Int J Comput Vis (2011) 91: 157–174
DOI 10.1007/s11263-010-0350-x

Omnidirectional Image Stabilization for Visual Object Recognition

Akihiko Torii · Michal Havlena · Tomáš Pajdla

Received: 4 May 2009 / Accepted: 29 April 2010 / Published online: 21 May 2010

Abstract In this paper, we present a pipeline for camera pose and trajectory estimation, and image stabilization and rectification for dense as well as wide baseline omnidirectional images. The proposed pipeline transforms a set of images taken by a single hand-held camera to a set of stabilized and rectified images augmented by the computed camera 3D trajectory and a reconstruction of feature points facilitating visual object recognition. The paper generalizes previous works on camera trajectory estimation done on perspective images to omnidirectional images and introduces a new technique for omnidirectional image rectification that is suited for recognizing people and cars in images. The performance of the pipeline is demonstrated on real image sequences acquired in urban as well as natural environments.

Keywords Omnidirectional vision · Structure from motion · Image rectification · Object recognition

1 Introduction

Image stabilization using camera poses and trajectory estimated by reliable structure from motion (SfM) plays an important role in 3D reconstruction (2d3. Boujou 2001; Hartley and Zisserman 2003; Akbarzadeh et al. 2006; Cornelis et al. 2006; Davison and Molton 2007; Williams et al. 2007), self localization (Goedemé et al. 2007), and reducing the number of false alarms in detection and recognition of pedestrians, cars, and other objects in video sequences (Hoiem et al. 2006; Leibe et al. 2007a, 2007b; Torii et al. 2008).

Contrary to existing SfM algorithms, which solve the problem when the camera motion is small or once the 3D structure is initialized, we aim at a more general situation when neither the relationship between the cameras nor the 3D structure is available. In such case, 2-view camera matching and relative motion estimation is a natural starting point to camera tracking and structure from motion. This is an approach used by the state of the art wide baseline structure from motion algorithms, *e.g.* Brown and Lowe (2003), Snavely et al. (2006), Martinec and Pajdla (2007), Snavely et al. (2008), Microsoft (2008), that start with pairwise image matches and epipolar geometries which they next clean up and make consistent by a large scale bundle adjustment.

The state of the art wide baseline SfM methods often work with perspective cameras because of the simplicity of their projection models and the ease of their calibration. On the other hand, due to the limited field of view of perspective cameras, occlusions and sharp turns of the camera may cause consecutive frames to look completely different when the baseline becomes longer or the change of the view direction becomes larger. These make image feature matching very difficult (or even impossible) and camera pose and trajectory estimation fails under such conditions. These problems can be avoided if the SfM method uses omnidirectional cameras, *e.g.* fish-eye lens convertors (Mičušík and Pajdla 2006), catadioptric cameras (Geyer and Daniilidis 2001; Mičušík and Pajdla 2006), or compound cameras (Scara-

Electronic supplementary material The online version of this article (doi:10.1007/s11263-010-0350-x) contains supplementary material, which is available to authorized users.

A. Torii (✉) · M. Havlena · T. Pajdla
Center for Machine Perception, Department of Cybernetics,
Faculty of Elec. Eng., Czech Technical University in Prague,
Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
e-mail: torii@cmp.felk.cvut.cz

M. Havlena
e-mail: havlem1@cmp.felk.cvut.cz

T. Pajdla
e-mail: pajdla@cmp.felk.cvut.cz

muzza et al. 2008; Tardif et al. 2008). Large field of view also facilitates the analysis of activities happening in the scene since moving objects can be tracked for longer time periods (Leibe et al. 2007b).

The closest related SfM approach (Tardif et al. 2008) employs guided matching by using epipolar geometries computed in previous frames and estimates camera trajectory robustly by computing camera orientations and positions individually. The performance of their SfM is demonstrated on sufficiently dense image sequences acquired by a car-mounted Ladybug 2 spherical camera (Point Grey Research 2005).

The main contribution of this paper is to present an integrated pipeline for camera pose and trajectory estimation followed by image stabilization and rectification for dense as well as wide baseline omnidirectional images acquired by a single hand-held camera. Our wide baseline SfM is capable of recovering camera poses and trajectories from sequences having large and non-smooth camera motions between consecutive frames. Therefore, the recovery can be accomplished even from sequences in which some frames are contaminated by unexpected accidents, *e.g.* blurred images, extreme change of the view direction, and lack of features to match. Furthermore, we show that the proposed approach is capable of facilitating visual object recognition by using the stabilized and rectified images augmented by the computed camera trajectory and the 3D reconstruction of the detected feature points.

There are some essential issues for reliable camera pose and trajectory estimation:

- The choice of camera, its geometric projection model, and a suitable calibration technique (Sect. 2.1).
- Image feature detection, description (Sect. 2.2), and robust relative motion estimation (Sect. 2.3).
- Robust 3D structure computation (Sects. 3 and 4).
- The choice of a suitable omnidirectional image stabilization and rectification method (Sect. 5).

Moreover, the pipeline has a natural capability to deal with unorganized images, regarding them as a sequence after sorting them by an image indexing method based on visual words and visual vocabulary (Sivic and Zisserman 2006; Knopp et al. 2009), as described in Sect. 6.2.

Robust Estimation of Relative Camera Poses The state of the art technique for finding relative camera poses from image matches first establishes tentative matches by pairing image points with mutually similar features and then uses RANSAC (Fischler and Bolles 1981; Hartley and Zisserman 2003; Chum and Matas 2005) to look for a large subset of the set of tentative matches which is, within a predefined threshold ε, consistent with an epipolar geometry (EG) (Hartley and Zisserman 2003). Unfortunately, this strategy does not always recover the epipolar geometry generated by the actual camera motion, which has been observed in Li and Hartley (2005), Nistér and Engels (2006), Torii and Pajdla (2008).

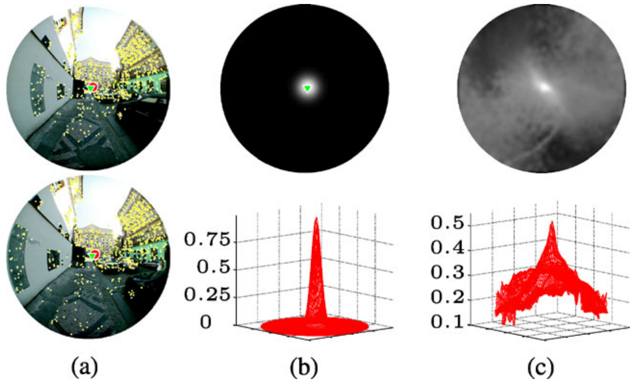


Fig. 1 (Color online) Easy camera motions. (**a**): The first image (*top*) and the second image (*bottom*). *Red* ○ and *green* ▽ show the true epipoles and the epipoles computed by soft voting for the position of the epipole, respectively. *Small dots* show the matches giving *green* ▽. (**b**): Voting space for the motion direction in the first image generated by 50 soft votes cast by the 500-sample PROSAC, visualized on the image plane (*top*) and as a 3D plot (*bottom*). White color corresponds to a large number of votes. The peak corresponds to the *green* ▽. (**c**): The maximal support for every possible epipole (*i.e.* CIF image from Nistér and Engels (2006)). *White color* corresponds to high support. The image space has been uniformly sampled by 10,000 epipoles and the size of the support of the best model found by the 500-sample PROSAC has been recorded for each epipole

It has been demonstrated in Chum and Matas (2005) that ordering the tentative matches by their similarity may help to reduce the number of samples in RANSAC. PROSAC (Chum and Matas 2005) sampling strategy has been suggested which allows uniform sampling from the list of tentative matches in descending order by the similarity of their descriptors. The promising samples are drawn first which often leads into hitting a sufficiently large configuration of good matches early.

Often there are several models that are supported by a large number of matches. The chance that the correct model will be found by running a single RANSAC is then small, even when it has the largest support. Work of Li and Hartley (2005) suggested to generate models by randomized sampling as in RANSAC but to use soft (kernel) voting for a physical parameter, the radial distortion coefficient in their case, instead of looking for the maximal support. The best model is then selected as the one with the parameter closest to the maximum in the accumulator space. This strategy works when the correct, or almost correct, models provide consistent values of the parameter while the incorrect models with high support generate different values. Here, as in Nistér and Engels (2006), we show that this strategy works also when used for voting in the space of motion directions. To illustrate the problem, we shall now discuss two inter-

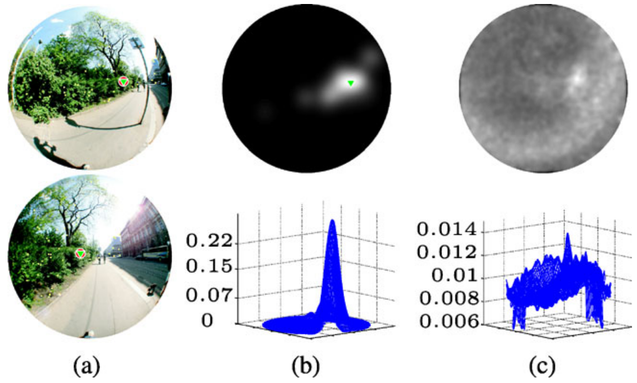


Fig. 2 Difficult camera motions. See Fig. 1 for description. Notice that the true motion has the highest support but its peak is very sharp and thus difficult to find in limited time

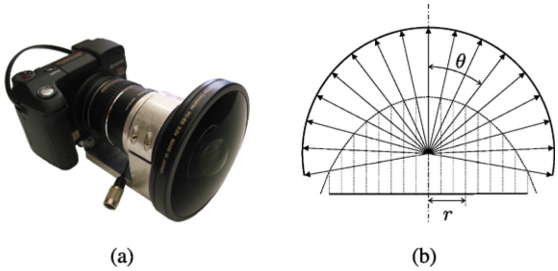


Fig. 3 (**a**) Kyocera Finecam M410R camera and Nikon FC-E9 fish-eye lens convertor. (**b**) The equi-angular projection model. The angle θ between the casted ray of a 3D point and the optical axis can be computed from the radius r of a circle in the image circular view field

esting examples of camera motions which have a gradually increasing level of difficulty.

Figure 1(a) shows an easy pair of images which can be solved by a standard RANSAC estimation (Hartley and Zisserman 2003). 57%, *i.e.* 1,400, of tentative matches are consistent with the true motion. Figure 1(c) shows that there is a dominant peak in the data likelihood $p(M|\mathbf{e})$ of the matches given the motion direction (Nistér and Engels 2006), meaning that there is only one motion direction which explains a large number of matches. Figure 1(b) shows the voting space for the motion direction in the first image generated by 50 soft votes cast by the result of a 500-sample PROSAC, visualized on the image plane (top) and as a 3D plot (bottom). White represents a large number of votes. The peak corresponds to the green $\triangledown$.

Figure 2(a) shows a difficult pair of images since only 1.4%, *i.e.* 50, tentative matches are consistent with the true motion. There are many wrong tentative matches on the bushes where nearly all the local image features are small and green. Thus many motion directions get high support from wrong matches. The true motion has the highest support but its peak is very sharp and thus difficult to find in limited time. Even this difficult example can be solved correctly by the technique presented in Sect. 2.

Robust SfM by Detecting Too Small Translations The problem of detecting too small translation in structure from motion has been addressed in Martinec and Pajdla (2007). Camera motions were considered pure rotations if at least 90% of matches verified by an epipolar geometry were also verified by fitting a pure rotation. Another recent work (Clipp et al. 2008) looks at a related problem of determining the scale of the motion of a stereo rig with non-overlapping fields of view.

In Sect. 3, we propose a method providing a reliable detection of too small camera translation from two images and demonstrate that such capability enhances SfM and object recognition from a video sequence taken by a moving camera. Since the scale of the reconstruction cannot be determined from two images acquired by a moving camera, the amount of camera translation can be measured only relatively w.r.t. the observed scene. We use the dominant apical angle (DAA) (Torii et al. 2008) of the 3D points reconstructed from the matches for measuring the amount of camera translation from pairwise image matches.

The apical angle of a 3D point $\mathbf{X}$ is the angle under which the camera centers are seen from the perspective of the point $\mathbf{X}$. We show on simulated data that the dominant apical angle is a linear function of the length of the true translation for general as well as planar scenes and that it can be reliably estimated in the presence of outliers. Furthermore, we demonstrate in real experiments that the proposed measure enables robust computation of camera poses and trajectory even from sequences acquired with the presence of large changes of motion acceleration.

Hereafter, we describe the details of our pipeline with some illustrative examples.

2 Robust Estimation of Relative Camera Motion

2.1 Camera Calibration

The setup used in this work is a combination of a Nikon FC-E9 lens, mounted via a mechanical adaptor, and a Kyocera Finecam M410R digital camera, see Fig. 3(a). Nikon FC-E9 is a megapixel omnidirectional add-on convertor with 183° view angle which provides high-quality images. Kyocera Finecam M410R delivers 2,272 × 1,704 pixels large images at 3 frames per second. The resulting combination yields a circular view of the diameter slightly under 1,600 pixels in the image.

The calibration of omnidirectional cameras is non-trivial but crucial for achieving good accuracy of the resulting 3D reconstruction. We calibrate our camera off-line using the state of the art technique (Bakstein and Pajdla 2002) and

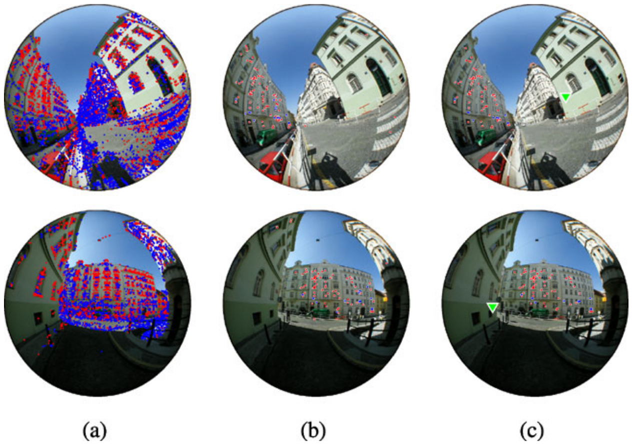


Fig. 4 (Color online) Wide baseline image matching. The colors of the *dots* correspond to the detectors (*red*) MSER-Intensity+ and (*blue*) MSER-Intensity−. (**a**) All detected features. (**b**) Tentative matches constructed by selecting pairs of features which have the mutually closest descriptors. (**c**) The epipole (*green* ▽) computed by maximizing the support. Note that the scene dominated by a single plane does not induce degeneracy in computing calibrated epipolar geometry by solving the 5-point minimal relative pose problem

Mičušík's two-parameter model (Mičušík and Pajdla 2006), that links the radius of the image point r to the angle θ of its corresponding rays w.r.t. the optical axis, see Fig. 3(b) as

$$\theta = \frac{ar}{1 + br^2}. \tag{1}$$

After a successful calibration, we know the correspondence of the image points to the 3D optical rays in the coordinate system of the camera. The following steps aim at finding the transformation between the camera and the world coordinate systems, *i.e.* the pose of the camera in the 3D world, using 2D image matches.

2.2 Detecting Features and Constructing Tentative Matches

For computing the 3D structure, a set of tentative matches is constructed by detecting image features. We have tested several feature detectors: Maximally Stable Extremal Regions (MSER) (Matas et al. 2004), Laplacian-Affine, Hessian-Affine (Mikolajczyk et al. 2005), Scale-Invariant Feature Transform (SIFT) (Lowe 2004), and Speeded Up Robust Features (SURF) (Bay et al. 2008). We can conclude that the choice of the feature detector is not crucial for the resulting 3D models. We use MSER and SIFT since they have potential to match features under large changes of view direction and are more efficient than the features from Mikolajczyk et al. (2005). Parameters of the detectors were chosen to limit the number of regions to 1–2 thousands per image. For MSER, the detected regions are assigned Local Affine Frames (LAF) (Obdržálek and Matas 2002) and transformed into the standard positions w.r.t. their LAFs. Discrete Cosine Transform (DCT) descriptors (Obdržálek and Matas 2003) are computed for each region in the standard position. For SIFT, keypoints are detected based on the Difference of Gaussians (DoG) and SIFT keypoint descriptors are created from sets of histograms of the gradient information computed from the neighbors of the keypoints.

Finally, tentative matches are constructed by searching the mutually closest descriptors between the given images. We use Fast Library for Approximate Nearest Neighbors (FLANN) (Muja and Lowe 2009) which performs approximate nearest neighbors search based on random kd-trees. Figures 4(a) and (b) show two examples of feature detection and matching for pairs of wide baseline images.

When all camera motions between consecutive frames are small and moderate, short baseline matching using simpler image features (Cornelis et al. 2006; Havlena et al. 2009) can be used efficiently under assumptions on the proximity of the consecutive projections. However, in practical situations, some frames may be contaminated or lost by unexpected accidents, *e.g.* an extremely fast camera movement, while acquiring a long sequence. The view point and direction can change a lot between the usable consecutive frames and the short baseline matching often fails. By using wide baseline matching, one can handle such situations as it is possible to make a link between the non-contaminated frames.

2.3 Epipolar Geometry Computation by RANSAC+ Soft-voting

3D structure can be robustly computed by RANSAC (Fischler and Bolles 1981) which searches for the largest subset of the set of tentative matches which is, within a predefined threshold ε, consistent with an epipolar geometry (Hartley and Zisserman 2003). We use ordered sampling as suggested in Chum and Matas (2005) to draw 5-tuples from the list of tentative matches which may help to reduce the number of samples in RANSAC. From each 5-tuple, relative pose is computed by solving the 5-point minimal relative pose problem for calibrated cameras (Nistér 2004a; Stewénius 2005). Figure 4(c) shows the results of computing the epipolar geometry for two pairs of wide baseline images.

Ordered Randomized Sampling Samples are drawn from tentative matches ordered ascendingly by the distance of their descriptors as suggested in Chum and Matas (2005). On the other hand, we keep the original RANSAC stopping criterion (Hartley and Zisserman 2003) and limit the maximum number of samples to 1,000. We have observed that pairs which could not be solved in 1,000 samples got almost never solved even after many more samples. Using the stopping criterion from Chum and Matas (2005) often leads to ending the sampling prematurely since the criterion is designed to stop as soon as a large non-random set of matches is found. Our objective is, however, to find a globally good model and not to stop as soon as a local model having a sufficiently large support is found.

Orientation Constraint A given essential matrix can be decomposed into four different camera and point configurations which differ by the orientations of the cameras and points (Hartley and Zisserman 2003, p. 260). Without enforcing the constraint that all points have to be observed in front of the cameras, some epipolar geometries may be supported by many matches but it may not be possible to reconstruct all points correctly, *i.e.* in front of both cameras.

A point $\mathbf{X}$ is in front of the perspective camera when it has a positive z coordinate in the camera coordinate system. For omnidirectional cameras, the meaning of 'in front' is a generalization of the classical case for perspective cameras: a point $\mathbf{X}$ is in front of the camera if its coordinates can be written as a positive multiple of the direction vector which represents the half-ray by which $\mathbf{X}$ has been observed.

In general, it is beneficial to use only the matches which generate points in front of the cameras. However, it takes time to verify this for all matches. On the other hand, it is fast to verify whether the five points in the minimal sample generating the epipolar geometry can be reconstructed in front of both cameras and to reject such epipolar geometries which do not allow it. Furthermore, the orientation constraint on average reduces the computational cost because it avoids evaluating the residuals corresponding to many incorrectly estimated camera motions.

Soft Voting In this paper, we vote in a two-dimensional accumulator for the estimated motion direction. However, unlike in Li and Hartley (2005), Nistér and Engels (2006), we do not cast votes directly by each sampled epipolar geometry but by the best epipolar geometries recovered by the ordered sampling of PROSAC. This way the votes come only from the geometries that have a very high support. We can afford to compute more, *e.g.* 5, epipolar geometries since the ordered sampling is much faster than the standard RANSAC. Altogether, we need to evaluate maximally 1,000 × 5 = 5,000 samples to generate 5 soft votes, which is comparable to running a standard 5-point RANSAC for the expected contamination by 77% of mismatches (Hartley and Zisserman 2003, p. 119). Yet, with our technique, we could go up to 98.5% of mismatches with a comparable effort. Finally, the relative camera pose with the motion direction closest to the maximum in the voting space is selected.

The proposed robust estimation of relative camera motion is summarized as the pseudocode in Algorithm 1 with the actual parameters used in the real experiments.

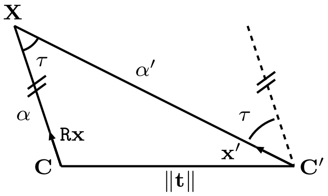


Fig. 5 The apical angle τ at the point $\mathbf{X}$ reconstructed from the correspondence $(\mathbf{x}, \mathbf{x}')$ relatively depends on the length of the camera translation $\mathbf{t}$ and on the distances of $\mathbf{X}$ from the camera centers $\mathbf{C}$, $\mathbf{C}'$

3 Measuring the Amount of Camera Translation by the Dominant Apical Angle

Consider a pair of calibrated cameras with the normalized camera matrices (Hartley and Zisserman 2003), $\mathtt{P} = [\mathtt{I}|0]$ and $\mathtt{P}' = [\mathtt{R}| - \mathbf{t}]$ and an image point correspondence given by a pair of homogeneous coordinates $(\mathbf{x}, \mathbf{x}')$ represented by unit direction vectors, *i.e.* $\|\mathbf{x}\| = \|\mathbf{x}'\| = 1$. There holds

$$\alpha'\mathbf{x}' = \alpha\mathtt{R}\mathbf{x} - \mathbf{t}, \tag{2}$$

with real α, α', rotation $\mathtt{R}$ and translation $\mathbf{t}$.

Should there be no noise then pure camera rotation, *i.e.* $\|\mathbf{t}\| = 0$, could be detected by finding out that $\mathbf{x}' = \mathtt{R}\mathbf{x}$ holds true for all the correspondences. However, this does not occur, even when the physical camera really does rotate, due to noise in image measurements. Thus, in real situations, a non-zero essential matrix $\mathtt{E}$ can always be computed from noisy image matches, *e.g.*by the 5-point algorithm (Nistér 2004a).

Having n matches $\{(\mathbf{x}_i, \mathbf{x}_i')\}_{i=1,\dots,n}$ and the essential matrix $\mathtt{E}$ computed from them, we can reconstruct n 3D points $\{\mathbf{X}_i\}_{i=1,\dots,n}$. Figure 5 shows a point $\mathbf{X}$ reconstructed from an image match $(\mathbf{x}, \mathbf{x}')$. For each point $\mathbf{X}$, the apical angle τ, which measures the length of the camera translation from the perspective of the point $\mathbf{X}$, is computed. If the cameras are related by pure rotation, all angles τ are equal to zero. The larger is the camera translation, the larger are the angles τ. The closer is the point $\mathbf{X}$ to the midpoint of the camera baseline, the larger is the corresponding τ. In fact, measuring the apical angles is equivalent to measuring disparities on a spherical retina as the corresponding angle, *i.e.* the apical angle τ is easily computed with relative rotation $\mathtt{R}$ such that

$$\tau = \measuredangle(\mathtt{R}\mathbf{x}, \mathbf{x}'). \tag{3}$$

For a given $\mathtt{E}$ and matches $\{(\mathbf{x}_i, \mathbf{x}_i')\}_{i=1,\dots,n}$, one can select the decomposition of $\mathtt{E}$ to $\mathtt{R}$ and $\mathbf{t}$, which reconstructs the largest number of 3D points in front of the cameras. The apical angle τ_i, corresponding to the match $(\mathbf{x}_i, \mathbf{x}_i')$, is computed by solving a set of linear equations for the relative distances α_i, α_i'

$$\alpha_i'\mathbf{x}_i' = \alpha_i\mathtt{R}\mathbf{x}_i - \mathbf{t} \tag{4}$$

Algorithm 1 Robust estimation of relative camera motion

```
Input   Image pair $I_1, I_2$.
        $N_V := 5$... the number of soft votes.
        $N_T := 1000$... the maximum number of random samples.
        $\varepsilon := 0.1°$... the tolerance for establishing matches.
        $\sigma := 0.4°$... the standard deviation of Gaussian kernel for soft voting.
Output  Relative camera motion $\mathtt{E}^*$ and its supports $M^*$.

1. Detect features and compute descriptors, (MSER-INT±, LAF+DCT) (Obdržálek and Matas 2002, 2003) and
   (SIFT) (Lowe 2004).
2. Construct the list $M = [\mathbf{m}]_1^N$ of tentative matches with mutually closest descriptors (Chum and Matas 2005). Order the
   list ascendingly by the distance of the descriptors. $N$ is the length of the list.
3. Find a camera motion consistent with a large number of tentative matches (Torii and Pajdla 2008):
   Set $D$ to zero. // Initialize the accumulator of camera translation directions.
   for $i := 1, \ldots, N_V$ do
     $t := 0$ // The counter of samples.
     while $t \le N_T$ do
       $t := t + 1$ // New sample.
       Select the 5 tentative matches $M_5$ of the $t^{\text{th}}$ sample from the ordered list $M$ (Chum and Matas 2005)
       $\mathtt{E}_t :=$ the essential matrix by solving the 5-point minimal problem for $M_5$ (Nistér 2004a; Stewénius 2005).
       if $M_5$ can be reconstructed in front of the cameras (Hartley and Zisserman 2003, p. 260) then
         $S_t :=$ the number of matches which are consistent with $\mathtt{E}_t$, i.e. the number of all matches $\mathbf{m} = [\mathbf{u}_1, \mathbf{u}_2]$ for
         which $\max(\measuredangle(\mathbf{u}_1, \mathtt{E}_t\mathbf{u}_2), \measuredangle(\mathbf{u}_2, \mathtt{E}_t^\top\mathbf{u}_1)) < \varepsilon$.
       else
         $S_t := 0$
       end if
       $N_R := \log(\eta)/\log(1 - \binom{S_t}{5}/\binom{N}{5})$ // The termination length defined by the maximality constraint (Hartley and
       Zisserman 2003, p. 119).
       $N_T := \min(N_T, N_R)$ // Update the termination length.
     end while
     $\hat{t} = \arg_{t=1,\ldots,N_T} \max S_t$ // The index of the sample with the highest support.
     $\hat{\mathtt{E}}_i := \mathtt{E}_{\hat{t}}$, $\hat{\mathbf{e}}_i :=$ camera motion direction for the essential matrix $\mathtt{E}_{\hat{t}}$.
     Vote in accumulator $D$ by the Gaussian with sigma $\sigma$ and the mean at $\hat{\mathbf{e}}_i$.
   end for
   $\hat{\mathbf{e}} := \arg_{\mathbf{x} \in domain(D)} \max D(\mathbf{x})$ // Maximum in the accumulator.
   $i^* := \arg_{i=1,\ldots,50} \min \measuredangle(\hat{\mathbf{e}}, \hat{\mathbf{e}}_i)$ // The motion closest to the maximum.
   $\mathtt{E}^* := \hat{\mathtt{E}}_{i^*}$ // The "best" camera motion.
   $M^* := [\mathbf{m}^*]_1^{N^*}$ // The inlier matches supporting $\mathtt{E}^*$. $N^*$ is the number of the inlier matches.
4. Return $\mathtt{E}^*$ and $M^*$.
```

in the least square sense and by using the law of cosines

$$2\alpha_i \alpha_i' \cos(\tau_i) = {\alpha_i}^2 + {\alpha_i'}^2 - \|\mathbf{t}\|^2. \qquad (5)$$

For a small translation w.r.t. the distance to the scene points, it is natural to use the approximation $\alpha_i = \alpha_i'$. Then, the apical angle τ_i becomes a linear function of $\|\mathbf{t}\|$. This is instantly proven by using the approximated equation of the law of cosines

$$\cos(\tau_i) = 1 - \frac{\|\mathbf{t}\|^2}{2\alpha_i^2} \qquad (6)$$

and the cosine series expansion

$$\cos(\tau_i) = 1 - \frac{\tau_i^2}{2!} + \mathrm{O}(\tau_i^4). \qquad (7)$$

If all matches were correct, the largest τ would best represent the amount of the translation. However, all matches are rarely correct and thus we need to find a robust measure of the translation. The distribution of the values of τ_i depends on the distribution of the points in the scene and on mismatches, if present. We have observed that for many

Algorithm 2 Measuring camera motion by computing the dominant apical angle

```
Input   The relative camera motion E and its supports $\{\mathbf{m}_i\}_{i=1}^N$
        $\sigma := 0.4°$ ... the standard deviation of Gaussian kernel for soft voting.
Output  Dominant apical angle $\tau^*$.

Decompose E into the rotation R and the translation t (Hartley and Zisserman 2003, p. 260).
for $i := 1, \ldots, N$ do
    Compute the apical angle $\tau_i$ from the match $\mathbf{m}_i$, R and t (see Sect. 3).
end for
Compute the 10th percentile $q^{10}$ and the 90th percentile $q^{90}$ from $[\tau]_1^{N^*}$. // Lower and upper bounds on apical angles to exclude outliers.
for $i := 1, \ldots, N^*$ do
    if $q^{10} < \tau_i < q^{90}$ then
        Vote in accumulator $B$ by the Gaussian with sigma $\sigma$ and the mean at $\tau_i$.
    end if
end for
$\tau^* := \arg_{y \in domain(B)} \max B(y)$ // Maximum in the accumulator.
Return $\tau^*$.
```

general 3D as well as planar scenes, the distribution has a dominant mode

$$\tau^* = \arg_{\{\tau_i\}_{i=1}^n} \max \; g(\tau_i) \tag{8}$$

where $g(\tau)$ performs kernel voting with Gaussian smoothing (Li and Hartley 2005), and that the mode τ^* predicts the length of the translation well. The pseudo code of DAA computation is listed in Algorithm 2.

3.1 Too Small Motion Detection on Simulated Data

Figures 6, 7, and 8 show the results of simulated experiments for three different scenes, different motion directions, and for the length of the translation increasing from zero to a large value. The amount of camera translation was computed by the method based on RANSAC (Fischler and Bolles 1981), which is described in Algorithm 1. Notice that we use a combination of ordered sampling (Chum and Matas 2005) with kernel voting to maximize the chance of recovering the correct epipolar geometry (Torii and Pajdla 2008). We also enforce the reconstructed points to be in front of both cameras before counting the support size in RANSAC.

Figure 6 shows an experiment with a general 3D scene consisting of 1,000 points uniformly distributed in a hemisphere with the center at $(0, 0, 10)^\top$ and radius 25, see

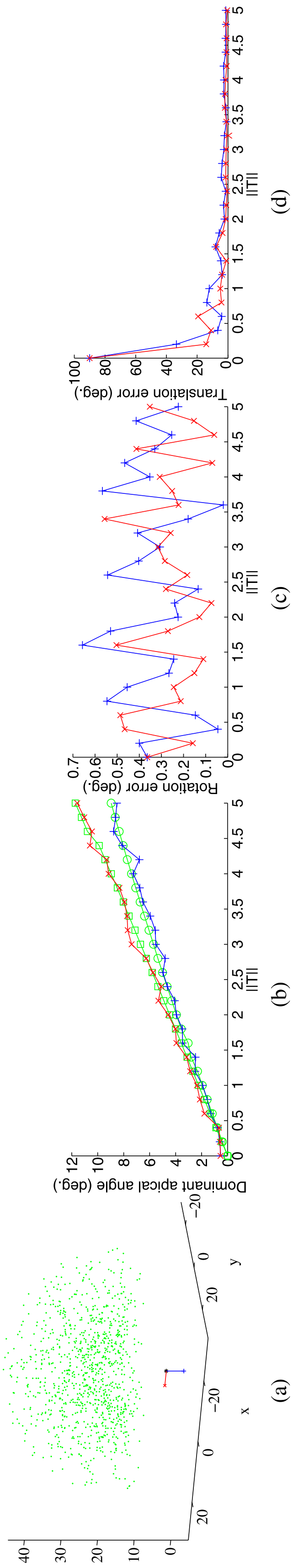


Fig. 6 (Color online) Measuring the length of the camera translation for a general 3D scene. **(b)** Dominant apical angle. Noisy data: *blue line* with "+" markers—backward motion, *red line* with "×" markers—lateral motion; exact data: *green lines* with "○" and "□"—backward and lateral motions, respectively. **(c)** Camera rotation error. **(d)** Camera translation error

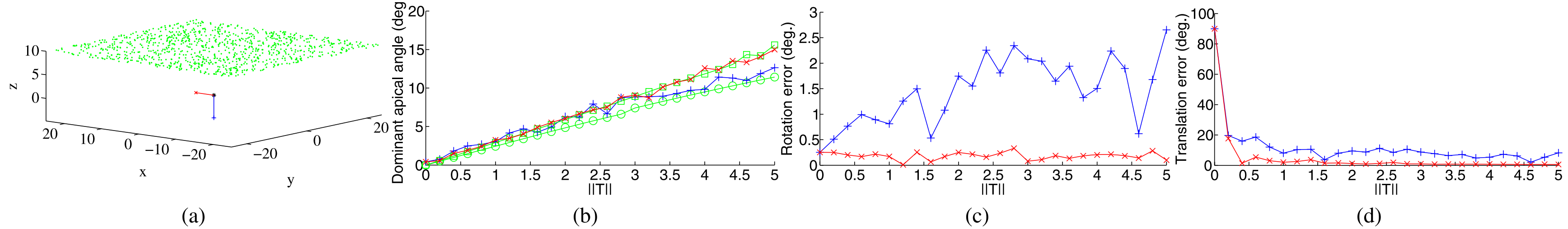


Fig. 7 (Color online) Measuring the length of the camera translation for a planar scene. (**b**) Dominant apical angle. Noisy data: *blue line* with "+" markers—backward motion, *red line* with "×" markers—lateral motion; exact data: *green lines* with "○" and "□"—backward and lateral motions, respectively. (**c**) Camera rotation error. (**d**) Camera translation error

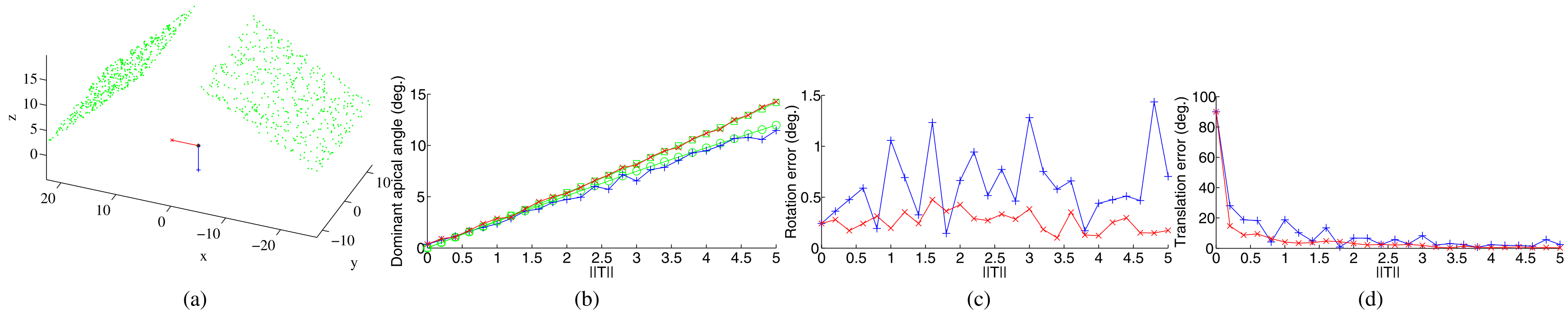


Fig. 8 (Color online) Measuring the length of the camera translation for the scene consisting of two planes. (**b**) Dominant apical angle. Noisy data: *blue line* with "+" markers—backward motion, *red line* with "×" markers—lateral motion; exact data: *green lines* with "○" and "□"—backward and lateral motions, respectively. (**c**) Camera rotation error. (**d**) Camera translation error

Fig. 6(a). The first camera was placed at $\mathbf{T}_1 = (0,0,0)^\top$ looking towards the scene points. Two motions of the second camera were tested. The backward motion was constructed as $\mathbf{T}_2 = (0,0,-s)^\top$, *i.e.* the camera was moving away from the scene. The sideways motion was constructed as $\mathbf{T}_3 = (s,0,0)^\top$. In both cases, s ranged from 0 to 5.

3D points were projected by normalizing their coordinate vectors, constructed w.r.t. the respective camera coordinate systems, to unit length. To simulate the imprecision due to image sampling during digitization and image measurement, Gaussian noise with a standard deviation $\sigma = 3^\circ$, corresponding to 1.3 pixels in a 800 × 800 pixels large image capturing 180° field of view, was added to the normalized vectors.

Figure 6(b) shows the dominant apical angle (DAA) as a function of the length of the true translation. DAA for the backward motion is shown by the blue line with "+" markers, whereas DAA for the lateral motion is shown by the red line with "×" markers, both computed from noisy measurements. The green lines with "∘" and "□" markers, respectively, show the respective DAA of the backward and lateral motions computed from exact measurements. We see that the DAA is a linear function of the length of the true motion for translations longer than 0.25 meters. The slope of the lateral DAA is slightly larger (2.5°/meter) than the slope of DAA for the backward motion (2.0°/meter) in this case. DAA of the zero translation computed from noisy matches is slightly above the zero due to noise in image measurements. Figure 6(c) shows the difference in the estimated camera rotation $\mathtt{R}_{est}$ w.r.t. the true rotation $\mathtt{R}$ evaluated as the angle of rotation of $\mathtt{R}_{est}^{-1}\mathtt{R}$. Notice that the error is constant for all lengths of the translation which shows that the rotation is computed correctly even if the direction of the translation, Fig. 6(d), cannot be found reliably.

Figures 7 and 8 show the same experiment as above on a planar scene and on a 3D scene consisting of two planes. The results are comparable to those shown in Fig. 6. In particular, we can see that we are able to measure the amount of translation in all three cases. It is interesting to notice that the error in rotation is constant for general 3D scenes, Figs. 6(c) and 8(c), but grows linearly for the planar scene, Fig. 7(c). This reflects the fact that the angle which is occupied by scene points determines, to a large extent, the quality of rotation estimation from scenes with shallow depth. At the same time, we can see that the quality of estimating the amount of camera translation has not been affected.

Algorithm 3 Keyframe selection

Input N images I_t.
$\eta\ldots$ the minimum amount of translation.
Output Flags k_t initialized as all FALSE.

```
$i := 1, k_i :=$ TRUE
while $i < N$ do
    $j := 0, q := 1$
    while $q = 1 \;\wedge\; (i + j < N)$ do
        $j := j + 1$
        Compute the relative motion $E_{i,i+j}$ between $I_i$ and $I_{i+j}$.
        $N_s :=$ number of supports of $E_{i,i+j}$.
        $\tau^* :=$ DAA computed from $E_{i,i+j}$ and its supports.
        $\omega^* :=$ sum of the weighted apical angles computed from $E_{i,i+j}$ and its supports.
        $q := (\tau^* < \eta) \;\wedge\; (\omega^* < N_s)$
    end while
    if $q = 0$ then
        $i := i + j$
        $k_i :=$ TRUE
    end if
end while
```

4 Sequential Wide Baseline Structure from Motion

Camera poses in a canonical coordinate system are recovered by chaining the EGs of pairs of consecutive images in a sequence. The essential matrix $\mathtt{E}_{ij}$ encoding the relative camera pose between frames i and $j = i + 1$ can be decomposed into $\mathtt{E}_{ij} = [\mathbf{t}_{ij}]_\times \mathtt{R}_{ij}$. Although there exist four possible decompositions, the right one can be selected as that which reconstructs the largest number of 3D points in front of both cameras. Having the normalized camera matrix (Hartley and Zisserman 2003) of the i-th frame $\mathtt{P}_i = [\mathtt{R}_i|\mathbf{t}_i]$, the normalized camera matrix $\mathtt{P}_j$ can be computed by

$$\mathtt{P}_j = [\mathtt{R}_{ij}\mathtt{R}_i|\mathtt{R}_{ij}\mathbf{t}_i + \gamma_{ij}\ \mathbf{t}_{ij}] \tag{9}$$

where γ_{ij} is the scale of the translation between frames i and j in the canonical coordinate system. This scale can be computed from any 3D point seen in at least three consecutive frames but the precision depends on the uncertainty of the reconstructed 3D point. Therefore, a robust selection from the possible candidates of the scales has to be done while evaluating the quality of the computed camera pose. The best scale is found by RANSAC maximizing the number of points that pass the "cone test" (Havlena et al. 2009) which checks the intersection of pixel ray cones, *i.e.* the feasibility test of L_1- or L_∞- triangulation (Kahl 2005; Ke and Kanade 2007). During the cone test, quarter-pixel wide cones formed by four planes (up, down, left, and right) are cast around the matches and we test whether the intersection of the cones is empty or not.

Contrary to standard sequential SfM techniques, which compute camera translation and rotation from the estimated 3D point cloud, we compute camera rotation and translation from EG, as it has been shown in Tardif et al. (2008) that computing camera rotation from EG is more accurate. Contrary to the decoupled SfM technique presented in the aforementioned paper, the maximum number of samples in our scale selection is bounded by the number of 3D points seen in three consecutive frames because we do not need to draw pairs of 2D-3D matches to compute the camera translation.

On the other hand, when also recovering the camera translation from EG, one must take care whether or not a particular EG contains a sufficient amount of translation. EGs inaccurately computed from image pairs having too small translation disturb the chaining of camera poses and do not contribute by reconstructing new 3D points in the scene. It is also important that a sufficient number of 3D points with large apical angles exists in the pairwise reconstruction for obtaining accurate scales when chaining the EGs. For producing the EGs capable of stable recovery of camera poses and trajectory, we propose to use only the images that satisfy one of the two quality scores computed between the pairs of consecutive frames. We will call such images keyframes.

One of the quality scores is the DAA τ^*, which has been described in Sect. 3 already. Setting the minimum amount of DAA to 0.2°–2° enables the detection of too small translation. The other quality score ω^* is the sum of the weighted scores computed based on apical angles. The weighted score ω for the apical angle τ of a 3D point $\mathbf{X}$ is defined by the following formula:

$$\omega = q_1 + 4q_2 + 20q_3, \tag{10}$$

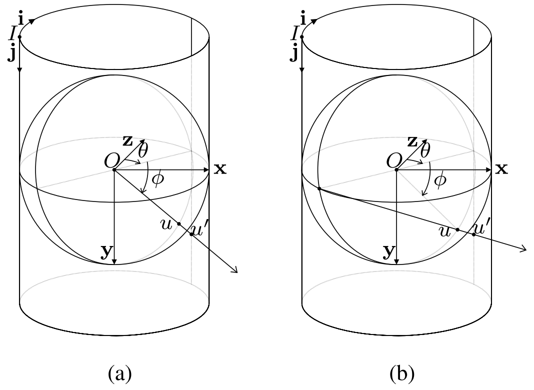
$$q_1 = \begin{cases} 1 & \tau \geq 5^\circ \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$q_2 = \begin{cases} 1 & \tau \geq 10^\circ \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$q_3 = \begin{cases} 1 & \tau \geq 15^\circ \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Quality score ω^* checks whether there is a sufficient number of 3D points with sufficiently large apical angles. The threshold value for ω^* is set to the number of the reconstructed 3D points, *i.e.* either all the reconstructed points must have the apical angles at least 5° or some of them are having even larger apical angles as the weighting constants set to 4 and 20 are favoring such 3D points. The pseudocode of keyframe selection is summarized in Algorithm 3. Note that Algorithm 3 may not select the last frame of the sequence as a keyframe. In that case, we regard the last keyframe as the end of the sequence.

After recovering the camera poses and 3D points using only the keyframes, the camera poses corresponding to the images not selected as the keyframes are estimated by solving the camera resectioning task (Nistér 2004b). Since every non-keyframe is interleaved between two keyframes, the tentative 2D-3D matches are efficiently constructed by extracting the 3D points associated with the two keyframes. RANSAC is used to find the camera pose having the largest support of the tentative 2D-3D matches evaluated by the cone test again. Local optimization is achieved by repeated camera pose computation from all the inliers (Schweighofer and Pinz 2008) via SDP and SeDuMi (Sturm 2006). Camera resectioning is considered successful when the inlier ratio is higher than 70%.



(a) (b)

Fig. 9 Projection of a pixel u' of the resulting cylindrical image onto a pixel u on a unit sphere. Column index u'_i is transformed into angle θ and row index u'_j into angle ϕ. These angles are then transformed into the coordinates u_x, u_y, and u_z of a unit vector. (**a**) Central cylindrical projection. (**b**) Non-central cylindrical projection

In the final step, very distant points, *i.e.* likely outliers, are filtered out. Sparse bundle adjustment (Lourakis and Argyros 2004), modified in order to work with unit vectors, refines both points and cameras.

5 Omnidirectional Image Stabilization

5.1 Image Rectification Using Camera Pose and Trajectory

The recovered camera poses and trajectory can be used to rectify the original images to the stabilized ones. If there exists no constraint on camera motion in the sequence, the simplest way of stabilization is to rectify images w.r.t. the up vector in the coordinate system of the first camera and all the other images will then be aligned with the first one. This can be achieved by taking the first image with care.

When the sequence is captured by walking or driving on the roads, the images can be stabilized w.r.t. the ground plane with a natural assumption that the motion direction is parallel to the ground plane. For the fixed gravity direction $\mathbf{g}$ and the motion direction $\mathbf{t}$, we compute the normal vector of the ground plane

$$\mathbf{d} = \frac{\mathbf{t} \times (\mathbf{g} \times \mathbf{t})}{\|\mathbf{t} \times (\mathbf{g} \times \mathbf{t})\|}. \tag{14}$$

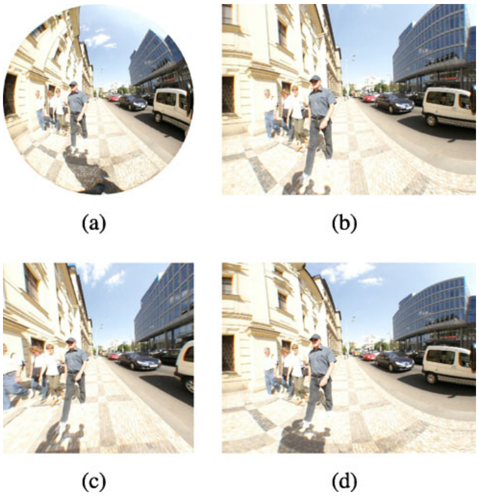


Fig. 10 Omnidirectional image rectification. (**a**) Original omnidirectional image (equi-angular). (**b**) Central cylindrical projection. (**c**) Perspective projection. (**d**) Non-central cylindrical projection. Note the large deformation at the borders of the perspective image and at the top and bottom borders of the central cylindrical image. The borders of the non-central cylindrical image are less deformed

Then, we construct the stabilization and rectification transform $\mathtt{R}_s$ for the image point represented as a unit 3D vector such that $\mathtt{R}_s = [\mathbf{a}, \mathbf{d}, \mathbf{b}]$ where

$$\mathbf{a} = \frac{(0,0,1)^\top \times \mathbf{d}}{\|(0,0,1)^\top \times \mathbf{d}\|} \tag{15}$$

and

$$\mathbf{b} = \frac{\mathbf{a} \times \mathbf{d}}{\|\mathbf{a} \times \mathbf{d}\|}. \tag{16}$$

This rectification preserves yaw (azimuth) which is sufficient for producing panorama images having the same field of view as the original images.

5.2 Central and Non-central Cylindrical Image Generation

Having perspective cutouts rectified w.r.t. the ground plane, an arbitrary object recognition routine designed to work with images acquired by perspective cameras can be used without any further modifications. Furthermore, some object recognition methods, *e.g.* (Leibe et al. 2007a), could benefit from image stabilization. On the other hand, as a true perspective image is able to cover only a small part of the available omnidirectional view field, we propose to use cylindrical images which can cover a much larger part of it.

Knowing the camera and lens calibration, we represent our omnidirectional image as a part of the surface of a unit sphere, each pixel being represented by a unit vector. It is straightforward to project such surface on a surface of a unit cylinder surrounding the sphere using rays passing through the center of the sphere, see Fig. 9. We transform the column index u'_i of a pixel of the resulting cylindrical image into angle θ and the row index u'_j into angle ϕ using

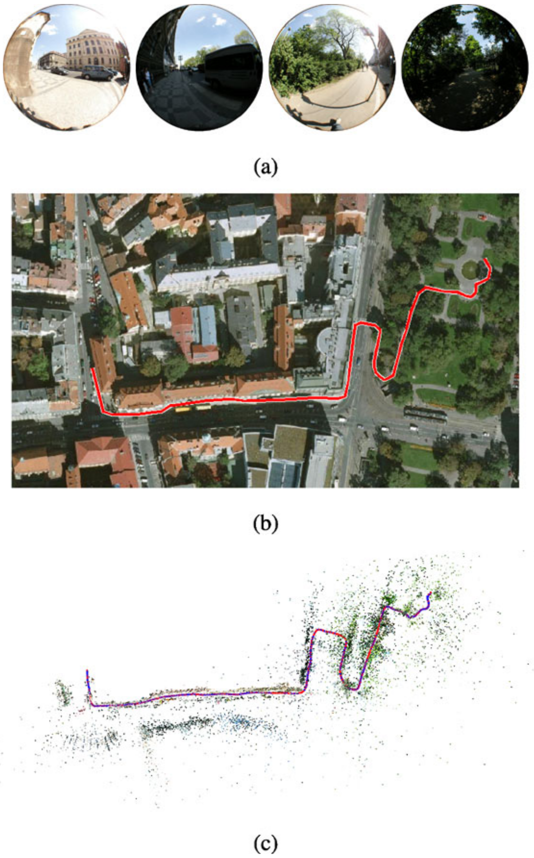


Fig. 11 (Color online) Camera trajectory of sequence CITY WALK. (**a**) The sequence contains moving objects occluding large parts of the view, rapid changes of illumination, and a natural complex environment. (**b**) A bird's eye view of the city area used for the acquisition of our test sequence. The trajectory is drawn with a *red line*. (**c**) The bird's eye view of the resulting 3D model. *Red cones* represent the keyframe camera poses recovered by our SfM. *Blue cones* represent the camera poses of the non-keyframes. *Small dots* represent the reconstructed world 3D points

$$\theta = \left(u'_i - \frac{I_W}{2}\right)\frac{\theta_{max}}{I_W}, \tag{17}$$

$$\phi = \arctan\left(\left(u'_j - \frac{I_H}{2}\right)\frac{\theta_{max}}{I_W}\right), \tag{18}$$

where I_W and I_H are the dimensions of the resulting image and θ_{max} is the horizontal field of view of the omnidirectional camera. These angles are then transformed into the coordinates u_x, u_y, and u_z of a unit vector as

$$u_x = \cos\phi \sin\theta, \qquad u_y = \sin\phi, \qquad u_z = \cos\phi\cos\theta. \tag{19}$$

Note that the top and bottom of the rectified image look rather deformed for the vertical field of view reaching π if the height of the resulting image I_H is being increased, see

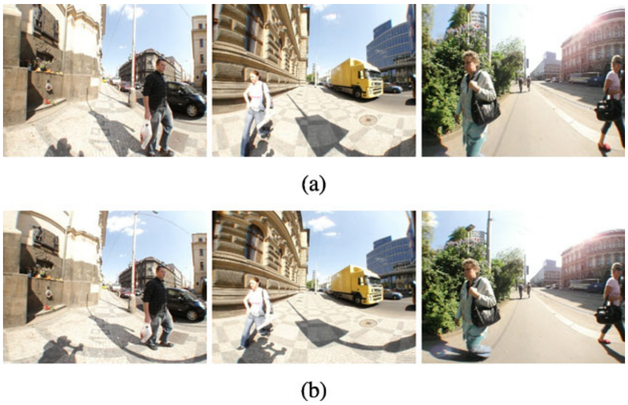


Fig. 12 Results of image transformations in sequence CITY WALK. The images are stabilized w.r.t. the ground plane and panoramic images transformed by (**a**) central cylindrical projection and (**b**) non-central cylindrical projection. Note that the pedestrians are less deformed when using the non-central cylindrical projection while convening a larger field of view than the central one

Fig. 10. We propose to use a generalization of the stereographic projection which we call a non-central cylindrical projection. Projecting rays do not pass through the center of the sphere but are cast from points on its equator. The desired point is the intersection of the plane determined by the column of the resulting image and the center of the sphere with the equator of the sphere. The equation for angle θ remains the same but angle ϕ is now computed using

$$\phi = 2\arctan\left(\frac{(u'_j - \frac{I_H}{2})\frac{\theta_{max}}{I_W}}{2}\right). \tag{20}$$

When generating the images, bilinear interpolation is used to suppress the artifacts caused by image rescaling.

6 Experimental Results

6.1 Omnidirectional Image Sequences

The experiments with real data demonstrate the use of the proposed image stabilization method. Five image sequences of a city scene captured by a single hand-held fish-eye lens camera are used as our input.

CITY WALK Sequence CITY WALK is 949 frames long and the distance between consecutive frames is 0.2–1 meters. This sequence is challenging for recovering the camera trajectory due to sharp turns, objects moving in the scene, large changes of illumination, and natural complex environments, see Fig. 11(a).

The camera motions are reasonably recovered by using the features detected from stationary rigid objects. Figure 11(c) shows the obtained camera poses and the world 3D points reconstructed by our SfM. The red cones represent the keyframe camera poses while the blue cones represent the non-keyframe camera poses computed by solving camera resectioning. The reconstructed camera trajectory fits well the walking trajectory shown in Fig. 11(b).

Since the sequence was captured while walking along a planar street, all the images were stabilized using the recovered camera poses and trajectory w.r.t. the ground plane. Figure 12 shows the images generated by using the central and the non-central cylindrical projections. It can be seen that the non-central cylindrical projection in Fig. 12(b) successfully suppresses the deformation at the top and the bottom of the image and makes people standing close to the camera look much more natural.

GO AND STOP Sequence GO AND STOP is 303 frames long and the distance between consecutive frames is 0–1 meters. The observer was standing still at a fixed spot in frames 1–14, 51–68, and 157–170, otherwise walking along a street. We can detect when the observer was standing by finding the “too small” DAA on the graph in Fig. 13(e) which shows the DAAs between every pair of consecutive frames. In Figs. 13(a)–(d), the green ▽ shows the relative camera motion direction estimated from pairs of images (a) and (b), and (c) and (d), respectively. The red and yellow dots are the tentative matches and the supports of the motion ▽. It can be seen that the motion direction is estimated incorrectly when the motion is too small even though the size of the support is sufficiently large.

Figures 13(f) and (g) show the camera poses and the world 3D points reconstructed by our SfM visualized from two different viewpoints. Again, the red cones represent the keyframe camera poses and the blue cones represent the non-keyframe camera poses.

ABNORMAL MOTION Sequence ABNORMAL MOTION is 410 frames long and the distance between consecutive frames is 0.2–1 meters. The observer was walking along a street when performing abnormal motions three times as spotted in yellow markers in Fig. 14(d). Figures 14(a), (b), and (c) show the frames 100–115, 333–348, and 381–396 respectively, where the abnormal motions were acted. Figure 14(d) shows a bird’s eye view of the city area used for the acquisition and the red dots are the computed camera poses of the keyframes superposed on it.

Figure 14(e) shows the camera poses of the keyframes (red cones) and of the non-keyframes (blue cones), and the world 3D points (color dots). The significant utility of the wide baseline SfM on large field of view images can be seen on the reliable recovery of the sequence having abnormal motions which are fatal for classic sequential SfM methods working under the assumption of limited motions.

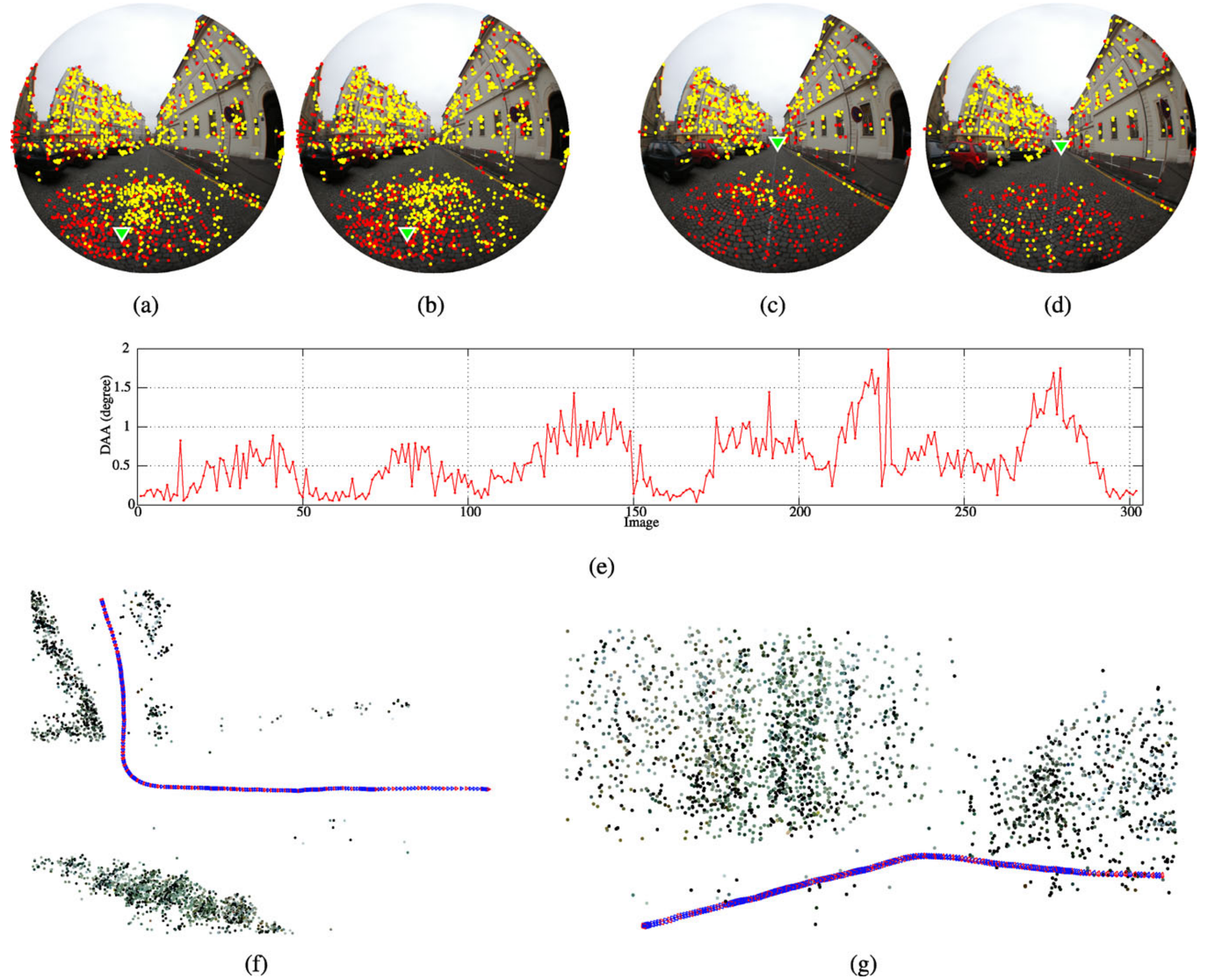


Fig. 13 (Color online) Detection of "too small motion" in sequence GO AND STOP. (**a**) and (**b**) Pair of images with too small motion. (**c**) and (**d**) Pair of images with a sufficiently large motion. The *green* ▽ shows the relative camera motion direction estimated from pairs of images (**a**) and (**b**), and (**c**) and (**d**), respectively. The *red* and *yellow dots* are the tentative matches and the supports of the motion ▽. It can be seen that the motion direction is estimated incorrectly when the motion is too small even though the size of the support is sufficiently large. (**e**) The DAA computed from pairs of consecutive images in the sequence. (**f**) and (**g**). The recovered camera poses and trajectory of keyframes (*red cones*) and non-keyframes (*blue cones*), and the world 3D points (*color dots*) from two different views

FREE MOTION Sequence FREE MOTION is 187 frames long and the distance between consecutive frames is 0.2–1 meters. This sequence is also challenging for recovering the camera poses and trajectory due to the large view changes caused by extreme camera rotation and translation. Figure 15(b) shows several examples of the original images in the sequence. Figure 15(a) shows the camera poses recovered by our SfM visualized from a bird's eye view. Figure 15(c) shows the panoramic images generated by the non-central cylindrical projection. As the motion is completely irrelevant w.r.t. the ground plane, all images are stabilized w.r.t. the gravity vector in the coordinate system of the first camera. Figure 15(d) shows the panoramic images stabilized using the recovered camera poses and trajectory. It can be seen clearly from this result that even large image rotations are successfully canceled using the recovered camera poses and trajectory.

PED DETECTION Sequence PED DETECTION is 404 frames long and the distance between consecutive frames is 0–1 meters. The images are stabilized w.r.t. the ground plane by using the estimated trajectory and rectified by adopting the non-central cylindrical projection, see Fig. 16. The multi-body pedestrian tracker (Dalal and Triggs 2005; Ess et al. 2008) is applied to the sequence of the stabilized cylindrical images and the results are shown in

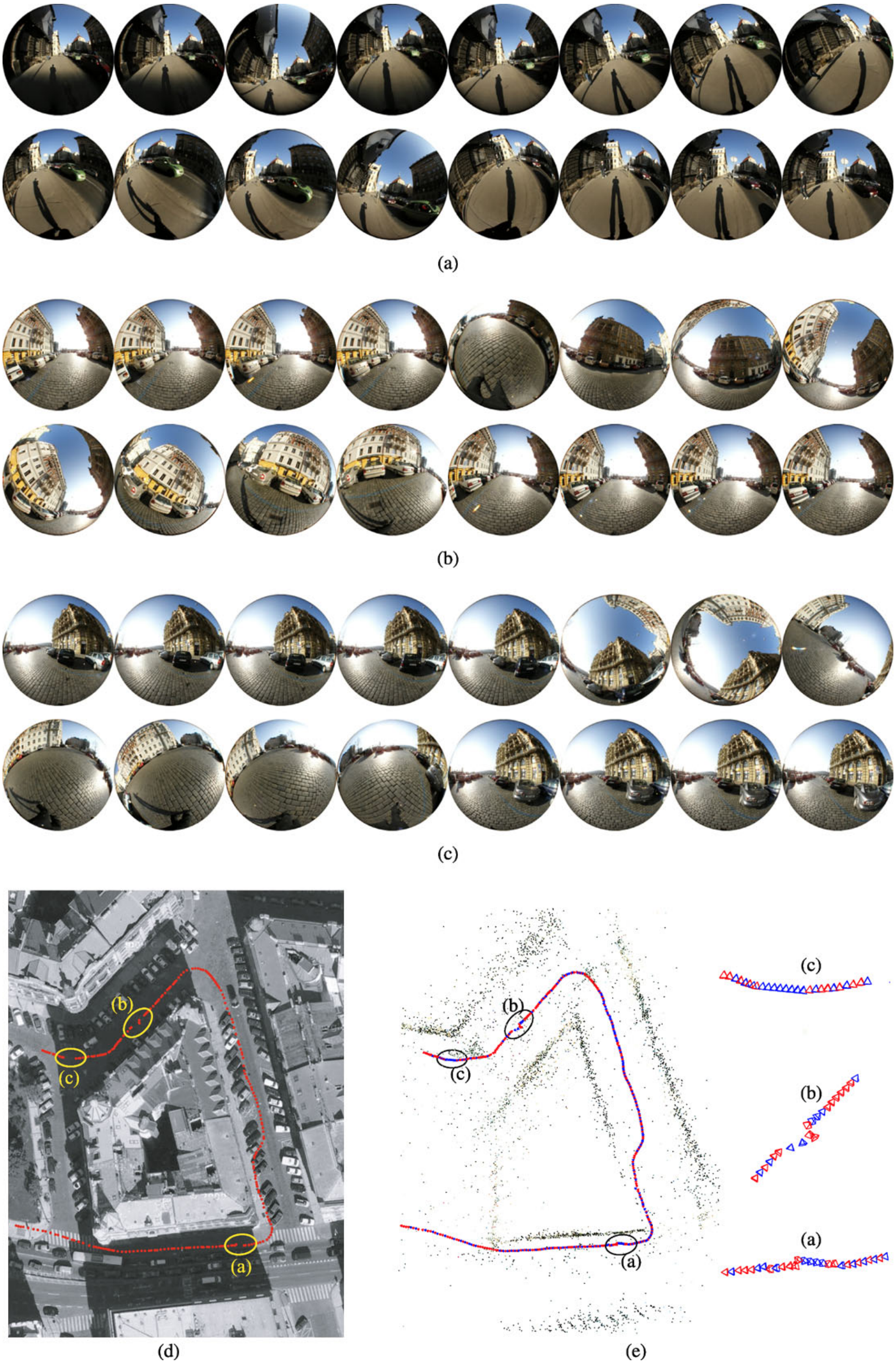


Fig. 14 (Color online) Camera poses and trajectory of sequence ABNORMAL MOTION. The sequence was acquired with abnormal motions at the frames (**a**) 100–115, (**b**) 333–348, and (**c**) 381–396 while walking along a street. (**d**) The recovered camera poses of the keyframes are superimposed on the map of a bird's eye view. (**e**) The recovered camera poses of the keyframes (*red cones*) and of the non-keyframes (*blue cones*), and the world 3D points (*color dots*) from a bird's eye view. See detailed views of the recovered camera poses in (**a**), (**b**), and (**c**) on *the right side*

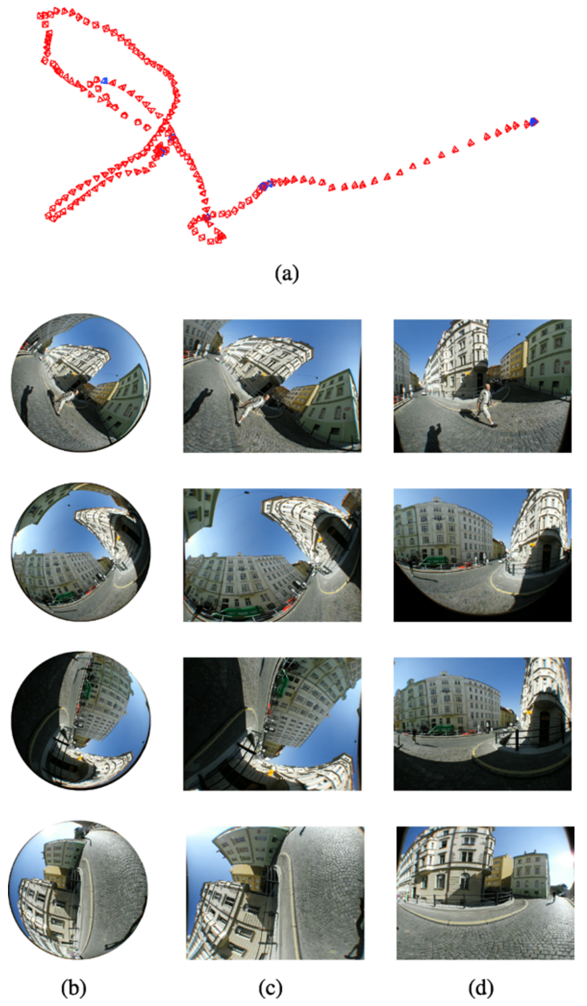


Fig. 15 Results of our image stabilization and transformation in sequence FREE MOTION. (**a**) The camera poses and the world 3D points reconstructed by our SfM visualized from a bird's eye view. *Red cones* represent the keyframe camera poses recovered by our SfM. *Blue cones* represent the non-keyframe camera poses estimated by solving camera resectioning. (**b**) Original images. (**c**) Non-stabilized images. (**d**) Stabilized images w.r.t. the gravity vector in the first camera coordinates. Image rotations are successfully canceled and all images are stabilized using the recovered camera poses and trajectory

Figs. 16(b)–(e). Thanks to proper image rectification, a pedestrian detector using Histograms of oriented Gradients (HoG) (Dalal and Triggs 2005) trained on perspective images could be used. The tracker used can greatly benefit from our ability of producing stable image sequences as it uses the ground plane position to reject false positive detections which is otherwise possible only for sequences acquired by vehicle-mounted cameras.

6.2 Unorganized Omnidirectional Images

We demonstrate the capability of our sequential SfM on unorganized images by applying an image indexing method based on visual words and visual vocabulary (Sivic and Zisserman 2006; Knopp et al. 2009) for ordering images into a sequence. Data set CASTLE ENTRANCE originally consisted of three sequences acquired at different times. To reveal the ability of wide baseline SfM, we randomly selected 40 out of 109 images of the whole data to make cameras sparser than general sequential images. Furthermore, 10 images of different locations were added as outliers, see Figs. 17(a)–(d).

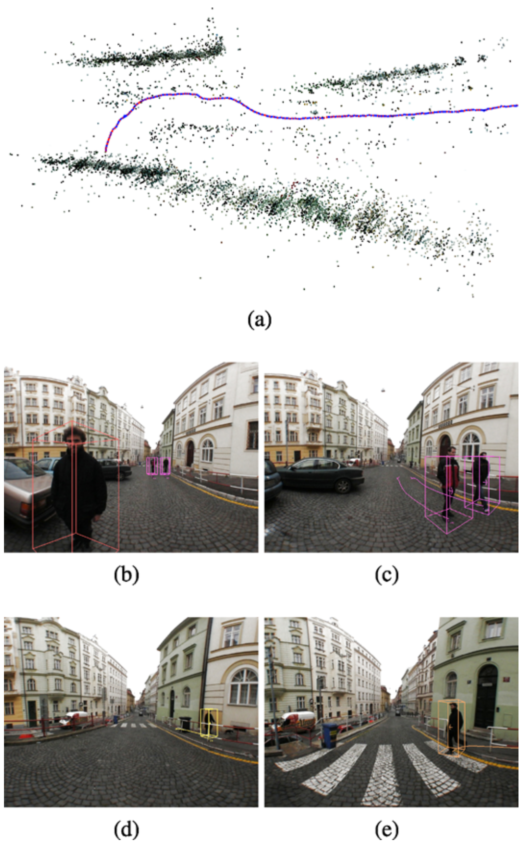


Fig. 16 Result of multi-body pedestrian tracking in sequence PED DETECTION on the cylindrical images stabilized w.r.t. the ground plane. (**a**) The camera poses and the world 3D points reconstructed by our SfM visualized from a bird's eye view. In (**b**), (**c**), (**d**), and (**e**), the color boxes and curves indicate the positions of pedestrians and their trajectories estimated by using the previous frames

For each image, a term frequency–inverse document frequency (tf-idf) vector (Sivic and Zisserman 2006; Knopp et al. 2009) was computed using a visual vocabulary containing 130,000 words trained from urban area omnidirectional images. Image similarities between the pairs of images were computed as the cosines of the angles between the normalized tf-idf vectors. Then, a pseudo-sequence was constructed by randomly selecting one image as the first frame and concatenating the most similar image as the successive frame. 33 camera poses were successfully recovered and none of outlier frames were selected. See Fig. 17(e) for the recovered camera poses of the keyframes (red cones),

Table 1 Details of the experimental results for all sequences. (# Frames) The number of frames. (# Keyframes) The number of keyframes selected and used in our wide baseline SfM. (Min. DAA) The minimum DAA, *i.e.* the minimum size of motion, in degrees. The rest is the computational time in different steps for each sequence in minutes. (Detection) Feature detection and description. (Matching) Tentative match construction and EG computation. (SfM) Chaining EGs, scale estimation, and bundle adjustment. (Resectioning) Estimating camera poses of non-keyframes

Name	# Frames	# Keyframes	Min. DAA	Detection	Matching	SfM	Resectioning
CITY WALK	949	503	1°	147	77	3	30
GO AND STOP	303	73	2°	41	30	2	35
ABNORMAL MOTION	410	198	1°	57	43	4	25
FREE MOTION	187	176	0.2°	32	18	3	2
PED DETECTION	404	74	2°	54	26	2	51
CASTLE ENTRANCE	50	31	2°	6	4	0.5	0.5

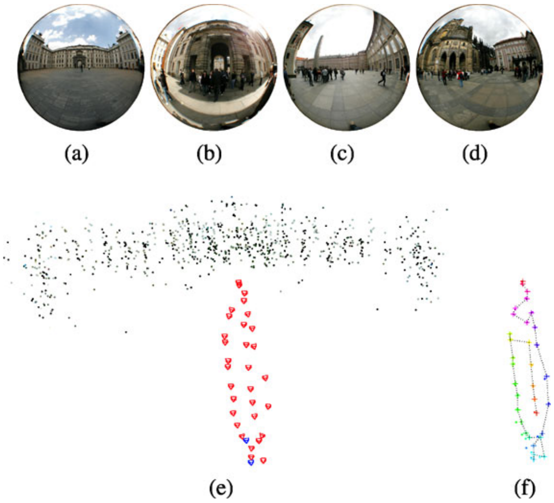


Fig. 17 Result of our SfM performed on ordered unorganized omnidirectional images from data set CASTLE ENTRANCE. The data set consists of 40 typical landmark images of the entrance to the Prague Castle (**a**) and (**b**) and 10 images from other locations acting as outliers (**c**) and (**d**). (**e**) The camera poses and the world 3D points reconstructed from images ordered as a sequence by using image similarity computed based on visual words and visual vocabulary indexing. (**f**) Visualization of the camera poses (+) and trajectory (*dashed line*) estimated by our method and the camera poses (•) estimated by the randomized SfM (Havlena et al. 2009)

the non-keyframes (blue cones), and the world 3D points (color dots).

109 camera poses of the same scene reconstructed by the state of the art randomized SfM method (Havlena et al. 2009) were used as the ground truth data of evaluating the accuracy of the camera pose estimation. We measured the error between the camera poses computed by our method and those computed by the randomized SfM method after giving the corresponding image indices and finding the similarity transform bringing the data into correspondence. The mean of the translational and rotational errors are 0.024 and 0.031 respectively, where the translational error is the fraction of the diameter of the smallest sphere containing all cameras and the rotational error is in radians. Both sets of camera poses can be seen in Fig. 17(f).

6.3 Details of Experimental Settings and Computations

We used the same parameter values except the minimum DAA η, *i.e.* the minimum size of motion, for all sequences. The actual values used in the experiments are listed in Algorithms 1 and 2, and Table 1. For all sequences but FREE MOTION, there was no significant difference in the visual quality of the reconstruction as long as setting the minimum DAA η between 1° and 2°. In sequence FREE MOTION, we set the minimum DAA $\eta = 0.2°$ which is smaller than in other sequences because larger values of the minimum DAA selected too sparse keyframes due to the lack of matches in consecutive frames and thus the camera trajectory could not be recovered stably.

The time spent in different steps of the pipeline having a MATLAB+MEX implementation running on a standard Core2Duo PC can be found in Table 1. The average computation time is about 18 seconds per frame and the performance can be further improved by using GPU implementations of feature detection and by speeding up the data storing processes which are caching all the results used in the pipeline on a hard drive.

The proposed pipeline is available on-line[1] through the CMP SfM web service (Heller et al. 2010). One can upload her own images and run the pipeline after being registered to the site. There is no need to install any code on a client's computer and all the computations are performed on our dedicated computing cluster. The service can be accessed through a web browser based interface or by a command line interface based utility.

[1] http://ptak.felk.cvut.cz/sfmservice

7 Conclusions

We presented a pipeline for camera pose and trajectory estimation, and image stabilization and rectification, for image sequences acquired by a single omnidirectional camera. The experiments demonstrated that the robust camera pose and trajectory estimation based on epipolar geometry is useful to stabilize image sequences. Furthermore, the non-central cylindrical projection which generates perspective-projection-like images while preserving a large field of view can be instantly used as the pre-process for the detection and tracking techniques (Leibe et al. 2007a, 2007b; Ess et al. 2008) that assume ground plane positions and have codebooks trained on perspective images.

Acknowledgements The authors were supported by EC project FP7-SPA-218814 PRoVisG, project CAT, and by Czech Government under the research program MSM 6840770038. We would like to thank Bastian Leibe and Andreas Ess for providing us the results of the multi-body pedestrian tracking. Also, we would like to thank Horst Bischof for the fruitful discussions at Computer Vision Winter Workshop 2009.

References

2d3. Boujou (2001). http://www.boujou.com.

Akbarzadeh, A., Frahm, J. M., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Merrell, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., Towles, H., Nistér, D., & Polleeys, M. (2006). Towards urban 3D reconstruction from video. In *3DPVT*, Invited paper.

Bakstein, H., & Pajdla, T. (2002). Panoramic mosaicing with a 180° field of view lens. In *OMNIVIS '02*, Copenhagen, Denmark (pp. 60–67).

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359.

Brown, M., & Lowe, D. G. (2003). Recognising panoramas. In *ICCV '03*, Washington, DC, USA.

Chum, O., & Matas, J. (2005). Matching with PROSAC—progressive sample consensus. In *CVPR '05*, Los Alamitos, USA (Vol. I, pp. 220–226).

Clipp, B. Kim, J.-H., Frahm, J.-M., Pollefeys, M., Hartley, R. (2008). Robust 6DOF motion estimation for non-overlapping, multi-camera systems. In *WACV '08* (Vol. I, pp. 1–8).

Cornelis, N., Cornelis, K., & Van Gool, L. (2006). Fast compact city modeling for navigation pre-visualization. In *CVPR '06*, New York, USA (Vol. II, pp. 1339–1344).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR '05*, Los Alamitos, USA (Vol. I, pp. 886–893).

Davison, A. J., & Molton, N. D. (2007). Monoslam: Real-time single camera SLAM. *IEEE Transactions on Patern Analysis and Machine Intelligence*, *29*(6), 1052–1067.

Ess, A., Leibe, B., Schindler, K., & Van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In *CVPR '08*, Anchorage, AK, USA.

Fischler, M., & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Geyer, C., & Daniilidis, K. (2001). Structure and motion from uncalibrated catadioptric views. In *CVPR '01* (pp. 279–286).

Goedemé, T., Nuttin, M., Tuytelaars, T., & Van Gool, L. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, *74*(3), 219–236.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.

Havlena, M., Pajdla, T., & Cornelis, K. (2008). Structure from omnidirectional stereo rig motion for city modeling. In *VISAPP '08*, Funchal, Portugal.

Havlena, M., Torii, A., Knopp, H., & Pajdla, T. (2009). Randomized structure from motion based on atomic 3D models from camera triplets. In *CVPR '09*, Miami, FL, USA.

Heller, J., Havlena, M., Torii, A., & Pajdla, T. (2010). *CMP SfM web service v1.0.* (Research Report CTU–CMP–2010–01). CMP Prague.

Hoiem, D., Efros, A. A., & Hebert, M. (2006). Putting objects in perspective. In *CVPR '06* (Vol. II, pp. 2137–2144).

Kahl, F. (2005). Multiple view geometry and the L-infinity norm. In *ICCV '05*, China, Beijing.

Ke, Q., & Kanade, T. (2007). Quasiconvex optimization for robust geometric reconstruction. *IEEE Transactions on Patern Analysis and Machine Intelligence*, *29*(10), 1834–1847.

Knopp, J., Šivic, J., & Pajdla, T. (2009). *Location recognition using large vocabularies and fast spatial matching* (Research Report CTU–CMP–2009–01). CMP Prague.

Leibe, B., Cornelis, N., Cornelis, K., & Van Gool, L. (2007a). Dynamic 3D scene analysis from a moving vehicle. In *CVPR '07*, Minneapolis, MN, USA.

Leibe, B., Schindler, K., & Van Gool, L. (2007b). Coupled detection and trajectory estimation for multi-object tracking. In *ICCV '07*, Rio de Janeiro, Brazil.

Li, H., & Hartley, R. (2005). A non-iterative method for correcting lens distortion from nine point correspondences. In *OMNIVIS '05* China: Beijing.

Lourakis, M., & Argyros, 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. http://www.ics.forth.gr/~lourakis/sba.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Martinec, D., & Pajdla, T. (2007). Robust rotation and translation estimation in multiview reconstruction. In *CVPR '07*, Minneapolis, MN, USA.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, *22*(10), 761–767.

Microsoft (2008). Photosynth: Use your camera to stitch the world. http://livelabs.com/photosynth.

Mičušík, B., & Pajdla, T. (2006). Structure from motion with wide circular field of view cameras. *IEEE Transactions on Patern Analysis and Machine Intelligence*, *28*(7), 1135–1149.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, *65*(1–2), 43–72.

Muja, M., & Lowe, D. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP '09*, Lisboa, Portugal.

Nistér, D. (2004a). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Patern Analysis and Machine Intelligence*, *26*(6), 756–770.

Nistér, D. (2004b). A minimal solution to the generalized 3-point pose problem. In *CVPR '04*, Washington, DC, USA (Vol. I, pp. 560–567).

Nistér, D., & Engels, C. (2006). Estimating global uncertainty in epipolar geometry for vehicle-mounted cameras. In *SPIE, unmanned systems technology VIII* (Vol. 6230).

Obdržálek, Š., & Matas, J. (2002). Object recognition using local affine frames on distinguished regions. In *BMVC '02*, London, UK (Vol. I, pp. 113–122).

Obdržálek, Š, & Matas, J. (2003). Image retrieval using local compact DCT-based representation. In *LNCS: Vol. 2781. DAGM '03* (pp. 490–497). Berlin: Springer.

Point Grey Research (2005). Ladybug 2 Spherical Digital Camera System. http://www.ptgrey.com/products/ladybug2.

Scaramuzza, D., Fraundorfer, F., Siegwart, R., & Pollefeys, M. (2008). Closing the loop in appearance guided SfM for omnidirectional cameras. In *OMNIVIS '08*, Marseille, France.

Schweighofer, G., & Pinz, A. (2008). Globally optimal O(n) solution to the PnP problem for general camera models. In *BMVC '08*, Leeds, UK.

Sivic, J., & Zisserman, A. (2006). Video Google: Efficient visual search of videos. In *CLOR '06* (pp. 127–144).

Snavely, N., Seitz, S., & Szeliski, R. (2006). Photo Tourism: Exploring image collections in 3D. In *SigGraph '06*, Boston, USA (pp. 835–846).

Snavely, N., Seitz, S., & Szeliski, R. (2008). Skeletal graphs for efficient structure from motion. In *CVPR '08*, Anchorage, AK, USA.

Stewénius, H. (2005). *Gröbner basis methods for minimal problems in computer vision*. PhD thesis, Centre for Mathematical Sciences LTH, Lund University, Sweden.

Sturm, J. (2006). Sedumi: A software package to solve optimization problems. http://sedumi.ie.lehigh.edu.

Tardif, J., Pavlidis, Y., & Daniilidis, K. (2008). Monocular visual odometry in urban environments using an omdirectional camera. In *IROS '08*, Nice, France.

Torii, A., & Pajdla, T. (2008). Omnidirectional camera motion estimation. In *VISAPP '08*, Funchal, Portugal.

Torii, A., Havlena, M., Pajdla, T., & Leibe, B. (2008). Measuring camera translation by the dominant apical angle. In *CVPR '08*, Anchorage, AK, USA.

Williams, B., Klein, G., & Reid, I. (2007). Real-time SLAM relocalisation. In *ICCV '07*, Rio de Janeiro, Brazil.