Detection and Identification of Rare Audiovisual Cues

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than things you do expect) Plautus (ca 200(B.C.)

Project no: 027787

DIRAC

Detection and Identification of Rare Audio-visual Cues

Integrated Project
IST - Priority 2

DELIVERABLE NO: D6.2
Omnidirectional Camera Tracking

Date of deliverable:30.6.2007
Actual submission date:

Start date of project: 01.01.2006 Duration: 60 months

Organization name of lead contractor for this deliverable: **CTU**

Revision [1]

Project co-funded by the European Commission within the Sixth Framework Program (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

DIRAC
Detection and Identification of Rare Audiovisual Cues

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than
things you do expect) Plautus (ca 200(B.C.)

D6.2 OMNIDIRECTIONAL CAMERA TRACKING

Czech Technical University in Prague (CTU)
Katholieke Universiteit Leuven (KUL)
Eidgenoessische Technische Hochschule Zuerich (ETHZ)

Abstract:
This deliverable presents work on DIRAC demonstrator D6.2 - Omnidirectional Camera Tracking. A web page (http://cmp.felk.cvut.cz/projects/dirac/omni_track) has been set up to complement this report with the demonstrator summary and up to date videos and VRML models of reconstructed camera trajectories. We describe a SW platform developed for calibration of omnidirectional cameras in the DIRAC project and we show how it is integrated into an existing 3D reconstruction module. We also describe an alternative approach to camera motion recovery using wide baseline stereo with omnidirectional images. We discuss integration of omnidirectional images into cognitive loops for detection of objects in the scene. Technical details and experimental results are given in an appendix.

Table of Content

1. Introduction

This deliverable presents technical material for DIRAC demonstrator D6.2—Omnidirectional Camera Tracking. A web page (`http://cmp.felk.cvut.cz/projects/dirac/omni_track`) has been set up to complement this report with the demonstrator summary and up to date videos and VRML models of reconstructed camera trajectories.

The report consists of three parts. The summary of the work on omnidirectional camera tracking and its integration into the KUL scene modeling framework is given in Section 2. Section 3 presents the relationship to a higher-level scene modeling, object and event recognition and cognitive loops. The appendix contains more detailed technical descriptions of omnidirectional camera calibration in Appendix A, image matching and epipolar geometry estimation in Appendix B, and tracking of a pair of omnidirectional cameras and its integration into the KUL scene modeling framework in Appendix C.

2. Omnidirectional Camera Tracking

We follow on work described in deliverable D1.1 "Omnidirectional Sensors and Features for Tracking and 3D reconstruction" which describe prototype omnidirectional sensors and discussed several feature descriptors suitable for processing of omnidirectional images. The sensors were successfully used in acquisition of several datasets for the DIRAC project. Figure 1 depicts contribution of using different feature detectors in images. While MSERs (Matas et. al. 2004) are good in urban environment, APTS (Mikolajczyk et. al. 2002) are more suitable for natural scenes.

Figure 1 Features detected in omnidirectional images. Left, an urban environment results in sufficient number of MSERs. Right, a natural scene requires APTS to obtain enough features.

We have integrated support for omnidiretional input images into two different 3D reconstruction frameworks; the wide-base line stereo tool at CMP (Matas et. al. 2004) and the strusture from motion (SfM) framework at KUL (Cornelis et. al. 2006). The first integration required error measures for determining the quality of the estimated epipolar geometry and the second one required inclusion of calibrated omnidirectional images and changes in interest point detector. We will discuss these modifications in more detail in the next

paragraphs and with technical details in the appendix. The error for quality measure of epipolar geometry has to be defined by angles
instead of pixel distances for omnidirectional images. This is because omnidirectional images have generally non-uniform image resolution. Figure 2 shows different camera motions resulting in varying quality of a standard epipolar geometry computation with image reprojection error, while the angular error gives stable results.

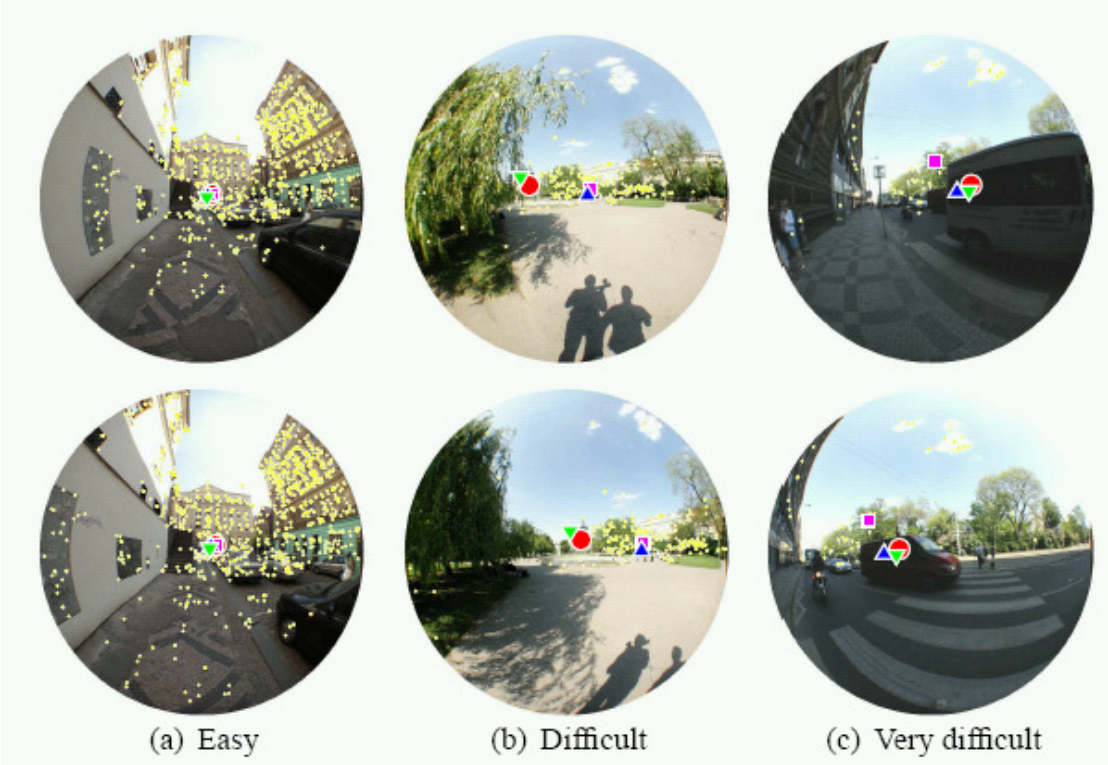


Figure 2 Four examples of camera motions. Red circle, blue up facing triangle, and green down facing triangle represent the true epipole, the epipole computed by maximizing the number of matches, and the epipole computed by soft voting for the position of the epipole, respectively. Small dots show the matches giving blue up facing triangle. (a) The camera translates forward. Many regions have been detected. Finding the camera motion is easy. (b) Camera translates forward and rotates 45◦ to the right. Most of the overlapping field of view has been occluded by a moving vehicle in the left image. Regions have been detected only in a small part of the view field. Finding the motion by reprojection error failed but angular error was successful. (c) The camera translates forward and rotates by 30◦. Many regions detected on bushes do not correspond to stable 3D structures and their descriptors are all very similar. Finding camera motion is more difficult due to the low fraction of correct matches. The Algorithm B.3.3 can find the correct motion.

Calibrated omnidirectional images mean that we can define a light ray for each pixel. In contrary for classical pinhole model, the rays are not defined as a vector between the image center and the respective pixel in the image plane, we have to consider a spherical “retina” and the light rays have to be oriented and rays represented by unit vectors have to be used instead of image pixels, and forcing the stereo constraints. Our experiments proved that using two cameras bound into a stereo rig improves the stability of the reconstruction and helps to keep its overall scale.

Moreover, omnidirectional images have also visible boundary between the (usually circular) part with image data and “black pixels” around this part, see Figure 3, left. The feature detector in the SfM framework had to be modified to ignore matches at this boundary. These are the necessary changes that had to be made for inclusion of support for the omnidirectional images into the KUL SfM framework.

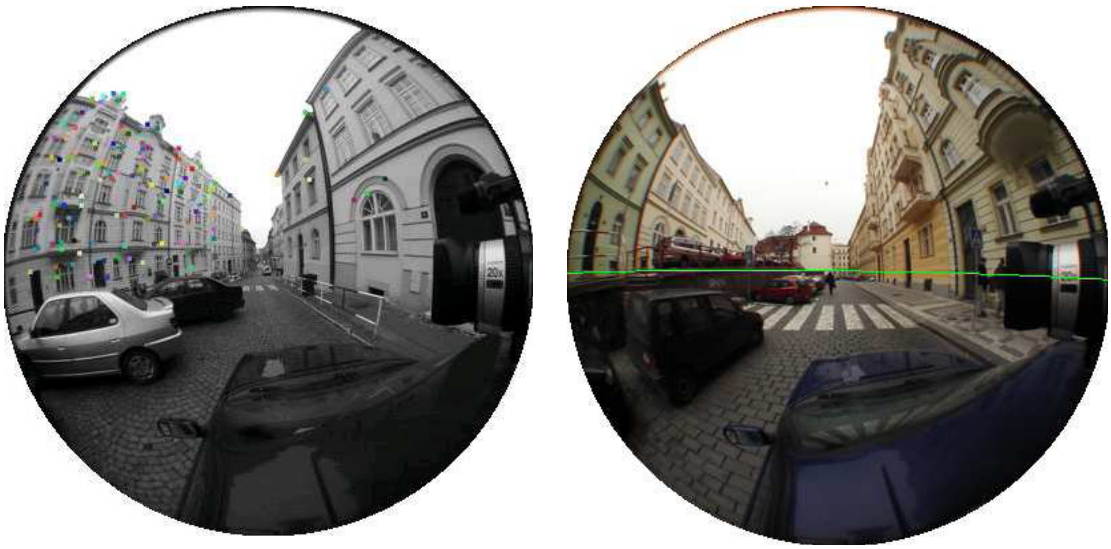


Figure 3 Left, features detected in omnidirectional images using the KUL SfM framework with proper treating of the image boundary. Right, ground plane comcomputed from the camera motion.

3. Relation to Cognitive loops — Deliverable D3.1, Expected Improvements

In deliverable D3.1 "Framework for Bottom-up 3D Reconstruction" a system was described which was able to reconstruct an a-priori unknown environment from a mobile stereo platform. At the time this platform was equipped with two normal perspective cameras. This framework already succeeded in computing the platforms trajectory for general motions and in building up a dense textured reconstruction of the scene.

Among the challenges faced by this first reconstruction system are the handling of fast turns in the platform motion, which impedes feature tracking as features disappear quickly out of the field of view. In addition, the texture of the final dense 3D model can only be extracted from the narrow field-of-view images offered by the perspective cameras. This results in typically good texture quality for the lower levels of the scene which are in view the longest, whereas higher parts, such as roof tops, etc disappear quickly from the viewing frustum of the cameras. This leads to poor texture quality for the aforementioned parts of the scene.

Omni-directional camera tracking will remedy both problems. The increase in viewing volume as seen by the cameras will allow feature tracks to be tracked much longer, even in fast turns of the mobile platform. In addition, the near 180 degree field-of-view will allow to get a more global view of the objects. Enabling to get textures for these objects from much closer viewpoints.

Now that omni-directional camera tracking is available, the new camera model requires changes in both the 3D dense reconstruction module and the object recognition module, which are coupled together into a cognitive loop in order to achieve robust dynamic scene analysis. This work was presented in "Bastian Leibe, Nico Cornelis, Kurt Cornelis, and Luc Van Gool, *Dynamic 3D Scene Analysis from a Moving Vehicle" at CVPR 2007.* Figure 4 demonstrates the cognitive loop setup.

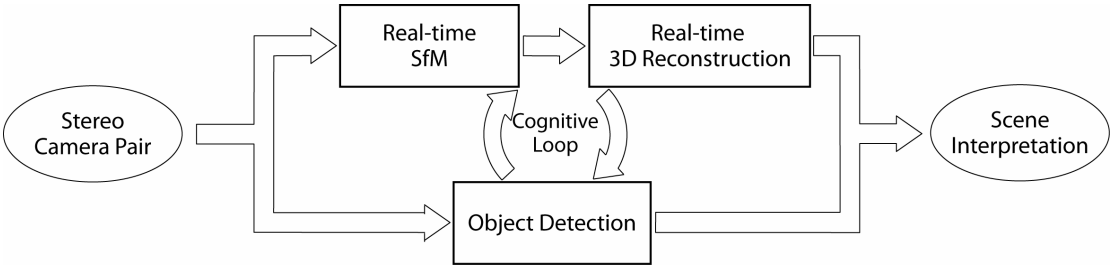


Figure 4: *Overview of our combined system integrating recognition and 3D reconstruction.*

The power of cognitive loops, as explained in more detail the aforementioned publication, will become even more apparent when we will start to use omni-directional images and tracking results. The use of omni-directional images makes object recognition a much harder task because of the image distortion caused by the lens optics. They lead to deformations of the objects to be recognized and a quick change in scale of these object in the image as they approach the image borders.

The ground plane, as can be computed from the results from the omni-directional camera tracking, will therefore be used to constrain the search for pedestrians, cars and other actors on the road surface. Using the ground plane and the tracked camera poses, the world sizes of detected objects (when assuming that they are on the ground plane) can be computed and compared to a size prior. This helps in throwing out many false positives which will undoubtedly occur because the object detection thresholds need be lowered to account for the image formations caused by the fish-eye lens. Figure 5 shows several representative examples of Omni images in which the indicated objects will need to be identified despite their deformations.

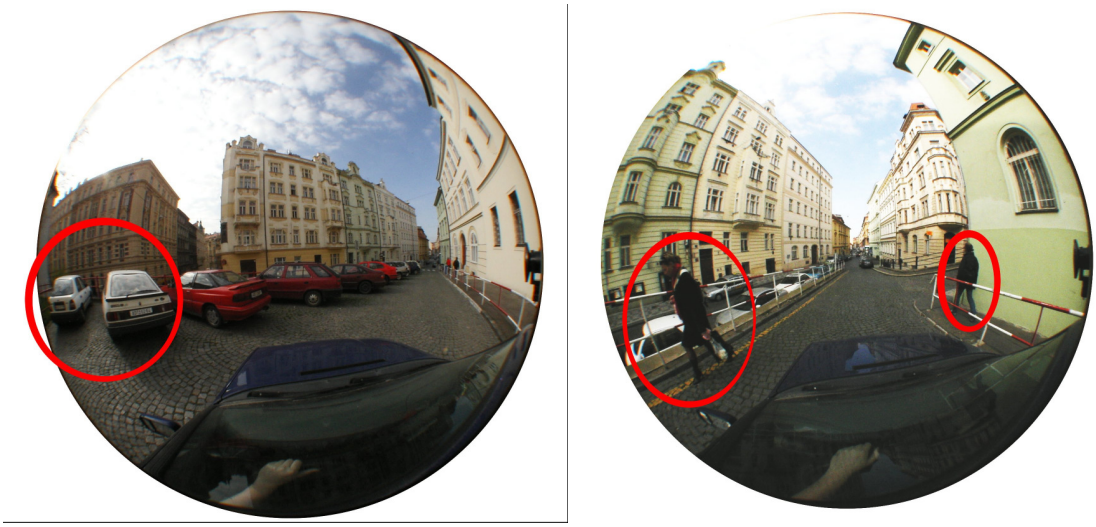


Figure 5: Object recognition is challenged by the image deformations caused by fish-eye optics.

Also the dense reconstruction module as mentioned in D3.1, which is based on fast dense stereo matches of vertical lines (see Figure 6 for a summary of dense stereo matching in D3.1) will need to undergo a drastic change. This is the result of the fact that vertical lines in the 3D world are no longer displayed as straight lines in the image, as was the case for normal perspective images.

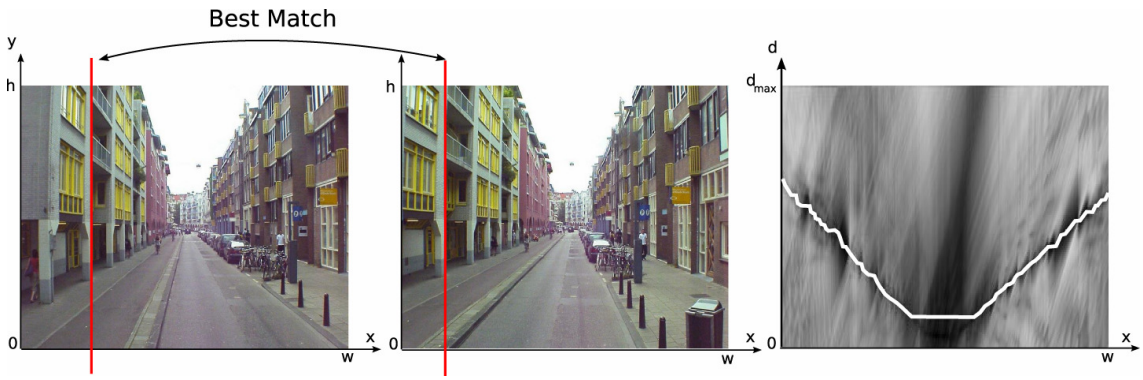


Figure 6: Building up correlation space by calculating the pixel difference when matching each vertical line of image 1 with image 2. Dynamic path programming is used to find the optimal path with lowest cost throughout this correlation space.

4. Conclusion

We have presented work on implementing support for omnidirectional images into two existing 3D reconstruction tools, the SfM framework at KUL and the wide-baseline stereo tool at CMP. We have also discussed inclusion of omnidirectional inputs into the cognitive loop for object detection. Technical details are given in the appendix and a web page (http://cmp.felk.cvut.cz/projects/dirac/omni_track) has been set up to present up-to-date videos and VRML models of results of omnidirectional camera tracking and 3D reconstruction.

5. Reference

Nico Cornelis, Kurt Cornelis, and Luc Van Gool. Fast compact city modeling for navigation pre-visualization. In *CVPR 2006*, volume 2, pages 1339–1344, 2006.

Jiří Matas, Ondřej Chum, Martin Urban, and Tomáš Pajdla. Robust widebaseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, September 2004.

K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the European Conference of Computer Vision (ECCV)*, volume 1, pages 128–142, 2002.

DIRAC deliverable 6.2: Omnidirectional Camera Tracking — technical details

Michal Havlena Akihiko Torii Hynek Bakstein
Tomáš Pajdla

A Omnidirectional Camera Calibration

Omnidirectional camera calibration was implemented as a SW module in MATLAB with, see Figure 1 and can be distributed among the partners. The calibration can be both off-line using a known calibration target or autocalibration, using point correspondences detected in the images. Off-line calibration uses minimization of the camera model parameters a and b for the model

$$r = \frac{a - \sqrt{a^2 - 4\theta^2 b}}{2b\theta} \quad , \tag{1}$$

where r is the distance from the principal point and θ is the angle between the optical axis and a light ray of the corresponding image point. The model has an iverse form

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

To relate the camera center and the origin of the coordinate system of the calibration object, a rigid body transformation has to be included in the minimization adding additional 6 parameters.

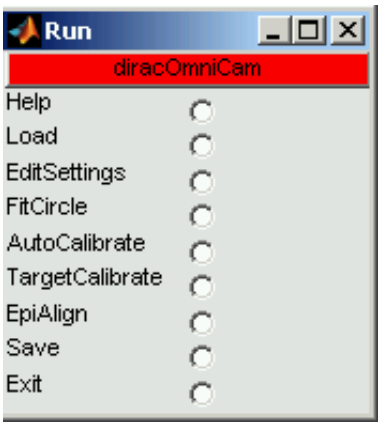


Figure 1: diracOmniCam - an SW platform for sensor calibration.

The autocalibration uses pairs of images with correspondences established with some (wide baseline) matching tool to compute epipolar geometry relating the two images. The epipolar constraint formulation for a fisheye lens leads to a Quadratic eigenvalue problem (QEP) [10]:

$$(D_1 + aD_2 + a^2 D_3)\mathbf{l} = 0 \ ,$$

where D_i is composed from the image coordinates of the points and a is the model parameter. The vector l contains the elements of the essential matrix and the parameter b. QEP problem can be solved linearly using 15 point correspondences and can be included in a RANSAC loop. This procedure was implemented into the software platform under the option 'AutoCalibrate'.

Both off-line and auto calibration techniques can make a benefit from using a reliable estimate of the principal point and the maximal distance (radius) of an image point from the principal point, called r_{max}. A simple procedure can be used to robustly estimate this radius provided with an image of a white wall. This step is available under option 'FitCirle'. In fact, a simple equidistant approximation $r = a\theta$ of the projection of the omnidirectional camera can be determined only from r_{max} and the maximal viewing angle θ_{max}, provided by the manufacturer of the lens. This one parameter approximation is useful for outlier rejection in a matching procedure but cannot be used for 3D reconstruction and epipolar geometry estimation, since it deviates from the true projection mainly at the border of the field of view.

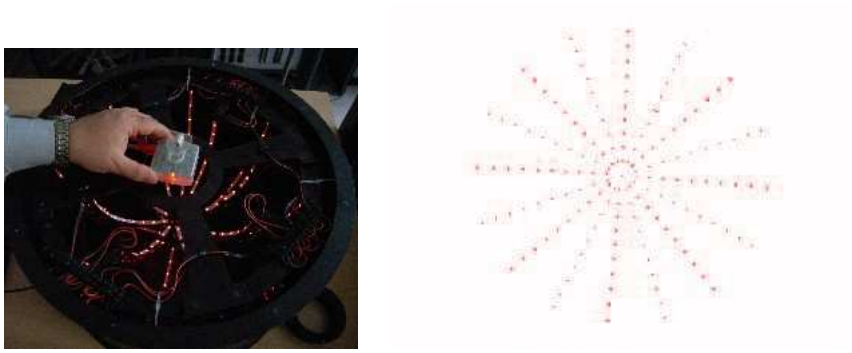
Two approaches to off-line calibration were implemented, the first one uses a calibration object developed in CMP and the second one employs an object developed KUL. The next two subsections describe both calibration objects in detail.

A.1 Calibration object PLANETARIUM

Calibration object 'planetarium' is designed for calibration of lenses with hemispherical field of view. It is composed of LEDs arranged on two half-spheres with different diameters and a common centre. The LEDs are placed on metal ribs and the target is mounted in a black cylinder with detachable lid so that the camera observes only the light from the LEDs and no parasite reflections. The target has additional single LED for establishing automatic correspondences between detected points in images and respective LEDs on the target and the calibration process is fully automatic. The target is available at CMP only. Another object was designed in KUL and can be easily reproduced by other partners.

A.2 Calibration object BOX

Calibration object 'box' was designed in KUL to calibrate omnidirectional lenses. Five square $400 \times 400mm$ big plastic boards covered by $2.5 \times 2.5mm$ checkerboards were mounted together to form five sides of a cube. Some of the checkerboard squares are labeled and top-left corners of the squares with labels are used for



(a) (b)

Figure 2: Calibration with PLANETARIUM. (a) the calibration object and (b) an image captured by the camera.

calibration. Known 3D positions of these 160 points together with their known re-projections into an acquired image give us both the external and internal parameters of a camera with an omnidirectional lens as the result of a non-linear optimization. The shape of the calibration object ensures good coverage of the whole view-field when a camera with an omnidirectional lens is inserted inside the box viewing its rear side.

(a) (b)

Figure 3: Calibration with BOX. (a) the calibration object and (b) an image captured by the camera.

A.3 Results

Figure 4 shows the mapping between the angle θ and the radius in the image r using the model (1) for the calibration object BOX and PLANETARIUM. The resulting parameter values are summarized in Table 1. The right part of Figure 4 shows an enlarged part of the graph of the mapping, where the difference between the parameter sets is the most visible. It should be noted that the b parameter affect

Method	a	b
Autocalibration	1.5550	-0.0612
BOX	1.5689	-0.0461
PLANETARIUM	1.6002	-0.0049

Table 1: Summary of values of parameters a and b for different calibration objects and methods.

the shape of the mapping function. As it was shown in [10], for $b \to 0$, the model function (2) becomes

$$\theta = ar \quad ,$$

which is in fact an equidistant projection. We can observe from Figure 4 that the model parameters extiimated by different methods or objects give slightly different results, with the difference most aparent at the border of the field of view. The difference is caused by coverage of the field of view by the calibration objects and by detected features for autocalibration and also by precision of detection of features or calibration marks at the edge of the field of view.

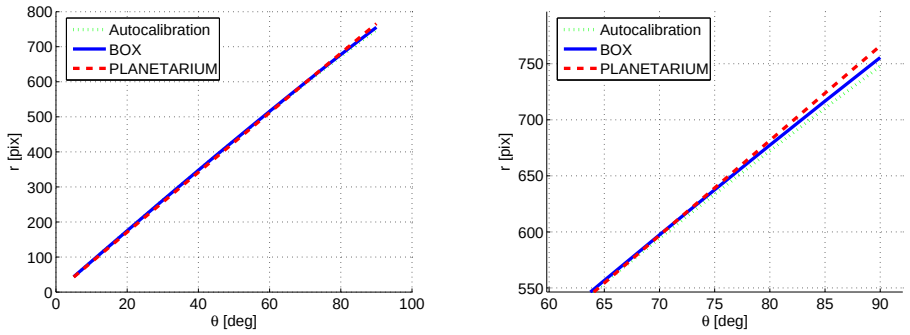


Figure 4: Comparison of calibration using different calibration objects and methods. Right: enlarged part of the left graph.

B Omnidirectional Image Matching and Epipolar Geometry Estimation

In this section, we describe a procedure for computing relative motion of one moving omnidirectional camera. We assume that the camera has been calibrated, e.g. by the technique described above.

(a) Easy (b) Difficult (c) Very difficult

Figure 5: Four examples of camera motions. Red $\circ$, blue $\triangle$, and green $\triangledown$ represent the true epipole, the epipole computed by maximizing the number of matches, and the epipole computed by soft voting for the position of the epipole, respectively. Small dots show the matches giving blue $\triangle$. (a) The camera translates forward. Many regions have been detected. Finding the camera motion is easy. (b) Camera translates forward and rotates 45° to the right. Most of the overlapping field of view has been occluded by a moving vehicle in the left image. Regions have been detected only in a small part of the view field. Finding the motion by reprojection error failed but angular error was successful. (c) The camera translates forward and rotates by 30°. Many regions detected on bushes do not correspond to stable 3D structures and their descriptors are all very similar. Finding camera motion is more difficult due to the low fraction of correct matches. The Algorithm B.3.3 can find the correct motion.

We combine the following four principles and obtain a practical algorithm for epipolar geometry estimation. First, we show that the correct motion is found much sooner if the tentative matches are sampled after ordering them by the similarity of their descriptors. Secondly, we show that the correct camera motion can be better found by soft voting for the direction of the motion that by selecting the motion that is supported by the largest set of matches. Third, we show that the residuals

computed as the angle between a ray and its corresponding epipolar plane work better than the image reprojection error. Finally, we show that it is useful to filter out the epipolar geometries which are not generated by points reconstructed in front of cameras.

B.1 The algorithm

Algorithm B.3.3 presents the pseudocode of the algorithm used to generate results described in this work. Next we describe the key parts of the algorithm in detail.

B.1.1 Detecting tentative matches and computing their descriptors.

MSER [8] and Harris-Affine and Hessian Affine [12] affine covariant feature regions are detected in images. Parameters of the detectors are chosen to limit the number of regions to 1-2 thousands per image. The detected regions are assigned local affine frames (LAF) [15] and transformed into standard positions w.r.t. their LAFs. Discrete Cosine Descriptors [16] are computed for each region in the standard position. Finally, mutual distances of all regions in one image and all regions in the other image are computed as the Euclidean distances of their descriptors and tentative matches are constructed by selecting the mutually closest pairs.

MSER region detector is approximately 100 times faster than the Harris and Hessian Affine region detector but MSERs alone were not able to solve all image pairs in our data. MSERs perform great in urban environment with contrast regions, such as windows, doors and markings. However, they often provide many useless regions on natural scenes because they tend to extract contrast regions which often do not correspond to real 3D structures, such as regions formed by tree branches against the sky or shadows casted by leaves.

B.1.2 Angular error

We compared the image reprojection error with the residuals evaluated as the angle between rays and their corresponding epipolar planes, which we refer as the *angular error* here. When cameras are calibrated, the angular error can safely be used instead of the image reprojection error. To be absolutely correct, every ray should be accompanied by a covariance matrix determining its uncertainty. The matrix depends on (i) image measurement error model and (ii) on the point position in the image. The point position determines how the unit circle around the point maps into the cone around the ray. In this paper we neglected the variability of the covariance matrix across the field of view and assumed it to be a scaled identity.

We have observed that the number of epipolar geometries generated by randomized sampling with correct motion directions, i.e. which differ not more than 8° from the ground truth, was higher when using the angular error, Fig. 8. It is clearly visible when sampling from matches ordered by their similarity (Sec. B.1.4), rows

sim-r with sim-a, but it also holds for sampling by the standard RANSAC, rows rnd-r and rnd-a.

The quality of the angular error also shows after selecting the best motion either by maximizing the number of matches which support it, rows (a,b)S, or by using the soft voting (Sec. B.2), rows (a,b)V in Fig. 8.

Fig. 7 shows that motions estimated using angular error less often provide epipolar geometries generated by a 5-tuple of matches which cannot be reconstructed in front of both cameras.

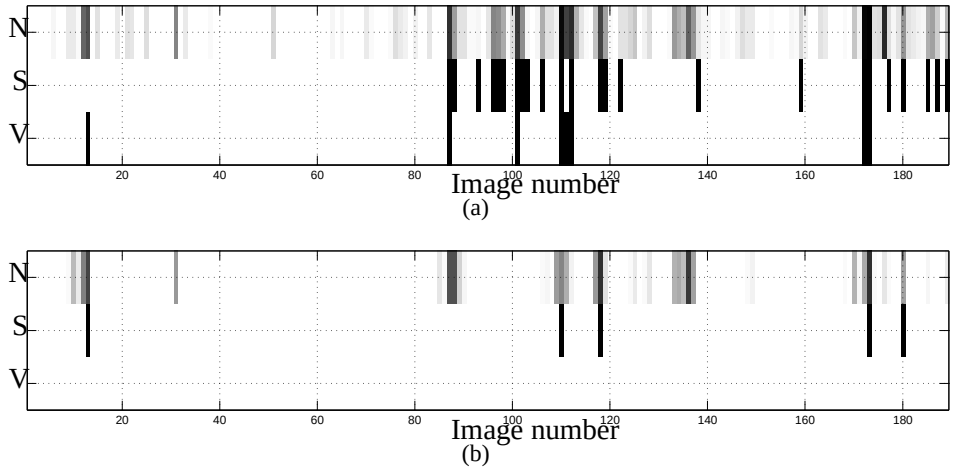


Figure 6: Camera motion estimation by soft voting compared for the (a) residuals evaluated on reprojection into images (reprojection error) vs. (b) residuals evaluated as the angle between rays and their corresponding epipolar planes (angular error). Ordered sampling with 500 samples has been run 50 times providing 50 camera motions. N: The number of the camera motions with the motion direction not more than 8° from the ground truth. Lighter colors represent higher numbers, white $= 50$, black $= 0$. S: The motion direction, which is supported by the largest number of tentative matches in 50 motions, is not (white) or is (black) more than 8° from the true motion direction. V: The estimated motion direction, which is closest to the global maximum in the accumulator, is not (white) or is (black) more than 8° from the true motion direction. Many more motion directions are within the 8° limit when using the angular error compared to using the reprojection error. The best strategy, (b)V, for estimating the camera motion is to use the ordered sampling with the angular error and to select the motion which is closest to the global maximum in the accumulator. limit.

B.1.3 Orientation constraint

An essential matrix can be decomposed into four different camera and point configuration which differ by the orientation of cameras and points [5]. Without enforcing the constraint that all points have to be observed in front of the cameras, some epipolar geometries may be supported by many matches but it need not be possible to reconstruct all points in front of both cameras.

For omnidirectional cameras, the meaning of infrontness is a generalization of the classical infrontness for perspective cameras. With perspective cameras, a point X is in front of the camera when it has a positive z coordinate in the camera coordinate system. For omnidirectional cameras, a point X is in front of the camera if its coordinates can be written as a positive multiple of the direction vector which represents the halfray by which X has been observed.

In general, it is beneficial to use only those matches which generate points in front of cameras. However, this takes time to verify it for all matches. On the other hand, it is easy to verify whether the five points in the minimal sample generating the epipolar geometry can be reconstructed in front of both cameras and to reject such epipolar geometries which do not allow it.

Fig. 7 shows that the number of incorrectly estimated motions decreased when such epipolar geometries were excluded for both selection strategies.

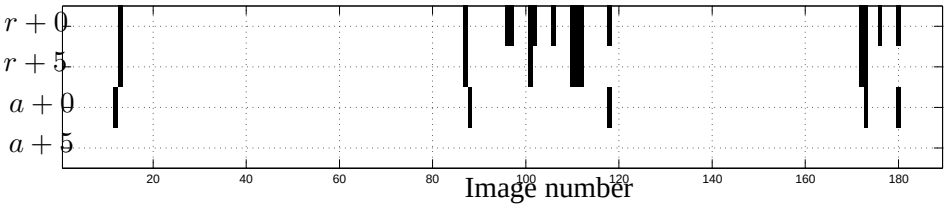


Figure 7: Camera motion directions that are not (white) and are (black) further than 8° from the true direction. The four rows show the four combinations of the two residual errors with using and not using the orientation constraint. r: residuals evaluated on images (reprojection error), a: residuals evaluated as the angle between rays and their corresponding epipolar planes (angular error). 0: all epipolar geometries contribute to the soft voting, 5: only epipolar geometries for which the 5 points of their generating minimal sample get reconstructed in front of both cameras (orientation constraint) contribute to the soft voting. Using the angular error significantly increases the number of correctly estimated directions. Using the orientation constraint further improves the result.

B.1.4 Ordered randomized sampling

We use *ordered sampling* as suggested in [2] to draw samples from tentative matches in ascending order by the distance of their descriptors. However, we keep the orig-

inal RANSAC stopping criterion plus we limit the maximum number of samples to 500. We have observed that pairs which could not be solved by the ordered sampling in 500 samples get almost never solved even after many more samples. Using the stopping criterion from [2] often leads to ending the sampling prematurely since the criterion is designed to stop as soon as a large non-random set of matches is found. Our objective is, however, to find a globally optimal model and not to stop as soon as a local model with large support is met.

We have observed that there are often several alternative models with the property that the right model of the camera motion has a similar or only slightly larger support than other models which are not correct. Algorithm B.3.3 would provide almost identical results even without the RANSAC stopping criterion but the criterion helps to end simple cases sooner than after 500 samples.

Having a calibrated camera, we draw 5-tuples of tentative matches from the list $M = [\mathbf{m}]_1^N$ of tentative matches in ascending order by the distance of their descriptors. From each five-tuple, relative orientation is computed by solving the 5-point minimal relative orientation problem for calibrated cameras [13, 17].

Figure 8 shows that many more correct motions have been sampled in 500 samples of PROSAC using ordered matches than by using the same number of samples on a randomly ordered list of matches.

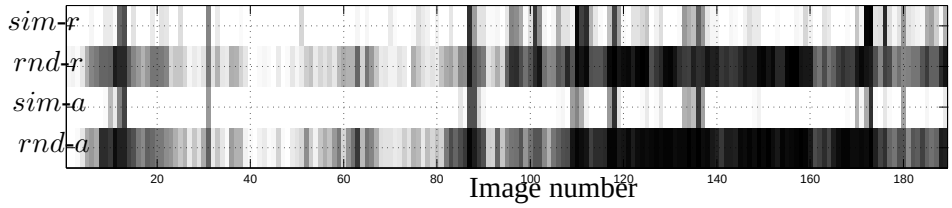


Figure 8: The number of estimated camera motion directions that are not further than 8° from the ground truth. The motion direction were computed by soft voting from the first 500 5-point samples drawn by the ordered sampling. Lighter colors represent higher numbers. The four rows show the four combinations of the two orderings of tentative matches with the two residual errors. sim: tentative matches ordered by feature similarity, rnd: tentative matches ordered randomly, r: residuals evaluated on reprojections to images (reprojection error), a: residuals evaluated as the angle between rays and their corresponding epipolar planes (angular error). Ordering of tentative matches by similarity greatly increases the number of correctly estimated epipoles. The angular error further improves results, often on difficult image pairs.

B.2 Soft voting

In this work, we vote in two-dimensional accumulator for the estimated motion direction. However, unlike in [7], we do not cast votes directly by each sampled

epipolar geometry but by the best epipolar geometries recovered by ordered sampling of PROSAC. This way the votes come only from the geometries that have high support. We can afford to compute more, e.g. 50, epipolar geometries since the ordered sampling is much faster than the standard RANSAC. Altogether, we need to evaluate maximally $500 \times 50 = 25000$ samples to generate 50 soft votes, which is comparable to running a standard 5-point RANSAC for expected contamination by $84\ \%$ of outliers. The relative camera orientation with the motion direction which is closest to the maximum in the voting space is finally selected.

Figure 6 shows the improvement of using soft voting for finding the relative motion when casting 50 soft votes. On several difficult image pairs, such as Fig. 5(c), the motion supported by the largest number of tentative matches gave incorrect motion but soft voting provided a motion close to the ground truth.

B.3 Experiment

B.3.1 Image data

Experimental data consist of 189 image pairs obtained by selecting consecutive images of an image sequence. The distance between two consecutive images was 1-3 meters. Most of the camera motions have rotations up to 15° but large rotations of 45° are also present. Images were acquired by Kyocera Finecam M410R with Nikon FC-E9 fisheye-lens. The field of view is 183°. The image projection is equiangular and was internally calibrated [11]. Images were digitized in resolution 800 pixels$/183^\circ$, i.e. $0.2^\circ/$pixel, which is comparable with 240×180 pixels for more standard 40° field of view. The sequence starts in a narrow street with buildings on both sides, then it continues to a wider street with many driving cars, and finally leads to a park with threes, bushes and walking people.

B.3.2 Ground truth motion

For most image pairs, the "true" camera motions were recovered by running the Algorithm B.3.3 a number of times and checking that (i) the true motion has been repeatedly generated by correctly matched 5-tuples of matches and that (ii) the motion direction pointed to the same object in both images. In a few image pairs, for which we could not get a decisive number of consistent results, the true motion has been generated from a 5-tuple of correct matches selected manually. We estimate the precision of our ground truth motion estimation to be higher than $4\ \%$ of the view field, which corresponds to 8° and 32 image pixels.

B.3.3 Results

Figure 6(b)V shows the quality of the estimated camera motion by Algorithm B.3.3. The algorithm looks for the motion with motion direction closest to the global maximum in the accumulator after casting soft votes from 50 motions. The 50 motions

for soft votes are estimated in 500 samples by the ordered sampling based on residuals evaluated as the angle between rays and their corresponding epipolar planes. All but two motions were estimated with motion directions within $8°$, i.e. $4\ \%$ of the view angle, from the ground truth. The two image pairs, for which the motion has been estimated incorrectly, are very difficult to solve due to small overlap of their fields of view and large occlusions, Fig. 5(d).

B.4 Residual errors

In the following, image points are represented by unit vectors, e.g. as $\mathbf{x}$ and $\mathbf{x}'$. For a pair of correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$, the epipolar constraint becomes

$$\mathbf{x'}^{\top} \mathtt{E}\, \mathbf{x} = \mathbf{x}^{\top} \mathtt{E}^{\top} \mathbf{x}' = 0,$$

epipoles are normalized

$$\mathtt{E}\, \mathbf{e} = 0, \quad ||\mathbf{e}|| = 1 \qquad \text{and} \qquad \mathbf{e'}^{\top} \mathtt{E} = 0, \quad ||\mathbf{e}'|| = 1,$$

and normals of epipolar planes yielded by $\mathbf{x}$ and $\mathbf{x}'$ are given as

$$\begin{aligned} \mathbf{y}' &= \frac{\mathtt{E}^{\top}\mathbf{x}'}{||\mathtt{E}^{\top}\mathbf{x}'||} = \frac{([\mathbf{e}']_{\times}\mathtt{R})^{\top}\mathbf{x}'}{||([\mathbf{e}']_{\times}\mathtt{R})^{\top}\mathbf{x}'||} = \frac{\mathtt{R}^{\top}(\mathbf{x}' \times \mathbf{e}')}{||\mathbf{x}' \times \mathbf{e}'||} \\ \mathbf{y} &= \frac{\mathtt{E}\,\mathbf{x}}{||\mathtt{E}\,\mathbf{x}||} = \frac{\mathtt{R}\,[\mathbf{e}]_{\times}\,\mathbf{x}}{||\mathtt{R}\,[\mathbf{e}]_{\times}\,\mathbf{x}||} = \frac{\mathtt{R}\,(\mathbf{e} \times \mathbf{x})}{||\mathbf{e} \times \mathbf{x}||} \end{aligned} \tag{3}$$

assuming $\mathtt{E} = [\mathbf{e}']_{\times}\mathtt{R} = \mathtt{R}\,[\mathbf{e}]_{\times}$.

B.4.1 Image reprojection error

The closest point $\mathbf{v}$ on the epipolar plane generated by $\mathbf{x}'$ in the right image to the point $\mathbf{x}$ in the left image is obtained by rotating $\mathbf{x}$ around $\mathbf{e}$ and along the great circle passing through $\mathbf{x}$ and $\mathbf{y}'$ into $\mathbf{v}$, Fig. 9,

$$\mathbf{v} = \frac{(\mathbf{x} - (\mathbf{x}^{\top}\mathbf{y}')\,\mathbf{y}')}{||(\mathbf{x} - (\mathbf{x}^{\top}\mathbf{y}')\,\mathbf{y}')||}$$

and similarly in the right image

$$\mathbf{v}' = \frac{(\mathbf{x}' - (\mathbf{x'}^{\top}\mathbf{y})\,\mathbf{y})}{||(\mathbf{x}' - (\mathbf{x'}^{\top}\mathbf{y})\,\mathbf{y})||}.$$

We compute the "image reprojection error" as

$$\alpha^L = ||f(\mathbf{x}) - f(\mathbf{v})||, \;\; \alpha^R = \big|\big|f(\mathbf{x}') - f(\mathbf{v}')\big|\big|,$$

Algorithm 1 Camera motion estimation by ordered sampling from tentative matches with geometrical constraints

- Input: Image pair I_1, I_2.

  $\theta := 0.3\,^\circ$ ...the tolerance for establishing matches

  $\sigma := 4\,^\circ$ ...the standard deviation of Gaussian kernel for soft voting

  $N_V := 50$ ...the number of soft votes

  $N_S := 500$ ...the maximum number of random samples.

  $\eta := 0.95$ ...the termination probability of the standard RANSAC [5, p. 119].

- Output: Essential matrix $\mathtt{E}^*$.

1. Detect tentative matches and compute their descriptors.

   1.1 Detect affine covariant feature regions MSER-INT+, MSER-INT-, MSER-SAT+, MSER-SAT-, APTS-LAP, and APTS-HES in left and right images, Sec. B.1.1.

   1.2 Assign local affine frames (LAF) [15] to the regions and transform the regions into a standard position w.r.t. their LAFs.

   1.3 Compute Discrete Cosine Descriptors [16] for each region in the standard position.

2. Construct the list $M = [\mathbf{m}]_1^N$ of tentative matches with mutually closest descriptors. Order the list ascendingly by the distance of the descriptors. N is the length of the list.

3. Find a camera motion consistent with a large number of tentative matches:

   Set D to zero. // Initialize the accumulator of camera translation directions.
   for $i := 1, \ldots, N_V$ **do**
      $t := 0$ // The counter of samples. $n := 5$ // Initial segment length.
      $N_T := N_S$ // Initial termination length.
      while $t \leq N_T$ **do**
         if $t = \lceil 200000 \binom{n}{5} / \binom{N}{5} \rceil$ [2] **then**
            $n := n + 1$ // The maximum number of samples for the current initial segment reached, increase the initial segment length.
         end if
         $t := t + 1$ // New sample
         Select the 5 tentative matches M_5 of the t^{th} sample by taking 4 tentative matches from $[\mathbf{m}]_1^{n-1}$ at random and adding the 5^{th} match $\mathbf{m}_n$.
         $\mathtt{E}_t :=$ the essential matrix by solving the 5-point minimal problem for M_5 [13, 17].
         if M_5 can be reconstructed in front of cameras [5, p. 260] **then**
            $S_t :=$ the number of matches which are consistent with $\mathtt{E}_t$, i.e. the number of all matches $\mathbf{m} = [\mathbf{u}_1, \mathbf{u}_2]$ for which $\max(\measuredangle(\mathbf{u}_1, \mathtt{E}_t\mathbf{u}_2), \measuredangle(\mathbf{u}_2, \mathtt{E}_t^\top \mathbf{u}_1)) < \theta$.
         else
            $S_t := 0$
         end if
         $N_R := \log(\eta) / \log\left(1 - \binom{S_t}{5} / \binom{N}{5}\right)$ //The termination length defined by the maximality constraint [5, p. 119].
         $N_T := \min(N_T, N_R)$ // Update the termination length.
      end while
      $\hat{t} = \arg_{t=1,\ldots,N_S} \max S_t$ // The index of the sample with the highest support.
      $\hat{\mathtt{E}}_i := \mathtt{E}_{\hat{t}}$, $\hat{\mathbf{e}}_i :=$ camera motion direction for the essential matrix $\mathtt{E}_{\hat{t}}$.
      Vote in accumulator D by the Gaussian with sigma σ and mean at $\hat{\mathbf{e}}_i$.
   end for
   $\hat{\mathbf{e}} := \arg_{\mathbf{x} \in domain(D)} \max D(\mathbf{x})$ // Maximum in the accumulator.
   $i^* := \arg_{i=1,\ldots,50} \min \measuredangle(\hat{\mathbf{e}}, \hat{\mathbf{e}}_i)$ // The motion closest to the maximum.
   $\mathtt{E}^* := \hat{\mathtt{E}}_{i^*}$ // The "best" camera motion.

4. Return $\mathtt{E}^*$.

where f is the mapping from a ray directional vector to the corresponding image point, Fig. 9.

This is, of course, only an approximation of the true image reprojection error, since the closest points $\mathbf{v}$, $\mathbf{v}'$ are found in space, not in the image, but this approximation is very close to the true reprojection error, since the image epiplolar curves are very close to circles, and is much easier to compute.

The main character of the image reprojection error, i.e. that it is evaluated in the image where the point localization error is happenning, is preserved.

B.4.2 Angular error

The angular error

$$\begin{aligned} \beta^L &= \pi/2 - \arccos(\mathbf{x}^\top \mathbf{y}') = \arcsin(\mathbf{x}^\top \mathbf{y}') \\ \beta^R &= \pi/2 - \arccos(\mathbf{x}'^\top \mathbf{y}) = \arcsin(\mathbf{x}'^\top \mathbf{y}) \end{aligned}$$

corresponds to the angles between the ray direction vectors and the corresponding epipolar planes, Fig. 9. The angles are in general different in different cameras.

B.4.3 Longitudal error

The longitudal error, Fig. 9,

$$\begin{aligned} \gamma &= \arccos\left(\frac{(\mathbf{x}\times\mathbf{e})^\top}{||\mathbf{x}\times\mathbf{e}||}\frac{\mathtt{E}^\top\mathbf{x}'}{||\mathtt{E}^\top\mathbf{x}'||}\right) = \arccos\left(\mathbf{y}^\top \mathtt{R}\,\mathbf{y}'\right) = \arccos\left(\mathbf{y}'^\top \mathtt{R}^\top\,\mathbf{y}\right) \\ &= \arccos\left(\frac{(\mathbf{x}'\times\mathbf{e}')^\top}{||\mathbf{x}'\times\mathbf{e}'||}\frac{\mathtt{E}\,\mathbf{x}}{||\mathtt{E}\,\mathbf{x}||}\right). \end{aligned}$$

is same in both cameras and equals the angle between the epipolar planes.

B.5 Error sensitivity of 5-point algorithm

Figure 10 shows sensitivities of the 5-point algorithm [13] to the errors in point localization. The errors are evaluated and the sensitivities tested by the following algorithm.

1. Generate five points $\mathbf{X}$ in the space.
2. Project $\mathbf{X}$ into unit ray direction vectors $\mathbf{x}$ and $\mathbf{x}'$ in the left and right camera.
3. for $n = 1 : 5$

   Repeat the following steps 1000 times

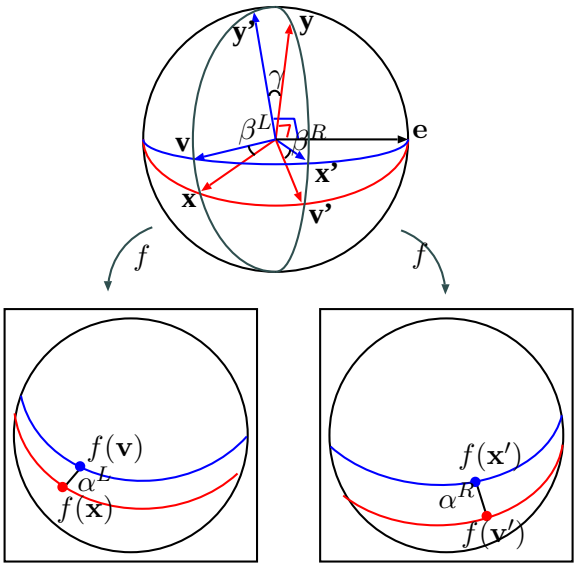


Figure 9: Residual errors. Vectors $\mathbf{x}$, $\mathbf{x}'$ represent a pair of corresponding image points represented by unit vectors. Vectors $\mathbf{y}$, $\mathbf{y}'$ represent normals of epipolar planes yielded by $\mathbf{x}$ and $\mathbf{x}'$. Vectors $\mathbf{v}$, $\mathbf{v}'$: The closest point $\mathbf{v}$ on the epipolar plane generated by $\mathbf{x}'$ to the point $\mathbf{x}$ in the left image obtained by rotating $\mathbf{x}$ around $\mathbf{e}$ and along the great circle passing through $\mathbf{x}$ and $\mathbf{y}'$ into $\mathbf{v}$. Angles α_L, α_R are the image reprojection errors. Angles β_L, β_R are the angular errors. Angle γ: is the longitudal error.

- Add uniformly distributed noise in the range s to $\mathbf{x}'^n_{i=1}$.
- Estimate E by the 5-points algorithm.
- Extract the epipoles from E and evaluate the angular errors to the ground truth (GT).

4. end

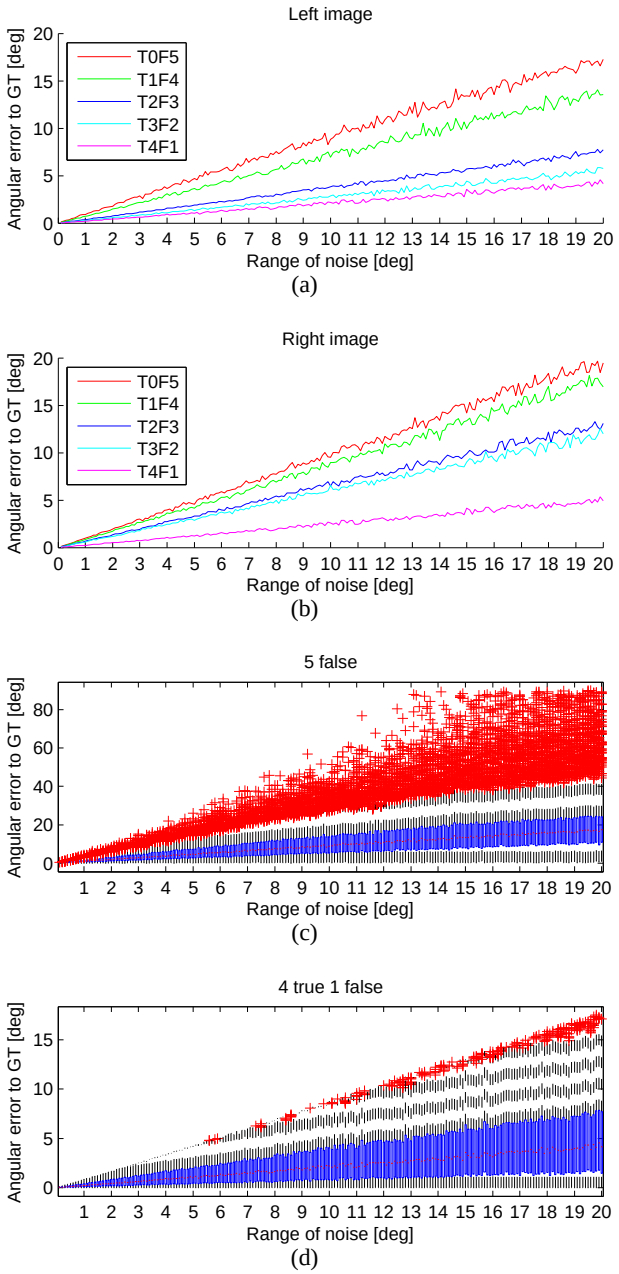


Figure 10: Sensitivity to errors of five points. x-axis: the size of the range of the uniformly ditributed noise. y-axis: the angle between the epipole estimated by the 5-point algorithm and the ground truth (GT). For each noise range, epipoles are estimated for 1000 nise realizations . Noise is added either to only one point (T4F1) or to all five points (T0F5). (a) and (b): the median of the angular error w.r.t the ground truth. (c) and (d): the angular errors of 'T4F1' and 'T0F5', respectively.

C Omnidirectional Camera Motion Estimation, Tracking and 3D Scene Modeling

In this section we describe a technique for tracking a pair of rigidly connected oimnidirectional cameras from a realatively dense video sequence of images. The main difference to the approach described in the previous section is in the density of the sequence which allows to use simpler and faster image feature detecton and extraction. The second difference that here we reconstruct the scene 3D structure to relate all camera positions to the original camera location.

A SfM framework [3] using a single perspective camera and a GPS/INS has been extended for using a stereo rig of omnidirectional cameras acquiring an omnidirectional stereo video sequence. The following paragraphs describe how the modified framework works in detail.

C.1 Features

Detection, description, and tracking of corner-like image features is a crucial part of the SfM framework. The green image channel is divided into square sections containing 8×8 pixels and at most one feature per section is found to achieve fast processing.

The feature strength is computed from the values of the pixels inside a square, which is divided into four subsquares and an average pixel value inside each of these subsquares is computed. The feature strength F is then evaluated as

$$F = |(M_{UL} + M_{DR}) - (M_{UR} + M_{DL})|, \tag{4}$$

where M_{UL}, M_{UR}, M_{DL}, and M_{DR} are average pixel values inside the up-left, up-right, down-left, and down-right subsquares respectively.

These features were primarily intended to detect corners of buildings and their windows and they reliably detect corners where horizontal and vertical lines meet. The detection becomes worse for rotated corners. Furthermore, objects captured in an omnidirectional image are radially distorted as they come closer and closer to the border of the circular view field and the feature strength can differ dramatically if computed on an object located in the center of the view field or on the same object few frames later when it moves closer to the border. However, when high video framerate is used and only strengths of features from consecutive frames are compared, this does not cause any problems.

Another problem arises from the shape of the image itself. Only a circular area near the center of the frame is covered with the view field, the rest of the frame is black. An additional position test had to be implemented in order not to allow the detection of the corners in the area out of the view field or on its border.

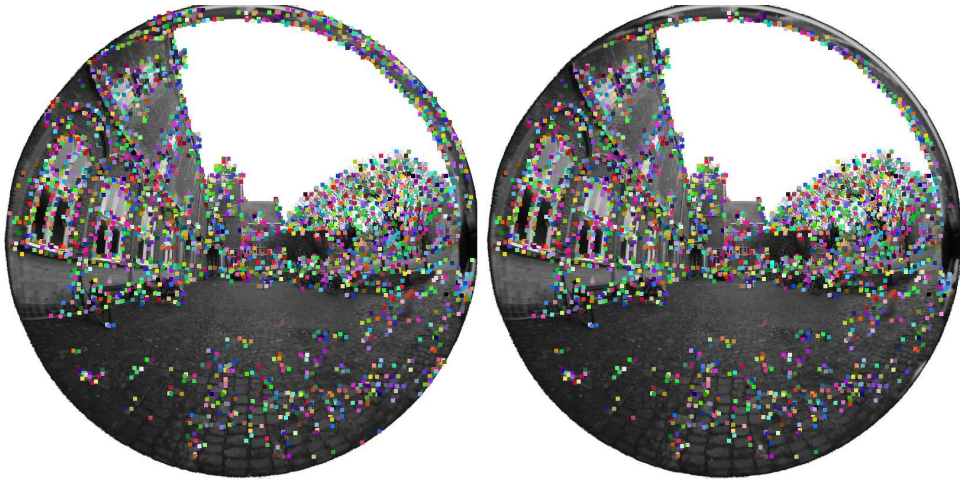


Figure 11: The output of the feature detector without an additional test for its position can be seen on the left image, the right image shows the corners that passed this test.

C.2 Omnidirectional Camera Calibration

Omnidirectional cameras are calibrated off-line using the technique describe above which builds on [1] and Mičušík's two-parameter model [9], 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 12, as

$$\theta = \frac{ar}{1 + br^2}. \tag{5}$$

Projecting via this model provides good results even when a low quality fish-eye lens is used because the second parameter 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. Conversion from pixel positions to rays is precomputed into a table to save computation time when performing the conversion.

C.3 Initialization of a Euclidean Reconstruction

The structure from motion computation starts by initialization. Known internal camera calibrations, which are then held constant for the whole video sequence, and a few initial camera poses are needed. Corners are tracked over 5-10 images and those, which are not lost, are then triangulated into world 3D points using known camera poses. The whole initialization is done independently for the left and for the right cameras, so two sets of world 3D points are computed.

Tracking is accomplished by connecting corners with small relative distances and small differences in the feature strengths for each section of the previously

(a) (b)

Figure 12: Diagram (a) shows the equi-angular projection performed by a fisheye lens. 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).

processed image. Correlation is computed to accept or to refuse matches and only the best 1-to-1 matches pass through the final test. As the images come from a high framerate video sequence, corners do not change their positions much, which is used to search only in small neighbourhoods.

C.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 corners 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 13, every reconstructed 3D point e.g. $X^R_{i,i+j}$ triangulated from corner positions x^R_i and x^R_{i+j} (if this point has not been refined yet) or x^R_i and y^L_{i+j} (if it has been refined already) is projected into the left and right images. We get matches $(X^R_{i,i+j}, x^R_{i+j+1})$ and $(X^R_{i,i+j}, y^L_{i+j+1})$ where x^R_{i+j+1} is the position of a similar corner near the projection to the image acquired by the camera where the reconstructed 3D point originated from and y^L_{i+j+1} is the position of a similar corner near the projection to image acquired by the other camera. Corners in images from both the left and the right cameras must be found to form a match $(X^R_{i,i+j}, x^R_{i+j+1}, y^L_{i+j+1})$.

Matches like $(X^R_{i,i+j}, y^L_{i+j+1})$ (not conventional matches like $(X^R_{i,i+j}, x^R_{i+j+1})$)

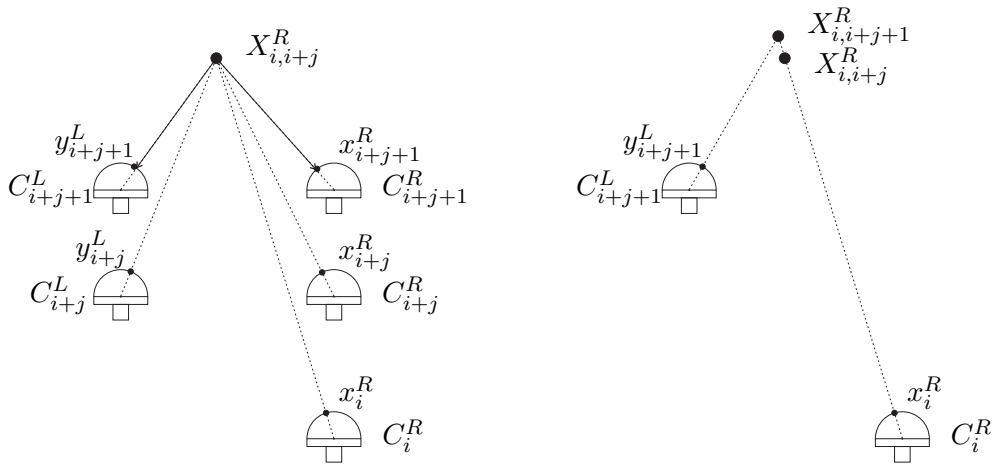


Figure 13: On the left you can see a 3D point $X^R_{i,i+j}$ triangulated from x^R_i and x^R_{i+j} or x^R_i and y^L_{i+j} projected into new images acquired by cameras C^L_{i+j+1} and C^R_{i+j+1}. Positions of the most similar corners are denoted by y^L_{i+j+1} and x^R_{i+j+1}. The diagram on the right shows 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}.

are used as the input to a hypothesis-and-test loop to force the pre-calibrated rigid stereo constraint which binds the left and right cameras into a stereo rig. We compute the rotation and the translation from three 3D-to-rays correspondences by Nister's algorithm [14]. The main advantage connected with this algorithm 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.

Nister's algorithm leads to an 8-degree polynomial. As there is no analytical way how to solve it, a numerical approach has to be used. The method described in Appendix A of [13] uses Sturm sequences and bisection with a fixed number of iterations and gives accurate results in reasonable time.

The RANSAC [4] stopping condition ensures stopping dependent on the probability of finding a better sample. Not to exceed the maximal processing time available, an upper threshold for the number of iterations is used. Having the size of the sample needed for setting up a hypothesis as small as 3 has a huge influence on early stopping of the RANSAC loop. To save even more time, the test for inliers is performed gradually on partitions of the matches and is stopped as soon as it is clear that the new hypothesis cannot be better then the best known at the time. 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 refinements using the Levenberg-Marquardt non-linear optimization are used to process all the inliers. The first refinement uses reprojection error as the cost function. As one cannot be sure that the set of inliers is correct and an outlier

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. Again, reprojection errors in both the left and 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 rigid stereo constraint is enforced in here again as corner positions x_i^R and y_{i+j+1}^L are used to triangulate the 3D point $X_{i,i+j+1}^R$. 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 corner 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. Even these tracks are prolonged but additional geometry constraints derived from the established camera poses are also used to restrict the set of possible locations of the corners that could prolong the tracks. First, a homography through a non-existent plane in a fixed distance in front of the camera is used to get an estimate of the position of the corner and a circular neighbourhood around this location is searched. This distance should be set to the expected average distance of the features. 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 corner position and the perpendicular projection of the ray going through the position of the corner into the matching epipolar plane, projected to the image.

C.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. The bundle adjustment routine running in parallel with the SfM framework refines the camera poses and the positions of world 3D points iteratively using the already finished data blocks. 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 eyes are treated completely independently.

During this procedure, only the tracks visible in 4 frames or more are used because they are considered to be more reliable than those which disappear very quickly. No global geometrical constraints are being used, that is why large errors in camera poses cannot be repaired by this simple bundle adjustment.

C.6 Experiments

Several experiments were performed to prove the functionality of the modified SfM framework. First, some hardware suitable for acquiring omnidirectional stereo data had to be chosen. The acquired image sequences were then used as the input for

Figure 14: Kyocera Finecam M410R cameras with Nikon FC-E9 fisheye lens convertors and two conventional perspective cameras mounted on a survey vehicle. Perspective cameras are not used in our experiments.

the SfM framework in two setups: using the rigid stereo constraint described in Section C and without using it.

C.6.1 Hardware

Finding an appropriate hardware is not an easy task. First, one has to choose whether to use a small or a high quality fisheye lens. Small lenses like Sunex DSL 125 can be mounted on industrial cameras like Unibrain Fire-I providing 30fps framerate. Unfortunatelly, not the whole circular field of view is captured when using this camera, the resulting view field is only 120×90 degrees. Other cameras like Pixeling are able to capture the whole view field (185 degrees) but the image quality is poor for desired feature detection because the radius of the resulting view field is only about 280 pixels and the image is blurred due to a low quality optics. There are also problems with synchronizing two cameras as they are capturing a video sequence and not single images.

When looking for a high quality fisheye lens, Nikon FC-E9 offering 183 degrees field of view seems like a good choice. This lens is rather big and heavy but it can be mounted on nearly any consumer camera. We have chosen Kyocera Finecam M410R because it was the only camera providing 3fps of high resolution images with the radius of the captured view field approximately 800 pixels and a very good image quality. We disassembled two cameras and connected them to an external trigger. The resulting compound device mounted on a survey vehicle can be seen in Figure 14.

C.6.2 Parameter Values

The SfM framework contains many parameter values which have to be set before running it. The first group of parameters concerns the detection and tracking of

Parameter name	value
feature halfwindow size	8×8 pixels
gradient threshold	16
minimum feature strength	16
search radius	64 pixels
maximum feature strength difference	4 per pixel
correlation window size	16×16 pixels
maximum correlation value difference	16 per pixel

Table 2: Parameters affecting the detection and the tracking of the corners.

the corners. The size of the feature window, the minimum gradient of the pixel intensity function, the minimum feature strength, the maximum allowed difference between the feature strengths of the two consecutive corners along a track, and the search radius constraining the area which is searched for the continuation of a track are some of them.

Experiments showed that while working with fisheye images shrinked to half size having the radius of the view field slightly smaller than 400 pixels, a 16×16 feature square gives best results. It is big enough to be well discriminative and severe radial distortion caused by the fisheye lens does not destroy the shape of the corners completely. Other parameters should be set carefully according to a concrete video sequence. Search radius should not be set smaller than the biggest expected movement in consecutive images, otherwise some good tracks would be lost. On the other hand, setting the thresholds too loose can cause a lot of false features and false tracks to appear and to make the reconstruction more difficult or even incorrect. Parameter values used with our test sequence can be found in Table 2.

The second group of parameters contains various thresholds used for world 3D points reconstruction. These thresholds should be independent on the input video sequence and their main significance lies in setting the ratio between the number of tracks that survive for a long time and the number of recently reconstructed tracks. Proposed parameter values are mentioned in Table 3.

C.6.3 Structure from Motion with the Rigid Stereo Constraint

There are several ways how 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. As we have the pre-calibrated rigid stereo constraint, starting camera poses for the right camera can be obtained by a simple transformation.

Another approach does not rely on the pre-calibrated rigid stereo constraint but computes the starting camera poses together with this constraint. An omnidirec-

Parameter name	value
max triangulation angle cosine	0.9995
min track length	2 frames
max initialization 3D point reprojection error	6 pixels
max inlier rep. error before refinement	6 pixels
max inlier rep. error after refinement	6 pixels
max distance from epipolar line	6 pixels
max new 3D point reprojection error	6 pixels

Table 3: Parameters affecting the reconstruction of the 3D points.

tional WBS software can be used to get epipolar geometries between the first left and first right, first left and sixth left, and first right and sixth left cameras. These geometries can be then combined together to get a consistent rigid stereo constraint and movement estimation.

Both approaches were tested and both work good. The main advantage of the first approach lies in the fact that one needs no additional WBS software to start the reconstruction so it is easier to use it in the final non-experimental setup. That is why the first approach was used in our experiments.

Our test sequence is 870 frames long and the first and the sixth image were used for the initialization with more than 200 correct tracks for each eye reconstructed into world 3D points. Straight street segments are quite easy, the support of the RANSAC winner is usually more than 70% 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. This is caused mostly by imprecise 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.

C.6.4 Structure from Motion without the Rigid Stereo Constraint

During the adaptation of the original SfM into an omnidirectional one, we first adapted the geometry and RANSAC without forcing the rigid stereo constraint [6]. 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 from the estimated left camera pose afterwards.

The framework worked fine when using additional GPS/INS data but failed when these data were not used. The reconstruction fails in the first sharp turn because the positions of world 3D points are not estimated well as the scale is being gradually lost.

A comparision with the original framework using perspective cameras was not performed but the result would be even worse because not only the missing rigid

stereo constraint but also the lack of features caused by a very small field of view would play a role.

C.6.5 Performance

The original SfM framework is able to work in realtime and it would be exciting to achieve the same speed even with fisheye cameras. Until now, we were interested more in functionality than in performance and the actual speed on a standard 2GHz Intel Pentium 4 computer is about 1.3fps. 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 corners and making the images much smaller will be possible only if a fisheye-oriented 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 3fps provided by our omnidirectional cameras are enough for the reconstruction from a moving vehicle because corners do not get lost from the image as quickly as when perspective cameras are used. That is why it is not necessary to achieve 25fps computational performance, 3fps are enough for realtime processing.

References

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

[2] Ondřej Chum and Jiří Matas. Matching with PROSAC - progressive sample consensus. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 220–226, Los Alamitos, USA, June 2005. IEEE Computer Society.

[3] N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *CVPR 2006*, volume 2, pages 1339–1344, 2006.

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

[5] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University, Cambridge, 2nd edition, 2003.

[6] Michal Havlena, Kurt Cornelis, and Tomáš Pajdla. Towards city modeling from omnidirectional video. In *CVWW 2007*, page ?, 2007. Submitted paper.

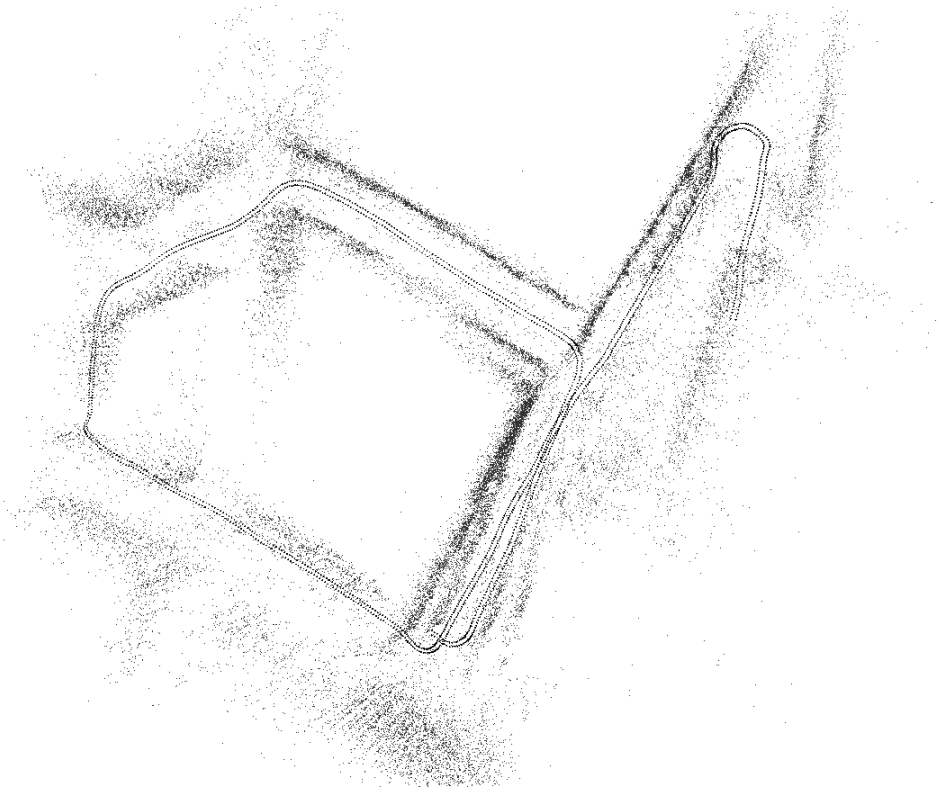


Figure 15: The resulting 3D model from the top view. The trajectory consists of a straight segment followed by a sharp turn and then a round-trip around a block of houses. The loop is not closed, mostly because of the errors arising in the sharp turns where the number of active tracks drops dramatically. Notice that the reconstruction nearly failed in the sharp turn in the bottom left corner because of a lack of features.

[7] H. Li and R. Hartley. A non-iterative method for correcting lens distortion from nine point correspondences. In *Proceedings of OMNIVIS*, 2005.

[8] Jiří Matas, Ondřej Chum, Martin Urban, and Tomáš Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, September 2004.

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

[10] Branislav Micusik and Tomas Pajdla. Structure from motion with wide circular field of view cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1135–1149, 2006.

[11] Branislav Mičušík and Tomáš Pajdla. Structure from motion with wide circular field of view cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1135–1149, July 2006.

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

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

[14] D. Nistér. A minimal solution to the generalized 3-point pose problem. In *CVPR 2004*, volume 1, pages 560–567, 2004.

[15] Štěpán Obdržálek and Jiří Matas. Object recognition using local affine frames on distinguished regions. In Paul L. Rosin and David Marshall, editors, *Proceedings of the British Machine Vision Conference*, volume 1, pages 113–122, London, UK, September 2002. BMVA.

[16] Štěpán Obdržálek and Jiří Matas. Image retrieval using local compact dct-based representation. In Bernd Michaelis and Gerald Krell, editors, *DAGM 2003: Proceedings of the 25th DAGM Symposium*, volume 1 of *LNCS*, pages 490–497, Berlin, Germany, 9 2003. Springer-Verlag.

[17] H. Stewenius. *Gröbner basis methods for minimal problems in computer vision*. PhD thesis, Lund University, 2005.