Appearance Based Recognition of Complex Objects by Genetic Prototype-Learning

Martin Stommel
Realtime Learning Systems
University of Siegen
Hoelderlinstrasse 3
57068 Siegen, Germany
stommel@fb12.uni-siegen.de

Klaus-Dieter Kuhnert
Realtime Learning Systems
University of Siegen
Hoelderlinstrasse 3
57068 Siegen, Germany
kuhnert@fb12.uni-siegen.de

ABSTRACT

This paper describes a method to recognize and classify complex objects in digital images. To this end, a uniform representation of prototypes is introduced. The notion of a prototype describes a set of local features which allow to recognize objects by their appearance. During a training step a genetic algorithm is applied to the prototypes to optimize them with regard to the classification task. After training the prototypes are compactly stored in a decision tree which allows a fast detection of matches between prototypes and images. The proposed method is tested with natural images of highway scenes, which were divided into 15 classes (including one class for rejection). The learning process is documented and the results show a classification rate of up to 93 percent for the training and test samples.

Keywords
Pattern Recognition, Decision Trees, Genetic Algorithms

1. INTRODUCTION

The recognition of complex objects in digital images is one of the major tasks in computer vision with many applications in automatisation. Traditional graph based or grammar based approaches model complex objects as sets of homogeneous image regions [BB97][KC03]. The challenge in this approach is to achieve a robust segmentation under varying lighting conditions. Recent appearance based approaches avoid the segmentation step. Instead, pose, lighting conditions and the object composition are part of the model. Murase and Nayar [MN95] showed experimentally that it is possible to estimate the pose and illumination projecting an unknown image into a low dimensional subspace constructed from the sample images. Ettelt [Ett02] applied a decision tree to accomplish a fast object detection with an intensity based model. Low-pass filters were used to reduce the search space. Olson and Huttenlocher [OH97] used an edge-pixel based object model for object recognition in a military application. Nelson and Selinger [NS00] researched unsupervised learning in the context of appearance based object recognition.

VRBF RGNQS o`odqr oqnbddchmfr+ HRAM 7/,8/20//,8,4
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

The method presented in this paper uses a decision tree for the representation of a prototype based object model. This model combines edge-based and intensity based features in a uniform way. A genetic feature selection algorithm is used to obtain prototypes which generalize over one class but separate between different classes. The algorithm is tested with car images from highway videos using the background of the images as a natural rejection class.

2. OBJECT RECOGNITION

Structure of Prototypes

The use of prototypes is motivated by a probabilistic understanding of the object recognition process. From this point of view, an image of width *w* and height *h* is perceived as the joint ensemble $C_{0,0}C_{1,0}\cdots C_{w,0}\cdots C_{w,h}$ where $C_{x,y}$ is a triple (c_{xy}, A_C, P_{Cxy}) with c_{xy} denoting a random value (the colour at coordinate x,y) which takes on values from the alphabet A_C (a discrete colour space) with the probabilities P_{Cxy}. Assuming we have a certain number of sample images which represent objects of different classes a simple nearest neighbour classification algorithm would execute a pixelwise comparison between a new image and the sample images and then return the most similar one as the result. Following the probabilistic approach this can be regarded as the maximisation of the conditional probability

$$P(k \mid c^{u}_{0,0} = c^{k}_{0,0}, \ldots, c^{u}_{w,h} = c^{k}_{w,h})$$

that the unknown image corresponds to a certain sample image given the result of the pixelwise comparison. Here c^{k}_{xy} denote the pixel value for the coordinate *x,y* of sample image *k* and c^{u} denote the pixel values of the unknown image. From the observation that in natural images the colour is nearly constant over small distances follows that some pixels have a negligible influence on *P(k).* To achieve a viable computation time we exclude these pixels from the recognition process and benefit more from the information contained in the geometric relationship between the remaining pixels. In this paper the notion of a prototype is used to refer to such a set of pixels. More abstractly, a prototype is regarded as a list of local features $F=\{f_1, f_2,...,f_n\}$. Each feature $f = \{x,y,T,v\}$ describes an object with regard to the the position *(x,y)* of the feature measured in pixels, the type *T* of the feature and the discrete number *v* giving the value of the feature up to a sufficient accuracy. In this paper the red, green and blue component of a pixel, as well as the gradient orientation serve as features.

The comparison of a prototype with an unknown image is executed using special thresholds t_R, t_G, t_B, t_ϕ, according to the type of the feature. We obtain the box classifier

$$box(x,y) = \begin{cases} 1, if \left| c_{xy} - v \right| < t_{R,G,B} \\ 0, otherwise \end{cases}$$

for RGB-features and an analogous classifier for the gradient direction which takes care of the fact that the domain of *v* is a modulo ring here. With the number

$$M_{feat} = \sum_{i=1}^{n} box(x_i + x_{Offs}, y_i + y_{Offs})$$

of positive classifications for the features $\{f_1, f_2,...,f_n\}$ shifted by the offset (x_{Offs},y_{Offs}) relatively to the image we obtain a matching function

$$match(\{f_1,..., f_n\}, x_{Offs}, y_{Offs}) = \begin{cases} 1, if\, n = M_{feat} \\ 0, otherwise \end{cases}$$

which returns one if a prototype matches to the relative image position (x_{Offs},y_{Offs}) with regard to a number of fixed thresholds.

Genetic Feature Selection

After creating an initial set of prototypes a number of training cycles are executed which consist of a generation and a mutation operation. After each operation a measure of fitness is computed for every new prototype. The best prototypes are kept for further processing, while the worst prototypes are discarded. They will be replaced by new ones during the next generation or mutation step.

2.1.1 Prototype Generation

New prototypes are initialised with a random number of features at random positions. If the gradient at a feature position is above a certain threshold, the direction of the gradient is used as feature. Otherwise the red, green and blue component of the pixel serve as features. A [-0.5 -1 0 +1 +0.5] filter matrix is used to compute the horizontal and vertical gradient. The orientation is computed via the arc tangent. Since the box classificator is sensitive to impulsive noise, the sample images are first filtered with the known olympic filter.

2.1.2 Selection

The fitness of prototypes is measured in terms of selectivity and generalisability. A high selectivity means that a prototype matches only the desired class and produces few false positive results. A high generalisability on the other hand means that a prototype matches many sample images from one class. To determine the generalisability the number of matching relative positions

$$M_{prot} = \sum_{x,y \in I} match(F, x, y)$$

between an image *I* and the prototype *F* from class *k* is determined. An image is covered by *F* if there is at least one match. Denoting the number of images in class *k* covered by *F* with M_{cov}, the generalisability *g* is computed according to the equation

$$g = \frac{M_{\mathrm{cov}}}{class\ size(k)}. \tag{1}$$

The division by the class size is used to normalize the result to the interval [0 1]. To determine the selectivity the total number of matches of *F* to images from other classes than *k* is computed according to the equation

$$H = \sum_{I \notin k} \sum_{x,y \in I} match(F, x, y).$$

To speed up the computation a upper limit H_{Stop} of total matches is introduced, so the matching procedure can be stopped early for non-selective prototypes. The selectivity *s* is then computed as

$$s = 1 - \frac{\log 10(1+H)}{\log 10(H_{Stop})}. \tag{2}$$

The logarithm expresses that the selectivity is more interesting for lower numbers of matches. Like the generalisability the selectivity is normalized to the interval [0 1]. The fitness is then computed as the product of the selectivity and the generalisability.

2.1.3 *Mutation*

Mutation is carried out here by adding zero-mean gaussian noise of a certain standard deviation to the feature coordinates. If an edge feature moves into a homogenous region or vice versa, the type of the feature is adapted. Features are discarded if their coordinates leave the image borders. The prototypes are randomly chosen for mutation. The probability of a prototype to be chosen depends linearly on its position in the list of prototypes sorted by fitness.

Classification via Decision Trees

After training the prototypes are converted into a more compact decision tree representation which allows a fast image recognition. The tree structure is vaguely related to Quinlan's C4.5-Algorithm [Qui93]. The decision tree is used to classify all pixel positions of an image, which corresponds to adding an offset to the position of every feature *{x,y,T,v}* of a prototype. Since single non-selective prototypes have a strong influence on the search result, a further threshold is introduced to exclude bad prototypes from the tree generation.

3. EXPERIMENTS

Data Base

The experiments were conducted on a data base of 1900 car images and 120 images of highway scenes without cars. The car images stem from different highway videos and were cut out by hand and classified by a human. To make the object recognition process independent of the context, the background of the car images was removed manually. Figure 3.1 shows four cars from the data base in different positions, sizes and lighting conditions.

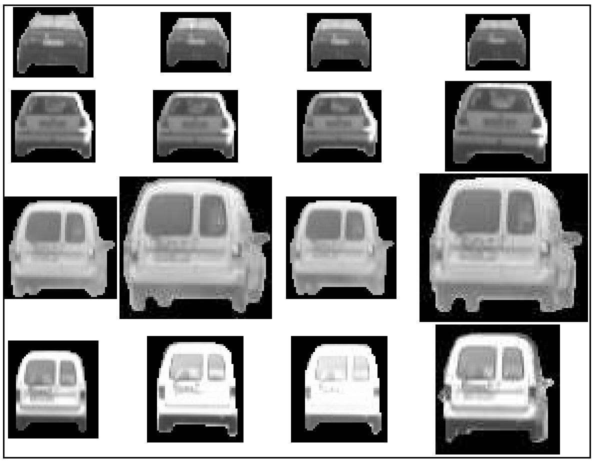

Figure 3.1. Examples from the car data base

The Learning Process

For the experiments the number of prototypes was set to 35 per class. Initial experiments on a two class problem showed that the number of prototypes is uncritical for values between 20 and 200. The learning process consisted of 899 training cycles. The number of prototypes obtained by a generation and mutation operation in every cycle was set to five.

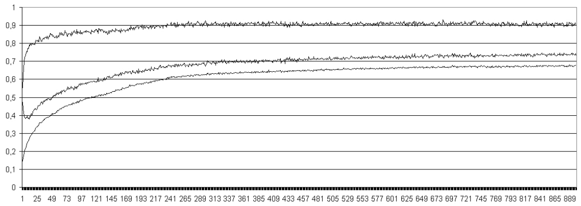

Figure 3.2. Learning rate during training: mean selectivity (upper curve), mean generalisability (mid), mean fitness (lower curve)

Figure 3.2 shows the learning curves for the genetic feature selection algorithm. The learning rate is measured in terms of mean selectivity and mean generalisability over all prototypes. It can be seen that the curves steadily increase during the first 300 training cycles and later converge to fixed values. The theoretical maximum of 1.0 resulting from the equations 1 and 2 is not reached in practice. From figure 3.3 it can be seen that for some classes no optimal set of prototypes can be determined.

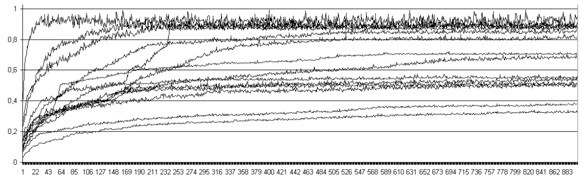

Figure 3.3. Mean fitness separated by classes

The reproduction of prototypes by mutation reduces the variety of sample images whose features are represented by the prototypes. Figure 3.4. illustrates this effect showing the number of different sample images used to form the prototypes of one class averaged over all classes. This observation has lead to the fixed assignment between classes and a set of prototypes.

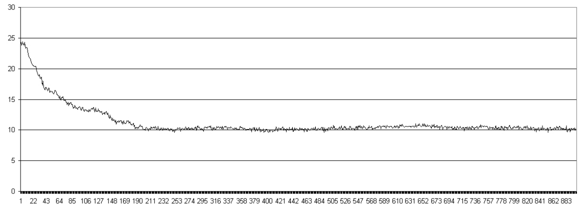

Figure 3.4. Variety among prototypes

Figure 3.5 shows the coverage of the training samples by the prototypes. The rate of recognized training samples stays between 92 and 100 percent during the training process. The curve does not approach 100 percent because the set of prototypes is not trained for a complete coverage of the sample images as a whole. Instead the algorithm searches for single prototypes which represent good features to recognize different classes of objects.

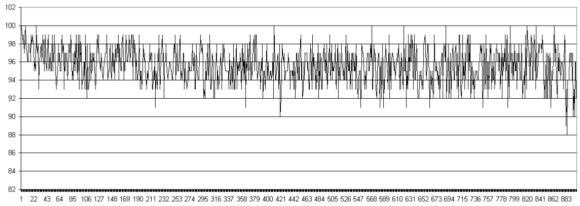

Figure 3.5. Coverage of the training samples

Classification Results

The classification results for the trained prototypes are shown as confusion matrices. To facilitate the computation of confusion matrices objects and background were treated separately. All samples images were classified giving one vote to every image, i.e. the distribution of positive results was normalised to a sum of 1.0 for every image. The background images were used to compute the classification rate for the rejection class. Here both positive and negative results are counted. Table 3.1 shows the proportion of true positives and the total number of predictions (from another training process with identical parameters). A classification rate of 93.1 percent is reached after 25 training cycles, which is an improvement of 6.8 percent over the untrained prototype set. For the following training cycles the results are worse, which seems to be a result of the decreasing variety among the prototypes. For later training cycles the accuracy increases again as a result of the growing generalisability. The similarity for the training and test samples indicates that the results are representative.

Training cycles	Training samples	Test samples
0	0.883	0.863
25	0.926	0.931
50	0.848	0.859
100	0.870	0.874
200	0.850	0.847
400	0.890	0.894
800	0.927	0.908

Table 3.1. Accuracy of the object recognition

Figure 3.6 shows the confusion matrix after 800 training cycles. Row 1 represents the rejection class. As apparent from the pictures the false matches are not evenly distributed over the confusion matrices but can be divided into objects which are not recognized at all and classes which are often misclassified as one single other class. The first group results from the training of single prototypes rather than sets of prototypes. The second kind of error is caused by single ambiguous prototypes.

4. CONCLUSION, FUTURE WORKS

The object recognition method presented in this paper employs trained prototypes to find and classify objects by their appearance. The training is conducted with objects from real world images using their natural background as rejection class.

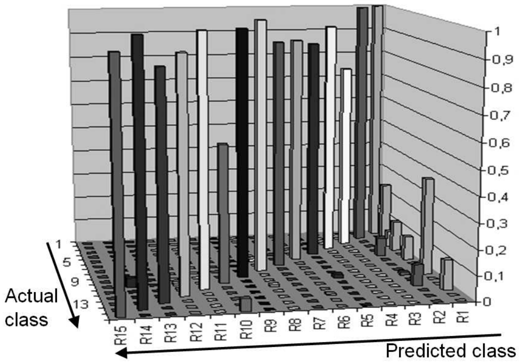

Figure 3.6. Confusion matrix for the test samples

As our experiments show, the structure of prototypes is capable of recognizing complex objects without a previous segmentation step and an explicit higher-level model. The problem of model construction and comparison to noisy segmentation results is avoided. The analysis of the learning process shows that single prototypes can be trained successfully to generalize over one class and at the same time to distinguish between objects of different classes. The experiments show also a certain potential for improvements concerning the coverage of the samples by the prototypes. Thus, in our future work we will investigate how whole sets of prototypes can be trained instead of single prototypes.

5. REFERENCES

[BB97] Burge, M., Burger, W., Learning Visual Ideals. Proc. of the 9th ICIAP, Florence, Italy, Sept. 1997.

[Ett02] Ettelt, E. Appearence Based Object Recognition by Use of Optimized Template Trees. Doctoral Thesis, TU Muenchen, Germany, 2002.

[KC03] Kerr, J., Compton, P., Towards Generic Model-based Object Recognition by Knowledge Acquisition and Machine Learning. Mixed Initiative Intelligent Systems, Eds. Gheorghe Tecuci, Morgan Kaufmann, Acapulco, Mexico, 2003, pp. 107-113.

[MN95] Murase, H., and Nayar, S.K. Visual learning and recognition of 3d objects from appearance. Int. Journal of Computer Vision, 14(1):5-24, Kluwer, January 1995.

[NS00] Nelson, R.C., Selinger, A., Learning 3D Recognition Models for General Objects from Unlabeled Imagery: An Experiment in Intelligent Brute Force. 15th Int. Conf. on Pattern Recognition (ICPR 2000), Barcelona, Spain, Sept. 3-8, 2000, Vol.1, pp.1-8.

[OH97] Olson, C.F., Huttenlocher, D.P., Automatic Targed Recognition by Matching Oriented Edge Pixels. IEEE Transactions On Image Processing, Vol.6, No.1, January 1997.

[Qui93] Quinlan, J.R. C4.5: Programms for Machine Learning. Morgan Kaufmann, San Mateo, California, 1993.