A New Formulation for Non-Linear Camera Calibration Using Virtual Visual Servoing

Eric Marchand, François Chaumette

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A new formulation for non-linear camera calibration using virtual visual servoing

Éric Marchand and François Chaumette

N˚4096

Janvier 2001

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--4096--FR+ENG

A new formulation for non-linear camera calibration using virtual visual servoing

Éric Marchand* and François Chaumette†

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projets Vista

Rapport de recherche n ° 4096 — Janvier 2001 — 22 pages

Abstract: This paper presents a new formulation for the non-linear calibration problem. We propose a method based on the well known visual servoing approach. We consider pose computation and lens calibration as the dual problem of visual servoing. It allows to take advantage of all the research that have been carried out in this domain in the past. The proposed method features accuracy, efficiency, scalability wrt. to the camera model as well as wrt. to the features extracted from the image, and simplicity.

Key-words: Calibration, non-linear approach, virtual visual servoing

(Résumé : tsvp)

* Projet Vista, Eric.Marchand@irisa.fr
† Projet Vista, Francois.Chaumette@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Calibration par asservissement visuel virtuel

Résumé : Ce rapport présente une formulation originale pour le problème de la calibration des caméras. Nous proposons une approche reposant sur l'utilisation de l'asservissement visuel 2D. L'idée consiste à considérer la calibration (c'est à dire le calcul des paramètres intrinsèques d'une caméra et le calcul de pose entre la caméra et une mire) comme le problème dual de l'asservissement visuel. Cette méthode considère l'image de l'objet comme le motif désiré à atteindre, le motif courant étant obtenu par la reprojection du modèle 3D de l'objet sur le plan image d'une caméra virtuelle. Cette méthode présente de nombreux avantages : précision identique aux méthodes de minimisation non-linéaire classiques, simplicité. efficacité, et possibilité de considérer différente informations visuelles

Mots-clé : Calibrage, méthode non-linéaire, asservissement visuel virtuel

1 Introduction

Calibration is an important issue in computer vision as soon as accurate metric information have to be extracted from a set of 2D images. This is a research area that received much attention since the early 70's first in the photogrammetry community (e.g., [1]) then in the computer vision and robotics communities (e.g., [3, 9, 11, 10, 6], etc.).

Most of the approaches consider the calibration issue as a registration problem that consists in determining the relationship between 3D coordinates of points (or other features: lines, ellipses, ...) and their 2D projections in the image plane. These 3D features are usually part of a calibration rig and the position of these features in a world frame have to be known with a very good accuracy. Performing the calibration leads to the estimation of the intrinsic camera parameters (image center, focal length, distortion) and extrinsic camera parameters (i.e., the pose: position and orientation of the camera frame wrt. the rig frame).

Different calibration techniques exist and may be classified in three main categories:

- *linear techniques.* These techniques (e.g., [3]) use a least-squares method to estimate both the intrinsic parameters and the pose. They cannot handle lens distortions.

- *full-scale non-linear optimization techniques.* These techniques have been introduced by the photogrammetry community [1]. It consists in minimizing the error between the observation and the back-projection of the model. Minimization is handled using numerical iterative algorithms such as Newton-Raphson or Levenberg-Marquartd. The main advantage of these approaches are their accuracy and their versatility: the camera model can be very general. The main drawback is that they may be subject to local minima and, worse, divergence. Therefore they usually require a good guess of the solution to ensure a correct convergence.

- *two-steps techniques.* These approaches (e.g., [9, 11, 10]) consider a linear estimation of some parameters while the others are estimated iteratively. Constraints (such as the Tsai's radial alignment constraint) are considered to linearize the estimation of some parameters. These algorithms allow a faster convergence of the algorithm.

Let us finally note that other techniques, called *self calibration* approaches, do not use any calibration rig [6, 5]. The calibration is performed using a moving camera

that observes a static scene. Constraints are provided by the scene rigidity. Reliable results are difficult to obtain using these approaches.

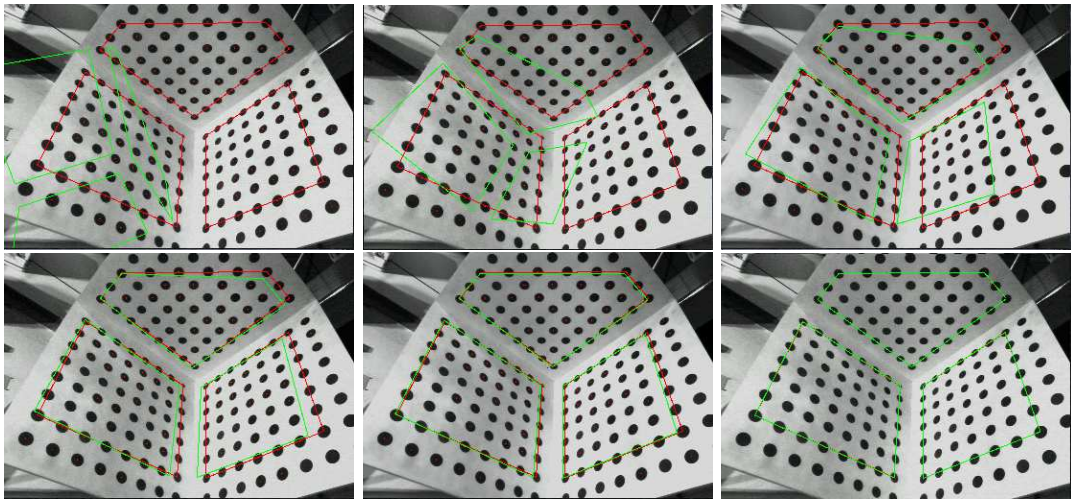

Figure 1: Calibration by virtual visual servoing: principle. The goal is to iteratively modify by visual servoing the parameters (pose and intrinsic parameters) of a virtual camera in order to register the desired features extracted from the image (in red) and the current one obtained by back-projection (in green). This figure illustrates the registration convergence and the minor influence of the initialization.

In this paper we propose a new formulation of the camera calibration involving a full scale non-linear optimization. We consider the pose computation problem as the dual of 2D visual servoing [8] and we extend this issue to camera calibration. Considering calibration as an image-based visual servoing problem [4, 2] takes advantage of all the background and the results in this field. It allows us to propose a very simple and versatile formulation of camera calibration. Various camera models can be performed and calibration from different features may be considered within the same process. Furthermore we can easily consider data from multiple images to improve the quality of the intrinsic parameters estimation.

In the remainder of this paper, we present in Section 2 the principle of the approach. In Section 2.3 we show how our approach is modified to handle multi-images calibration. In Section 3, we present the chosen camera model and we derive the visual servoing equation related to this model. In Section 4, we present some results on various image sets and we compare our method with other approaches.

2 Virtual visual servoing based camera calibration

2.1 Principle

As already stated, the basic idea of our approach is to define the pose computation and the calibration problem as the dual problem of 2D visual servoing [2, 4]. In visual servoing, the goal is to move the 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 $\mathbf{p_d}$ and the current state $\mathbf{p}$. If the vector of visual features is well chosen, there is only one final position of the camera that allows to achieve this minimization. We now explain why the calibration problem is very similar.

Let us define a virtual camera with intrinsic parameters ξ located at a position such that the object frame is related to the camera frame by the homogeneous 4×4 matrix ${}^c\mathbf{M}_o$. ${}^c\mathbf{M}_o$ defines the pose whose parameters are called extrinsic parameters. The position of the object point ${}^c\mathbf{P}$ in the camera frame is defined by:

$$ {}^c\mathbf{P} = {}^c\mathbf{M}_o {}^o\mathbf{P} $$

and its projection in the digitized image by:

$$ \mathbf{p} = pr_\xi({}^c\mathbf{P}) = pr_\xi({}^c\mathbf{M}_o {}^o\mathbf{P}) \tag{1} $$

where $pr_\xi(.)$ is the projection model according to the intrinsic parameters ξ. The goal of calibration is to minimize the error between the observed data denoted $\mathbf{p_d}$ (usually the position of a set of features on a calibration rig) and the position of the same features computed by back-projection according to the current extrinsic and intrinsic parameters $\mathbf{p}$ (as defined in Equation 1). In order to ensure this minimization we move the virtual camera (initially in ${}^{ci}\mathbf{M}_o$ and modify the intrinsic camera parameters (initially ξ_i) using a visual servoing control law. When the minimization is achieved, the parameters of the virtual camera will be ${}^{cf}\mathbf{M}_o$, that is the real pose, and ξ_f.

This process is illustrated in Figure 1.We will show in the next paragraphs how to perform this minimization using visual servoing an the interests of considering this approach.

2.2 Visual servoing and calibration

The goal is to minimize the error $\|\mathbf{p} - \mathbf{p_d}\|$. We therefore define the error in the image $\mathbf{e}$ by the simple relation:

$$ \mathbf{e} = \mathbf{p} - \mathbf{p_d} \tag{2} $$

The motion of the features in the image is related to the camera velocity $\mathbf{T_c}$ and the time variation of the intrinsic parameters by:

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} + \frac{\partial \mathbf{p}}{\partial \xi} \frac{d\xi}{dt} \tag{3}$$

that can be rewritten as:

$$\dot{\mathbf{p}} = \mathbf{H_p V} \qquad \text{with} \qquad \mathbf{V} = \begin{pmatrix} \mathbf{T_c} \\ \\ \dot{\xi} \end{pmatrix} \tag{4}$$

Matrix $\mathbf{H_p}$ is classically called interaction matrix or image Jacobian in the visual servoing community. It is given:

$$\mathbf{H_p} = \begin{pmatrix} \frac{\partial \mathbf{p}}{\partial \mathbf{r}} & \frac{\partial \mathbf{p}}{\partial \xi} \end{pmatrix} \tag{5}$$

If we specify an exponential decoupled decrease of the error $\mathbf{e}$ that is:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{6}$$

where λ is a proportional coefficient that tunes the decay rate, we finally get:

$$\mathbf{V} = -\lambda {\mathbf{H_p}}^{+} \mathbf{e} \tag{7}$$

where $\mathbf{H_p^+}$ is the pseudo inverse of matrix $\mathbf{H_p}$ ($\mathbf{H}^+ = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ if $\mathbf{H}$ is a full rank matrix).

Comments about the choice of p. Any kind of feature can be considered within this control law as soon as we are able to compute the image Jacobian $\mathbf{H_p}$. In [2], a general framework to compute $\frac{\partial \mathbf{p}}{\partial r}$ is proposed. On the other side $\frac{\partial \mathbf{p}}{\partial \xi}$ is seldom difficult to compute as will be shown in the next section. This is one of the advantages of this approach with respect to other non-linear calibration approaches. Indeed we are able to perform calibration from a large variety of primitives (points, lines, circles, etc...) within the same framework. Furthermore, considering various kind of primitives within the same calibration step is also possible.

2.3 Multi-images calibration

The intrinsic parameters obtained using one image may be, in practice, very different from the parameters obtained with another image taken from another viewpoint, even

if the same lens, the same camera, the same frame grabber and the same calibration rig are used (see the results presented in Tab 6). It is therefore important to consider a multi-image calibration process that integrates within a unique minimization process data from various images.

The underlying idea is to compute a unique set of intrinsic parameters that is correct for all the images (i.e., for all the camera positions). Puget and Skordas [7] considered this idea. They computed the final intrinsic parameters as the mean value of the parameters computed for each images, the camera positions are then recomputed wrt. to these new intrinsic parameters. Our approach is different. We consider a visual servoing scheme that computes the motion of n virtual cameras and the variation of the l intrinsic parameters that have to be the same for all the images. For n images, we therefore have $6n + l$ unknown variables and $\sum_{i=1}^{n} m_i$ equations (where m_i is the number of features observed in the i^{th} image).

If $\mathbf{p}^i$ is the set of features extracted from i^{th} image, the interaction matrix used in the calibration process is then given by the relation:

$$\begin{pmatrix} \dot{\mathbf{p}}^1 \\ \dot{\mathbf{p}}^2 \\ \vdots \\ \dot{\mathbf{p}}^n \end{pmatrix} = \mathbf{H} \begin{pmatrix} {\mathbf{T_c}}^1 \\ {\mathbf{T_c}}^2 \\ \vdots \\ {\mathbf{T_c}}^n \\ \\ \dot{\xi} \end{pmatrix} \tag{8}$$

with

$$\mathbf{H} = \begin{pmatrix} \frac{\partial \mathbf{p}^1}{\partial \mathbf{r}} & 0 & \dots & 0 & \frac{\partial \mathbf{p}^1}{\partial \xi} \\ 0 & \frac{\partial \mathbf{p}^2}{\partial \mathbf{r}} & \dots & 0 & \frac{\partial \mathbf{p}^2}{\partial \xi} \\ \vdots & \vdots & & & \vdots \\ 0 & \dots & 0 & \frac{\partial \mathbf{p}^n}{\partial \mathbf{r}} & \frac{\partial \mathbf{p}^n}{\partial \xi} \end{pmatrix} \tag{9}$$

Minimization is handled using the same methodology:

$$\begin{pmatrix} {\mathbf{T_c}}^1 \\ {\mathbf{T_c}}^2 \\ \vdots \\ {\mathbf{T_c}}^n \\ \\ \dot{\xi} \end{pmatrix} = -\lambda \mathbf{H}^+ \begin{pmatrix} \mathbf{p}^1 - \mathbf{p}_\mathbf{d}^1 \\ \mathbf{p}^2 - \mathbf{p}_\mathbf{d}^2 \\ \vdots \\ \mathbf{p}^n - \mathbf{p}_\mathbf{d}^n \end{pmatrix} \tag{10}$$

3 Calibration from points

3.1 Camera model

In this section, we use a simplification of the model used in photogrammetry [1]. We will only consider in this model the radial distortion. However, considering other kind of geometrical distortion is possible (see Appendix A) but experiments show that such more complex model does not really improve the quality of the results.

Let us define by $\mathbf{M} = (X, Y, Z)^T$ the coordinates of a point in the camera frame. The coordinates of the perspective projection of this point in the image plane is given by $\mathbf{m} = (x, y)^T$ with:

$$\begin{cases} x = X \,/\, Z \\ \\ y = Y \,/\, Z \end{cases} \tag{11}$$

If we denote (u, v) the position of the corresponding pixel in the digitized image, this position is related to the coordinates (x, y) by:

$$\begin{cases} u = u_0 + p_x x + \delta_u(u, v) \\ \\ v = v_0 + p_y y + \delta_v(u, v) \end{cases} \tag{12}$$

where δ_u and δ_v are geometrical distortions introduced in the camera model. These distotions are due to imperfections in the lenses design and assembly there usually are some positional errors that have to be taken into account. δ_u and δ_v can be modeled as follow:

$$\begin{cases} \delta_u(u, v) = K_d \, r^2 \, \tilde{u} \\ \\ \delta_v(u, v) = K_d \, r^2 \, \tilde{v} \end{cases} \tag{13}$$

with $\tilde{u} = u - u_0$, $\tilde{v} = (v - v_0)$ and $r^2 = \tilde{u}^2 + \tilde{v}^2$.

The five parameters to be estimated are thus $\xi = \{p_x, p_y, u_0, v_0, K_d\}$.

3.2 Deriving the interaction matrix

We have to compute the interaction matrix $\mathbf{H_p}$ that links the motion $\dot{\mathbf{p}} = (\dot{u}, \dot{v})$ of a point $\mathbf{p} = (u, v)$ in the image to $[\mathbf{T_c} \quad \dot{\xi}]^T$. For one point, this Jacobian is given by:

$$\mathbf{H_p} = \begin{pmatrix} \frac{\partial \mathbf{p}}{\partial \mathbf{r}} & \frac{\partial \mathbf{p}}{\partial \xi} \end{pmatrix} \tag{14}$$

where $\frac{\partial \mathbf{p}}{\partial \mathbf{r}}$ is a 2×6 matrix and $\frac{\partial \mathbf{p}}{\partial \xi}$ is a 2×5 matrix. Considering a calibration with n points, the full image Jacobian is given by the $2n \times 11$ matrix:

$$\mathbf{H} = (\mathbf{H_{p_1}}, \quad \mathbf{H_{p_2}}, \quad \dots \quad \mathbf{H_{p_n}})^T \tag{15}$$

One of the interest of this approach is that it is possible to consider the background in visual servoing. The image Jacobian $\frac{\partial \mathbf{p}}{\partial \mathbf{r}}$ that relates the motion of a point in the image to the camera motion is quite classical [4][2] and is given by:

$$\frac{\partial \mathbf{p}}{\partial \mathbf{r}} = \begin{pmatrix} p_x & 0 \\ 0 & p_y \end{pmatrix} \mathbf{L_p} \tag{16}$$

with

$$\mathbf{L_p} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \tag{17}$$

Furthermore, from (12), differentiating u and v for ξ leads very easily to:

$$\frac{\partial \mathbf{p}}{\partial \xi} = \begin{pmatrix} x & 0 & 1 - K_d\,(r^2 + 2\tilde{u}^2) & -2\,K_d\,\tilde{u}\tilde{v} & \tilde{u}\,r^2 \\ 0 & y & -2\,K_d\,\tilde{u}\tilde{v} & 1 - K_d\,(r^2 + 2\tilde{v}^2) & \tilde{v}\,r^2 \end{pmatrix} \tag{18}$$

We are now able to write the relation that link the motion of a point in the image to the camera motion and the variation of the intrinsic camera parameters.

Let us finally note that, If we do not want to consider distortion within the camera model, this equation can be simplified and we replace $\frac{\partial \mathbf{p}}{\partial \xi}$ by:

$$\frac{\partial \mathbf{p}}{\partial \xi} = \begin{pmatrix} x & 0 & 1 & 0 \\ 0 & y & 0 & 1 \end{pmatrix} \tag{19}$$

with $\xi = (p_x, p_y, u_0, v_0)^T$

These equations are then really simple with respect to the equation usually derive for classical non-linear minimization techniques (see Appendix B for a comparison issue).

4 Experimental Results

The proposed algorithm has been tested on various images. We considered image acquired on our calibration rig with various lens (12.5mm: Figure 2.a and 6mm: Figure 2.b) and image from the Calibrated Imaging Laboratory at CMU [1] (Figure 2.c).

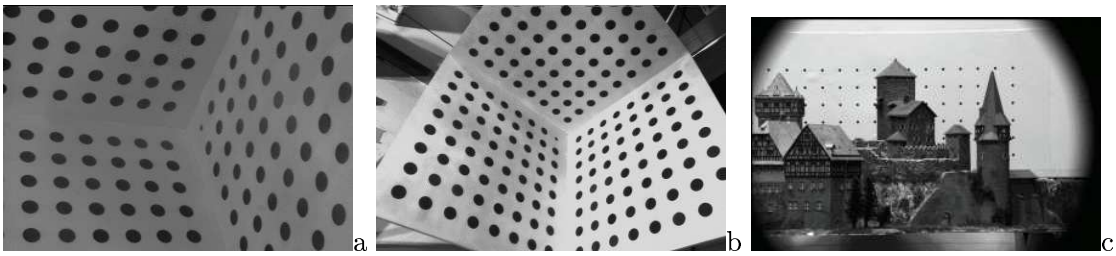

Figure 2: Image from the test database (a) image of our calibration rig with 12.5mm (b) 6mm (c) image from the CIL at CMU

4.1 Mono image calibration results

To illustrate the behavior of our algorithm we have first performed the calibration of a 6mm lens that features an important distortion using our calibration rig. The image size is 730×512. This rig is made of 192 points on three orthogonal planes (see Figure 2.b). Only 170 points are visible and then considered within the calibration process. The image coordinates of each point is computed with a precision of 0.1 pixel.

Algorithm behavior. Figure 1 shows the back-projection of the "model" of a sub-part of the rig from six different virtual position of the camera during the convergence. Figure 3 shows the mean error (in pixel) computed at each iteration of the minimization algorithm. As required by equation (6), the error decreases exponentially. In this experiment, we have first assumed a distortion free camera in order to get a first good guess of the intrinsic and extrinsic parameters. The computed solution is then refined considering the full model. We can see the improvement of the minimization when the distortion is considered (after iteration 180 on Figure 3.

[1] The images were taken with a scientific camera in an indoor setting. These datasets contain multiple images of static scenes with accurate information about object locations in 3D.

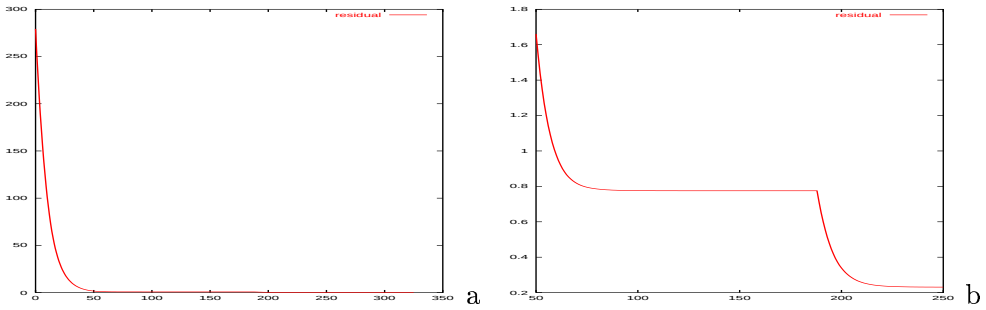

Figure 3: Mean error in pixel (we consider the radial distortion after iteration 130) (a) iteration 0 to 300 (b) iteration 50 to 250

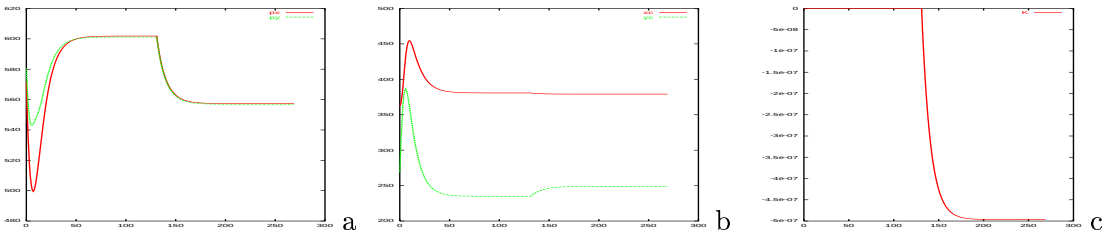

Figure 4: Evolution of the intrinsic parameters (a) p_x and p_y (b) u_0 and v_0 (c) K_d

Figure 4 shows the evolution of the intrinsic parameters over time during the minimization process. The minimization algorithm is fast and does not require many iterations (around 100 in this experiment where $\lambda = 0.1$ and where the initial guess for the solution is very different from the correct solution). Table 1 shows the number of iterations required to minimize the error with respect to the proportional coefficient λ (see equation (7)) that controls the decay rate. The algorithm ended when the difference between the computed error $\mathbf{e}$ at iteration t and at iteration $t+1$ is less than 10^{-8}. The first column ("VVS") shows the number of required iterations from an arbitrary initialization (that is the intrinsic parameters given by the manufacturer and a position corresponding to the first image on Figure 1. The second column ("Toscani+VVS") considers the same algorithm but the calibration parameters are initialized using the Toscani-Faugeras algorithm [3]. Times are in milliseconds and are obtained on a Pentium II 400MHz.

gain	VVS		Toscani then VVS
λ	# iterations	time (ms)	# iterations
0.1	161	826	117
0.25	63	513	48
0.5	25	208	23
0.7	20	152	15
1	13	93	7

Table 1: Convergence rate with respect to λ (see equation (7)).

Convergence issue. Initialization is often considered as a problem for this kind of approach. Indeed, non-linear optimizations method such as Gauss-Newton, Newton-Raphson or Levenberg-Marquartd are supposed to require a good guess of the initial parameters. Tsai points out that this violates the principle of automation. However, our tests show, for this particular application, that important errors may be introduced in the initial vector of parameters. This is reported in Table 2 and 3. Results show that even with important errors in the pose initialization, the algorithm converges toward the same solution. However, for few initial positions or some initial intrinsic parameters, it cannot converge (line with NC after the initial mean error in the tables). Figure 5 displays some of the virtual camera trajectories between their initial position toward their final position.

It remains that, in practice, initialization is then not really a fundamental issue. Considering the manufacturer data for the intrinsic parameters provides, if not a good calibration, a good guess of the solution. Furthermore, the problem can be fully solved if initialization is performed through a simple linear calibration algorithm such as the Toscani-Faugeras algorithm.

Comparisons with other calibration algorithms. For comparison issue we considered various approaches: the Toscani-Faugeras algorithm [3], the Tsai's calibration method[2] [9], a full scale non-linear calibration (non-linear optimization is performed using the Levenberg-Marquartd algorithm from the NAG library), our virtual visual servoing (VVS) algorithm using only the radial distortion (i.e., the model presented in Section 3.1), our VVS algorithm using the full model as presented by Weng [11] (see the Appendix A).

[2]We use the optimized version of the implementation by Reg Willson available at `http://www.cs.cmu.edu/ rgw/TsaiCode.html`

initial position						initial mean
Tx	Ty	Tz	Rx	Ry	Rz	error
1	0.5	0.5	20	0	100	12129.5
1	0.5	1.5	20	0	100	545.51
1	-1	1.5	20	0	100	655.57
1	-1	1.5	20	-20	80	707.11
-1	0	2	0	0	0	411.37
1	-1	0.5	50	-50	60	3088.66
1	-1	0.5	50	-50	30	4127.71
3	3	3	0	0	0	909.97 (NC)
initial position						final mean
0.038	0.039	1.08	3.42	21.87	118.97	0.2308

Table 2: Convergency test for various initial position of the virtual camera: this table show the initial position of the camera and the resulting mean error. The initial intrinsic camera parameters are those published by the camera manufacturer. Last line of the table is the final computed pose. (NC means that the algorithm did not converge)

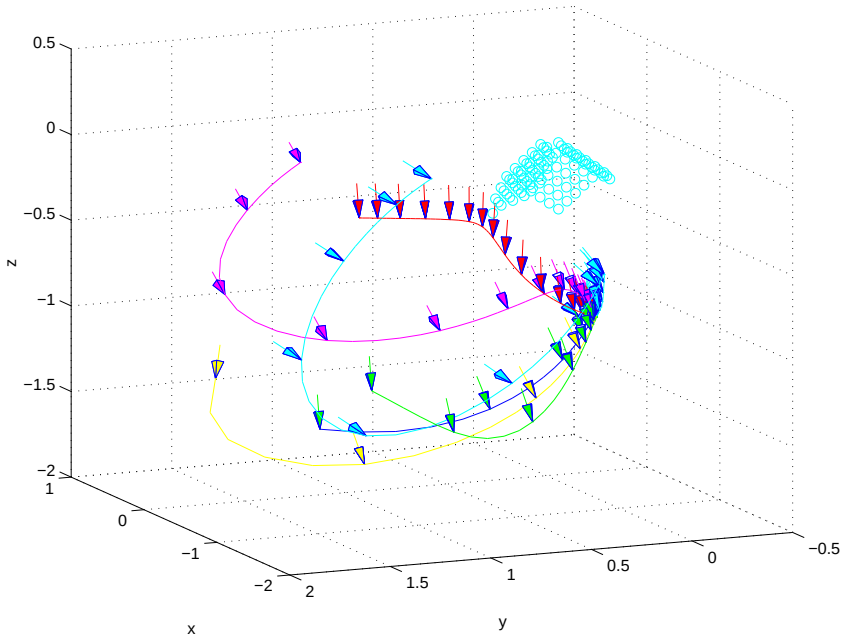

Figure 5: Virtual camera trajectory toward the computed pose

initial intrinsic parameters					initial mean
u_0	v_0	p_x	p_y	K_d	error
365	256	600	600	0	267.53
365	256	600	600	-1e-6	269.84
0	0	600	600	0	564.63
0	0	600	600	-1e-6	694.85
0	0	600	600	-1e-5	1961.2 (NC)
0	0	1000	1000	0	669.36
5000	5000	2500	2500	0	6757.3
computed intrinsic parameters					final mean
379.10	248.84	557.38	556.93	-4.96e-07	0.2308

Table 3: Convergency test for various initial intrinsic parameters and the resulting mean error. The initial position is constant ($T = (-0.2,\ 0.2,\ 1.3)$ and $R = (0,\ 0,\ 180)$). Last line of the table is the final computed intrinsic parameters. (NC means that the algorithm did converge)

6mm lens	p_x	p_y	u_0	v_0	mean error
Toscani	600.42	600.10	381.28	235.44	0.689
Tsai	557.18	556.75	379.16	248.95	0.2312
LM	557.38	556.93	379.10	248.84	0.2308
VVS	557.38	556.93	379.10	248.84	0.2308
VVS (full)	558.21	557.76	558.21	557.76	0.2221

Table 4: Comparison of calibration algorithms for a 6 mm lens. We consider the Toscani-Faugeras algorithm, the Tsai algorithm, a full scale non-linear calibration (Levenberg-Marquartd optimization), our VVS algorithm and our VVS algorithm with the full camera model

Table 4 shows calibration results using these various algorithms. We do not show the distortion parameters since the considered camera are not always the same. Table 5 shows the final mean error, the standard deviation and the maximum error for the three images of Figure 2 and three different methods. Only non-linear approaches feature results with a sufficient accuracy. Our method can be favorably compared with the Tsai approach and gives same results than the full scale non-linear minimization using the Levenberg-Marquartd algorithm (see Appendix B) even if the derivation of the Jacobian is very different.

method	criterion	12mm	6mm	c-00101
Toscani	mean	0.778	0.1830	0.4972
	std dev	0.9772	0.2257	0.5620
	max	3.3434	0.6230	1.7121
Tsai	mean	0.105	0.231	0.1190
	std dev	0.059	0.194	0.0636
	max	0.297	1.798	0.3206
VVS	mean	0.094	0.231	0.1159
	std dev	0.053	0.196	0.0642
	max	0.328	1.819	0.3119

Table 5: Comparison between calibration methods

4.2 Multi image calibration

For each image, the parameters are initialized with the parameters computed for one of the images using the process presented in the previous section. The multi image calibration process is then considered to achieve a global optimization. This approach allows to greatly improves the quality of the calibration (see Table 6). If the global mean error may increase, the global mean error in mosy of the images decreases significantly as can be seen on Figure 6. Furthermore the convergence of the algorithm is quite fast: less than 10 iterations (see Figure 6).

image	u_0	v_0	p_x	p_y	K_d	residual
1	382.327	234.100	1149.49	1150.06	-6.325e-08	0.0938
2	383.157	234.676	1150.42	1151.38	-6.123e-08	0.0904
3	383.352	237.415	1154.62	1154.70	-5.360e-08	0.1076
4	383.259	235.726	1150.76	1150.92	-6.292e-08	0.0853
5	383.485	236.712	1152.96	1153.00	-5.709e-08	0.0948
all	383.470	234.891	1149.59	1150.04	-6.281e-08	0.0985

Table 6: Mono and multi image calibration using different views of the same calibration rig. We can observe some variability in the estimation of the intrinsic parameters that validates the necessity to consider multiple images in a calibration process.

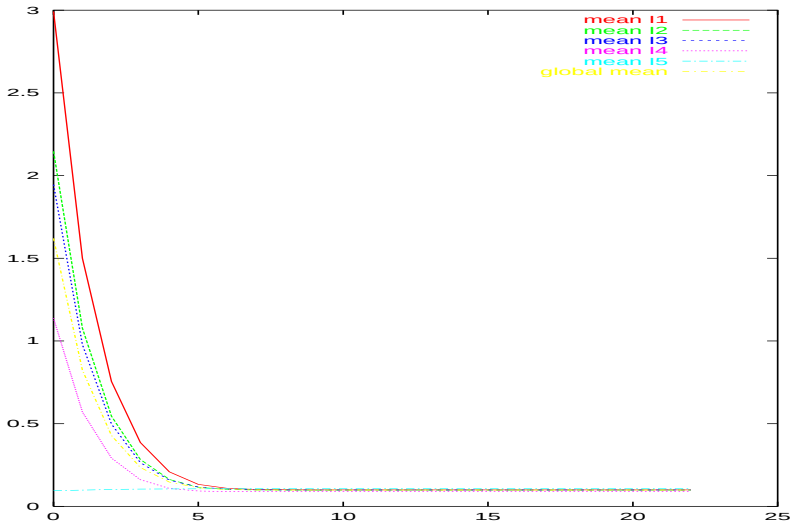

Figure 6: Multi image camera calibration: Evolution of the mean error (in pixel) in each image during the calibration process

5 Conclusion

We proposed in this paper an original formulation of the camera calibration problem. It consists in modifying using the visual servoing paradigm the parameters of a virtual camera (position, orientation, and intrinsic parameters) in order to register the back projection of the calibration rig with the data extracted from the images. We presented experimental results obtained using several cameras, lens, and calibration rig. Even if our approach seems to give better results in most cases, the thin difference between our approach and the other non-linear approaches makes a fair comparison not so obvious. However, our algorithm features many advantages:

- this algorithm is really simple. To illustrate this point, we give in appendix B the basic ideas leading to the derivation of the Jacobian in the usual case. Furthermore, we give in Appendix C a 20 lines source code of the virtual servoing closed-loop that compute the camera calibration proposed in this paper ;
- despite this simplicity, the obtained results may be favorably compared to best algorithms currently available ;

- with respect to Tsai's method it scales easily to more complex camera models (see Appendix A for the derivation of the image Jacobian related to the more complex camera model presented in [11]) ;
- Furthermore, it allows the integration of multiple images within the same calibration process in order to get a better estimate of the intrinsic parameters.
- last, and this is certainly the main advantage of this approach, we can consider various kinds of feature within the same calibration algorithm since the image Jacobian related to most of the geometrical primitive can analytically be computed [2]. To experimentally validate this point is one of the perspective of this work.
- We considered in this paper the calibration issue, It is obvious that the same approach provides a very efficient pose computation method if the intrinsic parameters are considered to be known.

A More complete camera model

In photogrammetry, we found in [1, 11] more complete camera models. Other type of distortion are added: thin prism distortion (e.g., slight tilt of the image array) and decentering distortion (e.g., the optical center of lens elements are not strictly collinear). This added four new variables in the distortion model g_1, g_2, g_3 and g_4. The new expression for distortion (equation (13)) becomes:

$$\begin{cases} \delta_u(u,v) = K_d\, r^2\, \tilde{u} + (g_1+g_2)\tilde{u}^2 + g_4\tilde{u}\tilde{v} + g_1\tilde{v}^2 \\ \\ \delta_v(u,v) = K_d\, r^2\, \tilde{v} + g_2\tilde{u}^2 + g_3\tilde{u}\tilde{v} + (g_2+g_4)\tilde{v}^2 \end{cases} \tag{20}$$

According to these new equations, the new interaction matrix is easily obtained form :

$$\begin{array}{ll}
\frac{\partial u}{\partial u_0} = 1 - K_d\, r^2 - 2K_d\, \tilde{u}^2 - 2(g_1+g_3)\tilde{u} - g_4\tilde{v} & \\
\frac{\partial u}{\partial v_0} = -2\, K_d\, \tilde{v}\tilde{u} - 2g_2\tilde{u} - g_3\tilde{v} & \\
\frac{\partial v}{\partial u_0} = -2\, K_d\, \tilde{u}\tilde{v} - g_4\tilde{v} - 2g_1\tilde{u} & \\
\frac{\partial v}{\partial v_0} = 1 - K_d\, r^2 - 2\, K_d\, \tilde{v}^2 - 2(g_2+g_4)\tilde{v} - g_3\tilde{u} & \\
\frac{\partial u}{\partial p_x} = x & \frac{\partial v}{\partial p_x} = 0 \\
\frac{\partial u}{\partial p_y} = 0 & \frac{\partial v}{\partial p_y} = y \\
\frac{\partial u}{\partial K_d} = \tilde{u}\, r^2 & \frac{\partial v}{\partial K_d} = \tilde{v}\, r^2 \\
\frac{\partial u}{\partial g_1} = r^2 & \frac{\partial v}{\partial g_1} = 0 \\
\frac{\partial u}{\partial g_2} = 0 & \frac{\partial v}{\partial g_2} = r^2 \\
\frac{\partial u}{\partial g_3} = \tilde{u}^2 & \frac{\partial v}{\partial g_3} = \tilde{u}\tilde{v} \\
\frac{\partial u}{\partial g_4} = \tilde{u}\tilde{v} & \frac{\partial v}{\partial g_4} = \tilde{v}^2
\end{array} \tag{21}$$

The other part $\frac{\partial \mathbf{p}}{\partial \mathbf{r}}$ of the interaction matrix remains unchanged.

B Comparison with a classical non-linear approach

For comparison issue we give in this appendix the derivation of the Jacobian related to the classical discrete full-scale non-linear approach considering no radial distortion in the camera model.

Rewriting the pose ${}^c\mathbf{M}_o$ as follows:

$${}^c\mathbf{M}_o = \begin{pmatrix} a_{11} & a_{12} & a_{13} & T_x \\ a_{21} & a_{22} & a_{23} & T_y \\ a_{31} & a_{32} & a_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{22}$$

the camera model is given by:

$$u = u_0 + p_x \frac{a_{11}X + a_{12}Y + a_{13}Z + T_x}{a_{31}X + a_{32}Y + a_{33}Z + T_z} \tag{23}$$

$$v = v_0 + p_y \frac{a_{21}X + a_{22}Y + a_{23}Z + T_x}{a_{31}X + a_{32}Y + a_{33}Z + T_z} \tag{24}$$

The Jacobian related to the intrinsic parameter and to the translation are equivalent to the Jacobian presented in the previous section. However, the Jacobian related to the rotation is more complex.

To use a classical iterative minimization method, we have to select a minimal representation for the rotation. The most common choice is $\mathbf{r} = \theta\mathbf{u}$ where θ is the angle and $\mathbf{u}$ the unit axe. More precisely we have:

$$\mathbf{r}_k = \theta\mathbf{u}_k \quad \text{with} \quad k = \{x, y, z\} \tag{25}$$

$$\theta = \sqrt{r_x^2 + r_y^2 + r_z^2} \qquad \text{and} \qquad \mathbf{u}_k = \frac{\mathbf{r}_k}{\theta} \tag{26}$$

This representation is related to the rotation matrix ${}^c\mathbf{R}_o$ by the relation

$${}^c\mathbf{R}_o = \cos\theta\mathbf{I}_{3\times3} + (1 - \cos\theta)\mathbf{u}\mathbf{u}^T + \sin\theta\tilde{\mathbf{u}} \tag{27}$$

where $\tilde{\mathbf{u}}$ is the skew matrix related to $\mathbf{u}$.

The x-position u of a point in the image is thus function of a_{ij} which depends of θ and $\mathbf{u}_k$. Therefore computing the Jacobian of u with respect to r_x, r_y and r_z leads to[3]:

$$\frac{\partial u}{\partial r_l} = \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial u}{\partial a_{ij}} \left(\frac{\partial a_{ij}}{\partial \theta}\frac{\partial \theta}{\partial r_l} + \sum_{k\in\{x,y,z\}} \frac{\partial a_{ij}}{\partial \mathbf{u}_k}\frac{\partial \mathbf{u}_k}{\partial r_l} \right) \tag{28}$$

$\frac{\partial u}{\partial a_{ij}}$ is computed from equation (23), $\frac{\partial a_{ij}}{\partial \theta}$ and $\frac{\partial a_{ij}}{\partial \mathbf{u}_k}$ are computed from equation (27). $\frac{\partial \theta}{\partial r_l}$ and $\frac{\partial \mathbf{u}_k}{\partial r_l}$ from equation (26).

Deriving the full Jacobian in this case is not in the scope of this paper. Indeed, from this point, the derivation is quite systematic, although it is error prone and the final result quite complex.

This illustrates the simplicity of our virtual visual servoing approach with respect to the classical discrete derivations. Simplicity is not the only advantage of our approach. As already stated the interaction matrix related to many geometrical features is now well known [2] while the discrete approach hardly scales to more complex geometrical features.

C VVS calibration source code

To illustrate the simplicity of the method, we propose here a pseudo-code implementation of our virtual visual servoing calibration algorithm. This code implement the calibration process presented in this paper. We have just assumed that we have a matrix computation library and that some operators may be overloaded (e.g., multiply

[3] The equation considering $\frac{\partial v}{\partial r_k}$ are similar and will no be derived here.

an homogeneous matrix by a point to perform a frame transformation). Vector `u` and `v` contains the coordinates of the observed points while `oP` contains the coordinates of the corresponding 3D points in the object frame.

```
Initialize(cMo,             // the pose
           px,py,u0,v0) ;   // the intrinsic camera parameters

for (i=0 ; i < n_points; i++)
{
    Pd[2*i]   = u[i] ;                        //     desired
    Pd[2*i+1] = v[i]  ;
}

while(abs(P-Pd) > eps)    // convergence test
{
  for (i=0 ; i < n_points; i++)
  {
    cP[i] = cMo * oP[i] ;    // compute position of each point
                             // in camera frame
    X = cP[i].X() ;        Y = cP[i].Y() ;      Z = cP[i].Z() ;
    x = X/Z  ;             y = Y/Z ;

    up = u[i] - u0 ;       vp = v[i] - v0 ;
    r2 = SQR(up)+SQR(vp) ;

    P[2*i] =    u0 + px*x  + K*r2*up ;       //    current
    P[2*i+1] =  v0 + py*y  + K*r2*vp ;

    // dP/dr                              // dP/dxi
    L[2*i][0] =  -px /Z ;                 L[2*i][6] = 1-K*(r2 +2*SQR(up)) ;
    L[2*i][1] =   0 ;                     L[2*i][7] = -2*K*up*vp ;
    L[2*i][2] =   px*x/Z ;                L[2*i][8] = x ;
    L[2*i][3] =   px*x*y ;                L[2*i][9] = 0 ;
    L[2*i][4] =  -px*(1+x*x)      ;       L[2*i][10] = up*r2;
    L[2*i][5] =   px*y ;

    L[2*i+1][0] =  0 ;                    L[2*i+1][6] = -2*K*up*vp ;
    L[2*i+1][1] =  py*(-1/Z) ;            L[2*i+1][7] = 1-K*(r2 +2*SQR(vp)) ;
    L[2*i+1][2] =  py*y/Z      ;          L[2*i+1][8] = 0;
    L[2*i+1][3] =  py* (1+y*y) ;          L[2*i+1][9] = y ;
    L[2*i+1][4] = -py*x*y ;               L[2*i+1][10] = vp*r2;
    L[2*i+1][5] = -py*x ;
  }

  Lp = (L.t()*L).inverse()*L.t() ;  // compute the pseudo inverse
```

```
  Tc = -lambda*Lp*(P-Pd) ;          // compute the control law

  u0 += Tc[6] ;    v0 += Tc[7] ;
  px += Tc[8] ;    py += Tc[9] ;
  K  += Tc[10] ;

  UpdatePose(Tc[0..6],cMo) ;
}
```

References

[1] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37:855–866, March 1971.

[2] 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.

[3] O.D Faugeras and G. Toscani. Camera calibration for 3-d computer vision. In *Proc International Workshop on Machine Vision and Machine Intelligence*, pages 240–247, Tokyo, February 1987.

[4] 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.

[5] Q.-T. Luong and O. Faugeras. Self calibration of a moving camera from point correspondences and fundamental matrices. *Int. Journal of Computer Vision, IJCV*, 22(3):261–289, 1997.

[6] S.J.. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. Journal of Computer Vision, IJCV*, 8(1):123–152, 1992 1992.

[7] P. Puget and T. Skordas. An optimal solution for mobile camera calibration. In *European Conf. on Computer Vision, ECCV'90*, pages 187–198, Antibes, France, April 1990.

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

[9] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.

[10] G.-Q. Wei and S.D. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE trans. on Pattern Analysis and Machine intelligence*, 16(5):469–480, May 1994.

[11] J Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE trans. on Pattern Analysis and Machine intelligence*, 14(10):965–980, October 1992.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399