Correcting Vehicle Heading in Visual Odometry by Using Image Appearance

Davide Scaramuzza and Roland Siegwart

Swiss Federal Institute of Technology Zurich (ETHZ)
Autonomous System Laboratory (ASL)
CH-8092, Zurich, Switzerland
{davide.scaramuzza,r.siegwart}@ieee.org
http://www.asl.ethz.ch

Abstract. This paper describes a visual odometry algorithm for on-road vehicles under planar motion assumption. The algorithm uses as only input images taken by a single omnidirectional camera mounted on the roof of the vehicle. To achieve accurate motion estimation results, we decouple the rotation and translation estimation. In particular, we use a feature based ground plane tracker to estimate the translation and an appearance based method to estimate the rotation. As we show, the use of image appearance to estimate the vehicle heading outperforms the pure feature based approach. This pose estimation method has been successfully applied to videos from an automotive platform. We give an example of camera trajectory estimated purely from omnidirectional images over a distance of 400 meters.

Key words: omnidirectional camera, visual odometry, vehicle ego-motion estimation, homography, SIFT features

1 Introduction

Accurate estimation of the ego-motion of a vehicle relative to the road is a key component for autonomous driving and computer vision based driving assistance. Most of the work in estimating robot motion has been produced using stereo cameras and can be traced back to Moravec's work [1]. Similar work has been reported also elsewhere (see [2, 3]). Furthermore, stereo visual odometry has also been successfully used on Mars by the NASA rovers since early 2004 [4]. Nevertheless, visual odometry methods for outdoor applications have been also produced, which use a single camera alone. Very successful results have been obtained over long distances using either perspective or omnidirectional cameras (see [3,5]). In [3], the authors deal with the case of a stereo camera but they also provide a monocular solution implementing a structure from motion algorithm that takes advantage of the 5-point algorithm and RANSAC robust estimation [16]. In [5], the authors provide two approaches for monocular visual odometry based on omnidirectional imagery. In the first approach, they use optical flow computation while in the second one structure from motion.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 285-296

The benefit of camera trajectory determination using omnidirectional cameras was firstly demonstrated in [6] and further recognized in [7]. However, in those works structure and motion estimation were performed only over short distances (up to a few meters) while in this paper we concentrate on accurate trajectory recovery over long distances.

In our approach, we use a single calibrated omnidirectional camera mounted on the roof of the car (Fig. 4). We assume that the vehicle undergoes a purely two-dimensional motion over a predominant flat ground. Furthermore, because we want to perform visual odometry in city streets, flat terrains, as in well as in motorways where buildings or 3D structure are not always present, we estimate the motion of the vehicle by tracking the ground plane.

Ground plane tracking has been already exploited by the robotics community for indoor visual navigation and most works have been produced using standard perspective cameras ([8–11]). In those works, the motion of the vehicle is estimated by using the property that the projection of the ground plane into two different camera views is related by a homography.

In this paper, we propose a similar approach for central omnidirectional cameras but our goal is to estimate the ego-motion of the vehicle in outdoor environments and over long distances. Thanks to the large field of view of the panoramic camera, SIFT keypoints [12] from the scene all around the car are extracted and matched between consecutive frames. After RANSAC based outlier removal [16], we use these features only to compute the translation in the heading direction. Conversely, to estimate the rotation angle of the vehicle we use an appearance based method. We show that by adding this second approach the result outperforms the pure feature based approach.

This paper is organized as follows. Section 2 describes our homography based ground plane navigation. Section 3 describes the appearance based method. Section 4 details the steps of the whole visual odometry algorithm. Finally, Section 5 is dedicated to the experimental results.

2 Homography Based Ground Plane Navigation

The motion information that can be extracted by tracking 2D features is central to our vehicle navigation system. Therefore, we briefly review here a method that uses planar constraints and point tracking to compute the motion parameters.

2.1 Homography and Planar Motion Parameters

Early work on exploiting coplanar relations has been presented by Tsai and Huang [13], Longuet-Higgins [14], and Faugeras and Lustman [15]. The coplanar relation between two different views of the same plane can be summarized as follows.

$$\lambda \mathbf{x_2} = \mathbf{K}(\mathbf{R} + \frac{\mathbf{T}\mathbf{n}^\mathbf{T}}{h})\mathbf{K}^{-\mathbf{1}}\mathbf{x_1} = \mathbf{H}\mathbf{x_1} \qquad (1)$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^{\mathbf{3}}$ are the rotation and the translation matrices encoding the relative position of the two views; $\mathbf{n} \in \mathbb{R}^{\mathbf{3}}$ is the plane normal and $h \in \mathbb{R}$ is the distance to the plane; $\mathbf{x_1}$, $\mathbf{x_2}$ are the images of the same scene points expressed in homogeneous coordinates ($[x, y, 1]^T$); $\mathbf{K}$ is a 3×3 matrix describing the camera intrinsic parameters; λ is a scalar; $\mathbf{H}$ is a 3×3 matrix called homography that relates the two camera projections of the same plane points. Note that matrix $\mathbf{K}$ in equation (1) is defined only for perspective cameras. However, in this paper we assume that our omnidirectional camera is already intrinsically calibrated and that the image points $\mathbf{x_1}$ and $\mathbf{x_2}$ are already normalized to have the third component equal to 1. This allows us to write $\mathbf{K} = \mathbf{I}$. If not stated otherwise, in the remainder of this paper we will assume that the image coordinates are always normalized. To calibrate our omnidirectional camera, we used the method proposed in [17].

In our experiments, we mounted the omnidirectional camera on the roof of the car (as in Fig. 4) with the z-axis of the mirror perpendicular to the ground plane. By fixing the origin of our coordinate system in the center of projection of the omnidirectional camera, we have that $\mathbf{n} = [0, 0, -1]^T$. The distance h of the origin to the ground plane can be manually measured.

2.2 Decomposing H

If a camera is internally calibrated, it is possible to recover $\mathbf{R}$, $\mathbf{T}$, and $\mathbf{n}$ from $\mathbf{H}$ up to at most a two-fold ambiguity. A linear method for decomposing $\mathbf{H}$ was originally developed by Wunderlich [19] and later reformulated by Triggs [20]. The algorithm of Triggs is based on the singular value decomposition of $\mathbf{H}$. The description of this method as well as its Matlab implementation can be found in [20]. This algorithm outputs two possible solutions for $\mathbf{R}$, $\mathbf{T}$, and $\mathbf{n}$ which are all internally self-consistent. In the general case, some false solutions can be eliminated by sign (visibility) tests or geometric constraints, while in our case we can disambiguate the solutions by choosing the one for which the computed plane normal $\mathbf{n}$ is closer to $[0, 0, -1]^T$. In the remainder of this paper, we will refer to this method as the "Triggs algorithm".

In our implementation, we used the Triggs algorithm but we combined it also with another method that we are now going to describe. Indeed, the Triggs algorithm works in general very well if the image points are spatially uniformly distributed on the camera image (see Fig. 1.a). If the image points are too close to a degenerate configuration or they are spatially distributed within one side of the whole omnidirectional image (Fig. 1.b), then it is better to use the following Euclidean approximation:

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{T}\mathbf{n}^{\mathbf{T}}}{h} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} + \frac{\mathbf{1}}{\mathbf{h}} \begin{bmatrix} t_1 \\ t_2 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}^{\mathbf{T}} \tag{2}$$

Here we describe how to use the Euclidean approximation to derive the rotation and translation parameters. From (2), we have:

$$\begin{cases} x_2 = cx_1 - sy_1 - a \\ y_2 = sx_1 + cy_1 - b \end{cases} \tag{3}$$

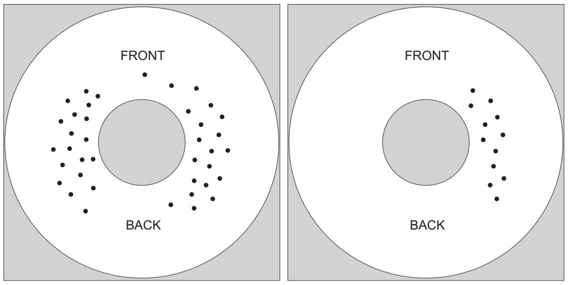
with $c = \cos\theta$, $s = \sin\theta$, $a = t_1/h$, $b = t_2/h$, $\mathbf{x_1} = [x_1, y_1]$, and $\mathbf{x_2} = [x_2, y_2]$. Each point pair gives two equations, and hence given two point pairs we can linearly recover c, s, a, b. When more point correspondences are given (say n corresponding pairs) a linear least-squares solution can be found with the pseudo-inverse matrix method. To this end, observe that (3) can be rewritten as:

$$\begin{bmatrix} x_1 & -y_1 & -1 & 0 \\ y_1 & x_1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} c \\ s \\ a \\ b \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \triangleq \mathbf{A} \cdot \begin{bmatrix} c \\ s \\ a \\ b \end{bmatrix} = \mathbf{B} \tag{4}$$

where $\mathbf{A}$ is a $2n \times 4$ matrix and $\mathbf{B}$ is a $2n \times 1$ vector. The linear least squares solution of (4) is $[c, s, a, b]^T = \mathbf{A}^+\mathbf{B}$, where $\mathbf{A}^+ = (\mathbf{A^T A})^{\mathbf{-1}}\mathbf{A^T}$ is the pseudoinverse of $\mathbf{A}$.

Observe that the matrix $\mathbf{Q} = [c, -s; s, c]$ may not be orthonormal because of the method used to compute its coefficients s and c. However, we can compute an orthonormal matrix that better approximate $\mathbf{Q}$.
The best rotation matrix $\mathbf{R_{2D}}$ to approximate $\mathbf{Q}$ in the Frobenius sense is $\mathbf{R_{2D}} = \mathbf{UV^T}$, where $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = SVD(\mathbf{Q})$ and $SVD(\mathbf{Q})$ is the singular value decomposition of $\mathbf{Q}$.
Here, "best" is in the sense of the smallest Frobenius norm of the difference $\mathbf{R_{2D}} - \mathbf{Q}$, which solves the problem:

$$\min_{\mathbf{R_{2D}}} \|\mathbf{R_{2D}} - \mathbf{Q}\|_F^{\mathbf{2}} \quad \textit{subject to} \quad \mathbf{R_{2D}} \cdot \mathbf{R_{2D}^T} = \mathbf{I} \tag{5}$$



(a) (b)

Fig. 1. (a) Uniform distribution of features. (b) Non uniform distribution.

Finally, from $\mathbf{R_{2D}}$ the rotation angle θ can be easily computed. At the same time, t_1, t_2 can be directly computed from a and b knowing h.

In the remainder of this paper, we will refer to this last method as the "Euclidean method".

In the final implementation of our algorithm, we implemented both the Triggs algorithm and the Euclidean method. The trigger condition to use the one or the other is given by the spatial distributions of the image points. If the image points occupy both the left and the right half of the omnidirectional image (like in Fig. 1.a), then the Triggs algorithm is used. If the image points are only in one half (i.e. either the left or the right one) of the image (like if Fig. 1.b), then the Euclidean method is used.

2.3 Coplanarity Check

So far we have assumed that the corresponding image pairs $\mathbf{x_1}$ and $\mathbf{x_2}$ are correctly matched and that their correspondent scene points are coplanar. Even though in omnidirectional images taken from the roof of the car the ground plane is predominant, there are also many feature points that come from other objects than just the road, like cars, buildings, trees, guardrails, etc. Furthermore, there are also many unavoidable false matches that are more numerous than those usually output by SIFT on standard perspective images (about 20-30% according to [12]) because of the large distortion introduced by the mirror. To discard the outliers, we use the Random Sample Consensus paradigm (RANSAC) [16].

Observe that if a prior estimate of the rotation angle is available (consider for example that coming from the appearance based method of Section 3) then this can be used as a pre-stage in the first step of RANSAC to constrain the selection of putative matches. This has the effect of speeding up the search of inliers and reducing the percentage of outliers. In our implementation in fact, we do this in the following manner: under planar motion assumption the motion can be computed from at least two putative point pairs. If the rotation estimate is consistent (within a certain threshold) with that given by the appearance based method, then the two point pairs are retained (see also Section 4).

3 Visual Compass

In the previous session, we described how to use point features to compute the rotation and translation matrices. Unfortunately, when using 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 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 2 is depicted in Fig. 5.

To improve the accuracy of the rotation estimation, we use an appearance based approach. This approach was inspired by the work of Labrosse [18], which describes a method to use omnidirectional cameras as visual compass.

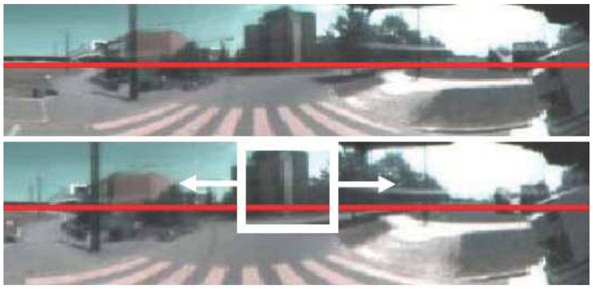


Fig. 2. Two unwrapped omnidirectional images taken at consecutive time stamps. For reasons of space, here only one half of the whole 360 *deg* is shown. The red line indicates the horizon.

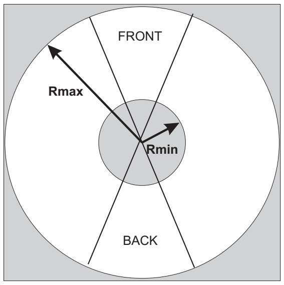


Fig. 3. The cylindrical panorama is obtained by unwrapping the white region.

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. 2). The unwrapping considers only the white region of the omnidirectional image that is depicted in Fig 3. 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:

$$d(I_i, I_j, \alpha) = \sqrt{\sum_{k=1}^{h} \sum_{h=1}^{w} \sum_{l=1}^{c} |I_i(k,h,l) - I_j(k,h-\alpha,l)|^2} \tag{6}$$

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:

$$\varDelta\vartheta = \alpha_m \cdot \frac{360}{w} \tag{7}$$

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 (6) 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. 3). 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. 3). A reduced field of view of about 30 *deg* around the front part is shown by the white window in Fig. 2. 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. 2).

4 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. Here, we describe how we combined the rotation angle estimates of Section 3 with the camera translation estimates of Section 2.

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 in this way:

$$\begin{cases} x_{i+1} = x_i \ + \ \delta\rho_i \cos(\theta_i + \frac{\delta\theta_i}{2}) \\ y_{i+1} = y_i \ + \ \delta\rho_i \sin(\theta_i + \frac{\delta\theta_i}{2}) \\ \theta_{i+1} = \theta_i \ + \ \delta\theta_i \end{cases} \tag{8}$$

where we use $\delta\rho = |\mathbf{T}|\ h$ and $\delta\theta = \varDelta\vartheta$. $|\mathbf{T}|$ is the length of the translation vector; 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 $\varDelta\vartheta$ is computed as in (7). Observe that we did not use at all the rotation estimates provided by the feature based method of Section 2.

Now, let us resume the steps of our motion estimation scheme, which have been detailed in Section 2 and 3. 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. 3).
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 camera model to normalize the feature coordinates by reprojecting them onto a plane perpendicular to the z-axis and distant 1 from the origin.
3. Unwrap the two images and compare them using the appearance based method described in Section 3. In particular, minimize (6), with reduced field of view, to compute the column-wise shift between the appearances and use (7) to compute the rotation angle $\varDelta\vartheta$.
4. Use RANSAC to reject points that are not coplanar (Section 2.3); in particular, use the available rotation estimate $\varDelta\vartheta$ from the visual compass to speed up RANSAC as explained at the end of Section 2.3.
5. Apply the algorithm described in Section 2.2 to estimate $\mathbf{R}$ and $\mathbf{T}$ from the remaining inliers. In doing this, switch between the Triggs algorithm and the Euclidean method as described in Section 2.2.
6. Use $\delta\rho = |\mathbf{T}|\ h$ and $\delta\theta = \varDelta\vartheta$ and integrate the motion using (8).
7. Repeat from step 1.

5 Results

The approach proposed in this paper has been successfully tested on a real vehicle equipped with a central omnidirectional camera. A picture of our vehicle (a Smart) is shown in Fig 4.

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.

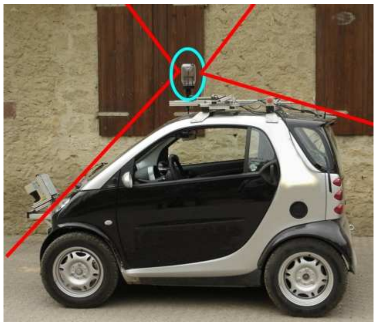


Fig. 4. Our vehicle with the omnidirectional camera (blue circle). The field of view is indicated by the red lines.

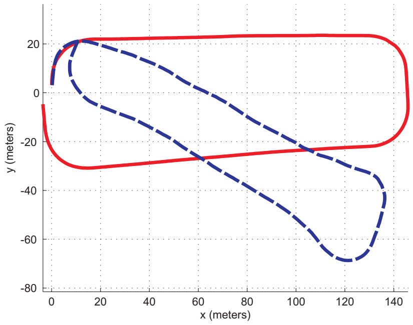


Fig. 5. Comparison between the standard feature based approach (dashed blue) and the approach combining features with visual compass proposed in this paper (red).

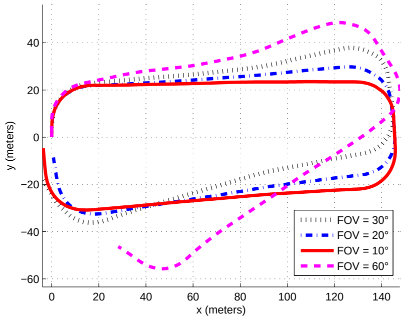


Fig. 6. A comparison of camera trajectory recovered by using different FOVs.

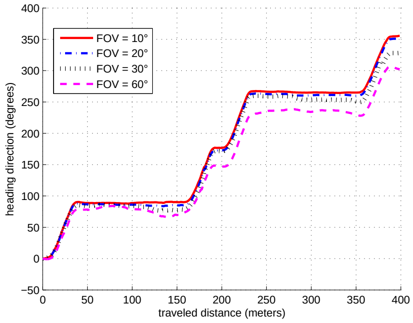


Fig. 7. The heading direction θ (degrees) vs. the traveled distance (meters). The results are shown for different FOVs.

The resulting path estimated by our visual odometry algorithm using a horizontal reduced FOV of 10 *deg* is shown in figures 5, 8, and 9. Our ground truth is a aerial image of the same test environment provided by Google Earth (Fig. 8). The units used in the three figures are in meters.

In this experiment, the vehicle was driven along a 400-meter loop and returned to its starting position (pointed to by the yellow arrow in Fig. 8). The estimated path is indicated with red dots in Fig. 8 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 this

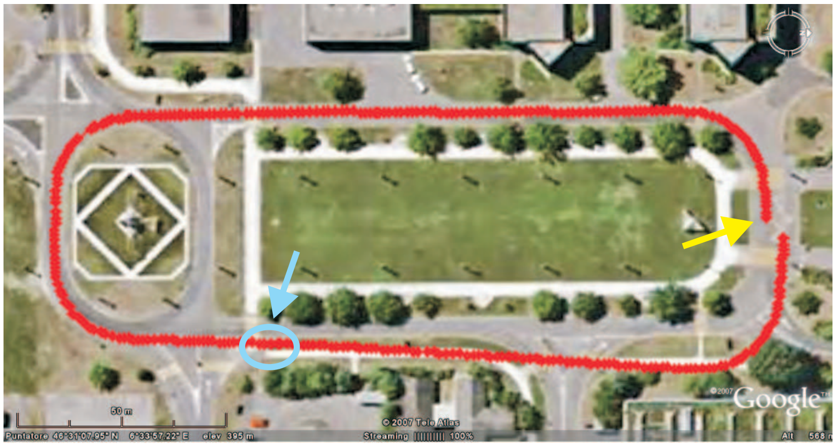


Fig. 8. The estimated path superimposed onto a Google Earth image of the test environment. The scale is shown at the lower left corner.

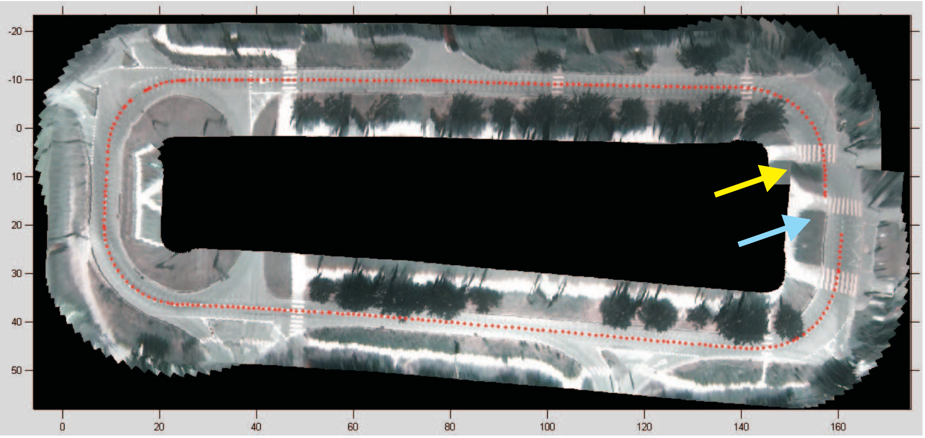


Fig. 9. Image mosaicing that shows a textured 2D reconstruction of the estimated path. The two arrows point out the final error at the loop closure (the pedestrian crossing pointed to by the cyan arrow should theoretically coincide with that pointed to by the yellow arrow).

deviation was due to three 0.3 meter tall road humps (pointed to by the cyan arrow in Fig. 8) that violate the planar motion assumption.

The content of Fig. 9 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 you visually and carefully compare the mosaic (Fig. 9) with the corresponding aerial image (Fig. 8), you 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, you can verify that the location of these elements in the mosaic fits well the location of the same elements in the aerial image.

We also evaluated the effect of the reduced horizontal FOV on the final motion estimation. Fig. 6 shows the recovered estimated trajectory respectively using FOV=10 *deg*, FOV=20 *deg*, FOV=30 *deg*, and FOV=60 *deg*. Observe that the estimation of the trajectory improves as the FOV decreases. Indeed, as we mentioned already in Section 3, the fact of reducing the field of view allows us to reduce most of the problems introduced by the translation, like sudden changes in parallax. The best performance in terms of closeness to the ground truth of Fig. 8 is obtained when FOV=10 *deg*.

In Fig. 7, the effect of the FOV on the estimation of the heading direction is also shown. Also from here, it is clear that the best performance is obtained when FOV=10 *deg*. In this case in fact 90-degree turns are very well estimated. Furthermore, when FOV=10 *deg* the heading direction stays quite constant after each turn, that is when the vehicle covers a straight path. Note that when the vehicle returns to its start position, the estimated heading direction is equal to 355 *deg*, that means the orientation error at the loop closure is 5 *deg*.

6 Conclusion

In this paper, we described an algorithm for computing the ego-motion of an on-road vehicle under planar motion assumption. The algorithm uses as only input images provided by a single omnidirectional camera. The front ends of the system are two different methods. The first one is a pure feature based method that is used to estimate the translation. The second one is an appearance based approach which gives high resolution estimates of the rotation angle of the vehicle. The use of image appearance to compute the vehicle heading proved to give very good estimates against the pure feature based method.

The proposed algorithm was successfully applied to videos from an automotive platform. We gave an example of camera trajectory estimated purely from omnidirectional images over a distance of 400 meters. For performance evaluation, the estimated path was superimposed onto a aerial image of the same test environment and a textured mosaic of the path was built.

References

1. H. Moravec, *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.* PhD thesis, Stanford University, 1980.

2. I. Jung and S. Lacroix, “Simultaneous localization and mapping with stereovision,” in *Robotics Research: the 11th International Symposium*, 2005.
3. D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications,” *Journal of Field Robotics*, 2006.
4. M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers: Field reports,” *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.
5. P. I. Corke, D. Strelow, and S. Singh, “Omnidirectional visual odometry for a planetary rover,” in *IROS*, 2004.
6. T. Svoboda and T. Pajdla and V. Hlavac, Motion Estimation Using Central Panoramic Cameras, IEEE Int. Conf. on Intelligent Vehicles, 1998.
7. Chang, P. and Hebert, M., Omni-Directional Structure from Motion, IEEE Workshop on Omnidirectional Vision, 2000.
8. H. Wang, K. Yuan, W. Zou, and Q. Zhou, “Visual odometry based on locally planar ground assumption,” in *IEEE International Conference on Information Acquisition*, pp. 59–64, 2005.
9. Q. Ke and T. Kanade, “Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection,” in *CVPR 2003*, jun 2003.
10. J. J. Guerrero, R. Martinez-Cantin, and C. Sagues, “Visual map-less navigation based on homographies,” *Journal of Robotic Systems*, vol. 22, no. 10, pp. 569–581, 2005.
11. B. Liang and N. Pears, “Visual navigation using planar homographies,” in *IEEE ICRA*, pp. 205–210, 2002.
12. D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 20, pp. 91–110, 2003.
13. R. Tsai and T. Huang, “Estimating three-dimensional motion parameters of a rigid planar patch,” *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 29, no. 6, pp. 1147–1152, 1981.
14. H. C. Longuet-Higgins, “The reconstruction of a plane surface from two perspective projections,” in *Royal Society London*, vol. 277, pp. 399–410, 1986.
15. O. D. Faugeras and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” *International Journal of Pattern Recognition and Artificial Inteligence*, no. 3, pp. 485–508, 1988.
16. M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
17. D. Scaramuzza, A. Martinelli, and R. Siegwart, “A flexible technique for accurate omnidirectional camera calibration and structure from motion,” in *ICVS*, jan 2006.
18. F. Labrosse, “The visual compass: performance and limitations of an appearance-based method,” *Journal of Field Robotics*, vol. 23, no. 10, pp. 913–941, 2006.
19. W. Wunderlich, “Rechnerische Rekonstruktion eines ebenen Objekts aus zwei Photographien,” *Mitteilungen der geodaetischen Institute, TU Graz*, vol. 40, pp. 365-377, 1982.
20. B. Triggs, “Autocalibration from planar scenes,” in *ECCV98*, pp. 89-105, 1998.