Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Outdoor Visual Path Following Experiments

Albert Diosi, Anthony Remazeilles, Siniša Šegvić and François Chaumette

Abstract— **In this paper the performance of a topological-metric visual path following framework is investigated in different environments. The framework relies on a monocular camera as the only sensing modality. The path is represented as a series of reference images such that each neighboring pair contains a number of common landmarks. Local 3D geometries are reconstructed between the neighboring reference images in order to achieve fast feature prediction which allows the recovery from tracking failures. During navigation the robot is controlled using image-based visual servoing. The experiments show that the framework is robust against moving objects and moderate illumination changes. It is also shown that the system is capable of on-line path learning.**

I. INTRODUCTION

Intelligent autonomous vehicles have performed amazing feats outdoors. They have driven thousands of kilometers on freeways [11], they have navigated on the surface of Mars [2] and they have driven over 200km on a challenging desert route [17]. However, autonomous navigation outdoors using one camera and no other sensor still remains an exciting challenge.

One of the approaches for autonomous navigation using monocular vision is visual path following. In visual path following a path to follow can be represented by a series of reference images and corresponding robot actions (go forward, turn left, turn right) as in [9]. There a mobile robot navigated in indoor corridors by applying template matching to current and reference images and by using the stored actions. However, storing the robot actions is not necessary for navigation. In [13] a robot navigates a 127m long path outdoors while saving only a series of images from a camera with a fish-eye lens. To enable pose-based control of the robot in a global metric coordinate frame, a precise 3D reconstruction of the camera poses is necessary of the frequently (approx. every 70cm) saved reference images. In the 3D reconstruction process applied to feature points of the reference images, a bundle adjustment is used which results in a long (1 hour) learning phase unsuitable for on-line use. The length of the path measured by odometry is used to correct the scale of the map. After learning the path the robot can very accurately reproduce the path at 50cm/s velocity.

It turns out that reconstructing the robot's path, or having 3D information is not necessary. In [1] a robot navigated 140m outdoors at a speed of 35cm/s with 2D image information only. During mapping, image features were tracked and their image patches together with their x image coordinates were saved approx. every 60cm traveled. During navigation, the robot control was based on simple rules applied to the tracked feature coordinates in the next reference and current image. The robot however relied on frequent reference image switches to recover from occlusions due to moving objects. A person walking across the camera's field of view between two reference image switches would have caused a problem due to covering up each tracked feature.

The work described in [4] aimed at indoor navigation, can deal with occlusion at the price of using 3D information. A local 3D reconstruction is done between two reference omnidirectional images. During navigation, tracked features which have been occluded get projected back into the current image. The recovered pose of the robot is used to guide the robot towards the target image.

Building an accurate and consistent 3D representation of the environment can also be done using SLAM. For example in [7] a robot mapped a 100m path outdoor using a monocular camera and odometry. There were only 350 features in the map which in our view approaches the limit that a simple Kalman filter SLAM implementation can handle in real time on current PCs. However the simulation result in [3] of closing million landmark loops predict that monocular SLAM will be soon a viable choice for creating accurate maps with large numbers of landmarks.

In this paper the experimental evaluation of a visual path following framework is presented. This framework is similar to [4] in that only local 3D reconstruction is used and that occluded features get projected back into the image. However the rest of the details are different. For example in this paper a standard camera is used, tracking is used for mapping instead of matching, experiments are done outdoors and the centroids of image features are used to control the robot.

The concept of the framework has been evaluated using simulations in [12], while the feature tracker and the complete vision subsystem have been described in [14], [15]. As already mentioned, we thus focus in this paper on the numerous experimental results that have been obtained using a car-like vehicle.

II. VISUAL NAVIGATION

This section briefly describes the implemented visual navigation framework. The teaching of the robot i.e. the mapping of the environment is described first, followed by the description of the navigation process consisting of localization and robot control.

A. Mapping

Learning a path (i.e. mapping) starts with the manual driving of the robot on a reference path while processing

The presented work has been performed within the French national project Predit Mobivip and project Robea Bodega.

The authors are with IRISA/INRIA Rennes, Campus Beaulieu, 35042 Rennes cedex, France. Email: firstname.lastname@irisa.fr

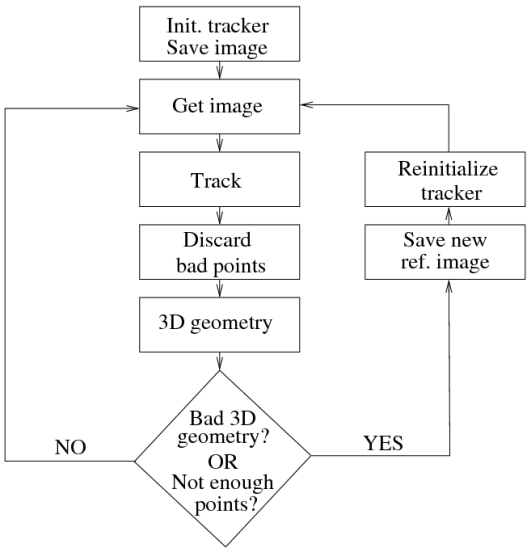

Fig. 1. The steps involved in building a representation of a path from a sequence of images, i.e. mapping.

(or storing for off-line mapping) the images from the robot's camera. From the images an internal representation of the path is created, as summarized in fig. 1. The mapping starts with finding Harris points [5] in the first image, initializing a Kanade-Lucas-Tomasi (KLT) feature tracker [16] and by saving the first image as the first reference image. The KLT[1] tracker was modified as proposed in [6] in order to improve performance in outdoor sequences acquired from a moving car. In the tracker postion, scale and contrast parameters of features are tracked. In the next step a new image is acquired and the tracked features are updated. The tracking of features which appear different than in the previous reference image is abandoned. The rest of the features are then used to estimate the 3D geometry between the previous reference and the current image. In the 3D geometry estimation, the essential matrix is recovered using the calibrated 5 point algorithm[2] [10] used in the MLESAC [18] random sampling framework. If the 3D reconstruction error is low and there are enough tracked features a new image is acquired. Otherwise the current image is saved as the next reference image. The relative pose of the current image with respect to the previous reference image and the 2D and 3D coordinates of the point features shared with the previous reference image are also saved. Then the tracker is reinitialized with new Harris points added to the old ones and the processing loop continues with acquiring a new image.

The resulting map (fig. 2) is used during autonomous navigation in the localization module to provide stable image points for image-based visual servoing.

[1]The source code of the KLT tracker maintained by Stan Birchfield can be found at http://www.ces.clemson.edu/~stb/klt/

[2]Free implementation is available in the VW library downloadable from http://www.doc.ic.ac.uk/~ajd/Scene/index.html.

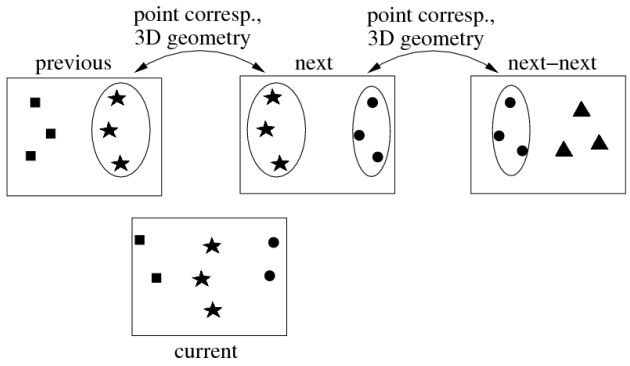

Fig. 2. The map consists of reference images, 2D and 3D information. During navigation, the point features from the map are projected into the current image and tracked.

B. Localization

The localization process during navigation is depicted in fig. 3. The navigation process is started with initial localization where the user selects a reference image close to the robot's current location. Then an image is acquired and matched to the selected reference image. The wide-baseline matching is done using SIFT descriptors [8]. The estimation of the camera pose using the matched points enables to project map points from the reference image into the current image. The projected points are then used to initialize a KLT tracker.

After the initial localization a new image is acquired and the point positions are updated by the tracker. Using the tracked points a three-view geometry calculation is performed between the previous reference, current and next reference image (fig. 2). If the current image is found to precede the next reference image, then points from the map are reprojected into the current image. The projected points are used to resume the tracking of points currently not tracked and to stop the tracking of points which are far from their projections. A new image is acquired next and the whole cycle continues with tracking. However, if it is found that the current image comes after the next reference image, a topological transition is made i.e. the next-next reference image (fig. 2) becomes the next reference image. The tracker is then reinitialized with points from the map and the process continues with acquiring a new image.

Wide-baseline matching is only used outside the initial localization phase if most features are lost for example due to a total obstruction of the camera's field of view. In such case automatic reinitialization is carried out by matching with the nearest reference images.

C. Motion Control

In the motion control scheme the robot is not required to accurately reach each reference image of the path, nor to follow accurately the learned path since it may not be useful during navigation. In practice, the exact motion of the robot should be controlled by an obstacle avoidance module which we plan to implement soon. Therefore a simple control

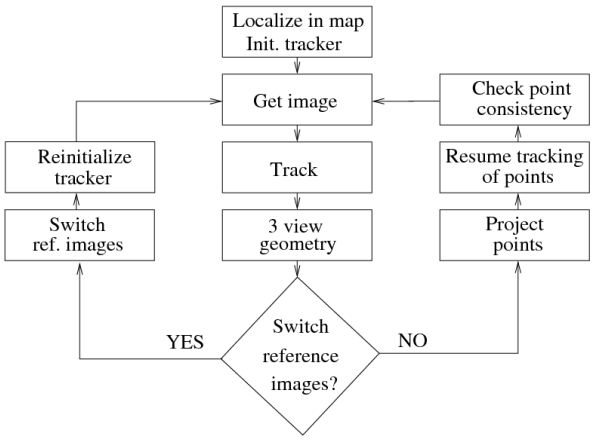

Fig. 3. Visual localization during navigation.

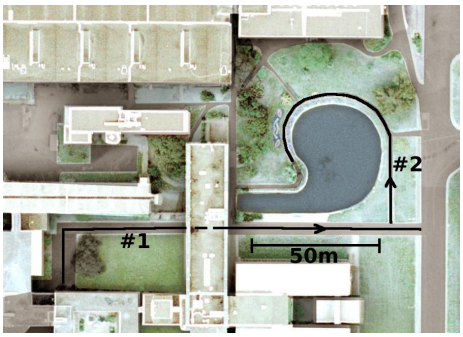

Fig. 4. Paths for experiments 1 and 2.

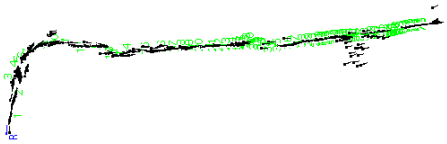

Fig. 5. Navigation results in exp. 1 shown as reconstructed robot poses (black) overlaid on 77 reconstructed reference image poses (green dots and numbers). "R" at the bottom marks the first reference image pose.

algorithm was implemented where the difference in the x-coordinates (assuming the forward facing camera's horizontal axis is orthogonal with the axis of robot rotation) of the centroid of features in the current (x_c) and next reference image (x_n) are fed back into the motion controller of the robot as steering angle Φ:

$$\Phi = -\ a(x_c - x_n)$$

The translational velocity is set to a constant value, except during turns, where it is reduced (to a smaller constant value) to ease the tracking of quickly moving features in the image. Such turns are automatically detected during navigation, by the analysis of the difference in the feature centroids in the current, next and next-next image.

Deciding when to stop when reaching the goal position, is carried out similarly to the reference image switching strategy of [1] by observing when the error between current and last-reference image features starts to rise.

III. EXPERIMENTAL RESULTS

In the experiments a CyCab, a French-made 4 wheel drive, 4 wheel steered intelligent vehicle designed to carry 2 passengers was used. In our CyCab all computations except the low-level control were carried out on a laptop with a 2GHz Centrino processor. A 70° field of view, forward looking, B&W Allied Vision Marlin (F-131B) camera was mounted on the robot at a 65cm height. Except in experiment 3 the camera was used in auto shutter mode, with the rest of the settings constant.

During all experiments, no software parameters were changed except that of the forward and turning speed. Mapping has been performed off-line, except in experiment 6. The image resolution in the experiments was 320x240.

A. Experiment 1

Experiment 1 (see fig. 4) was conducted on an overcast day with a short time between mapping and navigation. Most views on the 158m long path contained buildings which provided stable image features. The main challenges in this experiment were (i) motion blur in the teaching sequence caused by fast driving, (ii) driving under a building which caused a quick illumination change and (iii) people (more than 10) and cars covering up features during navigation.

In the teaching phase, 958 logged images were reduced into 77 reference images in 257s (3.7fps). While the robot was moving at 50cm/s in turns and at 90cm/s otherwise during navigation, 934 images were processed at 4.1fps on average. Statistics regarding mapping and navigation are shown in tab. I. Reconstructed robot and reference image poses shown in fig. 5 were only used for assessing the performance of the system.

The quick illumination change when driving under the building was easily handled due to the implemented illumination compensation in the tracker. Motion blur in the teaching sequence did not impair the performance of the system. The moving objects and persons did not affect the navigation because the tracking of features re-appearing after occlusion were resumed immediately due to the feature reprojection scheme. Figure 6 contains images processed at the end of the navigation. They describe an interesting situation where a moving car progressively occludes most features. It can be seen that the tracking of re-appearing features is resumed.

B. Experiment 2

Experiment 2 was conducted on a narrow path along a small lake (fig. 4 and 10). Mapping was carried out in June, under the strong summer sun. Navigation took place in October, when vegetation and light conditions were very different (fig. 8). Despite the large change in the environment, CyCab managed to navigate about 80% of the path with only one human intervention. At one place CyCab started brushing the rose plants on the left side of the path in fig. 8 therefore we stopped the vehicle. Without stopping

Fig. 6. Every second frame of a sequence from experiment 1 demonstrates robust feature (yellow crosses) tracking resumption after occlusion by a passing car.

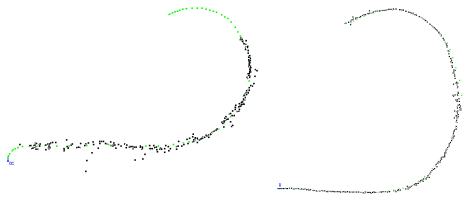

Fig. 7. Navigation results in experiment 2 (left) using a map created 3 months earlier. CyCab completed about 80% of the path. CyCab could navigate the whole path using a new map (right).

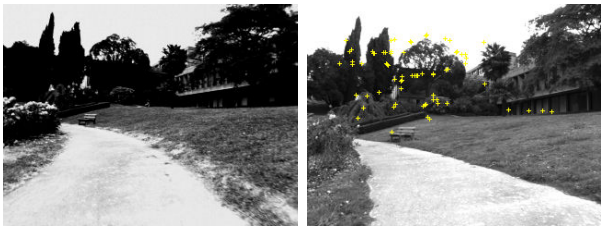

Fig. 8. Large difference in illumination and in the vegetation between a 3 month old reference image (left) and a current image used in navigation in exp. 2.

Fig. 9. Difference between the reference image (left) and current image (right) in exp. 2 which the vision system could not handle any more. Notice the missing flowers in the flowerbed.

Fig. 10. CyCab driving on the narrow path in experiment 2.

the vision system, CyCab was moved 50cm to the right and its automatic motion was resumed. CyCab's vision system gave up close to the end of the track when the change in the environment was too large (see fig. 9). Even though CyCab did not complete the whole path (see the left image in fig. 7 where it failed), this experiment still represents a large success because of the difficult conditions CyCab could handle.

Shortly after CyCab got lost, we have repeated the experiment using a new map. As it can be seen in the right image of fig. 7, CyCab completed the path without any problems.

The frame rates during navigation are lower in this experiment (see tab. I) due to implementation and processing platform limitations.

C. Experiment 3

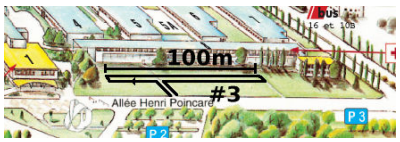

Fig. 11. The path for experiment 3.

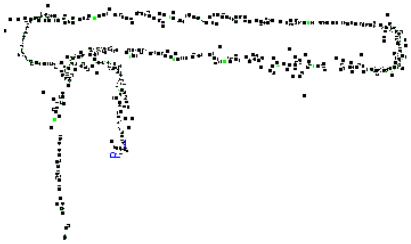

Fig. 12. Larger noise in the reconstructed robot poses where all features are far away in experiment 3.

Fig. 13. Sun shining into the camera in the reference image (left), but not in the current image (right) during navigation in exp. 3.

In experiment 3 CyCab completed an approximately 304m track, where in some places (right side in fig. 11), the closest features were more than 100m away. The CyCab wide track enabled us to examine the lateral error in CyCab's motion under such conditions. The mapping and navigation part of the experiment was conducted in succession, under very bright lighting conditions. Instead of the usual auto-shutter mode, the camera was used in its high dynamic range mode.

As one can expect, the error in the estimated pose during navigation was the largest at those places, where there were no close features. The large pose error is represented by noisy points in the right bottom part of the path in fig. 12. The 3D pose error resulted in an early switching of a few reference images during turning, and subsequently following the learned path with a 1m lateral error on a short section of the path. Other than that, CyCab performed excellently even when the sun was shining into its camera as in fig. 13.

D. Experiment 4

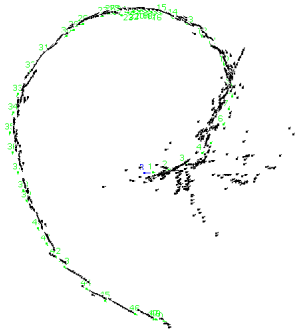

Fig. 14. Navigation results in the loop closing experiment (exp. 4).

Fig. 15. Sun shining into the camera in the reference image (left) of exp. 4, but not in the current image (right) during navigation.

The aim of this experiment was to investigate navigation in a loop. The teaching was performed by driving CyCab in a full loop in a circular parking lot of approx. 119m circumference. The beginning and end of the loop were closed by matching the first and last image of the teaching sequence. If neighboring nodes were connected with line segments, then the first and the last green dot in fig. 14 were connected.

Between mapping and navigation, 4 cars left the parking lot. One of these cars provided the only close features at the beginning of the loop, which resulted in noisy pose estimates. CyCab successfully completed 1.25 loops, while the small change of illumination (see fig. 15) did not matter.

Fig. 16. The first images during navigation in exp. 5 (left) and in 6 (right). In exp. 5 the robot drove until the end of the road. In exp. 6 the robot parked itself into the garage close to the center of the image.

E. Experiment 5

Fig. 17. Navigation results in experiment 5.

In experiment 5 (see fig. 16 and 17), CyCab completed a 100m straight path at a fast, 1.8m/s speed. A short video clip of this experiment (together with the next experiment) is included as supplementary material.

F. Experiment 6

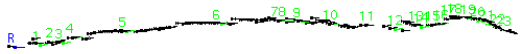

Fig. 18. Navigation results in experiment 6.

In this experiment on-line mapping (i.e. processing the images as they are grabbed) and a practical application is demonstrated. In the current state of the navigation system, i.e. without obstacle detection-avoidance, etc. the practical applications are limited. However, even now the framework can be used for automatic parking in private properties which are under the control of the user.

During the experiment a map was created on-line while driving CyCab from the entrance of IRISA to the CyCab garage approx. 50m away (see fig. 16 and 18) at about 50cm/s. Then CyCab was manually driven to the entrance of IRISA where the driver got out and CyCab drove itself into the garage. During mapping clouds covered the sun, while during navigation the sun was not covered.

G. Discussion

By performing simple image-based visual servoing instead of position-based control of the robot, one can have many advantages. Since there is no need for an accurate robot pose during navigation, one can allow a larger 3D reconstruction error during mapping. Because of this, there is no need to perform a computationally costly global bundle adjustment and mapping can be performed on-line. During the experiments it was noticed that, after the baseline between reference images increased beyond a certain distance, the 3D reconstruction error increased as well. Therefore if a larger 3D reconstruction error is allowed, then one can have

TABLE I

SUMMARY OF THE VISUAL PATH FOLLOWING EXPERIMENTS

	Learning						Navigation					
exp.	raw images	ref. images	proc. time [s]	fps	path [m]	meters per ref. image	images	time [s]	fps	v forw. [cm/s]	v turning [cm/s]	human interv.
1	958	77	257	3.7	158	2	934	226	4.1	90	50	0
2	862	51	208	4.1	96	1.9	532	262	2	50	30	1
3	2454	97	592	4.1	304	3.1	2272	516	4.4	80	30	0
4	1425	48	237	6	119	2.5	1812	385	4.7	50	40	0
5	785	32	167	4.7	100	3.1	280	78	3.6	180	40	0
6	371	22	102	3.6	50	2.4	406	94	4.3	80	40	0

larger distances between reference images, and the memory requirement for storing the map is reduced. This can be seen for example in experiment 3 where the average distance between reference images was 3.1m.

The implemented contrast compensation in the tracker is able to handle large affine changes of illumination between the reference and current images which was crucial for example during experiment 2 (fig. 8).

The use of 3D information enables to resume the tracking of features just becoming visible after occlusion as can be seen in fig. 6. This property is important in dynamic environments. Also, having 3D information also enables to check the consistency of the tracked features. Tracked points which "jump" from the background onto a moving object in the foreground are discarded. Even though having 3D information may not be necessary for path following as stated in the introduction, it may extend the area of applicability of an outdoor path following system.

The framework enables the learning and navigation of long paths since the memory and computational requirements for mapping grow linearly with the length of the path. The computational cost during navigation is approx. constant.

The main weakness in the current implementation of the framework is the reliance on 3D pose to switch reference images. In cases when there is a large 3D error, it can happen that a reference image switch is not performed, or it is performed in the wrong direction. Such misbehavior occasionally happens when most of the observed points are located on a plane or on a tree. To address this issue, we are planning to investigate a reference image switching strategy based on the more stable image information.

A further limitation is that of the illumination. Extreme illumination changes such as the sun shining into the camera during mapping but not during navigation, or the lack of light may impair the performance of the framework, especially that of the matcher.

At last, navigation frameworks for uncontrolled environments such as the one described in this paper should be able to detect and avoid obstacles. Since this is not implemented in the framework yet, it constitutes part of the future work.

IV. CONCLUSIONS

An experimental evaluation of a framework for visual path following in outdoor urban environments using only monocular vision was presented in this paper. In the framework no other sensor than a camera was used. It was shown that the use of local 3D information, contrast compensation and image-based visual servoing can lead to a system capable of navigating in diverse outdoor environments with reasonably changing lighting conditions and moving objects.

V. ACKNOWLEDGMENTS

The help of Fabien Spindler and Andrea Cherubini during experiments are gratefully acknowledged, as well as Geoffrey Taylor's comments during the preparation of this paper.

REFERENCES

[1] Z. Chen and S. T. Birchfield. Qualitative vision-based mobile robot navigation. In *ICRA*, Orlando, 2006.

[2] Y. Cheng, M.W. Maimone, and L. Matthies. Visual odometry on the Mars exploration rovers - a tool to ensure accurate driving and science imaging. *Robotics & Automation Magazine*, 13(2), 2006.

[3] U. Frese and L. Schroder. Closing a million-landmarks loop. In *IROS*, Beijing, 2006.

[4] T. Goedeme, T. Tuytelaars, G. Vanacker, M. Nuttin, and L. Van Gool. Feature based omnidirectional sparse visual path following. In *IROS*, Edmonton, Canada, August 2005.

[5] C. Harris and M.J. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147–152, 1988.

[6] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *ICCV*, volume 1, pages 684–689, 2001.

[7] T Lemaire, C. Berger, I. Jung, and S. Lacroix. Vision-based SLAM: Stereo and monocular approaches. *IJCV/IJRR special joint issue*, 2007.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[9] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *ICRA*, Minneapolis, April 1996.

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

[11] T. Pomerleau, D.and Jochem. Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert*, 11(2), 1996.

[12] A. Remazeilles, P. Gros, and F. Chaumette. 3D navigation based on a visual memory. In *ICRA'06*, 2006.

[13] E. Royer, J. Bom, M. Dhome, B Thuillot, M. Lhuillier, and F. Marmoiton. Outdoor autonomous navigation using monocular vision. In *IROS*, pages 3395–3400, Edmonton, Canada, August 2005.

[14] S. Segvic, A. Remazeilles, and F. Chaumette. Enhancing the point feature tracker by adaptive modelling of the feature support. In *ECCV*, Graz, Austria, 2006.

[15] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette. Large scale vision based navigation without an accurate global reconstruction. In *CVPR'07*, Minneapolis, June 2007.

[16] Jianbo Shi and Carlo Tomasi. Good features to track. In *CVPR'94*, pages 593–600, 1994.

[17] S. Thrun et al. Stanley, the robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9), 2006.

[18] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.