Landmark Based Global Self-localization of Mobile Soccer Robots

Abdul Bais[1,⋆] and Robert Sablatnig[2]

[1] Institute of Computer Technology,
Vienna University of Technology,
Vienna, Austria
bais@ict.tuwien.ac.at

[2] Pattern Recognition and Image Processing Group,
Institute of Computer Aided Automation,
Vienna University of Technology,
Vienna, Austria
sab@prip.tuwien.ac.at

Abstract. We present a stereo vision based global self-localization strategy for tiny autonomous mobile robots in a well-known dynamic environment. Global localization is required for an initial startup or when the robot loses track of its pose during navigation. Existing approaches are based on dense range scans, active beacon systems, artificial landmarks, bearing measurements using omni-directional cameras or bearing/range calculation using single frontal cameras, while we propose feature based stereo vision system for range calculation. Location of the robot is estimated using range measurements with respect to distinct landmarks such as color transitions, corners, junctions and line intersections. Unlike methods based on angle measurement, this method requires only two distinct landmarks. Simulation results show that robots can successfully localize themselves whenever two distinct landmarks are observed. As such marked minimization of landmarks for vision based self-localization of robots has been achieved.

1 Introduction

In mobile robotics the basic requirement for autonomous navigation in any environment is self-localization. There are two different approaches for position estimation: global position estimation and local position tracking. Methods for local position tracking suffer from accumulation of minute measurements to obtain the final estimate, whereas, techniques for global position estimation, are less accurate and often require significantly more computational power [1].This leads to techniques [2, 3, 4, 5, 6, 7, 8, 9, 10] where local measurements are fused with measurements from the robot environment. However, the robot must be able to estimate its position from the very beginning or when/if it loses track of its position during navigation.

⋆ Supported by Higher Education Commission of Pakistan.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 842–851, 2006.

Currently, soccer robots of the size of Tinyphoon are marked on their top with special patterns, which are then tracked for position estimation using a global camera and a host computer. We aim at a shift towards complete autonomy, where all sensing and processing is onboard. However, we look at localization techniques of other (much bigger and slower) soccer robots and also of indoor robots.

Gutmann et al. use a localization method based on dense range scans of the surrounding walls [11]. Whereas, in another approach line segments from range data are matched with the field model to estimate robot position [12]. These methods require that the environment must be surrounded by rectangular walls.

There are several approaches using omni-directional cameras. The major advantage of these approaches is that the robot has a panoramic view of its environment and consequently can acquire more landmark features. Marques and Lima [13] detect field lines using the Hough transform [14] and correlate them with the field model to estimate the robot position. In [8] odometry is used to calculate the expected position of landmarks and then a local search algorithm finds their exact position. Whereas, Motomura et al. localize their robots using dead-reckoning and angle measurements between two landmarks [5].

Approaches using single frontal cameras in conjunction with odometric sensors are widely used for self-localization. These methods are either based on calculating range and bearing based on known shape and size of landmarks or enforce special constrains on environment features[15, 7, 16, 17].

Herrero-Pérez et al. [18] detect features such as goal posts and corners made by the field lines. These features are treated as landmarks in a technique that uses fuzzy logic to account for errors and imprecision in visual recognition.

Approaches using omni-directional cameras with viewing angle of 360 °provide more landmarks but suffer from high cost of the mirror, low resolution of the camera, and requirement of an additional space to fit the mirror and the camera. With frontal cameras one can have high resolution but the field of view is limited. Furthermore, range measurement using single image is too erroneous and the approach cannot be used all the time [19].

To overcome these limitations we propose a stereo vision system with pivoted camera head. This approach would enable us to measure the distance to landmarks and to use bi/trilateration approach to calculate robot position. The pivoted camera head enables the robot to have a 360 °view of its environment. The major advantage of our approach is that it requires less landmarks as compared to the angle based methods.

In this paper we focus on global localization using stereo range measurements. The robot environment consists of visual landmarks i.e. lines, corners, junctions, line intersections and color transitions [20]. The test bed for our algorithm is a soccer playing robot called Tinyphoon (`http://www.tinyphoon.com`) [21].

The balance of the paper is organized as follows: Section 2 discusses robot localization using range or bearing to distinct features in the environment. Potential landmarks are discussed in Section 3. Experimental results are presented in Section 4, finally the paper is concluded in Section 5.

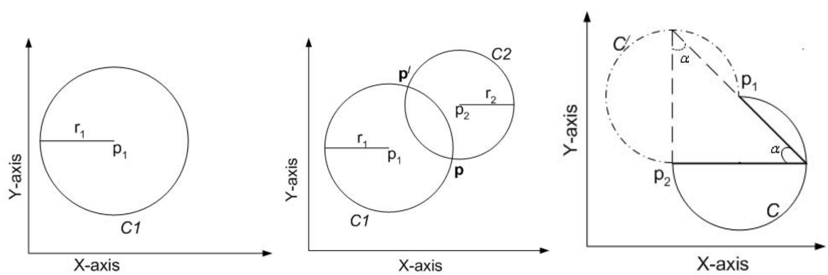
2 Landmark-Based Methods

Landmarks are distinct features that a robot can recognize from its sensory input. Landmarks can be geometric shapes (e.g., rectangles, lines, circles), and they may include additional information (e.g., in the form of bar-codes). In general, landmarks have a fixed and known position, relative to which a robot can localize itself [1].

The input data for position estimation in landmark-based systems may be of range or bearing type. This leads to two different techniques, trilateration and triangulation, respectively. Trilateration is the determination of a robot's position based on distance measurements to known landmarks, whereas, in triangulation, bearing to different landmarks in the environment is used [1].

Fig 1(a) shows the case when the robot identifies a landmark, p_1, and measures the distance r_1. This constrains the robot position to a circle, *C1*. Similarly, detection of landmark point p_2 and its measured distance r_2 will constrain the position of the robot to a circle *C2*. If two points, say p_1 and p_2, are detected at one time then the robot position will be constrained to two points $\mathbf{p}$ or $\mathbf{p}'$, determined by the intersection of the circles *C1* and *C2* (see Fig 1(b)). The ambiguity between these two points can be resolved by considering a fixed order of landmarks.

Fig 1(c) illustrates the case when the robot can only measure the angle α between two landmarks p_1 and p_2. The angle between p_1 and p_2 remains equal to α if the robot is moving along the circular arc *C* or *C'* (shown dotted in Fig 1(c)) [22, 23]. In this case there are infinite number of possible positions and the robot must detect a third landmark point.



(a) The robot is somewhere on the circle (b) Robot position constrained to two points (c) The robot position is constrained to circular arcs

Fig. 1. Constraining robot position with landmarks

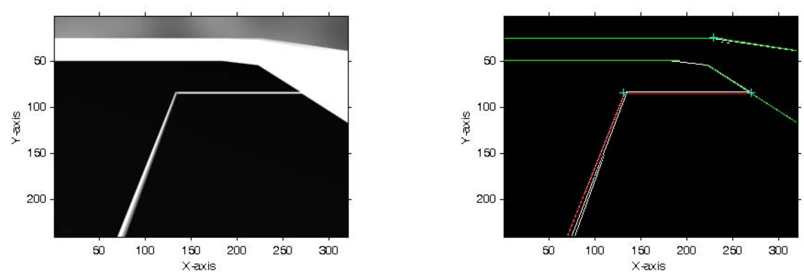
Error free measurements will result in perfect localization. However, measurements are never perfect and errors in distance and angle estimates can vary significantly [23]. Position of the robot will be constrained to a thick ring instead of a perfect circle if there is any error in distance measurement. The intersection of such thickened circles/rings will determine the uncertainty in robot position when two or more landmarks are used.

In addition to measurement errors there could be error in landmark identification and matching with the world map. For the identification errors, some landmarks may not be detected, and some spurious landmarks may be detected. Errors in correspondence could be such that what has been identified as point x on the map may really be point y [23].

3 Landmarks for Self-localization

We use color transitions, corners, junctions and line intersections as landmarks. These landmarks are detected using semantic interpretation of line segments extracted using gradient based Hough transform [20]. Fig 2 illustrate detection of these features.

The vertical edges of the goal corners are normally missed during edge detection and consequent line segments extraction as the change in y-channel value between white and yellow is not significant and the length of the edge is small as compared to other lines in the environment. Therefore, we extract goal corners based on color transitions as discussed in the following section.



(a) Left camera image

(b) Detected features superimposed over the edge map

Fig. 2. Line based landmarks for self-localization

3.1 Detecting Goal Corners

Goals are marked with different colors (blue and yellow). We use color segmentation of the camera images to detect corners of the goal. The process is outlined as follows.

In the left camera image, pixels are tested if they belong to either blue or yellow color. This 'segmentation' is done at a lower scale. Every fourth pixel in a row of every fourth row is tested, which results in a rectangular window around the blue or yellow color patches, if any. The neighborhood of this 'rectangular' window is searched for color transition (color transition from white to yellow, yellow to white, white to blue, or blue to white represents a goal corner), using a full scale. If a color transition is detected in the left image, the corresponding feature points are searched in the right image. The search in the right image is based on parameters of the feature points in the left image. If the corresponding feature point is detected in the right image, its distance from the current robot position is calculated. Detection of two such points determine the robot position as shown in Fig 1(b).

The use of two colors to detect a transition makes the process robust against outliers. All rows inside the rectangular window are searched for transition pixels. One value in a group of pixels is taken as the x-component of the edge between the wall and the colored goal. Outliers in the group are eliminated using simple statistical measures. The calculated stereo range is used to estimate robot position and orientation as discussed in the following sections.

3.2 Calculating Robot Position

We assume that the robot's motion is two dimensional where pose of the robot has 3 degrees of freedom i.e. x, y and θ. The global coordinate system is represented by X and Y axis, whereas the robot coordinate system by x_r and y_r axis. Rotation of robot coordinate system with respect to the global coordinate system is represented by the angle θ. Suppose the robot detects two distinct landmark points p_1 and p_2 at (x_1, y_1) and (x_2, y_2) in the global coordinate system and measures their distances r_1 and r_2, respectively. The two circles at p_1 and p_2 can be described by (1) and (2) as follows.

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2 \tag{1}$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2 \tag{2}$$

Solution of these equations, which is the intersection of the two circles, will give the possible robot position in the global coordinate system. Subtracting (2) from (1) and re-arranging terms we have

$$x = A + By \tag{3}$$

where

$$A = \frac{r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2}{2(x_2 - x_1)}$$

$$B = \frac{y_1 - y_2}{x_2 - x_1}$$

further simplification results in

$$y = \frac{-D \pm \sqrt{D^2 - 4CE}}{2C} \tag{4}$$

where

$$C = B^2 + 1$$

$$D = 2AB - 2x_1B - 2y_1$$

$$E = A^2 + x_1^2 - 2x_1A - r_1^2$$

One of the solution pairs (p_{x1}, p_{y1}) and (p_{x2}, p_{y2}) (if any) from (3) and (4) will qualify for the possible robot position. The ambiguity between the two positions is resolved by considering a fixed order of landmark points.

3.3 Calculating Robot Orientation

In this section we discuss calculation of robot orientation with respect to goal corners which is done after position is estimated. The process is illustrated in Fig 4. When robot calculates its position with respect to the blue goal θ can be calculated using (5) or (6). One of these equations is used depending on the y coordinate of the robot position.

$$\theta = \alpha_1 - \alpha_2 \tag{5}$$

$$\theta = -(\alpha_1 - \alpha_2) \tag{6}$$

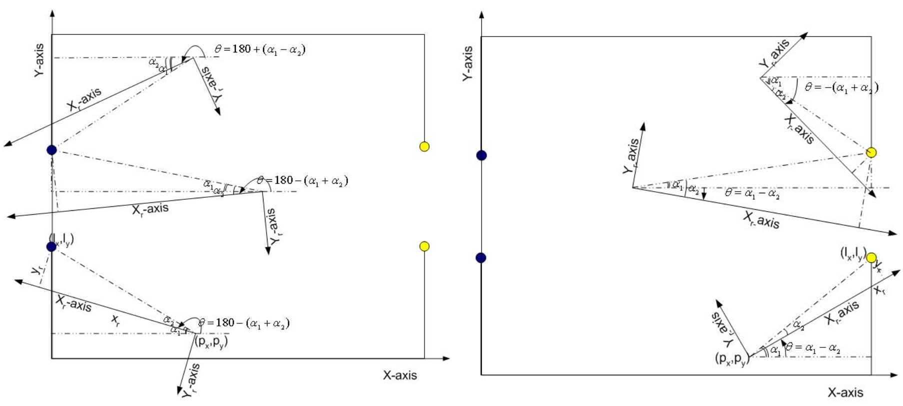
where $\alpha_1 = \arctan(\frac{l_y - p_y}{l_x - p_x})$ and $\alpha_2 = \arctan(\frac{y_r}{x_r})$.

In these equations (l_x, l_y) is the location of one of the landmarks, (p_x, p_y) is robot position and (x_r, y_r) is the location of the selected landmark in robot coordinate system. The landmark and robot position are in global coordinate system. Similarly when the robot position is calculated with respect to the yellow goal θ can be calculated using (7) or (8) as shown in Fig 3(b).

$$\theta = 180^\circ - (\alpha_1 + \alpha_2) \tag{7}$$

$$\theta = 180^\circ + (\alpha_1 - \alpha_2) \tag{8}$$

where $\alpha_1 = \arctan(\frac{l_y - p_y}{p_x})$ and $\alpha_2 = \arctan(\frac{y_r}{x_r})$.



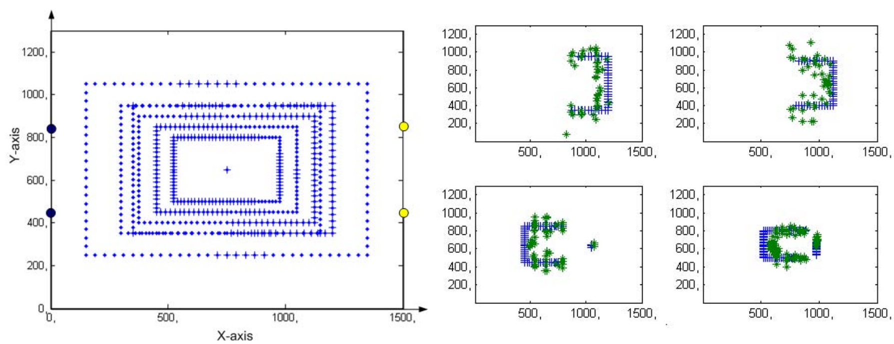
(a) Blue goal (b) Yellow goal

Fig. 3. Robot orientation with respect to goal corners

4 Experimental Results

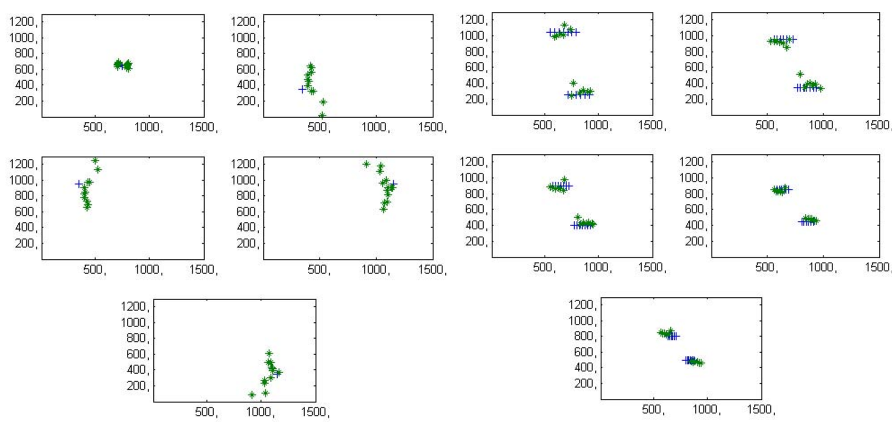
We simulate the performance of our algorithm using only goal corners as landmarks. We have conducted 14 trials where each one has 100 steps. These trials are further grouped into motion without rotation, rotation without motion and motion with rotation. At every step the robot is taking images of its environment, search for color transitions and calculates its position if it finds both the corners.

Fig 4(a) shows the path followed by the robot. Locations where images were taken and searched for color transitions are shown as dots $(\cdot)$. However, depending on the instantaneous pose of the robot both corners are not visible all the time therefore robot pose is estimated only at limited locations shown as plus $(+)$ superimposed on the dots$(\cdot)$.



(a) Actual locations from where position was calculated

(b) The robot is following rectangular paths but is looking only at the blue or yellow goal i.e no rotation



(c) In this case the robot is following rectangular paths and is rotating as well

(d) The robot is rotating in small steps but without any motion

Fig. 4. Experimental trials

Table 1. Error in x,y and θ for the motion only case

	Mean	Std	Min	Max
δx	63.55	37.10	0.46	157.5
δy	58.99	49.42	0.01	247.14
$\delta\theta$	3.19 $^\circ$	2.19 $^\circ$	0.04 $^\circ$	11.50 $^\circ$

Table 2. Error in x,y and θ for the rotation only case

	Mean	Std	Min	Max
δx	58.00	26.90	8.10	123.91
δy	78.53	73.52	0.51	240.00
$\delta\theta$	2.05 $^\circ$	1.58 $^\circ$	0.03 $^\circ$	8.53 $^\circ$

Table 3. Error in x,y and θ for the final case (motion and rotation)

	Mean	Std	Min	Max
δx	36.01	14.32	3.48	69.86
δy	35.92	30.98	1.03	166.24
$\delta\theta$	3.09 $^\circ$	2.52 $^\circ$	0.03 $^\circ$	13.40 $^\circ$

Table 4. Normalized range error

Mean	Std	Min	Max
6.98%	5.26%	0.01%	25.00%

Fig 4(b) shows the case where the robot follows a rectangular path around the field but its orientation remains fixed at 0 $^\circ$or 180 $^\circ$. The plus (+) show the actual position whereas the calculated position is shown as star ($*$). A rotation-only case is illustrated in Fig 4(c). The robot is placed at five locations: near the four corners and at the center of the field. Both motion and rotation is shown in Fig 4(d). Here in this case the robot is moving on a rectangular path and is rotating in fixed steps. In the motion-only and motion-with-rotation cases the robot follows rectangular paths of different sizes.

Statistical results for error in pose are shown in Table 1, Table 2 and Table 3 for all the three cases as discussed above. Whereas, normalized error in range measurements is shown in Table 4. The first column in all tables show values for the average absolute error. The standard deviation (Std), minimum (Min) and maximum (Max) values for each group are presented in the 2nd, 3rd and 4th columns. Error δx and δy in Table 1, Table 2 and Table 3 is expressed in millimeters.

As can be seen from Table 4 the normalized error of range is very high. This error is due to several reasons, like, we use a narrow baseline stereo since the construction of the robot does not allow the use of a wide baseline. Again, all processing has to be done by the onboard processors we use low resolution images(QVGA, 320×240). Moveover, due to the size and concavity of the goal, it is often difficult to determine which point on the goal is being observed. This results in inconsistent ranges and inconsistent landmark positions [15].

5 Conclusion

The method presented in this paper demonstrates that the robot can successfully localize itself with two distinct landmarks. The distinct and bright color of the goals makes them the strongest candidates to be selected as landmarks. Furthermore, calculating robot position and orientation with respect to goal corners is very efficient as only $N/16$ pixels are tested to determine the rectangular boundaries around the color patches (if any), N being the total number of pixels. This results in localization of color patches which are then searched for the actual corners. The error in range estimation is acceptable as we are using just a single shot localization and have not incorporated any kind of temporal redundancy. The robot pose could be refined once a rough estimate is available.

Currently we are working on methods for efficient interpretation of landmarks other than the goal corners, tracking of landmarks, tracking robot position with local sensors and information fusion. Furthermore, we are also investigating self-localization using range measurement to a single landmark where orientation could be obtained with some other means i.e compass or line segments such as the center line of the soccer field.

References

1. Borenstein, J., Everett, H.R., Feng, L.: Navigating Mobile Robots: Systems and Techniques. A. K. Peters, Ltd. (1996)
2. L.Iocchi, Nardi, D.: Hough localization for mobile robots in polygonal environments. Robotics and Autonomous Systems **40** (2002) 43–58
3. Enderle, S., Ritter, M., Fox, D., Sablatnög, S., Kraetzschmar, G., Palm, G.: Soccer robot localization using sporadic visual features. In et al., E.P., ed.: International Conference on Intelligent Autonomous Systems 6 (IAS-6). (2000) 959–966
4. Adorni, G., Cagnoni, S., Enderle, S., Kraetzschmar, G.K.: Vision-based localization for mobile robots. Robotics and Autonomous Systems **36** (2001) 103–119
5. Motomura, A., Matsuoka, T., Hasegawa, T.: Self-localization method using two landmarks and dead reckoning for autonomous mobile soccer robots. In: RoboCup 2003: Robot Soccer World Cup VII. LNCS (2003) 526–533
6. Adorni, G., Cagnoni, S., Mordonini, M.: Landmark-based robot self-localization: a case study for the robocup goal-keeper. In: Proceedings of the International Conference on Information Intelligence and Systems. (1999) 164–171
7. de Jong, F., Caarls, J., Bartelds, R., Jonker, P.: A two-tiered approach to self-localization. In: RoboCup 2001: Robot Soccer World Cup V. LNCS (2002) 405–410
8. Tehrani, A.F., Rojas, R., Moballegh, H.R., Hosseini, I., Amini, P.: Analysis by synthesis, a novel method in mobile robot self-localization. In et al., G.K., ed.: RoboCup 2004: Robot Soccer World Cup VIII. Volume 3276 of LNCS. (2005) 586–593

9. Utz, H., Neubeck, A., Mayer, G., Kraetzschmar, G.: Improving vision-based self-localization. In et al., G.K., ed.: RoboCup-VI. Number 2752 in LNCS (2002) 25–40
10. Christensen, H.I., Kirkeby, N.O., Kristensen, S., Knudsen, L.: Model-driven vision for in-door navigation. Robotics and Autonomous Systems **12** (1994) 199–207
11. Gutmann, J., Schlegel, C.: Amos: Comparison of scan matching approaches for self-localization in indoor environments. In: 1st Euro micro Workshop on Advanced Mobile Robots, IEEE Computer Society Press (1996)
12. Grisetti, G., Iocchi, L., D.Nardi: Global Hough localization for mobile robots in polygonal environments. In: IEEE International Conference on Robotics and Automation (ICRA-02). (2002) 353–358
13. Marques, C.F., Lima, P.U.: A localization method for a soccer robot using a vision-based omni-directional sensor. In et al., P.S., ed.: RoboCup 2000: Robot Soccer World Cup IV. Number 2109 in LNCS (2001) 96–107
14. Duda, R., Hart, P.: Use of the Hough transformation to detect lines and curves in the pictures. Communications of the ACM **15** (1972) 11–15
15. Stroupe, A.W., Sikorski, K., Balch, T.: Constraint-based landmark localization. In Kaminka, G., Lima, P., Rojas, R., eds.: RoboCup 2002:Robot Soccer World Cup IV. Volume 2752 of LNCS., Springer-Verlag (2003) 8–24
16. Bandlow, T., Klupsch, M., Hanek, R., Schmitt, T.: Fast image segmentation, object recognition and localization in a robocup scenario. In: RoboCup-99: Robot Soccer World Cup III. (1999) 174–185
17. Choi, W., Ryu, C., Kim, H.: Navigation of a mobile robot using mono-vision and mono-audition. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC '99). Volume 4. (1999) 686–691
18. Herrero-Pérez, D., Martínez-Barberá, H., Saffiotti, A.: Fuzzy self-localization using natural features in the four-legged league. In et al., D.N., ed.: RoboCup 2004: Robot Soccer World Cup VIII. LNCS, Springer-Verlag (2005) 110 – 121
19. Nickerson, S.B., Jasiobedzki, P., Wilkes, D., Jenkin, M., Milios, E., Tsotsos, J., Jepson, A., Bains, O.N.: The ark project: Autonomous mobile robots for known industrial environments. Robotics and Autonomous Systems **25** (1998) 83–104
20. Bais, A., Sablatnig, R., Novak, G.: Line-based landmark recognition for self-localization of soccer robots. In: IEEE International Conference on Emerging Technologies (ICET '05), Islamabad, Pakistan (2005) 132–137
21. Novak, G., Mahlknecht, S.: TINYPHOON a tiny autonomous mobile robot. In: IEEE International Symposium on Industrial Electronics (ISIE' 05). (2005) 1533–1538
22. Sugihara, K.: Some location problems for robot navigation using a single camera. Computer Vision, Graphics, and Image Processing **42** (1988) 112–129
23. Sutherland, K.T., B.Thompson, W.: Inexact navigation. In: IEEE International Conference on Robotics and Automation (ICRA' 93). (1993) 1–7