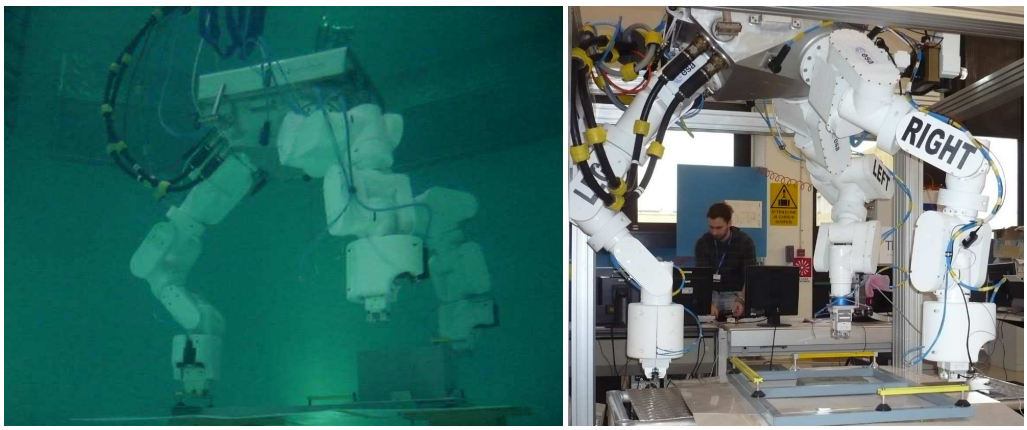


Figure 13: Eurobot wet model built by Thales Alenia Space in Torino (TAS-I) (a) Eurobot in a pool (image courtesy of ESA) (b) Eurobot in the configuration used for the VIMANCO experiments

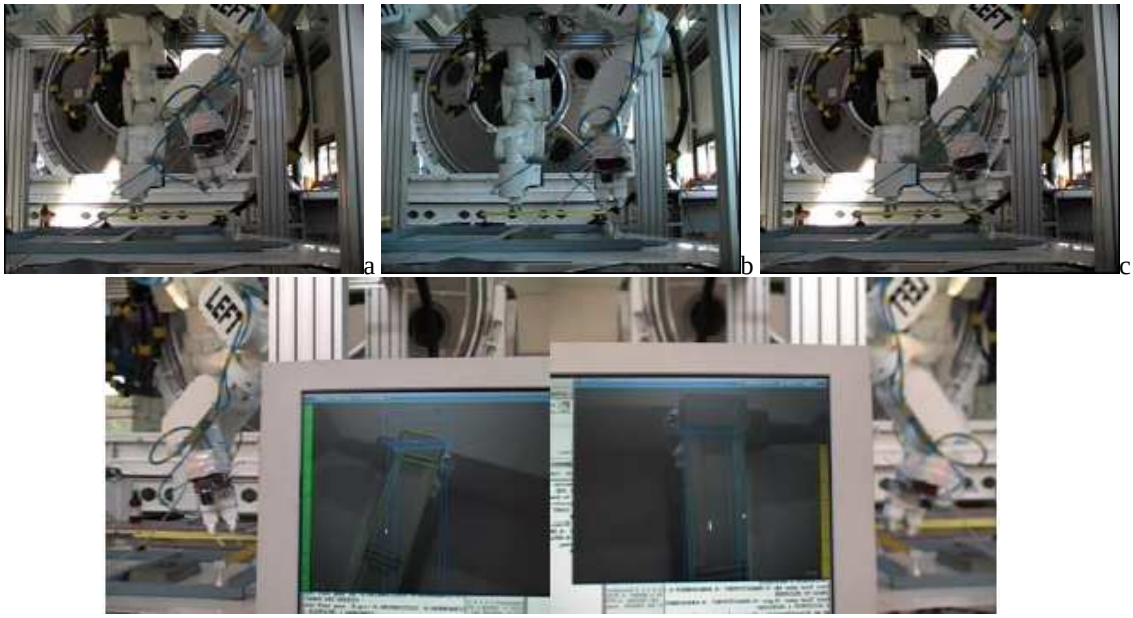


Figure 14: External view of a visual servoing experiment on an handrail at Thales Alenia Space on the Eurobot wet model. On the back is the the ESA Jules Verne module. First image shows the robot at its initial position. In the second image the robot is in grasping position. On the second row is a the view of the camera