This is an accepted version of the article: Dörfler, M., Přeučil, L., and Kulich, M. (2016). **Vision-Based Pose Recognition, Application for Monocular Robot Navigation**. *Robot 2015: Second Iberian Robotics Conference*, which has been published in a final form at https://doi.org/10.1007/978-3-319-27146-0_35.

Vision-Based Pose Recognition, Application for Monocular Robot Navigation

Martin Dörfler[1], Libor Přeučil[2⋆]and Miroslav Kulich[2⋆]

[1] Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 166 27 Prague 6, Czech Republic
dorflmar@fel.cvut.cz

[2] Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Zikova 1903/4, 166 36 Prague 6, Czech Republic
{preucil, kulich}@ciirc.cvut.cz

Abstract. This paper presents improvements made to previous method for monocular teach-and-repeat navigation of mobile robots. The method is based on recording the position of image features in camera image, and moving the robot so their position matches during the recall. The method has shown good reliability, though requires odometry to perform well. This paper targets improvements of the method by replacement of a simple odometry by visual pose recognition approach. Thus, localization becomes independent of preceding pose computation. This prevents accumulation of error during the run of the algorithm.
A pose recognition method based on angle differences is presented herein. The substitution of odometry implies necessary adjustments to the aforementioned method to be used. Suitability of the method for pose recognition is evaluated experimentally. The method has shown to be feasible for the nav task, although the achieved accuracy is lower than the original method.

Keywords: Pose Recognition · Vision-based · Robust Image Features · Monocular Localization and Navigation

1 Introduction

With increasing computational power making real-time image processing possible, multiple navigation methods based on vision have been investigated in the last two decades.

One of the most popular approaches is monocular SLAM (monoSLAM). A feature-based variant of monoSLAM extracts discrete feature observations from consecutive images and matches them in order to determine full pose of the camera and all the features themselves [5, 17, 9]. Dense monocular SLAM approaches work with raw images, model the environment as a dense surface and align these complete images to determine the camera pose [8].

⋆ This research was supported by the Grant Agency of the Czech Republic (GACR) with the grant no. 15-22731S entitled "Symbolic Regression for Reinforcement Learning in Continuous Spaces" and Technology Agency of the Czech Republic under the project no. TE01020197 "Centre for Applied Cybernetics"

Another stream takes advantage of a pre-learnt map built during a human-guided teleoperated drive. Afterwards, the robot is controlled by a kind of image-based visual servoing in the autonomous navigation mode, i.e. difference between the expected and current image is to be minimized [3].

For example, Bekris et al. [2] present a local control law for a homing strategy exploiting bearings of at least three salient features in a panoramic image of the scene. Determination of the actual control then relies on measuring the angle between pairs of features. Long-range homing is managed by extracting milestone positions on the trajectory with partially overlapping sets of observable features. The milestones are then traversed sequentially.

Another approach was used by Diosi et al. [6]. Path is stored as a sequence of reference images. Localization is performed by computing local geometry between current image and nearest reference frames. Navigation is simpler, relying on comparing expected and actual positions of detected landmarks in the current view and next reference image in sequence.

Similar approach is used SURFnav proposed by Krajník et al. [11]. In contrast to previous method, no attempt to reconstruct local geometry is performed. Learned trajectory is split into a sequence of linear segments. A lateral deviation during forward motion along a particular segment is reduced via comparison of 2D positions of currently observed features with positions of features expected to be visible in the pre-learnt map. A longitudinal position within the segment is determined from odometry.

Using these more relaxed constraints sacrifices localization of the robot, without impacting the ability to navigate the path. In return, it is possible to perform faster computation and use less reference frames, leading to sparser map and thus smaller representation. This makes the method suitable for use on resource-constrained onboard computers of smaller robots.

It has been proven, that the position error for closed polygonal trajectories does not diverge [10] with the following weaknesses:

1. The salient feature existence, their stability and recognition/matching may not be sufficient if displacement of the robot is not known with sufficient precision and confidence.
2. The limited relative dead-reckoning precision, which accumulates robot positioning errors along the trajectory may raise above the limits if the robot trajectory does not exhibit sufficient curvature to keep the odometry error within certain bounds in all degrees of freedom.

Consequently, the improper robot positioning may lead to weaker determination of correspondences between subsequent observations (frames) in feature matching. This brings the method [10] to less precise performance as the trajectory is followed with lower precision. This may even lead to complete failure if the robot looses ability to observe features in the previous frame completely. In this aspect, the robustness of the robot dead-reckoning system stands crucial for stable performance of feature-based visual servoing approaches (*FB-nav*).

Nitsche et al. [15] overcome dead-reckoning precision by employing computationally efficient Monte Carlo localization to estimate the robot position within a segment

relative to the segment start. Nevertheless, their approach still depends on odometry measurements.

The herein introduced approach addresses adjustment of positioning errors of the robot on the fly, using the same salient features only as applied in the *FB-nav* method itself (or a similar one, to limit the computational intensity of the process) without a need for odometry. Primarily, as the relative dead-reckoning error typically grows proportionally over time, or the path driven, a periodic (or even a steady) corrective process shall be elaborated. This keeps the positioning deviation within specific bounds and thus avoids failure of the method.

The paper is structured as follows: The next section explains the principles of the presented method and motivation for our approach. Section 3 deals with the algorithm itself. Necessary changes to *FB-nav* are presented, followed by explanation of the method we replace the dead-reckoning with, and technical details. The last part details the performed experiment and the results.

2 Motivation

The aim of the work is to elaborate a method for robust visual odometer, that may serve as a localization of a mobile robot along its trajectory. The method relies on visual features that are used for robot navigation along the given trajectory through application of [11]. The features used are natural salient properties of the observed scene described by suitable robust image descriptors [14].

As a regular odometry suffers from accumulative errors with non-zero mean, the overall error displacement of the robot grows without limits with the path driven. This featuring may cause limitation on performance of the used navigation method, that determines the robot heading control and requires guidance of the robot based on odometry for a certain portion of time, or path. Erroneous odometry may cause a complete failure of the method. Therefore, a procedure that refines the robot position steadily along the trajectory it drives is foreseen.

Normally, the SURFnav method allows the robot to drive along a linear path segment under assumption that the positioning error on the segment is constrained since position re-calibration can be only accomplished at the next turn — a rapid change of the robot heading along a piecewise linear trajectory.

In the cases, where the path segment is oversized, the dead-reckoning error may rise high and the robot may loose its capability to observe the desired scene features at all, causing the method to fail. Therefore, limiting of the robot displacement error on the fly is desired. Our approach is an extension of the original SURFnav approach and relies on permanent observation of salient features by omni-directional camera along the robot path, providing their observation angles only. Our contribution shows that the obtained directional information can then be processed to compute robot displacement errors and to suggest the necessary correction of its position without a need for any further calibration.

Moreover, we show prospective properties of the method, that appears incrementally stable i.e. it is capable to bring the robot into the correct position (= the previous position, where the robot is expected to be with respect to its preceding occurrence in

this location) if the correction step is primarily finite and under specific limits, imposed by the constrains of the camera, manoeuvering capabilities, etc.

2.1 Problem Specification

Generally the *FB-nav* methods [11, 10], irrespective of the type of image feature being used, are based on extraction of those stable image regions from either omni- or forward looking camera. Knowing the assignment of particular features to a specific observation position (comprising robot heading and coordinates), correspondences of these within subsequent frames can be established and used for robot heading adjustment. In other words, the method, dead-reckoning serves two necessary roles:

1. To determine expected features to be observed from each position on the path.
2. To decide about a path segment end point and which is to be continued by the next path segment. In a typical case, this situation is accompanied by a change in direction.

For the first point, dead-reckoning is not strictly necessary. Expected features can be determined based on segment rather than precise position. Resulting larger feature set increases computational costs of feature matching, but should not hamper the function of the method. Shorter path segments can be used to keep the costs from becoming unbearable.

The key issue is the detection of the segment endings. Relying on the dead-reckoning is very unsafe in real cases. Therefore, replacing the end-of-segment detection by visual means allows to omit the dead-reckoning process completely and decouples the results on a segment from previous path steps and accumulative errors along the direction of the movement.

3 Method Description

3.1 Approach

To determine whether the end of a segment has been reached or passed, the method based on angle differences is used. In a way similar to the original method, the first step is to employ robust features in the camera image to detect landmarks in the scene. As the feature detection has already been performed by the *FB-nav* method in each step, this data can be reused. For this step, angles between the landmark directions are exploited instead of the directions themselves.

The core idea of the method is based on a simple geometric intuition: when observing a pair of landmarks, the difference of observation angles between them appears larger from closer distance and smaller, if they are more far away. Therefore by comparing views to the same landmarks from two diverse positions, it can be determined which one is located closer to the pair of landmarks. Considering a convex region from which the observations are done, while preserving their ordering in the view, the afore rule can be applied: if starting at a distant position and moving towards a landmark pair, the distance will decrease and the observations will turn more similar to a closer view case.

Expanding this idea for multiple landmarks, it becomes possible to state whether the motion direction is forwards or backwards with respect to each landmark pair. Depending on the use case, this piecewise information can be fused to estimate direction to the other position, or to perform voting to ascertain whether the motion appears forwards or backwards in a single specified direction.

In this work, the latter approach is addressed. For each segment of the *FB-nav* recorded path, the visual information of the segment end is described in terms of features observed. This information enables to later determinate whether the robot is located near the recorded position and subsequently navigate precisely to the same location.

The method used is based on comparison of angular measurements from the two positions. In both the recorded and current position, the image of surrounding environment is acquired. The matching landmarks are detected in these images, and identified using either SURF [13], SIFT [1], ORB [16] or BRISK [12] descriptors. The angle sizes between the landmarks are compared. Difference in each single angle yields only limited information regarding the relative position, but with sufficient number and good distribution of detected landmarks, an approximate vector towards the next position can be gained.

Since the robot used can only move on ground, we can simplify the computation and project all landmarks onto a plane, performing the computation completely in 2D.

3.2 Geometry of Single Angle

According to the generalized Thales Theorem, all the points observing two landmarks under a given angle are located on a circumference segment. Figure 1 gives an overview of the situation.

Distance d of observation point from the center and distance v of the center of the circumference from the line connecting the landmarks are given by following equations. s denotes scalar distance of the landmarks, α is the observation angle.

$$v = \frac{s}{2\tan\alpha} \tag{1}$$

$$d = \frac{s}{2\sin\alpha}, \tag{2}$$

If the landmark positions are known and fixed, the parameters of the circumference containing all possible observation points are dependent only on the observation angle. A smaller angle corresponds to a larger circumference and vice versa. Thus, we can state that moving towards the center of the circumference means a shift towards positions with a greater observation angle, while moving away means the opposite.

When the position of the landmarks is unknown, the information about landmark positions is limited to the corresponding direction vectors and angle α between them. In such case, the direction vector towards the center cannot be computed. However, it can be approximated by the axis of the angle α. As shown in [7], the error of this estimation is bounded by Eq. 3. The tightness of the bound is dependent on the angle size only. Angles with sufficiently small error can thus be selected.

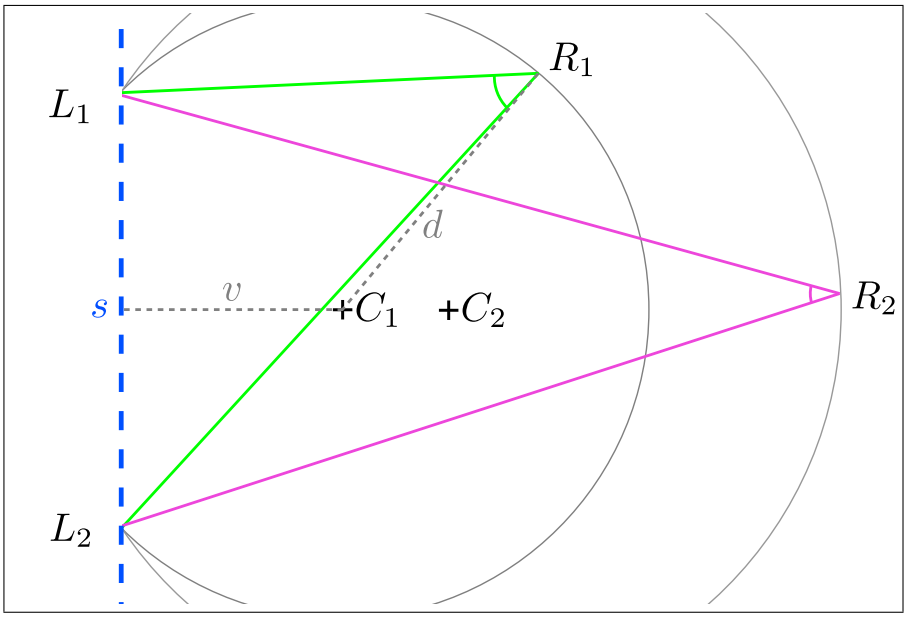

Fig. 1. Two landmarks observed under different angles. The points L_1, L_2 are landmarks, R_1, R_2 are two observing positions, while the points C_1, C_2 are the respective circumference centers. The corresponding d and v are also shown for the point R_1.

$$\beta \leq \frac{\pi}{2} - \frac{\alpha}{2} \tag{3}$$

3.3 Estimation

From each angle, relative position of the two viewpoints can be estimated in the direction of the angle axis. Without knowing the respective landmark distance distance, only the direction and orientation can be computed, not absolute scale. In the next step, these partial estimates need to be aggregated into a single correction vector. As the goal is limited to distinguishing whether the target position at the end of a segment is located still ahead or behind the robot, a simplified aggregation by voting can be used.

For each pair of observed landmarks, consider the angle between them. Let a be the direction of the angle axis. Compare the angle with the corresponding one recorded at the target location in terms of size. Result determines the direction on the axis vector. The landmark pair contributes to voting according to Eq. 4 with the magnitude equal to the angle between the vectors representing angle axis a and the direction of robot movement r. The contribution is weighted by maximum error β_a, calculated by Eq. 3.

$$C = \sum_a \frac{\frac{\pi}{2} - |a - r|}{\max(\beta_a, 1)} \tag{4}$$

Fig. 2. Unwrapped image from the robot camera.

If the sum of contributions C is positive, the target is considered to be still ahead of the robot. The negative C signifies the opposite, and is interpreted as passing the end of the current path segment.

3.4 Order-Based Filtration

Nevertheless, the method exhibits a considerable weakness. While the influence of noised landmark positions has little effect on the final method performance, the method remains highly sensitive to mismatches of landmark pairs. As the landmarks are handled pairwise in the computation, even small number of mismatches can introduce a large systemic error. To counteract this fact, a method to detect and discard outliers is necessary.

Assuming that viewpoints of two subsequent images are nearby, the geometry of the scene will be similar in the both. Therefore, horizontal order of majority of landmarks will be the preserved. Any deviations from the previous rule can be caused either by mismatching of features between the two frames, or by landmarks whose distance from the observer is significantly dissimilar to others. Neither of those is favorable to the computation, and all such landmarks can be disregarded.

The filtration process is iterative. Landmark ordering is considered pairwise. Landmark pairs that have different ordering between the two images are considered in conflict. In each cycle, landmark contained in greatest number of conflicting pairs is removed. Process is repeated until conflicts are eliminated.

Although only a small number of detected points is necessary to perform direction estimation, effect of any erroneous detection is significant in such small sets. To increase robustness, a lower bound on a number of detected matches can be set and any result calculated from lesser amount of matches is not considered valid.

4 Experimental Results

4.1 Experiment Setup

The proposed method has been validated in a real environment. The mobile robot used is Evolutionary Robotics, model ER1 fitted with a laptop PCB camera with catadioptric lens. Video stream with resolution 2048x1536 was recorded, and the omni-directional image was unwrapped to 3111x241 pixels before processing (Fig. 2).

The camera placement in a sufficient height above the top of the robot prevents observing of the robot body. The robot takes the advantage of a reliable dead-reckoning system used in the experiment as the ground truth. Nevertheless, the robot design is tailored for indoor use only.

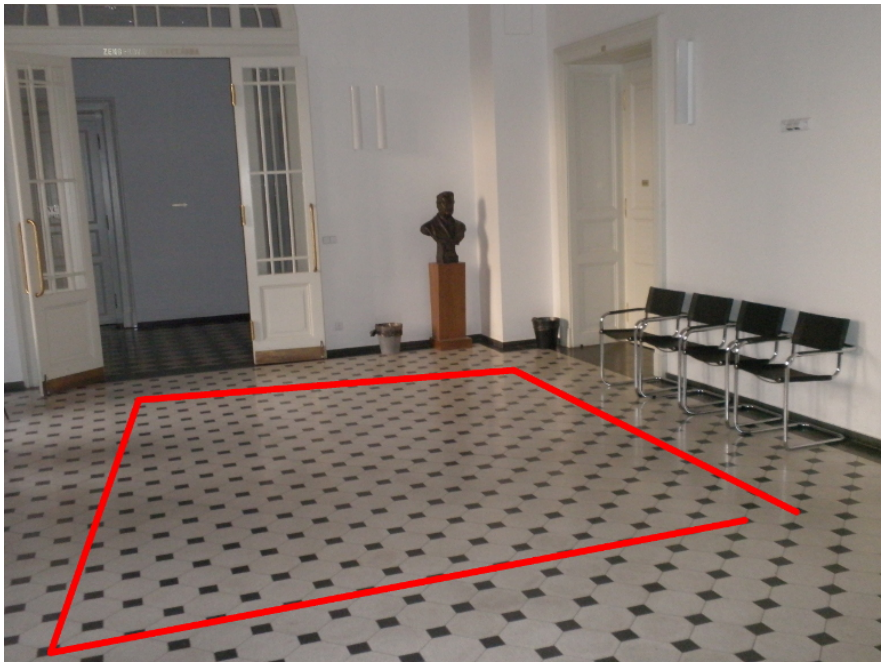

Fig. 3. The area of the experiment. The red line shows the trajectory of the robot.

The experiment was performed in the indoor office-type of environment, see Fig. 3. The experimental scene has primarily been set as static with minor variation in shape and structure. The variations due to presence of the operator and other pedestrians have not been observed as imposing any substantial influence on the experimental results.

The robot was manually driven multiple times along a trajectory consisting of several straight-line segments (Fig. 3). To test the error of the segment-end detection, each segment was required to be recorded past the end (therefore, past the point when the dead-reckoning would terminate it). For this reason, the original method was not used for the data gathering itself.

The first run was considered for a training, during which the endings of segments used by *FB-nav* in its learning phase are saved as a collection of observation angles. During the following runs, segment ends are detected by the presented method. A success level has been calculated as a distance between the closest position to a previously saved target point and the point indicated by the method as the end of a segment (as explained in Fig. 4).

4.2 Results

Localization was performed on the recorded experimental data. Various image feature detectors (SIFT, SURF, ORB, BRISK) were used in the first step of the computation. The OpenCV implementation [4] of all these detectors was used. The images underwent compensation of the camera embedded distortion from use of the catadioptric lens

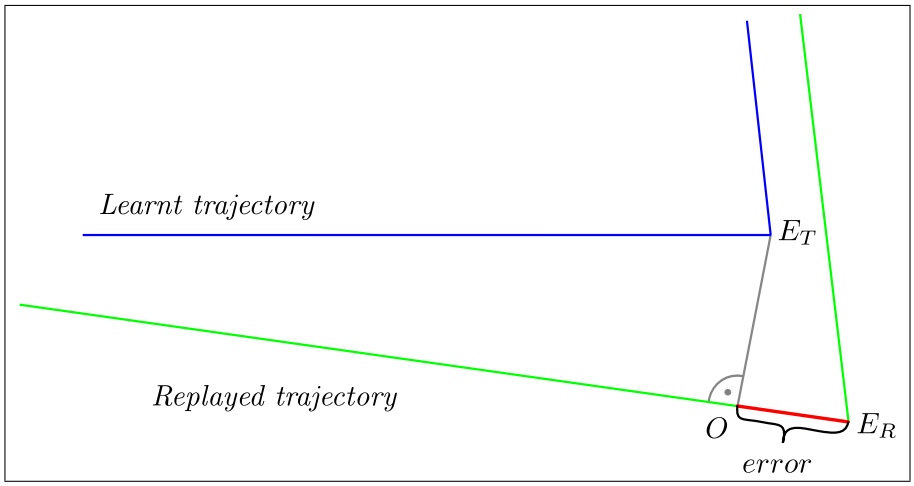

Fig. 4. The error is measured as a distance of the proposed segment end E_R from closest point O on the replayed trajectory.

and thus potentially contain deformations. Consequently, the suitability of feature detectors was in question. Results obtained of diverse features are reported side-by-side for comparison.

Principal results of the experiment are in the Table 1, which shows errors in calculation of the segment end. For each feature type, the five point statistics is presented.As observed herein, the error in finding the segment ending is mostly in the order of centimeters. Even this exhibits less precision than afforded by a dead-reckoning approach, this error is not accumulative thus does not increase beyond any limits over the run time of the system.

Considering the size of robot and scale of the environment, the precision in average case is sufficient for the task of segment ending detection in the *FB-nav* method. The results for ORB and BRISK detectors are less reliable.

The efficiency of any feature-based method is conditioned by existence of a sufficient amount of detected features along the recorded path. To investigate limits of the presented method in this parameter, comparison based on the number of detected features was also performed. Fig. 5 details relation between the number of successfully

Table 1. Error in segment termination (five point statistic)

	SIFT	SURF	ORB	BRISK
Average difference (cm)	9.5	8.9	13	13
Median difference (cm)	7.2	6	7.5	10
Standard deviation	6.4	6.6	13	9.3
Maximum difference (cm)	23	26	50	38
Minimum difference (cm)	1.8	1.8	0	1.4

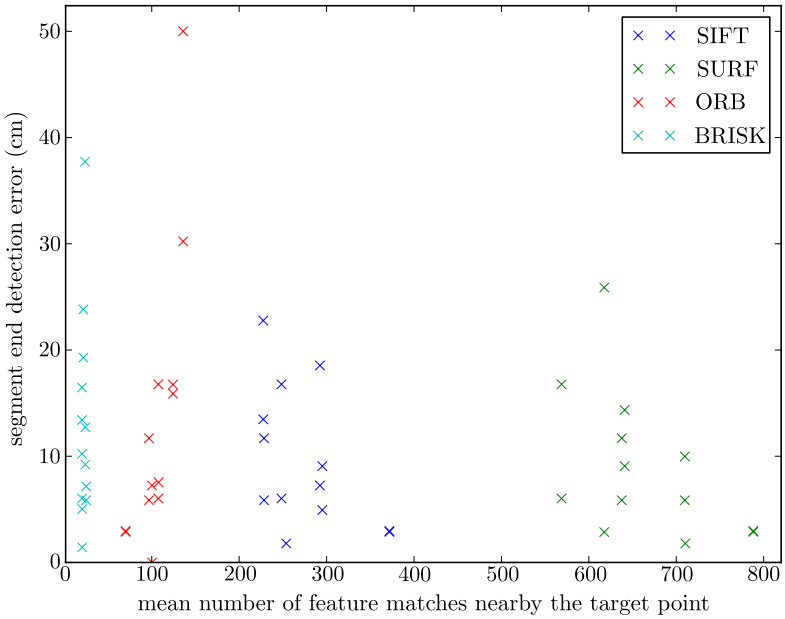

Fig. 5. Segment end detection error, plotted against number of detected features. Higher number of features is a slight advantage.

matched image features in the vicinity of the target point, and the error in the segment end detection.

We can see that detectors yielding higher amount of features have lower median error as shown in Table 1. However, the advantage that higher amount of features convey is not signifficant. We conclude that the method is not dependent on a high number of features with corresponding high computational costs. Past certain necessary minimum, additional features are of a limited benefit.

Also important is the fact how large vicinity of the target contains sufficient feature matches for navigation. Figure 6 shows the relation of distance from the target to the number of matched features. The number of detected features is highest at the target location and declines further afield. Note, that these values are dependent on the experiment location, visibility and a distance to landmarks.

Contrary to expectations, ORB and BRISK features were considerably less successful in matching than the other ones, with greater fallout caused by distance from target. This might be caused by deformation of the image imposed by unwrapping the image from omni-directional camera. Broad invariance of SURF feature detector brings minor advantages in this case.

These results also explain greater variance of error of endpoint detection based on ORB and BRISK-based features. If the target point neighborhood containing sufficient amount of features is smaller, even small deviations in the trajectory can have substantial impact.

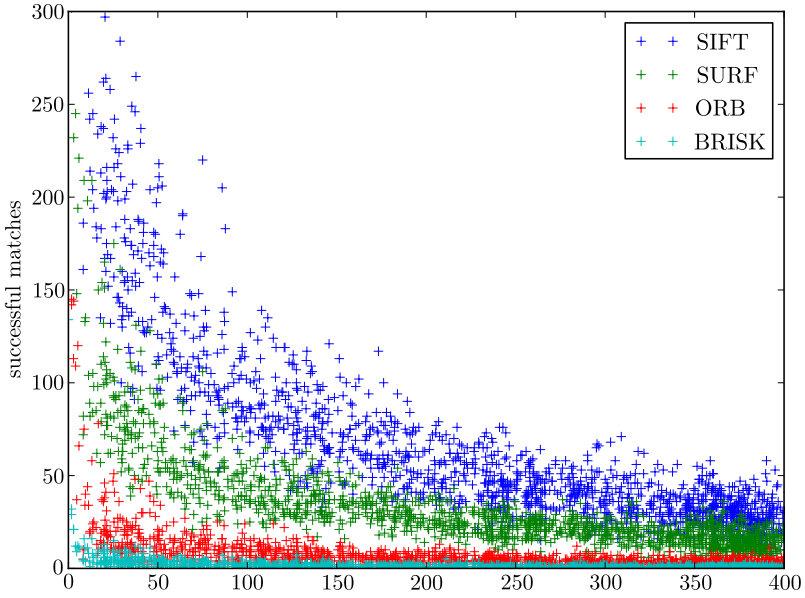

Fig. 6. The number of matches in relation to a distance from the target location

By selecting a desirable amount of features for localization, the afore data can also be used to estimate the maximum admissible length of a path segment. Longer segments run risk of the robot losing localization in the run in a given environment. In this case, the SURF detector enables segments of 2 meters, SIFT half that value, and other detectors require even shorter segments. These estimates necessarily vary between scenes.

5 Conclusion

The presented approach builds a novel *FB-nav* method as a generalization of the original *SURFnav* approach, deriving corrections of the robot bearing from forward-looking camera via usage of robust image features. This work aims to improve its performance by removal of the dependency on a reliable dead-reckoning system. Classical dead-reckoning approaches suffer accumulative errors and may fail completely under certain constraints (skid-control or aerial robots, wheel slippage, etc.) loosing their precision substantially. Hereby we suggest a novel method reusing the same modality – the same image features as used for the navigation in its previous outfit.

Angles between observed locations of such image feature-based landmarks are partial information that is readily available through most scenes. Although insufficient for absolute triangulation, this information is satisfactory for direction estimation, under certain constraints. In a limited form, it is able to support decisions, whether a pre-learned location has been achieved or allow judgements on a relative distance of these features for each particular observation point.

In this paper, a method using this information is presented. We show, that such method is able to replace dead-reckoning in the task of segment end detection, thus upgrading *FB-nav* to fully visual navigation. Experiments on real data verify feasibility of the presented approach.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. 110(3), 346–359 (Jun 2008)
2. Bekris, K.E., Argyros, A.A., Kavraki, L.E.: Exploiting Panoramic Vision for Angle-Based Robot Navigation, pp. 229–251. Springer (2006)
3. Blanc, G., Mezouar, Y., Martinet, P.: Indoor navigation of a wheeled mobile robot along visual routes. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. pp. 3354–3359 (April 2005)
4. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
5. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Trans. Pattern Anal. Mach. Intell. 29(6), 1052–1067 (Jun 2007)
6. Diosi, A., Remazeilles, A., Segvic, S., Chaumette, F.: Outdoor visual path following experiments. In: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on. pp. 4265–4270 (Oct 2007)
7. Dörfler, M., Přeučil, L.: Position correction using angular differences. In: POSTER 2014 - 18th International Student Conference on Electrical Engineering. Prague: Czech Technical University (Oct 2014)
8. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II. pp. 834–849 (2014)
9. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07). Nara, Japan (November 2007)
10. Krajník, T., Faigl, J., Vonásek, V., Košnar, K., Kulich, M., Přeučil, L.: Simple yet stable bearing-only navigation. Journal of Field Robotics 27(5), 511–533 (2010)
11. Krajník, T., Přeučil, L.: A simple visual navigation system with convergence property. In: Bruyninckx, H., Přeučil, L., Kulich, M. (eds.) European Robotics Symposium 2008, Springer Tracts in Advanced Robotics, vol. 44, pp. 283–292. Springer Berlin Heidelberg (2008)
12. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary robust invariant scalable keypoints. In: Computer Vision (ICCV), 2011 IEEE International Conference. pp. 2548–2555 (Nov 2011)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (Nov 2004)
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27(10), 1615–1630 (Oct 2005)
15. Nitsche, M., Pire, T., Krajník, T., Kulich, M., Mejail, M.: Monte carlo localization for teach-and-repeat feature-based navigation. In: Advances in Autonomous Robotics Systems - 15th Annual Conference, TAROS 2014, Birmingham, UK, September 1-3, 2014. Proceedings. pp. 13–24 (2014)
16. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 2564–2571 (Nov 2011)
17. Strasdat, H., Montiel, J.M.M., Davison, A.: Scale drift-aware large scale monocular slam. In: Proceedings of Robotics: Science and Systems. Zaragoza, Spain (June 2010)