Exploiting Mutual Camera Visibility in Multi-Camera Motion Estimation

Christian Kurz[1], Thorsten Thormählen[1], Bodo Rosenhahn[2], and Hans-Peter Seidel[1]

[1] Max Planck Institute for Computer Science (MPII), Saarbrücken, Germany
[2] Leibniz University Hannover, Institut für Informationsverarbeitung

Abstract. This paper addresses the estimation of camera motion and 3D reconstruction from image sequences for multiple independently moving cameras. If multiple moving cameras record the same scene, a camera is often visible in another camera's field of view. This poses a constraint on the position of the observed camera, which can be included into the conjoined optimization process. The paper contains the following contributions: Firstly, a fully automatic detection and tracking algorithm for the position of a moving camera in the image sequence of another moving camera is presented. Secondly, a sparse bundle adjustment algorithm is introduced, which includes this additional constraint on the position of the tracked camera. Since the additional constraints minimize the geometric error at the boundary of the reconstructed volume, the total reconstruction accuracy can be improved significantly. Experiments with synthetic and challenging real world scenes show the improved performance of our fully automatic method.

1 Introduction

Simultaneous estimation of camera motion and 3D reconstruction from image sequences is a well-established technique in computer vision [1–3]. Often this problem is referred to as Structure-from-Motion (SfM), Structure-and-Motion (SaM), or Visual Simultaneous Location and Mapping (Visual SLAM). This paper investigates the special scenario of multiple independently moving cameras that capture the same scene. In such a scenario it is often the case that a camera can be observed by another camera. This puts an additional constraint on the position of the observed camera. The additional constraint can be exploited in the estimation process in order to achieve more accurate results.

Multi-camera systems, e.g., stereo cameras, light field capturing systems [4], and markerless motion capturing setups [5] employing multiple cameras, are very common in computer vision. Until now, almost all of these camera setups have been static.

For static cameras, Sato [6] analyzed the epipolar geometry for cases where multiple cameras are projected into each other's images. In these cases, the epipoles are directly given by the projection of the camera centers. Therefore, the epipolar geometry can be calculated from less feature correspondences between the images.

Sometimes static setups are mounted on a moving platform [7], e.g., Stewénius and Åström investigated the structure-and-motion problem for multiple rigidly moving cameras in an autonomous vehicle [8], and Frahm et al. [9] mounted several rigidly coupled cameras on a moving pole.

Recently, Thormählen et al. [10] presented a solution for multiple independently moving cameras that capture the same scene. This scenario frequently occurs in practice, e.g., multi-camera recordings of TV shows or multi-camera shots in movie productions. Camera motion estimation and 3D reconstruction is performed independently for each sequence with a feature-based single camera structure-and-motion approach. The independent reconstructions are then merged into a common global coordinate system, followed by a conjoined bundle adjustment [2, 11] over the merged sequences.

This paper adapts a similar approach and extends it for the case where a moving camera is located in the field of view of another moving camera. The following two contributions are made:

- A detection and tracking algorithm is used to determine the projection of a camera center in the image of another camera. Thereby, the user has the choice between a fully automatic and a semi-automatic approach. For the fully automatic approach, the cameras have to be retrofitted with a color pattern. For the semi-automatic approach the user manually defines the position of the camera center projection in the first image where the camera is visible. For both approaches the camera center is automatically tracked in the subsequent images, whereby the tracking algorithm is guided by the available initial camera center estimates.
- A sparse bundle adjustment algorithm is presented that allows incorporating the additional constraints given by the tracked camera centers. These constraints minimize the geometric error at the boundary of the reconstructed volume, which is usually the most sensitive part for reconstruction. Consequently, the total reconstruction accuracy can be improved significantly.

2 Scene Model

Consider a total number of N moving cameras, which capture the image sequences S_n, with $n = 1, \ldots, N$, consisting of K images $I_{k,n}$, with $k = 1, \ldots, K$, each. The cameras are synchronized, so that images $I_{k,n}$ for all n are recorded at the same point in time k. Let $\mathtt{A}_{k,n}$ be the 3×4 camera matrix corresponding to image $I_{k,n}$. A set of J 3D object points $\mathbf{P}_j = (P_x, P_y, P_z, 1)^\top$, with $j = 1, \ldots, J$, represents the static scene geometry, where the individual 3D object points are visible in at least a subset of all the images. In addition, the 2D feature points corresponding to $\mathbf{P}_j$, as seen in image $I_{k,n}$, are given by $\mathbf{p}_{j,k,n} = (p_x, p_y, 1)^\top$. This notation is clarified by Fig. 1.

Let $\mathbf{C}_{k,n} = (C_x, C_y, C_z, 1)^\top$ be the center of camera n at time k. The 2D image position of $\mathbf{C}_{k,n}$, as seen from camera $\tilde{n}$, with $n \neq \tilde{n}$, is now defined as $\mathbf{c}_{k,n,\tilde{n}} = (c_x, c_y, 1)^\top$. Likewise, the position of the projection of $\mathbf{C}_{k,n}$ in $I_{k,\tilde{n}}$ is defined as $\hat{\mathbf{c}}_{k,n,\tilde{n}} = \mathtt{A}_{k,\tilde{n}}\mathbf{C}_{k,n}$. Note that, in an ideal noise-free case $\hat{\mathbf{c}}_{k,n,\tilde{n}} = \mathbf{c}_{k,n,\tilde{n}}$; however, in real situations, it can usually be observed that $\hat{\mathbf{c}}_{k,n,\tilde{n}} \neq \mathbf{c}_{k,n,\tilde{n}}$.

3 Unconstrained Reconstruction

In a first step, synchronization of the N individual image sequences S_n is achieved using a method similar to the one presented by Hasler et al. [12]. This method analyzes

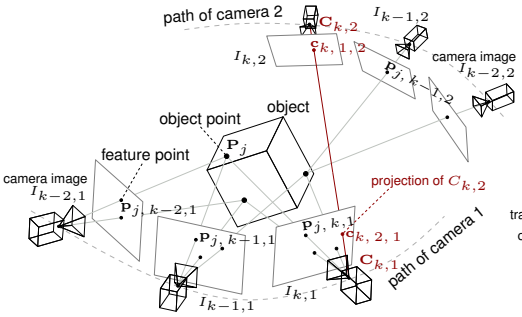

Fig. 1. Multiple cameras observe the same object. The camera center $\mathbf{C}_{k,2}$ of camera 2 is visible in image $I_{k,1}$ of camera 1, and vice versa.

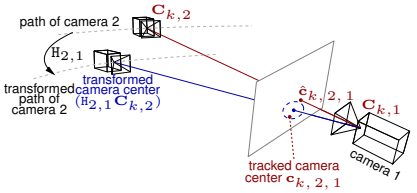

Fig. 2. Compensation of the systematically erroneous path of camera 2 by applying a similarity transformation $\mathtt{H}_{2,1}$.

the audio data, which is recorded simultaneously with the video data. A synchronization offset of at most half a frame is usually achieved. This approach allows the application of standard consumer cameras; a hard-wired studio environment is not required and the recordings can take place at arbitrary sets, including outdoor locations.

In a second step, each camera sequence is processed independently with a standard structure-from-motion algorithm. This establishes initial estimates for every single camera matrix $\mathtt{A}_{k,n}$ and every 3D object point $\mathbf{P}_j$ of the rigid scene. The 2D feature points $\mathbf{p}_{j,k,n}$ are detected and tracked through the image sequences. For each tracked 2D feature point $\mathbf{p}_{j,k,n}$ a corresponding 3D object point $\mathbf{P}_j$ is estimated. The applied algorithms are robust against outlier feature tracks introduced by moving objects, repetitive structures, or illumination changes. Intrinsic camera parameters are determined by self-calibration [3]. The estimation is finalized by a bundle adjustment.

In a third step, a similar approach as in [10] is employed to register the independent reconstructions into a common global coordinate system. The required similarity transformation for each individual reconstruction is estimated from corresponding feature tracks found via wide baseline matching between the image sequences. A conjoined bundle adjustment over all N reconstructions is performed to achieve equal distribution of the residual error over the whole scene. This minimization problem requires finding

$$\underset{\mathtt{A},\mathbf{P}}{\arg\min} \quad \sum_{n=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \mathrm{d}(\mathbf{p}_{j,k,n}\,,\, \mathtt{A}_{k,n}\,\mathbf{P}_j)^2 \quad , \tag{1}$$

where $\mathrm{d}(\dots)$ denotes the Euclidean distance. It is solved using the sparse Levenberg-Marquardt (LM) algorithm, as described in [2].

After these processing steps, an initial reconstruction of the scene has been established, which will be referred to as *unconstrained reconstruction* henceforth. Though the residual error is usually small, the inhomogeneous distribution of the corresponding feature tracks found by wide baseline matching may lead to estimation results not accurately reflecting the true structure of the scene. These inhomogeneities can arise because reliable merging candidates can usually be found more easily at the center of the reconstructed volume where the individual camera's fields of view overlap.

4 Detection and Tracking of Camera Centers

The unconstrained reconstruction can be improved by exploiting the visibility of the camera center in the field of view of another camera. To incorporate this additional constraint into the bundle adjustment, the determination of the 2D image positions of the visible camera centers $\mathbf{c}_{k,n,\tilde{n}}$ is necessary. The user has the choice to either use a fully automatic or a semi-automatic approach. The fully automatic approach comprises the detection and tracking of the camera centers, whereas the semi-automatic approach requires the user to provide the positions of the projection of the camera centers for the first image they appear in.

4.1 Detection

The automatic detection of the camera centers requires the image of the cameras to be descriptive. One possibility would be to use a learning-based approach trained on the appearance of the camera. However, as small consumer cameras are used, reliable detection is challenging. As a consequence, the cameras were retrofitted with descriptive color patterns to facilitate the automatic detection.

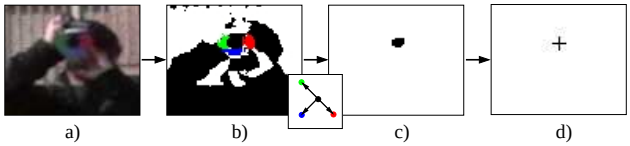

Fig. 3. Steps of the detection algorithm: a) input image detail, b) image after the conversion to HSV color space and color assignment, c) pixels that pass the geometric structure evaluation, d) detected camera center.

Fig. 3 summarizes the automatic detection process and also shows the used color pattern, which consists of three patches with different colors. The pattern colors red, green, and blue were chosen, as they can easily be separated in color space. Since the front of the cameras is usually visible, the camera lens serves as additional black patch.

At first, the image is converted from RGB to HSV color space. All the pixels are then either assigned to one of the three pattern colors, black, or the background based on their proximity to the respective colors in HSV color space. Thereby, the value parameter (V) of the HSV color space model is ignored to achieve illumination invariance. For each black pixel the geometric structure of the pixels in a window around the pixel is examined. To be more specific, for each red pixel in the neighborhood of the black pixel, a green pixel is required to lie in the exact same distance in the opposite direction. Furthermore, a blue pixel must be located in the direction perpendicular to the connection line between the red and the green pixel. Again, the distance of the blue pixel from the black pixel must be exactly the same as the distance from the red to the black pixel. In addition, it must lie on the correct side of the connection line (see Fig. 3). Since there are usually multiple detections per camera, the centers of the clusters yield the desired positions of the camera centers.

4.2 Tracking

If fully automatic detection is not used, e.g., because color patterns for the camera are not available, the user is required to input the initial positions of the camera centers for the first image the camera appears. To simplify the notation, it is assumed for a moment that all other cameras are visible in the first image of every camera. Thus, the positions $\mathbf{c}_{1,n,\tilde{n}}$ are now determined, either through user input or automatic detection.

For the tracking of the camera positions through the image sequences, a tracking algorithm based on Normalized Cross Correlation (NCC) matching is employed. A special feature of the algorithm is the guided matching process, which relies on the known initial estimates for the camera positions given by the unconstrained reconstruction to improve the robustness of the tracking.

Starting from $\mathbf{c}_{k,n,\tilde{n}}$ in image $I_{k,\tilde{n}}$, the algorithm searches for $\mathbf{c}_{k+1,n,\tilde{n}}$. This is done by calculating NCC matching scores for a window around $\mathbf{c}_{k,n,\tilde{n}}$ in image $I_{k+1,\tilde{n}}$. The results of this operation are stored in a sorted list with positions producing the highest matching scores at the front.

Starting with the best match, it is checked whether the NCC score is above a user-defined threshold t_0 or not. If no matches with sufficiently high score are present, the algorithm aborts. In case of a valid match, the solution is cross-checked by calculating a second NCC score, between the current best match and the initial camera position $\mathbf{c}_{1,n,\tilde{n}}$ (assuming the initial position to originate from $I_{1,\tilde{n}}$).

If the score for the second NCC matching is below another user-defined threshold t_1, instead of terminating, the algorithm simply processes the match with the next-lower score in the list. The cross-check reduces the effects of slow deviation of the feature point's description over time, since it assures that the original position can be found by reverse tracking.

Albeit performing very well and producing results of high tracking accuracy, this unguided tracking fails under certain conditions. Mismatches can occur due to similar image regions in the search window. Moreover, camera centers can leave and reenter the camera's field of view, or might get occluded by foreground objects, which causes traditional unguided tracking algorithms to lose the target.

Therefore, an additional constraint is introduced. As stated before, a set of good initial estimates for the camera projection matrices $\mathtt{A}_{k,n}$ is available from the unconstrained reconstruction. These estimates contain estimates for the camera centers $\mathbf{C}_{k,n}$, since $\mathtt{A}_{k,n}\mathbf{C}_{k,n} = \mathbf{0}$.

Due to registration errors, the tracked positions of the camera centers $\mathbf{c}_{k,n,\tilde{n}}$ and the positions resulting from reprojection of the camera centers $\hat{\mathbf{c}}_{k,n,\tilde{n}} = \mathtt{A}_{k,\tilde{n}}\mathbf{C}_{k,n}$ systematically deviate from each other (see Fig. 2).

These registration errors are compensated by estimating a common similarity transformation for the camera centers $\mathbf{C}_{i,n}$, with $i = 1,\ldots,k$, represented by a 4×4 matrix $\mathtt{H}_{n,\tilde{n}}$. More formally, it is required to find

$$\underset{\mathtt{H}_{n,\tilde{n}}}{\arg\min} \quad \sum_{i=1}^{k} \mathrm{d}(\mathbf{c}_{i,n,\tilde{n}}\,,\ \mathtt{A}_{i,\tilde{n}}\,\mathtt{H}_{n,\tilde{n}}\,\mathbf{C}_{i,n})^2 \qquad . \tag{2}$$

The similarity transformation $\mathtt{H}_{n,\tilde{n}}$ allows for 7 degrees of freedom (3 for translation, 3 for rotation, and 1 for scale), and therefore a minimum of 4 measurements $\mathbf{c}_{i,n,\tilde{n}}$

is needed to prevent ambiguities. For that reason it is clear that this approach cannot be applied to the first 3 images after the initial one, but starting from the fourth one it provides a sophisticated means of determining whether the match is a false positive or not, as described in the following.

The projection $(\mathtt{A}_{k,\tilde{n}}\,\mathtt{H}_{n,\tilde{n}}\,\mathbf{C}_{k,n})$ gives a quite accurate estimate of the true $\hat{\mathbf{c}}_{k,n,\tilde{n}}$ that can be used to determine if $\mathbf{c}_{k,n,\tilde{n}}$ lies within a certain distance t_2 from the estimated projection of the camera center (see Fig. 2). The expectation of the residual error of Eq. (2) is given by $\epsilon_{res} = \sigma(1-(d/M))^{1/2}$, where $d = 7$ is the number of parameters of $\mathtt{H}_{n,\tilde{n}}$ and $M = 2k$ is the number of measured $\mathbf{c}_{i,n,\tilde{n}}$ (see Hartley and Zisserman [2] for details on the expectation of residual errors). Using the relation $t_2 = \epsilon_{res} + b$, with user-defined values for standard deviation σ and bias b, t_2 can be changed adaptively. The bias value accounts for systematic errors, which cannot be compensated by the similarity transformation.

If the currently best match from the sorted list does not fulfill the requirements, the next match in the list is processed. Once a match is accepted, the transformation $\mathtt{H}_{n,\tilde{n}}$ is refined and the algorithm moves on to the next image.

If a tracked camera center leaves the camera's field of view or gets occluded by foreground objects, the remainder of the image sequence is checked for possible reappearance of the camera. The reappearance point can be predicted with $(\mathtt{A}_{k,\tilde{n}}\,\mathtt{H}_{n,\tilde{n}}\,\mathbf{C}_{k,n})$ using the last transformation $\mathtt{H}_{n,\tilde{n}}$ that was estimated before the track was lost. This prediction is then used to reinitialize the NCC matching process.

5 Sparse Bundle Adjustment With Additional Camera Center Constraints

Given tracked positions of camera centers $\mathbf{c}_{k,n,\tilde{n}}$, Eq. (1) is expanded to accommodate for the additional constraints:

$$\underset{\mathtt{A},\mathbf{P}}{\arg\min} \quad \sum_{n=1}^{N}\sum_{j=1}^{J}\sum_{k=1}^{K} \mathrm{d}(\mathbf{p}_{j,k,n}\,,\,\mathtt{A}_{k,n}\,\mathbf{P}_j)^2 + w\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\tilde{n}=1}^{N} \mathrm{d}(\mathbf{c}_{k,n,\tilde{n}}\,,\,\mathtt{A}_{k,\tilde{n}}\,\mathbf{C}_{k,n})^2 \tag{3}$$

for $n \neq \tilde{n}$, with w being a user-defined weight factor.

As in the unconstrained case, this minimization problem can be solved by the sparse LM algorithm, as derived in the following. A similar notation as in the book by Hartley and Zisserman [2] is used.

The measurement vector $\tilde{\mathbf{p}} = \left(\bar{\mathbf{p}}^\top,\, \bar{\mathbf{c}}^\top\right)^\top$ is assembled from the vector $\bar{\mathbf{p}}$ of all 2D feature points $\mathbf{p}_{j,k,n}$ placed one after another in a single column, and the vector $\bar{\mathbf{c}}$ constructed alike from all tracked camera centers $\mathbf{c}_{k,n,\tilde{n}}$.

In a similar fashion, the parameter vector $\mathbf{q} = \left(\mathbf{a}^\top,\, \mathbf{b}^\top\right)^\top$ can be obtained by assembling a parameter vector $\mathbf{a}$ denoting the corresponding set of parameters describing the cameras, and parameter vector $\mathbf{b}$ denoting the corresponding set of parameters describing the points.

In each step of the LM algorithm the following linear equation system needs to be solved:

$$\mathtt{J}\boldsymbol{\delta} = \boldsymbol{\epsilon} \tag{4}$$

with the Jacobian matrix $\mathtt{J} = \partial\tilde{\mathbf{p}}/\partial\mathbf{q}$, the residual vector $\boldsymbol{\epsilon}$, and the update vector $\boldsymbol{\delta}$ of the LM algorithm, which is the solution to the least squares problem. The residual vector $\boldsymbol{\epsilon} = \left(\boldsymbol{\epsilon_p}^\top, \boldsymbol{\epsilon_c}^\top\right)^\top$ is assembled from the residual vector of the 2D feature points $\boldsymbol{\epsilon_p}$ and the residual vector of the camera centers $\boldsymbol{\epsilon_c}$.

The Jacobian matrix $\mathtt{J}$ has a block structure

$$\mathtt{J} = \begin{bmatrix} \bar{\mathtt{A}} & \bar{\mathtt{B}} \\ \bar{\mathtt{C}} & \mathtt{0} \end{bmatrix} \quad , \quad \text{where} \quad \bar{\mathtt{A}} = \begin{bmatrix} \dfrac{\partial\bar{\mathbf{p}}}{\partial\mathbf{a}} \end{bmatrix} \quad , \quad \bar{\mathtt{B}} = \begin{bmatrix} \dfrac{\partial\bar{\mathbf{p}}}{\partial\mathbf{b}} \end{bmatrix} \quad \text{and} \quad \bar{\mathtt{C}} = \begin{bmatrix} \dfrac{\partial\bar{\mathbf{c}}}{\partial\mathbf{a}} \end{bmatrix} \quad . \tag{5}$$

The linear equation system of Eq. (4) evaluates to

$$[\tilde{\mathtt{A}}|\tilde{\mathtt{B}}] \begin{pmatrix} \boldsymbol{\delta_a} \\ \boldsymbol{\delta_b} \end{pmatrix} = \boldsymbol{\epsilon} \quad , \quad \text{with} \quad \tilde{\mathtt{A}} = \begin{bmatrix} \bar{\mathtt{A}} \\ \bar{\mathtt{C}} \end{bmatrix} \quad \text{and} \quad \tilde{\mathtt{B}} = \begin{bmatrix} \bar{\mathtt{B}} \\ \mathtt{0} \end{bmatrix} \quad . \tag{6}$$

The normal equations corresponding to Eq. (4) are given as

$$\mathtt{J}^\top\Sigma^{-1}\mathtt{J}\boldsymbol{\delta} = \mathtt{J}^\top\Sigma^{-1}\boldsymbol{\epsilon} \quad , \quad \text{with} \quad \Sigma = \begin{bmatrix} \Sigma_{\mathbf{p}} & \mathtt{0} \\ \mathtt{0} & \Sigma_{\mathbf{c}} \end{bmatrix} \quad , \tag{7}$$

where $\Sigma_{\mathbf{p}}$ is the covariance matrix of the 2D feature points, and $\Sigma_{\mathbf{c}}$ the covariance matrix of the tracked camera centers. In absence of other knowledge, the matrix $\Sigma_{\mathbf{c}}$ is chosen to be the identity matrix. The normal equations evaluate to

$$\begin{bmatrix} \tilde{\mathtt{A}}^\top\Sigma^{-1}\tilde{\mathtt{A}} & \tilde{\mathtt{A}}^\top\Sigma^{-1}\tilde{\mathtt{B}} \\ \tilde{\mathtt{B}}^\top\Sigma^{-1}\tilde{\mathtt{A}} & \tilde{\mathtt{B}}^\top\Sigma^{-1}\tilde{\mathtt{B}} \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta_a} \\ \boldsymbol{\delta_b} \end{pmatrix} = \begin{pmatrix} \tilde{\mathtt{A}}^\top\Sigma^{-1}\boldsymbol{\epsilon} \\ \tilde{\mathtt{B}}^\top\Sigma^{-1}\boldsymbol{\epsilon} \end{pmatrix} \quad , \tag{8}$$

which can be simplified by back-substitution:

$$\begin{bmatrix} \bar{\mathtt{A}}^\top\Sigma_{\mathbf{p}}^{-1}\bar{\mathtt{A}} + \bar{\mathtt{C}}^\top\Sigma_{\mathbf{c}}^{-1}\bar{\mathtt{C}} & \bar{\mathtt{A}}^\top\Sigma_{\mathbf{p}}^{-1}\bar{\mathtt{B}} \\ \bar{\mathtt{B}}^\top\Sigma_{\mathbf{p}}^{-1}\bar{\mathtt{A}} & \bar{\mathtt{B}}^\top\Sigma_{\mathbf{p}}^{-1}\bar{\mathtt{B}} \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta_a} \\ \boldsymbol{\delta_b} \end{pmatrix} = \begin{pmatrix} \bar{\mathtt{A}}^\top\Sigma_{\mathbf{p}}^{-1}\boldsymbol{\epsilon_p} + \bar{\mathtt{C}}^\top\Sigma_{\mathbf{c}}^{-1}\boldsymbol{\epsilon_c} \\ \bar{\mathtt{B}}^\top\Sigma_{\mathbf{p}}^{-1}\boldsymbol{\epsilon_p} \end{pmatrix} \quad . \tag{9}$$

The corresponding block structure is

$$\begin{bmatrix} \mathtt{U}^* & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{V}^* \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta_a} \\ \boldsymbol{\delta_b} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\epsilon}_{\mathtt{A}} \\ \boldsymbol{\epsilon}_{\mathtt{B}} \end{pmatrix} \quad , \tag{10}$$

where $\mathtt{U}^*$ denotes $\mathtt{U}$ augmented by multiplying its diagonal entries by a factor of $1 + \lambda$, and $\mathtt{V}^*$ likewise. Left multiplication with $\begin{bmatrix} \mathtt{I} & -\mathtt{W}\mathtt{V}^{*-1} \\ \mathtt{0} & \mathtt{I} \end{bmatrix}$, where $\mathtt{I}$ is the identity matrix, yields

$$\begin{bmatrix} \mathtt{U}^* - \mathtt{W}\mathtt{V}^{*-1}\mathtt{W}^\top & \mathtt{0} \\ \mathtt{W}^\top & \mathtt{V}^* \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta_a} \\ \boldsymbol{\delta_b} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\epsilon}_{\mathtt{A}} - \mathtt{W}\mathtt{V}^{*-1}\boldsymbol{\epsilon}_{\mathtt{B}} \\ \boldsymbol{\epsilon}_{\mathtt{B}} \end{pmatrix} \quad . \tag{11}$$

The equation

$$\left(\mathtt{U}^* - \mathtt{W}\mathtt{V}^{*-1}\mathtt{W}^\top\right)\boldsymbol{\delta}_{\mathbf{a}} = \boldsymbol{\epsilon}_{\mathtt{A}} - \mathtt{W}\mathtt{V}^{*-1}\boldsymbol{\epsilon}_{\mathtt{B}} \tag{12}$$

can be used to find $\boldsymbol{\delta_a}$, which may be back-substituted to get $\boldsymbol{\delta_b}$ from

$$\mathtt{V}^*\boldsymbol{\delta_b} = \boldsymbol{\epsilon}_{\mathtt{B}} - \mathtt{W}^\top\boldsymbol{\delta_a} \quad . \tag{13}$$

These derivations are closely related to those of standard sparse bundle adjustment. It is thus very easy to incorporate the modifications into existing implementations without introducing significant additional computational overhead.

This *constrained bundle adjustment* can improve the unconstrained reconstruction of Sec. 3. At first, a projective constrained bundle adjustment is performed (12 parameters per 3×4 camera matrix $\mathtt{A}$). Afterwards, a new self-calibration and a metric constrained bundle adjustment with 7 parameters per camera view (3 for translation, 3 for rotation, and 1 for focal length) is executed.

6 Results

In this section, experiments with synthetic and real scenes are shown. The experiments on real scenes are also presented in the video provided with this paper, which can be found at `http://www.mpi-inf.mpg.de/users/ckurz/`.

6.1 Experiments with synthetic data

To evaluate if the constrained sparse bundle adjustment of Sec. 5 achieves higher accuracy than the standard bundle adjustment used for the generation of the unconstrained reconstruction of Sec. 3, a comparison with synthetic data is performed.

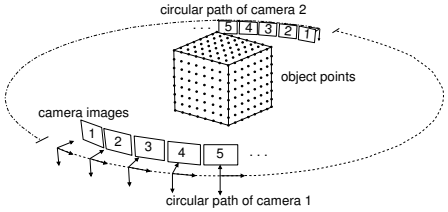

Fig. 4. Setup of the scene to generate synthetic measurement values with known ground truth camera parameters.

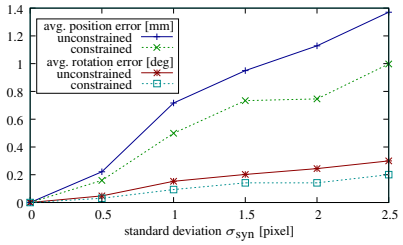

Fig. 5. Average absolute position error and average absolute rotation error of the estimated camera motion over standard deviation σ_{syn}.

Fig. 4 shows the setup of the scene to generate synthetic measurement values for the 2D feature points $\mathbf{p}_{j,\,k,\,n}$ and the tracked camera centers $\mathbf{c}_{k,\,n,\,\tilde{n}}$. Two virtual cameras with known ground truth camera parameter are observing the same 296 object points $\mathbf{P}_j$, which are placed in a regular grid on the surface of a cube with an edge length of 100 mm. The two virtual cameras have an opening angle of 30 degrees and rotate on a circular path with a radius of 300 mm around the object points. Each virtual camera generates 20 images, and the camera centers are mutually visible in every image. Using the ground truth camera parameters and ground truth positions of object points, ground truth measurements are generated for the 2D feature points $\mathbf{p}_{j,\,k,\,n}$ and the tracked camera centers $\mathbf{c}_{k,\,n,\,\tilde{n}}$. These measurements are then disturbed with Gaussian noise with a

standard deviation σ_{syn}. Furthermore, 20 percent of the 2D feature points are disturbed with a very large offset to simulate outliers. The structure-from-motion algorithm is then applied with constrained and standard bundle adjustment for 50 times, each time with different randomly disturbed measurements.

Each resulting reconstruction is registered to the ground truth reconstruction by aligning both with an estimated similarity transformation.

In Fig. 5 the average absolute position error and average absolute rotation error for the estimated camera motion for different standard deviations are shown. It can be observed that the constrained always outperforms the standard bundle adjustment; e.g., for a standard deviation of $\sigma_{\text{syn}} = 1.0$ pixel, the average absolute position error is reduced by 30.0 percent and the average absolute rotation error by 38.7 percent.

6.2 Experiments with real scenes

The presented approach is applied to several real image sequences. These image sequences are first processed by a standard bundle adjustment, resulting in an unconstrained reconstruction as described in Sec. 3. Afterwards, the camera positions are obtained with the described detection and tracking algorithm of Sec. 4, and the constrained sparse bundle adjustment including an updated self-calibration is applied to the sequences.

In accordance with Eq. (3), two different error measures are introduced: The root-mean-squared residual error of the tracked camera centers

$$r_1 = \left(\frac{1}{C} \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{\tilde{n}=1}^{N} \mathrm{d}(\mathbf{c}_{k,n,\tilde{n}} \, , \, \mathtt{A}_{k,\tilde{n}} \, \mathbf{C}_{k,n})^2 \right)^{\frac{1}{2}} \tag{14}$$

with C the total number of all tracked camera centers, and the root-mean-squared residual error of the 2D feature points

$$r_2 = \left(\frac{1}{P} \sum_{n=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \mathrm{d}(\mathbf{p}_{j,k,n} \, , \, \mathtt{A}_{k,n} \, \mathbf{P}_j)^2 \right)^{\frac{1}{2}} \quad , \tag{15}$$

where P is the total number of all 2D feature points.

Obviously, the introduction of additional constraints restrains the bundle adjustment, so that a value for r_2, as obtained in the unconstrained case, can usually not be achieved in the constrained case. Therefore, if r_1 is significantly reduced and the value of r_2 increases only slightly, it can be assumed that a plausible solution was found.

Both r_1 and r_2 are first evaluated for the unconstrained reconstruction. Then the constrained bundle adjustment is applied and r_1 and r_2 are measured again.

In the following paragraphs results for three scenes are presented. The image sequences of these scenes have a resolution of 1440×1080 pixel and were recorded by 4 moving HDV consumer cameras. The lengths of the sequences in the first scene are 80 images per camera (320 images total), 400 images per camera for the second scene (1600 images total), and 400 images per camera for the third scene. The parameters $\sigma = 3$ pixel, $b = 2$ pixel, $t_0 = 0.8$, and $t_1 = 0.6$ are used for the tracking algorithm.

For the first scene, depicting a runner jumping over a bar, the weight factor $w = 0.1 \cdot P/C$ is used for the constrained bundle adjustment. The error measures for this scene can be found in Tab. 1 and the result is shown in Fig. 6.

Scene	"Running"		"Ramp"		"Statue"	
Method	r_1 [pixel]	r_2 [pixel]	r_1 [pixel]	r_2 [pixel]	r_1 [pixel]	r_2 [pixel]
unconstrained	16.85	1.67	99.37	0.77	123.78	0.88
constrained	6.26	1.75	5.92	0.98	4.43	0.93

Table 1. Results for r_1 and r_2 of the three scenes.

The second scene depicts a skateboard ramp. The wide baseline matching used for the generation of the unconstrained reconstruction finds mainly corresponding feature tracks at the center of the reconstruction volume. As can be verified in Fig. 7, this leads to acceptable results in the center of the reconstruction volume but results in large deviations at the borders of the reconstruction volume. This becomes evident because the overlay geometry in the center (green rectangle) does fit but the projections of the camera centers show large errors. In contrast, the constrained bundle adjustment with $w = 0.001 \cdot P/C$ is able to guide the estimate parameters to a solution, which generates plausible results for the whole reconstruction volume. In particular, the self-calibration benefits from improved estimates, as can be verified by the overlay geometry in Fig. 8, where the perpendicular structure is slightly off in the unconstrained reconstruction and fits well after the constrained bundle adjustment. Tab. 1 shows the results.

The third scene shows an art statue in a park. For this scene the automatic camera detection algorithm is employed, whereas for the two previous examples only the automatic tracking with manual initialization is used. The automatic detection works reliably and similar results as for the previous examples are achieved. Results for a weight factor $w = 0.001 \cdot P/C$ are shown in Figs. 8 and 9, and Tab. 1).

To evaluate the automatic detection algorithms, it is applied to all images of the sequence and the result is checked manually. In spite of severe illumination changes due to the grazing incidence of the sunlight, the detection algorithm reliably determines all camera center positions without producing any false positives.

7 Conclusion

An algorithm for multi-camera motion estimation is presented, which takes advantage of mutual visibility of cameras. An automatic detection and tracking algorithm using color patterns, NCC matching, and homography estimation is proposed that is capable of tracking the camera positions through the image sequences. Furthermore, a constrained bundle adjustment is introduced, which allows to include the additional constraints for the tracked camera centers. It is an extended version of the widely used sparse Levenberg-Marquardt algorithm for bundle adjustment. Despite the introduction of additional constraints, the sparse matrix structure of the equation systems is preserved, so that the computational effort does not increase.

Evaluations have been conducted on both, synthetic and real-world image sequences. On the synthetic sequences the average absolute error could be reduced by approximately 30 percent. For the real world image sequences, a very significant improvement of the estimated camera parameters could be observed. It turns out that the additional constraints minimize the geometric error at the boundary of the reconstructed volume and thereby can also ameliorate the self-calibration process.

An obvious drawback is the necessity of the cameras to be at least visible in a subsequence of frames, to allow our algorithm to generate estimation results with improved accuracy. However, during recording image sequences, it turned out to be quite hard to avoid situations where other cameras are visible. Therefore, the presented algorithm can find numerous applications in multi-camera computer vision. A current limitation is that the tracking algorithm does determine only the position of the camera lens and not the true mathematical camera center point. However, as the projection of the camera is small in the images, this is a good approximation.

Future work will address the automatic determination of the weighting factor w of the camera center constraints as well as the inclusion of other constraints to further improve the accuracy and robustness of camera motion estimation.

References

1. Gibson, S., Cook, J., Howard, T., Hubbold, R., Oram, D.: Accurate camera calibration for off-line, video-based augmented reality. In: ISMAR, Darmstadt, Germany (2002)
2. Hartley, R.I., Zisserman, A.: Multiple View Geometry. Cambridge University Press (2000)
3. Pollefeys, M., Gool, L.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. IJCV **59** (2004) 207–232
4. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M.: High performance imaging using large camera arrays. Proceedings of Siggraph 2005, ACM Trans. Graph. **24** (2005) 765–776
5. Rosenhahn, B., Schmaltz, C., Brox, T., Weickert, J., Cremers, D., Seidel, H.P.: Markerless motion capture of man-machine interaction. In: CVPR, Anchorage, USA (2008)
6. Sato, J.: Recovering multiple view geometry from mutual projections of multiple cameras. IJCV **66** (2006) 123ff
7. Jae-Hak, K., Hongdong, L., Hartley, R.: Motion estimation for multi-camera systems using global optimization. In: CVPR, Anchorage, AK, USA (2008)
8. Stewénius, H., Åström, K.: Structure and motion problems for multiple rigidly moving cameras. In: ECCV, Prague, Czech Republic (2004) 238ff
9. Frahm, J.M., Köser, K., Koch, R.: Pose estimation for multi-camera systems. In: 26th DAGM Symposium, Tübingen, Germany (2004) 27–35
10. Thormählen, T., Hasler, N., Wand, M., Seidel, H.P.: Merging of unconnected feature tracks for robust camera motion estimation from video. In: CVMP, London, UK (2008)
11. Triggs, B., Mclauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment – a modern synthesis. Lecture Notes in Computer Science **1883** (2000) 298ff
12. Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., Seidel, H.P.: Markerless motion capture with unsynchronized moving cameras. In: CVPR, Miami Beach, FL, USA (2009)

Fig. 6. Image sequence "Running" recorded with 4 moving cameras: Example images of camera 2 (the 4 leftmost images) and camera 4 (the 4 rightmost images) are presented. The two left images of camera 2 depict the path of camera 3 and 4 (in blue), prior to (left) and after constrained optimization (right), and the two left images of camera 4 depict the path of camera 2 in a similar fashion. The two right images show detail magnifications in each case. The detected camera positions are indicated by a red circle. Deviations of the estimated camera positions from the actual positions are depicted by red lines and are clearly visible in the magnifications.

Fig. 7. Image sequence "Ramp" recorded with 4 moving cameras: Example images of camera 1 and camera 3 are presented, showing the path of camera 3 and camera 1, respectively.

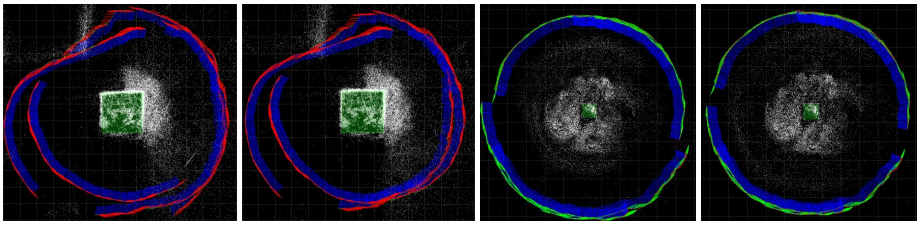

Fig. 8. Top views of the reconstructions for scenes "Ramp" (leftmost images) and "Statue" (rightmost images): Comparison between the unconstrained reconstruction (left) and the result of the constrained bundle adjustment (right). The estimated camera positions and orientations are depicted by small coordinate systems (optical axis, horizontal image direction, and vertical image direction, in blue, red, and green, respectively). The 3D object points are displayed as white dots.

Fig. 9. Image sequence "Statue" recorded with 4 moving cameras: Example images of camera 2 and camera 3 are presented, showing the path of camera 3 and camera 2, respectively.