Model-Based Bundle Adjustment with Application to Face Modeling

Ying Shan*, Zicheng Liu, Zhengyou Zhang
Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA
Email: yshan@ieee.org, {zliu, zhang}@microsoft.com

Abstract

We present a new model-based bundle adjustment *algorithm to recover the 3D model of a scene/object from a sequence of images with unknown motions. Instead of representing scene/object by a collection of isolated 3D features (usually points), our algorithm uses a surface controlled by a small set of parameters. Compared with previous model-based approaches, our approach has the following advantages. First, instead of using the model space as a regularizer, we directly use it as our search space, thus resulting in a more elegant formulation with fewer unknowns and fewer equations. Second, our algorithm automatically associates tracked points with their correct locations on the surfaces, thereby eliminating the need for a prior 2D-to-3D association. Third, regarding face modeling, we use a very small set of face metrics (meaningful deformations) to parameterize the face geometry, resulting in a smaller search space and a better posed system. Experiments with both synthetic and real data show that this new algorithm is faster, more accurate and more stable than existing ones.*

Keywords: *Bundle adjustment, model-based bundle adjustment, model acquisition, structure from motion, face modeling.*

1. Introduction

Bundle adjustment (BA) has been a popular technique to refine in an optimal way both 3D structures and camera motions for a given sequence of images. The reader is referred to [17] for an excellent survey of the theory of BA as well as many implementation strategies. Conceptually BA can handle a very wide variety of scenarios. However, previous work mainly aims at recovering *isolated 3D features* (usually points and/or lines) [6, 12, 10], which we will refer to as *classical bundle adjustment* (CBA). In this paper, we describe a new type of BA aiming at directly recovering a surface model, which we will call *model-based bundle adjustment* (MBA).

1.1. Representation: Point cloud or surface model?

Whether to use a point cloud or surface model depends on the complexity of the objects/scenes to be modeled. A point cloud is definitely a more general representation, and can be used in almost all scenarios such as those shown in Fig. 1. However, in many cases such as indoor scenes and faces (see Fig. 1), 3D geometric structures exhibit strong regularities, and it is more compact to use some surface models.

In structure from motion, image features can only be extracted with limited precision, and their matching across images usually contains errors. To deal with these problems, several techniques exploiting robust statistics were reported in the last few years [18, 16]. Another approach is to make full use of geometric properties of objects/scenes to reduce errors in 3D reconstruction and motion estimation. As we will argue in this paper, surfaces can usually be represented by a smaller number of parameters than isolated points, making the overall system better posed. This has also been noted by other researchers [15, 11].

(a) (b)

Figure 1. *Examples of scenes that are difficult in (a) and possible in (b) to represent by a few parameters*

1.2. Model: regularizer or search space?

Most previous work uses a generic surface model as a *regularizer*. Surface properties are formulated as soft constraints, and are added to the traditional bundle adjustment formulation to regularize the motion and structure estimation [3, 7, 15].

*Currently at Sarnoff Corporation Princeton, NJ, USA.

Fua [3] used the model-driven bundle adjustment algorithm to compute camera motions. Based on a generic model, they impose smoothness constraints to ensure that the resulting model is not too far from the generic model. This approach improved the algorithm's robustness. However, the 3D model obtained with the model-driven bundle adjustment algorithm is in general inaccurate, and they have to throw away the model and use an additional step to recompute the 3D structure.

Kang et al [7] proposed appearance-based structure from motion using a linear combination of 3D face models (scanned from real people) as their model space. The model space is also used as regularization constraints to ensure that the resulting model is not too far from the model space. The coefficients of the 3D face models are added as additional variables to solve for. Therefore the resulting formulation is a much larger optimization problem with more unknowns and more constraints, thus more difficult to achieve convergence. Another difference between their work and ours is that they know the association between each image track and the vertex it corresponds to on the face mesh.

Debevec et al [2] used a model-based structure-from-motion algorithm to recover architecture models. Their algorithm also requires the associations of their feature tracks to be known: edges marked in the images are projections of some specific edges in the model. Another work with a similar requirement is Lowe's work on tracking deformable objects [9].

Szeliski and Torr [15] were probably the first to consider coplanarity constraints directly in structure from motion, although not implemented in a statistically optimal way. They interleave structure and motion estimation stages, and the coplanarity constraints are only imposed in structure estimation.

In this paper, we propose a new *model-based bundle adjustment* algorithm (MBA) that takes as input a set of image tracks without necessarily a prior 2D-to-3D association. The algorithm simultaneously recovers motion and surface model by *directly using the model space as our search space*. Robustness is obtained without adding additional regularization constraints. The resulting formulation is more concise and has less unknowns and constraints than previous approaches.

The closest work we found in the literature is that of McLauchlan et al. [11], where the planarity constraints are imposed by applying the technique of Lagrange multipliers. Besides the 3D coordinates of each point, they also estimate the plane parameters and the corresponding Lagrange multiplier for each plane. Our formulation can be considered as the generalization to arbitrary parametric surfaces.

We also propose a technique to reduce considerably the number of unknowns by eliminating the 3D coordinates of each point using a first order approximation. Another contribution of our work is that our algorithm automatically associates tracked points with their correct locations on the surfaces. Such a feature track is used as a point-on-surface constraint in our formulation. Our algorithm thereby eliminates the need for a prior 2D-to-3D association, and is thus suitable for a wide variety of applications. We have applied this technique to face modeling from images. We use a very small set of face metrics (meaningful deformations) to parameterize the face geometry, resulting in a smaller search space and a better posed system.

The paper is organized as follows. We formulate the MBA problem in Section 2, and derive a first order approximation of the MBA formulation that eliminates the position variables in Section 3. Section 4 describes applications to face modeling. The experimental results with both computer simulation and real data are shown in Section 5. We conclude the paper in Section 6.

In addition to those mentioned above, there are numerous works on face modeling from images such as [1, 14]. Due to space limitation, we will not go into the details.

2. Problem Statement and Model-Based Bundle Adjustment

We are interested in recovering the structure of a 3D scene/object from multiple images. Instead of representing the 3D scene/object by a collection of isolated 3D features, we consider the situations where the scene/object is a surface defined by a set of M parameters $\mathtt{C} = \{c_m | m = 1, \ldots, M\}$, called *model parameters*. Let us denote the surface by $\mathcal{S}(\mathtt{C})$. Furthermore, we assume there are Q semantically meaningful points (*semantic points* for short) $\{\mathtt{Q}_j | j = 1, \ldots, Q\}$ on the scene/object. If there are no semantic points available, then $Q = 0$. The relationship between the j^{th} semantic point $\mathtt{Q}_j$ and the scene/object parameters $\mathtt{C}$ is described by

$$\mathtt{Q}_j = \mathcal{Q}(\mathtt{C}, j) \,. \tag{1}$$

Obviously, point $\mathtt{Q}_j \in \mathcal{S}(\mathtt{C})$.

We are given a set of N images/frames, and assume we have matched a number of points of interest across images using for example the technique described in [18]. Because of occlusion, feature detection failure and other reasons, a scene/object point may be observed and detected in a subset of images. Let us call the set of image points corresponding to a single scene/object point a *feature track*. Let P be the total number of feature tracks, Θ_k be the set of frame numbers of the k^{th} feature track, $\mathbf{p}_{i,k}$ ($i \in \Theta_k$) be the feature point in the i^{th} frame that belongs to the kth feature track, and $\mathtt{P}_k$ be the corresponding 3D point, which is unknown, on the scene/object surface. Furthermore, we assume that the j^{th} semantic point $\mathtt{Q}_j$ is observed and detected in zero or more images. Let Ω_j be the, possibly empty, set of frame

numbers in which $\mathtt{Q}_j$ are detected, and $\mathbf{q}_{l,j}$ ($l \in \Omega_j$) be the detected semantic point in the l^{th} frame.

Image i is taken by a camera with unknown pose parameters $\mathtt{M}_i$ which describes the position and orientation of the camera with respect to the world coordinate system in which the scene/object is described. Although many camera models exist, we use the standard perspective projection. The relationship between a 3D point $\mathtt{P}_k$ and its image point $\mathbf{p}_{i,k}$ is described by function ϕ such that

$$\mathbf{p}_{i,k} = \phi(\mathtt{M}_i, \mathtt{P}_k) \ . \tag{2}$$

For simplicity, the internal camera parameters are assumed to be known, although it is trivial to add them in our formulation; indeed, these parameters are incorporated in our implementation.

We can now state the problem as follows: Given P tracks of feature points $\{\mathbf{p}_{i,k} | k = 1, \ldots, P; i \in \Theta_k\}$ and Q tracks of semantic points $\{\mathbf{q}_{l,j} | j = 1, \ldots, Q; l \in \Omega_j\}$, determine the scene/object model parameters $\mathtt{C}$ and the camera pose parameters $\mathtt{M} = [\mathtt{M}_1^T, \ldots, \mathtt{M}_N^T]^T$.

Our objective is to solve this problem in an optimal way. By *optimal* we mean to find simultaneously the scene/object parameters and camera parameters by minimizing some statistically and/or physically meaningful cost function. As in the classical point-based bundle adjustment, it is reasonable to assume that the image points are corrupted by independent and identically distributed Gaussian noise because points are extracted independently from images by the same algorithm. In that case, the maximum likelihood estimation is obtained by minimizing the sum of squared errors between the observed image points and the predicted feature points. More formally, the problem becomes

$$\begin{aligned} \min_{\mathtt{M},\mathtt{C},\{\mathtt{P}_k\}} \Bigg(& \sum_{k=1}^{P} \sum_{i \in \Theta_k} \|\mathbf{p}_{i,k} - \phi(\mathtt{M}_i, \mathtt{P}_k)\|^2 \\ & + \sum_{j=1}^{Q} \sum_{l \in \Omega_j} \|\mathbf{q}_{l,j} - \phi(\mathtt{M}_l, \mathtt{Q}_j)\|^2 \Bigg) \\ & \text{subject to } \mathtt{P}_k \in \mathcal{S}(\mathtt{C}), \end{aligned} \tag{3}$$

where $\mathtt{Q}_j = \mathcal{Q}(\mathtt{C}, j)$ as defined in (1). Note that although the part for the general feature points (the first term) and the part for the semantic points (the second term) have the same form, we should treat them differently. Indeed, the latter provides stronger constraint in bundle adjustment than the former. We can simply substitute $\mathtt{Q}_j$ in the second part by $\mathcal{Q}(\mathtt{C}, j)$ while $\mathtt{P}_k$ must be searched on the surface $\mathcal{S}(\mathtt{C})$.

In most practical problems, not all possible values of parameters $\mathtt{C}$ are acceptable, and it is often necessary or desirable to impose constraints. There are many forms of constraints: linear or nonlinear, equality or inequality. The reader is referred to [5] for various techniques to deal with constrained optimization problems. An important case is when a parameter c_m is subject to bounds: $l_m \le c_m \le u_m$. For each such constraint, we add to (3) two penalty terms for the lower and upper bound. For the lower bound, the penalty term is defined by

$$p_m = \begin{cases} 0 & \text{if } c_m \ge l_m; \\ \rho(l_m - c_m)^2 & \text{otherwise,} \end{cases}$$

where the non-negative value ρ is the penalty parameter.

2.1 Comparison Between CBA and MBA

Compared with the classical point-based bundle adjustment (CBA), our model-based bundle adjustment (3) (MBA) has a similar form except that we have model parameters $\mathtt{C}$ and points $\mathtt{P}_k$ are constrained. In CBA, there are no model parameters but points $\mathtt{P}_k$ are free. Although it appears that MBA has more parameters to estimate, the real number of free parameters is usually smaller because of the constraint on points $\mathtt{P}_k$. Indeed, the total number of free parameters in CBA is equal to $6(N-1) - 1 + 3P$ ("-1" is due to the fact that the global scale cannot be determined), while the total number of free parameters in MBA is equal to $6(N-1) - 1 + M + 2P$ because each point on a surface has two degrees of freedom. As long as $P > M$ (the number of feature tracks is larger than that of model parameters), the parameter space for MBA is smaller than for CBA.

Take a simple example of recovering a plane from images. A plane can be defined in the first camera coordinate system with three parameters. Therefore, the parameter space for MBA is smaller if there are more than 3 point tracks across images, and we do need at least 4 point tracks to recover a plane.

In Section 3 we will show how to eliminate $\mathtt{P}_k$ in (3) and derive a much smaller minimization problem.

3. Simplifying the Minimization Problem

We observe that in (3), unknown 3D points $\{\mathtt{P}_k\}$, which correspond to feature tracks, are not involved at all in the second term. Furthermore, in the first term, the second summation only depends on each individual $\mathtt{P}_k$. We can therefore rewrite (3) as

$$\begin{aligned} \min_{\mathtt{M},\mathtt{C}} \Bigg(& \sum_{k=1}^{P} \min_{\mathtt{P}_k} \sum_{i \in \Theta_k} \|\mathbf{p}_{i,k} - \phi(\mathtt{M}_i, \mathtt{P}_k)\|^2 \\ & + \sum_{j=1}^{Q} \sum_{l \in \Omega_j} \|\mathbf{q}_{l,j} - \phi(\mathtt{M}_l, \mathtt{Q}_j)\|^2 \Bigg) \\ & \text{subject to } \mathtt{P}_k \in \mathcal{S}(\mathtt{C}). \end{aligned} \tag{4}$$

Henceforward, we will consider only the first term and try to eliminate $\{\mathtt{P}_k\}$ in order to simplify the minimization problem. Instead of estimating $\{\mathtt{P}_k\}$, it is equivalent to estimating its projection in each image. We can therefore rewrite the inner minimization problem with respect to $\mathtt{P}_k$

as the following problem:

$$\min_{\{\widehat{\mathbf{p}}_i\}} \sum_{i\in\Theta_k} \|\mathbf{p}_i - \widehat{\mathbf{p}}_i\|^2$$
$$\text{subject to } \widehat{\mathbf{p}}_i = \boldsymbol{\varphi}_i(\mathcal{S}(\mathtt{C}), \widehat{\mathbf{p}}_r) \text{ for } i \in \Theta'_k, \quad (5)$$

where $\mathbf{p}_i \equiv \mathbf{p}_{i,k}$, $\widehat{\mathbf{p}}_i \equiv \widehat{\mathbf{p}}_{i,k}$, $\{\widehat{\mathbf{p}}_i \mid i \in \Theta_k\}$ is the set of projected 2D points of a 3D point on surface $\mathcal{S}(\mathtt{C})$, $\boldsymbol{\varphi}_i(\mathcal{S}(\mathtt{C}), \widehat{\mathbf{p}}_r)$ is an image transfer function (see below), and $\Theta'_k = \Theta_k - \{r\}$. Here, we choose image r as reference, and $\widehat{\mathbf{p}}_r$ is the projection of the 3D point to be estimated in the reference image. The image transfer function $\boldsymbol{\varphi}_i(\mathcal{S}(\mathtt{C}), \widehat{\mathbf{p}}_r)$ finds the intersection $\widehat{\mathtt{P}}$ of a ray from the reference point $\widehat{\mathbf{p}}_r$ with surface $\mathcal{S}(\mathtt{C})$ and then projects $\widehat{\mathtt{P}}$ onto the ith image; it is thus a function of $\widehat{\mathbf{p}}_r$, $\mathcal{C}$, and $\mathtt{M}_i$. Note that at this point there is no approximation at all. Regardless of which reference image used, problem (5) is the same as the original one except that we are estimating a 2D point $\mathbf{p}_r$ instead of a 3D point $\mathtt{P}$ on a surface.

Defining

$$\Delta\mathbf{p}_i = \mathbf{p}_i - \widehat{\mathbf{p}}_i \quad \Rightarrow \quad \widehat{\mathbf{p}}_i = \mathbf{p}_i - \Delta\mathbf{p}_i \quad (6)$$

and substituting (6) into (5), we can combine all the constraints as

$$\mathcal{F}_k \equiv \sum_{i\in\Theta'_k} \|\mathbf{p}_i - \Delta\mathbf{p}_i - \boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p}_r - \Delta\mathbf{p}_r)\|^2 = 0\ . \quad (7)$$

By applying Taylor expansion to $\boldsymbol{\varphi}_i$ at $\mathbf{p}_r$ and keeping the first order terms, we have:

$$\boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p}_r - \Delta\mathbf{p}_r) = \boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p}_r) - \boldsymbol{\Phi}_{i,r}\Delta\mathbf{p}_r\ , \quad (8)$$

where $\boldsymbol{\Phi}_{i,r} = \partial\boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p})/\partial\mathbf{p}\,|_{\mathbf{p}=\mathbf{p}_r}$ is the Jacobian matrix of $\boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p})$ evaluated at $\mathbf{p}_r$. Equation (7) then becomes:

$$\begin{aligned}\mathcal{F}_k &= \sum_{i\in\Theta'_k} \|\mathbf{p}_i - \Delta\mathbf{p}_i - \boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p}_r) + \boldsymbol{\Phi}_{i,r}\Delta\mathbf{p}_r\|^2 \\ &\approx \sum_{i\in\Theta'_k} (2\mathbf{a}_i^T\boldsymbol{\Phi}_{i,r}\Delta\mathbf{p}_r - 2\mathbf{a}_i^T\Delta\mathbf{p}_i + \mathbf{a}_i^T\mathbf{a}_i) \quad (9)\end{aligned}$$

where $\mathbf{a}_i = \mathbf{p}_i - \boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p}_r)$, and the second term of (9) is achieved by keeping only first order terms. Using the Lagrange multiplier, we can transform (5) into an unconstrained minimization problem, and the objective function is given by

$$\mathcal{J}_k = \sum_{i\in\Theta_k} \|\Delta\mathbf{p}_i\|^2 + \lambda\mathcal{F}_k\ . \quad (10)$$

To minimize $\mathcal{J}_k$, its first derivatives with respect to $\Delta\mathbf{p}_i$ must be equal to 0, that is:

$$\frac{\partial\mathcal{J}_k}{\partial\Delta\mathbf{p}_r} = 2\Delta\mathbf{p}_r + 2\lambda\sum_{i\in\Theta'_k}\boldsymbol{\Phi}_{i,r}^T\mathbf{a}_i = 0$$
$$\frac{\partial\mathcal{J}_k}{\partial\Delta\mathbf{p}_i} = 2\Delta\mathbf{p}_i - 2\lambda\mathbf{a}_i = 0 \quad \text{for } i \in \Theta'_k$$

The solution is

$$\Delta\mathbf{p}_r = -\lambda\mathbf{d} \quad (11)$$
$$\Delta\mathbf{p}_i = \lambda\mathbf{a}_i \quad \text{for } i \in \Theta'_k \quad (12)$$

where $\mathbf{d} = \sum_{i\in\Theta'_k}\boldsymbol{\Phi}_{i,r}^T\mathbf{a}_i$. Substituting (11) and (12) into (9) leads to:

$$\mathcal{F}_k = \sum_{i\in\Theta'_k} (-2\lambda\mathbf{a}_i^T\boldsymbol{\Phi}_{i,r}\mathbf{d} - 2\lambda\mathbf{a}_i^T\mathbf{a}_i + \mathbf{a}_i^T\mathbf{a}_i)\ . \quad (13)$$

Solving (13) for λ gives:

$$\begin{aligned}\lambda &= \frac{\sum_{i\in\Theta'_k}\mathbf{a}_i^T\mathbf{a}_i}{2[\sum_{i\in\Theta'_k}(\mathbf{a}_i^T\boldsymbol{\Phi}_{i,r}\mathbf{d} + \mathbf{a}_i^T\mathbf{a}_i)]} \\ &= \frac{\sum_{i\in\Theta'_k}\mathbf{a}_i^T\mathbf{a}_i}{2(\mathbf{d}^T\mathbf{d} + \sum_{i\in\Theta'_k}\mathbf{a}_i^T\mathbf{a}_i)}\ . \quad (14)\end{aligned}$$

Substituting $\Delta\mathbf{p}_i$ and λ into (10)gives:

$$\begin{aligned}\mathcal{J}_k &= \Delta\mathbf{p}_r^T\Delta\mathbf{p}_r + \sum_{i\in\Theta'_k}\Delta\mathbf{p}_i^T\Delta\mathbf{p}_i \\ &= \lambda^2(\mathbf{d}^T\mathbf{d} + \sum_{i\in\Theta'_k}\mathbf{a}_i^T\mathbf{a}_i) \\ &= \frac{\left(\sum_{i\in\Theta'_k}\mathbf{a}_i^T\mathbf{a}_i\right)^2}{4\left(\mathbf{d}^T\mathbf{d} + \sum_{i\in\Theta'_k}\mathbf{a}_i^T\mathbf{a}_i\right)}\ . \quad (15)\end{aligned}$$

Therefore, under the first order approximation, the constrained minimization problem (4) becomes the following unconstrained one:

$$\min_{\mathtt{M},\mathtt{C}} \left(\sum_{k=1}^{P}\mathcal{J}_k + \sum_{j=1}^{Q}\sum_{l\in\Omega_j}\|\mathbf{q}_{l,j} - \boldsymbol{\phi}(\mathtt{M}_l, \mathtt{Q}_j)\|^2\right)\ . \quad (16)$$

3.1. How to choose the reference image?

Note that equation (16) is obtained from (4) through the linearization of the transfer function $\boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p}_r)$. The choice of the reference image r determines how well the transfer function can be approximated by the linear function (8).

Let's examine the image transfer function $\boldsymbol{\varphi}_i(\mathcal{S}, \mathbf{p}_r)$. Denote the relative motion (rotation and translation) from image r to image i by $(\mathbf{R}_i^r, \mathbf{t}_i^r)$. Let $\mathtt{P}$ be the point on surface $\mathcal{S}$ corresponding to $\mathbf{p}_r$. In the neighborhood around $\mathtt{P}$, surface $\mathcal{S}$ can be approximated by its tangent plane Π. Let Π be represented in the camera coordinate system of image r by a unit normal vector $\mathbf{n}$ and the distance d from the origin to the plane. It can be easily shown that $\mathbf{p}_i$ is related to $\mathbf{p}_r$ through a homography $\mathbf{H}_i^r$, i.e.,

$$\widetilde{\mathbf{p}}_i = \lambda\mathbf{H}_i^r\widetilde{\mathbf{p}}_r \quad \text{with } \mathbf{H}_i^r = \mathbf{R}_i^r + \frac{1}{d}\mathbf{t}_i^r\mathbf{n}^T\ , \quad (17)$$

where $\widetilde{\mathbf{p}} = [\mathbf{p}^T, 1]^T$ and λ is a non-zero scalar.

If $\mathbf{H}_i^r$ is an affine matrix (i.e., the first two elements of the third row are equal to 0), then $\mathbf{p}_i$ and $\mathbf{p}_r$ are related by a linear function, and the linear approximation error of the transfer function is 0. This happens if the following two conditions are satisfied:

1. the rotation axis is parallel to z-axis;
2. there is no translation in z (i.e., $t_z = 0$) *or* the plane is parallel to image plane r (i.e., $n_x = n_y = 0$).

We therefore should choose r such that the above two conditions can be satisfied as close as possible for all $i \in \Theta'_k$. In our implementation, we choose the central frame of a feature track as the reference image for the following two reasons. First, the central frame usually corresponds to the best viewpoint for that feature, i.e., with smallest value in n_x and n_y. This is because the first and last frames usually exhibit strong distortion, making the tracking broken. Second, the nonlinear effect of the transfer function due to rotation around axis not parallel to the z-axis is smaller if the central frame is chosen as the reference.

4. MBA for Face Modeling

In this section, we show how to apply MBA to the reconstruction of 3D face models from image sequences. We represent a face as a linear combination of a neutral face mesh and a certain number of face *metrics*. A metric is a vector that linearly deforms a face in certain way, such as to make the head wider, make the nose bigger, etc. To be more precise, let us denote the neutral face mesh as $\mathcal{S}^0 = ({\mathbf{v}_1^0}^T, \ldots, {\mathbf{v}_n^0}^T)^T$, where $\mathbf{v}_i^0$ $(i = 1, ..., n)$ are the vertices of the mesh. Denote the metrics as $\mathcal{M}^j = ({\delta\mathbf{v}_1^j}^T, \ldots, {\delta\mathbf{v}_n^j}^T)^T$, $j = 1, ..., m$. For each vector $C_{coef} = (c_1, c_2, ..., c_m)^T$, its corresponding face mesh is

$$\mathcal{S}^*(C_{coef}) = \mathcal{S}^0 + \sum_{j=1}^{m} c_j \mathcal{M}^j. \tag{18}$$

The c_j's are called *metric coefficients*.

Notice that Equation (18) evaluates the vertices in the coordinate system where the neutral mesh and metrics were designed. We use $\mathbf{M}_p$ to denote the transformation matrix from this original face mesh coordinate system to the camera coordinate system. Then $\mathcal{S}(C) = \mathbf{M}_p(\mathcal{S}^*(C_{coef}))$ is the face mesh in the camera coordinate system. Therefore, including the 6 parameters for $\mathbf{M}_p$, the total number of model parameters $M = m + 6$.

In our implementation, the neutral face and all the metrics are designed by an artist. The neutral face contains 194 vertices and 360 triangles (Figure 5). There are 65 metrics (i.e., $m = 65$). Each metric is associated with a valid range $[l_j, u_j]$ with $l_j \leq c_j \leq u_j$. We use a penalty function to handle these inequality constraints as described in Section 2.

All the vertices on the face mesh have meanings, and the user is allowed to add zero or more semantic point constraints. One way to specify semantic point constraints is to mark some face features (such as eye corners) on some of the images. The vertices corresponding to these features will be constrained so that they project to these markers (the second term of equation (16)).

To obtain a reasonable initial guess, we use the results from the face modeling system as reported in [8] as the initial guess of our MBA algorithm. Their system takes as input a sequence of images and five manual markers (inner eye corners, nose tip, and mouth corners) on two frontal views (i.e., $Q = 5$ in equation (16)). It computes the face geometry from the two frontal views, and then uses the computed face model to track the rest of the images. Finally it combines these images into a single cylindrical texture map. Notice that in their system, the geometry is computed from only two frontal views, so the results are in general not very accurate. On the other hand, the system is fast, robust, and the resulting face models and the camera motions are reasonable.

Equation (16) is solved using the Levenberg-Marquardt algorithm [13] with the gradients computed numerically.

5. Experiments

In this section, we provide experimental results of our MBA algorithm with both synthetic (Sect. 5.1) and real (Sect. 5.2) data for face modeling.

5.1. Synthesized data

With synthetic data, we compare MBA with CBA because we know the ground truth. We choose to use the structure error as the error metric for comparison since MBA results in a 3D face model and the structure error directly affects the visual quality of the 3D reconstruction. For CBA, we add a 3D geometrical fitting stage to fit a face model to the 3D point clouds reconstructed by CBA. From the five markers (see Section 4), we compute their 3D positions and use them to initialize the pose of the face model. The fitting process then searches for both the pose of the face and the metric coefficients to minimize the distances from the reconstructed 3D points (including the markers) to the face mesh. We use an iterative closest point approach to solve this problem (See [4, 8] for details).

The structure error is computed as follows. For each sampled point (corresponding to a track) on the ground truth face model, compute the corresponding point (with the same semantic meaning) on the estimated face model. Then we compute a *similarity transformation* (scale, rotation, and translation) from the point set on the estimated face model to that on the ground truth model. The remaining distance from the transformed point set to that on the ground truth model is defined as the structure error. The reason to compute a similarity transformation between the two point sets is due to the obvious fact that structure from motion can only be determined up to a similarity transformation. Notice that in all the experiments below, we normalize the ground truth mesh so that its bounding box size is 1.

The ground truth data was synthesized in such a way that the distribution of the tracks are similar to a real situation example where a person turns his head in front of a static

camera with corner matching algorithm being used to compute pair-wise image matches. There are 4 views and 232 tracks.

Sensitivity to initial guess and data noise. In the first experiment, the ground truth data was perturbed in the following way. First, noise with a given percentage (referred to as *initial guess perturbation* hereafter) is added to both the camera motions and face parameters to simulate inaccurate initial guesses. Second, zero-mean Gaussian noise with a given value of standard deviation (referred to as *image noise level* hereafter) is added to the image features. For each different initial guess, 30 independent random trials of experiments are conducted, and the mean of the errors as well as the mean of the used CPU time are recorded. Figure 2 plots the error (vertical axis) vs. the image noise level (horizontal axis). Notice that since the size of the ground truth mesh is 1, the vertical axis is actually the amount of error relative to the size of the face. Each curve corresponds to a different initial guess. The numbers inside the legend box denote the initial guess perturbation percentage (for example, MBA10 means 10 percent perturbation). We can see that the curves for MBA corresponding to the three initial guesses almost coincide; the same holds for CBA. We see clearly that MBA is significantly better than CBA. For example, when the image noise level is 1 pixel, the error for MBA is about 0.75% of the head size, and is 3% for CBA.

Computational time. Figure 3 is the plot of the mean of the used CPU time in seconds. We can see that the MBA method is in general faster than the CBA method. We should point out that we have taken advantage of the sparseness property in our CBA implementation to speed up computation but not in the MBA method (there are some sparseness properties in MBA formulation, though not as much as in CBA, but we are not taking advantage of it in our current implementation); thus, the computational gain with MBA is potentially even higher. Another interesting point is that the computation time of the MBA varies much less than CBA as the initial guess changes or the image noise level changes. This indicates that MBA is more stable than CBA.

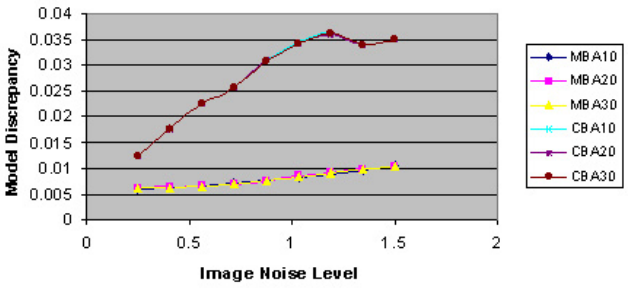

Figure 2. *Model validation.*

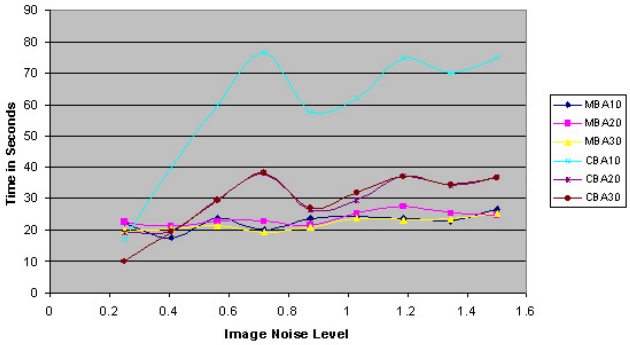

Figure 3. *Time duration.*

Sensitivity to outliers. The second experiment tests the robustness of the algorithm in the presence of outliers. We use the same synthesized data as in the first experiment. The initial guess perturbation is set to 10% and the image noise level is set to 0.5 pixels. We increase the percentage of the outlier from 5% to 30%, while the total number of tracks is kept fixed. The outliers are sampled from a Gaussian distribution with much larger standard deviation, 3 pixels in this case. Fig.4 shows the results. We can see that, as the outlier percentage increases, the error with MBA increases more slowly than that with CBA.

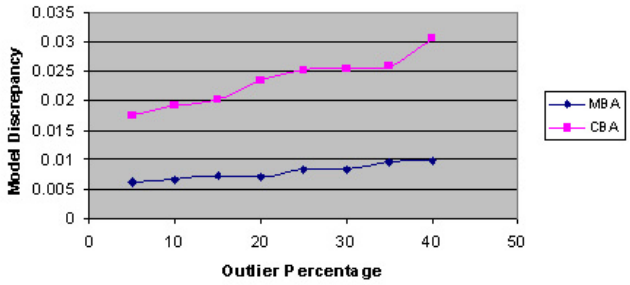

Figure 4. *Robust test.*

Visual comparison. Figure 5 shows visual comparison of the face meshs from the two methods with initial guess perturbation 10% and image noise level 0.25 pixels. The top are the front views and the bottom are the side views. On each row, the one in the middle is the ground truth, on the left is the result from CBA, and the result from MBA is on the right. By a close look, we can see that the MBA result is much closer to the ground truth mesh.

Sensitivity to interframe motion. The data in this experiment are synthesized differently. For each view, we randomly sample a set of 3D points from an existing face model (these points must be visible from this viewpoint). Each sampled point is then projected onto this view plus its two neighboring views (so the track length is 3). The number of tracks is 262. The cameras are assumed to be on a circle with center at the centroid of the face model. Figure 6 shows how the error varies as the camera rotation an-

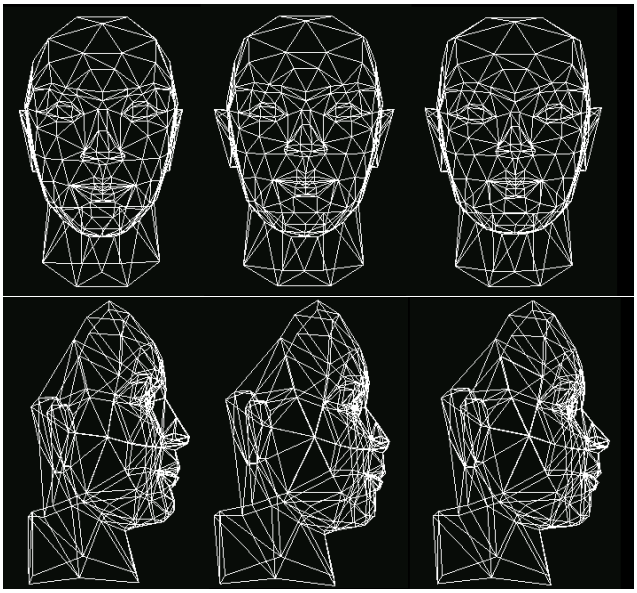

Figure 5. *Face mesh comparison. Left: CBA; Middle: Ground truth; Right: MBA.*

gle changes. As the camera motion decreases (from right to left), the errors grow for both methods as expected, but the error with MBA grows much more slowly than with CBA.

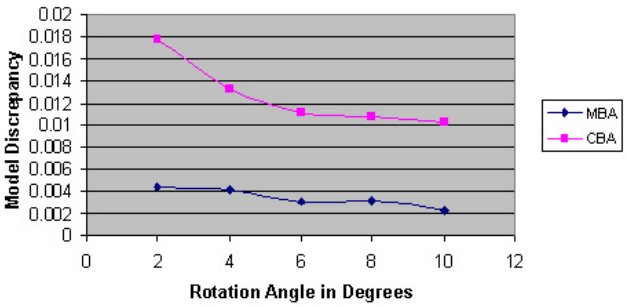

Figure 6. *Change of the rotation angle.*

Sensitivity to the number of feature tracks. In this experiment, there are 4 views, and the track length is 3. The rotation angle between every two neighboring views is 3 degrees. Figure 7 is the plot of the error vs. the number of tracks. As the number of tracks decreases (looking from right to left), the error with CBA grows much faster than with MBA.

5.2. Real data

In this section, we show some results of face modeling from a sequence of images. The image sequences are obtained by asking people turning their heads in front of a static camera. We first run the rapid face modeling system [8] and use the results as the initial guess of the MBA algorithm. The final result is a face mesh with a cylindrical texture map. For each example, we show both the initial

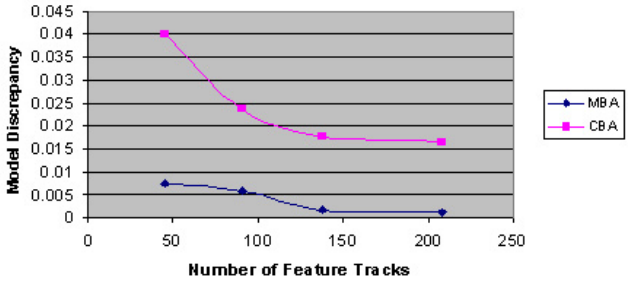

Figure 7. *Change of the number of sample points.*

guesses (results from the rapid face modeling system) and the final results from the MBA algorithm. Each sequence contains 23 to 26 images of resolution 640x480. The number of feaure tracks ranges from 1500 to 2400. There are 50 to 150 image matches between each pair of neighboring views. For each sequence, the total running time of the MBA algorithm is about 6 to 8 minutes on a 850MHz PentiumIII machine.

The first example is shown in Fig. 8. The left column is the initial guess. The right column is the result from the MBA algorithm. The images in the middle are the acquired images. We can see that the face of the initial guess is too narrow compared to the actual face, and there is a clear improvement with the result from the MBA algorithm.

The second example is shown in Fig. 9. Again, the left column is the initial guess and the right column is the result from MBA. We can see that the upper portion of the face (above the eyes) is too wide for the initial guess while the result from MBA looks more like the actual person.

The third example is shown in Fig. 10. We can see that the profile of the inital guess is quite different from the actual person. With MBA, the profile closely matches the profile of the actual person.

6. Conclusion and future directions

We have proposed model-based bundle adjustment as a general formulation to incorporate model knowledge into traditional bundle adjustment, and have derived a minimization problem that eliminates all the 3D point position variables, thus resulting in a much smaller optimization problem. We have implemented this approach for face modeling. Our experiment results have shown that, compared with the traditional bundle adjustment, the model-based bundle adjustment algorithm not only is more robust just like other model based approaches, but also is faster and more accurate.

The major computation cost of our current implementation is the gradient computation. We are planning to use the sparseness in the MBA formulation to speed up its computation. We would like to apply model-based bundle adjustment to other objects such as human body, arms, etc.

Figure 8. *First textured face model. Left: initial guess; Middle: original images; Right: MBA.*

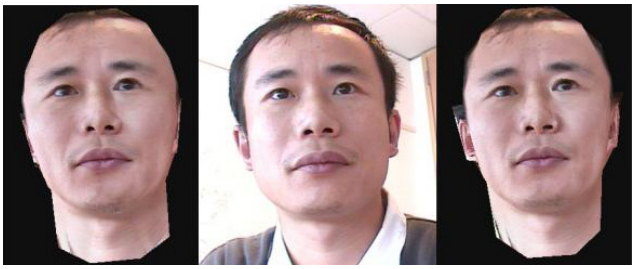

Figure 9. *Second textured face model.*

References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Computer Graphics, Annual Conference Series*, pages 187–194. Siggraph, August 1999.

[2] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Computer Graphics, Annual Conference Series*, pages 11–20. Siggraph, August 1996.

[3] P. Fua. Using model-driven bundle-adjustment to model heads from raw video sequences. In *Proceedings of the 7th International Conference on Computer Vision*, pages 46–53, Corfu, Greece, Sept. 1999. IEEE Computer Society Press.

Figure 10. *Third textured face model.*

[4] P. Fua and C. Miccio. From regular images to animated heads: A least squares approach. In *European Conference on Computer Vision*, pages 188–202, 1998.

[5] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.

[6] R. Hartley. Euclidean reconstruction from uncalibrated views. In J. Mundy and A. Zisserman, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*, pages 237–256, Berlin, Germany, 1993. Springer-Verlag.

[7] S. B. Kang and M. Jones. Appearance-based structure from motion using linear classes of 3-d models. *Manuscript*, 1999.

[8] Z. Liu, Z. Zhang, C. Jacobs, and M. Cohen. Rapid modeling of animated faces from video. In *Proc. 3rd International Conference on Visual Computing*, pages 58–67, Mexico City, Sept. 2000. Also available as MSR technical report from http://research.microsoft.com/˜zhang/Papers/TR00-11.pdf.

[9] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[10] P. McLauchlan and D. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proc. Fifth International Conference on Computer Vision*, pages 314–320, Cambridge, Massachusetts, June 1995.

[11] P. McLauchlan, X. Shen, P. Palmer, A. Manessis, and A. Hilton. Surface-based structure-from-motion using feature groupings. In *Proceedings of the Asian Conference on Computer Vision*, pages 699–705, Taipei, Taiwan, Jan. 2000. Springer-Verlag.

[12] R. Mohr, B. Boufama, and P. Brand. Accurate projective reconstruction. In J. Mundy and A. Zisserman, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*, pages 257–276, Berlin, 1993. Springer-Verlag.

[13] J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag, 1977.

[14] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *Computer Graphics, Annual Conference Series*, pages 75–84. Siggraph, July 1998.

[15] R. Szeliski and P. Torr. Geometrically constrained structure from motion: Points on planes. In *Proceedings of the European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE)*, pages 171–186, Freiburg, June 1998.

[16] P. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997.

[17] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, Corfu, Greece, Sept. 1999.

[18] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, Oct. 1995.