Advances in Robot Vision - From Domestic Environments to Medical Applications, Kragic, D. and Christensen, H. (Eds.), Sendai, Japan, September 2004.

Features tracking for visual servoing purpose

Éric Marchand, François Chaumette
IRISA - INRIA Rennes
Campus de Beaulieu, 35042 Rennes, France
E-Mail : Firstname.Lastname@irisa.fr

***Abstract*— Elaboration of objects tracking algorithms in image sequences is an important issue for research and application related to visual servoing and more generally for robot vision. A robust extraction and real-time spatio-temporal tracking process of visual cue is indeed one of the keys to success, or to failure, of a visual servoing task. To consider visual servoing within large scale applications, it is now fundamental to consider natural scenes without any fiducial markers and with complex objects in various illumination conditions. In this paper we give an overview of a few tracking algorithms developed for visual servoing experiments at IRISA-INRIA Rennes.**

I. Motivation

Elaboration of objects tracking algorithms in image sequences is an important issue for research and application related to visual servoing and more generally for robot vision. A robust extraction and real-time spatio-temporal tracking process of visual cue is indeed one of the keys to success, or to failure, of a visual servoing task. To consider visual servoing within large scale applications, it is now fundamental to consider natural scenes without any fiducial markers and with complex objects in various illumination conditions. From a historical perspective, the use of fiducial markers allows the validation of theoretical aspects of visual servoing research. If such features are still useful to validate new control laws, it is no longer possible to limit ourselves to such techniques if the final objectives are the transfer of these technologies in the industrial world. In this paper we give an overview of a few tracking algorithms developed for visual servoing experiments at IRISA-INRIA Rennes. On Figure 1 we present some features tracking results in visual servoing experiments ordered by subjective increasing difficulties.

Overview of tracking approaches for visual servoing purpose

Most of the available tracking techniques can be divided into two main classes: feature-based and model-based tracking. The former approach focuses on tracking 2D features such as geometrical primitives (points, segments, circles, . . .) or object contours, regions of interest . . . The latter explicitly uses a model of the tracked objects. This second class of methods usually provides a more robust solution (for example, it can cope with partial occlusion of the objects). If a CAD model is available, tracking is closely related to the pose computation problem and is then suitable for any visual servoing approach. The main advantage of the model-based methods is that the knowledge about the scene (the implicit 3D information) allows improvement of robustness and performance by being able to predict hidden movement of the object and acts to reduce the effects of outlier data introduced in the tracking process. Another approach may also be considered when the scene is too complex (due, for example, to texture, to the lack of specific object, etc.). It is not based on features extraction and tracking as in the two other cases but on the analysis of the motion in the image sequence. 2D motion computation provides interesting information related to both camera motion and scene structure that can be used within a visual servoing process.

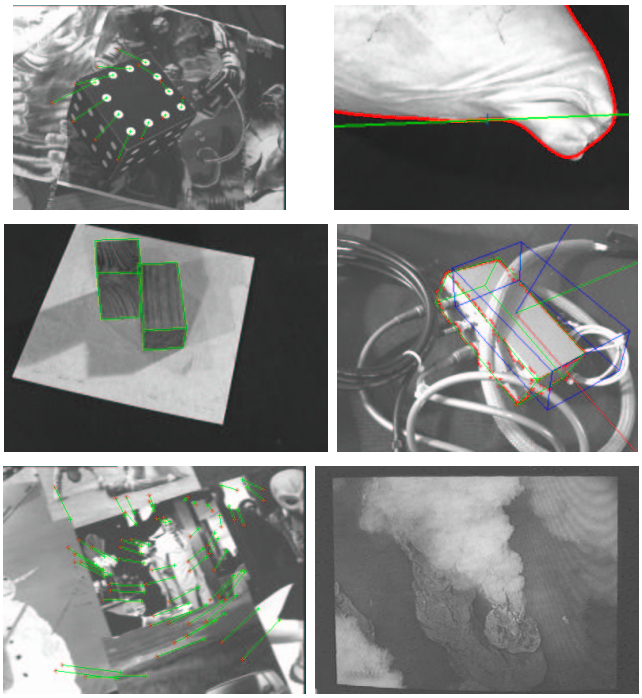

Fig. 1. Features tracking in visual servoing experiments (ordered by subjective increasing difficulties)

*a) Tracking 2*D *features:* In this approach, tracked objects are described using simple geometric features such as dots, points of interest [41], [52], angles, contours [2], [3], straight lines or segments [25], [4], [43], ellipses [57], [43], etc. This approach is the most common within the visual servoing context since there often exists a direct relationship between the 2D measures and the visual features used in the visual servoing control law.

XVision [25] is a good example of such system. Simple features (lines, contours, etc) are tracked in real-time (i.e., at video rate) ; to achieve this goal, some edge

points are matched over frames using a simple search along the normal of the contour. The tracking of more complex object is also possible if multiple elementary features are combined together and linked through a set of constraints. Our own software, ViSP [43], also features such capabilities. The tracking of elementary geometrical features (segments, ellipses, splines,...) is considered and is based on the moving edges algorithm [5] (see Section II-B). The main advantages of this class of approaches is that they are very simple and therefore very fast which is an important issue in the visual servoing context. On the other side, they do not allow the tracking of complex features that cannot be modeled trough a set of simple 2D features. Furthermore, the quality of the results (precision and robustness) cannot always be guaranteed and depends on the scene (for example, it is very dependent on the features density and on the occlusion phenomenons).

b) Regions tracking and motion estimation: Previous approaches rely mainly on the analysis of intensity gradients in the images. Another possibility is to directly consider the image intensity and to perform 2D matching on a part of the image without any feature extraction. The goal of such algorithms is to estimate a set of parameters that describes the transformation or the displacement of the considered area (a part of the image or in some cases the whole image) by minimizing a given correlation criterion (e.g., [29]). An exhaustive search of the transformation that minimizes the given criterion is not efficient. Furthermore, it is possible to solve the problem using efficient minimization techniques that allows to consider quite complex 2D transformation (such as affine or homographic motions). An approach that features such capabilities has been proposed in [24]. It allows to consider the variation of the parameters of a motion model as a linear function of an image of intensity differences. As in visual servoing, Hager defines an interaction matrix that links the variation of the motion parameters to the variation of the image intensity. An extension of these approaches has been proposed in [33] where the pseudo-inverse of the interaction is learned. These two methods are closely related to classical image motion estimation algorithms. In work carried out at IRISA, we use the approach proposed in [50]. We will see that this method is perfectly suitable for dynamic visual servoing approaches.

c) Model-based tracking: In order to handle any object or camera motions and to introduce important constraints in the spatio-temporal matching process, it is interesting to consider a model of the tracked object. If the considered model is a 2D model, it is necessary, in the general case, to augment the model with 2D local deformations [32], [23] in order to cope with the non-linear deformations of the projection of the object in the image plane due to perspective effects non handled by these 2D models. There exists a large set of deformable models more or less complex. Some had low constrained structure such as active contours [2], [34], [3], while other considered deformable templates [49], [13], [35]. However, when adding local deformations, we cannot ensure global 3D rigidity constraints which is not suitable for visual servoing purpose. Moreover, this is usually highly time consuming approaches. It is also possible to consider 3D model of the object. Tracking is then usually handled as a monocular 3D localization or pose computation issue [19], [39], [22], [36], [17], [55], [20], [56], [6], [37], [46], [9]. These approaches allow to handle any camera or rigid object motion and to consider implicitly the 3D global rigidity constraint. Furthermore they allow to handle partial occlusions of the objects. Finally, these approaches are suitable for any visual servoing control law (2D, 2 1/2 D and 3D).

The remainder of this paper is organized as follow: in a first part, we recall basic features extraction and tracking algorithms that are classically considered in visual servoing. In a second part, model-based algorithms will be presented for the tracking of 3D objects. Finally we will show how dominant image motion estimation can be used in visual servoing. In all cases we describe experimental results obtained on our experimental cells.

II. 2D TRACKING

A. Fiducial markers: past, present and future...

Most of papers related to visual servoing consider very basic image processing algorithm. Indeed the basic features considered in the control law are usually 2D points coordinates. Therefore, the corresponding object is usually composed of “white dots on a black background”. Such a choice allows using various *ad hoc* real-time algorithms. The main advantage of this choice is that tracking is very robust and very precise. It is then suitable for all visual servoing control laws (2D but also 2 1/2 D and 3D since the position between camera and target can easily be obtained using pose computation algorithm).

From a practical point of view, such algorithms are still useful to validate theoretical aspects of visual servoing research. This tracking approach has been widely used at Irisa to validate modeling aspects and designing new control laws [21], [42], [48], [45]. Furthermore in some critical industrial processes such simple approach ensures the required robustness. In such a way, it has been considered in the development of grasping tasks in nuclear environment for *Électricité de France*.

B. Tracking contour-based 2D features

In order to address the problem of 2D geometric feature tracking it is necessary to consider at the low level a generic framework that allows local tracking of edge points. From the set of tracked edges, it is then possible to perform a robust estimation of the features parameters using an Iteratively Reweighted Least Squares (IRLS, see appendix A) based on robust M-estimation.

For the first point, few systems feature real-time capabilities on a simple workstation. The XVision system [25] is a nice example of such systems. However, it does not feature all the tracking capabilities we wanted. In our case, we decided to use the Moving Edges (ME) algorithm [5] which is adapted to the tracking of parametric curves. It is

a local approach that allows to match moving contours. Primary works done to use this algorithm to track line segments [4] has been achieved on a dedicated IP board. Now it runs at video rate on a classical PC.

1) ME algorithm: When dealing with low-level image processing the contours are sampled at a regular distance. At these sample points a 1 dimensional search is performed to the normal of the contour for corresponding edges. An *oriented* gradient mask [5] is used to detect the presence of a contour. One of the advantages of this method is that it only searches for edges which are aligned in the same direction as the parent contour. An array of 180 masks is generated off-line which is indexed according to the contour angle. This is therefore implemented with convolution efficiency, and leads to real-time performance.

When referring to Figure 2, the process consists of searching for the corresponding point p^{t+1} in image I^{t+1} for each point p^t. A 1D search interval $\{Q_j, j \in [-J, J]\}$ is determined in the direction δ of the normal to the contour. For each position Q_j lying in the direction δ, a mask convolution M_δ corresponding to the square root of a log-likelihood ratio ζ_j is computed. Thus the new position p^{t+1} is given by:

$$Q^{j^*} = \arg \max_{j \in [-J,J]} \zeta_j$$

with

$$\zeta_j = \mid I^{t+1}_{\nu(Q_j)} * M_\delta + I^t_{\nu(p^t)} * M_\delta \mid$$

$\nu(.)$ is the neighborhood of the considered pixel. In our implementation of the algorithm, the neighborhood is limited to a 7×7 pixel mask. It should be noted that there is a trade-off to be made between real-time performance and mask stability. Likewise there is a trade-off to be made between the search distance, and real-time performance while considering the maximum inter-frame movement of the object.

This low level search produces a list of k points which are used to compute the parameters of the tracked features.

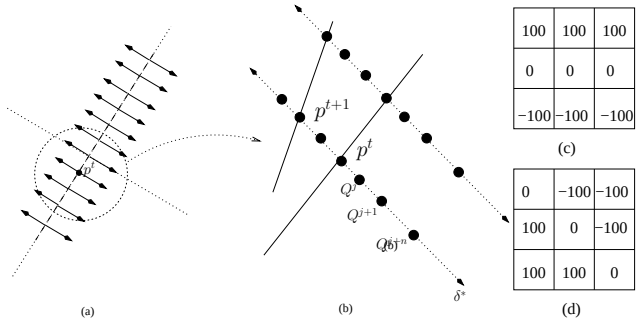

Fig. 2. Determining points position in the next image using the oriented gradient algorithm: (a) calculating the normal at sample points, (b) sampling along the normal (c-d) 2 out of 180 3x3 predetermined masks (in practice 7x7 masks are used) (c) 180^o (d) 45^o.

2) Tracking visual cues: Line segments. The simplest case we considered is the line segment [4]. The representation considered for the straight line are the polar coordinates (ρ, θ) such that:

$$x \cos\theta + y \sin\theta - \rho = 0, \forall (x, y) \in \mathcal{D}$$

This case is very simple as the direction θ is directly given by the parameters of the features. The choice of the convolution mask is then straightforward. A points insertion process either in the middle of the segment, to deal with partial occlusions or miss-tracking, and at the extremities of the segment to deal with sliding movements has been also introduced in the tracking method.

Ellipses. Dealing with an ellipse, many representations can be used, we choose to consider the coefficients K_i that are obtained from the polynomial equation of an ellipse:

$$x^2 + K_1 y^2 + 2K_2 xy + 2K_3 x + 2K_4 y + K_5 = 0$$

The ellipse correspond to the case $K_2^2 < K_1$. The parameters K_i can be estimated from the list of tracked points using a least square method or an IRLS method. From the parameters K_i, it is of course possible to derive the representation of any other representation parameters such as for instance $(x_c, y_x, \mu_{11}, \mu_{02}, \mu_{20})$ based on the moments.

Splines. A spline is defined by a parametric equation:

$$Q(t) = \sum_{j=-d}^{n-1} \alpha_j B_j(t), \quad t \in [0, 1]$$

where the α_j are the control points of the spline, d is the degree of the spline ($d = 3$ for a cubic spline) and B_j is the spline basis function. Since the number p of tracked points is usually greater than the number $n + d$ of desired control points, a least square or an IRLS can also be used.

3) Results: Figure 3 shows several results of features tracking (line, circle, contours,...) in visual servoing experiments. The proposed tracking approach based on the ME algorithm allows a real-time tracking of geometric features in an image sequence. It is robust with respect to partial occlusions and shadows. However, as a local algorithm, its robustness is limited in complex scenes with highly textured environment.

C. Tracking point of interests

When the objective is to track points of interest, it is necessary to make some conservation assumptions on some information related to the points. These hypotheses may concern the point motion, or a photometric/geometric invariance in a neighborhood of the point.

The usual assumption of luminance pattern conservation along a trajectory has led to build two kinds of methods. The first ones are intuitive methods based on correlation. The second ones are defined as differential trackers, built on a differential formulation of a similarity criterion. In particular, the well-known Shi-Tomasi-Kanade tracker [52] belongs to this latter class.

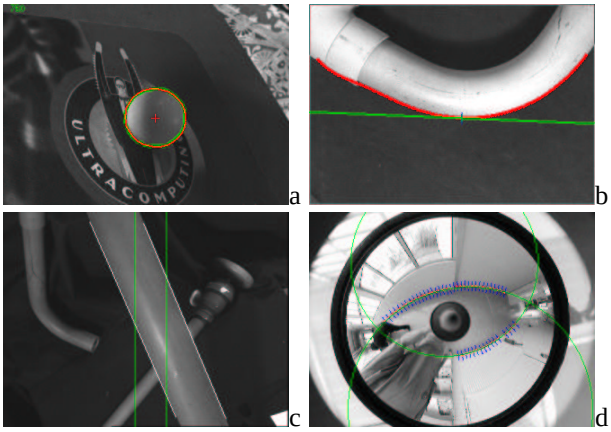

Fig. 3. Tracking 2D features using the Moving Edges algorithm within visual servoing experiments (a) 3D reconstruction of a sphere using active vision [7], (b) contour following [43], (c) positioning wrt. a cylinder with joint limits avoidance [8], (d) ellipses tracking (that correspond to the projection of 3D straight lines in catadioptric images).

1) Shi-Tomasi tracker: Consider an image sequence $I(\mathbf{x}, t)$ where $\mathbf{x} = (x, y)$ are the coordinates of an image point. If the baseline between two successive camera locations is small, it can be assumed that, though small image regions are displaced their intensities are unchanged. From which the classical equation is deduced: $I(\boldsymbol{x} + \dot{\boldsymbol{x}}_\Theta(\boldsymbol{x}), t + 1) - I(\boldsymbol{x}, t) = 0$ where $\dot{\boldsymbol{x}}_\Theta(\boldsymbol{x})$ is the motion field specifying the point displacement according to a motion model Θ.

The task of the tracker is then to compute the parameters of the motion model of selected points for each pair of successive frames. The problem is then to find the displacement $\dot{\boldsymbol{x}}_\Theta(\boldsymbol{x})$ (and then the parameters Θ of the motion model) which minimize the following residual:

$$\varepsilon = \sum_{\boldsymbol{x} \in \mathcal{W}} \left(I(\boldsymbol{x} + \dot{\boldsymbol{x}}_\Theta(\boldsymbol{x}), t + 1) - I(\boldsymbol{x}, t) \right)^2 \qquad (1)$$

where $\mathcal{W}$ is a small image region centered on the point to be tracked. To obtain a linear least-squares estimate of $\dot{\boldsymbol{x}}_\Theta(\boldsymbol{x})$. The derivative of the residual wrt. Θ are set to zero (see [54] for a detailed derivation when in the case of a translation and of an affine motion model).

As explained in [52], some locations in the initial template are more interesting than the others: those where the singular values of $\mathbf{R}$ are high, with :

$$\mathbf{R} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \qquad (2)$$

Indeed, they represent corners and similar entities, their high spatial gradient gives robust information about the 2D motion whereas a point located in an uniform area does not allow to detect the displacement.

2) Visual servoing based on point of interest: In the presented experiment, the position to reach is defined by a reference image (see Figure 4b). Points of interest are extracted (using the Harris detector [27]) and are matched with similar points extracted from the image acquired from the initial camera location. This matching process is done using the Image-Matching software [58] and is based on robust estimation of the fundamental matrix. Tracking during the visual servoing experiment is based on the Shi-Tomasi algorithm [52]. It appears that the feature tracking is not very reliable and poor images quality induces important errors into a classical visual servoing approach.

More precisely with the use of a classical control law and due to excessive miss-tracking, the camera was not able to reach the desired position. Therefore in the presented results, we have used a robust control law [45] that allows rejection of miss-tracked points. In Figure 4c red crosses are the initial points location, blue crosses are their desired locations while the green crosses are the final points location. Point trajectories are in red when points are correctly tracked and in blue when the appears to be miss-tracked (60 points are tracked).

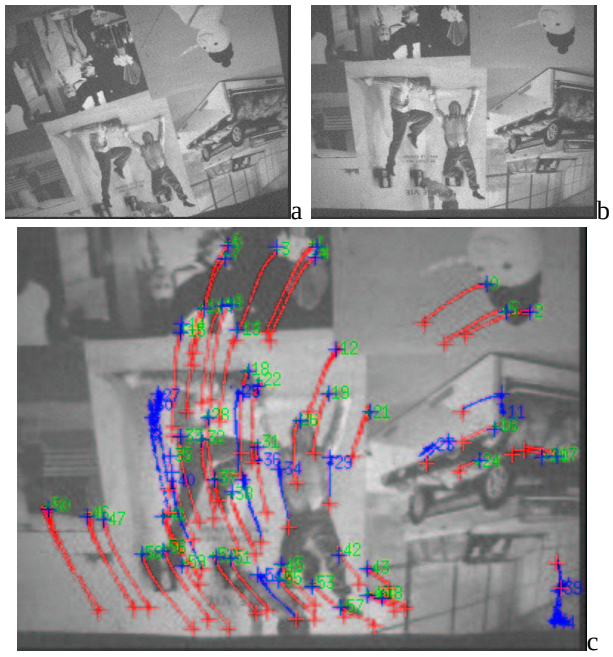

Fig. 4. Visual servoing based on the tracking of points of interest: (a) initial image, (b) desired image, (c) robust visual servoing.

D. Multimodal visual tracking: appearance and contours

Dealing with 2D tracking, numerous algorithms rely on the texture of the tracked object. This is the case for appearance or template-based tracking [24], [33], for point of interest [52] (Section II-C) or dominant motion estimation [50] (see Section IV). These techniques have proved to be efficient to track textured object. On the other hand, contour-based tracking approaches (section II-B, III) are more relevant for textureless objects and are very efficient when sharp edges are visible in the image sequence. In both cases the tracking algorithm rely on the optimization of an objective function. In order to develop algorithms robust to aberrant measurements and potential occlusions, it appears to be interesting to take into account visual information related to these different types.

Therefore, an alternative direction is to consider both information within the same minimization problem [47], [51]. The idea is to consider both motion or appearance

and edges. Contours of the object as well as its motion can indeed be modeled by parametric models whose estimation is possible. The multimodal tracking algorithm proposed in [51] fuses the motion estimation of contour points and of texture points motion estimation in an unique non-linear minimization process. Figure 5 shows preliminary results of the tracking (within a visual servoing experiment) of a rice box in a highly textured environment. Let us note that trackers that rely only on texture or on contour fail to track correctly the box over a long image sequence.

III. Model-Based Tracking: Robust virtual visual servoing

In this section, we now assume that the 3D CAD model of the tracked is available. We focus on the registration techniques that allow alignment of 2D features extracted from the image (in real-time by a moving camera) and the model of the object. In the related computer vision literature, geometric primitives considered for this pose estimation problem are often points [26], [18], [40], contours or points on the contours [39], [9], [20], segments, straight lines, conics, cylindrical objects, or a combination of these different features [44]. Another important issue is the registration method. *Purely geometric* (eg, [19]), or *numerical and iterative* [18] approaches may be considered. *Linear approaches* use a least-squares method to estimate the pose. *Full-scale non-linear optimization techniques* (e.g., [39], [20]) consists of minimizing the error between the observation and the forward-projection of the model. In this case, minimization is handled using numerical iterative algorithms such as Newton-Raphson or Levenberg-Marquardt. The main advantage of these approaches are their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence.

In our work, pose computation is formulated in terms of a full scale non-linear optimization: Virtual Visual Servoing (VVS). In this way the pose computation problem is considered as similar to 2D visual servoing as proposed in [53], [44]. This method is aligned with state of the art methods treating this issue [20], [38]. Essentially, 2D visual servoing [31], [21], [28] consists of specifying a task (mainly positioning or target tracking tasks) as the regulation in the image of a set of visual features. A closed-loop control law that minimizes the error between the current and desired position of these visual features can then be implemented which automatically determines the motion the camera has to realize. This framework is used to create an image feature based system which is capable of treating complex scenes in real-time. Advantages of the virtual visual servoing formulation are demonstrated by considering a wide range of performance factors. Notably the accuracy, efficiency, stability, and robustness issues have been addressed and demonstrated to perform in complex scenes. A robust control law that integrates an M-estimator has been integrated to improve robustness. The resulting pose computation algorithm is thus able to deal efficiently with incorrectly tracked features that usually contribute to a compound effect which degrades the system until failure.

A. Overview and motivations

As already stated, the fundamental principle of the proposed approach is to define the pose computation problem as the dual problem of 2D visual servoing [21], [31]. In visual servoing, the goal is to move a camera in order to observe an object at a given position in the image. An explanation will now be given as to why the pose computation problem is very similar.

To illustrate the principle, consider an object composed of various 3D features $\mathbf{P}$ (for instance, we denote ${}^{o}\mathbf{P}$ the value of these features in the object frame). A virtual camera is defined whose position in the object frame is defined by $\mathbf{r}$. The approach consists of estimating the real pose by minimizing the error Δ between the observed data $\mathbf{s}^*$ (usually the position of a set of features in the image) and the position $\mathbf{s}$ of the same features computed by forward-projection according to the current pose:

$$\Delta = \big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big) = \big[pr_{\xi}(\mathbf{r},{}^{o}\,\mathbf{P}) - \mathbf{s}^*\big], \tag{3}$$

where $pr_{\xi}(\mathbf{r},{}^{o}\,\mathbf{P})$ is the projection model according to the intrinsic parameters ξ and camera pose $\mathbf{r}$. It is supposed here that intrinsic parameters ξ are available but it is possible, using the same approach to also estimate these parameters.

In this formulation of the problem, a virtual camera initially at $\mathbf{r}_i$ is moved using a visual servoing control law in order to minimize this error Δ. At convergence, the virtual camera reaches the position $\mathbf{r_d}$ which minimizes this error (and which corresponds to the real camera pose).

An important assumption is to consider that $\mathbf{s}^*$ is computed (from the image) with sufficient precision. However, when outliers are present in the measures, a robust estimation is required. M-estimators can be considered as a more general form of maximum likelihood estimators [30]. They are more general because they permit the use of different minimization functions not necessarily corresponding to normally distributed data. Many functions have been proposed in the literature which allow uncertain measures to be less likely considered and in some cases completely rejected. In other words, the objective function is modified to reduce the sensitivity to outliers. The robust optimization problem is then given by:

$$\Delta_{\mathcal{R}} = \rho\Big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\Big), \tag{4}$$

where $\rho(u)$ is a robust function [30] that grows sub-quadratically and is monotonically nondecreasing with increasing $|u|$. Iteratively Re-weighted Least Squares (IRLS) is a common method of applying the M-estimators (see appendix A). It converts the M-estimation problem into an equivalent weighted least-squares problem.

B. Robust minimisation

The objective of the optimization scheme we had proposed [9], [11] is to minimize the objective function given in equation (4). This objective is incorporated into a robust

Fig. 5. Multimodal tracking: merging contour and appearance within a single non-linear minimization process.

visual servoing control law (see [12] for more details). Thus, the error to be regulated to 0 is defined as:

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \tag{5}$$

where $\mathbf{D}$ is a diagonal weighting matrix given by $\mathbf{D} = diag(w_1, \ldots, w_k)$. The computation of weights w_i that reflects the confidence in each feature is described in Appendix A. A simple control law can then be designed to try to ensure an exponential decoupled decrease of $\mathbf{e}$ around the desired position $\mathbf{s}^*$ [12]. The control law is given by:

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}\mathbf{L}_\mathbf{s}})^+\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big), \tag{6}$$

where $\mathbf{L_s}$ is the interaction matrix related to $\mathbf{s}$.

Clearly, this approach necessitates to ensure that a sufficient number of features will not be rejected so that $\mathbf{DL_s}$ is always of full rank (6 to estimate the pose). It has been shown that only local stability can be demonstrated [12]. This means that the convergence may not be obtained if the error $\mathbf{s} - \mathbf{s}^*$ is too large. However, in tracking applications $\mathbf{s}$ and $\mathbf{r}$ are obtained from the previous image, thus the motion between two successive images acquired at video rate is sufficiently small to ensure the convergence. In practice it has been observed that the convergence is obtained, in general, when the camera displacement has an orientation error less that 30^o on each axis. Thus, potential problems only appear for the very first image where the initial value for $\mathbf{r}$ may be too coarse. In the current algorithm the initialization is done by manually clicking on the images and calculating the pose using a 4 point algorithm [18].

C. Visual features and interaction matrices

Any kind of geometrical features can be considered within the proposed control law as soon as it is possible to compute its corresponding interaction matrix $\mathbf{L_s}$. In [21], a general framework to compute $\mathbf{L_s}$ is proposed. Indeed, it is possible to compute the pose from a large set of image information (points, lines, circles, quadratics, distances, etc...) within the same framework. The combination of different features is achieved by adding features to vector $\mathbf{s}$ and by "stacking" each feature's corresponding interaction matrix into a large interaction matrix of size $nd \times 6$ where n corresponds to the number of features and d their dimension:

$$\begin{pmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_n \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{s1} \\ \vdots \\ \mathbf{L}_{sn} \end{pmatrix} \mathbf{v} \tag{7}$$

The redundancy yields a more accurate result with the computation of the pseudo-inverse of $\mathbf{L_s}$ as given in equation (6). Furthermore if the number or the nature of visual features is modified over time, the interaction matrix $\mathbf{L_s}$ and the vector error $\mathbf{s}$ is easily modified consequently.

In most of our works [10], a distance feature is considered as a set of distances between local point features obtained from a fast image processing step and the contours of a more global CAD model. In this case the desired value of the distance is equal to zero. The assumption is made that the contours of the object in the image can be described as piecewise linear segments. All distances are then treated according to their corresponding segment. The derivation of the interaction matrix that links the variation of the distance between a fixed point and a moving straight line to the virtual camera motion is given in [10].

D. Results on 3D model-based tracking

In such experiments, the image processing is potentially very complex. Indeed, extracting and tracking reliable points in real environment is a non trivial issue. In all experiments, the distances are computed using the "oriented" gradient mask algorithm described in section II-B. In the experiment presented in Figure 6, images were acquired and processed at video rate (50Hz). Tracking is always performed at below frame rate (usually in less than 10ms). All the images given in Figure 6 depict the current position of the tracked object in green while its desired position appears in blue. The considered object is a video multiplexer. It was placed in a highly cluttered environment. Tracking and positioning tasks were correctly achieved. Multiple temporary and partial occlusions were made by a hand and various work-tools. Modification of the lighting conditions were also imposed. After a positioning task achieved using a 2D 1/2 visual servoing control law, the object is handled by hand and moved around. In this case, since the visual servoing task has not been stopped, the robot continues to follow the object in order to maintain the rigid link between the camera and the object. Note that some object faces appeared while others disappeared. Other

results using this algorithm are presented in [11], and in [9] for augmented reality application.

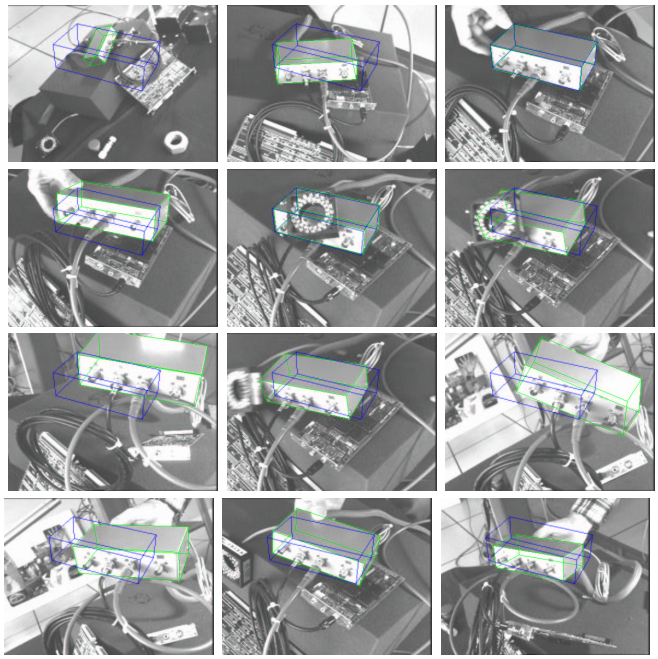

Fig. 6. 2D 1/2 visual servoing experiment: in the images the tracked object appears in green and its desired position in blue. The four first images have been acquired in initial positioning step. In the reminder images, object is moving along with the robot.

IV. Motion estimation in visual servoing

When images are to complex (textured or natural outdoor scene) retrieving geometric features has proved to be a difficult issue. Another possibility is to use motion in the image as input of the control scheme [14], [15], [16], since it can be estimated without any a priori knowledge of the observed scene. Thus, more realistic scenes or objects can be considered.

In these works the estimation of the 2D parametric motion model accounting for the dominant image motion is achieved with a robust, multi-resolution, and incremental estimation method exploiting only the spatio-temporal derivatives of the intensity function [50]. Let us note that other approaches such as those proposed in [24], [33], [1] may also be suitable to build such visual servoing systems.

A. Motion estimation

1) Quadratic motion models: In visual servoing based on image motion the goal is to control the motions of a robot from visual features without any a priori knowledge on the image content. Both methods presented in the next subsections rely on motion in the image since it is not dependent on the image content.

In the general case, a 2D image motion models cannot account for the global 3D motion of the object within the scene. A good compromise is thus to consider a 2D quadratic motion model which corresponds to the projection of the rigid motion of a planar surface. This model involves eight independent parameters. Let us denote $\Theta = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$, the velocity vector $\dot{\boldsymbol{x}}_\Theta(\boldsymbol{x})$ at pixel $\boldsymbol{x} = (x, y)$ corresponding to the quadratic motion is given by:

$$\dot{\boldsymbol{x}}_\Theta(\boldsymbol{x}) = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} + \begin{bmatrix} a_2 & a_3 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_6 & a_7 & 0 \\ 0 & a_6 & a_7 \end{bmatrix} \begin{bmatrix} x^2 \\ xy \\ y^2 \end{bmatrix} \quad (8)$$

Other model are also available (constant motion model $a_2 = \ldots = a_7 = 0$, affine motion model $a_6 = a_7 = 0$ or even model more complex with more parameters that may for example handle illumination variation). In fact, there is a necessary compromise to find between the accuracy provided by a model and the computation load, such that the control rate is the closest possible to the video rate. Indeed, the real motion in the image is generally complex, and only an approximation can be obtained using a polynomial model. In year 2000, only the parameters of the constant model can be estimated at video rate without any dedicated image processing board. Now an affine motion model can be easily computed at video rate. But due to computer increasing power, in one or two year from now, a complete model (8 parameters or more) may certainly be considered in real time)

2) Dominant image motion estimation: To estimate the dominant image motion between two successive images $I(t)$ and $I(t+1)$, the gradient-based multiresolution robust estimation method described in [50] is used. The constraint is given by the usual assumption of brightness constancy of a projected surface element over its 2D trajectory. The parameters of the motion model $\widehat{\Theta}$, that describes the dominant motion, are estimated by minimizing for Θ the difference of frame displacement. To ensure robustness to the presence of independent motion, a robust minimization of equation (1) is considered:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \sum_{\boldsymbol{x} \in I(t)} \rho\left(I(\boldsymbol{x} + \dot{\boldsymbol{x}}_\Theta(\boldsymbol{x}), t+1) - I(\boldsymbol{x}, t)\right) \quad (9)$$

$\rho(x)$ is a robust function [30]. In practice, it will be restricted to a specific area of the image. The minimization is embedded in a multi-resolution framework and follows an incremental scheme (see [50] for details).

B. Results in dynamic visual servoing

Two different methods are presented. In the first one, geometric features are retrieved by integration of motion, which allows to use classical control laws. This method is applied to a pan/tilt tracking and stabilization task. In the second method, the principle is to try to obtain a desired 2D motion field in the image sequence. This approach is illustrated with results for positioning a camera parallel to a plane.

1) visual servoing by integration of 2D motion: Motion estimation can be used to handle the classical task of mobile target tracking using a camera mounted on the end effector of a robot. In this example, we now consider as

input of the control law the estimated position of the target position in the image. The visual features are thus $\mathbf{s} = (x, y)$. They are simply obtained by successive summations of the a_0 and a_1 (see (9)):

$$s_i(k) = s_i(0) + \sum_{j=1}^{k} a_{ij}\ \delta t, \qquad i = 0, 1 \qquad (10)$$

where $\mathbf{s}_i(0) = (x_0, y_0)$ is the initial position of the target computed during a detection step and δt is the period of the control loop.

The aim of the tracking task is to control the camera pan and tilt such that the image of the mobile target is, first, brought at the image center ($\mathbf{s}^* = (0, 0)$), and then remains at this position whatever the target motions are. This task is quite simple from the control point of view. The contribution described in [15] is more concerned with the complexity of the considered targets.

A pedestrian tracking task is presented in [15] and in Figure 7. Let us point out that the estimation of 2D motion parameters with the algorithm we used involves the discarding of non-coherent local motions considered as outliers. Therefore, motions related to deformations of non-rigid objects (such as a human being) do not affect greatly the estimation of the dominant motion. Figure 7 contains one image over 10 of the sequence acquired during the tracking. Motion of the person is first sideways, and not always facing the camera. Then, the pedestrian comes back toward the camera. On each image, the estimated position is represented by a black cross (+) and the image center by a black square ($\diamond$). Despite the complexity of motion, the pedestrian always appears at the image center. This demonstrates the robustness of the motion estimation algorithm and of the control scheme. Finally, when another person crosses the tracked one. In spite of this perturbing supplementary motion, the camera is still fixating at the selected person.

This algorithm as been used in [14] for image stabilization of a camera mounted on an underwater engine. In that case, the motion in the image is, in part, due to the potential scene own motion and overall, to the undesirable motion of the engine, because of underwater currents. Even if the quality of the images used is poor (they had low spatio-temporal gradients), the results presented in [14] show that the drift in the image remains very weak (less than half a pixel after 250 iterations). A typical image sequence acquired during the stabilization is given in Figure 9 where the considered scene is a rock from which smoke and gas escape.

2) Camera positioning: In a second approach, the visual specification of the desired configuration is no more done with geometrical constraints, but with dynamic criteria, i.e. homogeneous to speed in the image. More precisely, we wish to control the camera motions in order that the current motion field in the image, such as the one presented in Figure 10.a, becomes equal to a desired one, such as, for example, the divergent field of Figure 10.b.

More precisely, visual features are selected from the

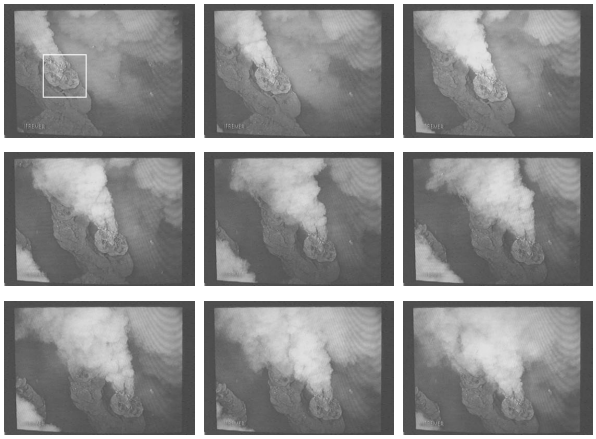

Fig. 8. "Rocks" sequence. One full image over 25 from the original sequence (one image every 2 s) with a non controlled camera

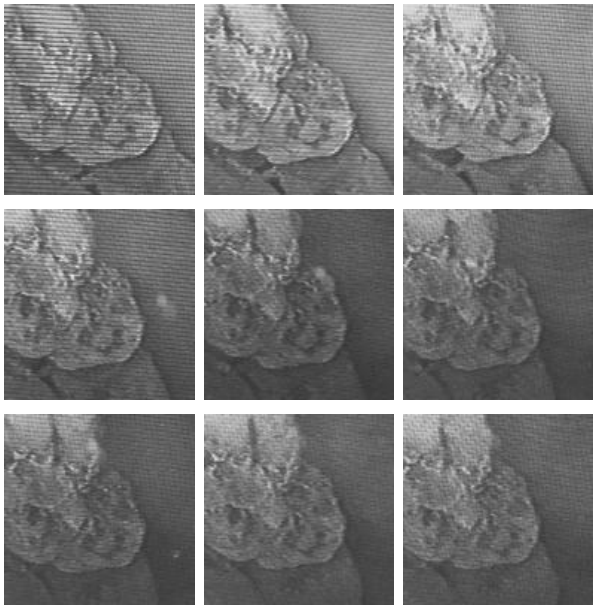

Fig. 9. One image over 25 of the "rocks" sequence acquired during stabilization

parameters of a polynomial motion model. Numerous tasks can be defined using such kind of dynamic visual features, some of them being impossible to perform using geometric visual features. They can be divided into three groups depending on the aim of the considered task [16]. In usual image-based visual servoing, variations of visual features are linearly linked to the camera velocity. In that case, the corresponding relation is more complex.

This approach is illustrated with a positioning task: the camera has to be parallel to a plane. The choice of feature vector $\mathbf{s}$ using the motion parameters $\widehat{\Theta}$ as well as the design of the control law is given in [16]. The images of the scene at the initial and final positions are respectively presented on Figure 11.a and Figure 11.b. One can notice that the planar object does not cover the totality of the field of view at the beginning. Moreover, non-planar objects are displayed on the plane. Nevertheless, due to the robustness of the motion 2D estimation, the task is correctly realized.

Fig. 7. Tracking of a pedestrian. An image upon ten of the acquired sequence (approximately 2 frames per second). Cross (+) stands for the estimated c.o.g. and diamond ($\diamond$) for the image center

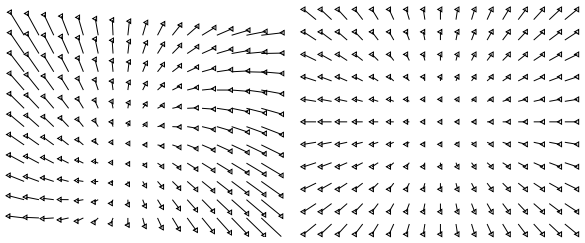

Fig. 10. Positioning the camera parallel to a plane : current (a) and desired (b) 2D motion field

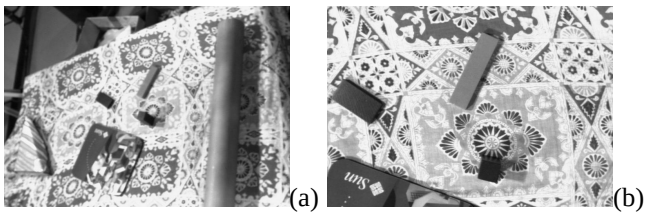

Fig. 11. Positioning the camera parallel to a plane : (a) initial image and (b) final image

APPENDIX

A. Robust estimation (IRLS)

In most of the algorithms presented in this paper, we have considered a parameters estimation process (for features extraction, pose estimation, and dominant image motion estimation). To handle properly the estimation in presence of noise and of corrupted data a robust estimation has to be performed. Therefore, we briefly recall in this appendix the Iteratively Reweighted Least Squares (IRLS) algorithm.

1) IRLS algorithm: The actual least squares problems aims at solving for $\boldsymbol{x}$ the following linear system $\mathbf{A}\boldsymbol{x} = \mathbf{b}$ where $\boldsymbol{x}$ and $\mathbf{b}$ are vectors and $\mathbf{A}$ is a matrix. However in each case since some inputs are likely to be outliers, it is necessary to handle a robust estimation of these parameters.

Iteratively Reweighted least squares algorithm aims at solving the following system $\mathbf{DA}\boldsymbol{x} = \mathbf{Db}$ where $\mathbf{D} = diag(w_1, \ldots, w_n)$ is a diagonal matrix where w_i reflect the confidence of each feature. The algorithm acts as follows: estimate weights using one of the many robust criterion suggested in the literature (see next paragraph for example), estimate the value of $\boldsymbol{x}x$ by solving the weighted system, and reiterate until convergence. These methods act like automatic outlier rejectors since large residual values lead to very small weights.

2) Confidence computation using M-estimation: This section gives a brief overview for the calculation of confidence we have in each image feature. The weights w_i, which reflect the confidence of each feature, are usually given by [30]:

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}, \tag{11}$$

where $\psi(u) = \frac{\partial \rho(u)}{\partial \boldsymbol{x}}$ (ψ is the influence function) and δ_i is the normalized residue given by $\delta_i = \Delta_i - Med(\Delta)$ (where $Med(\Delta)$ is the median operator).

Of the various influence functions that exist in the literature Tukey's hard re-descending function is considered. Tukey's function completely rejects outliers and gives them a zero weight. This is of interest in tracking applications so that a detected outlier has no effect on the estimation scheme. This influence function is given by:

$$\psi(u) = \begin{cases} u(C^2 - u^2)^2 & , \text{ if } |u| \leq C \\ 0 & , \text{else}, \end{cases} \quad (12)$$

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

In order to obtain a robust objective function, a value describing the certainty of the measures is required. The scale σ that is the estimated standard deviation of the inlier data, is an important value for the efficiency of the method. In non-linear regression, this estimate of the scale can vary dramatically during convergence. Scale may be manually chosen as a tuning variable or may be estimated online. One robust statistic used to estimate scale is the Median Absolute Deviation (MAD), given by:

$$\widehat{\sigma} = \frac{1}{\Phi^{-1}(0.75)} Med_i(|\delta_i - Med_j(\delta_j)|). \quad (13)$$

where $\Phi()$ is the cumulative normal distribution function and $\frac{1}{\Phi^{-1}(0.75)} = 1.48$ represents one standard deviation of the normal distribution.

ACKNOWLEDGMENT

The authors want to acknowledge the work of Armel Crétual dealing with dynamic visual servoing, of Andrew Comport dealing with robust virtual visual servoing, and of Muriel Pressigout dealing with multimodal tracking.

REFERENCES

[1] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJ International Conference on Intelligent Robots Systems*, Sendai, Japan, October 2004.

[2] M.-O. Berger. How to track efficiently piecewise curved contours with a view to reconstructing 3D objects. In *Int. Conf on Pattern Recognition, ICPR'94*, pages 32–36, Jerusalem, October 1994.

[3] A. Blake and M. Isard. *Active Contours*. Springer Verlag, April 1998.

[4] S. Boukir, P. Bouthemy, F. Chaumette, and D. Juvin. A local method for contour matching and its parallel implementation. *Machine Vision and Application*, 10(5/6):321–330, April 1998.

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

[6] P. Braud, M. Dhome, J.-T. Lapresté, and B. Peuchot. Reconnaissance, localisation et suivi d'objets polyhédriques par vision multioculaire. *Technique et Science Informatiques*, 16(1):9–38, Janvier 1997.

[7] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin. Structure from controlled motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):492–504, May 1996.

[8] F. Chaumette and E. Marchand. A redundancy-based iterative scheme for avoiding joint limits: Application to visual servoing. *IEEE Trans. on Robotics and Automation*, 17(5):719–730, Octobre 2001.

[9] A. Comport, E. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pages 36–45, Tokyo, Japan, October 2003.

[10] A. Comport, E. Marchand, and F. Chaumette. Complex articulated object tracking. In *Int. Workshop on articulated motion and deformable objects, AMDO'04, LNCS*, Palma de Mallorca, Spain, septembre 2004.

[11] A. Comport, E. Marchand, and F. Chaumette. Robust model-based tracking for robot vision. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, Sendai, Japan, september 2004.

[12] A. Comport, M. Pressigout, E. Marchand, and F. Chaumette. A visual servoing control law that is robust to image outliers. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'03*, volume 1, pages 492–497, Las Vegas, Nevada, October 2003.

[13] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models - their training and application. *CVGIP : Image Understanding*, 61(1):38–59, Janvier 1994.

[14] A. Crétual and F. Chaumette. Dynamic stabilization of a pan and tilt camera for sub-marine image visualization. *Computer Vision and Image Understanding*, 79(1):47–65, July 2000.

[15] A. Crétual and F. Chaumette. Application of motion-based visual servoing to target tracking. *Int. Journal of Robotics Research*, 20(11):878–890, November 2001.

[16] A. Crétual and F. Chaumette. Visual servoing based on image motion. *Int. Journal of Robotics Research*, 20(11):857–877, November 2001.

[17] N. Daucher, M. Dhome, J.T. Lapreste, and G. Rives. Modelled object pose estimation and tracking by monocular vision. In *British Machine Vision Conference, BMVC'93*, pages 249–258, Guildford, UK, September 1993.

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

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

[20] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(7):932–946, July 2002.

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

[22] D.B. Gennery. Visual tracking of known three-dimensional objects. *Int. J. of Computer Vision*, 7(3):243–270, 1992.

[23] N. Giordana, P. Bouthemy, F. Chaumette, F. Spindler, J.-C. Bordas, and V. Just. Two dimensional model-based tracking of complex shapes for visual servoing tasks. In M. Vincze and G. Hager, editors, *Robust vision for vision-based control of motion*, chapter 6, pages 67–75. IEEE Press, 2000.

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

[25] G. Hager and K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, January 1998.

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

[27] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Conference*, pages 189–192, 1988.

[28] 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.

[29] B. Horn. *Robot Vision*. MIT Press, Cambridge, 1987.

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

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

[32] A.K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 18(3):267–278, March 1996.

[33] F. Jurie and M. Dhome. Hyperplane approximation for template matching. *IEEE trans on Pattern Analysis and Machine Intelligence*, 24(7):996–1000, July 2002.

[34] M. Kass, A. Witkin, and D. Terzopolous. Snakes : Active contour models. In *Int. Conf. Computer Vision, ICCV'87*, pages 259–268, London, UK, 1987.

[35] C. Kervrann and F. Heitz. A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60(3):173–195, May 1998.

[36] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *Int. Journal of Computer Vision*, 10(2):257–281, June 1993.

[37] D. Kragic and H.I. Christensen. Model based techniques for robotic servoing and grasping. In *IEEE Int. Conf. on intelligent robots and systems, IROS'02*, volume 1, pages 299–304, Lausanne, Switzerland, October 2002.

[38] D. Kragic and H.I. Christensen. Confluence of parameters in model based tracking. In *IEEE Int. Conf. on Robotics and Automation, ICRA'03*, volume 4, pages 3485–3490, Taipe, Taiwan, September 2003.

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

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

[41] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conf. on Artificial Intelligence, IJCAI'81*, pages 674–679, 1981.

[42] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 D visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, April 1999.

[43] E. Marchand. Visp: A software environment for eye-in-hand visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'99*, volume 4, pages 3224–3229, Detroit, Michigan, Mai 1999.

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

[45] E. Marchand, A. Comport, and F. Chaumette. Improvements in robust 2D visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, volume 1, pages 745–750, New Orleans, April 2004.

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

[47] L. Masson, F. Jurie, and M. Dhome. Contour/texture approach for visual tracking. In *13th Scandinavian Conference on Image Analysis, SCIA 2003*, volume 2749 of *Lecture Notes in Computer Science*, pages 661–668. Springer, 2003.

[48] Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549, August 2002.

[49] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In *Int. Conf. on Computer Vision, ICCV'93*, pages 275–279, Berlin, Allemagne, 1993.

[50] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.

[51] M. Pressigout and Marchand E. Multimodal tracking for visual servoing. Internal Note, September 2004.

[52] J. Shi and C. Tomasi. Good features to track. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pages 593–600, Seattle, Washington, June 1994.

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

[54] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 178–183, Santa Barbara, USA, June 1998.

[55] M. Tonko and H.H. Nagel. Model-based stereo-tracking of non-polyhedral objects for automatic disassembly experiments. *Int. Journal of Computer Vision*, 37(1):99–118, June 2000.

[56] L. Vacchetti, V. Lepetit, and P. Fua. Stable 3–d tracking in real-time using integrated context information. In *IEEE Int. Conf. on Conference on Computer Vision and Pattern Recognition, CVPR'03*, volume 2, pages 241–248, Madison, WI, June 2003.

[57] M. Vincze. Robust tracking of ellipses at frame rate. *Pattern Recognition*, 34(2):487 – 498, February 2001.

[58] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, October 1995.