EXPERIMENTAL EVALUATION OF AN URBAN VISUAL PATH FOLLOWING FRAMEWORK

Albert Diosi [*] **Anthony Remazeilles** [*,2] **Siniša Šegvić** [*,3]
François Chaumette [*]

[*] *IRISA/INRIA Rennes, Campus Beaulieu, 35042 Rennes cedex, France. Email:* `firstname.lastname@irisa.fr`

Abstract: Robot cars will likely play an important role in the future. In this paper a visual path following framework for urban environments is experimentally evaluated. The framework's hybrid topological-metric approach for representing the environment provides stable interest points for image-based visual servoing during navigation. The presented experimental results with a robot car show that the framework is robust against changing illumination and moving objects covering up parts of the field of view of the monocular camera. Furthermore, there is no need to perform bundle adjustment nor to use odometry.

Keywords: robot vision, mobile robots, tracking applications

1. INTRODUCTION

Intelligent autonomous vehicles have performed amazing feats outdoors. They have driven thousands of kilometers on freeways (Pomerleau, 1996), they have navigated on the surface of Mars (Cheng *et al.*, 2006) and they have driven over 200km on a challenging desert route (Thrun *et al.*, 2006). 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 (Matsumoto *et al.*, 1996). 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 (Royer *et al.*, 2005) 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 of the frequently (approx. every 70cm) saved reference images is necessary. 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 (Chen and Birchfield, 2006) 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

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

[2] Currently with CEA-LIST DTSI/SRI/LTC, route du panorama BP6, 92265 Fontenay aux Roses Cedex, France

[3] Currently with EMT, TU Graz, Kopernikusgasse 24/4, A-8010 Graz, Austria

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 (Goedeme *et al.*, 2005) 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 (Lemaire *et al.*, 2007) 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 which a simple Kalman filter SLAM implementation can handle in real time. However the simulation result in (Frese and Schroder, 2006) of closing million landmark loops predict that monocular SLAM will be soon a viable choice for creating accurate maps with a large number of landmarks.

In this paper the experimental evaluation of a visual path following framework is presented. This framework is similar to (Goedeme *et al.*, 2005) 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 (Remazeilles *et al.*, 2006), and the feature tracker and the implemented vision system have been described in (Segvic *et al.*, 2006) and in (Segvic *et al.*, 2007) respectively.

2. 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.

2.1 Mapping

Learning a path i.e. mapping starts with the manual driving of the robot on a reference path while storing 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 (Harris and Stephens, 1988) in the first image, initializing a Kanade-Lucas-Tomasi (KLT) feature tracker (Shi and Tomasi, 1994) and by saving the first image as the first reference image. The KLT[4] tracker was modified to compensate for changes in the illumination as proposed in (Jin *et al.*, 2001). Besides the illumination compensation only isotropic warping is applied to the tracked 15×15 pixel image patches. In the next step a new image is acquired and the features are tracked. As the appearance of features may changes as the robot moves away from the previous reference image, the tracking of features with a high RMS error towards their reference appearance 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[5] (Nister, 2004) or the uncalibrated 7 point[6] (Hartley and Zisserman, 2004) algorithms used in the MLESAC (Torr and Zisserman, 2000) random sampling framework. If the 3D reconstruction error (evaluated as the reprojection error) is low and there are enough tracked features a new image is acquired. Otherwise the previous image is saved as the next reference image. The relative pose of the previous image with respect to the previous reference image and the 2D and 3D coordinates of the point

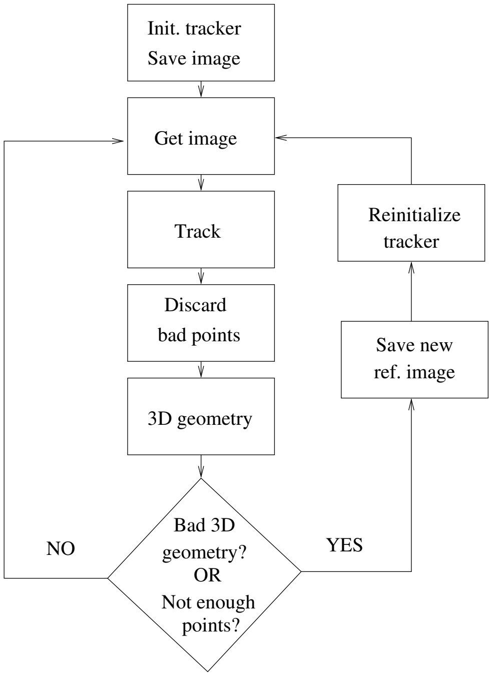

Fig. 1. Steps involved in building a representation of a path from a sequence of images, i.e. mapping.

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

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

[6] Free implementation is available in the VXL library downloadable from http://vxl.sourceforge.net.

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.

If the change between two consecutive images is too large to be handled by the tracker, matching (as described in the next section) is used to fill in the gap.

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

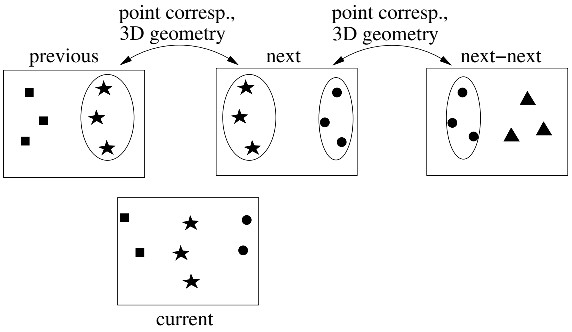

Fig. 2. The map consists of reference images, 2D and 3D information. During navigation, the point features from the map are projected into the current image and tracked.

2.2 *Localization*

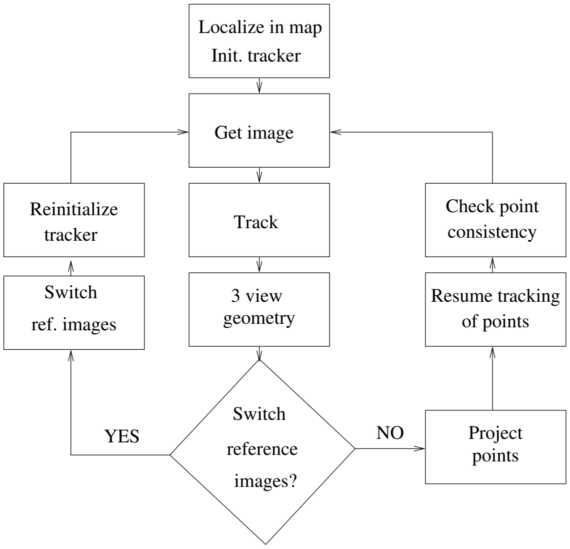

Fig. 3. Visual localization during navigation.

The localization process during navigation is depicted in fig. 3. The navigation process is started with the user selecting a reference image close to the robot's current location. Then an image is acquired and matched to the selected reference image. The matching is based on a correlation-based (Zhang *et al.*, 1995) approach and on matching SIFT descriptors (Lowe, 2004) determined at (i) the maxima of the difference of Gaussians (Lowe, 2004), (ii) multi scale Harris corners (Mikolajczyk and Schmid, 2004) and (iii) maximally stable external detector (Matas *et al.*, 2002). The estimation of the camera pose using the matched points enables to project map points in every iteration from the reference image into the current image. The projected points are then used to initialize a KLT tracker. Next, 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 be before 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 becomes the next reference image. The tracker is then reinitialized with points from the map and the process continues with acquiring a new image.

2.3 *Motion Control*

In the motion control scheme the robot is not required to accurately reach each reference image of the path, since the exact motion of the robot should be controlled by an obstacle avoidance module which is planned be to implemented soon. Therefore a simple control 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 and next reference image are fed back into the motion controller of the robot as steering angle.

The translational velocity is set to a constant value, except during sharp turns, where it is reduced to ease the tracking of quickly moving features in the image. Such sharp turns are automatically detected during navigation by thresholding the differences between the *x*-coordinates of feature centroids in the current image, next and next-next reference image.

3. EXPERIMENTAL RESULTS

All experiments were carried out with a CyCab. CyCabs are French-made 4 wheel drive, 4 wheel steered intelligent vehicles designed to carry 2 passengers. 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 Marlin (F-131B) camera was mounted on the robot at a 65cm height. The camera was used in auto shutter mode, with the rest of the settings constant in all experiments.

Although a large number of successful navigation experiments were conducted, only 2 experiments are shown here. During all experiments, only the maximum forward speed of the robot and the 3D reconstruction algorithm was changed. The image resolution in the experiments was 320x240.

3.1 *Experiment 1*

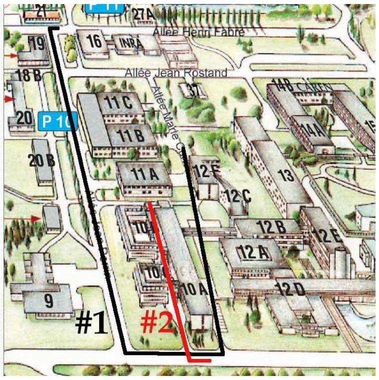

Fig. 4. Map of the university campus with the path for experiment 1 and 2 marked with different colors.

Experiment 1 (fig. 4) was carried out on an overcast day on an over 740m long path. The path entailed a variety of driving conditions including driving on a slope and under a building. In the first part of the experiment the robot was manually driven on the path. Next, a map was created from the over 4700 images logged during driving. Building the map took 47 minutes which corresponds to 1.6fps. It contained 215 reference images (on average 3.4m per image) and 30000 image points. In this experiment the 7-point algorithm was used in the 3D geometry estimation.

2.5 hours after the first part of the experiment, the second part was carried out consisting of the autonomous navigation. During navigation, the robot's speed was 30cm/s in turns, otherwise 80cm/s. The frame rate including displaying, logging and control was around 1Hz. In fig. 5 the visual odometry results of the reference path are depicted. Visual odometry is only a side effect of our vision system and it is not used during navigation. During navigation a car progressively covered up most of the tracked features (see fig. 6), however the tracking of re-appearing features was immediately resumed due to feature reprojection.

3.2 *Experiment 2*

The second experiment was carried out on an over 240m long path (fig. 4). The teaching of the robot took place 2-3 months before the navigation experiment. During teaching there were no clouds in the sky and the bright, midday summer sun cast strong shadows

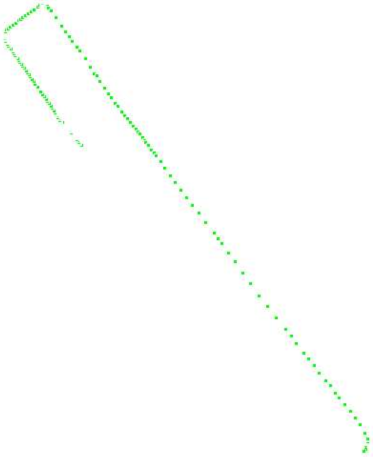

Fig. 5. Representation of the 740m reference path using visual odometry for experiment 1.

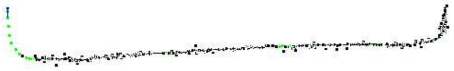

Fig. 7. Representation of the more than 240m reference path (green dots) using visual odometry for experiment 2. Robot locations during driving are show as black dots.

and over and under exposed the images. However, during navigation the sky was overcast. In fig. 8 the illumination difference in the image used for navigation (left image) and during teaching (right image) is shown.

The map (fig. 7) was created from about 2100 logged images and consisted of 95 reference images and 20000 point features. The average distance between reference images was 2.5m.

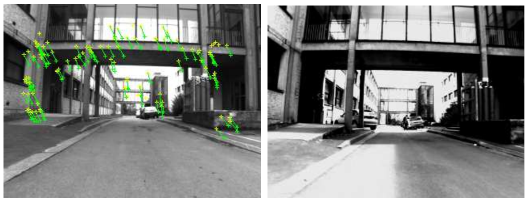

Fig. 8. Difference between the lighting conditions during navigation (left image) and mapping (right image) in experiment 2. Lines connect corresponding features in the current and reference image.

During navigation, the maximum speed of the robot was reduced to 50cm/s for safety reasons, because the first half of the path consisted of a single lane road full of parked cars. The control loop including image processing, logging and robot control ran at 1-2Hz. The faster 5-point algorithm was used for 3D reconstruction.

The autonomous navigation of the robot went without any human intervention even when people covered a large proportion of the image (fig. 9), or during the

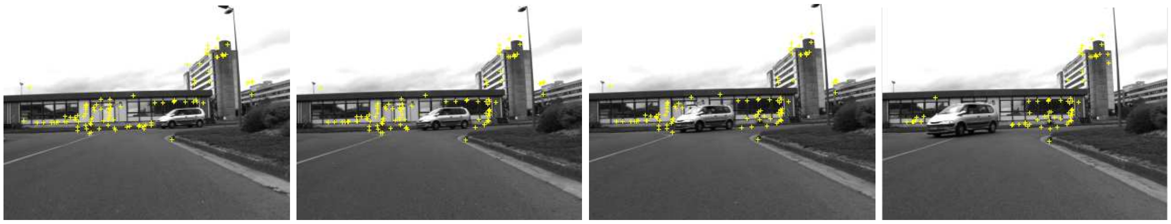

Fig. 6. Sequence from experiment 1 demonstrates robust feature (yellow crosses) tracking resumption after occlusion by a passing car.

left turn (fig. 10) where there was a large change in the scene due to change in the parked cars and only features from trees were available. This experiment well demonstrates the robustness of the framework against changes in the illumination and environment during teaching and navigation.

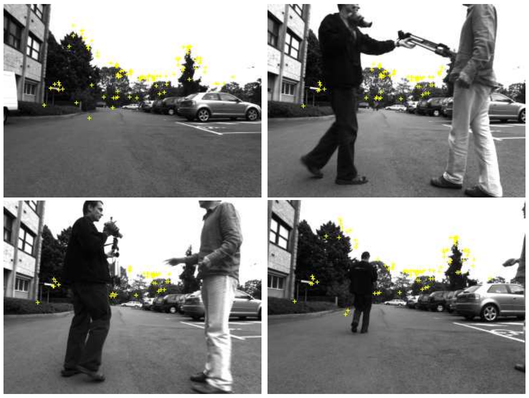

Fig. 9. Robust feature tracking resumption after occlusion during navigation in experiment 2.

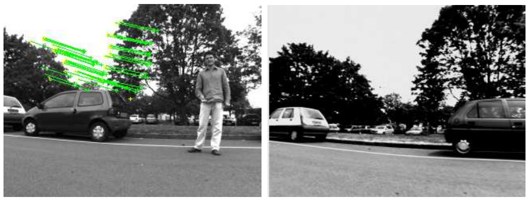

Fig. 10. Current (left) and reference (left) image during the left turn in experiment 2.

3.3 Discussion

As can be seen from the experiments, by performing 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 larger distances between reference images, and the memory requirement for storing the map is reduced. This can be seen for example in experiment 1 where the average distance between reference images was 3.4m.

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 and 9. This property is important in dynamic environments. 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 can be discarded.

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 approximately constant.

The framework works not only with a high quality camera, but also with an inexpensive webcam. The mapping and localization part was also successfully tested with a Logitech Quickcam Pro 4000 webcam.

Odometry is not used in the framework at any stage to make the problem more challenging. Omitting odometry also extends the area of possible applications of the vision framework to vehicles with no odometry e.g. hovercrafts and blimps.

The main weakness in the current implementation of the framework is the reliance on 3D pose to switch reference images. In cases where there are large 3D errors, 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.

There are other limitations of the framework. If the number of tracked points becomes too low, for example due to a large car occluding the field of view, the 3D reconstruction process stops, no image features are reprojected and no reference images are switched. We are planning to enable the robot to recover from such situations by matching.

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.

Since feature reprojection during mapping has not been implemented, stationary features covered up by moving objects are discarded. A higher rate of feature loss may reduce the distance between reference images. In severe conditions, the performance of the system may decline in spite of the use of matching to recover from large occlusions.

At last, navigation frameworks for uncontrolled environments should be able to detect and avoid obstacles. A panning mechanism may be used to keep map features in the field of view during large maneuvers. Since this is not implemented in the framework yet, it constitutes part of the future work.

4. 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 sensors than a camera were 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 long paths in outdoor environments with reasonably changing lighting conditions and moving objects.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the help of Fabien Spindler with problems regarding the robot and his help during experiments. Hai Tran is gratefully acknowledged for helping with an experiment.

REFERENCES

Chen, Z. and S. T. Birchfield (2006). Qualitative vision-based mobile robot navigation. In: *ICRA'07*. Orlando, USA. pp. 2686–2692.

Cheng, Y., M.W. Maimone and L. Matthies (2006). Visual odometry on the Mars exploration rovers - a tool to ensure accurate driving and science imaging. *Rob. & Aut. Mag.* **13**, 54–62.

Frese, U. and L. Schroder (2006). Closing a million-landmarks loop. In: *IROS'06*. Beijing,China. pp. 5032–5039.

Goedeme, T., T. Tuytelaars, G. Vanacker, M. Nuttin and L. Van Gool (2005). Feature based omnidirectional sparse visual path following. In: *IROS'05*. Edmonton, Canada. pp. 1806–1811.

Harris, C. and M.J. Stephens (1988). A combined corner and edge detector. In: *Proceedings of the Alvey Vision Conference*. Manchester, UK. pp. 147–152.

Hartley, R. I. and A. Zisserman (2004). *Multiple View Geometry in Computer Vision*. second ed.. Cambridge University Press, ISBN: 0521540518.

Jin, H., P. Favaro and S. Soatto (2001). Real-time feature tracking and outlier rejection with changes in illumination. In: *ICCV'01*. Vol. 1. Seattle, USA. pp. 684–689.

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

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

Matas, J., O. Chum, U. Martin and T. Pajdla (2002). Robust wide baseline stereo from maximally stable extremal regions. In: *BMVC'02*. Vol. 1. Cardiff. pp. 384–393.

Matsumoto, Y., M. Inaba and H. Inoue (1996). Visual navigation using view-sequenced route representation. In: *ICRA'96*. Minneapolis, USA.

Mikolajczyk, K. and C. Schmid (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision* **60**, 63–86.

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

Pomerleau, D.and Jochem, T. (1996). Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert* **11**, 19–27.

Remazeilles, A., P. Gros and F. Chaumette (2006). 3D navigation based on a visual memory. In: *ICRA'06*. Orlando, USA. pp. 2719–2725.

Royer, E., J. Bom, M. Dhome, B Thuillot, M. Lhuillier and F. Marmoiton (2005). Outdoor autonomous navigation using monocular vision. In: *IROS'05*. Edmonton, Canada. pp. 3395–3400.

Segvic, S., A. Remazeilles, A. Diosi and F. Chaumette (2007). Large scale vision based navigation without an accurate global reconstruction. In: *Accepted to CVPR'07*. Minneapolis, USA.

Segvic, S., A. Remazeilles and F. Chaumette (2006). Enhancing the point feature tracker by adaptive modelling of the feature support. In: *ECCV'06*. Graz, Austria.

Shi, Jianbo and Carlo Tomasi (1994). Good features to track. In: *CVPR'94*. Seattle, USA. pp. 593–600.

Thrun, S. et al. (2006). Stanley, the robot that won the DARPA Grand Challenge. *Journal of Field Robotics* **23**, 661–692.

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

Zhang, Z., R. Deriche, O. D. Faugeras and Q.-T. Luong (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence* **78**, 87–119.