Case-Based Object Recognition with Application to Biological Images

Petra Perner

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

Abstract. There are many biotechnological applications where 3-dimensional objects are represented as 2-d objects in a digital image. The dynamic and variable nature of the microorganism thus creates a formidable challenge to the design of a robust 2-d image inspection system with the ideal characteristics of high analysis accuracy but wide generalization ability. We have developed a novel case-based object recognition method for this kind of problems. The method is able to recognize objects and learn incrementally cases for the recognition process. Such a procedure requires capturing new cases for the further recognition process in order to be able to handle the variability of the natural objects. We describe the theory behind the method and how it works on our problem of fungi spore recognition. The developed case-based object recognition method is flexible and robust enough to be used for different recognition tasks in biotechnology.

Keywords: Object Recognition, Similarity Measure, Image Mining, Case-Based Reasoning.

1 Introduction

There are many biotechnological applications where 3-dimensional objects are represented as 2-d objects in a digital image. The dynamic and variable nature of the microorganism thus creates a formidable challenge to the design of a robust 2-d image inspection system with the ideal characteristics of high analysis accuracy but wide generalization ability.

We have developed a novel case-based object recognition method for this kind of problems. The method is able to recognize objects and learn incrementally cases for the recognition process. Such a procedure requires capturing new cases for the further recognition process in order to be able to handle the variability of the natural objects. Image mining is applied to the newly captured cases in order to keep the case base as small as possible. As a result we obtain groups of similar cases for which we are prototypically calculating cases that are stored into the case base. These learnt cases are applied for case-based object recognition. For the object recognition procedure we have developed a novel similarity measure that can determine similarity between the cases in the case base and the objects in the image. The similarity measure is flexible

J.F. Martínez-Trinidad et al. (Eds.): CIARP 2006, LNCS 4225, pp. 27 – 37, 2006.

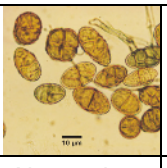
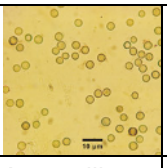
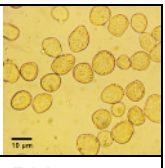
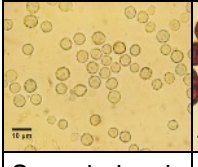
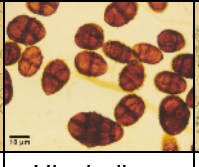
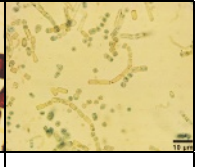
enough in order to get adjusted for different recognition purposes. We describe the theory behind it and how it works on our problem of fungi spore recognition.

The developed case-based object recognition method is flexible and robust enough to be used for different recognition tasks in biotechnology.

2 Material

Some digitized sample images are presented in Table 1 for the different fungal spores. The objects in the images are good representatives for the nature of different kind of biological objects such as yeast, cells, algae, and others.

Table 1. Images of fungal strains

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

3 Case-Base Object Recognition

The objects in the image are highly structured. Our study has shown that these images, represented in Table 1, cannot be segmented by thresholding. The objects in the image can be occluded, touching, or overlapping. It can also happen that only parts of the objects appear in the image. Therefore we decided to use a case-based object recognition procedure for the detection of objects in the image.

A case-based object recognition method uses cases that generalize the original objects and matches these cases against the objects in the image. During the match a score is calculated that describes the goodness of the fit between the object and the case. Well known similarity measures are the normalized cross correlation [1], the Hausdorff distance [2] and the chamfer matching [3]. We did not use the gray values of the objects, but used the object edges instead. For the score of the match between the case and the image we modified the normalized cross correlation in order to measure the average angle between the vectors of the case and the object.

The case can be an object model that describes the inner appearance of the object as well as the contour. In our case the appearance of the whole objects can be very diverse. The shape seems to be the feature that generalizes the objects. Therefore we decided to use the contour of the objects as case representation.

3.1 Similarity Measure

We determine the similarity measure based on the cross correlation by using the direction vectors of an image. This requires the calculation of the dot product between each direction vector of the model $\vec{m}_k = (v_k, w_k)^T$, $k = 1, \ldots, n$, and the corresponding image vector $\vec{i}_k = (d_k, e_k)^T$:

$$s_1 = \frac{1}{n}\sum_{k=1}^{n} \vec{m}_k \cdot \vec{i}_k = \frac{1}{n}\sum_{k=1}^{n} \langle \vec{m}_k, \vec{i}_k \rangle = \frac{1}{n}\sum_{k=1}^{n} (v_k \cdot d_k + w_k \cdot e_k) \cdot \tag{1}$$

The similarity measure of Equation (1) is influenced by the length of the vector. That means that *s1* is influenced by the contrast in the image and the model. In order to remove the contrast, the direction vectors are normalized to the length one by dividing them through their gradient:

$$s_2 = \frac{1}{n}\sum_{k=1}^{n} \frac{\vec{m}_k \cdot \vec{i}_k}{\|\vec{m}_k\| \cdot \|\vec{i}_k\|} = \frac{1}{n}\sum_{k=1}^{n} \frac{v_k \cdot d_k + w_k \cdot e_k}{\sqrt{v_k^2 + w_k^2} \cdot \sqrt{d_{k_i}^2 + e_k^2}} \cdot \tag{2}$$

Note that the normalization of *s2* differs from the normalized cross correlation (NCC): The NCC normalizes each pixel value by the expected mean of all values of the considered pixels. Therefore it is sensitive to nonlinear contrast changes whereas our method is not. The similarity measure in Equation (2) takes into account only the angle between the direction vectors, i.e. it is invariant against illumination changes. The value of $arccos\ s_2$ indicates the mean angle between the model vectors and the image vectors.

The values of *s2* can range from *-1* to *1*. In case of *s2 = 1* and *s2 = -1* the model and the image object are identical. If *s2* is equal to one, then all vectors in the model and the corresponding image vectors have the same direction. If *s2* is equal to *-1* then the vectors have exactly opposite directions, that means only the contrast between the model and the image is changed.

Contrast changes can be ignored if the absolute value of the dot product is calculated:

$$s_3 = \frac{1}{n}\sum_{k=1}^{n} \frac{\left|\vec{m}_p \cdot \vec{i}_p\right|}{\|\vec{m}_p\| \cdot \|\vec{i}_p\|}. \tag{3}$$

The aim is to store only one model for objects with similar shapes of different scale and rotation. Therefore a transformed model must be compared to the image at a particular location.

3.2 Case Generation

The acquisition of the templates is done semi-automatically. Prototypical images are displayed to an expert. The screenshot of our developed tool is shown in Fig. 1. The expert manually traces the contour of the object with the help of the cursor of the computer. Afterwards the number of the contour points are reduced for data reduction purposes by interpolating the marked contour by a 1st order polynom. The marked object shapes are then aligned by Procrustes Algorithm [6]. From a set of shapes general groups of shapes are learnt by clustering. Single-linkage technique is used for clustering [4]. The prototype of each cluster is calculated by estimating the median shape [5] of the set of shapes in the cluster and taken as an object model.

3.2.1 Shape Approximation

An approximation of the contour might reduce this set of contour 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 [6]. We use the first labeled point *s1* of the object *S* as the starting point *p1* for the first approximation. Next, we virtually draw a line segment from the starting point *p1* to the successor point in *B*. The area *A* between this line and the corresponding contour segment of the object *S* is measured. If the area divided by the length *L* of the line is smaller than a predefined threshold *T*, then the same process is repeated for the next successor point in the set *B*.

This procedure is repeated until the ratio exceeds the threshold *T*. In that case the current point of set *B* becomes the end point of the approximated line *P* and the starting point for the next approximation. The same process is then repeated until the last point in set *B* is reached. The result of the approximation is a subset *C* of *m* points *p1…pm* where $m < n$ and $pi \in N$.

The ratio A / L controls the maximal error of the approximation, since A is the area and L the side length of a virtual rectangle. If the ratio is low, then the other side of the virtual rectangle is small and wise-versa.

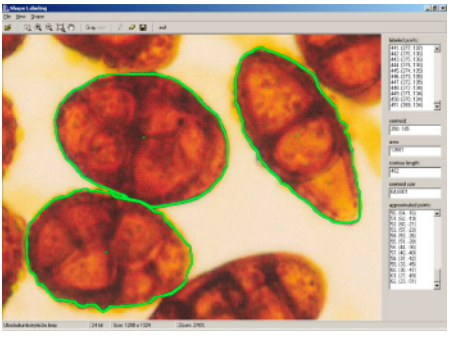


Fig. 1. Labeled and approximated shapes with coordinates

3.2.2 Shape 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 $p_i \in R^2$, $i = 1,2,...N_1$ and $o_j \in R^2$, $j = 1,2,...N_2$, 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 $P = \{p_i(x, y)\}_{i=1...N_1}$ is said to be aligned to the shape instance $O = \{o_j(x, y)\}_{j=1...N_2}$ if a distance $d(P, O)$ between the two shapes can not be decreased by applying a transformation ψ to P. The differences between various alignment approaches is the group of allowed transformations (similarity, rigidity, affinity…) on one side and the chosen distance function on the other side.

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

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

$$D(P,O) = \sum_{i=1}^{N} \left\| \frac{(p_i - \mu_P)}{\sigma_P} - R(\theta)\frac{(o_i - \mu_O)}{\sigma_O} \right\|^2 . \tag{4}$$

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

In the 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 a similarity transformation based on these centered pre-shapes is computed. Finally the Procrustes average shape and Procrustes residuals can be evaluated. Our algorithm is described in [7].

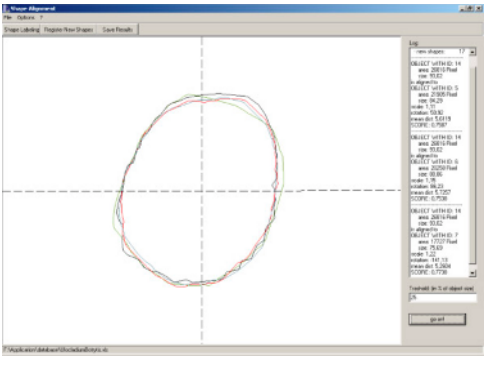


Fig. 2. Aligned shapes of objects Ulocladium Botrytis

4 Our Conceptual Clustering Algorithm

Conceptual clustering [8] is a type of flexible learning the hierarchy by observations. The partitioning of the cases is controlled by a category utility function [4]. Conceptual clustering algorithms can be distinguished by the type of this utility function which can be based on a probabilistic [9], [10] or a similarity concept [11]. Our conceptual clustering algorithm presented here is based on similarities, because we do not consider logical but numerical concepts. The algorithm works directly with structural objects. In our study this is a set of N acquired cases $\{S_1,\ldots,S_N\}$, each comprised by an ordered array of 2D contour points. In contrast to the agglomerative clustering methods where the distance matrix is used as input it is not necessary to calculate pair-wise distances at first.

In addition to merging cases our algorithm allows incorporating new cases into existing nodes, opening new nodes, and splitting of existing nodes at every position in the hierarchy. Each new case is successively incorporated, so the algorithm dynamically fits the hierarchy to the data. The result will be a sequence of partitions represented a directed graph (concept hierarchy) where the root node contains the complete set of input cases and each terminal node represents an individual case.

Initially the concept hierarchy only consists of an empty root node. The algorithm implements a top-down method. A new case is placed into the actual concept hierarchy level by level beginning with the root node until a terminal node is reached. In each hierarchy level one of these four different kinds of operations is performed:

The new case is incorporated into an existing child node,

A new empty child node is created where the new case is incorporated,

Two existing nodes are merged to form a single node where the new case is incorporated, and

An existing node is splitted into its child nodes.

The new case is tentatively placed into the next hierarchy level by applying all of these operations. Finally that operation is performed which gives the best score of the partition according to the evaluation criteria. A proper evaluation function prefers compact and well separated clusters. These are clusters with small inner-cluster variances and high inter-class variances. Thus we calculate the score of a partition comprised of the clusters $\{X_1, X_2, \ldots X_m\}$ by

$$SCORE = \frac{1}{m}\sum_{i=1}^{m} p_i \left(SB_i - SW_i\right) , \tag{5}$$

where m is the number of clusters in this partition, p_i is the relative frequency of the i-th cluster, SB_i is the inter-cluster variance and SW_i is the inner-cluster variance of the i-th cluster. The normalization according to m is necessary to compare partitions of different size. The relative frequency p_i of the i-th cluster is

$$p_i = \frac{n_i}{n} , \tag{6}$$

where n_i is the number of cases in the i-th cluster and n is the number of cases in the parent node. The output of our algorithm for applying the eight exemplary shape cases of strain Ulocladium Botrytis is shown in Fig. 4. On top level the root node is shown which comprises the set of all input cases. Successively the tree is partitioned into nodes until each input case forms its one cluster.

We also introduced a pruning criterion into the algorithm which can be used optionally. It says that the clusters $\{X_1, X_2, \ldots X_m\}$ in one partition are removed if the sum of their inner-cluster-variances is *zero*. The criterion is fulfilled if the following condition is met

$$\sum_{i=1}^{m} SW_i = 0 \ . \tag{7}$$

Fig. 3 shows the complete, un-pruned concept hierarchy, where a new case called $\{ub\_9\}$ was incorporated supplementary. The darker nodes were those clusters which had to be modified because the new case was incorporated into them. For these nodes the cluster has to be updated. The white nodes in the hierarchy are clusters which were not attached.

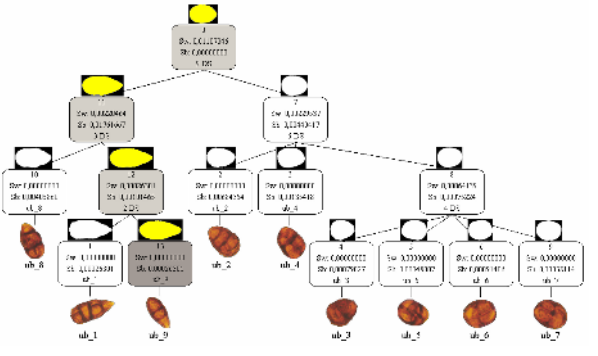


Fig. 3. Complete, not-pruned concept hierarchy after incrementally incorporating a new case

The main advantage of our conceptual clustering algorithm is that it brings along a concept description. Thus, in comparison to agglomerative clustering methods it is easy to understand why a set of cases forms a cluster. The algorithm calculates the inner-cluster-variances direct on the cases within this cluster or rather on their contour points instead of using a given distance matrix. During the clustering process the representative case, and also the variances and maximum distances in relation to this representative case are calculated since they are part of the concept description. It is also possible to incorporate new cases into the existing learnt hierarchy. Thus, the algorithm is of incrementally fashion.

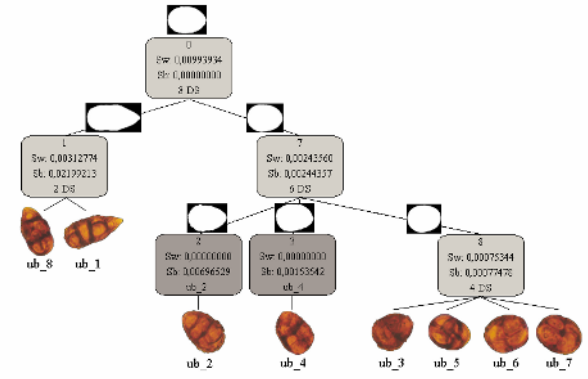


Fig. 4. The pruned version of the concept hierarchy resulting from the eight instances of strain Ulocladium Botrytis is shown

5 Calculation of General Cases

The representative case of a cluster is a more general representation of all cases hosted in this cluster. We select the medoid as a natural representative case for a cluster. The medoid x_{medoid} of a cluster X is the shape case which is positioned closest to the cluster centroid. It is the case which has the minimum distance to all other cases in the cluster

$$\bar{\mu}_X = x_{medoid} = \min_{x \in X} \sum_{i=1}^{n_x} d(x_i, x) \; . \tag{8}$$

In addition to the representative of a cluster we are interested in learning the maximum permissible distance from this generalized case. The maximum permissible distance D_X to the representative case is

$$D_X = \max_{x \in X} \; d(x, \bar{\mu}_X) \; . \tag{9}$$

When matching objects with a hierarchical casebase of increasing specialized cases it is important to know the degree of generalization for each case. This measure will be used as threshold for the similarity score while matching.

6 Experimental Results

We applied our method to six different airborne fungi spores (see Table 1). We labeled a total of 60 objects for each of the six fungal strains. These objects were taken for the case generation process according to the procedure as described in Section 4. The result was a casebase of 79 cases for the six different fungal strains. Despite the result in [12] where for each separate class the number of models was calculated we get a reduction in the number of models of 27.5%. That is because

some of the fungal strain have the same appreance in shape and get clustered into the same cluster with others although the dimension of these objects might be different. Table 3 illustrates this fact.

The 79 cases were inputed in our case base and were applied to a test set of images contaning the same number of images for each class.

The threshold for the score was set to 0.8. We calculated the recognition rate as the number of objects that were recognized correctly in the image to the total number of objects in the images. The results of the matching process are shown in Table 2 and Table 3. Compared to our former results in [12] we got a better recognition rate since with our conceptual clustering method we could better control the grouping of the objects and the level of the hierarchy where the clusters should be taken from. The strategy for finding the cut-off for the grouping by conventional hierarchical clustering methods is a bit more tricky and subjective.

Table 2. Recognized objects in the image

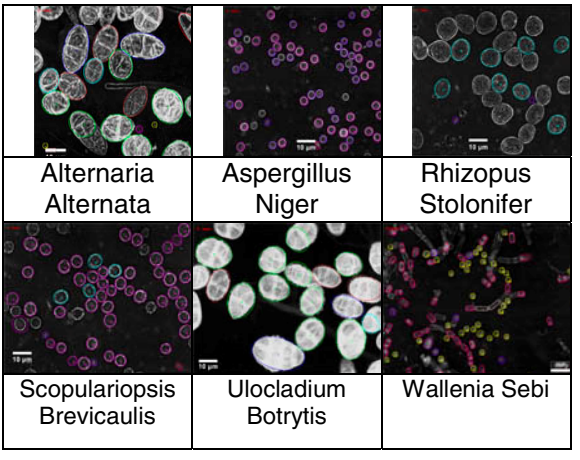


Table 3. Cluster number versus class membership

	Cluster Membership of Classes					
Number of Clusters	Alternaria Alterna	Ulocladium Botrytis	Aspergillus Niger	Rhizo. Stol.	Scopul. Brev.	Wallenia Sebi
1		X				
2						X
3	X	X				
...	...	...	...	...	...	...
11	X	X				X
12	X	X				
13		X				
14						X

Table 3. (*continued*)

...	...	...	...	...	...	...
28		X		X		X
29		X				
30				X		
31	X	X		X		
...	...	...	...	...	...	...
63	X	X	X		X	
64	X		X		X	
65	X			X		
66	X		X			
67	X				X	
68			X		X	
69					X	
70					X	
...	...	...	...	...	...	...
77					X	
78					X	
79					X	
Number of Member ship	26	33	9	25	18	26

Table 4. Results of matching

Class	Classification Accuracy in %
Alternaria Alternata	93%
Aspergillus Niger	92%
Rhizopus Stolonifer	94%
Scopulariopsis Brevicaulis	89%
Ulocladium Botrytis	92%
Wallenia Sebi	85%

6 Conclusions

We have described our method for the recognition of biological objects such as e.g. fungi spores in digital microscopic images.

We used a case-based recognition method. The cases are represented by edges and not by the gray-level itself. The similarity measure is based on the scalar product and

is invariant against illumination changes and contrast changes. The case generation was done by manually tracing the contour of the object, automatic shape alignment, conceptual shape clustering and prototype calculation. The clustering process is incremental and allows to incorporate a new object into the excisting cluster hierarchy.

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 BMWI under the grant number 16IN0147 and by the European Commission within the project "Multimedia Understanding through Semantics, Computation, and Learning" No. 507752.

References

1. Brown, A Survey of Image Registration Techniques, ACM Computer Surveys 24 (4), 1992, pp. 325-376.
2. C.F. Olson, D.P. Huttenlocher, Automatic Target Recognition by Matching Oriented Edge Pixels, IEEE Transactions on Image Processing 6(1), 1997, p 103-113.
3. G. Borgefors, Hierarchical Chamfer Matching: A Parametric Edge Detection Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 10(6), 1988, p. 848-865.
4. P. Perner, Data Mining on Multimedia Data, Springer Verlag, lnai 2558, 2003.
5. I.L. Dryden and K.V. Mardia, Statistical Shape Analysis, John Wiley&Sons, 1998.
6. K. Wall and P.-E. Daniellson, A fast sequential Method for Polygonal Approximation of digitized Curves, Comput. Graph. Image Process. 28, pp. 220-227, 1984.
7. P. Perner and S. Jänichen, Learning of Form Models from Exemplars, In In: Ana Fred, Terry Caelli, Robert P. W. Duin, Aurelio Campilho, and Dick de Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Springer Verlag 2004, lncs 3138, pp. 153-161
8. S. Jänichen and P. Perner, Conceptual Clustering and Case Generalization of 2-dimensional Forms, Journal on Computational Intelligence, to appear 2006
9. D. Fisher and P. Langley, Approaches to conceptual clustering, Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 691-697, Los Angeles, 1985.
10. W. Iba and P. Langley, Unsupervised Learning of Probabilistic Concept Hierarchies, In G. Paliouras, V. Karkaletsis, & C. D. Spyropoulos (Eds)., Machine learning and its applications. Springer Verlag, 2001.
11. P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, In B. Smith and P. Cunningham (Eds.), Advances in Case-Based Reasoning, pp. 251-261, Springer Verlag, lnai 1488, 1998.
12. P. Perner, H. Perner, and S. Jänichen Recognition of Airborne Fungi Spores in Digital Microscopic Images, Artificial Intelligence in Medicine, Volume 36, Issue 2 , February 2006, p. 137-157 (available on-line 3 October 2005)