Artificial Intelligence in Medicine (2006) **36**, 137—157

ELSEVIER

ARTIFICIAL
INTELLIGENCE
IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Case-based object recognition for airborne fungi recognition

Petra Perner[*], **Silke Jänichen, Horst Perner**

Institute of Computer Vision and Applied Computer Sciences, IBaI,
Körnerstrasse 10, 04107 Leipzig, Germany

Received 15 December 2004; received in revised form 27 March 2005; accepted 25 April 2005

KEYWORDS
Case-based object recognition;
Similarity measure;
Acquisition of form models;
Shape alignment;
Clustering;
Fungi spore recognition;
Biotechnological application

Summary

Objective: Model-based object recognition is a well-known task in Computer Vision. Usually, one object that can be generalized by a model should be detected in an image based on this model. Biomedical applications have the special quality that one object can have a great variation in appearance. Therefore the appearance of this object cannot be generalized by one model. A set of cases of the appearance of this object (sometimes 50 cases or more) is necessary to detect this object in an image. The recognition method is rather case-based object recognition than model-based object recognition. Case-based object recognition is a challenging task.
Methods and material: It puts special requirements to the similarity measure and needs a matching algorithm that can work fast in a large number of cases. It also needs a case acquisition procedure that can capture the great variation in appearance of an object and generalize these data into a case description. In this paper we describe the chosen case representation, the similarity measure and the matching as well as the case acquisition procedure. We evaluate our method based on a large enough set of digital images containing biological objects such as fungi spores.
Results: We can show that the similarity measure is superior to detect the objects in the images. The developed method for case acquisition and learning of generalized cases allows us to learn interactively a sufficient number of cases that are further stored into our case base. Finally, we give results on the performance of the system by calculating the recognition rate.
Conclusion: These result show that we have developed a novel similarity measure for object detection in digital grey-level images and a novel procedure for case acquisition and learning that allows us to learn a sufficiently large enough case base and to generalize over a group of cases.

[*] Corresponding author. Tel.: +49 341 8612 273; fax: +49 341 8612 275.
E-mail address: ibaiperner@aol.com (P. Perner).
URL: http://www.ibai-institut.de

0933-3657/$ — see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2005.04.002

1. Introduction

Case-based object recognition is used to detect objects of interest on a complex background where thresholding-based image segmentation methods fail. The basis for such a recognition method is a set of cases of the appearance of the object that should be recognized. A case is comprised of a set of contour points of an object and the object name. The recognition is done by retrieving similar cases from the case base and successively matching these cases against the images. Objects whose points match the case points give a high recognition score and they are marked in the actual image.

In this respect case-based object recognition differs from model-based object recognition [1,2]. Model-based object recognition is a well-known task in Computer Vision. Usually, one object that can be generalized by a model should be detected in an image based on this model. Biomedical applications have the special quality that one object can have a great variation in appearance. Therefore, the appearance of this object cannot be generalized by one model. A set of cases of the appearance of this object (sometimes 50 cases or more) is necessary to detect this object in an image.

Case-based object recognition is a challenging task. It puts special requirements on the similarity measure and needs a matching algorithm that can work fast in a large number of cases. The basis for such an object recognition method is a good set of cases that describes the variation in appearance of the object on different abstraction levels, so that it is possible to detect an object with a sufficiently high recognition score. Besides the case-based object recognition method, we developed a special case acquisition procedure which mines a large number of data for a more generalized case description. This case is put into the case base for case-based object recognition.

In this paper we describe our work on an application for health monitoring. Airborne microorganisms are ubiquitously present in the various fields of indoor and outdoor environments. The potential implication of fungal contaminants in bio-aerosols on occupational health is recognized as a problem in several working environments. The continuous monitoring of airborne biological agents is consequently a necessity, as well for the detection of risks for human health as for the smooth sequence of technological processes. We use our case-based object recognition method in order to recognize fungi spores in digital microscopic image. The object recognition unit is our first step on the way to an automatic image interpretation system for the detection and interpretation of airborne fungi spores. Note that the recognition method furnishes us information only if there is an object in the image that is very likely a fungi spore. It gives us also information about the shape of the object, but it does not give us the final information about the kind of fungi spore. This information can only be obtained after further processing steps.

In this paper we consider six digital microscopic images of fungal cultures. The acquisition technique and the sample images are given in Section 2. In Section 3 we describe the basic architecture of a case-based object recognition system. Different image and case representations are presented in Section 4. An overview about similarity measures for object recognition is given in Section 5, as well as the similarity measure that we developed for our application. In Section 6 the hypothesis-test strategy is presented. The overview about the recent matching algorithm is given in Section 7. We discuss the case acquisition in Section 8. Finally, we give results on the performance of the system in Section 9 and discuss implementation details. Conclusions and an outlook to further research are given in Section 10.

2. Image acquisition and sample images

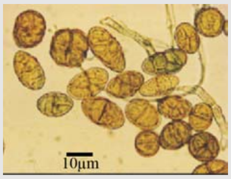
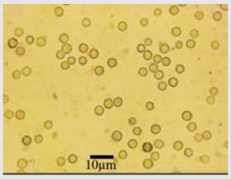
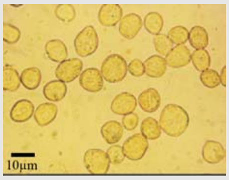
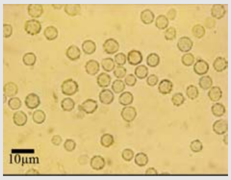
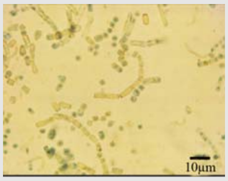
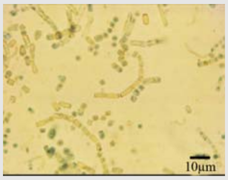
2.1. Fungal cultures

Six fungal strains representing species with different spore types were used for the study (Table 1). The strains were obtained from the fungal stock collection of the Institute of Microbiology, University of Jena/Germany and from the culture collection of JenaBios GmbH. All strains were cultured in Petri dishes on 2% malt extract agar (Merck) at 24 °C in an incubation chamber for at least 14 days. For microscopy fungal spores were scrapped off from the agar surface and placed on a microscopic slide in a drop of lactic acid. Naturally hyaline spores were additionally stained with lacto phenol cotton blue (Merck). From the spores of these species a database of images was produced.

2.2. Image acquisition

Image acquisition was conducted using a Zeiss-Axiolab transmission light microscope equipped with a 100× lens and a NIKON Coolpix 4500 digital color camera. The magnification is 1000×, using a 100× objective. The resulting pixel size ranges from 0.1 to 0.025 μm. The average spore size of common airborne fungi varies between 2 and 40 μm. Some

Table 1 Images of fungal strains

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

digitized sample images are presented in Table 1 for the different fungal spores. The objects in the images are good representatives of the different kinds of fungal spores cultured under optimal conditions.

3. Case-based object recognition

The heart of our case-based object recognition system is a case base of shapes. These shapes are represented as contour chains. Therefore, a case is comprised of a set of contour points $S_C = \{s_c(x_0, y_0); 1 \leq c \leq n\}$ where each contour point has the grey value of 1 and a class label for the shape. Based on this information we can transform the shape from the contour point list into a 2D image matrix, in the following called case image. The case base is filled up for the actual application by shapes that we learnt based on our tool for information gathering, case acquisition and case mining CACM (see Section 8). An index of the case base should allow us to find the closest case among the numerous cases in short time. A case image is matched against the image by constructing an image pyramid from the actual image and the case image. This allows us to reduce the computation time while matching. Beginning with the highest level of the image pyramid the scores are calculated and the areas of interest are marked. The area of interest is the area where an object can be detected. This area is recursively used for further matching by going the levels of the image pyramid downward. Finally the closest match is given to the output. Depending on the actual value of the similarity measure, the next level of the index structure is selected and the process repeats until a final node is reached.

The construction of this index is not part of this paper. It is left for further work. In this paper we want to describe the image representation, the image similarity, and the implementation of the system, the case acquisition method, as well as the matching results. The architecture of our case-based object recognition system is shown in Fig. 1.

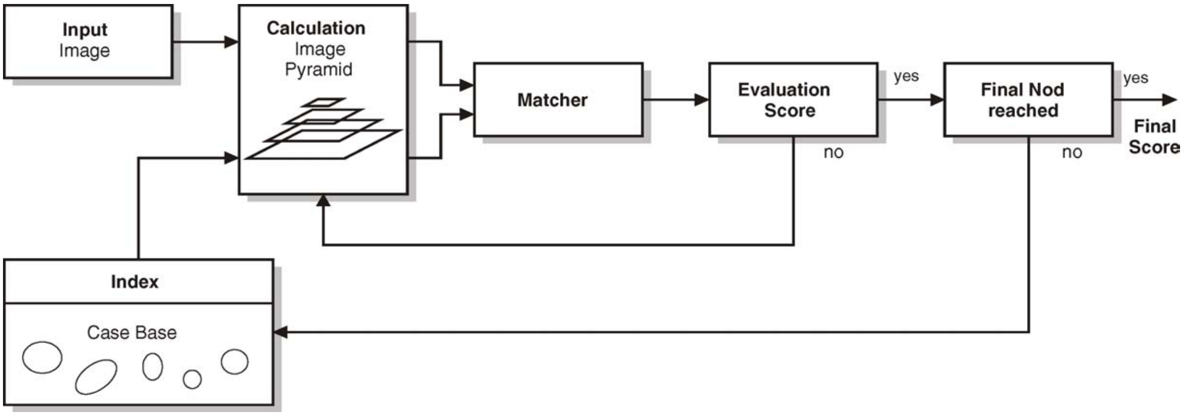


Figure 1 Architecture of a case-based object recognition system.

4. Data representation

4.1. Case representation

In general we can distinguish between three different case representations according to the pixels that are used for matching:

1. *Region of interest* (*ROI*): A region of interest ROI (see Table 2b) is obtained by taking a cut-out from the original image (see Table 2a). All pixels of the obtained image matrix are used as case pixels regardless of whether they are object or background pixels.
2. *Object case*: In the image matrix shown in Table 2b only those pixels are used as case points that lie inside and at the contour of the object (see Table 2c). In this case the shape and the inner structure of the object are taken into consideration.
3. *Contour case*: Only pixels that lie on the contour of an object are taken as case points (see Table 2d). Thus only the shape of the object of interest is matched.

Which kind of case representation is best suited depends on the application. Since we analyze fungi spores it is inefficient to use an object case, because the appearance of the structure inside the objects is very diverse. It would result in a case base where for each case is stored an object. The only representation that gives us a more generalized view of the objects is the shape. Therefore we use a contour case as case representation.

Note that an object might appear in an image with a different size and under a different rotation angle and on various locations in an image. But it is still the same object. It makes no sense to store all these identical objects, which however differ in size and rotation, in the case base. Rather there should be stored a unit object with the origin coordinates x_0 and y_0 that can be translated, resized and rotated during the matching process. Therefore the case pixels $\vec{p}_k = (t_k, u_k)^{\mathrm{T}}$ and the direction vectors $\vec{m}_k = (v, w)^{\mathrm{T}}$ have to be transformed with a matrix A to:

$$\vec{p}\,'_k = A \cdot \vec{p}_k, \qquad \vec{m}\,'_k = A \cdot \vec{m}_k \tag{1}$$

If φ denotes the angle of rotation and r the scaling factor the matrix may look like the following:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} r\cos\varphi & -r\sin\varphi \\ r\sin\varphi & r\cos\varphi \end{pmatrix} \tag{2}$$

4.2. Image representation

Since we are looking for the contour of the object which is the boundary between the background and the object and which is usually an area of high grey-level change, we are representing the image by the edges. The edges can be represented by the gradient of the pixels. In order to determine the gradient, first the direction vector $\vec{z}_{(x,y)} = (\Delta x, \Delta y)^{\mathrm{T}}$ of a pixel at the position (x, y) is calculated from the grey-level matrix. The direction vector indicates the change of the grey value in vertical and horizontal direction, respectively. The length of this vector is equal to the gradient and it is commonly determined from the direction vector through the following formula:

$$||\vec{z}_{(x,y)}|| = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{3}$$

Due to the discreteness of the grey-level matrix which represents the grey value function only in some well-chosen points, the direction vectors cannot be calculated by the known analytic derivation formula. Therefore, many operators were developed that allow us to determine the direction vectors from the grey-level matrix. We used the Sobel operator. The corresponding edge image is obtained by applying such an operator to the grey-level

Table 2 Different case representations

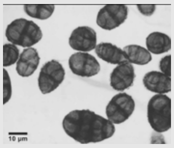
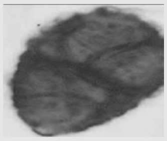
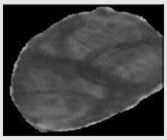
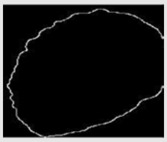
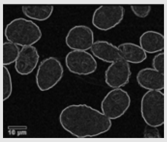
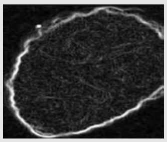
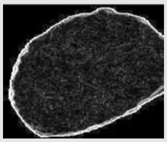
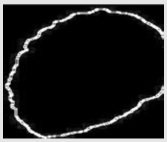
(a) Original image	(b) ROI case	(c) Object case	(d) Contour case
			
(e) Sobel filtered original image	(f) Sobel filtered ROI case	(g) Sobel filtered object case	(h) Sobel filtered contour case
			

image. After that the pixels represent the gradient instead of the grey-level value. Besides that the direction vectors for each pixel are stored. This representation is calculated for the case and the actual image before matching.

5. Similarity measure

5.1. Basics

Common similarity measures for object recognition are the normalized sum of squared differences [3] (NSSD) and the normalized cross correlation [3] (NCC). The measure NSSD calculates the difference between each normalized point of the model and each normalized point in the image. Then the difference is squared and summed up. The measure NCC calculates the sum of the product between each point of the model and the corresponding point of the image and normalizes this value to the second order moment of the image and the model points.

The normalized cross correlation and the normalized sum of squared differences are invariant to linear brightness changes, but they are very sensitive to nonlinear brightness changes. However, if the sum of the squared differences is not normalized, then it is also sensitive to linear brightness changes.

It can be shown that both measures behave similarly. The realization of the SSD measure is simpler than the correlation measure which was of advantage in times where computers were not so powerful. In general, the SSD measure is also more intuitive than the normalized cross correlation, because the meaning of the difference of two values is easier to understand for a human than the meaning of the product.

Another commonly used measure is the Hausdorff distance [4]. It is a measure for comparing binary features. A binary feature in an image can be produced by thresholding. The image pixels that go over a given threshold are set to one, otherwise they are set to zero.

Let A be the set of the model pixels and let B be the set of the image pixels with the value of one. The directed Hausdorff distance $h(A, B)$ ranks each point in A based on its closeness to a point in B. It takes the value of the most mismatched point of A. In general $h(A, B)$ is unequal to $h(A, B)$. The Hausdorff distance $H(A, B)$ of the model and the image is the maximum of the both directed Hausdorff distances. If the model and the image are identical, then the score takes on the value of zero. The similarity of the model and the image decreases as much as the score increases; but there is no distinct value of inequality.

The disadvantage of the Hausdorff distance is its sensibility to occlusion and clutter: The score increases if some of the model pixels are missing in the image (that means occlusion) and also when the image contains more pixels than the model (that means clutter). In order to overcome this weakness, Huttenlocher et al. [4] presented a modified Hausdorff distance which calculates the maximum of the kth largest distance between the image and the model pixels and the maximum of the lth largest distance between the model and the image pixels, respectively. If the number n of model pixels and the number m of image pixels are known, then the modified Hausdorff distance is robust against $100k/n\%$ occlusion and $100l/m\%$ clutter.

Olson and Huttenlocher [5] extend the Hausdorff distance to oriented edge pixels. An oriented edge pixel is described by the three-dimensional vector $p = (p_x, p_y, p_o)$. The Hausdorff distance is adapted to handle also oriented edge pixels. Olson and Huttenlocher point out that the recognition process is faster if the orientation is considered.

However, another disadvantage of the Hausdorff distance is that we have to search for each pixel in the set A the closest pixel of the set B. This process may be very time-consuming. Olsen and Huttenlocher [5] improve the performance by restricting the set of image pixels so that the distance between any image pixel and the next model pixel is lower than a predefined value δ.

The similarity measure introduced in Latecki and Lakämper [6] is based on the contour of an object which is represented by a polygon. This polygon is subdivided into a set of consecutive line segments and the corresponding tangent function is determined. The similarity of a model and an object is defined as the distance of their tangent functions. Since this distance is normalized with the lengths of the line segments, the similarity measure is invariant against scale differences of the model and the object.

5.2. Similarity measure based on the dot product

5.2.1. Global similarity

As we have pointed out above, the calculation of the Hausdorff distance is more costly than the calculation of the cross correlation. While we have to search for correspondences between case and image pixels in case of using the Hausdorff distance, we evaluate the image pixels that coincide with the case pixels by using the cross correlation. On the other hand we are interested in matching oriented edge pixels which Olson and Huttenlocher [5] described for the Hausdorff distance. Therefore we propose a similarity measure based on the cross correlation and by using the direction vectors of an image. This approach requires the calculation of the dot product between each direction vector of the case $\vec{m}_k = (v_k, w_k)^{\mathrm{T}}$ and the corresponding image vector $\vec{i}_k = (d_k, e_k)^{\mathrm{T}}$:

$$s_1 = \frac{1}{n}\sum_{k=1}^{n} \vec{m}_k \cdot \vec{i}_k = \frac{1}{n}\sum_{k=1}^{n} \langle \vec{m}_k, \vec{i}_k \rangle$$

$$= \frac{1}{n}\sum_{k=1}^{n} (v_k \cdot d_k + w_k \cdot e_k) \qquad (4)$$

with k = 1, . . ., n case pixels.

The similarity measure of Eq. (4) is influenced by the length of the vector. That means that s_1 is influenced by the contrast in the image and the case. In order to remove the contrast, the direction vectors are normalized to the length one by dividing them through their gradient:

$$s_2 = \frac{1}{n}\sum_{k=1}^{n} \frac{\vec{m}_k \cdot \vec{i}_k}{\|\vec{m}_k\| \cdot \left\|\vec{i}_k\right\|}$$

$$= \frac{1}{n}\sum_{k=1}^{n} \frac{v_k \cdot d_k + w_k \cdot e_k}{\sqrt{v_k^2 + w_k^2} \cdot \sqrt{d_k^2 + e_k^2}} \qquad (5)$$

In this respect the similarity measure 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, the normalized cross correlation is sensitive to nonlinear illumination changes, while our method is not because it takes only into account the angle between two corresponding direction vectors.

The values of s_2 can range from -1 to 1 (see Table 3). If s_2 is equal to one then all vectors in the case and the corresponding image vectors have the same direction (see Table 3a and b). If s_2 is equal to -1 then all the image vectors have exactly opposite directions compared to the case vectors. That means that only the contrast between the case and the image is changed (see Table 3a and c). The similarity values for these image constellations are also shown in Table 3.

The above described global contrast changes can be excluded by computing the absolute value of s_2:

$$s_3 = \left| \frac{1}{n}\sum_{k=1}^{n} \frac{\vec{m}_k \cdot \vec{i}_k}{\|\vec{m}_k\| \cdot \|\vec{i}_k\|} \right| \qquad (6)$$

However, in case half the vectors have the same contrast and the other half has the opposite contrast, then the similarity based on s_3 is zero. That might not be preferable for cases where objects are touching. To avoid this we calculate the similarity based on s_4:

$$s_4 = \frac{1}{n}\sum_{k=1}^{n} \frac{|\vec{m}_k \cdot \vec{i}_k|}{\|\vec{m}_k\| \cdot \|\vec{i}_k\|} \qquad (7)$$

5.2.2. Local similarity

The sum over all comparisons of pixels in the case and the image gives us a global similarity measure. The local similarity between a pixel in the case and the corresponding pixel in the image gives us a normalized local similarity SIM_k of these two vectors:

$$\mathrm{SIM}_k = \frac{\vec{m}_k \cdot \vec{i}_k}{\|\vec{m}_k\| \cdot \|\vec{i}_k\|} = \cos(\angle \vec{m}_k \vec{i}_k) = \cos\varphi$$

$$\text{with } \vec{m}_k \cdot \vec{i}_k = \|\vec{m}_k\| \cdot \|\vec{i}_k\| \cdot \cos(\angle \vec{m}_k \vec{i}_k)$$

$$\text{and } \mathrm{SIM}_k = 0 \text{ if } \|\vec{i}_k\| = 0 \qquad (8)$$

That means the local similarity value is defined by the cosine function and takes values between -1 and 1. A small angle φ which means a small deviation between the two vectors brings a high contribution to the similarity, while a big angle φ brings a small contribution to the similarity. The local similarity function is nonlinear).

The possible relations between the case and the image vector are shown in Table 4. Because of the

Table 3 The effect of contrast changes to the scores of the similarity measures

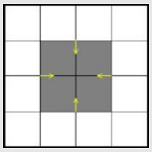
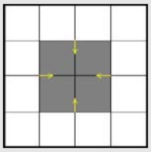
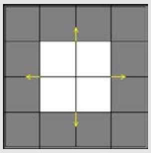
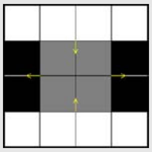
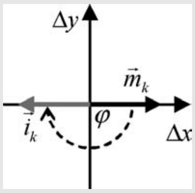
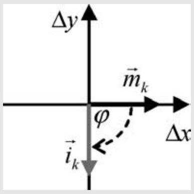
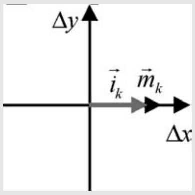
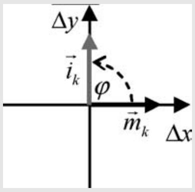
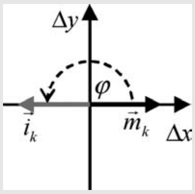
(a) Case	(b) Object with identical contrast	(c) Object with globally inversed contrast	(d) Object with locally inversed contrast
			
s_2	1	-1	0
s_3	1	1	0
s_4	1	1	1

Table 4 Relation between the case and the image vector for some values of φ

φ	$-\pi$	$-\frac{\pi}{2}$	0	$\frac{\pi}{2}$	π
Relation between $\vec{m}_k$ and $\vec{i}_k$					
SIM_k	−1	0	1	0	−1
Interpretation	Opposite	Neutral	Identical	Neutral	Opposite

periodicity of the angles we can restrict the values of φ to the interval $[-\pi, \pi]$. That means that we consider the lowest absolute difference between the case and the image direction as well as the orientation of φ, which is represented by the sign.

The interpretation of this similarity is that 1 is identity, −1 is opposite and 0 is neutral (see Table 4). If we want to consider similarity and dissimilarity, then we have to rescale our similarity measure to:

$$\overline{\mathrm{SIM}_k} = \frac{1 + \mathrm{SIM}_k}{2} \tag{9}$$

This rescaling results in a different interpretation of the values. A value of 0.8 by SIM_k has a higher value of 0.9 by SIM_k (see Fig. 2). If we use the first local similarity defined in Eq. (8) we will detect lower matches than in the case of the second local similarity defined in Eq. (9).

In case that the direction of the vectors should not influence the similarity value, we have to calculate the absolute value of the similarity:

$$\mathrm{ABS\_SIM}_k := |\mathrm{SIM}_k| = \left| \frac{\vec{m}_k \cdot \vec{i}_k}{||\vec{m}_k|| \cdot ||\vec{i}_k||} \right| = |\cos \varphi| \tag{10}$$

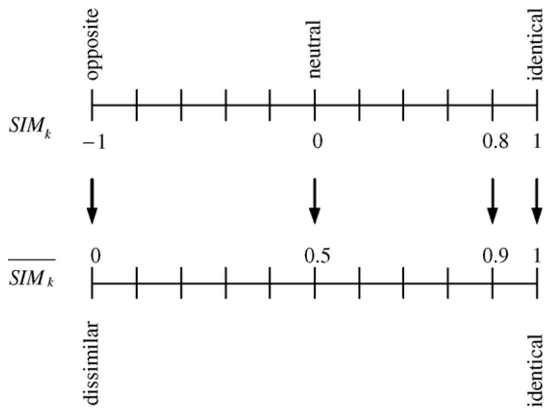


Figure 2 Influence of the rescaling to the local similarity values.

This modification also restricts the values of the local similarity $\mathrm{ABS\_SIM}_k$ to the interval [0, 1]. The modified similarity measure $\mathrm{ABS\_SIM}_k$ eliminates direction changes of the vector. It depends on the application whether such behavior of the similarity measure is requested.

The similarity measure is based on the angle of the vectors regardless of the length of the vector. Therefore already small edges that can appear in almost homogenous areas or in the background, increase the similarity. In order to eliminate the influence of unwanted edges we have to introduce a threshold for the gradient:

$$\frac{1}{||\vec{i}_k||} = \begin{cases} 0 & ||\vec{i}_k|| \leq l_{\min} \\ \dfrac{1}{||\vec{i}_k||} & \text{in all other cases} \end{cases} \tag{11}$$

How we select the threshold $l_{\min}$ depends on the application. In our case we did not want that the case fires in the background area. Therefore we calculated the mean $(\hat{\mu})$ and the standard deviation $(\hat{\sigma})$ of the gradient over several background images and set the threshold to the value of $\hat{\mu} + n \cdot \hat{\sigma}$. The introduction of a threshold prevents false matches in the image background. In fact, it has also an influence on real edges. If they do not exceed the threshold, there is no match. Therefore an optimum has to be found between no-background matches and object matches.

Note, if we consider homogeneous areas inside the case, then for those pixels the gradient will be zero. If we sum up over these pixels with the value zero and divide the resulting sum by n, the final similarity value will never reach the value 1. The similarity value will differ from the value of 1, depending on the number of zero values in the case. That means that we have to ensure that our case never includes homogenous or nearly homogenous areas.

5.2.3. Contextual similarity measure

The above described local similarity measures do not consider the spatial context of the direction changes. In case of touching objects with different grey-level values, the vectors have inversed directions. If we want these local deviations also increase the similarity, we have to use the absolute similarity value. Note, that in this case also arbitrary changes increase the similarity.

Therefore a contextual similarity measure could be introduced that counts the preceding direction changes and detects arbitrary direction changes based on the number of preceding direction changes. Then a criterion is introduced that tells us if these changes appear along touching objects. For this the mean and the standard deviation of the grey-level in the surroundings of the contour where the direction changes appear are calculated. If the values for the mean and the standard deviation correspond to those of the inner area of an object, then the local similarity values are accepted and they contribute to the global similarity.

5.3. Occlusion and touching of objects

Note that the global similarity measure also indicates the amount of occlusion: If some parts of an object are occluded, then the direction vectors of the case and the image will differ. This leads to a smaller value of the similarity measure. Thus the similarity score decreases as much as the case instance in the image is occluded. For example, if there is no noise in the image and an object in the image is occluded by 30% the similarity score cannot exceed the value 0.7.

In case two objects in the image touch each other, either the contrast of the case and the image objects can be inversed in the corresponding regions or there are no edges which match with case edges. The similarity measure will be decreased if we use s_2. If we apply the similarity measure s_4 local contrast changes are ignored.

6. Hypothesis-test verification

6.1. Related work

The aim of the hypothesis verification is to decide whether a match can be accepted as being correct or not. In the literature we can find different approaches for this process. Grimson and Huttenlocher [7] as well as Jurie [8] and Katartzis et al. [9] refer to features in the form of points or line segments. The common target of all these works is to find the best pose for the detected data features. However, all papers follow different strategies: Grimson and Huttenlocher [7] focus on the question how random matches can be prevented. They develop a formal means for finding the fraction of model features that have to be evaluated in order to ensure that the match occurs only with a given probability at random. The derivation of this fraction is done in three steps whereas the type of feature, the type of transformation from model to image and a bound on the positional and orientational error are known. First for every pairing of a model feature to data feature the set of transformations is determined. This set defines a particular volume in the transformation space. In the next step the probability of a common point of intersection between l or more volumes is calculated. This probability corresponds to a match of at least l pairings of model and image features. Last a second probability that describes that l or more volumes will intersect at random, is used to specify a threshold for the fraction of model features that have to be evaluated at least in order to ensure that the probability of a random match is lower than a given value.

The aim of the research of Jurie [8] is to find the pose of the model features that best matches the data features. The pose hypotheses are generated by correspondences between the model and the data features. Early researches propose to evaluate only some correspondences in order to find an initial pose hypothesis P that is refined by iteratively enlarging the number of correspondences. Jurie [8] describes that this way of hypothesis generation and verification is not optimal. Therefore, the paper suggests the opposite approach: A pose space is generated from different model-data-pairings. A 'box' of the pose space is computed including the initial position P that is large enough to compensate the data errors. Assuming that the distribution of model-data-correspondences is Gaussian, the maximal probability of the object to be matched is determined. Then the box can be refined. The process repeats until the 'box' only contains one pose.

A simpler method of model-based pose estimation and verification is described by Shahrokni et al. [10]. They deal with the automatic detection of polyhedral objects. Hypotheses are generated by the knowledge-based connection of corners and line segments. The model and the transformed hypotheses are evaluated with the method of the least squares. The best hypothesis minimizes the sum of squared differences between the model and the transformed hypothesis.

Katartzis et al. [9] discuss the automatic recognition of rooftops, which are characterized by lines and their connections. After detecting the line segments in the image, they are grouped in a hierarchical graph. The highest level of the hierarchy contains closed contours. Every node of the graph is assigned a value that on the one hand assigns the saliency of the hypothesis and on the other hand represents the likelihood of the presence of a 3D-structure, which depends on domain-specific knowledge. Based on the hierarchical graph a Markov random field (MRF) is defined. By maximization of the posterior probability of the MRF for the concrete graph a consistent configuration of the data is found.

In general the verification process for object hypotheses based on line segments is a widely discussed field.

An approach that differs totally from the discussed ones is given in Leibe et al. [11]. The heart of the described object recognition system is a database with different appearances of parts of the object that should be recognized. Additionally an 'Implicit Shape Model' is learnt in order to combine the parts to a correct object. If multiple objects are located in the image, then some hypotheses may overlap each other so that a verification step is required. The method follows the principle of minimal description length (MDL) that is borrowed from the information theory. The description length of a hypothesis depends on its area and the probability that the pixels inside the hypothesis are no object pixels. The description length of two overlapping hypotheses is generated in the same way. From the resulting values it is concluded whether two overlapping hypotheses refer to two objects or only to one.

6.2. Problem of random matches and our strategy for hypothesis verification

The matching process determines each possible match in the image. In the following we consider the found match as a hypothesis. To each hypothesis a matching score is assigned based on the similarity measure. It can range from 0 to 1, whereas the value 1 says identity and the value 0 says dissimilarity. By defining a threshold for the similarity value, we can exclude hypotheses. This is the simplest hypothesis verification process. However, there are still some hypotheses that remain in the image (see Table 5c).

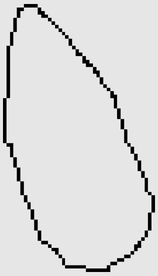
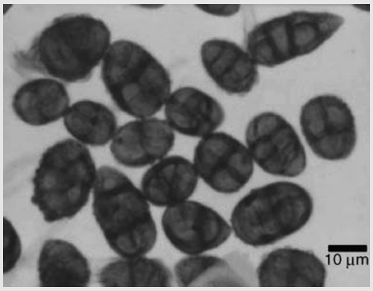
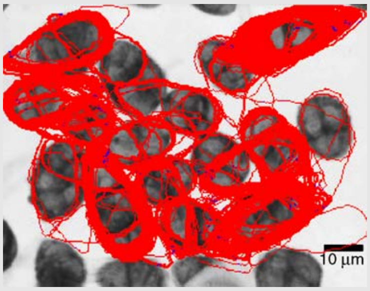
We can see in Table 5c that the case often matches the same object but the hypothesis is slightly transformed (shifted or twisted) with respect to other hypotheses. All these hypotheses in this image have scores greater than 0.8. Now we need to find a rule which allows us to remove false hypotheses. The spatial relation of the hypotheses to each other gives a rule for the hypothesis verification process. Let A and B be two hypothesized objects. The set $S(A)$ is the set of all image pixels that are inside the contour of object A, including also the image pixels of the contour. Equally, $S(B)$ is the set of all image pixels inside the contour of object B, including all image pixels of the contour. Now we can define two spatial relationships:

1. We say that the set $S(B)$ is inside the set $S(A)$ if all elements that are included in the set $S(B)$ are also included in the set $S(A)$, i.e. $S(B) \subset S(A)$ and there is $S(A) \cap S(B) = B$ or $S(A) \cup S(B) = S(A)$.
2. We say that the sets $S(A)$ and $S(B)$ overlap each other if they have some equal elements, i.e. $S(A) \cap S(B) \neq \emptyset$, $S(A) \cap S(B) \neq S(A)$ and $S(A) \cap S(B) \neq S(B)$.

Hypotheses that fulfill the described spatial relation are compared by their similarity value. The hypothesis which has the lower similarity value is removed.

Besides the similarity value there are some other measures that may be used to decide which hypothesis should be removed. For example, we can determine the variation of the gradient of the matched image pixels. The variation is defined as the standard deviation normalized by the mean. Assuming that the gradient of the matched image pixels

Table 5 Contour case, original image and hypothesized objects

(a) Case	(b) Original image	(c) Hypothesized objects
		

should be as homogenous as possible, the hypothesis is removed that has the higher variation.

7. Implementation in the software package CaseRec

We developed the case-based object recognition system CaseRec which is based on the similarity measure described in Section 5. The following subsections present an overview of how the case-based object recognition was realized if a contour case and an image are given. The case base is filled up with cases obtained by our program CACM described in the next section. Note that the case description is only comprised of the contour point coordinates, the case ID and that the grey values of the contour points are fixed to the value one.

7.1. Determining the low level feature representations

For the image and the case pixels the direction vectors have to be calculated. We use the Sobel operator to transform the image and the case into the necessary representation. Since the calculation is based on the 8-neighboorhood of a pixel, we exclude the margin pixels and set the directions of their vectors to value zero.

Each case is a data file which contains the case ID and the coordinates of the contour pixels. In order to determine the corresponding direction vectors we create a binary case image based on this set of pixels. The case image consists of white background and black object pixels. The object pixels are defined as all pixels of the area inside the contour and the contour pixels itself. Note that also the area inside the object has black object pixels. Since our matching is only based on the contour, we only calculate the direction vectors along the given object contour. Because the underlying image is binary, there is no noise in the direction vector and therefore the vectors are orthogonal to the edge.

7.2. Translation of the case

Usually the area which includes the contour is small compared to the image matrix. Therefore we only consider a part of the image, called *matching window*, which has the size of the surrounding box of the contour area. The similarity measure S is calculated between the contour and the corresponding image pixels inside this matching window.

During the matching process the matching window is pixel-wise shifted over the whole image. Let the centre of the contour be the origin of the contour, then $q = (x, y)$ denotes the image pixel, which is equal to the origin of the contour. In each translation q the similarity value $S(q)$ between the case and the matching window is calculated. Note that there will be a border effect where the case can only be partially matched against the image. In the recent version of our program we match the case only on that image positions where the window fully fits to the image.

7.3. Speeding up the computation time

The object recognition process includes three different transformations of the case image against the input image: translation, rotation, and scaling. Given the range of rotation and scaling we have to determine the similarity of each pixel on each possible rotation and scale. This requires much time and therefore we included two procedures into our matching process to speed up the computation time. The current method of choice is to use image pyramids and to introduce a minimal value of similarity.

7.3.1. Image pyramids

Image pyramids are applied for various tasks in image processing [12]. Each pyramid level concludes the information of the next lowest level, beginning with the original image on the bottom. The image generation in each level combines a decimation filter to reduce the image extensions with a function that assigns the grey values to the pixels. In general the image width and the image length are halved level by level. Thus the image of the new level only contains a quarter of the pixels of the image it is generated from. The corresponding pixel to (x, y) in the generated level is defined as the pixel $(2x, 2y)$ in the basic image.

In Gaussian pyramids the weighted average of a 5×5 window around the pixels in the basic level is taken [12]. The nearer a pixel is to the centre of the 5×5 window, the more its grey value influences the new grey value. The Laplacian pyramid [12] is based on the Gaussian pyramid: The images of two adjacent levels are subtracted after the image of the higher level is expanded to the size of the image of the lower level.

We implemented a simple and fast pyramid generation method to our system: The image is separated into 2×2-sized blocks. Each block represents a pixel in the next resolution level and the corresponding grey value is determined by averaging all grey values of the block.

The number of levels is restricted by the demand that the main image contents in the different pyramid levels must remain. The matching procedure will be as faster as the image size is lower. Note that you can only match cases and images with the same resolution, i.e. the same pyramid level. After matching the case against the image in the highest pyramid level, the approximate position, scale, and orientation of each object in the image will be known. So in the next levels the search is focused on the regions around these objects.

7.3.2. Minimal value of similarity

To speed up the calculation time for the whole image we introduce a termination criterion for each iteratively carried out matching process. We demand that the similarity between the case and the image in the actual matching window must reach a predefined minimal value $s_{\min}$ after having inspected j case points. If this is not the case, the matching process is terminated in the actual matching window

$$s_j < s_{\min} - 1 + \frac{j}{n} \Rightarrow \text{break} \qquad (12)$$

where

$$s_j = \frac{1}{n} \sum_{k=1}^{j} \frac{\vec{m}_k \cdot \vec{i}_k}{||\vec{m}_k|| \cdot |||\vec{i}_k||} \qquad (13)$$

7.4. Rotation and scaling

With the Eqs. (1) and (2) we presented a possibility to scale and rotate a given case. Applying these formulas to the pixel positions of the case lead to floating point coordinates. Since rounding of the coordinates sums up to a high error in pixel positions, we divided the transformation procedure into two parts: First we determined the new pixel positions by selecting some representative contour pixels from the whole set. After transforming these representatives with Eq. (1), we interpolated between the pixels by using a first order polynom. In the second part the corresponding direction vectors were created as described in Section 7.1.

8. Case acquisition

8.1. Related work

The acquisition of object shapes from real images is still an essential problem of image segmentation. For automated image segmentation often low-level methods, such as edge detection [13] and region growing [14,15] are used to extract the outline of objects from an image. Low-level methods yield good results if the objects have conspicuous boundaries and are not occluded. In the case of complex backgrounds and occluded or noisy objects, the shape acquisition may result in strong distorted and incorrect cases.

Therefore, segmentation is often performed manually at the cost of a very subjective, time-consuming procedure. Landmark coordinates [16—19] can be assigned by an expert to some biologically significant points of an organism. If there are objects with an absence of anatomical landmarks, it is a common procedure to determine landmark points according to the defined mathematical or geometrical properties of the objects. However, in some applications it is impossible or insufficient to describe the shape of an object only by means of these landmarks because important characteristics of the shape might be lost. To increase the total of landmarks it is usual to trace the complete outline of an object manually and subsequently determine corresponding points on each shape [20]. New semi-automatic approaches were developed [21,22] for interactive image segmentation. These approaches use live-wire segmentation algorithms which are based on a graph search to locate mathematically optimal boundaries in the image. If the user moves the mouse cursor in the proximity of an object edge, the labeled outline is automatically adjusted to the boundary.

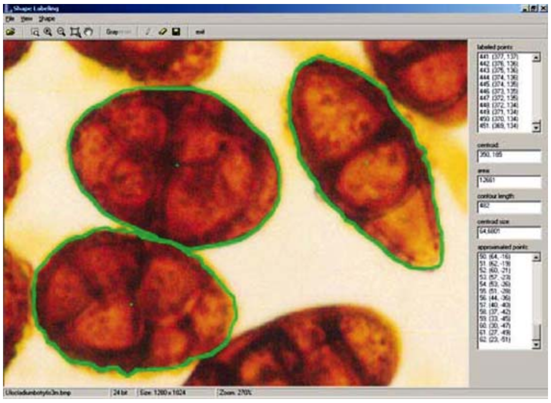
Based on an acquired set of shape instances it is usually desirable to describe and compare deformations and distances between these shapes. The problems of shape spaces and distances have been intensively studied by Kendall [19] and Bookstein [16] in a statistical theory of shape. All of them assume that point correspondences between two sets of landmarks are already known. However, at the beginning of many applications this condition is not held and various approaches are made to determine corresponding points for the automated generation of statistical shape models. Hill et al. [20] presented an interesting framework for the automated landmark identification on a set of two-dimensional, polygonal shapes. They assume that all acquired shapes are similar so that, proportionately, the arc path-lengths between consecutive points are equal. Bookstein [23] applied landmark methods to continuous contours represented as thin-plate splines, but his approach is not completely automatic. The softassign Procrustes matching algorithm [13] solves the correspondence problem using deterministic annealing. This algorithm works robustly with respect to outlier identification and noise, but it is also a computationally expensive procedure. Belongie et al. [24] found correspondences between points based on the shape context descriptor. Latecki et al. [6] used a tangent space representation of shapes to determine correspondences of visually significant parts and to define a shape similarity measure. Another interesting method was presented by Mokhtarian et al. [25] who calculate a similarity measure between two exemplars based on their maxima in the curvature scale-space. Nevertheless, especially in cases of noisy or distorted cases the analysis of these feature-based shape representations is problematic.

8.2. Acquisition of shape cases

Case-based object recognition can be done based on the case model or based on the case contour. 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 object shape S_C represented by a set of n_{S_C} boundary points $s_i(x, y)$, $i = 1, 2, \ldots, n_{S_C}$.

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. Notice that the sampled points are not required to be landmark coordinates or curvature extremes. The user starts labeling the contour S_C of object S at an arbitrary pixel $s_i(x, y)$, $i = 1, 2, \ldots, j, \ldots, n_{S_C}$ of the contour S_C. After having traced the complete object the labeling ends at a pixel $s_j(x, y)$ in the neighbourhood of $s_i(x, y)$. To obtain the complete set S_C of all boundary pixels we need to ensure that the contour is closed which means $s_j(x, y)$ is a direct neighbor of $s_i(x, y)$. Therefore, we insert missing boundary pixels using the Bresenham's [26] procedure. Fig. 3 presents a screenshot from our program CACM with three labeled shapes of the strain Ulocladium Botrytis with their coordinates on the right side of the screenshot.

It might be very difficult to exactly determine and meet every boundary pixel of an object when manually labeling the contour of an object. The quantization of a continuous image constitutes a reduction in resolution which causes considerable image distortion (Moiré effect). Furthermore the contour of an object in a digitized image may be blurred which means the contour is extended over a set of pixels with decreasing grey values. In fact, image digitization and human imprecision always implies small error rates in the object shapes. Therefore, in the next step our intention is to introduce into the program procedures that help the user to find the right boundary of an object and that speed up the labeling process.



labeled points:
441. (377, 137)
442. (376, 136)
443. (375, 136)
444. (374, 136)
445. (374, 135)
446. (373, 135)
447. (372, 135)
448. (372, 134)
449. (371, 134)
450. (370, 134)
451. (369, 134)

centroid: 350, 185

area: 12661

contour length: 482

centroid size: 64,6601

approximated points:
50. (64, -16)
51. (62, -19)
52. (60, -21)
53. (57, -23)
54. (53, -26)
55. (51, -28)
56. (44, -36)
57. (40, -40)
58. (37, -42)
59. (33, -45)
60. (30, -47)
61. (27, -49)
62. (23, -51)

Ulocladiumbotrytis3m.bmp | 24 bit | Size: 1280 x 1024 | Zoom: 270%

Figure 3 Labeled and approximated shapes with coordinates.

As a result of the labeling process we obtain the set S_C of n_{S_C} connected points that describes the boundary of the object S. Having labeled the contour S_C of the object S its boundary pixels are still defined by their absolute position in the 2D matrix of the original image. In order to describe and compare the shapes of objects it is useful to specify a common coordinate system that is invariant under translation and scale. Therefore, we transform the contour such that the centroid is at the origin and the maximal distance of the contour points from the origin is one. In a following approximation of the contour we reduce this set of pixels to a sufficiently large number of pixels that will speed up the succeeding computation time of the alignment and clustering process. The number of the pixels in this set will be influenced by the chosen order of the polygon and the allowed approximation error. For the polygonal approximation we used the approach based on the area/length ratio of Wall and Daniellson [27], because it is a very fast and simple algorithm without time-consuming mathematical operations.

8.3. Shape alignment and similarity calculation

8.3.1. 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 $p_i \in R^2$, $i = 1, 2, \ldots, N_1$ and $o_j \in R^2$, $j = 1, 2, \ldots, N_2$, 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 $P = \{p_i(x, y)\}_{i=1,\ldots,N_1}$ is said to be aligned to the shape instance $O = \{o_j(x, y)\}_{j=1,\ldots,N_2}$ if a distance $d_{\min}(P, O)$ between the two shapes cannot be decreased by applying a transformation ψ to P. Various alignment approaches are known [4,13,28,29]. They differ in the kind of mapping (similarity [13], rigid [30], and affine [20] and the chosen similarity measure [31]). For the similarity measure between P and O we use the Procrustes distance [32]:

$$d(P,O)=\sum_{i=1}^{N_{PO}}\left\|\frac{(p_i-\mu_P)}{\sigma_P}-R(\theta)\frac{(o_i-\mu_O)}{\sigma_O}\right\|^2 \qquad (14)$$

where $R(\theta)$ is the rotation matrix, μ_P and μ_O are the centroids of the objects P and O respectively, σ_P and σ_O are the sums of squared distances of each point-set from the centroids and N_{PO} is the number of point correspondences between the point-sets P and O. Thus, the point correspondences are required for calculating the Procrustes distance. Generally this method is applied to centered shape instances represented by sets of landmark coordinates. Each of these shapes is rescaled, so that the sum of squared distances of all landmarks to the centroid are identical ($\sigma_P = \sigma_O$). Then it is possible to compute a similarity transformation based on these centered pre-shapes. Finally, the Procrustes average shape and the Procrustes residuals can be evaluated.

8.3.2. Our approach to shape alignment

We are considering a set of shape instances where differences in the translation and the scale were already eliminated. To compare the shape of these two instances we still have to eliminate differences in the rotation. The measure of similarity is based on the Procrustes distance between all points of P and their correspondences in O. As it can be seen from Eq. (14) 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 labeled 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.
2. The point correspondences between the two shape instances P and O are completely unknown.
3. We have no information about point outliers.

The outline of our approach to shape alignment is as follows: For every pair of points $\{p_i, o_j\} \in P \times O$ we calculate the similarity transformation ψ_{ij} that aligns these two points. 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 $p'_k \in R^2$, $k = 1, 2, \ldots, N_1$. For every point p'_k we define the nearest neighbor $\mathrm{NN}(p'_k)$ in O as the point correspondence of p'_k. Note that we do not enforce one-to-one point correspondences. One point in O can have more than one point correspondences or even not one point correspondence in P. The sum of squared distances $d(P', O)$ between every pair of point correspondents is calculated. In addition to that we define the quantity $\sqrt{\frac{1}{N_1}d(P',O)}$ as the mean alignment error $\bar{\varepsilon}(P',O)$:

$$\bar{\varepsilon}(P',O)=\sqrt{\frac{1}{N_1}d(P',O)} \qquad (15)$$

with

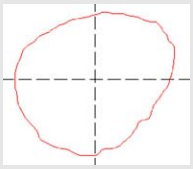
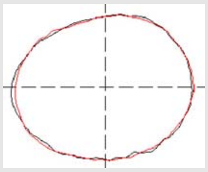
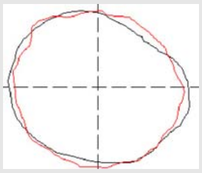
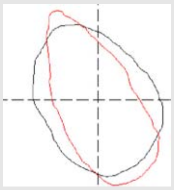
$$d(P',O)=\sum_{k=1}^{N_1}(p'_k-\mathrm{NN}(p'_k))^2 \qquad (16)$$

If the distance $d(P', O)$ is smaller than all earlier calculated distances, $d_{\min}(P, O)$ is set to $d(P', O)$, $\bar{\varepsilon}_{\min}(P,O)$ is set to $\bar{\varepsilon}(P',O)$ and $\psi_{\min}$ is set to ψ_{ij}. After having iteratively aligned every possible pair of points $\{p_i, o_j\} \in P \times O$, we may estimate the dissimilarity between the shape instances P and O based on the value of the minimum mean alignment error $\bar{\varepsilon}_{\min}(P,O)$. To ensure that our final measure of dissimilarity ranges from 0 to 1 we normalize the measure $\bar{\varepsilon}_{\min}(P,O)$ to a predefined maximum distance T:

$$\bar{\varepsilon}'_{\min}(P,O)=\frac{\bar{\varepsilon}_{\min}(P,O)}{T} \qquad (17)$$

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

Table 6 Aligned shape instances of strain Ulocladium Botrytis with distances

			
$\bar{\varepsilon}'_{\min}(P,O)=0$ [identical]	$\bar{\varepsilon}'_{\min}(P,O)=0.027$ [nearly identical]	$\bar{\varepsilon}'_{\min}(P,O)=0.199$ [similar]	$\bar{\varepsilon}'_{\min}(P,O)=0.5$ [neutral]

It is obvious that the constant T has a direct influence on the value of the resulting score. The parameter T can be defined by the user in the setting dialog of our program CACM. For our calculations we set T to 35% of the mean distance of all contour points to the centroid. Our investigations showed that this value leads to good results. Table 6 shows pair-wise aligned shape instances and calculated values of the dissimilarity measure. It can be seen that in case of identity the shapes are superposed. With an increasing value of dissimilarity we can see the increasing deviation of the two shapes.

8.3.3. Clustering

The alignment of every possible pair of objects in our database leads us to $N \times N$ pair-wise dissimilarity measures between N cases. These distances can be collected in a matrix where each row and each column corresponds to an instance of our dataset. The dissimilarity measure $\bar{\varepsilon}'_{\text{min}}(P, O)$ between shape instance P and shape instance O will be entered into the cell where the row of P and the column of O intersect. This results in a squared symmetric matrix with diagonal elements equal to value zero, since the dissimilarity between an instance and itself is zero.

This matrix is the input for the hierarchical cluster analysis [33]. It depends on the selected clustering method how the instances are merged together into groups. After having investigated different hierarchical clustering methods we chose single linkage (nearest neighbor) where the linkage is done at the minimum distance between the two most similar cases of two different clusters.

The result of the hierarchical cluster analysis can be represented graphically by a dendrogram. The dendrogram is drawn on a proximity scale to show the clustering and the proximities at which the cases are merged together. The merging is done with increasing distances until all cases are combined in only one cluster. The distance at which two cases are merged in the same cluster is called the cophenetic proximity measure [34]. Note that the cophenetic proximity measure $d_C(P, O)$ between two cases P and O is not identical with the dissimilarity measure $\bar{\varepsilon}'_{\text{min}}(P, O)$ between these two cases.

Table 7 Merging steps with cophenetic proximity measure

Merging step	Case 1	Case 2	Cophenetic proximity measure
1	ub_6	ub_7	0.0110
2	ub_5	ub_7	0.0162
3	ub_2	ub_4	0.0500
4	ub_3	ub_6	0.0579
5	ub_4	ub_5	0.0583
6	ub_1	ub_8	0.2489
7	ub_8	ub_2	0.2634

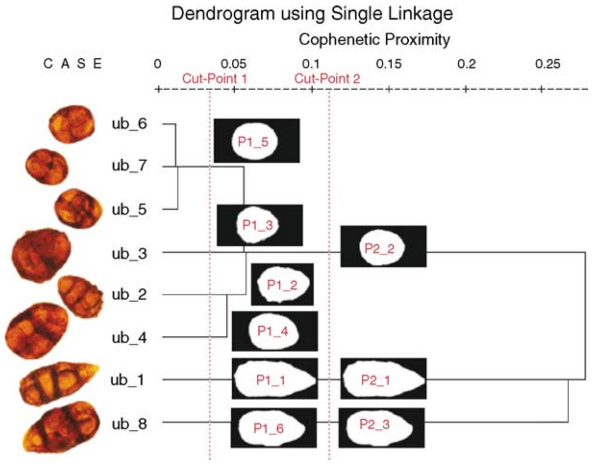


Figure 4 Dendrogram of eight instances of strain Ulocladium Botrytis using single linkage.

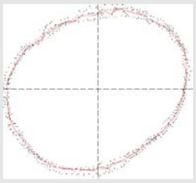
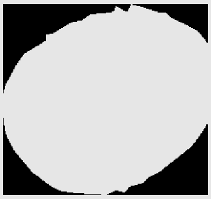
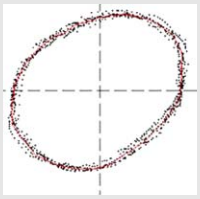
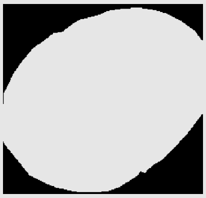
Table 7 presents the merging steps and the cophenetic proximity measures for clustering eight instances of strain Ulocladium Botrytis using single linkage.

The graphical output of this table is presented in the dendrogram in Fig. 4. In the dendrogram in Fig. 4 we marked two exemplary cut-points at different distances. A cut-point is a virtual vertical line in the dendrogram. The horizontal position of this line marks the cophenetic proximity measure at which the cases were split into several clusters. The level of the cut-point has a direct influence on the resulting number of clusters and the number of prototypes. The smaller the level is the more groups are created. That means the prototypes are more specialized which will result in matches with higher scores. The higher the chosen cophenetic proximity measure of a cut-point is, the fewer cases have to be created. The cases are more general and the innerclass variance inside a cluster is higher. In general, this will result in matches with lower scores.

8.3.4. Prototype calculation

We have divided our set of N shape instances $\{P_1, P_2, \ldots, P_N\}$ into k clusters $C_1, C_2, \ldots, C_k$. Each cluster C_i, $i = 1, 2, \ldots, k$ consists of a subset of n_i shape cases. For each cluster we need to compute a prototype $\bar{\mu}$ that will be the representative of the cluster. The full Procrustes estimate [18] of the mean shape is obtained by minimizing (over $\bar{\mu}$) the sum of the squared full Procrustes distances d_f^2 from each

Table 8 Aligned shapes and mean shape of a single cluster before and after rejecting outlier

(a) Alignment of all objects in a cluster	(b) Resulting synthetic mean prototype	(c) Alignment of all objects in a cluster after rejecting outlier	(d) Resulting synthetic mean prototype after
			

instance P_j, $j = 1, 2, \ldots, n_i$ to an unknown unit-size mean configuration $\bar{\mu}$, i.e.

$$[\bar{\mu}] = \arg \inf_{\bar{\mu}} \sum_{j=1}^{n_i} d_f(P_j, \bar{\mu})^2 \tag{18}$$

We have implemented and tested two different approaches of computing a prototype for the cluster. In the first approach we compared the similarity measures of all shape instances $\{P_1, P_2, \ldots, P_j\}$ included in the same cluster C_i, $i = 1, 2, \ldots, k$. As prototype we chose the median shape $P_{\text{median}}(C_i)$ of that cluster which is the shape instance that has the minimum distance to all other shape instances:

$$[\bar{\mu}] = P_{\text{median}}(C_i) = \arg\inf \sum_{j=1}^{n_i} d(P_j, P_{\text{median}})^2 \tag{19}$$

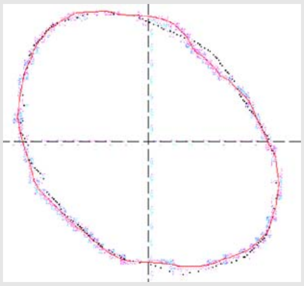
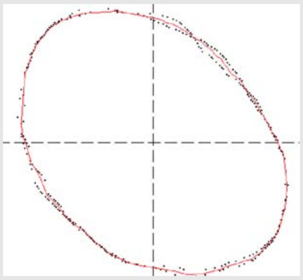
The main advantage of this solution is that the prototype represents a natural shape instance out of that cluster. An example of using a natural shape instance as prototype of a cluster is shown in Table 9a.

Another approach for the calculation of the prototype is to calculate the arithmetic mean of the shape instances of a cluster. Suppose a cluster C_i including a set of n_i shape instances $\{P_1, P_2, \ldots, P_{n_i}\}$. In the best case every point $p \in R^2$ of a shape instance $P_j \in C_i$ has exactly $n_i - 1$ corresponding points, one point on each single shape instance of that cluster. As described before we cannot guarantee a one-to-one point mapping. We are confronted with one-to-n correspondences where each point $p_m \in R^2$, $m = 1, 2, \ldots, N_1$ of P_j can have zero, one or even more then n_i point correspondences. In worst cases this may result in strong distortions in the prototype contour (see Table 8b).

To improve the results of the calculations for the arithmetic mean of corresponding points we reject all contour points as outliers of the prototype where at least 80% of all individuals of that cluster do not have a point correspondence. In general, this approach leads to a better representation of the mean contour for all instances in one cluster. The comparison of Table 8b and d shows that the sharp edges shown in Table 8b are eliminated after rejecting the outlier.

Table 9a shows the median shape and Table 9b shows the arithmetic mean as two different kinds of prototypes representing a cluster which consists of only two shape instances. The prototype in Table 9a is identical with one instance out of that cluster. In comparison the prototype in Table 9b is not a natural shape instance but it is a better representative for both members in that cluster. In addition to that we would favor the mean shape as prototype for the cluster since it appears visually smoother. With our program CACM it is possible to calculate both types of prototypes—the median of shapes and the arithmetic mean. Each prototype can be exported as an image, where the contour pixels are labeled by the grey-level one and the background pixels are labeled by zero, and as a list of coordinates of the contour points.

Table 9 Median of shapes in a cluster and arithmetic mean of the shapes

(a) Median shape instance as prototype	(b) Using the arithmetic mean as prototype
	

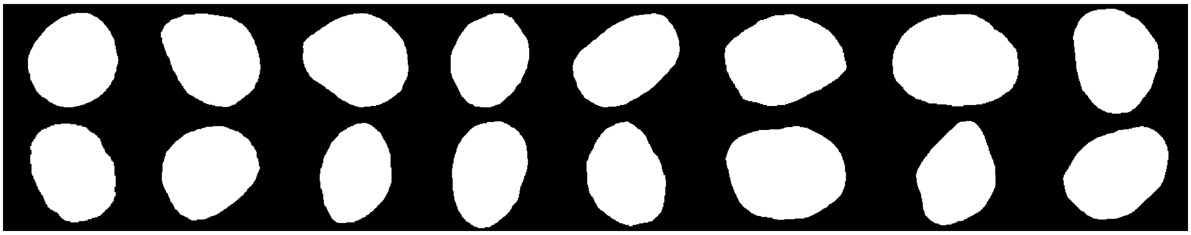


Figure 5 Case base of cases for strain Rhizopus Stolonifer representing the 16 resulting clusters.

8.3.5. The acquired cases

We have tested our approach on six different airborne fungi spores. Some digitized sample images for the analyzed spores are presented in Table 1. We have labeled a total of 60 objects for each of the six fungi strains. In the following registration process we have aligned every single object with all objects of the same strain to calculate the pair-wise similarity values. As a result we obtained six similarity matrices, one for each analyzed fungi strain. These matrices are the input for the following hierarchical cluster analysis. The outcome of this process was a dendrogram for each of the six fungi strains.

Now we need to define a cut-point on each dendrogram to obtain the groups of the shapes. As described earlier the choice of the right position for a cut-point is a central issue and it directly influences the recognition rate and performance. Visually the dendrograms impart only some limited information about the best position of the cut-point. This is one of the main disadvantages of classical hierarchical clustering methods. The algorithms only create clusters but do not explain why they were established. It is not possible to draw direct conclusions about the quality of the clusters from the classification hierarchy.

If it is possible to specify the number of clusters in advance, we can calculate the position of the cut-point d_w by taking the minimal and the maximal distance d_{min} and d_{max} in each cluster

$$d_w = \frac{d_{max} - d_{min}}{k} \tag{20}$$

where k is the maximal possible number of clusters.

In our application we can give no detailed information about the expected number of resulting clusters. Therefore, we started our investigations at two different cut-points for each class. Subsequently we calculated the corresponding set of cases for both cluster partitions. Table 10 presents the resulting number of cases for each class at these cut-points.

Both case bases were applied to our case-based object recognition system (see Section 7). We decided to split the dendrograms at the distances of cut-point (2) because it results in the best recognition rates. The cases from these clusters were incorporated as cases into the case base. Fig. 5 shows exemplary the resulting case base for the class Rhizopus Stolonifer on cut-point (2).

In fact, we know little about the accuracy of these cases now. Therefore, in the next step our intention is to introduce into the program procedures that improve the information gathering procedure and to develop an incremental clustering procedure [33].

9. Results

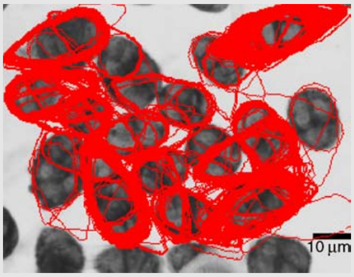
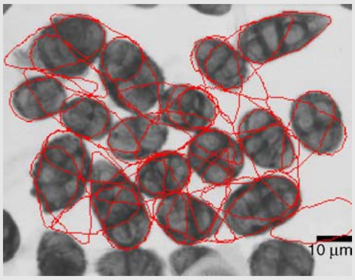
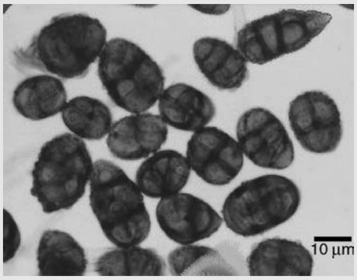
9.1. Results of hypothesis-test verification

We tested our verification algorithm for each spatial relation and counted the number of hypotheses that were removed. In Table 11a you can see the initial hypotheses that have to be verified. In case of the spatial relation ‘inside’ the reduction of the hypotheses was only 10%. We could achieve the best results for the spatial relation ‘overlapping’ by requiring that the areas of the two hypotheses overlap by

Table 10 Number of cases on two different cut-points

Classes	d_w (1)	Number of cases	d_w (2)	Number of cases
Alternaria Alternata	0.045	23	0.031	34
Aspergillus Niger	0.017	5	0.029	3
Rhizopus Stolonifer	0.027	22	0.037	16
Scopulariopsis Brevicaulis	0.063	8	0.050	10
Ulocladium Botrytis	0.025	24	0.020	30
Wallenia Sebi	0.158	7	0.065	17

Table 11 Hypothesis-test verification

(a) Initial hypotheses	(b) Remaining hypotheses after hypothesis-test verification	(c) Hypotheses with a threshold for the minimal gradient
		

more than 50%. In case of such an overlap the hypothesis that has the lower similarity value was rejected. We could achieve a reduction of the hypotheses of 95% (see Table 11b). For all the following tests we applied our hypothesis-test verification algorithm based on the overlap rule.

9.2. Evaluation of the similarity measures and determination of the minimal gradient

From Table 11b we can see that there are a lot of background matches. In order to prevent false matches in the image background, we test the similarity measure with a threshold for the minimal gradient. We determined the minimal gradient by calculating the mean and the standard deviation of the gradient from six background images.

As you can see from Table 11c we get only the true hypothesis if the threshold for the minimal gradient takes on the value 24.53. In order to investigate more intensively in the threshold of the minimal gradient, we carried out twelve experiments where each of the similarity measures is combined with one of four different thresholds for the minimal gradient. Our test image is shown in Table 12a and in Table 12b is shown the gradient image. All objects in this image are numbered. The dendrogram has been calculated based on the similarity between these objects (see Table 12c). We chose the case {ub_5} as matching case. Our aim was to determine which objects we can recognize with this case and how good the match is.

The test case {ub_5} represents the group {ub_3, ub_6, ub_7, ub_11, ub_12, ub_14} in the dendrogram and at a distance of 0.064. The other cases {ub_2, ub_4, ub_8, ub_9, ub_10, ub_13, ub_15, ub_16,} are grouped together at a distance of 0.0772. The two cases {ub_10} and {ub_1} are separated from the others based on the distance value 0.11.

Table 12 Test image and dendrogram of the stain Ulocladium Botrytis

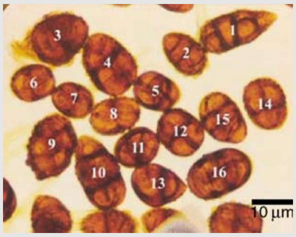
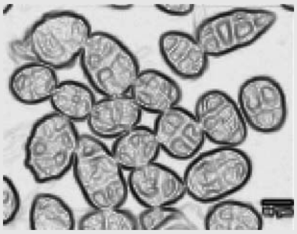
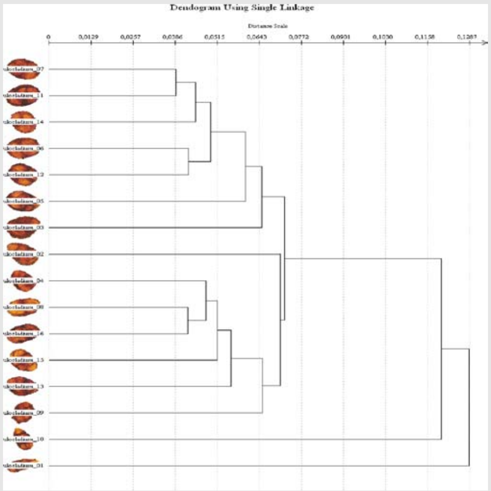
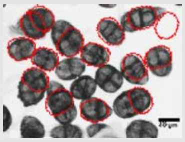
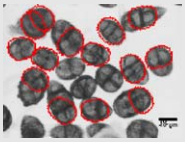
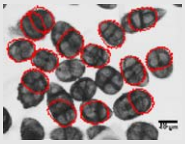
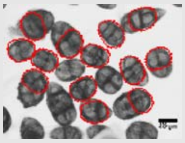
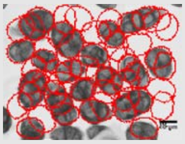
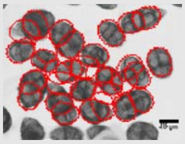
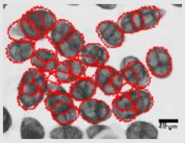
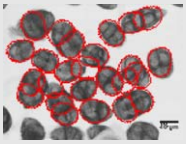
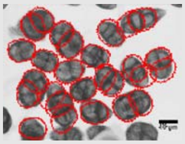
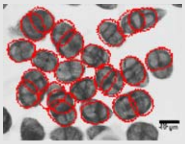
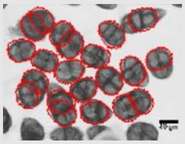
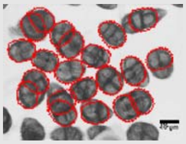
(a) Test image including the object numbers	(b) Gradient representation of the test image	(c) Dendrogram of the objects in the test image
		

Table 13 Comparison of the resulting matches by applying different variations of the similarity measure and demanding different minimal gradients

Similarity measure	Minimal gradient $l_{\min}$			
	(A) 0	(B) $\hat{\mu} = 11.01$	(C) $\hat{\mu} + \hat{\sigma} = 17.77$	(D) $\hat{\mu} + 2 \cdot \hat{\sigma} = 24.53$
$s_3 = \left\| \frac{1}{n} \sum_{k=1}^{n} \frac{\vec{m}_k \cdot \vec{i}_k}{\|\|\vec{m}_k\|\| \cdot \|\|\vec{i}_k\|\|} \right\|$				
$s_4 = \frac{1}{n} \sum_{k=1}^{n} \left\| \frac{\vec{m}_k \cdot \vec{i}_k}{\|\|\vec{m}_k\|\| \cdot \|\|\vec{i}_k\|\|} \right\|$				
$s_5 = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{2} \left(1 - \frac{\vec{m}_k \cdot \vec{i}_k}{\|\|\vec{m}_k\|\| \cdot \|\|\vec{i}_k\|\|} \right)$				

We carried out the experiments by using a four level pyramid of the image. The case is transformed in steps of 1% from 95 up to 105% of its original size. Each scaled case is rotated from 0° to 360° using 1° steps. The transformed pixels (i.e. scaled and rotated coordinates) are rounded to integers. Each transformed case is translated from the left to the right and from the top to the bottom of the image. The minimal matching score of an object is set to the value 0.8. The hypothesis-test strategy is based on the overlap rule. In Table 13 the detected objects of each experiment are labeled in the test image.

From Table 13 we can see that the minimal gradient has a strong influence on the number of the detected objects (compare column A with column B). Background matches are clearly eliminated by introducing a threshold for the minimal gradient. As we can see from the last column of Table 13 also some object matches are missing.

The chosen similarity measure has a strong influence on the results. While too many objects are detected by applying the similarity measure s_4 which does not consider the direction of the vector by using the absolute value of the dot product, there are too few objects detected by applying the measure s_3. We obtain the most consistent results by applying the last similarity measure s_5. However, there are too many matches inside an object which are not wanted.

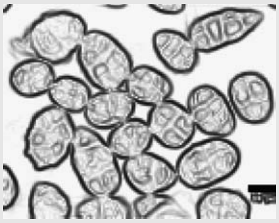
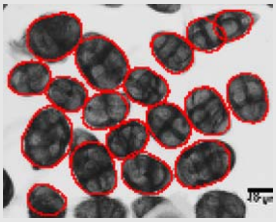
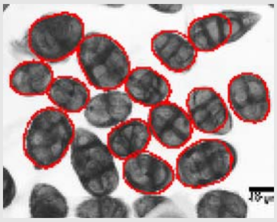
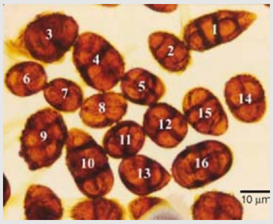
As a conclusion we can say that only considering the angle between two vectors is not sufficient in order to recognize the objects correctly, in particular if the objects are highly structured. A threshold for the minimal value of the gradient has to be introduced. It can be necessary to introduce a threshold or to require that the value of the gradient lies in a particular range depending on the application. If this is not done too many mismatches may occur which make the proceeding interpretation difficult.

The threshold of the similarity value corresponds to the variation among the shapes of a group of instances. Not every object that has a proximity over 0.1 (see Table 12c) could be detected by the matcher based on the chosen case. This result shows the correspondence between the similarity measure used for separating the set of input cases into groups and the threshold of the similarity value for the matching.

9.3. Evaluation of the similarity measure and contour gradient

In Section 9.1 we could see that the introduction of a threshold for the minimal gradient is a necessity if background matches are to be avoided. However, since the objects are highly structured, the matcher matches also along edges inside the objects. We can see from Table 14a that the edges in the objects have a lower gradient than the outer contour of the object. Therefore we estimated the mean and the standard deviation of the contour of the objects and used these values for the threshold of the minimal gradient (mean = 133.46 and standard deviation = 53.81). Note that the relatively high standard deviation results from the fact that the objects often touch each other. In the area where the objects touch each other the gradient is very low. Based on these statistics we determined a threshold of 100 for the gradient, which is lower than the mean of the contour gradient but higher than the difference of

Table 14 Comparison of the matched objects by applying different thresholds for the minimal gradient

(a) Gradient representation of the test image	(b) Threshold for the minimal gradient = 24.53	(c) Threshold for the minimal gradient = 100	(d) Test image including the object numbers
			
	Recognition rate: 112.5%	Recognition rate: 87.5%	

the mean and the standard deviation of the contour gradient.

Then we matched the test case {ub_5} again against the test image. The resulting matches are given in Table 14c. For comparison purposes we also give the results of the experiment in Section 9.2 in Table 14b where we used a lower threshold of the minimal gradient.

We can see from Table 14b and c that the recognition rate decreases if the threshold of the minimal gradient increases. As we can see from the image and the recognition rate that is lower than 100%, in the actual experiment some objects were not detected. The comparison of the images of both experiments shows that there is no match for object_10. Since also the matches in the object_1 change we conclude that the higher threshold for the minimal gradient fires in order to avoid matches with edges in the object. On the other hand also the match of object_8 is rejected. From Table 14d we can see that in general this object has a lower contrast than the objects in the image. This is also reflected by the gradient along the contour which is also slightly smaller than the other ones. Additionally this object touches four other objects which also decrease the local gradients. In consequence we have to conclude that the chosen threshold in too high in order to recognize this object. Therefore, we introduced our contextual similarity measure which enables us to get the missing objects.

Table 15 Results of matching

Classes	Number of cases	Recognition rate
Alternaria Alternata	34	81.0
Aspergillus niger	5	89.0
Rhizopus Stolonifer	22	96.2
Scopulariopsis Brevicaulis	8	98.2
Ulocladium Botrytis	30	85.0
Wallenia sebi	10	78.8

9.4. Performance of the case-based object recognition system

We applied our method to six different airborne fungi spores (see Table 1). We labeled 60 objects for each of the six fungal strains. These objects were taken for the case generation according to the procedure described in Section 8. The result was a database of cases. These cases were applied to images for the particular class which consist of unknown objects.

The threshold for the score was set to 0.8. We used similarity measure s_4 and a threshold for the minimal gradient that was derived from the mean and standard deviation of the gradient of the contour of the objects (see Section 9.3).

The recognition rate is defined as the ratio of the number of correct recognized objects to the total number of objects in the image. The results of the matching process are shown in Table 15. The highest recognition rate can be achieved for the objects Rhizopus Stolonifer and Scopulariopsis Brevicaulis, since the variation of their shape is expressed well by the number of cases. For those classes where the variation of the shape of the objects is high, the number of the cases is also high. In the other cases the recognition rate shows that we do not have enough cases to recognize the classes with good recognition rate (see Alternaria Alternata and Ulocladium Botrytis). Therefore, we need to increase the number of cases. For this task we should like to develop an incremental procedure for the case acquisition in our tool. Objects that have not been recognized well will be displayed automatically for tracing, then the similarity to all other shapes will be calculated, and the clustering will be done in an incremental fashion as well. This procedure will ensure that we can learn the natural variation of the shape during usage of the system.

10. Conclusion

In this paper we have proposed a case-based object recognition system. Such a system is necessary when objects of interest should be detected on a complex background, when objects are overlapping, or touching and when thresholding-based image segmentation methods are not useful. There are many applications in biotechnology where a single case is not enough for an object recognition process and where instead there is a set of cases describing the various appearances of an object required.

We have described our study of similarity measures for case-based object recognition that can work under illumination invariance, occlusion, and clutter. The implementation of the matcher has been presented and tested on a test data set. Implementation details were given as well as results of the matching process.

The cases are represented by edges and not by the grey-level itself. The similarity measure is based on the dot product and is invariant against illumination changes and contrast changes. We have described the behavior of the similarity measure in different perspectives and showed how the similarity measure should be used for different applications.

Natural objects have a great variation in shape which does not make it easy to specify a case by hand. Therefore it is necessary to have a computerized procedure that helps to acquire the case from the real objects. We have proposed a method for the acquisition of contour instances and learning of general shape cases. We use the Procrustes 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 prototype of the cluster. The methods are implemented in the program CACM which runs on a Windows PC. Further research will be carried out to improve the information-gathering procedure and to develop an incremental clustering procedure.

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] Chen Q, Petriu EM. Optical character recognition for model-based object recognition applications. In: Proceedings of the Second IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications HAVE 2003. Los Alamitos, CA, USA: IEEE Computer Society Press; 2003 . p. 77—82.

[2] Lamdan Y, Schwartz JT, Wolfson HJ. Affine invariant model-based object recognition. IEEE Trans Robot Autom 1990; 5:578—89.

[3] Brown LG. A survey of image registration techniques. ACM Comput Surv 1992;24(4):325—76.

[4] Huttenlocher DP, Klanderman GA, Rucklidge WJ. Comparing images using the Hausdorff distance. IEEE Trans Pattern Analy Mach Intell 1993;15(1):850—63.

[5] Olson CF, Huttenlocher DP. Automatic target recognition by matching oriented edge pixels. IEEE Trans Image Process 1997;6(1):103—13.

[6] Latecki LJ, Lakämper R. Shape similarity measure based on correspondence of visual parts. IEEE Trans Pattern Anal Mach Intell 2000;22(10):1185—90.

[7] Grimson WEL, Huttenlocher DP. On the verification of hypothesized matches in model-based recognition. IEEE Trans Pattern Anal Mach Intell 1991;13(12):1201—13.

[8] Jurie F. Hypothesis verification in model-based object recognition with a Gaussian error model. In: Proceedings of the Fifth European Conference on Computer Vision ECCV'98. Berlin, Germany: Springer Verlag; 1998. p. 642—56.

[9] Katartzis A, Sahli H, Nyssen E, Cornelis J. Detection of buildings from a single airborne image using a Markov random field model. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium IGARSS'2001. Piscataway, NJ, USA: IEEE Publications; 2001. p. 2832—4.

[10] Shahrokni A, Vacchetti L, Lepetit V, Fua P. Polyhedral object detection and pose estimation for augmented reality applications. In: Proceeding of the Computer Animation. Los Alamitos, CA, USA: IEEE Computer Society Press; 2002 . p. 65—72.

[11] Leibe B, Leonardis A, Schiele B. Combined object categorization and segmentation with an implicit shape model. In: Proceedings of the Workshop on Statistical Learning in Computer Vision 2004. New York, NY, USA: Springer Verlag; 2004 p. 17—32.

[12] Burt PJ, Adelson EH. The Laplacian pyramid as a compact image code. IEEE Trans Commun 1983;31:532—40.

[13] Rangarajan A, Chui H, Bookstein FL. The softassign Procrustes matching algorithm. In: Proceedings of the 15th International Conference on Information Processing in Medical Imaging IPMI. London, UK: Springer Verlag; 1997. p. 29—42.

[14] Cheng D-C, Schmidt-Trucksäss A, Cheng K-S, Burkhardt H. Using snakes to detect the intimal and aventitial layers of the common carotid artery wall in sonographic images. Comput Meth Prog Biomed 2002;67:27—37.

[15] Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. Int J Comput Vision 1987;1(4):321—31.

[16] Bookstein FL. Size and shape spaces for landmark data in two dimensions. Statist Sci 1986;1(2):181—242.

[17] Cootes TF, Taylor CJ. A mixture model for representing shape variation. Image Vision Comput 1999;17(8):567—74.

[18] Dryden IL, Mardia KV. Statistical shape analysis. New York, NY, USA: Wiley, 1998.

[19] Kendall DG. A survey of the statistical theory of shape. Statist Sci 1989;4(2):87—120.

[20] Hill A, Taylor CJ, Brett AD. A framework for automatic landmark identification using a new method of nonrigid correspondence. IEEE Trans Pattern Anal Mach Intell 2000;22(3):241—51.

[21] Haenselmann T, Effelsberg W. Wavelet-based semi-automatic live-wire segmentation. In: Proceedings of the SPIE Human Vision and Electronic Imaging VII. Bellingham, WA, USA: The International Society for Optical Engineering; 2000. p. 387—92.

[22] Mortensen EN, Barrett WA. Intelligent scissors for image composition. In: Proceedings of the 22nd Annual Conference on Computer graphics and Interactive Techniques SIGGRAPH. New York NY, USA: ACM Press; 1995. p. 191—8.

[23] Bookstein FL. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. Med Image Anal 1997;1(3):225—44.

[24] Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 2002;24(24):509—22.

[25] Mokhtarian F, Abbasi S, Kittler J. Efficient and robust retrieval by shape content through curvature scale space. In: Proceedings of the International Workshop on Image Databases and Multimedia Search. Amsterdam, Netherlands: Amsterdam University Press; 1996. p. 35—42.

[26] Bresenham JE. Algorithm for computer control of a digital plotter. IBM Syst J 1965;4(1):25—30.

[27] Wall K, Daniellson P-E. A fast sequential method for polygonal approximation of digitized curves. Comput Graph Image Process 1984;28:220—7.

[28] Alt H, Guibas LJ. Discrete geometric shapes: matching, interpolation and approximation. In: Sack J-R, Urrutia J, editors. Handbook of computational geometry. Amsterdam, Netherlands: Elsevier; 1996. p. 121—53.

[29] Feldmar J, Ayache N. Rigid, affine and locally affine registration of free-form surfaces. Int J Comput Vision 1996;18(3):99—119.

[30] Sclaroff S, Pentland A. Modal matching for correspondence and recognition. IEEE Trans Pattern Anal Mach Intell 1995;17(6):545—61.

[31] Veltkamp RC. Shape matching: similarity measures and algorithms. In: Shape Model Int. 2001. p. 1880—197.

[32] Lele SR, Richtsmeier JT. An invariant approach to statistical analysis of shapes. London, UK: Chapman and Hall/CRC Press, 2001.

[33] Perner P. Data mining on multimedia data. Berlin, Germany: Springer Verlag, 2002.

[34] Jain AK, Dubes RC. Algorithms for clustering data. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.