

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224135244>

From Google Street View to 3D City models

Conference Paper · November 2009

DOI: 10.1109/ICCVW.2009.5457551 · Source: IEEE Xplore

CITATIONS

78

READS

9,086

3 authors, including:



Michal Havlena

Parametric Technology Corporation

51 PUBLICATIONS 1,123 CITATIONS

[SEE PROFILE](#)



Tomas Pajdla

Czech Technical University in Prague

265 PUBLICATIONS 16,097 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



3D camera calibration [View project](#)

From Google Street View to 3D City Models

Akihiko Torii Michal Havlena Tomáš Pajdla
Center for Machine Perception, Department of Cybernetics
Faculty of Elec. Eng., Czech Technical University in Prague
{torii,havlem1,pajdla}@cmp.felk.cvut.cz

Abstract

We present a structure-from-motion (SfM) pipeline for visual 3D modeling of a large city area using 360° field of view Google Street View images. The core of the pipeline combines the state of the art techniques such as SURF feature detection, tentative matching by an approximate nearest neighbour search, relative camera motion estimation by solving 5-pt minimal camera pose problem, and sparse bundle adjustment. The robust and stable camera poses estimated by PROSAC with soft voting and by scale selection using a visual cone test bring high quality initial structure for bundle adjustment. Furthermore, searching for trajectory loops based on co-occurring visual words and closing them by adding new constraints for the bundle adjustment enforce the global consistency of camera poses and 3D structure in the sequence. We present a large-scale reconstruction computed from 4,799 images of the Google Street View Pittsburgh Research Data Set.

1. Introduction

Large scale 3D models of cities built from video sequences acquired by car mounted cameras provide richer 3D contents than those built from aerial images only. A virtual reality system covering the whole world can be brought by embedding such 3D contents into Google Earth or Microsoft Virtual Earth in near future. In this paper, we present a structure-from-motion (SfM) pipeline for visual 3D modeling of such a large city area using 360° field of view omnidirectional images.

Recently, work [27] demonstrated 3D modeling from perspective images exported from Google Street View images using piecewise planar structure constraints. Another recent related work [38] demonstrated the performance of the SfM which employs the guided matching by using epipolar geometries computed in previous frames, and the robust camera trajectory estimation by computing camera orientations and positions individually for the calibrated perspective images acquired by Point Grey Ladybug Spher-

ical Digital Video Camera System [32]. This paper shows a large scale sparse 3D reconstruction using the original omnidirectional panoramic images.

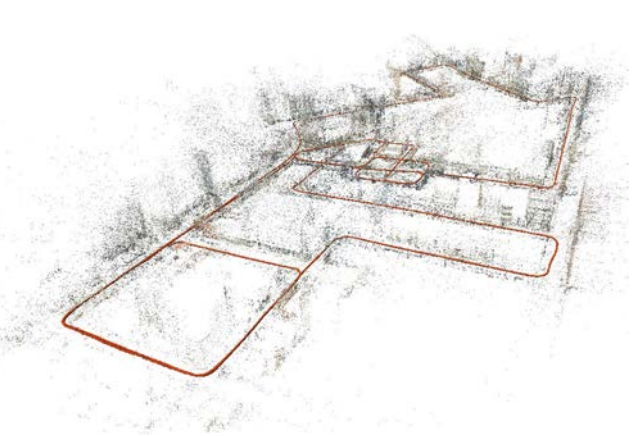
Previously, city reconstruction has been addressed using aerial images [9, 3, 10, 22, 40, 41] which allowed reconstructing large areas from a small number of images. The resulting models, however, often lacked visual realism when viewed from the ground level since it was impossible to texture the facades of the buildings.

A framework for city modeling from ground-level image sequences working in real-time has been developed, *e.g.* in [1] and [5]. Work [5] uses SfM to reconstruct camera trajectories and 3D key points in the scene, fast dense image matching, assuming that there is a single gravity vector in the scene and all the building facades are ruled surfaces parallel to it. The system gives good results but 3D reconstruction could not survive sharp camera turns when a large part of the scene moved away from the limited view field of cameras. A recent extension of [5] using a pair of calibrated fisheye lens cameras [12], which have hemispherical fields of view, could successfully reconstruct a trajectory with sharp turns. In this work, we assume a single moving camera which provides sparse image sequences only.

Short baseline SfM using simple image features [5], which performs real-time detection and matching, recovers camera poses and trajectory sufficiently well when all camera motions between consecutive frames in the sequence are small. On the other hand, wide baseline SfM based methods, which use richer features such as MSER [25], Laplacian-Affine, Hessian-Affine [28], SIFT [21], and SURF [2], are capable of producing feasible tentative matches under large changes of visual appearance between images induced by rapid changes of camera pose and illumination. Work [7] presented the SfM based on wide baseline matching of SIFT features using a single omnidirectional camera and demonstrated the performance on indoor environments. We use SURF features [2] since they are the fastest among those features used for the wide baseline matching and produce sufficiently robust tentative matches even on distorted omnidirectional images.



(a)



(b)

Figure 1. Camera trajectory computed by SfM. (a) Camera positions (red circles) exported into Google Earth [8]. To increase the visibility, every 12th camera position in the original sequence is plotted. (b) The 3D model representing 4,799 camera positions (red circles) and 123,035 3D points (color dots).

The problem inevitable for sequential SfM is to have drift errors accumulated while proceeding along the trajectory. Loop closing [16, 34] is essentially capable of removing the drift errors since it brings the global consistency of camera poses and 3D structures by giving additional constraints for the final refinement accomplished by bundle adjustment. In [16], the loop closing is achieved by merging partial reconstructions of overlapping sequences which are extracted using an image similarity matrix [36, 17]. Work [34] finds loop endpoints by using the image similarity matrix and verifies the loops by computing the rotation transform between the pairs of origins and endpoints under the assumption that the position of the origin and the endpoint of each loop coincide. Furthermore, they constraint the camera motions on a plane to reduce the number of parameters in bundle adjustment. Unlike in [34], we aim at proposing a pipeline which recovers camera poses in 3D and tests the loops by solving camera resectioning [31] in order to accomplish large scale 3D modeling of cities, see Figure 1.

The main contribution of this paper is in demonstrating that one can achieve SfM from a single sparse omnidirectional sequence with only an approximate knowledge of calibration as opposed to [5, 38] where the large scale models are computed from dense sequences and with precisely calibrated cameras. We present an experiment with the Google Street View Pittsburgh Research Data Set¹, which has denser images than data freely available at Google Maps. Therefore, we processed every second image and could have processed even every fourth image with a small degradation of the results.

¹Provided and copyrighted by Google.

2. The Pipeline

The proposed SfM pipeline is an extension of the previous work [39] which demonstrated the performance of the recovery of camera poses and trajectory on the image sequence acquired by a single fisheye lens camera. We refer [39] for more technical details of each step in the pipeline.

2.1. Calibration

Assuming that the input omnidirectional images are produced by the equirectangular projection, see Figure 2, the transformation from image points to unit vectors of their rays can be formulated as follows. For the equirectangular image having the dimensions I_W and I_H , a point $\mathbf{u} = (u_i, u_j)^\top$ in the image coordinates is transformed into a unit vector $\mathbf{p} = (p_x, p_y, p_z)^\top$ in spherical coordinates:

$$p_x = \cos \phi \sin \theta, \quad p_y = \sin \phi, \quad p_z = \cos \phi \cos \theta. \quad (1)$$

where angles θ and ϕ are computed as:

$$\theta = \left(u_i - \frac{I_W}{2} \right) \frac{2\pi}{I_W}, \quad (2)$$

$$\phi = \left(u_j - \frac{I_H}{2} \right) \frac{\pi}{I_H}. \quad (3)$$

2.2. Generating Tracks by Concatenating Pairwise Matches

Tracks used for SfM are generated in several steps. First, up to thousands of SURF features [2] are detected and described on each of the input images.

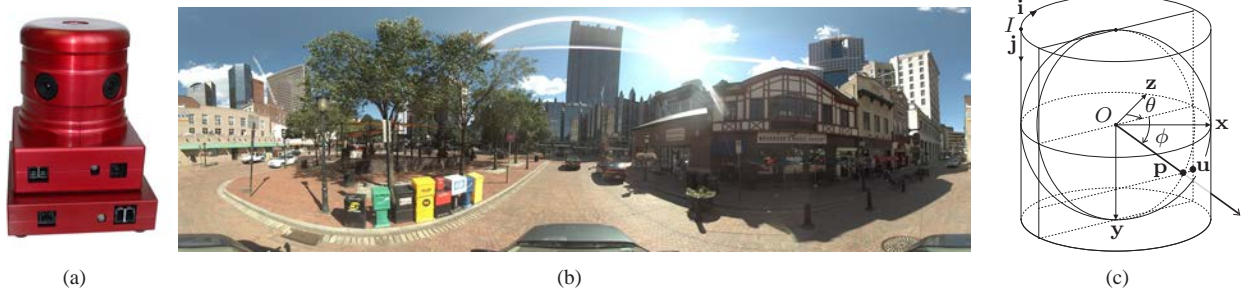


Figure 2. Omnidirectional imaging. (a) Point Grey Ladybug Spherical Digital Video Camera System [32] used for acquiring the Street View images. (b) Omnidirectional image used as input data for SfM. (c) Transformation between a unit vector \mathbf{p} on a unit sphere and a pixel \mathbf{u} of the equirectangular image. The coordinates p_x , p_y , and p_z of the unit vector \mathbf{p} are transformed into angles θ and ϕ . Column index u_i is computed from the angle θ and row index u_j from the angle ϕ .

Secondly, sets of tentative matches are constructed between pairs of consecutive images. The matching is achieved by finding features with closest descriptors between the pair of images, which is done for each feature independently. When conflicts appear, we select the most discriminative match by computing the ratio between the first and the second best match. We use Fast Library for Approximate Nearest Neighbors (FLANN) [29] which delivers approximate nearest neighbours significantly faster than exact matching thanks to using several random kd-trees.

Thirdly, tentative matches between each pair of consecutive images are verified through epipolar geometry (EG) computed by solving the 5-point minimal relative pose problem for calibrated cameras [30]. The tentative matches are verified with a RANSAC based robust estimation [6] which searches for the largest subset of the set of tentative matches consistent with the given epipolar geometry. We use PROSAC [4], a simple modification of RANSAC, which brings a good performance [33] because of reducing the number of samples by using the ordered sampling [4]. The 5-tuples of tentative matches are drawn from the list ordered ascendingly by their discriminativity scores, which are the ratios between the distances of the first and the second nearest neighbours in the feature space. Finally, the tracks are constructed by concatenating inlier matches.

The pairwise matches, obtained by epipolar geometry validation, often contain incorrect matches lying on epipolar lines or in the vicinity of epipoles since they may support the epipolar geometry even without violating geometric consistency. In practice, such incorrect matches can be mostly filtered out by selecting only the tracks having a longer length. We reject tracks containing less than three features.

2.3. Robust Initial Camera Pose Estimation

Initial camera poses and positions in a canonical coordinate system are recovered by using the epipolar geometries of pairs of consecutive images computed in the stage

of verifying tracks. The essential matrix \mathbf{E}_{ij} , encoding the relative camera pose between frames i and $j = i + 1$, can be decomposed into $\mathbf{E}_{ij} = [\mathbf{t}_{ij}]_{\times} \mathbf{R}_{ij}$. 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 [11] of the i -th frame $\mathbf{P}_i = [\mathbf{R}_i | \mathbf{T}_i]$, the normalized camera matrix \mathbf{P}_j can be computed by

$$\mathbf{P}_j = [\mathbf{R}_{ij} \mathbf{R}_i | \mathbf{R}_{ij} \mathbf{T}_i + \gamma \mathbf{t}_{ij}] \quad (4)$$

where γ is the scale of the translation between frames i and j in the canonical coordinate system. The scale γ can be computed by 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 possible candidates of scales has to be done while evaluating the quality of the computed camera position. The best scale is found by RANSAC maximizing the number of points that pass the ‘‘cone test’’ [13] which checks the intersection of pixel ray cones in a similar way as the feasibility test of L_1 - or L_∞ -triangulation [14, 15], see Algorithm 1. During the cone test, one pixel wide cones formed by four planes (up, down, left, and right) are casted around the matches and we test whether the intersection of the cones is empty or not using the LP feasibility test [23] or an exhaustive test [13] which is faster when the number of the intersected cones is smaller than four.

2.4. Bundle Adjustment Enforcing Global Camera Pose Consistency

Even though the Google Street View data is not primarily acquired by driving the same street several times, there are some overlaps suitable for constructing loops that can compensate drift errors induced while proceeding the trajectory sequentially. We construct loops by searching pairs of images observing the same 3D structure in different times in the sequence.

Algorithm 1 Construction of the Initial Camera Poses by Chaining Epipolar Geometries

Input $\{\mathbf{E}_{i,i+1}\}_{i=1}^{n-1}$ Epipolar geometries of pairs of consecutive images.
 $\{\mathbf{m}_i\}_{i=1}^{n-1}$ Matches (tracks) supporting the epipolar geometries.

Output $\{\mathbf{P}_i\}_{i=1}^n$ Normalized camera matrices.

- 1: $\mathbf{P}_1 := [\mathbf{I}^{3 \times 3} \mid \mathbf{0}^{3 \times 1}]$... Set the first camera to be the origin of the canonical coordinates.
- 2: **for** $i := 1, \dots, n - 1$ **do**
- 3: Decompose $\mathbf{E}_{i,i+1}$ and select the right rotation \mathbf{R} and translation \mathbf{t} where $\|\mathbf{t}\| = 1$.
- 4: $\{\mathbf{U}_i\} := 3\text{D}$ points computed by triangulating the matches $\{\mathbf{m}_i\}_i^{i+1}$ using \mathbf{R} and \mathbf{t}
- 5: **if** $i = 1$ **then**
- 6: $\mathbf{P}_{i+1} := [\mathbf{R}\mathbf{A} \mid \mathbf{R}\mathbf{b} + \mathbf{t}]$ where $\mathbf{P}_i = [\mathbf{A} \mid \mathbf{b}]$.
- 7: $\{\mathbf{X}\} := \{\mathbf{U}_i\}$... Update 3D points
- 8: **else**
- 9: Find 3D points $\{\mathbf{U}_{i-1,i+1}\}$ in $\{\mathbf{U}_i\}$ in the i th camera coordinates seen in three images.
- 10: Find 3D points $\{\mathbf{X}_{i-1,i+1}\}$ in $\{\mathbf{X}\}$ in the canonical coordinates seen in three images.
- 11: $t := 0, S_{\max} := 0, N := |\{\mathbf{X}_{i-1,i+1}\}|$... Initialization for RANSAC cone test.
- 12: **while** $t \leq N$ **do**
- 13: $t := t + 1$... New sample.
- 14: $\gamma := \|\mathbf{X}_{i-1,i+1}\| / \|\mathbf{A}^\top (\mathbf{U}_{i-1,i+1} - \mathbf{b})\|$... The scale to be tested.
- 15: $\mathbf{P}_t := [\mathbf{R}\mathbf{A} \mid \mathbf{R}\mathbf{b} + \gamma\mathbf{t}]$ where $\mathbf{P}_i = [\mathbf{A} \mid \mathbf{b}]$.
- 16: $S_t :=$ the number of matches $\{\mathbf{m}_i\}_{i-1}^{i+1}$ which are consistent with the motions $\mathbf{P}_{i-1}, \mathbf{P}_i$ and \mathbf{P}_t .
- 17: **if** $S_t > S_{\max}$ **then**
- 18: $\mathbf{P}_{i+1} := \mathbf{P}_t$... The best motion with scale so far.
- 19: $S_{\max} := S_t$... The maximum number of supports so far.
- 20: Update the termination length N .
- 21: **end if**
- 22: **end while**
- 23: Update $\{\mathbf{X}\}$ by merging $\{\mathbf{U}_{i-1,i+1}\}$ and adding $\{\mathbf{U}_i\} \setminus \{\mathbf{U}_{i-1,i+1}\}$
- 24: **end if**
- 25: **end for**

The knowledge of GPS locations of Street View images truly alleviates the problem of image matching for loop closing but does not completely reduce it since common 3D structures can be seen even among relatively distant images. In this paper, we do not rely on GPS locations because the image matching achieved by using the image similarity matrix is potentially capable to match such distant images and it is always important for the vision community to see that certain problem can be solved entirely using vision.

Building Image Similarity Matrix SURF descriptors of each image are quantized into visual words using the visual vocabulary containing 130,000 words computed from urban area omnidirectional images. Next, term frequency-inverse document frequency (tf-idf) vectors [36, 17], which weight words occurring often in a particular document and downweight words that appear often in the database, are computed for each image with more than 50 detected visual words. Finally, the image similarity matrix \mathbf{M} is constructed by computing the image similarities, which we define as cosines of angles between normalized tf-idf vectors, between all pairs of images.

Loop Finding and Closing First, we take the upper triangular part of \mathbf{M} to avoid duplicate search. Since the diagonal entries of \mathbf{M} which are the neighbouring frames in the sequence essentially have high scores, the 1st to 50th diagonals are zeroed in order to exclude very small loops. Next, for the image I_i in the sequence, we select the image I_j as the one having the highest similarity score in the i th row of \mathbf{M} . Image I_j is a candidate of the endpoint of the loop which starts from I_i . Note that the use of an upper triangular matrix constraints $j > i$.

Next, the candidate image I_j is verified by solving the camera resectioning [31]. Triplets of the tentative 2D-3D matches constructed by matching the descriptors of 3D points associated to the images I_i and I_{i+1} with the descriptors of the features detected in the image I_j are sampled by RANSAC to find the camera pose having the largest support evaluated by the cone test again. The image I_{i+1} , which is the successive frame of I_i , is additionally used for performing the cone test with three images in order to enforce geometric consistencies in the support evaluation of the RANSAC. Local optimization is achieved by repeated camera pose computation from all inliers [35] via SDP and

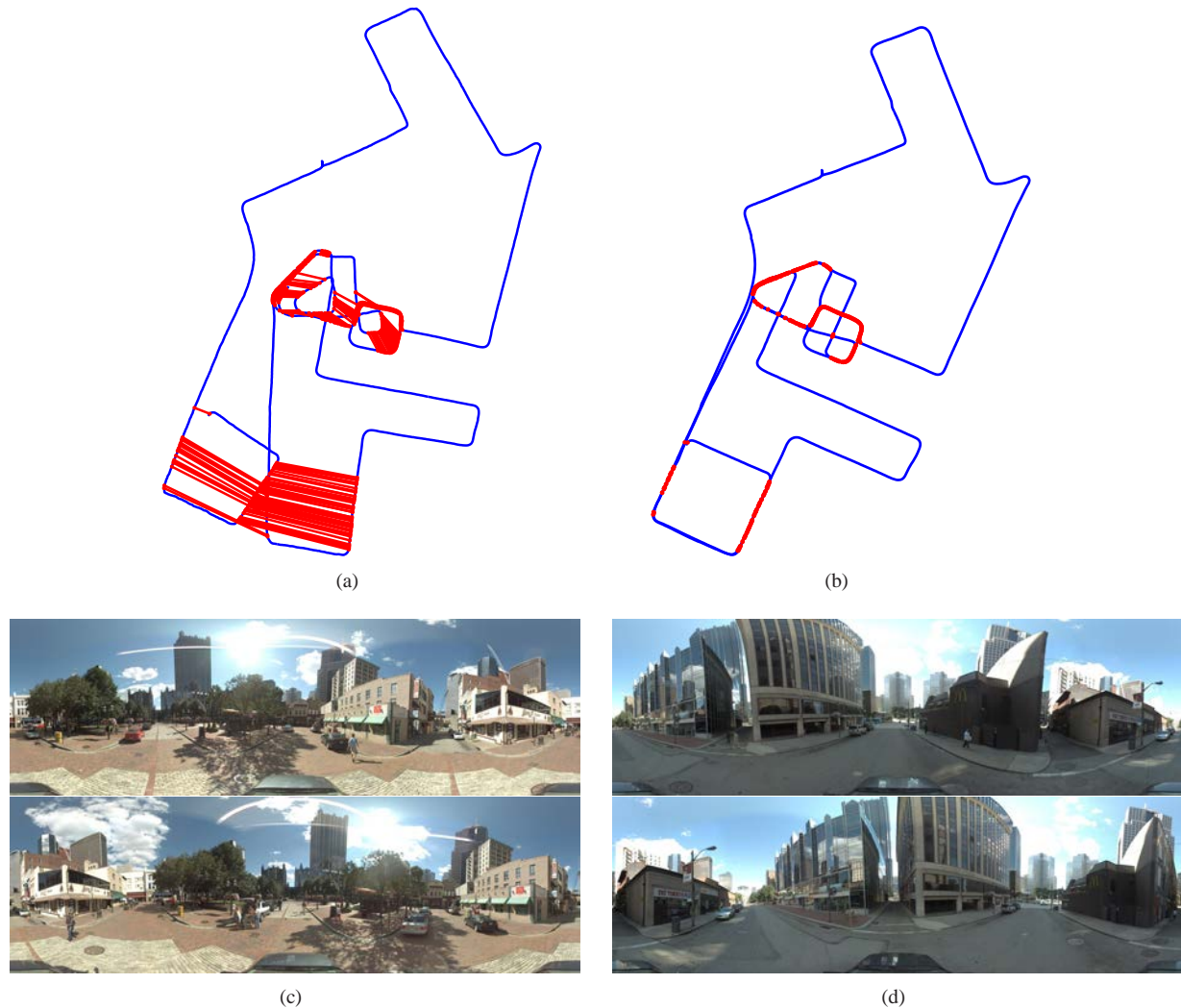


Figure 3. Results of SfM with loop closing. (a) Trajectory before bundle adjustment. (b) Trajectory after bundle adjustment with loop closing. Examples of the images used for the loop closing: (c) Frames 6597 and 8643. (d) Frames 6711 and 6895.

SeDuMi [37]. If the inlier ratio is higher than 70%, the camera resectioning is considered successful and the candidate image I_j is accepted as the endpoint of the loop. The inlier matches are used to give additional constraints on the final bundle adjustment. We perform this loop search for every image in the sequence and test only the pair of images having the highest similarity score. If one increased the number of candidates to be tested, our pipeline would approach SfM [24, 19, 26] for unorganized images based on exhaustive pairwise matching.

Finally, very distant points, *i.e.* likely outliers, are filtered out and sparse bundle adjustment [20] modified in order to work with unit vectors, which is the approach similar to [18], refines both points and cameras.

3. Experimental Results

We used 4,799 omnidirectional images of the Google Street View Pittsburgh Research Data Set. Since the input omnidirectional images have large distortion at the top and bottom, we clipped original images by cropping 230 pixels from the top and 410 pixels from the bottom to obtain $3,328 \times 1,024$ pixel large images, see Figure 2(b). Since the tracks are generated based on wide baseline matching, it is possible to save computation time by constructing initial camera poses and 3D structure from a sparser image sequence. Our SfM was run on every second image in the sequence, *i.e.* 2,400 images were used to create a global reconstruction. The remaining 2,399 images were attached to the reconstruction in the final stage.

The initial camera poses were estimated by comput-

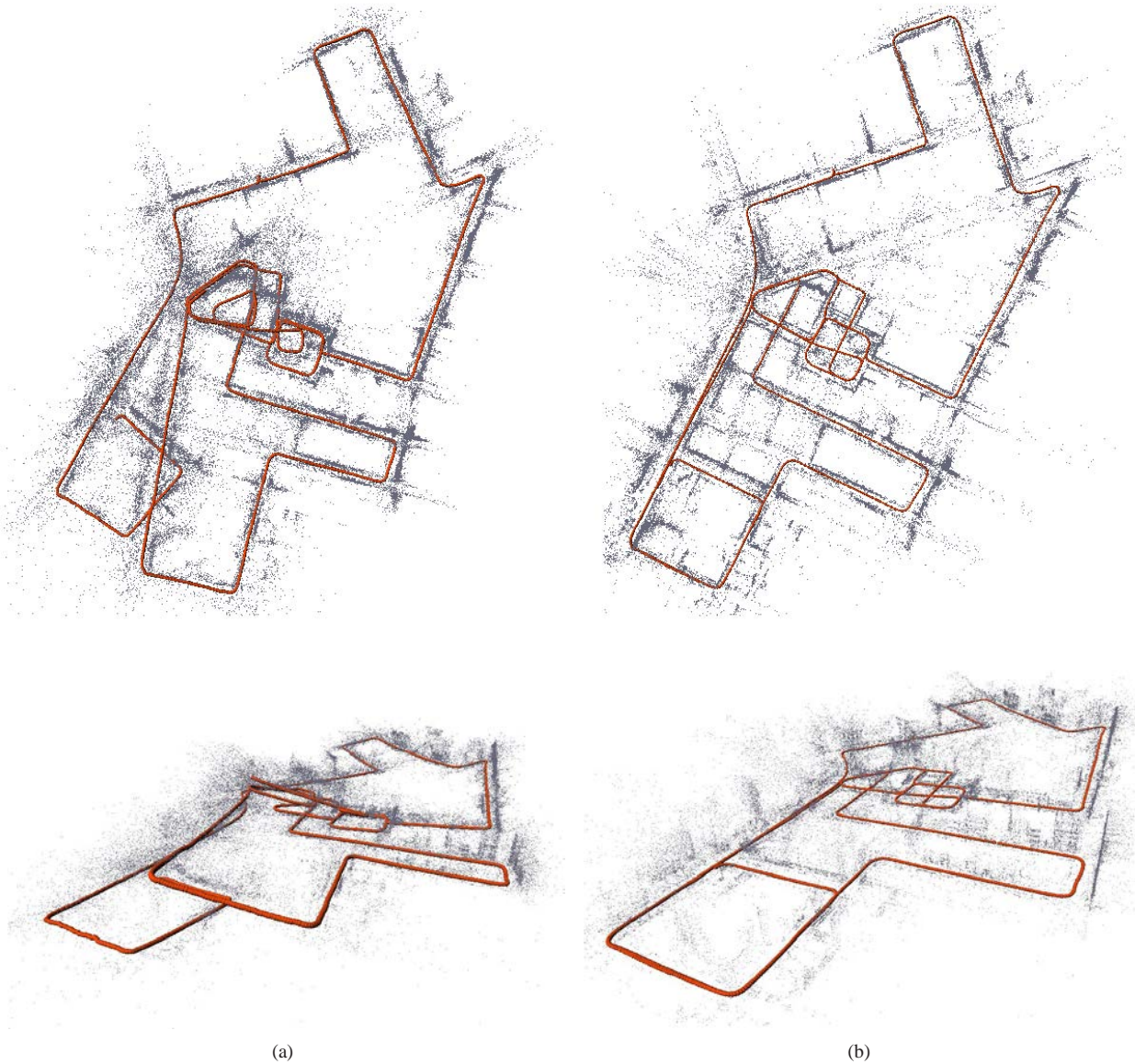


Figure 4. Resulted 3D model consisting of 2,400 camera positions (red circles) and 124,035 3D points (blue dots) recovered by our pipeline. (a) Initial estimation. (b) After bundle adjustment with loop closing.

ing epipolar geometries of pairs of successive images, and chaining them by finding the global scale of camera translation, see Algorithm 1. The resulting trajectory is shown in Figure 3(a). After estimating the initial camera poses and reconstructing 3D points, the pairs of images acquired at the same location in different times were searched for. The red lines in Figure 3(a) indicate links between the accepted image pairs. Figure 3(b) shows the camera trajectory after the bundle adjustment with the additional constraints obtained from loop closing. Figures 3(c) and (d) show the examples of pairs of images used for closing the loops at frames (6597, 8643) and (6711, 6895) respectively. Furthermore,

Figure 4 shows the camera positions and the 3D points of the initial recovery (a) and after the loop closing (b) in different views. In Figure 5, the recovered trajectory is compared to the GPS positions provided in the Google Street View Pittsburgh Research Data Set. The computational time spent in different steps of the pipeline implemented in MATLAB+MEX running on a standard Core2Duo PC is shown in Table 1. Since the method is scalable and therefore storing the intermediate results of the computation on a hard drive instead of in RAM, performance could be improved by using a fast SSD drive instead of a standard SATA drive.

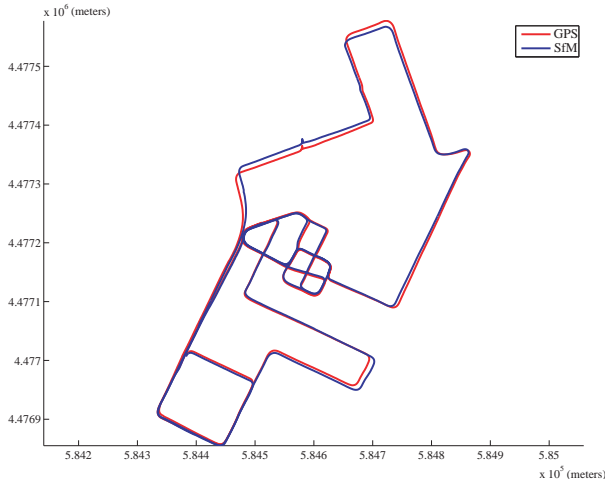


Figure 5. Comparison to the GPS provided in the Google Street View Pittsburgh Research Data Set. Camera trajectory by GPS (red line) and estimated camera trajectory by our SfM (blue line).

Detection	12.8
Matching	4.5
Chaining	1.0
Loop Closing	6.3
Bundle	14.5

Table 1. Computational time in hours. (Detection) SURF detection and description. (Matching) Tentative matching and computing EGs. (Chaining) Chaining EGs and computing scales. (Loop Closing) Searching and testing loops. (Bundle) Final sparse bundle adjustment.

Finally, the remaining 2,383 camera poses were computed by solving the camera resectioning in the same manner as used in the loop verification. Linear interpolation was used for the 16 cameras that could not be resectioned successfully. Figure 1(b) shows the 4,799 camera positions (red circles) and the 124,035 world 3D points (color dots) of the resulted 3D model.

4. Conclusions

We demonstrated the recovery of camera trajectory and 3D structure of a large city area from omnidirectional images and showed that the world can in principle be reconstructed from Google Street View images. We also showed that finding loops and using additional constraints on final bundle adjustment significantly improve the qualities of resulting camera trajectory and 3D structures. Since the street view images on Google Maps are approximately 10 times sparser than the original sequence from the Google Street View Pittsburgh Research Data Set, testing the performance of the proposed pipeline on such sparse sequences will be our next challenge.

Acknowledgment

The authors were supported by EC project FP6-IST-027787 DIRAC. T. Pajdla was supported by Czech Government under the research program MSM-684 0770038. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

References

- [1] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénus, R. Yang, G. Welch, H. Towles, D. Nistér, and M. Pollefeys. Towards urban 3d reconstruction from video. In *3DPVT06*, May 2006.
- [2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *CVIU*, 110(3):346–359, June 2008.
- [3] C. Brenner and N. Haala. Fast production of virtual reality city models. *IAPRS98*, 32(4):77–84, 1998.
- [4] O. Chum and J. Matas. Matching with PROSAC: Progressive sample consensus. In *CVPR05*, pages I: 220–226, 2005.
- [5] N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *CVPR06*, pages 1339–1344, 2006.
- [6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.
- [7] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool. Omnidirectional vision based topological navigation. *IJCV*, 74(3):219–236, 2007.
- [8] Google. Google earth - <http://earth.google.com/>, 2004.
- [9] A. Grün. Automation in building reconstruction. In *Photogrammetric Week '97*, pages 175–186, 1997.
- [10] N. Haala, C. Brenner, and C. Stätter. An integrated system for urban model generation. In *ISPRS Congress Comm. II*, pages 96–103, 1998.
- [11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.
- [12] M. Havlena, T. Pajdla, and K. Cornelis. Structure from omnidirectional stereo rig motion for city modeling. In *VIS-APP08*, pages II: 407–414, 2008.
- [13] M. Havlena, A. Torii, and T. Pajdla. Randomized structure from motion based on atomic 3d models from camera triplets. In *CVPR09*, 2009.
- [14] F. Kahl. Multiple view geometry and the L_∞ -norm. In *ICCV05*, pages II: 1002–1009, 2005.
- [15] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *PAMI*, 29(10):1834–1847, 2007.
- [16] M. Klopschitz, C. Zach, A. Irschara, and D. Schmalstieg. Generalized detection and merging of loop closures for video sequences. In *3DPVT*, 2008.
- [17] J. Knopp, J. Sivic, and T. Pajdla. Location recognition using large vocabularies and fast spatial matching. Research Report CTU-CMP-2009-01, CMP Prague, January 2009.

- [18] M. Lhuillier. Effective and generic structure from motion using angular error. In *ICPR06*, pages I: 67–70, 2006.
- [19] X. Li, C. Wu, C. Zach, S. Lazebnik, and J. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV08*, pages I: 427–440, 2008.
- [20] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Tech. Report 340, Institute of Computer Science – FORTH, August 2004.
- [21] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.
- [22] H. Maas. The suitability for airborne laser scanner data for automatic 3d object reconstruction. In *Ascona01*, pages 291–296, 2001.
- [23] A. Makhorin. GLPK: GNU linear programming kit - <http://www.gnu.org/software/glpk>, 2000.
- [24] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR07*, 2007.
- [25] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *IVC*, 22(10):761–767, September 2004.
- [26] Microsoft. Photosynth - <http://livelabs.com/photosynth>, 2008.
- [27] B. Micusik and J. Kosecka. Piecewise planar city 3d modeling from street view panoramic sequences. In *CVPR09*, 2009.
- [28] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 65(1-2):43–72, 2005.
- [29] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP09*, 2009.
- [30] D. Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6):756–770, June 2004.
- [31] D. Nister. A minimal solution to the generalized 3-point pose problem. In *CVPR04*, pages I: 560–567, 2004.
- [32] I. Point Grey Research. Ladybug2 - <http://www.ptgrey.com/products/ladybug2/index.asp>, 2005.
- [33] R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *ECCV08*, pages 500–513, 2008.
- [34] D. Scaramuzza, F. Fraundorfer, R. Siegwart, and M. Pollefeys. Closing the loop in appearance guided SfM for omnidirectional cameras. In *OMNIVIS08*, 2008.
- [35] G. Schweighofer and A. Pinz. Globally optimal $O(n)$ solution to the PnP problem for general camera models. In *BMVC08*, 2008.
- [36] J. Sivic and A. Zisserman. Video google: Efficient visual search of videos. In *CLOR06*, pages 127–144, 2006.
- [37] J. Sturm. Sedumi: A software package to solve optimization problems - <http://sedumi.ie.lehigh.edu>, 2006.
- [38] J. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IROS08*, 2008.
- [39] A. Torii, M. Havlena, and T. Pajdla. Omnidirectional image stabilization by computing camera trajectory. In *PSIVT09*, pages 71–82, 2009.
- [40] C. Vestri and F. Devernay. Using robust methods for automatic extraction of buildings. In *CVPR01*, pages I:133–138, 2001.
- [41] G. Vosselman and S. Dijkman. Reconstruction of 3d building models from laser altimetry data. *IAPRS01*, 34(3):22–24, 2001.