# A Variational Approach to Calibration of Multiple Cameras

Gozde Unal[1], Anthony Yezzi[1], Harlyn Baker, Bruce Culbertson
Mobile and Media Systems Laboratory
HP Laboratories Palo Alto
HPL-2004-219
December 7, 2004*

camera calibration, variational methods, function minimization

This report addresses the problem of calibrating camera parameters such as lens distortion and color scaling using variational methods. We have presented multi-view stereo techniques based on variational ideas to (i) evolve pose parameters of a 3D model object to take advantage of the known shape of calibration objects, and to reduce computational complexity, (ii) evolve distortion parameters of cameras given a 3D model shape, (iii) evolve color calibration parameters of cameras given a 3D model shape.

* Internal Accession Date Only
[1]Georgia Institute of Technology, Atlanta, GA 30332-0250

Approved for External Publication

# A Variational Approach to Calibration of Multiple Cameras

[‡]Gozde Unal, [‡]Anthony Yezzi, [†]Harlyn Baker, [†]Bruce Culbertson

August 2003

[†]Hewlett-Packard Labs, Palo Alto, CA, 94301

[‡]Georgia Institute of Technology, Atlanta, GA, 30332-0250

**Abstract**

This report addresses the problem of calibrating camera parameters such as lens distortion and color scaling using variational methods. We have presented multi-view stereo techniques based on variational ideas to (i) evolve pose parameters of a 3D model object to take advantage of the known shape of calibration objects, and to reduce computational complexity, (ii) evolve distortion parameters of cameras given a 3D model shape, (iii) evolve color calibration parameters of cameras given a 3D model shape.

## 1 Introduction

The problem of recovering 3D representation of scenes 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 using epipolar geometry techniques 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] proposed a joint region-based image segmentation and simultaneous 3D stereo reconstruction technique. This report addresses camera calibration techniques built on this later stereo reconstruction framework which avoids searches for local correspondences and is versatile enough to accomodate the new applications to be shown.

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 repre-

sentations of 3D scenes, accurate camera calibrations are essential. For geometrical measurements for instance, 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.

The contributions presented in this report are 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 2. 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 3 and a novel technique for relative inter-camera color calibration in Section 4. Conclusions and future work directions are given in Section 5.

## 2 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 $f$ (radiance of background $g$), and the actual measurements from multiple camera views. Let $g_i$ denote the transformation from world coordinates to camera coordinates: $g_i : \vec{X} \longrightarrow \vec{X}_i$, and $\pi$ denote the perspective transformation from camera frame to the image plane: $\pi : \vec{X}_i = \longrightarrow \vec{x}_i = (x_i = \frac{X_i}{Z_i}, y_i = \frac{Y_i}{Z_i})^T$.

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 $I_i$:

$$E = \sum_{i=1}^{n} \int_{R_i} [f((\pi \circ g_i)^{-1}(\vec{x}_i)) - I_i(\vec{x}_i)]^2 d\vec{x}_i + \sum_{i=1}^{n} \int_{R_i^c} [g - I_i]^2 d\vec{x}_i \tag{1}$$

This energy can be lifted back onto surface $S$ :

$$E(S) = \sum_{i=1}^{n} \int_{S} [(f(\vec{X}) - I_i(\pi \circ g_i(\vec{X})))^2 - (g - I_i)^2] \, \mathcal{X}_i(\vec{X}) \sigma(\vec{X}_i) dA, \tag{2}$$

where $\sigma$ is the Jacobian of the change of coordinates from the image plane to the surface, and $\mathcal{X}_i$ is the visibility function of a voxel on the surface. 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) which is the gradient descent flow of the energy E. A popular class of numerical techniques known as Level Sets Methods [3], 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 [4], 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, we will derive the equations to update the parameters of the surface motion, which is more general than a similarity but less general than a fully affine (which we refer to as semi-affine) transformation, using the energy $E$ in Eq. (2).

Let the original rigid surface be denoted by $S_o$, then $S = g(S_o)$, or $\vec{X} = g(\vec{X}_o) = R_s\vec{X}_o + T_s$, and let $\lambda$ denote parameters of the rigid motion $g$. Then the gradient of the energy $E$ w.r.t. $\lambda$ is given by:

$$\begin{aligned}\frac{\partial E(\lambda)}{\partial \lambda} &= \int_S \sum_i F_i(\vec{X}) < \frac{\partial \vec{X}}{\partial \lambda}, \vec{N} > dA, \\ &= \int_{S_o} \sum_i F_i(g(\vec{X}_o)) < \frac{\partial (g\vec{X}_o)}{\partial \lambda}, R_s\vec{N}_o > dA_o\end{aligned}$$

where $F_i$ includes data and visibility related terms. Note that the visibility function $\mathcal{X}_i(g(\vec{X}_o))$, included in the data term $F_i(\cdot)$ is computed using the original visibility function but compensated for by the $R_s^T(\vec{C}_i - T_s)$ ), where $\vec{C}_i$ is a camera center.

For translation parameters:

$$< \frac{\partial (g\vec{X}_o)}{\partial \lambda}, R_s\vec{N}_o >= R_s\vec{N}_o.$$

For rotation parameters:

$$\begin{aligned}< \frac{\partial (g\vec{X}_o)}{\partial \lambda}, R_s\vec{N}_o > &= < R_s \begin{bmatrix} 0 & Z_o & -Y_o \\ -Z_o & 0 & X_o \\ Y_o & -Xo & 0 \end{bmatrix}, Rs\vec{N}_o > \\ &= < -R_s\hat{\vec{X}}_o, R_s\vec{N}_o >= - < R_s^T R_s\hat{\vec{X}}_o, \vec{N}_o >= -\vec{N}_o^T \cdot \hat{\vec{X}}_o, \end{aligned} \quad (3)$$

where we utilize exponential coordinates (see [5] for details on this representation) for the global rotation parameters of the surface.

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

$$\int_{S_o} \sum_i F_i(g(\vec{X}_o)) < \frac{\partial (g\vec{X}_o)}{\partial \lambda}, R_s\vec{N}_o > dA_o$$

where

$$< \frac{\partial(g\vec{X}_o)}{\partial\lambda}, R_s\vec{N}_o >=< R_s\frac{\partial S}{\partial\lambda}\vec{X}_o, R_s\vec{N}_o >= \vec{X}_o \cdot \vec{N}_o. \tag{4}$$

The evolution equations for the motion parameters $\lambda_j$ are given by:

$$\frac{\partial\lambda_j}{\partial t} = -\frac{\partial E}{\partial\lambda_j} = \int_{S_o}\sum_i F_i(g(\vec{X}_o)) \ < \vec{N}_o^T \cdot \hat{\vec{X}}_o > \ dA_o, \quad \text{(for rotation)}, \tag{5}$$

$$\frac{\partial\lambda_j}{\partial t} = -\frac{\partial E}{\partial\lambda_j} = -\int_{S_o}\sum_i F_i(g(\vec{X}_o)) \ < R_s\vec{N}_o > \ dA_o, \quad \text{(for translation)}. \tag{6}$$

The evolution equations for the scaling parameters $\lambda_j$ are given by:

$$\frac{\partial\lambda_j}{\partial t} = -\frac{\partial E}{\partial\lambda_j} = -\int_{S_o}\sum_i F_i(g(\vec{X}_o)) \ (\vec{X}_o \cdot \vec{N}_o) \ dA_o, \quad \text{(for scaling)}. \tag{7}$$

Note that we can generalize this idea in a straightforward fashion by considering $S$ to be more general than a simple diagonal matrix in order to accomodate a fully affine motion of the surface.

We will use equations (5-6-7) in updating the pose of the surface $S$ to estimate its correct placement in the 3D space for the calibration applications presented in Sections 3 and 4.

## 3 Lens Distortion Calibration

The linear model in an ideal pinhole camera model [6] leads to imaging of world lines as lines on the image plane, and simplifies many computations and considerations. However, for most real cameras with low-cost or wide-angle lenses this assumption does not hold, and nonlinearities introduced by a well-known phenemonen 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 external camera parameters (relative pose of the cameras with respect to the origin of world coordinate frame), and internal parameters (focal length, image aspect ratio, principal point, ...) are first obtained by a pinhole camera model neglecting lens distortion. Then distortion calibration is performed while holding the other parameters fixed [7–9]. 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. The distortion mapping is usually 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 [7]. 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 in the context of many computer vision applications. 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 is often used as the center for radial distortion [6], which we will also adopt.

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. Other techniques such as [10] rely on point correspondences. Given a set of 3D points, the associated epipolar and trilinear (among three cameras) 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.

Next we present a new distortion calibration technique in which we do not rely on extraction of edges and search for point correspondences. 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. The details are presented next.

### 3.1 Calibration of the Lens Distortion Parameters

Our notation is as follows:

(World) $\vec{X} \underbrace{\longrightarrow}_{g_i} \vec{X}_i \underbrace{\longrightarrow}_{\pi} \vec{x}_i = \begin{pmatrix} \frac{X_i}{Z_i} = x_i \\ \frac{Y_i}{Z_i} = y_i \\ 1 \end{pmatrix} \underbrace{\longrightarrow}_{D} \vec{\hat{x}}_i \underbrace{\longrightarrow}_{\begin{pmatrix} L_u & 0 & u_0 \\ 0 & L_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}} (u, v)$ (image coordinates),

where $D$ is the distort function:

$$\vec{\hat{x}}_i = D\vec{x}_i = (1 + k_1^i r^2 + k_2^i r^4 + k_3^i r^6 + ...)\vec{x}_i, \tag{8}$$

$r^2 = (x_i^2 + y_i^2)$, and $k_j^i$ is the $j^{th}$ distortion coefficient for camera $i$.

The gradient of the energy E (1), defined over the distorted image plane instead of the ideal plane, w.r.t. distortion parameters $k_j^i$ is given by

$$\frac{\partial k_j^i}{\partial t} = -\frac{\partial E}{\partial k_j^i} = -\int_{\hat{C}_i} F_i((D \circ \pi \circ g_i)^{-1} \vec{\hat{x}}_i) < \frac{\partial \vec{\hat{x}}_i}{\partial k_j^i}, \vec{\hat{n}}_i > d\hat{s} \tag{9}$$

where $\hat{s}$ is the arclength of the contour $\hat{C}_i$ on the image plane (the distorted or actual observed image coordinates), and note that we assumed a piecewise constant radiance model on the surface and the background to simplify the computations.

We can compute $\frac{\partial \vec{\hat{x}_i}}{\partial k_j^i}$ as follows:

$$\frac{\partial \vec{\hat{x}_i}}{\partial k_1^i} = r^2 \vec{x_i}, \;\; \frac{\partial \vec{\hat{x}_i}}{\partial k_2^i} = r^4 \vec{x_i}, \;\; \frac{\partial \vec{\hat{x}_i}}{\partial k_3^i} = r^6 \vec{x_i}, \cdots \tag{10}$$

hence $\frac{\partial \vec{\hat{x}_i}}{\partial k_j^i} = r^{2j} \vec{x_i}$, and

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

where $J$ denotes the $2 \times 2$ ninety degree rotation matrix, $D' = (1 + k_1^i r^2 + k_2^i r^4 + k_3^i r^6 + ...)$, and $\pi'$ denotes the Jacobian of $\pi$. After lifting the integral in Eq.(9) back onto occluding boundary $C_i$ of the surface, and some further manipulations, we get

$$\frac{\partial k_j^i}{\partial t} = \int_{C_i} F_i \, \frac{r^{2j} D' ||\vec{X}_i||}{Z_i^3} < \vec{N}_i, \begin{pmatrix} X_i \\ Y_i \\ 0 \end{pmatrix} > ds \tag{12}$$

as the calibration equation for the lens distortion parameter $k_j^i$.

Note that the distortion calibration method we propose can handle different models of distortion by changing the $D$ function, and related derivatives in Eq.(10).

### 3.2 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 $k_i$ for each camera $i$. Once we obtain rough estimates for the object pose and distortion coefficients $k_i$, we can fuse a "common distortion" $\tilde{k_i}$ from these separate experiments for each camera $i$ and then jointly evolve $\tilde{k_i}$'s as follows:

$$\frac{\partial \tilde{k_i}}{\partial t} = \sum_{m=1}^{M \text{poses}} \int_{C_m} F_{i,m} \, \frac{r^{2j} D' ||\vec{X}_{i,m}||}{Z_{i,m}^3} < \vec{N}_{i,m}, \begin{pmatrix} X_{i,m} \\ Y_{i,m} \\ 0 \end{pmatrix} > ds \tag{13}$$

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

$$\frac{\partial \lambda}{\partial t} = -\sum_i \int_{S_o} F_i(g(\vec{X}_o)) < \frac{\partial (g\vec{X}_o)}{\partial \lambda}, R_s \vec{N}_o > dA_o \tag{14}$$

where $F_i$ includes computation of $I_i(D \cdot \pi \cdot g_i\ (g(\vec{X}_o)))$ with the new common distortion coefficients $\tilde{k_i}$ in the multiplying distortion factor $D$.

### 3.3 Experimental Results

We tested our distortion calibration algorithm using a white bar object, made from a foam core as shown in Fig.1. We captured 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 [11]. The images of the calibration object captured from three of the five cameras in the rig are given in Fig.2. Many desktop multi-camera systems use wide angle and inexpensive camearas which produce severe distortion effects as can be observed in these camera images. For our calibration algorithm, we initialize a surface model of the real calibration object which is shown from several vantage points in Fig.3.

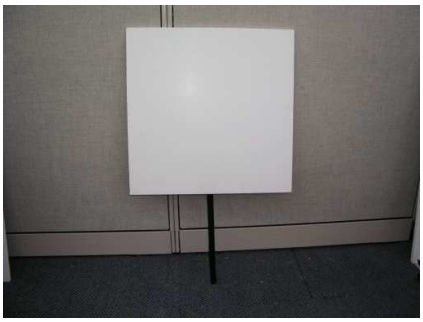

Figure 1: Photograph of the calibration object.

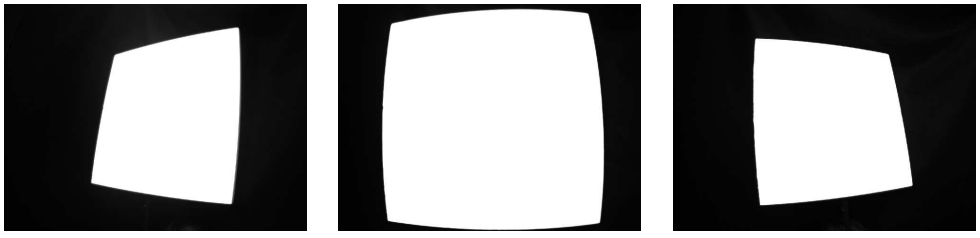

Figure 2: Three out of five camera views of the real calibration object.

Figure 3: Initialized surface model shown from three different vantage points.

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 in this report, we have 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 PDE is small in the first phase. In the experiments, the distortion function $D$ in Eq.8 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.

# 4 Color Calibration

In this section, we present a variational approach to color calibration problem in multi-camera systems. Variations in the characteristics and responses of each camera due to ambient conditions like temperature, manufacturing differences, and so on, result in different color measurements between cameras. This affects 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.

The current color calibration object is a color cube with patches of known colors whose images are captured from each camera. Demosaicing coefficients are calculated independently for each camera based upon the absolute colors on 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 discontinuties 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 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 again 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.

The differences in absolute colors measured in the response of each camera are modeled by a simple multiplicative factor in its gray scale measurements and an additive offset parameter. Note that we will extend this framework to color RGB and to RGGB images in a straightforward fashion later.

The first variation of our energy functional $E$ using this model leads to a gradient descent flow for the color calibration parameters $\alpha_i$ and $\beta_i$ for each camera $i$:

$$\frac{\partial E}{\partial \alpha_i} = -2\int_{R_i}[f-(\alpha_i I_i+\beta_i)]I_i d\vec{x} - 2\int_{R_i^c}[g-(\alpha_i I_i+\beta_i)]I_i d\vec{x},$$

$$\frac{\partial E}{\partial \beta_i} = -2\int_{R_i}[f-(\alpha_i I_i+\beta_i)]d\vec{x} - 2\int_{R_i^c}[g-(\alpha_i I_i+\beta_i)]d\vec{x}.$$

In our experiments, we utilized white noise additive offsets and multiplicative scaling coefficients to perturb the measured images, thereby exaggerating the effect of color miscalibrations. First, we initialize a model cube, and a radiance function on the cube, as shown in top row of Fig.4 along with two out of five camera views. Evolutions of the pose parameters of the cube, and the radiance function are shown in rows 2 and 3. The next three rows show how the greyscale values in these views change as the calibration coefficients are evolved ( the first row: the original captured three views, the second row: views with simulated miscalibration effects, the third row: views after the evolution of color calibration coefficients are completed).

# 5 Conclusions and Future Work

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

Pros and cons of these techniques along with future research directions are listed as follows:

- For the distortion calibration technique, the algorithm assumes knowledge of external and internal camera parameters. A future research direction would be to devise and integrate the calibration techniques for those parameters into the system as well. Indeed, an external camera calibration technique built on the same framework was developed in [12]. This is a nice feature of the methodology adopted in this report, because it can integrate a whole bunch of 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, extrinsics, and other parameters as well.
- A simplifying assumption of piecewise constant radiance model on the foreground 3D object and the background is used. More realistic models such as the piecewise smooth assumption for the foreground exist (see [13]). Our goal is also to devise and integrate the equations for a piecewise smooth model for the background to lift the simplifying assumption mentioned above. With more complex foreground and background models, these algorithms have an advantage in treating complex scenes over point-matching-based algorithms where the point picks have to be made manually.
- 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 much sensitive to noise as the former.
- Possible use of these techniques exist in constructing a 3D representation of the visual hulls generated by the image-based visual hull systems such as the one currently used at HP Labs. In our framework, a 3D volumetric object model is already available as a surface, and this representation can be deformed, smoothed or processed as required in a convenient way by the associated PDEs for geometry refinement or other purposes. The 3D shape representation is also suitable for high

level perceptual uses, such as object recognition, face detection, and so on with its available vast set of features. For instance, the surface curvature, normals, and other such geometric features are easily computable from the current representation.

To visualize the mentioned potential, we constructed the surface representation of the visual hull using silhouettes extracted by the Coliseum system of HP Labs. Three out of five camera views, along with the visual hull from different vantage points are shown in rows 1 and 2 of the Fig.9 respectively. In rows 3 and 4, reconstruction of a 3D scalar radiance field (can be a vector field for an RGB representation) on the surface is shown where the last row shows smoothing effects on the very rugged initial visual hull representation.

- The framework onto which the techniques in this report are built, was based on only diffusivity model of surfaces, however they are extendable to more general reflectance models with specularity and diffusitivity as shown in [14].

- Another advantage of our framework is that it easily accomodates 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, provide more tolerance towards errors and nothing else.

- As for near future work of the distortion calibration method, more improvements will follow with utilizing more poses, hence many more camera images of the calibration object, and more than one distortion coefficient in the model selected. We can also utilize more general/complicated distortion models than the simple polynomial $D$ function. Also, accuracy assesment tests will be carried out.

- Camera calibration application is particularly suited to our framework, since it does not have to be done in real-time which is not supported by these techniques for now, 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).

- Right now, 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 cube

with its nine parameters (scale, pose, orientation) to more accurately describe the occluding boundaries. With this approach, we won't need 3D grids for the data structures in the algorithm, and will increase both accuracy and speed.

As a result, the 3D stereo techniques based on variational ideas, and implemented via level set techniques, leads to new research directions, and finds a wide variety of applications to various problems in computer vision. We demonstrated the potential and versatility of these techniques with two applications to camera calibration problem, and many possibilities and directions of improvement in this report.

## 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] 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.

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

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

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

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

[8] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Pattern Analysis, and Machine Intelligence*, vol. 14, pp. 965–980, 1992.

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

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

[11] 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.

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

[13] H. Jin, A. J. Yezzi, Y-H. Tsai, L-T. Cheng, and S. Soatto, "Estimation of 3d surface shape and smooth radiance from 2d images: A level set approach," Tech. Rep., Preprint, 2003.

[14] H. Jin, S. Soatto, and A. Yezzi, "Multi-view stereo beyond lambert," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

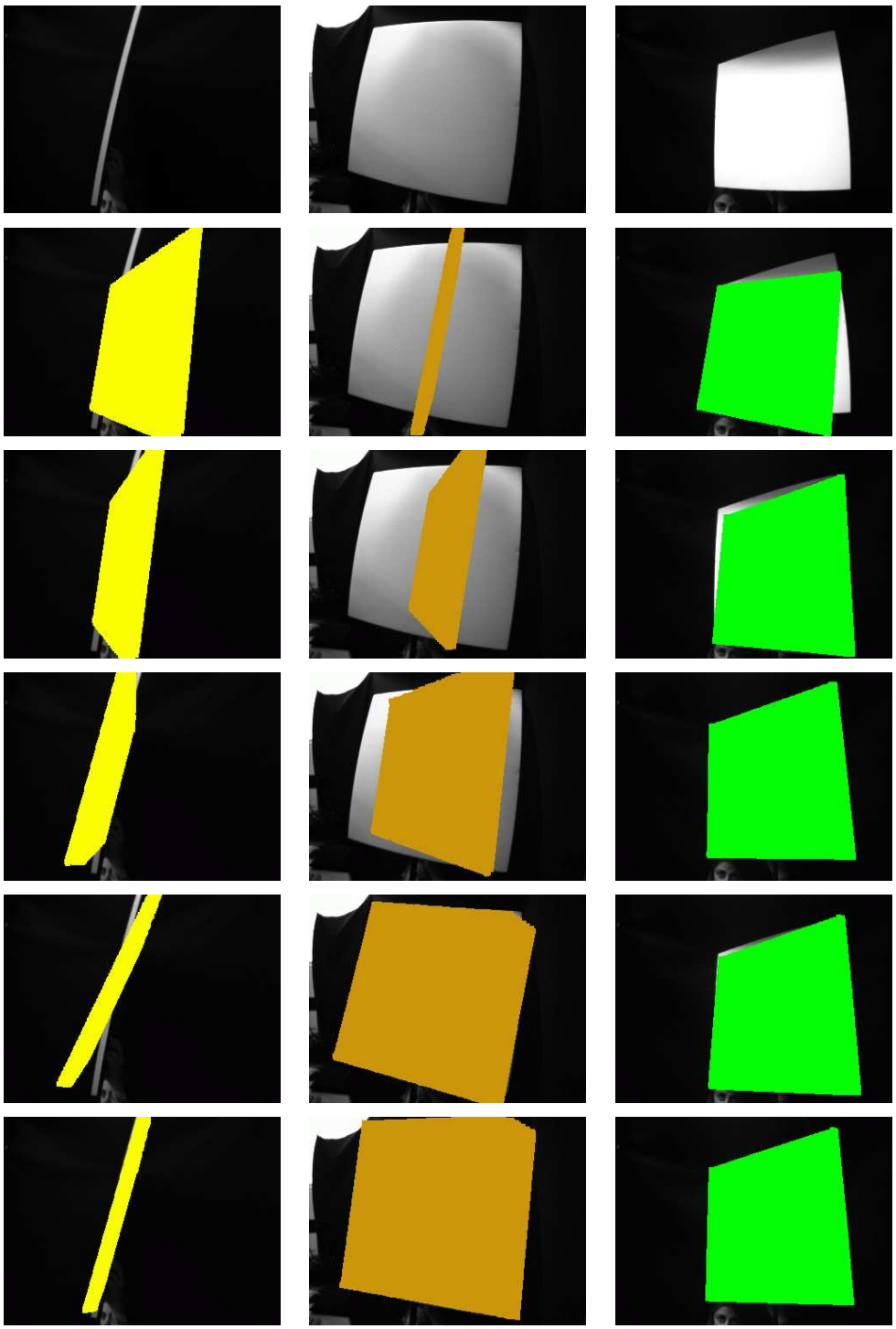

Figure 4: Column 1: Pose1. Row 1: one camera image shown, Row2: with projection of initialized surface (orange mask), Rows 3-5: during evolution of the pose parameters of the surface, Row 6: with converged pose parameters. Columns 2-3: same as column 1 for poses 2 and 3, respectively.

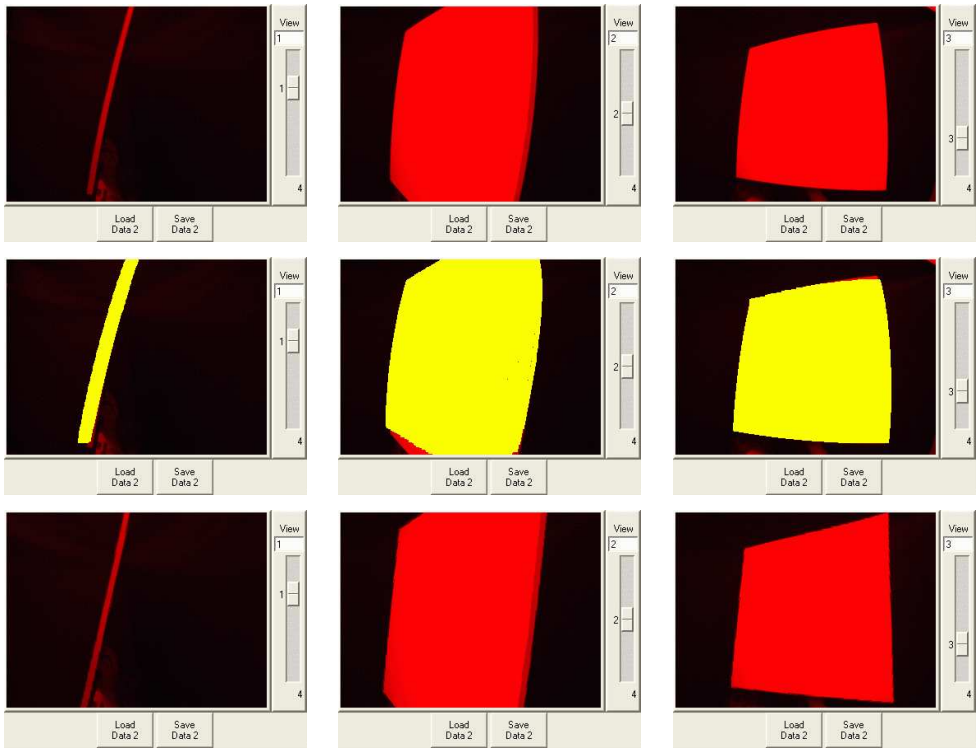


Figure 5: Pose 1. Row 1: Three out of five captured views. Row 2: Projected surface after distortion parameters have converged. Row 3: Undistorted with the obtained distortion coefficients.

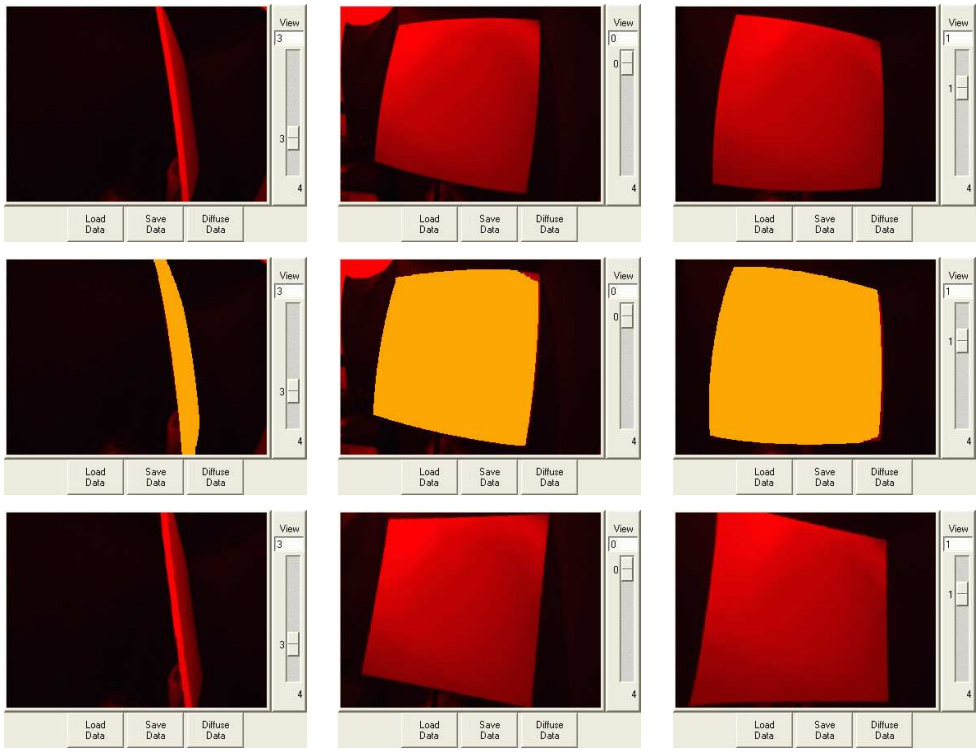


Figure 6: Pose 2. Row 1: Three out of five captured views. Row 2: Projected surface after distortion parameters have converged. Row 3: Undistorted with the obtained distortion coefficients.

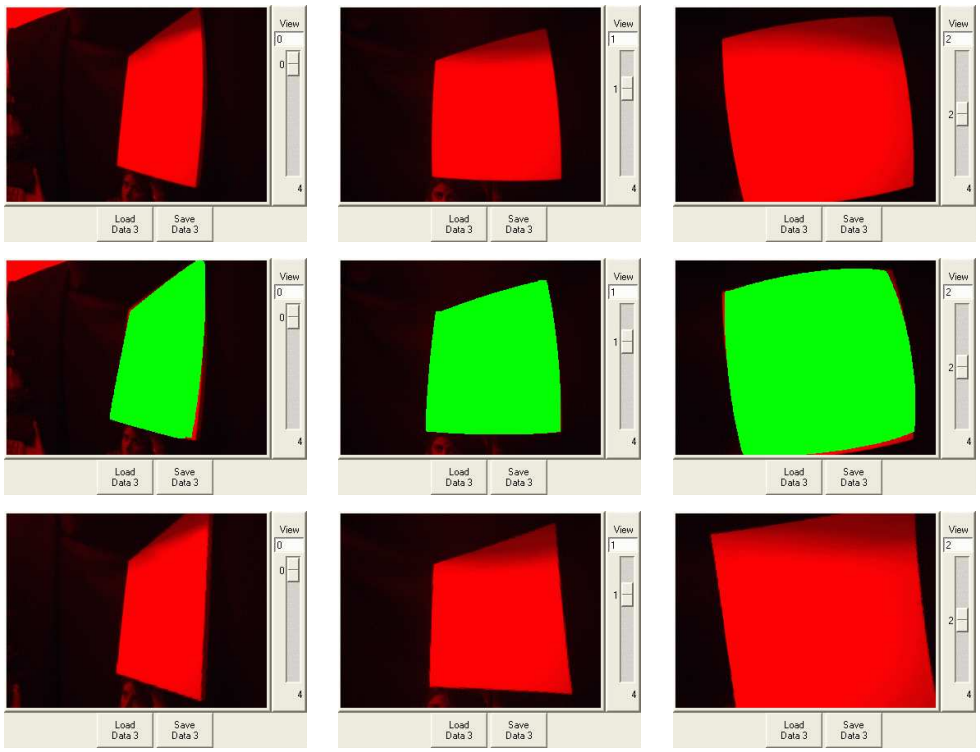


Figure 7: Pose 3. Row 1: Three out of five captured views. Row 2: Projected surface after distortion parameters have converged. Row 3: Undistorted with the obtained distortion coefficients.

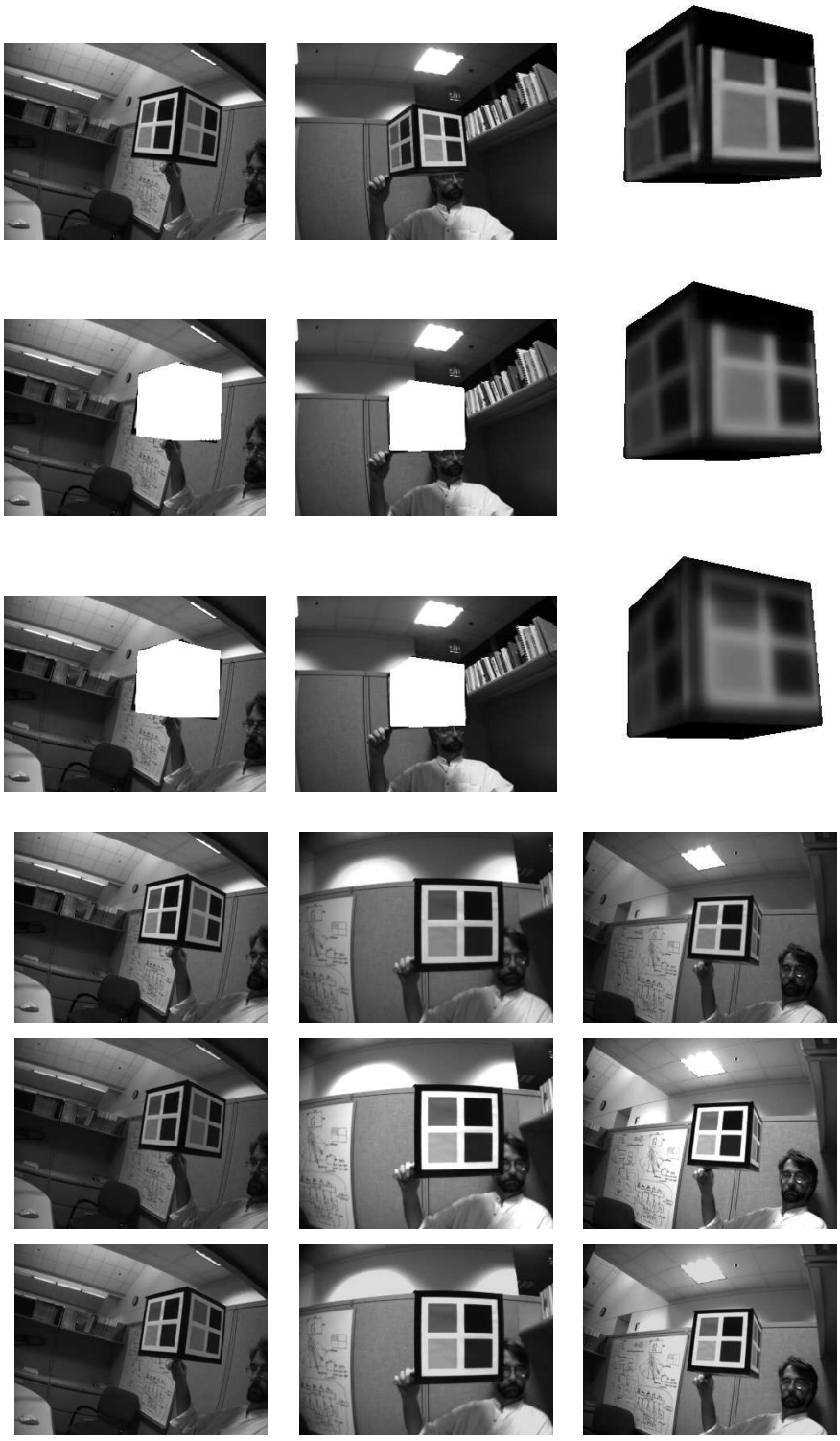

Figure 8: Top: Row 1: Two of the original views and an initialized cube. Row 2-3: Evolution of the radiance on the cube and the pose parameters of the cube. Bottom: Row 1: Three original views. Row 2: The same three different after deliberate simulated miscalibration of the greyscales. Row 3: The views after evolution of the calibration parameters has completed.

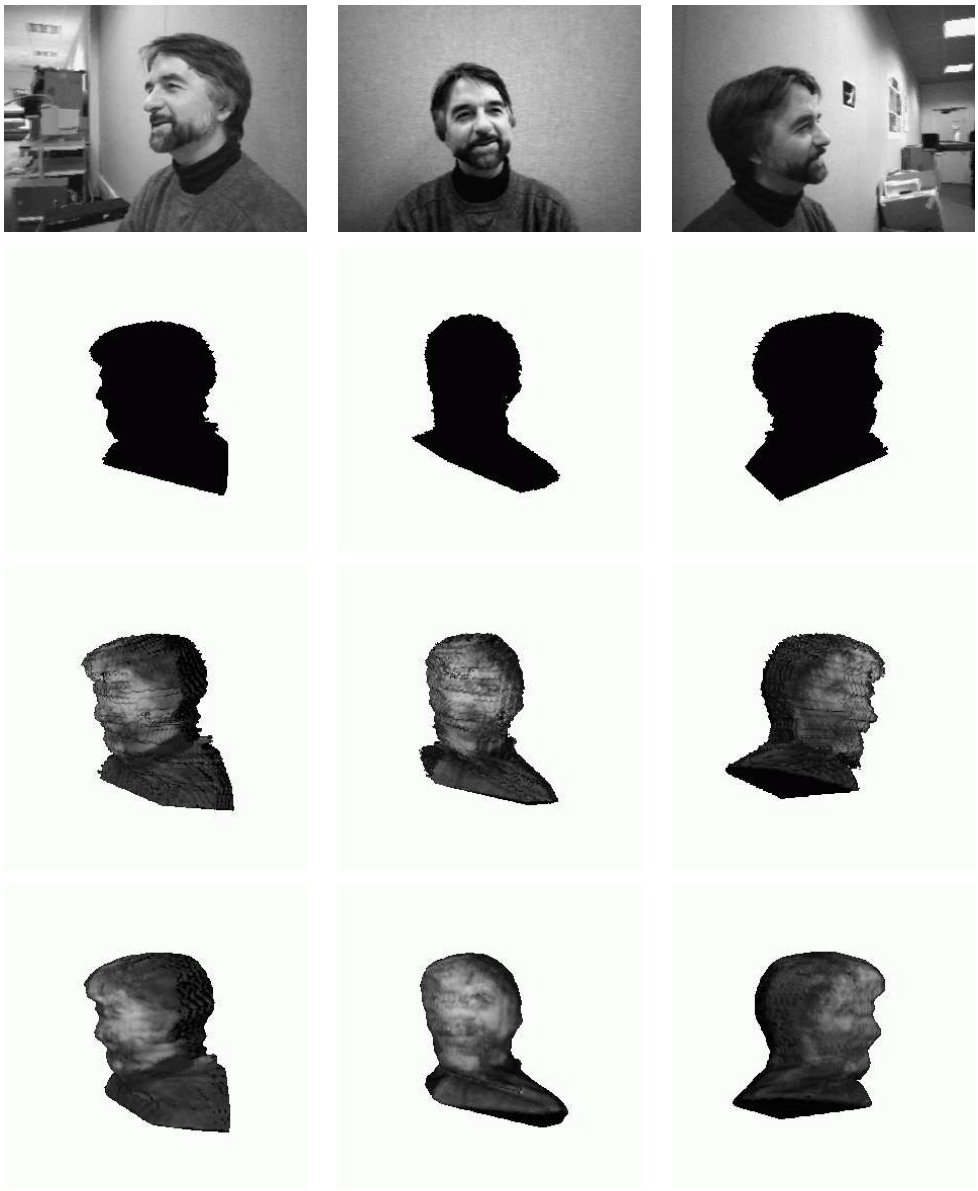

Figure 9: Row 1: Three different views, Row 2: Visual hull initialization from the extracted silhouettes shown from 3 different vantage points, Row 3: radiance initialization on the surface, Row 4: After smoothing the visual hull.