Reduced egomotion estimation drift using omnidirectional views

Yalin Bastanlar

Department of Computer Engineering, Izmir Institute of Technology, Urla, Izmir, Turkey

Abstract

Estimation of camera motion from a given image sequence is a common task for multi-view 3D computer vision applications. Salient features (lines, corners etc.) in the images are used to estimate the motion of the camera, also called egomotion. This estimation suffers from an error built-up as the length of the image sequence increases and this causes a drift in the estimated position. In this letter, this phenomenon is demonstrated and an approach to improve the estimation accuracy is proposed. The main idea of the proposed method is using an omnidirectional camera (360° horizontal field of view) in addition to a conventional (perspective) camera. Taking advantage of the correspondences between the omnidirectional and perspective images, the accuracy of camera position estimates can be improved.

In our work, we adopt the sequential structure-from-motion approach which starts with estimating the motion between first two views and more views are added one by one. We automatically match points between omnidirectional and perspective views. Point correspondences are used for the estimation of epipolar geometry, followed by the reconstruction of 3D points with iterative linear triangulation. In addition, we calibrate our cameras using sphere camera model which covers both omnidirectional and perspective cameras. This enables us to treat the cameras in the same way at any step of structure-from-motion.

We performed simulated and real image experiments to compare the estimation accuracy when only perspective views are used and when an omnidirectional view is added. Results show that the proposed idea of adding omnidirectional views reduces the drift in egomotion estimation.

Key Words: Omnidirectional Cameras, Structure-from-motion, Egomotion Estimation, Visual Odometry

1 Introduction

Estimating the 3D motion of a camera (egomotion) and the 3D scene structure simultaneously from a sequence of images is known as simultaneous localization and mapping (SLAM) in robotics community and as structure from motion (SfM) in computer vision community. This problem has been studied extensively over the years and different approaches were proposed. However, satisfactory results are not easily obtained due to the high dimension of the space of unknowns and challenges regarding the nature of the work such as unreliable reconstruction due to short baseline between the views and erroneous feature matching when the scene is composed of repetitive patterns.

Most of the methods that estimate the camera motion employ the approach of adding views to the structure one by one [1]. In this approach, when a sequence of views is available, geometry extraction is initially performed for the first two views. The three dimensional structure built with these two views is used to estimate the pose of a new view with the help of point detection and matching between the new view and

already reconstructed points. At each step, point correspondences with another view of the camera are extracted and motion of the camera is extended. Due to this sequential nature, this kind of SfM algorithms is called sequential algorithms. Unfortunately, the error at each view addition builds up and the estimation of camera poses suffers from a drift. This phenomenon is depicted in Fig.1.

Our proposal in this letter is using an omnidirectional camera in addition to a conventional camera to reduce the drift occurring in the sequential SfM algorithms. We show that adding omnidirectional views to a SfM consisting of only perspective views improves the accuracy of camera position estimation. With its 360° view, an omnidirectional image can provide us point correspondences with all perspective images taken from a nearby location. Fig. 2 shows the case where a perspective camera is moved and two images from that camera do not share much field-of-view. However, an omnidirectional camera has common viewing angles with both perspective images. Our hypothesis is that these point correspondences help to obtain more accurate egomotion estimation. We tested the effect of adding an omnidirectional view to a perspective-only SfM by measuring the drift occurred. In our experiments, we consider both loop and non-loop sequences.

The organization of the paper is as follows. In Section 2, we present the related work and the differences in our approach. In Section 3, we explain the steps of sequential SfM when an omnidirectional camera is added and give visual examples. The results of our experiments, given in Section 4, indicate that the proposed approach helps to reduce egomotion estimation drift.
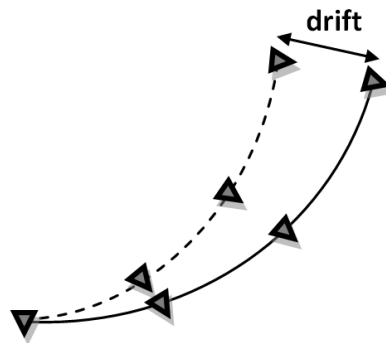

Figure 1: Drift can be defined as the difference between the actual and estimated camera positions after a sequence of images or video frames. Actual path of the camera is shown by the dashed line, whereas estimated camera positions (triangles on the solid line) are drifted from the actual path. The error grows as more views are added since the motion estimation depends on the position of the previous view.
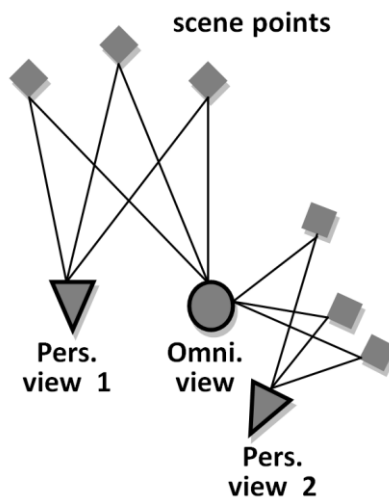

Figure 2: An omnidirectional image can provide us point correspondences with different perspective images that do not share much field-of-view.

2   Related Work

Performing egomotion estimation for a moving platform over a long distance can also be called as visual odometry since it uses visual information instead of other sources such as GPS, inertia sensors, wheel encoders, etc. [2]. If scene points are constructed together with the motion of the camera, then the task is called structure-from-motion. An automated system that handles large-scale reconstructions was presented in [3].

As mentioned in the previous section, due to the sequential nature of these methods error propagates as more images are added and egomotion estimation suffers from a drift. To solve the problem, authors in [4] proposed a non-sequential SfM. Their method depends on rotational consistency and uses convex optimization. They also performed experiments regarding the drift and showed that their method is more robust when compared with the approach in [3].

Regarding sequential algorithms, a method for drift removal was presented in [5]. That method is based on the assumption that the image sequence forms a closed loop. In other words, image sequence ends up at the starting point and this information is used to reduce the accumulated error. There are other previously proposed loop closing methods as well [6, 7]. However, real life applications do not necessarily contain loop sequences. Therefore, some techniques should be proposed to obtain accurate egomotion estimations with sequential algorithms even when the image sequences do not contain loops. With this motivation, we propose an approach that exploits the wide field-of-view capability of an omnidirectional camera and does not assume that the image sequence forms a loop. Details will be given in Section 3.

It is known that using wide field-of-view cameras provides point correspondences from a variety of angles which enables more stable structure estimation and degenerate cases like viewing only a planar surface are less likely to occur [8]. Thus, one can think of employing only an omnidirectional camera for SfM. However, due to their low resolution and high distortion, matching of image feature points is less reliable than with perspective cameras. In an omnidirectional visual odometry study [9], the authors point out that the rotation estimated from feature points alone gives rise to large errors. For better rotation estimation, they convert the omnidirectional image to a panoramic image and estimate the horizontal shift in the panoramic image. It was also mentioned that when the feature points are distributed in an unbalanced fashion, estimation is degraded. In our study, we use the omnidirectional camera as an aid to the perspective camera approach.

The studies on SfM with mixed camera types started about a decade ago. The epipolar geometry between a para-catadioptric (catadioptric camera with a parabolic mirror) and a perspective camera was first explained in [10]. Later, the framework was extended to catadioptric cameras with hyperbolic mirrors and cameras with lens distortion [11]. Regarding a complete SfM task, a work for hybrid systems was presented in [12]. They employed a highly generic non-parametric imaging model where the cameras are modelled with sets of projection rays. They mentioned that directly applying SIFT [13] did not provide good results for their fisheye-perspective image pairs and used manually selected feature point correspondences to estimate the epipolar geometry. In our recent work [14], we proposed a technique to perform point matching automatically so that no manual interaction is needed for the complete SfM task. Moreover, a single model, namely sphere camera model, was used for mixed types of cameras, which enabled us to perform most of the computations regarding the omnidirectional images linearly.

The SfM computation in this letter uses the algorithms proposed in [14]. Our contribution here is that we investigate, with experiments, the effect of adding an omnidirectional view to a perspective-only SfM. In other words, we compare hybrid SfM with perspective-only SfM in terms of egomotion estimation accuracy.

3   Sequential Structure-from-Motion with Mixed Camera Types

The pipeline of sequential SfM algorithm can be summarized with the steps shown in Fig. 3. Let us go over these steps regarding that our approach involves using omnidirectional cameras with perspective ones.

To calibrate our cameras of two different types, we use the sphere camera model [15] which is able to cover single viewpoint omnidirectional cameras as well as perspective cameras. The main characteristics of the model are given in Fig. 4. The most important aspect is that the non-linear image formation in omnidirectional cameras is converted to two linear projections; one is the projection of 3D point onto a

sphere, the other is a perspective projection from the sphere to the image plane. The usage of this model was also extended to fisheye cameras [16]. Using the sphere model, there is no need to change the camera model between the steps of the SfM algorithm. Another advantage is that, projections and epipolar geometry estimation can be computed linearly to a great extent. Thus, the need to use polynomial equations for calculating the direction of rays is avoided.
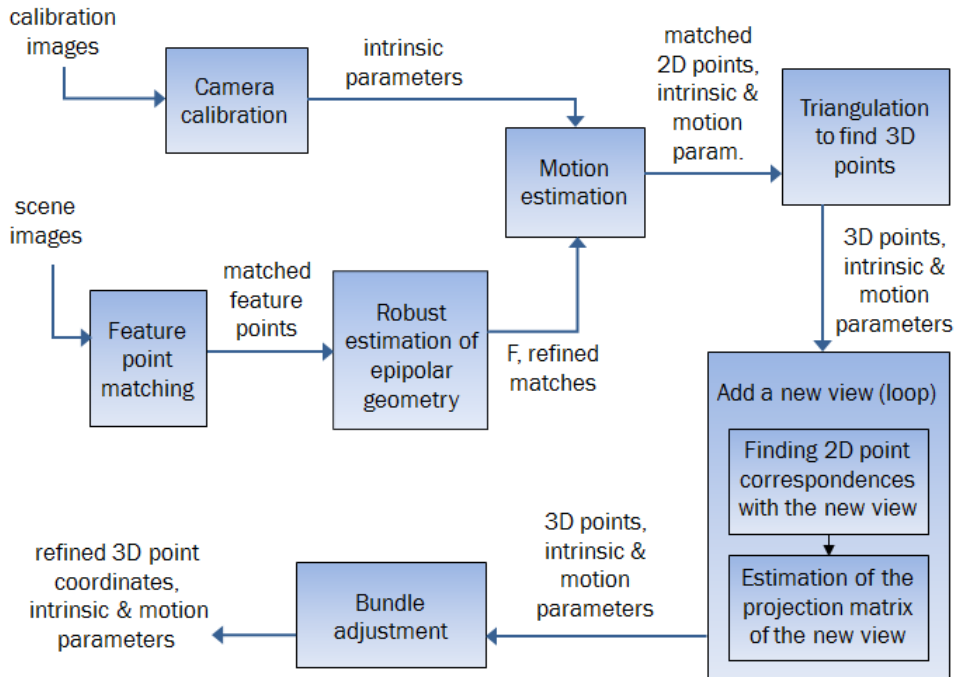

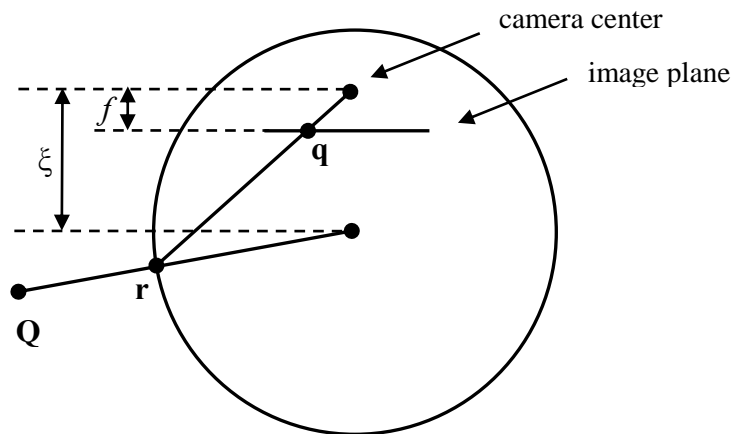Figure 3: Steps of the applied sequential SfM pipeline.

Figure 4: Projection of a 3D point onto the image plane in sphere camera model. The first step is the projection of 3D point **Q** onto a sphere, resulting in point **r**, and the second step is a perspective projection from the sphere to the image plane, resulting in point **q**. $\xi = 0$ for perspective cameras, $\xi = 1$ for para-catadioptric cameras (the ones using a paraboloidal mirror), $0 < \xi < 1$ for hyper-catadioptric cameras (the ones using a hyperboloidal mirror). Image plane is drawn inside the sphere for convenience, however it may be at any elevation depending on the scale of the image, parameterized by f.

The sequential SfM pipeline starts with feature point matching between the first two views. These are preferably two images captured with the perspective camera so that SIFT [13] can directly be applied for point matching. When an omnidirectional image is added to the sequence at a later stage, it is required to find point matches between that image and the perspective camera images. To do this automatically, we use an algorithm to eliminate false SIFT matches [17]. We used this algorithm in [14] and showed that the majority of the false matches in SIFT output are due to matching a high-resolution feature in the perspective image to a feature in the omnidirectional image which does not have such high-resolution. By pre-processing perspective images, which mainly consists of a low-pass filtering step, the probability of matching the features between the incorrect scales (octaves) significantly decreases. Fig. 5 shows the result of the employed matching method for a hybrid image pair used for the experiments in this letter. We observe most of the false matches are eliminated. Please also note that the matches that do not conform to the estimated epipolar geometry will be eliminated later by RANSAC [18].
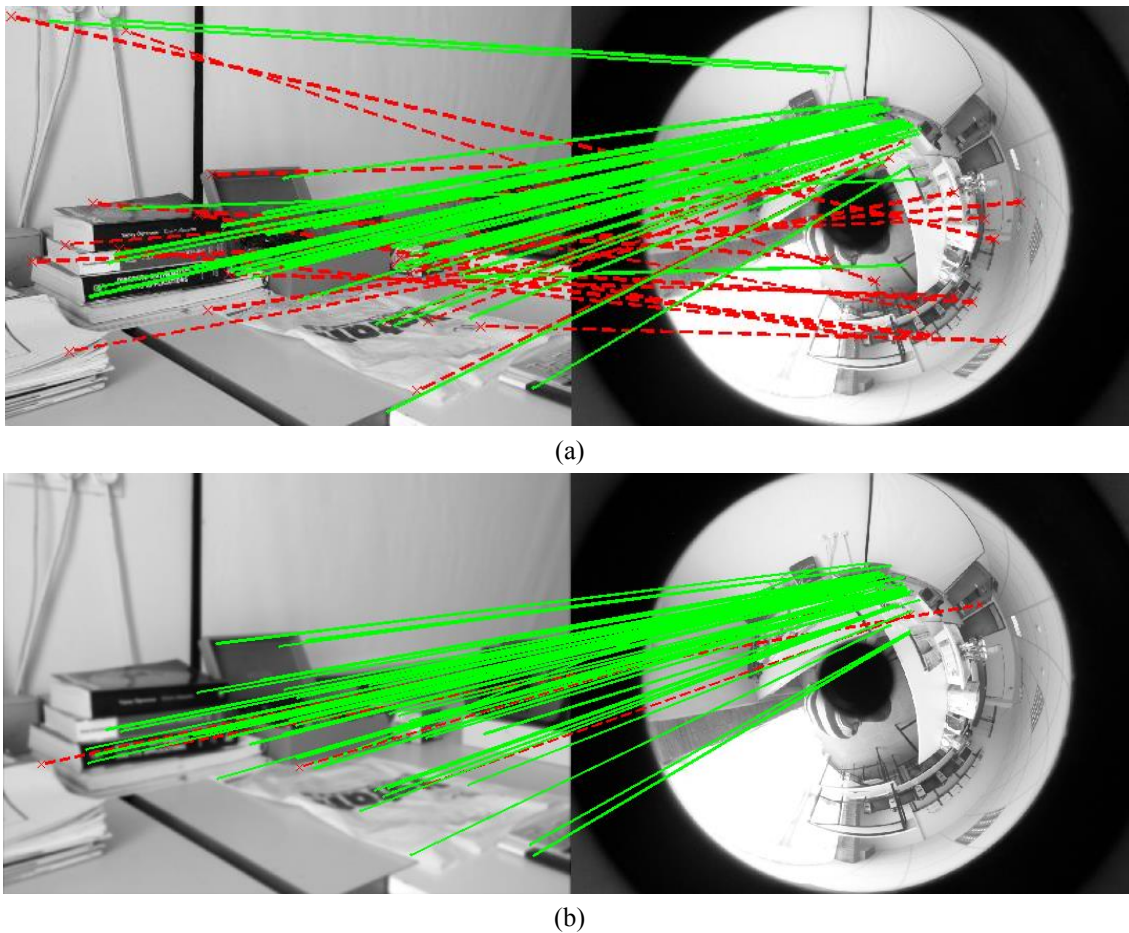

(a)

(b)

Figure 5: Point matching results for an example hybrid image pair. (a) Standard SIFT matching results where correct/total match ratio is 39/61. (b) Results after the employed automatic false match elimination method. Correct/total match ratio is 53/57. Red dashed lines show false matches, green solid lines show the correct ones.

Point correspondences are then used for the estimation of epipolar geometry. Following the technique which was first explained in [16], we represent the points in catadioptric omnidirectional images with so-called lifted coordinates. Lifting for para-catadioptric cameras can be performed by $\hat{\mathbf{q}}_{\mathbf{c}} = (x^2 + y^2, x, y, 1)^{\mathrm{T}}$. Matching point in the perspective image, $\mathbf{q}_{\mathbf{p}}$, is not lifted and represented with a 3-vector in homogeneous coordinates. A 3x4 hybrid fundamental matrix ($F_{\mathbf{pc}}$) expresses the epipolar constraint between these points:

$$\mathbf{q}_{\mathbf{p}}^{\mathrm{T}} \, F_{\mathbf{pc}} \, \hat{\mathbf{q}}_{\mathbf{c}} = 0$$

A point in the perspective image corresponds to a curve, namely epipolar conic, in the catadioptric image. Hybrid epipolar geometry can be visualized in Fig. 6. For the image pair given in Fig. 5, the set of corresponding epipolar lines/conics for the matched points are shown in Fig. 7. RANSAC [18] is used to select matches that conform to the estimated epipolar geometry. Therefore, false matches coming from the point matching step are eliminated if they violate the estimated epipolar geometry.

With the final set of point correspondences, and using the calibration information, the essential matrix, E, is computed. E in hybrid case is 3x3 similar to the perspective camera case since the 3D rays outgoing from the camera center are used. The motion parameters, namely the rotation matrix, R, and the translation vector, **t**, can be obtained from E using one of the known techniques [19]. These steps are followed by the reconstruction of 3D points with iterative linear triangulation [20].
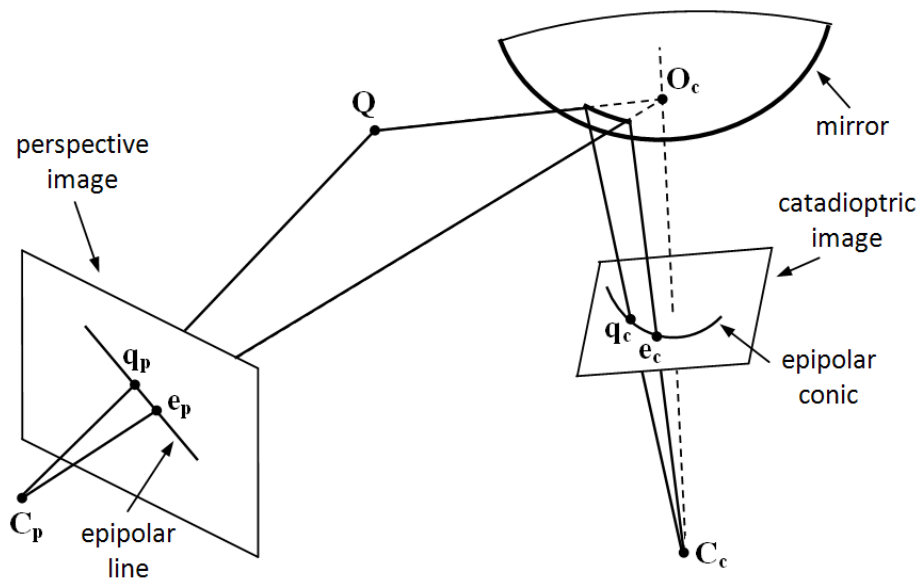

Figure 6: Hybrid epipolar geometry between a perspective and a catadioptric image. $\mathbf{q_p}$ and $\mathbf{q_c}$ are the projections of a 3D point **Q** on perspective and catadioptric images respectively. $\mathbf{e_p}$ and $\mathbf{e_c}$ are the epipoles in the perspective and catadioptric images respectively.
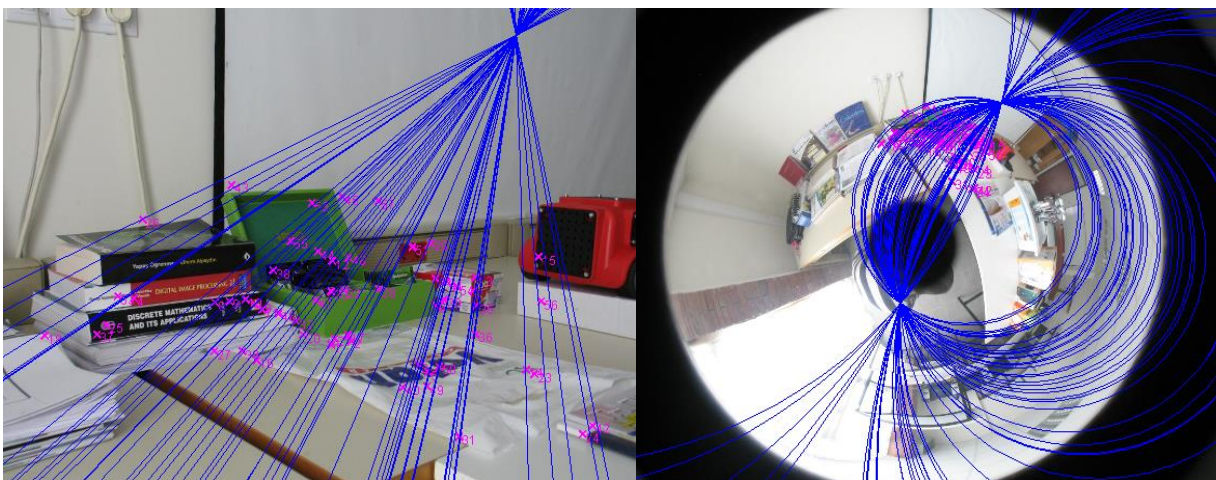

Figure 7: Example catadioptric-perspective image pair and corresponding epipolar conics/lines of point matches. 53 of the 57 input matches are kept after RANSAC.

As shown in Fig. 3, to continue with egomotion estimation, more views are added one by one. The 3D structure built with the first two views is used to estimate the pose of a new view with the help of the matches between the new view and already reconstructed points. Perspective views are added with standard point matching, motion estimation and triangulation stages. When an omnidirectional view is added, the hybrid image pairs are handled with algorithms described above which are the modified versions of perspective camera methods.

No public implementations or software packages were used for the steps explained so far. Algorithms were coded in MATLAB for this work and two of our previous papers [10,14], to which we refer the readers for details. MATLAB code can be provided upon request. Hybrid image sequences used in this letter can be downloaded from our website: http://cvrg.iyte.edu.tr/.

As the last step of the SfM pipeline, the final reconstructions are improved by minimizing reprojection errors. This process is called bundle adjustment. We performed this using the Levenberg-Marquardt algorithm offered within MATLAB. There is a powerful publicly available C++ package for bundle adjustment, called 'sparse bundle adjustment' [21]. It also uses Levenberg-Marquardt algorithm but more efficient since it exploits the sparseness in the data. However, we could not use this package since it does not offer the flexibility to define different intrinsic parameters for different camera types.

4   Experiments

Our primary contribution in this paper is providing an experimental proof of the idea that omnidirectional views decrease the drift in sequential SfM. For this aim, simulated and real image experiments are presented separately in the following.

4.1   Experiments with simulated images

First, we analyse the proposed idea in a simulated environment. Generated points were randomly distributed in a 3D space. Simulated perspective camera sees only a part of the scene points at a certain position. Then, we started to move the camera. As the camera moves, some of the previously viewed points are lost and new points appear in the image. Finally, we added an omnidirectional camera viewing all the points and it is positioned at the half of the distance travelled by the perspective camera. A top view of the simulated environment is given in Fig. 8. Gaussian noise (with σ=1 pixel) is added to the pixel coordinates.

We measured the egomotion estimation drift for changing number of views (frames) to compare three cases; (i) initial estimation without bundle adjustment (BA), (ii) after perspective BA and (iii) after mixed camera BA. Please note that the same number of BA iterations was applied in both perspective and mixed cases and the experiments were repeated 50 times and the average drift was taken.
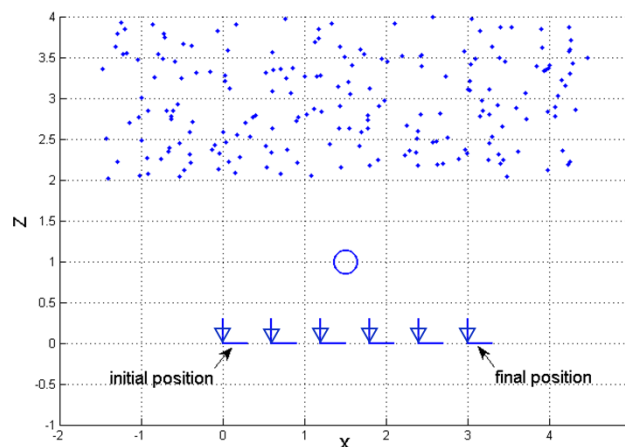

Figure 8: Top-view of the simulated environment with 6 perspective views. Perspective camera, pointing towards the scene points, is moving to the right. Omnidirectional camera is indicated with O. For fixed length experiment (Fig. 9a), different number of views (4 to 8) are taken with the same length of the camera path. For increasing length experiment (Fig. 9b), travelled distance by the perspective camera increases 60 cm with each view.
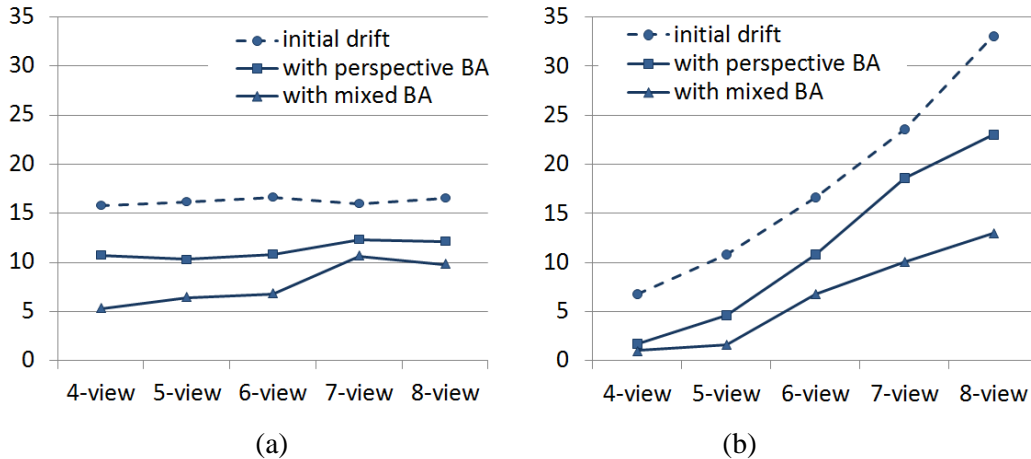
(a)   (b)

Figure 9: Egomotion estimation drift (y axis, in cm.) for compared approaches and changing number of perspective views (x axis). (a) The distance travelled by the perspective camera is fixed to 3 m. (b) The distance increases 60 cm with each view.

Fig. 9a shows the drift (y axis, in cm.) for changing number of perspective views (x axis) when the travelled distance by the perspective camera is fixed to 3 meters. We observe that changing the number of views within a fixed distance does not affect the initial drift. Although one may expect the drift to increase with more views, since the travelled distance is fixed, positions of the views come close to each other and the estimation of motion becomes more successful due to the increased number of matched points. Regarding the effect of the proposed approach of including an omnidirectional camera, it can be easily inferred from Fig. 9a that, bundle adjustment with the omnidirectional view (mixed BA) outperforms the perspective-only approach. The improvement gained by the omnidirectional view decreases with increasing number of perspective views. This is an expected result since the relative importance of the point correspondences coming from omnidirectional view decreases.

Fig. 9b shows the measured drifts when the travelled distance by the perspective camera increases 60 cm with each view, i.e. the total distances travelled in 4-view and 8-view sequences are 1.8 m and 4.2 m. respectively. Notice that 6-view data is the same in both graphs since it corresponds to the same setup, 6 views 60 cm apart. This time, the drift increases with the addition of new views. Mixed BA consistently gives better results than perspective-only BA, which indicates that adding an omnidirectional camera improves the accuracy of egomotion estimation. Fig. 10 can be given as a visual example to this experiment where the mixed approach (Fig. 10b) reduces the drift more than perspective-only approach does (Fig. 10a).

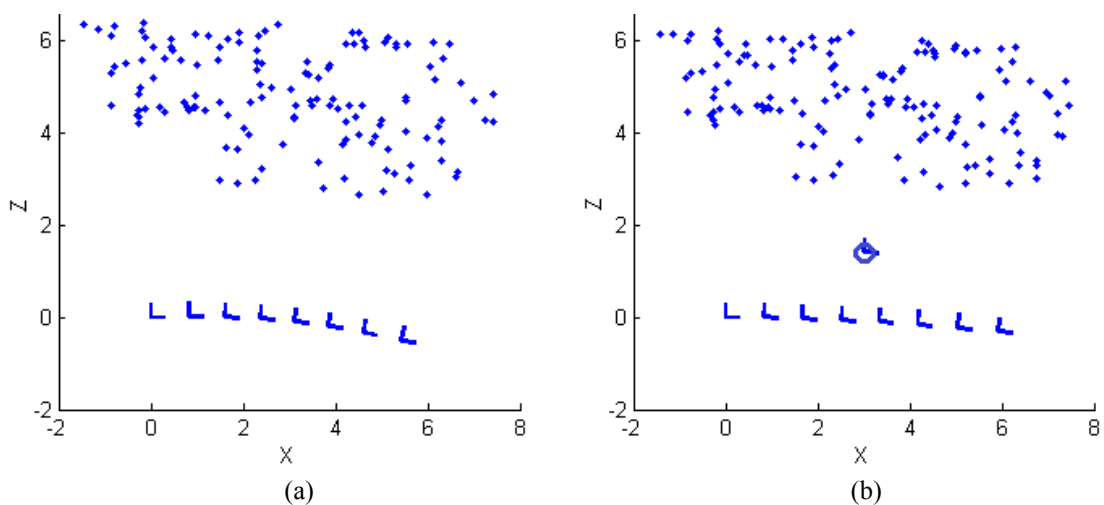(a)   (b)

Figure 10: Comparison of perspective-only BA (a) and mixed BA (b). The reconstructed camera positions are supposed to be on a straight line along x axis. Mixed BA reduces the drift more.

4.2 Experiments with real images

We performed an experiment where the positions of the camera were known while capturing the images of the scene, which enabled us to compare the estimated camera positions with the real ones. We positioned the viewing center of the camera to certain 2D coordinates on the floor via a tripod while taking images. One omnidirectional and seven perspective images used in this experiment are shown in Fig. 11. Fig. 12a shows the initial estimation of the structure and camera motion. The camera movement from the first to the last view is indicated with arrows. The actual distance between the consecutive views are equal to 40 cm (accomplished by precisely positioning the viewing center of the camera). A drift can easily be observed in the estimated motion towards the end of the sequence. Fig. 12b shows the result after perspective-only BA and Fig. 12c shows the result when the omnidirectional camera is added (mixed BA).

Figure 11: Omnidirectional and perspective images of the first real image experiment.

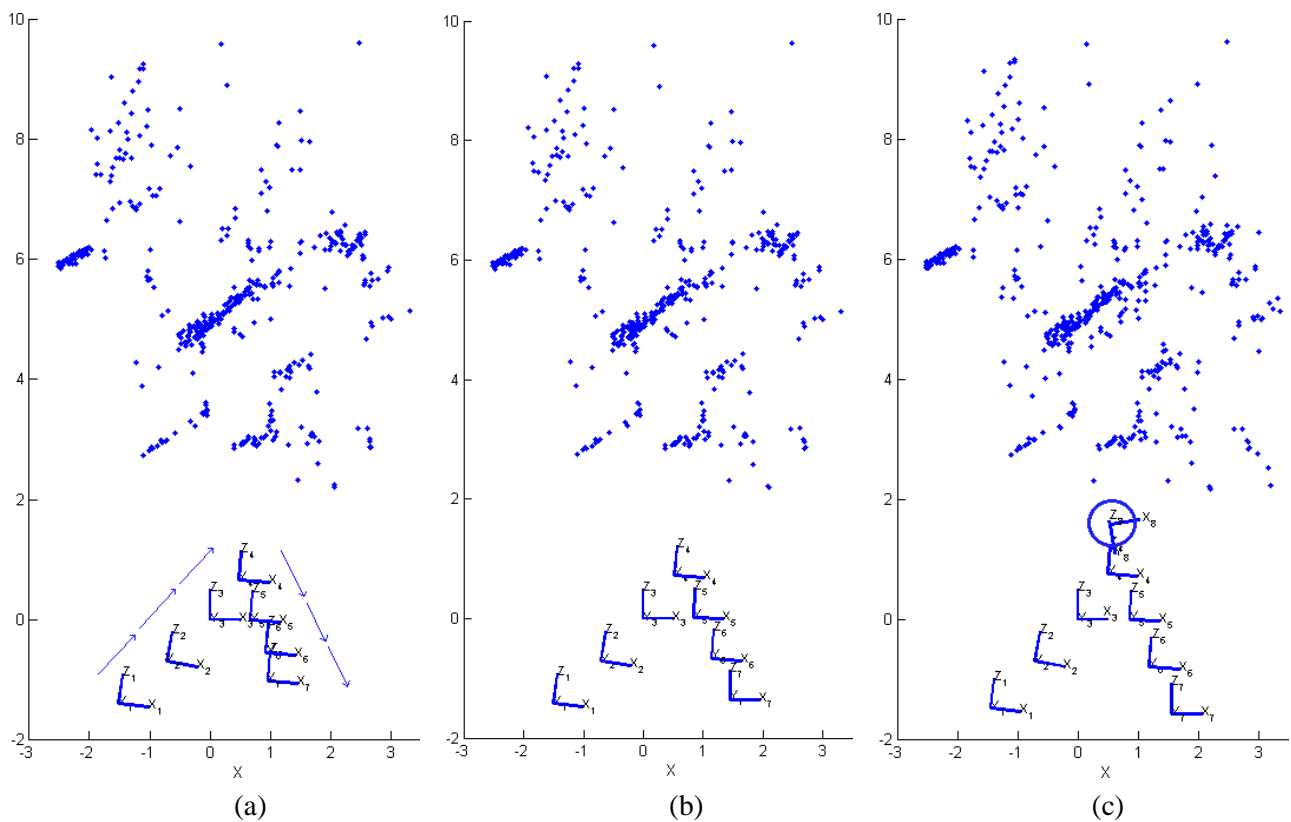(a)   (b)   (c)

Figure 12: Structure and motion estimated from images in Fig. 11. Figures show top-view and the z axes of the estimated perspective camera views are toward the reconstructed scene points. (a) The initial estimation with perspective-only SfM. The results after bundle adjustment for (b) perspective-only BA and (c) mixed BA. The omnidirectional camera is indicated with a circle around it, z axis of which is looking down.

Since the reconstructions are correct up to a scale factor, we normalized the measured drift using the known real world distances. Initial estimation drift is 27.5 cm whereas the drifts after perspective BA and mixed BA are 15.2 cm and 6.9 cm respectively. This result clearly indicates that adding an omnidirectional camera improves the accuracy of camera motion estimation. There are around 550 reconstructed points. Approximately, one fifth of the scene points have a match in the omnidirectional view i.e. one fifth of the points were reconstructed with the omnidirectional view and at least one of the perspective views.

Perhaps the easiest way to measure the drift is using a sequence of images that form a closed loop since the drift is the distance between the estimated positions of the first and last views. We performed such an experiment again with a sequence of six images which can be seen in Fig. 13.

Figure 13: Sample perspective and omnidirectional view from the experiment with a closed loop sequence.

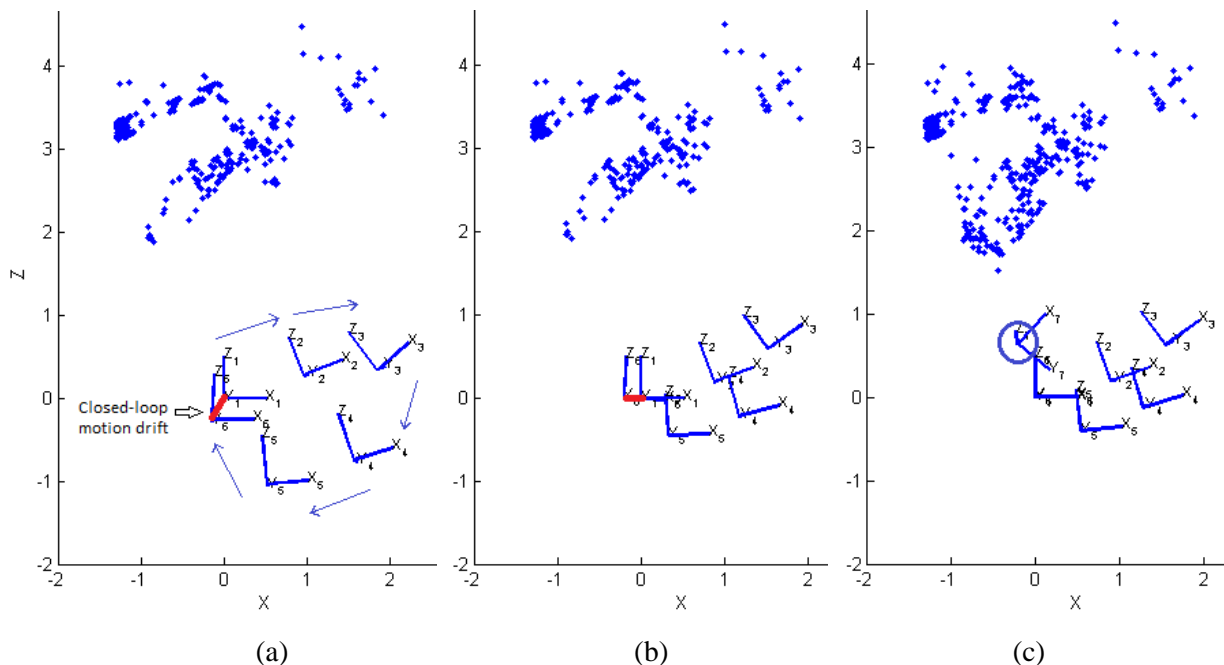(a)   (b)   (c)

Figure 14: SfM experiment with a closed loop sequence. Figures show top-view and the z axes of the estimated perspective camera views are toward the reconstructed scene points. (a) The initial estimation, the drift is depicted in red. (b) The result after perspective-only BA, the drift is depicted in red. (c) The result after mixed BA. The drift is almost disappeared.

The reader should note that, we pretended not to know that there is a closed loop and did not perform point matching between the first and the last frames because this normally requires a loop detection algorithm such as [6,7]. Here, we used the loop sequence to measure the drift easily.

After the SfM pipeline was performed, the drift is measured as the distance between the first and the last (sixth) camera positions, since these two views are actually the same. Fig. 14a shows the initial estimation of the structure and camera positions, the order of the views together and the drift. Fig. 14b shows the perspective-only BA result and Fig. 14c shows the result with the omnidirectional view. Normalized initial drift was 20.55 cm, drift values after bundle adjustment were 18.05 cm and 0.95 cm for perspective-only SfM and hybrid SfM, respectively. The improvement gained by the omnidirectional view in this experiment is more significant than the previous real image experiment.

Our experiments were restricted to ~8 views and ~500 3D points due to computational resources required for bundle adjustment. Bundle adjustment is an error minimization typically carried out with Levenberg-Marquardt algorithm. Due to the large number of unknowns contributing to the minimized reprojection error, the computation takes considerable amount of time. In our case these unknowns are point coordinates and camera extrinsic parameters. As the number of unknowns increases the required time increases exponentially.

The sparse bundle adjustment (SBA) implementation [21] which also uses Levenberg-Marquardt algorithm is much more efficient since it exploits the sparseness in the data. However, we could not use SBA for our hybrid case, since it does not offer the flexibility to define different intrinsic parameters for different camera types. In non-sparse bundle adjustment the number of parameters to be optimized is limited due to computational resources.

5   Conclusion

We have demonstrated the fact that the egomotion estimation of a moving camera suffers from a drift in the estimated path when a sequential algorithm is employed. We proposed an approach to increase the estimation accuracy, the main idea of which is using an omnidirectional camera in addition to the conventional camera. Taking advantage of the enlarged view, correspondences between the omnidirectional and perspective images improve the accuracy of camera position estimates. Unlike other methods in the literature, our approach does not assume loop closure.

We performed simulated and real image experiments to compare the estimation accuracy when only perspective views are used and when an omnidirectional view is added. Results show that the proposed idea of adding an omnidirectional camera reduces the drift.

In this paper, we did not work on egomotion estimation over long distances, rather we concentrated on SLAM and SfM which definitely requires an accurate camera pose estimation. In our experiments, the length of the image sequences was restricted since the bundle adjustment method we employed is computationally expensive. However, our method can also be valuable for visual odometry over long distances [2] if more efficient implementations of hybrid SfM become available. As the length of the sequence increases, number of omnidirectional views can be increased gradually. For respectively short sequences as we used in this work, increasing the accuracy of SfM can be useful for 3D reconstruction from multiple images like in [22].

References

[1]   P. Beardsley, A. Zisserman, D. Murray, "Sequential updating of projective and affine structure from motion", *International Journal of Computer Vision*, 23(3): 235–259, 1997.

[2]   D. Nister, O. Naroditsky, J. Bergen, "Visual odometry", *European Conference on Computer Vision*, 2004.

[3]   N. Snavely, S.M. Seitz, R. Szeliski, "Modeling the world from Internet photo collections". *International Journal of Computer Vision*, 2008, 80(2). <http://dx.doi.org/10.1007/s11263-007-0107-3>

[4]   O. Enqvist, F. Kahl, C. Olsson, Non-sequential structure from motion, Proc. of IEEE Workshop on Omnidirectional Vision, 2011. <http://dx.doi.org/10.1109/iccvw.2011.6130252>

[5]   K. Cornelis, F. Verbiest, L. Van Gool, "Drift detection and removal for sequential structure from motion algorithms", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10): 1249-1259, 2004. <http://dx.doi.org/10.1109/tpami.2004.85>

[6]   K.L. Ho, P. Newman, "Detecting loop closure with scene sequences", *Int. Journal of Computer Vision*, 74(3): 261-286, 2007. <http://dx.doi.org/10.1007/s11263-006-0020-1>

[7]   M. Cummins, P. Newman, "Probabilistic appearance based navigation and loop closing", *Int. Conference on Robotics and Automation* (ICRA), 2007. <http://dx.doi.org/10.1109/robot.2007.363622>

[8]   P. Chang, M. Hebert, Omnidirectional Structure from Motion, Proc. of IEEE Workshop on Omnidirectional Vision, 2000. <http://dx.doi.org/10.1109/omnvis.2000.853819>

[9]   D. Scaramuzza, R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles", *IEEE Transactions on Robotics*, 24(5): 1015-1026, 2008. <http://dx.doi.org/10.1109/tro.2008.2004490>

[10]   P. Sturm, Mixing Catadioptric and Perspective Cameras, in: Proc. of Workshop on Omnidirectional Vision (OMNIVIS), 37–44, 2002. <http://dx.doi.org/10.1109/omnvis.2002.1044489>

[11]   J. Barreto, K. Daniilidis, Epipolar Geometry of Central Projection Systems using Veronese Maps, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1258–1265, 2006. <http://dx.doi.org/10.1109/cvpr.2006.111>

[12]   S. Ramalingam, S. Lodha, P. Sturm, A Generic Structure-from-motion Algorithm for Cross-camera Scenarios, in: Proc. of IEEE Workshop on Omnidirectional Vision, 2004.

[13]   D. Lowe, "Distinctive image features from scale invariant keypoints", *International Journal of Computer Vision*, 60(2): 91-110, 2004. <http://dx.doi.org/10.1023/b:visi.0000029664.99615.94>

[14]   Y. Bastanlar, A. Temizel, Y. Yardimci, P. Sturm, "Multi-view structure-from-motion for hybrid camera scenarios", *Image and Vision Computing*, 30(8): 557–572, 2012. http://dx.doi.org/10.1016/<j.imavis.2012.06.001>

[15]   C. Geyer, K. Daniilidis, "A unifying theory for central panoramic systems", *European Conference on Computer Vision* (ECCV), 2000.

[16]   X. Ying, Z. Hu, "Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model?", *European Conference on Computer Vision* (ECCV), 2004. <http://dx.doi.org/10.1007/978-3-540-24670-1_34>

[17]   Y. Bastanlar, A. Temizel, Y. Yardimci, "Improved SIFT matching for image pairs with scale difference", *Electronics Letters*, 46(5): 346–348, 2010. <http://dx.doi.org/10.1049/el.2010.2548>

[18]   M. Fischler, R. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM 24(6) (1981) 381–395.

[19]   R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge Univ. Press, 2nd Ed., 2004. <http://dx.doi.org/10.1007/978-0-387-31439-6_100010>

[20]   R. Hartley, P. Sturm, "Triangulation", *Computer Vision and Image Understanding*, 68(2): 146–157, 1997.

[21]   M. Lourakis, A. Argyros, "A generic sparse bundle adjustment C/C++ package based on the Levenberg-Marquardt algorithm", http://www.ics.forth.gr/~lourakis/sba/, accessed 30th June 2014.

[22]   ARC 3D Webservice: A Family of Web Tools for Remote 3D Reconstruction, http://www.arc3d.be/ accessed 30th June 2014.