An Evaluation of Image-Based Robot Orientation Estimation

Cao, Juan; Labrosse, Frédéric; Dee, Hannah Mary

Published in:
Towards Autonomous Robotic Systems

DOI:
10.1007/978-3-662-43645-5_15

Publication date:
2013

Citation for published version (APA):
Cao, J., Labrosse, F., & Dee, H. M. (2013). An Evaluation of Image-Based Robot Orientation Estimation. In Towards Autonomous Robotic Systems (pp. 135-147). (Lecture Notes in Artificial Intelligence). Springer Nature. https://doi.org/10.1007/978-3-662-43645-5_15

An evaluation of image-based robot orientation estimation

Juan Cao, Frédéric Labrosse, Hannah Dee

Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK
{juc3,ffl,hmd1}@aber.ac.uk
http://www.aber.ac.uk/en/cs/

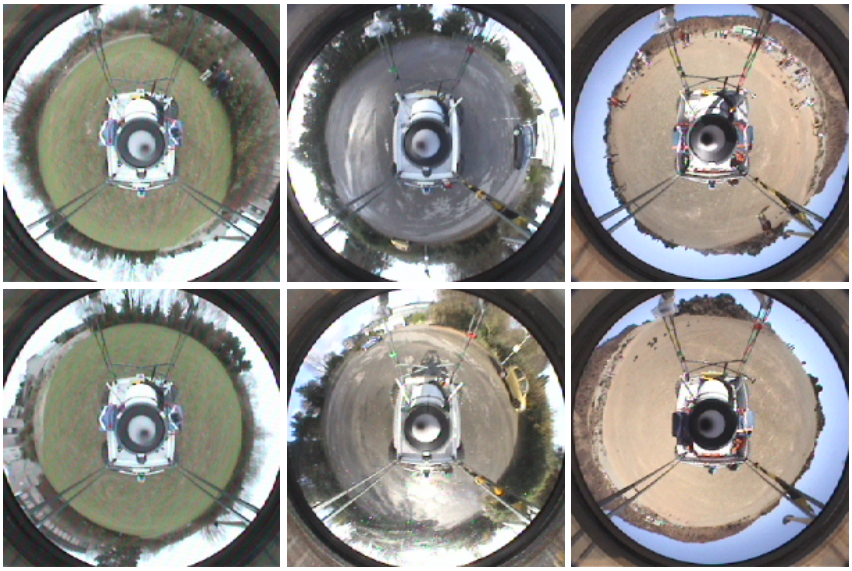
Abstract. This paper describes a novel image based method for robot orientation estimation based on a single omnidirectional camera. The estimation of orientation is computed by finding the best pixel-wise match between images as a function of the rotation of the second image. This is done either using the first image as the reference image or with a moving reference image. Three datasets were collected in different scenarios along a "Gummy Bear" path in outdoor environment. This carefully designed path has the appearance of a gummy bear in profile, and provides many curves and sets of image pairs that are challenging for visual robot localisation. We compare our method to a feature based method using SIFT and another appearance based visual compass. Experimental results demonstrate that the appearance based methods perform well and more consistently than the feature based method, especially when the compared images were grabbed at positions far apart.

Keywords: Robot orientation, Quadtree, SIFT, Visual compass

1 Introduction

When mobile robots move, one of the basic problems which needs to be solved is for the robot to know its orientation as accurately as possible. A range of sensors, such as laser, sonar, digital compass, wheel encoders, gyroscope and GPS can be used to perform this measurement. However, since digital cameras have become more affordable, more research has been devoted to supplement traditional systems using visual cues.

Various solutions to the problem have been proposed. We can categorize these solutions into two main groups: Feature-based and appearance-based. Feature-based methods try to detect distinctive and robust points or regions between consecutive images, while appearance-based methods concentrate their efforts on the information extracted from the pixel intensity, the whole image being represented by a single descriptor, without local feature extraction. The change in orientation between frames is then computed by aligning the features or images, using a calibration of the projection onto the image plane. This process can be achieved incrementally from frame to frame as the robot moves, which can have the drawback that it generally becomes less and less accurate as integration introduces additive errors at each step. Alternatively, a fixed reference image can



(a) (b) (c)

Fig. 1. Example images of the (a) FIELD, (b) CARPARK, and (c) TENERIFE datasets

be used and all orientations calculated from it. However, as the frames become more different, the orientation estimation becomes less reliable.

In this paper we aim to address the question "What image based techniques are best for orientation estimation?". We do this by comparing appearance based methods such as the visual compass [5], image-based techniques which use quadtree methods to reduce noise [1] and feature matching techniques such as SIFT (Scale-invariant feature transform) [6]. In order to thoroughly compare these methods, we measure their performance on a number of datasets captured in outdoor dynamic environments: FIELD, CARPARK, and TENERIFE. Some images from these datasets are shown in in Fig. 1 while details can be found in Section 4.1. These datasets consist of sequences of images captured along a "Gummy Bear" path (see Fig. 2) by our 4 wheel drive, 4 wheel steering, electric vehicle *Idris*. This carefully designed path has the appearance of a gummy bear in profile, and provides many curves and sets of image pairs that are challenging for visual robot localisation (for example the ear region contains a sequence of images on a tight curve).

The rest of this paper is organised as follows: Section 2 reviews related work. Section 3 outlines the three different methods that we compare. Section 4 details the experiments undertaken, and reports results. Finally, Section 5 concludes the paper and outlines possible future improvements.

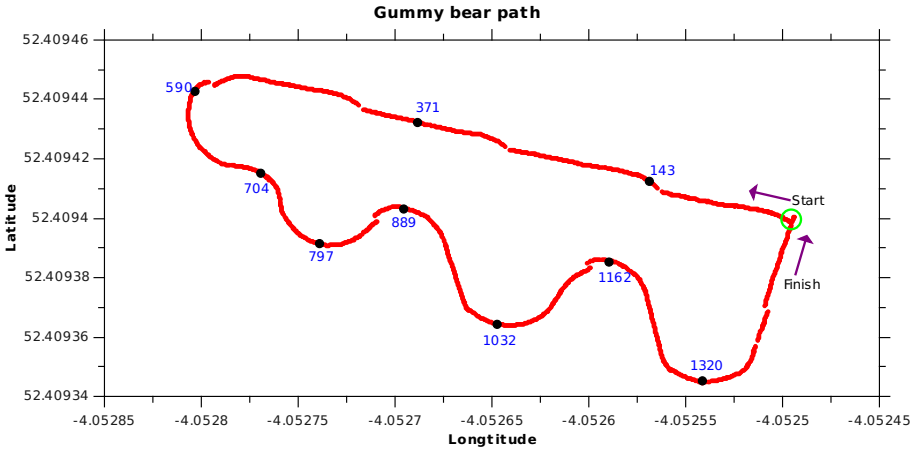


Fig. 2. RTK-GPS track from the "Gummy Bear" path for the FIELD dataset. Image numbers of key positions are marked in blue. This path is designed so that there are pinch points (at the "neck" and "knees") where the robot is quite close to where it has been before, but is clearly not in the same place (e.g. images 143 and 1162 might be expected to be similar). The path finishes at the start point but with Idris rotated through 90°.

2 Related work

Vision based motion estimation has a long history in robotics. Methods have been proposed using both monocular cameras [14,11] and stereo cameras [10,8,12,11,7]. Related work can be divided into two categories: feature-based and image-based methods. Here, we review some of these works, more extensive surveys can be found in [13].

The earliest work on estimating a vehicle's motion from visual imagery alone is [10], where the basic algorithm identifies corner features in each camera frame and estimates the depth of each feature using stereo. Then potential matches are found by normalized cross correlation. Finally, motion is computed by estimating the rigid body transformation that best aligns the features at two consecutive robot positions. However, this kind of system suffers from poor accuracy and is unstable, partly because it relies on scalar models of measurement error in triangulation. Based upon this work, [8] uses 3D Gaussian distributions to model triangulation error and incorporates the error covariance matrix of the triangulated features into the motion estimation between successive stereo pairs. The motion estimation in this work was pure translation, without considering orientation. This is extended in [12] by incorporating an absolute orientation sensor such as a compass, a sun sensor or a panoramic camera providing periodic orientation updates, with the Förstner corner detector used as feature detector. The results indicated that the error growth can be reduced to a linear function of the distance travelled and outperform previous visual odometry results. A Harris corner detector was used in [11].

All the works reviewed above are feature-based methods; this kind of method tries to detect distinctive points or regions between consecutive image pairs. Although feature extraction can be fast, it often requires assumptions about the type of features being extracted and natural environments can sometimes present no obvious visual landmarks, such as desert or plain regions.

Some successful works using the whole appearance of the images have been proposed in the literature, e.g. [4,9,2,3]. In [4] a Fourier-Mellin transform was applied to omnidirectional images in order to obtain a visual descriptor for vehicle's motion estimation. The vehicle's motion was decomposed into a rotation and a translation component. The rotation angle estimate is taken as the median of the observed angular displacements using a mapping from camera coordinates to the ground plane. In the same manner, the low frequency components of Fourier coefficients are used in [2] to represent each panoramic image captured. When the Fourier signature has been captured in two nearby points, the relative orientation of two points will be computed using the shift theorem. An other example is [9] where the colour images captured from a perspective camera are first converted to grey images, then each pixel column is summed and normalized to form a one-dimensional array. The resulting arrays were used to extract the rotation information.

More recently, both appearance-based and feature-based methods were presented in [3] to compute the motion transformation between two consecutive images incrementally. The phase information of the Fourier signature was used to compute the robot orientation, and SURF features were used to detect the interest points for image comparisons by means of looking for corresponding points.

3 Computing Robot Orientation

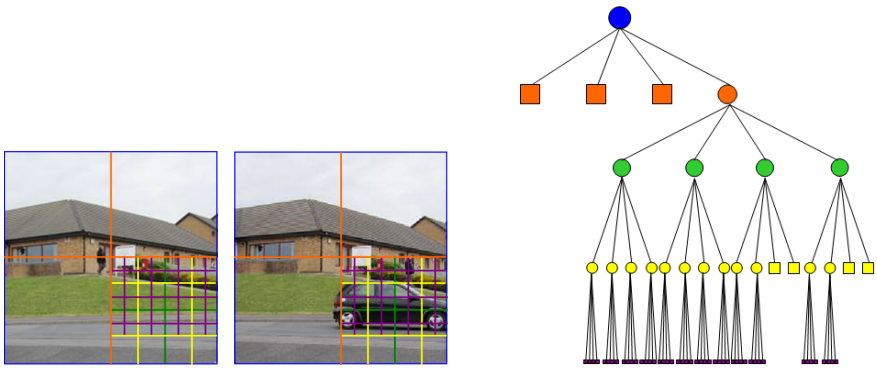
For the purpose of this paper, we compare the performance of our proposed method with a feature based method and the visual compass presented in [5] using real images in different environments.

3.1 Feature Based Method: SIFT

In this method, SIFT features [6] are used to align panoramic images. Features are extracted from the two frames that need to be aligned and they are matched using a distance ratio of 0.6 (two features are matched if their distance in feature space is less than 0.6 times the distance of second closest feature). Once features have been matched, the two images can be aligned by computing the average horizontal displacement of the features, taking care to wrapp around positions in the panoramic images. In order to obtain a reliable solution in the presence of outliers, we use a Gaussian distribution to model the error and cut off the matches that are one standard deviation away from the mean.

3.2 Visual Compass

In the visual compass proposed in [5], the relative rotation between pairs of panoramic images is obtained by finding the best match between images based



(a) Recursively decomposed image pair using quadtree

(b) Corresponding quadtree representation of image pair comparison

Fig. 3. Quadtree decomposition. Note that there has been camera motion between the two images but the only difference detected is the car, showing robustness to small changes.

on their column shift and using the Euclidean distance in image space. The orientation estimation is done from a moving reference image, the decision on when to change it being made using a measure of difference between images. This offers a compromise between accumulating error and comparing similar images to get a better estimation of the change in orientation. Because the sides of the panoramic images contain rotational and translational information, only the parts of the images that correspond to the front and back of the moving robot are used in the matching process.

3.3 Quadtree Method

The core technique we use for image pair comparison is quadtree decomposition combined with a number of standard image distance measures. This technique enables robustness of perceptual aliasing (the images we tested are mostly made of repetitive features) and can cope with the appearance of new objects in the robot environment without prior information. Fig. 3(a) is a visualisation of recursive image comparison and Fig. 3(b) the corresponding tree-based visualisation. Fig. 3(a) shows that the decomposition into sub regions provides us not only with robustness to noise, but also with an indication of the locations of visual change between image pairs. In Fig. 3(b), the root of the tree corresponds to the comparison of the two original images. Circles represent internal nodes of the tree, and leaf nodes correspond to quadrants that are either similar or too small.

Our method recursively compares quadrant of the two images to be compared using a metric (discussed below) until either the two quadrants are judged similar (distance below a threshold) or too small. When two quadrants are not similar enough they are split into four quadrants and the process is repeated. This is described in Algorithm 1. The similarity measure of the two images is then

```
// Base case
begin
    I_n ← I_new;
    I_r ← I_ref;
    // Calculate the distance between I_n and I_r
    Dist=distance(I_n, I_r);
    if Dist > THRESHOLD then
        BuildQuadTree(rootNode);
    else
        Quadtree building stopped;
    end
end
// Quadtree building
BuildQuadTree(Node *n)
begin
    if n → dist > THRESHOLD and n → size > MIN then
        // Break image or patch into 4 patches
        for n=0 to 3 do
            nodeIn=BuildNode(n → child[i], n, i);
            BuildQuadTree(nodeIn));
        end
    end
end
```

Algorithm 1: Pseudocode representation of the image comparison algorithm using quadtree

given as the area of the two images that are similar. The value of the threshold largely influences the similarity between images. We experimentally determined thresholds based on the metric and the colour space used.

To estimate the change of orientation between two images, one image is column-wise shifted (the shift corresponding to a rotation of the camera) and the similarity computed for each rotation. The maximum similarity gives the rotation between the two images.

Image metrics are used to quantitatively evaluate the similarity or dissimilarity between two images or image regions. There are a range of metrics; we have tried Euclidean distance, χ^2 distance and Pearson's correlation coefficient. Here we discuss details of implementation, including the choice of image distance metric to use within the quadtree algorithm, the choice of the threshold and the smallest region size.

To determine which distance metric is appropriate for our application, we compute the similarity between one image at the middle of a sequence of images and all other images of the same sequence, using our quadtree algorithm with the three metrics mentioned. The sequence of images was captured in our laboratory using the panoramic camera mounted on a robot moving on a straight line, an image being captured every 10 cm. We seek a similarity measure that is a) smooth and b) not too "steep" around the reference image. Fig. 4 shows the similarity

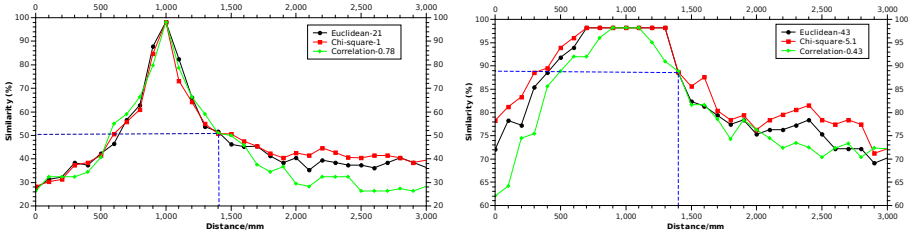


Fig. 4. Comparison of different metrics applied to our quadtree similarity measurement; The left plot has a threshold set low, the right high, iso-similarity point occurs approximately 1,400mm on the x-axis.

values for the three metrics and different values of the threshold used in the quadtree construction.

In order to produce a fair comparison between the three different measures (with thresholds on different scales) we define an iso-similarity point for each test; this sets the threshold for quadtree decomposition so that the three metrics produce identical similarity measures. It can be seen that for low thresholds, our quadtree measure is sensitive to small displacements but that with higher thresholds we are able to determine similarity between images on a broader scale.

Briefly summarising our tests, we can see that Euclidean distance, χ^2 distance and Pearson's Correlation Coefficient behave in much the same way when we find a threshold that defines an iso-similarity point. Between the two results given here, the similarity of the iso-similarity point increases from 50% to 89%. Pearson's correlation coefficient seems to be the most sensitive to small displacements (has a narrower peak) and it is also the most computationally intensive metric we have considered. Euclidean distance and χ^2 distance are both fast and easy to compute, with the comparison results showing little difference between them. Thus, for the rest of this paper, we will present results on Euclidean distance only due to simplicity.

In our implementation, the smallest region size is set to 10×10 pixels, which represents a horizontal field of view of 10°. Note that the angular resolution of the images in our experiments is 1° per pixel. Such a field of view implies that we can see objects 35 cm wide 2 m away. For the quadtree decomposition this value is a good compromise between reducing computational load and distinguishing between objects in the robot's environment.

4 Experiments

4.1 Experimental setup

Test images were captured from an omni-directional camera approximately one and a half meters above the ground surface as the robot moves. The CIE $L^*a^*b^*$ colour space is used in our experiments. The FIELD dataset was collected in a field-type area, with some buildings in sight but mostly trees and grass. The CARPARK dataset was captured from a carpark with trees around, where few cars

Table 1. Characteristics of datasets

Dataset	Frames	Length	Rate	Notes
FIELD	1525	60m	6Hz	Flat but rough surface, can see about 50 m
CARPARK	2101	60m	8Hz	Flat, can see 30 m, light changes, moving objects
TENERIFE	2156	60m	8Hz	Bumpy, can see 100 m, moving objects

Table 2. Root Mean Square Error, Mean Error and Standard Deviation Error for each dataset (F: FIELD; C: CARPARK; T: TENERIFE; VC: visual compass, subscript indicates fixed reference or moving with the corresponding skip value).

	RMSE			Mean			SD		
Method	F	C	T	F	C	T	F	C	T
VC	**11.42**	19.51	41.18	**4.63**	-1.33	-24.47	**10.43**	19.47	33.12
SIFT_f	24.15	40.67	**21.83**	10.61	-3.57	**-2.89**	21.70	40.52	21.63
SIFT_5	53.06	68.00	93.48	-35.28	-48.82	-69.55	39.68	47.38	62.52
SIFT_20	106.02	108.39	107.35	-17.01	48.63	-5.36	105.30	96.87	107.69
QT_f	29.93	**16.62**	23.88	16.00	**-0.76**	-11.59	25.30	**16.60**	**20.88**
QT_5	62.79	81.43	79.29	-38.42	-55.79	9.70	49.73	59.32	78.76
QT_20	12.51	24.06	24.43	5.90	5.66	-10.00	11.10	23.38	22.39

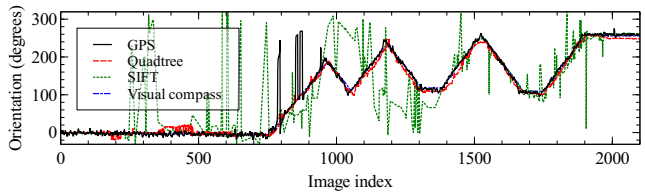
were parked and one moved and some parts of ground were wet from rain providing challenging reflections and shadows. The TENERIFE dataset was obtained at the El Teide National Park, Tenerife. Its flat landscape with fine textures of volcanic sand, pebbles and occasional rocky outcrops are similar to those encountered on the surface of Mars. Some tourists were walking around during the data acquisition. The characteristics of the datasets are described in Table 1.

The performance of each method is evaluated using a real-time kinematic (RTK) GPS system, which is theoretically accurate up to 1 cm in an horizontal plane. However, the accuracy and reliability to be achieved depend on a few factors, such as satellite availability, baseline length and sufficient redundancy of GPS observations. To allow comparison, absolute GPS heading is filtered using a Kalman filter and converted to relative bearing by subtracting the absolute heading of the starting point of "Gummy Bear" trajectory, and changed to a range between 0 and 360. This is used as ground truth.

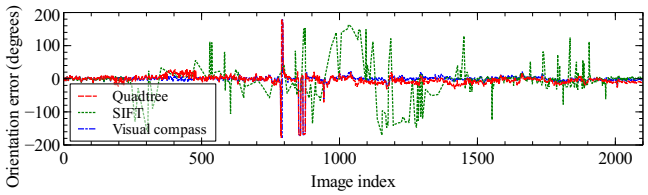
4.2 Experimental Results

For both quadtree and feature based methods we estimate the orientation in two ways. The first uses the first image as a reference from which the orientation is calculated. The second uses a moving reference image and accumulates changes in orientation. In the second case we present results skipping a fixed number of images between pairs. The visual compass method from [5] uses a moving reference with automatically adjusted skips. It is therefore compared to the methods using a fixed reference image. Table 2 gives quantitative results for all cases.

Figures 5 to 7 show the results for the three methods with a fixed reference image for quadtree and SIFT (the visual compass uses a moving reference image but this is all internal to the method and not exposed). These show that both appearance-based methods perform well and consistently for the whole path of

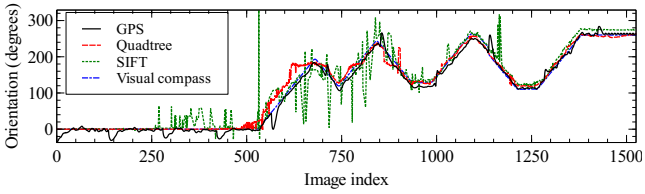


(a) Orientation estimation and groundtruth

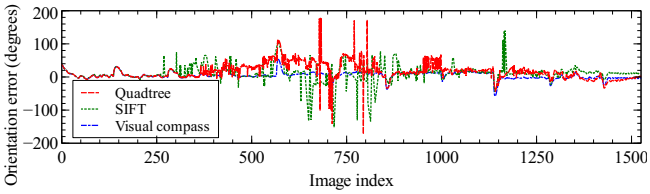


(b) Orientation error from groundtruth

Fig. 5. Experimental results for dataset CARPARK with a fixed reference image



(a) Orientation estimation and groundtruth

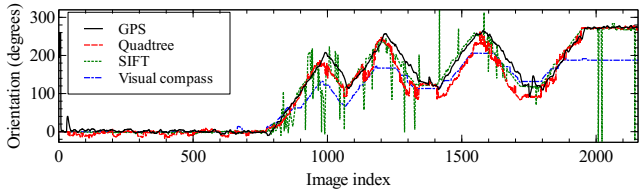


(b) Orientation error from groundtruth

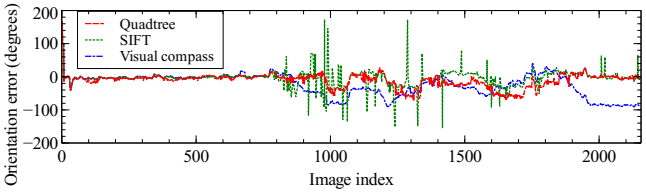
Fig. 6. Experimental results for dataset FIELD with a fixed reference image

the robot. The feature-based method performs well when the images are close to the reference image but poorly when away with many frames where no features were found to match. This is due to a lack of matched SIFT features due to the distortions introduced by the camera. Indeed, the orientation could not be calculated using the SIFT method for many frames: 36% for CARPARK, 62% for FIELD and 67% for TENERIFE

The Sift method performs better than both appearance-based methods on the TENERIFE dataset. This is because the boundary between sky and land is

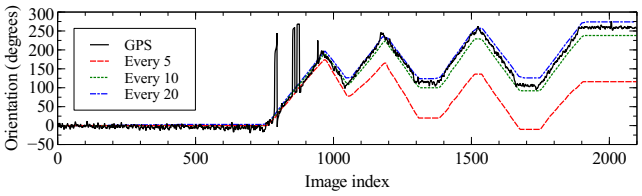


(a) Orientation estimation and groundtruth

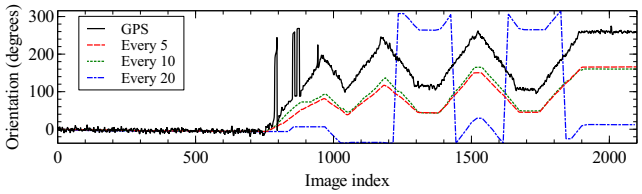


(b) Orientation error from groundtruth

Fig. 7. Experimental results for dataset TENERIFE with a fixed reference image



(a) Orientation from the quadtree method

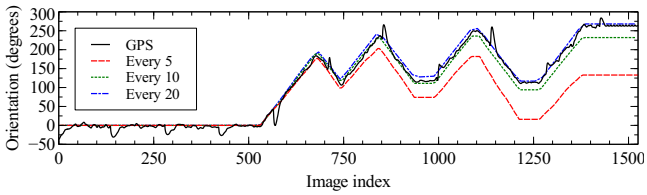


(b) Orientation from the SIFT method

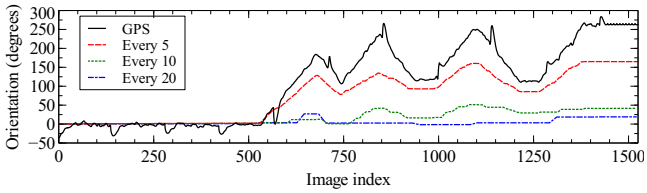
Fig. 8. Experimental results for dataset CARPARK with a moving reference image

very strong, not visible all around the robot and slanted. Alignment of the images using pixel values will therefore tend to align the skyline introducing a bias due to the slant, in particular in the narrow field of view of the visual compass.

Figures 8 to 10 show the results for the incremental quadtree and SIFT methods that use a moving reference. For both methods, pairs were created skipping a fixed number of images and results are given for different values of the number of images skipped. These clearly show that choosing the correct compromise between better short term rotation estimation and often accumulating error is

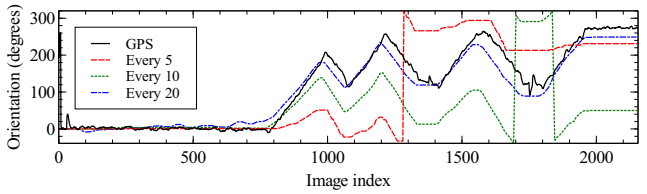


(a) Orientation from the quadtree method

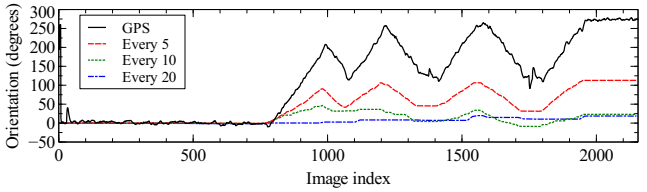


(b) Orientation from the SIFT method

Fig. 9. Experimental results for dataset FIELD with a moving reference image



(a) Orientation from the quadtree method



(b) Orientation from the SIFT method

Fig. 10. Experimental results for dataset TENERIFE with a moving reference image

critical. In fact, none of these results are as good as that of the visual compass. This is due to the subpixel processing and the automatic, adaptive estimation of the best compromise done in the visual compass. Nevertheless, the quadtree method performs similarly to the SIFT method but skipping more images. This is in line with the fact that the SIFT method performs better when the reference image is not too different from the processed images.

5 Conclusions

In this paper we have evaluated three methods for robot orientation estimation with panoramic images in order to investigate what image based techniques are suitable for this task. The results gathered from the three different scenarios in two different ways show that the quadtree method perform better than the SIFT method when the distance between pairs of images becomes high, while the SIFT based method does well over short distances. This implies that the appearance-based method is likely to work better at lower frame rates and be more appropriate for loop closure tasks, at least for orientation estimation. Moreover, the appearance-based methods (quadtree and visual compass) perform better than the feature-based method when the environment is visually variable but not too contrasty. In future work, we will use the quadtree method for localisation tasks where its handling of partially similar images will be of importance.

References

1. J. Cao, F. Labrosse, and H. Dee. A novel image similarity measure for place recognition in visual robotic navigation. In *TAROS*, pages 414–415, 2012.
2. L. Fernández, L. Payá, Ó. Reinoso, and F. Amorós. Appearance-based visual odometry with omnidirectional images - a practical application to topological mapping. In *ICINCO (2)*, pages 205–210, 2011.
3. D.V. García, L.F. Rojo, A.G.Aparicio, L.P.Castelló, and O.R. García. Visual odometry through appearance- and feature-based method with omnidirectional images. *Journal of Robotics*, pages 1–13, 2012.
4. R. Goecke, A. Asthana, N. Pettersson, and L. Petersson. Visual vehicle egomotion estimation using the fourier-mellin transform. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 450–455, 2007.
5. F. Labrosse. The visual compass: Performance and limitations of an appearance-based method. *Journal of Field Robotics*, 23(10):913–941, 2006.
6. D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
7. M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.
8. L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):239 – 250, June 1987.
9. M.J. Milford and G.F. Wyeth. Single camera vision-only slam on a suburban road network. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3684–3689, 2008.
10. H. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University, 1980.
11. D. Nistr, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23:2006, 2006.
12. C.F. Olson, L.H. Matthies, M. Schoppers, and M.W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, 2003.
13. D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *Robotics Automation Magazine, IEEE*, 18(4):80–92, 2011.
14. C. Tomasi and J. Shi. Direction of heading from image deformations. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 422–427, 1993.