Closing the Loop in Appearance-Guided Omnidirectional Visual Odometry by Using Vocabulary Trees

Davide Scaramuzza[1], Friedrich Fraundorfer[2],
Marc Pollefeys[2], and Roland Siegwart[1]

[1]Autonomous Systems Lab, ETH Zurich
[2]Computer Vision and Geometry Lab, ETH Zurich

Abstract. In this paper, we present a method that allows us to recover the trajectory of a vehicle purely from monocular omnidirectional images very accurately. The method uses a combination of appearance-guided structure from motion and loop closing. The appearance-guided monocular structure-from-motion scheme is used for initial motion estimation. Appearance information is used to correct the rotation estimates computed from feature points only. A place recognition scheme is employed for loop detection, which works with a visual word based approach. Loop closing is done by bundle adjustment minimizing the reprojection error of feature matches. The proposed method is successfully demonstrated on videos from an automotive platform. The experiments show that the use of appearance information leads to superior motion estimates compared to a purely feature based approach. And we demonstrate a working loop closing method which eliminates the residual drift errors of the motion estimation.

1 Introduction

Robust and reliable trajectory recovery for automotive applications using visual input only needs a very accurate motion estimation step and loop closing for removing the inevitably accumulated drift. The first focus of this paper is in using the appearance information to improve the results of feature based motion estimation. The second focus is in removing accumulated drift with loop closing.

In this paper, we use a single calibrated catadioptric camera mounted on the roof of the car (Fig. 1). The first step of our approach is to extract SIFT keypoints [1] from the scene all around the car and match them between consecutive frames. After RANSAC based outlier removal [2], we use these features to compute the translation in the heading direction only. To estimate the rotation angle of the vehicle we instead use an appearance based method. We show that by using appearance information our result outperforms the pure feature based approach. At the same time as motion estimation, a loop detection algorithm is running. We use a visual word based approach [3–5] that is very fast and highly scalable. In addition, we designed a geometric loop verification especially for omnidirectional

Fig. 1. Our vehicle with the omnidirectional camera (blue circle). The field of view is indicated by the red lines.

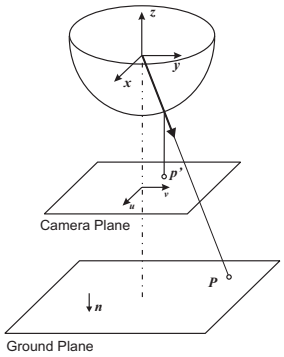


Fig. 2. Our omnidirectional camera model.

images. Loop closing is then finally done in an optimization step using bundle adjustment. The method is demonstrated on motion estimation. We demonstrate this method on video data from a 400m trajectory and show that the initial motion estimation is already very accurate.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes our homography based motion estimation which is used for translation estimation. Section 4 describes the details about the appearance guided Structure from Motion (SfM) which corrects the rotation estimates. Section 5 details the steps of the whole SfM algorithm. Section 6 describes the loop closing algorithm. Finally, Section 7 is dedicated to the experimental results.

2 Related Work

Structure from motion and motion estimation (also called visual odometry) with omnidirectional cameras has already been investigated from various groups [6–8]. The benefit of camera trajectory determination using large field of view was firstly demonstrated by Svoboda *et al.* [9] and further recognized by Chang and Hebert [10]. However, in those works SfM was performed only over short distances (up to a few meters). Conversely, in this paper we concentrate on SfM and accurate trajectory recovery over long distances (hundreds of meters).

Motion estimation with omnidirectional cameras for long trajectories has been investigated by [11–13]. In [11], Corke *et al.* provided two approaches for monocular visual odometry based on omnidirectional imagery. As their approach was conceived for a planetary rover, they performed experiments in the desert and therefore used keypoints from the ground plane. In the first approach, they used optical flow computation with planar motion assumption while in the second one SfM with no constrained motion using an extended Kalman filter. The optical flow based approach gave the best performance over 250 meters but the

trajectory was not accurately recovered showing a large drift of the rotation estimation. Another approach with robust and reliable motion estimation was presented by Lhuillier [12] where only keypoint tracking and bundle adjustment were used to recover both the motion and the 3D map. None of these methods however address loop detection and loop closing.

Loop closing in general was described by Bosse *et al.* [14]. The approach worked by map matching of laser scans using 3D points and 3D lines as features. Map matching with 3D point and 3D line features from image data was demonstrated in [15]. Experiments were shown on indoor and urban areas with plenty of line features. No result is shown on image data similar to ours, where almost no line features are present. Loop detection in a similar manner to ours is also described in [16]. In their case a loop detection probability is computed from the visual similarity of images. In contrast to our approach no geometric verification of visual features is performed and no actual loop closure using the detected loops is done. Loop detection and loop closure is successfully demonstrated in [17] where they use a laser range finder to get the initial trajectory and image data for loop detection. Loop closure however also relies on information from the laser range finder, whereas our proposed approach uses visual features only.

In this paper we extend our previous work on appearance-guided visual odometry for outdoor ground vehicles [18, 19] by addressing the problem of loop detection and closing. To make this paper self consistent, we summarize our previous work in Section 3 and 4.

3 Homography Based Motion Estimation

Our initial motion estimation proceeds along the lines of our previous work (please refer to [18, 19] for major details). The method uses planar constraints and point tracking to compute the motion parameters. As we assume planar motion and that the camera is orthogonal to the ground plane with quite a good approximation, only two points are needed to estimate the motion parameters up to a scale (the scale is then recovered from the height of the camera). Then, after a two-point RANSAC based outlier removal, rotation and translation parameters between consecutive frames are computed from the remained inliers. For this the homography decomposition of Triggs [20] is used being adapted to omnidirectional images. A subsequent non-linear refinement improves the accuracy. In this we constrain the minimization so that the rotation is about the plane normal and the translation is parallel to the same plane as we assume planar motion. More details about our motion estimation step can be found in our previous work [18].

4 Visual Compass

Unfortunately, when using point features to estimate the motion, the resulting rotation is extremely sensitive to systematic errors due to the intrinsic calibration of the camera or the extrinsic calibration between the camera and the ground

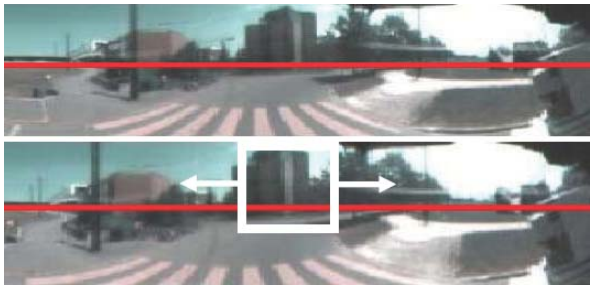


Fig. 3. Two unwrapped omnidirectional images taken at consecutive time stamps. For reasons of space, here only one half of the whole 360 *deg* is shown. The red line indicates the horizon.

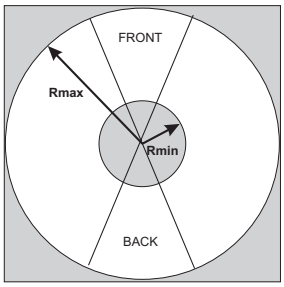


Fig. 4. The cylindrical panorama is obtained by unwrapping the white region.

plane. This effect is even more accentuated with omnidirectional cameras due to the large distortion introduced by the mirror. In addition to this, integrating rotational information over the time has the major drawback of generally becoming less and less accurate as integration introduces additive errors at each step. An example of camera trajectory recovered using only the feature based approach described in Section 3 is depicted in Fig. 6 (blue trajectory).

To improve the accuracy of the rotation estimation, we use an appearance based approach. This approach was inspired by the work of Labrosse [21], which describes a method to use omnidirectional cameras as visual compass.

Directly using the appearance of the world as opposed to extracting features or structure of the world is attractive because methods can be devised that do not need precise calibration steps. Here, we describe how we implemented our visual compass.

For ease of processing, every omnidirectional image is unwrapped into cylindrical panoramas (Fig. 3). The unwrapping considers only the white region of the omnidirectional image that is depicted in Fig 4. We call these unwrapped versions "appearances". If the camera is perfectly vertical to the ground, then a pure rotation about its vertical axis will result in a simple column-wise shift of the appearance in the opposite direction. The exact rotation angle could then be retrieved by simply finding the best match between a reference image (before rotation) and a column-wise shift of the successive image (after rotation). The best shift is directly related to the rotation angle undertaken by the camera. In the general motion, translational information is also present. This general case will be discussed later.

The input to our rotation estimation scheme is thus made of appearances that need to be compared. To compare them, we use the Euclidean distance. The Euclidean distance between two appearances I_i and I_j, with I_j being column-

wise shifted (with column wrapping) by α pixels, is:

$$d(I_i, I_j, \alpha) = \sqrt{\sum_{k=1}^{h} \sum_{h=1}^{w} \sum_{l=1}^{c} |I_i(k,h,l) - I_j(k,h-\alpha,l)|^2} \tag{1}$$

where $h \times w$ is the image size, and c is the number of color components. In our experiments, we used the RGB color space, thus having three color components per pixel.

If α_m is the best shift that minimizes the distance between two appearances I_i and I_j, the rotation angle $\Delta\vartheta$ (in degrees) between I_i and I_j can be computed as:

$$\Delta\vartheta = \alpha_m \cdot \frac{360}{w} \tag{2}$$

The width w of the appearance is the width of the omnidirectional image after unwrapping and can be chosen arbitrarily. In our experiments, we used $w = 360$, that means the angular resolution was 1 pixel per degree. To increase the resolution to 0.1 *deg*, we used cubic spline interpolation with 0.1 pixel precision. We also tried larger image widths but we did not get any remarkable improvement in the final result. Thus, we used $w = 360$ as the unwrapping can be done in a negligible amount of time.

The distance minimization in (1) makes sense only when the camera undergoes a pure rotation about its vertical axis, as a rotation corresponds to a horizontal shift in the appearance. In the real case, the vehicle is moving and translational component is present. However, the "pure rotation" assumption still holds if the camera undergoes small displacements or the distance to the objects (buildings, tree, etc.) is big compared to the displacement. In the other cases, this assumption does not hold for the whole image but an improvement that can be done over the theoretical method is to only consider parts of the images, namely the front and back part (Fig. 4). Indeed, the contribution to the optical flow by the motion of the camera is not homogeneous in omnidirectional images; a forward/backward translation mostly contributes in the regions corresponding to the sides of the camera and very little in the parts corresponding to the front and back of the camera, while the rotation contributes equally everywhere.

Because we are interested in extracting the rotation information, only considering the regions of the images corresponding to the front and back of the camera allows us to reduce most of the problems introduced by the translation, in particular sudden changes in appearance (parallax).

According to the last considerations, in our experiments we use a reduced Field Of View (FOV) around the front and back of the camera (Fig. 4). A reduced field of view of about 30 *deg* around the front part is shown by the white window in Fig. 3. Observe that, besides reducing the FOV of the camera in the horizontal plane, we operate a reduction of the FOV also in the vertical plane, in particular under the horizon line. The objective is to reduce the influence of the changes in

appearance of the road. The resulting vertical FOV is 50 *deg* above and 10 *deg* below the horizon line (the horizon line is indicated in red in Fig. 3).

5 Motion Estimation Algorithm

As we already mentioned, the appearance based approach provides rotation angle estimates that are more reliable and stable than those output by the pure feature based approach. However, in case of occlusions by other passing vehicles, the rotation estimate computed by the visual compass may be wrong. This situation can be easily detected by comparing it with the rotation estimate obtained from the features, indeed we found out that during motion without occlusions the difference between the two estimates is very small ($\sim$ 1 deg). Thus, when this difference is too big (in practice we used 5 deg) we replace the rotation estimate from the visual compass with that from the feature based approach.

Here, we describe how we combined the rotation angle estimates of Section 4 with the camera translation estimates of Section 3.

In our experiments, the speed of the vehicle ranged between 10 and 20 Km/h while the images were constantly captured at 10 Hz. This means that the distance covered between two consecutive frames ranged between 0.3 and 0.6 meters. For this short distance, the kinematic model of the camera configuration (x, y, θ), which contains its 2D position (x,y) and orientation θ, can be approximated with that used for differential drive robots [22]:

$$\begin{cases} x_{i+1} = x_i \ + \ \delta\rho_i \cos(\theta_i + \frac{\delta\theta_i}{2}) \\ y_{i+1} = y_i \ + \ \delta\rho_i \sin(\theta_i + \frac{\delta\theta_i}{2}) \\ \theta_{i+1} = \theta_i \ + \ \delta\theta_i \end{cases} \tag{3}$$

where we use $\delta\rho = |\mathbf{T}|\ h$ and $\delta\theta = \Delta\vartheta$. $|\mathbf{T}|$ is the length of the translation vector assuming the camera at unit distance from the ground plane; h is the scale factor (i.e. in our case this is the height of the camera to the ground plane). The camera rotation angle $\Delta\vartheta$ is computed as in (2). Observe that we did not use at all the rotation estimates provided by the feature based method of Section 3.

Now, let us resume the steps of our motion estimation scheme, which have been described in Sections 3 and 4. Our omnidirectional visual odometry operates as follows:

1. Acquire two consecutive frames. Consider only the region of the omnidirectional image, which is between *Rmin* and *Rmax* (Fig. 4).
2. Extract and match SIFT features between the two frames. Use the double consistency check to reduce the number of outliers. Then, use the calibrated omnidirectional camera model to normalize the feature coordinates so that the z-coordinate is equal to -1 (see Fig. 2).
3. Use 2-point RANSAC to reject points that are not coplanar; the homography is finally computed from the remaining inliers.
4. Apply the Triggs algorithm followed by non-linear refinement described in Section 3 to estimate $\mathbf{R}$ and $\mathbf{T}$ from the remaining inliers.

5. Unwrap the two images and compare them using the appearance method described in Section 4. In particular, minimize (1), with reduced field of view, to compute the column-wise shift α_m between the appearances and use (2) to compute the rotation angle $\Delta\vartheta$.
6. Use $\delta\rho = |\mathbf{T}|\ h$ and $\delta\theta = \Delta\vartheta$ and integrate the motion using (3).
7. Repeat from step 1.

6 Vision based loop detection and closing

Loop closing is essential to remove accumulated drift errors in the camera trajectory. Our loop closing proceeds along three steps. The first step is loop detection which is done by a visual place recognition scheme. Next, geometric correspondences between the two matching places are established. Finally, loop closing is performed by optimization of structure and motion using bundle adjustment.

6.1 Loop detection by visual place recognition

For loop detection we use a visual word based place recognition system. Each image that got acquired is assigned a place in the world. To recognize places that have been visited before, image similarity (based on local features) is used. Our approach is along the lines of the method described in [4]. Firstly, local features are extracted from images. We use Difference of Gaussian (DoG) keypoints and compute a SIFT feature vector for each keypoint [1]. Each SIFT feature vector is quantized by a hierarchical vocabulary tree. The quantization assigns a single integer value, called a visual word (VW), to the originally 128-dimensional SIFT feature vector. This results in a very compact image representation, where each image is represented by a list of visual words, each only of integer size. The list of visual words from one image forms a document vector which is a v-dimensional vector where v is the number of possible visual words (we use $v = 10^6$). The document vector is a histogram of visual words normalized to 1. For place recognition the similarity between the query document vector to all document vectors in a database is computed. As similarity score we use the L_2 distance between document vectors. The document vectors are very sparse and therefore we only save the non-zero entries of it. Furthermore, the organization of the database as an inverted file structure [4] enables a very efficient scoring. For scoring, the different visual words are weighted based on the Inverse Document Frequency (IDF) measure [4]. Visual words that occur in many images of the database are not very discriminative and have therefore a low weight, while visual words that appear only rarely have a higher weight. The visual word based place recognition approach is illustrated in Fig. 5. The place recognition in our case works as an online approach. For each new image the similarity to all the images in the database is computed. The n top ranked images are stored as loop hypotheses. After similarity computation the current image is then added to the database as well. The loop hypotheses are then geometrically verified. If one hypothesis passes this verification the loop closing optimization will be invoked. The visual

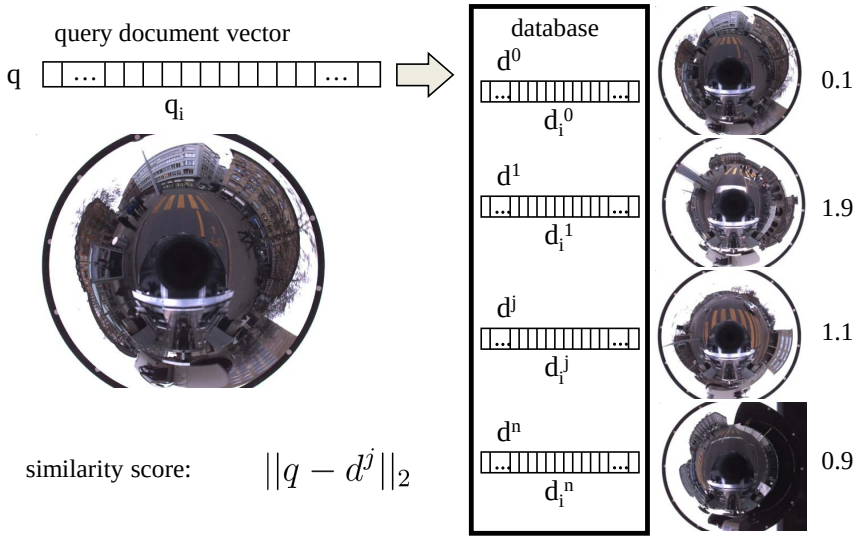


Fig. 5. Illustration of the visual place recognition system. A query image is represented by a query document vector (a set of visual words). A query works by computing the L_2-distance between the query document vector and all the document vectors in a database which represent the stored images, i.e. places. The L_2-distance is used as similarity score and is in the range of $[0, 2]$. A similarity score close to zero stands for a very similar image, while a similarity score close to 2 stands for a different image. The computation of the similarity score is using an inverted file structure for efficiency.

word based place recognition system is efficient and maintains online speed for large databases. Adding a new image to the database is done in constant time, i.e. independent of the database size. The query time theoretically grows linearly with the database size. See Table 1 for typical query times of our system. We created databases of different sizes up to 1 million images, with an average of 1000 visual words per image. For a database with a couple of hundred images as used in our experiments the query times are less than 1ms on a Intel Xeon PC with 2.8GHz. The visual word based image representation is extremely compact, 32 bit per visual word, which means that a database of 1 million images (with an average of 1000 visual words per image) takes 4GB of memory.

6.2 Geometric correspondences and loop verification

For loop closing geometric correspondences need to be established for the loop hypothesis. The necessary 2D point matches between the images from matching places are created by the geometric verification step. The geometric verification is designed for omnidirectional images. Besides the image similarity the place recognition also returns 2D point correspondences between the two images. These are the point correspondences used for geometric verification. For verification we first compute a polar coordinate representation (r, ϕ) from the (x, y) image

database size [number of images]	query time
300	0.175ms
1000	0.435ms
5000	2.038ms
10000	7.317ms
50000	48.035ms
100000	107.251ms
500000	0.873s
1000000	3.661s

Table 1. Query times for the place recognition system for different database sizes on an Intel Xeon 2.8GHz PC with an average of 1000 features per image.

coordinates. We assume that the vehicle poses for both images differ only by a planar rotation, neglecting a possible translation. In our omnidirectional images a planar rotation of the vehicle is visible as an in-plane rotation around the center of the omnidirectional image. The angle of the in-plane rotation can be computed from a single 2D point match and all the other 2D point matches will have the same rotation angle if they are correct matches. This property is used for the geometric verification algorithm. For every 2D point match the in-plane rotation angle is computed and we search for the largest set of 2D point matches having the same in-plane rotation angle (up to some ϵ). The number of points in the largest set is our new geometric similarity score. If the value is higher than a threshold S the loop hypothesis is counted as correct. The set of 2D point matches is then used in the loop closing process.

6.3 Loop closing

Loop closing is carried out as bundle adjustment [23] over camera poses and structure (3D points). Bundle adjustment performs non-linear optimization of camera (vehicle) poses $\mathbf{P}$ and 3D points $\mathbf{X}$ to minimize the reprojection error, i.e. the L_2-distance between measured 2D points $\mathbf{x}$ and 3D points projected into the image $\mathbf{x}'$ with the current camera pose estimate. This cost function is computed as

$$e = \sum_{i,n} ||x_{i,n} - P_n X_i|| \tag{4}$$

where n is an index over all cameras and i is an index over all 3D points. A Levenberg-Marquardt optimization method is used [23].

Only planar motion is assumed, so that we have to optimize 3 parameters (x, y, θ) per camera. Each 3D point also has 3 parameters (x, y, z coordinates). Initial poses and 3D points come from the SfM algorithm. The detected loop now adds an extra constraint to the optimization. From loop detection and geometric verification we get a set of 2D point matches x_0 and x_1 for the camera positions P_0 and P_1. From the SfM algorithm we know the 3D coordinates of

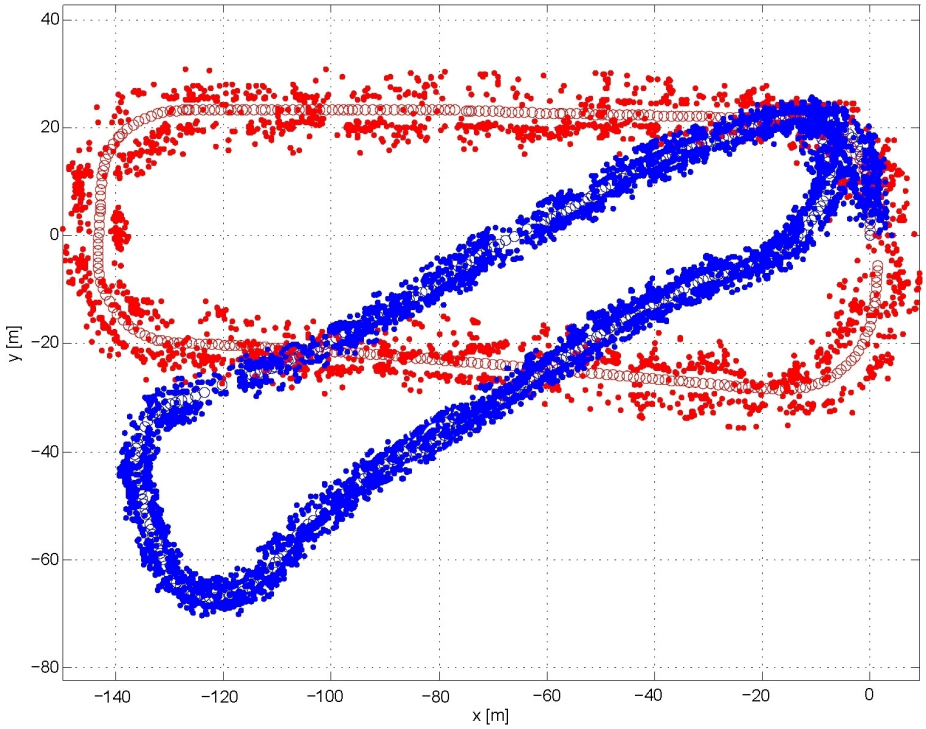


Fig. 6. Comparison between the standard feature based approach (blue) and the approach combining features with visual compass proposed in this paper (red).Circles are camera poses, dots are reconstructed feature points.

these points as well, denoted as X_0 and X_1. As the SfM algorithm does not know the correspondence between x_0 and x_1 the corresponding 3D points differ. The loop closing constraint tells us that the 3D points need to be the same and that a reprojection of the 3D points must be equal to x_0 and x_1. In the BA data structure we add the information that X_0 needs to reproject to x_1 as well and not only into x_0. After adding this loop constraint the optimization can be started and this will correct the camera poses and 3D points so that the loop is closed. Loop closing using full optimization of 3D points and camera poses usually cannot be done with online speed, but it is scalable up to large maps using for instance the method described in [24].

7 Results

7.1 Motion estimation

This experiment shows motion estimation results using the proposed approach and also compares it to a feature only based approach. It was tested with data

Fig. 7. The estimated path (before loop closing) superimposed onto a Google Earth image of the test environment. The scale is shown at the lower left corner. The cyan arrow indicates the position of the three road humps. The yellow arrow indicates the starting position of the vehicle.

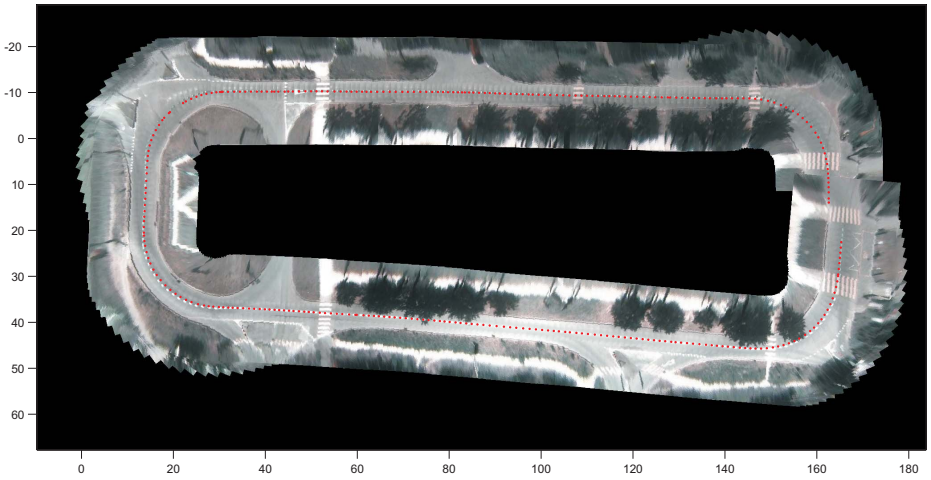


Fig. 8. Image mosaicing that shows a textured 2D reconstruction of the estimated path before closing the loop.

from a real vehicle equipped with a central omnidirectional camera. A picture of our vehicle (a Smart) is shown in Fig 1. Our omnidirectional camera, composed of a hyperbolic mirror (KAIDAN 360 One VR) and a digital color camera (SONY XCD-SX910, image size 640×480 pixels), was installed on the front part of the roof of the vehicle. The frames were grabbed at 10 Hz and the vehicle speed

ranged between 10 and 20 Km/h. The resulting path estimated by our SfM algorithm using a horizontal reduced FOV of 10 *deg* is shown in figures 6, 7, and 8. The results of the feature only based approach in Fig. 6 are shown in blue. The trajectory with our appearance based method is shown in red. From the aerial view in 7 it is clear that our method produces the correct trajectory.

In this experiment, the vehicle was driven along a 400 meter long loop and returned to its starting position (pointed to by the yellow arrow in Fig. 7). About 800 images were collected overall. The estimated path is indicated with red dots in Fig. 7 and is shown superimposed on the aerial image for comparison. The final error at the loop closure is about 6.5 meters. This error is due to the unavoidable visual odometry drift; however, observe that the trajectory is very well estimated until the third 90-degree turn. After the third turn, the estimated path deviates smoothly from the expected path instead of continuing straight. After road inspection, we found that the reason for this were most likely three 0.3 meter tall road humps (pointed to by the cyan arrow in Fig. 7).

The content of Fig. 8 is very important as it allows us to evaluate the quality of motion estimation. In this figure, we show a textured top viewed 2D reconstruction of the whole path. Observe that this image is not an aerial image but is an image mosaicing. Every input image of this mosaic was obtained by an Inverse Perspective Mapping (IPM) of the original omnidirectional image onto an horizontal plane. After being undistorted through IPM, these images have been merged together using the 2D poses estimated by our visual odometry algorithm. The estimated trajectory of the camera is shown superimposed with red dots. If the reader visually compares the mosaic (Fig. 8) with the corresponding aerial image (Fig. 7), he will recognize in the mosaic the same elements that are present in the aerial image, that is, trees, white footpaths, pedestrian crossings, roads' placement, etc. Furthermore, as can be verified, the position of these elements in the mosaic fits well the position of the same elements in the aerial image.

7.2 Motion estimation with loop closing

We performed the same experiment as in the previous section with loop closing running in parallel. For each new image we query the database to search for similar images that indicate that we are returning to a known place. Afterwards the image is added to the database. We use a vocabulary tree pre-trained on a general 100000 image database. The images of the test sequence were not included. Fig. 9 shows the similarity matrix for the experiment. The depicted similarity value is the number of geometrically consistent point matches between the current image and the images in the database. The top-5 query results were used as entries to the similarity matrix. Temporally close images (neighboring places) have usually a large overlap and therefore high similarity. This accounts for the high similarity along the diagonal. Images that are temporally too close are not used as loop hypothesis. For this experiment we required a distance of 50 frames between two frames to be used as a loop hypothesis. The bottom left corner of the similarity matrix shows a spot of high similarity. This is when the car returns to the starting position. In particular frame 298 has a very

high similarity to frame 2. The similarity value is higher than the set threshold of 10 and the pair $298 \leftrightarrow 2$ is accepted as a loop. The 2D point matches of the detected loop are used as additional constraints in the bundle adjustment. Fig. 10 shows the results after loop closing. The optimized trajectory is shown in red, while the initial parameters are shown in blue and black. The circles are the camera position and the dots are reconstructed feature points. The loop got nicely closed. Fig. 11 shows the path after loop closing superimposed on a Google Earth aerial view. It is an almost perfect fit. Fig. 12 shows the image mosaic created from the optimized path. Fig. 13 visualizes the motion estimation error before and after loop closing using feature matches. 3D points from the start of the sequence are projected into frame 2 (shown in red). The corresponding 3D points from the end of the sequence are projected also and shown in blue. Due to drift the 3D points do not have the same coordinates although the should. Thus the projected points differ largely. After bundle adjustment which incorporates the loop constraint, that the 3D points correspond, both projections overlap. Quantitatively the optimization reduced the sum of the reprojection error of the matching features from 851.8 pixel to 13.8 pixel.

8 Conclusion

It this paper, we presented a reliable and robust method for structure from motion for omnidirectional cameras. Initial motion estimates from feature tracks were corrected using appearance information. Finally loop detection and loop closing removed accumulated drift. The proposed method runs in real-time (30 fps) and is scalable. The method however assumes planar motion which is in particular the case in many automotive and robotics applications. Only features from the ground plane are tracked, i.e. this approach will also work outside urban areas and it is perfectly suited for outdoor areas. In our experiments we demonstrated motion estimation on a challenging 400m long path that is one of the longest distances ever reported with a single omnidirectional camera. We showed that the use of appearance information enormously improved the motion estimates resulting in a very accurate estimate. Loop closing finally again improved the accuracy by removing accumulated drift.

9 Acknowledgment

The research leading to these results has received funding from the European Commission Division FP6-IST Future and Emerging Technologies under the contract FP6-IST-027140 (BACS: Bayesian Approach to Cognitive Systems).

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **20** (2003) 91–110

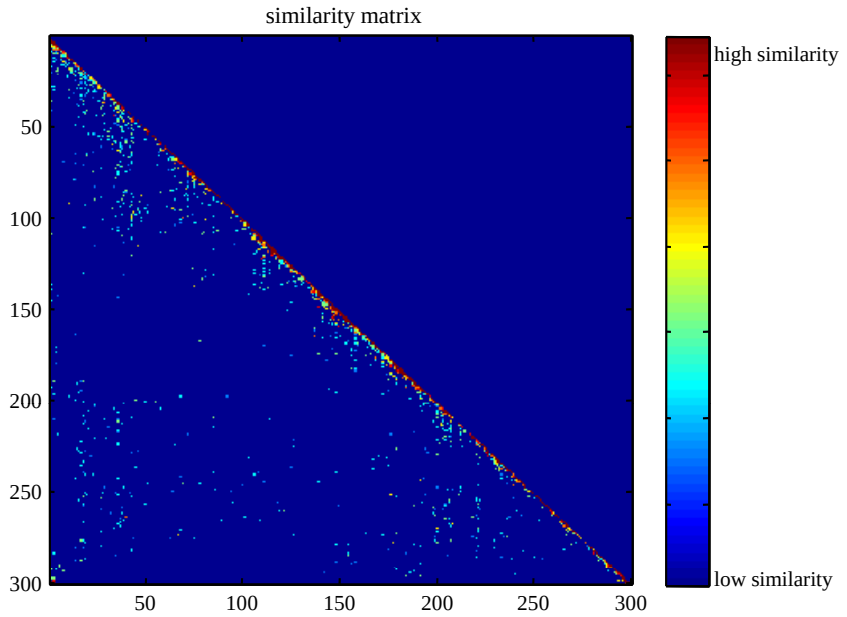


Fig. 9. Similarity matrix for our dataset. The value of similarity is the number of geometrically consistent matches between the current image and the images in the database. The top-5 query results were used to create the similarity matrix. Note, the small spot of high similarity in the bottom left corner is the detected loop when the vehicle returns to the starting position. There is also high similarity along the diagonal as the temporally close images are also very similar.

2. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24** (1981) 381–395
3. Engels, F.F.C., Nister, D.: Topological mapping, localization and navigation using image collections. In: IROS. (2007)
4. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, New York City, New York. (2006) 2161–2168
5. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. 9th International Conference on Computer Vision, Nice, France. (2003) 1470–1477
6. Micusik, B., Pajdla, T.: Autocalibration and 3d reconstruction with non-central catadioptric cameras. In: CVPR 2004. (2004)
7. Micusik, B., Pajdla, T.: Structure from motion with wide circular field of view cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 1135–1149

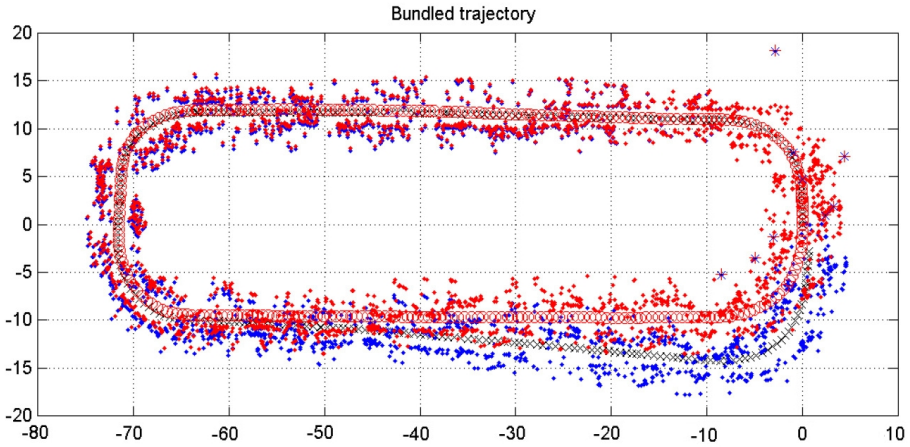


Fig. 10. Structure and motion after loop closing (red). Initial estimates are shown in blue and black. The loop is nicely closed.

Fig. 11. The estimated path after loop closing superimposed onto a Google Earth.

8. Geyer, C., Daniilidis, K.: Structure and motion from uncalibrated catadioptric views. In: CVPR 2001. (2001)
9. Svoboda, T., Pajdla, T., Hlavac, V.: Motion estimation using central panoramic cameras. In: IEEE Int. Conf. on Intelligent Vehicles. (1998)
10. Chang, P., Hebert, M.: Omni-directional structure from motion. In: IEEE Workshop on Omnidirectional Vision. (2000)

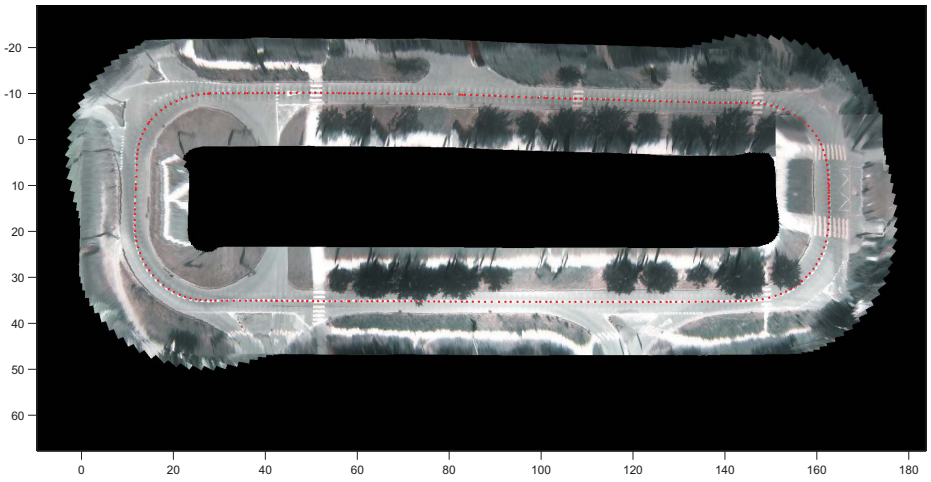
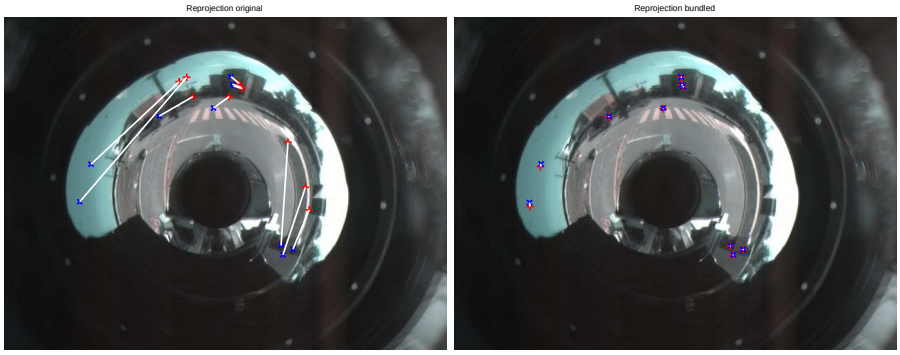


Fig. 12. Image mosaicing of the estimated path after loop closing.



(a) (b)

Fig. 13. Visualization of the motion estimation error before and after loop closing using feature matches. (a) Before loop closing. Reprojected features do not coincide. (b) After loop closing. Reprojected features coincide.

11. Corke, P.I., Strelow, D., , Singh, S.: Omnidirectional visual odometry for a planetary rover. In: IROS. (2004)
12. Lhuillier, M.: Automatic structure and motion using a catadioptric camera. In: IEEE Workshop on Omnidirectional Vision. (2005)
13. Bosse, M., Rikoski, R., Leonard, J., Teller, S.: Vanishing points and 3d lines from omnidirectional video. In: ICIP02. (2002) III: 513–516
14. Bosse, M., Newman, P., Leonard, J., Teller, S.: Simultaneous localization and map building in large cyclic environments using the atlas framework. The International Journal of Robotics Research **23** (2004) 1113–1139
15. Bosse, M.: ATLAS: A Framework for Large Scale Automated Mapping and Localization. PhD thesis, Massachusetts Institute of Technology (2004)

16. Cummins, M., Newman, P.: Probabilistic appearance based navigation and loop closing. In: IEEE International Conference on Robotics and Automation (ICRA'07), Rome (2007)
17. Ho, K.L., Newman, P.: Detecting loop closure with scene sequences. International Journal of Computer Vision **74** (2007) 261–286
18. Scaramuzza, D., Siegwart, R.: Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. IEEE Transactions on Robotics, Special Issue on Visual SLAM **24** (2008)
19. Scaramuzza, D., Siegwart, R.: Monocular omnidirectional visual odometry for outdoor ground vehicles. In: 6th International Conference on Computer Vision Systems. (2008) 206–215
20. Triggs, B.: Autocalibration from planar scenes. In: ECCV98. (1998)
21. Labrosse, F.: The visual compass: performance and limitations of an appearance-based method. Journal of Field Robotics **23** (2006) 913–941
22. Siegwart, R., Nourbakhsh, I.: Introduction to Autonomous Mobile Robots. MIT Press (2004)
23. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment: A modern synthesis. In: Vision Algorithms Workshop: Theory and Practice. (1999) 298–372
24. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3d reconstruction. In: Proc. 11th International Conference on Computer Vision, Rio de Janeiro, Brazil. (2007) 1–8