HAL
open science

Calibrage itératif de caméras à partir de scènes planes

Frédéric Jurie, Eric Noirfalise, Peter Sturm

To cite this version:

Frédéric Jurie, Eric Noirfalise, Peter Sturm. Calibrage itératif de caméras à partir de scènes planes. 14ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Jan 2004, Toulouse, France. pp.261-269. inria-00524413

HAL Id: inria-00524413

https://inria.hal.science/inria-00524413

Submitted on 25 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Calibrage itératif de caméras à partir de scènes planes

Iterative Camera Autocalibration based on Planar Scenes

Frédéric Jurie[1] Eric Noirfalise[1] Peter Sturm[2]

[1] LASMEA, UMR6602 du CNRS, Université Blaise Pascal
[2] INRIA Rhône-Alpes, Montbonnot, France

LASMEA,
UMR6602 du CNRS,
Université Blaise Pascal,
63177 Aubière Cedex
jurie@lasmea.univ-bpclermont.fr

INRIA
Zirst - 655 avenue de l'Europe,
Montbonnot,
38334 Saint Ismier Cedex
peter.sturm@inrialpes.fr

Résumé

Nous présentons une méthode permettant le calibrage de caméras vidéo à partir de séquences d'images d'un objet plan texturé, mais inconnu. Les méthodes permettant de répondre à ce problème procèdent généralement par optimisation d'un système non-linéaire global (utilisant toutes les images simultanément) alors que notre méthode est par nature séquentielle, puisque basée sur un filtre de Kalman. Le coût de traitement est très faible, ce qui permet de calibrer des caméras vidéo en temps réel, en utilisant directement le flux des images produites par la caméra. Le fait de pouvoir utiliser le flux vidéo efficacement est profitable, puisqu'un auto-calibrage stable requière que les images soient prises avec des angles de vue différents. Les méthodes globales sont limitées par le nombre d'images qu'elles peuvent utiliser et il faut donc soit choisir des vues clés, soit utiliser des séquences de vues fortement espacées ce qui rend la mise en correspondance difficile. Notre méthode ne souffre pas de ces limites, et peut donc traiter des séquences de plusieurs milliers d'images.
Bien que non-optimale (au sens de l'ajustement de faisceaux), notre méthode permet de traiter l'ensemble des informations d'une séquence. Des résultats expérimentaux montrent que, aussi bien en terme de précision que de convergence, notre méthode donne des résultats équivalents à un ajustement de faisceaux, et permet de plus l'auto-calibrage en temps réel.

Mots Clef

Vision par ordinateur, calibrage de caméras.

Abstract

We present a method for camera autocalibration from images of an unknown *planar object. The few existing methods for that problem proceed by non-linear batch optimization, whereas our method is of sequential nature, being based on Kalman filtering. Its overall computational cost is much smaller, thus enabling the use of video sequences as input for real-time, on-the-fly, autocalibration. Being able to use video efficiently is very beneficial, since a stable autocalibration requires images taken with significantly varying camera orientations : batch methods put limits on the number of images used, hence require either to select key frames from a video, or to use discrete image sequences, for which the matching is more difficult. Our method is not concerned by these issues.*
Although not optimal (in the bundle adjustment sense), our method compensates by using indeed all available information in an image sequence. Experimental results show that it performs similarly to a global batch method, in terms of accuracy and convergence, while allowing real-time autocalibration in practice.

Keywords

Computer vision, camera calibration.

1 Introduction

Le calibrage de caméra est une étape importante pour de nombreux algorithmes et applications de la vision par ordinateur, en particulier ceux visant à inférer des informations 3D telles que la structure d'une scène où le mouvement de la caméra par rapport à la scène.
Le calibrage consiste à déterminer les paramètres intrinsèques (distance focale, etc), généralement à partir de l'observation d'objets 3D ayant une structure géométrique connue (mires de calibrage).
Le concept *d'autocalibrage* [13], traite le problème du calibrage, mais à partir d'images de la scène ayant une géométrie complètement inconnue, et sans aucune information sur la trajectoire de la caméra ni sur les paramètres

intrinsèques[1].

Cela permet évidemment une plus grande flexibilité, puisque :

- le calibrage ne nécessite plus de mires de calibrage spécifiques qui sont généralement onéreuses et difficiles à concevoir,
- calibrer à partir des images provenant d'une application, sans avoir à interrompre cette application (imageurs embarqués sur des satellites par exemple),
- mise à jour en ligne des paramètres intrinsèques qui sont modifiés lors d'actions sur le zoom ou la mise au point de l'objectif.

Bien que la plupart des méthodes d'auto-calibrage requièrent des images de *scènes 3D*, il a pu être montré que même dans le cas de scènes *planes*, sans connaissance de leur structure, il est possible de résoudre ce problème [17].

L'un des avantages principaux d'utiliser une scène plane plutôt qu'une scène 3D est que la mise en correspondance d'indices visuels - nécessaire au calibrage - est plus simple : le problème des occultations ne se pose plus, même pour des images prises à des points de vue très différents, ce qui est un facteur clé pour la mise au point de méthodes d'auto-calibrage stables.

De plus, les surfaces planes étant omniprésentes dans l'environnement construit par l'homme, l'auto-calibrage à partir d'observations de telles surfaces peut constituer l'une des méthodes les plus pratiques et les plus flexibles.

Remarquons que le cas « intermédiare » qui consiste à calibrer des caméras à partir de structures planes mais de géométrie connue [20, 15, 4] est de nos jours utilisé quotidiennement.

Un travail proche de notre contexte mais pas directement lié est celui de l'auto-calibrage à partir de l'observation de plusieurs objets plans, de structure inconnue, et de position inconnue[18]. Dans ce cas, l'auto-calibrage « classique », c'est-à-dire celui basé sur l'utilisation de scènes 3D quelconques serait possible, mais il est montré dans [18] que le fait de prendre en compte la planéité par morceaux peut être profitable.

Nous allons dans un premier temps décrire les concepts et les méthodes déjà publiés, en rapport avec nos travaux. Nous pouvons distinguer deux approches principales pour résoudre le problème de l'auto-calibrage : les méthodes directes et les méthodes stratifiées.

- les méthodes directes d'autocalibrage, telle que celle présentée dans [2], consistent à estimer directement les paramètres intrinsèques et extrinsèques des caméras, en même temps que se fait l'estimation de la structure Euclidienne 3D de la scène.
- dans les méthodes stratifiées, [6, 14, 3, 17], ces paramètres sont estimés de manière indirecte. Généralement, une reconstruction projective de la scène et des caméras est calculée dans un premier temps, et l'auto-calibrage est ensuite obtenu en estimant une transformation qui rectifie la reconstruction projective en reconstruction Euclidienne.

Les travaux antérieurs sur l'auto-calibrage à l'aide de structures planes incluent la méthode directe proposée dans [1], laquelle est une méthode dérivée de [2]. Comme [2], la méthode estime la distance focale de la caméra et le mouvement de cette dernière de manière récursive, en utilisant un filtre de Kalman.

Les autres méthodes, stratifiées, sont du types de celles proposées dans [17, 11, 5]. Elles reposent sur une méthode d'optimisation globale, non-linéaire[2].

Un auto-calibrage fiable n'est possible que si les images sont prises à partir de différents points de vue (en particulier d'orientations différentes), pour éviter les mouvements critiques [16], et les instabilités numériques qui les entourent.

Puisque le coût algorithmique d'optimisation non-linéaire croît avec le nombre d'images utilisées, cela impose en pratique que :

- il faille prendre peu d'images, prises de points de vue très différents,
- ou bien il faille choisir quelques images clef issues d'une séquence vidéo.

Dans le premier de ces deux cas, la mise en correspondance d'indices est difficile et est plus coûteuse que dans le second. Dans le second cas il faut une méthode pour choisir les vues à retenir.

Notre approche se place dans cette problématique. Il s'agit d'une méthode stratifiée, mais qui comme dans [1], repose sur une estimation récursive.

Cela permet d'un côté d'utiliser des séquences vidéo comme entrée, ce qui simplifie la mise en correspondance, et d'un autre côté, cela évite d'avoir à sélectionner des vues clef, puisque le coût algorithmique est très faible, car toutes les images sont traitées séparément.

L'inconvénient potentiel d'une méthode récursive est bien entendu la possibilité qu'elle diverge, ou qu'elle ne converge pas vers les valeurs optimales, contrairement à une optimisation non-linéaire globale. En revanche, contrairement à une optimisation globale, notre méthode permet de traiter toutes les images d'une séquence, c'est-à-dire plusieurs centaines ou milliers d'images. Les expérimentations nombreuses que nous avons conduites nous montrent qu'il est possible d'obtenir des résultats similaires qu'avec une optimisation globale, mais en temps réel.

Dans cet article, nous supposerons que les paramètres intrinsèques sont constants.

La première étape de notre méthode, la reconstruction projective du plan observé est obtenue en plus, comme résultat indirect de la méthode, comme nous l'expliquons à la section 2. Dans la section 3, nous formulons notre problème d'auto-calibrage, qui exploite le lien entre la

[1] ce n'est pas entièrement vrai : au moins un des paramètres intrinsèques doit être connu ou doit être supposé constant [8]

[2] La méthode propose dans [11] consiste à résoudre alternativement deux sous-ensembles d'inconnues, ce qui revient à une optimization non-linéaire.

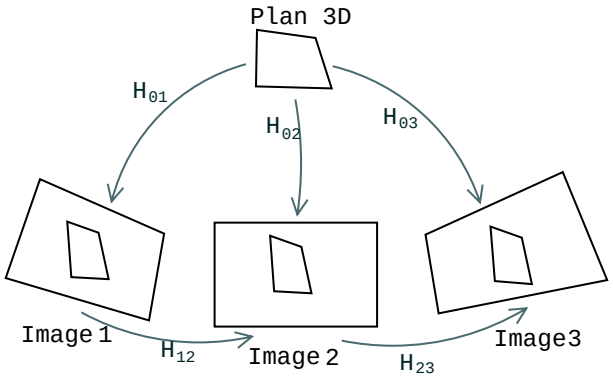

FIG. 1 – Le caractère plan de l'objet permet d'établir qu'il existe des relations homographiques entre le plan 3D et sa projection – les H_{0i} – mais aussi entre des paires d'images du plan (homographies inter-images H_{ij}).

structure Euclidienne de l'objet plan et les paramètres intrinsèques de la caméra. Ce lien est exprimé au moyen de transformations homographiques du plan entre les images, qui proviennent d'un algorithme de mise en correspondance, comme décrit dans la section 2.1. La section 4 décrit notre méthode d'auto-calibrage itérative. Enfin, des résultats expérimentaux sont donnés dans la section 5 Nous montrons également des comparaisons entre notre méthode et une méthode classique d'ajustement de faisceaux.

2 Reconstruction projective d'un objet plan

Dans cette section, nous considérons la première étape de notre approche stratifiée, qui est l'estimation de la structure projective plane de l'objet observé.

Il est bien connu que la projection perspective, lorsqu'elle se limite à la projection de points coplanaires peut être exprimée par une homographie (se référer par exemple à [7]). Pour chaque image i, il existe une homographie, c'est-à-dire une matrice 3×3, H_{0i}, qui transforme tout point $\mathbf{X}$ de l'objet plan en point de l'image, de coordonnées $\mathbf{x}_i$ (les deux points sont exprimés au moyen de leur coordonnées homogènes) :

$$H_{0i}\mathbf{X} \sim \mathbf{x}_i$$

Le symbole "$\sim$" exprime l'égalité à un facteur d'échelle près, de vecteurs ou de matrices.

Une des conséquences de ce qui est énoncé ci-dessus est qu'il existe aussi des homographies inter-images (entre des paires d'images) qui lient entre eux les points qui sont les projections des mêmes points 3D de l'objet plan :

$$\mathbf{x}_j \sim H_{ij}\mathbf{x}_i$$

où H_{ij} (voir aussi la figure 1) est donnée par :

$$H_{ij} \sim H_{0j}H_{0i}^{-1} \tag{1}$$

2.1 Reconstruction projective et mise en correspondance de primitives visuelles

A partir de ce qui précède, il découle que chaque image d'un objet plan constitue directement une reconstruction projective de cet objet : si l'on connaît H_{0i} pour plusieurs images i, il est alors possible de l'appliquer pour rectifier l'image et ainsi obtenir une image qui révèle la structure Euclidienne de l'objet.

Pour notre problème, les H_{0i} sont inconnues et ne peuvent être obtenues directement. Cependant, les homographies inter-images H_{ij} peuvent être calculées, par exemple à partir de la mise en correspondance de points d'intérêt (voir [7] pour trouver de tels algorithmes). En pratique, nous estimons les homographies entre images en utilisant l'algorithme de suivi similaire à celui décrit dans [9]. Ce système ne nécessite pas de mise en correspondance de points, il met directement en correspondance les niveaux de gris de deux images, lorsque le déplacement de la caméra entre les deux images n'est pas trop important. Précis, rapide, il est également résistant à des changements d'illumination de la scène et nous permet ainsi d'obtenir des homographies entre images en temps réel.

En utilisant l'équation (1), nous pouvons représenter « l'homographie de rectification » de toutes les vues au moyen d'une seule homographie. Sans perte de généralité, nous pouvons supposer que la première image est choisie comme image de référence, et exprimer les H_{0i}, pour $i \neq 1$, comme :

$$H_{0i} \sim H_{1i}H_{01}$$

Nous avons ainsi à estimer les homographies entre l'image 1 et les autres images. Cela peut se faire en accumulant les homographies entre paires d'images successives. En pratique, cependant, nous les estimons directement ; notre algorithme de suivi utilise la première image pour « apprendre » la texture [3] [9], et la sortie de l'algorithme de suivi, pour chaque nouvelle image, est l'homographie qui lie l'image courante a la première image (d'une certaine manière, c'est cette homographie qui est suivie). L'algorithme ne prend que quelques *ms* pour traiter chaque image, sur un PC standard.

Cette approche nécessite que chaque image ait un recouvrement important avec l'image initiale, ce qui est facile à garantir en pratique. De plus, cette approche évite tout risque de dérive, qui pourrait arriver si l'on accumulait les homographies.

Remarquons que les méthodes globales peuvent combiner les homographies inter-images pour effectuer des reconstructions projectives multi-vues, comme dans [17, 11, 19]. Cela peut être utile dans une phase de pré-traitement, en particulier si les homographies sont estimées à partir de correspondances de points.

[3] Cet apprentissage dure environ une seconde, après quoi le calibrage peut commencer.

3 Structure Euclidienne et Autocalibrage

La majorité des définitions suivantes peuvent être trouvées dans [7] (par exemple).

3.1 Modèle de caméra

Nous supposons que la caméra peut se modéliser à l'aide d'un modèle projectif basé sur la représentation sténopé.
La matrice de projection 3×4 appelée P , associée à ce modèle, regroupe les paramètres internes et externes de la camera :

$$P = KR(I_3|t) \tag{2}$$

R est une matrice orthogonale et t un vecteur de $\Re^3$ qui représentent respectivement l'orientation et la position de la caméra. La matrice de calibrage K contient les paramètres internes :

$$K = \begin{pmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

où (u_0, v_0) sont les coordonnées du point principal dans l'image, (f_u, f_v) représente la focale de la caméra, exprimée en pixel (dans les directions verticales et horizontales), et s représente le paramètre de *skew*.

3.2 Image de la Conique Absolue (ICA)

La Conique Absolue (CA) $\mathbf{\Omega}_\infty$ est une conique imaginaire du plan à l'infini π_∞. Les points du plan à l'infini $\mathbf{X} = (X_1, X_2, X_3, 0)^\top$ qui forment la Conique Absolue satisfont :

$$X_1^2 + X_2^2 + X_3^2 = 0 \Leftrightarrow \mathbf{X}^\top \mathbf{\Omega}_\infty \mathbf{X} = \mathbf{X}^\top I \mathbf{X} = 0$$

Etant donnée une matrice de projection $P = KR(I|t)$ et un point $\mathbf{X}_\infty = (X_1, X_2, X_3, 0)^\top$ sur π_∞, son image satisfait :

$$\begin{aligned} \mathbf{x} &\sim P\mathbf{X}_\infty \\ &= KR(I|t)(X_1, X_2, X_3, 0)^\top \\ &= KR\tilde{\mathbf{X}} \end{aligned}$$

où $\tilde{\mathbf{X}} = (X_1, X_2, X_3)^\top$.
Soit H l'homographie qui relie des points du plan à l'infini à leurs projections dans le plan image. En raison de la décomposition proposée, l'homographie H est donnée par $H = KR$.
La propriété d'othogonalité de $R^\top R = I$ peut être utilisée pour obtenir ω, l'Image de la Conique Absolue (ICA), sous la forme :

$$\omega = H^{-\top} I H^{-1} = (KR)^{-\top}(KR)^{-1} = (KK^\top)^{-1} \tag{3}$$

L'ICA est donc indépendante de la pose de la camera et ainsi n'est liée exactement qu'aux paramètres intrinsèques de la caméra. Plutôt que de déterminer directement la matrice de calibrage K, la majorité des algorithmes d'autocalibrage, y compris le nôtre, estiment d'abord ω, à partir de laquelle K peut alors être obtenue au moyen d'une décomposition de Cholesky.

3.3 Relation entre la matrice de projection et les homographies du plan

Considérons le plan 3D Π et son image. Sans perdre en généralité, nous supposons que Π est le plan donné par $Z = 0$. La matrice de projection de la i^{e} image peut être obtenue par :

$$P_i = KR_i(I|t_i)$$

Soit $\mathbf{X} = (X_1, X_2, 0, X_4)^\top$ un point de Π, et $\tilde{\mathbf{X}} = (X_1, X_2, X_4)^\top$ ses coordonnées homogènes dans ce plan.
Ses coordonnées dans l'image $\mathbf{x}_i$ peuvent être obtenues au moyen de :

$$\begin{aligned} \mathbf{x}_i \sim P_i\mathbf{X} &= KR_i \begin{pmatrix} 1 & 0 & 0 & \\ 0 & 1 & 0 & t_i \\ 0 & 0 & 1 & \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ 0 \\ X_4 \end{pmatrix} \\ &= KR_i \begin{pmatrix} 1 & 0 & \\ 0 & 1 & t_i \\ 0 & 0 & \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_4 \end{pmatrix} \end{aligned}$$

Dans les équations ci-dessus, nous pouvons identifier les matrices d'homographies 3×3 H_{0i}, introduites dans la section 2 :

$$H_{0i} = KR_i \begin{pmatrix} 1 & 0 & \\ 0 & 1 & t_i \\ 0 & 0 & \end{pmatrix} \tag{4}$$

3.4 Equations d'(Auto-) Calibrage

A partir de la forme des homographies de plans telle que donnée dans l'équation 4), des équations les liant aux paramètres intrinsèques uniquement (c'est-à-dire sans qu'elles ne dépendent des poses des caméras) peuvent être obtenues. Soit $\mathbf{h}_k$ le vecteur de dimension 3 contenant la k^{e} colonne de l'homographie H_{0i}.
Les équations qui suivent ont été démontrées dans [15, 20] :

$$\mathbf{h}_1^\top \omega \mathbf{h}_1 - \mathbf{h}_2^\top \omega \mathbf{h}_2 = 0 \tag{5}$$

$$\mathbf{h}_1^\top \omega \mathbf{h}_2 = 0 \tag{6}$$

Elles ont été conçues pour développer un algorithme de calibrage linéaire. Cependant cet algorithme suppose que les H_{0i} soient connues, ou de manière équivalente, que la structure Euclidienne du plan soit connue, ce qui n'est pas le cas dans notre problème.
Dans le prochain paragraphe, nous allons décrire comment utiliser ces équations pour notre problème.

3.5 Définition du problème

Réduction du nombre d'inconnues. Lors de la phase initiale de reconstruction, nous utilisons la base projective définie par la projection du plan Π dans la première image. Comme nous l'avons dit dans la section précédente, les équations (5) et (6) sont définies pour les homographies H_{0i}, qui sont inconnues dans notre cas. Néanmoins, comme montré section 2.1, la matrice $\{H_{0i}, i = 2 \ldots n\}$

peut être exprimée comme le produit de la matrice H_{01} (inconnue) avec $\{H_{1j}, j = 2 \ldots n\}$ (estimée par l'algorithme de suivi) :

$$H_{0i} \sim H_{1i} H_{01}$$

Il est donc possible de réduire le nombre de paramètres à estimer pour résoudre le problème d'auto-calibrage : l' ICA ω et la transformation H_{01}. Cet ensemble de paramètres est indépendant de la longueur de la séquence d'images.
Chaque image, hormis la première, génère une paire d'équations (5) et (6). Ici, les vecteurs $\{\mathbf{h}_k, k = 1, 2\}$ représentent maintenant les colonnes de la matrice $H_{0i} = H_{1i} H_{01}$ (nous avons deux équations par image i).
Il peut être noté que ces équations sont non-linéaires par rapport aux paramètres ci-dessus. Nous montrerons ensuite, en section 4, comment incorporer ces équations dans un filtre de Kalman.

Estimation de H_{01} et ω. Comme mentionné dans la section 2, la matrice d'homographie H_{01} contient la structure métrique du plan Π. Elle peut être décomposée en un produit de matrices : $H_{01} = H_A H_P$, où H_P (resp. H_A) permet de retrouver les propriétés affines (resp. métriques) du plan [7]. Ces matrices peuvent s'écrire :

$$H_A = \begin{pmatrix} I & \mathbf{0} \\ \mathbf{v}^\top & 1 \end{pmatrix} \text{ and } H_P = \begin{pmatrix} M & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

où M est une matrice triangulaire supérieure normalisée telle que : $m_{11} = 1$. Cette formulation nous permet d'obtenir une forme particulière de H_{01} :

$$H_{01} = \begin{pmatrix} 1 & a & 0 \\ 0 & b & 0 \\ c & d & 1 \end{pmatrix}$$

qui dépend uniquement des quatre paramètres a, b, c, d. Ceci peut s'interpréter en termes de rectification d'un plan, ou bien en termes de la ligne de fuite et des points cycliques du plan, comme dans [10].
Remarquons que fixer les paramètres a, b, c, d à $(a = c = d = 0, b = 1)$ correspond au fait que la première image de la séquence est fronto-parallèle au plan. Nous utilisons ces valeurs lors de l'initialisation de notre système. Il est relativement aisé de vérifier cette hypothèse initiale en pratique, en commençant la séquence par une image où le plan est approximativement fronto-parallèle à la caméra.
L'ICA ω est une matrice symétrique, définie à un facteur d'échelle près. Il est donc possible de fixer $\omega_{33} = 1$, et d'estimer les cinq autres coefficients : $\{\omega_{11}, \omega_{12}, \omega_{13}, \omega_{22}, \omega_{23}\}$.
Nous avons vérifié expérimentalement que cette contrainte donne les mêmes résultats que d'imposer que la norme de ω soit constante, et de plus, cela réduit le nombre d'inconnus.
En résumé, il nous faut estimer un total de neuf paramètres : cinq pour l'image de la conique absolue, et quatre autres paramètres (a, b, c, d) pour définir l'homographie H_{01} et donc la structure Euclidienne de l'objet plan.

Remarquons enfin que, très souvent en pratique, des valeurs *a priori* de certains de ces paramètres sont disponibles, tels que le rapport des focales f_u et f_v, ou la position du point principal. Notre méthode peut être adaptée très simplement pour n'estimer que certains de ces neuf paramètres.

4 Auto-calibrage itératif de caméra

Comme expliqué ci-dessous, nous proposons de calibrer itérativement la caméra. Dans ce cas, les images sont utilisées une par une, au fur et à mesure de leur acquisition. Chaque nouvelle image amène deux contraintes, fournies par les équations (5) and (6). Ces équations sont non-linéaires par rapport aux paramètres à estimer (les coefficients portant sur les termes de ω sont linéaires, ceux concernant a, b, c, d sont quadratiques).
Ce type de non-linéarité a été étudié abondamment dans la communauté du traitement du signal. Une des techniques permettant l'estimation récursive de l'état d'un système à partir d'observations est le filtre de Kalman étendu [12]. Dans la suite de cet article, nous présentons un aperçu du filtre de Kalman, suivi par des explications sur les moyens de l'utiliser pour notre problème d'auto-calibrage.

4.1 Le filtre de Kalman étendu

Le filtre de Kalman prend en compte l'évolution de l'état du système au cours du temps $\{\mathbf{x}_k, k \in N\}$ donné par $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1})$, où $\mathbf{v}_{k-1}$ est un processus de bruit de covariance Q. $\mathbf{f}$ est une fonction, potentiellement non-linéaire de l'état $\mathbf{x}_{k-1}$.
Dans notre cas, le calibrage est supposé constant, et par conséquent $\mathbf{f}$ est fixé à l'identité.
L'objectif est d'estimer récursivement :
$\{\mathbf{x}_k, k \in N\}$.
L'état est observable au moyen d'*observations* $\mathbf{z}_k$, qui lui sont liées par l'équation :

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k)$$

où $\mathbf{h}_k : R^{n_x} \times R^{n_n} \to R^{n_z}$ est une fonction (potentiellement non-linéaire) $\{\mathbf{n}_k,\ k \in N\}$ est un processus de bruit, de covariance R.
Nous cherchons la meilleure solution filtrée de l'estimée de $\mathbf{x}_k$ basée sur l'ensemble des mesures jusqu'au temps k :

$$\mathbf{z}_{1:k} = \{\mathbf{z}_i,\ i = 1, \ldots, k\}$$

Des solutions optimales existent dans certains cas, en fonction des *a priori* possibles sur les modèles des incertitudes. Le filtre de Kalman suppose que la densité *a posteriori* de l'état $\mathbf{x}_k$ à chaque itération est Gaussienne, et suppose également que les fonctions $\mathbf{f}$ sont $\mathbf{h}$ linéaires. L'estimée de l'état est donc caractérisée par sa moyenne et sa covariance.
Si les conditions de linéarité ne sont pas satisfaites, alors le filtre de Kalman étendu (EKF) permet d'obtenir une solution approchée de la solution optimale, en utilisant une approximation au premier ordre des fonctions $\mathbf{f}$ et $\mathbf{h}$.

Afin d'estimer l'évolution de l'état, le filtre utilise les équations :

$$\begin{aligned} p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) &= N(\mathbf{x}_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) \\ p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) &= N(\mathbf{x}_k; m_{k|k-1}, P_{k|k-1}) \\ p(\mathbf{x}_k|\mathbf{z}_{1:k}) &= N(\mathbf{x}_k; m_{k|k}, P_{k|k}) \end{aligned}$$

Ici, $N(x; m, P)$ est un bruit Gaussien caractérisant l'incertitude sur l'évolution de l'état, de moyenne m et de variance P.
Le filtre estime l'état du système à un instant donné au moyen des équations de *mise à jour* suivantes :

$$\begin{aligned} m_{k|k-1} &= \mathbf{f}(m_{k-1|k-1}, 0) \\ P_{k|k-1} &= W_k Q_{k-1} W_k^T + F_k P_{k-1|k-1} F_k^T \end{aligned}$$

Reposant sur les observations au moyen de :

$$\begin{aligned} m_{k|k} &= m_{k|k-1} + K_k(\mathbf{z}_k - \mathbf{h}(m_{k|k-1}, 0)) \\ P_{k|k} &= P_{k|k-1} - K_k H_k P_{k|k-1} \end{aligned}$$

Ces définitions utilisent les Jacobiens des fonctions $\mathbf{f}$ et $\mathbf{h}$ par rapport au vecteur d'état $\mathbf{x}$ et au bruit de mesure $\mathbf{v}$.
Concrètement :
– F est le Jacobien de $\mathbf{f}$ par rapport à $\mathbf{x}$,
– H est le Jacobien de $\mathbf{h}$ par rapport à $\mathbf{x}$,
– W est le Jacobien de $\mathbf{f}$ par rapport à $\mathbf{v}$
Le terme *d'innovation* est donné par $\mathbf{z}_k - H_k m_{k|k-1}$, et sa covariance S_k et le gain du filtre K_k sont calculés au moyen de :

$$\begin{aligned} S_k &= H_k P_{k|k-1} H_k^T + R_k, \\ K_k &= P_{k|k} H_k^T S_k^{-1} \end{aligned}$$

Ici, comme indiqué ci-dessus, R est la covariance sur le bruit d'observation.

4.2 Auto-calibrage par utilisation d'un filtre de Kalman étendu

Dans cette partie, nous expliquons comment utiliser le principe de filtrage que nous venons d'exposer pour notre application d'auto-calibrage.
Comme les paramètres internes de la camera sont supposés constants, F est fixée à l'identité, et Q à la matrice nulle. La première image est supposée être l'image de référence montrant l'objet plan qui doit être suivi. Ainsi, chaque nouvelle image i est liée à la première image de la séquence à travers l'homographie H_{1i}. Cette homographie peut être directement utilisée dans un filtre de Kalman étendu, en utilisant les équations (5) et (6). Ces équations sont les équations non-linéaires $\mathbf{h}$ requises par le filtre de Kalman. Elles sont utilisées au moyen du jacobien H de $\mathbf{h}$. C'est une matrice de taille 2×9 contenant les dérivées partielles des (5) et (6) par rapport à ω_{ij} et $a, \ldots, d$. Elles peuvent être obtenues analytiquement, de manière triviale, à la main ou en utilisant un logiciel de calcul symbolique. La quantité de calculs impliqués dans notre processus d'estimation itérative réside principalement dans l'inversion de la matrice R de taille 9×9. Ces faibles coûts algorithmiques nous permettent d'obtenir sans aucune difficulté une implémentation en temps réel de la méthode proposée.
Néanmoins, il est important de noter deux inconvénients de notre méthode, liés à l'utilisation du filtre de Kalman. D'abord, les distributions des différentes variables d'état n'ont aucune raison d'avoir une distribution Gaussienne, après qu'elles soient passées à travers le processus non-linéaire. Ensuite, la linéarisation de $\mathbf{h}$ autour de la valeur courante estimée est seulement une approximation de la véritable fonction non-linéaire sous-jacente. Par conséquent, nous ne pouvons apporter de preuve théorique de la convergence de notre système. Nous ne pouvons donc pas nous passer de preuves expérimentales.

4.3 Trajectoire de la caméra

Si l'on souhaite avoir un processus d'auto-calibrage fiable, il est important d'éviter les mouvements critiques. Bien que les mouvements critiques n'aient pas encore été étudiés dans le cas d'objets plans, il semble de bon sens de penser que la trajectoire de la caméra doive contenir des variations, en particulier en ce qui concerne la rotation de la caméra. On peut remarquer que la *position* (translation) de la caméra n'est, en théorie, pas importante lorsque l'on s'intéresse au calibrage de caméra à partir d'objets plans. En effet, les équations d'auto-calibrage (5) et (6) peuvent être interprétées en terme de projection des points cycliques du plan [10], et ces points ne dépendent pas de la position de la caméra. Bien sûr, la position de la caméra a un impacte sur le suivi de l'objet, et en particulier sur la précision des mesures.
De même, les *rotations* de la caméra autour de la normale à l'objet plan n'ont pas d'impact sur les équations de calibrage. Nous avons donc l'intuition que seuls deux paramètres doivent varier pour obtenir un auto-calibrage fiable : l'angle que l'axe optique de la caméra forme avec la normale au plan de l'objet, et la rotation de la caméra autour de son centre optique.
Nous avons imaginé une trajectoire qui satisfasse ces exigences : la trajectoire commence dans une position approximativement fronto-parallèle. Soit F le point de l'objet plan qui appartient à l'axe optique de la caméra. Nous déplaçons ensuite la caméra approximativement sur un cercle autour de F, en direction de l'objet plan, tout en restant centré sur F. Cela permet de faire varier l'angle entre l'axe optique et la normale au plan. Pendant ce mouvement, la caméra tourne autour de son axe optique.
Cette trajectoire, bien que très simple, n'est pas facile à réaliser avec une caméra tenue à la main. Nous avons utilisé dans nos expériences des séquences comportant des mouvements tels que celui que nous venons de décrire, mais aussi d'autres séquences réalisées plus librement.
Nous avons obtenu des résultats identiques dans les deux

cas, et nous n'avons jamais eu le sentiment que le mouvement réalisé ait une importance dans les performances du processus de calibrage.

5 Résultats expérimentaux

Nous avons évalué les performances de la méthode proposée sur des données simulées et sur des données réelles. Dans les deux cas, nous avons comparé notre méthode avec une méthode d'optimisation globale non-linéaire permettant d'estimer les mêmes neuf paramètres, en commençant dans les deux cas avec les mêmes valeurs initiales et les mêmes mesures.

La méthode globale minimise la fonction de coût non-linéaire donnée par la somme des carrés, sur l'ensemble des images des équations (5) et (6). L'optimisation se fait au moyen de l'algorithme de Levenberg-Marquardt (implémentation Matlab).

5.1 Données de synthèse

Nous simulons l'objet plan, ses images et les trajectoires de caméra.

L'objet consiste en 40 points coplanaires répartis à l'intérieur d'un carré. Les paramètres internes de la caméra sont : $f_u = f_v = 1, u_0 = v_0 = 0.256, s = 0$. Nous travaillons de manière implicite avec des coordonnées image normalisées.

Nous simulons les enchaînements de vues grâce à un modèle de trajectoire décrit dans la section 4.3, mais aussi avec des positions arbitraires (avec la contrainte de garder la distance à l'objet plan à peu près constante, et que l'objet reste dans le champ de vision). Les deux types de positionnement donnent le même type de résultat. Nous avons simulé des calibrages en utilisant jusqu'à 220 images par expérience.

Les coordonnées des points co-planaires de l'objet sont bruitées par ajout d'un bruit blanc Gaussien ayant un écart-type de 0.005, ce qui correspond à environ 5 pixels dans des images réelles (rappelons que la focale utilisée est de 1).

Les homographies inter-images sont calculées en utilisant les correspondances des 40 points, pour chaque paire de vues $(1, i)$.

L'initialisation des paramètres est calculée comme suit :

- paramètres intrinsèques : les paramètres intrinsèques sont initialisés à leurs vraies valeurs $(f_u, \dots)$ auxquelles s'ajoute un bruit Gaussien tel que les valeurs puissent être bruitées de plus de 50% de leur valeur réelle. A partir de ces valeurs, les inconnues estimées par la méthode (les coefficients de ω), sont alors calculées et utilisées comme valeurs initiales.
- l'homographie H_{01} : est initialisée à l'identité (c'est-à-dire $a = c = d = 0$ et $b = 1$) ce qui correspond à l'hypothèse que la première vue est fronto-parallèle. Nous simulons la première vue à partir d'une pose initiale pour laquelle l'angle que la caméra forme avec le plan de l'objet peut aller jusqu'à $50°$.

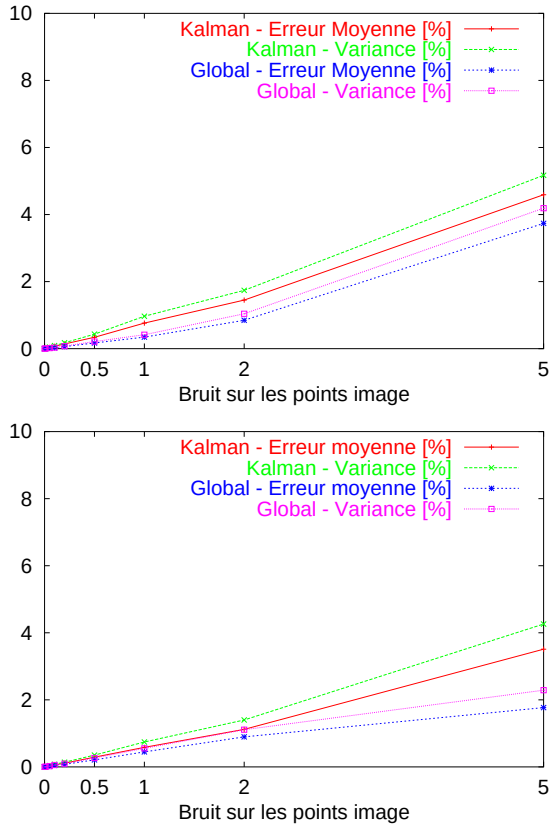

FIG. 2 – Evaluation des performances en fonction du bruit présent dans les images. Haut : résultats pour f_u. Bas : résultats pour for u_0. Les valeurs initiales des paramètres intrinsèques sont obtenues à partir des valeurs réelles auxquelles un bruit uniforme de 20% est ajouté. La première image est simulée pour une pose telle que l'axe optique forme un angle de $20°$ avec la normale du plan de l'objet. L'unité utilisée pour le bruit correspond au nombre de pixels pour une focale de 1000 et une taille d'image de 512×512.

Les graphiques suivants montrent les erreurs moyennes et les variances, pour 100 expériences aléatoires pour chaque combinaison de paramètres initiaux.

La Figure 2 montre les résultats pour f_u et u_0, en fonction du niveau de bruit ajouté aux points de l'image. La méthode globale est meilleure, les deux méthodes utilisant le même nombre d'images. Néanmoins, la méthode itérative permet d'obtenir des résultats très précis pour une méthode d'auto-calibrage, avec une erreur moyenne en dessous de 2% , et ceci pour des niveaux de bruit tout à fait réalistes.

Ces graphes correspondent à des erreurs d'initialisation des paramètres intrinsèques de 20%. Pour évaluer le domaine de convergence de la méthode, nous perturbons les valeurs initiales jusqu'à 50% de leur valeur et plaçons la première vue telle que l'axe optique forme un angle pouvant atteindre $50°$, par rapport à la normale au plan de l'objet. Cela signifie que notre initialisation de H_{01} est très mau-

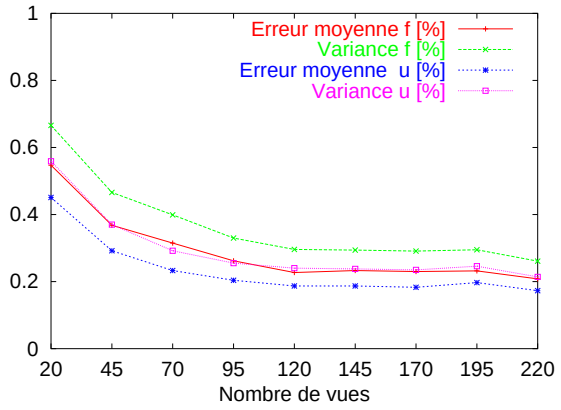

FIG. 3 – Performance de la méthode itérative en fonction du nombre de vues. Pour ces expériences, les coordonnées des points de l'image sont perturbées par un bruit Gaussien d'un écart-type de 1 pixel.

vaise.
Même dans de telles conditions initiales, notre méthode converge dans plus de 95% des cas vers des solutions présentant le même niveau d'erreur que ceux montrés dans la figure 2, par exemple.
La Figure 3 montre les résultats obtenus avec la méthode itérative, en fonction du nombre de vues utilisées. Comme nous nous y attendions, l'erreur moyenne décroît en fonction du nombre de vues, plus ou moins de manière asymptotique.
Des erreurs relatives en dessous du tiers d'un pourcent sont obtenues lorsque plus de 100 vues sont utilisées.

5.2 Expériences avec des données réelles

Nous avons pris des séquences vidéo d'objets plans texturés. Chaque prise débute par une vue approximativement fronto-parallèle. Les séquences sont obtenues par une caméra tenue à la main. Les séquences sont traitées en ligne au moyen de la méthode décrite en section 2.1.
Nous avons évalué les performances de notre méthode et de la méthode globale en comparant les résultats obtenus avec ceux obtenus par un ajustement de faisceaux, minimisant directement les erreurs de reprojection des points dans les images. Les deux méthodes donnent des résultats identiques pour les paramètres intrinsèques, donnant des résultats à 2% près à ceux fournis par l'ajustement de faisceaux.
Nous avons pris des vidéos avec différentes trajectoires, en déplaçant l'objet à la main devant la caméra statique. Nous avons essayé des mouvements tels que ceux suggérés dans la section 4.3. Puisque ces mouvements ne sont pas très intuitifs, et ne sont pas faciles à réaliser, nous avons également essayé des mouvements quelconques, et obtenu des résultats similaires. Afin d'illustrer ces résultats, nous présentons une image issue de l'une de ces séquences, après rectification, montrant que la structure Euclidienne du plan est correctement estimée (voir la figure 4).

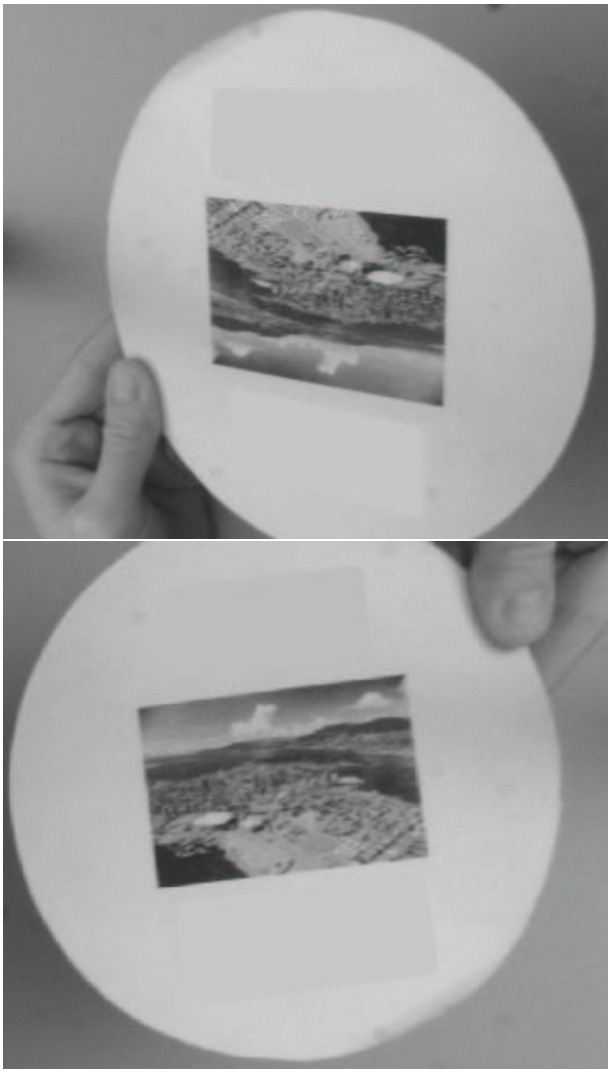

FIG. 4 – Représentation d'une image issue d'une vidéo (à gauche) qui est rectifiée de manière à rétablir sa structure Euclidienne en utilisant les résultats de l'auto-calibrage (droite). Notons que la zone rectangulaire est correctement reconstruite.

5.3 Discussion

La méthode itérative proposée donne des résultats similaires à ceux obtenus au moyen d'une méthode d'optimisation globale non-linéaire utilisant simultanément toutes les images, tant en ce qui concerne le domaine de convergence que la précision des estimations. Le facteur principal qui peut faire échouer la méthode proposée est le cas ou la caméra demeure fixe pendant une très grande période de temps. Cela peut se détecter aisément en pratique. Nos séquences vidéos réelles ont été traitées telles quelles.

Nous avons observé que notre méthode semble plus sensible à des erreurs sur les valeurs initiales des paramètres de H_{01} qu'à des erreurs sur les paramètres intrinsèques. Nous avons pu montrer en pratique que ce problème ne se posait pas si l'on démarrait la séquence par une vue fronto-parallèle de l'objet.

Il semble que 15 ou 20 images soit le nombre minimal d'images nécessaires pour que notre méthode puisse donner des résultats convenables. Il faut pendre cette remarque avec précaution, puisque ce n'est pas tant le nombre d'images qui compte que la variation du point de vue entre ces images (en particulier la rotation de la caméra). Ainsi nous avons proposé un type de trajectoire de caméra qui satisfasse intuitivement les critères requis qui donne en pratique de bons résultats. Il semble de plus que tant que des trajectoires trop régulières sont évitées (par exemple sans rotation de la caméra) de bons résultats puissent être attendus.

6 Conclusions

Nous avons proposé une méthode itérative de calibrage de caméra dont le principe repose sur l'observation d'un plan de géométrie inconnue par la caméra vidéo. Cette méthode ne suppose que la connaissance de matrices d'homographie inter-images qui peuvent être calculées facilement et efficacement. La méthode proposée constitue une alternative pratique à des méthodes de calibrage déjà existantes qui sont basées sur des algorithmes globaux d'optimisation non-linéaires. Nous avons montré que des performances similaires peuvent être obtenues en terme de précision et de domaine de convergence, pour un coût algorithmique beaucoup plus faible.

Un système expérimental complet a été implémenté : il reçoit en entrée des flots d'images vidéo et permet d'auto-calibrer en ligne la caméra. Ce système fonctionne de manière robuste, tant en ce qui concerne le suivi qu'en ce qui concerne l'auto-calibrage. Il est plus flexible que les systèmes déjà existants, [4] et nous espérons ainsi qu'il sera utilisé dans l'avenir.

[4] Tel que par exemple le système de calibrage plan proposé par Intel, cf. `http ://www.intel.com/research/mrl/research/opencv/`.

Références

[1] J. Alon et S. Sclaroff, "Recursive Estimation of Motion and Planar Structure," *CVPR*, pp. 550-556, 2000.

[2] A. Azarbayejani et A. Pentland, "Recursive Estimation of Motion, Structure, and Focal Length," *IEEE PAMI*, Vol. 17, No. 6, pp. 562-575, 1994.

[3] O. Faugeras, "Stratification of Three-Dimensional Vision : Projective, Affine and Metric Representations," *Journal of the Optical Society of America – A*, Vol. 12, pp. 465-484, 1995.

[4] P. Gurdjos, A. Crouzil et R. Payrissat, "Another Way of Looking at Plane-Based Calibration : The Centre Circle Constraint," *ECCV*, Vol. 4, pp. 252-266, 2002.

[5] P. Gurdjos et P. Sturm. "Methods and Geometry for Plane-Based Self-Calibration," *CVPR*, pp. 491-496, 2003.

[6] R. Hartley, "An Algorithm for Self Calibration from Several Views," *CVPR*, pp. 908-912, 1994.

[7] R. Hartley et A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[8] A. Heyden et K. Åström, "Flexible Calibration : Minimal Cases for Auto-Calibration," *ICCV*, pp. 350-355, 1999.

[9] F. Jurie et M. Dhome, "A Simple and Efficient Template Matching Algorithm," *ICCV*, Vol. 2, pp. 544-549, 2001.

[10] D. Liebowitz, *Camera Calibration and Reconstruction of Geometry from Images*, PhD Thesis, Dept. Engineering Science, University of Oxford, 2001.

[11] E. Malis et R. Cipolla, "Camera Self-Calibration from Unknown Planar Structures Enforcing the Multi-View Constraints between Collineations," *IEEE PAMI*, Vol. 24, No. 9, pp. 1268-1272, 2002.

[12] P.S. Mayback, *Stochastic Models, Estimation and Control*, Vol. 1/2, Academic Press, 1979.

[13] S.J. Maybank et O.D. Faugeras, "A Theory of Self Calibration of a Moving Camera," *IJCV*, Vol. 8, No. 2, pp. 123-151, 1992.

[14] M. Pollefeys, R. Koch et L. Van Gool, "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters," *ICCV*, pp. 90-95, 1998.

[15] P. Sturm et S. Maybank. "On Plane-Based Camera Calibration : A General Algorithm, Singularities, Applications," *CVPR*, pp. 432-437, 1999.

[16] P. Sturm, "Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction," *CVPR*, pp. 1100-1105, 1997.

[17] B. Triggs, "Autocalibration from Planar Scenes," *ECCV*, pp. 89-105, 1998.

[18] G. Xu, J.-I. Terai et H.-Y. Shum, "A Linear Algorithm for Camera Self-Calibration, Motion and Structure Recovery for Multi-Planar Scenes from Two Perspective Images," *CVPR*, Vol. II, pp. 474-479, 2000.

[19] L. Zelnik-Manor et M. Irani, "Multi-View Subspace Constraints on Homographies," *ICCV*, Vol. 2, pp. 710-715, 1999.

[20] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE PAMI*, Vol. 22, No. 11, pp. 1330-1334, 2000.