Learning of Form Models from Exemplars

Petra Perner and Silke Jähnichen

Institute of Computer Vision and Applied Computer Sciences IBaI
Körnerstr. 10, 04107 Leipzig
eibaiperner@aol.com
www.ibai-institut.de

Abstract. Model-based image recognition requires a general model of the object that should be detected. In many applications such models are not known a priori, but have to be learnt from examples. In this paper we describe our procedure for the acquisition and learning of general contour models. We developed a modified Procrustes algorithm for alignment and similarity calculation of shapes. Based on the calculated pair-wise similarity we learn groups of shapes. For each group we calculated prototypes. The set of prototypes will be used as models for the detection of object instances in new images.

1 Introduction

One of the most commonly encountered problems in image analysis is the recognition of objects in an image. This can be done by a model-based object recognition method. Such a method works as follows: A shape model is applied to the image and matched according to the pixel points of the image. If the considered pixels have an appearance that is similar to the model points, then the result of the matching process will be a score equal to one for identity and less then one for similarity. The basis for such a method is a good model of the object to be recognized and a good similarity measure. We will consider in this paper the generation of the models from exemplars.

The model can be generated synthetically or from the original image. We are considering the process where the model or a set of models should be learnt from a set of instances elicited from a set of real images. Generally we are attempting to solve the following problems: Create a set of *m* shape instances from real images, each is represented by a set of arbitrary boundary points. Align these shapes and calculate the pairwise similarity, partition them into a set of clusters and, for each shape cluster, compute a prototype. The set of prototypes will be used as models for the detection of object instances in new images.

In Section 2 of this paper we give the basic notion and we briefly describe the material used for this study. The acquisition of shape instances is described in Section 3. The alignment of shapes and the computation of the pair-wise similarity is described in Section 4. Clustering and prototype calculation is presented in Section 5 and Section 6. Finally we give results in Section 7. The methods described in this paper are implemented in a program named *CACM Version 1.0* that assists the user in the acquisition of 2-D shapes and learns groups of shape models and their prototypes. An outlook to our research is given in Section 8.

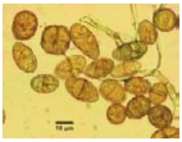
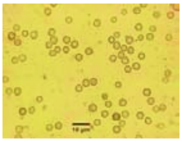
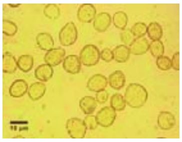
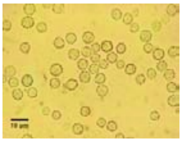
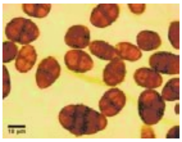
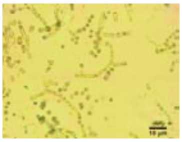
A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 153–161, 2004.

2 Basic Notions and Material Used for this Study

Model-based object recognition can be done based on the object model or based on the contour model. We are considering the contour of an object *S* but not the appearance of the object inside the contour. Therefore we want to elicit from the real image the shape $S_c = \{(x_{ci}, y_{ci})\}\, i = 1 \ldots n_c$ represented by a set of n_c boundary points (x_c, y_c).

The material we used for our study are fungal strains that are naturally 3-D objects, but which are acquired in a 2-D image. These objects have a great variance in the appearance of the shape of the object because of their nature and the imaging constraints. Six fungal strains representing species with different spore types were used for the study (Table 1). A database of images from the spores of these species was produced.

Table 1. Images of Six Different Fungi Strains

		
Alternaria Alternata	Aspergillus Niger	Rhizopus Stolonifer
		
Scopularioupsis Brevicaulis	Ulocladium Botrytis	Wallenia Sebi

3 Acquisition of Shape Cases

We obtain the set of boundary pixels by implementing into our program a function that allows the user to mark the contour S_C of an object *S* by moving the mouse cursor of the computer or by moving an electronic pen over a digitizer tablet (Figure 1). Notice that the sampled points are not required to be landmark co-ordinates [1] or curvature extrema. The user starts labelling an object *S* at an arbitrary pixel $\mathrm{s}_{\mathrm{start}}$ of its contour. After having traced the complete object the labelling ends at a pixel s_j in the 8-neighbourhood of s_{start}. To obtain the complete set S_C of all boundary pixels we need to ensure that the contour is closed, which means s_j is a direct neighbour of s_{start}. Therefore we insert missing boundary pixels using the Bresenham [2] procedure.

As a result of the labelling process we obtain set S_C with an amount of n_C ordered, connected points that describe the boundary of object *S* with the highest possible accuracy as far as is possible with this kind of labelling procedure.

Having labelled the contour S_C of the object *S* its boundary pixels are still defined by their absolute position in the 2-D matrix of the original image. In order to describe and compare the shapes of objects it is useful to specify a common co-ordinate system that is invariant for translation. Therefore we transform the set S_c of boundary points to the origin $x = 0$ and $y = 0$.

A following approximation of the contour might reduce this set of pixels to a sufficiently large set of pixels that will speed up the succeeding computation time of the alignment and clustering process. The numbers of pixels in this set will be influenced by the chosen order of the polygon and the allowed approximation error. Our approach to the polygonal approximation is based on the area/length ratio according to Wall and Daniellson [3].

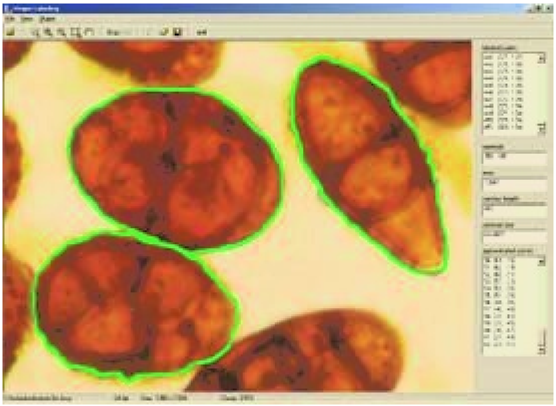


Fig. 1. Labeled and Approximated Shape with Co-ordinates

4 Shape Alignment and Similarity Calculation

4.1 Theory of Procrustes Alignment

The aim of the alignment process is to compare the shapes of two objects in order to define a measure of similarity between them. Consider two shape instances *P* and *O* defined by the point-sets $p_i \in R^2, \quad i = 1, 2, ..., n_C$ and $o_j \in R^2, \quad j = 1, 2, ..., n_K$ respectively. The basic task of aligning two shapes consists of transforming one of them (say *P*) so that it fits in some optimal way the other one (say *O*). Generally the shape instance $P = \{ \; (\; x_i^P , y_i^P \;) \; \}$ with $i = 1...n_c$ is said to be aligned to the shape instance $O = \{ \; (\; x_j^O , y_j^O \;) \; \}$ with $j = 1...n_k$ if a distance function *d(P, O)* between the two shapes cannot be decreased by applying to *P* a transformation Θ. The differences between various alignment approaches is the group of allowed transformations (similarity, rigid, affinity) on one side and the definition of the distance function on the other side.

In our application we use for the Procrustes distance [4] [5], a least-squares type distance function. The alignment of shapes is limited to similarity transformation in order to eliminate differences in rotation and scale of the two shapes *P* and *O*.

After computing a similarity transformation between *P* and *O* the Procrustes distance is defined by

$$d(P,O)=\sum_{i=1}^{N_c}\left\|\frac{(p_i-\mu_P)}{\sigma_P}-R(\theta)\frac{(o_i-\mu_O)}{\sigma_O}\right\|^2 \tag{1}$$

where θ is the rotation matrix, μ_P and μ_O are the centroids of the object *P* and *O* respectively and σ_P and σ_O are the sums of squared distances of each point-set from the centroids.

In its basic form, the Procrustes alignment centers and scales each set of points so that the sum of squared distances of all points in each point-set is unity. Then it is possible to compute a similarity transformation based on these centered pre-shapes. Finally the Procrustes average shape and Procrustes residuals can be evaluated.

4.2 Our Approach to Shape Alignment

As described in Section 3 we are considering a set of shape instances where differences in translation were already eliminated. To compare the shapes of two instances we still have to eliminate differences in rotation and scale. As a measure of similarity we use the Procrustes distance between all points of *P* and their correspondences in *O*.

As can be seen from equation (1) the Procrustes distance requires the knowledge of point correspondences between the shapes *P* and *O*. Therefore we are confronted with the following problems:

1. In our application we use an approximation of the manually labelled set of contour points instead of a predefined number of landmark coordinates. Therefore we cannot guarantee that all shape instances are defined by an identical number of contour points. We can only assume to have nearly the same amount of contour points regardless of which size or shape an object has.
2. The point correspondences between the two shapes of the instance *P* and *O* are completely unknown.
3. As a result of the above-mentioned facts we do not have information about point outliers either.

The Softassign Procrustes Matching algorithm [7] solves the point correspondence problem using deterministic annealing. This algorithm works robust with respect to outlier identification and noise, but is it also a computationally-expensive procedure. Belongie et al. [8] found correspondences between points on the basis of the shape context descriptor.

We are solving the problem of unknown point correspondences by applying an iterative robust point matching algorithm. The outline of our approach to shape alignment is as follows:

For every pair of points $(p_i, o_j) \in P \ x \ O$ we calculate the similarity transformation ψ_{ij} that aligns these two points $\{ p_i, o_j \}$. The transformation ψ_{ij} is applied to all points in *P* to obtain the transformed shape instance *P'*, which is defined by the point-set $p'_i \in R^2, \quad i=1, 2, ..., n_C$. For every point p'_i we define the nearest neighbour $NN(p'_i)$ in *O* as a point correspondence of p'_i. Note that we do not enforce one-to-one point correspondences. One point in *O* can have more than one point correspondence or even not a single point correspondence in *P*. The sums of squared distances $d_{ij}(P, O)$ between every point correspondence were added. In addition to that we define the quantity $\sqrt{\frac{1}{k} * d_{ij}(P,O)}$ as the mean alignment error $\bar{\varepsilon}_{ij}(O,P)$:

$$\bar{\varepsilon}_{ij}(P,O) = \sqrt{\frac{1}{k} * d_{ij}(P,O)} \tag{2}$$

with

$$d_{ij}(P,O) = \sum_{i=1}^{k} (pi'.x - NN(pi').x)^2 + (pi'.y - NN(pi').y)^2 \tag{3}$$

If the distance $d_{ji}(P, O)$ is smaller then all earlier calculated distances, $d_{min}(P, O)$ is set to $d_{ji}(P, O)$, $\bar{\varepsilon}_{\min}(O,P)$ is set to $\bar{\varepsilon}_{ij}(O,P)$ and ψ_{min} is set to ψ_{ij}.

After having cyclically aligned every possible pair of points $(p_j, o_i) \in P \, x \, O$, we may estimate the similarity between the shape instances *P* and *O* based on the value of $d_{min}(P, O)$.

To ensure that the final measure of similarity ranges from 0 to 1, we normalize the measure $\bar{\varepsilon}_{\min}(O,P)$ to a predefined maximum distance *T*:

$$\bar{\varepsilon}'_{\min} = \frac{\bar{\varepsilon}_{\min}(P,O)}{T} \tag{4}$$

If $\bar{\varepsilon}_{\min}(O,P) = 0$ then the shape instance *P* is identical with the shape instance *O*. With an increasing value of $\bar{\varepsilon}_{\min}(O,P)$ the shape instance *P* is less similar to the shape instance *O*. If $\bar{\varepsilon}_{\min}(P,O) > T$ then the term $\frac{\bar{\varepsilon}_{\min}(P,O)}{T}$ is automatically set to value one.

It is obvious that the constant *T* has a direct influence to the value of the resulting score. The parameter *T* can be defined by the user in the settings dialog of our program *CACM*. For our calculations we set *T* to 35% of the centroid size of *O*. Our investigations showed that this value leads to good results. Figure 2 shows pair-wise aligned shapes and the calculated values of the dissimilarity measure. It can be seen that in case of identity the shapes are superposed. If the similarity score is less than one, we can see a deviation of the two shapes.

5 Clustering

The alignment of every possible pair of objects in our database leads us to *N*N* pair-wise similarity measures between the *N* shape instances. These distances can be collected in an *N x N* matrix where each row and each column corresponds to an instance of our data set.

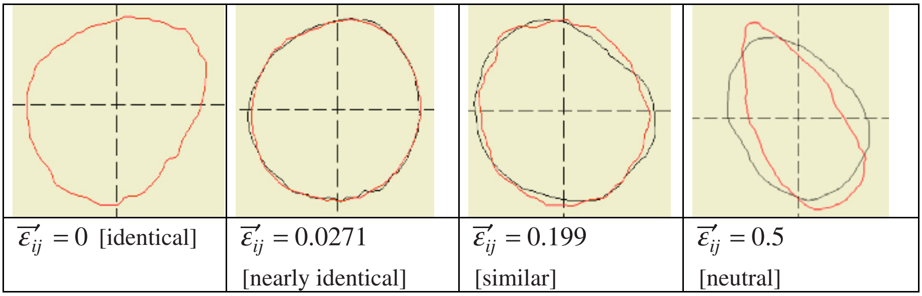

$\overline{\varepsilon}'_{ij} = 0$ [identical]
$\overline{\varepsilon}'_{ij} = 0.0271$ [nearly identical]
$\overline{\varepsilon}'_{ij} = 0.199$ [similar]
$\overline{\varepsilon}'_{ij} = 0.5$ [neutral]

Fig. 2. Aligned Shape Instances of Strain Ulocladium Botrytis with Distances

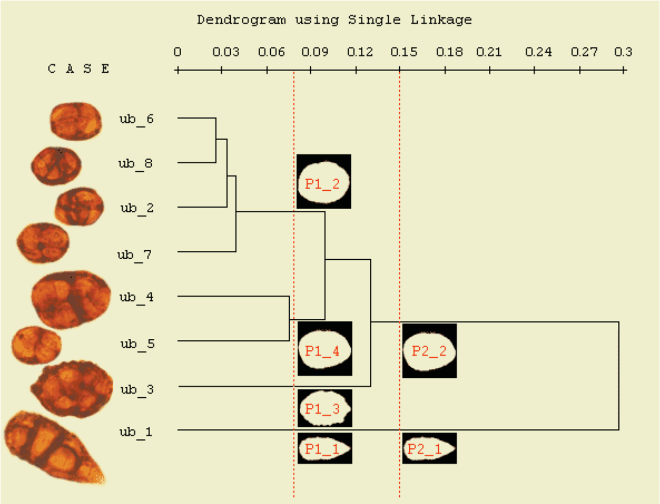


Fig. 3. Dendrogram of Shapes of the Object Ulocladium Botrytis and the calculated Prototypes

We want to point out that we do not obtain a symmetric square matrix where the distance *d(A, B)* between an individual *A* and an individual *B* is identical to the distance *d(B, A)* between individual *B* and individual *A*. This is obviously a lack of precision, but until we do not enforce a strict one-to-one mapping between corresponding points we can only assume that $d(A, B) \approx d(B, A)$.

Based on this similarity matrix we can divide our set of shape instances into groups or clusters. The clustering is done using the single linkage method [6]. The result of the hierarchical cluster analysis can be graphically represented by a dendogram. The dendogram for the shapes of the fungi strain ulocladium botrytis is shown in Figure 3. The dendogram shows the relative distances between all individuals. The merging of individuals into clusters is done with increasing distances (from left to right) until all individuals are combined in only one cluster. The exemplary cut-point (vertical red-dotted line) at a distance 0.15 results in two different clusters. The first cluster which consists only of the object with ub_1 is represented by prototype *P2_1*. The second cluster consists of the other seven objects $\{ub\_2, ub\_3, \ldots, ub\_8\}$ and is represented by prototype $P2\_2$.

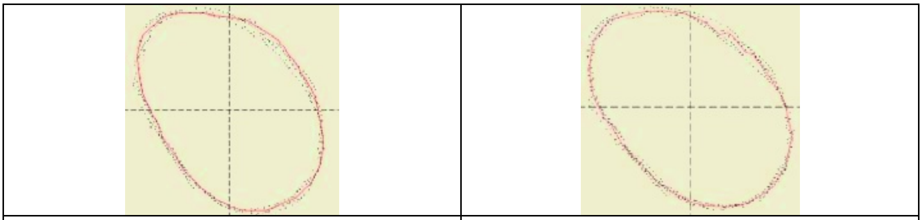
6 Prototype Calculation

Each cluster consists of a subset of j shapes S_1, S_2...S_j. For each cluster we can now compute a prototype μ that will be the representative of the cluster. This prototype can be calculated by computing the mean over all shapes in a cluster. As result we will get an artificial prototype, a prototype that does not exist in reality. Therefore we decided to calculate the median of all shapes in a cluster.

As the median shape of that cluster we choose the shape instance which has the minimum distance to all other shape instances

$$[\hat{\mu}] = S_{\min} = \arg\inf \sum_{i=1}^{j} d\left(S_{\min}, S_i\right) \tag{5}$$

The main advantage of this solution is that the model represents a natural shape that is included in the cluster. An example of using a natural shape instance as the prototype of a cluster is shown in Figure 4a. In contrast to this the arithmetic mean as prototype is shown in Figure 4b. Visually we would favor the median shape as prototype for the cluster since it appears to be more smooth.



(a) using a natural shape instance as prototype

(b) using the arithmetic mean as prototype

Fig. 4. Median of Shapes in a Cluster and Arithmetic Mean of the Shape

7 Results

We have applied our method to six different airborne fungi spores (see Table 1). We have labelled a total of 60 objects for each of the 6 fungal strains. In the following registration process we have aligned every single object with all objects of the same strain to calculate the measure of similarity between them. As a result we have obtained six squared similarity matrices, one for each analyzed fungal strain. These matrices were the input for the following cluster analysis. The outcome of this process was a dendrogram for each of the six different fungi strains. Table 2 presents the number of models for each class at two different cut-points.

For both cut-points we calculated the corresponding set of clusters in each strain class. The prototypes of these clusters were used as models for the later recognition process which is not part of this work. Figure 5 shows as an exemple the database of models for the class Rhizopus Stolonifer at cut-point (1). We can see that a large number of models is required for good detection of the object Rhizopus Stolonifer.

Table 2. Number of Models for two different Cut Points

Classes	max. Distance	Cut-Point (1)	Number of Models	Cut-Point (2)	Number of Models
Alternaria Alternata	0.5264	0.035	23	0.031	34
Aspergillus Niger	0.2936	0.098	3	0.094	5
Rhizopus Stolonifer	0.4275	0.058	16	0.055	22
Scopulariopsis Brevicaulis	0.4911	0.095	8	0.083	10
Ulocladium Botrytis	0.5332	0.043	24	0.040	30
Wallenia Sebi	0.5202	0.050	7	0.046	10

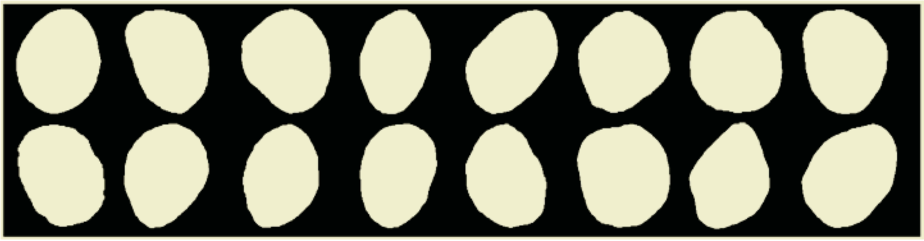


Fig. 5. Database of Models for Strain Rhizopus Stolonifer representing the 16 resulting Clusters

8 Conclusions

The recognition of objects in images can be done based on a model-based recognition procedure. That requires to have a model from the objects which should be recognized. Natural objects have a great variation in shape that does not make it easy to specify a model by hand. Therefore it is necessary to have a computerized procedure that helps to acquire the model from the real objects. We have proposed a method for the acquisition of contour instances and the learning of general shape models. We use the Procustes similarity measure for aligning and determining the similarity between different shapes. Based on the calculated similarity measure we create clusters of similar shapes by using the single linkage method. The mean shape or the median of the cluster is calculated and taken as a prototype of the cluster. The methods are implemented in the program *CACM Version 1.0* which runs on a window PC.

Acknowledgement

The project “Development of methods and techniques for the image-acquisition and computer-aided analysis of biologically dangerous substances BIOGEFA” is sponsored by the German Ministry of Economy BMWA under the grant number 16IN0147.

References

1. A. Hill, Ch. J. Taylor, and A.D. Brett, A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence, IEEE PAMI, vol. 22, No. 3, pp. 241-251
2. P. Zamperoni, Methoden der digitalen Bildverarbeitung, Vieweg Verlag 1995
3. K. Wall and P.-E. Daniellson, ”A fast sequential method for polygonal approximation of digitized curves”, Comput. Graph. Image Process. 28, 220-227
4. I.L. Dryden and K.V. Mardia, Statistical Shape Analysis, John Wiley&Sons, 1998.
5. S.R. Lele and J.T. Richtsmeier, An Invariant Approach to Statistical Analysis of Shapes, Chapman &Hall/CRC 2001
6. P. Perner, Data Mining on Multimedia Data, Springer Verlag, lnai 2558, 2003
7. A. Rangarajan, H. Chui and F.L. Bookstein, The Softassign Procrustes Matching Algorithm, Proc. Information Processing in Medical Imaging, pp. 29-42, 1997
8. S. Belongie, J. Malik and J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 24, pp. 509-522, 2002