RBF Networks for Object Recognition

Josef Pauli, Michael Benkwitz and Gerald Sommer

Christian-Albrechts-University
Department of Computer Science
Preusserstrasse 1-9
24105 Kiel
Germany

Abstract

A predominant task occurring in Computer Vision is to localize and recognize the two-dimensional view of an object in the image. In particular, for controlling our vision based roboter system autonomous object detection is necessary for grasping scene objects. The work being reported here uses RBF networks for learning and representing object detectors. The primary goal of learning is to generate detectors which are robust with respect to various imponderables, e.g., illumination condition, camera properties, viewing directions, reflections on the object. We show the approach exemplary for real world images taken under varying illumination conditions and varying object background.

1 Introduction

Frequently a model-based approach is used for detecting a scene object in the image [Dickinson *et al.*, 1990]. First, a three-dimensional model object is assumed to be known a priori as well as the geometric projection laws of the camera system [Faugeras, 1993, pp. 33-68]. Second, by taking the projection laws into account the model object is transformed into a two-dimensional model and this one is matched with the image [Faugeras, 1993, pp. 483-558]. Third, the place with the highest matching score describes the location of the object. The model object is defined by geometrical attributes whereas the image consists of gray levels. Obviously, these are different types of representation and image segmentation is the usual approach to bridge the gap [Maxwell and Shafer, 1994]. Unfortunately, image segmentation is only useful if surface discontinuities of the scene object have corresponding gray level edges in the image. It is a matter of fact, nearly all problems in object recognition can be traced back to the mentioned gap of representation.

Our system bridges the gap of representation by using a fundamentally different category of models dispensing with geometric attributes. Rather the models will be defined directly as two-dimensional views in terms of gray level attributes incorporating implicitly the photometric laws of the camera [Murase and Nayar, 1995]. Consequently, we do object detection without image segmentation and apply instead appearance grounded detectors.[1] The detectors will be learned using a Radial Basis Function (RBF) network [Poggio and Girosi, 1990].

An RBF network interpolates a transformation of m-dimensional vectors into p-dimensional vectors by a linear combination of n nonlinear basis functions. Each basis function operates as a localized receptive field and therefore responds most strongly for input vectors localized in the heighbourhood of the center of the field. E.g., the radially symmetric Gaussian is the most popular basis function mapping the distance between an input vector and the center vector into a real value of the unit interval. The centers and the extent of the receptive fields are learned with unsupervised methods, while the factors for combining the basis functions are learned in a supervised manner. Our system uses RBF networks both for detecting objects in the image and for determining size and orientation of objects. In this report we use an RBF network to reason about the existence of an object in the image.

2 Learning photometry grounded detectors

First, we take appearance patterns from the object of interest under varying conditions. Second, these patterns will be transformed into a specific structure in order to apply efficient search strategies (see section 3). Third, the specific patterns have to be clustered and a small number of typical specific patterns selected. Fourth, the typical patterns are combined appropriately for defining an object detector.

2.1 Taking sample images

We take sample images containing the object which have to be detected at a later date. The images may differ from each other with respect to varying illumination conditions, varying object background, varying camera parameters, varying viewing directions, varying distances between camera and object, and so on. In this work we only consider sample images under varying illumination conditions and varying object background (Figure 1 shows three exemplary images from an overall collection of 16, the relevant object is outlined with a white rectangle).

[1] For Physics-Based Vision see several sections in the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1994.

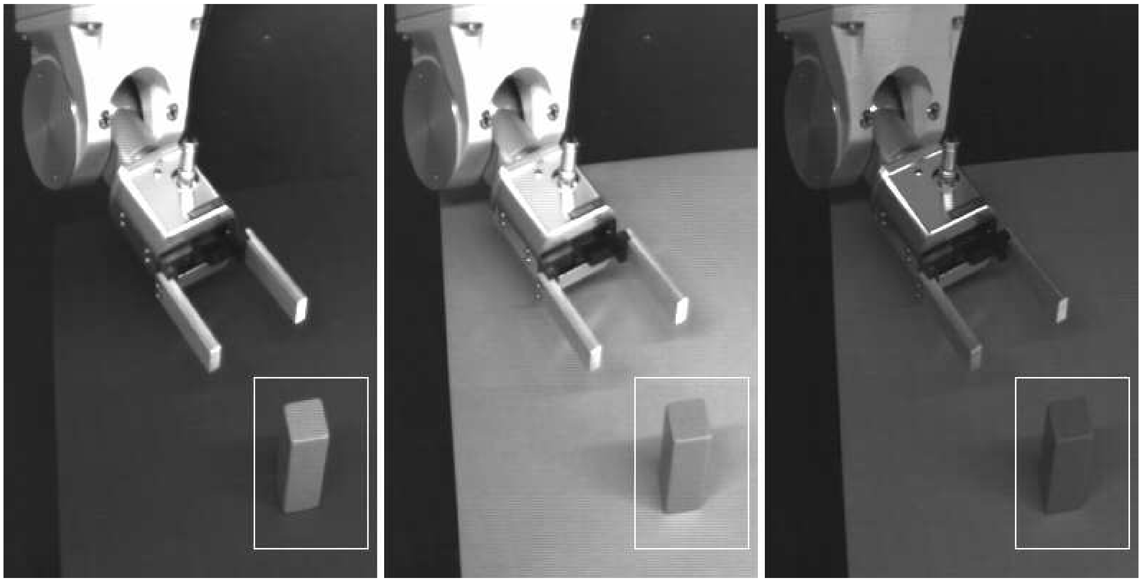

Figure 1: Three sample images with different object background or illumination.

2.2 Transforming the sample images

We apply an operator to the sample images in order to transform the gray levels of the depicted object into a specific structure, respectively. Based on this structure of the operator outcome it is intended to efficiently look for the two-dimensional depiction of the object in the image. For example, our operator transforms the gray levels of the depicted object into a unique peak structure with the peak being localized approximately at the center of the depicted object. The operator is designed on the basis of a gauss-modulated cosinus wave (for short, wavelet [Rioul and Vetterli, 1991]) which is two-dimensionally extended (see Figure 2). As a special case of the complex Gabor function (see [Reed and Wechsler, 1990]) we only take the real (cosinus) part into account:

$$G_{\sigma_1,\sigma_2,\lambda}(x,y) = e^{-\frac{1}{2}(\frac{x^2}{{\sigma_1}^2}+\frac{y^2}{{\sigma_2}^2})} cos(\frac{2\pi x}{\lambda})$$

The parameter λ defines the wavelength of the wavelet, σ_1 defines the turning point of the Gaussian in the spreading direction of the wavelet, and σ_2 defines the turning point of the Gaussian in normal direction. The spreading direction of the stated function is parallel to the X-axis, and if needed, by rotating the x and y components we can get any 2D direction. We are faced with the problem of allocating appropriate values to the parameters (λ, σ_1, and σ_2) to get a unique peak structure in the area of the depicted object (see Figure 3 for the following deduction).

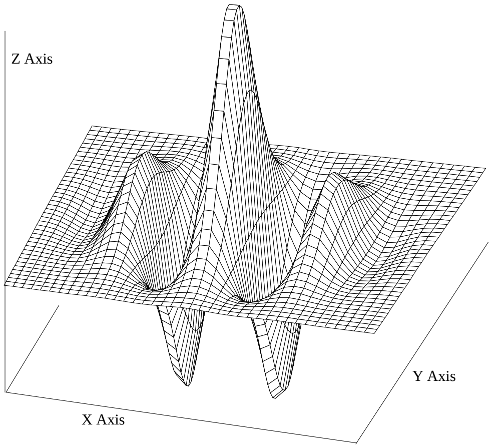

Figure 2: Gauss-modulated 2D cosinus function (wavelet).

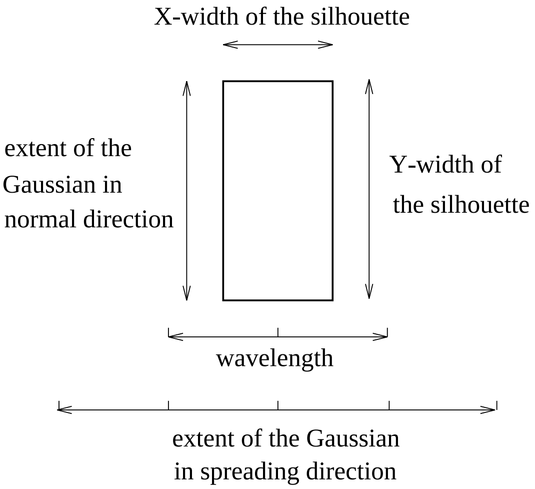
X-width of the silhouette
extent of the Gaussian in normal direction
Y-width of the silhouette
wavelength
extent of the Gaussian in spreading direction

Figure 3: Defining values for the parameters of the wavelet.

The value for λ is used to make the operator sensitive to objects having a certain extension in the direction of the X-axis. The values for σ_1 and σ_2 are used to specify the range of definition of the function in spreading direction and in normal direction. E.g., we define a finite extension of the Gaussian in spreading direction by $2 * \pi * \sigma_1$, and a finite extension in normal direction by $2 * \pi * \sigma_2$. Take X_w and Y_w to be the X-width and Y-width of the silhouette of the object (in pixel), and I_r the resolution of the image (in pixel). For example, in Figure 1 the extension of the silhouette is $X_w = 30$ Pixel and $Y_w = 80$ Pixel, respectively. The resolution is $I_r = 512$ Pixel.[2] The X-width and Y-width are divided by the image resolution to get normalized values X_{nw} and Y_{nw}. Taking the double of the normalized X-width as the wavelength and equating the extension of the Gaussian in spreading direction with the double wavelength we get values for λ and σ_1. Finally, in order to incorporate the Y-width of the silhouette we equate the extension of the Gaussian in the normal direction with the normalized Y-width of the silhouette and get a value for σ_2. Therefore, the following formulas arise:

$$X_{nw} = \frac{X_w}{I_r}, \quad Y_{nw} = \frac{Y_w}{I_r}, \quad \lambda = 2 * X_{nw}, \quad \sigma_1 = \frac{2 * X_{nw}}{\pi}, \quad \sigma_2 = \frac{Y_{nw}}{2 * \pi}$$

Taking these formulas into account we transform both an image and the wavelet into the frequency space, make a convolution, transform the image back to the spatial space and get a unique peak structure for the depicted object. Figure 4 shows the operator response for the left image in Figure 1, and Figure 5 shows especially for the area of the object the structure of operator response in a three-dimensional manner.

[2] In Figure 1, we only show 260x400 Pixel subimages of 512x512 Pixel images.

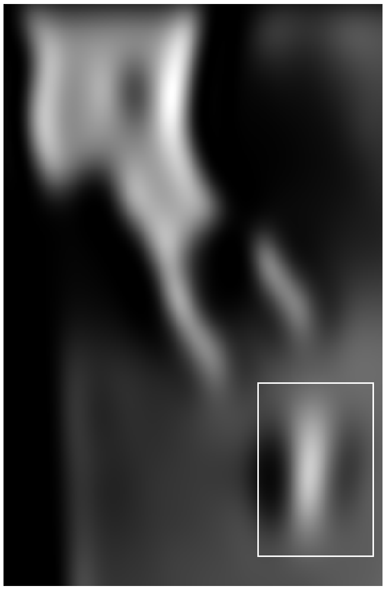

Figure 4: Operator response of a wavelet directed parallel to the X-axis, depicted as gray levels.

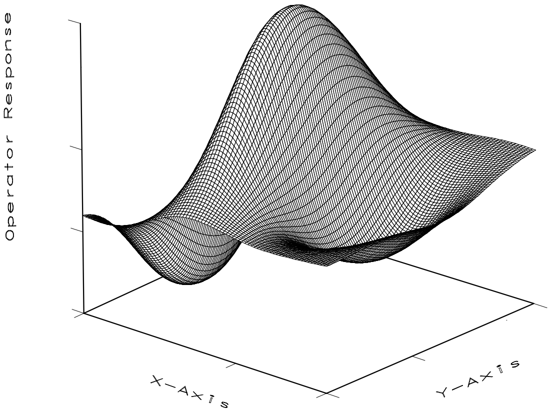

Figure 5: Operator response of a wavelet directed parallel to the X-axis, depicted for a small patch in three dimensions.

So far we have parameterized the wavelet for a spreading direction along the X-axis. Alternatively, we can define a wavelet spreading along the Y-axis. The values for the parameters have to be defined in a similar manner. In particular, now the λ-value depends on the Y-width of the object silhouette. The response of this operator in Figure 6 is different compared that in Figure 5. Although the peak is approximately at the same position it is much harder to determine because of the lower curvature on the top. In order to reach a greater degree of robustness we use both wavelets in parallel and multiply both operator responses point by point (see Figure 7).

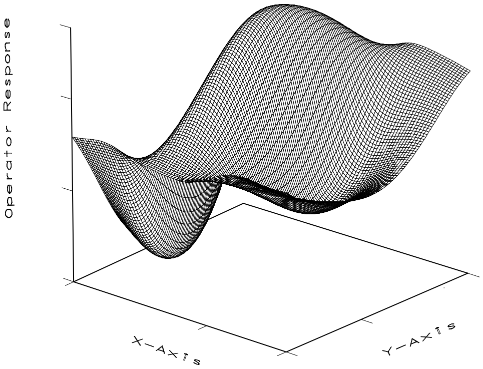

Figure 6: Operator response of a wavelet directed parallel to the Y-axis.

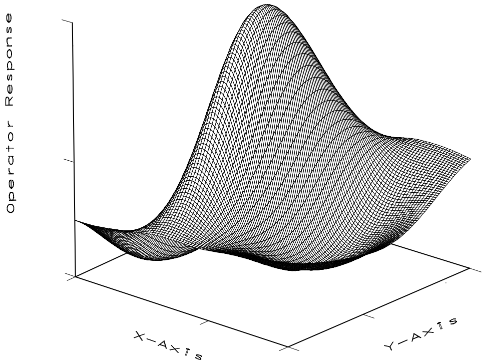

Figure 7: Operator response of two combined wavelets directed parallel to the X-Axis and the Y-axis.

2.3 Clustering specific patterns of the object

From each of the 16 sample images we are interested in the operator response pattern of a small rectangular area having the relevant object inside (see Figure 4). These specific patterns are extracted in order generate a training set from which to automatically learn the object detector. For example, Figure 8 shows the relevant patterns computed for the three sample images in Figure 1. In order to qualitatively show all 16 patterns we select the intensity structure along a middle staight line in X-direction and along a middle staight line in Y-direction (see Figure 9). Figure 10 shows the overlay of intensity structures in X-direction and Figure 11 in Y-direction.

According to the mentioned approach for learning an RBF network, we first have to group the 16 patterns for getting a smaller number of typical patterns. In this work, we used a clustering approach which is similar to the error-based ISODATA clustering algorithm in [Schalkoff, 1992, pp. 109-125]. The algorithm initially groups the patterns using a standard K-means clustering approach. Then, clusters exhibiting large variances are split in two, and clusters that are too close together are merged. Next, K-means is reiterated taking the new clusters into account. This sequence is repeated until no more clusters are split or merged.

The algorithm groups the 16 patterns into four clusters and computes for each cluster a typical pattern. The typical pattern of the four clusters are shown in Figure 12, respectively. Additionally, associated with each typical pattern is variance information representing point for point the deviations from the mean value within the cluster. Figure 13 shows for the second cluster in Figure 12 the intensity structure of the typical pattern (middle line) together with the deviations for each point (upper and lower line).[3] The four patterns are used as centers of four Gaussians buildung up the hidden layer of the RBF network. The dimension number of a Gaussian is equal to the number of pixels in the pattern. For simplicity, each of the four Gaussians will be defined symmetrically, therefore only one σ-value has to be computed, respectively. We do specify this value by the mean deviation over all points of a typical pattern.

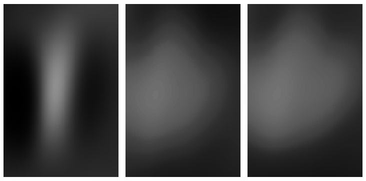

Figure 8: Three exemplary operator response patterns.

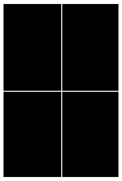

Figure 9: Two orthogonal straight lines for selectively showing the intensity structures.

[3] Notice, only the intensity structure and the deviations along the middle straight line in X-direction is depicted.

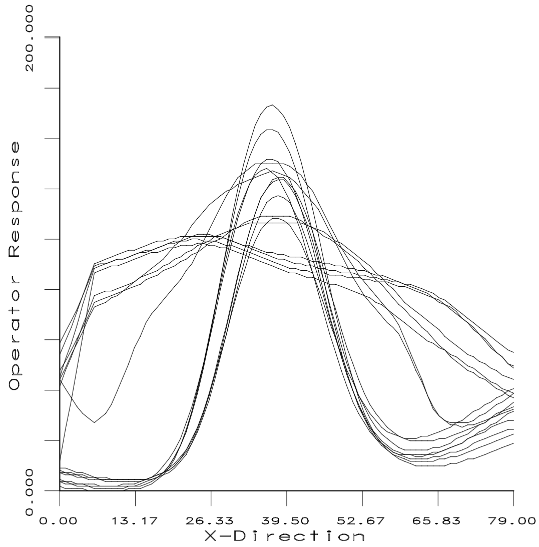

Figure 10: Overlay of 16 intensity structures in X-direction.

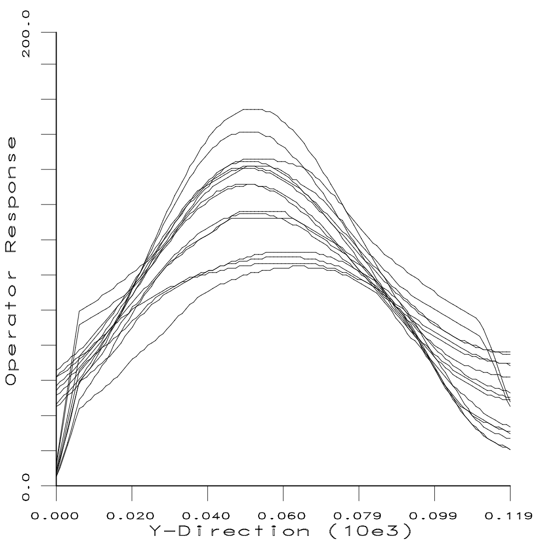

Figure 11: Overlay of 16 intensity structures in Y-direction.

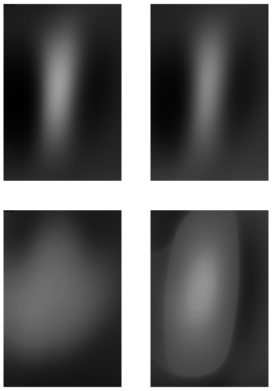

Figure 12: Typical patterns of four clusters.

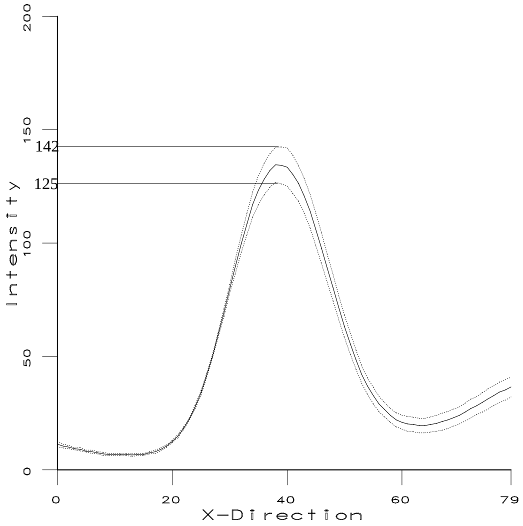

Figure 13: Deviations between the patterns of the second cluster.

2.4 Assembling the object detector

According to the approach for learning an RBF network, we finally assign to each training pattern a desired output vector. Based on the difference between the actual outcome and the desired outcome the weights for combining the nodes of the hidden layer are trained. E.g., the LMS rule can be used for tuning these weights between the hidden layer and the output layer in a supervised manner [Hush and Horne, 1993]. In our specific application the output vector is only a scalar taking a real value between 0 and 1. This value codifies a probability as to whether the object of interest is inside a certain image area. Therefore, we assign the real value 1 to each of the 16 patterns and use the LMS rule for learning the weights.[4]

3 Using the learned object detector

The learned detector is defined as an RBF network and is mainly composed of a number of typical patterns together with possible deviations from each typical pattern. Figure 14 illustrates the useful effect of the detector. The image on the right shows the well known object under arbitrary illumination and background. The square in the object silhouette marks the location of the object, and the accompanying bright gray value codifies a high probability.

Based on the specific arrangement of the detector we implemented an efficient method to look for the depicted object in the image. The efficiency arises from three factors. First, the typical patterns of the detector can be used in parallel for matching with the image. Second, the unique peak type of the operator response (of the depicted object) is to a certain degree invariant with respect to shrinking the image. In fact, all the experiments done for this report can been carried out with a shrinking factor of 8 leading to similar results. Third, by making use of the unique peak structure of the typical patterns we can efficiently select small areas of the image as candidate places for possible locations of the object.

The approach mentioned in the last statement is as follows: First, we take a new image of the object using any illumination and object background (see the right image in Figure 14). Second, we apply the operator used during the training phase (see operator response in the left image in Figure 14). Third, for each typical pattern of the learned detector we take the point having the maximal intensity value and compute for this point an interval of possible intensity deviations. For example, in Figure 13 the two horizontal lines define variations of the maximal response in the interval beginning at the intensity of 125 and ending at 142. Fourth, we look for local maxima in the filtered image, and only those local maxima being inside the relevant interval will be considered in detail. For example, the dark areas in the middle image of Figure 14 depict the operator response in the intensity interval between 125 and 142. Fifth, only the image areas which surround a relevant local maximum must be put into the RBF network. The network computes a hypothesis for the existence of the object in an area. Sixth, the image area for which the RBF network computes a global maximal value is the location of the desired object (see the square in the object of the right image in Figure 14).

Figure 15 shows three additional applications of the detector. First, the object used during the training phase has been rotated and under this appearance it will be localized only with a low degree of evidence (see the dark square in the left image of Figure 15). The low evidence is a desired result because the detector has not been trained on rotated objects and therefore the learned detector is not robust with

[4] The number of output nodes is equal to the number of objects which must be detected. For short, we illustrate the approach only for one object and therefore applying the LMS-rule is trivial.

respect to object rotation. Second, the operator is applied to an image depicting a new object whose brightness is higher than that of the training object. Once again, the detector computes low evidence for the existence of the desired object in the image (the dark square on the object of the middle image in Figure 15 indicates a low evidence). Third, the operator is applied to an image depicting another new object whose shape is different compared to the shape of the training object. The detector can not find any hints at all for the existence of the desired object in the image (see right image in Figure 15).

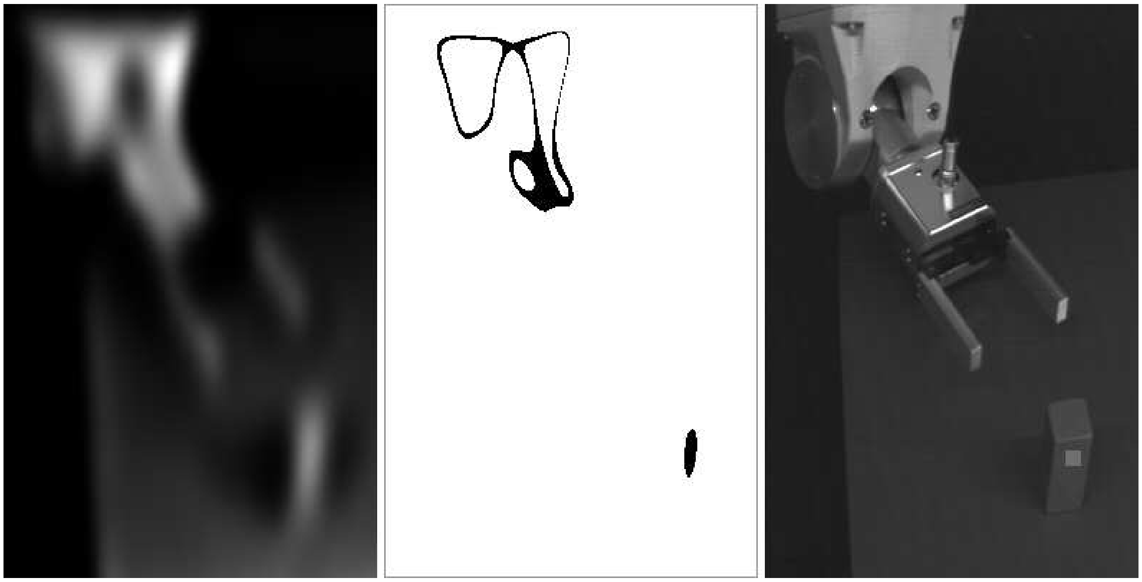

Figure 14: Operator response of a test image (left), candidate places of possible object locations (middle), square patch on the object specifies the location (right).

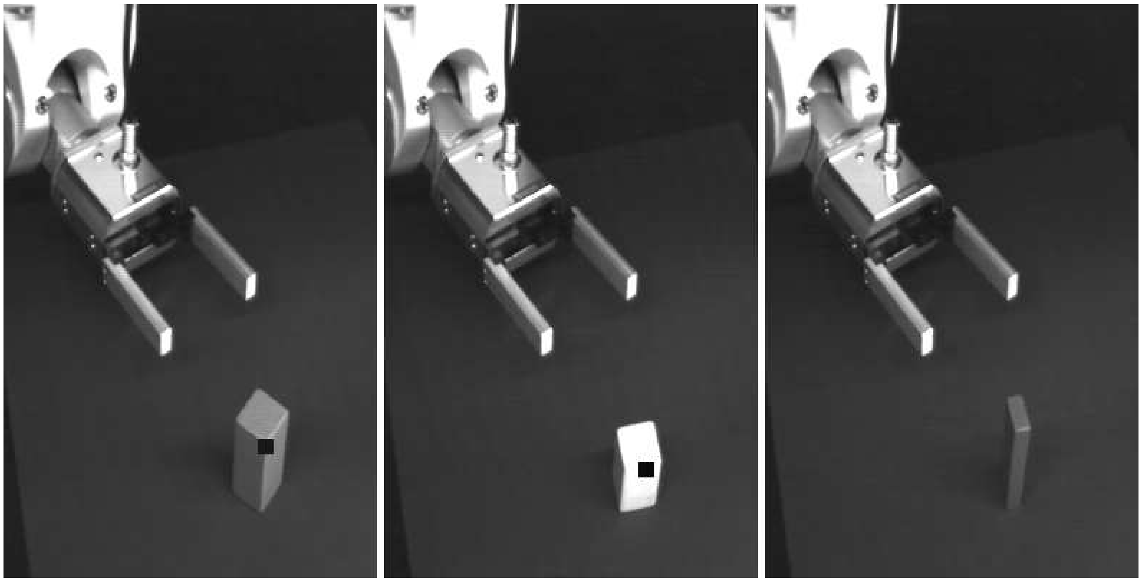

Figure 15: Applying the detector to images of a well known but rotated object and two new objects.

4 Conclusion

We have presented an approach for object detection which does not require a priori knowledge of the three-dimensional geometric shape. Rather the meaning of an object is grounded in photometric appearance. That is, the detector for an object must be learned online on the basis of elementary filter operators. For representing and learning the detector an RBF network is used. Object detection has been performed successfully in images taken under arbitrary illumination and object background. In our current and future work we would like to extend this approach for detecting objects under arbitrary orientation and scale.

References

[Dickinson *et al.*, 1990] S. Dickinson, A. Pentland, and A. Rosenfeld. A Representation for Qualitative 3D Object Recognition Integrating Object-Centered and Viewer-Centered Models. In K. Leibovic, editor, *Science of Vision*, pages 398–421. Springer Verlag, Berlin, 1990.

[Faugeras, 1993] O. Faugeras. *Three-Dimensional Computer Vision.* The MIT Press, Massachusetts, 1993.

[Hush and Horne, 1993] D. Hush and B. Horne. Progress in Supervised Neural Networks. *IEEE Signal Processing Magazine*, 10:8–39, January 1993.

[Maxwell and Shafer, 1994] B. Maxwell and S. Shafer. A Framework for Segmentation Using Physical Models of Image Formation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–368, 1994.

[Murase and Nayar, 1995] H. Murase and S. Nayar. Visual Learning and Recognition of 3D Objects from Appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[Poggio and Girosi, 1990] T. Poggio and F. Girosi. Networks for Approximation and Learning. *Proceedings of the IEEE*, 78:1481–1497, 1990.

[Reed and Wechsler, 1990] T. Reed and H. Wechsler. Segmentation of Textured Images and Gestalt Organization Using Spatial/Spatial-Frequency Representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:1–12, 1990.

[Rioul and Vetterli, 1991] O. Rioul and M. Vetterli. Wavelets and Signal Processing. *IEEE Signal Processing Magazine*, pages 14–38, October 1991.

[Schalkoff, 1992] R. Schalkoff. *Pattern Recognition - Statistical, Structural, and Neural Approaches.* John Wiley and Sons, New York, 1992.