A Variational Approach to Problems in Calibration of Multiple Cameras

Gozde Unal[1], Anthony Yezzi[2], Stefano Soatto[3], and Greg Slabaugh[1]

Abstract

This paper addresses the problem of calibrating camera parameters using variational methods. One problem addressed is the severe lens distortion in low cost cameras. For many computer vision algorithms aiming at reconstructing reliable representations of 3D scenes, the camera distortion effects will lead to inaccurate 3D reconstructions and geometrical measurements if not accounted for. A second problem is the color calibration problem caused by variations in camera responses that result in different color measurements and affects the algorithms that depend on these measurements. We also address the extrinsic camera calibration that estimates relative poses and orientations of multiple cameras in the system, and the intrinsic camera calibration that estimates focal lengths and the skew parameters of the cameras. To address these calibration problems, we present multi-view stereo techniques based on variational methods that utilize partial and ordinary differential equations. Our approach can also be considered as a coordinated refinement of camera calibration parameters. To reduce computational complexity of such algorithms, we utilize prior knowledge on the calibration object, making a piecewise smooth surface assumption, and evolve the pose, orientation, and scale parameters of such a 3D model object without requiring a 2D feature extraction from camera views. We derive the evolution equations for the distortion coefficients, the color calibration parameters, the extrinsic and intrinsic parameters of the cameras, and present experimental results.

Index Terms

calibration, variational methods, color calibration, lens distortion calibration, camera parameters refinement.

[1]Siemens Corporate Research, Princeton NJ 08540. Corresponding author: gozde.unal@siemens.com

[2]School of Electrical Engineering, Georgia Institute of Technology, Atlanta GA 30332.

[3]Computer Science Department, University of California, Los Angeles, CA 90095.

I. Introduction

The problem of recovering a 3D representation of a scene from multiple 2D images has been one of the main research interests in computer vision. Many of the existing stereo techniques involve pre-processing the camera images to extract 2D features such as corners, lines, and contours of objects in the scene. These features are then used to find correspondences between camera views. In practice, searching for features and establishing correspondences is not an easy task due to noise and local extrema. Early variational approaches to the 3D reconstruction problem were pioneered by Faugeras et.al. [1] who also relied on local feature matching. A more recent variational approach by Yezzi and Soatto [2, 3] proposed a joint region-based image segmentation and simultaneous 3D stereo reconstruction technique. This paper addresses camera calibration techniques built on this later stereo reconstruction framework that avoids searches for local correspondences and is versatile enough to accommodate the new applications to be shown. A tradeoff is achieved by making a piecewise smooth object assumption and a constant background assumption, however, extraction of 2D features from given camera views are not required.

Camera calibration refers to the problem of finding the mapping between the 3D world and the camera or image plane. For most computer vision algorithms aimed at reconstructing reliable digital representations of 3D scenes, accurate camera calibrations are essential. There has been a great deal of research on camera calibration problem as early as in 70's [4]. In most of the previous techniques, some set of features are extracted from images of a known calibration pattern, and intrinsic camera parameters as well as camera pose and orientation (extrinsic camera parameters) are estimated by a minimization of an overall cost functional [5–13]. Many calibration techniques use both nonlinear minimization and closed form solutions as in [14].

In this paper, we develop a coordinated refinement technique for the extrinsic camera parameters, intrinsic camera parameters: lens distortion, focal lengths, skew, and also estimation of camera color

calibration parameters in a coupled way within a multiple camera system [1]. For geometrical measurements, an intrinsic camera parameter, the camera lens distortion, is an important issue, and will result in inaccurate 3D reconstructions if not taken into account. Another common problem in multi-view stereo techniques is caused by color miscalibrations between cameras due to different sensor characteristics. Extrinsic parameters of the cameras on the other hand, determine the relative poses and orientations of cameras, and their correct estimation is one of the first phases of a camera calibration system.

A. Relation to Previous Work and Contributions

1) Lens Distortion: The ideal pinhole camera model leads to imaging of world lines as lines on the image plane, and simplifies many computations and considerations [6]. However, for most real cameras with wide-angle or inexpensive lenses this assumption does not hold, and nonlinearities introduced by a well-known phenomenon referred to as a lens distortion should be taken into account. The corresponding distortion parameters should be estimated for each camera.

In many existing calibration techniques, good estimates for extrinsic and intrinsic camera parameters are first obtained by a pinhole camera model neglecting lens distortion. Then distortion calibration is performed while holding the other parameters fixed [17–19]. This is possible because the mapping from 3D world coordinates to the 2D image plane can be decomposed into a perspective projection and a mapping that models the deviations from the ideal pinhole camera.

A popular group of lens distortion calibration methods in the literature, mainly under the category known as plumb line methods, rely on a first step of extracting edges from the images. Either a user manually selects the image curves or there must be a way to reliably estimate image edges which correspond to linear 3D segments in the world. An optimization problem is set up by defining a measure of how much each detected segment is distorted. The curved lines in the image which do not really correspond to 3D line segments will constitute outliers in this optimization procedure [17, 20–23]. Other techniques such

[1]An initial version of this work that addresses lens distortion and color calibration can be found in [15] and [16], and an initial work addressing extrinsic camera calibration appears in [3].

as [24] rely on point correspondences. Given a set of 3D points, the associated epipolar and trilinear constraints are arranged into a tensor, which is computed with estimated distortion parameters at each step to minimize a reprojection error in an iterative manner. In another group of methods as in [25–27], a direct solution strategy is employed to find camera calibration parameters by incorporating lens distortion as well.

Our contribution is a new distortion calibration technique that does not rely on extraction of edges and search for point correspondences. The former may not be an easy task due to noise and local extrema. Instead, we devise an integrated calibration technique in which the distortion parameters of cameras are computed in a tightly coupled framework. The desired coupling of multiple camera views comes from estimating a common 3D object (in this case the calibration object). In other words, we minimize the cost between the reprojection of the 3D calibration object and the image measurements by evolving the distortion parameters of the cameras. In our distortion calibration algorithm, we use a white bar object, made from a foam core as shown in Fig.1 on the left. We capture its views before a dark background with the multi-view stereo rig system, a desktop multi-camera system designed for remote multimedia collaboration, developed by HP Labs [28]. The images of the calibration object captured from three of the five cameras in the rig are given in Fig.1. Many desktop multi-camera systems use wide angle and inexpensive cameras which produce severe distortion effects as can be observed in the given images.

Fig. 1. Three out of five camera views of the real calibration object shown on the left.

As we will show, with this technique we can also incorporate other parameters of calibration into the same variational framework and get their locally optimal estimates as well.

2) Color Calibration: Another common problem in multi-view stereo techniques is caused by color miscalibrations between cameras resulting from variations in camera responses due to different sensor

characteristics, ambient conditions like temperature, manufacturing differences, and so on. These yield different color measurements between cameras, and affect the algorithms that depend on these measurements. Camera color calibration refers to the problem of estimating the color calibration parameters of cameras to overcome these unwanted effects. A common approach taken toward this problem is to calibrate each camera independently through comparisons with known colors on a color calibration object/environment [28, 29].

The color calibration object we use, shown in Fig.2 is a color cube with patches of known colors whose

Fig. 2. Photograph of the color calibration object.

images are captured from each camera. Demosaicing coefficients are calculated independently for each camera based upon the absolute colors of the calibration object and the measured color responses of each camera. Slight errors and differences that arise from this independent calibration procedure sometimes lead to noticeable seams or discontinuities in the texture mapping process during the transition of the texture map between neighboring cameras. Our goal is to help even out these discrepancies by devising a *relative inter-camera* color calibration technique in which the demosaicing parameters of cameras are calculated jointly in a tightly coupled framework rather than just one camera at a time.

Similar to our approach to lens distortion calibration, the desired coupling of the multiple camera views comes from estimating a common 3D shape, and in addition a common radiance function for the calibration object (in this case, the color cube). We take advantage of the fact that the object shape is known up to location and scale to simplify the problem. Hence, we estimate the pose parameters of the cube, the radiance function on the cube, and the color calibration coefficients for each camera.

3) Extrinsic and Intrinsic Calibration: Following the same philosophy as mentioned in the other two calibration problems above, extrinsic and intrinsic calibration parameters can be estimated in a variational framework using the general stereoscopic framework of Yezzi-Soatto.

It should be noted that due to differential nature of the estimation equations derived, the extrinsic and intrinsic update equations require rough initial values. This is a well-known feature of almost all of the recent state-of-the-art energy functionals used in segmentation (e.g., Mumford-Shah energy, geodesic energy, ...), i.e., the solutions are locally optimal, hence starting far away from the real solution may lead to solutions that get stuck at local extrema far from the desired solution. Nevertheless, the usefulness of a refinement stage in extrinsic and intrinsic camera parameters will be demonstrated via the improvement in the final 3D reconstructions. A nice feature of the methodology presented in this paper is that it can integrate several different problems in geometric and color calibration into an overall unified system based on the joint segmentation framework to evolve pose, color, distortion, extrinsics, and intrinsics.

The organization of this paper is as follows. We first present a variant of the Yezzi-Soatto algorithm in which a 3D object is allowed to move with a semi-affine motion model in Section II. We developed this scheme for our applications in calibration, where the 3D object shape is roughly known (up to 3 scales and rigidity) to obtain more efficient and faster algorithms. We then present a novel technique for lens distortion calibration in Section III and a novel technique for relative inter-camera color calibration in Section IV. We apply the same calibration ideas for intrinsic camera calibration in Section V, and for extrinsic camera calibration problem in Section VI. Conclusions and discussions are given in Section VII.

II. Evolution Equations of 3D Object Motion Parameters

The Yezzi-Soatto 3D stereo reconstruction model builds a cost on the discrepancy between the reprojection of a model surface with a radiance $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$, the background (infinitely far away) with radiance $b : \mathbb{R}^3 \longrightarrow \mathbb{R}$, and the actual measurements from multiple camera views. Let g_i denote the transformation from world coordinates to camera coordinates: $g_i : \mathbf{X} \longrightarrow \mathbf{X}_i = (X_i, Y_i, Z_i)^T$, and π denote the perspective transformation from camera frame to the image plane: $\pi : \mathbf{X}_i \longrightarrow \mathbf{x}_i = (x_i = \frac{X_i}{Z_i}, y_i = \frac{Y_i}{Z_i})^T$.

On the image plane, the cost functional for the Yezzi-Soatto model can be written as a joint segmentation problem over regions of n camera images I_i with domain $\Omega_i = R_i \cup R_i^c$ (R_i denotes the foreground region),

and with 3 color channels $k \in (R, G, B)$:

$$E = \sum_{k=R,G,B} \sum_{i=1}^{n} \int_{R_i} [f^k((\pi \circ g_i)^{-1}(\mathbf{x}_i)) - I_i^k(\mathbf{x}_i)]^2 d\Omega_i + \sum_{k=R,G,B} \sum_{i=1}^{n} \int_{R_i^c} [b^k - I_i^k]^2 d\Omega_i \tag{1}$$

This energy can be lifted back onto surface S :

$$E(S) = \sum_{k=R,G,B} \sum_{i=1}^{n} \int_S [(f^k(\mathbf{X}) - I_i^k(\pi \circ g_i(\mathbf{X})))^2 - (b^k - I_i^k)^2] \ \mathcal{X}_i(\mathbf{X}) \sigma(\mathbf{X}_i) dA, \tag{2}$$

where σ is the Jacobian of the change of coordinates from the image plane to the surface, $\mathcal{X}_i$ is the visibility function of a voxel on the surface, and dA is the area measure of surface S. The deformation of the surface S w.r.t. this energy or data fidelity measure is then obtained by finding the partial differential equation (PDE) that is the gradient descent flow of the energy E. A popular class of numerical techniques known as Level Sets Methods [30], is utilized to evolve the surface S via the evolution of a 3D function $\Psi : \mathbb{R}^3 \longrightarrow \mathbb{R}$. Nevertheless, an update of the level set function is required after each iteration of the associated PDE, and even with more efficient narrowband schemes [31], there is a considerable amount of computation involved. For our intended applications, in which there is a calibration object whose shape can be roughly known a priori, rather than deforming the surface of the 3D object, we will evolve its pose and scale parameters instead. Next, using the energy E in Eq. 2 we will derive the ordinary differential equations (ODEs) to update the parameters of the surface motion modeled by a semi-affine transformation, which is more general than a similarity but less general than a fully affine transformation.

Let the original rigid surface be denoted by S_o, then $S = g^s(S_o)$, or $\mathbf{X} = g^s(\mathbf{X}_o) = \mathbf{R}_s\mathbf{X}_o + \mathbf{T}_s$, and let λ denote parameters of the rigid motion g^s of the surface S_o with rotation $\mathbf{R}_s$ and translation $\mathbf{T}_s$. Then the gradient of the energy E w.r.t. λ is given by:

$$\begin{aligned}
\frac{\partial E(\lambda)}{\partial \lambda} &= \sum_{k=R,G,B} \int_S \sum_i F_i^k(\mathbf{X}) < \frac{\partial \mathbf{X}}{\partial \lambda}, \mathbf{N} > \ dA + \frac{\partial F_i^k(\mathbf{X})}{\partial \lambda} \ dA \\
&= \sum_{k=R,G,B} \int_{S_o} \sum_i F_i^k(g^s(\mathbf{X}_o)) < \frac{\partial (g^s \mathbf{X}_o)}{\partial \lambda}, \mathbf{R}_s \mathbf{N}_o > dA_o \\
&\quad + 2(f^k(g\mathbf{X}_o) - I_i^k) < \frac{\partial (g^s \mathbf{X}_o)}{\partial \lambda}, \nabla f_S^k > dA_o
\end{aligned} \tag{3}$$

where $F_i^k = [(f^k(\mathbf{X}) - I_i^k(\pi \circ g_i(\mathbf{X})))^2 - (b^k - I_i^k)^2]\ \mathcal{X}_i(\mathbf{X})\sigma(\mathbf{X}_i)$ is the Mumford-Shah term from Eq. 2 (also in [2]). The derivation follows from shape optimization tools [32] that provide the shape derivatives in curve and surface evolution framework. $\mathbf{N}$ denotes the surface normal vector. Note that the visibility function $\mathcal{X}_i(g^s(\mathbf{X}_o))$, included in the data term $F_i^k(\cdot)$ is computed using the original visibility function but compensated by $\mathbf{R}_s^T(\mathbf{C}_i - \mathbf{T}_s)$), where $\mathbf{C}_i$ is a camera center. The second term in Eq.(3) is the region term corresponding to the foreground object whereas the first one is the boundary term. In our applications, the background is modeled by a piecewise constant radiance, therefore we omit the background region term in the equation.

For translation parameters:

$$< \frac{\partial(g^s\mathbf{X}_o)}{\partial\lambda}, \mathbf{R}_s\mathbf{N}_o >= \mathbf{R}_s\mathbf{N}_o.$$

For rotation parameters:

$$< \frac{\partial(g^s\mathbf{X}_o)}{\partial\lambda}, \mathbf{R}_s\mathbf{N}_o >=< \mathbf{R}_s \underbrace{\begin{bmatrix} 0 & Z_o & -Y_o \\ -Z_o & 0 & X_o \\ Y_o & -Xo & 0 \end{bmatrix}}_{\hat{\mathbf{X}}_o}, \mathbf{R}_s\mathbf{N}_o >=< -\mathbf{R}_s\hat{\mathbf{X}}_o, \mathbf{R}_s\mathbf{N}_o >, \tag{4}$$

where we utilize exponential coordinates (see [33] for details on this representation) for the global rotation parameters of the surface. We note that a matrix in an inner product expression, when operated on a vector, will incorporate each of its row vectors in the inner product to result in a vector: $< \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, \mathbf{y} >= (< \mathbf{x}_1, \mathbf{y} >, < \mathbf{x}_2, \mathbf{y} >, ..., < \mathbf{x}_n, \mathbf{y} >)$.

For further flexibility in initializing a model surface, we add three scaling parameters along the $X, Y,$ and Z axes. Then the semi-affine transformation for a point $\mathbf{X}_o$ on the surface becomes: $\mathbf{X} = g^s(\mathbf{X}_o) = \mathbf{RSX}_o + \mathbf{T}$, where $\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$. The gradient of the energy w.r.t. the scaling parameters $\lambda = s_j$ is derived similarly to the above:

$$\sum_{k=R,G,B} \int_{S_o} \sum_i F_i^k(g^s(\mathbf{X}_o)) < \frac{\partial(g^s\mathbf{X}_o)}{\partial\lambda}, \mathbf{R}_s\mathbf{N}_o > dA_o$$

where

$$
\begin{aligned}
< \frac{\partial (g^s \mathbf{X}_o)}{\partial \lambda}, \mathbf{R}_s \mathbf{N}_o > &= < \mathbf{R}_s \frac{\partial S}{\partial \lambda} \mathbf{X}_o, \mathbf{R}_s \mathbf{N}_o > \quad \text{with e.g.} \\
\frac{\partial (g^s \mathbf{X}_o)}{\partial s_1} &= \mathbf{R}_s \frac{\partial S}{\partial s_1} \mathbf{X}_o = \mathbf{R}_s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{X}_o = \mathbf{R}_s \begin{bmatrix} \mathbf{X}_o.x \\ 0 \\ 0 \end{bmatrix} \\
< \frac{\partial (g^s \mathbf{X}_o)}{\partial \lambda}, \mathbf{R}_s \mathbf{N}_o > &= < \underbrace{\mathbf{R}_s \begin{bmatrix} \begin{bmatrix} X_o \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ Y_o \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ Z_o \end{bmatrix} \end{bmatrix}}_{\mathbf{R}_s^X}, \mathbf{R}_s \mathbf{N}_o > . \qquad (5)
\end{aligned}
$$

The evolutions for the rigid motion parameters λ are then given by the gradient descent equations:

$$
\frac{\partial \lambda}{\partial t} = -\frac{\partial E}{\partial \lambda} = - \sum_{k=R,G,B} \int_{S_o} \underbrace{\sum_i F_i^k (g^s(\mathbf{X}_o))}_{F^k} \; \mathbf{R}_s \mathbf{N}_o \; dA_o, \qquad \text{(for translation)}. \qquad (6)
$$

$$
\frac{\partial \lambda}{\partial t} = -\frac{\partial E}{\partial \lambda} = - \sum_{k=R,G,B} \int_{S_o} F^k \; < -\mathbf{R}_s \hat{\mathbf{X}}_o, \mathbf{R}_s \mathbf{N}_o > \; dA_o, \qquad \text{(for rotation)}. \qquad (7)
$$

$$
\frac{\partial \lambda}{\partial t} = -\frac{\partial E}{\partial \lambda} = - \sum_{k=R,G,B} \int_{S_o} F^k \; < \mathbf{R}_s^X, \mathbf{R}_s \mathbf{N}_o > \; dA_o, \qquad \text{(for scaling)}. \qquad (8)
$$

Here note that the visibility function $\mathcal{X}_i(g^s(\mathbf{X}_o))$ is computed using the original visibility function but compensated by the $\mathbf{S}^{-1}\mathbf{R}_s^T(\mathbf{C}_i - \mathbf{T}_s)$), where $\mathbf{S}^{-1} = \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1/s_z \end{bmatrix}$. Note that we can generalize this idea in a straightforward fashion by considering $\mathbf{S}$ to be more general than a simple diagonal matrix in order to accommodate a fully affine motion of the surface.

We will use equations (6-7-8) in updating the pose of the surface S to estimate its correct placement in the 3D space for the calibration applications presented in Sections III, and IV.

III. Lens Distortion Calibration

The lens distortion is usually modeled by a function defined from the ideal image plane to the distorted image plane. One approach is to decompose it into two terms: radial and tangential distortion [17]. The radial distortion is a deformation along the radial direction from a center of distortion point to an image point, and the tangential distortion is a deformation in a direction perpendicular to the radial direction, and is negligible for many cameras. To model the radial distortion effects, a commonly used distortion function $D(r)$ is given by $(1 + k_1 r^2 + k_2 r^4 + ...)$ where r is the radius from the center of distortion to a point on the ideal image plane. The principal point (u_0, v_0) is often used as the center for radial distortion [6], which we will also adopt. Below $\hat{\mathbf{x}}_i$ is the distorted image coordinates, and D is the distort function:

$$\hat{\mathbf{x}}_i = D\mathbf{x}_i = (1 + k_1^i r^2 + k_2^i r^4 + ...)\mathbf{x}_i, \tag{9}$$

$r^2 = (x_i^2 + y_i^2)$, and k_j^i is the j^{th} distortion coefficient for camera i. In Eq. 9, we assume that $k_0 = 1$, which can be changed to an arbitrary k_0 value.

A. Calibration of the Lens Distortion Parameters

Notation: World $\mathbf{x} \underbrace{\longrightarrow}_{g_i} \mathbf{x}_i \underbrace{\longrightarrow}_{\pi} \mathbf{x}_i = \begin{pmatrix} \frac{X_i}{Z_i} = x_i \\ \frac{Y_i}{Z_i} = y_i \\ 1 \end{pmatrix} \underbrace{\longrightarrow}_{\begin{matrix} L_u & 0 & u_0 \\ 0 & L_v & v_0 \\ 0 & 0 & 1 \end{matrix}} (u, v)$ (image coordinates), where D is the distort function in Eq. 9, and L_u and L_v are the focal lengths. The gradient of the energy (1), assuming a single image channel over the distorted image plane, w.r.t. distortion parameters k_j^i is given by

$$\frac{\partial k_j^i}{\partial t} = -\frac{\partial E}{\partial k_j^i} = -\int_{\hat{c}_i} F_i((D \circ \pi \circ g_i)^{-1}\hat{\mathbf{x}}_i) < \frac{\partial \hat{\mathbf{x}}_i}{\partial k_j^i}, \hat{\mathbf{n}}_i > d\hat{s} \tag{10}$$

where $F_i = (f - I_i)^2 - (b - I_i)^2$, subscript i corresponds to each camera view, and $\hat{\mathbf{n}}_i$ denotes the normal vector to the occluding boundary $\hat{c}_i$ of region R_i on the distorted image plane. We only consider the boundary term ($\hat{s}$ is the arclength of the contour $\hat{c}_i$ on the image plane: the distorted or actual image coordinates) as we assume the foreground and background have constant radiance. We design the lens distortion calibration object to satisfy this assumption.

We want to lift this integral back onto occluding boundary C_i of the surface. Note that $\frac{\partial \hat{\mathbf{x}}_i}{\partial k_j}$ are given by

$$\frac{\partial \hat{\mathbf{x}}_i}{\partial k_1^i} = r^2 \mathbf{x}_i, \quad \frac{\partial \hat{\mathbf{x}}_i}{\partial k_2^i} = r^4 \mathbf{x}_i, \quad ... \ \frac{\partial \hat{\mathbf{x}}_i}{\partial k_j^i} = r^{2j} \mathbf{x}_i$$

hence

$$< \frac{\partial \hat{\mathbf{x}}_i}{\partial k_j^i}, \hat{\mathbf{n}}_i > d\hat{s} =< r^{2j} \pi(\mathbf{X}_i), J \frac{\partial}{\partial s}(D \circ \pi)\mathbf{X}_i > ds =< r^{2j} \pi(\mathbf{X}_i), JD' \circ \pi' \frac{\partial}{\partial s}\mathbf{X}_i > ds \tag{11}$$

where J denotes the 2×2 ninety degree rotation matrix, $D' = (1 + k_1 r^2 + k_2 r^4 + ...)$, and

$$\pi' = \frac{1}{Z_i^2} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \end{bmatrix} \tag{12}$$

is the Jacobian of the perspective projection π. We can continue to simplify:

$$\begin{aligned} & < \frac{\partial \hat{\mathbf{x}}_i}{\partial k_j^i}, \hat{\mathbf{n}}_i > d\hat{s} = r^{2j} D' < [\pi(\mathbf{X}_i)]_{2\times 1}, \frac{1}{Z_i^2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \end{bmatrix} \left[\frac{\partial \mathbf{X}_i}{\partial s}\right]_{3\times 1} > ds \\ = & \frac{r^{2j} D'}{Z_i^2} < \begin{pmatrix} X_i \\ Y_i \end{pmatrix} \frac{1}{Z_i}, \begin{bmatrix} 0 & Z_i & -Y_i \\ -Z_i & 0 & X_i \end{bmatrix} \left[\frac{\partial \mathbf{X}_i}{\partial s}\right]_{3\times 1} > ds \\ = & \frac{r^{2j} D'}{Z_i^3} < \begin{bmatrix} 0 & -Z_i \\ Z_i & 0 \\ -Y_i & X_i \end{bmatrix} \begin{pmatrix} X_i \\ Y_i \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} > ds = \frac{r^{2j} D'}{Z_i^3} < \begin{pmatrix} -Z_i Y_i \\ Z_i X_i \\ 0 \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} > ds \end{aligned}$$

Noting that $\begin{pmatrix} -Z_i Y_i \\ Z_i X_i \\ 0 \end{pmatrix} = \mathbf{X}_i \times \begin{pmatrix} X_i \\ Y_i \\ 0 \end{pmatrix}$, we have $< \frac{\partial \hat{\mathbf{x}}_i}{\partial k_j}, \hat{\mathbf{n}}_i > d\hat{s} = \frac{r^{2j} D'}{Z_i^3} < - \begin{pmatrix} X_i \\ Y_i \\ 0 \end{pmatrix} \times \mathbf{X}_i, \frac{\partial \mathbf{X}_i}{\partial s} > ds$, and

$$< \frac{\partial \hat{\mathbf{x}}_i}{\partial k_j^i}, \hat{\mathbf{n}}_i > d\hat{s} = -\frac{r^{2j} D'}{Z_i^3} < \mathbf{X}_i \times \frac{\partial \mathbf{X}_i}{\partial s}, \begin{pmatrix} X_i \\ Y_i \\ 0 \end{pmatrix} > ds = -\frac{r^{2j} D'}{Z_i^3} < ||\mathbf{X}_i|| \mathbf{N}_i, \begin{pmatrix} X_i \\ Y_i \\ 0 \end{pmatrix} > ds. \tag{13}$$

Substituting Eq.(13) into Eq.(10), we get the calibration equation

$$\frac{\partial k_j^i}{\partial t} = \int_{C_i} F_i \ \frac{r^{2j} D' ||\mathbf{X}_i||}{Z_i^3} < \mathbf{N}_i, \begin{pmatrix} X_i \\ Y_i \\ 0 \end{pmatrix} > ds \tag{14}$$

for the lens distortion parameters k_j^i. Note that the distortion calibration method we propose can handle different models of distortion by changing the D function, and related derivatives in Eq.(14).

B. Using Several Poses of the Object

When camera views from multiple poses of the object are available, we can take advantage of the existence of variously distorted views in calibrating the lens distortion. In the first phase, we estimate both pose and distortion coefficients from separate experiments. To simplify the explanation, let us assume that we want to solve for only one distortion coefficient k_1^i for each camera i. Once we obtain rough estimates for the object pose and distortion coefficients k_1^i, we can fuse a "common distortion" $\tilde{k}_1^i$ from these separate experiments for each camera i and then jointly evolve $\tilde{k}_1^i$'s as follows:

$$\frac{\partial \tilde{k}_1^i}{\partial t} = \sum_{m=1}^{M\text{poses}} \int_{C_{i,m}} F_{i,m} \, \frac{r^{2j} D' ||\mathbf{X}_{i,m}||}{Z_{i,m}^3} < \mathbf{N}_{i,m}, \begin{pmatrix} X_{i,m} \\ Y_{i,m} \\ 0 \end{pmatrix} > ds. \tag{15}$$

At the same time, we evolve the pose parameters of separate poses of the object as described in Section II, the only difference being the incorporation of the new "common distortion" in the equations. For instance, we evolve any of them for a given pose as follows:

$$\frac{\partial \lambda}{\partial t} = -\sum_i \int_{S_o} F_i(g^s(\mathbf{X}_o)) < \frac{\partial (g^s \mathbf{X}_o)}{\partial \lambda}, \mathbf{R}_s \mathbf{N}_o > dA_o \tag{16}$$

where F_i includes computation of $I_i(D \cdot \pi \cdot g_i \ (g^s(\mathbf{X}_o)))$ with the new common distortion coefficients $\tilde{k}_1^i$ in the multiplying distortion factor D.

C. Experimental Results

For our calibration algorithm, we initialize a surface model of the real calibration object which is shown from several vantage points in Fig.3. After initializing the surface, the first phase of our algorithm is to evolve its pose parameters to position the 3D object model roughly in the correct location in 3D space. For the experiments presented here captured via HP Labs' stereo rig system, we used three different poses of the calibration object, but we can increase the number of poses used in the process. Example evolutions

of the pose parameters are shown in Fig.4, for three different pose captures of the calibration object in each column (showing only one camera view for each pose). The distortion coefficients are also evolved at a slower pace. That is, the time step used in the associated ODE is small in the first phase. In the experiments, the distortion function D in (9) with one distortion coefficient k_1 for each camera is used.

After the separate evolutions for each of the poses have converged, common initial distortion coefficients are computed as the average of the results from phase 1. In the second phase of the algorithm, we evolve the distortion coefficients for each camera again separately but summed over different poses. We show sample views of pose 1, 2, and 3 in row 1 of Fig.s 5-6-7. As the distortion coefficients converge to true values, the reprojection of the surfaces onto the distorted views results in a better match to the image data and continues to minimize the overall energy. Such images with reprojections are shown on the second row of Figures 5-6-7. The undistorted views shown as well on the third row. The straightening effect of this operation on the curved lines can be clearly observed in these images.

IV. Color Calibration

For color calibration, the differences in absolute colors measured in the response of each camera are modeled by a simple multiplicative factor in each of color RGB channel measurements and an additive offset parameter.

The first variation of our energy functional E using this model leads to gradient descent flows:

$$\frac{\partial E}{\partial \alpha_{i,k}} = -\int_{R_i} [f^k - (\alpha_{i,k} I_i^k + \beta_{i,k})] I_i^k d\Omega_i - \int_{R_i^c} [b^k - (\alpha_{i,k} I_i^k + \beta_{i,k})] I_i^k d\Omega_i, \tag{17}$$

$$\frac{\partial E}{\partial \beta_{i,k}} = -\int_{R_i} [f^k - (\alpha_{i,k} I_i^k + \beta_{i,k})] d\Omega_i - \int_{R_i^c} [b^k - (\alpha_{i,k} I_i^k + \beta_{i,k})] d\Omega_i. \tag{18}$$

for the color calibration parameters $\alpha_{i,k}$ and $\beta_{i,k}$ for each camera i, and $k \in \{R, G, B\}$, where I_i^k, f^k, and b^k are from one of the three color channels $\{R, G, B\}$. Note that one can extend this framework to RGGB images in a straightforward fashion.

In our test calibration experiments, we utilized white noise additive offsets and multiplicative scaling coefficients to perturb the measured images, thereby exaggerating the effect of color miscalibrations. On

a synthetically created example in Fig.8, where the correct geometry and radiance function are known, we show such miscalibration effects on the original views, and views during the evolution of α_i's and β_i's in Eq.s (17-18), and views after these parameters have converged. In addition, in Fig.9, the curves depict the true α and β values for all nine camera views, and the convergence of the estimated parameters towards the real values.

Similarly in Figure 10, the color cube with original colors are shown from some camera views first, then shown after their color calibration parameters are perturbed. Finally, the convergence of the color parameters results in a corrected set of colors as shown in the views. Also shown in Figure 11 are the evolutions of the color calibration parameters for the shown views. We have to note here again that due to relative calibration framework among cameras, the updated parameters may not always result in absolute values but still provide useful outputs for the multi-camera systems.

V. Intrinsic Parameter Calibration

We show the evolution of three of the main intrinsic camera parameters: focal lengths, denoted by L_u and L_v for each of the coordinates on the image plane, and the skew parameter a. Inclusion of the skew parameter between the two plane coordinates leads to an intrinsics matrix of the form

$$\pi = \begin{pmatrix} L_u & a & u_0 \\ 0 & L_v & v_0 \\ 0 & 0 & 1 \end{pmatrix},$$

then the Jacobian of the perspective transformation becomes (compare to Eq. 12):

$$\pi' = \begin{pmatrix} L_u/Z_i & a/Z_i & -L_u X_i/Z_i^2 - aY_i/Z_i^2 \\ 0 & L_v/Z_i & -L_v Y_i/Z_i^2 \end{pmatrix} = \frac{1}{Z_i^2} \begin{pmatrix} L_u Z_i & aZ_i & -L_u X_i - aY_i \\ 0 & L_v Z_i & -L_v Y_i \end{pmatrix}.$$

The derivatives of the image coordinates w.r.t. each of the intrinsic parameters are computed from the overall energy functional as before (similar to our derivations of the lens distortion calibration parameters in Section III):

$$\frac{\partial E}{\partial L_j} = \sum_{k=R,G,B} \int_{C_i} \underbrace{[(f^k - I_i^k)^2 - (b^k - I_i^k)^2]}_{F^k} \langle \frac{\partial \mathbf{x}_i}{\partial L_j}, \mathbf{n}_i \rangle ds.$$

For the focal length parameter L_u, we have

$$
\begin{aligned}
< \frac{\partial \mathbf{x}_i}{\partial L_u}, \mathbf{n}_i > ds &= \left\langle \frac{\partial \pi \mathbf{C}_i}{\partial L_u}, \frac{\partial}{\partial s} J \pi' C_i \right\rangle ds \\
&= \left\langle \frac{\partial}{\partial L_u} \begin{pmatrix} L_u X_i/Z_i + a Y_i/Z_i \\ 0 \end{pmatrix}, \frac{1}{{Z_i}^2} \begin{bmatrix} 0 & L_v Z_i & -L_v Y_i \\ -L_u Z_i & -a Z_i & L_u X_i + a Y_i \end{bmatrix} \left[\frac{\partial \mathbf{X}_i}{\partial s} \right] \right\rangle ds \qquad (19) \\
&= \frac{1}{Z_i^3} \left\langle \begin{bmatrix} 0 & -L_u Z_i \\ L_v Z_i & -a Z_i \\ -L_v Y_i & L_u X_i + a Y_i \end{bmatrix} \begin{pmatrix} X_i \\ 0 \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} > ds = \frac{1}{Z_i^3} < L_v \begin{pmatrix} 0 \\ Z_i X_i \\ -Y_i X_i \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} \right\rangle ds.
\end{aligned}
$$

Noting that $\begin{pmatrix} 0 \\ Z_i X_i \\ -Y_i X_i \end{pmatrix} = \mathbf{X}_i \times \begin{pmatrix} X_i \\ 0 \\ 0 \end{pmatrix}$, then for the focal length parameter L_u we obtain:

$$
\begin{aligned}
< \frac{\partial \mathbf{x}_i}{\partial L_u}, \mathbf{n}_i > ds &= \frac{1}{Z_i^3} < -L_v \begin{pmatrix} X_i \\ 0 \\ 0 \end{pmatrix} \times \mathbf{X}_i, \frac{\partial \mathbf{X}_i}{\partial s} > ds \\
&= -\frac{L_v}{Z_i^3} < \mathbf{X}_i \times \frac{\partial \mathbf{X}_i}{\partial s}, \begin{pmatrix} X_i \\ 0 \\ 0 \end{pmatrix} > ds = -\frac{L_v ||\mathbf{X}_i||}{Z_i^3} < \mathbf{N}_i, \begin{pmatrix} X_i \\ 0 \\ 0 \end{pmatrix} > ds. \qquad (20)
\end{aligned}
$$

Due to the skew parameter, the equations for the second focal length parameter L_v will be slightly different. This time incorporating the derivative w.r.t. L_v in Eq. 19 :

$$
< \frac{\partial \mathbf{x}_i}{\partial L_v}, \mathbf{n}_i > ds = \frac{1}{Z_i^3} < \begin{bmatrix} 0 & -L_u Z_i \\ L_v Z_i & -a Z_i \\ -L_v Y_i & L_u X_i + a Y_i \end{bmatrix} \begin{pmatrix} 0 \\ Y_i \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} > ds = \frac{1}{Z_i^3} < \begin{pmatrix} -L_u Z_i Y_i \\ -a Z_i Y_i \\ L_u X_i Y_i + a Y_i^2 \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} > ds
$$

Again noting that $\begin{pmatrix} -L_u Y_i Z_i \\ -a Z_i Y_i \\ L_u X_i Y_i + a Y_i^2 \end{pmatrix} = \mathbf{X}_i \times \begin{pmatrix} -a Y_i \\ L_u Y_i \\ 0 \end{pmatrix}$, then for the focal length parameter L_v, we have:

$$
< \frac{\partial \mathbf{x}_i}{\partial L_v}, \mathbf{n}_i > ds = -\frac{||\mathbf{X}_i||}{Z_i^3} < \mathbf{N}_i, \begin{pmatrix} -a Y_i \\ L_u Y_i \\ 0 \end{pmatrix} > ds. \qquad (21)
$$

Note that when the skew parameter a is 0, which is a widely used convention, the above equation reduces to a symmetric form of the Eq. 20 derived for L_u.

Finally, we derive similarly the update equations for the skew parameter a:

$$< \frac{\partial \mathbf{x}_i}{\partial a}, \mathbf{n}_i > ds = \frac{1}{Z_i^3} < \begin{bmatrix} 0 & -L_u Z_i \\ L_v Z_i & -aZ_i \\ -L_v Y_i & L_u X_i + aY_i \end{bmatrix} \begin{pmatrix} Y_i \\ 0 \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} >= \frac{1}{Z_i^3} < L_v \begin{pmatrix} 0 \\ Z_i Y_i \\ -Y_i^2 \end{pmatrix}, \frac{\partial \mathbf{X}_i}{\partial s} > ds.$$

This time noting that $\begin{pmatrix} 0 \\ Z_i Y_i \\ -Y_i^2 \end{pmatrix} = \mathbf{X}_i \times \begin{pmatrix} Y_i \\ 0 \\ 0 \end{pmatrix}$, then for the skew parameter a, we have:

$$< \frac{\partial \mathbf{x}_i}{\partial a}, \vec{n_i} > ds \quad = \quad -\frac{L_v ||\mathbf{X}_i||}{Z_i^3} < \mathbf{N}_i, \begin{pmatrix} Y_i \\ 0 \\ 0 \end{pmatrix} > ds. \tag{22}$$

The final evolution equations for the three intrinsic parameters for each camera i are then given by:

$$\frac{\partial L_{u,i}}{\partial t} \quad = \quad - \sum_{k=R,G,B} \int_{C_i} F^k \frac{L_v}{Z_i^3} < ||\mathbf{X}_i|| \mathbf{N}_i, \begin{pmatrix} X_i \\ 0 \\ 0 \end{pmatrix} > ds \tag{23}$$

$$\frac{\partial L_{v,i}}{\partial t} \quad = \quad - \sum_{k=R,G,B} \int_{C_i} F^k \frac{1}{Z_i^3} < ||\mathbf{X}_i|| \mathbf{N}_i, \begin{pmatrix} -aY_i \\ L_u Y_i \\ 0 \end{pmatrix} > ds \tag{24}$$

$$\frac{\partial a_i}{\partial t} \quad = \quad - \sum_{k=R,G,B} \int_{C_i} F^k \frac{L_v}{Z_i^3} < ||\mathbf{X}_i|| \mathbf{N}_i, \begin{pmatrix} Y_i \\ 0 \\ 0 \end{pmatrix} > ds \tag{25}$$

In Figure 12 a synthetic color cube example is shown. The intrinsic parameters, focal lengths L_u, and L_v, are initialized to perturbed values and when the intrinsic calibration update equations have converged, both the projections of the cube surface onto the images and the evolution of the focal lengths are shown.

VI. Extrinsic Camera Calibration

We now consider the same energy functional as a function of the extrinsic calibration parameters $\Lambda_i = (\lambda_{i1}, \ldots, \lambda_{i6})$ for each camera image I_i. Notice that the only term in our energy functional E which depends upon Λ_i is the corresponding fidelity term in E_{data} (due to the dependence of π_i^{-1}), assuming a constant background radiance in the scene :

$$E_{data,i}(S, f, b, \Lambda_i) = \sum_{k=R,G,B} \int_{R_i} \left(f^k(\pi_i^{-1}(\hat{\mathbf{x}})) - I_i^k(\hat{\mathbf{x}})\right)^2 d\Omega_i + \int_{R_i^c} \left(b^k - I_i^k(\hat{\mathbf{x}})\right)^2 d\Omega_i, \tag{26}$$

where $\hat{\mathbf{x}}$ denotes image coordinates as before (for simplicity of discussion, distortion $D = 1$).

A. Initial expression of gradient

If we let $\hat{c}_i = \partial R_i$ denote the boundary of R_i then we may express the partial derivative of E with respect to one of the calibration parameters λ_{ij} as follows.

$$\begin{aligned} \frac{\partial E}{\partial \lambda_{ij}} &= \text{boundary term} + \text{foreground term} \\ &= \sum_{k=R,G,B} \int_{\hat{c}_i} \left(\left(f^k(\pi_i^{-1}(\hat{\mathbf{x}})) - I_i^k(\hat{\mathbf{x}})\right)^2 - \left(b^k - I_i^k(\hat{\mathbf{x}})\right)^2 \right) \langle \frac{\partial \hat{c}_i}{\partial \lambda_{ij}}, \hat{\mathbf{n}}_i \rangle \, d\hat{s} \\ &+ \sum_{k=R,G,B} 2 \int_{R_i} \left(f^k(\pi_i^{-1}(\hat{\mathbf{x}})) - I_i^k(\hat{\mathbf{x}})\right) \left\langle \nabla_S f^k\left(\pi_i^{-1}(\hat{\mathbf{x}})\right), \frac{\partial}{\partial \lambda_{ij}} \pi_i^{-1}(\hat{\mathbf{x}}) \right\rangle d\Omega_i \end{aligned} \tag{27}$$

In the boundary term, $d\hat{s}$ denotes the arclength measure of $\hat{c}_i$, and $\hat{\mathbf{n}}_i$ denotes its outward unit normal. In the foreground term, ∇_S denotes the natural gradient operator on the surface S.

B. Rewriting the boundary term

Ultimately, we will compute all quantities by integrating along the current estimate of the surface since that is the actual object represented by our data structures. Thus, it is more convenient to express the contour integral around $\hat{c}_i(\hat{s})$ in the image plane as a contour integral around $C_i(s)$ on the surface S

instead, (where $\pi_i(C_i) = \hat{c}_i$ and where s is the arclength parameter of C_i). They may be related as follows.

$$\begin{aligned}
\langle \frac{\partial \hat{c}_i}{\partial \lambda_{ij}}, \hat{\mathbf{n}}_i \rangle \, d\hat{s} &= \left\langle \frac{\partial}{\partial \lambda_{ij}} \pi_i(C_i), \frac{\partial}{\partial s} J \pi_i(C_i) \right\rangle ds, \qquad \text{where } J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
&= \left\langle \frac{1}{Z_i^2} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \end{bmatrix} \frac{\partial \mathbf{x}_i}{\partial \lambda_{ij}}, \frac{1}{Z_i^2} \begin{bmatrix} 0 & Z_i & -Y_i \\ -Z_i & 0 & X_i \end{bmatrix} \frac{\partial \mathbf{X}_i}{\partial s} \right\rangle ds \\
&= \frac{1}{Z_i^3} \left\langle \frac{\partial \mathbf{X}_i}{\partial s}, \begin{bmatrix} 0 & -Z_i & Y_i \\ Z_i & 0 & -X_i \\ -Y_i & X_i & 0 \end{bmatrix} \frac{\partial \mathbf{X}_i}{\partial \lambda_{ij}} \right\rangle ds \\
&= \frac{1}{Z_i^3} \left\langle \frac{\partial \mathbf{X}_i}{\partial s}, \frac{\partial \mathbf{X}_i}{\partial \lambda_{ij}} \times \mathbf{X}_i \right\rangle ds = \frac{1}{Z_i^3} \left\langle \frac{\partial \mathbf{X}_i}{\partial \lambda_{ij}}, \mathbf{X}_i \times \frac{\partial \mathbf{X}_i}{\partial s} \right\rangle ds \\
&= \frac{\|\mathbf{X}_i\|}{Z_i^3} \left\langle \frac{\partial \mathbf{X}_i}{\partial \lambda_{ij}}, \mathbf{N}_i \right\rangle ds \qquad \text{(since } \mathbf{X}_i \text{ and } \tfrac{\partial \mathbf{X}_i}{\partial s} \text{ are perpendicular tangent vectors to } S\text{)}
\end{aligned}$$

Thus, the boundary term written as an integral on the surface S (along the occluding contour C_i) has the following form:

$$\frac{\partial \lambda_{ij}}{\partial t} = \sum_{k=R,G,B} \int_{C_i} \left(\left(f^k - I_i^k\right)^2 - \left(b^k - I_i^k\right)^2 \right) \frac{\|\mathbf{X}_i\|}{Z_i^3} \left\langle \frac{\partial g_i}{\partial \lambda_{ij}}, \mathbf{N}_i \right\rangle ds, \tag{28}$$

which is also the update equations for the extrinsic parameter j for camera i with a piecewise constant assumption on the foreground and the background radiance.

C. Rewriting the foreground term

The first step in rewriting the foreground/background integrals is to re-express the derivative of the back-projected 3D point $\mathbf{X} = \pi_i^{-1}(\hat{\mathbf{x}}, \Lambda_i)$ with respect to the calibration parameter λ_{ij} in terms of the derivative of the forward projection $\pi_i(\mathbf{x}, \Lambda_i) = \pi(g_i(\mathbf{X}, \Lambda_i))$, since π_i has an analytic form while π_i^{-1} does not. We begin by fixing a 2D image point $\hat{\mathbf{x}}$ and note that

$$\hat{\mathbf{x}} = \pi_i\big(\mathbf{X}(\hat{\mathbf{x}}, \Lambda_i), \Lambda_i\big) \qquad \Big(\text{where } \mathbf{X}(\hat{\mathbf{x}}, \Lambda_i) = \pi_i^{-1}(\hat{\mathbf{x}}, \Lambda_i) = g_i^{-1}\big(\pi^{-1}(\hat{\mathbf{x}}), \Lambda_i\big)\Big)$$

and thus differentiation with respect to λ_{ij} yields:

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \lambda_{ij}} \pi_i\big(\mathbf{X}, \Lambda_i\big) = \frac{\partial \pi_i}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \lambda_{ij}} + \frac{\partial \pi_i}{\partial \lambda_{ij}} \\
&= \frac{1}{Z_i^2} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \end{bmatrix} \frac{\partial g_i}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \lambda_{ij}} + \frac{1}{Z_i^2} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \end{bmatrix} \frac{\partial g_i}{\partial \lambda_{ij}} \\
&\begin{bmatrix} Z_i & 0 & -Z_i \\ 0 & Z_i & -Y_i \end{bmatrix} \frac{\partial g_i}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \lambda_{ij}} = - \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \end{bmatrix} \frac{\partial g_i}{\partial \lambda_{ij}}
\end{aligned} \tag{29}
$$

Notice, though, that (29) does not uniquely specify $\partial \mathbf{X}/\partial \lambda_{ij}$ but merely gives a necessary condition. We must supplement (29) with the additional constraint that $\partial \mathbf{X}/\partial \lambda_{ij}$ must be orthogonal to the unit normal $\mathbf{N}$ of S at the point $\mathbf{X}$ in order to obtain a unique solution.

$$
\frac{\partial \mathbf{X}}{\partial \lambda_{ij}} \cdot \mathbf{N} = 0 \qquad \left(\text{or equivalently } \frac{\partial g_i}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \lambda_{ij}} \cdot \mathbf{N}_i = 0 \right) \tag{30}
$$

Now, combining equations (29) and (30), we have

$$
\begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \\ N_{ix} & N_{iy} & N_{iz} \end{bmatrix} \frac{\partial g_i}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \lambda_{ij}} = - \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \\ 0 & 0 & 0 \end{bmatrix} \frac{\partial g_i}{\partial \lambda_{ij}}
$$

$$
\frac{\partial \mathbf{X}}{\partial \lambda_{ij}} = - \left(\frac{\partial g_i}{\partial \mathbf{X}} \right)^{-1} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \\ N_{ix} & N_{iy} & N_{iz} \end{bmatrix}^{-1} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \\ 0 & 0 & 0 \end{bmatrix} \frac{\partial g_i}{\partial \lambda_{ij}}
$$

$$
\frac{\partial \mathbf{X}}{\partial \lambda_{ij}} = \frac{-\left(\frac{\partial g_i}{\partial \mathbf{X}}\right)^{-1}}{Z_i(\mathbf{X}_i \cdot \mathbf{N}_i)} \begin{bmatrix} Z_i N_{iz} + Y_i N_{iy} & -X_i N_{iy} & X_i z_i \\ -Y_i N_{ix} & Z_i N_{iz} + X_i N_{ix} & Y_i z_i \\ -Z_i N_{ix} & -Z_i N_{iy} & Z_i z_i \end{bmatrix} \begin{bmatrix} Z_i & 0 & -X_i \\ 0 & Z_i & -Y_i \\ 0 & 0 & 0 \end{bmatrix} \frac{\partial g_i}{\partial \lambda_{ij}}
$$

$$
\frac{\partial \mathbf{X}}{\partial \lambda_{ij}} = \frac{-\left(\frac{\partial g_i}{\partial \mathbf{X}}\right)^{-1}}{\mathbf{X}_i \cdot \mathbf{N}_i} \begin{bmatrix} \mathbf{X}_i \cdot N_i - X_i N_{ix} & -X_i N_{iy} & -X_i N_{iz} \\ -Y_i N_{ix} & \mathbf{X}_i \cdot N_i - Y_i N_{iy} & -Y_i N_{iz} \\ -Z_i N_{ix} & -Z_i N_{iy} & \mathbf{X}_i \cdot N_i - Z_i N_{iz} \end{bmatrix} \frac{\partial g_i}{\partial \lambda_{ij}}
$$

$$
\frac{\partial \mathbf{X}}{\partial \lambda_{ij}} = - \left(\frac{\partial g_i}{\partial \mathbf{X}} \right)^{-1} \left(\mathbf{I} - \frac{\mathbf{X}_i \otimes \mathbf{N}_i}{\mathbf{X}_i \cdot \mathbf{N}_i} \right) \frac{\partial g_i}{\partial \lambda_{ij}}, \tag{31}
$$

where $\otimes$ is the Kronecker product, and $\mathbf{I}$ is the 3×3 identity matrix.

The second step proceeds in the same manner as outlined earlier in rewriting the data fidelity terms in E_{data} by noting that the measure in the image domain $d\Omega_i$ and the area measure on the surface dA are related by $d\Omega_i = \sigma(\mathbf{x}_i, \mathbf{N}_i)\, dA$ where $\sigma(\mathbf{X}_i, \mathbf{N}_i) = (\mathbf{X}_i \cdot \mathbf{N}_i)/Z_i^3$. Then the foreground term in Eq.(27) is given by

$$\begin{aligned} & \sum_{k=R,G,B} 2 \int_{R_i} \left(f^k - I_i^k\right) \left\langle \nabla_S f^k\left(\pi_i^{-1}(\hat{\mathbf{x}})\right), \frac{\partial}{\partial \lambda_{ij}} \pi_i^{-1}(\hat{\mathbf{x}}) \right\rangle d\Omega_i \\ = & \sum_{k=R,G,B} 2 \int_{\pi_i^{-1}(R_i)} \left(f^k - I_i^k\right) \left\langle \nabla_S f^k(\mathbf{X}), \frac{\partial \mathbf{X}}{\partial \lambda_{ij}} \right\rangle \frac{\mathbf{X}_i \cdot \mathbf{N}_i}{Z_i^3}\, dA \end{aligned} \tag{32}$$

Therefore, the following foreground term will be added to the update equation of the extrinsic parameter in Eq.(28) :

$$\frac{\partial \lambda_{ij}}{\partial t} = \sum_{k=R,G,B} -2 \int_{\pi_i^{-1}(R_i)} \frac{\left(f^k - I_i^k\right)}{Z_i^3} \left\langle \nabla_S f^k(\mathbf{x}), \left(\frac{\partial g_i}{\partial \mathbf{X}}\right)^{-1} \left((\mathbf{X}_i \cdot \mathbf{N}_i) \frac{\partial g_i}{\partial \lambda_{ij}} - \left(\frac{\partial g_i}{\partial \lambda_{ij}} \cdot \mathbf{N}_i\right) \mathbf{X}_i \right) \right\rangle dA. \tag{33}$$

In Figure 13, several photos from a set of 32 images of a toy skater doll are shown. When the initial extrinsic parameters are off as observed in the projections of the foreground object onto the images (shown by an orange mask), a visual hull created using the uncorrected extrinsic camera parameters is significantly away from the real doll surface. After the extrinsic calibration equations (28) plus (33) are evolved to convergence, visual hull created using the updated extrinsic parameters demonstrates the correction and true refinement provided by the derived equations. In Figure 14, we depict the extrinsic refinement stability by showing the uncertainty ellipsoids drawn around each camera center. Parameters were perturbed in x, y, z directions randomly several times, and converged properly for variations up to 8%.

VII. Results and Conclusions

The toy skater example shown in Figure 15 demonstrates the simultaneous evolution of the extrinsic and intrinsic parameters for the 32 cameras, along with the projections of the foreground surface. The visual hulls created with again the initial set of camera parameters and the evolved set of camera parameters display a correct refinement of the camera parameters.

For most of the experiments we utilized a 128^3 volume, and a $140 \times 150 \times 360$ volume for the Bust dataset. With a volumetric signed distance representation in our C++ implementation without any code optimization on a Pentium 2.40 GHz processor, each single iteration to compute all calibration gradients takes on the order of 10 seconds depending on the number of camera views as well, and convergence takes about 50-400 iterations depending on the initialization, hence a computation time of about 8-60 minutes. However, a mesh representation on the object may be easier to work with since the parameter update equations we derived are ordinary differential equations.

A common issue for any calibration procedure is that when there are shape symmetries or constant radiance on the object, camera pose parameter estimation is not stable, however, these do not affect the 3D reconstruction (e.g. multiple views on a sphere do not allow estimating camera pose, but they still allow estimating the shape of the sphere). Regarding the radiance assumptions, because our algorithm integrates information globally on the entire collection of images, it is far less sensitive to this accident than algorithms based on local statistics, such as point feature correspondences. Therefore, symmetries are not an obstacle since our goal is not to obtain the absolute calibration parameters (ground truth) but to help refine 3D reconstruction. From this perspective, the only criterion of concern is the re-projection error. We experimented with a full turn head sequence using Intel's Vang Gogh Bust data for testing the issue of shape complexity. We utilized only 16 camera views from the available 330 camera images for ease and speed of computations. We computed re-projection errors: a Type II error (error of omission) and Type I error (error of commission) by counts of voxels for several camera views used during our experiments both after perturbation of the camera parameters and after evolution of the parameters as shown in Table 1. After refinement stage, the Type I error dropped by 95%, and Type II error dropped by 40%. As remarked above, our goal is not to obtain absolute camera parameters but to help 3D reconstruction algorithm to obtain objects correctly, which is achieved.

The Bust data comprises of numerous views, and this facilitated the following experiment to show the practicality of our calibration correction. For the three camera views, out of the 16 views, we deliberately

used wrong camera calibration parameters, which belong to that of the neighbor views in the sequence in Figure 16. This represents a possible perturbation in a real life scenario, i.e. the cameras are accidentally moved a little bit after the calibration and the views that are captured afterwards are a little bit off. The 3D reconstruction of the Bust object on the top right shows the erroneous surfaces obtained in this case. With our coordinated refinement of the extrinsic parameters using Eq.(28) and (33), the improvements in the reprojection errors and the 3D reconstruction are observed in Figure 16.

A real color calibration experiment is carried out using HP Labs stereo rig system. We captured images, shown in Fig.17, of the color calibration object from five cameras. Notice that the first picture is somewhat darker than the others, second and third pictures appear lighter, and there is a color mismatch. A cube surface is rigidly registered with the scene, also the radiance function on the cube is estimated as shown in bottom row of Fig.17. The second row shows views after the evolution of color calibration coefficients are completed. The third row shows the projections of the model surface onto the views. It can be visually assessed that color responses of the cameras have achieved a balancing effect, and helped to obtain a better texture mapping as well.

Next we demonstrate a calibration experiment using pictures from a handheld camera with no camera calibration information available. In this scenario, the variational calibration techniques we presented require some rough initial values that we obtained through a self calibration software currently under development. We have a 13 set of pictures taken around the Statue of Liberty, covering about 220/360 degrees of a circle around the statue, a few of the views shown in Figure 18. [2] We obtained initial camera parameters: extrinsics and intrinsics including the skew parameter. A rough calibration results in the projections shown in Figure 18. After evolution of the camera parameters: extrinsics, intrinsics including the skew parameter, and color parameters, the comparison is done with the visual hulls of before and after evolution camera parameters in Figure 19. One can observe the correction in the Statue of Liberty surface with a better set of camera parameters obtained with the derived update equations throughout the

[2]We thank our colleague Irwin Sobel at HP Labs for providing these pictures.

paper. We also show blow-up regions in Figure 20 from some of the camera views before and after the evolution of the color camera parameters, and the colors are modified towards achieving some relative agreement among the cameras which can however only be subjectively judged.

A. Discussions

One may argue that the requirement of some rough initial extrinsic and intrinsic camera parameters limits the usability of this technique. However, the refinement or correction of camera parameters from a perturbed state of a previous calibration is a real world problem that constantly presents obstacles to the usage of multiple camera systems. After a very good initial calibration, the cameras over time may see small changes in their parameters. For instance, extrinsic parameters will often be changed particularly due to unwanted accidental motion. Similarly, the intrinsics and color parameters of the cameras may go through small variations due to ambient conditions and wear-off. Therefore, the presented camera calibration framework proves to be a useful tool for multi-camera systems.

B. Conclusions

In this paper, we employed the 3D stereo techniques based on variational ideas to various camera calibration refinement problems. We have presented new multi-view stereo techniques to:

- evolve pose parameters of a 3D model object to take advantage of the known shape of calibration object, and to reduce computational complexity,
- evolve distortion parameters of cameras given a 3D model shape,
- evolve color calibration parameters of cameras given a 3D model shape,
- evolve intrinsic parameters of cameras,
- evolve extrinsic parameters of cameras.

Pros and cons of this technique are discussed as follows:

- A nice feature of the methodology presented in this paper is that it can integrate several small and different problems such as distortion calibration, color calibration into an overall unified system based on the joint segmentation framework, and simultaneously evolve pose, color, distortion, extrinsic, and other parameters as well.

- We make piecewise smooth object assumption and a constant background assumption, which may be a limitation if the background is to be modeled as well. However, a background model may be added to this framework if needed.
- The presented methods eliminate the need for search of image edges, point correspondences from images, which can be very sensitive to pixel-level noise whereas our approach being based on image regions for comparisons, is not as sensitive to noise.
- Another advantage of our framework is that it easily accommodates additional data. In the more classical approaches to stereo, bringing in more data, or adding more images to the algorithm might not help all the time, that is if something goes wrong in the independent segmentation phase of even one image, it destroys the whole process of reconstructions and geometry. On the other hand, adding more data to this joint segmentation framework will only improve robustness, providing more tolerance towards errors.
- For the distortion calibration method, more improvements may be obtained with utilizing more poses, hence many more camera images of the calibration object, and more than one distortion coefficient in the model selected. One can also utilize more general/complicated distortion models than the simple polynomial D function.
- Currently, we have an implicit representation of the calibration objects, i.e. the cube or the rectangular bar. Computing surface normals, visibility functions for the surface occluding boundary from this implicit representation is not perfectly exact, and the quantities are slightly smeared. A future direction towards more efficient algorithms, is to use an explicit representation of the calibration object to more accurately describe the occluding boundaries. With this approach, 3D grids are not needed for the data structures, resulting in increased accuracy, speed and decreased memory requirements.
- Camera calibration is particularly suited to our framework, since it does not have to be done in real-time, and also the environmental conditions may be allowed to vary to a degree (e.g. our choice of a constant colored foreground object before a dark background).

ACKNOWLEDGEMENTS

We acknowledge HP Labs, Palo Alto, CA, for their support to G. Unal and A. Yezzi through grants to Georgia Tech. for funding of this work. We thank our colleagues at HP Labs: Bruce Culbertson, Harlyn Baker, Irwin Sobel, Tom Malzbender, and Donald Tanguay for fruitful discussions and their support. We also thank Hailin Jin for providing us the Intel's Bust dataset.

Fig. 3. Initialized surface model shown from three different vantage points.

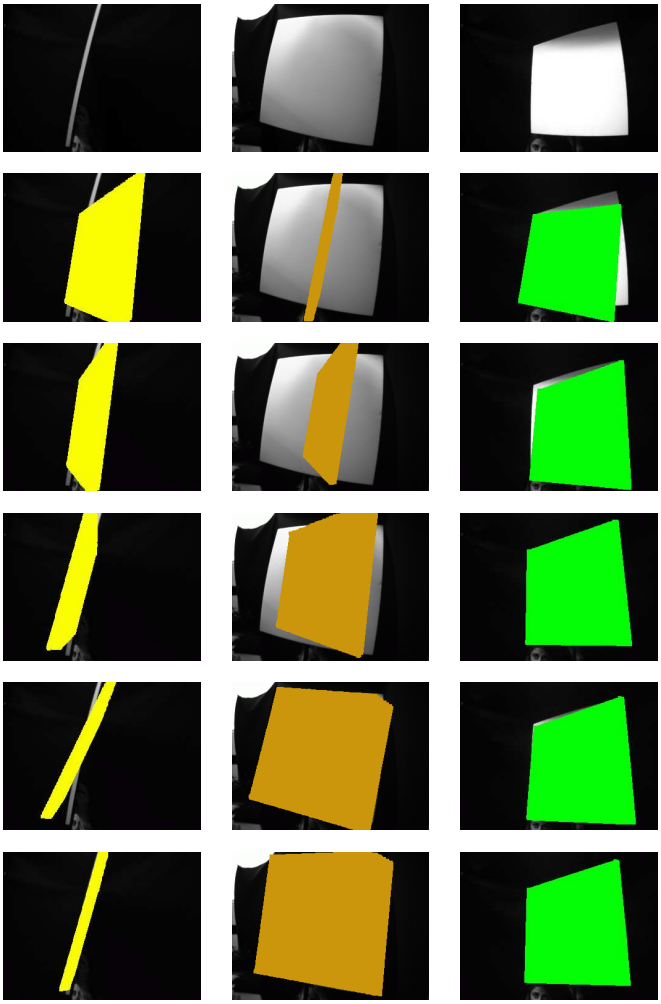

Fig. 4. Column 1: Pose1. Row 1: one camera image shown, Row2: with projection of initialized surface (orange mask), Rows 3-5: during evolution of the pose parameters of the surface, Row 6: with converged pose parameters. Columns 2-3: same as column 1 for poses 2 and 3, respectively.

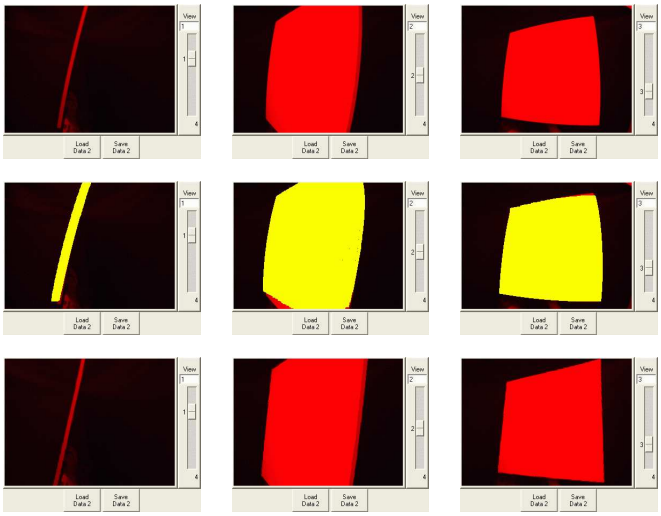

Fig. 5. Pose 1. Row 1: Three out of five captured views. Row 2: Projected surface after distortion parameters have converged. Row 3: Undistorted with the obtained distortion coefficients.

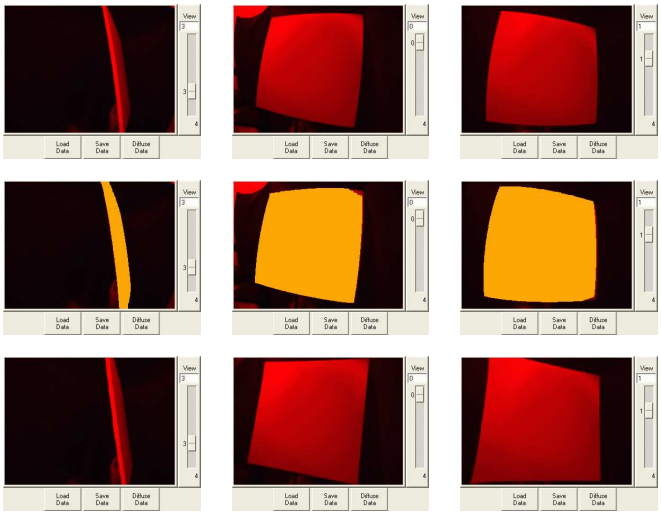

Fig. 6. Pose 2. Row 1: Three out of five captured views. Row 2: Projected surface after distortion parameters have converged. Row 3: Undistorted with the obtained distortion coefficients.

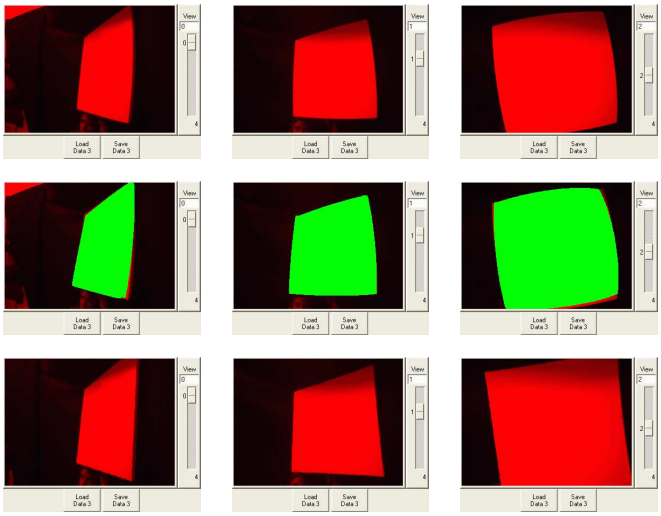

Fig. 7. Pose 3. Row 1: Three out of five captured views. Row 2: Projected surface after distortion parameters have converged. Row 3: Undistorted with the obtained distortion coefficients.

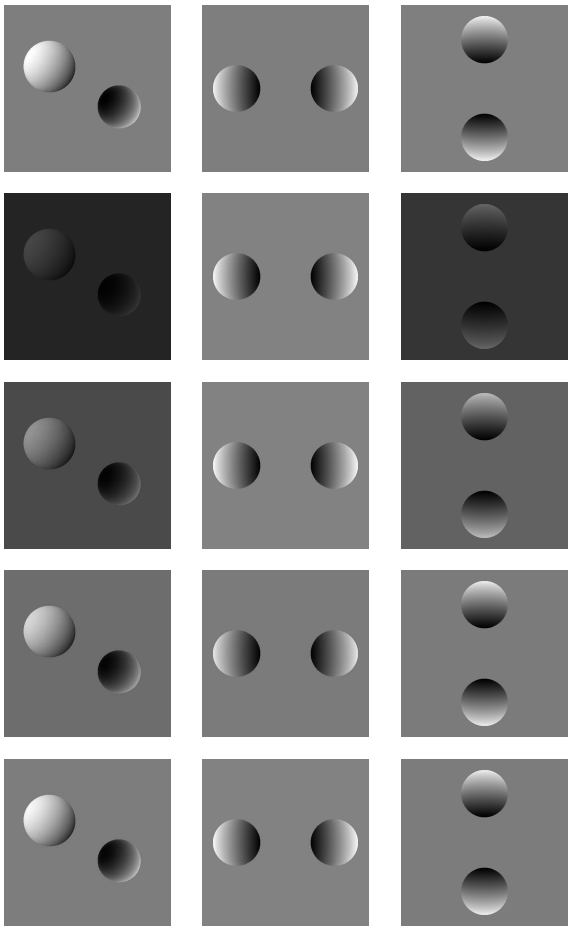

Fig. 8. Row 1: Three original views (cameras 1-7-9). Row 2: The same three different after deliberate simulated miscalibration of the greyscales. The same three views while evolving the calibration parameters: Rows 3-4 intermediate stages, Row 5: The views after evolution of the calibration parameters has completed.

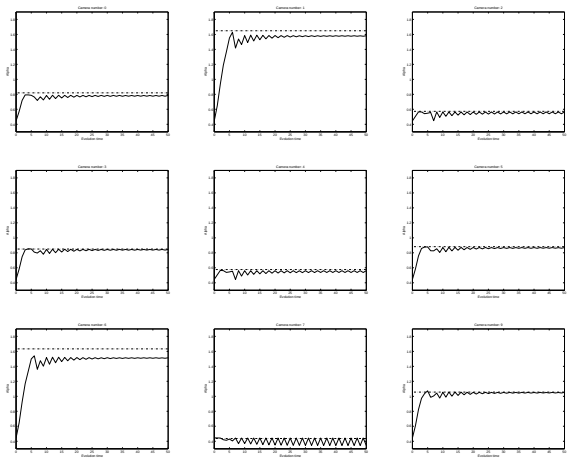

Fig. 9. Evolution of the parameter α for different camera views. True α value is shown as a dotted line.

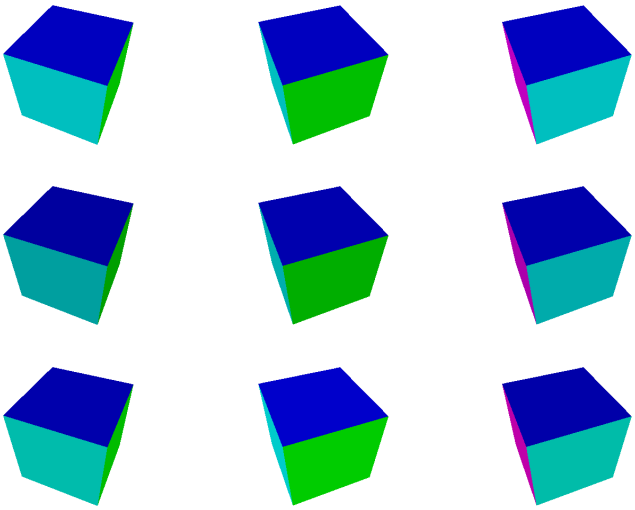

Fig. 10. Some camera views shown during the evolution of the color calibration. Top: original views, Middle: Perturbed views, Bottom: Final views after convergence. Note the color similarity in top and bottom rows.

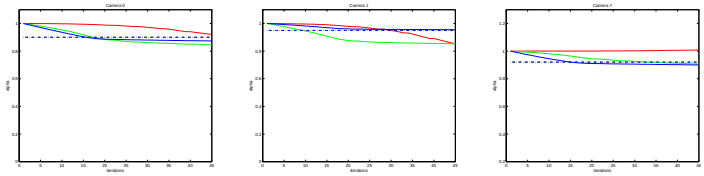

Fig. 11. Evolution of the parameter α for different views for R,G,B channels of the synthetic color cube. True α value is shown as a dotted line.

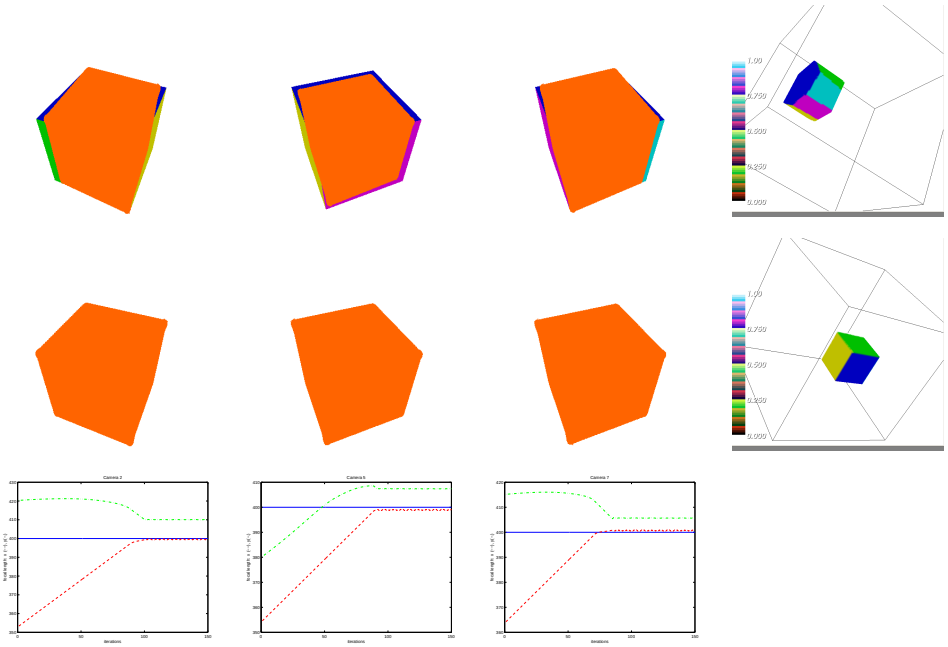

Fig. 12. Top: Three camera views shown during the evolution of the intrinsic parameters of an initial cube with projections from the initial surface, Middle: Final views after convergence of the intrinsic parameters of the surface. Also shown at the bottom are the evolution of the two focal length parameters for each shown camera view (red and green curves) along with the true (blue curve) focal lengths.

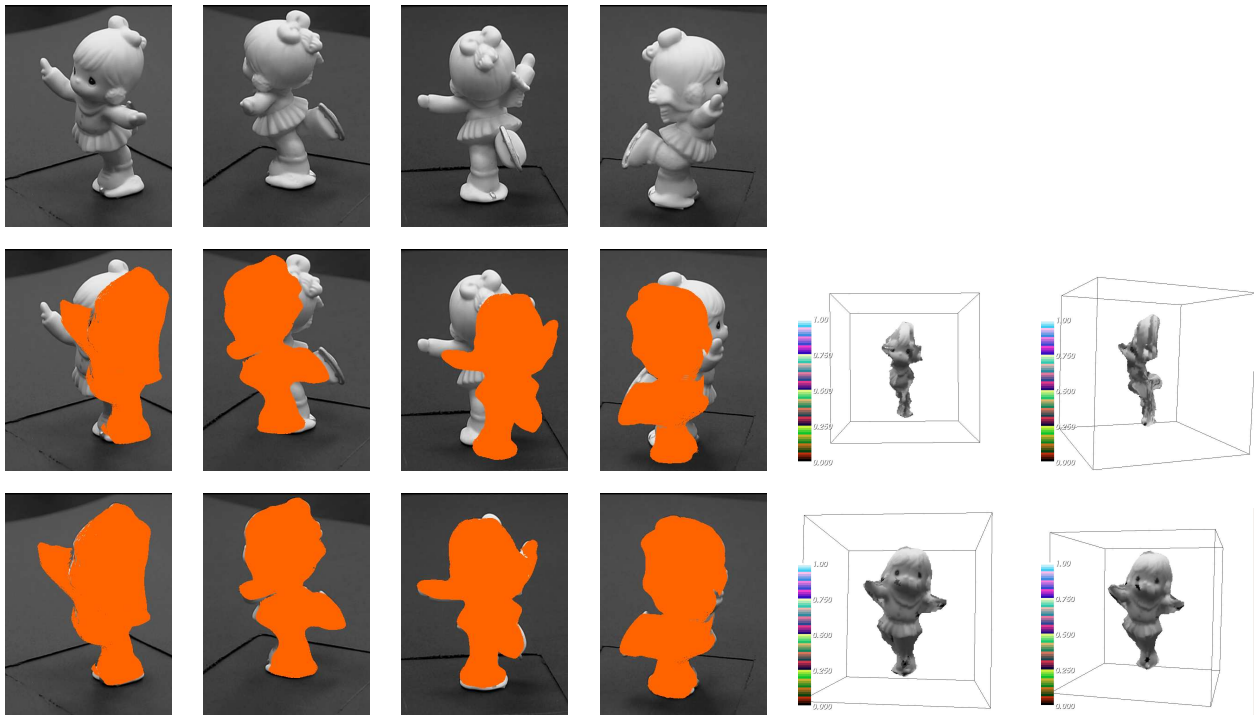

Fig. 13. Four camera views shown (top) during the evolution of the extrinsic parameters of an initial surface of a toy skater, Row 2: Views shown with projections from the initial surface, Row 3: Final views after convergence of the extrinsic camera parameters. Visual hull generated using the miscalibrated initial extrinsic parameters (row 2 right); visual hull generated using the converged extrinsic parameters (row 3 right).

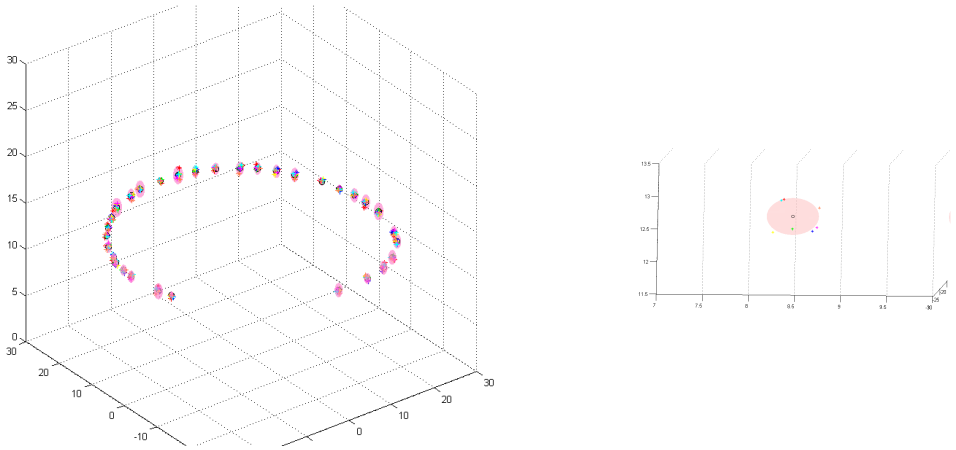

Fig. 14. Uncertainty ellipsoids drawn around each camera center for the toy skater data show the extrinsic refinement stability (right: zoomed into one camera's perturbations).

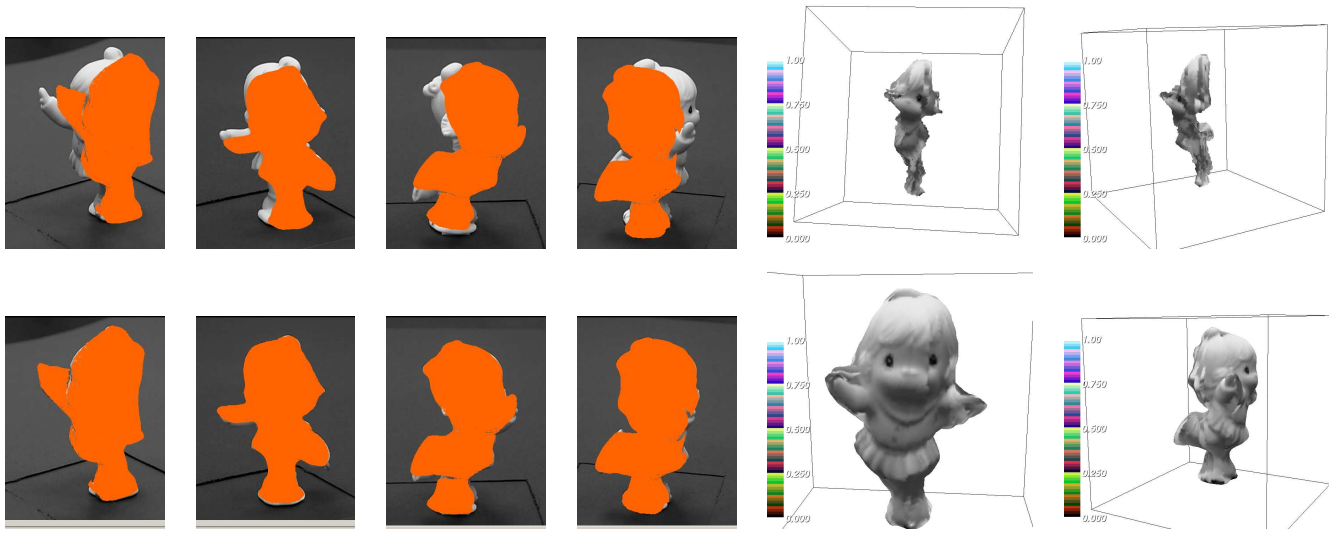

Fig. 15. Row 1: Four camera views during the evolution of the extrinsic plus intrinsic parameters of a toy skater with projections of the initial surface, Row 2: Final views after convergence of the camera parameters. Visual hull generated using the miscalibrated initial parameters (row 1 right); visual hull generated using the converged parameters (row 2 right).

	Type II error		Type I error	
Camera	Initial	Final	Initial	Final
Cam 1	3584	47	5593	4476
Cam 2	13831	77	17762	4807
Cam 55	2841	191	5574	4618
Cam 77	8894	46	14013	14700
Cam 105	8339	1344	10032	6724
Cam 166	1005	170	4021	4001
Cam 200	4901	467	7971	6414
Cam 207	7339	97	9764	6783
Cam 239	9251	213	10615	6492
Cam 244	12467	263	16956	3110
Cam 321	1733	65	6501	7365

Table 1. Type I and Type II errors in counts of voxels for several camera views for Bust data (Fig. 16) after perturbation of camera parameters (Initial), and after evolution of parameters (Final).

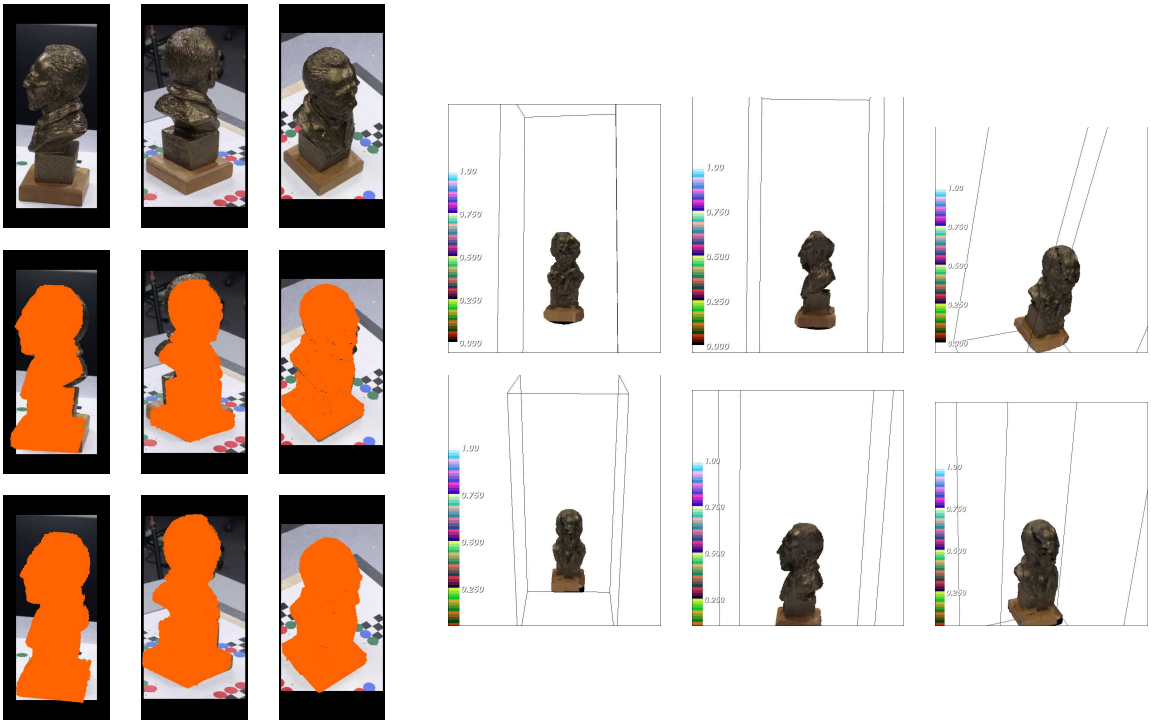

Fig. 16. Camera views 78,167, and 240 in top row are used deliberately with camera calibration parameters of camera views 77, 166, and 239 of the Van Gogh Bust dataset. Top: Three camera views shown with projections from the initial surface in row 2, here note the resulting initial mismatch in projected silhouettes. Row 3: Final views after convergence of the camera parameters. Visual hull surfaces obtained by using wrong calibration parameters for views 78, 167, 240 on the right (top row) and surfaces with corrected calibration parameters in bottom row.

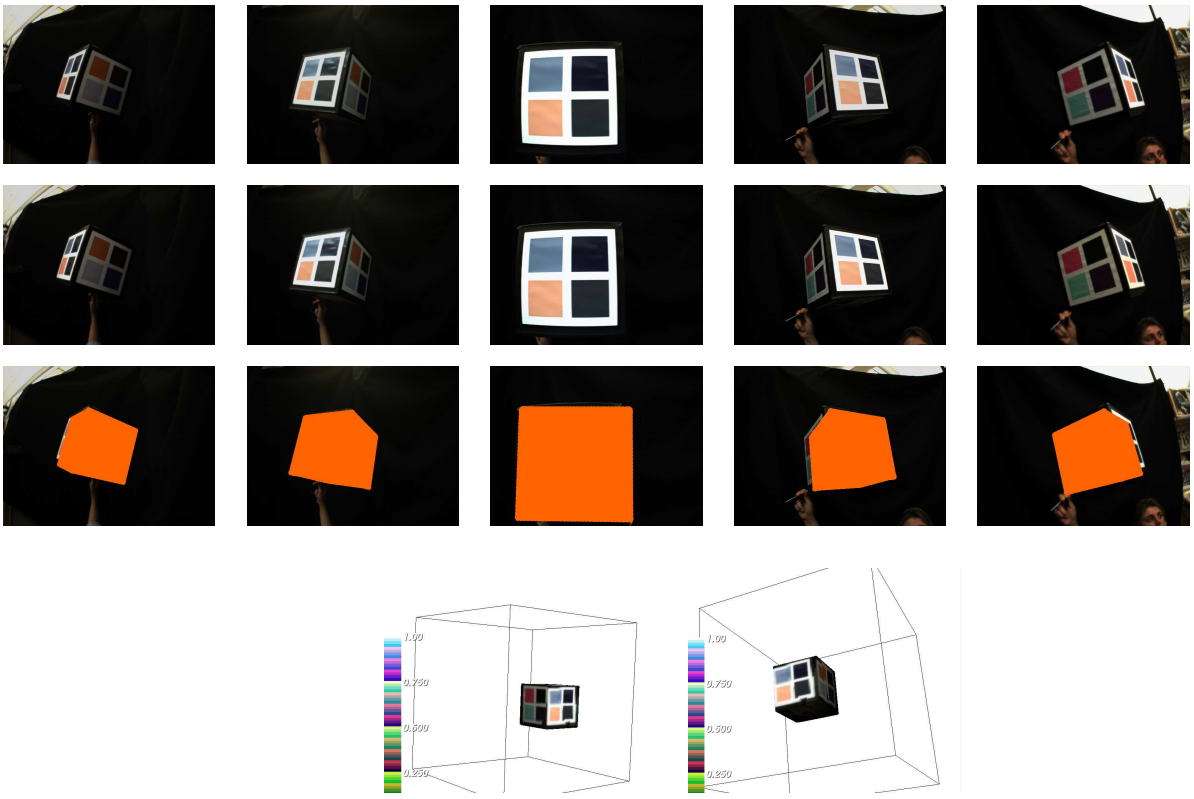

Fig. 17. Some camera views shown during the evolution of the color calibration parameters of the HP color calibration object surface. Top: Five camera views; Row 2: Final views after convergence of the extrinsic camera parameters; Row 3: Same shown with projections of the converged cube; Bottom: Color calibration cube with reconstructed radiance on the surface from two different vantage points.

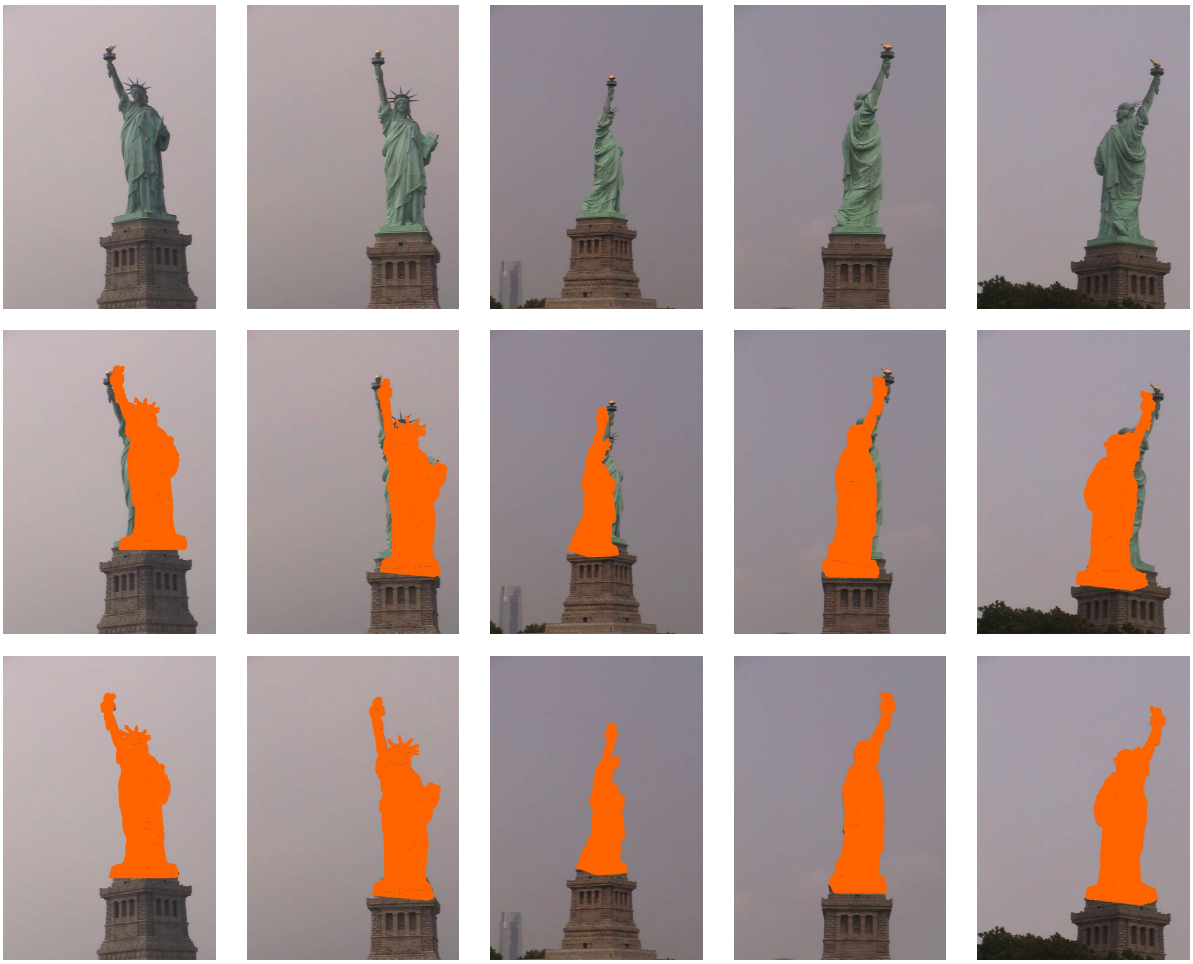

Fig. 18. Some camera views shown during the evolution of the camera calibration parameters of the Statue of Liberty surface. Top: Five camera views shown with projections from the initial surface in Row 2; Row 3: Final views after convergence of the camera parameters.

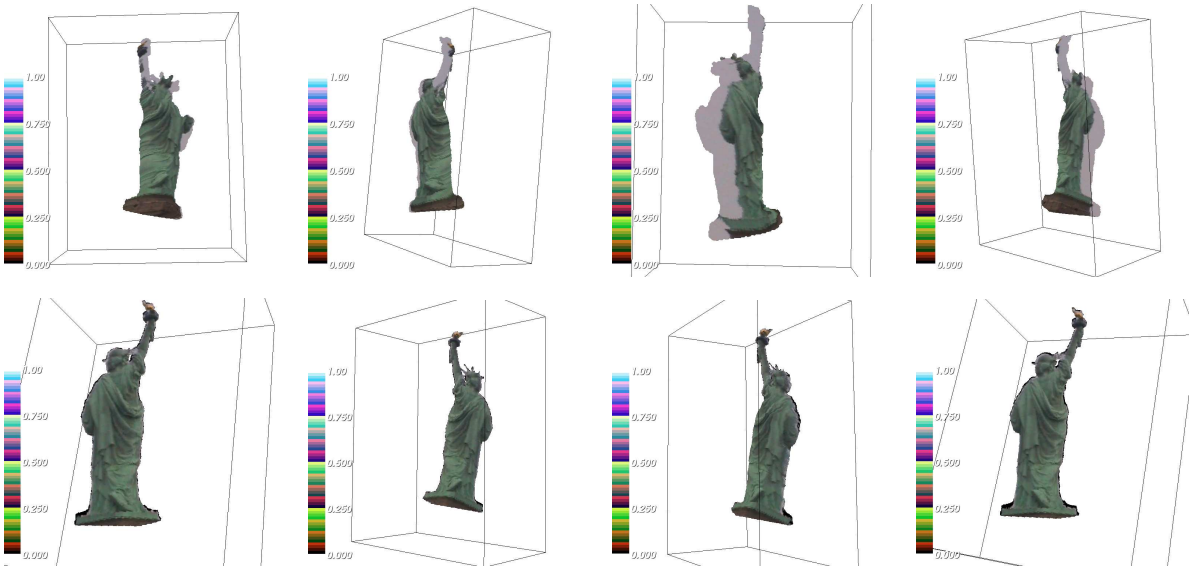

Fig. 19. Visual hull surfaces with initial rough calibration parameters(top), and with refined calibration parameters (bottom), also with radiance texture mapped onto the surfaces.

Fig. 20. Some camera views before and after the color calibration for the statue of liberty.

REFERENCES

[1] O. Faugeras and R. Keriven, "Variational principles, surface evolution pdes, level set methods and the stereo problem," Tech. Rep., INRIA, 1996.

[2] A. Yezzi and S. Soatto, "Stereoscopic segmentation," *Int. J. Computer Vision*, vol. 53, pp. 31–43, 2003.

[3] A. Yezzi and S. Soatto, "Structure from motion for scenes without features," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, pp. 525–532.

[4] I. Sobel, "On calibrating computer controlled cameras for perceiving 3d scenes," *AI*, vol. 5, pp. 185–198, 1974.

[5] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *J. Robotics and Automation*, vol. 3, pp. 323–44, 1987.

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

[7] B. Triggs, "Camera pose and calibration from 4 or 5 known 3d points," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1999, pp. 278–284.

[8] P. Gurdjos and P. Sturm, "Methods and geometry for plane-based self-calibration," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, pp. 491–496.

[9] O.D. Faugeras, Q.T. Luong, and S.J. Maybank, "Camera self-calibration: Theory and experiments," in *Proc. European Conf. Computer Vision*, 1992, pp. 321–334.

[10] Y. Seo, A. Heyden, and R. Cipolla, "A linear iterative method for auto-calibration using the DAC equation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001, pp. 880–885.

[11] J. Oliensis, "Fast and accurate self calibration," in *Proc. Int. Conf. on Computer Vision*, 1999, pp. 745–752.

[12] M. Pollefeys, R. Koch, and L. Van Gool, "Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters," *Int. J. Computer Vision*, vol. 32, no. 1, pp. 7–25, 1999.

[13] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multi-camera self-calibration for virtual environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, no. 4, 2005.

[14] J. Heikkila and O.Silven, "A four-step camera calibration procedure with implicit image correction," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1997, pp. 1106–1112.

[15] G. Unal and A. Yezzi, "A variational approach to problems in calibration of multiple cameras," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004, pp. 172–178.

[16] G. Unal, A. Yezzi, H. Baker, and B. Culbertson, "A variational approach to calibration of multiple cameras," Tech. Rep., HP Labs, HPL-2004-219, Palo Alto, CA, 2004.

[17] F. Devernay and O. Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, vol. 13, pp. 14–24, 2001.

[18] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE PAMI*, vol. 14, pp. 965–980, 1992.

[19] Z. Zhang, "A flexible new technique for camera calibration," Tech. Rep., Microsoft Research, MSR-TR-98-71, 1998.

[20] D.C. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering*, vol. 32, 1966.

[21] M. T. El-Melegy and Aly A. Farag, "Nonmetric lens distortion calibration: Closed-form solutions, robust estimation and model selection," in *Proc. Int. Conf. on Computer Vision*, 2003, pp. 554–559.

[22] S.B. Kang, "Semiautomatic methods for recovering radial distortion parameters from a single image," Tech. Rep., CRL Tech. Report CRL 97/3, 1997.

[23] R. Swaminathan and S.K. Nayar, "Nonmetric calibration of wide-angle lenses and polycameras," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1999, pp. 2413–2418.

[24] G.P. Stein, "Lens distortion calibration using point correspondences," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1997, pp. 602–608.

[25] A.W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *CVPR*, 2001.

[26] F. Du and M. Brady, "Self-calibration of the intrinsic parameters of cameras for active vision systems," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1993, pp. 477–482.

[27] H.S. Sawhney and R. Kumar, "True multi-image alignment and its application to mosaicing and lens distortion correction," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1997, pp. 450–456.

[28] H.H. Baker, D. Tanguay, I. Sobel, D. Gelb, M.E. Goss, B.W. Culbertson, and T. Malzbender, "The coliseum immersive teleconferencing system," Tech. Rep., HP Labs, HPL-2002-351, 2002.

[29] E. Marszalec, "On-line color camera calibration," in *Proc. Int. Conf. on Pattern Recognition*, 1994, pp. 232–237.

[30] S. Osher and J.A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton-Jacobi formulation," *J. Computational Physics*, vol. 49, pp. 12–49, 1988.

[31] D. Adalsteinsson and J.A. Sethian, "A fast level set method for propagating interfaces," *J. Computational Physics*, vol. 118, pp. 269–277, 1995.

[32] J. Sokolowski and J.P. Zolesio, *Introduction to Shape Optimization, Shape Sensitivity Analysis*, Springer Verlag, Berlin, Heilderberg, 1992.

[33] R.M. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.