International Journal of
Computer & Software Engineering

Research Article Open Access

Determining the Similarity between Two Arbitrary 2-D Shapes and its Application to Biological Objects

Petra Perner

Institute of Computer Vision and applied Computer Sciences, IBaI, Leipzig, Germany

Abstract

Generalized shape models of objects are necessary to match and identify an object in an image. To acquire these kind of models special methods are necessary that allow to learn the similarity pair-wise similarity between shapes. They mainly concern is the establishment of point correspondences between two shapes and the detection of outlier. Known algorithm assume that the aligned shapes are quite similar in a way. But special problems arise if we have to align shapes that are very different, for example aligning concave to convex shapes. In such cases it is indispensable to take into account the order of the point-sets and to enforce legal sets of correspondences; otherwise the calculated distances are incorrect. We present our novel shape alignment algorithm which can also handle such cases. The algorithm establishes symmetric and legal one-to-one point correspondences between arbitrary shapes, represented as ordered sets of 2D-points and returns a distance measure which runs between *0* and *1*.

Publication History:

Received: Ocober 22, 2018
Accepted: November 14, 2018
Published: November 16, 2018

Keywords:

Shape Alignment, Correspondence Problem, Shape Acquisition

Introduction

The analysis of shapes and shape variation is of great importance in a wide variety of disciplines. Initially in 1917 D'Arcy Thompson [1] studied the field of geometrical shape analysis from a biological point of view. Intuitively it is especially interesting for biologists since shape is one of the most concise features of an object class and may change over time due to growth or evolution. The problems of shape spaces and distances have been intensively studied by Kendall [2] and Bookstein [3] in a statistical theory of shape. We follow Kendall [2] who defined shape as all the geometrical information that remains when location, scale, and rotational effects are filtered out from the object. Sometimes it is also interesting to retain the size of the objects, but we will not consider this case here. In most applications this invariance is not given a-priori, thus it is indispensable to transform the acquired shapes into a common reference frame. Different terms are equivalently used for this operation, i.e. superimposition, registration, and alignment. Only if the shapes are aligned, it will be possible to compare them in order to describe deformations or to define a measure distance between them.

Bookstein geometrically analyzed shapes and measured their change in many biological applications, e.g. bee wings, skulls, and i.e. human schizophrenia brains [4]. To give an example, with his morphometric studies he found out that the region connecting the two hemispheres is narrower in schizophrenic brains as in normal brains. In digital image processing the statistical analysis of shape is a fundamental task in object recognition and classification. It concern applications in a wide variety of fields, e.g. [5].

The Problem of Alignment of 2-D Shapes

Consider two shape instances P and O defined by the point-sets $p_i \varepsilon R^2$, $i= 1,2,...,N_p$ and $o_k \varepsilon R^2$, $k=1,2,...,N_0$ 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) (see Figure 1).

Generally the shape instance $P=\{p_i\}$ is said to be aligned to the shape instance $O=\{o_k\}$ if a distance $d(P,O)$ between the two shapes can not be decreased by applying a transformation ψ to P. Various alignment approaches are known [6-12]. They mainly differ in the kind of mapping (i.e. similarity [8], rigid [13], affine [14]) and the chosen distance measure. A survey of different distance measures used in the field of shape matching can be found in [15].

For calculating a distance between two shape instances the knowledge of corresponding points is required. If the shapes are defined by sets of landmarks [16,17] the knowledge of point correspondences is implicit. However at the beginning of many applications this condition is not hold and often it is hard or even impossible to assign landmarks to the acquired shapes. Then it is necessary to automatically determine point correspondences between the points of two aligned shapes *P* and *O*, see Figure 2.

One of the most essential demands on these approaches is symmetry. Symmetry means obtaining the same correspondences when mapping instance *P* to instance *O* and vise versa instance *O* to instance *P*. This requirement is often bound up with the condition to establish one-to-one correspondences. This means a point o_k in shape instance *O* has exactly one corresponding point p_k in shape instance P. If we compare point sets with unequal point numbers under the condition of one-to-one mapping then it is clear that some points will not have a correspondence in the other point set. These points are called outlier.

Special problems arise if we have to align shapes that are very different, for example aligning concave to convex shapes. In these cases it is indispensable to take into account the order of the point-sets and to enforce legal sets of correspondences, otherwise the calculated distances are incorrect. This paper presents our novel algorithm for aligning arbitrary 2D-shapes, represented as ordered point-sets. In our work natural shapes are acquired manually from real images. The object shapes can appear with varying orientation, position, and scale in the image. The shapes are arbitrary and there is nothing special about them. They might have a great natural variation. They also might

***Corresponding Author:** Prof. Dr. Petra Perner, Institute of Computer Vision and applied Computer Sciences, IBaI, Arno-Nitzsche-Str. 45, 04277 Leipzig, Germany; E-mail: pperner@ibai-institut.de

Citation: Perner P (2018) Determining the Similarity between Two Arbitrary 2-D Shapes and its Application to Biological Objects. Int J Comput Softw Eng 3: 139. doi: https://doi.org/10.15344/2456-4451/2018/139

Citation: Perner P (2018) Determining the Similarity between Two Arbitrary 2-D Shapes and its Application to Biological Objects. Int J Comput Softw Eng 3: 139. doi: https://doi.org/10.15344/2456-4451/2018/139

be very similar or very dissimilar. They even might have concave or convex shapes. The algorithm establishes symmetric and legal one-to-one point correspondences between arbitrary shapes, represented as ordered sets of 2D-points and returns a distance measure which runs between *0* and *1*.

Related Work

The problems of shape spaces and distances have been intensively studied [3,2] in a statistical theory of shape. The well-known Procrustes distance [8,16] between two point-sets P and O is defined as the sum of squared distances between corresponding points

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

where *R(θ)* is the rotation matrix, μP and μ_O are the centroids of the object *P* and *O* respectively, σ_P and σ_O are the standard deviations of the distance of a point to the centroid of the shapes and N_{PO} is the number of point correspondences between the point-sets *P* and *O*. This example shows that the knowledge of correspondences is an important prerequisite for calculation of shape distances.

There has been done a lot of work which concerns the problem of automatic finding point correspondences between two unknown shapes. Hill et al. [14] presented an interesting framework for the determination of point correspondences. First, the algorithm calculates pseudo-landmarks based on a set of two-dimensional polygonal shapes. The polygon approximation is controlled by an automatic calculated threshold in order to identify a subset of points on each shape. In the next step they establish an initial estimate of correspondences based on the arc-path length of the polygons. Finally,

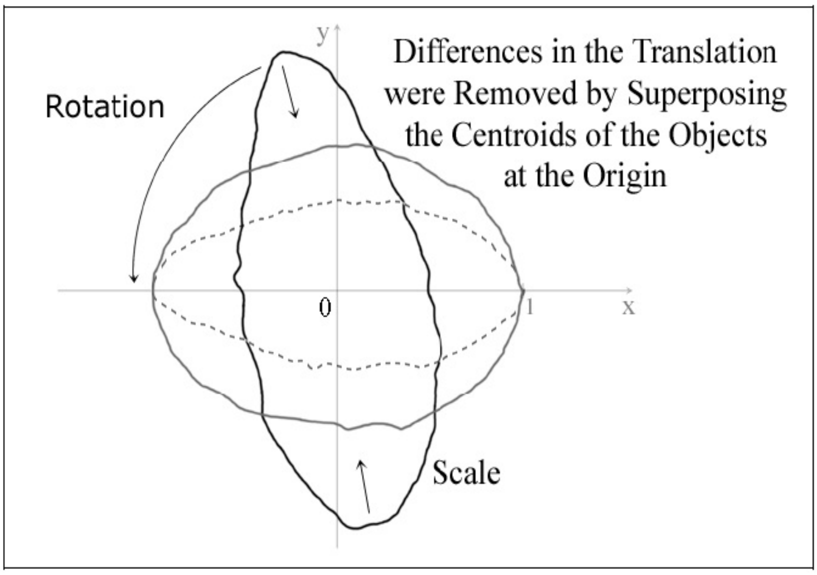

Rotation
Differences in the Translation were Removed by Superposing the Centroids of the Objects at the Origin
y
0
1
x
Scale

Figure 1: Alignment of two shapes instances with superimposition and similarity transformation.

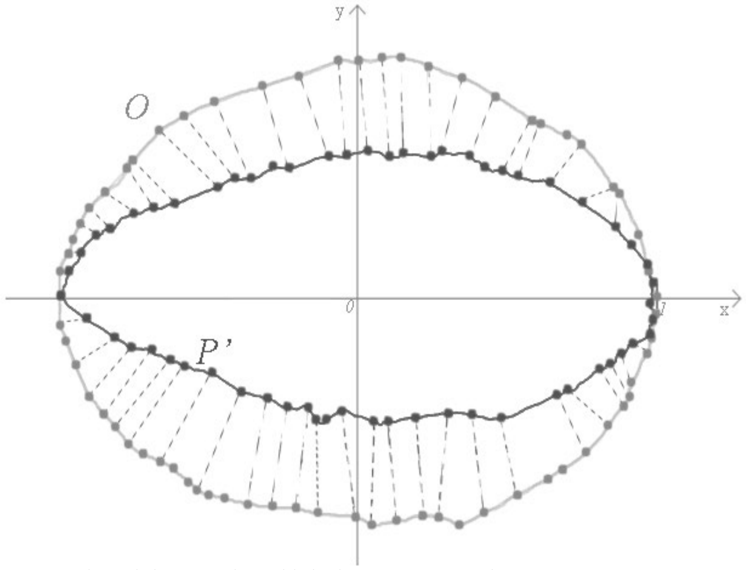


Figure 2: Aligned shapes with established point correspondences.

the greedy algorithm is used as an iterative local optimization scheme to modify the correspondences in order to minimize the distance between both polygons. The complexity of the greedy algorithm is *O(nlog n)* for sorting the elements and *O(n)* for the selection. Brett and Taylor [12] presented the extension of this method to three-dimensional surfaces. However, the algorithm was only applied to groups of objects from the same category. They assume that all acquired shapes are similar and compared them under non-rigid transformation.

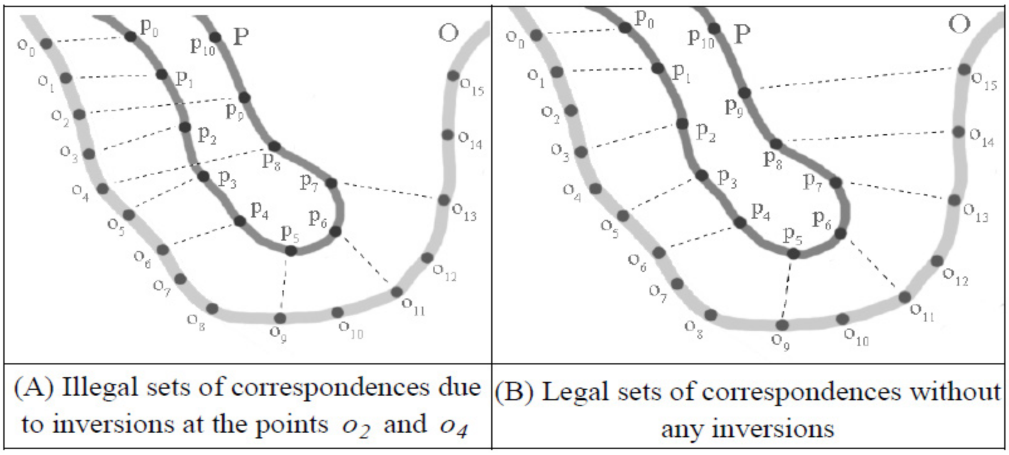
Another popular approach in solving the correspondence problem is called *Iterative Closest Point* (ICP) developed by Besl and McKay [18] and further improved in [10,11,19]. Given a set of initial, estimated registrations the *ICP* automatic converges to the nearest local minimum of a mean squared distance metric. It establishes correspondences by mapping the points of one shape to their closest points on the other shape. In the original version of the *ICP* [18] the complexity of finding for each point p_k in P the closest point in the point-set O is $O(N_pN_o)$ in worst case. Marte et al. [11] improved this complexity by applying a spatial subdivision of the points in the set O. They used clustering techniques to limit the search space of correspondences for a point p_k to points which are located within a defined range around p_k. But this can only be done because they assume that the point-sets are already in a close proximity, i.e. a reasonably good initial registration state is already given. In general, the ICP is a very simple algorithm which can also be applied to shapes with geometric data of different representations (e.g. point-sets, line segments, curves, surfaces). On the other hand it is not very robust with respect to noise and outlier. Fitzgibbon [10] replaced the closed-form inner loop of the ICP by the Levenberg-Marquardt algorithm, a non-linear optimization scheme. His results showed an increased robustness and a reduction of the dependence of the initial estimated registrations without a significant loss of speed.

The main problem of the ICP is that is does not guarantee to produce a legal set of correspondences. By a legal set of correspondences is meant that there are no inversions between successive pairs of correspondences (see figure 1). In detail, when starting from a reference pair of corresponding points and traveling successive around the complete boundaries to pairs of correspondences, the arc path lengths in relation to the reference points always have to be increasing.

An extension of the classical Procrustes alignment to point-sets of differing point counts is known as the *Softassign Procrustes Matching* algorithm [8]. It alternates between solving for correspondence, the spatial mapping, and the Procrustes rescaling. The *Softassign Procrustes Matching* algorithm is also an iterative process and uses deterministic annealing to solve the optimization problem. In deed it is a very time-consuming and computationally expensive procedure. They applied the algorithm to dense point sets not to closed boundaries. One-to-one correspondences between points were established and points without correspondences are rejected as outlier. The establishment of point correspondences is only held in a nearest neighbor framework so they do not guarantee to produce legal sets of correspondences.

Another solution of the correspondence problem was presented by Belongie et al. [20]. He added to each point in the set a descriptor called shape context. The shape context of a point is a histogram which contains information about the relative distribution of all other points in the set. The histogram can provide invariance to translation, scale, and rotation. The cost of mapping two points is calculated by comparing their histograms using the χ^2 test statistic. The χ^2 distances between all possible pairs of points between the two shapes have to be calculated which results in a square distance matrix. The best match is found where the sum of distances between all matched histograms has reached its minimum. To solve this square assignment problem they applied a shortest augmenting path algorithm for bipartite graph matching which has a time complexity of $O(n^3)$. The result is a set of one-to-one correspondences between points with similar shape contexts. The algorithm can also handle point-sets with different point counts by integrating dummy points. But it is also not guaranteed that legal sets of correspondences were established.

In none of these works is described the problem of aligning a convex to a concave shape. There arise special problems: Let us suppose that the concave shape representing the letter *C* is compared with the shape of the letter *O* (see figure 4). If the pair-wise correspondences were established between nearest neighbored points the resulting distance between both shape instances will be very small (see figure 4A). But intuitively we would say that these shapes are not very similar. In particular in such cases it is necessary to regard the order of point correspondences and to remove correspondences if they produce inversions. In addition to legality of correspondences it is important to take into account the complete contours of both shapes (see figure 4B). In result it can be seen that there arise big distances between corresponding points which leads to an increased distance measure.



(A) Illegal sets of correspondences due to inversions at the points o_2 and o_4

(B) Legal sets of correspondences without any inversions

Figure 3: Illegal and legal sets of correspondences.

Material Used for this Study

The materials we used for this study are fungal strains that are naturally 3-D objects but that are acquired in a 2-D image. These objects have a great variance in the appearance of the shape of the object because of their nature and the imaging constraints. Six fungal strains representing species with different spore types were used for the study. Figure 5 shows one of the acquired images for each analyzed fungal strain.

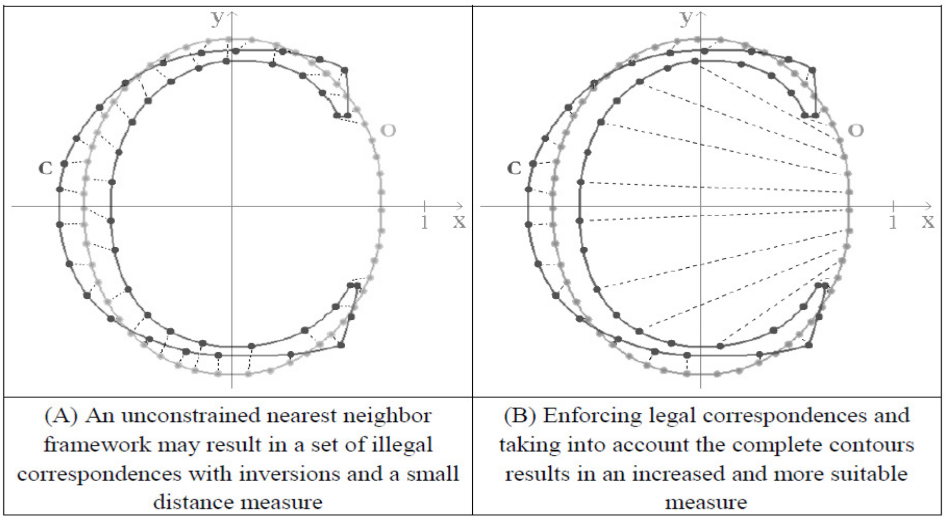
The strains were obtained from the fungal stock collection of the Institute of Microbiology, University of Jena / Germany and from the culture collection of JenaBios GmbH. All strains were cultured in Petri dishes on 2% malt extract agar (Merck) at 24°C in an incubation chamber for at least fourteen 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 lactophenol cotton blue (Merck). A database of images from the spores of these species was produced.

Acquisition of Shape Cases

Background

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 [8] and region growing [21,22] 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.

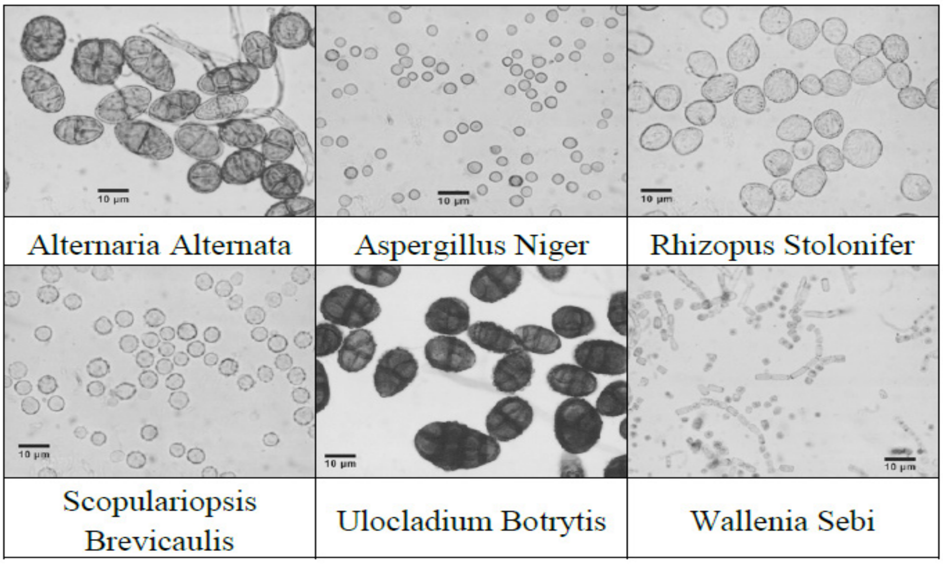
Therefore the acquisition is often performed manually at the cost of a very subjective, time-consuming procedure. In some studies it might be sufficient to reduce the shapes of objects to some characteristic points which are common features of the object class. Landmark coordinates [2,4,17,23] are manually assigned by an expert to some biologically or anatomically significant points of an organism. This has



(A) An unconstrained nearest neighbor framework may result in a set of illegal correspondences with inversions and a small distance measure

(B) Enforcing legal correspondences and taking into account the complete contours results in an increased and more suitable measure

Figure 4: Establishing correspondences while mapping a concave and convex shape.



Alternaria Alternata

Aspergillus Niger

Rhizopus Stolonifer

Scopulariopsis Brevicaulis

Ulocladium Botrytis

Wallenia Sebi

Figure 5: Images of six different fungi strains.

to be done for every single object separately. Additional landmarks can be automatic constructed using a combination of existing landmarks. The calculation of the position of these constructed landmarks has to by precisely defined, i.e. landmark *C* is located in the midpoint of the shortest line between landmark *A* and landmark *B*. Every single landmark is not only defined by its position but also by the specific and unique feature that it represents.

Landmarks might provide information about special features of an object contour but cannot capture the shape of an object because important characteristics might be lost. In many applications it is insufficient or impossible to describe the shape of an object only by landmarks. Then it is a common procedure to trace and capture the complete outlines of the objects in the images manually [14]. Indeed, manually image segmentation may be a very time-consuming and also inaccurate procedure. Therefore new semi-automatic approaches were developed [24,25] 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. The manual acquisition of objects from real images with the help of semi-automatic segmentation approaches is faster and more precise.

As described in above we are actually studying the shape of airborne fungi strains. The great natural variance in the shape of these objects and the imaging constraints have the effect that a manual assignment of landmark coordinates is quite impossible. We also reject a fully automatic segmentation procedure because this might produce outlier shapes due to the objects might be touching and overlapping. Therefore we decided to use a manually labeling procedure in our application.

Acquisition of object contours from real images

As a shape we are considering the outline of an object but not the appearance of the object inside the contour. Therefore we want to elicit from the real image the object contour P represented by a set of N_p boundary points p_k, $k=1,2,...,N_p$. The user starts labeling the shape of the object at an arbitrary pixel p_1, of the contour *P*. In our application we are only interested in the complete and closed outline of the objects. So after having traced the object, the labeling should end at a pixel p_m, $k=1,2,...,m,...,N_p$ in the 8-neighbourhood of P_1. Because it might be difficult to exactly meet this pixel manually, the contour will be closed automatic using the Bresenham [26] procedure. This algorithm connects two pixels in a raster graphic with a digital line. In addition to that we demand that two successive points may not be defined by exactly the same point coordinates in the image. In consequence we obtain an ordered, discrete, and closed sequence of points where the Euclidean distance between two successive points is either 1 or $\sqrt{2}$. In Figure 6 the scheme of the acquisition of a shape is shown for illustration.

Figure 7 presents a screenshot from our developed program *Case Acquisition and Case Mining - CACM* while labeling a shape of the strain *Ulocladium Botrytis* with the point coordinates on the right side of the screenshot.

In fact image digitization and human imprecision always implies an error during the acquisition of the object shapes. It might be very difficult to exactly determine and meet every boundary pixel of an object when manually labeling the contour of an object. Also 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.

Approximation

The amount of acquired contour points N_p of a shape *P* depends on the resolution of the input image and on the area and the contour length of the object. To speed up the succeeding computation time of the following alignment process we introduced a polygonal approximation. The resulting number of points from a polygonal approximation of the shape will be influenced by the chosen order of the polygon and the allowed approximation error. We apply the

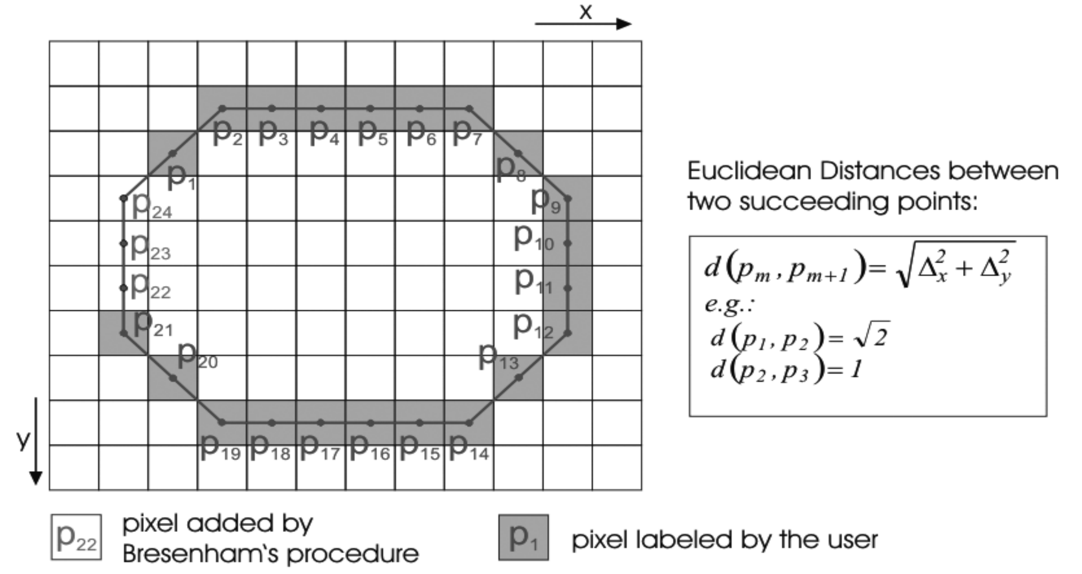

x
y
p2 p3 p4 p5 p6 p7
p1
p8
p9
p24
p10
p23
p22
p11
p21
p12
p20
p13
p19 p18 p17 p16 p15 p14
Euclidean Distances between two succeeding points:
$d(p_m, p_{m+1}) = \sqrt{\Delta_x^2 + \Delta_y^2}$
e.g.:
$d(p_1, p_2) = \sqrt{2}$
$d(p_2, p_3) = 1$
p22 pixel added by Bresenham's procedure
p1 pixel labeled by the user

Figure 6: Scheme of the labeled object outline.

approximation only to shapes which consist out of more than *200* points. This limit was introduced because the contour of very small objects in images might be defined by a few pixels only. A further reduction of information about the contour of these objects would be disadvantageous.

For the polygonal approximation, we used the approach based on the area/length ratio according to Wall and Daniellson [27] because it is a very fast and simple algorithm without time-consuming mathematic operations. Suppose the set of N_p points $p_1, p_2, \ldots, p_n$, defining the contour of the object *P*, for which a polygonal approximation is desired. We use the first labeled point p_1, as the starting point for the first approximation. Next we virtually draw a line segment from this starting point to the successor point in *P*. The area *A* between this line and the corresponding contour segment of *P* is measured. If the area *A* divided by the length of the line *L* is smaller then a predefined threshold *T*, then the same process is repeated for the next successor point in the set P (see Figure 8).

This procedure repeats until the ratio exceeds the threshold *T*. In that case the current point p_m, becomes the end point of the approximated line and the starting point for the next approximation. The same process is then repeated until the first point p_1, is reached. The result of the approximation is a subset of points of the contour *P*.

The ratio $\frac{A}{L}$ controls the maximal error of the approximation since A is the area and *L* the side length of a virtual rectangle. If the ratio is low then the other side of the virtual rectangle is small.

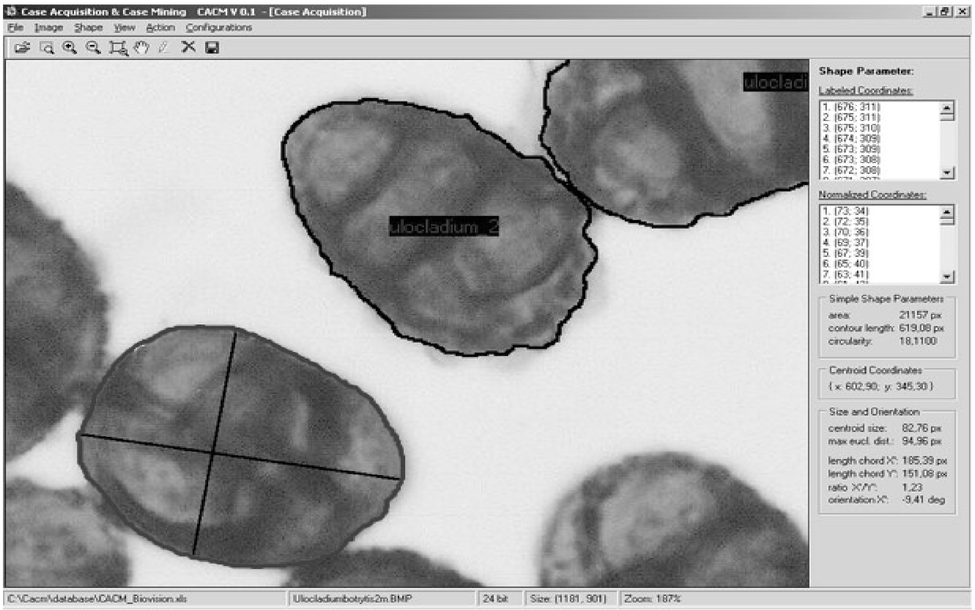
Normalization of shape cases

In our application we demand the distance measure between two shapes to be invariant under translation, scale, and rotation. Thus, in a preprocessing step we remove differences in translation and we rescale the shapes so that the maximum distance of each point from the centroid of the shape will not be larger than *one*. The invariance to rotation will be calculated during the alignment process.

The centroid $\bar{\mu}$ of a set of N_p points is given by:

$$x_{\bar{\mu}} = \frac{1}{N_P} \sum_{i=1}^{N_P} x_i$$
$$y_{\bar{\mu}} = \frac{1}{N_P} \sum_{i=1}^{N_P} y_i \qquad (2)$$

To obtain invariance under translation we translate the shape so that its centroid is at the origin


Case Acquisition & Case Mining CACM V 0.1 - [Case Acquisition]
File Image Shape View Action Configurations
uloclad
ulocladium_2

Shape Parameter:

Labeled Coordinates:

1. (676; 311)
2. (675; 311)
3. (675; 310)
4. (674; 309)
5. (673; 309)
6. (673; 308)
7. (672; 308)

Normalized Coordinates:

1. (73; 34)
2. (72; 35)
3. (70; 36)
4. (69; 37)
5. (67; 39)
6. (65; 40)
7. (63; 41)

Simple Shape Parameters

area:	21157 px
contour length:	619,08 px
circularity:	18,1100

Centroid Coordinates

(x: 602,90; y: 345,30)

Size and Orientation

centroid size:	82,76 px
max eucl. dist.:	94,96 px
length chord X':	185,39 px
length chord Y':	151,08 px
ratio X'/Y':	1,23
orientation X':	-9,41 deg

C:\Cacm\database\CACM_Biovision.xls | Ulocladiumbotrytis2m.BMP | 24 bit | Size: (1181, 901) | Zoom: 187%

Figure 7: Labeled shape with coordinates.

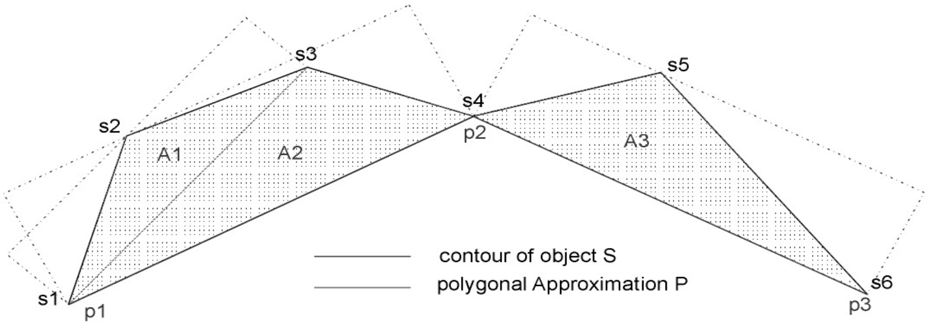


Figure 8: Polygonal approximation based on the area/length ratio [27].

$$x_i^{'} = x_i - x_{\bar{\mu}} \qquad i=1,2,\ldots,N_P \qquad y_i^{'} = y_i - y_{\bar{\mu}} \qquad (3)$$

Furthermore for each shape we calculate the scale factor t_p:

$$t_p = \frac{1}{\max_i \sqrt{x_i'^2 + y_i'^2}};\ i=1,2,\ldots,N_P \qquad (4)$$

This scale factor is applied to all points of the shape *P* to obtain the transformed shape instance *P'*. The set of acquired shapes are now superposed on the origin of a common coordinate system. The maximum Euclidean distance of a point to the origin of the coordinate system is *one*.

Shape Alignment and Distance Calculation

The alignment algorithm

In the Procrustes Distance (see Eq. 1) each of the shapes is rescaled in such a way that the standard deviation of all points to the centroid is identical ($\sigma_P=\sigma_O$). This normalization does not ensure that the resulting distance runs between *0* and *1*. Since we are interested in comparing objects of different categories we rescale the shapes so that the maximum distance of a point to the centroid is not larger than *1*. In comparison to Procrustes Distance we also do not need μ_P and μ_O because we have already removed the translation by transforming the objects into the origin.

The differences in rotation will be removed during our iterative alignment algorithm. In each iteration of this algorithm the first shape is rotated stepwise while the second shape is kept fixed. For every transformed point in the first shape we try to find a corresponding point on the second shape. Based on the distance between these corresponding points the alignment score is calculated for this specific iteration step. When the first shape is rotated once around its centroid, finally that rotation is selected and applied where the minimum alignment score was calculated.

We are regarding arbitrary shapes with varying orientations and different point counts and do not have any information about how the points of two shape instances have to be mapped onto each other. As already stated, we have to solve the correspondence problem before we are able to calculate the distance between two shape instances. In summary, for the establishment of point correspondences we demand the following facts:

1. Produce only legal sets of correspondences,
2. Produce one-to-one point correspondences,
3. Determine points without a correspondence as outlier, and
4. Produce a symmetric result, which is obtaining the same results when aligning instance *P* to instance *O* as when aligning instance *O* to *P*.

It was shown above that the establishment of legal sets of correspondences is an important fact to distinguish between concave and convex shapes. The drawback of this requirement is that the acquired shapes have to be ordered point-sets. The demand for establishing only legal sets of correspondences is also the main reason why we decided to extend the initial version of our alignment algorithm [28] that can only align concave with concave and convex with convex shapes. We extended our nearest neighbor-search algorithm presented there so that it can handle the correspondence problem while aligning concave with convex object shapes.

The input for our alignment algorithm is a pair of two normalized shapes *P* and *O* as described above. We define the shape instance *P* as the one which has less contour points then the second shape instance *O*. The instance with more points is always that one, which will be aligned to shape instance with less points. That means shape P will be kept fixed while shape *O* will be stepwise rotated to match *P*.

Before the iterative algorithm starts we define a range where to search for potential correspondences. This range is defined by a maximum deviation of the orientation according to the centroid (see Figure 9). This restriction will help us to produce legal sets of correspondences. The maximum permissible deviation of orientation γ_{dev} will be calculated in dependence of the amount of contour points n_O of the shape *O*, which is the instance that has more points than the other one. Our investigations showed that the following formula leads to a well-sized range

$$\gamma_{dev} = \pm\frac{4\pi}{n_o} \qquad (5)$$

The outline of our iterative alignment algorithm is as follows:

```
Initialize $\psi$    /* stepwise rotation angle */
SET $\psi_i = 0$       /* actual rotation angle */
REPEAT UNTIL $\psi_i \ge 2\pi$ or $SCORE(P,O_i)=0$
    (A) Rotation of $O$ with $\psi_i = \psi_{i-1} + \psi$
    (B) Calculate point correspondences between $P$ and $O$
    (C) Calculate distance $SCORE(P,O_i)$
END
RETURN $SCORE(P,O) = min\{SCORE(P,O_i)\}$
```

Calculation of point correspondences

The most difficult and time-consuming part is the establishment of point-correspondences. For each point p_k with $1 \le k \le n_p$ in the set P we try to establish exactly one corresponding point o_j with $1 \le j \le n_o$ in the set O. To create a list of potential correspondents we first select all points in O which are located in the permissible range of orientation (see Figure 9) and insert them into this list of potential correspondents $\{CorrList(p_k)\}$ of p_k. Each of the selected points meets the following condition

$$(\gamma_{P_k} - \gamma_{dev}) \le \gamma_{o_j} \le (\gamma_{pk} + \gamma_{dev}) \qquad (6)$$

with γ_{pk} the angle of point p_k and γ_{dev} the permissible range of orientation around p_k.

In Figure 10 is shown a part of two aligned shape instances at this state of registration. Each point in shape *P* is connected by lines to all of its potential correspondences. It can also be seen that there are outlier on shape *O*. As an outlier we define all points where no correspondence could be established. Resulting points without correspondences are a logical consequence if we demand one-to-one correspondences when mapping shapes with unequal number of points. Such kind of points does not have any influence on the calculated distance.

If no point in O is located in the permissible range $\{CorrList(p_k)\}$ is empty), p_k is marked as an outlier. Normally in case of aligning convex objects there is no point in P which has not at least one potential correspondence. The permissible range of orientation is chosen big enough anyway. Indeed, this kind of outliers occurs only in cases of aligning convex to concave objects. Each point p_k which has not a single potential corresponding point in O is included with the maximum distance of one when calculating the distance measure.

If there is at least one potential correspondent we calculate the squared Euclidean distances between p_k and each point in this list. Then the points in this list $\{CorrList(p_k)\}$ were sorted with ascending distances in relation to p_k. In succession we check for each point in this list if there was already assigned a corresponding point in P. The first found point without a correspondence is selected and we establish a one-to-one correspondence between them. If all points in the list $\{CorrList(p_k)\}$ have already a correspondence the point p_k is marked as an outlier. In contrast to those outliers in P which do not even have one potential correspondence in O, this kind of outlier will not be included in calculating the distance measure.

Let us assume we had gone through the complete set P and assigned to each point p_k either a correspondence in O or marked it as an outlier. In the next step we want to ensure to produce a legal set of correspondences. Therefore we go through the set P again and remove all one-to-one correspondences that are inversions (see figure 3). Suppose we have found a point kp assigned to a correspondence o_g in O that produces an inversion with the point p_m which is assigned to the point o_f $(k,m \leq n_p; g,f \leq n_o)$. First we try to remove the inversion by switching the correspondences, i.e. p_k will be assigned to o_f and p_m

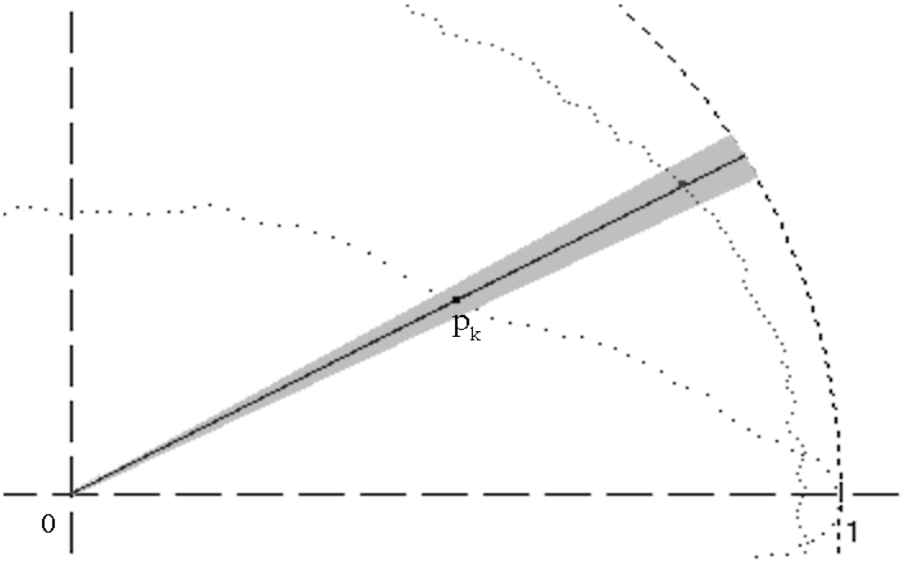


Figure 9: The permissible deviation of orientation for finding a point correspondent of the point p_k is illustrated. The line marks the orientation of the point p_k. The range where to search for a correspondence is shown.

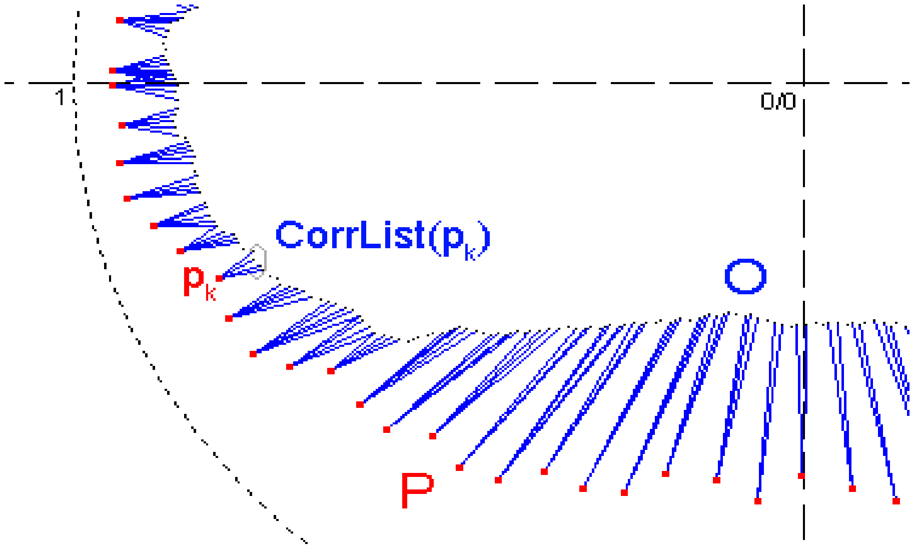


Figure 10: Part from a screenshot made during the alignment of two shape cases. The outer dotted shape *(P)* is aligned to the inner dotted shape *(O)*. Each point p_k in P is connected by lines with all of its potential correspondences that are stored in $\{CorrList(p_k)\}$.

will be assigned to o_g. Otherwise we remove the correspondence of p_k and check in succession all the remaining points in the list *{CorrList(p_k)}* if it is possible to assign a correspondence with one of these points that does not produce an inversion. If we find a point in this list, we establish the one-to-one correspondence between these two points. Otherwise we have to mark p_k as an outlier.

Calculation of the distance measure

The distance between two shapes *P* and *O* is calculated based on the sum of Euclidean distances. If there was assigned a correspondence to the point p_k, $1 \le k \le n_p$ in *p*, the Euclidean distance between this point and its corresponding point o_k in *O* is calculated by

$$d(P_k, o_k) = \sqrt{(X_p - X_o)^2 + (y_p - y_o)^2} \qquad (7)$$

If the point p_k is an outlier because there could not be assigned at least one potential correspondence and the list *{CorrList(p_k)}* is empty, the distance is set with the maximal value of one

$$d(p_k, o_k) = 1 \qquad (8)$$

The sum of all pair-wise distances between two shapes *P* and *O* is defined by

$$\varepsilon(P, O) = \sum_{k=1}^{n_p} d(P_k, O_k) \qquad (9)$$

Due to the fact that we interested in obtaining a distance measure which runs between *zero* and *one*, we normalize the sum according to n_p, the number of points in *P*

$$\bar{\varepsilon}(P, O) = \frac{1}{n_p} \varepsilon(P, O) \qquad (10)$$

The mean Euclidean distance alone is not a sufficiently satisfactory measure of distance. Particularly in cases of wavy or *jagged* contours important information get lost using the averaged distance. Therefore we also calculated the maximum distance between a pair of established correspondences

$$\varepsilon_{\max}(P, O) = \max \sum_{k=1}^{n_p} \sqrt{d(p_k, o_k)} \qquad (11)$$

The final similarity measure is the weighted sum of the mean Euclidean distance and the maximum Euclidean distance

$$Score(P, O) = \alpha \bar{\varepsilon}(P, O) + \beta \varepsilon_{\max}(P, O) \qquad (12)$$

The choice of the value for the weights α and β depends on the importance the user wants to respective distance measure. We chose for α and β a value of 0.5 which results in an equal influence of both distances to the overall distance.

Evaluation of our alignment algorithm

At first we will demonstrate that the algorithm is symmetric (see Figure 11), i.e. the same distance is calculated when aligning instance *P* to instance *O* as when aligning instance *O* to *P*. As described in above the instance with more contour points is always that one which will be aligned to shape instance with less points. Therefore, the order in which the cases are given as input into the algorithm doesn't matter. A special case where the order matters is if both shape instances are defined by the same amount of contour points.

The following table presents some results of pair-wise aligned shape cases. In the left column of the table the visual results are shown with connecting lines between corresponding points. The right column of the table presents the calculated scores and the number of outlier which are included in the alignment score with a distance value of *one*. The alignment score of value *zero* means identity and the value of 0.5 can be understood as neutral. Up to a value of *one* the shapes become more and more dissimilar.

Again we take a closer look at the pair-wise alignment of a concave and a convex shape. Reconsider the example from above where the shape of the letter *O* has to be aligned to the shape of a letter *C*. Figure 13 presents a screenshot from our program where a similar situation occurred. There were twelve outlier determined on the convex shape which are included in the calculation of the alignment score. As more of these outliers are included as more dissimilar the two shapes become because each of them is assigned with the maximum distance of one.

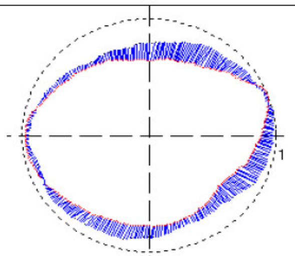
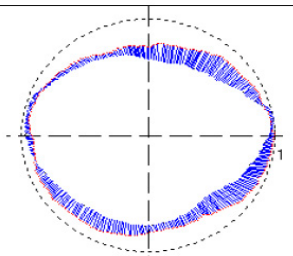
	
ulocladium_12 is aligned to *ulocladium_13* (each shape is defined by *340* contour points) $\psi_{min} = 0.2094$ $\bar{\varepsilon} = 0.0842$ $\varepsilon_{max} = 0.1835$ $\underline{\underline{Score = 0.1339}}$ Outlier included: *0*	*ulocladium_13* is aligned to *ulocladium_12* (each shape is defined by *340* contour points) $\psi_{min} = -0.2094$ $\bar{\varepsilon} = 0.0856$ $\varepsilon_{max} = 0.1822$ $\underline{\underline{Score = 0.1339}}$ Outlier included: *0*

Figure 11: Evaluation of Symmetry.

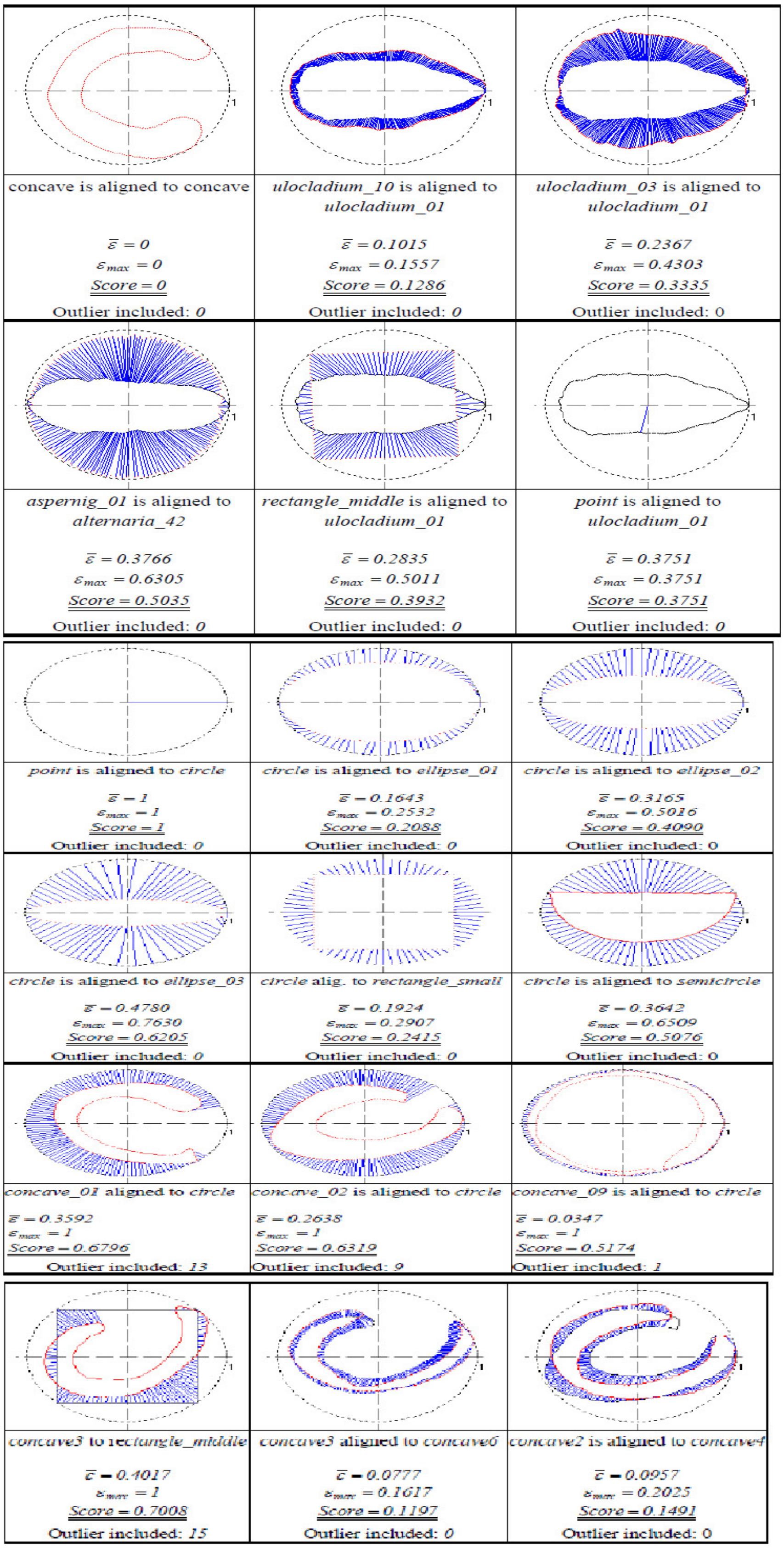
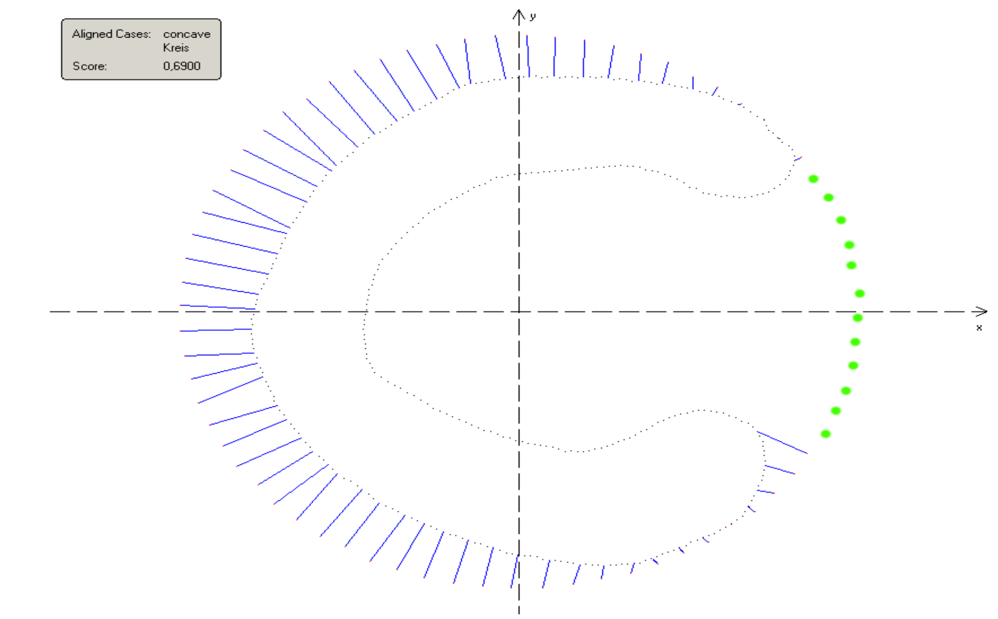


Figure 12: Different shapes with corresponding distance measures and the alignment scores.



Aligned Cases: concave Kreis
Score: 0,6900

Figure 13: Screenshot with outlier, marked as green dots, occurred during the alignment of a circle and a concave shape.

Under these circumstances the points on the inside of the concave shape may not have any influence on the resulting alignment score. This is due to the fact that they were not mapped onto opposite points of the contour of the convex shape (see figure 4B). In deed, this is an error of the algorithm but otherwise it may happen that the alignment score exceeds the value one.

Conclusions

We have proposed a method for the acquisition of shape instances and our novel algorithm for aligning arbitrary 2D-shapes, represented by ordered point-sets of varying size. Our algorithm aligns two shapes under similarity transformation; differences in rotation, scale, and translation are removed. It establishes one-to-one correspondences between pairs of shapes and ensures that the found correspondences are symmetric and legal. The method detects outlier points and can handle some amount of noise. We have evaluated that the algorithm also works well if the aligned shapes are very different, like i.e. the alignment of concave and convex shapes. A distance measure which runs between *0* and *1* is returned as result.

The methods are implemented in the program *CACM Version 1.4* which runs on a *Windows* PC.

Funding

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.

Competing Interests

The author declare no competing interests.

References

1. Thompson D (1917) On Growth and Form. Cambridge University Press, Cambridge, UK.
2. Kendall DG (1989) A Survey of the Statistical Theory of Shape. Statistical Science 4: 87-120.
3. Bookstein FL (1986) Size and Shape Spaces for Landmark Data in Two Dimensions. Statistical Science 1: 181-242.
4. Bookstein FL (1997) Landmark Methods for Forms without Landmarks: Morphometrics of Group Differences in Outline Shape. Med Image Anal 1: 225-244.
5. Alon J, Athitsos V, Sclaroff S (2005) Online and Offline Character Recognition Using Alignment to Prototypes. IEEE Computer Society Press.
6. Huttenlocher D, Klanderman G, Rucklidge W (1993) Comparing Images Using the Hausdorff Distance, IEEE Trans. Pattern Analysis and Machine Intelligence 15: 850-863.
7. Alt H, Guibas LJ (1996) Discrete Geometric Shapes: Matching, Interpolation and Approximation A Survvey. Elsevier Science Publishers.
8. Rangarajan A, Chui H, Bookstein FL (1997) The Softassign Procrustes Matching Algorithm, Proc. Information Processing in Medical Imaging.
9. Sclaroff S, Pentland A (1995) Modal Matching for Correspondence and Recognition. IEEE Trans Pattern Analysis and Machine Intelligence 17: 545-561.
10. Fitzgibbon AW (2001) Robust Registration of 2D and 3D Point Sets. In Proc British Machine Vision Conference 2: 411-420.
11. Marte OC, Marais P (2002) Model-Based Segmentation of CT Images. In South African Computer Journal 28: 54-59.
12. Brett AD, Taylor CJ (1999) A Framework for Automated Landmark Generation for Automated 3D Statistical Model Construction. Proc Information Processing in Medical Imaging.
13. Feldmar J, Ayache N (1996) Rigid, Affine and Locally Affine Registration of Free-Form Surfaces. The International Journal of Computer Vision 18: 99-119.
14. Hill A, Taylor CJ, Brett AD (2000) A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence. IEEE Transactions on Pattern Analysis and Machine Intelligence 22: 241-251.
15. Veltkamp RC (2001) Shape Matching: Similarity Measures and Algorithms. Shape Modelling International.
16. Lele SR, Richtsmeier JT (2001) An Invariant Approach to Statistical Analysis of Shapes. Chapman & Hall / CRC.

17. Dryden IL, Mardia KV (1998) Statistical Shape Analysis. John Wiley & Sons Inc.

18. Besl P, McKay N (1992) A Method for Registration of 3-D Shapes. IEEE Trans. Pattern Analysis and Machine Intelligence 14: 239-256.

19. Aksenov P, Clark I, Grant D, Inman A, Vartikovski L, et al. (2003) 3D Thermography for Quantification of Heat Generation Resulting From Inflammation. Proc 8th 3D Modelling symposium, Paris, France.

20. Belongie S, Malik J, Puzicha J (2002) Shape Matching and Object Recognition Using Shape Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24: 509-522.

21. Kass M, Witkin A, Terzopoulos D (1987) Snakes: Active contour models. In 1st International Conference on Computer Vision.

22. Cheng DC, Schmidt-Trucksäss A, Cheng KS, Burkhardt H (2002) Using Snakes to Detect the Intimal and Aventitial Layers of the Common Carotid Artery Wall in Sonographic Images. Computer Methods and Programs in Biomedicine 67: 27-37.

23. Cootes TF, Taylor CJ (1999) A Mixture Model for Representing Shape Variation. Image and Vision Computing 17: 567-574.

24. Mortensen EN, Barrett WA (1995) Intelligent Scissors for Image Composition. In Computer Graphics Proceedings.

25. Haenselmann T, Effelsberg W (2003) Wavelet-Based Semi-Automatic Live-Wire Segmentation. Proceedings of the SPIE Human Vision and Electronic Imaging 4662: 260-269.

26. Bresenham JE (1965) Algorithm for Computer Control of a Digital Plotter. IBM Systems Journal 4: 25-30.

27. Wall K, Daniellson PE (1984) A Fast Sequential Method For Polygonal Approximation of Digitized Curves. Comput Graph Image Process 28: 220-227.

28. Perner P, Jänichen S (2004) Learning of Form Models from Exemplars. Lisbon/Portugal, Springer Verlag Incs 3138: 153-161.