

Calcul de pose et calibration par asservissement visuel virtuel : application à la réalité augmentée

Éric Marchand, Jérôme Royan, François Chaumette

IRISA - INRIA Rennes

Campus de Beaulieu,

35042 Rennes Cedex, France

Email {Eric.Marchand, Francois.Chaumette}@irisa.fr

Résumé

Cet article présente une formulation originale pour le problème du calcul de pose et de la calibration des caméras. Nous proposons une approche reposant sur l'utilisation de l'asservissement visuel 2D. L'idée consiste à considérer la pose et la calibration 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 reprojction 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 différentes informations visuelles.

Abstract

This paper presents a new formulation for the pose computation and camera 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 simplicity, accuracy, efficiency, and scalability wrt. the camera model as well as wrt. to the features extracted from the image.

1 Introduction

Dans cet article nous considérons deux problèmes importants en vision par ordinateur : le calcul de pose et la calibration des caméras. La plupart des solutions proposées dans la littérature considèrent la pose et la calibration comme des problèmes de recalage qui consistent à déterminer la relation existant entre un ensemble de primitives 3D (en général des points mais aussi, plus rarement, d'autres primitives géométriques : droites, ellipses, ...) et leur projection 2D dans le plan image. Calculer la pose consiste à calculer la position et l'orientation de la caméra dans le repère de la scène alors que la calibration nécessite en plus l'estimation des paramètres intrinsèques de la caméra

(point principal, distance focale, distorsion, ...).

Différentes approches ont été développées pour estimer la position relative de la caméra par rapport à la scène. Les primitives géométriques considérées pour l'estimation de la pose sont, le plus souvent, des points [14, 13, 7] mais on trouve aussi des segments [9], des contours [18, 10] des coniques [25, 6], ou des objets cylindriques [8]. Peu d'approches considèrent cependant l'utilisation conjointe de plusieurs types de primitives (voir cependant [23] pour l'utilisation de points et de droites). Pour la calibration, là encore, la grande majorité des méthodes existantes repose sur l'extraction de points (e.g., [12, 2, 28, 30, 29]) mais il existe des approches utilisant des droites [3] ou des ellipses [6, 27]. Un autre facteur discriminant important est la méthode retenue pour réaliser le recalage entre les primitives 2D extraites des images et leurs correspondants 3D. Pour le calcul de pose, des méthodes purement géométriques (e.g., [14, 9]), ou numériques et itératives [7] ont été proposées. D'autres méthodes dites linéaires (e.g., [12]) reposent sur la résolution de systèmes linéaires aux moindres carrés pour estimer la pose et éventuellement les paramètres intrinsèques de la caméra. Ces méthodes ne fournissant généralement pas un résultat de très bonne qualité, les techniques de minimisation non-linéaire (e.g., [18] pour le calcul de pose ou [2, 4] pour la calibration) consistent à directement minimiser l'erreur entre les observations dans l'image et la projection du modèle de l'objet pour une pose/calibration donnée. La minimisation est généralement réalisée en utilisant des algorithmes numériques itératifs de type Newton-Raphson ou Levenberg-Marquardt. Le principal avantage de ces approches est la précision du résultat obtenu. En contrepartie, l'algorithme de minimisation est sensible aux minima locaux, et peut, dans certains cas critiques, diverger. C'est pourquoi, une bonne initialisation du vecteur de paramètres à estimer est souvent nécessaire. Pour pallier ces problèmes, des techniques multi-étapes (e.g., [28, 30, 29]), plus dédiées à la calibration, considèrent une estimation linéaire (en ajoutant des contraintes comme la contrainte de Tsai

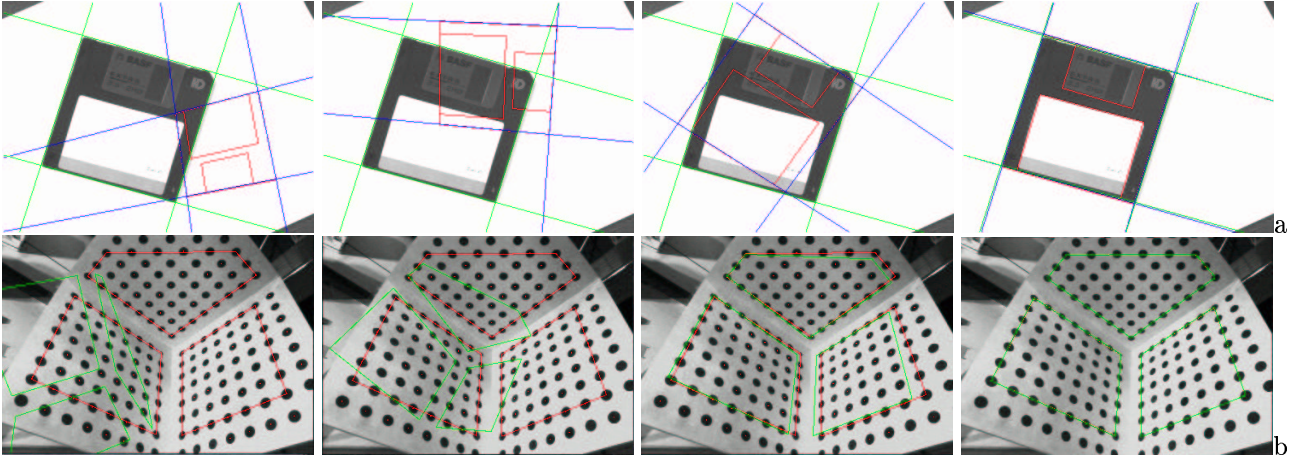


FIG. 1 – Pose et calibration par asservissement visuel virtuel: principe. L’objectif est de modifier itérativement la position de la caméra en utilisant une loi de commande classique en asservissement visuel afin de recaler les primitives visuelles désirées extraites de l’image avec les primitive courante calculer par reprojection du modèle 3D. Sur la première ligne, des droites sont utilisés pour calculer la pose. Sur la seconde ligne des points sont utilisés pour calculer à la fois la pose et les paramètres intrinsèques de la caméra.

sur l’alignement radial pour linéariser le problème) de certains paramètres et réalisent une estimation itérative des autres. Ces algorithmes autorisent une convergence plus rapide vers une bonne solution. Finalement, précisons que certaines techniques, dites d’autocalibration, ne nécessitent pas de mire de calibration [22, 19]. La méthode décrite dans cet article est une nouvelle formulation du calcul de pose et de la calibration utilisant un recalage non-linéaire global : l’asservissement visuel virtuel (AVV) [26]. Nous considérons en effet que l’asservissement visuel [15, 11] est le problème dual du calcul de pose et nous étendons cette approche à la calibration. Considérer ainsi le calcul de pose permet d’utiliser toutes les techniques issues de l’asservissement visuel (par exemple, les matrices d’interaction ou Jacobiens associés à un grand nombre de primitives sont en effet parfaitement connus). De plus il est aisé de considérer dans le même processus d’estimation différentes primitives géométriques.

Dans la suite de cet article, nous présenterons dans la Section 2 le principe de cette approche et son application au calcul de pose. Dans la Section 3 nous verrons comment étendre l’asservissement visuel à la calibration mono-image puis multi-images. Nous donnerons ensuite divers résultats pour le calcul de pose ou la calibration. Dans chacun des cas, nous considérerons aussi l’application particulière de la réalité augmentée.

2 Calcul de pose par asservissement visuel virtuel

2.1 Principe

L’idée fondamentale de notre approche est de définir le calcul de pose et la calibration comme le problème

dual de l’asservissement visuel 2D [11, 15]. L’objectif de l’asservissement visuel est de déplacer une caméra afin d’observer un objet à une position donnée dans l’image. Ceci est réalisé en minimisant l’erreur entre la position désirée des informations visuelles (\mathbf{p}_d) dans l’image et leur position courante (\mathbf{p}). Si le vecteur des informations visuelles est bien choisi, il y a une unique position de la caméra qui assure la régulation à zéro de cette erreur. Nous décrivons maintenant pourquoi le calcul de pose est un problème très similaire.

Pour illustrer notre propos, nous considérerons ici, le cas d’un objet constitué de points. Définissons une caméra virtuelle par ses paramètres intrinsèques ξ et ses paramètres extrinsèques donnés par la matrice homogène 4×4 ${}^c\mathbf{M}_o$. ${}^c\mathbf{M}_o$ définit la pose, c’est-à-dire la position de l’objet dans le repère de la caméra. La position d’un point ${}^c\mathbf{P}$ dans le repère de la caméra est donnée par :

$${}^c\mathbf{P} = {}^c\mathbf{M}_o {}^o\mathbf{P} \quad (1)$$

où ${}^o\mathbf{P}$ définit les coordonnées homogènes du point dans un repère lié à l’objet. La projection de ce point dans l’image est donnée par:

$$\mathbf{p} = pr_\xi({}^c\mathbf{P}) = pr_\xi({}^c\mathbf{M}_o {}^o\mathbf{P}) \quad (2)$$

où $pr_\xi(\cdot)$ est le modèle de projection fonction des paramètres intrinsèques de la caméra ξ . L’objectif du processus de calcul de pose est de minimiser l’erreur entre les données observées et notées \mathbf{p}_d et la position des mêmes informations visuelles \mathbf{p} calculées par reprojection en fonction des paramètres extrinsèques et intrinsèques de la caméra (définie par l’équation 2). De manière à réaliser cette minimisation, nous déplaçons la caméra virtuelle (initialement en ${}^c\mathbf{M}_o$) en utilisant

une loi de commande classique en asservissement visuel. Quand la minimisation est réalisée, la position finale de la caméra est donnée par ${}^c f \mathbf{M}_o$ c'est à dire la pose. Cet exemple est illustré ici pour des points, pour d'autres types de primitives géométriques les équations (1) et (2) sont bien évidemment différentes mais le principe général reste identique.

Ce processus est illustré par la Figure 1.a en considérant des droites comme information visuelle. Nous détaillerons ce processus dans les paragraphes suivants et nous montrerons comment l'étendre à la calibration des caméras.

2.2 AVV et calcul de pose

Pour le calcul de pose, les paramètres intrinsèques de la caméra étant connus, nous considérerons que la position des informations visuelles est exprimée dans l'espace métrique. Nous noterons donc $\mathbf{p}_{m,d}$ le vecteur d'information visuel extrait de l'image et \mathbf{p}_m la valeur du même vecteur calculée par reprojection.

L'objectif est donc de minimiser l'erreur $\|\mathbf{p}_m - \mathbf{p}_{m,d}\|$. Nous définissons donc simplement l'erreur dans l'image \mathbf{e} par la simple équation :

$$\mathbf{e} = \mathbf{p}_m(\mathbf{r}) - \mathbf{p}_{m,d} \quad (3)$$

où \mathbf{r} est la pose.

Le déplacement des informations visuelles est relié à la vitesse \mathbf{T}_c de déplacement de la caméra par :

$$\dot{\mathbf{p}}_m = \frac{\partial \mathbf{p}_m}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \mathbf{L}_{p_m} \mathbf{T}_c \quad (4)$$

La matrice \mathbf{L}_{p_m} est classiquement appelée matrice d'interaction ou Jacobien image.

Si l'on souhaite une décroissance exponentielle de l'erreur \mathbf{e} , c'est à dire :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (5)$$

où λ est un scalaire qui définit la vitesse de convergence, on obtient finalement la loi de commande suivante :

$$\mathbf{T}_c = -\lambda \mathbf{L}_{p_m}^+ \mathbf{e} \quad (6)$$

où $\mathbf{L}_{p_m}^+$ est la pseudo inverse de la matrice \mathbf{L}_{p_m} ($\mathbf{L}^+ = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$ si \mathbf{L} est une matrice $m \times n$ de rang plein avec $m > n$).

2.3 Les matrices d'interaction

Tout type d'information visuelle peut *a priori* être utilisé dans cette loi de commande si l'on est capable d'établir la matrice d'interaction \mathbf{L}_{p_m} qui lui est associée. Dans [11], un cadre général permettant le calcul de \mathbf{L}_{p_m} a été proposé. C'est en fait le principal avantage de cette formulation par rapport à la formulation classique de minimisation non-linéaire. En effet, ces matrices d'interaction ont déjà été établies pour un grand nombre d'informations visuelles (points, droites,

segments, cercles, sphères, quadriques, distances, etc.). De plus différentes informations visuelles peuvent être combinées entre elles au sein du même processus de calcul de pose.

Nous rappelons ici les matrices d'interaction relatives aux trois primitives géométriques (le point, la droite, et le cercle) que nous utiliserons dans cet article.

Le cas du point. Considérons $\mathbf{M} = (X, Y, Z)^T$ les coordonnées d'un point dans le repère de la caméra. La projection perspective de ce point dans le plan image est donnée par $\mathbf{m} = (x, y)^T$ avec :

$$\begin{cases} x = X / Z \\ y = Y / Z \end{cases} \quad (7)$$

La matrice d'interaction \mathbf{L}_{p_m} qui lie le mouvement $\dot{\mathbf{p}}_m = (\dot{x}, \dot{y})$ d'un point $\mathbf{p}_m = (x, y)$ dans l'image à la vitesse \mathbf{T}_c de la caméra est bien connue [15][11] :

$$\mathbf{L}_{p_m} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \quad (8)$$

Précisons que si l'on utilise que des points, l'asservissement visuel virtuel est très similaire à la méthode proposée par Lowe [18].

Cas d'une droite. Une droite est définie par l'intersection de deux plans :

$$\begin{cases} A_1 X + B_1 Y + C_1 Z = 0 \\ A_2 X + B_2 Y + C_2 Z + D_2 = 0 \end{cases} \quad (9)$$

L'équation de la projection de cette droite dans le plan image est donnée par :

$$x \cos \theta + y \sin \theta - \rho = 0. \quad (10)$$

Il est alors possible de calculer la matrice d'interaction relative aux paramètres $\mathbf{p}_m = (\theta, \rho)$ définissant la droite. Elle est donnée par [11]:

$$\mathbf{L}_\theta = \begin{bmatrix} \lambda_\theta \cos \theta & \lambda_\theta \sin \theta & -\lambda_\theta \rho & \rho \cos \theta & -\rho \sin \theta & -1 \end{bmatrix}$$

$$\mathbf{L}_\rho = \begin{bmatrix} \lambda_\rho \cos \theta & \lambda_\rho \sin \theta & -\lambda_\rho \rho \\ (1 + \rho^2) \sin \theta & -(1 + \rho^2) \cos \theta & 0 \end{bmatrix} \quad (11)$$

où $\lambda_\theta = (A_2 \sin \theta - B_2 \cos \theta) / D_2$ et $\lambda_\rho = (A_2 \rho \cos \theta + B_2 \rho \sin \theta + C_2) / D_2$.

Cas d'un cercle. Un cercle est défini comme l'intersection d'une sphère et d'un plan. Son équation est donc donné par :

$$\begin{cases} (X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 - R^2 = 0 \\ A(X - X_0) + B(Y - Y_0) + C(Z - Z_0) = 0 \end{cases} \quad (12)$$

Sa projection dans le plan image est une ellipse dont l'une des paramétrisations est $\mathbf{p}_m = (x_c, y_c, \mu_{02}, \mu_{20}, \mu_{11})$. x_c, y_c représente le centre de l'ellipse et μ_{02}, μ_{20} et μ_{11} sont ses moments centrés d'ordre 2. La matrice d'interaction relative à ces paramètres est donnée par l'équation (13) [11].

$$\mathbf{L}_{\mathbf{p}_m} = \begin{pmatrix} -1/Z_c & 0 & x_c/Z_c + A\mu_{20} + B\mu_{11} & x_c y_c + \mu_{11} & -1 - x_c^2 - \mu_{02} & y_c \\ 0 & -1/Z_c & y_c/Z_c + A\mu_{11} + B\mu_{02} & 1 + y_c^2 + \mu_{02} & -x_c y_c - \mu_{11} & -x_c \\ -2(A\mu_{20} + B\mu_{11}) & 0 & 2[(1/Z_c + Ax_c)\mu_{20} + Bx_c\mu_{11}] & 2(y_c\mu_{20} + x_c\mu_{11}) & -4\mu_{02}x_c & 2\mu_{11} \\ 0 & -2(A\mu_{11} + B\mu_{02}) & 2[(1/Z_c + By_c)\mu_{02} + Ay_c\mu_{11}] & 4y_c\mu_{02} & -2(y_c\mu_{11} + x_c\mu_{02}) & -2\mu_{11} \\ -A\mu_{11} - B\mu_{02} & -A\mu_{20}B\mu_{11} & Ay_c\mu_{20} + (3/Z_c - C)\mu_{11} + Bx_c\mu_{02} & 3y_c\mu_{11} + x_c\mu_{02} & -y_c\mu_{20} - 3x_c\mu_{11} & \mu_{02} - \mu_{20} \end{pmatrix} \quad (13)$$

3 Calibration et AVV

3.1 Principe

Il est possible, en utilisant la même approche de réaliser la calibration des caméras. Pour la pose, nous avons considéré que le vecteur d'information visuelle \mathbf{p} était exprimé dans l'espace métrique (noté alors \mathbf{p}_m). Pour la calibration, les paramètres intrinsèques de la caméra étant inconnus, \mathbf{p} ne peut être exprimé que dans l'espace pixel (nous le noterons donc \mathbf{p}_p). Dans ce cas l'erreur à minimiser est donnée par :

$$\mathbf{e} = \mathbf{p}_p(\mathbf{r}, \xi) - \mathbf{p}_{p_d} \quad (14)$$

Le déplacement des informations visuelles dans l'image est donc relié non seulement au déplacement de la caméra \mathbf{T}_c mais aussi à la variation temporelle des paramètres intrinsèques :

$$\dot{\mathbf{p}}_p = \mathbf{L}_{p_p} \mathbf{T}_c + \frac{\partial \mathbf{p}_p}{\partial \xi} \frac{d\xi}{dt} \quad \text{avec} \quad \mathbf{L}_{p_p} = \frac{\partial \mathbf{p}_p}{\partial \mathbf{p}_m} \mathbf{L}_{p_m} \quad (15)$$

qui peut se réécrire en :

$$\dot{\mathbf{p}}_p = \mathbf{H}_p \mathbf{V} \quad \text{avec} \quad \mathbf{V} = \begin{pmatrix} \mathbf{T}_c \\ \dot{\xi} \end{pmatrix} \quad (16)$$

La nouvelle matrice d'interaction est alors donnée par :

$$\mathbf{H}_p = \left(\frac{\partial \mathbf{p}_p}{\partial \mathbf{p}_m} \mathbf{L}_{p_m} \quad \frac{\partial \mathbf{p}_p}{\partial \xi} \right) \quad (17)$$

et la loi de commande assurant la minimisation de l'erreur est donnée par :

$$\mathbf{V} = -\lambda \mathbf{H}_p^+ \mathbf{e}. \quad (18)$$

3.2 Les matrices d'interaction

Nous donnons ici les matrices d'interaction relatives à la calibration pour le point et le cercle. Celles relatives à la droite ou à toutes autres primitives géométriques peuvent être calculées de façon similaire.

Cas du point Dans ce paragraphe nous considérons un modèle de caméra utilisé en photogrammétrie [2] avec un seul terme de distorsion : la distorsion radiale. Cependant prendre en compte des modèles plus complexes ne pose aucun problème (voir [21]) mais l'expérience montre que leur utilisation n'améliore que faiblement la qualité des résultats.

Si l'on note (u, v) la position d'un point dans l'espace pixel, cette position est reliée à la position du même point dans l'espace normalisé par :

$$\begin{cases} u = u_0 + p_x x + \delta_u(u, v) \\ v = v_0 + p_y y + \delta_v(u, v) \end{cases} \quad (19)$$

où δ_u et δ_v sont des termes de distorsion géométrique introduits dans le modèle de la caméra. δ_u et δ_v peuvent être modélisés par :

$$\begin{cases} \delta_u(u, v) = K_d r^2 \tilde{u} \\ \delta_v(u, v) = K_d r^2 \tilde{v} \end{cases} \quad (20)$$

avec $\tilde{u} = u - u_0$, $\tilde{v} = v - v_0$ et $r^2 = \tilde{u}^2 + \tilde{v}^2$. Les cinq paramètres à estimer (en plus de la position de la caméra) sont donc $\xi = \{p_x, p_y, u_0, v_0, K_d\}$.

Nous devons donc établir la matrice d'interaction \mathbf{H}_p qui lie le mouvement $\dot{\mathbf{p}} = (\dot{u}, \dot{v})$ du point $\mathbf{p} = (u, v)$ dans l'image à $[\mathbf{T}_c \quad \dot{\xi}]^T$. \mathbf{L}_{p_p} est une matrice 2×6 et $\frac{\partial \mathbf{p}}{\partial \xi}$ est matrice 2×5 . Plus précisément :

$$\mathbf{L}_{p_p} = \begin{pmatrix} p_x & 0 \\ 0 & p_y \end{pmatrix} \mathbf{L}_{p_m} \quad (21)$$

où \mathbf{L}_{p_m} est donnée par l'équation (8)

De plus, à partir de (19), dériver u et v par rapport à ξ permet d'établir simplement :

$$\frac{\partial \mathbf{p}}{\partial \xi} = \begin{pmatrix} x & 0 & 1 - K_d(r^2 + 2\tilde{u}^2) & -2K_d \tilde{u}\tilde{v} & \tilde{u} r^2 \\ 0 & y & -2K_d \tilde{u}\tilde{v} & 1 - K_d(r^2 + 2\tilde{v}^2) & \tilde{v} r^2 \end{pmatrix} \quad (22)$$

Cas d'un cercle. Pour ne pas alourdir les équations, nous ne considérerons pas ici de distorsion dans le modèle de caméra. Notons $(u_c, v_c, m_{20}, m_{02}, m_{11})$ les paramètres décrivant l'ellipse dans l'espace pixel. On montre qu'ils sont reliés à $(x_c, y_c, \mu_{20}, \mu_{02}, \mu_{11})$ (i.e., les même paramètres dans l'espace normalisé) par :

$$\begin{aligned} u_c &= u_0 + p_x x_c & v_c &= v_0 + p_y y_c \\ m_{20} &= \mu_{20} p_x^2 & m_{02} &= \mu_{02} p_y^2 & m_{11} &= \mu_{11} p_x p_y \end{aligned} \quad (23)$$

$\frac{\partial \mathbf{p}_p}{\partial \mathbf{p}_m}$ est donc une matrice diagonale 5×5 dont la diagonale est donnée par : $(p_x, p_y, p_x^2, p_y^2, p_x p_y)$. Pour l'es-

timisation des paramètres intrinsèques on calcule $\frac{\partial \mathbf{p}_p}{\partial \xi}$:

$$\frac{\partial \mathbf{p}_p}{\partial \xi} = \begin{pmatrix} (u - u_0)/p_x & 0 & 1 & 0 \\ 0 & (v - v_0)/p_y & 0 & 1 \\ 2m_{20}/p_x & 0 & 0 & 0 \\ 0 & 2m_{02}/p_y & 0 & 0 \\ m_{11}/p_x & m_{11}/p_y & 0 & 0 \end{pmatrix} \quad (24)$$

3.3 Calibration multi-image

Les paramètres intrinsèques issus d'une calibration mono-image peuvent être, en pratique, très différent des paramètres calculés à partir d'une seconde image prise d'un autre point de vue et ce avec la même caméra, le même objectif, la même carte de numérisation, et la même mire de calibration (voir les résultats présentés dans le tableau 6). Il est donc important de considérer plusieurs images dans le processus de calibration et d'intégrer l'ensemble des données disponibles dans le processus d'estimation.

Il convient donc de calculer un jeu unique de paramètres intrinsèques pour l'ensemble des images (c'est à dire pour l'ensemble des positions différentes de la caméra). Puget et Skordas [24] ont proposé un processus de ce type. Ils calculaient les paramètres intrinsèques finaux comme la moyenne des paramètres calculés pour chaque image. Les paramètres extrinsèques étant alors recalculés en fonctions des nouveaux paramètres intrinsèques. L'approche utilisée ici est différente. Nous utilisons un processus d'asservissement visuel virtuel qui a à calculer le déplacement de n caméras virtuelles et la variations des l paramètres intrinsèques supposés être les mêmes pour les n images. Pour n images, nous avons donc $6n + l$ variables à estimer $\sum_{i=1}^n m_i$ équations (où m_i est le nombre d'informations visuelles observées dans la i^{th} image). Une méthode similaire est présentée dans [17].

Si \mathbf{p}^i est l'ensemble des informations visuelles extraites de l'image i , la matrice d'interaction utilisée dans ce processus de calibration est donnée par la relation :

$$\begin{pmatrix} \dot{\mathbf{p}}^1 \\ \dot{\mathbf{p}}^2 \\ \vdots \\ \dot{\mathbf{p}}^n \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{L}_p^1 & 0 & \dots & 0 & \frac{\partial \mathbf{p}^1}{\partial \xi} \\ 0 & \mathbf{L}_p^2 & \dots & 0 & \frac{\partial \mathbf{p}^2}{\partial \xi} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \mathbf{L}_p^n & \frac{\partial \mathbf{p}^n}{\partial \xi} \end{pmatrix}}_{\mathbf{H}} \begin{pmatrix} \mathbf{T}_c^1 \\ \mathbf{T}_c^2 \\ \vdots \\ \mathbf{T}_c^n \\ \xi \end{pmatrix} \quad (25)$$

La minimisation est ensuite réalisée en utilisant la même approche que précédemment :

$$\begin{pmatrix} \mathbf{T}_c^1 \\ \mathbf{T}_c^2 \\ \vdots \\ \mathbf{T}_c^n \\ \xi \end{pmatrix} = -\lambda \mathbf{H}^+ \begin{pmatrix} \mathbf{p}^1 - \mathbf{p}_d^1 \\ \mathbf{p}^2 - \mathbf{p}_d^2 \\ \vdots \\ \mathbf{p}^n - \mathbf{p}_d^n \end{pmatrix} \quad (26)$$

4 Résultats expérimentaux

4.1 Calcul de pose

Dans ce paragraphe, nous présentons les résultats relatifs au calcul de pose. Nous comparons dans un premier temps la méthode présentée avec les algorithmes classiques de calcul de pose puis nous considérons une application particulière : la réalité augmentée.

Pose à partir de points Dans un premier temps, nous avons réalisé une comparaison entre notre algorithme et les principaux algorithmes "classiques" de calcul de pose : une estimation linéaire de la pose, une estimation numérique itérative par l'algorithme de Dementhon [7], une estimation non-linéaire en utilisant une minimisation de type Levenberg-Marquardt [18]. Dans cette expérience, la cible est faite de quatre

méthode	mean error (mm)	
	image 1	image 2
linéaire	7.3441	1.1402
Dementhon	1.189	5.6942
Lowe	1.1425	0.7641
AVV	1.1316	0.7463

TAB. 1 – Calcul de pose : comparaison avec trois algorithmes classique de calcul de pose

points coplanaires. Nous avons considéré deux images. Pour la seconde, le plan de l'image est presque parallèle au plan de la cible. Le tableau 1 affiche le résidu final ($\|\mathbf{p} - \mathbf{p}_d\|$) pour chacune des quatre méthodes. Si la qualité des résultats obtenus par la méthode de l'asservissement visuel virtuel sont très similaires de ceux obtenus par la minimisation non-linéaire par Levenberg-Marquardt, ils sont, comme on pouvait l'attendre, bien meilleurs que ceux obtenus par une approche linéaire ou par l'algorithme de Dementhon. Une analyse plus complète du comportement de cet algorithme sera proposée dans le paragraphe 4.2 en considérant le problème de la calibration.

Application à la réalité augmentée L'application considérée ici est la réalité augmentée. Cette application est intéressante pour juger de la qualité d'un algorithme de calcul de pose. En effet, bien que quelques algorithmes supposent que les paramètres de la caméra ne sont pas disponibles [16], la plupart des systèmes actuels de réalité augmentée repose sur l'utilisation de la calibration et la pose de la caméra.

La figure 2 montre les résultats de l'incrutation d'objets virtuels dans deux images après calcul de pose. Cette pose est calculée en utilisant comme information visuelle quatre points et un cercle. La séquence initiale est composée de 1600 images et la pose est calculée à la cadence vidéo. La statue qui "augmente" la séquence ainsi que le polygone texturé sont très stable dans toute la séquence. Il reste que, dans ce cas, l'extraction des primitives visuelles est très simple.



FIG. 2 – Pose à partir de point et de cercle pour une application de réalité augmentée

Pour confronter notre approche à des scènes plus complexes, nous avons considéré des images “réelles” acquises avec un caméscope classique. Dans cette première expérience dont les résultats sont montrés sur la figure 3, les primitives visuelles considérées dans le processus de calcul de pose sont des droites et des cercles. Quatre droites non coplanaires et trois cercles (sur la porte) ont été utilisés. Le modèle de la scène a été réalisé à la main à partir de plans du bâtiment et les primitives sont extraites et suivies dans la séquence en utilisant l’algorithme des ECM [1, 20]. Après le calcul de pose, la séquence a été augmentée avec plusieurs objets.



FIG. 3 – Calcul de pose à partir de droites et de cercles

Nous avons aussi considéré un contexte plus industriel (voir Figure 4). La caméra est située sur l’effecteur d’un robot à 6 degrés de liberté. Les primitives utilisées dans le processus de calcul de pose par asservissement visuel virtuel sont deux cylindres (nous n’avons pas présenté dans cet article la matrice d’interaction relative au cylindre, elle est disponible dans [11]) et une droite. Le suivi est, là encore, effectué en utilisant l’algorithme des ECM. La pose est calculées en moins de 3 ms et l’ensemble du processus d’augmentation (suivi, calcul de pose et rendu) est ici réalisé en temps réel.

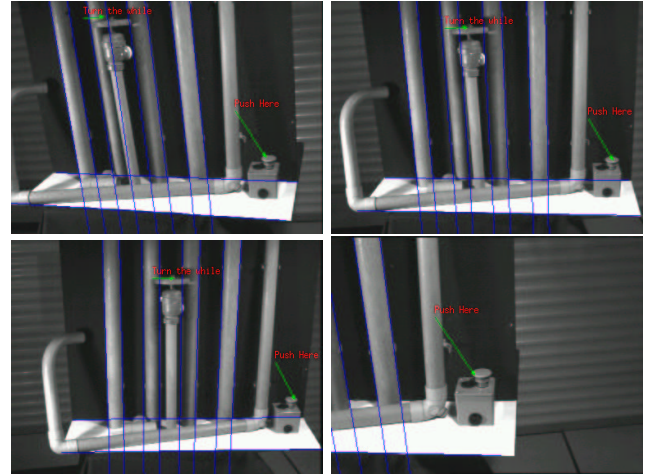


FIG. 4 – Calcul de pose et réalité augmentée à partir de cylindres et de droites

4.2 Calibration

Considérons maintenant le cas de la calibration de la caméra.

Comportement de l’algorithme. L’algorithme de calibration par asservissement visuel virtuel a été testé sur différentes images : images d’une mire de calibration acquises avec des objectifs de 6 et 12.5 mm et des images provenant du “*Calibrated Imaging Laboratory*” au CMU (voir figure 5).

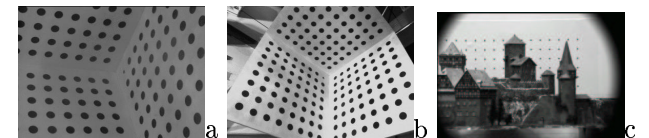


FIG. 5 – Images utilisées pour la calibration (a) image de notre mire de calibration acquise avec un objectif de (a) 12.5 mm et (b) 6mm. (c) image de la base de donnée du CIL à CMU

Pour illustrer le comportement de notre algorithme nous avons, dans un premier temps, considéré la calibration d’une caméra équipée d’un objectif de 6mm qui présente une importante distorsion radiale. La mire de calibration utilisée est composée de 192 points sur trois plan orthogonaux (voir figure 5.b). Seul 170 des 192 points sont visibles sur l’image utilisée. Les coordonnées de chaque point sont extraites avec une précision de 0.1 pixel. La figure 6 montre le résidu en pixel calculé à chaque itération de la minimisation. Comme spécifié par l’équation (5), cette erreur décroît de manière exponentielle. Dans cette expérience nous avons considéré dans un premier temps un modèle de caméra sans distorsion afin d’obtenir une première estimation des paramètres extrinsèques et intrinsèques. La solution trouvée a ensuite été améliorée en utilisant le modèle complet de caméra. L’amélioration de

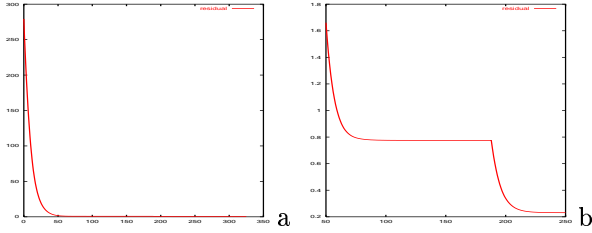


FIG. 6 – Résidu en pixel (la distorsion radiale est considérée après l’itération 180) (a) iteration 0 à 300 (b) iteration 50 à 250

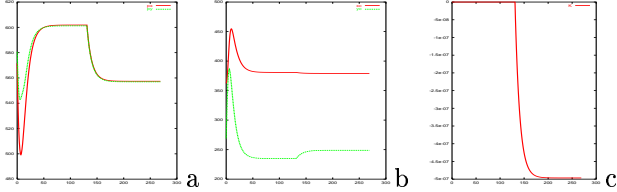


FIG. 7 – Évolution des paramètres intrinsèques (a) p_x et p_y (b) u_0 et v_0 (c) K_d

la minimisation de l’erreur en considérant la distorsion est visible sur la courbe de la figure 6 après l’itération 180.

La figure 7 montre l’évolution des paramètres intrinsèques de la caméra pendant le processus de minimisation. Cette minimisation est rapide : 161 itérations avec $\lambda = 0.1$ (ce coefficient λ contrôle la vitesse de convergence de l’algorithme, 13 itération seulement sont nécessaire avec $\lambda = 1$).

Convergence. L’initialisation est souvent considérée comme le problème principal de ce type d’approche. En effet, les méthodes d’optimisation non-linéaire comme Gauss-Newton, Newton-Raphson ou Levenberg-Marquardt requièrent une initialisation “correcte” (i.e., proche de la solution) des paramètres à estimer. Il en est de même pour notre approche. Cependant les tests réalisés montrent que, pour cette application, des erreurs importantes sur le vecteur initial de paramètres peuvent être considérées (voir les tableaux 2 et 3). Ces résultats montrent que, même avec d’importantes erreurs dans l’initialisation de la pose et/ou des paramètres intrinsèques, l’algorithme converge vers la même solution dans la grande majorité des cas. Cependant, dans quelques cas, il diverge (ligne où NC apparaît après le résidu initial).

Il reste qu’en pratique, pour ces approches de minimisation non-linéaire (AVV ou Levenberg-Marquardt), l’initialisation n’est pas un problème fondamental. Considérer les données constructeurs comme initialisation fournit, si ce n’est une bonne calibration, pour le moins une bonne initialisation de l’algorithme. De plus ce problème peut être totalement résolu en initialisant la minimisation par les résultats d’une calibra-

Position initiale						résidu initial
Tx	Ty	Tz	Rx	Ry	Rz	error
1	0.5	0.5	20	0	100	12129.5
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
3	3	3	0	0	0	909.97 (NC)
Position finale						résidu final
0.038	0.039	1.08	3.42	21.87	118.97	0.2308

TAB. 2 – Tests de convergence pour différentes positions initiales de la caméra virtuelle : ce tableau donne la position initiale de la caméra ainsi que l’erreur moyenne initiale en pixel). Les paramètres intrinsèques initiaux sont ceux donnés par le constructeur de la caméra. La dernière ligne du tableau donne la pose calculée avec le résidu final.

Paramètres intrinsèques initiaux					résidu initiale
u_0	v_0	p_x	p_y	K_d	
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
Paramètres intrinsèques finaux					résidu final
379.10	248.84	557.38	556.93	-4.96e-07	0.2308

TAB. 3 – Test de convergence pour différents jeux de paramètres intrinsèques initiaux. La pose initiale est ($T = (-0.2, 0.2, 1.3)$ et $R = (0, 0, 180)$). La dernière ligne du tableaux donne les paramètres intrinsèques calculés et le résidu finale.

tion réalisée par des algorithmes linéaire comme, par exemple, l’algorithme de Toscani-Faugeras.

Comparaisons avec d’autres algorithmes de calibration. Nous avons comparé notre approche avec d’autres algorithmes classiques de calibration : l’approche de Toscani-Faugeras [12], l’approche de Tsai [28], une méthode reposant sur minimisation non-linéaire avec une minimisation de type Levenberg-Marquardt, et notre approche non-linéaire reposant sur l’asservissement visuel virtuel.

objectif 6mm	p_x	p_y	u_0	v_0	résidu
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
AVV	557.38	556.93	379.10	248.84	0.2308

TAB. 4 – Comparaison de différents algorithmes de calibration pour une caméra équipée d’un objectif de 6mm. Nous avons considéré ici les algorithmes de Toscani-Faugeras, de Tsai, une approche de minimisation non-linéaire (Levenberg-Marquardt), et l’algorithme de minimisation non-linéaire d’AVV

Le tableau 4 montre les résultats de la calibration en

utilisant ces différentes méthodes. Le tableau 5 montre le résidu final, l'écart type et l'erreur maximale pour la calibration à partir des 3 images de la Figure 4. Seules les méthodes non-linéaires donnent un résultat de bonne qualité. Notre approche peut être favorablement comparé à la méthode de Tsai et donne des résultats similaires à l'approche non-linéaire avec une minimisation par Levenberg-Marquardt (même si le calcul du Jacobien est très différent [5]).

méthode	critère	12mm	6mm	c-00101
Toscani	résidu	0.778	0.689	0.4972
	écart type	0.9772	0.2257	0.5620
	dev max	3.3434	0.6230	1.7121
Tsai	résidu	0.105	0.231	0.1190
	écart type	0.059	0.194	0.0636
	max	0.297	1.798	0.3206
VVS	résidu	0.094	0.231	0.1159
	écart type	0.053	0.196	0.0642
	max	0.328	1.819	0.3119

TAB. 5 – Comparaison entre différentes méthodes de calibration

Calibration multi-images Nous considérons ici le cas où plusieurs images sont disponibles pour réaliser la calibration. Pour chaque image, les paramètres intrinsèques et extrinsèques sont initialisés en utilisant l'approche de calibration par AVV mono-image. Le processus de calibration multi-image est ensuite considéré pour réaliser une minimisation globale. Cette approche permet une amélioration significative de la qualité de la calibration (voir tableau 6). Si le résidu augmente légèrement pour certaine image, le résidu global moyen diminue comme on peut le voir sur la figure 8 (l'algorithme réalise en fait un compromis). Notons aussi que la convergence est relativement rapide : moins de 10 itérations (voir figure 8).

image	u_0	v_0	p_x	p_y	K_d	résidu
1	382.33	234.10	1149.49	1150.06	-6.32e-08	0.0938
2	383.16	234.68	1150.42	1151.38	-6.12e-08	0.0904
3	383.35	237.41	1154.62	1154.70	-5.36e-08	0.1076
4	383.26	235.73	1150.76	1150.92	-6.29e-08	0.0853
5	383.49	236.71	1152.96	1153.00	-5.71e-08	0.0948
toutes	383.47	234.89	1149.59	1150.04	-6.28e-08	0.0985

TAB. 6 – Calibration mono et multi-images en utilisant différentes vues de la même mire de calibration. Une certaine variabilité dans les paramètres intrinsèques estimés peut être observée ce qui milite pour un processus de calibration multi-image.

Calibration multi-primitives Dans les expériences précédentes, afin de pouvoir comparer notre approche avec d'autres algorithmes de la littérature, nous avons considéré uniquement des points comme information visuelle. L'un des avantages importants de notre approche est sa capacité à prendre en compte de façon évidente différents types de primitives. Dans cette expérience, nous avons considéré à la fois des

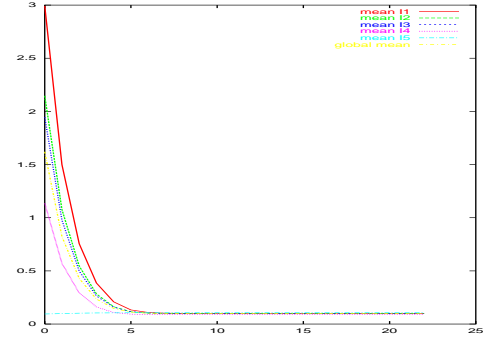


FIG. 8 – Calibration multi-images: évolution du résidu (en pixel) pour chaque image

points et des cercles et nous chercherons à estimer les paramètres intrinsèques suivants : u_0, v_0, p_x, p_y . Comme dans le cas du calcul de pose nous sommes intéressés ici par une application de réalité augmentée. Trois points et trois cercles sont suivis dans cette séquence d'images. Pendant l'acquisition de la séquence nous avons fait varier la distance focale de la caméra de façon significative.

Bien que la qualité de la calibration ne puisse en aucun cas être comparée avec les résultats obtenus dans le paragraphe précédent (à cause du faible nombre de primitives considéré) l'expérience montre (voir figure 9) que celle-ci est très suffisante pour des applications de réalité augmentée.

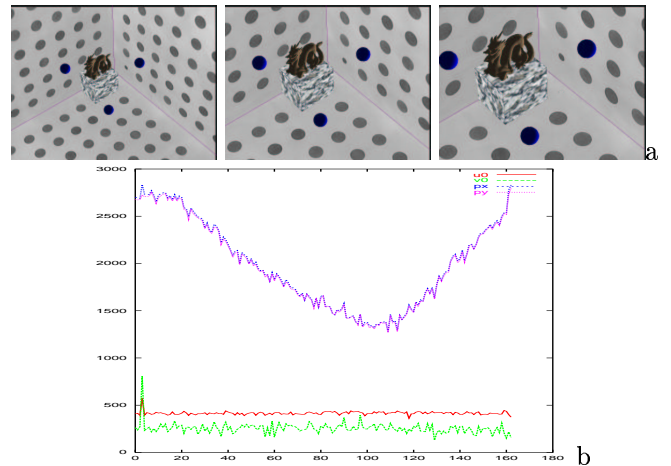


FIG. 9 – Calibration à partir de points et de cercles: application à la réalité augmentée (a) séquence d'images après incrustation d'objets virtuels (b) évolution des paramètres de calibration au cours de la séquence (le zoom du caméscope a été utilisé).

5 Conclusion

Nous avons présenté dans cet article une formulation originale du calcul de pose et de la calibration. Il est en effet possible de résoudre le recalage en modifiant les

paramètres d'une caméra virtuelle (position, orientation et éventuellement les paramètres intrinsèques) en utilisant une loi de commande classique en asservissement visuel. Nous avons présenté des résultats démontrant la validité de notre approche. Il faut bien reconnaître que cette approche est comparable avec les approches non-linéaires classiques tant d'un point de vue méthodologique que sur la qualité des résultats obtenus. Elle présente cependant un grand nombre d'avantages :

- d'abord, cet algorithme est réellement simple. Nous donnons dans [21] quelques lignes de code réalisant la calibration à partir de points présentée dans cet article ;
- malgré cette simplicité, les résultats obtenus sont très favorablement comparables aux meilleurs algorithmes disponibles actuellement (tant pour la pose que pour la calibration) ;
- Concernant le cas particulier de la calibration, contrairement à des méthodes comme celle de Tsai, il est aisé de modifier le modèle de la caméra (voir [21] pour l'utilisation du modèle plus complexe présenté dans [30]). De plus, il est possible de considérer aisément plusieurs images dans le même processus de calibration afin d'obtenir une meilleure estimation des paramètres intrinsèques ;
- enfin, cette méthode permet de considérer un ensemble hétérogène de primitives dans le processus de recalage. La matrice d'interaction nécessaire à l'AVV est évidente à établir. De plus le calcul de ces matrices pour d'autres cas que ceux présentés dans cet article est facilement réalisable (et déjà réalisé dans beaucoup de cas...)

Finalement, si nous avons considéré ici deux applications classiques en vision par ordinateur, cette approche devrait permettre de résoudre d'autres problèmes comme, par exemple, l'estimation non-linéaire de la matrice fondamentale (les informations visuelles à considérer seraient alors une distance entre des points et la droite épipolaire correspondante) ou le suivi d'objets [10].

Références

- [1] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 11(5):499-511, May 1989.
- [2] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 4(2):127-140, March 1971.
- [3] B. Caprile and V. Torre. Using vanishing points for camera calibration. *Int. J. of computer Vision*, pages 127-140, 1990.
- [4] F. Chaumette and P. Rives. Modélisation et calibration d'une caméra. In *7ème congrès AFCEC Reconnaissance des formes et intelligence artificielle, RFIA '89*, volume 1, pages 527-536, Paris, December 1989.
- [5] F. Chaumette and P. Rives. Vision-based-control for robotic tasks. In *IEEE Int. Workshop on Intelligent Motion Control*, pages 395-400, Istanbul, Turquie, August 1990.
- [6] S. de Ma. Conics-based stereo, motion estimation and pose determination. *Int. Journal 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 tracking of complex structures with on-line camera calibration. In *BMVC99*, 1999.
- [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] 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.
- [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] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-points problem. *Computer Vision, Graphics and Image Processing*, 47(1):33-44, July 1989.
- [15] 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.
- [16] K. Kutulakos and J. Vallino. Calibration-free augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):1-20, January 1998.
- [17] J.-M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration? In *European Conference on Computer Vision, ECCV'98*, volume 1, pages 158-174, Freiburg, Germany, June 1998.
- [18] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355-394, 1987.
- [19] Q.-T. Luong and O. Faugeras. Self calibration of a moving camera from point correspondances and fundamental matrices. *Int. Journal of Computer Vision, IJCV*, 22(3):261-289, 1997.

- [20] 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.
- [21] E. Marchand and F. Chaumette. A new formulation for non-linear camera calibration using virtual visual servoing. Technical Report 1366, IRISA, January 2001.
- [22] 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.
- [23] T.-Q. Phong, R. Horaud, A. Yassine, and P.-D. Tao. Object pose from a 2d to 3d point and line correspondance. *Int. J. of Computer Vision*, 15(3):225–243, July 1995.
- [24] 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.
- [25] R. Safae-Rad, I. Tchoukanov, B. Benhabib, and K.C. Smith. 3d-pose estimation from a quadratic-curved feature in two perspective views. In *IAPR Int. Conf. on Pattern Recognition, ICPR'92*, volume 1, pages 341–344, La Haye, Pays Bas, August 1992.
- [26] V. Sundareswaran and R. Behringer. Visual servoing-based augmented reality. In *IEEE Int. Workshop on Augmented Reality*, San Francisco, November 1998.
- [27] J.-P. Tarel and A. Gagalowicz. Calibration de caméra à base d'ellipses. *Traitement du Signal*, 12(2):177–187, 1995.
- [28] 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.
- [29] 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.
- [30] 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.