Dynamic Sensor Planning in Visual Servoing

Éric Marchand[1] and Greg D. Hager
Dept. of Computer Science, Yale University
New Haven, CT 06520–8285
Email: marchand@irisa.fr, hager@cs.yale.edu

abstract

*We present an approach to dynamic sensor planning problems in visual servoing. Specifically, one of the main problems in image-based visual servoing is to plan the camera trajectory in order to avoid undesired configurations (*e.g.*, features out of view, collision with obstacles, ...). Our approach uses the robot redundancy and employs a control scheme based on the task function approach. It combines the regulation of the selected vision-based task with the minimization of a secondary cost function, which reflects given constraints on the manipulator trajectory. We describe how this methodology is applied to common problems in robotic vision: occlusion avoidance, field of view constraint and obstacle avoidance. We have demonstrated the validity of this approach with various experiments.*

1 Overview

One of the key points of the perception action cycle is the automatic generation of the camera motion. Visual servoing [8][4][9] appears to be a very efficient approach to this problem. Although many of the theoretical control issues are now well known, the integration of visual servoing into complex robotics systems remains difficult for various reasons. Considering the case of an eye-in-hand architecture, planning camera trajectory remains an important issue. Indeed, if the control law computes a motion that leads the camera to undesired configurations (such as manipulator joint limits, occlusions or obstacles), visual servoing will fail. Control laws taking into account these "bad" configurations have thus to be considered.

We have chosen to build in avoidance of undesirable configurations using a control scheme based on the task function approach [14][4]. It combines the regulation of the vision-based task with the minimization of a cost function which reflects the constraints imposed on the trajectory. The visual task is considered as a primary and priority task. The cost function is then embedded in a secondary task which only the components which are compatible with the primary task are taken into account (*i.e.*, the minimization of the cost function is performed under the constraint that the visual task is realized). This cost function to be minimized is based on a measure of the risk of the occurrence of an undesired configuration. It must reach its maximal value when these configurations are likely to occur and its gradient must be equal to zero when the cost function reaches its minimal value [14]. In this paper, we applied the proposed methodology to various problems such as occlusions avoidance, constraints on the field of view (*i.e.*, keeping an object inside view), 3D contact in a cluttered environment (*i.e.*, obstacle avoidance). This method as been previously used for singularities and joint limits avoidance [11].

A similar approach has been proposed by Nelson and Khosla. It consists of minimizing an objective function which realizes a compromise between the visual task (a target tracking using a camera mounted on the end effector of a manipulator) and the avoidance of kinematic singularities, joint limits singularities but also with some other constraints on the field of view, the focus measure [12]. This function is used by exploiting the robot degrees of freedom which are redundant with respect to the visual task. However, the resulting camera motions can produce major perturbations in the visual servoing since they are generally not compatible with the regulation to zero of the selected image features.

The next section of this paper, taken from [11], recalls the application of the task function approach to visual servoing and the expression of the resulting control law. Section 3 describes the approach proposed to dynamic sensor planning. We finally present real time experimental results dealing with various robotic tasks. These results have been obtained using an eye-in-hand system composed of a camera mounted on the end-effector of a six d.o.f Zebra Zero robot.

2 Visual Servoing

The *image-based visual servoing* consists in specifying a task as the regulation in the image of a set of visual features[4][8]. Embedding visual servoing in the task function approach [14] allows us to take advantage of general results helpful for the analysis and the synthesis of efficient closed loop control schemes. A good review and introduction to visual servoing can be found in [9].

Let us denote $\underline{P}$ the current value of the set of selected visual features used in the visual servoing task and measured from the image at each iteration of the control law. To ensure the convergence of $\underline{P}$ to its desired value $\underline{P}_d$, we need to know the interaction matrix $L_{\underline{P}}^T$ defined by the classical

[1] Current address is Éric Marchand, IRISA - INRIA Rennes, Campus de Beaulieu, F-35042 Rennes Cedex, France.

equation [4] :

$$\dot{\underline{P}} = L_{\underline{P}}^T(\underline{P}, \underline{p})T_c \quad (1)$$

where $\dot{\underline{P}}$ is the time variation of $\underline{P}$ due to the camera motion T_c. The parameters $\underline{p}$ involved in $L_{\underline{P}}^T$ represent the depth information between the considered objects and the camera frame.

A vision-based task $\underline{e}_1$ is defined by:

$$\underline{e}_1 = C(\underline{P} - \underline{P}_d) \quad (2)$$

where C, called combination matrix, has to be chosen such that $CL_{\underline{P}}^T$ is full rank about the desired trajectory $q_r(t)$. It can be defined as $C = WL_{\underline{P}}^{T+}(\underline{P}_d, \underline{p}_d)$. Assumptions on the shape and on the geometry of the considered objects in the scene have thus generally to be done in order to compute the desired values $\underline{P}_d$ and $\underline{p}_d$. In that case, we set W as a full rank matrix such that Ker W = Ker $L_{\underline{P}}^T(\underline{P}_d, \underline{p}_d)$.

If the vision-based task does not constrain all the n robot degrees of freedom, a secondary task $\underline{g}_s$ can also be performed and we obtain the following task function:

$$\underline{e} = W^+\underline{e}_1 + (I\!I_n - W^+W)\underline{g}_s^T \quad (3)$$

where

- W^+ and $I\!I_n - W^+W$ are two projection operators which guarantee that the camera motion due to the secondary task is compatible with the regulation of $\underline{P}$ to $\underline{P}_d$. Indeed, due to the choice of matrix W, $I\!I_n - W^+W$ belongs to Ker $L_{\underline{P}}$, which means that the realization of the secondary task will have no effect on the vision-based task ($L_{\underline{P}}^T(I\!I_n - W^+W)\underline{g}_s^T = 0$). On the other hand, if errors are introduced in $L_{\underline{P}}^T$, $I\!I_n - W^+W$ no longer exactly belongs to Ker $L_{\underline{P}}$. This will induce perturbations on the visual task due to the secondary task. Let us finally note that, if the visual task constrains all the n degrees of freedom of the manipulator, we have $W = I\!I_n$, which leads to $I\!I_n - W^+W = 0$. It is thus impossible in that case to consider any secondary task.
- $\underline{g}_s$ is the gradient of a cost function h_s to be minimized ($\underline{g}_s = \frac{\partial h_s}{\partial \underline{r}}$). This cost function is minimized under the constraint that $\underline{e}_1$ is realized.

In order to make $\underline{e}$ exponentially decrease and then behave like a first order decoupled system, we get:

$$T_c = -\lambda\underline{e} - W^+\widehat{\frac{\partial \underline{e}_1}{\partial t}} - (I\!I_n - W^+W)\frac{\partial \underline{g}_s^T}{\partial t} \quad (4)$$

where:

- T_c is the camera velocity;
- λ is the proportional coefficient involved in the exponential convergence of $\underline{e}$;
- $\widehat{\frac{\partial \underline{e}_1}{\partial t}}$ represents an estimation of a possible autonomous target motion. If the scene is static, we can assume that $\frac{\partial \underline{e}_1}{\partial t} = \widehat{\frac{\partial \underline{e}_1}{\partial t}} = 0$.

3 Dynamic sensor planning

As already stated, when the vision-based task does not constrain all the six camera degrees of freedom, a secondary task can be combined with $\underline{e}_1$. Thus we can use the redundant degrees of freedom to propose a dynamic sensor planning strategy.

3.1 Avoiding occlusions

The main goal here is to avoid the occlusion of the target by static or moving (with unknown motion) objects. Therefore, the manipulator has to perform adequate motion in order to avoid the risk of occlusion while it ensures the desired constraints between the camera and the target (see Figure 1).

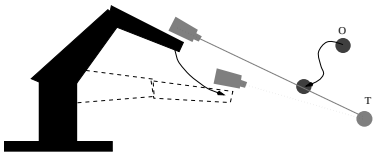


Figure 1: Reactive behavior for occlusion avoidance

Let us consider $\mathcal{O}$ the projection in the image of the set of objects in the scene which can possibly occlude the target T: $\mathcal{O} = \{O_1, \ldots O_n\}$. According to the presented methodology we have to define a function h_s which reaches its maximum when the target is occluded by another object of the scene. We thus define h_s as:

$$h_s = \frac{1}{2}\alpha \sum_{i=1}^{n} e^{-\beta\left(\|T - O_i\|^2\right)} \quad (5)$$

where α and β are two scalar constants. α sets the amplitude of the control law due to the secondary task. The components of $\underline{g}_s$ and $\frac{\partial \underline{g}_s}{\partial t}$ involved in (4) are then:

$$\underline{g}_s = \frac{\partial h_s}{\partial \underline{r}} = \frac{\partial h_s}{\partial \underline{P}}\frac{\partial \underline{P}}{\partial \underline{r}}, \qquad \frac{\partial \underline{g}_s}{\partial t} = 0$$

Computing $\frac{\partial h_s}{\partial \underline{P}}$ is seldom difficult. $\frac{\partial \underline{P}}{\partial \underline{r}}$ is nothing but the interaction matrix $L_{\underline{P}}^T$ or image jacobian.

Let us consider the case of a single occluding object ; a generalization to multiple objects is straightforward. We want to see the target T at the center of the image. Thus we will consider the coordinates $\underline{P} = (X, Y)$ as its center of gravity. If we also consider the occluding object $\mathcal{O}$ by a point $\underline{P}_{\mathcal{O}} = (X_{\mathcal{O}}, Y_{\mathcal{O}})$, defined as the closest point of $\mathcal{O}$ to T, we have:

$$h_s = \frac{1}{2}\alpha e^{-\beta\|\underline{P} - \underline{P}_{\mathcal{O}}\|^2}$$

and

$$\underline{g}_s = \frac{\partial h_s}{\partial \underline{r}} = \frac{\partial h_s}{\partial X}\begin{pmatrix} -1/z_{\mathcal{O}} \\ 0 \\ X_{\mathcal{O}}/z_{\mathcal{O}} \\ X_{\mathcal{O}}Y_{\mathcal{O}} \\ -(1+X_{\mathcal{O}}^2) \\ Y_{\mathcal{O}} \end{pmatrix} + \frac{\partial h_s}{\partial Y}\begin{pmatrix} 0 \\ -1/z_{\mathcal{O}} \\ Y_{\mathcal{O}}/z_{\mathcal{O}} \\ 1+Y_{\mathcal{O}}^2 \\ -X_{\mathcal{O}}Y_{\mathcal{O}} \\ -X_{\mathcal{O}} \end{pmatrix}$$

with

$$\frac{\partial h_s}{\partial X} = -\alpha\beta(X - X_{\mathcal{O}})e^{-\beta\|\underline{P}-\underline{P}_{\mathcal{O}}\|^2}$$

and

$$\frac{\partial h_s}{\partial Y} = -\alpha\beta(Y - Y_{\mathcal{O}})e^{-\beta\|\underline{P}-\underline{P}_{\mathcal{O}}\|^2}$$

3.2 Field of view constraints

Let us consider that we want to keep a set $\mathcal{O}$ of objects in the field of view of the camera while ensuring a positioning with respect to another object T.

We have here to define a cost function h_s which is equal to 0 when $\mathcal{O}$ are located at the middle of the image and which is maximal near the border if the image. However, it is not always realistic with respect to the primary task to define h_s as a linear function of the distance to the center of the image. Furthermore, seeing $\mathcal{O}_i$ at the middle of the image is not required by this process. Therefore we do not use a linear function as proposed in [12] but the following cost function:

$$h_s = \sum_{i=1}^{n} h_s(O_i) \tag{6}$$

with:

$$h_s(O_i) = \alpha e^{\beta(dc(O_i)^2)} \tag{7}$$

where $dc(O_i)$ denotes the distance between O_i to the center of the image. Such a function increases quickly in the vicinity of the border of the image but reaches a nearly null value when O_i is located in a circle located around the center of the image and which radius can be easily tuned using β.

O_i can be either any object of the scene (not necessary related to the focused object T – see for example the results proposed in the next section) or any point of the edge of the focused object. It is possible to choose $\mathcal{O}$ as the set of points located on the edge of the focused object. It is thus possible to ensure that the whole object will be observed by the camera.

Note that the function h_s as proposed in (7) is not the only possible one. Another function (more similar to the one proposed by Nelson [12]) can be defined as:

$$h_s(O_i) = \begin{cases} \alpha\left(1 - \frac{db(O_i)}{dmin}\right) & \text{if } db(O_i) \leq dmin \\ 0 & \text{otherwise} \end{cases}$$

where $db(O_i)$ defines the distance between O_i to the nearest border of the image and $dmin$ is a predetermined threshold.

4 Trajectory planning

In the previous section, we have considered a single camera mounted on the end effector of a manipulator and the secondary task can be seen as a constraint introduced in the camera trajectory. In this section, we will consider a different problem: obstacle avoidance. To achieve this task, we cannot consider only one camera. A second motionless camera is added to the system and provides a global view of the scene by observing the gripper and the target (see Figure 2). Let us call $\mathcal{C}^l$, the camera mounted on the robot, the "local" camera and $\mathcal{C}^g$ the "global" camera. The main difference with the approach presented in the previous section is that the secondary task will no longer be used to *constrains* the camera trajectory but will *provide* to the system a collision free path.

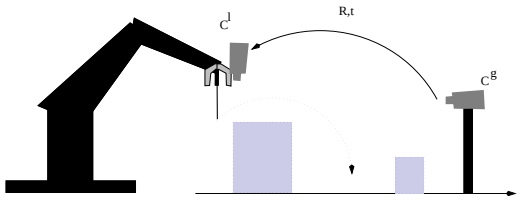


Figure 2: Two cameras system: a local and global view of the scene

4.1 Achieving contact...

The first task we want to perform is simply a point to point contact in 3D space. This kind of task can be achieved using the stereo visual servoing approach [6]. In our case, we will used two cameras as described above. To achieve the described task in an object free scene is straightforward. The primary task is to minimize the error between the current position of the target in the "local" image $\underline{P}^l$ and the position of the gripper $\underline{P}^l_d$ while the secondary task is nothing but the distance between gripper $\underline{P}^g$ and the target $\underline{P}^g_d$ in the "global" image. Note that $\underline{P}^l_d$ and $\underline{P}^g_d$ are fixed in the corresponding images.

$$h_s = \|\underline{P}^g - \underline{P}^g_d\|^2 \text{ and } \underline{g}_s = \frac{\partial h_s}{\partial \bar{r}} = \frac{\partial h_s}{\partial \underline{P}^g}\frac{\partial \underline{P}^g}{\partial \bar{r}}, \quad \frac{\partial \underline{g}_s}{\partial t} = 0$$

where $\frac{\partial \underline{P}^g}{\partial \bar{r}}$ defines the relation between the velocity of $\underline{P}^g$ in $\mathcal{C}^g$ and $\mathcal{C}^l$ velocity. It is given by:

$$\frac{\partial \underline{P}^g}{\partial \bar{r}} = L^T_{\underline{P}^g}\begin{pmatrix} R & -R\left[sk(-R^T t)\right] \\ 0 & R \end{pmatrix}$$

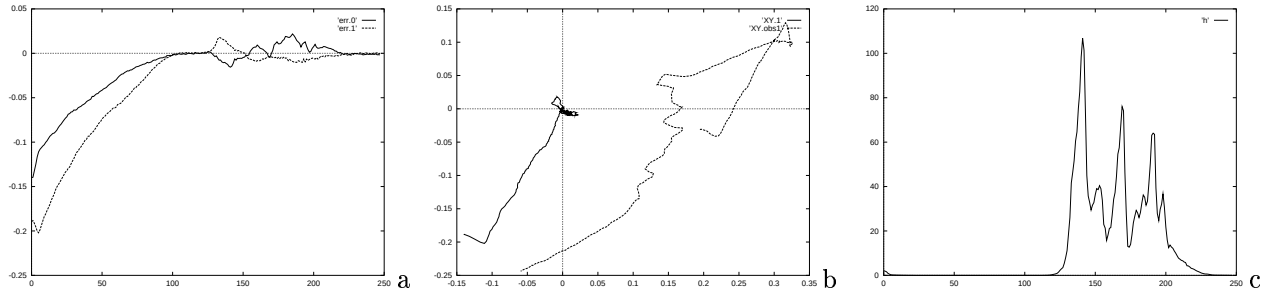
where R and t are the rotation matrix and translation vector associated to the $\mathcal{C}^l$-to-$\mathcal{C}^g$ (or the gripper-to-$\mathcal{C}^g$) rigid transformation and $sk(a)$ is the skew-symmetric matrix associated with vector a. R and t are computed using the method proposed in [5].

4.2 ... in cluttered environment

The problem is quite different if we consider a cluttered environment. Let us actually consider not the whole robot but just the extremity of the gripper defined by a point. We will also consider that we have only 3 d.o.f in translation.

It is possible to modify the previous formulation (Section 4.1) in order to introduce in the cost function h_s a term which increases when the robot moves toward an object. However, if it is possible to ensure that the gripper will not encounter an obstacle, it is hardly possible to propose an object-free path toward the target (*i.e.*, between $\underline{P}^g$ and $\underline{P}^g_d$). To this purpose we propose to use a method derived from the potential fields methods: the navigation functions [10][13].

Let us define by $\mathcal{C}_{free}$ the set of object-free positions of the robot in the image: $\mathcal{C}_{free} = \mathcal{C} \setminus \bigcup_{i=1}^{n} \mathcal{B}_i$ where $\mathcal{B}_i$ is an obstacle and $\mathcal{C}$ is the configuration space. A navigation function is a potential function $U : \mathcal{C}_{free} \mapsto \mathbb{R}$ with a minimum located at the goal and whose domain of attraction includes



a b c

Figure 3: Positioning with respect to a point using an occlusion avoidance process (a) error between the current position of the point in the image and the desired position ($\underline{P} - \underline{P}_d$), (b) position of the two objects in the image, and (c) cost function h_s

the entire subset of $\mathcal{C}_{free}$ connected to the goal [10]. To compute this navigation function we have used the algorithm proposed in [1].

The resulting navigation function U is strictly decreasing and admits only one minimum located at the goal. Knowing U, the cost function can be defined as:

$$h_s = \beta U(X_d^g, Y_d^g)$$

where β is a scalar constant which sets the amplitude of the control law due to the secondary task.

The term $\partial h_s / \partial \underline{s}$ involved in the computation of g_s is merely the spatial gradients of h_s, *i.e.*:

$$\frac{\partial h_s}{\partial \underline{s}} = -\beta \begin{pmatrix} \nabla U_X \\ \nabla U_Y \end{pmatrix}$$

Using this formulation, the gripper will avoid any obstacle visible from the fixed camera. Furthermore if a path exists, then the specified task will be also achieved. If the obstacle and camera $\mathcal{C}_g$ are static then only the spatial gradients of U have to be recomputed. However, this approach can also deal with moving obstacles. In that case, the navigation function has to be recomputed at each iteration of the control law.

5 Experimental results

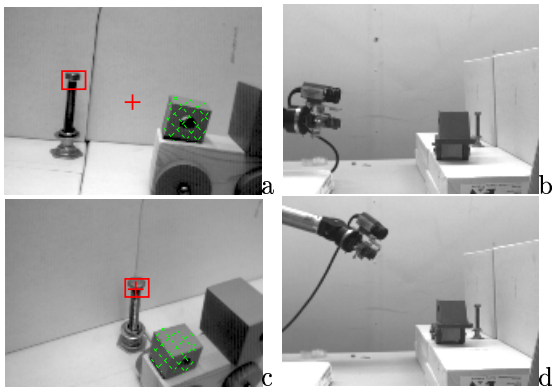
The method described above has been implemented on an experimental cell at Yale University. We have used a CCD camera mounted on the effector of a 6 d.o.f Zebra Zero robot arm. Image processing is performed at video rate using the XVision system [7].

5.1 Handling occlusions

In this experiment (as well as in the next one) we will consider a gaze control task. If $\underline{P} = (x, y)$ describes the position of the projection of the center of gravity of the "target", the goal (*i.e.*the primary task) is to observe this object at the center of the image: $\underline{P}_d = (0, 0)$. Only two degrees of freedom are necessary to perform the vision-based task, thus four motion components are redundant and can be used to avoid the undesirable configurations.

In this experiment the distance between the camera and the target is approximatively 400mm. An object is moving with an *a priori* unknown motion between the camera and the target in order to cause an occlusion.

Using the proposed occlusion avoidance process, when the distance in the image between the target and the occluding object decreases, the cost function h_s increases (see Figure 3.c) and the other degrees of freedom are used to avoid the occlusion. When the occluding object moves away from the target, the cost function decreases (oscillations are due to a non-constant velocity of the occluding object).



a b c d

Figure 4: Positioning with respect to a point: (a) initial image acquired by the camera, (b) initial position of the camera with respect to the target (the screw) and the occluding object (the vehicle toy), (c) final image acquired by the camera, and (d) final position of the camera with respect to the target and the occluding object

We can observe that during the occlusion avoidance process, the vision based task is not perfectly achieved (see Figure 3.a). This is due to the fact that we do not have any information on the relative position between the camera and the target: the depth information z involved in the interaction matrix is unknown. As $L_{\underline{P}}^T$ cannot be updated at each iteration of the control law, assumptions have been made about the depth of the target.

5.2 Field of view

In this experiment we want to focus on a target (here the top hole on the cylinder) while keeping the trailer (see right of Figure 6) within the field of view of the camera. If no specific strategy is achieved, the vehicle moves out of the image during the camera motion due to the focusing task.

Figures 6.a and 6.b depict the initial image from the camera and an external view of the scene. Figures 6.c and 6.d show the final views. Figure 5.a shows the error between the current position of the target in the image and its desired position. Figure 5.b shows the position in the image of the focused target and the position of the tow which is the closest from the image border. Figure 5.c shows the value of the

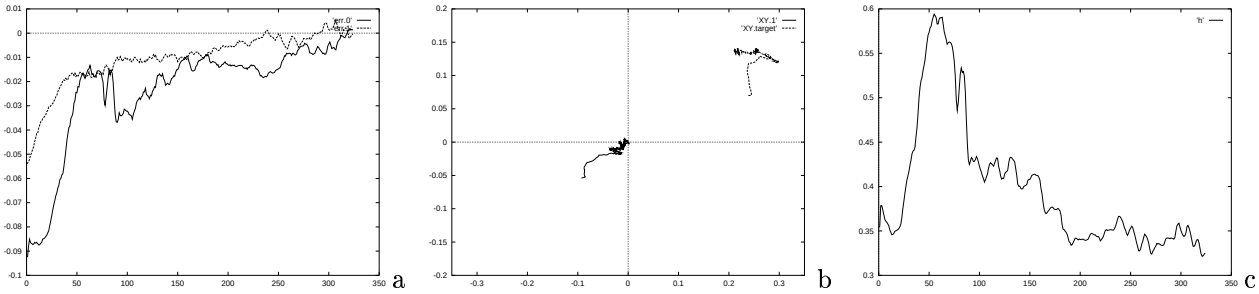


Figure 5: Field of view experiment (a) error $\underline{P} - \underline{P}_d$ (b) position of the focused target and of the vehicle tow in the image (c) cost function h_s

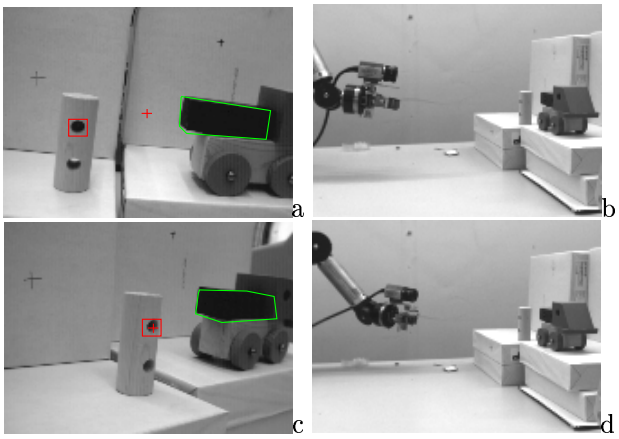


Figure 6: Field of view experiment

cost function h_s. Note that the system reacts rapidly to motion of the vehicle toward the border of the image. A motion along one of the previously unused translational degrees of freedom (x axis) is performed.

5.3 Avoiding obstacles

In this experiment we want to insert a 4mm wide screwdriver in a 5mm hole. It can be shown [6] that if the screwdriver and the hole are superimposed in the two images, then they are also superimposed in the 3D space. To achieve this task, we have used the three translational d.o.f of the robot. Introducing the rotational d.o.f into the process will require a more complex planning strategy to avoid the obstacles. Figure 9). As a result, in this experiment, the primary task will control 2 d.o.f (x and y in the mobile camera frame) while the secondary task controls the last one (z in this frame).

Figure 7 depicts the first and the last image acquired during the insertion process. The navigation function U is computed in the vicinity of group obstacle/target/robot using the method proposed in [1]. Figure 8 shows the resulting navigation function. Light areas correspond to high value of the navigation function while dark ones correspond to lower values. Obstacles are shown in black.

Figure 9.a depicts the error between the current and desired position in the image of the mobile camera. A small error (2 or 3 pixels which corresponds to a 1mm error in 3D space) can be observed due to a bad estimation of projection operators (W and $\mathbb{I} - W^+W$). Here again, the position of

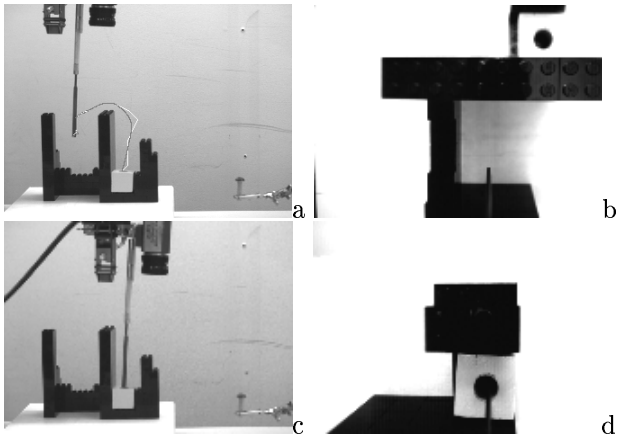


Figure 7: Insertion task in a cluttered environment: (a-b) initial image acquired by the cameras (a) global camera (b) local camera, and (c-d) final images. Note that the camera trajectory is depicted in image (a).

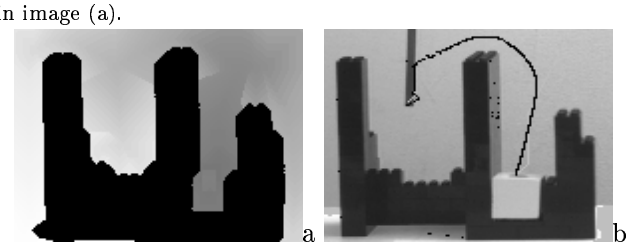


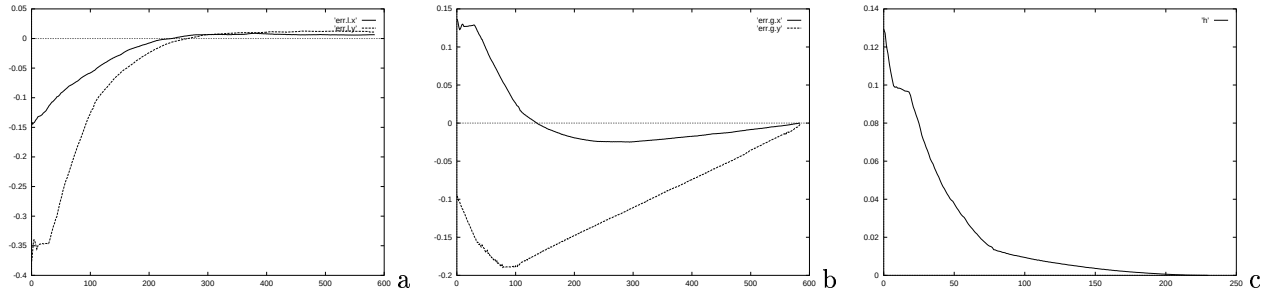
Figure 8: Obstacle avoidance (a) Navigation function (b) camera trajectory around the obstacle

the target w.r.t. the camera is unknown.

Figure 9.b depicts the error between the current and desired position of the screwdriver in the fix (global) image. In a first time (iterations 0–80) the error increases due to obstacle avoidance process (see robot trajectory on Figure 8.b). Figure 9.c depicts the cost function h_s which is the value of the navigation given the position of the robot.

6 Conclusion and future work

We have shown in various cases that it was possible to use redundancy to achieve dynamic sensor planning in visual servoing. Each time the constraints on the camera trajectory has been expressed as a function to be minimized with respect to the specified task. Results have been proposed for occlusion avoidance, field of view constraints and in a simple case for obstacle avoidance. Previous work has demonstrated



a b c

Figure 9: Obstacle avoidance experiment (a) error $\underline{P}^l - \underline{P}^l_d$ (b) error $\underline{P}^g - \underline{P}^g_d$ (c) cost function h_s

that the proposed method is able to deal with other problems such as singularities and joint limits avoidance [11] or for trajectory tracking[2]. Other tasks may be achieved such as introducing constraints on the measure of focus, resolvability (Nelson gives some cost functions for those purposes in [12]), visibility, resolution [3]. Most of these tasks have been solved in the case of static sensor planning (see for example [15]) and cost functions have been proposed. Using these cost functions within the framework proposed here is possible. However, here in each case, only image based control has been used. We have never used 3D information and our system has never been calibrated. To solve some others problems, more *a priori* informations (*e.g.*, a CAD of the scene) may be required.

Future work will be dedicated to incorporating an on-line estimation of depth of the objects using dynamic vision into the closed loop. Therefore we should be able to know if, in the occlusion avoidance problem, an object will actually occlude the target. Furthermore, this will give us a better estimation of the interaction matrix and thus of the projection operators. Other work has to be done dealing with the obstacle avoidance process. Obviously, the method presented here may fail (a path in the global image may not exist) and may not be optimal (a shorter path may exists using the other image). Future work will be devoted to determine a good path by computing new camera viewpoints. We will also consider the whole robot and be able to deal with motion along the rotation axes.

Acknowledgments

Éric Marchand was funded by INRIA (Institut National de la Recherche en Informatique et Automatique, France) under a postdoctoral fellowship. Work by Greg Hager was supported by NSF IRI-9420982. Authors whish to thank F. Chaumette for the discussions they had.

References

[1] J. Barraquand, B. Langlois, J.-C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):224–241, February 1992.

[2] F. Chaumette. Visual Servoing using image features defined upon geometrical primitives. *Conf on decision and Control*, Vol. 4, pp. 3782–3787, Orlando, December 1994.

[3] C.K. Cowan, P.D. Kovesi. Automatic sensor placement from vision task requirements. *IEEE Trans. on PAMI*, 10(3):407–416, May 1988.

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

[5] N. Fischler, R.C. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communication of the ACM*, 24(6):381–395, June 1981.

[6] G. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE Trans. on Robotics and Automation*, 13(4):582–595, August 1997.

[7] G. Hager, K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 1998.

[8] K. Hashimoto, editor. *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, Singapor, 1993.

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

[10] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[11] E. Marchand, F. Chaumette, A. Rizzo. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'96*, Vol. 3, pp. 1083–1090, Osaka, Japan, November 1996.

[12] B.J. Nelson, P.K. Khosla. Integrating sensor placement and visual tracking strategies. In *IEEE Int. Conf. Robotics and Automation*, Vol. 2, pp. 1351–1356, San Diego, May 1994.

[13] E. Rimon, D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Trans. on Robotics and Automation*, 8(5):501–518, October 1992.

[14] C. Samson, M. Le Borgne, B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[15] K. Tarabanis, P.K. Allen, R. Tsai. A survey of sensor planning in computer vision. *IEEE Trans. on Robotics and Automation*, 11(1):86–104, February 1995.