Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Vision based terrain reconstruction for planet rover using a special binocular bundle adjustment*

Min-yi SHEN[1,2], Zhi-yu XIANG[†‡1,2], Ji-lin LIU[1,2]

([1]*Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China*)

([2]*Zhejiang Provincial Key Laboratory of Information Network Technology, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: xiangzy@zju.edu.cn

Received Nov. 14, 2007; revision accepted Feb. 18, 2008

Abstract: This paper presents a pure vision based technique for 3D reconstruction of planet terrain. The reconstruction accuracy depends ultimately on an optimization technique known as 'bundle adjustment'. In vision techniques, the translation is only known up to a scale factor, and a single scale factor is assumed for the whole sequence of images if only one camera is used. If an extra camera is available, stereo vision based reconstruction can be obtained by binocular views. If the baseline of the stereo setup is known, the scale factor problem is solved. We found that direct application of classical bundle adjustment on the constraints inherent between the binocular views has not been tested. Our method incorporated this constraint into the conventional bundle adjustment method. This special binocular bundle adjustment has been performed on image sequences similar to planet terrain circumstances. Experimental results show that our special method enhances not only the localization accuracy, but also the terrain mapping quality.

Key words: 3D reconstruction, Binocular bundle adjustment (BBA), Scale-invariant feature transform (SIFT), Re-projection error, RANSAC

doi:10.1631/jzus.A0720057 **Document code:** A **CLC number:** TP317.4; TP391

INTRODUCTION

In planetary exploration missions, high-precision landing-site topographic information is crucial for engineering operations and the achievement of scientific goals. In particular, large-scale landing-site mapping will be extremely important for current and future landing missions such as the Mars Exploration Rovers (Di *et al*., 2004). Several systems have been proposed (Li *et al*., 2002; Di *et al*., 2005; Labrie and Hebert, 2007) for the Mars Exploration Rover Mission. In their implementation, the initial location and heading information of each rover was provided by the telemetry data acquired by onboard sensors. The onboard navigation system consists mainly of an IMU, an odometer, and some solar imaging cameras. Local rover localizations on the landing site were based on the onboard navigation sensors. The rover automatically estimates its position using wheel odometry and IMU data. A visual odometry experiment will improve localization accuracy by overcoming problems associated with wheel odometry such as slippage and low accuracy. Finally bundle adjustment (BA) was applied and constraints were imposed on the projection model or movement of the camera between images, resulting in high precision landing site topographic mapping products.

BA was originally used in photogrammetry (Wolf and DeWitt, 2000). It is basically a steepest-descent algorithm that searches for an optimal model by minimizing the error between the observed 2D feature points and the re-projected feature points from the reconstructed model. Most of the 3D reconstruction systems available today use a turntable to rotate an object to capture its images from different

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 60505017 and 60534070) and the Science Planning Project of Zhejiang Province, China (No. 2005C14008)

view points (Wong and Chang, 2004; Labrie and Hebert, 2007). For these systems, the object is constrained to rotate around a fixed axis. By incorporating this constraint into the traditional BA method, a more accurate reconstruction model can be obtained.

This paper presents a pure vision based terrain reconstruction for a planet rover using SIFT (scale-invariant feature transform) features and a special binocular bundle adjustment (BBA). In contrast to the work in (Li *et al*., 2004), which still needs manual help for feature matching, a SIFT feature has the characteristics of scale and rotation invariant and it guarantees a full automatic match between the projected features produced by the same 3D point. We initialized the localization process using stereo vision on some stereo images to obtain the structure and motion without INS (inertial navigation system) data. As shown in Fig.1, after merging local 3D structure models into a common coordinate system, a final BBA can be applied to the whole sequence. This paper aims to study how reconstruction accuracy could be improved by employing constraints inherent between the binocular views in an efficient way.

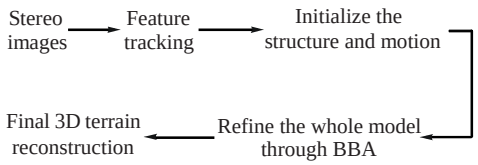


Fig.1 Steps performed for the proposed binocular bundle adjustment method

The remainder of this paper is structured as follows. First, we briefly describe the detection and tracking of features in Section 2. Then we show how to initialize the 3D model reconstruction in Section 3. In Section 4 we present the special BBA based pose estimation, using both photometric and geometric constraints. Section 5 shows how to merge all local 3D models to obtain a final 3D model and describes the experimental results of this work. Finally, we conclude the paper in Section 6.

FEATURE TRACKING

We use the SIFT features. This approach has been named the 'scale-invariant feature transform' (Lowe, 2004) as it transforms image data into scale-invariant coordinates relative to local features. The SIFT is invariant to translation, scaling (Lowe, 1999), and rotation. It is also partially invariant to illumination variations as well as affine for 3D projection. These features locate interest points at maxima/minima of a difference of Gaussian function in scale space. Each interest point has an associated orientation, which is the peak of a histogram of local orientations. The resulting feature descriptor, which captures the orientation information of the local image region, contains 128 elements. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features. All the feature points can be automatically selected.

The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. We perform feature space outlier rejection to remove incorrect matches. It has been found that comparing the distance of a potential match to the distance of the best incorrect match is an effective strategy for outlier rejection (Brown *et al*., 2005). Only the verified candidates are accepted as inliers.

Suppose that the match distance of the second closest neighbor is d_{sc}, as it is the best matching outlier. In order to verify a match, we compare the match distance of a potentially correct match d_c to the outlier distance, accepting the match if

$$d_c < r_d \cdot d_{sc}. \tag{1}$$

Typically the distance ratio r_d is set as 0.8.

Fig.2 shows an example of matched SIFT features in frames.

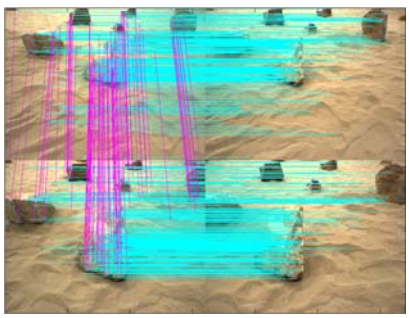


Fig.2 Matching lines of tracked SIFT features. Horizontal lines are intra-frame matches and vertical lines are inter-frame matches

INITIALIZATION

Motion estimation and outlier rejection

Three-dimensional geometric reconstruction is a process to recover a 3D scene or object from multiple images by simultaneously recovering the 3D structure of a scene or object and the camera motion (rotation and translation) associated with the images. In our work, initial estimate of the pose is determined with SVD (singular value decomposition) and quaternion based representation of the camera pose. The initial refinement of the pose estimate is achieved using RANSAC.

The method first eliminates the translation component by centering the data around the mean values and next estimates the rotation matrix $\hat{\boldsymbol{R}}$. When $\hat{\boldsymbol{R}}$ is determined, the translation $\hat{\boldsymbol{t}}$ is calculated.

Assume that we have at our disposal n noise-free and matched 3D measurements $U=\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ and $V=\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$. These ideal values satisfy the rigid motion constraint $\boldsymbol{v}_i=\boldsymbol{R}\boldsymbol{u}_i+\boldsymbol{t}$, where $\boldsymbol{R}$ is a 3×3 rotation matrix and $\boldsymbol{t}$ is the translation vector (Fig.3). The rotation can be parameterized by quaternion $\boldsymbol{q}=[q_0\ q_1\ q_2\ q_3]^{\mathrm{T}}$, which is a 4D unit vector. The rotation can then be estimated on the basis of the eigenvector corresponding to the smallest eigenvalue of the cross-correlation matrix (Kwolek, 2007).

Finally, the estimate of translation $\hat{\boldsymbol{t}}$ is computed in the following manner (Umeyama, 1991):

$$\hat{\boldsymbol{t}} = \tilde{\boldsymbol{v}} - \hat{\boldsymbol{R}}\tilde{\boldsymbol{u}},$$

where $\tilde{\boldsymbol{u}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}_i$, $\tilde{\boldsymbol{v}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i$.

Those matched SIFT features which were used to obtain the structure and motions in Section 2 have not been ready for the re-projection optimization. The correspondence algorithm we utilize in pose estimation can sporadically generate mismatches. The position of the structure points might not totally accord with the motions we have obtained. The RANSAC algorithm (Fischler and Bolles, 1981) is employed. It starts with as little data as possible to fit a model and increases the subset of points during the operation. All points that are consistent with the model are called 'inliers', whereas the non-consistent points are discarded.

Given a couple of image points (M_l, M_r) based on the SIFT features, a 3D point X may be accepted if it satisfies

$$|\, p_\mathrm{l}(X) - M_\mathrm{l} \,| < \tau_\mathrm{l},\ \ |\, p_\mathrm{r}(X) - M_\mathrm{r} \,| < \tau_\mathrm{r}, \qquad (2)$$

where τ_l and τ_r are the user-defined thresholds, p_l and p_r are the re-projection functions from 3D points onto the left-hand and right-hand images, respectively. To further reduce the pose error we employ the special BBA on resulting inlier points of several consecutive frames.

Sequence processing

During the construction of the 3D models, the lists of observed features are linked through consecutive pairs of images. A dynamic array is used as shown in Fig.4. It evolves as new pairs of images are coming in. The array is first initialized with matches found in the first pair. It is then updated for any new input pair of images by comparing new matches with those related to the nearest start pair of images in the array. An image match is declared if the number of RANSAC inliers $n_\mathrm{in} > \lambda n_\mathrm{m}$, where λ is a threshold (we use λ=0.9), and n_m is the total number of matches to be initially refined at a time. If the new pair is not

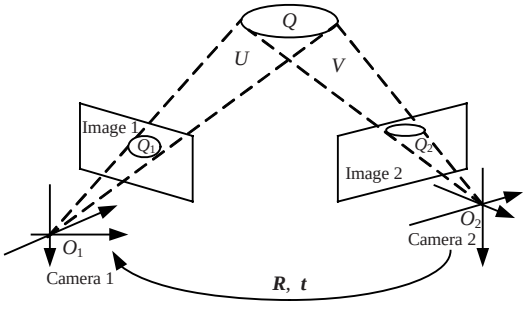


Fig.3 Rotation and translation between two cameras

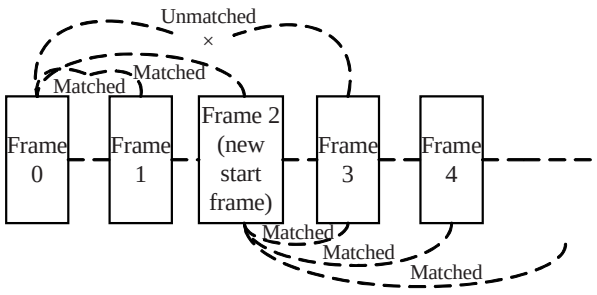


Fig.4 Matching process of initial estimation

matched to the last start frame, the frame ahead of the new one is declared as the new start frame.

Pairs of images to be processed should share a common subset of 3D points (60%~90%). When inter frame overlap is small, false matches between features can lead to a breakdown of the whole link (Olson *et al*., 2003). A good feature point in an image represents the projection of a unique physical point. If a physical point has been captured by more than one image, its projections in the images must match. The set of all those correspondences established between images will ultimately lead to the 3D positions of the physical points as well as to the poses of the camera.

The position and orientation of each new pair of images are estimated as soon as the pair comes in. It is also possible to optimize the 3D model using BBA after each new image's arrival or at the end of the entire sequence.

SPECIAL BINOCULAR BUNDLE ADJUSTMENT

Three-dimensional reconstruction is usually achieved in two stages (Reitinger *et al*., 2007). The first stage creates an approximate model, which we have obtained in Section 3. The second stage refines the model using an optimization technique known as 'bundle adjustment'. The reconstruction accuracy depends critically on the second stage. BA (Triggs *et al*., 1999) is a non-linear optimization problem solved through iterative non-linear least squares methods. It performs simultaneous optimization of the 3D point and camera placements by minimizing the squared error between estimated and measured image feature locations (Shum *et al*., 1999). It can be utilized at a refining stage of estimation and the algorithm requires a good initial estimate of the pose.

The classical BA method described above does not make full use of the constraints available, which come from the fact that stereo-vision-based reconstruction is obtained by binocular views. The optimized structure and motion of the left-hand images might not be well re-projected on the right-hand ones during stereo-vision-based reconstruction. This constraint, if estimated reliably, could be used to further constrain the adjustment as well, much like the process of stereo matching (Fig.5). Usually, the motions of the left-hand camera and the right-hand one could be optimized separately with the constraint between them (Di *et al*., 2004). But considering there is a fixed relative rotation between the stereo cameras, the motions of the right-hand camera follow the left-hand one. We will refine only the left-hand motions for simplification and optimize re-projection errors (REs) in both views to ensure the precision.

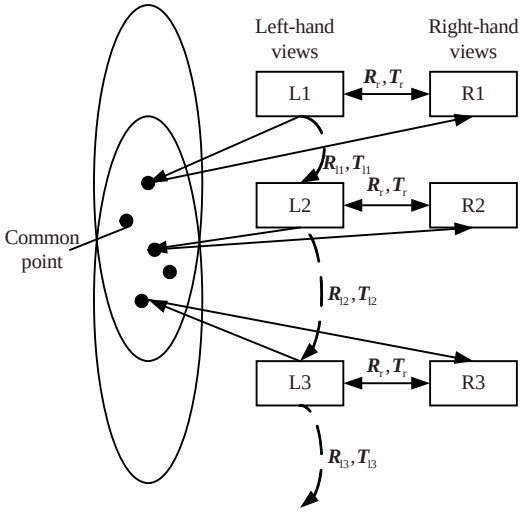


Fig.5 Motion relations in the special binocular bundle adjustment system. The constraint between the left-hand view and the right-hand view together with the 3D space points constructs a 'triangle', which may intensify the refinement. R_r and T_r stand for the rotation and translation of the right-hand views to the left-hand ones, respectively; R_{li} and T_{li} stand for the rotation and translation between the left-hand views, respectively

Motion parameters and a 3D structure could be obtained through a pure vision based method. However, the accuracy of space point correspondences is very sensitive to the disparity distance in each image pair. We choose to implement the constraint optimization with a disparity based weight multiplier. According to stereo vision (Matthies and Shafer, 1987), the depth distance between the 3D points and the camera center $z_i=bf/d_i$ (b is the baseline, f is the focal length, and d_i is the disparity of the ith image pair) is correlated to the disparity. The resolution of this distance is $\Delta z=bf\Delta d/d^2$, where Δz defines the resolution of the depth distance z and Δd represents the resolution of the disparity. Higher resolution of the depth distance could be obtained with larger disparity. It means that for the same deviation of disparity, points with larger disparity will have less influence on the resolution of the distance pose information, as well as on the resolution of the 2D re-projection of those 3D

points. Thus, in our method, measurements with larger disparity will have greater weight than those with smaller disparity pairs.

The error function (Eaton, 2005) is the sum squared error between the projected 3D point and the measured feature position. Therefore, the constraints are included in the original cost function to improve the robustness and accuracy of the initial results as follows:

$$e=\sum_{i\in M}\sum_{j\in\chi(i)}\rho_i[(f(\boldsymbol{r}_{lij}))^2+(f(\boldsymbol{r}_{rij}))^2], \tag{3}$$

where M is the set of all images, $\chi(i)$ is the set of 3D points projecting to image i, and the multiplier ρ_i is a weighting parameter in the cost function, which can be chosen based upon the dynamic ranges of the disparity d_{ij} of the ith point in the jth pair of images:

$$\rho_i=\left|d_{ij}/\overline{d}\right|,\quad \overline{d}=\frac{1}{N}\sum_{i\in M}d_{ij},$$

where N is the number of pairs. The robust function $f(x)$ in the error function Eq.(3) denotes the Euclidean distance (Lourakis and Argyros, 2004). $\boldsymbol{r}_{lij}$ and $\boldsymbol{r}_{rij}$ are the difference between the measured feature position and the projected 3D point in left-hand views and right-hand views, respectively:

$$\begin{cases}\boldsymbol{r}_{lij}=\boldsymbol{m}_{lij}-\tilde{\boldsymbol{u}}_{lij}=\boldsymbol{m}_{lij}-\boldsymbol{K}_{li}(\boldsymbol{R}_i\boldsymbol{X}_j+\boldsymbol{t}_i),\\ \boldsymbol{r}_{rij}=\boldsymbol{m}_{rij}-\tilde{\boldsymbol{u}}_{rij}=\boldsymbol{m}_{rij}-\boldsymbol{K}_{ri}[\boldsymbol{R}_{l2r}(\boldsymbol{R}_i\boldsymbol{X}_j+\boldsymbol{t}_i)+\boldsymbol{T}_{l2r}],\end{cases} \tag{4}$$

where $\boldsymbol{m}_{lij}$ and $\boldsymbol{m}_{rij}$ are the measured feature positions of the left and the right cameras, respectively; $\tilde{\boldsymbol{u}}_{lij}$ and $\tilde{\boldsymbol{u}}_{rij}$ are the projection of 3D point $\boldsymbol{X}_j$ in the left-hand and the right-hand images i, respectively; $\boldsymbol{K}_{li}$ and $\boldsymbol{K}_{ri}$ are the intrinsic parameter metrics of the left-hand and the right-hand cameras, respectively; $\boldsymbol{R}_{l2r}$ is the rotation matrix from the left-hand view to the right-hand one, and $\boldsymbol{T}_{l2r}$ is the translation between the binocular cameras, i.e., the distance from the origin of the left camera to the centre of the right one.

The vector of the projection of the left- and right-hand views $\tilde{\boldsymbol{u}}_l=[\boldsymbol{u}_{l11},\boldsymbol{u}_{l12},\ ...,\ \boldsymbol{u}_{l1N},\boldsymbol{u}_{l21},\ \boldsymbol{u}_{l22},\ ...,\ \boldsymbol{u}_{l2N},...]^{\mathrm{T}}$ and $\tilde{\boldsymbol{u}}_r=[\boldsymbol{u}_{r11},\ \boldsymbol{u}_{r12},...,\ \boldsymbol{u}_{r1N},\boldsymbol{u}_{r21},\ \boldsymbol{u}_{r22},\ ...,\ \boldsymbol{u}_{r2N},...]^{\mathrm{T}}$ can be written as $\tilde{\boldsymbol{u}}_l=h(\tilde{\boldsymbol{X}},\tilde{\boldsymbol{\theta}})$ and $\tilde{\boldsymbol{u}}_r=g(\tilde{\boldsymbol{X}},\tilde{\boldsymbol{\theta}})$. If $\boldsymbol{X}$ and $\boldsymbol{\theta}$ are the current estimates of the parameters such that $\tilde{\boldsymbol{X}}=\boldsymbol{X}+\delta\boldsymbol{X}+...$ and $\tilde{\boldsymbol{\theta}}=\boldsymbol{\theta}+\delta\boldsymbol{\theta}+...$, then by Taylor expansion, we have

$$\begin{cases}\tilde{\boldsymbol{u}}_l=h(\boldsymbol{X},\boldsymbol{\theta})+\dfrac{\partial h}{\partial\boldsymbol{X}}\delta\boldsymbol{X}+\dfrac{\partial h}{\partial\boldsymbol{\theta}}\delta\boldsymbol{\theta}+\dots\\ \tilde{\boldsymbol{u}}_r=g(\boldsymbol{X},\boldsymbol{\theta})+\dfrac{\partial g}{\partial\boldsymbol{X}}\delta\boldsymbol{X}+\dfrac{\partial g}{\partial\boldsymbol{\theta}}\delta\boldsymbol{\theta}+\dots\end{cases}$$

Keeping only the linear terms, we have

$$\begin{cases}\boldsymbol{r}_l=\tilde{\boldsymbol{u}}_l-\boldsymbol{m}_l=\dfrac{\partial h}{\partial\boldsymbol{X}}\delta\boldsymbol{X}+\dfrac{\partial h}{\partial\boldsymbol{\theta}}\delta\boldsymbol{\theta},\\ \boldsymbol{r}_r=\tilde{\boldsymbol{u}}_r-\boldsymbol{m}_r=\dfrac{\partial g}{\partial\boldsymbol{X}}\delta\boldsymbol{X}+\dfrac{\partial g}{\partial\boldsymbol{\theta}}\delta\boldsymbol{\theta},\end{cases} \tag{5}$$

and

$$\boldsymbol{r}=\rho_i\begin{bmatrix}\boldsymbol{r}_l\\ \boldsymbol{r}_r\end{bmatrix},\quad \boldsymbol{J}=\begin{bmatrix}\dfrac{\partial h}{\partial\boldsymbol{X}} & \dfrac{\partial h}{\partial\boldsymbol{\theta}}\\ \dfrac{\partial g}{\partial\boldsymbol{X}} & \dfrac{\partial g}{\partial\boldsymbol{\theta}}\end{bmatrix},\quad \boldsymbol{s}=\begin{bmatrix}\delta\boldsymbol{X}\\ \delta\boldsymbol{\theta}\end{bmatrix}.$$

The non-linear least squares problem can be solved using the Levenberg-Marquardt algorithm (Brown and Lowe, 2005). Each iteration step is of the form

$$\boldsymbol{s}=(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}+\lambda\boldsymbol{I})^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{r}. \tag{6}$$

After each iteration, decrease λ by a factor if the error has decreased, or increase λ by a factor if the error has increased (and reject the step).

$\boldsymbol{J}$ is mostly zero (since the derivatives of residuals for image i are zero except with respect to the parameters of image i), so the elements of $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ should be computed directly, instead of computing $\boldsymbol{J}$ first. Examining the structure of $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$,

$$\begin{aligned}&\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\\ &=\begin{bmatrix}\left(\dfrac{\partial h}{\partial\boldsymbol{X}}\right)^{\mathrm{T}}\dfrac{\partial h}{\partial\boldsymbol{X}}+\left(\dfrac{\partial g}{\partial\boldsymbol{X}}\right)^{\mathrm{T}}\dfrac{\partial g}{\partial\boldsymbol{X}} & \left(\dfrac{\partial h}{\partial\boldsymbol{X}}\right)^{\mathrm{T}}\dfrac{\partial h}{\partial\boldsymbol{\theta}}+\left(\dfrac{\partial g}{\partial\boldsymbol{X}}\right)^{\mathrm{T}}\dfrac{\partial g}{\partial\boldsymbol{\theta}}\\ \left(\dfrac{\partial h}{\partial\boldsymbol{\theta}}\right)^{\mathrm{T}}\dfrac{\partial h}{\partial\boldsymbol{X}}+\left(\dfrac{\partial g}{\partial\boldsymbol{\theta}}\right)^{\mathrm{T}}\dfrac{\partial g}{\partial\boldsymbol{X}} & \left(\dfrac{\partial h}{\partial\boldsymbol{\theta}}\right)^{\mathrm{T}}\dfrac{\partial h}{\partial\boldsymbol{\theta}}+\left(\dfrac{\partial g}{\partial\boldsymbol{\theta}}\right)^{\mathrm{T}}\dfrac{\partial g}{\partial\boldsymbol{\theta}}\end{bmatrix}\\ &=\begin{bmatrix}\boldsymbol{U} & \boldsymbol{W}\\ \boldsymbol{W}^{\mathrm{T}} & \boldsymbol{V}\end{bmatrix},\end{aligned}$$

where the structure parameter inverse covariance matrix

$$\boldsymbol{U}=\text{diag}\{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_j, \ldots\}$$

with

$$(\boldsymbol{U}_j)_{3\times3}=\sum_i\left[\left(\frac{\partial h_{ij}}{\partial \boldsymbol{X}_j}\right)^{\mathrm{T}}\frac{\partial h_{ij}}{\partial \boldsymbol{X}_j}+\left(\frac{\partial g_{ij}}{\partial \boldsymbol{X}_j}\right)^{\mathrm{T}}\frac{\partial g_{ij}}{\partial \boldsymbol{X}_j}\right], j=1,2,...$$

and the camera parameter inverse covariance matrix

$$\boldsymbol{V}=\text{diag}\{\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_i, \ldots\}$$

with

$$(\boldsymbol{V}_i)_{7\times7}=\sum_j\left[\left(\frac{\partial h_{ij}}{\partial \boldsymbol{\theta}_i}\right)^{\mathrm{T}}\frac{\partial h_{ij}}{\partial \boldsymbol{\theta}_i}+\left(\frac{\partial g_{ij}}{\partial \boldsymbol{\theta}_i}\right)^{\mathrm{T}}\frac{\partial g_{ij}}{\partial \boldsymbol{\theta}_i}\right], i=1,2,...$$

The camera/structure cross covariance is a full matrix

$$\boldsymbol{W}=\begin{bmatrix}\boldsymbol{W}_{11} & \boldsymbol{W}_{12} & \cdots\\ \boldsymbol{W}_{21} & \boldsymbol{W}_{22} & \cdots\\ \vdots & \vdots & \ddots\end{bmatrix}$$

with

$$\boldsymbol{W}_{11}=\left(\frac{\partial h_{11}}{\partial \boldsymbol{X}_1}\right)^{\mathrm{T}}\frac{\partial h_{11}}{\partial \boldsymbol{\theta}_1}+\left(\frac{\partial g_{11}}{\partial \boldsymbol{X}_1}\right)^{\mathrm{T}}\frac{\partial g_{11}}{\partial \boldsymbol{\theta}_1},$$

$$\boldsymbol{W}_{12}=\left(\frac{\partial h_{12}}{\partial \boldsymbol{X}_2}\right)^{\mathrm{T}}\frac{\partial h_{12}}{\partial \boldsymbol{\theta}_1}+\left(\frac{\partial g_{12}}{\partial \boldsymbol{X}_2}\right)^{\mathrm{T}}\frac{\partial g_{12}}{\partial \boldsymbol{\theta}_1},$$

$$\boldsymbol{W}_{21}=\left(\frac{\partial h_{21}}{\partial \boldsymbol{X}_1}\right)^{\mathrm{T}}\frac{\partial h_{21}}{\partial \boldsymbol{\theta}_2}+\left(\frac{\partial g_{21}}{\partial \boldsymbol{X}_1}\right)^{\mathrm{T}}\frac{\partial g_{21}}{\partial \boldsymbol{\theta}_2},$$

$$\boldsymbol{W}_{22}=\left(\frac{\partial h_{22}}{\partial \boldsymbol{X}_2}\right)^{\mathrm{T}}\frac{\partial h_{22}}{\partial \boldsymbol{\theta}_2}+\left(\frac{\partial g_{22}}{\partial \boldsymbol{X}_2}\right)^{\mathrm{T}}\frac{\partial g_{22}}{\partial \boldsymbol{\theta}_2}.$$

With this new set of unknown parameters, we repeat the steps described in Eq.(6). The estimate is improved iteratively by making $\boldsymbol{X}\rightarrow\boldsymbol{X}+\delta\boldsymbol{X}$ and $\boldsymbol{\theta}\rightarrow\boldsymbol{\theta}+\delta\boldsymbol{\theta}$. The procedure is repeated until the re-projection error $\boldsymbol{r}$ cannot be reduced significantly.

This special BBA algorithm is shown as follows:

Input:

$\boldsymbol{\theta}$: camera parameter vectors.

$\boldsymbol{X}$: 3D point parameter vectors.

h: left projection functions employing the $\boldsymbol{\theta}$ and $\boldsymbol{X}$ to compute the predicted projections $\tilde{\boldsymbol{u}}_{\mathrm{l}ij}$.

g: right projection functions employing the $\boldsymbol{\theta}$ and $\boldsymbol{X}$ to compute the predicted projections $\tilde{\boldsymbol{u}}_{\mathrm{r}ij}$.

$\boldsymbol{m}_{\mathrm{l}ij}$: the observed point locations of the left-hand image.

$\boldsymbol{m}_{\mathrm{r}ij}$: the observed point locations of the right-hand image.

d_{ij}: the disparity of the ith point on the jth pair of images.

λ: the damping term for LM (Levenberg-Marquardt) algorithm.

Output:

$\boldsymbol{s}=\begin{bmatrix}\delta\boldsymbol{X}\\ \delta\boldsymbol{\theta}\end{bmatrix}$: the solution to the normal equations involved in LM-based special BBA.

$\boldsymbol{p}=\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{\theta}\end{bmatrix}$: the parameter vector minimizing the error function.

Compute

$$\boldsymbol{A}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}=\begin{bmatrix}\left(\frac{\partial h}{\partial \boldsymbol{X}}\right)^{\mathrm{T}}\frac{\partial h}{\partial \boldsymbol{X}}+\left(\frac{\partial g}{\partial \boldsymbol{X}}\right)^{\mathrm{T}}\frac{\partial g}{\partial \boldsymbol{X}} & \left(\frac{\partial h}{\partial \boldsymbol{X}}\right)^{\mathrm{T}}\frac{\partial h}{\partial \boldsymbol{\theta}}+\left(\frac{\partial g}{\partial \boldsymbol{X}}\right)^{\mathrm{T}}\frac{\partial g}{\partial \boldsymbol{\theta}}\\ \left(\frac{\partial h}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}}\frac{\partial h}{\partial \boldsymbol{X}}+\left(\frac{\partial g}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}}\frac{\partial g}{\partial \boldsymbol{X}} & \left(\frac{\partial h}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}}\frac{\partial h}{\partial \boldsymbol{\theta}}+\left(\frac{\partial g}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}}\frac{\partial g}{\partial \boldsymbol{\theta}}\end{bmatrix}$$

and the error vectors

$$\boldsymbol{r}=\rho_i\begin{bmatrix}\boldsymbol{r}_{\mathrm{l}}\\ \boldsymbol{r}_{\mathrm{r}}\end{bmatrix}=\rho_i\begin{bmatrix}\tilde{\boldsymbol{u}}_{\mathrm{l}}-\boldsymbol{m}_l\\ \tilde{\boldsymbol{u}}_{\mathrm{r}}-\boldsymbol{m}_r\end{bmatrix}.$$

k=0; v=2; $\boldsymbol{B}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{r}=\sum_i\rho_i\begin{bmatrix}\frac{\partial h}{\partial \boldsymbol{X}}\boldsymbol{r}_{\mathrm{l}}+\frac{\partial g}{\partial \boldsymbol{X}}\boldsymbol{r}_{\mathrm{r}}\\ \frac{\partial h}{\partial \boldsymbol{\theta}}\boldsymbol{r}_{\mathrm{l}}+\frac{\partial g}{\partial \boldsymbol{\theta}}\boldsymbol{r}_{\mathrm{r}}\end{bmatrix}$; $\boldsymbol{p}=\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{\theta}\end{bmatrix}$;

stop=($\|\boldsymbol{B}\|_\infty\le\varepsilon_1$); $\lambda=\tau\cdot\max_{i=1,2,\ldots,m}(A_{ii})$;
while (not *stop*) and ($k<k_{\max}$)
  $k=k+1$;
  repeat
    solve ($\boldsymbol{s}=(\boldsymbol{A}+\lambda\boldsymbol{I})^{-1}\boldsymbol{B}$);
    if ($\|\boldsymbol{s}\|\le\varepsilon_2\|\boldsymbol{p}\|$)
      stop=true;
    else
      $\boldsymbol{p}_{\mathrm{new}}=\boldsymbol{p}+\boldsymbol{s}$;
      $\omega=\left(\|\boldsymbol{r}\|^2-\left\|\begin{matrix}h(\boldsymbol{p}_{\mathrm{new}})-\boldsymbol{m}_{\mathrm{l}}\\ g(\boldsymbol{p}_{\mathrm{new}})-\boldsymbol{m}_{\mathrm{r}}\end{matrix}\right\|^2\right)\Big/[\boldsymbol{s}^{\mathrm{T}}(\lambda\boldsymbol{s}+\boldsymbol{B})]$;
      if $\omega>0$
        $\boldsymbol{p}=\boldsymbol{p}_{\mathrm{new}}$;
        $\boldsymbol{A}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$; $\boldsymbol{r}=\rho_i\begin{bmatrix}h(\boldsymbol{p})-\boldsymbol{m}_{\mathrm{l}}\\ g(\boldsymbol{p})-\boldsymbol{m}_{\mathrm{r}}\end{bmatrix}$; $\boldsymbol{B}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{r}$;
        stop=($\|\boldsymbol{B}\|_\infty\le\varepsilon_1$);
        $\lambda=\lambda\cdot\max(1/3, 1-(2\omega-1)^3)$; v=2;
      else
        $\lambda=\lambda v$; $v=2v$;
      endif
    endif
  until ($\omega>0$) or (*stop*)
endwhile

Brown and Lowe (2005) exploited the structure of $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ by manipulating the system Eq.(6). $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ can in fact be computed in $O(n_{\boldsymbol{\theta}}n_{\boldsymbol{X}})$ operations (the cost of computing $\boldsymbol{W}$), where $n_{\boldsymbol{X}}$ and $n_{\boldsymbol{\theta}}$ are the number of structure and camera parameters, respectively. The total computational cost of sparse BA is $O(mn_{\boldsymbol{\theta}}^2)$, where m is the number of residuals in each image. Thus the complexity of this special BBA is $O(2mn_{\boldsymbol{\theta}}^2)$.

EXPERIMENTAL RESULTS

We used Visual C++ 6.0 and MATLAB 7.1 to realize our proposed method and to display the results in 3D space both on a 1.00 GHz AMD 64×2 Dual Core Processor. The reconstruction time depends on the image resolution and the number of extracted features.

Comparison of the mean re-projection error

We compared the mean RE of the initial 3D reconstruction model and the model with BA and our special BBA. Fig.6 shows an example of the use of the proposed algorithm to perform a 3D reconstruction from a sequence of pairs of images. The traditional BA method is clearly better than the initial model. Although the conventional BA on one sequence of images (the left-hand views) has a smaller 2D RE, the optimized reconstruction model brought a much larger RE on the right-hand views. The inherent constraints between the binocular cameras help to model the object more realistically thereby producing a more accurate model.

Table 1 compares the average 2D RE using the methods mentioned above. The average REs obtained by both methods were smaller than those in the initial model. The original BA method on the left-hand views produced a smaller RE than the special BBA on both views at every sequence of pairs of images. This is expected because the lack of constraints made it easier for the conventional BA to over-fit the data and to produce a smaller RE.

Experimental results under simulated planet environment

In this subsection, models of simulated planet terrain environment are reconstructed. In each case, the entire sequence pair has been incrementally processed frame by frame. For each sequence pair, the output model includes the computed camera positions as well as the set of observed 3D points. We have used our vision-based system to reconstruct the initial 3D models for several long sequences. Then, we optimized the models using BA and our special BBA for comparison.

1. Simulated moon terrain

A simulated moon terrain sequence pair is shown in Fig.7. The reconstructed dense model using the proposed special BBA method is shown in Fig.8. The actual process is as follows. First, dense 3D models for every pair of frames were generated by dense depth maps of input images using a stereo method (Sato *et al*., 2003). Then, combined with camera motions resulting from the optimization method BBA, all the dense 3D models were merged into a common

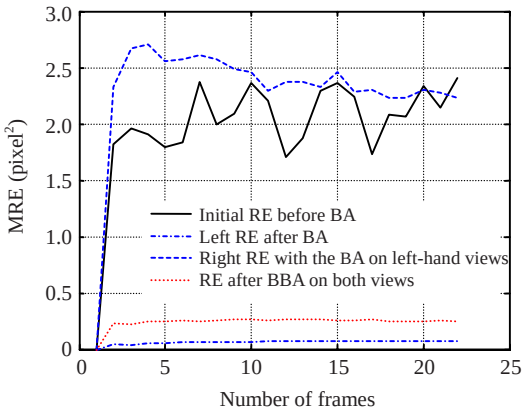


Fig.6 Comparison of the mean re-projection error (MRE) of different methods. BA: bundle adjustment; BBA: binocular bundle adjustment

Table 1 Mean re-projection errors (MREs) of the initial 3D model and reconstruction after bundle adjustment (BA) and our special binocular bundle adjustment (BBA) (unit: pixel^2)

Image resource	Number of points	Initial MRE before BA	MRE after BA on the left-hand views	MRE of the right-hand views with BA on left-hand views only	MRE after BBA on both views
Lipton tea box indoor	1706	0.994343	0.0240338	4.62589	0.1554160
Simulated moon terrain	9387	2.405740	0.0711450	2.67138	0.2485860
Tree-planting hole	40471	1.052530	0.0757512	20.31650	0.0810317
Outdoor ground	10477	2.276480	0.0494920	22.76480	0.1070060

(a) (b)

Fig.7 The first (a) and last (b) images (640×480) from the sequence of 22 pairs of frames of simulated moon terrain indoors using the 3D terrain reconstruction

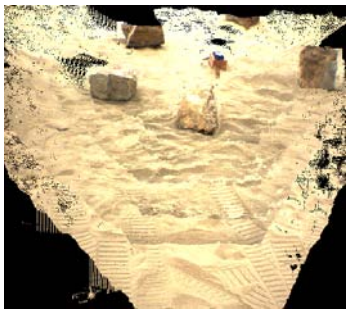
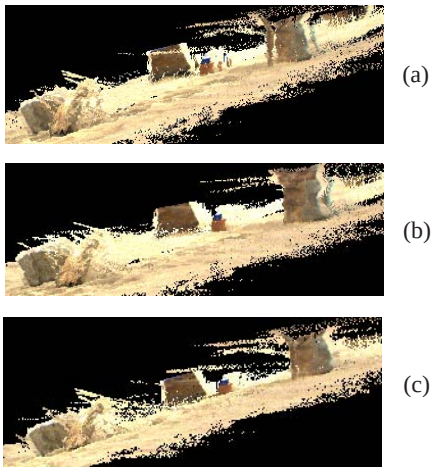


Fig.8 Reconstructed 3D model of the simulated moon terrain using the special BBA

single coordinate system. Finally color and texture were mapped onto the model and the final dense 3D model was obtained. After applying the traditional BA to the model, the RE reduced from 2.405 740 to 0.071 145 0 pixel^2. The result can be seen in Fig.9. Only some small clutters can be observed around the resulting model.

Three side views of the simulated moon terrain model we obtained are shown in Fig.9. It can be seen that the initial terrain model does not construct well. The top right rock has an obvious multi-image effect. It is due to the combination of imperfect initial camera motion estimates. This highlights the effect of the accumulated error on the camera position over the sequence. Moreover, if a position in the sequence has been estimated more or less precisely due to an orientation error provided initially or due to a small number of matches, all following positions will be affected. This is a drawback of incremental modeling approaches.

The model can be optimized by using a conventional BA. As can be seen in Fig.9b, the simulated moon terrain model was greatly improved. The multi-image effect is better. Fig.9c shows the results of the final model after applying the proposed special BBA. The multi-image effect is almost eliminated. The



(a)

(b)

(c)

Fig.9 Side view of the 3D reconstruction of the simulated moon terrain. (a) Initial model without optimization; (b) Optimized model using traditional BA; (c) Optimized model using special BBA

complete algorithm ran in 837.5056 s, of which the whole process of feature matching and initial modeling took a total of 799.3966 s using a MATLAB implementation, and 38.109 s were spent during the special BBA using a VC++ implementation (By contrast, the total required processing time for the conventional BA using a VC++ implementation was 18.125 s).

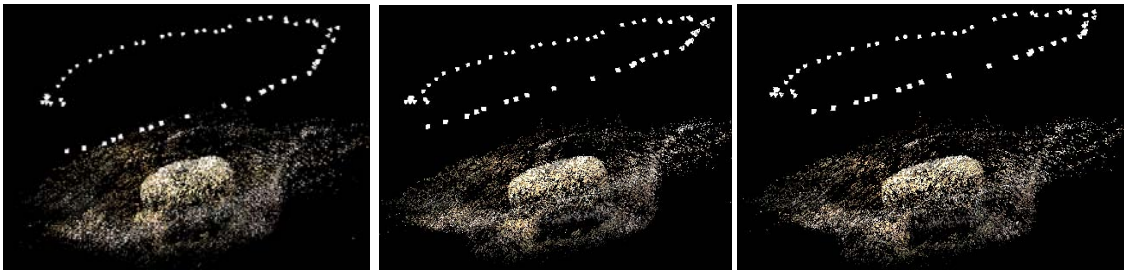
2. Cupped terrain of a tree-planting hole

This example is a cupped tree-planting hole sequence pair outdoors which is similar to the Lunar crater terrain. The entire sequence is composed of 76 pairs of stereo images. One of the sequence of images is shown in Fig.10. Two different views of the reconstructed sparse model using the proposed BBA method are shown in Fig.11. These sparse 3D points were all simultaneously optimized with camera motions by the new refined method BBA, as shown in Fig.12. And the dense models are shown in Fig.13.

Fig.10 Image 45 from the sequence of 76 pairs of frames (640×480) of tree-planting hole outdoors using the 3D terrain reconstruction

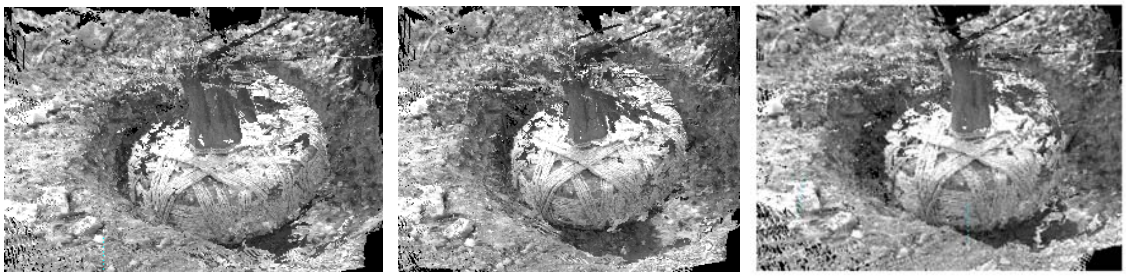
(a) (b)

Fig.11 Reconstructed 3D model of the tree-planting hole using special BBA. (a) Top view; (b) Side view



(a) (b) (c)

Fig.12 Sparse model of the 3D reconstruction (40 471 points) of the tree-planting hole. The white squares denote camera positions. (a) Initial model without optimization; (b) Optimized model using traditional BA; (c) Optimized model using special BBA



(a) (b) (c)

Fig.13 Dense model of the 3D reconstruction of the tree-planting hole. (a) Initial model without optimization; (b) Optimized model using classical BA; (c) Optimized model using special BBA

The operating process is similar to that in the previous case. After applying the traditional BA to the model, the RE reduced from 1.052 530 to 0.075 751 2 pixel^2. The results can be seen in Figs.12b and 13b. Our special BBA has been implemented to help refine the model through the sequence pairs. The results are shown in Figs.12c and 13c. The orbit of the cameras does not close well in the initial model and it improves after applying optimization BA and special BBA. Fig.13 shows the textured surface of multiple projections and the output dense 3D models are obtained using dense depth estimation. The tree trunk in the special BBA model is much more convergent than that in the initial model. The complete algorithm ran in 2501.8268 s, of which the whole process of feature matching and initial modeling took a total of 2447.7168 s using a MATLAB implementation, and 54.11 s were spent during the special BBA using a VC++ implementation (By contrast, the total computation time for the original BA using a VC++ implementation was 25.328 s).

CONCLUSION

We have presented a special terrain reconstruction method for planet rover using a special BBA. Our method is based on pure vision without INS data. We found that although classical BA produced smaller 2D REs on the left-hand views, it actually brought much larger errors on the right-hand ones. We believe that the 2D REs in the right-hand views are also a meaningful and accurate measure of the reconstruction results. We refined only the left-hand motions for simplification, and optimized REs in both views to ensure the precision. The experimental results show that the special BBA method can reduce the RE and lead to more accurate pose estimation.

We will further enhance our algorithms and software. In the near future we plan to replace the point cloud models with true surface meshes generated by a robust and incremental depth map integration technique. Future work will enhance the output dense 3D models by using interpolation, triangulation and texture mapping, which would lead to no clutter.

References

Brown, M., Lowe, D.G., 2005. Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets. Proc. 5th Int. Conf. on 3-D Digital Imaging and Modeling, p.56-63. [doi:10.1109/3DIM.2005.81]

Brown, M., Szeliski, R., Winder, S., 2005. Multi-image Matching Using Multi-scale Oriented Patches. Proc. Int. Conf. on Computer Vision and Pattern Recognition, San Diego, p.510-517. [doi:10.1109/CVPR.2005.235]

Di, K.C., Xu, F.L., Li, R.X., 2004. Constrained Bundle Adjustment of Panoramic Stereo Images for Mars Landing Site Mapping. Proc. 4th Int. Symp. on Mobile Mapping Technology, Kunming, China, p.1-6.

Di, K.C., Xu, F.L., Wang, J., Niu, X.T., Serafy, C., Zhou, F., Li, R.X., Matthies, L., 2005. Surface Imagery Based Mapping and Rover Localization. Proc. ASPRS Annual Conf., Baltimore, MD (CD-ROM). [doi:10.1029/2005JE002483]

Eaton, D., 2005. Answering 'Where Am I?' by Nonlinear Least Squares. Proc. Int. Conf. on Computer Vision, Beijing, p.1-15.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, **24**(6):381-385. [doi:10.1145/358669.358692]

Kwolek, B., 2007. Visual Odometry Based on Gabor Filters and Sparse Bundle Adjustment. IEEE Int. Conf. on Robotics and Automation, Roma, Italy, p.3573-3578.

Labrie, M., Hebert, P., 2007. Efficient Camera Motion and 3D Recovery Using an Inertial Sensor. Fourth Canadian Conf. on Computer and Robot Vision, p.55-62. [doi:10.1109/CRV.2007.23]

Li, R.X., Ma, F., Xu, F.L., Matthies, L.H., Olson, C.F., Arvidson, R.E., 2002. Localization of Mars rovers using descent and surface-based image data. *J. Geophys. Res.*, **107**(E11):8004. [doi:10.1029/2000JE001443]

Li, R.X., Di, K.C., Matthies, L.H., Arvidson, R.E., Folkner, W.M., Archinal, B.A., 2004. Rover localization and landing site mapping technology for the 2003 Mars exploration rover mission. *J. Photo. Eng. Remote Sensing*, **70**(1):77-90.

Lourakis, M.I.A., Argyros, A.A., 2004. The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm. Technical Report 340. Institute of Computer Science, FORTH, Heraklion, Crete, Greece, p.1-21.

Lowe, D.G., 1999. Object Recognition from Local Scale-invariant Features. Proc. Int. Conf. on Computer Vision, Corfu, Greece, p.1150-1157. [doi:10.1109/ICCV.1999.790410]

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, **60**(2):91-110. [doi:10.1023/B:VISI.0000029664.99615.94]

Matthies, L.H., Shafer, S.A., 1987. Error modeling in stereo navigation. *IEEE J. Rob. Autom.*, **3**(3):239-248.

Olson, C.F., Matthies, L.H., Schoppers, M., Maimone, M.W., 2003. Rover navigation using stereo ego-motion. *Rob. Auton. Syst.*, **43**:215-229. [doi:10.1016/S0921-8890(03)00004-6]

Reitinger, B., Zach, C., Schmalstieg, D., 2007. Augmented Reality Scouting for Interactive 3D Reconstruction. IEEE Virtual Reality Conf., Charlotte, North Carolina, USA, p.219-222. [doi:10.1109/VR.2007.352485]

Sato, T., Kanbara, M., Yokoya, N., 2003. Outdoor Scene Reconstruction from Multiple Image Sequences Captured by a Hand-held Video Camera. Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems, p.113-118. [doi:10.1109/MFI-2003.2003.1232642]

Shum, H.Y., Ke, Q.F., Zhang, Z.Y., 1999. Efficient Bundle Adjustment with Virtual Key Frames: A Hierarchical Approach to Multi-frame Structure from Motion. Proc. Int. Conf. on Computer Vision and Pattern Recognition, **2**:538-543. [doi:10.1109/CVPR.1999.784733]

Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A., 1999. Bundle Adjustment—A Modern Synthesis. Proc. Int. Workshop on Visual Algorithm: Theory and Practice, Corfu, Greece, p.298-372.

Umeyama, S., 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Anal. Machine Intell.*, **13**(4):376-380. [doi:10.1109/34.88573]

Wolf, P.R., DeWitt, B.A., 2000. Elements of Photogrammetry with Applications in GIS (3rd Ed.). McGraw Hill, Boston, MA, USA, p.608.

Wong, K.H., Chang, M.M.Y., 2004. 3D Model Reconstruction by Constrained Bundle Adjustment. Proc. 17th Int. Conf. on Pattern Recognition, **3**:902-905. [doi:10.1109/ICPR.2004.1334674]