

Wide Area Multiple Camera Calibration and Estimation of Radial Distortion

João P. Barreto^{1,2} and Kostas Daniilidis¹

¹ GRASP Laboratory, University of Pennsylvania, Philadelphia PA, 19104

² ISR/DEEC, University of Coimbra, Coimbra, Portugal
{jpbar, kostas}@grasp.cis.upenn.edu

Abstract. The calibration of cameras distributed in a wide area is a challenging task because it is impossible to use reference objects visible to all cameras and because wide field-of-view cameras suffer under radial distortion. The present work proposes the first algorithm in the literature for radial distortion estimation from multiple views without involving non-linear minimization. The correspondences between views are obtained by deliberately moving an LED in thousands of unknown positions in front of the cameras. Then both projection matrices and radial distortion parameters are simultaneously computed using a factorization approach. The algorithm is based on the application of two subspace approximation steps. At these steps, the estimated approximate solution for a matrix can be projected to the manifold of the parameter space by adjusting the singular values. It is remarkable, that our system does not involve a single non-linear minimization or outlier treatment and still produces accurate results which have been tested in a multi-camera reconstruction algorithm. In addition to real imagery results, we have analyzed the behavior of the algorithm in simulations.

1 Introduction

Immersive environments as well as surveillance systems require multiple cameras to visually capture environments of considerable extent like big rooms or even outdoors scenes. To be able to register everything to the same world coordinate system, be it 3D models or human activities, we have to calibrate all cameras. This means the estimation of projective matrices from world to image coordinates as well as the removal of lens distortions, assumed here to be radial. In summary, we have to be able to map every pixel at each camera to a ray in the scene defined in world coordinates. Single camera calibration is regarded as a solved problem and open source packages are widely disseminated [9]. However, multiple camera calibration poses a challenge because of lacking overlap of the field of views among cameras. It is easy to realize that reference objects unless they are very elongated lines [6] will not be visible in all cameras. We have to abandon the idea of a metric reference object and switch to an arbitrary structure. If the world itself is used as an arbitrary structure, we have to identify features and solve the correspondence problem which would make calibration vulnerable to matching errors. Instead, like others [5, 11, 7] we avoid the correspondence problem by moving a single light source in a dark environment of thousands of arbitrary positions.

Obviously, the only reconstruction possible is up to a projective transformation. To fix our cameras to a projective world, we conventionally calibrate two of them and we integrate this euclidean adjustment in the projective factorization set-up. At this point,

we could have run a bundle adjustment [11] introducing as additional unknowns the radial distortion. Instead, we propose a novel algorithm for estimating the radial distortion over multiple views. The only requirement is a rough estimation of the distortion parameters of two cameras in the set. This algorithm does not include any nonlinear minimization but only two steps where the closest rank deficient matrix to a full-rank matrix is obtained via SVD.

The main contribution of this paper is this simultaneous treatment of projection matrices and radial distortion over multiple views and without any nonlinear minimization. The algorithm differs from others in the literature [5, 4, 11] for being a one-shot method able to recover the projection matrices and the radial distortion of multiple cameras with minimum computational effort. Moreover it makes use of all multiple view constraints simultaneously, instead of just pairwise fundamental matrices [5, 4]. Compared with [11] it provides an estimation for the projection matrices and radial distortion without requiring euclidean stratification and nonlinear minimization.

2 Geometry of Multiple Views

Consider a 3D point \mathbf{X} and a set of K cameras with projection matrices \mathbf{P}_i ($i = 1, \dots, K$). For each view \mathbf{C}_i the 3D point \mathbf{X} is projected in the image point \mathbf{x}_i . Equation 1 shows the mathematical relation in homogeneous coordinates

$$\lambda_i \mathbf{x}_i = \mathbf{P}_i \mathbf{X}, \quad i=1, \dots, K \quad (1)$$

Where \mathbf{X} and \mathbf{x}_i are 4 and 3 dimension vectors respectively. The scalar λ_i denotes the depth and \mathbf{P}_i is a 3×4 matrix which can be split in the way shown in equation 2 (\mathbf{R}_i is a 3×3 matrix and \mathbf{t}_i is a 3×1). Please note that \mathbf{R}_i does not have to be orthogonal. Notice that both \mathbf{X} and \mathbf{P}_i , with $i = 1, \dots, K$, must be defined in a common coordinate system in the world

$$\mathbf{P}_i = [\mathbf{R}_i \ \mathbf{t}_i], \quad i=1, \dots, K \quad (2)$$

2.1 Depth Estimation

Assume that $\mathbf{X} = (\bar{\mathbf{X}}, 1)^t$ is simultaneously viewed by M cameras \mathbf{C}_i for which the projection matrices \mathbf{P}_i are known ($i = 1, \dots, M$). We aim to estimate the depth λ_1 with respect to the first view. From equations 1 and 2 arises

$$\bar{\mathbf{X}} = \mathbf{R}_1^{-1}(\lambda_1 \mathbf{x}_1 - \mathbf{t}_1).$$

Replacing \mathbf{X} in the projection equations of the remaining $M - 1$ views and multiplying both members by the skew symmetric matrix $\hat{\mathbf{x}}_i$ yields

$$\lambda_1 \hat{\mathbf{x}}_i \mathbf{R}_i \mathbf{R}_1^{-1} \mathbf{x}_1 + \hat{\mathbf{x}}_i (\mathbf{t}_i - \mathbf{R}_i \mathbf{R}_1^{-1} \mathbf{t}_1) = 0, \quad i=2, \dots, M. \quad (3)$$

Equation 4 is derived by gathering together the previous $M - 1$ equations

$$\underbrace{\begin{bmatrix} \hat{\mathbf{x}}_2 \mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{x}_1 & \hat{\mathbf{x}}_2 (\mathbf{t}_2 - \mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{t}_1) \\ \hat{\mathbf{x}}_3 \mathbf{R}_3 \mathbf{R}_1^{-1} \mathbf{x}_1 & \hat{\mathbf{x}}_3 (\mathbf{t}_3 - \mathbf{R}_3 \mathbf{R}_1^{-1} \mathbf{t}_1) \\ \vdots & \vdots \\ \hat{\mathbf{x}}_M \mathbf{R}_M \mathbf{R}_1^{-1} \mathbf{x}_1 & \hat{\mathbf{x}}_M (\mathbf{t}_M - \mathbf{R}_M \mathbf{R}_1^{-1} \mathbf{t}_1) \end{bmatrix}}_{\mathbf{A}_\lambda} \underbrace{\begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix}}_{\boldsymbol{\eta}} = 0. \quad (4)$$

In ideal circumstances the $3(M-1) \times 2$ matrix \mathbf{A}_λ should be rank 1. However, due to the measurement errors, the measurement vectors occupy in general a higher dimensional linear manifold. Matrix \mathbf{A}_λ is usually full rank and we need to estimate the underlying null subspace in order to determine the depth λ_1 . The solution to the stated subspace problem is the direction which is more likely to be the null space of the true unperturbed matrix \mathbf{A}_λ . The key to correctly identify the subspace is given by the Eckart-Young-Mirsky (EYM) theorem [10]. According to the EYM theorem, vector $\boldsymbol{\eta}$ can be estimated by performing the SVD decomposition of matrix \mathbf{A}_λ ($\mathbf{A}_\lambda = \mathbf{U}\mathbf{S}\mathbf{V}^t$) and selecting the column of \mathbf{V} corresponding to the smallest singular value. The derived $\boldsymbol{\eta}$ minimizes function $\epsilon(\boldsymbol{\eta}) = \boldsymbol{\eta}^t \mathbf{A}_\lambda^t \mathbf{A}_\lambda \boldsymbol{\eta}$ under the constraint $\boldsymbol{\eta}^t \boldsymbol{\eta} = 1$ [10]. In order to achieve robust and accurate results using subspace methods a proper equilibration of the design matrices is required.

2.2 Computing the Projection Matrix \mathbf{P}_i

Assume a set of N points projected at coordinates $\mathbf{x}_1^1, \mathbf{x}_1^2, \dots, \mathbf{x}_1^N$ in the first view \mathbf{C}_1 . The projection matrix $\mathbf{P}_1 = [\mathbf{R}_1 \ \mathbf{t}_1]$ is known as well as the point depths $\lambda_1^1, \lambda_1^2, \dots, \lambda_1^N$. These points are also viewed by camera \mathbf{C}_i . The image points are $\hat{\mathbf{x}}_i^1, \hat{\mathbf{x}}_i^2, \dots, \hat{\mathbf{x}}_i^N$ and the corresponding projection matrix $\mathbf{P}_i = [\mathbf{R}_i \ \mathbf{t}_i]$ is unknown ($M < i \leq K$).

Consider the 3×3 matrix $\boldsymbol{\Psi}_i$ and the 3×1 vector ϕ_i (equation 5). Concatenating the columns of $\boldsymbol{\Psi}_i$ we obtain a 9×1 vector ψ_i

$$\begin{cases} \boldsymbol{\Psi}_i = \mathbf{R}_i \mathbf{R}_1^{-1} \\ \phi_i = \mathbf{t}_i - \mathbf{R}_i \mathbf{R}_1^{-1} \mathbf{t}_1 \end{cases} \quad (5)$$

The relation of equation 3 can be written in the form of equation 6 where \otimes denotes the Kronecker product [10]

$$\lambda_1^j (\hat{\mathbf{x}}_i^j \otimes \mathbf{x}_1^j) \psi_i + \hat{\mathbf{x}}_i^j \phi_i = 0, \quad j=1, \dots, N \quad (6)$$

Gathering together the N points yields

$$\underbrace{\begin{bmatrix} \lambda_1^1 (\hat{\mathbf{x}}_i^1 \otimes \mathbf{x}_1^1) & \hat{\mathbf{x}}_i^1 \\ \lambda_1^2 (\hat{\mathbf{x}}_i^2 \otimes \mathbf{x}_1^2) & \hat{\mathbf{x}}_i^2 \\ \vdots & \vdots \\ \lambda_1^N (\hat{\mathbf{x}}_i^N \otimes \mathbf{x}_1^N) & \hat{\mathbf{x}}_i^N \end{bmatrix}}_{\mathbf{A}_i} \begin{bmatrix} \psi_i \\ \phi_i \end{bmatrix} = 0. \quad (7)$$

Vectors ψ_i and ϕ_i can be estimated by applying the EYM theorem to matrix \mathbf{A}_i . Notice that we need a minimum of six points in general position ($N \geq 6$). Matrix $\boldsymbol{\Psi}_i$ is obtained by rearranging ψ_i , and, since \mathbf{R}_1 and \mathbf{t}_1 are known, the projection matrix $\mathbf{P}_i = [\mathbf{R}_i \ \mathbf{t}_i]$ can be recovered from equation 5.

| | |
|---------------|--|
| Step 1 | Consider the set of points which are projected on camera C_1 . |
| Step 2 | Remove the points which are not viewed by any other calibrated camera C_i ($i = 2, \dots, M$) in order to obtain the set S_1 |
| Step 3 | For each point of S_1 : 3.1 - Build matrix A_λ (equation 4) 3.2 - Use the SVD decomposition of A_λ to compute the depth λ_1 |
| Step 4 | For each camera C_i ($i = 2, \dots, K$): 4.1 - Determine the subset S_i , of S_1 , containing the points viewed by C_1 and C_i 4.2 - If S_i has not enough elements ($N < 6$) then move to the next camera. 4.3 - Build matrix A_i (equation 7) 4.4 - Use the SVD decomposition of A_i to estimate ψ_i and ϕ_i 4.5 - Update the projection matrix P_i (equation 5) |

Table 1. Algorithm to estimate the projection matrices P_i

3 Calibration Algorithm

The present section proposes an algorithm to calibrate a large number of cameras spread through a wide area. We aim to determine the projection matrices P_1, \dots, P_K of a set of K cameras. Our approach is inspired by the multiple-view factorization of point features presented in [8, 14]. The method requires at least two cameras to be calibrated in advance. We will assume, without loss of generality, that the projection matrices of the first M views are known ($M \geq 2$). The goal is to determine the projection matrices of the remaining $K - M$ cameras.

The estimation of the projection matrices can be accomplished using the theory introduced in the previous section. The input for the calibration procedure is the multiple view of a set of N_p points in the scene. However, finding points x_i^j and establishing correspondences between multiple images is a difficult task. If the cameras are synchronized then the problem can be solved using a laser pointer or a LED in a similar way as proposed in [11, 4, 5]. The user is required to move the laser/LED throughout the working volume. The room should be as dark as possible for an easy detection of the virtual point. The illumination conditions provide enough contrast such that the point projection can be accurately measured by performing a simple image threshold.

Table 1 outlines the proposed calibration procedure. Notice that by waving the laser/LED in a free way there is no guarantee that the virtual point is simultaneously viewed by all cameras. We start by determining the set of points S_1 which are simultaneously viewed by the reference camera C_1 and by at least one camera C_i with $1 < i \leq M$. Since the points are viewed by a minimum of two calibrated cameras then the corresponding depths λ_1^j , with respect to C_1 , can be determined following the procedure outlined in section 2.1. For each camera C_i , with $1 < i \leq K$, we select the points of S_1 which lie on the corresponding field of view. The image points x_1^j , x_i^j and corresponding depths λ_1^j ($j = 1, \dots, N$) are used to obtain matrix A_i (equation 7). The projection matrix P_i is determined in the way explained in section 2.2.

3.1 Initialization Stage

In order to compute the point depths λ_1^j the algorithm requires a minimum of two calibrated cameras ($M \geq 2$). The projection matrices \mathbf{P}_1 and \mathbf{P}_2 can be determined just from N point correspondences between the views. If $N \geq 8$ then the computation of the fundamental matrix \mathbf{F} , such that $\mathbf{x}_1^j \mathbf{F} \mathbf{x}_2^j = 0$, is trivial. Let's consider matrices $\mathbf{P}'_1 = [\mathbf{I} \mathbf{0}]$ and $\mathbf{P}'_2 = [\hat{\mathbf{e}} \mathbf{F} \mathbf{e}]$, where \mathbf{e} and $\hat{\mathbf{e}}$ are the epipole and the corresponding skew symmetric matrix, and \mathbf{I} is the identity matrix. It can be shown that, given a general 4×4 collineation \mathbf{H} , any pair $\mathbf{P}_1 = \mathbf{P}'_1 \mathbf{H}$ and $\mathbf{P}_2 = \mathbf{P}'_2 \mathbf{H}$ is a valid solution for the projection matrices [8].

Matrices \mathbf{P}'_1 and \mathbf{P}'_2 can be used to initialize the algorithm and calibrate the set of K cameras up to a 4×4 projective transformation \mathbf{H} . In order to obtain the Euclidean projection matrices \mathbf{P}_i we have to find the projective transformation \mathbf{H} such that $\mathbf{P}_i = \mathbf{P}'_i \mathbf{H}$ for $i = 1, \dots, K$. This process is called Euclidean stratification [12] and requires the assumption of certain geometrical constraints (orthogonality between image lines and columns, knowledge about the camera intrinsic parameters, etc). Depending on the assumptions and amount of information the stratification process can be more or less stable and accurate. In a similar way as proposed in [11], we can use the euclidean stratification in order to obtain automatic multi-camera self calibration .

Using a moving plane to calibrate a large scale multi-camera environment is problematic due to the fact that the calibration rig must be visible in all cameras. Nevertheless, methods like the ones in [9, 1] are perfectly suitable to calibrate a small number of cameras for which the FOV overlaps. The Bouguet Calibration Toolbox [9] is used in order to calibrate a minimum of two cameras and initialize the algorithm. Since the initial projection matrices $\mathbf{P}_1, \dots, \mathbf{P}_M$ are euclidean, the calibration of the camera set obtained from the point correspondences is also euclidean. This practical procedure avoids the stratification process leading in general to more accurate results.

3.2 Algorithm Iterations

If data points are noiseless, the initial projection matrices are quite accurate, and each camera views at least 6 points of \mathbf{S}_1 , then a single run of the algorithm is enough to accurately determine the projection matrices. In general nor the image points are noiseless, neither the initial projection matrices are perfect. The procedure outlined on Table 1 must run more than one time in order to obtain an accurate calibration of the camera set. The iterative process can stop after a pre-defined number of iterations, when all cameras are calibrated or when the re-projection error is below a certain threshold.

According to equation 7 the projection matrices seem to be estimated using pairs of views only. However this is not the case when the procedure is run iteratively. The set of M calibrated cameras is updated at the end of each iteration and the new projection matrices are used to recompute the depth of the data points. Thus, in the subsequent iteration, matrices \mathbf{P}_i are updated using the newly recovered λ_1^j , each of which was estimated using all available views. The algorithm makes use of all constraints simultaneously, being robust to noise, and to errors in the initial projection matrices. Our multiple view approach not only calibrates the cameras that are unknown, but also updates the initial projection matrices in order to enforce the correspondences.

Since the method relies on point correspondence between multiple views, the field of view (FOV) of the cameras must overlap. However it is not necessary for all cameras

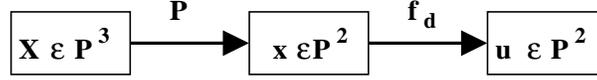


Fig. 1. Mapping model including radial distortion

to have a common FOV, as long as every camera is connected to every other camera through a common FOV. For a certain reference camera C_1 , the algorithm is run in order to calibrate the views for which the number of point correspondences is enough ($N \geq 6$). If views still remain uncalibrated, then the algorithm is run again assuming a newly calibrated camera as the reference view. The procedure can be repeated over and over till $M = K$ and the entire set is calibrated.

4 Calibration Including Radial Distortion

The calibration algorithm proposed in the previous section assumes a pin-hole model for the cameras. Many applications using multiple cameras require wide area visualization. In order to cover the working volume in an efficient way it is usually convenient for the cameras to have a large FOV. However, lenses with short focal length suffer from significant non-linear distortion which cannot be neglected. This section presents the paper's main contribution. We show how the calibration algorithm, summarized on Table 1, can be modified in order to include radial distortion. The final algorithm is still a one-shot method, relying on the solution of generalized eigensystems to estimate the projective matrices P_i and the radial distortion parameters ξ_i .

4.1 Model for Radial Distortion

The scheme of Fig. 1 shows the assumed mapping model. Point $\mathbf{u} = (u_x, u_y, u_z)^t$ is the image of a generic 3D point \mathbf{X} . The projection matrix \mathbf{P} transforms point \mathbf{X} into the 2D point $\mathbf{x} = (x, y, z)^t$ (equation 1). If the camera lens is distortion free then $\mathbf{u} = \mathbf{x}$ and the calibration algorithm of Tab. 1 applies without modification. However, in the presence of distortion, \mathbf{x} and \mathbf{u} are related by a non-linear mapping function f_d which must be taken into account.

We model the radial distortion using the so called division model [3]. Consider points \mathbf{x} and \mathbf{u} expressed in a 2D coordinate system with origin at the distortion center. The relation between the two points is given by equation 8, where the radial distortion is parameterized by ξ . This model is used by Fitzgibbon in order to simultaneously determine the fundamental matrix and the radial distortion between two views [3]. He shows that equation 8 approximates the radial distortion curve as well as the traditional first order model. Notice that the model requires the distortion center to be known. In the absence of any other information, we can place it at the image center without significantly affect the correction [13]

$$\mathbf{x} = \mathbf{f}_d^{-1}(\mathbf{u}, \xi) = \mathbf{u} + \xi \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{u_x^2 + u_y^2}{u_z} \end{bmatrix}}_{\mathbf{v}} \quad (8)$$

4.2 Including Radial Distortion in the Multiple View Equations

Consider the set of K cameras where the projection matrices \mathbf{P}_i and the distortion parameters ξ_i are known for the first M views ($M \geq 2$). Given a calibrated view, the radial distortion of a generic point \mathbf{u}_i can be corrected and the corresponding point \mathbf{x}_i determined. However if the distortion parameter is not known, then the distortion can not be removed and the multiple view constraints derived in section 2 do not apply.

Assume a set of N points simultaneously viewed by cameras C_1 and C_i ($i > M$). The points in the set are visible in more than one calibrated view and the corresponding depth λ_1^j is known (section 2.1). Since parameter ξ_i is unknown, then the radial distortion in the image points \mathbf{u}_i^j cannot be removed. However, equation 8 implies that $\mathbf{x}_i^j = \mathbf{u}_i^j + \xi_i \mathbf{v}_i^j$ with $j = 1, \dots, N$. Replacing in equation 6 yields

$$(\lambda_1^j (\hat{\mathbf{u}}_i^j \otimes \mathbf{x}_1^j) \psi_i + \hat{\mathbf{u}}_i^j \phi_i) + \xi_i (\lambda_1^j (\hat{\mathbf{v}}_i^j \otimes \mathbf{x}_1^j) \psi_i + \hat{\mathbf{v}}_i^j \phi_i) = 0, j=1, \dots, N \quad (9)$$

Gathering the N points together we obtain

$$\underbrace{\begin{bmatrix} \lambda_1^1 (\hat{\mathbf{u}}_i^1 \otimes \mathbf{x}_1^1) & \hat{\mathbf{u}}_i^1 \\ \lambda_1^2 (\hat{\mathbf{u}}_i^2 \otimes \mathbf{x}_1^2) & \hat{\mathbf{u}}_i^2 \\ \vdots & \vdots \\ \lambda_1^N (\hat{\mathbf{u}}_i^N \otimes \mathbf{x}_1^N) & \hat{\mathbf{u}}_i^N \end{bmatrix}}_{\mathbf{A}_i} \begin{bmatrix} \psi_i \\ \phi_i \end{bmatrix} + \xi_i \underbrace{\begin{bmatrix} \lambda_1^1 (\hat{\mathbf{v}}_i^1 \otimes \mathbf{x}_1^1) & \hat{\mathbf{v}}_i^1 \\ \lambda_1^2 (\hat{\mathbf{v}}_i^2 \otimes \mathbf{x}_1^2) & \hat{\mathbf{v}}_i^2 \\ \vdots & \vdots \\ \lambda_1^N (\hat{\mathbf{v}}_i^N \otimes \mathbf{x}_1^N) & \hat{\mathbf{v}}_i^N \end{bmatrix}}_{\mathbf{B}_i} \begin{bmatrix} \psi_i \\ \phi_i \end{bmatrix} = 0. \quad (10)$$

Equation 10 generalizes the result of equation 7 for cameras with radial distortion. If $\xi_i = 0$ then $\mathbf{u}_i^j = \mathbf{x}_i^j, \forall i$ and the second term of the equation vanishes. Another important observation is that matrix \mathbf{B}_i is always rank deficient. Columns 3, 6, 9 and 12 are constant and null due to the particular structure of vector \mathbf{v} (equation 8).

The result of equation 10 is stated in a more compact way in 11. \mathbf{A} and \mathbf{B} are $3N \times 12$ matrices, $\alpha = (\psi^t, \phi^t)^t$ is a 12×1 vector and ξ is a scalar

$$\mathbf{A}\alpha + \xi\mathbf{B}\alpha = 0 \quad (11)$$

Since $\text{rank}(\mathbf{B}) = 8$, the corresponding null space $\mathfrak{N}(\mathbf{B})$ has dimension 4. Any pair (ξ, α) with $\xi = \infty$ and $\alpha \in \mathfrak{N}(\mathbf{B})$ is a solution of equation 11. However these solutions are degenerate since they do not correspond to the camera calibration.

In order to apply square solvers we can multiply both members of the equation 11 by \mathbf{A}^t . The equation becomes $\mathbf{A}^t \mathbf{A} \alpha + \xi \mathbf{A}^t \mathbf{B} \alpha = 0$ and the solution (ξ, α) can be determined by solving a generalized eigenvalue problem (GEP). This approach is suggested in [3] where the author claims that, since \mathbf{A} is full rank, the premultiplication by \mathbf{A}^t does not change the solution. Matrix \mathbf{A}^t is a $12 \times 3N$ matrix and, even if \mathbf{A} is full rank, the corresponding right null space $\mathfrak{N}(\mathbf{A}^t)$ has dimension $3N - 12$. This means that any pair (ξ, α) , such that $(\mathbf{A}\alpha + \xi\mathbf{B}\alpha) \in \mathfrak{N}(\mathbf{A}^t)$, is a solution of the enunciated GEP. The solutions of the GEP do not necessarily verify equation 11. Nevertheless, if (ξ, α) is a solution of equation 11, then it can be determined by solving the GEP. The problem is that, in general and due to the noise effect, equation 11 has no solutions beside the degenerate ones. The camera calibration can be determined by solving the GEP if, and only if, the data is noiseless. The next section presents a method to estimate (ξ, α) in order to calibrate the camera in the presence of noise

4.3 Finding a Solution for Camera Calibration

Consider equation 12 where the order of the columns of \mathbf{A} and \mathbf{B} is rearranged such that the four null columns of \mathbf{B} appear at the end. The operation is performed by right multiplying both matrices by a 12×12 permutation matrix \mathbf{I}_p

$$\underbrace{[\mathbf{A}' \ \mathbf{A}'']}_{\mathbf{I}_p \alpha} \begin{bmatrix} \alpha' \\ \alpha'' \end{bmatrix} + \xi \underbrace{[\mathbf{B}' \ \mathbf{0}]}_{\mathbf{I}_p \alpha} \begin{bmatrix} \alpha' \\ \alpha'' \end{bmatrix} = 0 \quad (12)$$

Equation 13 is derived from 12 by suppressing the null columns. The degenerate solutions are discarded and the camera can be calibrated by finding the 20×1 vector β verifying the equation

$$\underbrace{[\mathbf{B}' \ \mathbf{A}' \ \mathbf{A}'']}_{\mathbf{C}} \underbrace{\begin{bmatrix} \xi \alpha' \\ \alpha' \\ \alpha'' \end{bmatrix}}_{\beta} = 0. \quad (13)$$

The solution $\beta \in \mathfrak{R}^{20}$, corresponding to the camera calibration, presents a specific structure. If $\beta = (\beta_1, \beta_2, \dots, \beta_{20})^t$ then vectors $(\beta_1, \dots, \beta_8)^t$ and $(\beta_9, \dots, \beta_{16})^t$ must be linearly dependent. Vectors verifying this property lie in a non-linear subspace that we will denote by \mathcal{M} , a manifold in \mathfrak{R}^{20} , defined by equation 14, with 13 degrees of freedom (DOF). Notice that the number of DOF of \mathcal{M} is equal to the number of unknowns in the calibration problem: 12 for the projection matrix \mathbf{P} (including scale) and 1 for the distortion parameter ξ .

$$\beta_1 \beta_{k+9} - \beta_9 \beta_{k+1} = 0, \quad k=1, \dots, 7 \quad (14)$$

In ideal circumstances matrix \mathbf{C} , provided in equation 13, is rank deficient and its null space intersects the manifold \mathcal{M} in a point $\beta \in \mathfrak{R}^{20}$, which encodes the camera calibration. However, due to noise in the data, matrix \mathbf{C} is in general full rank. In order to calibrate the camera, we must determine the vector, which minimizes the sum of the square distances ϵ , and verifies the constraints of equation 14

$$\epsilon(\beta) = \beta^t \mathbf{C}^t \mathbf{C} \beta \quad (15)$$

The constraints can be introduced in the objective function using Lagrange multipliers and the minimization could then be performed by gradient descendant methods. Strictly speaking the solution to problem requires the use of iterative minimization techniques. However an approximate solution can be found by solving two subspace problems. The proposed one-shot method determines the minima of function ϵ (equation 15), and then finds the vector in the manifold \mathcal{M} which is closest to the computed minima.

The vector $\beta = (\beta_1, \beta_2, \dots, \beta_{20})^t$ which minimizes the sum of the square distances ϵ is determined by applying the EYM theorem to matrix \mathbf{C} . In general the achieved solution does not lie on \mathcal{M} . In order to enforce the constraints of equation 14 we compute the vector $(\xi \alpha', \alpha', \alpha'')^t$ which is closest, in the least square sense, to β . Thus, we aim to find the solution (ξ, α', α'') which minimizes the following function

$$\gamma(\xi, \alpha', \alpha'') = (\xi\alpha' - \beta_{1\dots 8})^t(\xi\alpha' - \beta_{1\dots 8}) + (\alpha' - \beta_{9\dots 16})^t(\alpha' - \beta_{9\dots 16}) + (\alpha'' - \beta_{17\dots 20})^t(\alpha'' - \beta_{17\dots 20}) \quad (16)$$

Consider the 8×2 matrix $[\beta_{1\dots 8} \ \beta_{9\dots 16}]$, the corresponding SVD decomposition $[\beta_{1\dots 8} \ \beta_{9\dots 16}] = \mathbf{USV}^t$, and the 2×2 matrix \mathbf{S}_0 , obtained by zeroing the smallest diagonal element of \mathbf{S} . It is straightforward that the solution for α'' is the last 4×1 sub-vector of β . Moreover, accordingly to the EYM theorem, the solution for ξ and α' must be such that $[\xi\alpha' \ \alpha'] = \mathbf{US}_0\mathbf{V}^t$. The result is summarized on equation 17

$$\begin{cases} \alpha'' = \beta_{17\dots 20} \\ [\xi\alpha' \ \alpha'] = \mathbf{US}_0\mathbf{V}^t \end{cases} \quad (17)$$

4.4 Modifications to the Algorithm of Table 1

Some modifications have to be made to the calibration algorithm outlined in Tab. 1 to include the radial distortion. An additional **Step 0** is required at the start in order to correct the radial distortion in the cameras for which the calibration has already been estimated. Thus, for all views C_i , with $i = 1, \dots, M$, points \mathbf{x}_i are computed from the image points \mathbf{u}_i using the result of equation 8. Moreover at **Step 4.4** matrices \mathbf{A}_i and \mathbf{B}_i are generated accordingly to equation 10 and the parameter ξ , as well as vectors α' and α'' , are determined following the procedure outlined in section 4.3. On **Step 4.5** vectors ψ_i and ϕ are computed by $(\psi_i^t, \phi_i^t)^t = \mathbf{I}_p(\alpha'^t, \alpha''^t)^t$ where \mathbf{I}_p is the 12×12 permutation matrix of equation 12.

5 Performance Characterization

In this section we generate synthetic 640×480 images in order to characterize and evaluate the performance of the calibration algorithm. Consider four cameras, with a FOV close to 45° , positioned as depicted in Fig. 2(a). Each camera has different intrinsic parameters and radial distortion. The volume in front of the camera rig is uniformly sampled by 5000 virtual points. At each run a subset of N_p points, with $N_p \leq 5000$, is randomly selected accordingly to a uniform probability distribution. Notice that nothing guarantees that the chosen N_p points are simultaneously viewed by all cameras. The 3D points in the FOV of camera C_i ($i = 1, \dots, 4$) are projected in the corresponding image plane and the radial distortion is artificially introduced. Two-dimensional gaussian noise with zero mean and standard deviation σ is added to each image point.

Our goal is to calibrate the camera rig using the point correspondences between views. It is assumed that the projection matrices and the radial distortion of cameras C_1 (the reference view) and C_2 are known in advance. The calibration of the camera set is computed after 2 iterations of the algorithm. The errors in estimating the projection center and the radial distortion parameter are determined. The algorithm performance is characterized by computing the RMS error for the camera center, as well as the mean percentual relative error for the distortion, over 100 runs for each experiment. Henceforth we assume $N_p = 1000$ and $BL = 800$ unless otherwise stated (Fig. 2(a)).

Fig. 2(b) shows the error in estimating different magnitudes of radial distortion. These results were obtained by running simulations where cameras with different

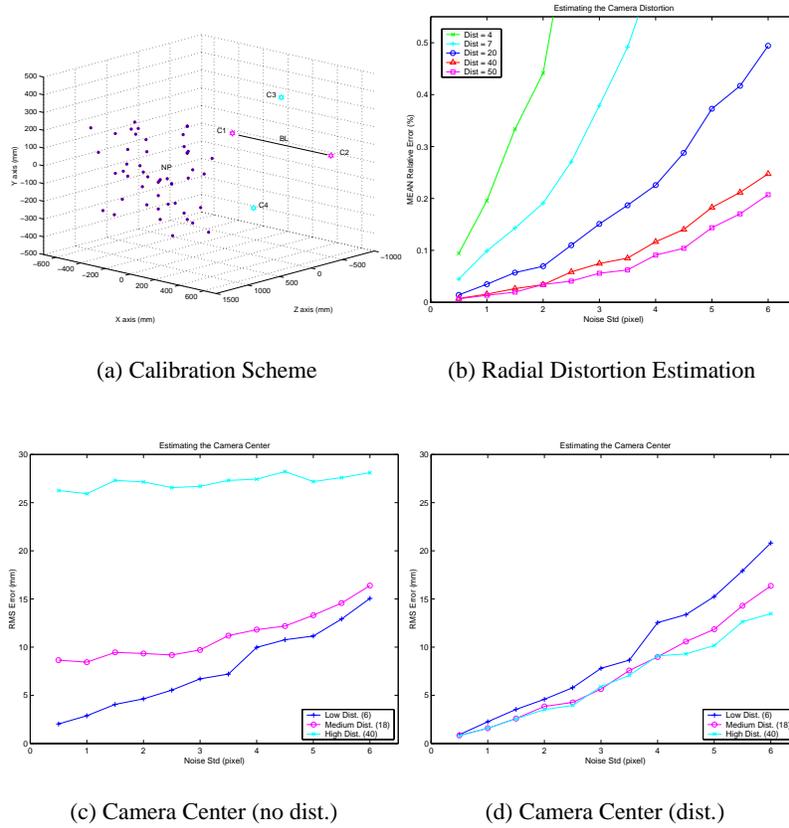


Fig. 2. Performance evaluation using synthetic images

amounts of distortion were mixed. The radial distortion is quantified by the deviation in pixels at the corner of the image. As expected, the estimation performance decreases with increasing noise. The increase on the mean relative error is much more pronounced in cameras with low distortion. If the radial distortion is not significant then the algorithm is not able to properly distinguish between distortion and noise. Nevertheless the estimation performance for moderate to high distortion is quite satisfactory. For $\sigma = 2.5$ the error in estimating radial distortion above 20 pixels is less than 10%.

The next experiment studies the algorithm behavior when cameras C_1 , C_2 , C_3 and C_4 present low, moderate, and high level of radial distortion. The curves on the left (Fig. 2 (c)) show the RMS errors for the projection center when the cameras are calibrated without taking into account the radial distortion (Tab 1). The graphic on the right (Fig. 2 (d)) illustrates the errors when the camera set is calibrated using the algorithm with the modifications proposed in section 4.4. In the presence of little or no distortion the calibration results are slightly worse for the modified version. This has already been observed in the results of Fig. 2(b). When the radial distortion is not very pronounced the estimator tends to confound noise and distortion. For moderate and

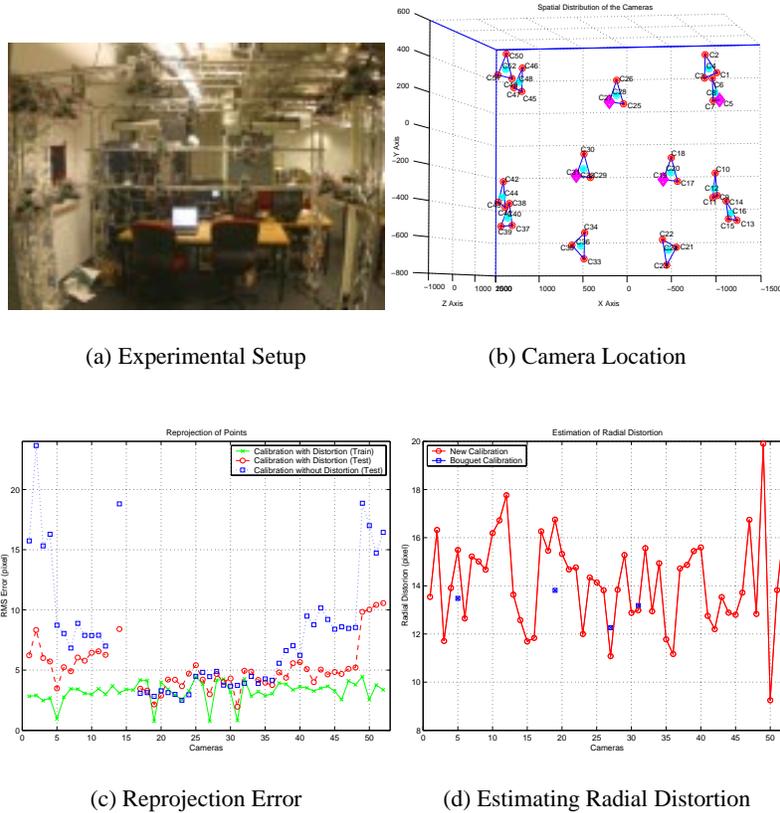


Fig. 3. Experimental results in calibrating a set of 52 cameras for tele-immersion application

high levels of distortion the modified version clearly outperforms the initial calibration method. Moreover the performance degradation due to increasing noise seems to be independent of the distortion magnitude. This suggests that the proposed calibration algorithm accurately estimates and compensates radial distortion.

6 Experimental Results

Fig. 3(a) is an image of our experimental setup used for tele-immersion. There are 52 cameras, grouped in 13 clusters of 4 cameras each, spread all over the room. Fig. 3(b), generated from the final calibration results, shows the location of the different cameras. An LED is waved through the room, in order to generate 2500 virtual points. The corresponding image points are determined by simple thresholding. Cameras 5, 19, 27 and 31 are initially calibrated using the Bouquet Calibration Toolbox [9].

The 3D points visible in at least two calibrated views are reconstructed [2]. From this set we select 100 test points, uniformly spread in the working volume. The remaining points (training set) are used in order to determine the projection matrices and the

radial distortion parameters of the 52 cameras. Fig. 3(d) compares the radial distortion, estimated using our method, with the distortion measured with Bouguet for the 4 initial cameras. The results present the same order of magnitude which is reasonable taking into account that the cameras are all similar. After calibrating, the training set is reconstructed and reprojected in the image plane. Fig. 3(c) shows, for each view, the corresponding RMS error. The error in reprojecting the test points, reconstructed using the Bouguet calibration, is also exhibited. Despite the fact that these test points are not ground truth, it is reasonable to claim that the obtained calibration is Euclidean. The third curve refers to the reprojection of the test points when the camera set is calibrated without taking into account the radial distortion. As can be observed the RMS error is significantly higher for the cameras laterally positioned in the room. The test points lying in the FOV of these cameras are projected at the image sides in the views initially calibrated with Bouguet. Due to the effect of distortion these points are not correctly reconstructed. This explains the observed RMS error and shows the importance of taking into account the radial distortion.

Acknowledgments

The authors are grateful for support through the following grants: NSF-IIS-0083209, NSF-IIS-0121293, NSF-EIA-0324977 and ARO/MURI DAAD19-02-1-0383.

References

1. Z. Zhang: *A Flexible New Technique for Camera Calibration*. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, 1330 – 1334, November 2000
2. T. Werner, T. Padjla and M. Urban: *REC3D: Toolbox for 3D Reconstruction from 2D Uncalibrated Views* Technical Report CTU-CMP-1999-4, Czech Technical University, 1999.
3. A. Fitzgibbon: *Simultaneous Linear Estimation of Multipleview Geometry and Lens Distortion*. in IEEE Int. Conf. on Computer Vision and Pattern Recognition. Hawaii, December 2001
4. H. Saito and T. Kanade: *Shape Reconstruction in Projective Grid Space for Large Number of Images*. in IEEE Int. Conf. on Computer Vision and Pattern Recognition, June 1999
5. P. Baker and Y. Aloimonos: *Complete Calibration of a Multiple Camera Network*. in Proc. of OMNI'2000 - Workshop on Omnidirectional Vision. Hilton Head, SC, June 2000
6. P. Baker and Y. Aloimonos: *Calibration of multicamera network*. in OMNI'2003 - Workshop on Omnidirectional Vision and Camera Networks. Madison, WI, June 2003
7. X. Chen, J. Davis and P. Slusallek: *Wide Area Camera Calibration Using Virtual Calibration Objects* in IEEE Int. Conf. on Computer Vision and Pattern Recognition, June 2000.
8. Y. Ma, S. Soatto, J. Kosecka and S. Sastry: *An Invitation to 3D Vision. From Images to Geometric Models* Interdisciplinary Applied Mathematics, Springer, 2003.
9. J Y Bouguet: *Camera Calibration Toolbox for Matlab*, 2003.
10. G. Golub and C. Van Loan: *Matrix Computations*, The John Hopkins University Press, 1983.
11. T. Svoboda: *Quick Guide to Multi-Camera Self Calibration* Technical Report BiWi-TR-263, Computer Vision Lab, Swiss Federal Institute of Technology, Zurich, August 2003.
12. R. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.
13. R. Willson and A. Shafer: *What is the center of the image?* in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition. 1993
14. B. Triggs: *Factorization Methods for Projective Structure and Motion*. In Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition. San Francisco, CA, 1996.