Absolute Localization using Visual Data for Autonomous Vehicles

Safa Ouerghi[1], Rémi Boutteau[2], Pierre Merriaux[2], Nicolas Ragot[2],
Xavier Savatier [2] and Pascal Vasseur[3]
[1]*Sup'Com Tunis, Ariana, Tunisia*
[2]*Research Institute for Embedded Systems (IRSEEM-ESIGELEC), Saint Etienne du Rouvray, France*
[3]*LITIS Lab, University of Rouen, Saint Etienne du Rouvray, France*

Keywords: Absolute Localization, Monocular vision, Structure-From-Motion, Autonomous vehicle, Vehicle Environment Perception.

Abstract: In this paper, we propose an algorithm for estimating the absolute pose of a vehicle using visual data. Our method works in two steps: first we construct a visual map of geolocalized landmarks, then we localize the vehicle using this map. The main advantages of our method are that the localization of the vehicle is absolute and that it requires only a monocular camera and a low-cost GPS. We firstly outline our method, then we present our experimental results on real images using a reference database: the KITTI Vision Benchmark Suite.

1 INTRODUCTION

In recent years, vision sensors have become ever more common in vehicles and employed in many Advanced Driver Assistance Systems (ADAS): pedestrian detection (Dollar et al., 2012), bird's eye view for vehicle surrounding monitoring (Liu et al., 2008), backup cameras and lane departure warnings (Kozak et al., 2006). ADAS were initially limited to luxury vehicles but have now become available on high-volume models. The multiplication of vision sensors in vehicles has led to a reduction of costs and several different cameras can now be embedded. Besides the fact that cameras are inexpensive sensors, their main advantage is their versatility due to the huge quantity of information they provide. A single camera can indeed perform several functions depending on the implemented algorithms.

Through research on ADAS in the past few years, vehicles have now reached level 2 of automation as defined by SAE International's On-Road Automated Vehicle Standards Committee (SAE, 2015). This level refers to partial automation, which means that the vehicle can execute steering, acceleration and deceleration but the driver has to monitor the driving environment. Researchers and car manufacturers are now seeking to reach level 5 of automation, which is the level of full automation. From levels 3 to 5, the vehicle must be able to localize itself accurately and to collect information from its surrounding environment. These localization and perception tasks can be solved quite easily by adding new sensors such as lidars (Light Detection and Ranging) since they provide a 3D structure of the scene with a high level of accuracy and at a high rate. However, the constraints of the automotive industry are not compatible with such sensors. Costs are high and robustness is not optimal due to the lidar's mechanical parts. That is why vision-based localization is of great interest and the scientific community is working actively on this topic.

In this paper, we propose to use a monocular camera to localize the vehicle on an absolute map, which is the first step towards vehicle automation. First, a map is built using additional sensors, in particular an IMU/RTK GPS system. This step can be carried out by a specific vehicle and the map-building can be done off-line. Then, we propose an on-line process to localize the vehicle using only a monocular camera and a low-cost GPS.

In Section 2, we set out the state of the art with regard to vehicle localization using visual data. In Section 3, we present our method to build the map, and then to localize the vehicle within this map. Section 4 is dedicated to the experimental results obtained for the map-building and for the localization stages. Lastly, in Section 5, we present a conclusion and identify several directions for future work.

Ouerghi, S., Boutteau, R., Merriaux, P., Ragot, N., Savatier, X. and Vasseur, P.
Absolute Localization using Visual Data for Autonomous Vehicles.
DOI: 10.5220/0005682105950601
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 3: VISAPP*, pages 597-603
ISBN: 978-989-758-175-5

2 RELATED WORK

Vehicle localization is a fundamental requirement in robotics and intelligent transportation systems that has been extensively tackled in the two main fields dealing with robotics navigation namely Simultaneous Localization and Mapping (SLAM) (Durrant-Whyte and Bailey, 2006) (Bailey and Durrant-Whyte, 2006) (Dissanayake et al., 2001) (Munguia and Grau, 2007) and real-time Structure From Motion or Visual Odometry (Nister et al., 2006) (Maimone et al., 2007) (Comport et al., 2010).

SLAM approaches have been rooted in the local methods, operating in unknown environments, constructing a 3D model and estimating the camera pose relative to it. One of the most successful approaches to date is PTAM (Parallel Tracking and Mapping) (Klein and Murray, 2007). PTAM builds keyframes-based maps from data acquired by a monocular camera and uses tracking to estimate camera pose relative to the map. This is achieved by two threads running in parallel: one thread tracks the camera position relative to the existing map, and a second mapping thread integrates keyframes in the map and refines the map by performing a global bundle adjustment. Though PTAM provides good performance in localization and map building, it was originally designed for augmented reality applications in small workspaces and is therefore not prepared to cope with large-scale mapping involving loop closures and big maps. To extend the range of map coverage, Parallel Tracking And Multiple Mapping (PTAMM) (Castle and Murray, 2009) was proposed, allowing multiple maps with automatic switching. However, maps are limited in size and they are maintained independently with local coordinate frames which could be more adapted to robot navigation rather than vehicles and transportation systems.

Another alternative to SLAM algorithms is the place recognition approach that has been adapted in SLAM to detect loop closures (Ho and Newman, 2007) (Mei et al., 2011) (Cummins and Newman, 2008). The most successful approach to date is FAB-MAP (Cummins and Newman, 2010), a probabilistic approach to image matching based on a ”visual bag-of-words” model. FAB-MAP performs localization on trajectories up to 1000km in length. Despite the impressive results of FAB-MAP, it is only demonstrated on trajectories with one loop closure for each location (Cummins and Newman, 2010). It could consequently fail when revisiting the same location several times, reducing therefore the recall performance over time. An attempt to improve the aforementioned performance is the CAT-SLAM system (Maddern et al., 2012) performing the fusion of the robot local movement with appearance information using a particle filter.

In addition, any previous localization and mapping methods fail when they are tested in dynamic environments, in environments with too many or very few salient features or where there are partial or total occlusions which could justify the resort to the use of a GPS in vulnerable situations. On the other hand, the absolute localization is needed by several applications used in transportation systems, especially for future ADAS systems based on car to car communication and assuming that all positions are known in a global and therefore exchangeable reference frame (Rockl et al., 2008). This motivated us to conceive a low-cost robust absolute localization system aimed at vehicles and transportation systems.

3 METHODOLOGY

The general idea of our algorithm is to localize a camera given a set of previously captured and geolocalized images as described in Figure 1. Our method is composed of two main steps: the map-building of the geolocalized landmarks using geolocalized images, and then the localization within this map. In this section, we describe the two stages.

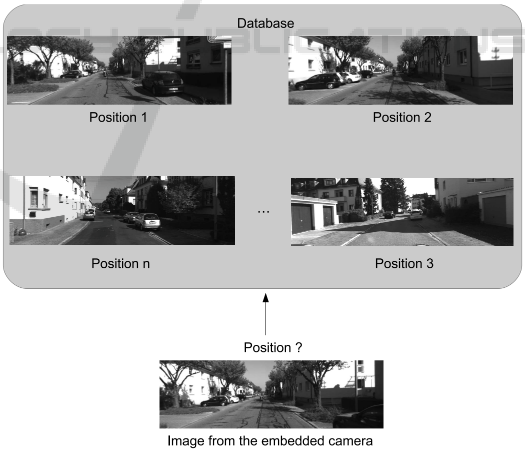

Figure 1: General idea of our method: given a set of geolocalized images, find the pose of a camera.

3.1 Map-building

Map-building is the key step to obtaining an accurate localization, so this step should be done carefully. If the map is not accurate, or does not contain sufficient robust features, localization will be very difficult, even impossible. To build the map, an instrumented vehicle has to be driven manually. This vehicle contains a very accurate localization system. In the KITTI dataset (Geiger et al., 2012) (Geiger et al., 2013), this system is composed of an IMU and an RTK-GPS. This system provides the vehicle's pose with accuracy within a few centimeters and high frequency (100 Hz). In our experiments, we only need the IMU/GPS measurements and the images from a monocular grayscale camera.

Once the acquisition has been done, we need to process the data to build a compact and representative map. Indeed, a sequence of a few kilometers generates several gigabytes of images, so we need to create a more compact database if we want it to be embedded in the vehicle. Instead of keeping all the images, the first step is to select only a few key images. This is achieved by detecting and matching keypoints between consecutive frames. Let the first image be considered as a keyframe. We detect and match the keypoints with the following frames. If the number of matches is too high, it means that the two images are very close visually, so the second frame is not a keyframe. We consequently need to define a threshold (500 in our experiments) below which there are not enough matched keypoints and a new keyframe is considered. The selection of keyframes has two advantages: first, the compression of the database by keeping only a few images, and second it ensures that there is motion between two frames. In the case of a stopped vehicle, if all the images are used, there is no distance between two images and the triangulation step is consequently impossible. Another way to build the map could be to use spatial discretization, for example by taking a keyframe every 3 meters. However, in the case of a rotation, we need more keyframes than in the case of a translation since the images vary quickly. To reject false matches, the fundamental matrix is computed using a RANSAC (Random Sample Consensus) scheme. The computation of the fundamental matrix is done using the Eight-Point algorithm (Hartley, 1997). The Five-Point algorithm (Nister, 2004) should provide the essential matrix with better accuracy than the Eight-Point algorithm but is considerably slower. In fact, what is important at this stage is not accuracy, but the rejection of false matches.

Our method relies on the use of keypoints so each image is compressed into a set of keypoints and their associated descriptors, as illustrated in Figure 2. Consequently, for each keyframe, keypoints are matched with the following keyframe. To do so, we use a binary descriptor, FREAK (Fast Retina Keypoint (Alahi et al., 2012)). It is advantageous because it detects keypoints quickly, and requires little memory. Moreover, the computation of similarities between descriptors is very fast because the Hamming distance can be used instead of the Euclidean distance. These keypoints are then triangulated using the poses provided by the IMU/GPS system. This is possible thanks to the high accuracy of this system, especially compared to the motion between two keyframes. This method has many advantages. 3D points are geolocalized and there is not the drift that exists in visual odometry systems. In addition, computation time is much shorter than that of SFM approaches that use bundle adjustment with loop-closure for example. For each keyframe, we store the keypoints, their descriptors and the computed 3D coordinates of the corresponding points. All this data represents only hundred of kilobytes, which is considerably smaller than the size of an image in memory.

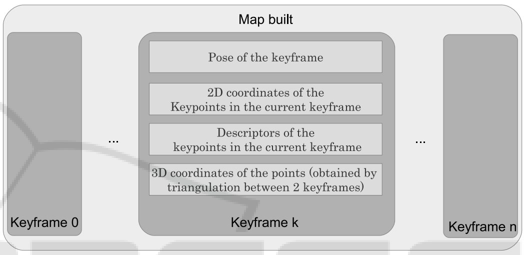
Map built
Pose of the keyframe
2D coordinates of the Keypoints in the current keyframe
Descriptors of the keypoints in the current keyframe
3D coordinates of the points (obtained by triangulation between 2 keyframes)
...
...
Keyframe 0
Keyframe k
Keyframe n

Figure 2: Constitution of the map built.

3.2 Localization

The mapping is now achieved and we have a database which contains, for each keyframe, the absolute position of the keyframe, the keypoints, their descriptors and the corresponding 3D points. The localization can now take place. In the localization process, we do not need the costly IMU/RTK-GPS system. The localization can be done using only a monocular camera and a low-cost GPS. The aim of the localization is, for each image captured by the camera, to find the pose of the camera in an on-line process (contrary to the map building which can be done off-line). To localize the vehicle, we need to find the nearest keyframe in the database. This is achieved using the low-cost GPS to have an initial guess of the position. Once the nearest keyframe is identified, the keypoints are detected in the current frame and then matched against it. The outliers are rejected using the fundamental matrix estimation with a RANSAC scheme. The 2D points of the current frame are now linked to those of the keyframe and consequently to their corresponding 3D points.

The localization is achieved by solving a Perspective-n-Point problem. The PnP problem, also known as pose estimation, was first introduced by

Fischler and Bolles (Fischler and Bolles, 1981). It consists in estimating the relative pose of a camera and an object knowing the position of n features in the object coordinate system and their projections in the image. Given a set of n correspondences between 3D points $\mathbf{M_i} = \begin{bmatrix} X_i & Y_i & Z_i & 1 \end{bmatrix}^T$ expressed in a world reference frame and their 2D projections $\mathbf{m_i} = \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T$, we look for the transformation $\mathbf{R}$, $\mathbf{t}$ from the world coordinate system to the camera coordinate system given by equation 1. To do so, we minimize the reprojection error E defined in equation 2, where P is the projection function which depends on the camera's pose and on the 3D coordinates of the points.

$$s \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{1}$$

$$E = \sum_{i=1}^{n} ||P(\mathbf{R}, \mathbf{t}, \mathbf{M}_i) - \mathbf{m}_i||_2 \tag{2}$$

The minimum number of correspondences to obtain a unique solution is $n = 4$. However, a larger set of points is required to obtain a more robust solution, using a RANSAC scheme (Fischler and Bolles, 1981). It is necessary because some 3D points may be incorrect (due to bad matches for example), and a linear estimation will provide an erroneous estimation of the relative pose if no outlier rejection is carried out.

4 EXPERIMENTAL RESULTS

We have evaluated our approach on the KITTI Vision Benchmark Suite. The KITTI database provides the images from the camera, their poses provided by the IMU/RTK GPS system, and the calibration matrix of the camera. We have built our map of geolocalized landmarks as described in Section 3 from the images of the left grayscale camera mounted on the vehicle. This camera is a Point Grey Flea 2 camera with a resolution of 1384x1032 (1.4MP) working at 10 frames per second. The images are rectified and cropped in the database so the final resolution is 1241x376 and the distance between two images is around 80cm.

4.1 Map building

Table 1 summarizes the results obtained for the map-building process. As we can see, the size of the database we obtained is forty times smaller than the one of the original KITTI database. The number of keyframes is four times smaller than the number of images of the KITTI database, which leads to a keyframe every 3m instead of 80cm.

Table 1: Comparison between the KITTI database and the map built.

	KITTI database	Map built
Number of poses	4540	1361
Mean distance between two poses (cm)	80	320
Size (MB)	1126	26

The rejection of outliers can be done using the Eight-Point or the Five-Point algorithms as discussed in Section 3. To justify our choice of using the Eight-Point algorithm, the computation time and the proportion of inliers are listed in Table 2. As we can see, the Five-Point algorithm is five times slower than the Eight-Point but the proportion of inliers is approximately the same for the two algorithms. As we are only interested in correctly matched points, and not in the estimated fundamental or essential matrix, the Eight-Point algorithm is sufficient. Figure 3 shows a matching example between two keyframes after outlier rejection.

Table 2: Comparison between 8-Point and 5-Point algorithms.

	Five-Point algorithm	Eight-Point algorithm
Computation Time	52 ms	11 ms
Proportion of inliers	59%	55 %

4.2 Localization

Our algorithm was evaluated on a sequence of the KITTI database. As there is no low-cost GPS in this dataset, we find the nearest keyframe using the ground truth value of the vehicle's pose and adding a random

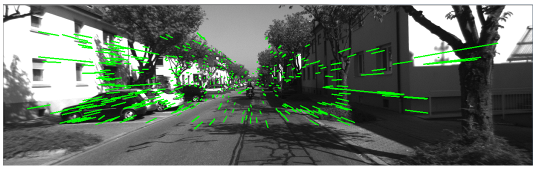

Figure 3: An example of matching (in green) between two keyframes after outlier rejection.

noise of ± 5 meters. Figures 4, 5 and 6 show the estimated trajectory compared with the ground truth from the IMU/RTK GPS. The mean error is around 17 centimeters, which is sufficient to control a vehicle autonomously. As shown in Figure 6, there are sometimes discontinuities between two estimated poses, which could be eliminated using a Kalman Filter (Kalman, 1960). The measure of the estimation confidence could be the reprojection error of the 3D points into the image.

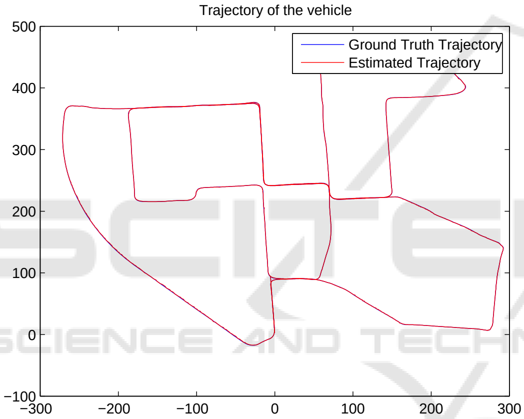

Figure 4: The estimated trajectory (red line) and the ground truth trajectory (blue line).

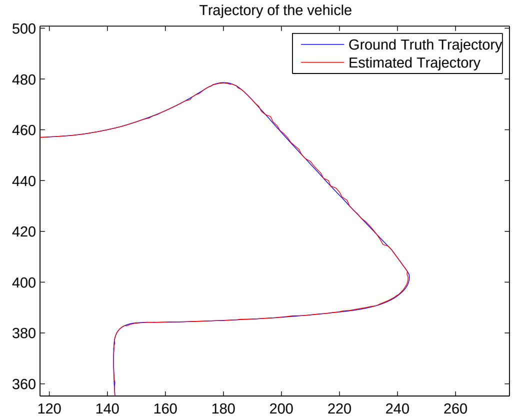

Figure 5: Zoom on a portion of the trajectory. The estimated trajectory is in red and the ground truth trajectory in blue.

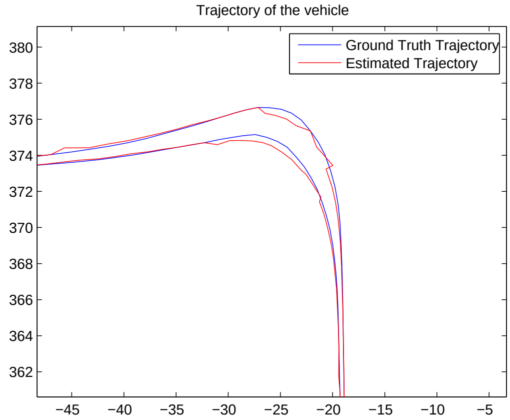

Figure 6: Zoom on a portion of the trajectory. The estimated trajectory is in red and the ground truth trajectory in blue.

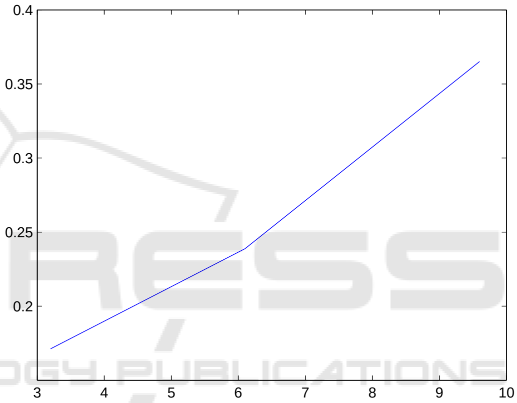

Figure 7: Evolution of the localization error with respect to the distance between two keyframes.

Table 3 and Figure 7 represent the error of the estimated trajectories when we decrease the threshold on the number of matches to determine whether an image is a keyframe or not. As we can see, the error grows slowly, and when there is only one image every 10 meters, the number of matches becomes insufficient to correctly estimate the camera's pose.

Table 3: Evolution of the mean error with respect to the distance between keyframes.

Mean distance between two keyframes (m)	3.2	6.1	9.6
Mean error of the estimated trajectory (m)	0.17	0.23	0.36

5 CONCLUSION AND FUTURE WORK

In this paper, we have presented a new method for the absolute localization of a vehicle using visual data. First, a selection of keyframes is made to reduce the complexity and the size of the database. Then, the keypoints of these keyframes are matched and triangulated to obtain 3D points. As our images are geolocalized, we obtain a map of geolocalized 3D points, their associated keypoints and descriptors. The map-building needs to be done only once, so this step can be carried out off-line. The localization is then achieved on-line, using the previously built map. For each frame from the camera mounted on the vehicle, the keypoints are detected, their descriptors computed and matched with the nearest keyframe. This provides the points in the image and their associated 3D points, so the pose of the camera can be found using a PnP approach. Our method has been evaluated on the KITTI dataset and gives precise results for the localization of the vehicle.

Our future works will focus on the problem of robustness to improve the results when changes appear between the acquisition of the map and the localization step. These changes can be due to light changes, season changes and/or appearance or disappearance of objects (cars, pedestrians, etc). This could be achieved for example by using multiple feature fusion.

ACKNOWLEDGEMENTS

This work was supported by the VIATIC project. VIATIC has been funded by the French Armaments Procurement Agency (DGA) and managed by the French National Research Agency (ANR) under the ASTRID-MATURATION 2014 call.

REFERENCES

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak : Fast retina keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, USA.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *Robotics and Automation Magazine*, 13(3):108–117.

Castle, R. and Murray, D. (2009). Object recognition and localization while tracking and mapping. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 179–180, Orlando, USA.

Comport, A., Malis, E., and Rives, P. (2010). Real-time quadrifocal visual odometry. *International Journal of Robotics Research (IJRR)*, 29(2-3):245–266.

Cummins, M. and Newman, P. (2008). Fab-map: probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research (IJRR)*, 27(6):647–665.

Cummins, M. and Newman, P. (2010). Highly scalable appearance-only slam fab-map 2.0. *International Journal of Robotics Research (IJRR)*.

Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17:229–241.

Dollar, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(4):743 – 761.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: Part i. *Robotics and Automation Magazine*, 13(2):99–110. ISSN: 1070-9932.

Fischler, M. and Bolles, R. (1981). Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages 381–395.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving ? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, USA.

Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(6):580–593.

Ho, K. and Newman, P. (2007). Detecting loop closure with scene sequences. *International Journal of Computer Vision (IJCV)*, 74(3):261–286.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan.

Kozak, K., Pohl, J., Birk, W., Greenberg, J., Artz, B., Blommer, M., Cathey, L., and Curry, R. (2006). Evaluation of lane departure warnings for drowsy drivers. In *Human Factors and ergonomics society*, pages 2400–2404.

Liu, Y., Lin, K., and Chen, Y. (2008). Birds-eye view vision system for vehicle surrounding monitoring. *Robot Vision*, 4931:207–218.

Maddern, W., Milford, M., and Wyeth, G. (2012). Cat-slam: probabilistic localisation and mapping using a continuous appearance-based trajectory. *International Journal of Robotics Research (IJRR)*, 31(4):429–451.

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics (JFR), Special issue on Space Robotics*, 24:169186.

Mei, C., Sibley, G., Cummins, M., Newman, P., and Reid, I. (2011). Rslam: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision (IJCV)*, 94(2):198–214.

Munguia, R. and Grau, A. (2007). Monocular slam for visual odometry. In *IEEE International Symposium on Intelligent Signal Processing*, pages 1–6.

Nister, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26:756–770.

Nister, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics (JFR)*, 23.

Rockl, M., Gacnik, J., and Schomerus, J. (2008). Integration of car-2-car communication as a virtual sensor in automotive sensor fusion for advanced driver assistance systems. In *FISITA 2008 World Automotive Congress*.

SAE (2015). Sae international. In *http://www.sae.org/*.