A real-time tracker for markerless augmented reality

Andrew Comport, E. Marchand, François Chaumette

ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03, pp. 36-45, Tokyo, Japan, October 2003.

A real-time tracker for markerless augmented reality

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

Abstract

Augmented Reality has now progressed to the point where real-time applications are being considered and needed. At the same time it is important that synthetic elements are rendered and aligned in the scene in an accurate and visually acceptable way. In order to address these issues a real-time, robust and efficient 3D model-based tracking algorithm is proposed for a 'video see through' monocular vision system. The tracking of objects in the scene amounts to calculating the pose between the camera and the objects. Virtual objects can then be projected into the scene using the pose. Here, non-linear pose computation is formulated by means of a virtual visual servoing approach. In this context, the derivation of point-to-curves interaction matrices are given for different features including lines, circles, cylinders and spheres. A local moving edges tracker is used in order to provide real-time tracking of points normal to the object contours. A method is proposed for combining local position uncertainty and global pose uncertainty in an efficient and accurate way by propagating uncertainty. Robustness is obtained by integrating a M-estimator into the visual control law via an iteratively re-weighted least squares implementation. The method presented in this paper has been validated on several complex image sequences including outdoor environments. Results show the method to be robust to occlusion, changes in illumination and miss-tracking.

1. Introduction

This paper addresses the problem of markerless real-time augmented reality (AR). Many different types of sensors have been used to achieve this including : GPS, gyroscopes, cameras, hybrid vision, accelerometers and many more which have been summarized in [1, 2]. Although the implementation oresented here is not restricted to a particular display technology, the problem is restricted to the use of a monocular vision sensor: a camera. This study will focus on the registration techniques that allow alignment of real and virtual worlds using images acquired in real-time by a moving camera. In such systems AR is mainly a pose (or viewpoint) computation issue. In this paper a markerless model-based algorithm is used for the tracking of 3D objects in monocular image sequences. The main advantage of a model based method 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.

Real-time 3D tracking. The most common geometric features used in pose computation which are suitable for AR applications include indoor fiducial/marker based [3, 19, 25, 32, 33] and outdoor fiducial/marker based [26], the latter shows how the size of the marker contributes to robustness and ease of use. In the related computer vision literature geometric primitives considered for the estimation are often points [13, 7], segments [9], lines [20], contours or points on the contours [21, 24, 10], conics [28, 6], cylindrical objects [8] or a combination of these different features [25]. Another important issue is the registration problem. *Purely geometric* (eg, [9]), or *numerical and iterative* [7] approaches may be considered. *Linear approaches* use a least-squares method to estimate the pose. *Full-scale non-linear optimization techniques* (e.g., [21, 23, 10]) 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. It is important to note that other approaches to on-line augmented reality do not rely on pose estimation but on relative camera motion [5], planar homography estimation [30] or optical flow based techniques [26]. These methods have been shown to work in real-time and in outdoor environments, however, they are restricted to planar surfaces which may be problematic in complex environments.

Statistically robust tracking. To handle occlusion, changes in illumination and miss-tracking a statistically robust estimation of the pose has to be considered. In related computer vision and statistics literature many different approaches exist to treat external sources of error. Amongst the robust outlier rejection algorithms, methods in computer vision have included the Hough Transform and RANSAC [12]. These approaches treats the standard deviation of the inlier data (scale) as a constant to be tuned. On the other hand the statistical methods such as Least Median Square (LMedS) and M-estimators [16] have been developed which treat scale as something to be estimated. Up to present most approaches focus only on a single geometric error function and reject any *outliers* which do not correspond to this definition. The reader is referred to [31] for a review of different robust techniques applied to computer vision. Statistically robust pose computation algorithm, suitable for real-time AR techniques, have been considered. Most of these approaches are related directly to computer vision literature [12, 13, 22, 20].

Outline of the paper and contributions In this paper, pose computation is formulated in terms of a full scale non-linear optimization: Virtual Visual Servoing (VVS). In this way the AR pose computation problem is considered as similar to 2D visual servoing as proposed in [32, 25]. 2D visual servoing or image-based camera control [17, 11, 14] allows control of a eye-in-hand camera wrt. to its environment. More precisely it consists in specifying a task (mainly positioning or target tracking tasks) as the regulation in the image of a set of visual features. A set of constraints are defined in the image space. A closed-loop control law that minimizes the error between the current and desired position of these visual features can then be built which determines automatically the motion the camera has to realize. This paper takes this framework and builds an image feature based system which is capable of treating complex scenes in real-time without the need for markers. Contributions can be exhibited at three different levels:

- the derivation of the Jacobian for complex visual features including ellipses, cylinders, points, distances and any combination of these is easily obtained. Determining an accurate approximation of the Jacobian, also called interaction matrix, is essential to obtain the convergence of the visual servoing. In this paper, a complete derivation of interaction matrices for distances to lines, ellipses and cylinders are given. A general framework for derivation is obtained by taking advantage of the duality of visual servoing methodologies. Furthermore, computational efficiencies are obtained by 'stacking' Jacobians and using a constant interaction matrix.
- the widely accepted statistical techniques of robust M-estimation [16] are employed. This is introduced directly in the virtual visual servoing control law by weighting the confidence on each feature. The Median Absolute Deviation (MAD) is used as an estimate of the standard deviation of the inlier data.
- this formulation for tracking objects is dependent on correspondences between local features in the image and the object model. In an image stream these correspondences are given by the local tracking of features in the image. In this paper low level tracking of the contours is implemented via the Moving Edges algorithm [4]. A local approach such as this is ideally suited to real-time tracking due to an efficient 1D search normal to a contour in the image. In a 'real world' scenario some features may be incorrectly tracked, due to occlusion, changes in illumination and miss-tracking. Since many point-to-curve correspondences are made, the method given here has many redundant features which favors the use of robust statistics. Furthermore a method is proposed for propagating uncertainty from the local edge features to a global pose determination algorithm which means that no arbitrary predetermined edge detection threshold is necessary.

It should be noted that Drummond and Cipolla [10] have recently proposed a similar approach to robust complex object tracking. Even though the formulation based on Lie Algebra is very different, it is also a full scale non-linear pose computation. It is also based on a 1D search along the edge normal in subsequent frames, as well as a robust M-estimation, however, only polyhedral objects were considered. The analytical form of the feature Jacobian was not determined, edge detection thresholds were needed and the orientation of the edges were not considered. The latter degrades the performance of the system in terms of accuracy of initial measures and subsequent computational efficiency.

In the remainder of this paper, Section 2.1 presents the principle of the approach. In Section 2.2 the details of the robust visual servoing control law are shown and a stability analysis is presented. In Section 2.3 the computation of the confidence in the local features extraction is introduced. Section 3 deals with the chosen visual features considered in the tracking process. Firstly the analytical formulation of the interaction matrices for various features are derived and then the algorithm used for tracking local features is presented. In Section 4, several experimental results including visual servoing experiments are demonstrated.

2. Robust virtual visual servoing

2.1. 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 [11, 17]. In visual servoing, the goal is to move a camera in order to observe an object at a given position in the image. This is achieved by minimizing the error between a desired state of the image features $\mathbf{s}^*$ and the current state $\mathbf{s}$. If the vector of visual features is well chosen, there is only one final position of the camera that allows this minimization to be achieved. An explanation will now be given as to why the pose computation problem is very similar.

To illustrate the principle, consider the case of an object with various 3D features $\mathbf{P}$ (for instance, ${}^o\mathbf{P}$ are the 3D coordinates of these features in the object frame). A virtual camera is defined whose position in the object frame is defined by $\mathbf{r}$. The approach consists in 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:

$$\Delta = \big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big) = \big[pr_\xi(\mathbf{r}, {}^o\mathbf{P}) - \mathbf{s}^*\big], \tag{1}$$

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 is moved (initially at $\mathbf{r}_i$) 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 ($\mathbf{r_d}$ will be the real camera pose).

Considering that $\mathbf{s}^*$ is computed (from the image) with sufficient precision is an important assumption. In visual servoing, the control law that performs the minimization of Δ is usually handled using a least squares approach [11][17]. 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 [16]. 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:

$$\Delta_{\mathcal{R}} = \rho\Big(\mathbf{s}(r) - \mathbf{s}^*\Big), \tag{2}$$

where $\rho(u)$ is a robust function [16] that grows subquadratically and is monotonically nondecreasing with increasing $|u|$. Iteratively Re-weighted Least Squares (IRLS) is a common method of applying the M-estimator. It converts the M-estimation problem into an equivalent weighted least-squares problem.

To embed robust minimization into visual servoing, a modification of the control law is required to allow outlier rejection.

2.2. Robust Control Law

The objective of the control scheme is to minimize the objective function given in equation (2). This new objective is incorporated into the control law in the form of a weight which is given to specify a confidence in each feature location. Thus, the task function to be regulated to 0 is defined as:

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

- where matrix $\mathbf{C}$ is a combination matrix of size $m \times k$ where k is the number of features and m the number of controlled robot degrees of freedom (6 to reach a unique desired position). this matrix allows to consider more visual features than the number of controlled d.o.f.
- $\mathbf{D}$ is a diagonal weighting matrix given by

$$\mathbf{D} = \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_k \end{pmatrix}$$

The computation of weights w_i is described in Section 2.3.

If $\mathbf{C}$ and $\mathbf{D}$ were constant, the derivative of equation 3) would be given by:

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \mathbf{CDL_s v}, \tag{4}$$

$\mathbf{v}$ is the camera velocity screw and $\mathbf{L_s}$ is called the image Jacobian [17] or interaction matrix [11] related to $\mathbf{s}$. This matrix depends on the value of the image features $\mathbf{s}$ and their corresponding depth Z in the scene (which is available here). If an exponential decrease of the task function $\mathbf{e}$ is specified:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}, \tag{5}$$

where λ is a positive scalar, the following control law is obtained from equation (4):

$$\mathbf{v} = -\lambda (\mathbf{C}\widehat{\mathbf{D}}\widehat{\mathbf{L}_s})^{-1}\mathbf{e}, \tag{6}$$

where $\widehat{\mathbf{L}_s}$ is a model or an approximation of the real matrix $\mathbf{L_s}$ and $\widehat{\mathbf{D}}$ a chosen model for $\mathbf{D}$.

To simplify the control law, $\mathbf{C}$ can be chosen to be the pseudo inverse $(\widehat{\mathbf{D}}\widehat{\mathbf{L}_s})^+$ of $\widehat{\mathbf{D}}\widehat{\mathbf{L}_s}$. This gives $\mathbf{C}\widehat{\mathbf{D}}\widehat{\mathbf{L}_s} = (\widehat{\mathbf{D}}\widehat{\mathbf{L}_s})^+\widehat{\mathbf{D}}\widehat{\mathbf{L}_s} = \mathbf{I}_m$, which finally leads to:

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

If $\widehat{\mathbf{D}}$ and $\widehat{\mathbf{L}_s}$ were constant, a sufficient criteria to ensure global asymptotic stability of the system would be given by [29]:

$$(\widehat{\mathbf{D}}\widehat{\mathbf{L}_s})^+\mathbf{D}\mathbf{L_s} > 0 \tag{8}$$

As usual, in image-based visual servoing, it is impossible to demonstrate the global stability. It is, however, possible to obtain the local stability for two cases of $\widehat{\mathbf{L}_s}$ and $\widehat{\mathbf{D}}$:

- the first case is to use the current value of the weights, an estimate of the depth at each iteration and the current feature:

  $$(\widehat{\mathbf{D}}\widehat{\mathbf{L}_s})^+ = \big[\mathbf{D}\mathbf{L_s}(\mathbf{s}, \hat{Z})\big]^+ \tag{9}$$

  This choice allows the system to follow, as closely as possible, the intended behavior ($\dot{\mathbf{e}} = -\lambda\mathbf{e}$). However, even when condition (8) is satisfied, only local stability can be demonstrated since $\mathbf{D}$ and $\mathbf{L_s}$ are not constant (refer to (4) that has been used to derive (8)).

- In the second case a constant Jacobian is considered using the initial depth $\mathbf{Z_i}$, the initial value of the features $\mathbf{s_i}$ and the first value of the weighting matrix $\widehat{\mathbf{D}} = \mathbf{I}_m$.

  $$(\widehat{\mathbf{D}}\widehat{\mathbf{L}_s})^+ = \big[\mathbf{L_s}(\mathbf{s_i}, \mathbf{Z_i})\big]^+, \tag{10}$$

  This choice leads to a simpler control law:

  $$\mathbf{v} = -\lambda\widehat{\mathbf{L}_s}^+\mathbf{e} = -\lambda\mathbf{L_s}(\mathbf{s_i}, \mathbf{Z_i})^+\mathbf{D}\big(\mathbf{s} - \mathbf{s}^*\big) \tag{11}$$

  and a simpler convergence criteria:

  $$\mathbf{L_s}(\mathbf{s_i}, \mathbf{Z_i})^+\mathbf{D}\mathbf{L_s} > 0. \tag{12}$$

  Note also that, even if the model (10) is constant, the evolution of the weights during the realization of the control law is taken into account through the computation of $\mathbf{e}$, as in (11). Furthermore, the weights $\mathbf{w_i}(0)$ could be computed instead of choosing them to be equal to 1, however, these initial weights may be equally incorrect. Once again, only the local stability of the system can be demonstrated since equation (12) is only satisfied around $\mathbf{s_i}$. In the results presented in section 4, we have used this second solution.

Of course it is also necessary 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. 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 manually.

2.3. Computing confidence

The weights w_i, which represent the different elements of the $\mathbf{D}$ matrix and reflect the confidence of each feature, are usually given by [16]:

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}, \tag{13}$$

where $\psi\Big(\delta_i/\sigma\Big) = \frac{\partial\rho\left(\delta_i/\sigma\right)}{\partial\mathbf{r}}$ (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 loss and corresponding 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 virtual camera motion. This influence function is given by:

$$\psi(u) = \begin{cases} u(C^2 - u^2)^2 & \text{, if } |u| \leq C \\ 0 & \text{, else,} \end{cases} \tag{14}$$

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 is the standard deviation of the inlier data and is an important value for the efficiency of the method. In nonlinear regression for pose computation, 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:

$$\widehat{\sigma} = \frac{1}{\Phi^{-1}(0.75)}Med_i(|\delta_i - Med_j(\delta_j)|). \tag{15}$$

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. To date, a convergence proof for non-linear regression using the MAD only exists if it is calculated once as an ancillary scale estimate due to the median's lack of asymptotic properties [15]. However, although convergence has yet to be proved, experimental results show that recomputing the MAD at each iteration gives better results (see Section 4).

3. Visual features

3.1. Interaction matrices

Any kind of geometrical feature can be considered within the proposed control law as soon as it is possible to compute its corresponding interaction matrix $\mathbf{L_s}$. In [11], 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. It is also easy to show that combining different features can be achieved by adding features to vector $\mathbf{s}$ and by "stacking" the corresponding interaction matrices. 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 [25], classical geometrical features (point, straight line, circle and cylinder) have been considered.

In this paper, a new distance feature '$\mathbf{s}$' 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 feature '$\mathbf{s}^*$' is considered zero. The assumption is made that the contours of the object in the image can be described as piecewise linear segments or portions of ellipses. All distances are then treated according to their corresponding segment or ellipse.

Case of a distance to a line. The derivation of the interaction matrix that links the variation of the distance between a fixed point and a moving line to the virtual camera motion is now given. In Figure 1 $\mathbf{p}$ is the tracked point feature position and $\mathbf{l}(r)$ is the current line feature position.

The position of the line is given by its polar coordinates representation,

$$x\cos\theta + y\sin\theta = \rho, \forall (x,y) \in \mathbf{l}(r), \tag{16}$$

The distance between point $\mathbf{p}$ and line $\mathbf{l}(r)$ can be characterized by the distance $d_\perp$ perpendicular to the line. In other words the distance parallel to the segment does not hold any useful information unless a correspondence exists between a point on the line and $\mathbf{p}$ (which is not the case). Thus the distance feature from a line is given by:

$$d_l = d_\perp(\mathbf{p}, \mathbf{l}(r)) = \rho(\mathbf{l}(r)) - \rho_d, \tag{17}$$

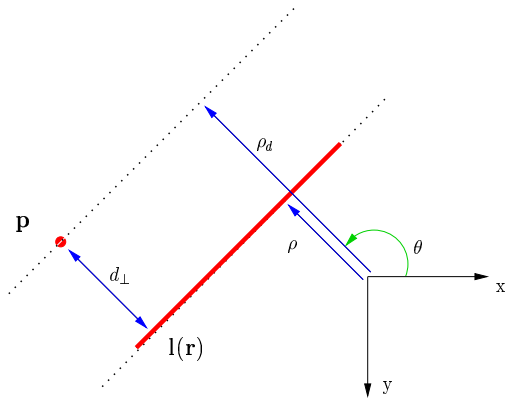

Figure 1. Distance of a point to a line

where

$$\rho_d = x_d\cos\theta + y_d sin\theta, \tag{18}$$

with x_d and y_d being the coordinates of the tracked point. Thus,

$$\dot{d}_l = \dot{\rho} - \dot{\rho_d} = \dot{\rho} + \alpha\dot{\theta}, \tag{19}$$

where $\alpha = x_d\sin\theta - y_d\cos\theta$. Deduction from (19) gives $\mathbf{L}_{d_l} = \mathbf{L}_\rho + \alpha\mathbf{L}_\theta$. The interaction matrix related to d_l can be thus derived from the interaction matrix related to a straight line given by (see [11] for its complete derivation):

$$\begin{aligned}\mathbf{L}_\theta &= \begin{pmatrix} \lambda_\theta\cos\theta & \lambda_\theta\sin\theta & -\lambda_\theta\rho & \rho\cos\theta & -\rho\sin\theta & -1 \end{pmatrix} \\ \mathbf{L}_\rho &= \begin{pmatrix} \lambda_\rho\cos\theta & \lambda_\rho\sin\theta & -\lambda_\rho\rho & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \end{pmatrix}\end{aligned} \tag{20}$$

where $\lambda_\theta = (A_2\sin\theta - B_2\cos\theta)/D_2$, $\lambda_\rho = (A_2\rho\cos\theta + B_2\rho\sin\theta + C_2)/D_2$, and $A_2X + B_2Y + C_2Z + D_2 = 0$ is the equation of a 3D plane which the line belongs to.

From (19) and (20) the following is obtained:

$$\mathbf{L}_{d_l} = \begin{pmatrix} \lambda_{d_l}\cos\theta \\ \lambda_{d_l}\sin\theta \\ -\lambda_{d_l}\rho \\ (1+\rho^2)\sin\theta - \alpha\rho\cos\theta \\ -(1+\rho^2)\cos\theta - \alpha\rho\sin\theta \\ -\alpha \end{pmatrix}^T, \tag{21}$$

where $\lambda_{d_l} = \lambda_\rho + \alpha\lambda_\theta$.

Let it be noted that the case of a distance between a point and the projection of a cylinder is very similar to this case and will be left to the reader.

Case of a distance to an ellipse. Here the derivation of the interaction matrix is given which relates the distance between a fixed point $\mathbf{p}$ and an ellipse that results from the projection in the image plane of a moving circle or a moving sphere. If the ellipse is parameterized by its center of gravity and by the moments of order 2 (that

is $(x_g, y_g, \mu_{02}, \mu_{20}, \mu_{11})$), the distance d_e between a point $\mathbf{p}(x, y)$ and an ellipse is defined by the ellipse equation:

$$\begin{aligned} d_e = \quad & \mu_{02}x^2 + \mu_{20}y^2 - 2\mu_{11}xy + 2(\mu_{11}y_g - \mu_{02}x_g)x \\ & +2(\mu_{11}x_g - \mu_{20}y_g)y + \mu_{02}x_g^2 + \mu_{20}y_g^2 \\ & -2\mu_{11}x_g y_g + \mu_{11}^2 - \mu_{20}\mu_{02} \end{aligned} \tag{22}$$

The variation the distance due to the variation of the ellipse parameters are thus given by:

$$\begin{aligned} \dot{d}_e \quad = \quad & \underbrace{\begin{pmatrix} 2(\mu_{11}(y - y_g) + \mu_{02}(x_g - x)) \\ 2(\mu_{20}(y_g - y) + \mu_{11}(x - x_g)) \\ ((y - y_g)^2 - \mu_{02}) \\ 2(y_g(x + x_g) + x_g y + \mu_{11}) \\ ((x - x_g)^2 - \mu_{20}) \end{pmatrix}^T}_{\mathbf{L}_{d_e}} \begin{pmatrix} \dot{x_g} \\ \dot{y_g} \\ \dot{\mu_{20}} \\ \dot{\mu_{11}} \\ \dot{\mu_{02}} \end{pmatrix} \\ = \quad & \mathbf{L}_{d_e}\mathbf{L}_{\mathbf{c}}\mathbf{v} \end{aligned} \tag{23}$$

where $\mathbf{L_c}$ is the interaction matrix related to an ellipse and is given in [11].

3.2. Tracking visual features

When dealing with image processing, the normal displacements are evaluated along the projection of the object model contours using the spatio-temporal Moving Edges algorithm (ME) [4]. One of the advantages of the ME method is that it does not require any prior edge extraction. Only point coordinates and image intensities are manipulated. For convenience, the word "contour" is used to refer to the list of tracked points. The ME algorithm can be implemented with convolution efficiency, and leads to real-time computation [4, 24]. The process consists in searching for the "correspondent" p^{t+1} in image I^{t+1} of 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 (see Figure 2). For each point p^t and for each entire position Q_j lying in the direction δ a criterion corresponding to the square root of a log-likelihood ratio ζ_j is computed. This ratio is nothing but the absolute sum of the convolution values, computed at p^t and Q_j, using a *pre-determined mask* M_δ function of the orientation of the contour. This improves accuracy and subsequent efficiency of the tracking by only finding edges with the same orientation and not all edges in the path.

The new position p^{t+1} is given by:

$$Q^{j^*} = \arg \max_{j \in [-J,J]} \zeta_j \text{ with } \zeta_j = \mid I^t_{\nu(p^t)} * M_\delta + I^{t+1}_{\nu(Q_j)} * M_\delta \mid$$

$\nu(.)$ is the neighborhood of the considered pixel. At this step, a list of k pixels exists from which distance s_{dl} or s_{de} to their corresponding 3D model feature projection can be computed. This is performed for each new frame and never requires the extraction of new contours.

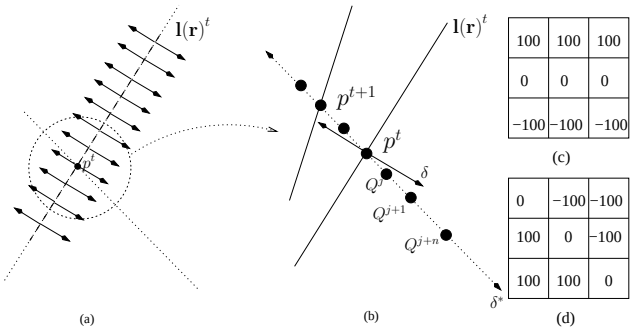

Figure 2. Determining points position in the next image using the ME algorithm: (a) calculating the normal at sample points, (b) sampling along the normal (c-d) 2 out of 180 3x3 predetermined masks (in practice 7x7 masks are used) (c) 180^o (d) 45^o.

3.3. Uncertainty Propagation

The local ME method described in Section 3.2 determines points along the normal of a contour using a maximum likelihood approach. The decision as to whether or not a spatio-temporal edge exists is made by thresholding the local likelihood value. ζ_{j^*} is chosen to be an edge providing that it is greater than a threshold λ. This threshold is usually chosen manually and it depends on both the contrast of the contours in the image as well as the size of the mask being applied. A method is presented here to propagate the local likelihood of the points to the global likelihood of the pose. Assuming that the local measure of uncertainty ζ_{j^*} is independent of the global measure of uncertainty w_i, the weights are thus given by:

$$w_{p_i} = w_i * \zeta_{j^*}, \tag{24}$$

where w_{p_i} is the propagated weight. Matrix $\mathbf{D}$ is then given by

$$\mathbf{D} = \begin{pmatrix} w_{p_1} & & 0 \\ & \ddots & \\ 0 & & w_{p_n} \end{pmatrix}$$

This has the effect of giving the most certainty to strong contours in terms of the local likelihood and amongst those correspondences the M-estimator converges upon those which conform globally to the 3D shape of the object. Effectively the robust estimator chooses which correspondences should be considered instead of a manually chosen threshold. This is advantageous when different scenes are considered along with different size masks.

4. Experimental results

In the four experiments presented,“real” images are acquired using a commercial digital camera. In such experiments, the image processing is potentially very complex. Indeed extracting and tracking reliable points in real environment is a non trivial issue. The use of more complex features such as the distance to the projection of 3D circles, lines, and cylinders is demonstrated. In all experiments, the distances are computed using the Moving Edges algorithm previously described. Tracking is always performed at below frame rate.

Tracking in an indoor environment. In the first experiment the result of the tracking of four 3D circles is shown. This long sequence (more than 2000 images) contains multiple occlusions of some of these circles. Although the images are quite simple in this experiment, if no robust estimation is considered tracking fails after a few images because the minimization process has to deal with miss-tracking and problems due to occlusion.

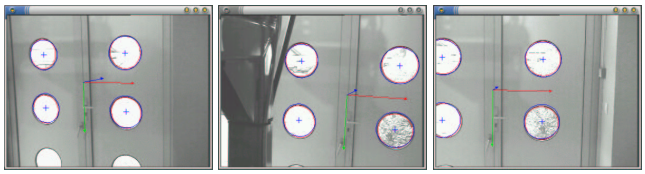

Figure 3. Tracking 3D circles. Four circles are tracked along a 2000 images sequence. This sequence features multiple occlusions of some of these circles.

In the second experiment an object whose model is composed of a cylinder, a circle and two straight lines is considered (See Figure 4). This illustrates the capabilities of our algorithm to consider various features within the same minimization process. This long sequence features numerous occlusions.

Tracking in an outdoor environment In the third experiment (see Figure 5), an outdoor scene is considered. Here, distance to the projection of a cylinder and to two straight lines are used to compute the pose. Despite very noisy images (wind in the trees, multiple occlusions, etc.) tracking is achieved along a 1400 image sequence. The images display the tracked lines and cylinder limbs as well as 3D information inserted after the pose computation (the reference frame and the projection of the cylinder and lines (in blue)). In Figure 6 this approach was applied to a real-time augmented reality application. These images are extracted from the sequence after the insertion of virtual objects. Due to both the introduction of the robust estimation and to the high redundancy of visual features, visual alignment is satisfactory to the human eye.

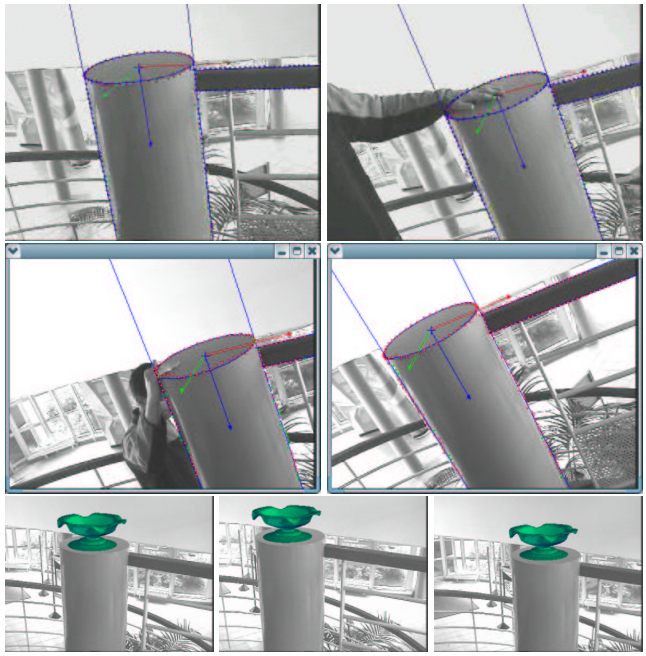

Figure 4. Tracking considering a circle, a cylinder and two straight lines and the resulting AR sequence.

An augmented reality application for system maintenance In this example a scenario is given for guided maintenance of an airconditioning system. Many lines with different contrasts along with two circles are considered in the pose computation. Large images of size 768x576 pixels are used in the tracking and real-time performance is still obtained The scenario deals with heavy occlusions and the effects of video interlacing. In the images displayed in Figure 8a, red dots correspond to inlier data, white dots correspond to data rejected by the ME algorithm and green dots correspond to the outliers rejected by M-estimation. Forward projection of the model appears in blue. On Figure 8c, one can see that most of the dots on the right part of the object are correctly detected as outliers. Despite this large occlusion tracking is correctly handled. The images in Figure 8b display the results of a simple AR scenario.

Tracking in a 3D visual servoing experiment. A positioning task using a CCD camera mounted on the end effector of a six d.o.f robot has been considered. This application requires both a fast and reliable tracking algorithm. From an initial position, the robot has to reach a desired position expressed as a desired position of the object in the image (depicted in blue in the images). The object of interest in this experiment was a micro-controller device. To prove the

Figure 5. Tracking in an outdoor environment. Despite multiple occlusions and disturbances, tracking is still very reliable and handled in real-time.

Figure 6. Tracking considering a cylinder and two straight lines with application to augmented reality.

robustness of the algorithm, the micro-controller was placed in a highly textured environment as shown in Fig. 7. Tracking and positioning tasks were correctly achieved. Images were acquired and processed at video rate (25Hz). Multiple temporary and partial occlusions by an hand and various tools as well as modification of the lighting conditions were imposed during the realization of the positioning task.

5. Conclusion and Future Perspectives

This paper has presented an accurate and efficient real-time AR algorithm that is robust to many sources of external error. 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 highly complex scenes. In particular, the interaction matrices that link the virtual camera velocity to the variation of a distance in the image were determined. The generality of the formulation was shown by determining distance features for more complex objects such as straight lines, spheres and cylinders. A new robust control law that integrates an M-estimator has been proposed. 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. Experimental results obtained using several cameras, lens, and environments were presented. The algorithm has been tested on various images sequences and for various applications (visual servoing, augmented reality,...) which demonstrates a real usability of this approach. Each time tracking is handled in real-time.

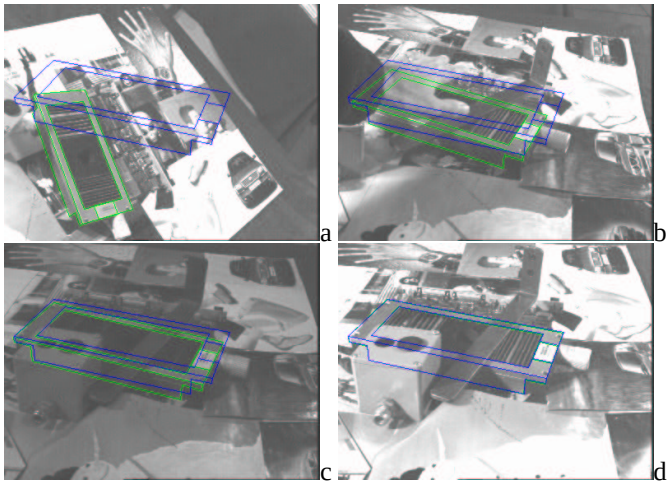

Figure 7. Tracking in complex environment within a classical visual servoing experiments: Images are acquired and processed at video rate (25Hz). Blue : the desired position defined by the user. Green : position measured after pose calculation. (a) first image initialized by hand, (b) partial occlusion with hand, (c) lighting variation, (d) final image with various occlusions

In perspective, the algorithm presented here has several limitations that need to be addressed in the future. Firstly it relies on a coarse manual localization of 4 points to initialize the pose. The experimental cone of convergence is about 30^o and the maximum speed of the camera or the object relies on a trade-off between real-time calculation and the search distance normal to the contour. With current computing power this distance is very large (10 to 15 pixels). A lack of contrast around contours and too large occlusions are classical failure modes. Finally, this method uses a CAD model which is approximately created by hand and a piecewise parametric representation of the object. Future work will be devoted to addressing these issues by considering deformable objects and the reconstruction of parametric objects models.

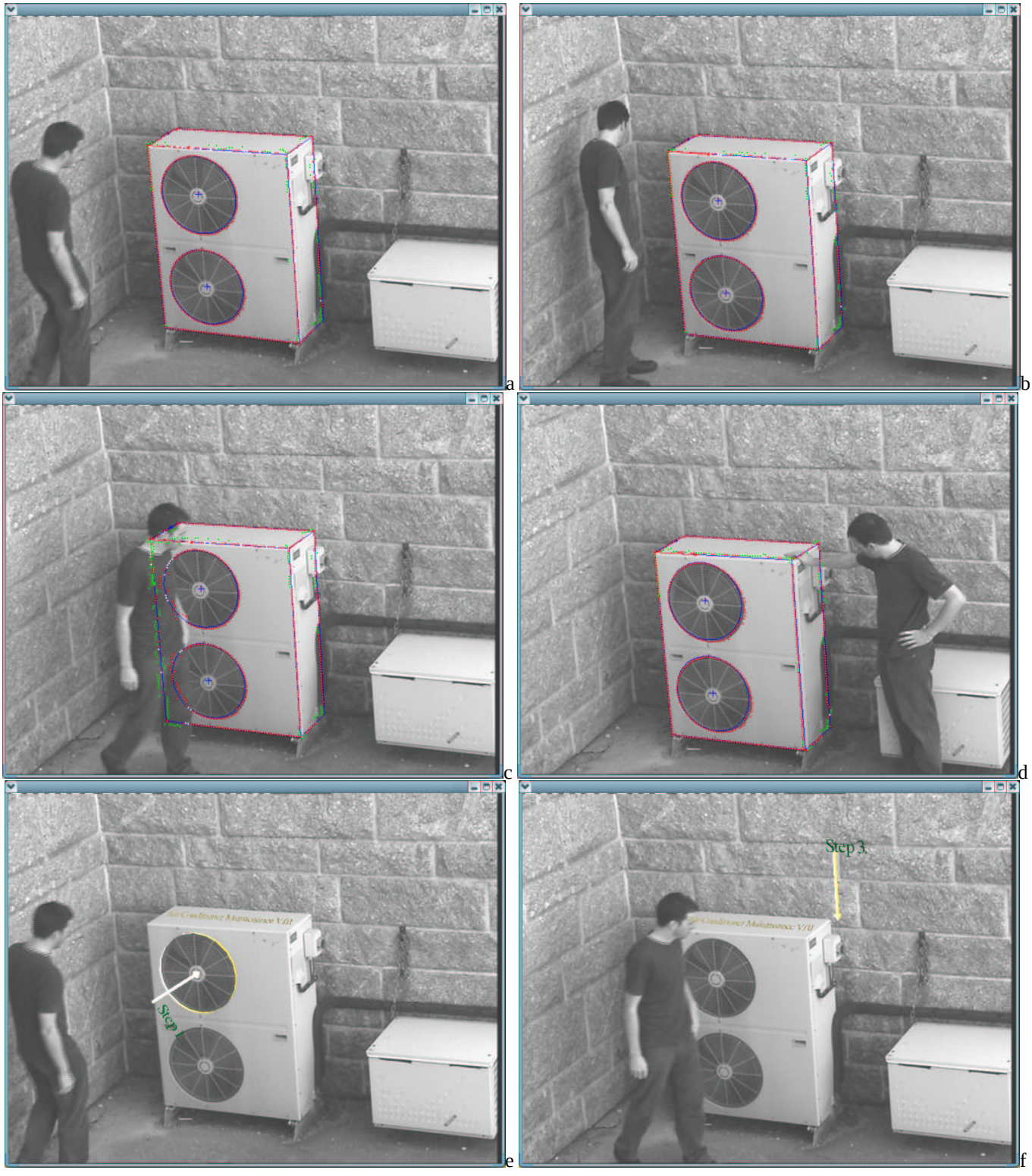

Figure 8. Tracking and AR for a constructed maintenance scenario. Even with heavy occlusion and disturbances, tracking is still very reliable and handled in real-time. (a-d) Tracking results (e-f) AR sequences corresponding to image a (resp c)

In this study the spatio-temporal aspect of the tracking process has not been considered in depth. Indeed robustness can also be handled from one time-step to another (as is possible in a Bayesian framework, such as with the Extented Kalman Filter e.g. [27] or a particle filter [18]). The measurements provided by our algorithm could then become part of the measurement model for such approach. Such filters may be considered in a future implementation of our system.

References

[1] R. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, Aug 1997.

[2] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Application*, 21(6):34–47, November 2001.

[3] M. Billinghurst, H. Kato, and I. Poupyrev. The magicbook: Moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, 21(3):6–8, May 2001.

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

[5] K.-W. Chia, A.-D. Cheok, and S. Prince. Online 6 dof augmented reality registration from natural features. In *IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR'02)*, pages 305–316, Darmstadt, Germany, September 2002.

[6] S. de Ma. Conics-based stereo, motion estimation and pose determination. *Int. J. of Computer Vision*, 10(1):7–25, 1993.

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

[8] M. Dhome, J.-T. Lapresté, G. Rives, and M. Richetin. Determination of the attitude of modelled objects of revolution in monocular perspective vision. In *European Conference on Computer Vision, ECCV'90*, volume LNCS 427, pages 475–485, Antibes, April 1990.

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] P.-W. Holland and R.-E. Welsch. Robust regression using iteratively reweighted least-squares. *Comm. Statist. Theory Methods*, A6:813–827, 1977.

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

[17] 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.

[18] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, January 1998.

[19] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proceedings of Int. Symp. on Augmented Reality 2000*, October 2000.

[20] R. Kumar and A. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Understanding*, 60(3):313–342, Novembre 1994.

[21] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–394, 1987.

[22] D. Lowe. Robust model-based motion tracking trough the integration of search and estimation. *Int. J. of Computer Vision*, 8(2):113–122, 1992.

[23] C. Lu, G. 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.

[24] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *IEEE Int. Conf. on Computer Vision, ICCV'99*, volume 1, pages 262–268, Kerkira, Greece, September 1999.

[25] 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.

[26] U. Neumann, S. You, Y. Cho, J. Lee, and J. Park. Augmented reality tracking in natural environments. In *International Symposium on Mixed Realities*, Tokyo, Japan, 1999.

[27] J. Park, B. Jiang, and U. Neumann. Vision-based pose computation: Robust and accurate augmented reality tracking. In *ACM/IEEE International Workshop on Augmented Reality*, pages 3–12, San Francisco, California, October 1998.

[28] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, and K. Smith. Three dimentional location estimation of circular features for machine vision. *IEEE trans on Robotics and Automation*, 8(2):624–639, october 1992.

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

[30] G. Simon and M.-O. Berger. Reconstructing while registering: A novel approach for markerless augmented reality. In *IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR'02)*, pages 285–294, Darmstadt, Germany, Sept 2002.

[31] C.-V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, September 1999.

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

[33] X. Zhang, S. Fronz, and N. Navab. Visual marker detection and decoding in ar systems: A comparative study. In *IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR'02)*, pages 79–106, Darmstadt, Germany, September 2002.