Proceedings of the 2001 IEEE/RSJ
International Conference on Intelligent Robots and Systems
Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001

Sensor-Based Navigation in Cluttered Environments

Ricardo Swain-Oropeza, Michel Devy, Seth Hutchinson

Abstract— **In this paper, we present a new approach to sensor-based navigation in cluttered environments. In our system, tasks are specified in terms of visual goals, and obstacles are detected by a laser range finder. To effect task performance, we introduce a new gain scheduling visual servo controller. Our approach uses a diagonal gain matrix whose entries are adjusted during execution according to one of several proposed gain schedules. Obstacle avoidance is achieved by allowing the detected obstacles to generate artificial repulsive potential fields, which alter the motion of the mobile robot base. Since this motion affects the vision-based control, it is compensated by corresponding camera motions. Finally, we combine the obstacle avoiding and visual servo components of the system so that visual servo tasks can be performed as obstacles are avoided. We illustrate our approach with both simulations and real experiments using our experimental platform Hilare2Bis.**

Keywords— **Sensor-based mobile robotics, visual servoing**

I. Introduction

In recent years, much progress has been made in the area of sensor-based robotics. In particular, visual servo control has seen increasing popularity as a method for specifying sensor-based robotic tasks [11], [15]. However, in spite of its growing popularity, there are still many tasks for which visual servo control is not well suited. Obstacle avoidance is one such task, since avoiding obstacles requires recognizing obstacles, and the recognition task is not yet readily solved in real time by computer vision systems. In contrast, obstacles can be easily recognized using real time range sensors (e.g., sonar or laser range finders). For these reasons, we have chosen to investigate a multi-sensor approach, in which tasks specified in terms of visual goals are performed using visual servo control techniques, while obstacles are detected using a laser range finder. The resulting system is able to perform fairly complex navigation tasks in a structured but cluttered environment.

Our multi-sensor approach requires that inputs from the vision system and the laser range finder be combined to determine control inputs for the robot. This has led us to develop a new visual servo controller, which we have integrated with an obstacle avoidance system. In particular, in Section II we describe a new, gain scheduling visual servo control system. The system a diagonal gain matrix whose entries are adjusted during execution according to one of several proposed gain schedules.

In Section III, we describe how obstacle avoidance is effected, using a potential field method that alters the motion of the mobile robot based on data from the laser range finder. Since motion of the robot base will induce a camera motion, it must be compensated. Our approach for providing this compensation is also described in Section III.

In Section IV we combine the results of Sections II and III, yielding a system that can avoid obstacles while performing visual servo tasks. Our approach uses a convex combination of inputs provided by the obstacle avoidance and visual servo systems, with the emphasis gradually shifting between obstacle avoidance and visual servo performance as the execution progresses.

Besides numerous simulation experiments, we have implemented our approach on our robot, Hilare2Bis (shown in Figure 8), a nonholonomic robot equipped with a SICK laser range finder and a camera mounted at the end effector of a six degree of freedom arm. At present, we use the arm to effect a pan/tilt mechanism for the camera (i.e., we do not fully use the six degrees of freedom). We present experimental results in Section V.

The work that we describe in this paper is built on our past research. In [32], [33], we have presented how visual servoing formalism can be used to execute high-level commands like *Follow a wall* or *Turn around a corner*. In this work, a planner was integrated with a visual servo system, and the planner determined the sequence of visual servo tasks to be performed, as well as trajectories to avoid obstacles. Later in [34], [35], [6] we presented some preliminary results for avoiding obstacles during visual servoing.

II. A Gain Scheduling Visual Servo Controller

In this section we provide a brief review of visual servo control, and then present our new gain scheduling approach.

A. A Brief Review of Visual Servoing

With image-based visual servo control, tasks are defined in terms of features that can be extracted from and tracked in an image sequence [11], [15]. The key tool in this approach is the interaction matrix, also called the image Jacobian matrix, which relates changes in the image to the velocity of the moving camera. In particular, if $\mathbf{s}$ denotes the features that are extracted from the image, then the interaction matrix satisfies

$$\dot{\mathbf{s}} = L^T T_c \tag{1}$$

in which $\dot{\mathbf{s}}$ represents the variation of the features in the image, and T_c is the camera kinematic screw. In our exper-

R. Swain-Oropeza is with ITESM CEM, Computer Science Department, México. E-mail: rswain@campus.cem.itesm.mx. This work was completed during the doctoral studies of Ricardo Swain at LAAS-CNRS

M. Devy is with LAAS-CNRS, Toulouse, France. E-mail: michel@laas.fr

S. Hutchinson is with Department of Electrical and Computer Engineering and with the Beckman Institute at University of Illinois at Urbana-Champaign, Illinois, USA. E-mail: seth@uiuc.edu

0-7803-6612-3/01/$10.00©2001 IEEE

iments we use only point features, (i.e., the image features are points in the image, which are projections of 3D points in the world), and thus L^T describes the relationships between point velocities in the image and the camera motion. In this case, L^T is a function of both $\mathbf{s}$ and the depth of the scene points relative to the camera. The interaction matrix also depends on the camera parameters (which must be determined during camera calibration), but several evaluations have shown that a coarse approximation to these parameters is sufficient.

For so called eye-in-hand systems, the camera is mounted on a robot arm, which may in turn be mounted on a mobile platform. If we denote by J the manipulator Jacobian for the robot, and by $\mathbf{q}$ the joint variables for the robot, then the relationship between the joint velocities and the image feature velocities is given by

$$\dot{\mathbf{s}} = L^T J \dot{\mathbf{q}}. \tag{2}$$

We can use this result to perform robotic tasks. We define the classical visual task function [11] as the error vector e

$$e = \mathbf{s}(\bar{r}) - \mathbf{s}^* \tag{3}$$

in which $\bar{r}$ denotes the camera position in the environment, $\mathbf{s}(\bar{r})$ is the value of the visual features currently observed by the camera, and $\mathbf{s}^*$ denotes the desired value for the feature vector. We choose to impose an exponential decrease on the error [11], which leads to $\dot{e} = -\lambda e$, with a scalar value for the gain λ. Combining this constraint with equations (2), (3) we obtain the following result:

$$\dot{\mathbf{q}} = -\lambda (L^T J)^+ (\mathbf{s}(\bar{r}) - \mathbf{s}^*). \tag{4}$$

The dimension of the error vector e depends on the number of features, and for most applications, the matrix product $(L^T J)$ is not a square matrix. In our previous work [32], we have explored two possible solutions to this problem: we have used the pseudo-inverse, $(L^T J)^+$, as in (4), and we have used the SVD method to solve the corresponding linear system (shown in (6) below). We have found the latter approach to be more reliable.

As mentioned above, L is a function of both the feature vector, $\mathbf{s}$, and of the depth of the corresponding scene points. Thus, it is necessary to estimate the depth of the scene points in order to compute the interaction matrix. There are several possible alternatives to directly estimating depth, and a good review of these can be found in [7]. A fairly popular approach is to use the value z^* (i.e., the depth of the scene points when the goal is achieved) for the depth. Further, it is sometimes beneficial to compute the interaction matrix off-line. To this end, the value $\mathbf{s}^*$ is sometimes used instead of $\mathbf{s}$, allowing the off-line computation of L, since $\mathbf{s}^*$ is constant. Figure 1 illustrates these choices. In the figure on the left, $\mathbf{s}$ and the actual depth value are used, while in the rightmost figure, $\mathbf{s}^*$ and the value z^* (i.e., the depth of the scene points when the goal is achieved) are used. This approximation to L^T (i.e., using the values $\mathbf{s}^*$ and z^*) affects the convergence, and the trajectory of the feature points in the image, increasing the risk that the features will exit the field of view. Therefore, this approximation is useful only when the required feature motion is small, i.e., when the features are initially near to their goal configuration. A second example is shown in Figure 2 for our mobile robot executing a visual servo task, computing L using z^* and $\mathbf{s}^*$. In this case, the features leave the image during execution.

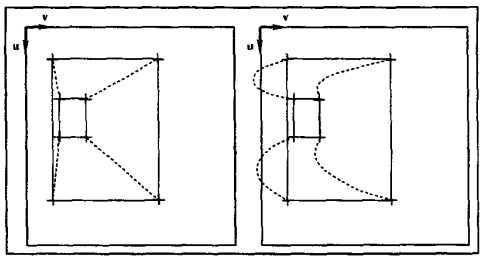


Fig. 1. Feature trajectories for two methods of computing L: On the left, L is computed using actual depth and $\mathbf{s}$. On the right L is computed using desired depth, z^* and $\mathbf{s}^*$.

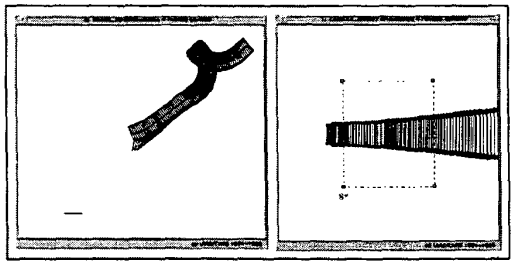


Fig. 2. The robot motion is illustrated on the left, and the feature trajectories in the image are shown on the right, both for the case when L is computed using z^* and $\mathbf{s}^*$.

B. The New Gain Scheduling Approach

The choice of λ in (4) is crucial for performance. At the beginning of the task, λ should be chosen rather small (less than 1), since at this stage of execution the error will be large, and a large value of λ would lead to undesired effects (loss of image features, saturation of the actuators, etc.). However, at the end of the task, the error will be very small and a large value of λ (greater than 1) will be needed to generate control values sufficiently large to be executed by the robot. Finally, instability problems occur when λ is chosen far too high. This phenomenon is linked to the proportional structure of the control law. For this kind of control law, control theory shows that the choice of a gain that is too high leads to instability. For these reasons, we have chosen to use an adaptive gain, an approach also known as gain scheduling. Gain scheduling has also been used by [3], [16]. An alternative approach has been proposed in [8], in which avoiding problematic configurations is posed as a secondary task.

There are several choices for scheduling the gains, including constant, linear, and exponential (see, e.g., [33] for more

detail). We have chosen an exponential schedule, given by

$$\lambda_k = \beta\, e^{-\gamma\, |e_{(k)}|_{max}} + \delta \quad \text{with} \quad \beta, \gamma, \delta > 0 \qquad (5)$$

where β, γ and δ are other gains that must be chosen, and $| e_{(k)} |_{max}$ is the maximal value of e in the iteration k. The performance difference between the schedule in (5) and a linear schedule is illustrated in Figure 3. Figure 3a shows visual servo results using a linear schedule for λ and Figure 3b shows results using (5). In each of these, the left figure illustrates the motion of the mobile robot, the center figure illustrates the feature trajectories in the image, and the right figure illustrates the variation of λ during the task execution. We note that, although the results appear very similar, in the linear case the convergence is more rapid, with larger motions for the time steps at the beginning of the trajectory.

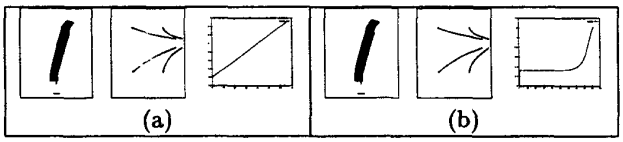


Fig. 3. (a) Linear schedule for λ. (b) Schedule given in (5).

To avoid some typical problems that arise during visual servoing (see, e.g., [7], [36], [10]) and to keep features from leaving the image, we use a diagonal gain matrix, Λ, instead of the scalar gain given in (5). Thus, (4) becomes

$$L^T J\, \dot{\mathbf{q}} = -\Lambda\, (\mathbf{s}(\bar{r}) - \mathbf{s}^*) \qquad (6)$$

in which $\Lambda = diag(\lambda_i) \in \Re^{n \times n}$, with n the dimension of the error vector. The value of each λ_i is chosen using (5), with $| e_{(k)} |_{max}$ replaced by the absolute value of the i^{th} component of e. This leads to the control law

$$\dot{\mathbf{q}}^{vs} = -(L\, J)^{\dagger} \Lambda (\mathbf{s}(t) - \mathbf{s}^*), \qquad (7)$$

in which $\dot{\mathbf{q}}^{vs}$ denotes the commanded joint velocities from the visual servo system. We use here the superscript vs to distinguish between the control provided by visual servo and control used for obstacle avoidance (which we discuss below, in Section III).

The use of matrix Λ provides more degrees of freedom in designing the controller, which allows us to adapt the dynamics of each component e_i of e. For example, if $\mathbf{s}$ is a 2-dimensional vector $(u\, v)^T$ representing the coordinates of a point in the image, then, imposing a faster decrease for $\mathbf{v}$ than for $\mathbf{u}$ leads to horizontal centering of the features before approaching the target. At the beginning of the task, the features will be brought to the horizontal center (closer to the desired position in $\mathbf{v}$), reducing the risk of the features leaving the image. As will be described in Section III, our main interest is to keep the center of gravity of the target in the middle of the image in order to eliminate the possibility of feature loss during the avoiding obstacle phase.

Figure 4 shows experimental results for a diagonal gain matrix, using the linear schedule (Figure 4A) and an exponential schedule (Figure 4B). As mentioned previously, we can see that the work area is almost always near the center of the image when we use a Λ matrix (Figure 4).

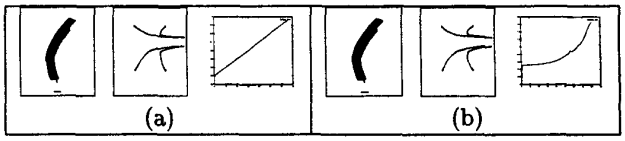


Fig. 4. (a) Linear schedule for Λ. (b) Schedule given in (5) for each λ_i.

Figure 5 shows the results of using our new approach for the same task as is illustrated above in Figure 2. Note that with the new control scheme the features do not exit the field of view and the task is successfully executed.

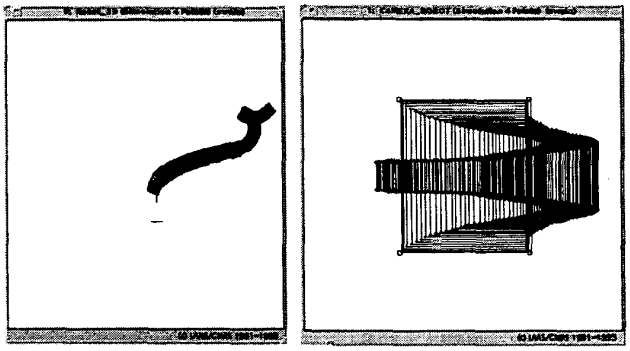


Fig. 5. The experiment of Figure 2 under the new control scheme.

III. Avoiding Obstacles

Now, let us consider the presence of unforeseen obstacles in the scene. Several methods of obstacle avoidance exist in the literature, and sensor-based navigation has been previously examined by different researchers, using sensors such as stereo vision [30], [24], [14], monocular vision using the normal flow [9], [30], ultrasonic sensors [17], and laser range finders [25].

In this paper, we wish to combine vision and range data (our system uses a SICK laser range finder) to perform a visual servoing task while avoiding obstacles. Our approach will be to construct repulsive potential fields around obstacles, and to allow these potential fields to affect only the motion of the base of the mobile robot. The idea of combining potential fields with visual servo has been previously discussed in [20], [22], and allowing the repulsive field to affect only the motion of the mobile robot base has been used in [38]. Of course the resulting motion of the base of the robot will affect the position of the camera, and these effects must be compensated for in order to keep the target features within the field of view.

In our previous research, we have considered three different approaches to this problem. In [36], the repulsive potential fields affect only the base of the robot and no compensation is made for the induced camera motion. In [34], [35], the base of the robot is controlled by the repulsive forces and a pan/tilt head moves the camera to exactly

cancel the effects of this motion in the image. In [37], [6], the movement of the pan/tilt head is decoupled from the motion of the robot base, allowing obstacle avoidance while performing visual servo tasks.

We will now briefly describe how the camera motion can be used to counteract the motion induced by the repulsive fields. Then, in Section IV we will describe our approach to obstacle avoidance while performing visual servo tasks.

Since the repulsive forces act only on the base of the robot, we partition the control system into two subsystems: one corresponding to the mobile robot base, and one that positions the camera (this could be a pan/tilt head, or, as in our case, a camera mounted to the end effector of a robot arm). Let $\dot{\mathbf{q}}_r^{rep}$ denote the vector of joint velocities for the robot base, induced by the repulsive fields around the obstacles. In our experiments, this potential field is generated using the methods described in [17], [18]. Let $\dot{\mathbf{q}}_c^{rep}$ be the vector of joint velocities for the camera positioning mechanism, commanded to compensate for the motion of the robot base. If we consider, for the moment, only the repulsive forces that act on the robot, we can write the Jacobian relationship for the system as

$$T_c = [J_r \; J_c] \begin{bmatrix} \dot{\mathbf{q}}_r^{rep} \\ \dot{\mathbf{q}}_c^{rep} \end{bmatrix}, \tag{8}$$

in which J_r is the Jacobian matrix for the base, and J_c is the Jacobian matrix for the camera positioning mechanism. Using this relationship along with (6) we can write

$$L\, J_r \dot{\mathbf{q}}_r^{rep} + L\, J_c \dot{\mathbf{q}}_c^{rep} = -\Lambda\, e. \tag{9}$$

We can solve this equation for the velocities $\dot{\mathbf{q}}_c^{rep}$ that will attempt to satisfy the visual servo goals, in effect canceling the effects of the motion of the base

$$\dot{\mathbf{q}}_c^{rep} = -(L\, J_c)^{-1} \left(\Lambda\, e + L\, J_r \dot{\mathbf{q}}_r^{rep}\right). \tag{10}$$

This represents a decoupled system, in which the robot base is controlled so that obstacles are avoided, while the camera positioning mechanism attempts to achieve visual servo goals. If we wish to exactly cancel the effects of the motion of the base on the image, then we merely set $e = 0$, eliminating the visual servo component of the control law. Note that, in general, the motion allowed by the camera positioning mechanism will not be sufficient to perform the visual servo task. Therefore, we now turn out attention to the problem of combining the obstacle avoidance with performing visual servo tasks.

IV. Avoiding Obstacles while Performing Visual Servo Tasks

We can view $\dot{\mathbf{q}}^{vs}$, given in (7), as a control that performs a visual servo task without regard to obstacles. In contrast, $\dot{\mathbf{q}}_r^{rep}$, given by (10), is a control that attempts to avoid obstacles by moving the robot base, while the remaining degrees of freedom (i.e., those that are used to position the camera) attempt to achieve a visual servo goal. As we have mentioned above, the control given in (10) is unlikely to be able to accomplish the visual servo task, since the robot base occupied solely by obstacle avoiding behavior.

To combine obstacle avoidance with visual servo task performance, we propose to use a convex combination of these two controls

$$\dot{\mathbf{q}} = \mu\, \dot{\mathbf{q}}^{vs} + (1 - \mu)\dot{\mathbf{q}}^{rep} \tag{11}$$

in which $0 < \mu < 1$ determines the relative weighting of task performance and obstacle avoidance.

This formalism essentially creates a virtual potential field in which the robot moves. The goal defined in the Cartesian space, is modeled by an attractive potential, the obstacles by a repulsive one, and the robot follows the gradient of the sum of these potentials until it reaches the minimum, see Figures 6 and 7. In this paper, the attractive part of the potential corresponds to $\dot{\mathbf{q}}^{vs}$, which is generated not by the goal in the Cartesian space, but by the goal in the image space, using the visual servoing formalism.

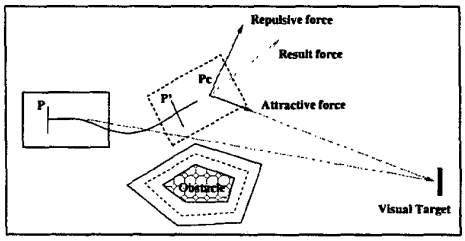


Fig. 6. Combining visual information and repulsive forces

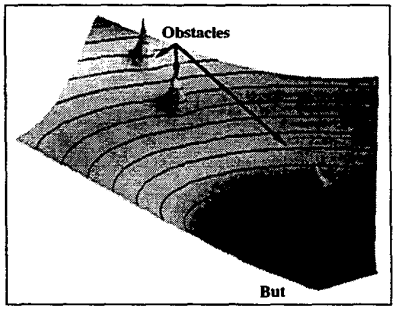


Fig. 7. Potential fields approach

Clearly, the choice of μ has a crucial impact on performance, as it fixes the weight of the visual servoing task during the obstacle avoidance behavior. Therefore, μ must be chosen so that the target features won't be lost because of obstacle avoidance. This problem occurs when the error, e, between the current and desired features becomes too large. Therefore, we have decided to make μ depend on this error. Since μ is a scalar quantity, we will use the error for the center of gravity of the target. We propose the law

$$\mu = 1 - e^{-(s_{gc} - s_{gc}^*)} \tag{12}$$

where s_{gc} and s_{gc}^* represent the current and desired centers of gravity of the image feature points (respectively). Thus, when $s_{gc} - s_{gc}^*$ is large, μ tends to 1, giving a more

significant weight to visual servoing, which brings back the target in the camera's line of view. This leads to a decrease of the error $s_{gc} - s^*_{gc}$ and therefore of μ. Visual servoing is then less important and the robot avoids obstacles without losing the target. Relation (12) shows that μ is less than 1 by construction; however, it can be zero if $s_{gc} = s^*_{gc}$. But, as visual servoing is perturbed by avoiding obstacles movement, it cannot be realized perfectly. So, this problem cannot occur during obstacle avoidance.

Finally, our control law is given by relations (11) and (12). This first attempt to couple vision with US in order to realize a specified task, allows us to perform visual servoing, while avoiding obstacles by a potential field method. The switch between the two tasks is done via μ. When the robot is close to an obstacle, the velocities on the wheels are modified by the repulsive potential, while the one of the arm is still given by the visual servoing.

V. Experimental Results

Our algorithms have been completely implemented on Unix desktops, and then, integrated under the real-time OS VxWorks using a tool for the specification and the implementation of functional modules in a distributed robot architecture named GenoM [13]. To test our algorithms in real-time, we have a simulation system (see Figures 12, 13, 14).

A. The Experimental Testbed: Hilare2Bis

We have performed several experiments using our mobile robot Hilare2Bis, shown in Figure 8. Hilare2Bis is a nonholonomic mobile robot with two actuated wheels, equipped with a six degree of freedom robot arm that holds a camera.

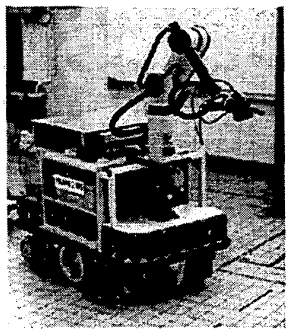


Fig. 8. The robot Hilare2Bis.

In practice, the presence of non holonomic constraints in a mechanical system prevents it from following some paths [23]. Therefore, the application of visual servoing to nonholonomic mobile robots is not direct. It is for this reason that we mount the camera at the end of the robot arm. When the whole mechanical system (robot plus arm) is considered as a single kinematic chain, it becomes possible to fully control the camera motion without being limited by the non holonomic constraint of the mobile basis [27].

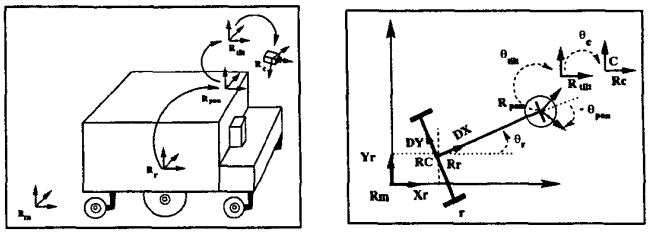


Fig. 9. The mobile robot kinematic model.

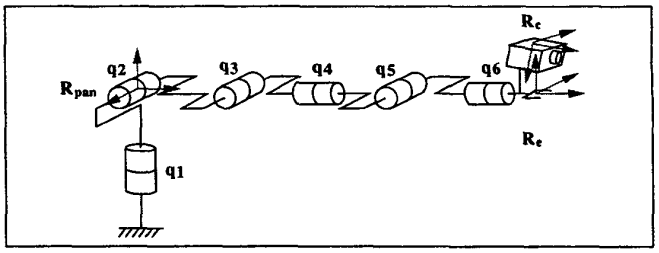


Fig. 10. The arm kinematic model.

The kinematic model of Hilare2Bis and its configuration is represented in Figure 9. Even though the robot arm has six degrees of freedom, for our experiments we have chosen to use the arm as a pan mechanism, as illustrated in Figure 10. We denote by $\dot{q_{pl}}$ the effective panning velocity of the camera, which is effected by the first joint (q_1) for large motions and by the wrist joints for small motions. We denote by $\dot{X_r}$ and $\dot{Y_r}$ the linear speed of the midpoint of the two actuated wheels (which we denote by, C). The angular velocity of the mobile base is given by $\dot{\theta_r}$. Finally, $\dot{q_{ri}}$, $\dot{q_{le}}$, are the speeds of the right and left wheels, respectively. Using this notation, we can apply the formalism proposed by Samson [29] to express the kinematic model of the system as

$$\begin{pmatrix} \dot{X_r} \\ \dot{Y_r} \\ \dot{\theta_r} \\ \dot{\theta_{pl}} \end{pmatrix} = \begin{pmatrix} \frac{r}{2}\cos\theta_r & \frac{r}{2}\cos\theta_r & 0 \\ \frac{r}{2}\sin\theta_r & \frac{r}{2}\sin\theta_r & 0 \\ \frac{r}{2DY} & -\frac{r}{2DY} & 0 \\ \frac{r}{2DY} & -\frac{r}{2DY} & 1 \end{pmatrix} \begin{pmatrix} \dot{q_{ri}} \\ \dot{q_{le}} \\ \dot{q_{pl}} \end{pmatrix}. \quad (13)$$

Using the appropriate coordinate frame transformations, we can express the kinematic model of the entire system (base plus arm) [27] as

$$T_c = J\,\dot{\mathbf{q}} \quad (14)$$

in which T_c is the camera kinematic screw (expressed with respect to the camera frame), J represents the Jacobian matrix of the mechanical system, and finally, $\dot{q} = (\dot{q_{ri}}\ \ \dot{q_{le}}\ \ \dot{q_{pl}})^T$. We note that J depends on several parameters that must be either dynamically read from encoders $(X_r, Y_r, \theta_r, \theta_{pl})$, or estimated off line during the arm configuration. An example Jacobian matrix is given in (15), shown in Figure 11. For this case, the camera is horizontal,

has fixed elevation (i.e., it cannot translate along its own X-axis), and can rotate only about the vertical axis (i.e., its X-axis). More details can be found in [28], [32], [37].

Thus, for our experimental platform, we write (8) as

$$T_c = [J_r \; J_c] \begin{bmatrix} \dot{q}_{ri}^{rep} \\ \dot{q}_{le}^{rep} \\ \dot{q}_{pan}^{rep} \end{bmatrix} \tag{16}$$

and we write (10) as

$$\dot{q}_{pan}^{rep} = (L\, J_c)^{-1} \left(\Lambda\, e + L\, J_r \begin{bmatrix} \dot{q}_{ri}^{rep} \\ \dot{q}_{le}^{rep} \end{bmatrix} \right). \tag{17}$$

To exactly cancel the effect of the robot base motion on the image, we set $e = 0$, and for this system we thus obtain

$$\dot{q}_{pan}^{rep} = (\dot{q}_{le}^{rep} - \dot{q}_{ri}^{rep}) \left(\frac{r}{2DY} \right) \tag{18}$$

B. Simulation Results

We present now simulation results for several complex visual servoing tasks. We define the repulsive potential fields as in [14], [17]. This approach has previously been implemented on the robots in our lab, and runs in real time. A dedicated software module, called *AVOID* [17], is able to give us the linear and angular velocities needed for avoiding obstacles. Using these, it is a simple matter to express the velocities of the right and left wheels, namely, $\dot{q}_{ri}^{rep}$ and $\dot{q}_{le}^{rep}$. We calculate μ as explained above.

In the first simulation, the goal is to position the camera in front of the target while avoiding the obstacles present in the scene (Figure 12A). Figure 12B illustrates the performance of the robot, which avoids the different obstacles without losing the target.

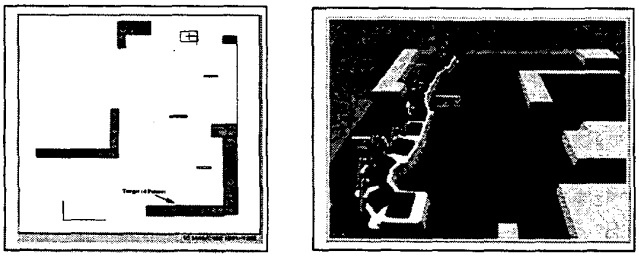


Fig. 12. Initial configuration & achievement of a visual servoing task with obstacle avoidance.

A second simulation is illustrated in Figure 13. Again, the robot is able to perform the task while avoiding obstacles, and, in this case, without being trapped in a local minimum that could arise due to the nonconvex obstacle. More details regarding this simulation can be found in [6].

Figure 14 illustrates a more complex task. Here, the robot is commanded to follow a wall, turn around a corner, follow a wall, and finally to move to a location in front of a target. Several obstacles are present in the workspace, and the robot is able to successfully navigate around these obstacles while performing the appropriate visual servo tasks.

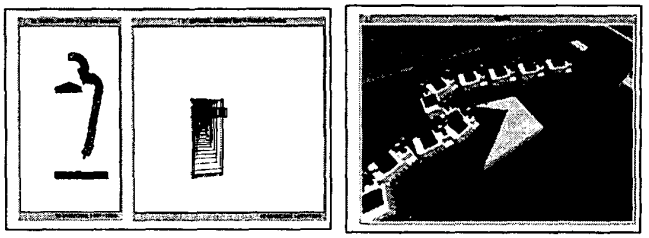


Fig. 13. Achievement of a visual servoing task with obstacle avoidance.

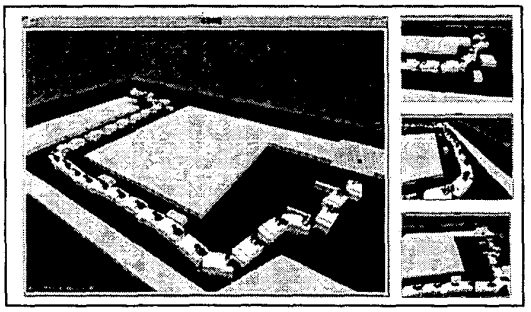


Fig. 14. A more complex task.

C. Experiments using Hilare2Bis

Our first experiment demonstrates the visual servo approach described in Section II (no obstacle avoidance is used in this experiment). The commanded task is *Go to object*. The robot trajectory is illustrated in Figure 15A, and the linear and angular speeds of the robot are shown in Figure 15B. The initial and final configurations for the features in the camera are shown in Figure 16A, and the evolution of the feature trajectories is shown in Figure 16B. Note that the robot performs the task and the features reach the desired positions. For this experiment, the interaction matrix, L, was computed using s^* and z^* as the feature and depth values. Thus, for this experiment, it was possible to compute L off line. The robot motion is periodically computed on line by the solution of equation (6). As for the linear and angular velocities given by the control law (Figure 15B), they do not violate the acceleration and speed constraints of our robot, and therefore can be applied to Hilare2Bis.

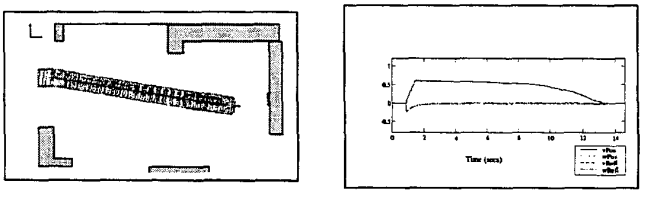
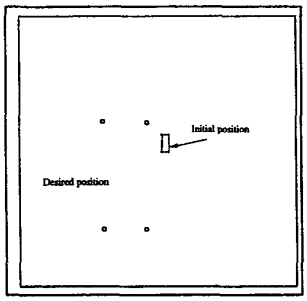


Fig. 15. A) Go to 4 points. B) Linear & Angular Speeds.

Our second experiment illustrates obstacle avoidance while performing a visual servo task. For this experiment, obstacles were detected by a SICK laser range finder mounted on the robot with a cycle time of approximately 25 *ms*. The data given by the SICK laser range finder is

$$J = \begin{pmatrix} 0 & 0 & 0 \\ -\frac{r}{2}(\sin(\theta_{pl}) + \frac{DX}{DY}\cos(\theta_{pl}) + \frac{a}{DY}) & -\frac{r}{2}(\sin(\theta_{pl}) - \frac{DX}{DY}\cos(\theta_{pl}) - \frac{a}{DY}) & a \\ \frac{r}{2}(\cos(\theta_{pl}) + \frac{DX}{DY}\sin(\theta_{pl}) - \frac{b}{DY}) & \frac{r}{2}(\cos(\theta_{pl}) - \frac{DX}{DY}\sin(\theta_{pl}) + \frac{b}{DY}) & -b \\ -\frac{r}{2DY} & \frac{r}{2DY} & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (15)$$

Fig. 11. An example system Jacobian matrix.



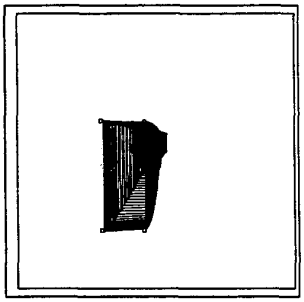
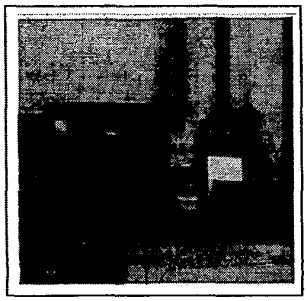


Fig. 16. Go to an object: A) initial position. B) Final position.



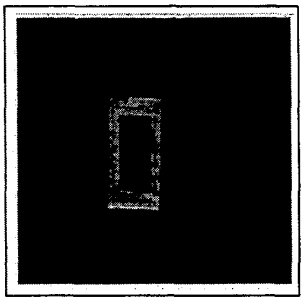


Fig. 17. A) First image. B) Final image.

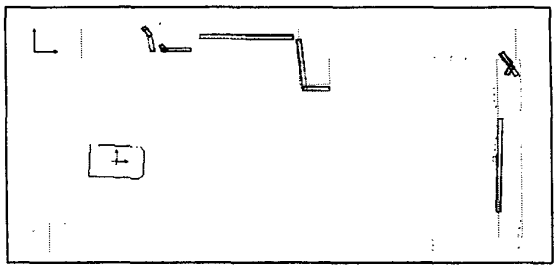


Fig. 18. Environment given by the SICK

Fig. 19. Reel-time obstacle avoidance.

illustrated in Figure 18. When the robot is far from the target, the visual tracker runs at approximately 4Hz (with a window size of 50x90 pixels, see Figure 17A) but when the robot is closer to the target, the frequency of the tracker slows to as low as 2Hz (with the image size of 240x460 pixels, see Figure 17B). Figure 19 shows the execution of the visual servoing task with obstacle avoidance in our robot Hilare2Bis.

VI. Conclusions

In this paper, we have presented a new multi-sensor approach to navigation in cluttered environments. We began by presenting a new gain scheduling visual servo control system. We developed an obstacle avoidance mechanism, and demonstrated how obstacle avoiding motions could be compensated by the vision system. We then combined these two systems, yielding a system that could avoid obstacles while performing visual servo tasks. Finally, we presented both simulations and real experiments using Hilare2Bis.

In our future work, we plan to improve our tracking method; the integration of a prediction step using Kalman filtering will allow us to intelligently modify the size of the search window making tracking process faster and more robust. We also plan integrate the selection of natural landmarks as we proposed in [19], where the landmarks will determine the task class that should be performed.

Acknowledgments

The authors would like to acknowledge the suggestions and wish to thanks M. Khatib, H. Haddad, S. Qutub for their contribution on the use of potential formalism, M. Herrb and S. Fleury for the technical assistance during the experiments. This material is based in part upon work supported by the National Science Foundation under Award Nos. CCR-0085917 and IIS-0083275.

REFERENCES

[1] R. Alami, R. Chatila, S. Fleury, M. Ghallab and F. Ingrand. An architecture for autonomy. In *the International Journal of Robotics Research (IJRR)*, Vol. 17 Num 4, April 1998.

[2] V. Ayala-Ramirez and R. Swain-Oropeza and M. Devy. Seguimiento de objetos por un robot móvil. In *Proceedings of the Encuentro de Computación - Taller de Visión por Computadora (ENC' 99)*, Pachuca, México, September 1999.

[3] F. Bensalah. Estimation du mouvement par vision active. PhDThesis *Université de Rennes I*, Rennes, France, July, 1996.

[4] B. Bouilly and T. Siméon. A Sensor-Based Motion Planner for a Mobile Robot Navigation with Uncertainty. In *Reasoning with Uncertainty in Robotics*, Ed. L. Dorst and M.V. Lambalgen and F. Voorbraak, Springer, p. 235-247, 1995.

[5] H. Bulata and M. Devy. Incremental Construction of a Landmark-based and Topological Model of Indoor Environments by a Mobile Robot. In *the Int. Conf. on Robotics & Automation (ICRA)*, Minneapolis, USA, April 1996

[6] V. Cadenat, R. Swain, P. Souères and M. Devy. A controller to perform a visually guided tracking task in a cluttered environment. In *the Int. Conf. on Intelligent Robots & Systems (IROS)*, Kyungju, Korea, October 1999.

[7] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. The Confluence of Vision and Control. In *Lecture Notes in Control and Informations Systems* G. Hager, D. Kriegman, A. Morse - Springer Verlang, 1998.

[8] F. Chaumette and E. Marchand. A new redundancy-based iterative scheme for avoiding joint limits - Applications to visual servoing. *Proc. of the IEEE Int'l. Conf. on Robotics and Automation*, San Francisco, USA, April, 2000.

[9] D. Coombs et al. Real-time obstacle avoidance using central flow divergence and peripheral flow. In *the Int. Conf. in Computer Vision (ICCV)*, Cambridge, USA, June 1995.

[10] P.I. Corke and S. A. Hutchinson, A New Partitioned Approach to Image-Based Visual Servo Control, In *Proc. on 31st Int. Symposium on Robotics*, Montreal, May 2000.

[11] B. Espiau, F. Chaumette and P. Rives. A new approach to visual servoing in robotics. In *IEEE Transactions on Robotics & Automation*, Vol. 8 (Num. 3), June 1992.

[12] O. Faugeras. Three-Dimensional Computer Vision: A Geometric Viewpoint. *The MIT press*, Cambridge, Massachusetts, USA, 1993.

[13] S. Fleury and M. Herrb and R. Chatila. GenoM: a tool for the specification and the implementation of operating modules in a distributed robot architecture, In *the Int. Conf. on Intelligent Robots & Systems (IROS)*, Grenoble, France, Septemberr 1997.

[14] H. Haddad, M. Khatib, S. Lacroix and R. Chatila. Reactive navigation in outdoor environments using potential fields. In *the Int. Conf. on Robotics & Automation (ICRA)*, Leuven, Belgium, May 1998.

[15] S. Hutchinson, G.D. Hager and P.I. Corke. A tutorial on visual servo control. In *IEEE Transactions on Robotics & Automation*, Vol. 12 (Num. 5):651–670, October 1996.

[16] F. Huynh. Manipulation assistée par la vision pour des tâches de surveillance et de saisie d'un objet fixe ou mobile. PhDThesis *UPS, LAAS CNRS*, Toulouse, France, February, 1998.

[17] M. Khatib and R. Chatila. An extended potential field approach for mobile robot sensor-based motions. In *the Int. Conf. on Intelligent Autonomous Systems (IAS)*, Karlsruhe, Germany, March 1995.

[18] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *the Int. Journal of Robotics Research (IJRR)*, Vol. 5 (Num. 1), January 1986.

[19] M. Knapek, R. Swain-Oropeza and D.J. Kriegman. Selecting Promising Landmarks. In *the Int. Conf. on Robotics & Automation (ICRA)*, San Francisco, USA, April 2000.

[20] J. Kosecka and R. Bajcsy and M. Mintz. Control of Visually Guided Behaviors. In *Real-time Computer Vision*, Cambridge Press, Ed. Christopher Brown and Demetri Terzoupoulos, 1994.

[21] J. Kosecka. Visually Guided Navigation. In *the Symposium Int. of Robotics Systems (SIRS)*, Lisbon, Portugal, July 1996.

[22] J. Kosecka "A Framework for Modeling and Verifying Visually Guided Agents: Design, Analysis and Experiments. PhD Thesis, GRASP Laboratory, University of Pennsylvania, Philadelphia, USA, March 1996.

[23] J.P. Laumond et al. A Motion Planner for Nonholonomic Mobile Robots. In *IEEE Transactions on Robotics & Automation*, Vol. 10 (Num. 5), October 1986.

[24] L.M. Lorigo and R.A. Brooks and W.E.L. Grimson. Visually-Guided Obstacle Avoidance in Unstructured Environments. In *the Int. Conf. on Intelligent Robots & Systems (IROS)*, Grenoble, France, September 1997.

[25] A. Ohya and A. Kosaka and A. Kak. Vision based Navigation by a mobile robot with obstacle avoidance using single camera vision and ultrasonic sensors. In *IEEE Transactions on Robotics & Automation*, Vol.14 (Num. 6), December 1998.

[26] R. Pissard-Gibollet and P. Rives. Applying Visual Servoing Techniques to Control a Mobile Hand-Eye System. In *the Int. Conf. on Robotics & Automation (ICRA)*, Nagoya, Japon, May 1995.

[27] R. Pissard-Gibollet et al. Real-Time Programming of Mobile Robot Actions Using Advanced Control Techniques. In *the Int. Symposium on Experimental Robotics (ISER)*, Stanford, California, USA, June 1995.

[28] P. Rives and R. Pissard-Gibollet. Reactive Mobile Robots Based on a Visual Servoing Approach. In *the 13rd Annual Conference on Artificial Intelligence (AIS)*, Perth, Western Australia, July 1992.

[29] C. Samson, M.L. Borgne and B. Espiau. *Robot Control: The Task Function Approach.* Clarendon Press, Oxford, England, 1991.

[30] J. Santos-Victor and G. Sandini. Visual-based obstacle detection a purposive approach using the normal flow. In *the Int. Conf. on Intelligent Autonomous Systems (IAS)*, Karlsruhe, Germany, March 1995.

[31] P. Souères, T. Hamel, and V. Cadenat. A path following controller for wheeled roots wich allows to avoid obstacles during transition phase. In *the Int. Conf. on Robotics & Automation (ICRA)*, Leuven, Belgium, May 1998.

[32] R. Swain-Oropeza and M. Devy. Visually-guided navigation of a mobile robot in a structured environment. In *the Symposium Int. of Robotics Systems (SIRS)*, Stockholm, Swede, July 1997.

[33] R. Swain-Oropeza and M. Devy and S. Jonquières. Navegación de un robot móvil por medio de control visual en un ambiente estructurado. In *Computación y Sistemas : Experiments with Domain Knowledge in Unsupervised Learning*, Vol. 1 (Num. 3), pages: 161-169, México, January-March 1998.

[34] R. Swain-Oropeza and M. Devy and V. Cadenat. Visual navigation & obstacle avoidance for a service robot. In *Proceedings of the International Symposium on Robotics & Automation (ISRA'98)*, Saltillo, Coahuila, México, December, 1998.

[35] R. Swain-Oropeza and M. Devy. Motion control using visual servoing and potential fields for a rover-mounted manipulator. In *the Int. Conf. on Robotics & Automation (ICRA)*, Detroit, Michigan, USA, May 1999.

[36] R. Swain-Oropeza, M. Devy and V. Cadenat. Controlling the Execution of a Visual Servoing Task. In *Journal of Intelligent and Robotic Systems : Theory and Applications (Incorporating Mechatronic Systems Engineering)*, Vol. 25 (Num. 4), June 1999.

[37] R. Swain-Oropeza. Contrôle de Tâches Référencées Vision pour la Navigation d'un Robot Mobile en Milieu Structuré. PhD Thesis, LAAS-CNRS, Toulouse, France, June 1999.

[38] D.P. Tsakiris, K. Kapellos, C. Samson, P. Rives and J.J. Borrelly. Experiments in Real-Time Vision-based Point Stabilization of a Nonholinomic Mobile Manipulator. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, Barcelona, Spain, June 1997.