How Fusion of Multiple Views Can Improve Object Recognition in Real-World Environments

Marcin Grzegorzek*, Frank Deinzer†, Michael Reinhold*, Joachim Denzler, Heinrich Niemann

Friedrich-Alexander-Universität Erlangen-Nürnberg, Informatik 5
Lehrstuhl für Mustererkennung
Martensstr. 3, 91058 Erlangen, Germany
Email: grzegorz@informatik.uni-erlangen.de

Abstract

In the past decades most object recognition systems were based on passive approaches. But in the last few years a lot of research was done in the field of active object recognition. In this context there are several unique problems to be solved. One of them is how to fuse a series of images that might differ in their viewpoints.

In this paper we present a well-founded approach for the fusion of multiple views based on a recursive density propagation method. It uses particle filters for solving the fusion in a continuous pose space. Furthermore we will show by means of a statistical object recognition system how to integrate such systems into our fusion approach.

The experimental result will show, how the fusion can improve classification rates substantial, especially for difficult conditions like heterogeneous background within real world environments.

1 Introduction

Passive approaches for object recognition have been in the center of research in computer vision within the last decades. One of the main properties is that a decision for a certain class and pose or a rejection must be made based on a single image. Although such passive approaches are sufficient for the solution of many computer vision problems as shown in a lot of applications in the past, they neglect the fact that in many fields there is usually more than one image available or it can be easily acquired. These additional images could be perfectly used to gain more information about the scene and the observed objects. This is one reason, why research has focused on active object recognition recently [2, 4, 5, 9, 13].

One of the most important aspects in active object recognition is the fusion of a sequence of images taken from different viewpoints to obtain an overall classification and localization result that improves over time. Actually, this is what one expects from our human visual system – collect and merge information. These circumstances where several images are available to a computer vision system can be observed in a lot of today's applications. Think, for example, of robots that move around in real world environments to perform service tasks. Such tasks require a continuous fusion of the images taken by the robot – preferable in real time. Other situations where a fusion of multiple views might be helpful is when one has to deal with ambiguous objects (for which more than one view might be necessary to resolve the ambiguity) or heterogeneous background.

In this paper we present a general fusion scheme based on [6]. There are three main reasons for applying the Condensation algorithm. First, during fusion one has to deal inherently with multimodal distributions over the class and pose space of the objects. Second, moving the camera from one viewpoint to another will add uncertainty in the fusion process as the movement of the camera will always be disturbed by noise. This is especially true for applications where the information about the camera movement is obtained from a robot's odometric information. Thus, this uncertainty must be taken into account when fusing the current image with the results acquired so far. Third, it is not straight forward

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

[†] This work was partially funded by the German Science Foundation (DFG) under grant SFB 603/TP B2.
Only the authors are responsible for the content.

to model the involved probability distributions in closed form, especially if multiple hypothesis, i.e. multimodal distributions, shall be handled. These three aspects are strong criterions that the Condensation algorithm is perfectly suited. Especially, the ability to handle dynamic systems is advantageous because in viewpoint fusion the dynamics is given by the known but noisy camera motion between two viewpoints.

In Section 2, we will present the theoretical background of our fusion approach based on the Condensation algorithm. We will also describe the requirements to existing object recognition systems needed to allow an integration into our fusion approach. A statistical object recognition system based on wavelet features, that was successfully used together with the fusion approach, is presented in Section 3. The performed experiments in Section 4 will show the practicability of our method in the context of classification of objects with heterogeneous background in real world environments. The results will show a significant improvement in recognition results. Section 5 will close this paper with a conclusion and a short outlook to further investigations.

2 Fusion

Active object recognition extends the classic passive approach in a manner that object classification and localization is based on a sequence of images no matter whether they were taken randomly or in an intelligent way. These images are used to improve the robustness and reliability of the object classification and localization. In this active approach object recognition is not simply a task of repeated classification and localization for each image, but a well directed combination of the information acquired so far with the current image. As one will see in the following section, the fusion can be formulated as the recursive propagation of densities over time.

2.1 Density Propagation with the Condensation Algorithm

In active object recognition a series of observed images $\boldsymbol{f}_n, \boldsymbol{f}_{n-1}, \ldots, \boldsymbol{f}_0$ of an object are given together with the camera movements $\boldsymbol{a}_{n-1}, \ldots, \boldsymbol{a}_0$ between these images. Based on these observations of images and movements one wants to draw conclusions for a non-observable state $\boldsymbol{q}_n$ of the object. This state $\boldsymbol{q}_n$ must contain both the *discrete* class and the *continuous* pose of the object. This fact is important for the further proceeding.

In the context of a Bayesian approach, the knowledge on the object's state is given in form of the a posteriori density $p(\boldsymbol{q}_n|\boldsymbol{f}_n, \boldsymbol{a}_{n-1}, \boldsymbol{f}_{n-1}, \ldots, \boldsymbol{a}_0, \boldsymbol{f}_0)$ and can be calculated from

$$p(\boldsymbol{q}_n|\boldsymbol{f}_n, \boldsymbol{a}_{n-1}, \ldots, \boldsymbol{a}_0, \boldsymbol{f}_0) = \frac{1}{k_n} \underbrace{p(\boldsymbol{q}_n|\boldsymbol{a}_{n-1}, \boldsymbol{f}_{n-1}, \ldots)}_{(\star)} p(\boldsymbol{f}_n|\boldsymbol{q}_n) \quad (1)$$

where $k_n = p(\boldsymbol{f}_n, \boldsymbol{a}_{n-1}, \ldots, \boldsymbol{a}_0, \boldsymbol{f}_0)$ denotes a normalizing constant that is left out in the following considerations. Under the Markov assumption

$$p(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}, \boldsymbol{a}_{n-1}, \ldots) = p(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}, \boldsymbol{a}_{n-1})$$

for the state transition, the term $(\star)$ within (1) can be recursively rewritten as

$$p(\boldsymbol{q}_n|\boldsymbol{a}_{n-1}, \boldsymbol{f}_{n-1}, \ldots) = \int_{\boldsymbol{q}_{n-1}} p(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}, \boldsymbol{a}_{n-1}) \cdot p(\boldsymbol{q}_{n-1}|\boldsymbol{f}_{n-1}, \boldsymbol{a}_{n-2}, \boldsymbol{f}_{n-2}, \ldots) d\boldsymbol{q}_{n-1} \quad (2)$$

It is obvious that this probability depends only on the camera movement $\boldsymbol{a}_{n-1}$. The inaccuracy of the camera movement is modeled with a normally distributed noise component.

If continuous components in the state $\boldsymbol{q}_n$ can be avoided, the integral in (2) can be simplified to

$$p(\boldsymbol{q}_n|\boldsymbol{a}_{n-1}, \boldsymbol{f}_{n-1}, \ldots) = \sum_{\boldsymbol{q}_{n-1}} p(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}, \boldsymbol{a}_{n-1}) \cdot p(\boldsymbol{q}_{n-1}|\boldsymbol{f}_{n-1}, \boldsymbol{a}_{n-2}, \boldsymbol{f}_{n-2}, \ldots) \quad (3)$$

and can easily be evaluated in an analytical way. For example, to classify an object of class Ω_κ in a sequence of images with i.e. $\boldsymbol{q}_n = (\Omega_\kappa)$, $p(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}, \boldsymbol{a}_{n-1})$ in (3) degrades to

$$p(\boldsymbol{q}_n|\boldsymbol{q}_{n-1}, \boldsymbol{a}_{n-1}) = \begin{cases} 1 & \text{if } \boldsymbol{q}_n = \boldsymbol{q}_{n-1} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

since the object class does not change if the camera is moved, and consequently (3) must have an analytically solution.

But if one wants to use the fusion of multiple views in a general way with the possibility of continuous pose parameters in $\boldsymbol{q}_n$ it is no longer possible to simplify (2) to (3).

The classic approach for solving this recursive density propagation is the Kalman Filter [7, 1]. But in computer vision the necessary assumption for the Kalman Filter ($p(\boldsymbol{f}_n|\boldsymbol{q}_n)$ being normally distributed) are often not valid. In real world applications this density $p(\boldsymbol{f}_n|\boldsymbol{q}_n)$ usually is not normally distributed due to object ambiguities, sensor noise, occlusion, etc. This is a problem since it leads to a distribution which is not analytically computable. An approach for the complicated handling of such multimodal densities are the so called particle filters. The basic idea is to approximate the a posteriori density by a set of weighted particles. In our approach we use the Condensation algorithm [6]. It uses a sample set $Y_n = \{\langle \boldsymbol{y}_1^n, p_1^n\rangle, \ldots, \langle \boldsymbol{y}_K^n, p_K^n\rangle\}$ to approximate the multimodal probability distribution in (1). Please note that we do not only have a continuous state space for $\boldsymbol{q}_n$ but a *mixed discrete/continuous state space* for object class and pose as mentioned at the beginning of this section. The practical procedure of applying the Condensation to the fusion problem is illustrated in the next section.

2.2 Fusion of Multiple Views with the Condensation Algorithm

After the presentation of the density propagation theory we will show how to use the Condensation algorithm in a practical realization of sensor data fusion of multiple views. As noted above we need to include the class and pose of the object into our state $\boldsymbol{q}_n$ to classify and localize objects. This leads to the following definitions of the state

$$\boldsymbol{q}_n = \left(\Omega_\kappa,\ {}^1\phi^n, \ldots,\ {}^J\phi^n\right)^{\mathrm{T}} \tag{5}$$

and the samples

$$\boldsymbol{y}_i^n = \left(\Omega_\kappa,\ {}^1\phi_i^n, \ldots,\ {}^J\phi_i^n\right)^{\mathrm{T}} \tag{6}$$

where ${}^j\phi^n$ denotes the pose of the j-th degree of freedom for the camera position. The camera movements are defined accordingly as

$$\boldsymbol{a}_n = \left(\Delta^1\phi^n, \ldots, \Delta^J\phi^n\right)^{\mathrm{T}} \tag{7}$$

with $\Delta^j\phi^n$ denoting the relative changes of the viewing position of the camera.

In our experimental setup (see image "O" in Figure 3) we have only one degree of freedom. The camera can move on a circle around the object with an angle ${}^1\phi^n \in [0°; 360°)$ describing the pose of the object.

In the practical realization of the Condensation, one starts with an initial sample set $Y_0 = \{\langle \boldsymbol{y}_1^0, p_1^0\rangle, \ldots, \langle \boldsymbol{y}_K^0, p_K^0\rangle\}$ with samples distributed uniformly over the state space and $p_i^0 = 1/K$. If there is some knowledge available about the distribution in advance the samples can of course be distributed non-uniformly. For the generation of a new sample set Y^n, samples $\boldsymbol{y}_i^n$ are

1. drawn from Y^{n-1} with probability
$$p_i^{n-1} / \sum_{j=1}^{K} p_j^{n-1}\,, \tag{8}$$
2. propagated with the state transition model
$$\boldsymbol{y}_i^n = \boldsymbol{y}_i^{n-1} + (0, r_1, \ldots, r_J)^{\mathrm{T}} \tag{9}$$
with $r_j \sim \mathcal{N}(\Delta^j\phi^n, \sigma_j)$ and the variance parameters of the Gaussian transition noise σ_j. They model the inaccuracy of the camera movement under the assumption that the errors of the camera movements are independent between the degrees of freedom. The variances have to be determined in advance.
3. evaluated in the image by
$$p(\boldsymbol{f}_n|\boldsymbol{y}_i^n)\,. \tag{10}$$
This evaluation is performed by the classifier. The only need to the classifier that shall be used together with our fusion approach must be its ability to evaluate this density. In section 3.5 we will show how (10) can be evaluated by (16).

For a more detailed explanation on the theoretical background of the approximation of (1) by the sample set Y^N we refer to [6].

At this point we want to note that it is important to include the class Ω_κ into the object state $\boldsymbol{q}_n$ and the samples $\boldsymbol{y}_i^n$. An alternative would be to omit this by setting up several sample sets – one for each object class – and perform the Condensation algorithm separately on each set. But this would not result in an integrated classification/localization, but in separated localizations on each set under the assumption of observing the corresponding object

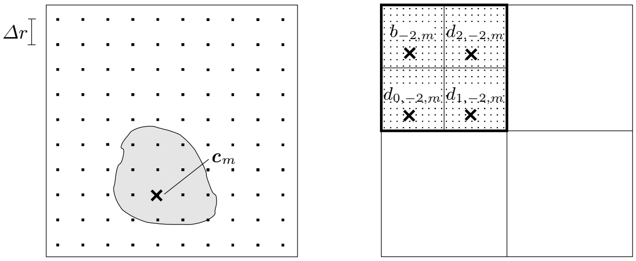

Figure 1: Feature calculating. *Left:* image with the grid size $\Delta r = 2^2$. *Right:* the wavelet multiresolution analysis was performed two times. Each grid point has exact one corresponding low-pass coefficient $b_{-2,m}$ and three high-pass coefficients $d_{0...2,-2,m}$, from which the two dimensional local feature vector will be computed.

class. Consequently, no fusion of the object class over the sequence of images would be done.

3 Statistical Object Recognition

In this section a statistical object recognition system that is successfully used together with the fusion approach will be presented.

3.1 Feature Vectors

To build the statistical model of any object we need first to define the feature vectors. Two dimensional local features are calculated with the wavelet transformation [3, 8]. A grid with the size $\Delta r = 2^{-s}$, whereby s is the scale of wavelet transformation, is laid over the quadratic image $\boldsymbol{f}$ (Figure 1). On each grid point a two dimensional local feature vector $\boldsymbol{c}_m$ is calculated. In this case we perform s-times the wavelet multiresolution analysis:

$$\boldsymbol{c}_m = \begin{pmatrix} c_{m1} \\ c_{m2} \end{pmatrix} =$$

$$= \begin{pmatrix} \ln(2^s\,|b_{s,m}|) \\ \ln[2^s\,(|d_{0,s,m}| + |d_{1,s,m}| + |d_{2,s,m}|)] \end{pmatrix} \quad (11)$$

The value $b_{s,m}$ is the low-pass coefficient and $d_{0...2,s,m}$ are the high-pass coefficients. Using the local feature vectors has an very important advantage: If only one pixel changes in the image, e.g. by noise or occlusion, only the local feature vectors in a small region around vary. Owing to wavelet multiresolution analysis the high-pass and low-pass information of the image could be stored in the feature vectors. We can now define the set of all feature vectors in the image:

$$C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_M\} \quad (12)$$

where M is the number of local feature vectors in the image.

3.2 Bounding Region of the Object

In natural environments the object takes usually only part of the image area. The rest belongs to the background. In order to model the object density function we do not need to regard feature vectors that belong to the background. That is, why we define for each object class in each training position a bounding region. A close boundary is laid around the object (Figure 2). The feature vectors inside this bounding region belong to the object and the feature vectors outside the bounding region belong to the background. When the object is rotated and translated inside the image plane $(\boldsymbol{\phi}_{int}, \boldsymbol{t}_{int})$ the appearance and size of the object do not change. This transformations are called internal transformations. To handle this case, the bounding region and the object grid are moved by the same transformations as the object. The new positions $\boldsymbol{x}'_m$ in the object grid are calculated from the old grid points $\boldsymbol{x}_m$ with following equation:

$$\boldsymbol{x}'_m = \boldsymbol{R}(\boldsymbol{\phi}_{int})\boldsymbol{x}_m + \boldsymbol{t}_{int} \quad (13)$$

whereby $\boldsymbol{R}(\boldsymbol{\phi}_{int})$ is the rotation matrix. For the external transformations $(\boldsymbol{\phi}_{ext}, \boldsymbol{t}_{ext})$ the size of the object in the image varies. We have to model the

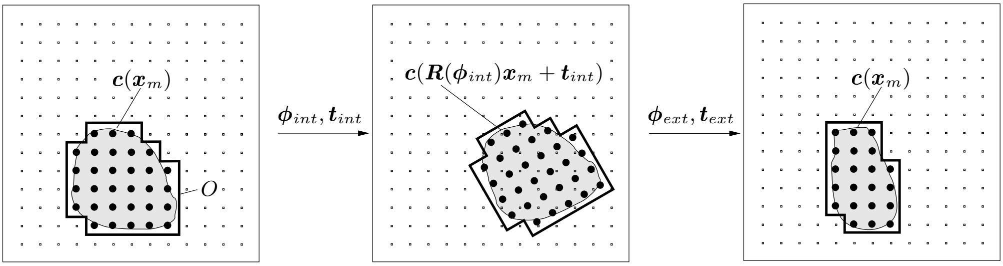

Figure 2: *Left* : all feature vectors within the bounding region belong to the object. *Middle*: under internal transformations moves the bounding region with the same intern transformations as the object. *Right*: under external transformations is the size of the object and the bounding region variable.

size of the bounding region as a function of these external transformations. For this purpose, we define for each local feature vector $\boldsymbol{c}_m$ a function, that assigns the feature vector to the bounding region, or to the background [12]:

$$\xi_m = \xi_m(\boldsymbol{\phi}_{ext}, \boldsymbol{t}_{ext}) = \begin{cases} 1 & \text{if} \quad \boldsymbol{c}_m \in O \\ 0 & \text{if} \quad \boldsymbol{c}_m \notin O \end{cases} \quad (14)$$

O symbolizes the object bounding region. These functions are calculated during training. We train these functions using images of objects taken from different viewpoints. During the recognition phase the size of the bounding region for a pose $(\boldsymbol{\phi}_{ext}, \boldsymbol{t}_{ext})$ is calculated by these trained functions $\xi_m(\boldsymbol{\phi}_{ext}, \boldsymbol{t}_{ext})$. The internal and external transformations could be written together: $\boldsymbol{\phi} = (\boldsymbol{\phi}_{int}, \boldsymbol{\phi}_{ext})^T$, $\boldsymbol{t} = (\boldsymbol{t}_{int}, \boldsymbol{t}_{ext})^T$.

3.3 Statistical Model for the Object

To handle noises and illumination changes in images we apply a statistical model. Each feature vector $\boldsymbol{c}_m$ is interpreted as random variable. We assume that the object features are statistically independent of the background features, so only object feature vectors have to be considered for the object model. We assume also the statistical independency of the single feature vectors and their components. The components of the feature vectors are modelled as normally distributed consequently. The density function for the object features could be written as:

$$p(C|\boldsymbol{B}_\kappa, \boldsymbol{\phi}, \boldsymbol{t}) =$$

$$= \prod_{\{m|\xi_{m\kappa}=1\}} p(\boldsymbol{c}_m|\boldsymbol{\mu}_{m\kappa}, \boldsymbol{\sigma}_{m\kappa}, \boldsymbol{\phi}, \boldsymbol{t}) \qquad (15)$$

where $\boldsymbol{B}_\kappa$ comprehends the trained mean vectors $\boldsymbol{\mu}_{m\kappa} = (\mu_{m\kappa 1}, \mu_{m\kappa 2})^T$ and standard deviation vectors $\boldsymbol{\sigma}_{m\kappa} = (\sigma_{m\kappa 1}, \sigma_{m\kappa 2})^T$ of the feature vectors $\boldsymbol{c}_{m\kappa} = (c_{m\kappa 1}, c_{m\kappa 2})^T$, $(\boldsymbol{\phi}, \boldsymbol{t})$ are the transformation parameters and index κ denotes the number of object class [10]. For internal transformations the mean values $\mu_{m\kappa 1}$, $\mu_{m\kappa 2}$ and the standard deviations $\sigma_{m\kappa 1}$, $\sigma_{m\kappa 2}$ are constant. Under external transformations the mean values vary and can be written as functions of these transformations $\mu_{m\kappa 1} = \mu_{m\kappa 1}(\boldsymbol{\phi}_{ext}, \boldsymbol{t}_{ext})$, $\mu_{m\kappa 2} = \mu_{m\kappa 2}(\boldsymbol{\phi}_{ext}, \boldsymbol{t}_{ext})$. In contrast to mean values we model the standard deviation values as constant in this case.

3.4 Statistical Model for the Background

In our task the objects are situated in heterogeneous background. As a consequence the feature vectors at the border of the object depend not only on gray values in the object bounding region, but also on the background. Components of such feature vectors could sometimes have strongly different values from the components of feature vectors that were observed during training. It has one very unpleasant consequence. The probabilities of such feature vectors $p(\boldsymbol{c}_m|\boldsymbol{\mu}_{m\kappa}, \boldsymbol{\sigma}_{m\kappa}, \boldsymbol{\phi}, \boldsymbol{t})$ are in the recognition phase near zero. The density function of an Object (15) as product of the single probabilities is also close to zero, i.e. a successful recognition is impossible. To solve this problem we have introduced a separate background model. The background is modelled as uniform distribution over all possible values of the feature vectors [11]. Therefore, a priori, nothing has to be known about the background in the recognition phase. Every possi-

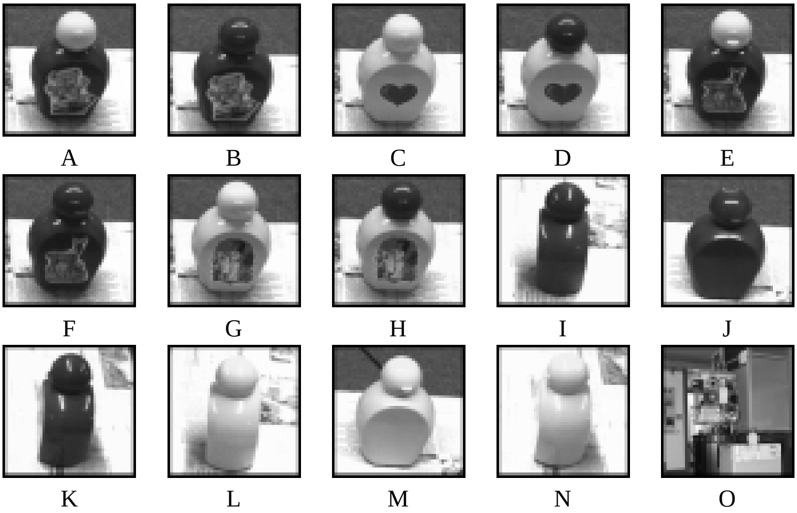

Figure 3: A - yellow bear (external rotation 0°), B - red bear (0°), C - yellow heart (0°), D - red heart (0°), E - yellow deer (0°), F - red deer (0°), G - yellow “star-money” (0°), H - red “star-money” (0°), I - red bear (90°), J - red bear (180°), K - red bear (270°), L - yellow heart (90°), M - yellow heart (180°), N - yellow heart (270°), O - complex scene with moving robot

ble background can be handled by the same background density.

3.5 Evaluation of the Probability

The classifier, that is described in this section, must be able to evaluate the density from the equation 10. The image $\boldsymbol{f}_n$ has to be transform into the set of feature vectors C (12). The computation of the feature vectors is described in the section 3.1. Then we get the trained class model $\boldsymbol{B}_\kappa$ from the first element of the vector $\boldsymbol{y}_i^n$ (6). The external rotations vector $\boldsymbol{\phi}_{ext}$ could be also obtained from the vector $\boldsymbol{y}_i^n$ (6). We consider in our task one external rotation, so it could be written: $\boldsymbol{\phi}_{ext} = ((y_i^n)_2, 0)^T$, whereby $(y_i^n)_2$ is the second component of the vector $\boldsymbol{y}_i^n$ and the second external rotation is always equal zero in our task. The internal rotation and external translation are equal zero. The evaluation of the probability (10) can be written as:

$$p(\boldsymbol{f}_n|\boldsymbol{y}_i^n) = \max_{\boldsymbol{t}_{int}} p(C|\boldsymbol{B}_\kappa, \boldsymbol{\phi}, \boldsymbol{t}) \qquad (16)$$

As one can see in the equation above we had to maximize the density with internal translations. The test images were taken from a moving robot, the moment of which is not precise. That is why the objects were shifted in the image plane.

4 Experiments and Results

We tested our approach on a data set that comprehends 8 objects which are illustrated in figure 3. These objects are represented from different viewpoints.

For the training the objects were put on a turntable, with $0^\circ \leq \phi_{table} \leq 360^\circ$, and from each object 270 gray value images with 256^2 pixels were taken by a camera mounted on a robot, so that we have one external rotation. The viewpoints are uniformly distributed on a circle and the angle between two adjacent viewpoints is 4°. Besides three different lighting conditions were applied. Although we used a dark background for taking the training images the background was not homogeneous. For the training of the bounding region we applied a threshold value for each training image. All pixels with gray values under this threshold value were set to zero (black).

The experiments were performed on 180 images for each class, i.e altogether 1440 images, thereby

Classification Rates				
	0°	90°	180°	270°
1	90%	60%	62,5%	62,5%
5	72,5%	70%	85%	75%
10	77,5%	70%	85%	80%
15	85%	87,5%	85%	80%
18	85%	90%	82,5%	85%

Figure 4: The classification rates in the column 0° were created when the fusion algorithm was started with objects in the position 0°. The next column contains the results of the fusion algorithm that was started with objects in the position 90° etc. The left column denotes the number of fused images.

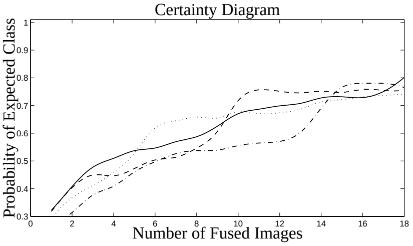

Figure 5: — algorithm starts with the image in 0° position, · · · algorithm starts with the image in 90° position, - - - algorithm starts with the image in 180° position, - · - algorithm starts with the image in 270° position.

the test images were different from the training images. The test images were taken from a robot. The robot moved on a circle around the object and took 90 images of each object. The objects were located on a box wrapped with a newspaper which is depicted in the figure 3 O. The search area for $\boldsymbol{t}_{int}$ within the image plane as described in (16) was restricted to ±12 pixels in x- and y-direction.

We took into account four different situations. First, we started with experiments where the pictures on the objects were visible already on the first image in the fusion (around 0°, figure 3A-H). Then our fusion algorithm was started with images where the objects were rotated about 90° (Figure 3 I and L). In this case a successful classification is impossible at the beginning, because, for example, red bear and red deer are identical. First when the pictures on the objects become visible is the classification and the probability value of the expected object class much better what you can see on the diagram. In the third situation the algorithm was started with images where the objects were rotated about 180° (Figure 3 J and M) and in the last case - about 270° (Figure 3 K and N).

The first part of the experiments is presented in figure 5. It shows the probability of the best class (its certainty) against the number of fused images. The number of fused images amounts 18 (the rotation between two adjacent objects is 20°). As we can see in the diagram the value of certainty increases, what is expected from the fusion algorithm. At the beginning the fusion algorithm used informations only about a couple of images and as time dragged on informations many a frames of images is being used. The classification rates are presented in the table (Figure 4).

In the 0° column in the table the classification rates are good also at the beginning of the fusion algorithm. In this case the pictures (bear red, heart yellow etc.) are visible in the first fused image. For example if our algorithm starts with the image in position 90°, some objects could not be distinguished at the beginning. First when the algorithm comes to the image with the object in the position (270°) the picture on the object is visible and the probability of the expected class and classification rate become greater. Four example test images could be seen in the figure 6. The objects in the images were correct classified although they are paired nearly identical. This is because we took into account the fusion of more images in the recognition phase. An static approach for object recognition would give in this case rather chaotic results.

The training of one object class (270 images for the training of the object model and 90 for the training of the bounding region) executed on Pentium III (800 MHz) takes actually 14s. The recognition of one image on the same computer takes about a minute, hence the using of our algorithm in real-time applications is presently not possible.

5 Conclusions

In this article we presented a powerful statistical approach for classification and localization of 3-D objects based on the fusion of multiple views. In contrast to passive approaches, where the decision about class and pose of an object has to be taken

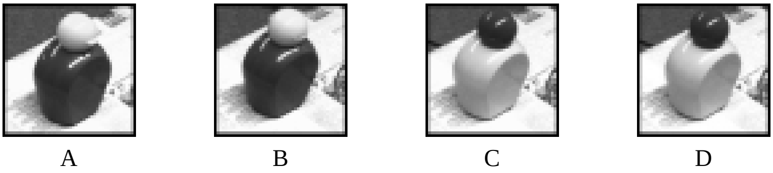

Figure 6: Example test images and results: A - class: yellow bear, result: yellow bear; B - class: yellow deer, result: yellow deer; C - class: red heart, result: red heart; D - red “star-money”, result: red “star-money”.

based on one image, we use more images. The additional images are used to gain more information about the scene and the observed objects. A general fusion scheme based on the Condensation algorithm [6] was presented. In section 3 we described the classifier which was used in our approach. To build the statistical model of any object we defined two dimensional feature vectors. The feature vectors were calculated with the wavelet transformation. A close boundary around the object (bounding region) was introduced to determine the object area in the image. The statistical model for the background was defined to avoid problems with occlusions and heterogeneous background. The results show that the using of information about more images makes the classification rate and the certainty better.

References

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, Boston, San Diego, New York, 1988.

[2] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz. Appearance based active object recognition. *Image and Vision Computing*, (18):715–727, 2000.

[3] C. K. Chui. *An Introduction to Wavelets*. ACP, San Diego, 1992.

[4] F. Deinzer, J. Denzler, and H. Niemann. Classifier Independent Viewpoint Selection for 3-D Object Recognition. In G. Sommer, editor, *Mustererkennung 2000*, pages 237–244, Heidelberg, September 2000. Springer.

[5] J. Denzler and C.M. Brown. Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2), 2002.

[6] M. Isard and A. Blake. CONDENSATION — Conditional Density Propagation for Visual Tracking. *IJCV*, 29(1):5–28, 1998.

[7] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, pages 35–44, 1960.

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

[9] N. Oswald and P. Levi. Object recognition with multiple observers. In *Autonome Mobile Systeme 2001*, pages 80–90. Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[10] J. Pösl. *Erscheinungsbasierte statistische Objekterkennung*. Shaker Verlag, Aachen, 1999.

[11] M. Reinhold, D. Paulus, and H. Niemann. Appearance-Based Statistical Object Recognition by Heterogenous Background and Occlusions. In B. Radig and S. Florczyk, editors, *Pattern Recognition, 23rd DAGM Symposium*, pages 254–261, München, September 2001. Springer-Verlag, Berlin, Heidelberg, New York. Lecture Notes in Computer Science 2191.

[12] M. Reinhold, D. Paulus, and H. Niemann. Improved Appearance-Based 3-D Object Recognition Using Wavelet Features. In T. Ertl, B. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *Vision, Modeling, and Visualization 2001*, pages 473–480, Stuttgart, November 2001. AKA/IOS Press, Berlin, Amsterdam.

[13] B. Schiele and J.L. Crowley. Transinformation for Active Object Recognition. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 249–254, Bombay, India, 1998.