Recognition of Objects Represented in Different Color Spaces

Marcin Grzegorzek; Institute for Web Science and Technologies, University of Koblenz-Landau; Koblenz, Germany
Alexandra Wolyniec; Institute for Computational Visualistics, University of Koblenz-Landau; Koblenz, Germany
Frank Schmitt; Institute for Computational Visualistics, University of Koblenz-Landau; Koblenz, Germany
Dietrich Paulus; Institute for Computational Visualistics, University of Koblenz-Landau; Koblenz, Germany

Abstract

In this article we present a statistical framework for automatic classification and localization of 3D objects in 2D images. The new functionality of the framework allows us to use objects represented in different color spaces including gray level, RGB, and Lab formats. First, the objects are preprocessed and described by local wavelet features. Second, statistical modeling of these features under the assumption of their normal distribution is performed in a supervised way. The resulting probability density functions are determined by the maximum likelihood estimation. The density functions describe a particular object class from a particular training viewpoint. In the recognition phase, local feature vectors are computed from an image with an unknown object in an unknown pose. Those features are then evaluated against the trained density functions which yields the classes and the poses of objects found in the scene. Experiments performed for more than 40.000 images with real heterogeneous backgrounds have delivered very good classification and localization rates for all investigated object representations. Moreover, they brought us to interesting conclusions considering the general performance of statistical recognition systems for different image representations.

Introduction

One of the most fundamental problems of computer vision is the recognition of objects in digital images. Throughout this paper the term object recognition comprehends both, the classification and the localization of objects. The task of object classification is to determine the classes of objects occurring in the image f from a set of predefined object classes $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_\kappa, \ldots, \Omega_{N_\Omega}\}$. Generally, the number of objects in a scene is unknown, however, in this work we assume that exactly one object is expected in an image. In the case of object localization, the recognition system estimates the poses of objects in the image, whereas the object classes are assumed to be known. The object poses are defined relatively to each other with a 3D translation vector $t = (t_x, t_y, t_z)^{\mathrm{T}}$ and a 3D rotation vector $\phi = (\phi_x, \phi_y, \phi_z)^{\mathrm{T}}$ in a coordinate system with an origin placed in the image center [1]. Figure 1 visualizes this definition.

For recognition of 3D objects in 2D images, two main approaches are known in computer vision: based on the result of object segmentation (shape-based), or by directly using the object texture (texture-based). Shape-based methods make use of geometric features such as lines or corners extracted by segmentation operations. These features as well as relations between them are used for object description [2]. However, the segmentation-based approach often suffers from errors due to loss of image details or other inaccuracies resulting from the segmentation process. Texture-based approaches avoid these disadvantages by using the image data, i. e., the pixel values, directly without a previous segmentation step. For this reason the texture-based method for object recognition has been chosen to develop the system presented in this contribution.

The object recognition problem has been intensively investigated in the past. Many approaches to object recognition, like the one presented in this paper, are founded on probability theory [3], and can be broadly characterized as either generative or discriminative according to whether or not the distribution of the image features is modeled [4]. Generative models such as principal component analysis (PCA) [5], independent component analysis (ICA) [6] or non-negative matrix factorization (NMF) [7] try to find a suitable representation of the original data [8]. In contrast, discriminative classifiers such as linear discriminant analysis (LDA) [9], support vector machines (SVM) [10], or boosting [11] aim at finding optimal decision boundaries given the training data and the corresponding labels [8]. The system presented in this paper represents the generative approaches.

There are further interesting approaches for object recognition. Amit et al. proposes in [12] an algorithm for multi-class shape detection in the sense of recognizing and localizing instances from multiple shape classes. In [13] a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene is presented. In [14] the problem of detecting a large number of different classes of objects in cluttered scenes is taken into consideration. [15] proposes a mathematical framework for constructing probabilistic hierarchical image models, designed to accommodate arbitrary contextual relationships. In order to compare different methods for object recognition, in [16] a new database specifically tailored to the task of object categorization is presented. In [17] an object recognition system is described that uses a new class of local image features. The features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. In [18] a multi-class object detection framework whose core component is the nearest neighbor search over object part classes is presented.

Classification and localization of objects in images is a useful, and often indispensable step, for many real life computer vision applications. Algorithms for automatic computational object recognition can be applied in areas such as: face classification [19], fingerprint classification [20], handwriting recognition [21], service robotics [22], medicine [23], visual inspection [24], the automobile industry [25], etc. Although successful applications have been developed for some tasks, e. g., fingerprint classification, there are still many other areas that could potentially benefit from object recognition. The system described in this article has been tested in real application scenarios. One of these is the classification of artefacts following a visit to a museum, another is the analysis of metallography images from an ironworks.

Our experimental study on a dataset with more than 40.000 real-world images has shown that the classification and localization rates are dependent on the color space which is used for feature extraction. Therefore, in this paper we experimentally

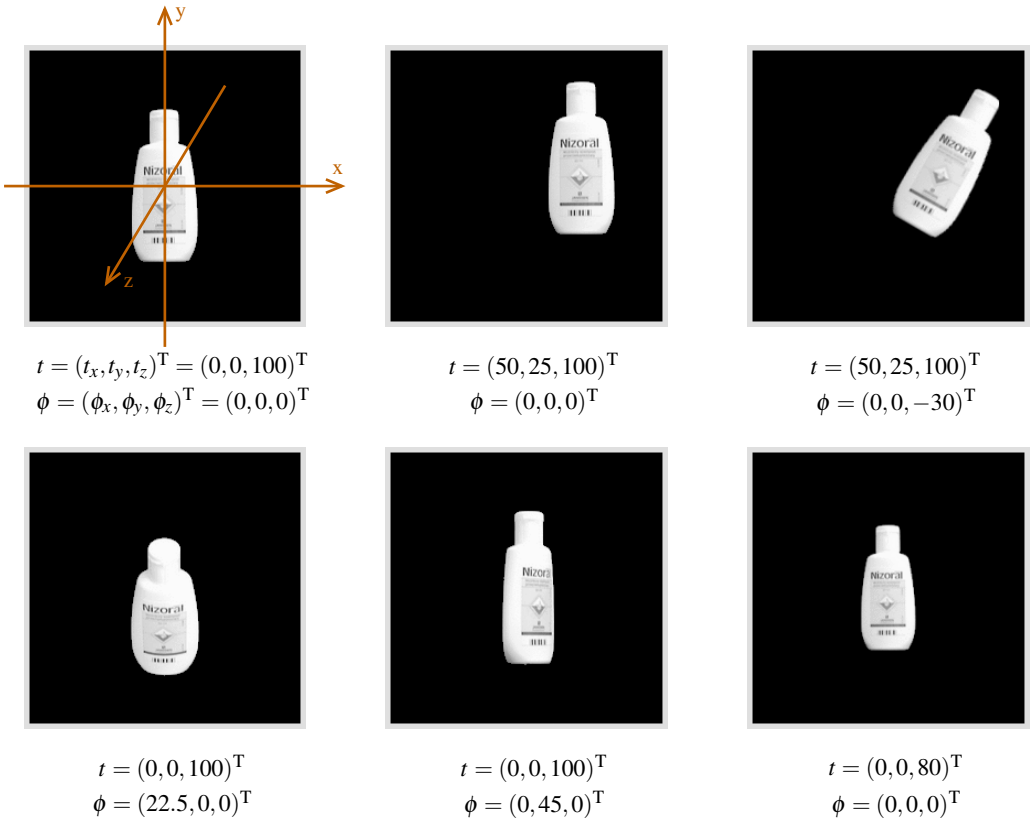

Bild 1. *Examples of object poses and their values. The components of the internal translation vector $t_{\mathrm{int}} = (t_x, t_y)^T$ are given in pixels, the components of the rotation vector $\phi = (\phi_x, \phi_y, \phi_z)^T$ in degrees [°], and the external translation (scaling) $t_{\mathrm{ext}} = t_z$ in percent [%] of a reference object size (top left).*

compare the system performance for gray level, RGB, and Lab images. The paper is structured as follows. Section presents the training phase of the system, Section deals with the classification and localization, Section describes and discusses the results, and finally, Section concludes the paper.

Supervised Statistical Learning

Since statistical learning is performed in the same way for all object classes, the index κ denoting the number of class will be skipped in this section, i. e., $\Omega_\kappa = \Omega$. Our framework performs the supervised statistical learning in following steps: (i) object acquisition from different viewpoints, (ii) preprocessing into one of the investigated color spaces, (iii) feature extraction, (iv) object area definition, and (v) estimation of the multivariate likelihood density function. These steps are described in the following subsections keeping their order.

Acquisition

In order to capture training data, objects are put on a turntable that rotates to set angles, and training images are taken for each of these angles. The camera is fixed on a mobile arm that can move around the object. The turntable position produces information about the rotation ϕ_y of the object around the vertical y axis. The position of the camera relative to the object yields the object's rotation ϕ_x around the horizontal x axis. The object's scale (translation t_z along the z) can be set with the zoom parameter of the camera, or by moving the camera closer or further from the object. By modifying the camera parameters and position, images can be captured from all top and sidewise views of the object with known external pose parameters $(\phi_{\mathrm{ext}}, t_{\mathrm{ext}})$ for each training image. The translation of the object in the image plane (internal translation) $t_{\mathrm{int}} = (t_x, t_y)^{\mathrm{T}}$ as well the internal rotation $\phi_{\mathrm{int}} = \phi_z$ can be determined after the acquisition process from the relative position of the object in the image. The object pose parameters are usually given relative to each other, as can be seen in Figure 1. For each object class, one image is chosen as the reference image. The pose of an object in an image is understood as being the 3D transformation (rotation and translation) that maps that object into the reference image.

Preprocessing

The original images taken as described in the previous subsection are now preprocessed. First, the scenes are resized to $2^n \times 2^n$ ($n \in \mathbb{N}$) pixels. Then, they are converted into three different representations, namely gray level, RGB, and Lab images.

Unlike the gray level and RGB representations, Lab color is designed to approximate human vision. It aspires to perceptual uniformity, and its L component closely matches human perception of lightness. It can thus be used to make accurate color balance corrections by modifying output curves in the a and b components, or to adjust the lightness contrast using the L component. In RGB space, which models the output of physical devices rather than human visual perception, these transformations can only be done with the help of appropriate blend modes in the editing application [26].

Features

The system determines a set of local feature vectors c_m for all preprocessed training images of an object via the discrete wavelet transform [27]. In order to calculate the c_m vectors, a grid with size $\Delta r = 2^{\lfloor \widehat{s} \rfloor}$, where $\widehat{s}$ is the minimum multiresolution sca-

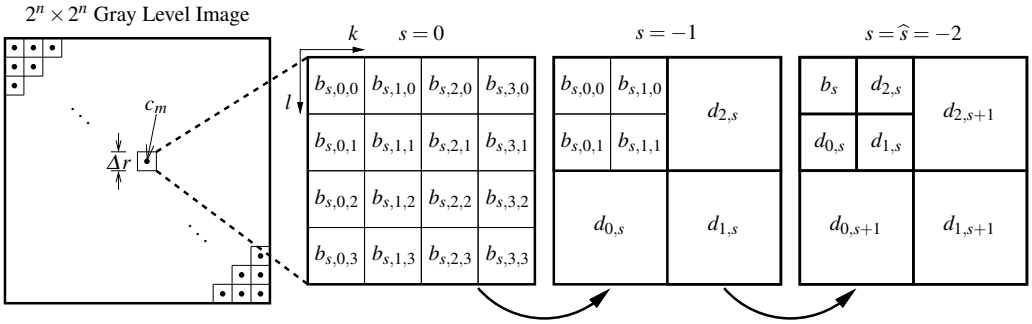

Bild 2. *2D signal decomposition with the wavelet transform for a local neighborhood of size* 4×4 *pixels. The final coefficients result from gray values* $b_{0,k,l}$ *and have the following meaning:* b_{-2} *: low-pass horizontal and low-pass vertical,* $d_{0,-2}$ *: low-pass horizontal and high-pass vertical,* $d_{1,-2}$ *: high-pass horizontal and high-pass vertical,* $d_{2,-2}$ *: high-pass horizontal and low-pass vertical.*

le parameter [1] s, is overlaid on the image [1]. Figure 2 depicts this procedure for the case of gray level scenes divided into local neighborhoods of size 4×4 pixels. Using the coefficients introduced in Figure 2, the local feature vector c_m for the gray level image is defined by,

$$c_m = \begin{pmatrix} c_{m,1} \\ c_{m,2} \end{pmatrix} = \begin{pmatrix} \ln(2^{\widehat{s}}|b_{\widehat{s}}|) \\ \ln[2^{\widehat{s}}(|d_{0,\widehat{s}}| + |d_{1,\widehat{s}}| + |d_{2,\widehat{s}}|)] \end{pmatrix} \quad . \quad (1)$$

In the feature vector, the first component $c_{m,1}$ stores information about the mean gray level (low-frequencies) in the local neighborhood, while the second component $c_{m,2}$ represents discontinuities (high-frequencies). The natural logarithm (ln) decreases the sensibility of the system to illumination changes and muffles any noises, which occur very often, especially in the real world environment. Its use is experimentally motivated in [28]. In the case of RGB and Lab images, each channel is treated independently. The feature computation for each channel is performed in the same way as for gray level images (see Figure 2). Therefore, the local feature vector for color images has six components,

$$c_m = (c_{m,1}, c_{m,2}, c_{m,3}, c_{m,4}, c_{m,5}, c_{m,6})^{\mathrm{T}} \quad . \quad (2)$$

The first $c_{m,1}$ and the second $c_{m,2}$ components are calculated from the first channel, the third $c_{m,3}$ and the fourth $c_{m,4}$ from the second channel, and the fifth $c_{m,5}$ and the sixth $c_{m,6}$ from the third channel [29]. Generally, the system is able to compute local feature vectors for any resolution scale $\widehat{s}$, but in practice $\widehat{s} \in \{-1, -2, -3\}$ is preferred.

Object Area

Clearly, some feature vectors in each training image describe the object, while others belong to the background. In real life applications it cannot be assumed that the background is a-priori known in the recognition phase. Therefore, only feature vectors describing the object are considered for statistical object modeling. Since the object usually composes a part of the image, a tightly enclosing bounding region O called *object area* is defined for each object class. For clarity, we will use the term object area to actually refer to the set of features belonging to the object. The object area can change its location, orientation, and size from image to image depending on the object pose parameters. For this reason, it is modeled as a function of the external pose parameters

$$O = O(\phi_{\mathrm{ext}}, t_{\mathrm{ext}}) \quad , \quad (3)$$

[1] i.e. Further decomposition of the signal with the wavelet transform is not possible.

ideally within a continuous domain. This is done by using the so called *assignment functions* ξ defined for all feature vectors c_m and all training viewpoints $(\phi_{\mathrm{ext}}, t_{\mathrm{ext}})$ as

$$\xi = \xi_m(\phi_{\mathrm{ext}}, t_{\mathrm{ext}}) \quad . \quad (4)$$

The assignment function ξ_m decides, whether the feature vector c_m belongs to the object in the pose $(\phi_{\mathrm{ext}}, t_{\mathrm{ext}})$ or to the background, as follows,

$$\left\{ \begin{array}{lcl} \xi_m(\phi_{\mathrm{ext}}, t_{\mathrm{ext}}) \geq S_O & \Rightarrow & c_m \in O(\phi_{\mathrm{ext}}, t_{\mathrm{ext}}) \\ \xi_m(\phi_{\mathrm{ext}}, t_{\mathrm{ext}}) < S_O & \Rightarrow & c_m \notin O(\phi_{\mathrm{ext}}, t_{\mathrm{ext}}) \end{array} \right\} \quad , \quad (5)$$

where the threshold value S_O is set experimentally and has the same value for all object classes. The assignment functions are trained for each training view separately

$$\xi_m(\phi_{\mathrm{ext}}, t_{\mathrm{ext}}) = \left\{ \begin{array}{lll} 1, & \mathrm{if} & c_{m,1} \geq S_\xi \\ 0, & \mathrm{if} & c_{m,1} < S_\xi \end{array} \right\} \quad , \quad (6)$$

where S_ξ is a threshold value. Since $c_{m,1}$ results from a low-level filtering of a small neighborhood it represents, the practical execution of the training is quite simple. The objects are taken on a nearly black background and a threshold value S_ξ decides whether they belong to the object or to the background. Since there is a finite number of training views $(\phi_{\mathrm{ext}}, t_{\mathrm{ext}})$, these are discrete functions initially, but after interpolation with the sine-cosine transformation they become continuous. Therefore, considering both the internal and external transformation parameters, the object area can be expressed by the function

$$O = O(\phi, t) \quad (7)$$

defined in a continuous six-dimensional pose parameter space (ϕ, t).

Likelihood Density Function

In order to handle illumination changes and low-frequency noise, the elements $c_{m,q}$ of the local feature vectors c_m are interpreted as random variables. Assuming the object's feature vectors $c_m \in O$ as statistically independent of the feature vectors outside the object area, the background feature vectors $c_m \notin O$ can be disregarded here. The elements of the object feature vectors are represented with Gaussian density functions $p(c_{m,q}|\mu_{m,q}, \sigma_{m,q}, \phi, t)$. The mean $\mu_{m,q}$ and standard deviation $\sigma_{m,q}$ values are estimated for all training views $(\phi_{\mathrm{ext}}, t_{\mathrm{ext}})$, which form a subspace of (ϕ, t). Assuming the statistical independence

of the elements $c_{m,q}$, which is valid due to their different interpretations in terms of signal processing, the density function for the object feature vector $c_m \in O$ can be written as,

$$p(c_m|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m, \phi, t) = \prod_{q=1}^{N_q} p(c_{m,q}|\mu_{m,q}, \sigma_{m,q}, \phi, t) \quad , \tag{8}$$

where $\boldsymbol{\mu}_m$ is the mean value vector, $\boldsymbol{\sigma}_m$ the standard deviation vector, and N_q the dimension of the feature vector c_m ($N_q = 2$ for gray level images, $N_q = 6$ for color images). Further, it is assumed that the feature vectors belonging to the object $c_m \in O$ are statistically independent. Under this assumption, an object can be described by the probability density p as follows,

$$p(O|\boldsymbol{B}, \phi, t) = \prod_{c_m \in O} p(c_m|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m, \phi, t) \quad , \tag{9}$$

where $\boldsymbol{B}$ comprises the mean value vectors $\boldsymbol{\mu}_m$ and the standard deviation vectors $\boldsymbol{\sigma}_m$. This probability density is termed the *object density* and, taking into account (8), can be written in more detail as,

$$p(O|\boldsymbol{B}, \phi, t) = \prod_{c_m \in O} \prod_{q=1}^{N_q} p(c_{m,q}|\mu_{m,q}, \sigma_{m,q}, \phi, t) \quad . \tag{10}$$

In reality, neighboring feature vectors might be statistically dependent, but considering the full neighborhood relationship, e. g., as a Markov Random Field [30], leads to a very complex model. In order to complete the object description with the object density (10), the means $\mu_{m,q}$ and the standard deviations $\sigma_{m,q}$ for all object feature vectors c_m have to be learned. For this purpose, N_ρ training images of each object f_ρ are used in association with their corresponding transformation parameters (ϕ_ρ, t_ρ). The mean vectors $\boldsymbol{\mu}_m$ concatenated, written as $\boldsymbol{\mu}$, and the standard deviation vectors $\boldsymbol{\sigma}_m$ concatenated, written as $\boldsymbol{\sigma}$, can be estimated from the maximization of the object density (10) over all N_ρ training images,

$$(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\sigma}}) = \underset{(\boldsymbol{\mu}, \boldsymbol{\sigma})}{\operatorname{argmax}} \prod_{\rho=1}^{N_\rho} p(O|\boldsymbol{B}, \phi_\rho, t_\rho) \quad . \tag{11}$$

As a result of a subsequent interpolation step, the mean vectors $\boldsymbol{\mu}_m$ and standard deviation vectors $\boldsymbol{\sigma}_m$ are trained for all pose parameters (ϕ, t) in a continuous sense.

Classification and Localization

Since for all object classes Ω_κ regarded in a particular recognition task corresponding object models $\mathscr{M}_\kappa$ have already been learned in the training phase, the system is able to classify and localize objects in images taken from a real world environment. First, a test image is taken, preprocessed, and feature vectors in it are computed. Second, the system starts the recognition algorithm integrated into it. In order to perform the classification and localization in the image f, the density values

$$p_{\kappa,h} = p(O_\kappa|\boldsymbol{B}_\kappa, \phi_h, t_h) \tag{12}$$

for all objects Ω_κ and for a large number of pose hypotheses (ϕ_h, t_h) are compared . The computation of the object density value $p_{\kappa,h}$ for the given object Ω_κ, and pose parameters (ϕ_h, t_h) starts with the estimation of the object area $O_\kappa(\phi_h, t_h)$ which has been learned in the training phase. For feature vectors from this object area $c_m \in O_\kappa(\phi_h, t_h)$ the mean value vectors $\boldsymbol{\mu}_{\kappa,m}$ and standard deviation vectors $\boldsymbol{\sigma}_{\kappa,m}$ have been trained and are stored in the object models. Therefore, their density values

$$p_{c_m} = p(c_m|\boldsymbol{\mu}_{\kappa,m}, \boldsymbol{\sigma}_{\kappa,m}, \phi_h, t_h) \tag{13}$$

can be easily determined. Now, the object density value is calculated as follows

$$p_{\kappa,h} = \prod_{c_m \in O_\kappa} \max\{p_{c_m}, T_p\} \quad , \tag{14}$$

where T_p is a threshold value ensuring that density values p_{c_m} close to zero are not taken into account by the product. These density values $p_{c_m} \approx 0$ may result from artifacts or occlusions. The object densities (14) normalized by a quality measure Q are maximized over all object classes Ω_κ and a large number of pose hypotheses (ϕ_h, t_h), as explained in Figure 3. The quality measure (also called geometric criterion), defined in the following way

$$Q(p_{\kappa,h}) = \sqrt[N_{\kappa,h}]{p_{\kappa,h}} \tag{15}$$

decreases the influence the object size has on the recognition results. $N_{\kappa,h}$ denotes the number of feature vectors that belong to the object area $O_\kappa(\phi_h, t_h)$. The classification and localization process presented in Figure 3 can be described by the following maximization term

$$(\widehat{\kappa}, \widehat{\phi}, \widehat{t}) = \underset{(\kappa, \phi_h, t_h)}{\operatorname{argmax}} Q(p(O_\kappa|\boldsymbol{B}_\kappa, \phi_h, t_h)) \tag{16}$$

where $(\widehat{\kappa}, \widehat{\phi}, \widehat{t})$ represent the final recognition result, i. e., the class index and the pose parameters of the object found in image f.

Experiments and Results

3D-REAL-ENV Image Database

In our experiments we used the 3D-REAL-ENV [29, 1] database consisting of the ten real world objects depicted in Figure 4. The pose of each object in the 3D-REAL-ENV database is defined by internal translations $t_{\text{int}} = (t_x, t_y)^{\text{T}}$ and external rotation parameters $\phi_{\text{ext}} = (\phi_x, \phi_y)^{\text{T}}$. The objects were captured in RGB at a resolution of 640×480 pixels under three different illumination settings $I_{\text{lum}} \in \{\text{bright}, \text{average}, \text{dark}\}$. For this experiment the captured images were resized to 256×256 pixels and converted into three different representations, namely gray level, RGB, and Lab images.

Training images were captured with the objects against a dark background from 1680 different viewpoints under two different illumination settings $I_{\text{lum}} \in \{\text{bright}, \text{dark}\}$. This produced 3360 training images in total for each 3D-REAL-ENV object. Each object was placed on a turntable performing a full rotation ($0^\circ \leq \phi_{\text{table}} < 360^\circ$) while the camera attached on a robotic arm was moved on a vertical to horizontal arc ($0^\circ \leq \phi_{\text{arm}} \leq 90^\circ$). The movement of the camera arm ϕ_{arm} corresponds to the first external rotation ϕ_x, while the turntable spin ϕ_{table} corresponds to the second external rotation parameter ϕ_y. The angle between two successive steps of the turntable corresponds to 4.5°. The rotation of the turntable induces an apparent translation in the object position in the image plane, which results in varying internal translation parameters $t_{\text{int}} = (t_x, t_y)^{\text{T}}$. These translations parameters were determined manually after acquisition.

For testing, the ten objects presented were captured from 288 different viewpoints under the average illumination setting ($I_{\text{lum}} = \text{average}$) and against three different backgrounds: homogeneous, weak heterogeneous, and strong heterogeneous. This resulted in three test sets of 2880 images each denoted according to the background used as $T_{\text{ype}} \in \{\text{hom}, \text{weak}, \text{strong}\}$.

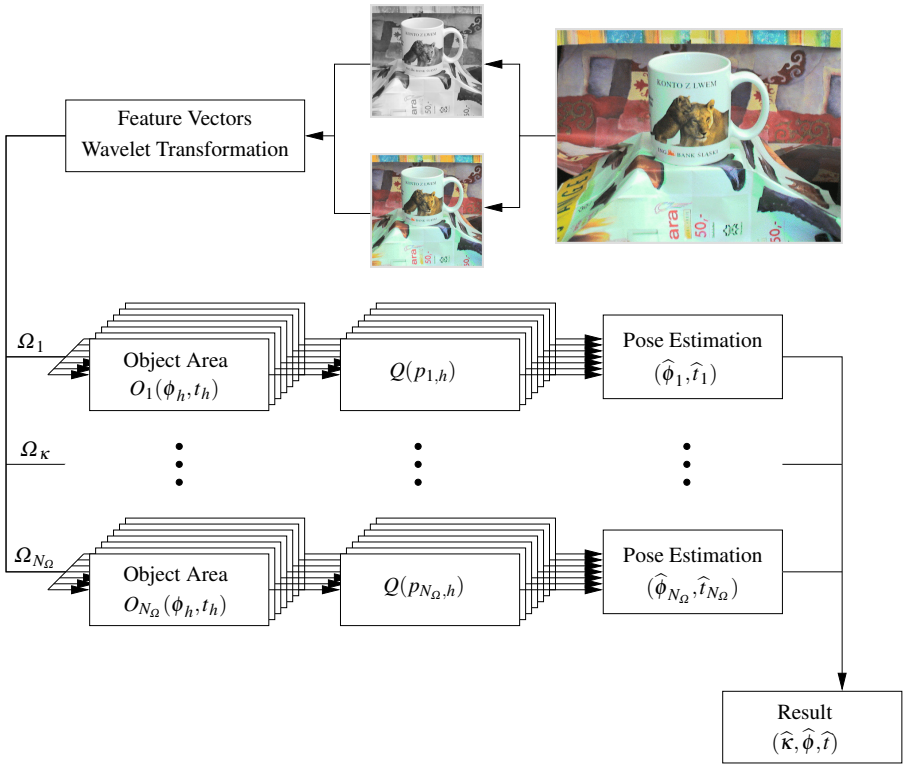

Bild 3. *Density maximization for object classification and localization. First, local feature vectors from the preprocessed test image are computed. Then, for each object class Ω_κ and each pose hypothesis (ϕ_h,t_h) the object area $O_\kappa(\phi_h,t_h)$ is determined and the object density $p_{\kappa,h}$ is calculated. The final recognition result $(\widehat{\kappa},\widehat{\phi},\widehat{t})$ corresponds to the highest density normalized by a quality measure $Q(p_{\kappa,h})$.*

Test scenes of the first type ($T_{\text{ype}} = \text{hom}$) were taken on homogeneous black background, while 200 different real backgrounds were used to create heterogeneous backgrounds ($T_{\text{ype}} \in \{\text{weak}, \text{strong}\}$). Examples of test images with all three types of background are shown in Figure 5. Similarly to the acquisition of training images, the objects were put on a turntable ($0^\circ \leq \phi_{\text{table}} < 360^\circ$) and the camera moved on a robotic arm from vertical to horizontal ($0^\circ \leq \phi_{\text{arm}} \leq 90^\circ$). However, for test images the turntable's rotation between two successive steps is 11.25°, thus test views are generally different from the views used for training. Also, the illumination in the test scenes is different from the illumination in the training images.

Experimental Results

The recognition algorithm was evaluated for the 3D-REAL-ENV image database presented in the previous section. The training of statistical object models was performed for 6 angle-steps (4.5°, 9°, 13.5°, 18°, 22.5°, 27°). Since this was done three times, i. e., for gray level, RGB, and Lab images it resulted in 18 training configurations. The classification and localization rates obtained for these configurations are summarized in Table 1. A classification result is counted as correct when the algorithm returns the correct object class. A localization result is counted as correct when the error for internal translations is not greater than 10 pixels and the error for external rotations not greater than 15°. The results show that color modeling brings a significant improvement in the classification and localization rates for test images with strong heterogeneous background. Here, the best results have been obtained for the Lab images. For scenes with weak heterogeneous background, gray level and RGB representations lead to reasonable results, while Lab color space fails in this case. For scenes with homogeneous background the recognition algorithm performs perfectly in all three cases. For this type of background the computational expense associated with color information can be avoided. In Table 1 one can also see how the recognition rates depend on the distance of training views. For test images with homogenous background, the classification algorithm works properly even for a high distance of 27°. However, it becomes worse in more complex scenarios. The localization rates show their strong dependency on the distance of training views for all three test datasets. The reason for this is the necessary interpolation between the training views. Obviously, this interpolation works more precisely for denser data sets in terms of training viewpoints. Object recognition takes 3.6s in one gray level image and 7s in one color image on a workstation equipped with a Pentium 4, at 2.66 GHz, and 512 MB of RAM.

Conclusions

This article presents a system for 3D texture-based probabilistic object classification and localization in 2D images and discusses its performance for three different image formats, namely the gray level, the RGB, and the Lab representations. Experimental results on an image database of over 40,000 images illustrate the high performance of our system. While for test images with homogeneous background the recognition rates are almost perfect for all three image representations, they vary in more complex environments. Color modeling, especially the Lab color space, brings the most benefits in very complex scenes (strong heterogeneous background). However, for images with weak heterogeneous background, the gray level and RGB formats present higher performance.

Improvements are possible with the approach and will form the basis of our future work. One line of research is to consider combining the appearance-based model with a shape-based mo-

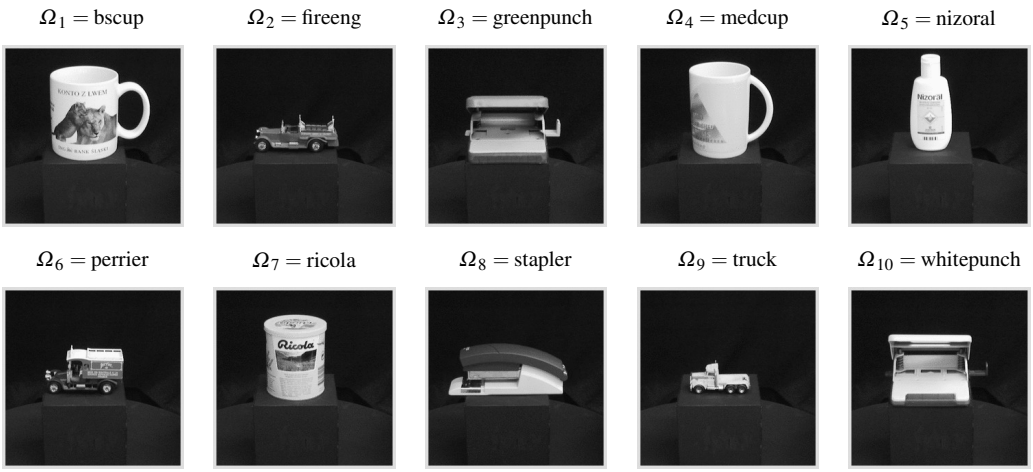

Bild 4. *Ten objects of the 3D-REAL-ENV image database with their short names. Top row from left to right: bank cup, toy fire engine, green puncher, siemens cup, nizoral bottle. Bottom row from left to right: toy passenger car, candy box, blue stapler, toy truck, white puncher.*

del for object recognition. There are objects with the same shape, which are distinguishable only by texture, but one can also imagine objects with the same texture features, which can be easily distinguished by shape. Finally, since our system is adaptable to many image classification tasks we also intend to apply it in the context of knowledge assisted image and video content retrieval.

Author Biography

Marcin Grzegorzek graduated in Computer Science at the Silesian University of Technology in Gliwice (Poland) in 2002. In 2007 he received his PhD (summa cum laude) in the field of statistical object recognition from the University of Erlangen-Nuremberg (Germany). Further, he worked as a Postdoc for the Multimedia and Vision Research Group at the Queen Mary University of London (UK). Currently, he is a Lecturer in the Computer Science Department at the University of Koblenz-Landau (Germany).

Alexandra Wolyniec is studying Computer Science at the University of Koblenz (Germany). She will graduate in 2010. Beside her studies she was working at the University of Koblenz as student assistant in the fields of e-Learning (2005-2007), Computer Graphics (2008), and now Image Processing. In 2009 she was working at the Fraunhofer Institute for Applied Information Technology in the fields of Augmented and Virtual Reality.

Frank Schmitt graduated in Computer Science at the University of Koblenz-Landau (Germany) in 2005. Since 2005 he is a PhD Student in the Image Recognition Working Group, Institute for Computational Visualistics, University of Koblenz-Landau. His main research interests are associated with the automatic detection and description of distinctive visual features in images of urban environments.

Dietrich Paulus obtained a Bachelor degree in Computer Science from University of Western Ontario, London, Canada, followed by a diploma (Dipl.-Inf.) in Computer Science and a PhD (Dr.-Ing.) from Friedrich-Alexander University Erlangen-Nuremberg, Germany. He obtained his habilitation in Erlangen in 2001. Since 2001 he is at the institute for computational visualistics at the University Koblenz-Landau, Germany where he became a full professor in 2002. His primary interests are computer vision and robot vision.

Literatur

[1] Grzegorzek, M., Reinhold, M., Niemann, H.: Feature extraction with wavelet transformation for statistical object recognition. In Kurzynski, M., Puchala, E., Wozniak, M., Zolnierek, A., eds.: 4th International Conference on Computer Recognition Systems, Rydzyna, Poland, Springer-Verlag, Berlin, Heidelberg (May 2005) 161–168

[2] Latecki, L.J., Lakaemper, R., Wolter, D.: Optimal partial shape similarity. Image and Vision Computing Journal **23** (2005) 227–236

[3] Schiele, B., Crowley, J.L.: Recognition without correspondence using multidimensional receptive field histograms. International Journal of Computer Vision **36**(1) (January 2000) 31–50

[4] Ulusoy, I., Bishop, C.M.: Generative versus discriminative methods for object recognition. In: International Conference on Computer Visions and Pattern Recognition (Volume 2), San Diego, USA, IEEE Computer Society (June 2005) 258–264

[5] Jolliffe, I.T.: Principal Component Analysis. Springer (2002)

[6] Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons (2001)

[7] Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791

[8] Roth, P.M., Winter, M.: Survey of appearance-based methods for object recognition. Technical Report ICG-TR-01/08, Inst. for Computer Graphics and Vision, Graz University of Technology, Austria (2008)

[9] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons (2000)

[10] Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer (1995)

[11] Freund, Y., Shapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer System Sciences **55** (1997) 119–139

[12] Amit, Y., Geman, D., Fan, X.: A coarse-to-fine strategy for multi-class shape detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(12) (December 2004) 1606–1621

[13] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (November 2004) 91–110

[14] Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection.

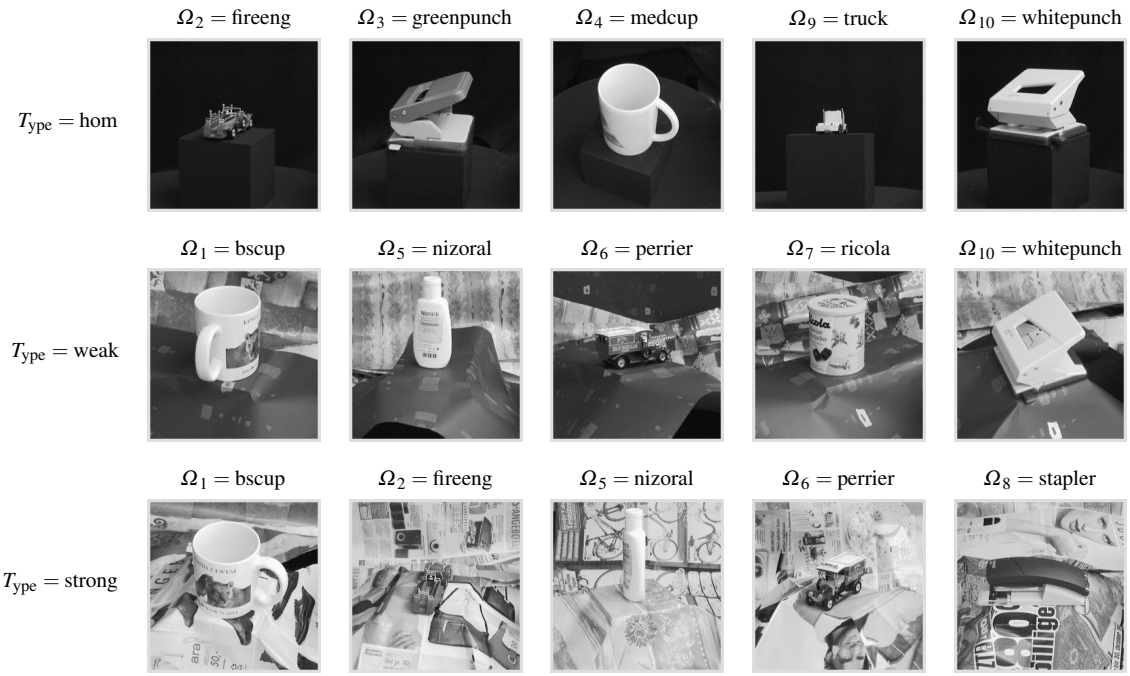

Bild 5. *Examples of test scenes on all three types of background $T_{\text{ype}} \in \{\text{hom}, \text{weak}, \text{strong}\}$. The top row shows images with homogeneous background ($T_{\text{ype}} = \text{hom}$), the middle row images with weak heterogeneous background ($T_{\text{ype}} = \text{weak}$), and the bottom row images with strong heterogeneous background ($T_{\text{ype}} = \text{strong}$).*

IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(5) (May 2007) 854–869

[15] Jin, Y., Geman, S.: Context and hierarchy in a probabilistic image model. In: IEEE Conference on Computer Vision and Pattern Recognition, New York, USA (June 2006) 2145–2152

[16] Leibe, B., Schiele, B.: Analyzing contour and appearance based methods for object categorization. In: IEEE Conference on Computer Vision and Pattern Recognition, Madison, USA (June 2003)

[17] Lowe, D.G.: Object recognition from local scale-invariant fearures. In: 7. International Conference on Computer Vision (ICCV), Corfu, Greece (September 1999) 1150–1157

[18] Mahamud, S., Hebert, M.: The optimal distance measure for object detection. In: IEEE Conf. on Computer Vision and Pattern Recognition, Madison, USA (June 2003)

[19] Gross, R., Matthews, I., Baker, S.: Appearance-based face recognition and light-fields. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(4) (April 2004) 449–465

[20] Park, C.H., Park, H.: Fingerprint classification using fast fourier transform and nonlinear discriminant analysis. Pattern Recognition **38**(4) (April 2005) 495–503

[21] Heutte, L., Nosary, A., Paquet, T.: A multiple agent architecture for handwritten text recognition. Pattern Recognition **37**(4) (April 2004) 665–674

[22] Zobel, M., Denzler, J., Heigl, B., Nöth, E., Paulus, D., Schmidt, J., Stemmer, G.: Mobsy: Integration of vision and dialogue in service robots. Machine Vision and Applications **14**(1) (April 2003) 26–34

[23] Li, C.H., Yuen, P.C.: Tongue image matching using color content. Pattern Recognition **35**(2) (February 2002) 407–419

[24] Ngan, H.Y., Pang, G.K., Yung, S., Ng, M.K.: Wavelet based methods on patterned fabric defect detection. Pattern Recognition **38**(4) (April 2005) 559–576

[25] Gausemeier, J., Grafe, M., Matysczok, C., Radkowski, R., Krebs, J., Oelschlaeger, H.: Eine mobile augmented reality versuchsplattform zur untersuchung und evaluation von fahrzeugergonomien. In Schulze, T., Horton, G., Preim, B., Schlechtweg, S., eds.: Simulation und Visualisierung, Magdeburg, Germany, SCS Publishing House e.V. (March 2005) 185–194

[26] Levkowitz, H.: Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications. Kluwer Academic Publishers, Norwell, MA, USA (1997)

[27] Mallat, S.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(7) (July 1989) 674–693

[28] Reinhold, M.: Robuste, probabilistische, erscheinungsbasierte Objekterkennung. Logos Verlag, Berlin, Germany (2004)

[29] Grzegorzek, M., Niemann, H.: Statistical object recognition including color modeling. In Kamel, M., Campilho, A., eds.: 2nd International Conference on Image Analysis and Recognition, Toronto, Canada, Springer-Verlag, Berlin, Heidelberg, LNCS 3656 (September 2005) 481–489

[30] Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. Chapman & Hall, London, UK (2005)

Distance of Training Views [°]	Type of Object Modeling	Classification Rate [%]			Localization Rate [%]		
		Hom. Back.	Weak Het.	Strong Het.	Hom. Back.	Weak Het.	Strong Het.
4.5	**Gray**	100	92.2	54.1	99.1	80.9	69.0
	RGB	100	88.0	82.3	98.5	77.8	73.6
	LAB	100	67.0	90.2	99.1	71.4	77.8
9.0	**Gray**	100	92.4	55.4	98.7	80.0	67.2
	RGB	100	88.3	81.2	98.2	76.4	72.1
	LAB	100	66.7	90.7	99.1	70.6	78.6
13.5	**Gray**	99.4	89.7	56.2	96.9	78.6	65.4
	RGB	99.6	82.7	80.3	94.9	68.4	66.6
	LAB	100	65.3	87.8	98.7	64.2	74.8
18.0	**Gray**	99.9	89.2	55.1	96.6	71.4	54.5
	RGB	97.3	80.6	68.6	94.3	64.9	60.7
	LAB	100	63.3	83.9	97.0	62.2	69.0
22.5	**Gray**	99.4	86.0	52.8	94.5	60.7	38.6
	RGB	94.7	74.8	59.2	89.4	52.2	46.2
	LAB	99.9	57.5	77.7	94.5	49.6	55.5
27.0	**Gray**	96.5	69.4	54.4	83.8	49.9	32.8
	RGB	93.8	53.6	50.2	78.3	35.8	35.6
	LAB	99.1	43.6	62.2	92.0	37.2	41.2

Classification and localization rates obtained for 3D-REAL-ENV image database with gray level, RGB, and Lab images. The distance of training views varies from 4.5° **to** 27° **in** 5 **steps. For experiments,** 2880 **test images with homogeneous,** 2880 **test images with weak heterogeneous, and** 2880 **images with strong heterogeneous background were used.**