A Symmetry Operator and Its Application to the RoboCup

Kai Huebner

Bremen Institute of Safe Systems, TZI, FB3
Universität Bremen, Postfach 330440, 28334 Bremen, Germany
khuebner@tzi.de

Abstract. At present, visual localization of soccer playing robots taking part in the RoboCup contest is mainly achieved by using colored artificial landmarks. As known, this method causes further vision problems like color classification and segmentation under variable light conditions. Additionally, robots confined to use visual sensor information from common cameras usually waste time in switching between the modi of playing soccer and searching landmarks for localization. An upcoming approach to solve these problems is the detection of field lines. Motivated by our research in using a compact symmetry operator for natural feature extraction in mobile robot applications, we propose its application to the RoboCup contest. Symmetry is a structural feature and as results show, it is highly independent of illumination changes and very compliant to the task of line detection. We will motivate symmetry as a natural feature, discuss the symmetry operator and finally present results of the field line extraction.

1 Introduction

For mobile robot tasks it is preferable to concentrate on visual sensor information only. In contrast to other types of sensor data, camera images offer additional information for range estimation, object classification, localization and navigation. Human vision proves that it is possible to realize these tasks without using laser scanners or ultrasonic sensors. In this case, performance is relying on robust image features to be recognized, tracked or classified. In structured workspaces, extraction of those features may be simple if constant light conditions, color markings and maps can be used. In arbitrary environments, however, it is necessary to extract natural and cognitive features.

One of those is symmetry. Most objects in our world have a high degree of symmetry, maybe because it is appropriate to a certain kind of beauty, simplicity or usefulness. Just like animals or plants are quite symmetrical in shape, humans are inclined to use symmetry in art, architecture and artifacts. Therefore, symmetry has been tested in psychological experiments to examine how and how effective human vision explores symmetric objects and scenes [5, 6].

Especially reflective symmetry and its orientation seem of high importance for human vision. Eye-tracking experiments pointed out that there are differences in quality and speed of detecting several types of symmetry, e.g. vertical

D. Polani et al. (Eds.): RoboCup 2003, LNAI 3020, pp. 274–283, 2004.

mirror symmetry (reflective symmetry about a vertical axis) features very quick and accurate detection in most cases. Assuming the presence of horizontal or vertical reflective symmetry in a scene, people are able to initially detect and take advantage of the symmetry axis for further visual exploration.

Because of its relevance in human perception, symmetry has been widely studied in psychological context. Palmer and Hemenway [6] studied the latency for detecting different types of reflective symmetry in a set of arbitrary oriented polygon shapes. Their results show that detection is fastest for vertical symmetry, next fastest for horizontal symmetry and slowest for skewed symmetries. Locher and Nodine [5] made experiments on visual detection and attention of symmetry in composed pictures and showed that the axis of symmetry is used as a perceptual landmark for visual exploration. Ferguson [2] presents the detection of symmetry using visual relations, adjustment of an object's reference frame using its symmetry axes and analyzation of orientation effects.

In computer vision tasks, symmetry has also been used in different applications. Sun [8] proposes a fast symmetry detection algorithm to detect the main symmetry axis of an image. Reisfeld et al. [7] define a generalized symmetry transform including reflective symmetry to extract regions of interest in arbitrary images and to determine an object's value of symmetry. They use a square mask detecting symmetry and gradient-based interest to establish a symmetry picture. Similar results are achieved by Kovesi [4] by analyzing the frequency components of an image. This is based on the idea that if most components have their minimum or maximum at a point of interest, it will correspond to a point of huge symmetry. Chetverikov [1] computes symmetry in order to find orientations of faces in portraits or to detect structural defects in industrial applications. Therefore, he defines a regularity value based on the symmetrical regularity of a pattern. Face detecting is done by Zabrodsky et al. [10] by utilizing symmetry. By using explicitely defined rotational symmetry, they are able to reconstruct partially occluded objects, in the case that they are roughly rotational symmetric (e.g. in an image with partially occluded asymmetric flowers).

2 The Symmetry Operator

Related approaches use symmetry in a global sense. Some use symmetry as a feature of the image itself [1, 8], detect reflective symmetries of any direction [1, 4, 7, 8] or additionally incorporate rotational symmetries [10].

Our line detection method is based on a compact 1-dimensional symmetry operator for arbitrary images [3]. For each pixel of the image, a qualitative value of reflective symmetry in horizontal or vertical direction is determined. *Vertical symmetry* is defined as symmetry about a vertical axis, thus only pixels in the same image row $R = [p_0, p_{w-1}]$ have to be considered for the detection of vertical symmetry about a pixel $p_i \in R$, where w is the width of the image. The same is applied for horizontal symmetry regarding only one column of the image. Furthermore, robot vision requires processing of real images. Because of the common image distortion in real images, an operator detecting exact, mathematic symmetry fails and offers erroneous symmetry images. Therefore, we propose the following qualitative symmetry operator based on a normalized mean square error function:

$$S(p_i, m) = 1 - \frac{1}{C \cdot m} \sum_{j=1}^{m} \sigma(j, m) \cdot g(p_{i-j}, p_{i+j})^2 \tag{1}$$

where $m > 0$ is the size of the surrounding of p_i in which its value of symmetry shall be detected. Thus, the complete number of pixels considered is $2m$. C is a normalization constant depending on the used color space and on $\sigma(j, m)$, which is a radial weighting function. The difference between two opposing points p_{i-j} and p_{i+j} is determined by a gradient function $g(p_{i-j}, p_{i+j})$, which usually is the Euclidian distance of the corresponding color vectors $\overline{p}_{i-j}$ and $\overline{p}_{i+j}$. For all presented experiments, we used 8-bit gray-scale representation with

$$g(p_{i-j}, p_{i+j}) = \begin{cases} \|\overline{p}_{i-j} - \overline{p}_{i+j}\| & \text{if } p_{i-j} \in R \wedge p_{i+j} \in R \\ c & \text{otherwise} \end{cases} \tag{2}$$

where c is the maximum error available (depending on color space), and a linear weighting function additionally depending on m

$$\sigma(j, m) = 1 - \frac{|j|}{m+1} \tag{3}$$

The larger m the more it may exceed the visible region R and the more the error function gets burdened with the maximum error c (see Eq. 2). In this case, image border regions (left and right for vertical symmetry, see Fig. 1b and 1d; top and bottom for horizontal symmetry, see Fig. 1c and 1e) get more influenced by the effect of fading.

Important symmetry axes can be found at places where not necessarily high symmetry values but symmetry peaks can be detected. Though the extraction of maxima and minima of a symmetry image causes more distortion in resulting binary images, it is more significant than using a threshold value. Thresholds may vary from application to application or even from image to image. Additionally, appropriate thresholds are difficult to find for normalized symmetry. A symmetry value of 0 corresponds to hard black-white transitions between each pair of opposing points p_{i-j} and p_{i+j}, while a value of 1 corresponds to exact parity. Thus, high symmetry values are more frequent and much more dense, which makes threshold setting very ineffective. Symmetry is more adapted for the application of local extrema, since it is a regional feature characterizing the local environment (in contrast to local features like edges). Since calculation of vertical symmetry in one row is independent of those in other rows or columns, maxima and minima can be detected line by line, respectively column by column for horizontal symmetry. Results of this symmetry axes detection are presented in Fig. 2. Note that each result has been achieved by only using the symmetry operator and maximum detection, without any kind of pre- or post-processing like Gauss filtering, segmentation or related techniques.

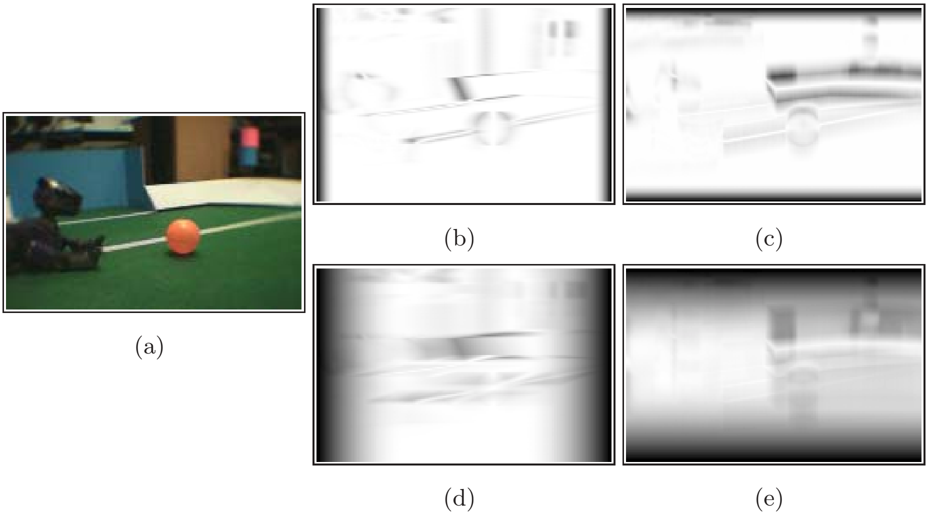


Fig. 1. Symmetry images of a RoboCup image (174×114) (a): vertical and horizontal symmetry using $m = 10$ (b,c) and $m = 50$ (d,e).

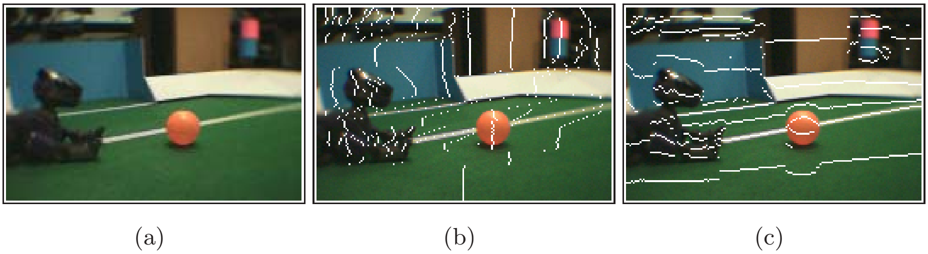


Fig. 2. Symmetry maxima of the RoboCup image (a) using $m = 5$ for vertical (b) and horizontal (c) symmetry axes extraction.

3 Experimental Results

In addition to the extraction of symmetry axes, symmetry feature points can be extracted as crossings of vertical and horizontal symmetry axes. Zhang and Huebner [11] used this feature type to track and classify points of interest with a mobile robot using an omnidirectional vision sensor. In the context of panoramic images, another application is the usage of histograms of vertical symmetry axes as a feature to recognize doors or the direction of the hallway. The several methods of feature extraction are presented in Fig. 3, but surely offer further expansion. Choosing the best feature method mainly depends on the specific application. Vertical symmetry histograms and symmetry feature points were useful in panoramic images for mobile robot tasks like range estimation. In this section, we will demonstrate our RoboCup line detection experiments based on symmetry axes extraction.

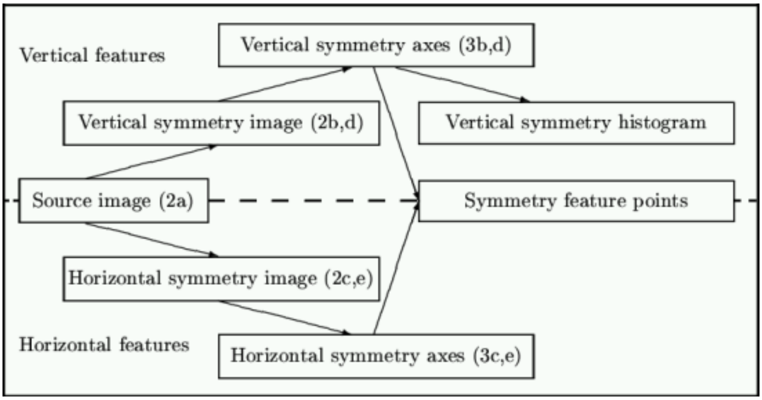


Fig. 3. Several methods of symmetry feature extraction.

3.1 Symmetry Line Filter

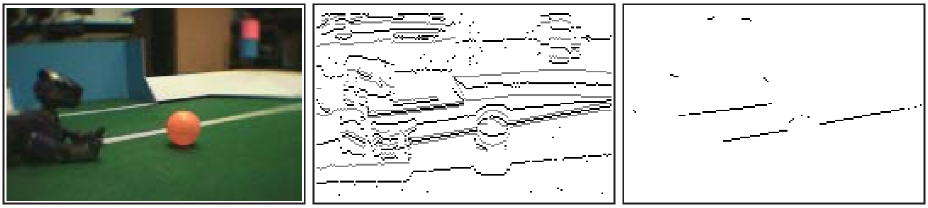
The task of line extraction was motivated by visual mobile robot localization in the RoboCup contest. In this context, localization highly depends on robust recognition of color markings that have to be explored around the field. For robots with common cameras, searching those marks deters from concentrating on game objects like the ball. Thus, it is only possible to either localize the robot or to capture the ball at a time, accordingly it would be more efficient to likewise concentrate on the field for localization. A possible and more intelligent solution could be offered by the extraction of field lines.

Line extraction techniques usually need some preprocessing, let edge detection, thresholding or thinning only be a few examples. Using symmetry, we can detect lines as a structure from arbitrary images. For example, a horizontal line is a structure where we should continuously detect a given A_1SA_2-pattern ($=$ Asymmetry$_1$$-$$S$ymmetry$-$$A$symmetry$_2$) in each small vertical neighborhood along the line. Actually, A_1S is sufficient, because the symmetry axis S implies that there is another A_2 symmetric to A_1. An example for detection of this structure is shown in Fig. 4, where only horizontal symmetry axes using $m = 3$ were detected. If the specific AS-pattern can be found in the same environment, we can assume it is part of a line.

3.2 Line Detection

In the following, two approaches are presented to extract lines from the images resulting from the proposed symmetry line filter. The first one is a modified Hough approach using the Wallace Muff space [9], which represents a line by its start and end point on the image border rectangle.

The Muff parameter space (Fig. 5b) shows that there are two lines that lead from the left to the right side of the symmetry line image (Fig. 5a). The result shown in Fig. 5c seems quite acceptable, but it needed several adaptation steps



(a) (b) (c)

Fig. 4. (a) RoboCup image. (b) Horizontal symmetry maxima and minima (gray = A, black = S) using $m = 3$. (c) Filtered line points by AS pattern.

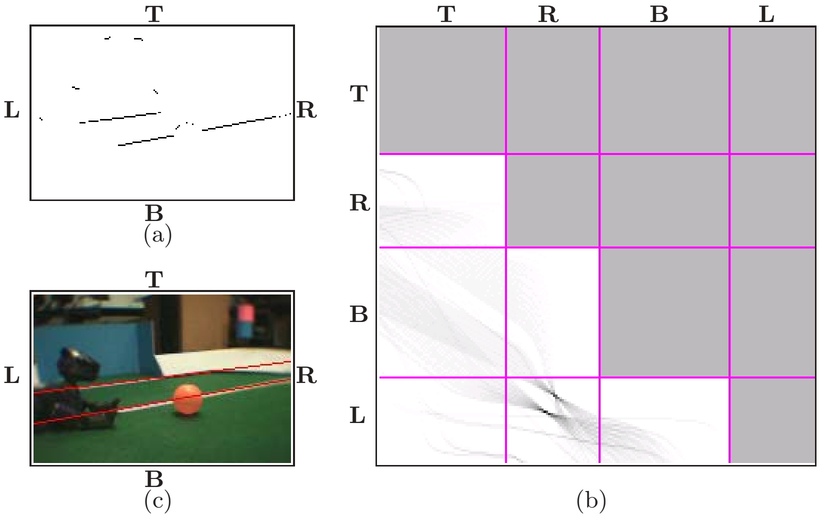


Fig. 5. (a) Symmetry line image. (b) Muff parameter space. (c) Lines found in Muff space.

for this result, mainly because of the difficulty to extract maxima in Muff space. There are further disadvantages of this approach, for example, a line now is represented by its image border points, thus information about line segments is lost. Additionally, curve segments may also be detected as lines with this method, that yet is complex enough without a modified Hough transform for circle detection.

Because of these disadvantages, we developed another approach that takes advantage of the fact that most feature points of the symmetry line image only have one or two neighboring feature points. Simply using the number of feature points in the 3×3-neighborhood of a point p, each feature point can be classified as follows:

- Type A: if p has **no** neighbor, it is not interesting for line extraction.
- Type B: if p has **one** neighbor, it is start or end point of a line.
- Type C: if p has **more** neighbors, it is part of a line.

Thus, we only have to search for feature points of type B (a line's start point) and recursively search the next neighboring point of type C, until we find another point of type B (the line's end point). Therefore, we use the search patterns described in Fig. 6 that are rotational invariant:

Fig. 6. The two search patterns for line segmentation.

Suppose X and Y are points of type B or C, and Y has been detected as the neighbor of X. Now we can start searching the neighboring fields as proposed, until we find a new feature point. If no feature point is found, Y is the end point of the current line, otherwise we proceed at Y in the same manner. Note that the fields left empty can not be occupied by feature points because of the symmetry maxima detection.

Each line segment can now be represented as the list of feature points found by this method. Based on this representation, we can access further information about the line, e.g. the variance of each point to the line described by start and end point. This measure is very useful to easily distinguish curves from straight lines, because the maximum variance will probably exceed a few pixels in the first case (see Fig. 7).

Some results of arbitrary RoboCup images are presented in Fig. 8. In each case, we had to search for thin white horizontal lines. Thus, we applied the hor-

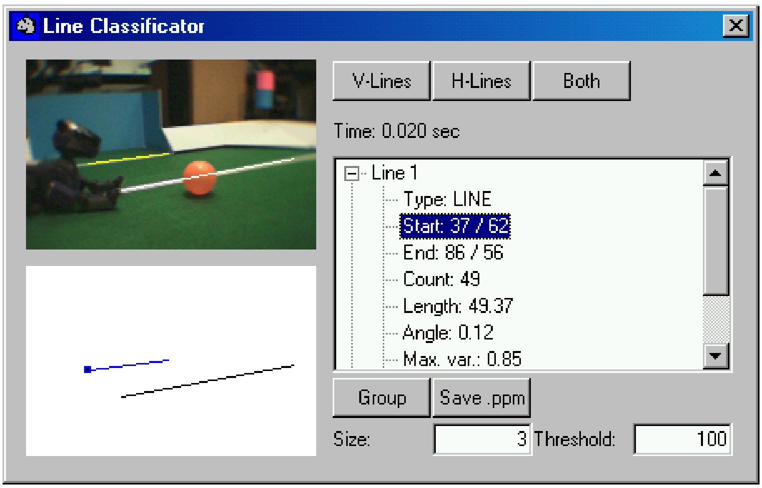


Fig. 7. Screenshot of the Line Classificator Dialog.

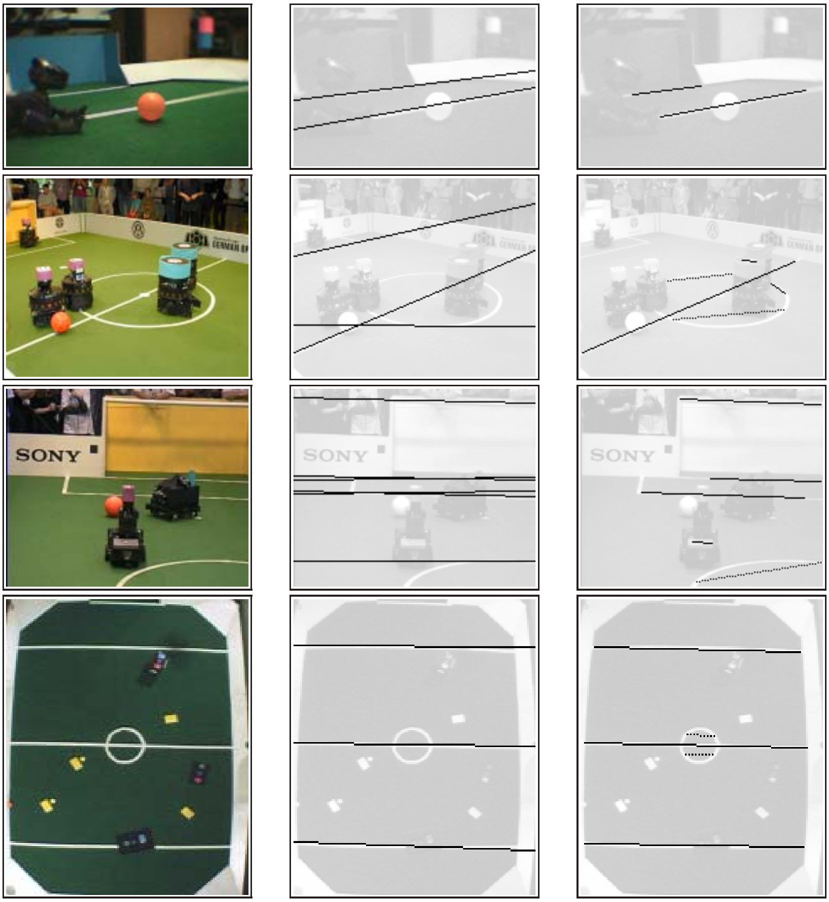


Fig. 8. Results of the horizontal symmetry line detection and classification. In each line: source image, lines detected by Muff space approach, lines and curves (dotted) detected by own approach.

izontal symmetry operator using $m = 3$ and included an illumination threshold neglecting those feature points having a gray-scale value smaller than 100 in the source image. Additionally, we did not care about lines shorter than a given threshold and implemented a heuristic to combine line segments, in the case that they seem to belong to the same field line, but are disrupted by occluding objects.

As proposed, the performance of this method is quite acceptable. Additionally, it is more compact and faster than the Muff space approach. It needs less adaptation, but offers extraction of line segments and classification of curves. In Fig. 8, all line segments are simply plotted as lines from start point to end point, but those identified as curve segments are dotted.

4 Conclusions

We proposed a compact symmetry operator detecting horizontal and vertical reflective symmetry. Resulting symmetry images offer multiple feature extraction methods, especially binary images derived from symmetry axis detection are interesting for further image processing. The operator can be applied to arbitrary images without prior adaptation and without thresholds, the only parameters to specify are the size of the operator mask and the resolution of symmetry data. As a structural feature, symmetry is additionally less dependant on illumination changes, thus no color table or classification is needed.

The operator and the proposed line extraction technique are able to detect lines in a promising and fast way, which is a basic condition for further applications like localization and navigation. As presented, horizontal and vertical symmetry are not purely restricted to detect exact horizontal or vertical lines. For all experiments, only the horizontal operator was used for extracting nearly horizontal lines, but for surely finding all lines in the image, both operators must be used.

Considering the algorithm in comparison to competitive approaches will be another aim. The effort of time seems quite acceptable compared to other gradient-based operators, but it is worse than approaches based on color segmentation. On the other hand, it is an advantage of the proposed operator that it is highly independent of color. In addition to this, the quality of the operator is very convincing with reference to its compactness and line detection ability.

Future work in the context of the RoboCup contest will also consist of supporting localization methods with the lines detected. Some possible approaches would be the use of lines as observations for Monte Carlo Localization or using them for position estimation by Spatial Reasoning.

References

1. D. Chetverikov. Fundamental Structural Features in the Visual World. In *Proceedings of the International Workshop on Fundamental Structural Properties in Image and Pattern Analysis*, pages 47–58, Budapest, 1999.
2. R. W. Ferguson. Modeling Orientation Effects in Symmetry Detection: The Role of Visual Structure. In *Proceedings of the 22nd Conference of the Cognitive Science Society*, Hillsdale, New Jersey: Erlbaum, 2000.
3. K. Huebner. A 1-Dimensional Symmetry Operator for Image Feature Extraction in Robot Applications. *The 16th International Conference on Vision Interface*, June 2003.
4. P. D. Kovesi. Symmetry and Asymmetry From Local Phase. In *The 10th Australian Joint Conference on Artificial Intelligence, Poster Proceedings*, pages 185–190, Perth, 1997.
5. P. Locher and C. Nodine. The perceptual value of symmetry. *Comput. Math. Applic.*, 17:475–484, 1989.
6. S. E. Palmer and K. Hemenway. Orientation and Symmetry: Effects of Multiple, Rotational, and Near Symmetries. *Journal of Experimental Psychology: Human Perception and Performance*, 4(4):691–702, 1978.

7. D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context Free Attentional Operators: the Generalized Symmetry Transform. *International Journal of Computer Vision*, 14:119–130, 1995.
8. C. Sun. Symmetry detection using gradient information. *Pattern Recognition Letters*, 16:987–996, 1995.
9. R.S. Wallace. A Modified Hough Transform For Lines. *IEEE Intrenational Conference on Computer Vision and Pattern Recognition*, 85:665–667, 1985.
10. H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a Continuous Feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12), 1995.
11. J. Zhang and K. Huebner. Using Symmetry as a Feature in Panoramic Images for Mobile Robot Applications. In *Proceedings of Robotik 2002*, volume 1679 of *VDI-Berichte*, pages 263–268, Ludwigsburg, 2002.