

Good Features to Map

Stephan H.G. ten Hagen

Faculty of Science, University of Amsterdam

Kruislaan 403, 1098 SJ Amsterdam

The Netherlands

stephanh@science.uva.nl

Abstract—In this paper visual homing is combined with an appearance model to obtain a navigation method that does not require estimates of the pose. The robot's current view is used to select a target from a collection of panoramic images that forms the map. This requires selecting features to find correspondences and in this paper some feature selection methods are evaluated to see which one is best suited for navigation.

I. INTRODUCTION

In mobile robot navigation the state of the robot is usually expressed as its pose (position and orientation) in some sort of global reference frame. This pose can be estimated from odometry, which is inaccurate on the long run, or from external sensors, such as range sensors or vision. In this case the robot needs an internal model of the environment. An explicit, geometric 'map' can be used to represent the environment, for example an occupancy grid or polygonal model. An alternative way is to use an appearance model that maps the current observation to a pose estimate. In [9] panoramic images are used to estimate the likelihood of being at a certain pose. The navigation process takes care that a path is planned and executed towards the desired pose.

Instead of expressing the desired state of the robot in the pose domain, the desired state can also be expressed in the image domain. In [2], [3] a robot arm navigates based on the current image and a target image. So here the destination is not given as a target pose. In each image trackable features are selected and using point correspondences and epipolar geometry the required change in camera pose can be computed. This method is called visual homing. When this method is applied to a mobile robot equipped with omni-directional vision, the robot is capable of moving to the pose where the image was taken. This idea can be extended by taking many potential target images and use them as way points in a navigation process. We can see these target images as a 'map' of the environment.

The purpose of this paper is to introduce a novel navigation method that is based on the combination of visual homing and an appearance model. Section 2 is about the action part of the navigation process and the requirements it places on the perception part. Section 3 explains how features can be selected that are good

to track. An empirical evaluation of the methods from section 3 is presented in section 4. Besides investigating which feature selection methods should be used, we also want to show that an implementation of our navigation method is feasible.

II. VISION AND NAVIGATION

A. Visual Navigation

Visual homing [2], [3] is a visual navigation method where the destination 'pose' is indicated by a target image. Suppose the robot currently observes image I and the target image I' is taken at the target pose. A point in the environment is projected onto point \mathbf{p}' in I' and on point \mathbf{p} in I . These points can be projected into homogeneous coordinates $\mathbf{h}_{\mathbf{p}'}$ and $\mathbf{h}_{\mathbf{p}}$. The following should hold:

$$\mathbf{h}_{\mathbf{p}'}^T \mathbf{E} \mathbf{h}_{\mathbf{p}} = 0. \quad (1)$$

Matrix $\mathbf{E} \in \mathbb{R}^{3 \times 3}$ is the essential matrix. This matrix is the product of the rotation matrix and the translation matrix, and so it encodes the six degrees of freedom of the camera motion, three for translation and three for rotation.

Points in I and I' are selected as features based on some criterion. Windows around these features in I are compared with windows in I' . If they are not too dissimilar the features form a corresponding pair. A true corresponding pair is when \mathbf{p}' and \mathbf{p} do belong to a true point in the environment and then (1) holds. When sufficient (at least 7) true corresponding pairs are found, matrix \mathbf{E} can be estimated by making sure that the right hand side of (1) is close to zero for all pairs.

It is possible that windows are very similar without the features belonging a real point in the environment. In that case (1) does not have to hold. From the two images alone it cannot be derived whether the pair is a false correspondence. For this reason robust estimation techniques [17] have to be used to reduce the influence of false correspondences on the estimation of the essential matrix.

When sufficient true correspondences are found and not too many false correspondences, then the direction D at which the target image was taken follows from the estimated essential matrix. Only the direction can be

estimated from two images and not the actual distance between the pose of the current image and the pose of the target. In [2], [3] also the rotation part from the essential matrix is used and [13] shows that asymptotically stable controllers exist for such configurations. Note that this is not the only 'pose free' visual navigation method. In [1] correlations between areas in the images are used to estimate the required change in pose. They show that a sequence of target images can be used to make the robot follow a path.

B. Pose Free Navigation

In an appearance model observations taken at different poses are stored, usually together *with* the poses. One example of an appearance model that does not require the pose is given in [5]. This is because the whole model consist of one panoramic image taken at a certain pose. This pose forms the global reference frame and the robot navigates by relating the current observation with the stored panoramic image. The panoramic images stored in [12] are associated with the nodes of a topological map. These nodes are called places and form a more abstract representation of the pose. Still the navigation is indirect by first estimating where the robot is and then plan the next place using the known spatial relation between the nodes of the map.

The navigation we propose is more direct and does not focus on the current location of the robot. Instead it focuses on where the robot can go to. Consider what happens when a robot with an omni-directional camera moves along a trajectory and stores images together with a time stamp. Just like in [1], these images form a sequence of target images. From all the stored images for which the direction D can be estimated the robot can pick the one with the oldest time stamp and the robot will follow a path.

When using a mobile robot with omni-directional vision the pose of the target images can be reached from every direction. Also the navigation to the target is less involved than in [2][13] when motion is restricted to 2D movements in a plane. For robots capable of turning around their axis it is enough to first rotate to face the target and then to move forward.

If the robot is in close proximity of the target then it is likely that windows around selected features match very well with those of the target image. The number of true correspondences will be high. Further away from the target the robot will observe elements in the environment differently and it becomes less likely to have enough true correspondences. In that case the robot cannot approach the target. Around each image stored we can think of an area A in which this image can be a target. Figure 1 shows a situation where the robot can choose between two targets. If it chooses target I_2 then eventually it will

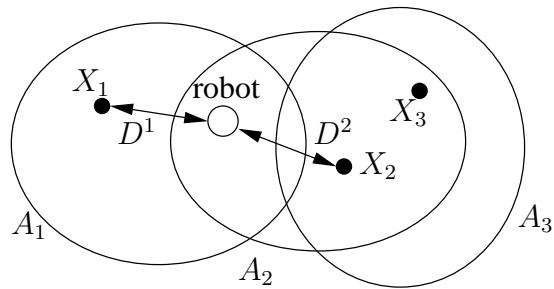


Fig. 1. The map $\mathcal{M} = \{I_1, I_2, I_3\}$ with images taken at pose X_1 , X_2 and X_3 . The robot (circle) is currently in area A_1 and A_2 so that directions D^1 and D^2 can be computed. The direction towards X_3 cannot be computed because the robot is outside A_3 .

enter the area A_3 and I_3 can be chosen as target. The pose of the target images are used as way points, but the robot can choose a new target when it enters a new area.

The selection of targets can be accomplished by assigning values to the stored images, just like the time stamps in the previously described example. By making the robot select targets with the highest values, navigation becomes a form of hill climbing. These values can be assigned by estimating the pose of the target images when the map is created, using a technique as in [14]. Then values can be assigned as function of the distance to the destination in the environment. It is also possible without estimating the poses by constructing an adjacency graph of all images. Two images are at distance one if they are in each others area. Values can be assigned as a function of the distance in the graph to the destination image. Assigning values can also be postponed until the robot is using these images. In [15] a method is presented to teach the robot to move to a certain destination by assigning values to targets.

The main benefit from navigation without estimating the pose is that it deals more natural with changing circumstances in the environment. When a robot has to navigate from the corridor into a room, then a closed door will block the view to target images in the room. The robot will stay in front of the door because it has reached the highest value. Opening the door results in a 'state transition' in sensor space and the robot will to move into the room. When estimating the pose first, images showing the open and closed door should result in the same pose estimate. An extra mechanism is required to prevent the robot from moving "through" the closed door. The same holds when the lights are switched off and it becomes dark. The robot cannot match the current images with a target so it will not move. A pose estimate will probably be wrong in this situation and it does not prevent the robot from moving.

There are a few requirements for the navigation:

- The need for collecting databases with new target images should be reduced and the robot should be

able to use the same set of images throughout the day. This implies that target selection and estimation of the directions D should be robust against small global variations like the light condition.

- The navigation assumes areas around poses of target images. If the robot is close to the target many true correspondences should be found. Far away the robot should not find too many true correspondences.

For the action part of the navigation, moving to target and assigning values, solutions exist. Now only the perception part that meets the above mentioned requirements is needed.

III. TRACKABLE FEATURES

The idea of feature tracking is to first select in an entire image features according to some criterion. After that around each feature a window is considered in which the feature in the next image will be sought. The criterion of feature selection is based on the sensitivity of the dissimilarity in a window around a point in the image. Let $W_{\mathbf{p}}$ be a window around pixel \mathbf{p} and let $I(\mathbf{x}, t) \in [0, 1]$ be the intensity at image location $\mathbf{x} = [u, v]^T$ relative to \mathbf{p} at time t . Also let $I'(\mathbf{x}, t + \tau)$ be the intensity from a second image taken τ seconds later. A dissimilarity measure $\varepsilon_{\mathbf{p}}$ for \mathbf{p} can be defined according to:

$$\varepsilon_{\mathbf{p}} = \iint |I'(\mathbf{x}, t + \tau) - I(\mathbf{x}, t)|^2 W_{\mathbf{p}}(\mathbf{x}) d\mathbf{x}. \quad (2)$$

We take $W_{\mathbf{p}}(\mathbf{x}) = 1$ if $\|\mathbf{x} - \mathbf{p}\|_1 < n_w$ and $W_{\mathbf{p}}(\mathbf{x}) = 0$ otherwise, so the value is one inside a rectangular window with size n_w centered around \mathbf{p} and the value is zero outside.

The dissimilarity (2) is defined for two different images, but to select suitable features only the current image I is available. The idea is to express image I' as a distorted version of I , where the distortion is parameterized by δ . In the Lucas-Kanade feature tracker [10], [16] the image I' is displaced compared to the original image I :

$$I'(\mathbf{x}, t + \tau) = I(\mathbf{x} + \mathbf{d}, t + \tau). \quad (3)$$

Displacement $\mathbf{d}^T = [d_u \ d_v]^T$ forms distortion δ . A first order Taylor expansion gives:

$$I(\mathbf{x} + \mathbf{d}, t + \tau) \approx I(\mathbf{x}, t) + \nabla_{\delta} I(\mathbf{x}, t)^T \delta + \nabla_{\tau} I(\mathbf{x}, t) \tau. \quad (4)$$

Here the constant part $I(\mathbf{x}, t)$ cancels the original image in (2) and the dissimilarity only depends on the gradients $\nabla_{\delta} I(\mathbf{x}, t)^T = [I_u \ I_v]^T = \left[\frac{\partial I(\mathbf{x} + \mathbf{d}, t + \tau)}{\partial d_u}(\mathbf{x}, t) \quad \frac{\partial I(\mathbf{x} + \mathbf{d}, t + \tau)}{\partial d_v}(\mathbf{x}, t) \right]$ and $\nabla_{\tau} I(\mathbf{x}, t) = I_t = \frac{\partial I(\mathbf{x} + \mathbf{d}, t + \tau)}{\partial \tau}(\mathbf{x}, t)$.

The task of tracking is to estimate the displacement \mathbf{d} , which is the value of δ for which (2) is minimal. This can be computed by setting $\nabla_{\delta} \varepsilon_{\mathbf{p}} = \mathbf{0}$, resulting in

$$\mathbf{C}_{\mathbf{p}} \delta + \mathbf{g}_{\mathbf{p}} = \mathbf{0}, \quad (5)$$

with

$$\mathbf{C}_{\mathbf{p}} = \iint \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix} W_{\mathbf{p}}(\mathbf{x}) d\mathbf{x}, \quad (6)$$

$$\mathbf{g}_{\mathbf{p}} = -\tau \iint I_t \begin{bmatrix} I_u \\ I_v \end{bmatrix} W_{\mathbf{p}}(\mathbf{x}) d\mathbf{x}. \quad (7)$$

Using (5) δ is given by

$$\delta = -\mathbf{C}_{\mathbf{p}}^{-1} \mathbf{g}_{\mathbf{p}}. \quad (8)$$

The $\mathbf{g}_{\mathbf{p}}$ cannot be computed from one image because I_t represents the change in dissimilarity not modeled by δ and τ is not known. Still, we see that the computation requires the inverse of $\mathbf{C}_{\mathbf{p}}$, so $\mathbf{C}_{\mathbf{p}}$ should be well conditioned with respect to inversion. In other words, the minimal eigenvalue of $\mathbf{C}_{\mathbf{p}}$ should be sufficiently large. This can be used as a measure of goodness $f_{\mathbf{p}}$ for tracking point \mathbf{p} :

$$f_{\mathbf{p}} = \lambda_{\min}(\mathbf{C}_{\mathbf{p}}). \quad (9)$$

Good features can be selected by introducing a threshold f_T and select those points \mathbf{p} as feature for which $f_{\mathbf{p}} > f_T$.

In [11] the distortion was modeled as a affine transformation:

$$I'(\mathbf{x}, t + \tau) = I(\mathbf{A}\mathbf{x} + \mathbf{d}, t + \tau). \quad (10)$$

with $\delta = [a_{uu} \ a_{uv} \ a_{vu} \ a_{vv} \ d_u \ d_v]^T$, and a_{uu} , a_{uv} , a_{vu} and a_{vv} the elements from \mathbf{A} . Solving $\nabla_{\delta} \varepsilon_{\mathbf{p}} = \mathbf{0}$ results in a $\mathbf{C}_{\mathbf{p}} \in \mathbb{R}^{6 \times 6}$ containing products of I_u , I_v , u and v (see [11] for the complete matrix). Again the minimal eigenvalue is used to select the features. This selection is computational more involved and is only used as a good initialization, after which the features are selected using (3).

To track features for even larger τ with respect to the first image, feature selection that takes into account illumination have been considered [8]:

$$I'(\mathbf{x}, t + \tau) = \mu I(\mathbf{A}\mathbf{x} + \mathbf{d}, t + \tau) + \nu, \quad (11)$$

where ν represents the brightness and μ the contrast. The distortion becomes $\delta = [a_{uu} \ a_{uv} \ a_{vu} \ a_{vv} \ d_u \ d_v \ \nu \ \mu]^T$ and $\mathbf{C}_{\mathbf{p}}$ is a 8 by 8 matrix. There are other possibilities to deal with illumination, [6] considers normalization of intensities in the windows and [7] introduces a more explicit modeling of the reflection of light.

For completeness we introduce

$$I'(\mathbf{x}, t + \tau) = \mu I(\mathbf{x} + \mathbf{d}, t + \tau) + \nu, \quad (12)$$

with $\delta = [d_u \ d_v \ \nu \ \mu]^T$.

These feature selection methods find windows in the images that are insensitive to certain changes. In our navigation method the robot has to compare the current image with those taken at different poses, so we need features that are insensitive to changes in camera pose.

We assume that features that are good to track at one pose are also good to track at another pose, so that the same features are selected and matched.

IV. EXPERIMENT

A. Setup

Our robot is equipped with a camera looking up to a hyperbolic mirror. The images are mapped to a cylinder to form gray-scaled panoramic images (see [4] for details). We placed the robot at known poses on a grid with 7 rows 1.0 meter apart and 5 columns 0.5 meter apart. A panoramic image was created for all these poses. For an 11 by 11 window around each pixel \mathbf{p} in all images we computed $\lambda_{\min}(\mathbf{C}_{\mathbf{p}})$ to find $f_{\mathbf{p}}$ using the methods described in the previous section. We will refer to the result of using equation (3)(10)(11) and (12) respectively as features A, B, C and D. The maxima of $f_{\mathbf{p}}$ were used as features, but maxima within a 21 by 21 window around a higher value were rejected. This was to prevent having too many features at the same part of the image. We ranked the feature based on $f_{\mathbf{p}}$ starting with the highest value $f_{\mathbf{p},\max}$ and selected all features for which $f_p > 0.2f_{\mathbf{p},\max}$.

B. Light

Purpose of this experiment is to investigate how different light condition introduce of remove selected features. We created sets of images on the grid under three different light conditions I, II and III. Figure 2 shows that set I was created will all lights switch on, set II only has light from behind and III has the lights on the left switched on.

We selected features using the four methods on all images. Images were taken at the same poses, so features should be found at the same positions in the image. Features within a 2 pixel distance were considered the same to take into account the possible slight differences in pose. We compared features in I with those in II and III and counted how many features were left. We also divided the rank of the features in II and III by their rank in I and call the average of this the Index Quotient. This indicates how many higher ranked features in I were not found in II and III. If this value would be 0.5 then the 10th ranked feature in II or III would be the 20th ranked feature in I. Unlike a tracking application, we can choose to select more features for the target images to increase the probability of finding corresponding pairs under different light conditions.

Table I shows the results. We see in table I(a) that the number of remaining features decreases when going from row 1 (light) to row 5 (dark). Row 6 and 7 are not shown because only a few features remained. We see also that a larger number of features remained for method C and D. This is because these methods select about twice as many features as methods A and B. Still, close to the darkness in row 5 we see that the number of remaining



(a) I: All Lights



(b) II: Light Behind



(c) III: Light Left

Fig. 2. Panorama images from the center of the grid under different light conditions.

features is about the same for all methods. The surprising result in table I(b) is that only a few features remain for column 4 while enough features remain for column 5. This is because images in column 5 were taken closer to the wall and more features were found in condition I. When looking at the Index Quotient we see that most values are above 0.5.

Result: All methods perform equally well in dealing with darkness. For the target images it is sufficient to select twice as much features as the robot selects from its current image. Then the robot can deal with changing light conditions. More than twice is not necessary because then the situation would be so dark that hardly any good features are found.

C. Distance

Purpose of this experiment is to investigate the influence of the distance to the target images on the number of true and false correspondences. One of the requirements in section 2 is that there are many true correspondences at a close distance and only a few at a far distance. We created the corresponding pairs by computing the dissimilarity (2) and the normalized dissimilarity (analog to [6]) between windows around the features. If the dissimilarity is below a threshold features were considered a corresponding pair.

Based on the corresponding pairs the essential matrix \mathbf{E} should be estimated. In this case we know the poses, so we can compute the essential matrix. In fact, this was very easy because all images were taken with the same orientation making the rotation matrix a unit matrix. Also

	row 1	row 3	row 5	IQ min	IQ max
A	26	21	10	0.55	0.80
B	22	18	8	0.60	0.79
C	43	27	11	0.58	0.68
D	47	30	10	0.49	0.69

(a) I→II

	col 3	col 4	col 5	IQ min	IQ max
A	16	5	13	0.67	0.77
B	12	3	9	0.49	0.72
C	23	6	18	0.62	0.82
D	33	8	23	0.63	0.81

(b) I→III

TABLE I

NUMBER OF FEATURES LEFT WHEN GOING FROM I TO II OR III. THE ROW AND COL INDICATE THE MEAN OVER THE ROWS OR COLUMNS ROUNDED TO THE CLOSEST INTEGER. THE IQ MIN AND IQ MAX ARE THE MINIMUM AND MAXIMUM VALUE OF THE INDEX QUOTIENT.

the difference in pose was restricted to changes in the x and y direction. So \mathbf{E} was given by

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & \Delta y \\ 0 & 0 & -\Delta x \\ -\Delta y & \Delta x & 0 \end{bmatrix}. \quad (13)$$

Now all corresponding pair can be projected into homogeneous coordinates and (1) can be used to find the true correspondences. If the left hand side of (1) is less than 0.03 for a given pair, we consider it a true corresponding pair.

We took the first column of data set I and computed the number of true correspondences for the poses at 1 to 5 meters distance. We only used one column because there are some differences between columns. The features of images near the center of the grid belong to points in the environment that are relatively far away and do not change much when the camera position changes. So areas A for images near the center are larger than for those at the border.

Table II shows the fraction of the true correspondences with respect to all pairs found. First thing to notice is that there is hardly any difference in using the dissimilarity or the normalized dissimilarity. At one meter distance all methods result in a situation where two out of three pairs are true correspondences. In these situations the directions D can be computed, so one part of the requirement from section 2 is fulfilled. At two meters distance the fraction for methods A and B already drop below a level where it is unlikely that D can be computed. The fractions for method C and D do not drop that much, so there is still a change that D can be computed. The second part of the requirements is that D can *not* be computed at a large

	1m	2m	3m	4m	5m
A	0.70	0.29	0.26	0.10	0.17
B	0.61	0.26	0.17	0.19	0.15
C	0.67	0.51	0.38	0.23	0.15
D	0.67	0.36	0.42	0.17	0.20

(a) Dissimilarity

	1m	2m	3m	4m	5m
A	0.62	0.24	0.26	0.11	0.19
B	0.70	0.27	0.12	0.13	0.19
C	0.66	0.42	0.34	0.20	0.13
D	0.65	0.38	0.42	0.14	0.17

(b) Normalized Dissimilarity

TABLE II

NUMBER OF TRUE CORRESPONDENCES DIVIDED BY TOTAL NUMBER OF CORRESPONDENCES FOR DIFFERENT DISTANCES AND METHODS.

distance, and all methods meet this requirement. Note that results of A resemble that of B and results of C resemble that of D. Considering an affine transformation in (10) and (11) does not lead to an increase in performance.

Result: All methods agree with the requirement from section 2. The direction D can be computed for one meter distance, but it cannot be computed for much larger distances. The fraction of true correspondences drop faster for method A and B and so these methods create a sharper boundary for the area A around the images. The extra computational effort for affine transformations does not result in a better performance.

V. CONCLUSION

In this paper a pose free navigation method was introduced that is based on the combination of visual homing and an appearance model. Certain requirements were placed on the feature selection methods for the navigation to work. All methods investigated performed equally well when confronted with a change in light conditions. To deal with such situations it is sufficient to select twice as many features from images that form the model. All methods investigated resulted in an area around the images in which sufficient correspondences are found for the visual homing. The areas of the illumination methods have a less sharp boundary. The affine transformation methods do not perform better than the computationally cheaper displacement methods.

Main conclusion is that all requirements are met and that the proposed navigation method is feasible. The next step is to test this navigation method on a robot.

VI. ACKNOWLEDGEMENT

This research is supported by the Netherlands Organization for Scientific Research (NWO).

VII. REFERENCES

- [1] C.S. Andersen, S.D. Jones, and J.L. Crowley. Appearance based processes for visual navigation. In *Proc. of the 5th Int. Symposium on Intelligent Robotic Systems (SIRS)*, pages 227–236, 1997.
- [2] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 863–869, 1998.
- [3] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. *Int. Journal of Computer Vision (IJCV)*, 33(2):1–21, September 1999.
- [4] R. Bunschoten and B.J.A. Kröse. 3-D scene reconstruction from cylindrical panoramic images. *Robotics and Autonomous Systems (special issue)*, 41(2/3):111–118, November 2002.
- [5] D. Cobzas and H. Zhang. Cylindrical panoramic image-based model for robot localization. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1924–1930, 2001.
- [6] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2(4):312–320, 1999.
- [7] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(10), 1998.
- [8] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *Proc. Int. Conference on Computer Vision (ICCV01)*, pages 684–689, 2001.
- [9] B.J.A. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, April 2001.
- [10] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.
- [11] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.
- [12] H.D. Tagare, D. McDermott, and H. Xong. Visual place recognition for autonomous robots. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1998.
- [13] C.J. Taylor and J.P. Ostrowski. Robust vision-based pose control. In *IEEE Conference on Robotics and Automation (ICRA)*, pages 2734–2740, 2000.
- [14] S.H.G. ten Hagen and B.J.A. Kröse. Towards global consistent pose estimation from images. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 466–471, Lausanne, Switzerland, September 2002.
- [15] S.H.G. ten Hagen and B.J.A. Kröse. Learning to navigate using a lazy map. In A.T. de Almeida and U. Nunes, editors, *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, pages 299–304, Coimbra, Portugal, June/July 2003.
- [16] C. Tomasi and T. Kanade. Detecting and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.
- [17] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. Journal of Computer Vision (IJCV)*, 1997.