7

Localisation using Output Relevant Features (ORFs)

In this chapter, a method for robot localisation by using omnidirectional images is presented. The idea is to estimate the robot's position in the environment using image features, which have been learned from a small number of training pictures. The learning is based on using ORFs (Output Relevant Features), which have yet been used in omnidirectional context by Schwert [20] and Zhang et al. [28]. These are produced by a single-layer feed-forward perceptron network with the Hebbian learning rules. After having learned the ORFs, they can be used to estimate the position out of any picture taken at any place in the environment. Experiments, results and possible enhancements are discussed.

7.1 Output Relevant Features

Due to the fact that it is difficult to find relevant features in a couple of taken pictures and because the search and evaluation of those features have to be directly implemented to the robots image processing algorithms, e.g. edge or symmetry detection, the simple ORF method is used.
This learning based method tries to find features itself by looking at given input vectors and teached output values. The approach is based on a single-layer feed-forward perceptron network for each output value (see Fig. 7.1). Each network is trained to maximize the correlation between the input vectors and the corresponding output values refering to the Hebbian learning rule

$$\Delta w_j = \eta(y_w - y)x_j \tag{7.1}$$

where η is a given learning rate.
Thus the learned weighting vector $\vec{w}$ includes the ORF values. In experiments, the omnidirectional round images of the environment are used as input vectors, which result in estimated position coordinates.

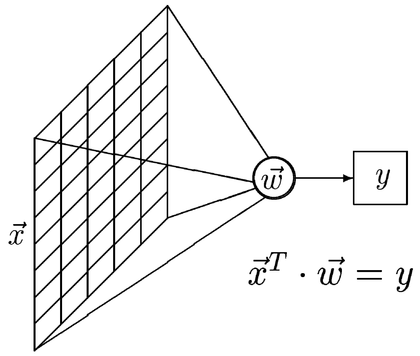

Figure 7.1: Perceptron principle

7.2 Experimental Setups

The pictures for the described tests have been taken across the hall of our working group (see section 3.4). Unfortunately, this environment is poor in colour, so that one can reduce it to gray floor and doors and white painted walls. Attention was paid to use constant environmental conditions, e.g. constant light influence, no active motion (like persons running around) or passive motion (like door opening and closing between two pictures).

The vision system delivered omnidirectional pictures (see Fig. 7.2), which were used as the first datasets for each experiment. To avoid the maybe linear influence of translational movement on the learning method, the image parts of the ceiling were cut. In addition to this, the region the robot looks back on itself is masked, supposing it should be constant (which it is not, because of light influences). The second datasets were established by calculating these masked pictures (see Fig. 7.2) from the original pictures. In order to speed up the calculation of the ORF vector and because the environment permitted this constraint, every picture was converted to grayscale.

Figure 7.2: Original image (left) and masked image (right)

7.2.1 Setup I

The first experiment includes 52 round pictures taken at the center of the hall with a constant distance of 10cm between two neighbouring pictures (these correspond to the 52 panoramic pictures in Fig. 3.6. Thus tests were done on a line with a length of 5.10m, lying between two pairs of opposing office doors (see Fig. 7.3). For this 1D problem one ORF network was trained to deliver the x position on the line. Note that the door crossing the hall is a glass door.

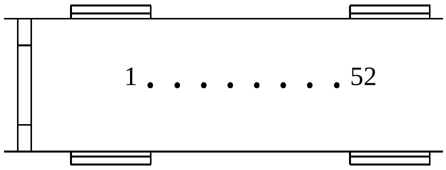

Figure 7.3: Experiment I (1D; 52 images on a straight line)

7.2.2 Setup II

The second experiment resembles the first. 36 pictures were taken on a line along the complete hall with a distance of 20cm to the side wall and a distance of 50cm between neighbouring pictures, which makes a total distance of 17.50m.

7.2.3 Setup III

For the third experiment, the first one is extended by including another dimension. 90 pictures were taken on a 15×6 grid. Keeping the distance of 10cm between neighbouring nodes makes an area of 1.40m $\times$ 0.50m, which was placed in the same environment (see Fig. 7.4) as experiment I. Two ORF networks were trained, one for each output value (x and y coordinate).

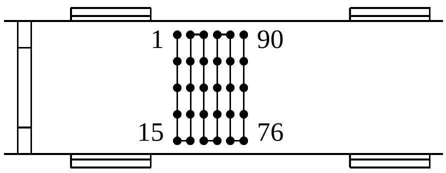

Figure 7.4: Experiment III (2D, 90 images on a 15×6 grid)

7.3 Experimental Results

Experiments were made by splitting each dataset into training data and test data. Like anticipated, the method works with neglectable error using all pictures for training. The main interest is to reduce the used training data without losing performance in results for not trained test data.

7.3.1 Result I

The database for the first experiment contained 52 pictures of size 612 $\times$ 384. First every fifth picture was taken (let $l = 5$ indicate this), which is equal to one picture every 0.5m, to train the ORF vector. It was started with picture 6 and finished with 46 in order to additionally get some information about the method's extrapolation abilities. Figure 7.5 shows the results for this test.
The training pictures correspond to the filled symbols, the test pictures to the unfilled. Note that in this case, the given image numbers are set on the x-axis and the estimated ones on the y-axis, which means that a perfect position estimation would results in a straight diagonal line of symbols. One can see that the extrapolation

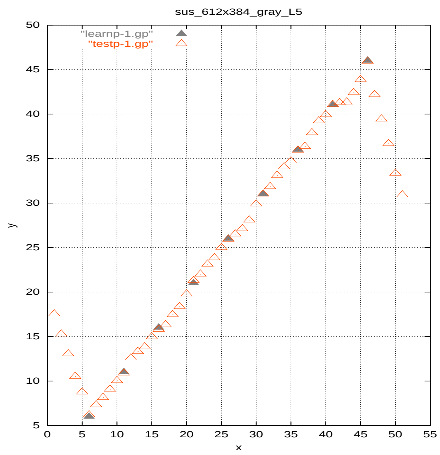

Figure 7.5: Unmasked results for (I), $l = 5$

results of pictures 1-5 and 47-51 were expectedly poor, but on the other hand the interpolation of unknown positions works very well with small variances. Fearing that the translational motion beneath the ceiling may have had advantaged this result, another test was made using the masked (see section 7.2) and also downscaled pictures of size 128 $\times$ 96 (see Fig. 7.6)without extrapolation. In spite of this loss of image information, the method still delivers good results. Thus the number of training pictures was decreased by setting $l = 10$, which means a distance of 1m between two training pictures. The results are presented for unmasked (Fig. 7.7) and masked and downscaled (Fig. 7.8) pictures with extrapolation. Results are still acceptable for the unmasked pictures, although the estimation error has grown. There is also a visible estimation problem regarding the last 10 interpolated points of the masked

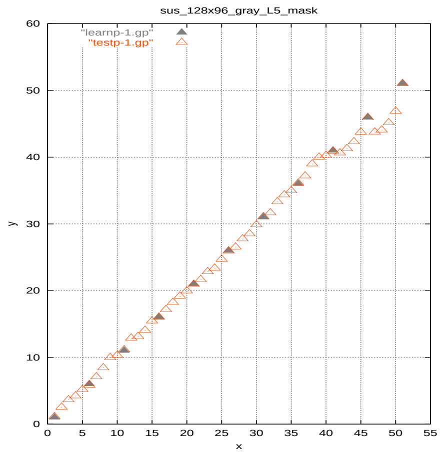

Figure 7.6: Masked results for (I), $l = 5$

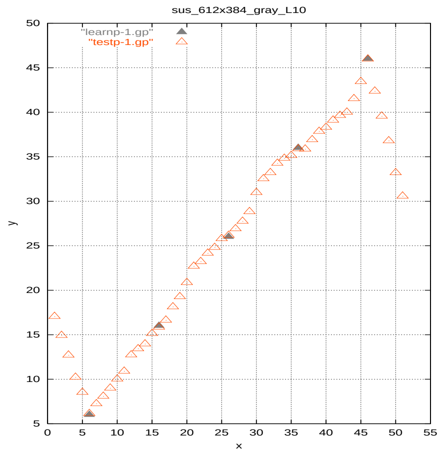

Figure 7.7: Unmasked results for (I), $l = 10$

series, which means that in this case masking and scaling obviously dropped out important feature information.

7.3.2 Result II

The database for the second experiment contained 36 pictures of size 400 $\times$ 300. Every second picture (distance 1m) was used for training. Tests belonging to extrapolation were not supposed to be reasonable regarding the results of the first experiment. Figure 7.9 shows the results. Results are poor, except those of the trained positions. The problem seems to be caused by the environmental conditions that situations at different places result in very similar pictures, e.g. there is more than one situation between two opposing doors (see section 3.4), which has not been the case in the first experiment.

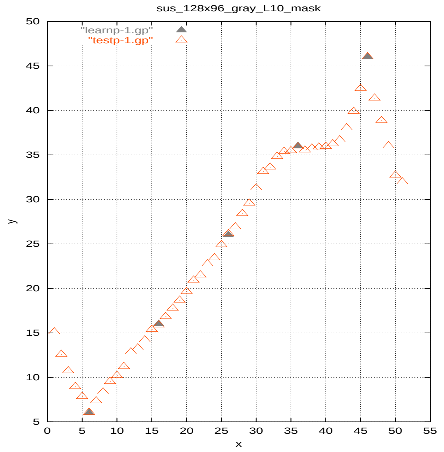

Figure 7.8: Masked results for (I), $l = 10$

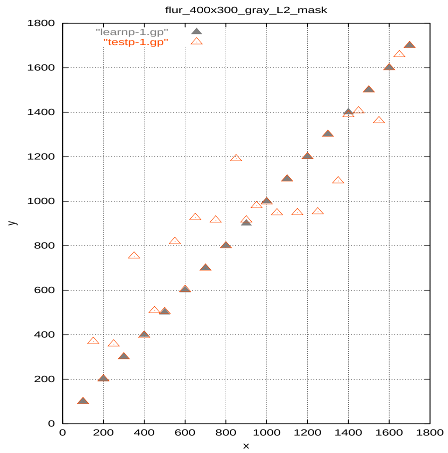

Figure 7.9: Masked results for (II), $l = 2$

7.3.3 Result III

For the third experiment 90 pictures of size 400×300 were taken on the described 15×6 grid. First tests have been made on a minimum part of 40cm $\times$ 40cm (5×5 grid) by only learning the overlaying 3×3 grid points and testing the points between (see Fig. 7.10).

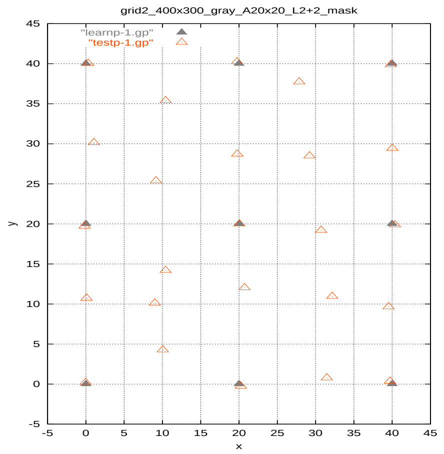

Figure 7.10: Masked results for (III), 5 x 5 grid

The interpolated estimation of the unkown positions is convincing, considering that each ORF tries to map more than one input picture to one and the same output value and there are only few resulting variances up to 5cm. It is also remarkable that (in this case) the estimation of the x-value works much better than that of the y-value. The results of testing across a large 15 x 5 grid, using every second point along each direction as before, are presented in Fig. 7.11. In this experiment, there are more

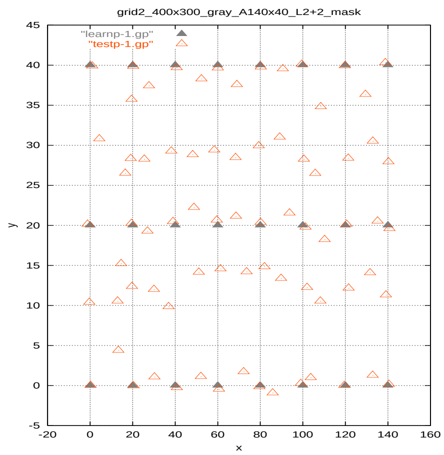

Figure 7.11: Masked results for (III), 15 x 5 grid

estimation errors about a few centimeters, but it can still be determined that almost no estimation exceeds an error of 5cm.

7.4 Conclusions

The ORF method with its underlying perceptron and Hebbian learning principles is in the same way simple as general in its application. For localisation or similar problems, it is reasonable to use the omnidirectional pictures almost directly as the input

vector, which may be not necessary for other vision tasks. Not described experiments with panoramic pictures show the same acceptable performance.
In our approach, the network was teached to estimate positions along one or two axes which corresponds to a question like „What is my position?“. It may also be possible to combine knowledge based methods with ORF-learned abstract boolean outputs like „Is there a door in front of me?“.
Maybe this could lead to more robustness by solving some of the problems of the presented ORF application, which can be summarized in

- No extrapolation abilities (see exp. I), which means that localisation in unknown, untrained environments is not possible.
- Similar input situations lead to similar output values, which is something that might not be desirable (see exp. II). Especially in applications like localisation this is a big problem, because images taken at two different locations should never lead to the same position result. For the cause that neither the ORF method nor human beings can differ two very similar locations only by analysing image data, other, e.g. time-dependent, methods have to be added to get a unique solution. Another idea is to check from the outset if there are ambiguous situations, like Zhang et al. [28] do by defining an overlap measure.
- Mapping different input situations on one and the same output value might lead to estimation errors not only in tested, but also in trained data (see exp. III).
- The perceptron rule is based on linear classification, thus it can only solve problems which are linear separable, though this can never be guaranteed

Those describe the problems to take care of in experiments similar to those mentioned in this approach, where disturbances like differing light conditions, moving objects or persons and a dynamic environment are frivolously omitted.
Beside this the effort for implementation and calculation of the ORF method is small. Calculation time for one ORF „train & test“ is mainly dependent on number and size of the input data (in this case width, height and colour depth of the pictures), but exp. I shows that its minimizing by reducing size and omitting colour will not necessarily lead to worse results.