A unified approach to visual tracking and servoing

Ezio MALIS
INRIA
Sophia-Antipolis, FRANCE
Ezio.Malis@sophia.inria.fr

Selim BENHIMANE
INRIA
Sophia-Antipolis, FRANCE
Selim.Benhimane@sophia.inria.fr

***Abstract*— In this paper, we present a generic and flexible system for vision-based robot control. The system integrates several research areas (visual matching, visual tracking and visual servoing) in a unifying framework. In this framework, the flexibility is obtained using a template matching algorithm based on an efficient second-order minimization. Contrarily to feature-based visual servoing schemes, we avoid the design of feature-dependent visual tracking algorithms. By integrating the visual tracking process with the visual servoing techniques, we can more easily deal with constrained tasks. This reduces the computation cost and improves the precision of the system. The experimental results prove the efficiency of the unified system in real conditions.**

I. Introduction

Besides the traditional domain of robotic manipulation and grasping, the vision-based control offers a wide spectrum of application possibilities entailing the use of computer vision and control theories : automatic driving, long range exploration, observation and surveillance by aerial robots, medical robotics... The achievement of such complex applications needs the integration of several research areas in vision and control such as visual matching, visual tracking and visual servoing (see for example Petersson *et al.* (2002)). A possible approach to the design of vision-based control schemes is to use, for special purposes, vision and control methods that have been conceived separately. With such approach, system integration can be very difficult due to the high number of different methods for visual tracking and visual servoing (many of them are described in Hashimoto (1993) and in Hutchinson *et al.* (1996)). Instead of considering vision and control systems separately, in this paper we propose to integrate, as far as possible, several research works in computer vision and robotic control in a unifying framework. Our objective is to build a generic, flexible and robust system that can be used for a variety of robotic applications. A class of vision-based control techniques having these requirements has been proposed in Malis & Chaumette (2002). These techniques have been designed to control a robot with respect to rigid objects of any shape and without the explicit knowledge of their CAD model. From the point of view of control theory, the "model-free" (i.e. object model-free) control laws proposed in Malis & Chaumette (2002) can deal with many different applications. In practice, the design of the visual tracking has made the overall system application dependent. For example, a complete system for matching, tracking and servoing has been proposed in Malis *et al.* (2003). That system was specifically designed for positioning a robot with respect to closed contours. If a closed contour is not available in the image, we need to design again several parts of the system in order to consider other image features (e.g. interest points, edges, straight lines...). In addition, feature-based visual servoing methods need explicit feature detection and cannot be applied in the case when the control is performed with respect to an object that does not contain special sets of features. In this paper, we will show that a more flexible system can be obtained by integrating template-based visual tracking algorithms and model-free vision-based control techniques. The key issues for the integration of such tracking techniques in a generic real-time control system are the flexibility, the efficiency, the precision and the robustness of the tracking algorithm. Indeed, template-based visual tracking algorithms estimate the deformation parameters of a certain template between two frames by minimizing an error measure based on image brightness. The explicit segmentation of features is not needed and the visual tracking can be applied to objects of generic shape and texture. Some methods learn the parameters variation in an off-line processing stage: difference decomposition in Gleicher (1997); Jurie & Dhome (2002), active blobs in Sclaroff & Isidoro (1998), active appearance models in Cootes *et al.* (1998). Although these methods work well, they can not be used in some real-time robotic applications when the learning step can not be processed on-line. Alternatively, there are methods that minimize a dissimilarity measure (e.g. the sum-of-squared-differences (SSD)) between the reference template and the current image using parametric models. Many minimization algorithms could be used to estimate the transformation parameters. Although Newton-like algorithms (i.e. based on a SSD second-order Taylor series approximation) have faster local convergence, most of the approaches that have been proposed in the literature are based on first-order approximations (see for example Lucas & Kanade (1981); Hager & Belhumeur (1998); Shum & Szeliski (2000); Baker & Matthews (2001)). The main advantages of a visual tracking approach with a first-order approximation is its efficiency (i.e. it does not need the Hessian computation and need only the Jacobian computation) and its stable behavior when the first-order approximation is not valid any more. On the contrary, the main drawbacks of a second-order approximation visual tracking approach is its high computational cost and its convergence problems when the second-order approximation is not valid any more. Recently, an Efficient Second-order Minimization (ESM) algorithm has been proposed in Malis (2004) in order to improve the performances of visual servoing techniques. Thanks to its generality, the ESM algorithm has been successfully used to build an efficient visual tracking algorithm in Benhimane & Malis (2004). The main contribution of this paper is to integrate into an unifying framework the ESM visual tracking and the model-free visual servoing techniques. In order to achieve real-time applications with non-dedicated hardware several improvements to the ESM visual tracking are proposed in the paper: an efficient coarse-to-fine strategy (a multiresolution strategy) and a sub-sampling strategy.

The paper is organized as follows. In Section II, we give an overview of "model-free" vision-based control systems. In Section III, we give an overview of the ESM visual tracking and we describe some improvements of the algorithm. In Section IV, we propose a new unified vision-based control system. Finally, in Section V and Section VI, we describe two experiments which show the flexibility of the proposed system.

II. Visual Servoing

The starting point of the unified scheme proposed in this paper is the model-free visual servoing proposed in Malis & Chaumette (2002). Model-free means that one does not need an explicit model of the object (e.g. a CAD model) in order to compute the control law. figure 1 shows the corresponding block scheme. The main advantage of model-free visual servoing approaches with respect to standard approaches (e.g. image-based Espiau *et al.* (1992) and model-based Wilson *et al.* (1996) visual servoing) is that they need less "a priori" knowledge on the observed objects. However, as in the other visual servoing approaches, the selection of image features is an important step and it influences the design of the visual tracking method. Indeed, the "Visual Tracking" block must be specifically designed for the selected image features. For example, in Malis *et al.* (2003) the selected features were closed contours. The visual tracking algorithm used in that work is not able to track different features (e.g. interest points, straight lines...). In this scheme, the output of the visual tracking are the current features position. They are used with the reference features position to compute a projective transformation from which it is possible to estimate the Cartesian representation of the state.

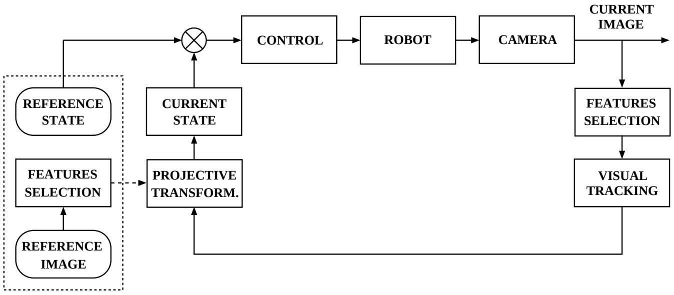


Fig. 1
STANDARD MODEL-FREE VISUAL SERVOING SCHEME WITH FEATURES SELECTION.

A. *Projective Reconstruction*

We suppose that we observe a planar object and that the selected image features are the image coordinates of some characteristic points. The reference features corresponding to the frame $\mathcal{F}^*$ have been selected during an off-line step. The current features acquired at each iteration of the control scheme corresponds to frame $\mathcal{F}$. A 3D point having homogeneous coordinates $\boldsymbol{\mathcal{X}}$ projects to an image point having homogeneous coordinates $\mathbf{p}^*$ in the reference frame $\mathcal{F}^*$. The same 3D point projects to an image point $\mathbf{p}$ in the current frame $\mathcal{F}$:

$$\mathbf{p} \propto \mathbf{K} \left[\begin{array}{cc} \mathbf{R} & \mathbf{t} \end{array} \right] \boldsymbol{\mathcal{X}}$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in^3$ are respectively the rotation matrix and the translation vector between the frames $\mathcal{F}$ and $\mathcal{F}^*$, and the matrix $\mathbf{K}$ contains the camera intrinsic parameters. From the knowledge of several matched points, it is possible to recover the projective transformation (in this case a homography) between the two views without knowing the structure of the object. Indeed, the image points in the current image are related to the reference points by the equation:

$$\mathbf{p} \propto \mathbf{G}\mathbf{p}^*$$

Knowing the camera intrinsic parameters, we can extract the camera displacement from the following equation:

$$\mathbf{R} + \frac{\mathbf{t}}{d^*}\mathbf{n}^{*\top} = \mathbf{K}^{-1}\mathbf{G}\mathbf{K}$$

where $\mathbf{n}^*$ is the unit vector normal to the plane expressed in $\mathcal{F}^*$ and d^* is the distance between the plane and the center of the frame $\mathcal{F}^*$. In general, there are two possible solutions to the homography decomposition Faugeras & Lustman (1988). In order to distinguish the right solution, an estimation of the normal $\mathbf{n}^*$ should be known.

B. *Control schemes*

In this paper, we will suppose that the Jacobian of the robot is perfectly known and full rank. However, even in the presence of small errors in the model of the robot, the vision feedback will compensate such errors. We suppose that the camera is mounted on the end-effector of the robot (an eye-in-hand configuration) and that the control output is the camera velocity. The velocity is obtained by regulating to zero a task function (see Samson *et al.* (1991)). The class of model-free visual servoing methods described in Malis & Chaumette (2002) is based on a reconstruction of the Cartesian representation of the state from which one can design the following control schemes:

- Position-based visual servoing: it is a full Cartesian state feedback. The camera is controlled by performing an explicit reconstruction of the Cartesian state. This control law should be used when the initial displacement is small since there is no control in the image and the observed object could get out of the image.

- Hybrid visual servoing: the state is a mixture of the Cartesian representation of the state (the rotation of the camera) and of measures from the image data. The use of image measures makes easier to keep the object in the field of view of the camera when the initial displacement is very large.

For both methods, an image trajectory planification step can also be added in order to make sure that the object stays in the camera field of view during the servoing (see Mezouar & Chaumette (2002)).

III. Visual Tracking

The model-free visual servoing methods make use of a given set of image features to estimate the displacement of the camera. Thus, the next step is to build a visual tracking algorithm that does not need an explicit feature selection. The tracking can be achieved by directly estimating the projective transformation between a selected reference template and the corresponding area in the current image.

A. *ESM Tracking*

As already mentioned in the introduction, the core of the tracking method is the Efficient Second-order Minimization (ESM) algorithm proposed in Malis (2004). The application of the ESM algorithm to visual tracking allows an efficient real-time homography estimation and template-based tracking with high inter-frame displacements. The figure 2 gives a general overview of the method. A detailed description of the tracking method can be found in Benhimane & Malis (2004). The homography matrix $\mathbf{G}$ is defined up to a scalar factor. Then, without loss of generality, it can always be considered as an element of the SL(3) group (i.e. the group of unimodular (3×3) matrices). Indeed, if $\det(\mathbf{G}) = 0$ then the plane passes through the optical center and all the points on the plane project on a line. Starting from an initial prediction of the homography we iteratively estimate the optimal homography which minimizes the SSD between a reference pattern $\mathbf{T}$ and the current pattern $\mathbf{W}$ reprojected using the current homography $\mathbf{G}$. If an initial prediction of the homography is not available we start with $\mathbf{G}$ equal to the identity matrix. Both the image derivatives of the template $\nabla\mathbf{T}$ and the image derivatives of the current pattern $\nabla\mathbf{W}$ are used to obtain an efficient second-order update. It is an efficient algorithm since only first derivatives are used and the Hessians are not explicitly computed thanks to the use of the average of the image Jacobians.

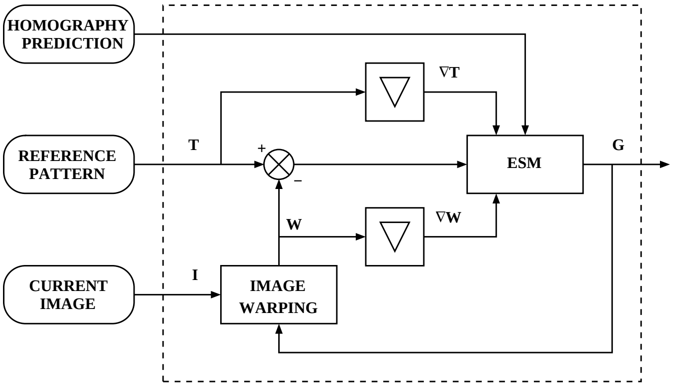


Fig. 2

Visual tracking based on an Efficient Second-order Minimization method.

B. *Multiresolution ESM Tracking*

In order to improve the tracking algorithm, we use a multiresolution method. A change of the resolution of the image can be obtained by an affine transformation and thus by a homography. Let ${}^{i}\mathbf{S}_j$ be the homography matrix which allows to warp the image from resolution j to resolution i. When the reference pattern $\mathbf{T}_0$ is selected in the reference image at full resolution, we warp it n times (using the homographies ${}^1\mathbf{S}_0$, ${}^2\mathbf{S}_0$, ..., ${}^n\mathbf{S}_0$) until we reach a minimum resolution (e.g. the size of the pattern at resolution n should be greater than 20×20 pixels). We obtain once and for all $n + 1$ reference patterns $\mathbf{T}_0$, $\mathbf{T}_1$,..., $\mathbf{T}_n$. If the homography transforming an area of the current image in the reference template at scale j is ${}^j\mathbf{G}$, then the homography transforming that area of the current image in the reference template at scale i is:

$${}^i\mathbf{G} = {}^i\mathbf{S}_j \, {}^j\mathbf{G}$$

The Multiresolution ESM tracking is illustrated in figure 3. The tracking is started at the scale n using an initial estimation homography ${}^n\mathbf{G}$ (if a prediction is not available, we set ${}^n\mathbf{G}$ equal to the matrix ${}^n\mathbf{S}_0$). Once the ESM algorithm has computed the homography ${}^n\mathbf{G}$, we simply obtain the homography ${}^{n-1}\mathbf{G}$ by changing the scale ${}^{n-1}\mathbf{G} = {}^{n-1}\mathbf{S}_n \, {}^n\mathbf{G}$. The ESM algorithm is repeated $n + 1$ times and at the scale 0, we obtain the homography ${}^0\mathbf{G}$. The loop is repeated by rescaling the homography at the higher scale ${}^n\mathbf{G} = {}^n\mathbf{S}_0 \, {}^0\mathbf{G}$. The algorithm can also be stopped at scale $k > 0$ of the pyramid (e.g. if the computation time is limited) by rescaling the homography ${}^0\mathbf{G} = {}^0\mathbf{S}_k \, {}^k\mathbf{G}$.

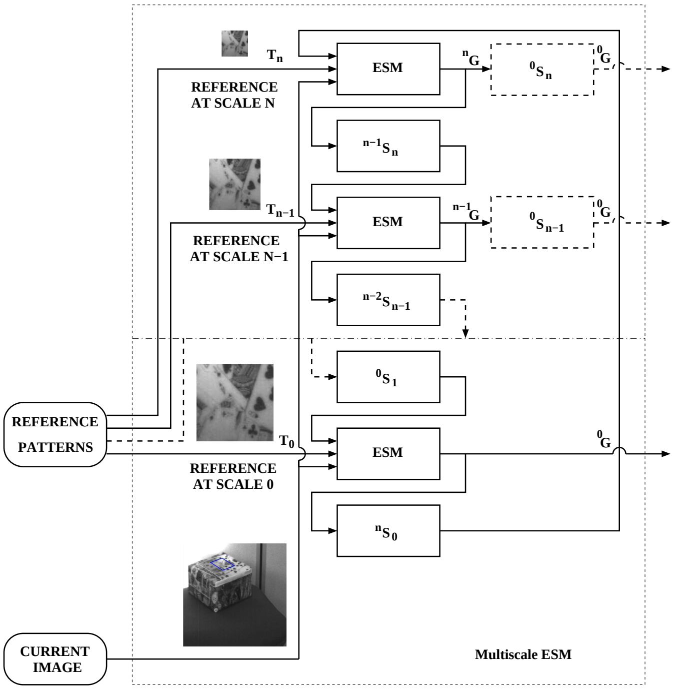


Fig. 3

MULTIRESOLUTION ESM TRACKING

C. Template sampling

In order to speed up the visual tracking, we select a subset of pixels belonging to the reference template. For example, we can stop the multiresolution ESM tracking at scale $k > 0$ and obtain the homography by rescaling ${}^0\mathbf{G} = {}^0\mathbf{S}_k \, {}^k\mathbf{G}$. By reducing the size of the reference pattern, we reduce the computational cost of each iteration. A further improvement of the computation speed can be obtained by selecting a subset of pixels in the reference template. Indeed, pixels with a strong gradient carry more useful information than pixels with a weak gradient. The simplest method to select the pixels is to use the image derivatives of the reference pattern. Since we do not want to choose any particular image features (e.g interest points or edges), we do not use Harris & Stephens (1988) or Canny (1986) filters. We simply threshold the norm of the derivatives $\nabla T_u^2 + \nabla T_v^2$. Figure 4 shows an example of the pixel selection. The reference pattern is (300×300) pixels. Selecting only pixels where the gradient is meaningful allows to speed up the estimation of the homography. The white points in figure 4 are the pixels selected by thresholding and they represent only 10% of the original pattern. Another method for template sampling is to use both the derivative of the reference pattern and the current image. This is more adapted to our minimization method. Indeed, in our minimization scheme, the meaningful information is the average of the image Jacobians. If the average is zero, the image pixels difference does not carry useful information and can be discarded. Thus, the sampled pixels are obtained by thresholding the norm of the average derivatives $(\nabla T_u + \nabla W_u)^2 + (\nabla T_v + \nabla W_v)^2$. In this case, the selected pixels change at each iteration.

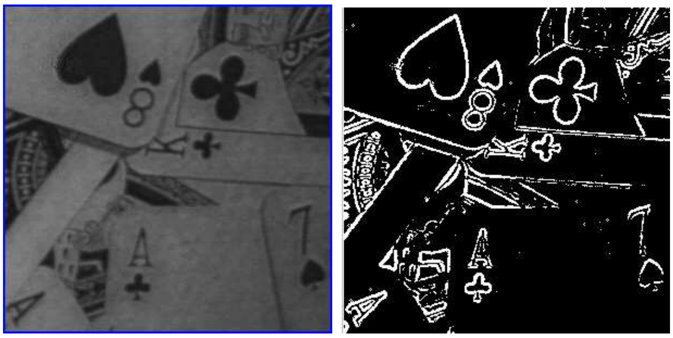


Fig. 4

TEMPLATE SAMPLING BASED ON IMAGE GRADIENTS.

IV. Unifying visual tracking and servoing

The ESM tracking algorithm has been the fundamental step for unifying visual tracking and servoing. Figure 5 shows the block scheme corresponding to the model-free visual servoing with the template-based tracking. When compared to figure 1, we can notice the absence of the blocks corresponding to features selection. The scheme is considerably simplified since we do not have to select any kind of feature to track. As a consequence, the projective transformation (i.e. the homography in the case of planar objects) is not computed with the current and the reference image features but it is directly estimated in the tracking algorithm. The new tracking method is generic and we only need to find the homography $\mathbf{G}$ that links an area of the current image with the reference pattern.

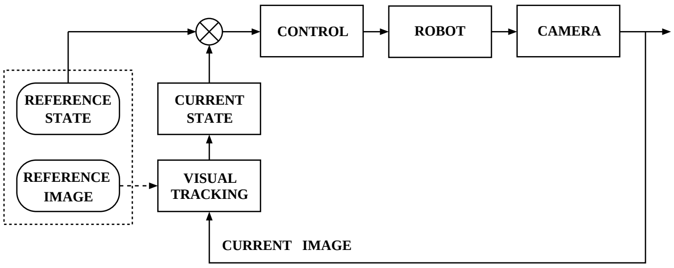


Fig. 5

Template-based visual servoing with direct estimation of the projective transformation.

Although the scheme described in the figure 5 is more flexible than the scheme in the figure 1, many issues remain unsolved.

- We need to extract the Cartesian representation of the state from the projective transformation (the homography) as detailed in section II-A. As we already mentioned, the homography decomposition is generally a critical step since there exist two possible solutions. Generally, having an approximation of $\mathbf{n}^*$ helps to select the right solution. Unfortunately, due to the unavoidable image noise, measure errors and estimation uncertainties, it is very hard to shun switching between the two solutions especially when the $\mathbf{n}^*$ approximation is unreliable. This may cause an undesirable discontinuity in the control law and may reduce the system performance.

- The projective transformation depends on 8 parameters: 3 parameters for the rotation, 3 parameters for the translation (up to the scale factor d^*) and 2 parameters for the normal $\mathbf{n}^*$. Note that the normal in the reference frame is constant and it is not a part of the state of the camera (which is fully defined by the other 6 parameters). Since an approximation of the normal is needed anyway for the homography decomposition, the 2 extra parameters should be fixed directly in the ESM tracking reducing the computation time (making the Jacobian matrices have smaller size).

- Tracking in the projective space makes it difficult to impose some constraints on the robot motion (which are generally defined in the Cartesian space). For example, if the robot is a car moving on a planar surface, the state is defined by only 3 parameters. Imposing this constraint can considerably reduce the computations and improve the results of the tracking algorithm. Similarly, adding in the projective space “a priori” knowledge on the observed objects can be possible only in some special cases. For example, when tracking an object composed of N planes rigidly attached to each other, we need to compute N homographies (each plane must be tracked separately). Thus, 8N parameters are estimated whereas only 6 parameters vary.

For these reasons, we propose to directly compute the camera Cartesian position given the reference pattern and the current image. In fact, the efficient second-order approximation used in the ESM tracking can be obtained considering any Lie Group. In the previous scheme, the output of the ESM tracking was the homography which belongs to $SL(3)$ group. In the new tracking scheme, the output is directly the Cartesian representation of the state (the rotation and the translation of the camera) which belongs to the $SE(3) = SO(3)\times^3$ group. As a consequence, few modifications have been done to deal with $SE(3)$ and the output of the ESM tracking has become the Cartesian representation of the state (see figure 6).

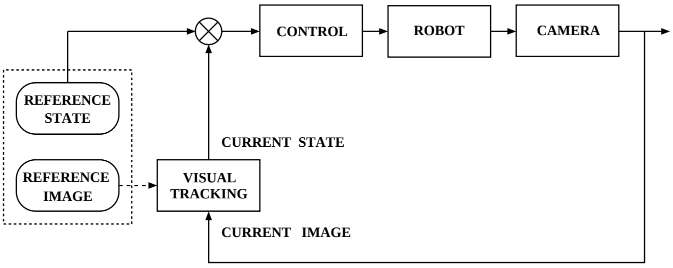


Fig. 6

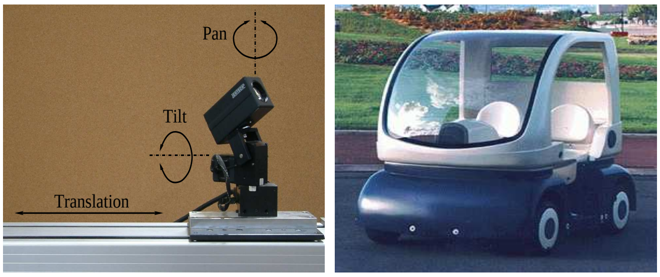
Template-based visual servoing with direct estimation of the Cartesian representation of the state.

Exactly as in the previous scheme, the complete system (tracking and control) needs an approximation of the camera intrinsic parameters $\mathbf{K}$, of the normal $\mathbf{n}^*$ and of the distance d^*. However, in the unified approach proposed in this paper, they are directly used in the tracking. We are currently studying how to adapt the self-calibration method described in Malis & Cipolla (2002) in order to estimate in a joint loop the camera intrinsic parameters $\mathbf{K}$ and the normal vector $\mathbf{n}^*$.

V. Experimental setup

A. The robotic systems

In the experiments, we use two different robotic systems. The first (called “Arges”) is a 3 d.o.f. robot (see figure 7(a)). It is a pan-tilt turret mounted on a linear rail and equipped with a camera. A single computer with a bi-processor Pentium III 730 MHz is used for the image capture, the visual tracking process and the robot control. The second robotic system (called “Cycab”) is an electric car (see figure 7(b)) equipped with a camera mounted on a pan-tilt turret situated behind its front windshield. The “Cycab” is equipped with two computers (Pentium III 700 MHz). One is used for the visual tracking and the control computation, while the other is devoted for the low-level control of the “Cycab” velocity and the wheel steering.



(a) Arges

(b) Cycab

Fig. 7

The robotic system used in our experiments.

B. ESM Tracking and Control Software

We have developed a software called “ESM Tracking and Control Software”. The software provides an interface for template tracking and makes it possible to easily choose the output of the tracking algorithm depending on the control scheme and the robotic system. The software displays the current image seen by the camera and gives statistical and visual information during the tracking task in order to supervise the different steps of the experiments. Screenshots of the current version of the software can be seen in figure 8.

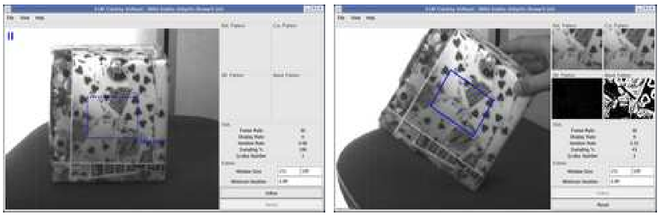


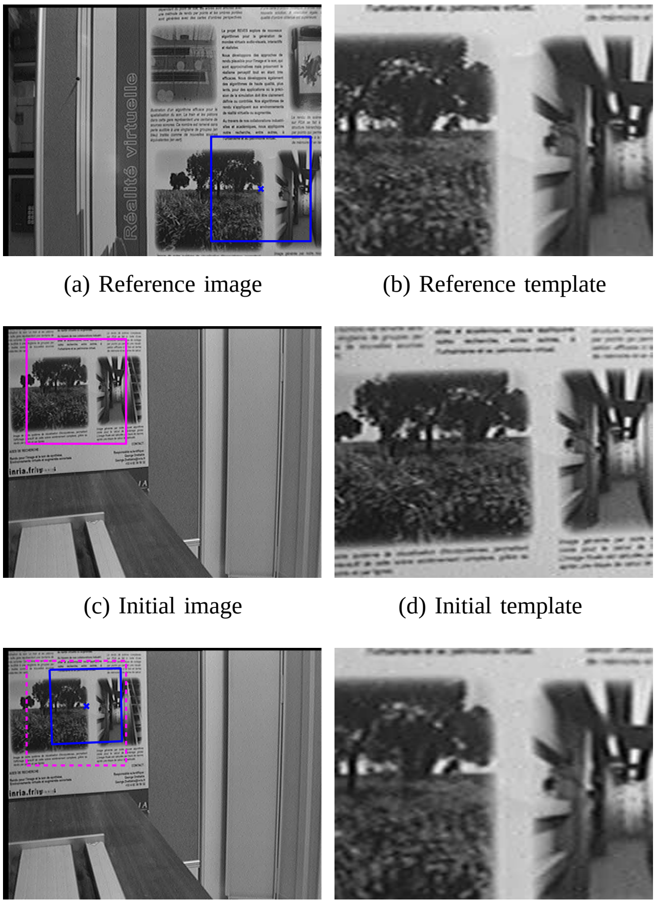
Fig. 8

ESM Tracking and Control Software.

VI. Experimental results

A. 2 1/2 D visual servoing

We use the ESM visual tracking algorithm to build a semi-automatic method for matching the reference pattern and the initial image. The method is semi-automatic since we only need a rough prediction of the homography matrix. In the experiments, the prediction of the homography is given by the user that approximatively selects the center of the template in the initial image. The figure 9 shows an example of the semi-automatic matching. A reference image is stored when the robot is in its goal position (see the figure 9(a)) and a 200x200 pattern of the planar object is chosen (see the figure 9(b)). Then, the robot is displaced to another position from which the pattern is still in the field of view of the camera (see the figure 9(c)). In order to guide the matching process, an approximative position of the pattern is given (see the magenta square in the figure 9(c) and the corresponding pattern in 9(d)). The patterns 9(b) and 9(d) are different proving that only a rough approximation has been given.



(a) Reference image

(b) Reference template

(c) Initial image

(d) Initial template

(e) Matching convergence

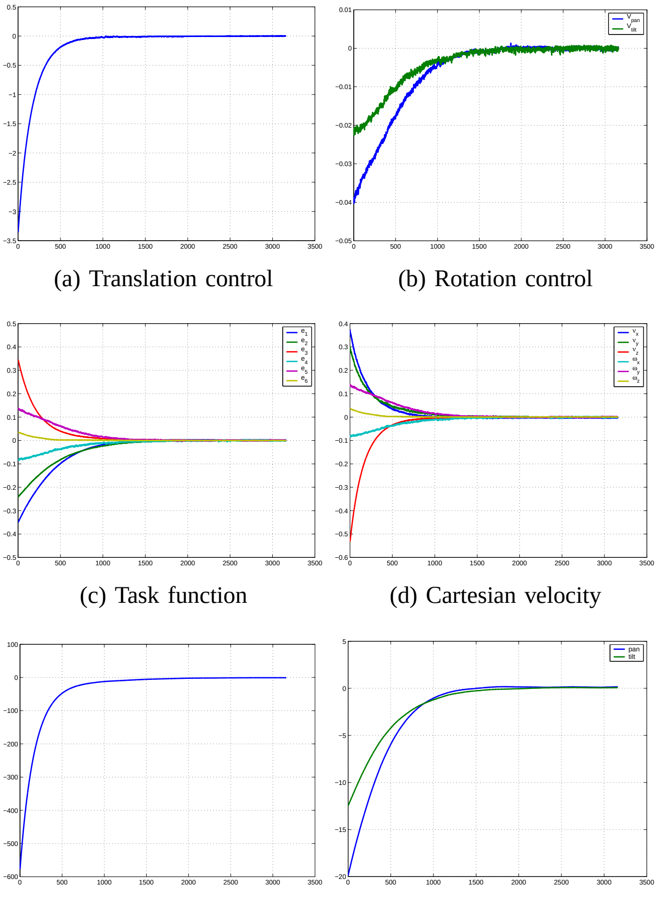
(f) Final template

Fig. 9

Semi-automatic matching.

Starting from the approximative position, it is possible to obtain the homography between the reference and the current frame by running off-line the visual tracking method described in paragraph III (see the blue square in the figure 9(e)). The obtained pattern in figure 9(f) is very close to the reference pattern 9(b) proving that the visual matching is accurately performed.

1) 2 1/2 D visual servoing without occlusion: Once the reference image is learned (see the figure 9(a)), the robot “Arges” is translated of 58 cm along the linear rail, and rotated of 20 degrees in the pan and of 12.5 degrees in the tilt. Then, the robot is controlled in order to reach the reference position. During this experiment, the pattern remains entirely visible in the current image i.e. the pattern is not occluded by an object during the experiment. The experiment shows that the control law is stable (see figures 10(a) 10(b)). The task function and the Cartesian velocity converge exponentially to zero (see figures 10(c) 10(d)). At the end of the servoing process, the robot is back to its reference position (see figures 10(e) 10(f)). The translation error is less than 1 mm and the rotation error is less than 0.2 degrees.



(a) Translation control

(b) Rotation control

(c) Task function

(d) Cartesian velocity

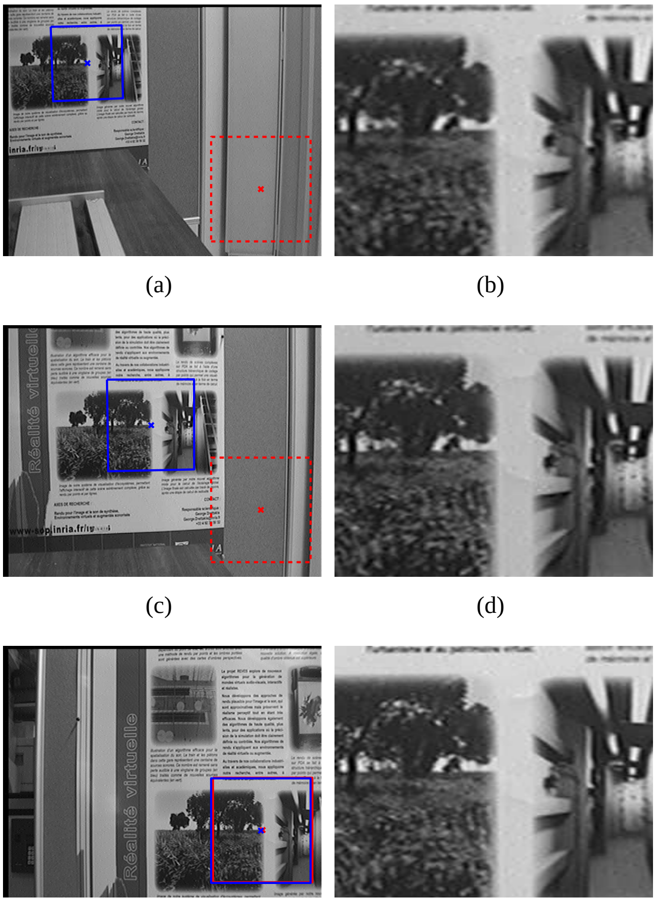
(e) Translation error

(f) Rotation error

Fig. 10

THE ROBOT POSITIONING WITHOUT OCCLUSIONS.

Despite the depth change and the image noise, the tracking algorithm provides a very good approximation of the homographies along the servoing process. Three images are extracted from the servoing sequence in the chronological order and they are shown in the left columns of figure 11. The red dashed square corresponds the reference position of the pattern and the blue squares are its current position. In the right columns of the figure 11, the current pattern warped back using the homography estimation is shown. For the different positions, the current pattern is very close to the reference one (see the figure 9(b)) proving that the tracking is accurately performed.



(a)

(b)

(c)

(d)

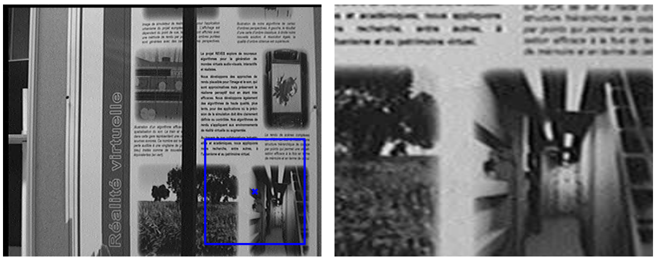
(e)

(f)

Fig. 11

TRACKING A PATTERN ON A PLANAR OBJECT.

2) 2 1/2 D visual servoing with occlusion: Contrarily to the first experiment, the tracked pattern is occluded by a vertical black ruler during the servoing. The ruler is placed at half distance between the robot and the planar object (the poster). It occludes from 15 to 20 pixels of the pattern width depending on the robot position. When the reference image is learned (see figure 12(a)), the pattern is not occluded (see figure 12(b)).

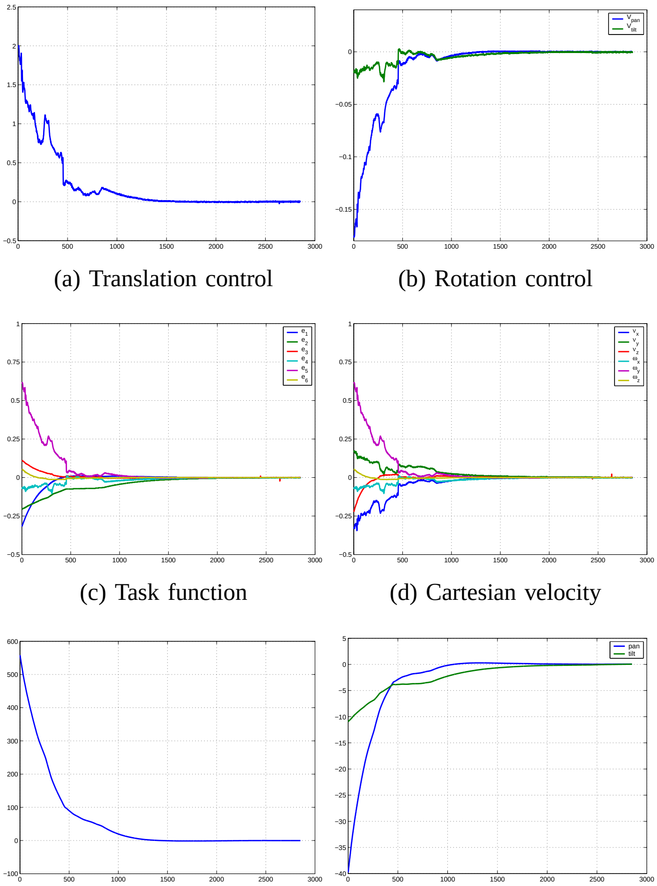


(a) Reference image (b) Reference pattern

Fig. 12

REFERENCE IMAGE AND REFERENCE PATTERN.

The robot is translated of 56 cm, and rotated of 40 degrees in the pan and 11 degrees in the tilt. The semi-automatic matching is performed again. Despite the partial occlusion, the experiment shows that the control law is still stable (see figures 13(a) 13(b)), the task function and the Cartesian velocity converge to zero (see figures 13(c) 13(d)). At the end of the servoing process, the robot is back to its reference position (see figures 13(e) 13(f)).



(a) Translation control (b) Rotation control

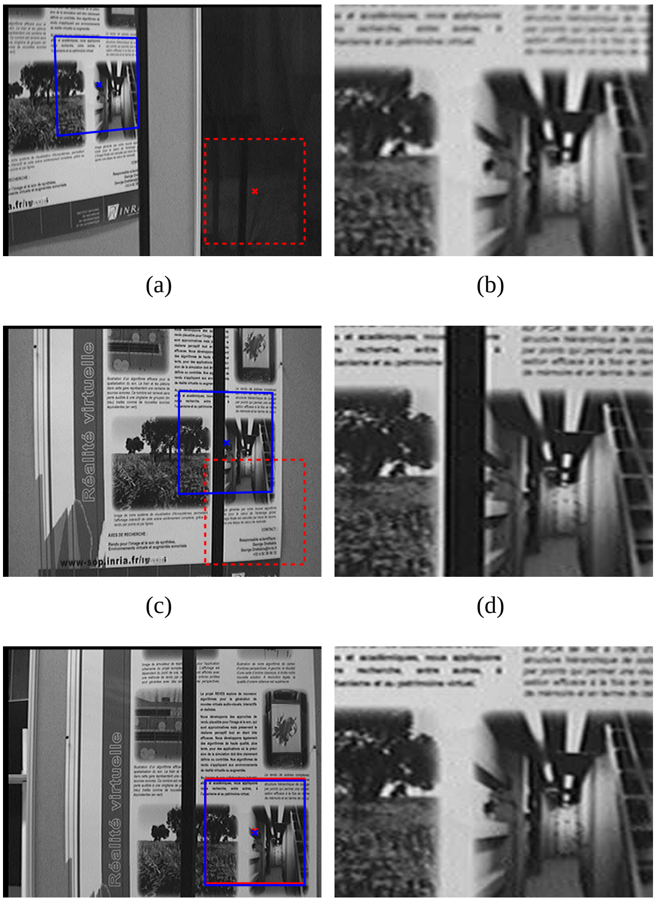
(c) Task function (d) Cartesian velocity

(e) Translation error (f) Rotation error

Fig. 13

THE ROBOT POSITIONING WITH OCCLUSIONS.

The translation error is less than 0.3 mm and the rotation error is less than 0.05 degrees. However, in the different curves, we can see some oscillations. These oscillations are due to the occlusion. In fact, the homographies estimated are not very accurate when the pattern is partially occluded. The amplitude of the oscillations depends on the part occluded of the pattern. In our case, the ruler that occludes the pattern is dark. If it occludes a white part of the pattern, the homography is not estimated very well. For example, the discontinuity observed in the different curves at the iteration 500 is due to the occlusion of the vertical blank between the 2 images of the pattern (see the figure 14(d)). The tracking process remains reliable despite the partial occlusion of the tracked pattern. In fact, along the servoing process, it provides a sufficient approximation of the homographies. In the images extracted from the servoing sequence, the current pattern (in the right column of the figure 14) is very close to the reference one (see the figure 12(b)).



(a) (b)

(c) (d)

(e) (f)

Fig. 14

TRACKING A PLANAR PATTERN WITH PRESENCE OF OCCLUSION.

B. 3D vision-based control of non-holonomic robots

In this experiment, a vision-based car-platooning is performed in a real outdoor environment. Two electric vehicles of type “Cycab” (see the figure 7(b)) are used one as a guider car and the other as a follower car. A driver guides the first car while the follower car is controlled by a 3D vision-based control scheme. The control scheme takes into account that the vehicle is non-holonomic and tries to keep the distance between the two vehicles constant and equal to the initial distance. The relative position is given by a vision-based system. The pan-tilt turret is controlled in order to keep the guider car in the field of view of the camera during the experiment.

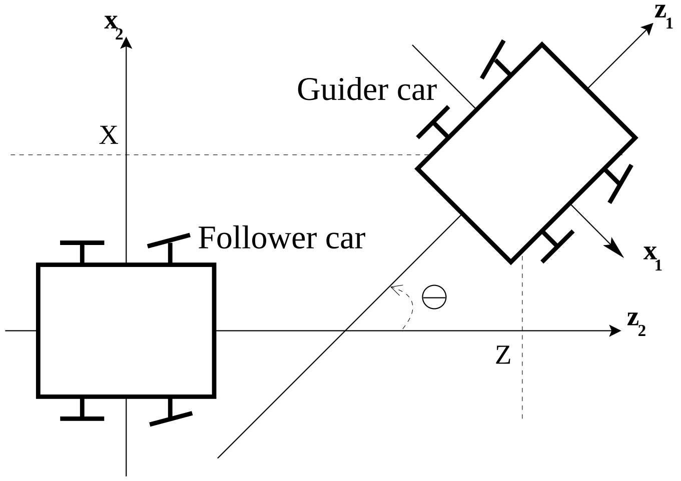
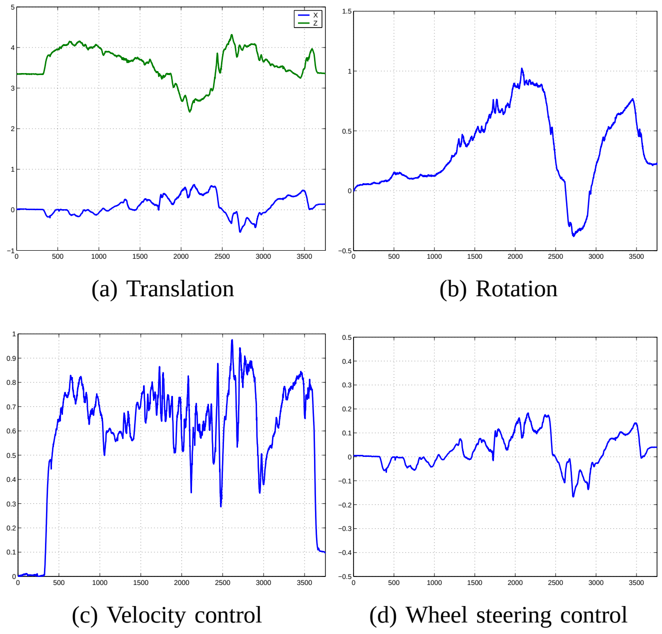


Fig. 15

DETAILS OF THE RELATIVE POSITION BETWEEN THE GUIDER CAR AND THE FOLLOWER CAR

In the starting situation, when the guider car is in front of the follower car, a window of (100×100) pixels is selected to be the reference pattern. In order to have a metric reconstruction, the camera has been roughly calibrated and the distance between the two cars is given to the control process. It is the distance between the camera of the follower car and a poster sticked on the back windshield of the guider car. Tracking this reference pattern provides the relative position between the two vehicles. The two vehicles are supposed to move on a planar surface. Consequently, the relative position provided by the tracking algorithm is composed of two translations along X and along Z and one rotation θ as shown in the figure 15. In this experiment, the guider car is driven in a 100 meter long closed loop. The initial relative X translation is $X_0 = 0$ and the initial relative rotation is $\theta_0 = 0$. The distance between the two vehicles is $Z_0 = 3.35m$. So the follower car is controlled to keep the distance constant (i.e. equal to Z_0). The relative translation and rotation are shown in the figures 16(a) and 16(b). They all variate around the reference values except when the leader vehicle turns. In this case, the distance cannot be kept constant due to the non-holonomy of the robotic system. The follower car velocity control and wheel steering control are shown in the figures 16(c) and 16(d). The description of the control law is out of the scope of this paper. Note however that the control law is stable proving that the estimation of the position of the car is good enough to achieve the task. In the figure 17, images of the experiment sequences are given. In the left column, the relative position between the guider car and the follower car can be seen. In the central column, the respective images grabbed by the follower car and used as input for the ESM visual tracking algorithm are shown. The blue square indicates the tracked region. In the right column, the reprojections of the tracked region using the estimated homographies are shown. The first row of the figure 17 corresponds to the initial position. The tracking algorithm performs well although the experiment takes place outdoor and sun reflection on the tracked region occurs. The current pattern reprojections are very close to the reference one.



(a) Translation

(b) Rotation

(c) Velocity control

(d) Wheel steering control

Fig. 16

RELATIVE POSITION PROVIDED BY THE ESM TRACKING ALGORITHM AND THE CORRESPONDENT CONTROL.

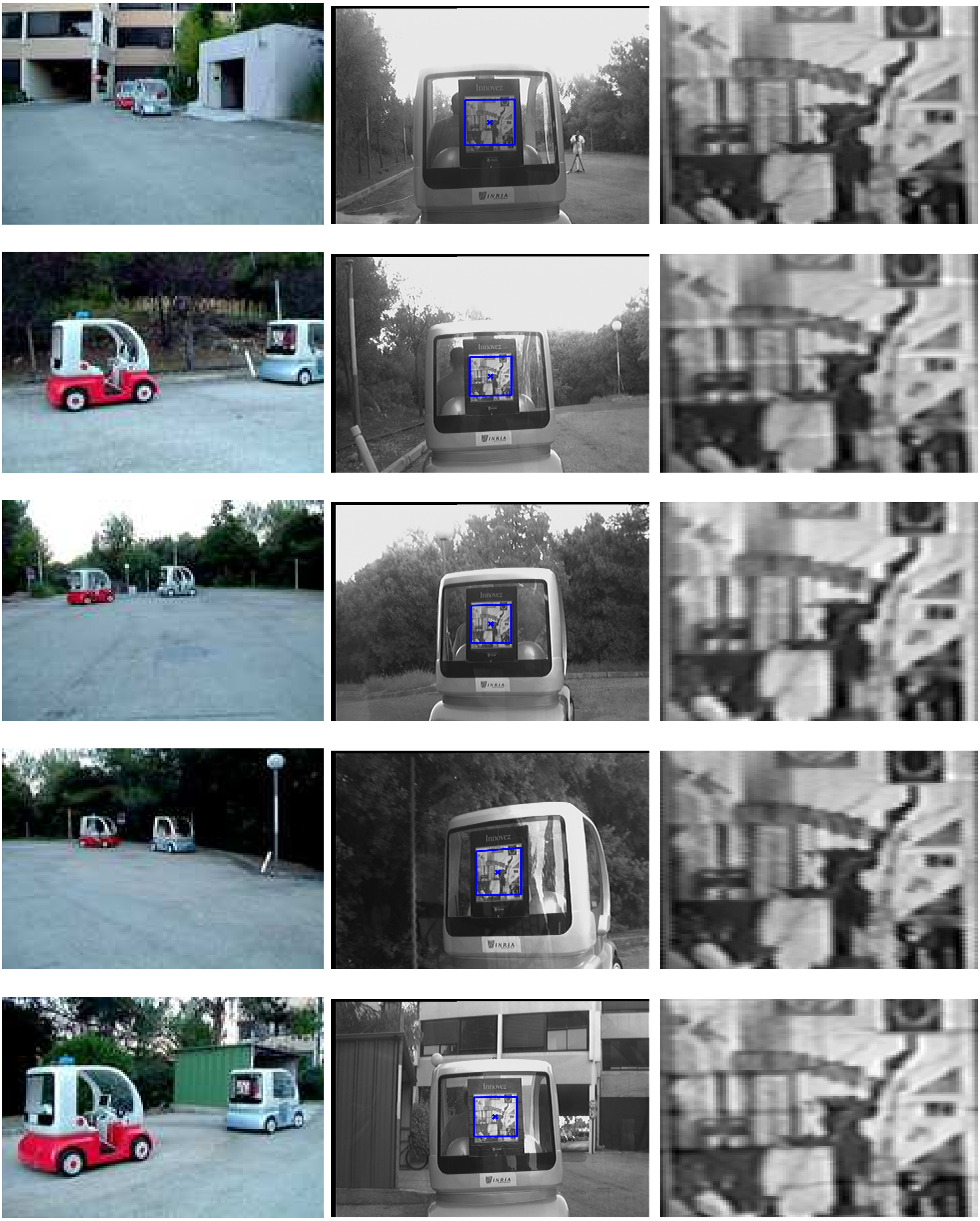


Fig. 17

Images of a car platooning application. The images in the left column are taken with an external camera. The images in the central column are taken with the camera on-board and show the current pattern. The images in the right column show the reprojection of the current pattern in the reference frame.

VII. Conclusion and future work

In this paper, we have presented a complete system for visual tracking and servoing with respect to planar objects having any shape and texture. The visual tracking and servoing are integrated in an unified framework. This allows to handle more easily constraints on robots and/or the observed objects. The main advantage of the system is its flexibility. Indeed, it can be used in many applications without modifying the low level modules. Further improvements of the ESM tracking will concern a robust estimation of the state (e.g. with M-estimators) for handling occlusions and illumination changes. Finally, we plan to extend the system for tracking 3D objects (e.g. supposing that they are piecewise planar).

Acknowledgments

We would like to thank Baptiste Pettiti for his help in programming the experiments and Jose Raul Azinhera and Patrick Rives for providing the non-holonomic control law of the “Cycab”.

References

Baker, S., & Matthews, I. 2001. Equivalence and Efficiency of Image Alignment Algorithms. *Pages 1090–1097 of: IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1.

Benhimane, S., & Malis, E. 2004. Real-time image-based tracking of planes using efficient second-order minimization. *In: IEEE/RSJ Int. Conf. on Intelligent Robots Systems*.

Canny, J. F. 1986. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **8**(6), 679–698.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. 1998. Active appearence models. *Pages 484–498 of: European Conf. on Computer Vision*, vol. 2.

Espiau, B., Chaumette, F., & Rives, P. 1992. A New Approach to Visual Servoing in Robotics. *IEEE Trans. on Rob. and Aut.*, **8**(3), 313–326.

Faugeras, O., & Lustman, F. 1988. Motion and Structure From Motion in a Piecewise Planar Environment. *Int. Journal of Pattern Recognition and Artificial Intelligence*, **2**(3), 485–508.

Gleicher, M. 1997. Projective registration with difference decomposition. *Pages 331–337 of: IEEE Int. Conf. on Computer Vision and Pattern Recognition*.

Hager, G. D., & Belhumeur, P. N. 1998. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **20**(10), 1025–1039.

Harris, C., & Stephens, M. 1988. A combined corner and Edge Detector. *Pages 147–151 of: Proceedings of the 4th Alvey Vision Conf.*.

Hashimoto, K. 1993. *Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback*. World Scientific Series in Robotics and Automated Systems, vol. 7. World Scientific Press.

Hutchinson, S., Hager, G. D., & Corke, P. I. 1996. A tutorial on Visual Servo Control. *IEEE Trans. on Rob. and Aut.*, **12**(5), 651–670.

Jurie, F., & Dhome, M. 2002. Hyperplane Approximation for Template Matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **24**(7), 996–1000.

Lucas, B., & Kanade, T. 1981. An iterative image registration technique with application to stereo vision. *Pages 674–679 of: Int. Joint Conf. on Artificial Intelligence*.

Malis, E. 2004. Improving vision-based control using efficient second-order minimization techniques. *Pages 1843–1848 of: IEEE Int. Conf. on Rob. and Aut.*.

Malis, E., & Chaumette, F. 2002. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Transaction on Rob. and Aut.*, **18**(2), 176–186.

Malis, E., & Cipolla, R. 2002. Camera self-calibration from unknown planar structures enforcing the multiview constraints between collineations. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **24**(9), 1268–1272.

Malis, E., Chesi, G., & Cipolla, R. 2003. 2 1/2 D visual servoing with respect to planar contours having complex and unknown shapes. *Int. Journal of Rob. Research*, **22**(10-11), 841–853.

Mezouar, Y., & Chaumette, F. 2002. Path planning for robust image-based control. *IEEE Trans. on Rob. and Aut.*, **18**(4), 534–549.

Petersson, L., Jensfelt, P., Tell, D., Strandberg, M, Kragic, D., & Christensen, H. I. 2002. System integration for real-world manipulation tasks. *Pages 2500–2505 of: IEEE Int. Conf. on Rob. and Aut.*.

Samson, C., Le Borgne, M., & Espiau, B. 1991. *Robot Control: the Task Function Approach*. Oxford Engineering Science Series, vol. 22. Clarendon Press.

Sclaroff, S., & Isidoro, J. 1998. Active blobs. *Pages 1146–1153 of: IEEE Int. Conf. on Computer Vision*.

Shum, H. Y., & Szeliski, R. 2000. Construction of panoramic image mosaics with global and local alignment. *Int. Journal of Computer Vision*, **16**(1), 63–84.

Wilson, W. J., Hulls, C. C. W., & Bell, G. S. 1996. Relative End-Effector Control Using Cartesian Position-Based Visual Servoing. *IEEE Trans. on Rob. and Aut.*, **12**(5), 684–696.