STRUCTURE FROM OMNIDIRECTIONAL STEREO RIG MOTION FOR CITY MODELING

Michal Havlena, Tomáš Pajdla
CMP, Department of Cybernetics, CTU in Prague, Czech Republic

Kurt Cornelis
PSI-VISICS, ESAT, Katholieke Universiteit Leuven, Belgium

Keywords: Structure from Motion, City Modeling, Omnidirectional Vision.

Abstract: This paper deals with a step towards a 3D reconstruction system for city modeling from omnidirectional video sequences using structure from motion together with stereo constraints. We concentrate on two issues. First, we show how the tracking and reconstruction paradigm were adapted to use omnidirectional images taken by lenses with 180 degrees field of view. This concerns mainly camera calibration transforming the pixel locations into rays and solving the minimal problem for 3D-to-2D matches using RANSAC. Secondly, we compare the results of the reconstruction using additional stereo constraints to the results when these constraints are not used and show that they are needed to make the reconstruction stable. Performance of the system is demonstrated on a sequence of 870 images acquired while driving in a city.

1 INTRODUCTION

3D scene modeling from images is an important problem of computer vision and photogrammetry. Albeit large progress made recently in understanding the key problems of geometry (Hartley and Zisserman, 2003), optimization (Triggs et al., 1999), and related algebra (Nistér, 2004a), the design of systems working on a large number of images is still an interesting and open engineering problem. For some applications, working or partially working solutions have already been introduced. For instance, Boujou (2d3 Boujou, 2001) system is capable of reconstructing the camera trajectory from a sequence containing several thousands of images when image sequences are acquired in a limited space and the camera does not make sharp turns.

In this paper we deal with the problem of the automatic reconstruction and modeling of real cities from dense image sequences acquired by a pair of cameras mounted on a survey vehicle. This application calls for the ability to process a very large number of images which span extended spaces and are acquired along trajectories containing large camera rotations. The processing must be done in, or at least close to, real-time.

Previously, the city reconstruction has been addressed using aerial images (Grün, 1997; Brenner and Haala, 1998; Haala et al., 1998; Maas, 2001; Vestri and Devernay, 2001; Vosselman and Dijkman, 2001) 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.

Alternatively, survey vehicles equipped with laser scanners and cameras were used to gather 3D depths and textures at ground level (Früh et al., 2005; Früh and Zakhor, 2001; Stamos and Allen, 2000; Sun et al., 2002). These systems gave very nice and accurate 3D models in some situations but they were complicated and expensive. A city modeling system (Akbarzadeh et al., 2006) from dense image sequences acquired simultaneously by 8 perspective cameras has been also designed. The system records images and processes them later off-line.

Recently, a framework for city modeling from image sequences working in real-time has been developed in (Cornelis et al., 2006a). It uses structure from motion (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

Havlena M., Pajdla T. and Cornelis K. (2008).
STRUCTURE FROM OMNIDIRECTIONAL STEREO RIG MOTION FOR CITY MODELING.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 407-414
DOI: 10.5220/0001081104070414

the scene and all the building facades are ruled surfaces parallel to it, and real-time texture mapping to generate visually correct models from a very large number of images.

The system gives good results but two major problems have been reported. First, cars parked along streets were not correctly reconstructed since they did not lie in the ruled surfaces representing either the ground or the buildings on the side. This problem has been solved by recognizing car locations and replacing them by corresponding computer generated models (Cornelis et al., 2006b). Secondly, 3D reconstruction could not survive sharp camera turns when a large part of the scene moved away from the limited view field of cameras. We propose to solve the second problem by using "omnidirectional" cameras with larger field of view.

Omnidirectional cameras have been used on cars and mobile platforms (Benosman and Kang, 2000) mainly to estimate ego-motion of the vehicles or for simultaneous localization and motion planning (Goedemé et al., 2007; Ehlgen and Pajdla, 2007). These works used catadioptric cameras with views optimized to see the complete surroundings of their vehicles in a limited resolution. They do not provide images of photographic quality needed for city modeling. We therefore use 180° fish-eye lenses which are compact and provide better image quality (Mičušík and Pajdla, 2006).

Omnidirectional vision was previously used also for city modeling to capture images with very large resolution. Panoramic mosaicing was preferred to using a fish-eye lens for recovering relative camera poses very accurately from a small number of images (Antone and Teller, 2000; Antone and Teller, 2001) and to generate high resolution and high dynamic range images (Teller et al., 2003) from georeferenced positions. This approach provides very detailed but large images and is not suitable for real-time processing. We use two compact 4 Mpixel omnidirectional cameras. Images of such size can be processed in real-time. On the other hand, our images are extremely radially distorted and a special projection model is needed to process them.

In this paper we present an extension of the framework (Cornelis et al., 2006a) for an omnidirectional stereo rig. We focus on presenting the extensions to the camera tracking and structure from motion and demonstrating the functionality of the modified SfM framework in experiments. We also show that using two omnidirectional cameras bound into a stereo rig prevents the undesirable drift in the estimation of the camera poses. Extensions to facade generation will be reported elsewhere.

2 THE SFM FRAMEWORK FOR AN OMNIDIRECTIONAL STEREO RIG

Figure 1: Omnidirectional stereo rig with Kyocera Finecam M410R cameras and Nikon FC-E9 fish-eye lens convertors.

Omnidirectional cameras differ from the perspective ones primarily in their image projection. This difference influences (i) camera calibration, (ii) feature extraction for image matching, and (iii) structure from motion computation. We shall next describe the extension of the SfM framework (Cornelis et al., 2006a) to be able to use the omnidirectional stereo rig of cameras with 180° field of view lens convertors shown in Figure 1.

2.1 Omnidirectional Camera Calibration

We calibrate omnidirectional cameras off-line using the technique (Bakstein and Pajdla, 2002) and Mičušík's two-parameter model (Mičušík and Pajdla, 2006), which links the radius of the image point r to the angle θ of its corresponding rays w.r.t. the optical axis, see Figure 2, as

$$\theta = \frac{ar}{1+br^2}. \quad (1)$$

Projecting via this model provides good results even when a low quality fish-eye lens is used because the additional parameter b can compensate for improper lens manufacturing.

All operations in the SfM framework that compute a projection of a world 3D point into the image or a ray casted through a pixel are using this lens model. The mapping from pixel positions to the corresponding rays is pre-computed and stored in a table to save time in actual computations.

2.2 Features

Images are matched by detecting, describing and tracking corner-like image features (Cornelis et al.,

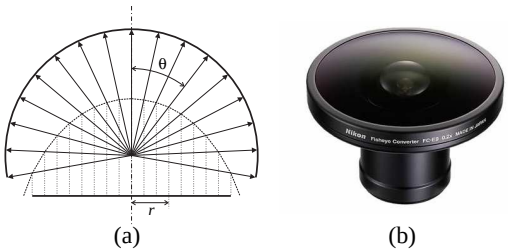

Figure 2: Diagram (a) shows the equi-angular projection of Nikon FC-E9 lens convertor. Angle θ measured between the casted ray and the optical axis determines the radius r of a circle in the image circular view field where the pixel representing the value of the projected 3D point will lie. The Nikon FC-E9 lens convertor can be seen in (b).

2006a). The green image channel is divided into sections of 8×8 pixels and at most one salient feature per section is used to limit the amount of computation.

The feature saliency F is computed from a square region of pixels as

$$F = |(M_{UL} + M_{WR}) - (M_{UR} + M_{WL})|, \qquad (2)$$

where M_{UL}, M_{UR}, M_{WL}, and M_{WR} are average pixel values inside the upper-left, upper-right, lower-left, and lower-right quadrants.

These features were designed to detect corners of buildings and their windows and they work reliably for corners where horizontal and vertical lines meet. The detection becomes worse for rotated corners. Furthermore, objects captured in omnidirectional images are radially distorted as they come closer to the border of the circular view field. The feature saliency can therefore differ dramatically if computed on an object located in the center of the view field or on the same object when it appears close to the border. This can be remedied by a local image rectification (Mauthner et al., 2006) but we observed that the difference is negligible when matching consecutive images of our dense image sequences. Figure 3 shows an input image and the detected feature points.

2.3 Initialization by 2D Tracking

The camera tracking and structure from motion computation has to be initialized by computing initial 3D structure. Internal parameters of the camera calibrations (held constant for the whole sequence) and a few initial camera poses are needed. Feature points are detected and tracked in 2D over several consecutive images and then triangulated into world 3D points using known camera poses.

Tracking in 2D is done by constructing tentative matches from pairs of feature points in consecutive

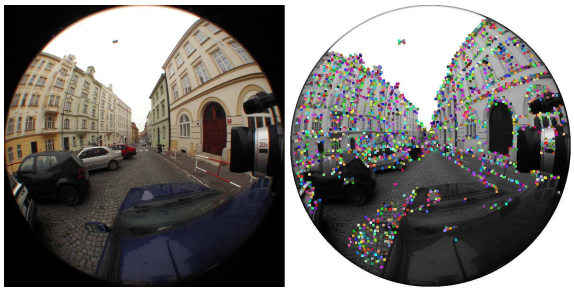

Figure 3: Left: Input image. Right: Detected feature points marked with coloured squares around them. Black area around the circular view field is excluded from feature detection.

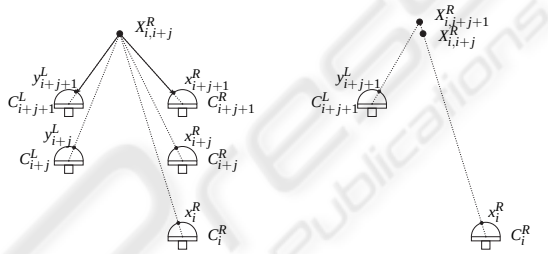

Figure 4: Left: 3D point $X^R_{i,i+j}$ triangulated from x^R_i and x^R_{i+j} or x^R_i and y^L_{i+j} and projected into new images acquired by cameras C^L_{i+j+1} and C^R_{i+j+1}. Positions of the most similar feature points are denoted by y^L_{i+j+1} and x^R_{i+j+1}. Right: The refinement of the 3D point $X^R_{i,i+j}$ into $X^R_{i,i+j+1}$ using triangulation from x^R_i and y^L_{i+j+1}.

images, which have small differences in positions as well as in their saliencies. Images used for initialization thus should come from a slow camera motion without sharp turns. Next, pixel regions of the tentative feature point pairs are correlated and only the sufficiently and mutually most similar tentative matches are joined to construct tracks. Only those tracks that are tracked during all frames of the initialization are used to triangulate cameras and compute the initial 3D structure. It is important to adjust the length of the initial sequence to retain a sufficient number of tracks corresponding to a sufficiently large camera motion. The initialization is done independently for the left and right camera, so two sets of world 3D points are computed.

2.4 Expansion of the Euclidean Reconstruction

Once the Euclidean reconstruction is initialized, the next image pair in the stereo sequence is taken and the reconstruction is expanded using it. The expansion consists of several steps described below in detail.

First, the camera poses of the new stereo pair

must be established. 3D points reconstructed in previous frames are projected into the new images using the last established camera poses. The feature points that could prolong the tracks connected with the projected 3D points are found in small neighbourhoods of the projections using the same tests as during the initialization. As can be seen in Figure 4, every reconstructed 3D point, e.g. $X^R_{i,i+j}$ triangulated from feature point positions x^R_i and x^R_{i+j} or x^R_i and y^L_{i+j} (depending on whether or not it has been re-triangulated already), is projected into the right and the left images as $\pi^R(X^R_{i,i+j})$ and $\pi^L(X^R_{i,i+j})$. To prolong tracks, we establish tentative 3D-to-2D matches $(X^R_{i,i+j}, x^R_{i+j+1}, y^L_{i+j+1})$ between the 3D point $X^R_{i,i+j}$, the feature point x^R_{i+j+1} found in the neighbourhood of $\pi^R(X^R_{i,i+j})$ as the feature point whose saliency is most similar to the saliency of x^R_{i+j}, and the feature point y^L_{i+j+1} found in the neighbourhood of $\pi^L(X^R_{i,i+j})$ as the feature point whose saliency is most similar to the saliency of x^R_{i+j+1}. The tentative 3D-to-2D matches are used as the input to RANSAC (Fischler and Bolles, 1981) robust estimation technique which estimates the camera poses and simultaneously rejects wrong tentative matches.

Left camera pose can be computed from a minimal sample of three 3D-to-2D correspondences by Nister's algorithm (Nistér, 2004b) and right camera pose is then obtained using the rigid left-right transformation computed from the known camera poses during the initialization. The main advantage of Nister's algorithm, originally designed for non-central cameras, lies in the fact that the rays do not need to be concurrent and thus rays going through both the left and the right cameras can be combined together in one sample. The algorithm (Nistér, 2004a) leads to solving an 8-degree polynomial using Sturm sequences and bisection with a fixed number of iterations and gives accurate results in constant time.

The RANSAC stopping condition ensures stopping dependent on the probability of finding a better sample. As we are using samples of size 3, RANSAC usually needs only tens of samples to meet the stopping condition. However, not to exceed the maximal processing time available, a threshold for the maximal number of samples has to be used. To save even more time, the test for inliers is performed gradually on partitions of the matches and the verification is terminated as soon as it is clear that the new hypothesis cannot be better than the best hypothesis known at the time. A similar idea is extended into a two-step evaluation procedure in (Chum and Matas, 2002) and further modified specially for on-line motion estimation in (Nistér, 2003). A match $(X^R_{i,i+j}, x^R_{i+j+1}, y^L_{i+j+1})$ is an inlier if and only if both matches $(X^R_{i,i+j}, x^R_{i+j+1})$ and $(X^R_{i,i+j}, y^L_{i+j+1})$ are inliers.

Two runs of the Levenberg-Marquardt non-linear optimization are used to refine the camera poses using the computed set of inliers. The first refinement uses reprojection error as the cost function and finds the best solution according to the computed set. As this set can be computed incorrectly and can contain true outliers which might have a big influence on the optimization, a fixed cost value is used when the reprojection error is bigger than a threshold during the second refinement to suppress this influence. Again, reprojection errors in both the left and the right images are measured.

The tracks of the resulting inliers are prolonged and 3D points connected with these tracks are refined by re-triangulation. The stereo rig rigidity constraint is enforced again when feature points x^R_i and y^L_{i+j+1} are used to triangulate the 3D point $X^R_{i,i+j+1}$. The rest of the tracks, i.e. the tracks of the outliers and the tracks that did not have a corresponding match, are ended. If the same feature point is detected later again, a new track with a new connected 3D point is created with no binding to the old one.

There are also tracks that do not have a 3D point connected with them because either they are too short or the angle between the two rays used for triangulation is not yet large enough. These tracks are prolonged using the following geometrical constraints derived from the established camera poses to restrict the set of possible locations of the feature points. First, a homography through a virtual plane in a fixed distance in front of the camera is used to get an estimate of the position of the feature point and a circular neighbourhood around this location is searched. This distance should be set to the expected average distance of the feature points. An additional condition is the proximity to the matching epipolar line. When having omnidirectional cameras, the residual distance is computed as the distance between the feature point position and the perpendicular projection of the ray going through the position of the feature point into the matching epipolar plane, projected to the image.

2.5 Bundle Adjustment

The data computed from the image sequences during the expansion are divided into blocks, each of them holding information from 60 images. Unlike the on-line local bundle adjustment routine described in (Mouragnon et al., 2006), our routine processes the already finished data blocks with no back coupling to the expansion. First, the positions of 3D points are refined with fixed camera poses and then the camera

poses are refined with fixed positions of 3D points. Left and right cameras are rigidly bound using the left-right transformation and 3D point reprojection errors in both the left and the right images are summed together in the cost function. The whole routine runs twice and a fixed cost value is used when the reprojection error is bigger than a threshold during the second run to suppress the influence of outliers.

The main reason of running the bundle adjustment routine is to smooth the camera trajectories and to remove noise from 3D point clouds as only the tracks of feature points visible in 4 frames or more are used for refinement and 3D points reconstructed from short tracks are thrown away because these tracks are considered to be less reliable.

3 EXPERIMENTS

Next we shall demonstrate the structure from motion with an omnidirectional camera stereo rig. We shall first describe the stereo rig and then compare the results of motion computation with and without the modifications that enforce the stereo rig rigidity constraint described in Section 2.

3.1 Omnidirectional Stereo Rig

The important parameters of a camera rig are: view angle, resolution, image quality, frame rate, exposure synchronization, size and weight, and the length of the base line. We have constructed a two-camera rig. Each camera of the rig is a combination of Nikon FC-E9 mounted via a mechanical adaptor onto a Kyocera Finecam M410R digital camera, see Figure 1.

Nikon FC-E9 is a megapixel omnidirectional add-on convertor with 180° view angle. It is designed to be mounted on top of lenses of standard Nikon digital cameras. The lens is larger and heavier than similar FC-E8 Nikon lens but it is designed for imagers with higher resolution than FC-E8 and provides images of photographic quality. Kyocera Finecam M410R delivers 2272×1704 images at 3 frames per second. Since the FC-E9 lens is originally designed for a different optical system, we used a custom made mechanical adaptor to fit it on top of the Kyocera lens. The resulting combination yielded a circular view of diameter 1600 pixels in the image.

Since the FC-E9 lens is close to equiangular projection (Bakstein and Pajdla, 2002), we obtain angular resolution $0.11 = 180/1600$ degrees per pixel in the radial direction of the image. The tangential resolution depends on the distance from the view center. It grows from 0.11 degrees per pixel in the center to 0.036 degrees per pixel at the periphery. For comparison, consider that a 1024×768 camera with a common angle of view 40° yields almost uniform resolution $0.039 = 40/1024$ degrees per pixel. Kyocera cameras do not have external synchronization but we were able to connect an external signal to start the acquisition at the same moment. Figure 5 shows four cameras mounted on a survey vehicle. The two cameras with large fish-eye lenses form our stereo rig with 0.95 m base line.

Figure 5: Kyocera Finecam M410R cameras with Nikon FC-E9 fish-eye lens convertors and two conventional perspective cameras mounted on a survey vehicle. Perspective cameras were not used in our experiments.

3.2 SfM with the Stereo Rig Rigidity Constraint

There are several ways to get the camera poses needed for the initialization. If the cameras are mounted on a vehicle riding at a constant known velocity with no changes in the direction of the movement during one second, starting camera poses for the left camera can be computed easily. If the relative camera pose of the right camera w.r.t. the left camera is known, starting camera poses for the right camera can be obtained by a simple transformation.

Another approach does not rely on a known stereo rig calibration but computes the starting camera poses directly. An extension of a WBS structure from motion (Matas et al., 2004) to omnidirectional images can be used to get epipolar geometries between the first left and first right, first left and e.g. sixth left, and first right and sixth left cameras. These geometries can be then combined together to get movement estimation fulfilling the stereo rig rigidity constraint.

Both approaches were tested and work well. The main advantage of the first approach lies in the fact that one needs no additional method to start the reconstruction. On the other hand, the second approach can be used even when the stereo rig calibration and/or the movement of the car are not known.

Figure 6 shows a city segment with several blocks of houses used for our experiments. We were driving our survey vehicle equipped with the camera rig

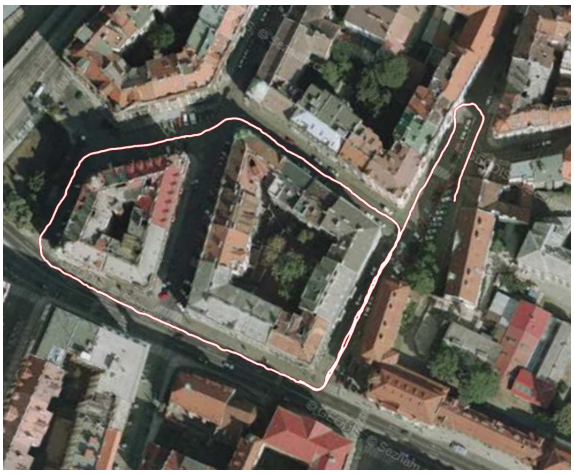

Figure 6: An aerial view of the city segment used for the acquisition of our test sequence, the designed car trajectory is drawn with a white line. The trajectory contains several sharp turns and a round-trip around a block of houses.

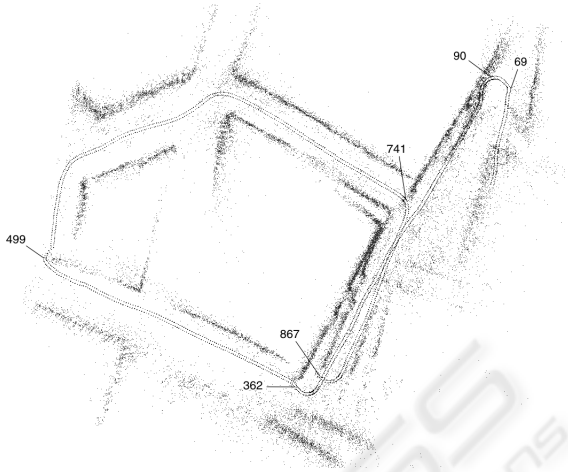

Figure 7: The resulting 3D model from the top view. Camera positions are represented by bigger dots, smaller dots represent the reconstructed world 3D points. The loop is not closed, mostly because of the errors arising in the sharp turns where the number of active tracks drops dramatically. Note that the reconstruction nearly failed in the sharp turn at frame number 499.

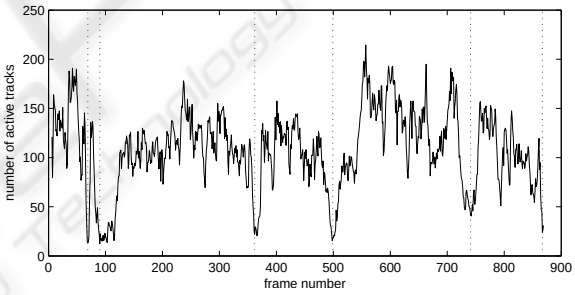

Figure 8: Variation of the number of active tracks for different frames in the sequence. Note that the number of active tracks drops dramatically in frames corresponding to sharp turns.

slowly following the path drawn in the map. The designed trajectory contains sharp turns to test the performance under difficult conditions and a closed loop which allows us to measure the accuracy of the reconstruction. The data were acquired under normal traffic conditions with cars and pedestrians moving in the streets.

Our test sequence was 870 frames long and the first and the sixth image were used to initialize the SfM with more than 200 correct tracks for each camera reconstructed into world 3D points. The top view of the resulting reconstructed 3D model can be seen in Figure 7. Straight street segments are quite easy, the support of the RANSAC winner is usually more than 60% and only few tens of runs of the RANSAC loop are needed to find it. Segments with sharp turns are much more difficult, the support of the RANSAC winner and also the number of active tracks drop dramatically, see Figure 8. We hypothesize that this is caused mostly by inaccurate camera and/or stereo rig calibration because the world 3D points come closer to cameras and start rotating, which causes the errors in the estimations of their depths to become much more important than when these 3D points are distant and the movement is rotation-free.

The shape of the reconstructed trajectory corresponds well to the actual one, we observe only small problems at the beginnings of the turns when the movement is still estimated as being forward although the car is just starting to turn. This is probably caused by finding a large number of feature points on the corner building and a lack of feature points in the other parts of the scene. These “corner building” feature points form a large set of inliers to a model which does not describe the whole scene well.

The error accumulated along the 420 meters long loop is less than 4.5 meters.

3.3 SfM Without the Stereo Rig Rigidity Constraint

During the adaptation of the original SfM into an omnidirectional one, we first adapted the geometry and RANSAC without enforcing the stereo rig rigidity constraint (Havlena et al., 2007) in the reconstruction. Stereo information was used only in the RANSAC loop where the left camera pose was estimated from 3D-to-2D matches from both cameras and the right camera pose was computed using the stereo rig calibration.

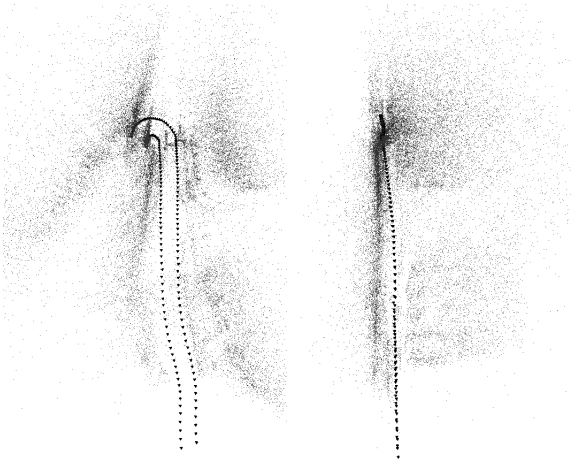

Figure 9: The resulting 3D model from the top (left) and the side views (right). As the stereo rig rigidity constraint is not enforced, scale of the reconstruction is being lost gradually so the cameras are approaching the ground plane although they were actually moving parallel to it. The reconstruction fails in the first sharp turn.

The SfM worked fine when using additional GPS/INS data but failed when these data were not used. The resulting model reconstructed for the same test sequence without enforcing the rigidity constraint can be seen in Figure 9. The number of active tracks drops under 10 in the first sharp turn because the positions of world 3D points were not estimated well as the scale of the reconstruction was gradually lost.

A comparison with the original framework using perspective cameras was not performed but we hypothesize that the result would be even worse not only because of the missing stereo rig rigidity constraint but also because of the lack of feature points caused by a very small field of view.

3.4 Performance

The original SfM framework is able to work in real-time and it would be exciting to achieve the same speed even with fish-eye cameras. Until now, we were interested more in functionality than in performance and the actual speed of our C++ implementation on a standard 2GHz Intel Pentium 4 computer is about 1.3 frames per second. This is primarily caused by the size of the input images which is 800×800 compared to 360×288 used with perspective cameras. Working with smaller images makes it more difficult to detect and to correctly describe enough feature points and making the images much smaller will be possible only if an extension to feature extraction would be proposed and implemented. This extension would describe the features on a locally unwarped image. As this unwarping would not be quick enough using the CPU, GPU programming techniques should be used via OpenGL.

On the other hand, it showed out that 3 frames per second provided by our omnidirectional cameras are enough for the reconstruction from a moving vehicle because feature points do not get lost from the image as quickly as when perspective cameras are used. That is why it is not necessary to achieve 25 frames per second computational performance, 3 frames per second are enough for real-time processing.

4 CONCLUSIONS

We succeeded in adapting the structure from motion part of a city modeling framework to using an omnidirectional stereo rig. The major changes of the framework originally working with one perspective camera involve significant changes in geometry, as rays represented by unit vectors have to be used instead of image pixels, and enforcing the stereo constraints. We have also shown how using two cameras bound into a stereo rig improves the stability of the reconstruction and helps to keep its overall scale.

Making the reconstruction more accurate is our main goal for the near future. We believe that better calibration of the cameras and of the stereo rig done directly from the images together with merging the tracks of the same feature points accidentally lost in one or more of the frames due to occlusions or fast camera movements should increase the number of inliers in difficult sharp turns and would therefore help us to get rid of the biggest sources of inaccuracies.

The rest of the future work lies in adapting the other parts of the city modeling framework to using omnidirectional cameras – starting with the facade reconstruction, through topological map generation, and texture generation, which could benefit from using omnidirectional vision.

ACKNOWLEDGEMENTS

This work is supported by the European IST Programme Project FP6-0027787 DIRAC. The author was also supported by the Czech Science Foundation under project 201/07/1136 and by the Grant Agency of the Czech Technical University under project CTU0705913.

This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

We would like to acknowledge Pavel Krsek for the construction of the external trigger and Hynek Bakstein for mounting the cameras to the survey vehicle.

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., Stewéius, H., Yang, R., Welch, G., Towles, H., Nistér, D., and Pollefeys, M. (2006). Towards urban 3d reconstruction from video. In *3DPVT*. Invited paper.

Antone, M. and Teller, S. (2000). Automatic recovery of relative camera rotations for urban scenes. In *CVPR 2000*, pages II:282–289.

Antone, M. and Teller, S. (2001). Scalable, absolute position recovery for omni-directional image networks. In *CVPR 2001*, pages I:398–405.

Bakstein, H. and Pajdla, T. (2002). Panoramic mosaicing with a 180° field of view lens. In *Proc. IEEE Workshop on Omnidirectional Vision*, pages 60–67.

Benosman, R. and Kang, S. (2000). *Panoramic Vision*. Springer-Verlag.

Brenner, C. and Haala, N. (1998). Fast production of virtual reality city models. *IAPRS*, 32(4):77–84.

Chum, O. and Matas, J. (2002). Randomized ransac with $t_{d,d}$ test. In *BMVC 2002*, pages 448–457.

Cornelis, N., Cornelis, K., and Van Gool, L. (2006a). Fast compact city modeling for navigation previsualization. In *CVPR 2006*, pages II:1339–1344.

Cornelis, N., Leibe, B., Cornelis, K., and Van Gool, L. (2006b). 3d city modeling using cognitive loops. In *3DPVT 2006*, pages 9–16.

Ehlgen, T. and Pajdla, T. (2007). Maneuvering aid for large vehicle using omnidirectional cameras. In *WACV 2007*, page 17.

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

Früh, C., Jain, S., and Zakhor, A. (2005). Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. *IJCV*, 61(2):159–184.

Früh, C. and Zakhor, A. (2001). 3d model generation for cities using aerial photographs and ground level laser scans. In *CVPR 2001*, pages II:31–38.

Goedemé, T., Nuttin, M., Tuytelaars, T., and Van Gool, L. (2007). Omnidirectional vision based topological navigation. *IJCV*, 74(3):219–236.

Grün, A. (1997). Automation in building reconstruction. In Fritsch, D. and Hobbie, D., editors, *Photogrammetric Week'97*, pages 175–186, Stuttgart.

Haala, N., Brenner, C., and Stätter, C. (1998). An integrated system for urban model generation. In *ISPRS Congress Comm. II*, pages 96–103.

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

Havlena, M., Cornelis, K., and Pajdla, T. (2007). Towards city modeling from omnidirectional video. In Grabner, M. and Grabner, H., editors, *CVWW 2007*, pages 123–130, St. Lambrecht.

Maas, H. (2001). The suitability for airborne laser scanner data for automatic 3d object reconstruction. In *Ascona01*, pages 291–296.

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

Mauthner, T., Fraundorfer, F., and Bischof, H. (2006). Region matching for omnidirectional images using virtual camera planes. In Chum, O. and Franc, V., editors, *CVWW 2006*, pages 93–98, Telč.

Mičušík, B. and Pajdla, T. (2006). Structure from motion with wide circular field of view cameras. *IEEE Trans. PAMI*, 28(7):1135–1149.

Mouragnon, E., Dekeyser, F., Sayd, P., Lhuillier, M., and Dhome, M. (2006). Real time localization and 3d reconstruction. In *CVPR 2006*, pages I:363–370.

Nistér, D. (2003). Preemptive ransac for live structure and motion estimation. In *ICCV 2003*, pages 199–206.

Nistér, D. (2004a). An efficient solution to the five-point relative pose problem. *IEEE Trans. PAMI*, 26(6):756–770.

Nistér, D. (2004b). A minimal solution to the generalized 3-point pose problem. In *CVPR 2004*, pages I:560–567.

Stamos, I. and Allen, P. (2000). 3-d model construction using range and image data. In *CVPR 2000*, pages I:531–536.

Sun, Y., Paik, J., Koschan, A., and Abidi, M. (2002). 3d reconstruction of indoor and outdoor scenes using a mobile range scanner. In *ICPR 2002*, pages III:653–656.

Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., and Master, N. (2003). Calibrated, registered images of an extended urban area. *IJCV*, 53(1):93–107.

Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (1999). Bundle adjustment: A modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372, Corfu.

Vestri, C. and Devernay, F. (2001). Using robust methods for automatic extraction of buildings. In *CVPR 2001*, pages I:133–138.

Vosselman, G. and Dijkman, S. (2001). Reconstruction of 3d building models from laser altimetry data. *IAPRS*, 34(3):22–24.