Perspective Factorization Methods for Euclidean Reconstruction

Mei Han Takeo Kanade

August 1999
CMU-RI-TR-99-22

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this paper we describe a factorization-based method for Euclidean reconstruction with a perspective camera model. It iteratively recovers shape and motion by weak perspective factorization method and converges to a perspective model. We discuss the approach of solving the reversal shape ambiguity and analyze its convergence. We also present a factorization-based method to recover Euclidean shape and camera focal lengths from multiple semi-calibrated perspective views. The focal lengths are the only unknown intrinsic camera parameters and they are not necessarily constant among different views. The method first performs projective reconstruction by using iterative factorization, then converts the projective solution to the Euclidean one and generates the focal lengths by using normalization constraints. This method introduces a new way of camera self-calibration. Experiments of shape reconstruction and camera calibration are presented. We design a criterion called back projection compactness to quantify the calibration results. It measures the radius of the minimum sphere through which all back projection rays from the image positions of the same object point pass. We discuss the validity of this criterion and use it to compare the calibration results with other methods.

Keywords: structure from motion, calibration, computer vision

1 Introduction

The problem of recovering shape and motion from an image sequence has received a lot of attention. Previous approaches include recursive methods (e.g., [12]) and batch methods (e.g., [17] and [14]). The factorization method, first developed by Tomasi and Kanade [17], recovers the shape of the object and the motion of the camera from a sequence of images given tracking of many feature points. This method achieves its robustness and accuracy by applying the singular value decomposition (SVD) to a large number of images and feature points. However, it assumes the orthographic projection and known intrinsic parameters. Poelman and Kanade [14] presented a factorization method based on the weak perspective and paraperspective projection models assuming known intrinsic parameters. They also used the result from the paraperspective method as an initial value of non-linear optimization process to recover Euclidean shape and motion under a perspective camera model.

In this paper we first describe a perspective factorization method for Euclidean reconstruction. It iteratively recovers shape and motion by weak perspective factorization method and converges to a perspective model. Compared with Poelman and Kanade's method, we do not apply non-linear optimization which is computation intensive and converges very slowly if started far from the true solution. In our method we solve the reconstruction problem in a quasi linear way by taking advantage of the lower order projection approximation (like weak perspective) factorization methods. We also discuss the approach of solving the reversal shape ambiguity and analyze its convergence.

Secondly, we deal with the problem of recovering Euclidean shape and camera calibration simultaneously. Given tracking of feature points, our factorization-based method recovers the shape of the object, the motion of the camera (i.e., positions and orientations of multiple cameras), and focal lengths of cameras. We assume other intrinsic parameters are known or can be taken as generic values. The method first performs projective reconstruction by using iterative factorization, then converts the projective solution to the Euclidean one and generates the focal lengths by using normalization constraints. This method introduces a new way of camera self-calibration. Experiments of shape reconstruction and camera calibration are presented.

We design a criterion called back projection compactness to quantify the calibration results. It measures the radius of the minimum sphere through which all back projection rays from the image positions of the same object point pass. We discuss the validity of this criterion and use it to compare the calibration results with other methods.

2 Related Work

Tomasi and Kanade [17] developed a robust and efficient method to recover the shape of the object and the motion of the camera from a sequence of images, called the factorization method. Like most traditional methods, the factorization method requires that the image positions of point features first be tracked throughout the stream. This method processes the feature trajectory information using the singular value decomposition (SVD) to these feature points. However, the method's applicability is somewhat limited due to its use of an orthographic projection model. Poelman and Kanade [14] presented a factorization method based on the weak perspective and paraperspective projection models assuming known intrinsic parameters. They also used the result from the paraperspective method as an initial value of non-linear optimization process to recover Euclidean shape and motion under perspective camera models. The non-linear process is computation intensive and requires good initial values to converge.

Szeliski and Kang [16] used a non-linear least squares technique for perspective projection models. Their method can work on partial or uncertain feature tracks. They also initialized the alternative representation of focal length as scale factor and perspective distortion factor which improved stability and accuracy of recovery results.

Yu [21] presented a new approach based on a higher-order approximation of perspective projection by using Taylor expansion of depth. A method called "back-projection" is used to determine the Euclidean shape and motion instead of normalization constraints. This method approximates the perspective projection effects by higher order Taylor expansion which does not solve the projective depth literally. The accuracy of the approximation depends on the order of Taylor expansion and the computation increases exponentially as the order increases.

Christy and Horaud [1] [2] described a method for solving the Euclidean reconstruction problem with a perspective camera model by incrementally performing Euclidean reconstruction with either a weak or a paraperspective camera model. Given a sequence of images with a calibrated camera, this method converges in a few iterations, is computationally efficient. To deal with the reversal shape ambiguity problem, the method keeps two shapes (the shape and its mirror shape) to refine and postpones the decision between the two shapes at the end. At each iteration, weak or paraperspective method generates two shapes with sign ambiguity for every one of the two shapes being kept. The method checks the consistency of the two newly generated ambiguous shapes with the current shape being refined and chooses the more consistent one as the refined shape. In this way the method avoids the explosion of the number of solutions. The drawback is that it still keeps two lines of shapes to converge which doubles the computation cost.

There has been considerable progress on projective reconstruction in the last few years. One can start with uncalibrated cameras and unknown metric structure, initially recovering the scene up to an projective transformation [3] [13]. Triggs [18] viewed the projective reconstruction as a matter of recovering a coherent set of projective depths – projective scale factors that represent the depth information lost during image projection. It is a well-known fact that it is only possible to make reconstruction up to an unknown projective transformations when nothing about the intrinsic parameters, extrinsic parameters or the object is known. Thus it is necessary to have some additional information about either the intrinsic parameters, the extrinsic parameters or the object in order to obtain the desired Euclidean reconstruction.

Hartley recovered the Euclidean shape by a global optimization technique assuming the intrinsic parameters are constant [5]. In [8] Heyden showed that theoretically Euclidean reconstruction is possible even when the focal length and principal point are unknown and varying. The proof is based on the assumption of generic camera motion and known skew and aspect ratio. They developed a bundle adjustment algorithm to estimate all the unknown parameters, including focal lengths, principle points, camera motion and object shape. However, if the camera motion is not sufficiently general, then this is not possible. Pollefeys assumed the focal length as the only varying intrinsic parameter and presented a linear method to recover focal length [15]. Then they used the epipolar geometry to obtain a pair-wise image rectification to get the dense correspondence matches based on which the dense 3D model is generated. Maybank and Faugeras [11] gave a detailed discussion of the connection between the calibration of a single camera and the epipolar transformation obtained when the camera undergoes a displacement.

Most current reconstruction methods either work only for minimal number of views, or single out a few views for initialization to the multiple views.

To achieve robustness and accuracy, it is necessary to uniformly take account of all the data in all the images like in factorization methods. Triggs proposed a projective factorization method in [19] which recovered the projective depths by estimating a set of fundamental matrices and epipoles to chain all the images together. Based on the reconstruction of projective depths, a factorization is applied to the rescaled measurement matrix to generate shape and motion. Strictly speaking, the first step which recovers the projective depths is not a uniform batch approach.

Heyden [6] [7] presented methods of using subspace multilinear constraints to perform projective structure from motion. In [6] Heyden proposed an iterative algorithm based on SVD of the projective shape matrices. The algorithm is similar to our bilinear projective recovery method while ours performs SVD on the scaled measurement matrix which is more direct and simpler.

3 Perspective Factorization Method with Calibrated Cameras

Assuming the intrinsic parameters of the cameras are known, perspective factorization method reconstructs the Euclidean object shape and the camera motion from the feature points correspondences under a perspective projection model.

3.1 Algorithm description

3.1.1 Perspective projection

We denote by $\mathbf{s}_j = (x_j \; y_j \; z_j)$ a 3D point represented in the world coordinate system C_w whose origin is usually chosen at the gravity center of the object. The representations of points in the camera coordinate systems C_c are $(\mathbf{I}_i \cdot \mathbf{s}_j + t_{xi} \;\; \mathbf{J}_i \cdot \mathbf{s}_j + t_{yi} \;\; \mathbf{K}_i \cdot \mathbf{s}_j + t_{zi})$ where $(\mathbf{I}_i \; \mathbf{J}_i \; \mathbf{K}_i)$ are the rotations of the ith camera represented in C_w and $(t_{xi} \; t_{yi} \; t_{zi})$ are the translations.

Assuming the camera intrinsic parameters are known, the relationship between the object points and the image coordinates can be written as:

$$u_{ij} = \frac{\mathbf{I}_i \cdot \mathbf{s}_j + t_{xi}}{\mathbf{K}_i \cdot \mathbf{s}_j + t_{zi}}$$

$$v_{ij} = \frac{\mathbf{J}_i \cdot \mathbf{s}_j + t_{yi}}{\mathbf{K}_i \cdot \mathbf{s}_j + t_{zi}} \tag{1}$$

We divide both the numerator and the denominator of the above equations by t_{zi},

$$\begin{aligned} u_{ij} &= \frac{\frac{\mathbf{I}_i \cdot \mathbf{s}_j}{t_{zi}} + \frac{t_{xi}}{t_{zi}}}{1 + \epsilon_{ij}} \\ v_{ij} &= \frac{\frac{\mathbf{J}_i \cdot \mathbf{s}_j}{t_{zi}} + \frac{t_{yi}}{t_{zi}}}{1 + \epsilon_{ij}} \end{aligned} \tag{2}$$

where

$$\epsilon_{ij} = \frac{\mathbf{K}_i \cdot \mathbf{s}_j}{t_{zi}} \tag{3}$$

3.1.2 Weak perspective iterations

Given feature points correspondence matches, i.e., $(u_{ij}\ v_{ij})$, shape and motion reconstruction of perspective projection can be regarded as non-linear parameter fitting of equation (2) with camera motions and object points as parameters.

The numerators in equation (2) are weak perspective projections. Whenever the object is at some reasonable distance from the camera, the ϵ_{ij}'s are very small compared to 1. We start the parameter fitting by iterations of weak perspective approximations starting with $\epsilon_{ij} = 0$. Put the image coordinates in a matrix W called **measurement matrix**:

$$W = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ u_{n1} & u_{n2} & \cdots & u_{nm} \\ v_{11} & v_{12} & \cdots & v_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ v_{n1} & v_{n2} & \cdots & v_{nm} \end{bmatrix} \tag{4}$$

where n is the number of cameras and m is the number of feature points. Taking the denominators $1 + \epsilon_{ij}$ as scales of the measurement $(u_{ij}\ v_{ij})$, we are performing weak perspective factorization on a **scaled measurement**

matrix W_s to get motion and shape parameters:

$$W_s = \begin{bmatrix} \lambda_{11}u_{11} & \lambda_{12}u_{12} & \cdots & \lambda_{1m}u_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_{n1}u_{n1} & \lambda_{n2}u_{n2} & \cdots & \lambda_{nm}u_{nm} \\ \lambda_{11}v_{11} & \lambda_{12}v_{12} & \cdots & \lambda_{1m}v_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_{n1}v_{n1} & \lambda_{n2}v_{n2} & \cdots & \lambda_{nm}v_{nm} \end{bmatrix} \tag{5}$$

where $\lambda_{ij} = 1+\epsilon_{ij}$. The current motion parameters are denoted as $(\mathbf{I'}_i\, \mathbf{J'}_i\, \mathbf{K'}_i)$ and $(t'_{xi}\ t'_{yi}\ t'_{zi})$. The current points are denoted as $\mathbf{s'}_j = (x'_j\ y'_j\ z'_j)$. Then we use these current parameters to generate a new measurement matrix W':

$$W' = \begin{bmatrix} u'_{11} & u'_{12} & \cdots & u'_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ u'_{n1} & u'_{n2} & \cdots & u'_{nm} \\ v'_{11} & v'_{12} & \cdots & v'_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ v'_{n1} & v'_{n2} & \cdots & v'_{nm} \end{bmatrix} \tag{6}$$

where

$$\begin{aligned} u'_{ij} &= \frac{\mathbf{I'}_i \cdot \mathbf{s'}_j + t'_{xi}}{\mathbf{K'}_i \cdot \mathbf{s'}_j + t'_{zi}} \\ v'_{ij} &= \frac{\mathbf{J'}_i \cdot \mathbf{s'}_j + t'_{yi}}{\mathbf{K'}_i \cdot \mathbf{s'}_j + t'_{zi}} \end{aligned} \tag{7}$$

The process of computing the new measurement matrix is equivalent to the back projection process in many non-linear optimization methods. The new measurement matrix W' provides a criterion to choose between the two ambiguous shapes which are up to an mirror-symmetry transformation and its difference from the original measurement matrix W also gives the convergence error. Refined scales λ_{ij} are calculated from the current parameters and a new scaled measurement matrix is generated on which another iteration of weak perspective factorization is performed. The goal of parameter fitting is to iteratively find the scales which make the back projection consistent with the measurement.

3.1.3 Reconstruction algorithm

The perspective factorization method can be summarized by the following algorithm.

1. set $\epsilon_{ij} = 0$, for $\forall i$, $i \in \{1 \cdots n\}$ and $\forall j$, $j \in \{1 \cdots m\}$;
2. compute $\lambda_{ij} = 1 + \epsilon_{ij}$ and scaled W_s by equation (5);
3. perform weak perspective factorization method on W_s, generate a pair of motions and shapes which are mirror symmetric;
4. calculate two new measurement matrices with the sign reversal motions and shapes by equations (6) and (7);
5. check the difference between the new measurement matrix and the original W, take error E as the Frobenius norm of the difference matrix;
6. choose the set of parameters with smaller error as the refined motion and shape;
7. if the smaller error is close to zero, stop; else reset the values of ϵ_{ij} and go to step 2.

3.2 Algorithm analysis

3.2.1 Approximation by weak perspective projection

Weak perspective assumes that the object points lie in a plane parallel to the image plane passing through the origin of the world coordinate system. That is, weak perspective is a zero-order approximation [2] of perspective projection model:

$$\frac{1}{1 + \epsilon_{ij}} \approx 1 \tag{8}$$

In the first iteration, weak perspective factorization performs a zero-order approximation reconstruction. As the ϵ_{ij}'s are refined, iterations of weak perspective factorization figure out a consistent set of motion and shape parameters for the equations of (2).

3.2.2 Choice between mirror-symmetric shapes

It is well known that there is an inherent ambiguity problem with any affine reconstruction method, that is, after any affine reconstruction we can get two mirror-symmetric shapes and corresponding "mirror-symmetric" motions. As

$$\epsilon_{ij}^{l} = \frac{\mathbf{K}_i^l \cdot \mathbf{s}_j^l}{t_{zi}} \quad l = 1, 2 \tag{9}$$

and

$$\begin{array}{ccc} K_{xi}^1 = -K_{xi}^2 & K_{yi}^1 = -K_{yi}^2 & K_{zi}^1 = K_{zi}^2 \\ x_j^1 = x_j^2 & y_j^1 = y_j^2 & z_j^1 = -z_j^2 \end{array} \tag{10}$$

so

$$\epsilon_{ij}^1 = -\epsilon_{ij}^2 \tag{11}$$

For objects at reasonable distance from the camera, the weak perspective factorization method generates relatively correct shape without considering the perspective effects. In the two new measurement matrices computed by equations (6) and (7), perspective effects are taken care of by ϵ_{ij}'s. The ratio between the corresponding items of two W''s is $\frac{1+\epsilon_{ij}}{1-\epsilon_{ij}}$ which is large enough to distinguish the right shape with its mirror one. Based on this analysis, we keep only one set of the motion and shape parameters in each iteration which is computation efficient.

3.2.3 Error measurement

We use the Frobenius norm of the difference matrix of the selected new measurement matrix and the original one as error E during the iteration. Following theorem proves that the convergence of E guarantees the convergence of λ_{ij}'s.

Definition Matrix Error E is defined as the Frobenius norm of the difference matrix of the selected new measurement matrix W' and the original one W.

Theorem If the matrix error E converges, the λ_{ij}'s converge.

Proof. The convergence of E means that the difference matrix of W' and W converges to zero:

$$\begin{aligned} u_{ij} &= u'_{ij} = \frac{\frac{\mathbf{I'}_i \cdot \mathbf{s'}_j}{t'_{zi}} + \frac{t'_{xi}}{t'_{zi}}}{\frac{\mathbf{K'}_i \cdot \mathbf{s'}_j}{t'_{zi}} + 1} \\ v_{ij} &= v'_{ij} = \frac{\frac{\mathbf{J'}_i \cdot \mathbf{s'}_j}{t'_{zi}} + \frac{t'_{yi}}{t'_z i}}{\frac{\mathbf{K'}_i \cdot \mathbf{s'}_j}{t'_{zi}} + 1} \end{aligned} \tag{12}$$

Currently,

$$\lambda_{ij} = \frac{\mathbf{K'}_i \cdot \mathbf{s'}_j}{t'_{zi}} + 1 \tag{13}$$

Using the above λ_{ij}'s to get the scaled measurement matrix on which an iteration of weak perspective factorization is performed, from equation (12) it is obvious that this iteration generates the same motion and shape as the last iteration. Therefore, λ_{ij}'s stay constant according to equation (13), i.e., λ_{ij}'s converge.

4 Perspective Factorization Method with Unknown Focal Lengths

Assuming the cameras are semi-calibrated, i.e., the intrinsic parameters are known or taken as generic values (like $skew = 0$, principle point is in the middle of the image) except the focal lengths, the perspective factorization method reconstructs the Euclidean object shape, the camera motion and the focal lengths given the feature points correspondences under a perspective projection model. This method first recovers projective shape and motion by iterative factorization and projective depths refinement, then reconstructs the Euclidean shape, the camera motion and the focal lengths by normalization process.

4.1 Projective Reconstruction

Suppose there are n perspective cameras: P_i, $i = 1 \cdots n$ and m object points $\mathbf{x}_j$, $j = 1 \cdots m$ represented by homogeneous coordinates. The image coordinates are represented by $(u_{ij}\ v_{ij})$. Using the symbol $\sim$ to denote equality up to a scale, the following holds

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} \sim P_i \mathbf{x}_j \tag{14}$$

or

$$\lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = P_i \mathbf{x}_j \tag{15}$$

where λ_{ij} is a non-zero scale factor which is commonly called projective depth. The equivalent matrix form is:

$$W_s = \begin{bmatrix} \lambda_{11}\begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \cdots & \lambda_{1m}\begin{bmatrix} u_{1m} \\ v_{1m} \\ 1 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \lambda_{n1}\begin{bmatrix} u_{n1} \\ v_{n1} \\ 1 \end{bmatrix} & \cdots & \lambda_{nm}\begin{bmatrix} u_{nm} \\ v_{nm} \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} [\mathbf{x}_1 \ \cdots \ \mathbf{x}_m] \tag{16}$$

W_s is the **scaled measurement matrix**. Quite similar to the iterative algorithm described in the previous section, projective factorization method is summarized as:

1. set $\lambda_{ij} = 1$, for $\forall i$, $i \in \{1 \cdots n\}$ and $\forall j$, $j \in \{1 \cdots m\}$;
2. get the scaled measurement matrix W_s by equation (16);
3. perform rank4 factorization method on W_s, generate projective shape and motion;
4. reset the values of $\lambda_{ij} = P_i^{(3)}\mathbf{x}_j$ where $P_i^{(3)}$ denotes the third row of the projection matrix P_i;
5. if λ_{ij}'s are the same as the previous iteration, stop; else go to step 2.

The goal of the projective reconstruction process is to estimate the values of the projective depths (λ_{ij}'s) which make the equation (16) consistent. The reconstruction results are iteratively improved by back projecting the projective reconstruction of an iteration to refine the depth estimates. Triggs pointed out in [19] that the iteration turned out to be extremely stable even starting with arbitrary initial depths. In practice we use rough knowledge of the focal lengths and the perspective factorization method with calibrated cameras to get the initial values of λ_{ij}'s which drastically improve the convergence speed.

4.2 Normalization

The factorization of equation (16) is only determined up to a linear transformation $B_{4\times 4}$:

$$W_s = \hat{P}\hat{X} = \hat{P}BB^{-1}\hat{X} = PX \tag{17}$$

where $P = \hat{P}B$ and $X = B^{-1}\hat{X}$. $\hat{P}$ and $\hat{X}$ are referred to as the projective motion and the projective shape. Any non-singular 4×4 matrix B could be inserted between $\hat{P}$ and $\hat{X}$ to get another pair of motion and shape. With the assumption that the focal lengths are the only unknown intrinsic parameters, we have the projective motion matrix P_i,

$$P_i \sim K_i \left[R_i | \mathbf{t}_i\right] \tag{18}$$

where

$$K_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_i = \begin{bmatrix} \mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} \\ \mathbf{k}_i^{\mathrm{T}} \end{bmatrix} \quad \mathbf{t}_i = \begin{bmatrix} t_{xi} \\ t_{yi} \\ t_{zi} \end{bmatrix}$$

The upper triangular calibration matrix K_i encodes the intrinsic parameters of the ith camera: f_i represents the focal length, the principal point is $(0, 0)$ and the aspect ratio is 1. R_i is the ith rotation matrix with $\mathbf{i}_i$, $\mathbf{j}_i$ and $\mathbf{k}_i$ denoting the rotation axes. $\mathbf{t}_i$ is the ith translation vector. Combining Equation (18) for $i = 1 \cdots n$ into one matrix equation, we get,

$$P = [M|T] \tag{19}$$

where

$$\begin{aligned} M &= \left[\mathbf{m}_{x1}\ \mathbf{m}_{y1}\ \mathbf{m}_{z1}\ \cdots\ \mathbf{m}_{xn}\ \mathbf{m}_{yn}\ \mathbf{m}_{zn}\right]^{\mathrm{T}} \\ T &= \left[T_{x1}\ T_{y1}\ T_{z1}\ \cdots\ T_{xn}\ T_{yn}\ T_{zn}\right]^{\mathrm{T}} \end{aligned}$$

and

$$\begin{array}{lll} \mathbf{m}_{xi} = \mu_i f_i \mathbf{i}_i & \mathbf{m}_{yi} = \mu_i f_i \mathbf{j}_i & \mathbf{m}_{zi} = \mu_i \mathbf{k}_i \\ T_{xi} = \mu_i f_i t_{xi} & T_{yi} = \mu_i f_i t_{yi} & T_{zi} = \mu_i t_{zi} \end{array} \tag{20}$$

The shape matrix is represented by:

$$X \sim \begin{bmatrix} S \\ \mathbf{1} \end{bmatrix} \tag{21}$$

where

$$S = [\mathbf{s}_1 \;\; \mathbf{s}_2 \;\; \cdots \;\; \mathbf{s}_m]$$

and

$$\begin{aligned} \mathbf{s}_j &= [x_j \;\; y_j \;\; z_j]^{\mathrm{T}} \\ \mathbf{x}_j &= \left[\nu_j \mathbf{s}_j^{\mathrm{T}} \;\; \nu_j\right]^{\mathrm{T}} \end{aligned}$$

We put the origin of the world coordinate system at the center of gravity of the scaled object points to enforce

$$\sum_{j=1}^{m} \nu_j \mathbf{s}_j = 0 \tag{22}$$

We get,

$$\sum_{j=1}^{m} \lambda_{ij} u_{ij} = \sum_{j=1}^{m} (\mathbf{m}_{xi} \cdot \nu_j \mathbf{s}_j + \nu_j T_{xi}) = \mathbf{m}_{xi} \cdot \sum_{j=1}^{m} \nu_j \mathbf{s}_j + T_{xi} \sum_{j=1}^{m} \nu_j = T_{xi} \sum_{j=1}^{m} \nu_j \tag{23}$$

Similarly,

$$\sum_{j=1}^{m} \lambda_{ij} v_{ij} = T_{yi} \sum_{j=1}^{m} \nu_j \quad \sum_{j=1}^{m} \lambda_{ij} = T_{zi} \sum_{j=1}^{m} \nu_j \tag{24}$$

Define the 4×4 projective transformation H as:

$$H = [A|B] \tag{25}$$

where A is 4×3 and B is 4×1.
Since $P = \hat{P} H$,

$$[M|T] = \hat{P}\,[A|B] \tag{26}$$

we have,

$$T_{xi} = \hat{P}_{xi} B \quad T_{yi} = \hat{P}_{yi} B \quad T_{zi} = \hat{P}_{zi} B \tag{27}$$

From Equations (23) and (24) we know,

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} u_{ij}}{\sum_{j=1}^{m} \lambda_j} \quad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} v_{ij}}{\sum_{j=1}^{m} \lambda_j} \tag{28}$$

we set up $2n$ linear equations of the 4 unknown elements of the matrix B. Linear least squares solutions are then computed.

As $\mathbf{m}_{xi}$, $\mathbf{m}_{yi}$ and $\mathbf{m}_{zi}$ are scaled rotation axes, we get the following constraints from Equation (20):

$$\begin{aligned} |\mathbf{m}_{xi}|^2 &= |\mathbf{m}_{yi}|^2 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} &= 0 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} &= 0 \\ \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} &= 0 \end{aligned}$$

We can add one more constraint assuming $\mu_1 = 1$:

$$|\mathbf{m}_{z1}|^2 = 1 \tag{29}$$

The above constraints are linear constraints on MM^{T}. Since

$$MM^{\mathrm{T}} = \hat{P}AA^{\mathrm{T}}\hat{P}^{\mathrm{T}} \tag{30}$$

Totally we have $4n + 1$ linear equations of the 10 unknown elements of the symmetric 4×4 matrix $Q = AA^{\mathrm{T}}$. Least squares solutions are computed, we then get the matrix A from Q by rank3 matrix decomposition.

4.3 Reconstruction of shape, motion and focal lengths

Once the matrix A has been found, the projective transformation is $[A|B]$. The shape is computed as $X = H^{-1}\hat{X}$ and the motion matrix as $P = \hat{P}H$. We first compute the scales μ_i:

$$\mu_i = |\mathbf{m}_{zi}| \tag{31}$$

We then compute the focal lengths as

$$f_i = \frac{|\mathbf{m}_{xi}| + |\mathbf{m}_{yi}|}{2\mu_i} \tag{32}$$

Therefore, the motion parameters are

$$\begin{array}{lll} \mathbf{i}_i = \frac{\mathbf{m}_{xi}}{\mu_i f_i} & \mathbf{j}_i = \frac{\mathbf{m}_{yi}}{\mu_i f_i} & \mathbf{k}_i = \frac{\mathbf{m}_{zi}}{\mu_i} \\ t_{xi} = \frac{T_{xi}}{\mu_i f_i} & t_{yi} = \frac{T_{yi}}{\mu_i f_i} & t_{zi} = \frac{T_{zi}}{\mu_i} \end{array} \tag{33}$$

5 Applications

5.1 Scene Reconstruction

The perspective factorization method described in section 3 provides an efficient and robust way to recover the object shape and the camera motion under a perspective projection model. It takes care of the perspective effects by incremental reconstruction using the weak perspective factorization methods. The method assumes the intrinsic parameters are known and the feature points trajectories in image sequences are given. In this section we apply the perspective factorization method with calibrated cameras on indoor and outdoor scenes, analyze the results and compare with non-linear optimization approaches.

5.1.1 LED reconstruction

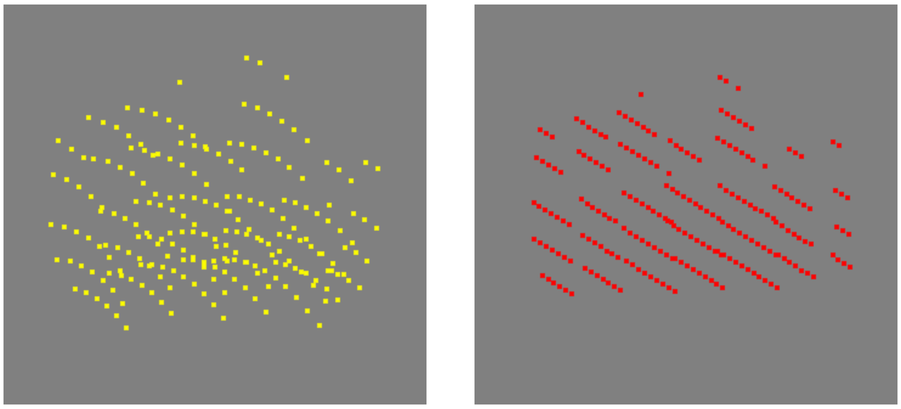
We take the data at the virtualized reality lab. The setup includes a bar of LEDs which moves around and works as object points (we have 232 feature points), and 51 cameras arranged in a dome above the LEDs. Tsai's approach [20] is used to calibrate intrinsic and extrinsic camera parameters. In this experiment we use intrinsic parameters calibrated by Tsai's method as known values. The perspective factorization method described in section 3 is applied on the feature points correspondences in 51 images.

- perspective effects

  Figure 1(a) shows the recovered object by the weak perspective factorization method. The distortions are obvious which are caused by the approximation of perspective projection with weak perspective projection. Figure 1(b) gives the recovered object by the perspective factorization method which takes care of the perspective effects by iterative weak perspective reconstruction.

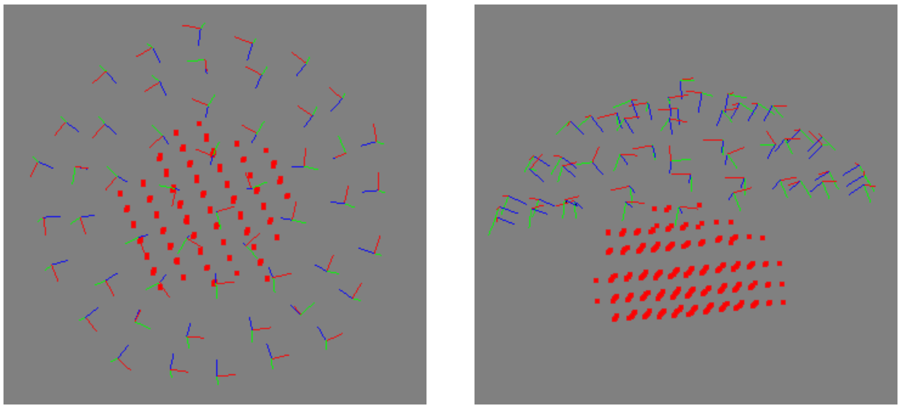
- reconstruction result

  The reconstruction results include the object shape (LEDs positions) and the camera extrinsic parameters which are the camera orientations and locations. The result is shown in figure 2. Figure 2(a) is the top view and (b) is the side view. Each camera is represented by its three axes where red, green and blue denoting x, y and z axis respectively. The intersections of the axes are the locations of the cameras.



(a) (b)

Figure 1: (a) Weak perspective (b) perspective reconstruction of LED positions.



(a) (b)

Figure 2: (a) Top view (b) side view of the LED reconstruction result. Red points denote the recovered LED positions. Red, green and blue lines denote the recovered x, y and z axes of the cameras.

Figure 3(a) and 3(b) are the comparisons of our reconstruction results with "known" LED positions and Tsai's extrinsic calibration results. As the perspective factorization recovers shape and motion up to a similarity transformation, it is necessary to find the transformation in order to compare the reconstruction results. There are two ways of finding the similarity transformation. One way is to set the scale and the orientation in the weak perspective factorization by giving the orientation and the position of any one of the cameras. Second way to handle this is to use Horn's absolute method [9] to compute the orientation between the recovered shape and the known shape since the correspondences are known between the two shapes.

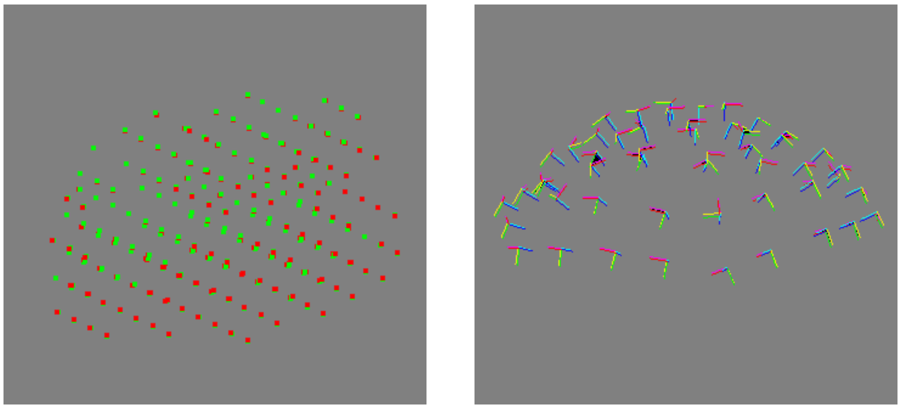
Figure 3 shows the comparison results after rotating and scaling the recovered shape. Figure 3(a) shows the differences of the recovered object points from the "known" positions. The recovered points are shown in red while the "known" ones shown in green. The maximal distance between the two sets of points is $8mm$ which is about 0.25 percent of the object size. The errors are partly due to the inaccurate "known" points positions. Figure 3(b) demonstrates the differences of the recovered camera orientations and the locations from the results generated by Tsai's extrinsic calibration method. The red, green and blue lines are the recovered camera axes and the pink, yellow and cyan ones the Tsai's results. The maximal distance between the two sets of camera locations is $21mm$ which is about 0.7 percent of the object size. The errors exist partly because of the calibration errors both in intrinsic and extrinsic parameters. The perspective factorization method converges in 8 iterations in this experiment.

5.1.2 Building reconstruction

We apply the perspective factorization method to campus building reconstruction. The images are taken by a hand-held camera whose intrinsic parameters are pre-calibrated. Figure 4 are the three images we use to recover the building. Feature points are manually selected, which are overlaid in the images . 34 feature points are used.

- reconstruction result

  Figure 5 shows the reconstruction result including the feature point positions, and the camera locations and orientatiaons. Figure 6 (a)

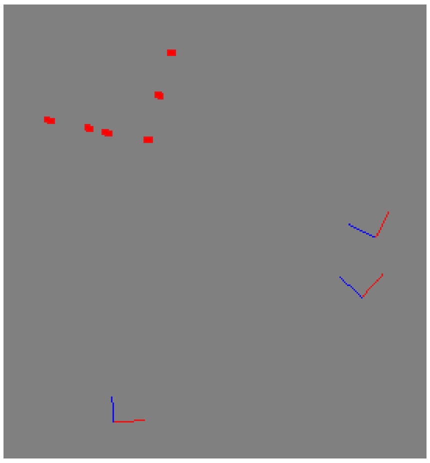


(a) (b)

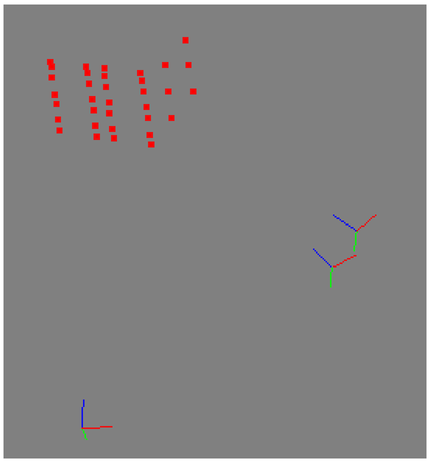
Figure 3: Comparison of (a) shape (b) motion of LED reconstruction. In (a) red dots denote the recovered LED positions by the perspective factorization method and green dots denote the "known" positions. In (b) red, green and blue lines represent the recovered x, y and z axes of the cameras by the perspective factorization method and pink, yellow and cyan lines represent the "known" x, y and z axes of the cameras.

Figure 4: Images for building reconstruction. Manually selected feature points are overlaid in the images.

is the top view of the building and (b) is the side view. It converges within 10 iterations and uses 0.69 seconds CPU time.



(a)



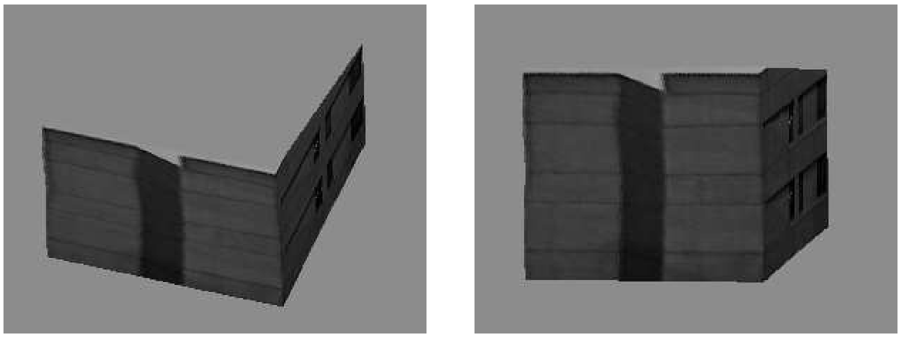
(b)

Figure 5: (a) Top view (b) side view of the building reconstruction result. Red points denote the recovered feature point positions. Red, green and blue lines denote the recovered x, y and z axes of the cameras.

- comparison with non-linear optimization

  We compare our results with non-linear optimization method. The non-linear method we use is starting with weak perspective factorization results as initial values then using bundle adjustment to refine the shape and motion. The method deals with the reversal shape ambiguity by comparing the optimization errors after two sweeps of non-linear optimizations on both shapes. The final result is the one with the smaller error. This brute force way at least doubles the computation cost. The non-linear method takes 18 steps to converge and the CPU time it uses is 38.83 seconds.

  Figure 7 shows the comparison results after putting the two recovered shapes together by a similarity transformation. Figure 7(a) shows the differences of the recovered feature points by the perspective factorization method and by the non-linear method. The perspective recovered points are shown in red while the non-linear recovered ones shown in green. The maximal distance between the two sets of points is about



(a) (b)

Figure 6: (a) Top view (b) side view of the reconstructed building with texture mapping.

1.52 percent of the size of the recovered partial building. Figure 7(b) shows the differences of the perspective recovered camera orientations and locations from the results generated by the non-linear method. The red, green and blue lines are the perspective recovered camera axes and the pink, yellow and cyan ones the non-linear results. The maximal distance between the two sets of camera locations is about 5.36 percent of the size of the recovered partial building. The results of the two methods are comparable.

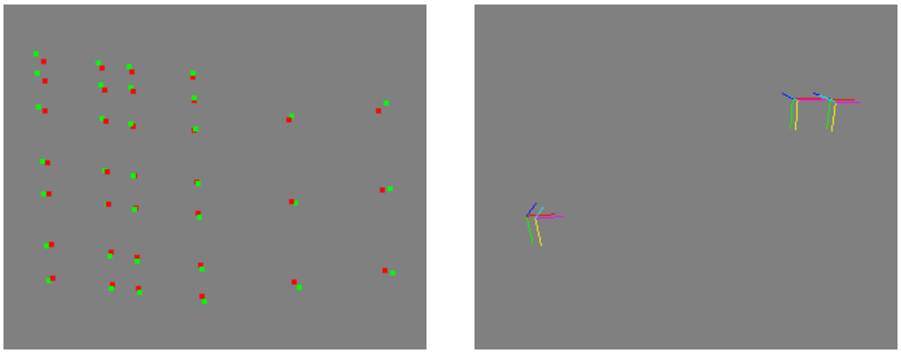
5.1.3 Terrain reconstruction

An aerial image sequence was taken from an airplane flying over the Grand Canyon area. The plane changed its altitude as well as the roll, pitch and yaw angles during the sequence. The intrinsic parameters of the camera are pre-calibrated. The sequence consists of 97 images and 86 feature points are tracked through the sequence. Several images from the sequence are shown in Figure 8.

- reconstruction result

  Figure 9 shows the reconstruction result including the feature point positions, and the camera orientations and the locations. Figure 10 (a) is the top view of the Grand Canyon terrain map and (b) is the side view. It converges within 18 steps and uses 41.13 seconds CPU time.

- comparison with non-linear optimization

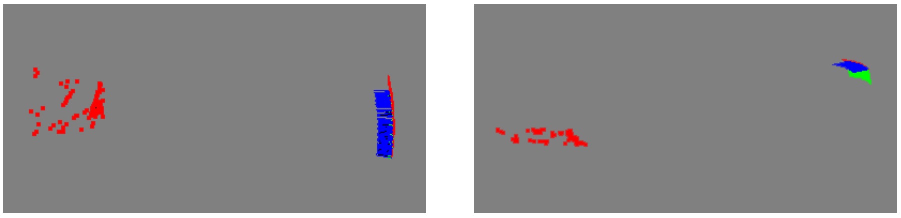


(a) (b)

Figure 7: Comparison of (a) shape (b) motion of building reconstruction. In (a) red dots denote the recovered feature points positions by the perspective factorization method and green dots denote the positions recovered by the non-linear method. In (b) red, green and blue lines represent the recovered x, y and z axes of the cameras by the perspective factorization method and pink, yellow and cyan lines represent the recovered x, y and z axes of the cameras by the non-linear method.

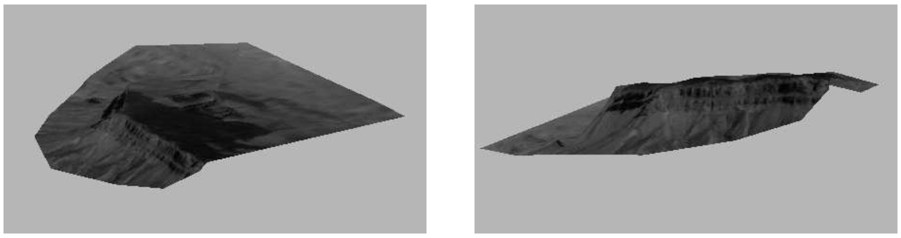
(a) (b) (c)

Figure 8: (a) 1st (b) 46th (c) 91st image of the Grand Canyon sequence.



(a) (b)

Figure 9: (a) Top view (b) side view of the Grand Canyon reconstruction result. Red points denote the recovered feature point positions. Red, green and blue lines denote the recovered x, y and z axes of the cameras.

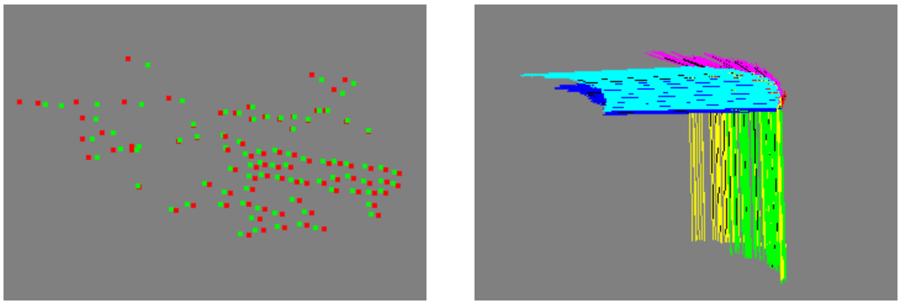


(a) (b)

Figure 10: (a) Top view (b) side view of the reconstructed Grand Canyon with texture mapping.

We compare our results with the non-linear optimization method as well. The non-linear method is described in section 5.1.2. It takes 38 steps to converge and the CPU time it uses is 9264 seconds.

Figure 11 shows the comparison results after putting the two recovered shapes together by a similarity transformation. Figure 11(a) shows the differences of the recovered feature points by the perspective factorization method and by the non-linear method. The perspective recovered points are shown in red while the non-linear recovered ones shown in green. The maximal distance between the two sets of points is about 5.16 percent of the terrain size of the recovered part. Figure 11(b) shows the differences of the perspective recovered camera orientations and the locations from the results generated by the non-linear method. The red, green and blue lines are the perspective recovered camera axes and the pink, yellow and cyan ones the non-linear results. The maximal distance between the two sets of camera locations is about 9.12 percent of the terrain size of the recovered part.



(a) (b)

Figure 11: Comparison of (a) shape (b) motion of Grand Canyon reconstruction. In (a) red dots denote the recovered feature points positions by the perspective factorization method and green dots denote the positions recovered by the non-linear method. In (b) red, green and blue lines represent the recovered x, y and z axes of the cameras by the perspective factorization method and pink, yellow and cyan lines represent the recovered x, y and z axes of the cameras by the non-linear method.

5.2 Self-Calibration

In this section we apply the perspective factorization method with unknown focal lengths to perform camera self-calibration. Given feature correspondences from multiple views, the perspective factorization method recovers feature points positions, camera locations and orientations as well as camera focal lengths. We introduce a measurement called back projection compactness to compare the calibration results.

5.2.1 Back Projection Compactness

- Definition

  Definition Back projection compactness is the radius of the minimum sphere through which all back projection rays from the image positions of the same object point pass.

  Back projection compactness measures quantitatively how well the back projection rays from multiple views converge. The smaller the back projection compactness is, the better the convergence is.

- Analysis

  We design the concept of back projection compactness to measure the calibration results quantitatively. Calibration is to find a set of parameters to transfer the object point position to its image coordinates. How to quantify the calibration results depends on the calibration applications. For applications including structure from motion, image based rendering, and augmented reality, measuring the compactness of the back projection rays provides a value to quantify how consistent the camera calibrations are.

  In our experiments we compare our self-calibration results with the results of the Tsai's method [20]. For Tsai's method, the object point positions in 3D are known and the corresponding image coordinates are given. The Tsai's method outputs the camera locations and orientations as well as the camera intrinsic parameters. Calibration error is caused by the pinhole projection assumption, the inaccurate 3D point positions and the noises of the image measurements. For the perspective method, only the feature correspondences among the multiple views are given. It outputs the camera locations and orientations, the object point positions and the camera focal lengths. The perspective method assumes the other intrinsic camera parameters except the focal lengths are generic. Comparing with the Tsai's method, it has the same error sources from the pinhole camera assumption and the image measurement noises. However, it avoids the inaccuracy of 3D point positions which are taken as hard constraints in the Tsai's calibration method.

The goal of the Tsai's method is to make each back projection ray of the same point to pass the known 3D position no matter whether it is accurate or not. The perspective factorization method is to make the back projection rays of the same point to converge to one position which is the recovered object point position. It takes far less constraints than the Tsai's method and distributes the reconstruction error to the calibration results and the recovered shape.

- Algorithm

  1. start with a cubic space and divide it into 8 cubes of equal size;
  2. for each small cube, compute the distance $D(C_i, L_j)$ of its center C_i to every back projection ray L_j, where $i = 1 \cdots 8$ and $j = 1 \cdots m$, m is the number of the cameras;
  3. take $D_i = \max_j D(C_i, L_j)$;
  4. choose the cube with the smallest D_i, start with this cube and divide it into 8 cubes of equal size;
  5. if the size of the small cube is close to zero, stop, set C_i as the center of the sphere and D_i as the back projection compactness; else go to step 2.

5.2.2 Experiments

We use the data from the virtualized reality lab for the self-calibration experiments. The setup includes a bar of LEDs which moves around and works as object points, and cameras above the LEDs. Taking the camera intrinsic parameters except the focal lengths as generic (like $skew = 0$, $aspect = 1$, the principle point is in the middle of the image), we apply the perspective factorization method with unknown focal lengths to the feature points correspondences. The outputs of our method include the camera focal lengths,

	Tsai	*Persp1*	*Persp2*		
			Init1	*Init2*	*Init3*
max b.p.c.	13.4421	14.3017	15.3703	15.3543	15.5826
mean b.p.c.	7.0366	7.1496	8.2491	8.1238	8.3107
median b.p.c.	7.1092	7.4294	7.7700	7.6798	7.8294
max shapeD	——	7.9137	7.4042	7.3847	7.3730
max sphereD1	——	8.1359	8.6033	7.7878	7.3842
max sphereD2	8.3386	8.4666	8.8477	9.1947	10.0894

Table 1: calibration results

the camera locations and orientations and the object point positions. To compare the calibration results, Tsai's approach [20] is used to calibrate the intrinsic and extrinsic camera parameters. The back projection compactness is calculated from both of the calibration results to quantify the calibration quality.

In this experiment 223 feature points are used for both of the calibration methods. The image number is 51. Table 1 shows the comparison results. *Tsai* denotes the Tsai's calibration method. *Persp1* and *Persp2* represent the perspective factorization method with calibrated cameras and with unknown focal lengths respectively. *Init1* and *Init2* indicate that the perspective factorization method start with the mean value and the median value of the "known" focal lengths from the Tsai's method as initial values. *Init3* indicate the initial focal lengths are any random numbers within the range (in this example the range is 365 to 385). It shows that the three results are very close which means that the rough knowledge of the focal lengths is enough for the perspective calibration method to converge.

The 5 values we compare are the maximal back projection compactness of 223 object points (denoted as *max b.p.c.* in Table 1), the mean and the median values of the back projection compactnesses (denoted as *mean b.p.c.* and *median b.p.c.* respectively), the maximal distance of the recovered object points and the "known" object point positions (denoted as *max shapeD*), the maximal distance of the recovered object points and the back projection compactness centers (denoted as *max shapeD1*) and the maximal distance of the "known" object point positions and the back projection compactness centers (denoted as *max shapeD2*). The unit representing the distances is *mm*.

6 Discussion

In this paper we first describe a perspective factorization method for Euclidean reconstruction. It iteratively recovers shape and motion by the weak perspective factorization method and converges to a perspective model. We solve the reconstruction problem in a quasi linear way by taking advantage of the factorization methods of the lower order projection. Compared with non-linear methods, our method is efficient and robust. We also present a new way of dealing with the sign ambiguity problem. However, the perspective factorization method is conceptually non-linear parameter fitting process. Common problems, such as local minima, still exist. We are working on theoretical analysis of its convergence.

We successfully recover the shape of the object and the camera calibrations simultaneously given tracking of feature points. The factorization-based method first performs projective reconstruction by using iterative factorization, then converts the projective solution to the Euclidean one and generates the focal lengths by using normalization constraints. This method introduces a new way of camera self-calibration which has various applications in autonomous navigation, virtual reality systems and video editing. The projective factorization method requires the rough range of focal lengths to generate an estimate of the projective depths for fast convergence. Accuracy of the calibrations and their effects on applications provide further research topics.

Freeman [4] described various bilinear model fitting problems in computer vision. Koenderink [10] also analyzed the bilinear algebra applied to camera calibration problem. They both indicated that the self-calibration of perspective camera is a "hard" problem. We are going to focus on application oriented bilinear algebra analysis.

We also design a criterion called back projection compactness to quantify the calibration results. It measures the radius of the minimum sphere through which all back projection rays of the same object point pass.We use it to compare the calibration results with other methods. The divide and conquer algorithm to compute the back projection compactness is efficient. However, it is still an open problem to prove the algorithm.

References

[1] S. Christy and R. Horaud. Euclidean reconstruction: From paraperspective to perspective. In *ECCV96*, pages II:129–140, 1996.

[2] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *PAMI*, 18(11):1098–1104, November 1996.

[3] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV92*, pages 563–578, 1992.

[4] W.T. Freeman and J.B. Tenenbaum. Learning bilinear models for two factor problems in vision. In *CVPR97*, pages 554–560, 1997.

[5] R.I. Hartley. Euclidean reconstruction from uncalibrated views. In *CVPR94*, pages 908–912, 1994.

[6] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *SCIA97*, 1997.

[7] A. Heyden. Reduced multilinear constraints: Theory and experiments. *IJCV*, 30(1):5–26, October 1998.

[8] A. Heyden and K. Astrom. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *CVPR97*, pages 438–443, 1997.

[9] B.K.P. Horn. Closed form solutions of absolute orientation using unit quaternions. *JOSA-A*, 4(4):629–642, April 1987.

[10] J.J. Koenderink and A.J. vanDoorn. The generic bilinear calibration-estimation problem. *IJCV*, 23(3):217–234, 1997.

[11] S. Maybank and O.D. Faugeras. A theory of self-calibration of a moving camera. *IJCV*, 8(2):123–151, August 1992.

[12] P.F. McLauchlan, I.D. Reid, and D.W. Murray. Recursive affine structure and motion from image sequences. In *ECCV94*, volume 1, pages 217–224, 1994.

[13] R. Mohr, L. Quan, and F. Veillon. Relative 3d reconstruction using multiple uncalibrated images. *IJRR*, 14(6):619–632, December 1995.

[14] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *PAMI*, 19(3):206–218, 1997.

[15] M. Pollefeys, L. Van Gool, and A. Oosterlinck. Euclidean reconstruction from image sequences with variable focal length. In *ECCV96*, pages 31–44, 1996.

[16] R. Szeliski and S.B. Kang. Recovering 3d shape and motion from image streams using non-linear least squares. Technical Report CRL 93/3, Digital Equipment Corporation, Cambridge Research Lab, 1993.

[17] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137–154, 1992.

[18] B. Triggs. Matching constraints and the joint image. In *ICCV95*, pages 338–343, 1995.

[19] B. Triggs. Factorization methods for projective structure and motion. In *CVPR96*, pages 845–851, 1996.

[20] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *RA*, 3(4):323–344, 1987.

[21] H. Yu, Q. Chen, G. Xu, and M. Yachida. 3d shape and motion by svd under higher-order approximation of perspective projection. In *ICPR96*, page A80.22, 1996.