Image-Based Localisation

Roberto Cipolla, Duncan Robertson and Ben Tordoff
Department of Engineering, University of Cambridge
Cambridge, CB2 1PZ, UK

Abstract. This paper describes a system for retrieving information about an urban scene using a single image from a mobile device such as a camera-phone or PDA. The captured image is sent to a service provider where it is compared to an existing database of views using a novel wide baseline matching algorithm. The best matching view is then used to recover both details of the viewed building and the precise location of the user.

1 Introduction

Recovering information about an urban location and the position and orientation of the user is a task with obvious practical benefits. By sending a single image of a nearby building to the service provider the user can discover their exact location and orientation presented on standard maps, as well as detailed information about the building façade being viewed. For historical sites information about the age, architect and purpose of the building may be presented, and for commercial premises sales or contact information might be available. Many current mobile phone and personal digital assistant (PDA) devices are already equipped with both high resolution digital image capture abilities and the means to communicate with a service provider, making this system immediately applicable.

Whereas current GPS or network-cell based methods can give only positional information, a singe image is sufficient to determine both position and orientation with great accuracy. In addition, this approach is most appropriate in exactly those situations where GPS and cell systems become least accurate — in town and city centres and inside buildings. Where GPS or cell information is available it can be used to greatly reduce the search for matching façades.

1.1 Relation to previous research

The system described here is largely synoptic, bringing together several strands of computer vision research to achieve image matching and user localisation, making full use of the geometric information provided by typical urban scenes.

The problem of determining absolute position and orientation has been considered in detail in the context of mobile robot navigation. The ubiquitous technique of Simultaneous Localisation and Mapping (SLAM) (see eg. [22, 6, 2, 23, 17]) updates both robot pose and a 3D map of the environment relative to the initial position. Such schemes are inherently continuous and there is no general way to re-initialise the robot from an existing map if, for instance, it is unable to observe data for an extended period of time or it is switched on in an unknown location — the "kidnapped robot" problem [18].

Solutions for the "kidnapped robot" have been varied, but all rely on observing recognisable features in known locations. In vision based navigation, such features are either deliberately placed fiducial markers [16] or distinctive scene points that have been accurately registered to a map [18]. In either case the initialisation problem then reduces to one of data association, finding one or more landmarks, followed by pose-estimation relative to them. The absolute 3D

location of all landmark points must be known and in many environments (such as city centres) adding fiducial markers to a scene is not possible.

Se *et al* [18] re-recognise visual landmarks using natural features detected using Lowe's SIFT operator [13]. Matching these and similar features for re-recognising parts of scenes is the subject of much study in the context of retrieving images from large databases [15, 19, 21]. Each feature is described by a characteristic descriptor based on the surrounding image, and descriptors are often chosen to be invariant to lighting changes, scale, rotation or even affine changes of the image. These invariants improve the redetection rate when the view point is changed.

Whilst Shao *et al* [19] recover images of buildings from a database using feature matching alone, the geometry of urban scenes provides much richer information which can aid the search, as in [3]. Given a view of a building, Coorg and Teller [4] used vertical façades to determine the orientation of buildings, and Košecká and Zhang [10] demonstrated how such façades can provide accurate camera orientation. Here this approach is extended to also provide positional information.

1.2 New work

Extending our previous work [9], here the "kidnapped robot" problem is solved using the dominant planar façades of buildings as landmarks. Comparing rectified façades greatly constrains the matching problem when compared to more general approaches. As described below, once rectified, manual alignment of a façade to a map allows recovery of both orientation and position. The overall scheme is

1. Rectify: warp both query view and database view to be fronto-parallel.
2. Match: compare features between query and database views.
3. Localise: from the best match calculate camera position and orientation.

The method of image rectification to obtain *canonical views* is described below, followed in section 3 by the feature detection and matching stages. Section 4 describes the method of registering the database view to a map, and the subsequent calculation that localises the query view. Some early results and discussion are presented in section 5 before conclusions are drawn.

2 Image rectification

Nearly all building façades are predominantly planar and mainly due to gravity and human preference also contain mostly vertical and horizontal features. By determining the orientation of the camera with respect to this plane, views may be transformed into a canonical frame by *rectification* [12].

Camera orientation is determined using the vanishing points belonging to the principal horizontal and vertical directions of the façade. Whilst these can be determined directly from image statistics [5], for buildings it is more computationally efficient to use only straight line segments. Edge elements (edgels) are detected using the method of Canny [1], and grouped into straight line segments in linear time using a "merge" algorithm (eg. [20]), as in figure 1.

Finding vanishing points is a two stage process. In order to estimate a vanishing point the set of lines which should pass through it must be identified. However, identifying which lines are horizontal, vertical or clutter requires an estimate of the vanishing points. A coarse initialisation is obtained by testing an evenly spaced set of possible vanishing directions and measuring the

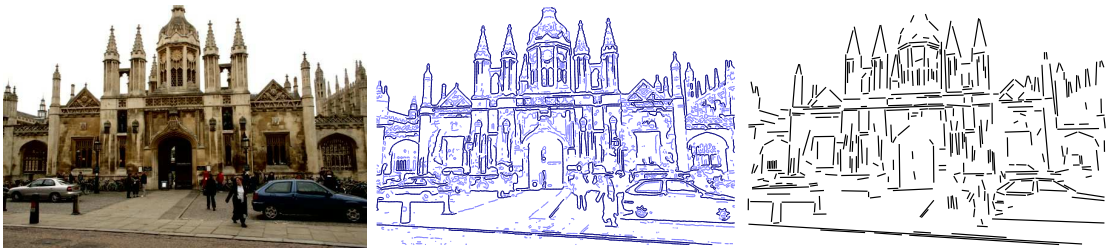

Figure 1. A building façade (left), the edgels output from Canny (middle) and the resulting straight line-segments (right).

support from the image lines. The maxima of this discrete global search must then be iteratively refined.

An efficient method for iteratively estimating both the vanishing points and which lines belong to them using expectation maximisation (EM) was described by Coorg and Teller [4], but in the form stated suffers poor convergence. Košecká and Zhang [10] overcome this by gradually relaxing the variance parameters rather than attempting to estimate them. Here we avoid this problem by using a general non-linear minimiser [11, 14] to give stable convergence and the ability to simultaneously remove radial distortion.

Having determined the two strongest vanishing directions, they must be assigned to the vertical and horizontal world directions. A simple but effective method is to just use their positions relative to the image centre — the vanishing point closest to the image vertical is assumed to represent the world vertical. When the image could be in an arbitrary orientation it is essential to have a more robust method, and higher level cues must be used (eg. finding the sky).

In homogeneous coordinates the normalised positions of the vanishing points are

$$\mathrm{v}_h = \mathtt{K}^{-1}\tilde{\mathrm{v}}_h \qquad \mathrm{v}_v = \mathtt{K}^{-1}\tilde{\mathrm{v}}_v$$

where $\tilde{\mathrm{v}}_h$, $\tilde{\mathrm{v}}_v$ are the pixel positions and $\mathtt{K}$ the camera calibration. In the canonical view horizontal lines become parallel to the image x-axis, and vertical to the image y-axis, giving infinite vanishing points at $[1,0,0]^\top$ and $[0,1,0]^\top$ respectively. This can be achieved using a 3D rotation around the camera centre

$$\mathtt{R}_\perp \left[\lambda_h \mathrm{v}_h \;\; \lambda_v \mathrm{v}_v\right] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad ,$$

where λ_h and λ_v are unknown scale factors and $\mathtt{R}_\perp$ is the rectifying rotation, all recoverable from this equation. The sole remaining step is to choose a calibration for the canonic view $\mathtt{K}_\perp$ such that it projects to an image whose resolution is as close as possible to the original. Here we use a calibration that preserves the area of the central image pixel — minimising distortion of pixels near to the image centre where the most salient information is likely to be. Points (ie. pixels) in the canonic view $\mathrm{p}_\perp$ are then related to the those in the original view p by the rectifying homography

$$\mathrm{p}_\perp = \mathtt{K}_\perp \mathtt{R}_\perp \mathtt{K}^{-1}\mathrm{p} = \mathtt{H}_\perp \mathrm{p} \tag{1}$$

where $\mathtt{H}_\perp$ is the 3×3 rectifying homography (an infinity homography). Examples of raw and canonical (rectified) views are shown in figure 2. Also shown in each canonical view is its horizon line, defined as the intersection of the horizontal plane containing the camera with the rectified image — the camera's "eye level".

Figure 2. Façades before (top) and after (bottom) rectification.

3 Matching canonical views

Between canonical views, a building facade will be related by an isotropic scaling α plus an image translation $(t_x, t_y)^\top$. Furthermore, knowledge of the horizon lines in both views removes the y-translation leaving a two degree of freedom relationship[1]

$$\mathrm{p}'_\perp = \begin{bmatrix} \alpha & 0 & t_x \\ 0 & \alpha & h' - \alpha h \\ 0 & 0 & 1 \end{bmatrix} \mathrm{p}_\perp = \mathtt{H}_m \mathrm{p}_\perp \tag{2}$$

where h, h' are the y-coordinates of the horizon lines in the two views, and $\mathtt{H}_m$ is the matching homography.

Although any matching strategy could be used (homography. [15, 19, 21]), the relationship between the images makes rotationally or affine invariant feature matching unnecessary, and allows a particularly simple scheme. In particular, instead of matching features or regions between views then imposing the geometric relationship to remove erroneous matches, we can search over the two parameters of the relationship using the matches to score each test.

3.1 Feature detection and characterisation

For matching we desire features that are well localised and can be found repeatedly. On the façade the canonical views do not suffer any perspective distortion, so a feature detector which is affine or perspective invariant is undesirable. Here we use the Harris-Stephens detector [8] to find image locations where the image autocorrelation function is most sharply defined. Other feature detectors could be substituted with minimal impact on the overall system.

The features are characterised by a *descriptor* based on the surrounding image. Since rotation and perspective distortion have been removed, the descriptor should vary with rotation, scale and other distortions — the exact opposite of the invariant schemes commonly used [13]. Although such a descriptor is the topic of current research, a straightforward method is to use the patch of the image surrounding the feature.

3.2 Matching by search

Since the feature detector and descriptors being used are sensitive to scale changes, the detection process must be repeated at a range of scales for both views being compared, as in [7]. Image pyramids are formed covering two complete scale octaves with three levels per octave, and features detected for each level. For negative scales ($\alpha < 0$), features from the largest level of the first image ($\mathrm{p}_\perp$) are compared with lower levels of the second ($\mathrm{p}'_\perp$), and for positive scales the converse. This allows for differences in the size of the rectified façade of up to four times, allowing for both differences in image resolution and in the distance of the user from the building.

[1] when matching arbitrary images there are usually seven

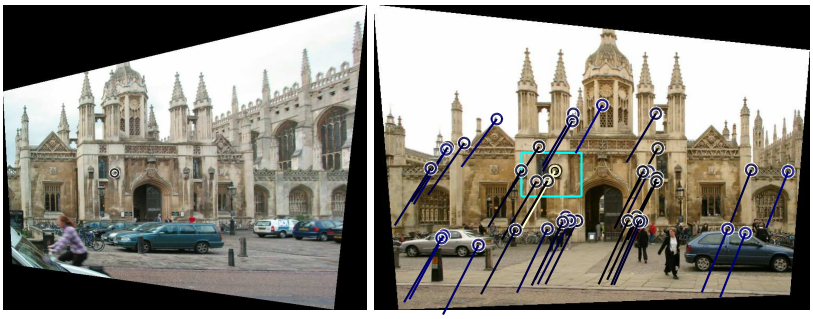

Figure 3. Features detected in two levels being compared. For the highlighted feature in image 1 (left), the search region and matching feature for a particular scale and translation are shown in image 2 (right).

At each scale level, vertical bounds on the position of a matching feature in the second image are determined from the inter-level spacing . Assuming that the images are captured at similar heights, the horizons should be aligned, so that the scale bounds translate into bounds on the y-coordinate proportional to the distance from the horizon line. Similarly, if t_x is varied in discrete steps, the step size dictates bounds on the possible x-coordinate of the match, giving a rectangular region that must be searched for a match (as in figure 3).

Even though the search is constrained by the scale and translation bounds, a number of features may lie within the search region. To decide between them each potential correspondence is scored using zero normalised cross-correlation. The zero-normalisation provides invariance to changes in lighting and a degree of contrast and colour variation.

The correlation score and ambiguity are used to derive the likelihood of having chosen the correct match. These likelihoods are then used to weight a robust estimation of the scale-translation transformation in the guided sampling and consensus scheme outlined in [24]. The output of this estimator is the transform with maximum likelihood and the posterior probability of each match being correct. Summing this posterior measures the expected number of correct matches.

In practise it is not necessary to run the robust estimation for every step of the search as the constrained matching returns a larger number of potential matches when correctly aligned. Instead the robust estimator is used on the strongest translation step for each tested scale and the best scale-translation combination kept as the match score. This process is repeated for each database image and the best matching image chosen for subsequent use in localisation.

4 Localisation

In order to determine the absolute position and orientation of the user it is first necessary to register the database views with a map. By rectifying the images this becomes a simple task, since the image lies in a vertical plane in the world and the map is a horizontal projection of the world. Vertical lines in the rectified image correspond to vertical lines in the world such that the user need only indicate the x-coordinate of the verticals which bound the façade. On the map these vertical lines project to points which are also manually chosen as in figure 4. The registration process therefore requires just four mouse clicks.

The intersection of the horizon line ($v = h$) with the left extent of the façade is $\mathrm{x}_l = (u_l, h)^\top$, and with the right $\mathrm{x}_r = (u_r, h)^\top$. The equivalent map points lie on the world X-Z plane $\mathrm{X}_l = (X_l, 0, Z_l)^\top$ and $\mathrm{X}_r = (X_r, 0, Z_r)^\top$, so that the relationship between image coordinates $(u, v)^\top$ and map coordinates $(X, 0, Z)^\top$ is an isotropic scaling β, a rotation by θ_y about the world Y-axis and a 3D translation T_w

$$\mathrm{X} = \mathtt{R}_w \mathtt{B} \mathrm{x} + \mathrm{T}_w \quad \text{where} \quad \mathtt{B} = \begin{bmatrix} \beta & 0 \\ 0 & \beta \\ 0 & 0 \end{bmatrix} \quad , \quad \mathtt{R}_w = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \quad ,$$

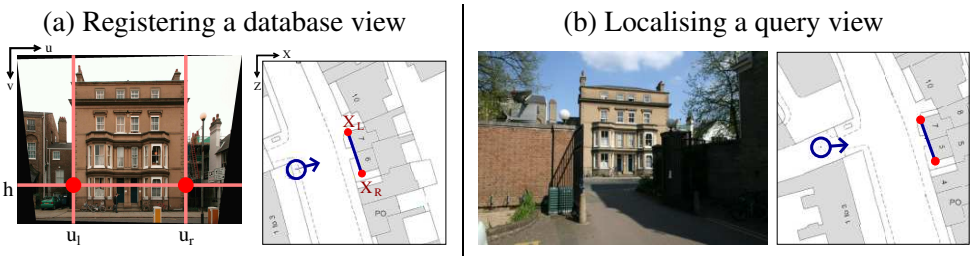

Figure 4. (a) Manually identifying the bounds of the façade (red dots) determines the registration. (b) A query view matched successfully with a registered database view allows recovery of the user position and orientation.

where the parameters β, θ_y and T are recovered as

$$\beta = \frac{|\mathrm{X}_r - \mathrm{X}_l|_2}{|\mathrm{x}_r - \mathrm{x}_l|_2} \qquad \theta_y = \tan^{-1}\frac{Z_r - Z_l}{X_r - X_l} \qquad \mathrm{T}_w = \begin{pmatrix} X_l - \beta x_l \cos\theta_y \\ -\beta h \\ Z_l + \beta x_l \sin\theta_y \end{pmatrix} \quad . \tag{3}$$

To determine the camera position for the database view the principal point of the *original image*, $\mathrm{c} = (u_0, v_0)^\top$, is first transferred homogeneously into the rectified view $\mathrm{c}_\perp = \mathtt{H}_\perp \mathrm{c}$, then transferred inhomogeneously into the map $\mathrm{C}_w = \mathtt{R}_w \mathtt{B} \mathrm{c}_\perp + \mathrm{T}_w$. The image normal direction is affected only by rotations $\hat{\mathrm{n}}_w = \mathtt{R}_w \mathtt{R}_\perp (0,0,-1)^\top$ and the focal length only by scale $f_w = \beta f_\perp$ where $f_\perp$ is the focal length chosen for the canonical view. Moving a distance f_w along the normal direction from the principal point cw gives the camera location as in 4c.

4.1 Localising the user

Having matched a query image to a registered database image, it is now straightforward to calculate the position of the user (ie. the camera position) by combining equations 1, 2 and 3. As described above, the three required quantities are the principal point map coordinates, the normal direction and the effective focal length. For the query image the match transformation must also be taken into account, giving

$$\mathrm{c}_\perp = \mathtt{H}_m \mathtt{H}_\perp \mathrm{c} \qquad \hat{\mathrm{n}}_w = \mathtt{R}_w \mathtt{R}_\perp (0,0,-1)^\top \qquad f_w = \alpha\beta f_\perp \quad .$$

An example is shown in figure 4. An alternative is to express the projection of homogeneous world (map) points, X, onto homogeneous points in the original image, x, as a single homogeneous equation $\mathrm{x} = \mathtt{P}\mathrm{X}$. In addition to the information used above, this requires the use of the calibration of the rectified database view $\mathtt{K}_{\perp db}$, its principal point (u_0, v_0) and focal length f:

$$\mathtt{P} = \mathtt{H}_\perp^{-1} \mathtt{H}_m^{-1} \mathtt{K}_{\perp db} \begin{bmatrix} \beta \mathtt{R}_w^\top & -\beta \mathtt{R}_w^\top \mathrm{T}_w - \begin{bmatrix} u_0 \\ v_0 \\ -f \end{bmatrix} \end{bmatrix} \quad .$$

Decomposing this projection matrix using the known calibration of the original image provides the pose of the user.

5 Evaluation

To test this system, a database was constructed from photographs of all the buildings in the main shopping street in Cambridge city centre, as well as a number of other buildings of interest around the city. These 200 views span around 2 km of building facades, and the area covered is several times greater than the typical mobile phone cell (which could be used to restrict the search).

Images were captured at different times of day and query views were obtained from a variety of viewpoints and distances from the façade. Usually distance differed by at least 30% and orientation by at

Figure 5. Typical results. The top row shows the query view with a region of interest manually outlined. The same region is projected onto the database view (bottom row) to illustrate the accuracy of the matching. In the final column the correct building has been found, but the recovered $\mathtt{H}_m$ is incorrect due to the repetitive nature of the façade.

least 30°. Many of the query images contained significant clutter (pedestrians, traffic, etc.) as is typical in a city centre environment.

Each comparison between the query data and a database view took around 100ms on a 3 GHz desktop PC, giving an overall time per query of around 20s (although our present implementation is not especially efficient). Pose determination results were verified by sketching building facade outlines in the database views and projecting them into the query views using the recovered homographies (since camera focal length is known, this method gives a good indication of the accuracy of the pose estimates). Overall, 93 out of 97 queries were registered correctly. Representative results are shown in Figure 5. Because of our robust matching strategy, usually only one or two matches are found between a query view and incorrect database views. When the system did fail, this was because the database contains some photographs of buildings (or parts of buildings) that are virtually identical to each other.

6 Conclusions

A prototype system has been described which identification of a building and recovery of absolute pose from a single image. This allows the user to navigate in an urban environment using a mobile telephone or PDA equipped with a camera. The system requires only limited computation on the actual device since the query can be sent to a central service provider at minimal cost.

In this work a database of views of the environment is captured and stored beforehand and each view registered to a map. The registration process requires manual identification of the vertical bounds of the façade in both the rectified view and on a map in a "four click" process.

One limitation is that some buildings (and parts of buildings) are very similar. This means that the system might be unable to distinguish between some viewpoints without more information, eg. extra query views. Another limitation is that conducting two-view matching between the query view and every nearby database view is slow. A more efficient strategy might be to use more 'global' image properties such as dominant colours to eliminate unlikely database views in advance.

In this paper, camera internal parameters have been assumed known. Whilst measuring the calibration of the camera used to capture the database is reasonable it is far more onerous having to know the calibration of the query device. In recent work, we have explored computing the focal length and radial distortion for the query view automatically using vanishing points. Where such calibration is not possible an extra parameter can be introduced into the matching search.

We are also exploring the possibility of acquiring and registering database views automatically using a camera attached to a moving vehicle. Using an inertial car navigation system, it should be possible to register views automatically in the world coordinate system, and hence onto standard maps.

Acknowledgements
This work was supported by the Cambridge-MIT Initiative.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), November 1986.

[2] J.A. Castellanos. *Mobile robot localization and map building: a multisensor fusion approach*. PhD thesis, University of Zaragoza, 1998.

[3] R. Collins and J.R. Beveridge. Matching perspective views of coplanar structures using projective unwarping and similarity matching. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition*, pages 240–245, 1993.

[4] S. Coorg and S. Teller. Extracting textured vertical facades from contolled close-range imagery. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition*, pages 625–632, 1999.

[5] J.M. Coughlan and A.L. Yuille. Manhattan world. *Neural Computation*, 2003.

[6] A.J. Davison and D.W. Murray. Sequential localisation and map-building using active vision. In *Proc 5th European Conf on Computer Vision, Freiburg, Germany*, pages 809–825, 1998.

[7] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition*, pages 612–618, 2000.

[8] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proc 4th Alvey Vision Conf, Manchester*, pages 147–151, 1988.

[9] B. Johansson and R. Cipolla. A system for automatic pose-estimation from a single image in a city scene. In *International Conference on Signal Processing, Pattern Recognition and Applications*, 2002.

[10] J. Košecká and W. Zhang. Video compass. In *Proc 7th European Conf on Computer Vision, Copenhagen*, pages 476–490, June 2002.

[11] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[12] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition*, pages 482–488, 1998.

[13] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc 7th Int Conf on Computer Vision, Kerkyra*, pages 1150–1157, 1999.

[14] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[15] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc 8th Int Conf on Computer Vision, Vancouver*, pages 525–531, 2001.

[16] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proc IEEE Int Conf on Robotics and Automation*, pages 321–328, 2000.

[17] P.M. Newman, J.J. Leonard, J. Neira, and J. Tardós. Explore and return: experimental validation of real time concurrent mapping and localization. In *Proc IEEE Int Conf on Robotics and Automation*, pages 1802–1809, 2002.

[18] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Int. Journal of Robotics Research*, 21(8):735–758, 2002.

[19] H Shao, T. Svoboda, T. Tuytelaars, and L. van Gool. HPAT indexing for fast object/scene recognition based on local appearance. In *International Conference on Image and Video Retrieval*, pages 71–80, 2003.

[20] Y. Shirai. Analyzing intensity arrays using knowledge about scenes. In *Psychology of Computer Vision*, pages 93–113, 1975.

[21] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proc 9th Int Conf on Computer Vision, Nice*, pages 1470–1477, 2003.

[22] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167–193, 1990.

[23] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Proc IEEE Int Conf on Robotics and Automation*, pages 321–328, 2002.

[24] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *Proc 7th European Conf on Computer Vision, Copenhagen*, pages 82–98, June 2002.