Elastic Template Matching as a Basis for Visual Landmark Recognition and Spatial Navigation

Christian Balkenius
Lars Kopp

Lund University Cognitive Science
Kungshuset, Lundagård
S-222 22 Lund, Sweden

christian.balkenius@fil.lu.se
lars.kopp@fil.lu.se

7/4/97

Abstract

We describe an efficient algorithm for visual landmark recognition based on elastic template matching. Each landmark is represented by a set of features and their spatial relations. The matching procedure allows for visual transformations such as translation and expansion, or even partial occlusion.

1 Introduction

Robot navigation in a natural environment calls for a sophisticated sensory system. In this paper we propose a method for visual landmark recognition based on elastic template matching that can be used for both positioning and navigation. We define positioning as a process that uses the sensory input of the robot to derive the location and orientation of the robot with respect to a map of the environment. Navigation, on the other hand, is the use of external cues and internal structures to derive a path through the environment. We will mainly be concerned with navigation, but the methods developed below can also be used for pure positioning.

The landmark recognition method we describe below is able both to recognize landmarks and to calculate the exact angle toward them. It is thus a possible first step in positioning techniques based on angles to landmarks (Zipser 1985, Gallistel 1990). The angle toward the landmark can also be used for so called stimulus-approach behavior (Gallistel 1980, Schmajuk and Thieme 1992, Balkenius 1995) which is the route we will follow in this paper.

2 Stimulus-Approach Navigation

Many different views on how animals navigate have been proposed over the years (see for example Ellen and Thinus-Blanc 1987). Much of the controversy on this issue has centered on whether animals use a method like reinforcement learning (Watkins 1992) or something more of the deliberative approach to artificial intelligence (Maes 1990). We believe, however, that animals use mechanisms over the whole spectrum from simple stimulus-response associations to complex planning abilities (Balkenius 1995).

Here we will concentrate on one such mechanism that lies somewhere in between the two extremes. The basis for this navigational system will be the stimulus-approach behavior (Gallistel 1980, Schmajuk and Thieme 1992, Balkenius 1995). Such a behavior is controlled by an external stimulus toward which the animal, or in our case the robot, moves. Figure 1a shows how a mobile robot can use a number of external stimuli to approach a goal stimulus E which is hidden around a corner.

Proc. AISB workshop on "Spatial Reasoning in Mobile Robots and Animals", Manchester 1997.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161.
Report number UMCS-97-4-1. http://www.cs.man.ac.uk/csonly/cstechrep/titles97.html

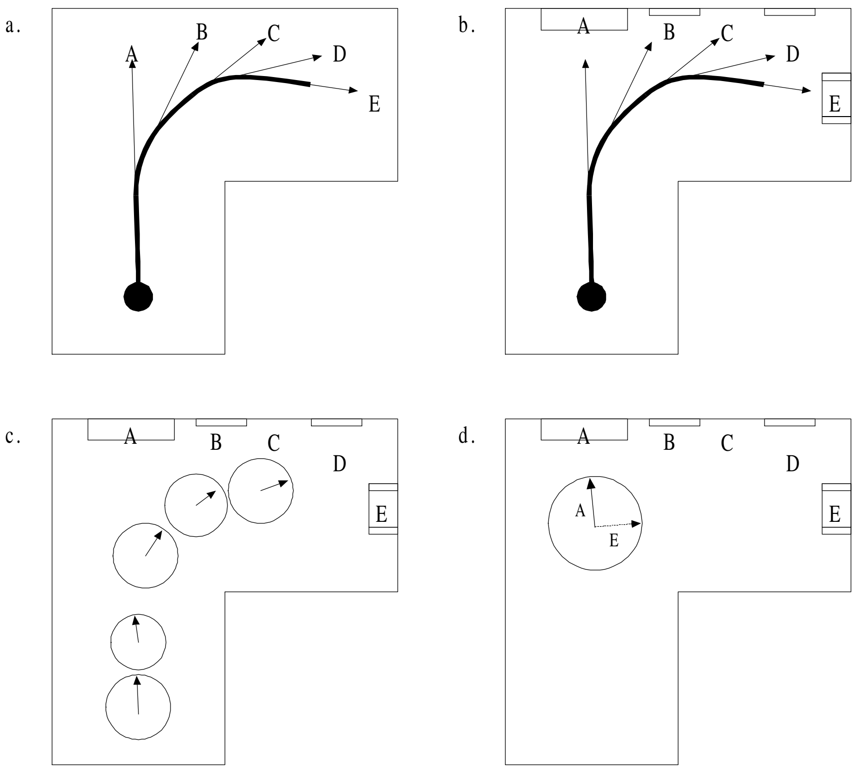

Figure 1: a. Moving around a corner using stimulus-approach. The robot negotiates the corner using a sequence of approach behaviors. First it approaches A, then B, and so on. b. Existing objects in the environment are used as landmarks in a stimulus-approach sequence. c. Since elastic templates can only recognize a landmark within a small region of the environment, the view-field, the same object may be represented by different templates at different view-fields. The view-fields are shown as circles with the direction toward the landmark shown as an arrow. For robust navigation, the view-field must completely cover the environment. d. The view-fields can be used much more economically if they are equipped with expectations of the location of other landmarks that are not in view. The figure shows how a view-field that is directed toward landmark A can generate an expectation of the landmark E. Such an expectation can be used to generate an anticipatory saccade which tries to localize the new target.

The robot keeps a list of stimuli (or landmarks) to approach, in this case A, B, C, D, and E. This sequence is assumed to be constructed by a 'planning' system (See Balkenius 1995 for an example). In the simplest case, it can be assumed that the robot keeps a list of its preference for each stimulus. In this case, E is preferred to D, which in turn is preferred to C, and so on.

To navigate using stimulus-approach, all the robot has to do is to constantly approach the most preferred stimulus in view. In the example, the robot would approach stimulus A until it is able to locate stimulus B. Then landmark B will be selected as the new target to approach until stimulus C becomes visible. It is obvious that with the right list of preferences, this strategy can take the robot between any two locations within an environment.

However, since we allow no preparation or change of the environment, it is necessary that the robot can use already existing features of the environment as landmarks. This is shown in figure 1b where the visual view of furniture along the walls are used as landmarks. Note that landmark identification, as it is used here, is much simpler than true object recognition. For example, the landmarks need not be segmented from the background. What makes these objects good as landmarks is, in fact, that they do stick to the background. A landmark can even consist of the blank wall between two objects, as is illustrated by landmark C and D in the figure.

The situation illustrated in figure 1b is unrealistic in one aspect, however. It assumes that the landmarks can be identified from everywhere within the environment, but that is not generally true. The image generated by a landmark is severely distorted when the viewing angle changes. It also expands considerably as the robot approaches it. If our task had been object identification, this would cause tremendous problems. Today, there exists no solution to the general problem. Fortunately, for spatial navigation, this does not matter. It is not necessary to identify a landmark from every location within the environment. It suffices that there is a small area from where it can be safely identified.

Figure 1c shows a number of small areas (as circles) from where a landmark can be recognized. Since the view of a landmark within such a region is constant enough for it to be identified, we call these regions *view-fields* (*cf.* Zipser 1985). When the robot is located within such a field, it can always calculate the precise direction toward a landmark. If the whole path the robot must take from start to goal is covered by such view-fields, a stimulus-approach behavior can be used by the robot to move through the environment.

A problem with the view-field approach is the large number of landmarks needed. Since the camera can only see a small portion of the environment at once, it is necessary, when the robot has to turn a large angle, that it contains a large number of landmarks. One way to overcome this problem, that we have used in our current implementation of the system, is to equip each stored view-field with a collection of expectations of the location of landmarks that are not currently in view. This is shown in figure 1c where a view-field for landmark A also generates the expectation that landmark E will be visible if the robot turns the specified angle to the right.

During navigation, such expectations can be used to more efficiently move through the environment. In the example above, only two landmarks are needed. There will still be a large number of view-fields, but the environment need not contain as many suitable landmarks. This scheme allows the robot to generate what we call *anticipatory saccades* toward the expected target, turning the camera in the expected direction to bring the landmark into view. In effect, the anticipatory saccades emulate a 360° view of the environment.

The complete navigational system is shown in figure 2. There are three behavior modules that cooperate to perform the navigating behavior: the *orienting* system, the *anticipatory* system, and the *approach* system.

During normal operation, the approach system is in charge of the robot, moving it toward the current landmark. The role of the approach system is thus primarily in control of the locomotion of the robot, but it also uses the movable camera of the robot to track the current target and to estimate its direction. During the approach to the current landmark, the anticipatory system has control over the movable camera of the robot. It uses the expectations of landmarks not in view to generate anticipatory saccades. If the anticipatory system succeeds in its task, the current target is replaced by the newly recognized landmark. Finally, if the robot should find itself in a situation where it cannot locate any landmark directly, the orienting system is activated. This system is used both to re-acquire lost targets and to find the first landmark in the list after the robot has been moved.

In the current implementation of the system in the POLUCS robot, we have made one simplification. Instead of moving the camera, the whole robot turns to do anticipatory saccades. While this functions very well, it slows down the behavior of the robot considerably. We are planning to add a movable head to the robot in the near future. This head will make it possible for the robot to navigate in a very efficient way using elastic template matching.

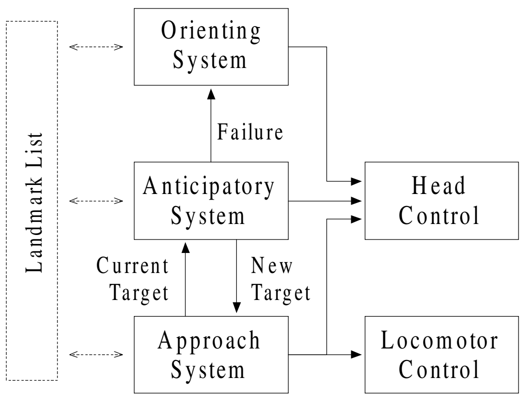

Figure 2: The navigational behavior system used in the POLUCS robot, which uses elastic template matching as its main sensory processing.

3 The XT-1 Vision Architecture

The main concept of this paper is the elastic template which is used to represent a visual landmark. For the landmark recognition to work well it is necessary that the landmarks can be recognized from a sufficiently large region of the environment. Modest translation or rotation of the camera should not disturb the recognition process. It is also necessary that the precise direction toward a landmark can be established.

The main idea behind the template representation is that the image is divided into a number of regions which contain patterns that are as distinct as possible. These regions are called *features* and are coded as small image matrices. To hold these features together the template also records the spatial relations between the features. In the simplest case, these spatial relations are stored as the x and y coordinates in the image of each feature (figure 3). Thus, each landmark is represented as a set of features together with their spatial relations. When the visual system tries to identify a landmark in the image, it attempts to locate each of the features at the approximately correct relative positions. The spatial relations are represented as springs to signify that a distortion of the template is allowed. If sufficiently many of the features are found at the approximately correct location in the image, they are used to estimate the center of the target in the image. The template matching method is described in more detail below in section 6.

Figure 4 shows an overview of the visual architecture used for stimulus-approach navigation. This is one of the two main subsystem of the XT-1 architecture (Balkenius and Kopp 1996a; Balkenius and Kopp 1996b). The other subsystem is used for target tracking and will not be described here. The architecture is an attempt to address a number of visual tasks with a single general architecture. The overall idea is to view visual perception as *eXpectation based Template matching*, hence, its name. The architecture can be divided into five conceptual levels. Each level will be described in the sections below.

Section 4 describes the lowest levels, pre-processing of the input-image. Its primary function is to extract edges from the image. Section 5 describes the second level handling low-level attention. At this level, operations are performed on the image to find areas that are suitable as features within a template.

Section 6 describes single feature processing and the processing of spatial relations. At the single feature level there are two main systems. The first is the feature correlator which searches for features within the image, and the second is the search field module which controls the region where the feature correlator searches. Since in general, the search field is much smaller than the whole image, the feature correlation is dramatically accelerated. Finally, at the spatial relation level, the features in an elastic template are brought together. The topmost level of the architecture is responsible for the stimulus-approach behavior described in section 2.

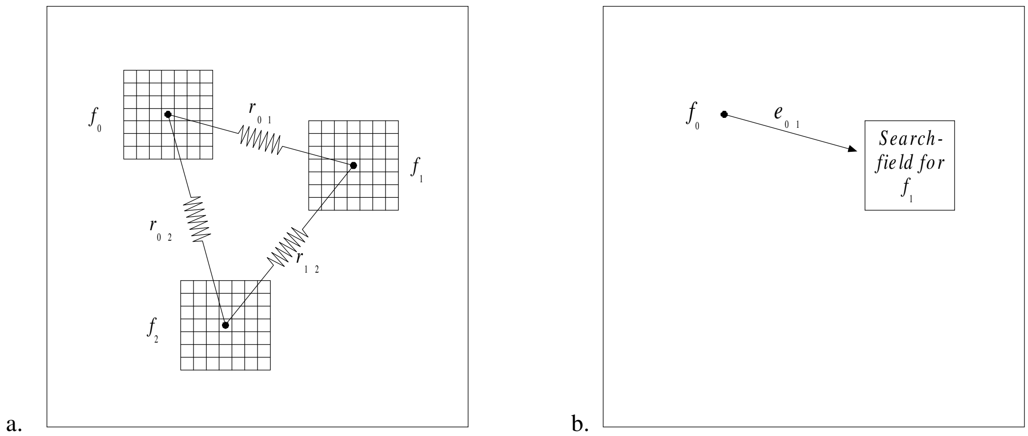

Figure 3: a. An elastic template is coded by a number of features f_0, f_1, f_2 together with their spatial relations r_{01}, r_{02}, r_{12}. b. The location of one feature constrains the search for the next feature.

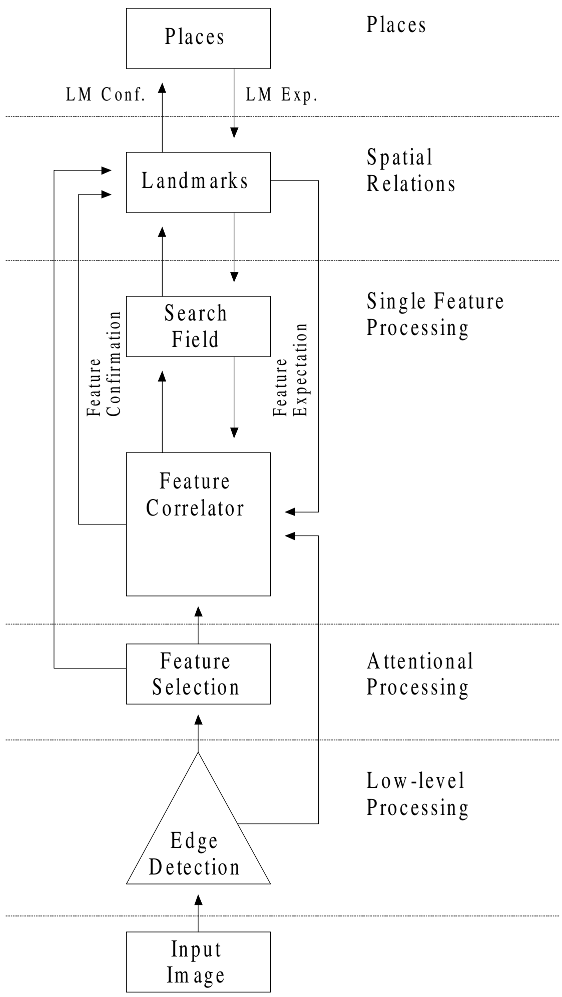

Figure 4: The navigational subsystem of the XT-1 architecture.

4 Pre-processing

The first level of the visual architecture is responsible for pre-processing of the input-image and renders it in a form suitable for attentional and feature-based processing. Although it would be possible to form features directly from the input image, it is not very practical to do so. The main reason for this is that direct matching to the image is very unreliable. For example, slightly changed lighting conditions would seriously disturb the ability to recognize a feature (Ullman 1996).

To make the image robust against variations in illumination we calculate the magnitude of the gradient of the gray level at each point in the image. This step is often referred to as *edge-detection* since it enhances the regions in the image with sharp edges. In our approach, however, this is not an accurate description since the goal of the preprocessing is not to find edges. We prefer to view this process as one which enhances the regions of the image that contain information and suppresses other areas.

In the general architecture, the pre-processing stage generates a multi-resolution pyramidal representation of the input image. In this paper, however, we will only consider as single scale. After an initial scale-reduction, the pre-processing stage constructs a 64×64 pixel image. This image is subsequently filtered with a simple edge filter. This operation is performed by a discrete convolution of the form,

$$E_{i,j} = \sum_{k=-n}^{n} \sum_{l=-n}^{n} K_{k,l} \cdot I_{i-k,j-l}, \tag{1}$$

where n=32, I is the input image, E is the output from the pre-processing stage, and K is the filter kernel (Faugeras 1993). We have experimented with different kernels to determine their influence on the behavior of the system. We initially used the horizontal Sobel operator (Gonzales and Woods 1992),

$$K = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}. \tag{2}$$

However, since the scale-reduction has already performed a low-pass filtering of the image, the smoothing properties of the Sobel operator are not required. As a consequence, we found it possible to get away with the simplest possible edge kernel:

$$K = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \tag{3}$$

Using such a simple edge kernel, the convolution operation in (1) is reduced to the following simple formula:

$$E_{i,j} = I_{i-1,j} - I_{i+1,j}\,. \tag{4}$$

There is a large number of alternatives to this simple method. A standard one is to use the magnitude of the gradient in all directions (Gonzales and Woods 1992). This can be computed by the following equation,

$$E_{ij} = \left\| \nabla I_{ij} \right\|_2 = \sqrt{H_{ij}^{\,2} + V_{ij}^{\,2}}\,, \tag{5}$$

where H_{ij} and V_{ij} together describe the gradient of the image at location $\langle i, j \rangle$. These values can be computed, for example, using the Sobel operator in (2) or using some other approximation method.

From a theoretical standpoint, it seems that the approach in (5) should give better a result when lighting conditions are changed. However, the result we have obtained are not consistent with this assumption. When we tested the template matching method with different lightning conditions, it turned out that the simple kernel in (3) has better generalization properties than the more advanced edge measure in (5). More experiments will have to be performed to determine whether this is generally true, but it appears that the simple edge filter works better since it retains more information about the feature. We are not primarily interested in extracting edges, and it should perhaps not be too surprising that a filter which enhances edges but discards other information does not work as well as a simpler filter.

5 Template Extraction

The second step in the XT-1 vision architecture is the selection of features within the input image. This level is only used during learning, when the robot selects new landmarks in the environment. The selection of a new landmarks proceeds in three steps. First the vision system determines that no existing landmark-template is sufficient to code the current view (See section 6). At this point the template construction mechanism is activated. Second, the feature extraction module at the attentional level is used to select a number of features in the image. These are the features which will be included in the template for the new landmark. Finally, the feature correlation mechanism stores the newly selected features, and the landmark module constructs a template for the new landmark using these features.

The critical step in this process is the extraction of features in the feature selection module. To select a feature, the pre-processed image is used to generate a probability density function defined over the image plane which in turn is used to select the location of a new feature. To do this, the pre-processed image is first threshold to avoid that features are located in areas of weak contrast which contain very little information. This process constructs a new image F, where

$$F_{ij} = \begin{cases} \left| E_{ij} \right| & \text{if} \left| E_{ij} \right| > \theta \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

The amount of information around a certain location is then calculated by summing the values of F around every location. If the heights and widths of the individual features are called h and w, the next image, G, is calculated as,

$$G_{ij} = \sum_{k=0}^{w} \sum_{l=0}^{h} F_{i+k,j+l}\,. \tag{7}$$

Finally, a probability density function, _P, is constructed by normalizing the values in G_{ij}:

$$P_{ij} = G_{ij} \Big/ \sum_i \sum_j G_{ij}. \tag{8}$$

To select a new feature location, a location $\langle x, y \rangle$ is drawn from the distribution described by P. The actual feature is subsequently constructed by copying the values of E around the selected location.

Each feature is represented as a pair $F = \langle p, f \rangle$, where p is the position of the feature in the image, and f is the feature matrix at that location in E when the template is constructed. We assume that there is an implicit reference direction in the middle of the input image which is used as the origin for the positions p. A template, T, is then constructed from a set of features. Thus, $T = \{ F_0, F_1, F_2, \ldots \}$.

6 Template Matching

The core of the navigational subsystem described in this paper is the template matching algorithm which incorporates all the levels of the vision architecture. Here we will only describe a somewhat restricted version of the matching process which is used to find landmarks in an image.

The straight forward approach to elastic template matching consists of two steps. In the first step, all possible locations for all features within the template are found in the input image. In the second step, the spatial relations are used to disambiguate the different matches of each feature and to calculate the overall match of the template. Such a procedure is, however, very costly since it involves a search for each feature over the whole input image.

A central idea behind the matching process we use is to view the spatial relations among the features as expectations of where another feature should be located, given the locations of the already found features. This is illustrated in figure 3b where the location of feature f_0 constrains the search for the subsequent feature, f_1.

The expectation, e_{01} can be generated during search, either from the last feature found, or from all features which have already been localized. In either case, the search process keeps an estimate $\hat{p}$ of the location of the reference position for the template and this estimate is used to set the center of the search field. If a feature is learned at position p and $\hat{p}$ is the current estimate of the reference direction of the template, the search field is centered at the location, $p + \hat{p}$.

Since a feature is only considered found if it lies within the expected search field, it is not necessary to search the regions of the image that fall outside the search field. Since the search field is considerably smaller than the input image, this speeds up the template matching considerably.

However, the search field can only be used when some other feature has already been found. Since this is not the case for the first feature, it is necessary to do a total search for this feature. If the first feature is found, its location will subsequently constrain the search for the next feature. When a feature is not found, a total search is performed for the next feature since no expectation will be present.

In our current robot, POLUCS, the camera is not allowed to move in the vertical direction. This means that features within a template do not move very much in the vertical directions. As consequence, a total search is limited to a small horizontal band across the image where the feature may be located.

There are a number of problems with this approach to template matching. The most serious is that the first feature is very critical. If it cannot be found, the matching process will be slowed down. A more serious case occurs when the first feature is found at the wrong location. In this case, the search field will not be correct and it is likely that the second feature will not be found.

In the experiments we have performed so far, however, these limitations have not been a serious problem. When they occur, they slow down the image processing somewhat, but the effect is not noticeable in the navigating robot.

To determine whether a template matches the image, we check whether a sufficiently large fraction of the features in the template are found with a sufficiently large correlation within the expected location in the image generated during the search process. If f is the feature matrix and E is the edge image generated by equation (4), the correlation $c_{ij}(f)$ of a feature, f, at point $\langle i, j \rangle$ is calculated using the Moravec correlation (Lawton, et al. 1988),

$$c_{ij}(f) = \frac{\sum_{k=0}^{h} \sum_{l=0}^{w} f_{k,l} E_{i+k,j+l}}{\frac{1}{2} \sum_{k=0}^{h} \sum_{l=0}^{w} f_{k,l}^2 + \frac{1}{2} \sum_{k=0}^{h} \sum_{l=0}^{w} E_{i+k,j+l}^2}. \tag{9}$$

Typically, we require that 75% of the features should be found with a correlation of at least 0.7. These values are not critical, however. An interesting extension of the system would be to change these parameters dynamically as a result of the confidence in the current estimate of robot's position.

7 Recognition Experiments

Figure 5 shows the typical behavior of the template matching process when the robot is moved around in the environment. The graphs show how well the robot recognizes a landmark when it is moved away from the location where the landmark was learned. The robot can recognize the landmark when it is moved as much as 90cm forward. When it is moved to the left or the right, recognition is fairly accurate within 60cm from the learned location. The two curves suggest that there exists a generalization surface around the learned location where the response of the recognition system is fairly high. When the robot is moved outside the view field, the response falls of drastically as can be seen in the graph. This is especially true when the robot is moved forwards or backwards. Such a translation of the robot corresponds to an expansion of the visual view. Obviously, this expansion depends on the distance to the landmarks. The further away the landmarks are, the smaller the expansion will be.

The robot can also be rotated around 10-15 degrees in either direction without losing sight of the landmark. This is shown in figure 6, where the graphs for template match is shown at three different locations. As can be seen, the sensitivity to rotation is very similar for different locations within a view field. When the robot is translated either 50 cm forwards or backwards, it can still recognize a landmark within a rotation of 10 degrees.

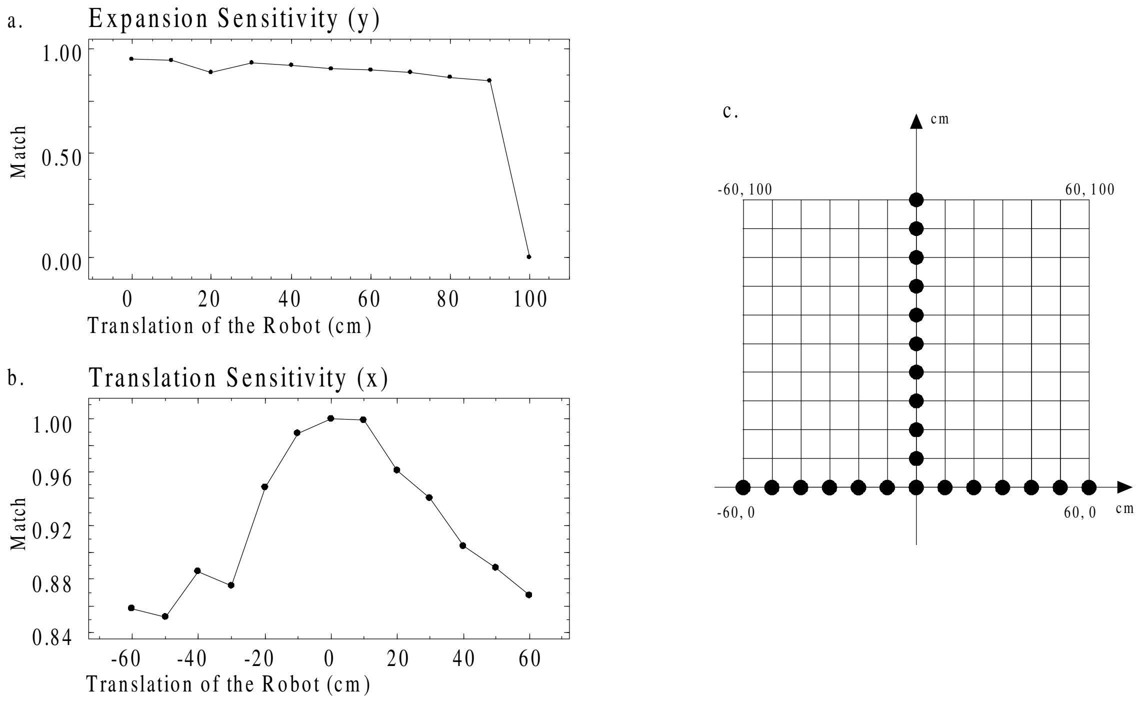

Figure 5: a, b. The sensitivity to translation of the robot. c. The tested spatial locations.

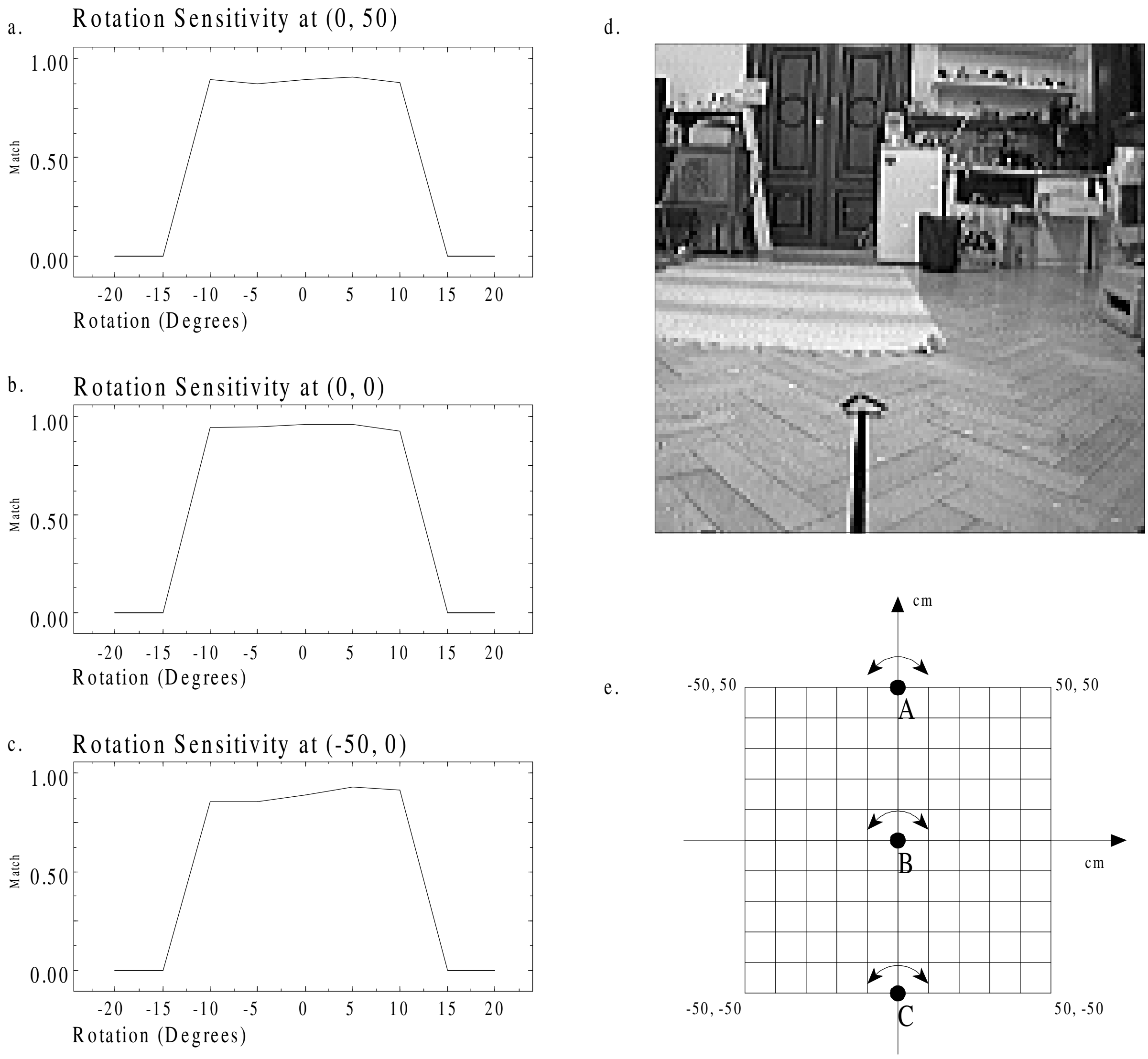

Figure 6: The rotation sensitivity of a typical landmark. a-c Measurements at three different locations. d. The look of a input image used as a landmark. e. The three different locations shown in a-c.

These properties of the template matching process make the size of a typical view field around 1m^2. With the resolution of the input image we are currently using (64×64 pixels), with 16 features of size 8×8, a total mapping of an environment requires approximately 16kByte/m^2, that is, 16×8×8 = 1k per template multiplied with the required number of templates for a total rotation of the robot.

The recognition process is extremely stable as long as there are some objects in the environment which can be used as landmarks. We have not found an environment where the algorithm does not work, however. The algorithm is also very insensitive to changes in the environment (Balkenius and Kopp 1997). Objects can be removed or added without disturbing the system much, although there are obviously limits to how much can be changed. It is also interesting to note that the described recognition method works best in the mess of a home rather than the sterile office environments often used for mobile robots.

8 Conclusion

We have described a template matching method which can be used for visual learning and recognition of landmarks in normal environments. The visual architecture described has been implemented in a mobile robot which is able to navigate using vision as the only sensory system. The heavy use of expectations makes it possible to run the complete system in real-time using a Pentium 133 MHz computer.

Acknowledgements

We would like to thank Lukas Böök for helpful comments on the paper. The support from the Swedish National Board for Industrial and Technical Development (NUTEK) is gratefully acknowledged.

References

Balkenius, C. (1995). *Natural intelligence in artificial creatures*. Lund University Cognitive Studies, 37.

Balkenius, C., and Kopp, L. (1996a). Visual tracking and target selection for mobile robots. In Jörg (ed.). *Proceedings of EUROBOT '96.* IEEE Press.

Balkenius, C., and Kopp, L. (1996b). The XT-1 Vision Architecture. In Linde, P., and Sparr, G. (eds) *Symposium on image analysis.* Lund University, Dept. of Mathematics, 1996.

Balkenius, C., and Kopp, L. (1997). Robust self-localization using elastic template matching, *submitted.*

Ellen, P., and Thinus-Blanc, C. (1987). *Cognitive processes and spatial orientation in animal and man*. Dordrecht: Nijhoff.

Faugeras, O. (1993). *Three-dimensional computer vision: a geometrical viewpoint*. Cambridge, MA: MIT Press.

Gallistel, C. R. (1980). *The organization of action: a new synthesis*. Hilldsdale, NJ: Erlbaum.

Gallistel, C. R. (1990). *The organization of learning*. Cambridge, MA: MIT Press.

Gonzales, R. C., and Woods, R. E. (1992). *Digital image processing*. Readings, MA: Addison-Wesely.

Lawton, D. T., Rieger, J., and, Steenstrup, M. (1988). Computational techniques in motion processing. In Arbib, M. A., and Hanson, A. R. (eds) *Vision, brain and cooperative computation*. Cambridge, MA: MIT Press. 419–488.

Maes, P. (1990). Designing autonomous agents. In Maes, P. (ed.) *Designing autonomous agents*. Cambridge, MA: MIT Press. 1-2.

Schmajuk, N. A., and Thieme, J. J. (1992). Purposive behavior and cognitive mapping: a neural network model. *Biological Cybernetics,* **67**, 165-174.

Ullman, S. (1996). *High-level vision: object recognition and visual recognition*. Cambrdige, MA: MIT Press.

Watkins, C. J. C. H. (1992). Q-learning. Machine Learning, **8**, 279-292.

Zipser, D. (1985). A computational model of hippocampal place fields. *Behavioral Neuroscience,* **99**, 1006-1018.