Marcin Grzegorzek, Michael Reinhold, and Heinrich Niemann

Feature Extraction with Wavelet Transformation for Statistical Object Recognition

appeared in:
Proceedings of the 4th International Conference
on Computer Recognition Systems
Rydzyna, Poland
pp. 161-168

Feature Extraction with Wavelet Transformation for Statistical Object Recognition

Marcin Grzegorzek [*], Michael Reinhold, and Heinrich Niemann

Chair for Pattern Recognition,
University of Erlangen-Nuremberg,
Martensstr. 3, 91058 Erlangen, Germany,
{grzegorz,reinhold,niemann}@informatik.uni-erlangen.de

Summary. In this paper we present a statistical approach for localization and classification of 3-D objects in 2-D images with real heterogeneous background. Two-dimensional local feature vectors are computed directly from pixel intensities in square gray level images with the wavelet multiresolution analysis. We use three different resolution levels for the feature computation. For the first one local neighborhoods of size 8×8 pixels, for the second one 4×4 pixels, and for the third one 2×2 pixels are taken into account. Then we define an object area as a function of 3-D transformations and represent the feature vectors as density functions. Our localization and classification algorithm uses a combination of object models created for the three different resolutions in the training phase. Experiments made on a real data set with 42240 images show that the recognition rates are much better using the resolution combination of the wavelet transformation.

1 Introduction

The automatic localization and classification of objects in real environment images is becoming more important lately. Object recognition systems can be applied for example: to face classification, to localization of obstacles on the road with a camera mounted on a driving car, to service robotics [10], to handwriting recognition, and so on. There exist two main approaches for localization and classification of 3-D objects in 2-D gray level images: based on results of a segmentation process [5], or directly on the object appearance [3, 8]. The appearance-based methods compute feature vectors from pixel intensities in images without previous segmentation process. Some of them use only one global feature vector for the whole image (e.g. eigenspace approach [2]), other describe objects with more local features (e.g. neural networks [7]).

[*] This work was partly funded by the German Research Foundation (DFG) Graduate Research Center 3D Image Analysis and Synthesis

In the present work two-dimensional local feature vectors are computed directly from pixel intensities (appearance-based approach) using the wavelet multiresolution analysis [6] and modeled by density functions [4]. The main advantage of the local feature vectors is that a local disturbance only affects the feature vectors in a small region around it. In contrast to this a global feature vector can totally change, if only one pixel in the image varies. We introduce feature extraction on three different resolution levels in each image and create three statistical object models for each object class in the training phase. Our new algorithm for object localization and classification uses a combination of the object models obtained for these different resolution levels, which significantly improves the recognition rates.

In Sect. 2 the training of statistical object models with all its steps, especially the computation of feature vectors, is presented. Beginning with the computation of the object density value, through the recognition algorithm for one resolution, until the combination of object models for different resolutions Sect. 3 describes the whole recognition phase. The experimental evaluation of the new approach made on a large image data set can be found in Sect. 4. Sect. 5 closes our contribution with conclusions.

2 Training of Statistical Object Models

In order to learn object models we preprocess training images (Sect. 2.1), compute feature vectors in the preprocessed images (Sect. 2.2), define an object area (Sect. 2.3), and model the feature vectors by density functions (Sect. 2.4). At the end of the training process we get three statistical models for each object class, because the feature vector calculation is applied for three different resolutions of the wavelet transformation.

2.1 Image Sample Set for Training

First we define a set of object classes $\Omega = \{\Omega_1, \ldots, \Omega_\kappa, \ldots, \Omega_k\}$ and take training images of them on a dark background. The original training images are preprocessed by converting them to gray level images sized $2^n \times 2^n$ pixels, where $n \in \{6, 7, 8, 9\}$. Then we set one image $\boldsymbol{g}_{\kappa,i}$ for each object class Ω_κ as a reference image. With a pose of an object in the image $\boldsymbol{g}_{\kappa,j}$ we denote the 3-D transformation (translations and rotations) that maps the object in the reference image $\boldsymbol{g}_{\kappa,i}$ to the object in $\boldsymbol{g}_{\kappa,j}$. The 3-D transformation can be described with translations $\boldsymbol{t} = (t_x, t_y, t_z)^{\mathrm{T}}$ and rotations $\boldsymbol{\phi} = (\phi_x, \phi_y, \phi_z)^{\mathrm{T}}$. The x and y axes lie in the image plane, and the z axis is orthographic to the image plane. With rotation around the x and y axes as well as with translation along the z axis (scaling) change the size and appearance of the object in the image. These are the so called external transformation parameters ($t_{ext} = t_z$ and $\boldsymbol{\phi}_{ext} = (\phi_x, \phi_y)^{\mathrm{T}}$). The remaining transformation parameters are called internal and do not change the object size and appearance. Until the end of Sect. 2 the number of object class κ is omitted, because the training is identical for all object classes.

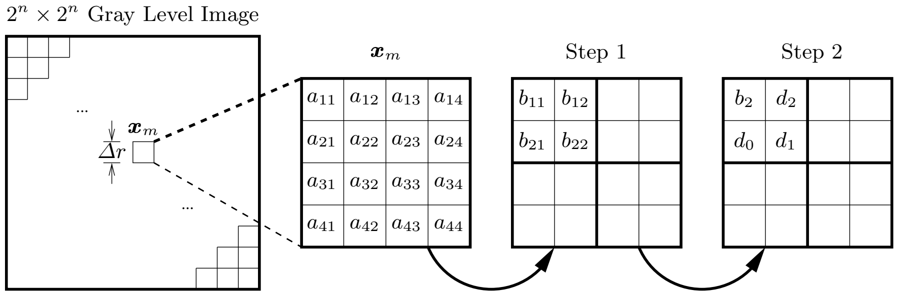
$2^n \times 2^n$ Gray Level Image
$\boldsymbol{x}_m$
Δr
$\boldsymbol{x}_m$
Step 1
Step 2
a_{11} a_{12} a_{13} a_{14}
a_{21} a_{22} a_{23} a_{24}
a_{31} a_{32} a_{33} a_{34}
a_{41} a_{42} a_{43} a_{44}
b_{11} b_{12}
b_{21} b_{22}
b_2 d_2
d_0 d_1

Fig. 1. Computation of a feature vector at a grid point $\boldsymbol{x}_m$ with a Haar wavelet (scale $s = 2$). In the first step low-pass coefficients $b_{ij} = 0.25 \cdot \sum_{k=0}^{1} \sum_{l=0}^{1} a_{k+2i-1,l+2j-1}$ are computed from the gray values a_{ij}. After the second step $b_2 = 0.25 \cdot \sum_{k=1}^{2} \sum_{l=1}^{2} b_{kl}$ is the low-pass coefficient. The other coefficients result from combinations of low-pass and high-pass filtering ($d_0 = 0.25 \cdot [-(b_{11}+b_{12})+(b_{21}+b_{22})]$, $d_1 = 0.25 \cdot [-(-b_{11}+b_{12})+(-b_{21}+b_{22})]$, $d_2 = 0.25 \cdot [(-b_{11}+b_{12})+(-b_{21}+b_{22})]$)

2.2 Computation of Feature Vectors

In all preprocessed training images feature vectors are computed using the wavelet transformation [1]. For the calculation of these vectors a grid with the size $\Delta r = 2^s$, where s is the index of the scale, is laid on an image (Fig. 1). On each grid point $\boldsymbol{x}_m$ a two-dimensional local feature vector $\boldsymbol{c}_m = \boldsymbol{c}(\boldsymbol{x}_m)$ is calculated. For this purpose we perform s-times the wavelet multiresolution analysis [6] using Haar wavelet. The components of the feature vector $\boldsymbol{c}_m$ are given by:

$$\boldsymbol{c}_m = \boldsymbol{c}(\boldsymbol{x}_m) = \begin{pmatrix} \ln(2^{-s}\,|b_{s,m}|) \\ \ln[2^{-s}\,(|d_{0,s,m}| + |d_{1,s,m}| + |d_{2,s,m}|)] \end{pmatrix} \tag{1}$$

$b_{s,m}$ is the low-pass coefficient and $d_{0\ldots2,s,m}$ result from combinations of low-pass and high-pass filtering. An illustration for the computation of a feature vector for $s = 2$ can be seen in Fig. 1 (indexes m and s are omitted). Our algorithm works with three resolution levels of the wavelet transformation: L_3 ($s = 3$), L_2 ($s = 2$), L_1 ($s = 1$). For each of these resolutions object models are created. The following training steps are identical for all resolution levels.

2.3 Modeling of Object Area

For the object model we want to consider only those feature vectors that belong to the object and not to the background. For each feature vector $\boldsymbol{c}_m$ in each external training pose $(\boldsymbol{\phi}_{ext,t}, \boldsymbol{t}_{ext,t})$ (for each training image) a discrete assignment function is defined [8]:

$$\widehat{\xi}_m(\boldsymbol{\phi}_{ext,t}, \boldsymbol{t}_{ext,t}) = \begin{cases} 1 & \text{if} \quad c_{m,1}(\boldsymbol{\phi}_{ext,t}, \boldsymbol{t}_{ext,t}) \geq S_t \\ 0 & \text{if} \quad c_{m,1}(\boldsymbol{\phi}_{ext,t}, \boldsymbol{t}_{ext,t}) < S_t \end{cases} \tag{2}$$

S_t is chosen manually. In the test images objects appear not only in the training poses, but also between them. In order to localize such objects we construct a continuous assignment function $\xi_m(\boldsymbol{\phi}_{ext}, \boldsymbol{t}_{ext})$ using values of

$\widehat{\xi}_m(\boldsymbol{\phi}_{ext,t}, t_{ext,t})$ by interpolation with trigonometric functions. The set of feature vectors belonging to the object for the given external pose $(\boldsymbol{\phi}_{ext}, t_{ext})$ (called object area $O(\boldsymbol{\phi}_{ext}, t_{ext})$) can be now determined with the following rule:

$$\xi_m(\boldsymbol{\phi}_{ext}, t_{ext}) \geq S_O \Longrightarrow \boldsymbol{c}_m(\boldsymbol{\phi}_{ext}, t_{ext}) \in O(\boldsymbol{\phi}_{ext}, t_{ext}) \qquad (3)$$

The threshold value S_O is also chosen manually. In the case of internal transformations the object area does not change the size and can be translated and rotated with these transformations. So, we can write the object area as a function of all transformation parameters: $O(\boldsymbol{\phi}, \boldsymbol{t})$.

2.4 Density Functions of the Feature Vectors

All feature vectors computed in the training phase (1) are interpreted as random variables. The object feature vectors are modeled with the normal distribution [4]. For each object feature vector $\boldsymbol{c}_m \in O$ we compute a mean value vector $\boldsymbol{\mu}_m$ and standard deviation vector $\boldsymbol{\sigma}_m$. The density of the object feature vector can be written as: $p(\boldsymbol{c}_m) = p(\boldsymbol{c}_m|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m, \boldsymbol{\phi}, \boldsymbol{t})$. The feature vectors, which belong to the background are modeled with the equal distribution $p(\boldsymbol{c}_m) = p_b$.

3 Localization and Classification

After for each object class Ω_κ three corresponding object models $\mathcal{M}_{\kappa,s}$ $(s \in \{1, 2, 3\})$ were created, we can localize and classify objects in test images. At the beginning test images are preprocessed and feature vectors are computed (1) with the same method as in the training phase (Sect. 2.1 and 2.2). Then we start our recognition algorithm that uses only one of the trained object models for each object class (Sect. 3.2). After that the results are refined by using the combination of object models for different resolutions (Sect. 3.3). In both cases object density values (Sect. 3.1) for many pose and class hypotheses are needed.

3.1 Object Density Value

In order to compute the object density value for the class Ω_κ in the pose $(\boldsymbol{\phi}, \boldsymbol{t})$ for a given test image we determine the set of feature vectors that belong to the object $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M\}$ (object area $O_\kappa(\boldsymbol{\phi}, \boldsymbol{t})$, Sect. 2.3) and compute their values. Then we compare the calculated object feature vectors with the corresponding density functions stored in the object model $\mathcal{M}_{\kappa,s}$ and read density values for these vectors $(p(\boldsymbol{c}_1), p(\boldsymbol{c}_2), \ldots, p(\boldsymbol{c}_M))$. The object density value for the object class Ω_κ in the pose $(\boldsymbol{\phi}, \boldsymbol{t})$ can be computed as:

$$p(C|\boldsymbol{B}_{\kappa,s}, \boldsymbol{\phi}, \boldsymbol{t}) = \prod_{i=0}^{M} \max\{p(\boldsymbol{c}_i), p_b\} \qquad (4)$$

$\boldsymbol{B}_{\kappa,s}$ comprehends the trained mean value vectors and standard deviation vectors from $\mathcal{M}_{\kappa,s}$ and p_b is the background density value (Sect. 2.4).

3.2 Recognition Algorithm for One Resolution

The localization and classification algorithm for one resolution (one object model) is realized with the maximum likelihood estimation [9] and can be described with the following equation:

$$(\widehat{\kappa}, \widehat{\boldsymbol{\phi}}, \widehat{\boldsymbol{t}}) = \underset{\kappa}{\operatorname{argmax}}\{\underset{(\boldsymbol{\phi},\boldsymbol{t})}{\operatorname{argmax}}\, G(p(C|\boldsymbol{B}_{\kappa,s}, \boldsymbol{\phi}, \boldsymbol{t}))\} \tag{5}$$

$\widehat{\kappa}$ is the classification result and $(\widehat{\boldsymbol{\phi}}, \widehat{\boldsymbol{t}})$ is the localization result. First the object density (normalized by G) is maximized according to the pose parameters $(\boldsymbol{\phi}, \boldsymbol{t})$ and then to the class κ. The norm function G is defined by:

$$G(p(C|\boldsymbol{B}_{\kappa,s}, \boldsymbol{\phi}, \boldsymbol{t})) = \sqrt[M]{p(C|\boldsymbol{B}_{\kappa,s}, \boldsymbol{\phi}, \boldsymbol{t})} \tag{6}$$

M is the number of feature vectors belonging to the object area $O_\kappa(\boldsymbol{\phi}, \boldsymbol{t})$. This norm function decreases the dependency between the maximization result and the object area size.

3.3 Combination of Object Models for Different Resolutions

Our recognition algorithm uses a combination of object models obtained for different resolutions of the wavelet transformation. We start for the resolution level L_3 ($s = 3$), where the feature vectors are computed from local neighborhoods of size 8 × 8 pixels. According to Sect. 3.2 we find a class and pose of the object in the scene $(\widehat{\kappa}_3, \widehat{\boldsymbol{\phi}}_3, \widehat{\boldsymbol{t}}_3)$ (5). Then the maximum likelihood estimation is applied for all object classes for the resolution level L_2 ($s = 2$) in the small neighborhood of the localization result from L_3 $(\widehat{\boldsymbol{\phi}}_3, \widehat{\boldsymbol{t}}_3)$ [1]. A refined recognition result $(\widehat{\kappa}_2, \widehat{\boldsymbol{\phi}}_2, \widehat{\boldsymbol{t}}_2)$ is obtained. Analogical for the resolution level L_1 ($s = 1$) we maximize the object density (normalized by the function G (6)) only in the small neighborhood of $(\widehat{\boldsymbol{\phi}}_2, \widehat{\boldsymbol{t}}_2)$ and get the finally recognition result $(\widehat{\kappa}_1, \widehat{\boldsymbol{\phi}}_1, \widehat{\boldsymbol{t}}_1)$.

4 Experiments and Results

We verified our approach on a 3D-REAL-ENV image data base (Sect. 4.1). Using the combination of object models for different resolutions (Sect. 3.3) the execution time increases (Sect. 4.2), but we obtain better localization and classification rates (Sect. 4.3).

4.1 Image Data Base

3D-REAL-ENV (Image Data Base for 3-D Object Recognition in Real World Environment) consists of 10 objects depicted in Fig. 2. We made the experi-

[1] The small neighborhood of $(\widehat{\boldsymbol{\phi}}_s, \widehat{\boldsymbol{t}}_s)$ is defined for rotations with $\pm 5(s-1)[^\circ]$, and for translations with $\pm 2^{s-1}$ pixels.

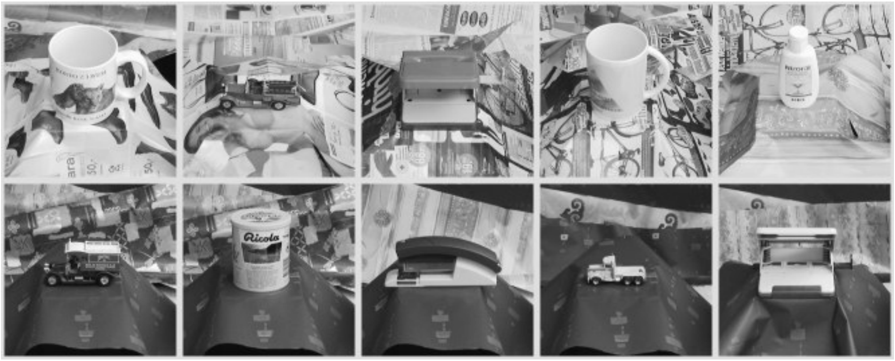

Fig. 2. 10 object classes used for experiments. In the first row examples of test images with "more heterogeneous" background can be seen. From left: bank cup, toy fire engine, green puncher, siemens cup, nizoral bottle. The second row contains examples of test images with "less heterogeneous" background. From left: toy passenger car, ricola container, stapler, toy truck, white puncher.

ments using gray level images of size 256 × 256 pixels. The pose of an object is defined with external rotations and internal translations $(\phi_x, \phi_y, t_x, t_y)^{\mathrm{T}}$. For the training we took 3360 images of each object with two different illuminations. The objects were put on a turntable ($0° \leq \phi_{table} \leq 360°$) and a robot arm with a camera was moved from horizontal to vertical ($0° \leq \phi_{arm} \leq 90°$). The angle between two adjacent training viewpoints amounts to 4.5°. For the tests 2880 images with homogeneous, 2880 images with "less heterogeneous", and 2880 with "more heterogeneous" background were taken. In the test images with "less heterogeneous" background the objects are easier to distinguish from the background than in the test images with "more heterogeneous" background. The object poses and the illumination in the test images were different from the training viewpoints and illuminations.

4.2 Execution Time

In Table 1 we compare the execution time in the recognition phase for different resolution levels and their combinations. The finest resolution level L_1 is very time consuming and can be used only in combination with L_2 and L_3.

Table 1. Execution time of the localization and classification algorithm for L_3, L_2, L_1, and combinations L_3–L_2, L_3–L_2–L_1.

Pentium 4 2.66 GHz 512 MB RAM	$\boldsymbol{L_3}$	$\boldsymbol{L_2}$	$\boldsymbol{L_1}$	$\boldsymbol{L_3}$–$\boldsymbol{L_2}$	$\boldsymbol{L_3}$–$\boldsymbol{L_2}$–$\boldsymbol{L_1}$
Recognition in 1 Test Image	3.6s	124.7s	73.7m	24.0s	96.5s

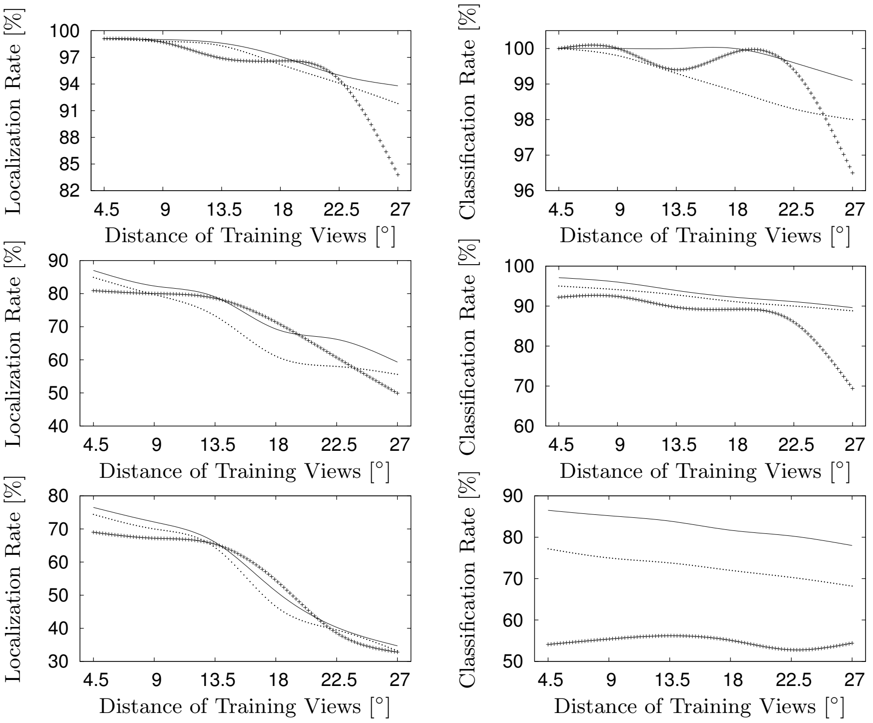

Fig. 3. Localization and classification rates depending on the distance of the training views for test images with homogeneous (first row), "less heterogeneous" (second row), and "more heterogeneous" background (third row). (— combination of L_3–L_2–L_1; $\cdots$ combination of L_3–L_2; +++ resolution level L_3).

4.3 Localization and Classification Rates

We count a localization result as correct, if the error for the external rotations (ϕ_x, ϕ_y) is not bigger than 15° and the error for the internal translations is not bigger than 10 pixels. Fig. 3 presents the recognition rates depending on the distance of the training views for test images with homogeneous, "less heterogeneous", and "more heterogeneous" background. The advantage of the combination of the resolution levels is visible especially for classification. Table 2 contains the recognition rates for 4.5° distance of training views.

Table 2. Recognition rates for 4.5° distance of training views.

Distance	**Localization**			**Classification**		
of Training	**Hom.**	**Less Het.**	**More Het.**	**Hom.**	**Less Het.**	**More Het.**
Views 4.5°	**Back.**	**Back.**	**Back.**	**Back.**	**Back.**	**Back.**
$\boldsymbol{L_3}$	99.1%	80.9%	69.0%	100%	92.2%	54.1%
$\boldsymbol{L_3}$–$\boldsymbol{L_2}$	99.1%	84.9%	74.4%	100%	95.0%	77.2%
$\boldsymbol{L_3}$–$\boldsymbol{L_2}$–$\boldsymbol{L_1}$	99.1%	87.0%	76.5%	100%	97.1%	86.5%

5 Conclusions

In this article a powerful statistical, appearance-based approach for 3-D object localization and classification in images with heterogeneous background is presented. After computation of local feature vectors for three different resolutions of the wavelet transformation we define an assignment function, which assigns the features to the object or to the background, and statistically model them by density functions. Our new algorithm for localization and classification of objects uses a combination of the statistical models obtained for the three resolutions. In the experiments we showed that the new algorithm brings an improvement of the recognition rates, especially for the classification, in relatively short execution time. In the future we will introduce color modeling to our system, because the color information of objects is presently lost in the image preprocessing step.

References

1. C. Chui. *An Introduction to Wavelets.* Academic Press, San Diego, USA, 1992.
2. Ch. Gräßl, F. Deinzer, and H. Nieman. Continuous parametrization of normal distribution for improving the discrete statistical eigenspace approach for object recognition. In V. Krasnoproshin, S. Ablameyko, and J. Soldek, editors, *Pattern Recognition and Information Processing 03*, pages 73–77, Minsk, Belarus, Mai 2003.
3. R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):449–465, April 2004.
4. M. Grzegorzek, F. Deinzer, M. Reinhold, J. Denzler, and H. Niemann. How fusion of multiple views can improve object recognition in real-world environments. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, and R. Westermann, editors, *Vision, Modeling, and Visualization 2003*, pages 553–560, Munich, Germany, November 2003. Aka/IOS Press, Berlin, Amsterdam.
5. J. Kerr and P. Compton. Toward generic model-based object recognition by knowledge acquisition and machine learning. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 9–15, Acapulco, Mexico, August 2003.
6. S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
7. S. Park, J. Lee, and S. Kim. Content-based image classification using a neural network. *Pattern Recognition Letters*, 25(3):287–300, February 2004.
8. M. Reinhold. *Robust Probabilistic Appearance-Based Object Recognition.* Logos Verlag, Berlin, Germany, 2004.
9. A. R. Webb. *Statistical Pattern Recognition.* John Wiley & Sons Ltd, Chichester, England, 2002.
10. M. Zobel, J. Denzler, B. Heigl, E. Nöth, D. Paulus, J. Schmidt, and G. Stemmer. Mobsy: Integration of vision and dialogue in service robots. *Machine Vision and Applications*, 14(1):26–34, 2003.