A Similarity-Based Object Recognition Algorithm

Petra Perner and Silke Jänichen

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

Abstract. Model-based object recognition is a challenging task especially if the objects of interest have a great natural variance. Such applications cannot be solved by only one general case, a case base is necessary to detect all objects with a sufficiently high accuracy. A flexible matching strategy is required that selects the most similar case out of the case base and fits it to an unseen object in a restricted distance transformation. In this paper we present our matching algorithm that establish point correspondences between a shape case and the found edges extracted from new images. A positive match will be found if the similarity measure holds.

Keywords: Case-based Object Recognition, Model-based Matching, Point Correspondence, Image Segmentation

1 Introduction

Model-based object recognition methods are used to detect objects of interest in images where thresholding based image segmentation methods fail. The model is required to be a generalization of the objects. In our model-based object recognition method the model consists of a set of pixel positions which describe the contour of the objects. To determine if a new unseen image contains an object, the model is matched with this image. The matching involves transforming the model with respect to the image and, for each transformation, calculating the similarity between the model and the image. A positive match is found if the similarity exceeds a defined threshold. Model-based object recognition methods might even allow classifying the found objects based on the model.

In case-based object recognition the general appearance of the model is not known in advance and cannot be directly extracted from the image, thus the generalized cases have to be learnt from real examples. Moreover, many real world applications such as biomedical applications cannot be solved by only one general case. To handle the natural variations of the appearances of the objects a case base is necessary to detect
all objects with a sufficiently high accuracy. We will present our case-based object recognition method that is used to detect objects with great natural variances in their shape.

Remind that our objects of interest have great variances in their shape and the learned cases represent only the mean shape of a group of similar shapes. This puts special requirements on the matching strategy since the algorithm must be flexible enough to fit the shape case to unseen objects. If we demand point-coincidence and calculate the evaluation score strictly based on this set of pixel positions (see Table 1, left), a lot of almost correct hypothesis would be discarded. In result a lot of shape cases would be necessary to detect all objects with a sufficiently high accuracy. Therefore, in the first version of our matching algorithm, presented in [1], we calculate the similarity score based on the orientation information of the pixels. We expanded the edges of the objects in the images so that it had no strong influence on the similarity if a case point was positioned beside an edge pixel instead of above. We have shown in [1] that our approach leads to good matching results if small dissimilarities between case and object are expected. Since we consider objects with a very high natural variation in their shape, in result a lot of cases were necessary to detect all of these objects.

Table 1. Comparison between matching with point coincidence and point correspondence

<table>
<thead>
<tr>
<th>Point Coincidence</th>
<th>Point Correspondence</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

In this paper we will present our novel matching algorithm which aims to reduce the amount of cases that are necessary. For that we have learnt for each case the degree of its generalization. It measures the permissible maximum dissimilarity from this generalized case and is used as a threshold for the similarity score while matching. The generalization measure was calculated based on the mean Euclidean distance and is normalized in order to run between zero and one. Given a mean case $C$ which represents a set of objects, and $d_C$ the maximum distance between an object from this set and $C$, the similarity threshold $SIM_C$ for matching $C$ has to be set to:
\[ \text{SIM}_C = 1 - d_C \]  

(1)

In order to do so, we will introduce a similarity measure which runs in conformity with the generalization measure of the cases. Our new similarity measure calculates the mean Euclidean distance between established point correspondences (see Fig. 1, right). It is not required that the case points are superimposed on edge pixels, since the algorithm can map a case point to an edge pixel in the neighborhood. The calculation of point correspondences is more flexible and efficient in comparison to point coincidences, because fewer cases are required for matching.

In this paper we describe the basic architecture of this case-based object recognition system. First we will discuss related work in Sect. 2. The material and the calculation of the case base are presented in Sect. 3. The matching procedure based on such a case and an unseen image is described in Sect. 4. In more detail we present the search for correspondences in Sect. 5 and, afterwards, the calculation of similarity in Sect. 6. The results for matching a set of test images are given in Sect. 7. Finally, conclusions and an outlook to further research are given in Sect. 8.

2 Related Work

A commonly used measure for object recognition is the Hausdorff distance [2], a measure for comparing binary features. The Hausdorff distance between two finite point-sets \( A = \{a_1, \ldots, a_n\} \) and \( B = \{b_1, \ldots, b_m\} \) is defined as:

\[ H(A, B) = \max(h(A, B), h(B, A)) \]  

(2)

where \( h(A, B) \) is the directed Hausdorff distance:

\[ h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \]  

(3)

The directed Hausdorff distance \( h(A, B) \) identifies the distance of the most mismatched point of \( A \) in relation to \( B \). In general \( h(A, B) \) is unequal to \( h(B, A) \), thus the Hausdorff distance is a bipartite matching where the maximum distance of both directed distances is selected. 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.

Another disadvantage of the Hausdorff distance is its sensibility to occlusion and clutter, since only a single outlier point has a strong influence on the distance. Huttenlocher et al. [2] presented a modified Hausdorff distance to overcome this weakness. They calculated the maximum of the \( k \)-th largest distances between the image and the model pixels and the maximum of the \( l \)-th largest distances between the model and the image pixels, respectively.

Olson and Huttenlocher [3] presented another modification. They included orientation information of edge pixels and presented a search strategy to examine the full
space of similarity transformations. Each orientation of an object model is treated as a separate object model. The left space of transformation, namely scale, translation in $x$, and translation in $y$, is discretized into a set of cells. Initially, that transformation out of a cell is considered, which is closest to the cell center. Pruning techniques allowed ruling out the complete cell if the initial transformation is poor enough. Otherwise the cells become divided into sub-cells which will be further analyzed.

Closely related to the Hausdorff distance is the chamfer matching [4]. The method minimizes the sum of distances from each model pixel to its closest image edge pixel over the space of possible transformations. In the original version by Barrow et al. [4] a good starting hypothesis is required in order to converge to a local optimum using an optimization procedure.

Other similarity measures use statistical information, like the maximum-likelihood image matching [5] that combines the Hausdorff distance or chamfer distance with probability density functions. The model is recognized in the image if the probability density function is maximal. Belongie et al. [6] also propose a statistical similarity measure using shape contexts: To each edge pixel is determined a histogram which counts the edge pixels around dependent on their positions relative to the pixel of interest. Using chi²-statistics and optimization methods the similarity between the model histograms and the image histograms is measured.

The similarity measure which is introduced in Latecki and Lakämper [7] 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.

3 Material Used for the 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. Six fungal strains representing species with different spore types were used for the study. Table 2 shows one of the acquired images for each analyzed fungal strain.

With a more detailed look onto the shape of these objects, it can be seen that they have a great variance in their appearance because of their nature and the imaging constraints. In fact, it is impossible to represent their shape by only one representative case, so multiple cases are necessary. But for the purpose that these object shapes should be effectively detected in new images, it is indispensable to generalize the shapes.

From the images we have manually acquired a set of shapes by tracing the object outlines. The shapes were normalized so that the centroid of the shape is superposed on the origin of the coordinate system and the maximum distance of a point from the
Afterwards, the shapes are pair-wise aligned to remove differences in their orientation and to define a measure of distance between them.

Clustering techniques can be used to mine for groups of similar shapes in this set of acquired shapes. Each group can be represented by a mean shape since this case averages over all instances in this group by generalizing their common properties. To learn generalized shape cases we have developed a novel conceptual clustering algorithm [8]. This algorithm is superior to conventional hierarchical clustering methods because it brings along a concept description. Conventional hierarchical methods divides the set of all input cases in a sequence of disjunctive partitions but give no further indication why a set of cases forms a cluster.

Table 2. Images of six different fungi strains

<table>
<thead>
<tr>
<th>Alternaria Alternata</th>
<th>Aspergillus Niger</th>
<th>Rhizopus Stolonifer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scopulariopsis Brevicaulis</td>
<td>Ulocladium Botrytis</td>
<td>Wallenia Sebi</td>
</tr>
</tbody>
</table>

Our conceptual clustering algorithm not only establishes the clusters but also explains why this set of instances forms a cluster. It provides adjuvant information for each established cluster since it is part of the concept description, namely cluster centroid, inner-cluster variance, inter-cluster variance, and the maximum distance of an instance from the centroid. Within our conceptual clustering algorithm we offer two different approaches to calculate a generalized representative shape for each cluster in the hierarchy. While the first one is to learn an artificial mean shape that is positioned in the centroid, the second one selects that instance out of a cluster which has the minimum distance to all other shape instances in this cluster.

It is also important to know the permissible distance from this generalized shape. This distance measures the error involved in the generalization and has to be taken into account in the matching process. The more groups are established, the less generalized these representative shapes will be. When matching the cases for object recognition, the similarity measure has to be set according to the degree of its abstraction. The less generalized the cases are, the higher the required similarity threshold for a positive match.
4 Matching Procedure

The input of our matching procedure is a generalized shape case from the case base, as described in Sect. 3 and the desired image for object recognition. We search for corresponding points between the case and the image pixels which belong to the boundary of an object in the image. Therefore the input image is preprocessed by a threshold-based edge detection algorithm [9]. The result of the edge extraction is a binary image where each edge pixel has a value of one.

The case \( C \) is defined by a set of \( n \) points in a normalized shape space which means all the geometrical information concerning location, orientation, and scale is removed. This dramatically reduces the amount of necessary shape cases, on the other hand we have to search the full space of similarity transformations. An object can be located everywhere in the image with an arbitrary rotation and in different size. That means, we have to shift the case across the complete image (see Fig. 1) in every possible size and every possible rotation. On every location we search for correspondences (see Sect. 5) between the transformed shape case and the assumed object edge pixel at the current position in the image. Based on the established correspondences we calculate the similarity measure (see Sect. 6). A positive match was found if this calculated similarity exceeds the threshold for similarity.

Fig. 1. Superimposing the transformed case on all possible locations on a regular grid
The outline of our matching procedure can be described as follows:

```
Extract edges from image \( I[N, M] \)
FOR scale \( \varsigma_i = \varsigma_{\text{min}}, \ldots, \varsigma_{\text{max}} \) DO
    FOR rotation \( \psi_k = -\pi, \ldots, \pi \) DO
        FOR translation \( x = 1, \ldots, N \) DO
            FOR translation \( y = 1, \ldots, M \) DO
                Calculate correspondences between transformed shape case and the image edge pixels
                Calculate similarity based on the established correspondences
                IF (calculated similarity \( \cdot \) threshold) Object detected
            ENDDF
        ENDDF
    ENDDF
ENDDF
ENDDF
ENDDF

To reduce the computational time of the algorithm we have introduced some pruning criterion. This input image is divided into single overlapping matching window. The minimum size of this window is the larger side of the surrounding rectangle of the actual shape case. Before we start to translate the case into a new matching window, we count the number of edge pixels \( E \) in it. We prune this complete image part if the amount of \( E \) is too minor, i.e. if the following condition is met:

\[
E \leq \frac{1}{2} n
\]  

(4)

with \( n \) the number of points in the shape case \( C \).

5 Search for Correspondences in the Image

Suppose the transformed shape case \( C_i \) which consists of a set of \( n \) ordered contour points \( c(x_1, y_1), c(x_2, y_2), \ldots, c(x_n, y_n) \) that are superimposed at the image pixels \( p(x_1, y_1), p(x_2, y_2), \ldots, p(x_n, y_n) \). We say that an image pixel is an edge pixel if \( p(x, y) = 1 \) otherwise it is not. As already stated, the likelihood that all contour points are exactly superimposed at an edge pixel is reasonable small if we consider objects with great variances in the contour. But it is likely that they are located in the neighborhood of their corresponding edge pixel. Thus, the search for correspondence is
finding a legal one-to-one mapping between each case point \( c(x_i, y_i) \) and the edge pixel \( s(x_k, y_k) = 1 \) in the image, where the distance between \( p(x_i, y_i) \) and \( s(x_k, y_k) \) is small. We do not strictly demand that \( s(x_k, y_k) \) has to be the nearest neighbor of \( p(x_i, y_i) \) for two reasons. The first is that we restrict the search space of the corresponding point to a special orientation in relation to \( p(x_i, y_i) \). And the second reason is to increase the robustness of our algorithm concerning occlusion and clutter.

The search for correspondence is graphically represented in Fig. 2. The search space for a potential corresponding edge pixel of a case point \( c_i \), which is superposed at the image pixel \( p_i \), is restricted to the perpendicular \( \perp \) of the line that connects \( p_i \) with \( p_{i-1} \). The maximum distance of a corresponding point on the perpendicular \( d_{\text{max}} \) is given by distance between \( p_i \) with \( p_{i-1} \). These restrictions were made in
order to promote the establishment of legal correspondences and to maintain the basic shape of the case. Starting from \( p_i \), there are always two possible directions \( \text{dir} \) where to search for correspondences on the perpendicular line. We have defined these directions as \textit{north} and \textit{south}, where their angles in relation to \( p_i \) are defined by:

\[
\varphi_{\text{north}} = \alpha - \frac{\pi}{2}, \quad \varphi_{\text{south}} = \alpha + \frac{\pi}{2}
\]  

(5)

with \( \alpha \) the angle between the line connecting \( p_i \) with \( p_{i-1} \) and the x-axis.

In Fig. 2 shows a schematic depiction of the search for correspondences. There are three case points which are directly superimposed on the edge pixel \( p_1, p_2, \) and also \( p_6 \) in the image. For these points it is not necessary to search for a corresponding point. If a case point \( c_j \) is not directly superimposed on an edge pixel (\( p(x, y) \neq 1 \)) and furthermore there is no edge pixel on the perpendicular in both directions, than \( c_i \) remains superimposed on \( p_i \). An example for such a case is given in Fig. 2 by the case point superimposed on the pixel \( p_3 \).

We apply the differentiation in both directions to increase the robustness of the algorithm against occlusion and clutter. To give an example, in Fig. 2 the first corresponding point \( s_1 \) was found in direction \textit{south} related to \( p_4 \), so that direction \textit{south} becomes our preferred direction. When searching a corresponding point for the next case point superimposed on \( p_2 \), we start to analyze pixels in the preferred direction with increasing distance first. That is the reason why the corresponding point \( s_2 \) is also established in direction \textit{south} although there is also a potential corresponding edge pixel in \textit{north} which has the same distance to \( p_5 \). The preferred direction will be switched, if an edge pixel can only be found in the non-preferred direction.

The result of the search for correspondences is the set of image pixels positions \( P \) where each instance is assigned to a point of \( C \). Points in \( C \) which have no corresponding edge pixel are marked as outlier. The left open problem is to calculate the distance between the pairs of point correspondences.
The outline for our matching algorithm based on the search of correspondences is as follows:

Set $DIR = \{\text{north, south}\}$ /* preferred direction to search for correspondences */

DO FOR all points $c_i$ with $1 \leq i \leq n$ of case C BEGIN
- Apply actual transformation so that $c_i$ superimposes $p_i$ in the image
- IF ( $p(x_i, y_i) = I$ ) /* if pixel $p_i$ is an edge pixel */
  - Define $p_i$ as point correspondence of $c_i$
- ELSE
  - Calculate $d_{max}$ /* distance between $c_{i-1}$ and $c_i */
  - Calculate $L$ /* perpendicular to the line that connects $p_{i-1}$ with $p_i */
  - Set $S = \{\emptyset\}$ /* initialize the set of possible correspondences */

  - Put into $S$ all pixels which are located on $L$ with a maximum distance of $d_{max}$ related to $p_i$
  - SORT $S$ with increasing distance related to $p_i$
  - DO FOR all points $s_k$ in $S$ WHERE $dir(s_k) \subseteq DIR$ BEGIN
    /* if pixel $s_k$ is an edge pixel and is located in preferred direction */
    - IF $s(x_k, y_k) = I$ THEN
      - Define $s_k$ as point correspondence of $c_i$
      - SET $DIR = \{dir(s_k)\}$ /* set/maintain preferred Direction */
      - BREAK
    END
  END /* if no edge pixel was found in the set $S$ */
  - IF (no point correspondence found) THEN
    Define $c_i$ as an outlier
END
6 Calculation of the Similarity

During the search for correspondences we have tried to assign to each point \(e(x_i, y_i)\) of the case \(C\) an edge pixel \(p(x_k, y_k) = 1\) in the image. Points in \(C\) without any corresponding edge pixel are marked as outliers. The Euclidean distance \(d(e_i, p_k)\) between the case point \(e_i\) and the corresponding image pixel \(p_k\) is defined by

\[
d(e_i, p_k) = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}.
\] (6)

Let \(D\) be the sum of Euclidean distances between each pair of correspondences. The distance increases as more corresponding point-pairs are established and as more distant these correspondences are, but there is no distinct value of inequality. Since we want to relate the dissimilarity between small objects with the dissimilarity between big objects, it is required to normalize the distance \(D\). Moreover we want our final similarity measure to run between zero and one. Therefore we first normalize \(D\) according to the actual size \(\phi_i\) of the case.

\[
\varepsilon = \frac{D}{\phi_i}.
\] (7)

For each point of the case \(C\) which was not assigned a corresponding edge pixel in the image we add the maximum value of inequality. Suppose that we have marked \(j\) points in \(C\) as outlier points. Then, the error \(\varepsilon\) is increased as follows

\[
\varepsilon = \varepsilon + j \phi_i.
\] (8)

Finally, we normalize this distance according to \(n\), the complete number of points in the case \(C\) and obtain a mean matching error \(\overline{\varepsilon}\)

\[
\overline{\varepsilon} = \frac{\varepsilon}{n}.
\] (9)

To run in conformity with the distance measure applied for learning the case base, we also need to consider the maximum matching error. The maximum matching error is given by

\[
\varepsilon_{\text{max}} = \max \sum_{i=1}^{n} d(e_i, p_k).
\] (10)

Our final measure of similarity is calculated by the mean of both error measures

\[
SIM = 1 - \left( \frac{\overline{\varepsilon} + \varepsilon_{\text{max}}}{2} \right).
\] (11)

This standardized measure of similarity reaches a value of one in case of identity and the maximum inequality is limited by value zero.
7 Experimental Results

First we have applied the matcher to synthetic images using synthetic standard shapes. After the method has proved under these conditions we applied it to our images using the learned shape cases as described in Sect. 3. The goal is to analyze if the all objects that are represented by a case are detected by this case. Furthermore we want to find out if the degree of generalization, which we have learnt for each case, can be directly set as threshold for similarity when matching this case.

In Fig. 3 are exemplarily shown the images of two cases with their maximum distances.

<table>
<thead>
<tr>
<th>case:</th>
<th>max_dist:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2_454</td>
<td>0.0452</td>
</tr>
<tr>
<td>2_499</td>
<td>0.1190</td>
</tr>
</tbody>
</table>

Fig. 3. Image of the applied case 2_454

The first applied case, depicted on the left side in Fig. 3, is a generalized mean shape representing 99 object shapes. The maximum distance of a represented object to this case is 0.0452. Therefore, the threshold for similarity while matching has to be set to 0.9548. In Table 3 is exemplary shown the result of a detected object using this case. The threshold for minimum similarity holds reliable.

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
<th>Case</th>
<th>Scale</th>
<th>Rotation</th>
<th>SIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>case 2_454 superimposed on the pre-processed edge image</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The object in Table 3 is isolated positioned in the images. In the following we will analyze what happens with the similarity measure if the objects are touching or overlapping? The two objects shown in Table 4 and Table 5 are touched by other objects. In result the edge extraction leads to fragmentary contours. Gaps in the contour will lead to increasing of outliers because there will not be found any corresponding edge pixels. The images were matched with the case 2_449, shown on the ride side of Fig. 3, which allows a maximum distance of 0.1190. Therefore, the similarity threshold was set to 0.8810. In spite of the gap in the contour of the object ub_01 in Table 4 and an increased number of detected outliers, the similarity threshold is exceeded. The opposite is given with the object ub_10 in Table 5. The three gaps in its contour are too big. We had to decrease the similarity threshold to a value of 0.8082 in order to detect this object.

Table 4. Detected object by case 2_449

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
<th>Case</th>
<th>Scale</th>
<th>Rotation</th>
<th>SIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ub_01</td>
<td>2_449</td>
<td>1.55</td>
<td>0.3665</td>
<td>0.89616407</td>
</tr>
</tbody>
</table>

Table 5. Object detected with reduced similarity threshold by case 2_449

<table>
<thead>
<tr>
<th>ID</th>
<th>Object</th>
<th>Case</th>
<th>Scale</th>
<th>Rotation</th>
<th>SIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ub_10</td>
<td>2_449</td>
<td>1.65</td>
<td>1.9896</td>
<td>0.80825652</td>
</tr>
</tbody>
</table>
In result, in order to detect all objects in the images we have to include into our similarity threshold a tolerance to consider gaps in the contour of the objects. We can allow a percentage of the object contour to be occluded, cluttered, or missing. Then we have to decrease our similarity threshold proportional:

$$SIM_i = SIM - t.$$  

For example, if we allow that 25% of the contour might be missing, than we decrease the similarity with $t = 0.25$. In fact, a reduction of the similarity threshold will also lead to an enhanced probability of false alarms.

8 Conclusions

In this paper we have presented our similarity-based matching algorithm. The input of this algorithm is a set of mean shapes that are generalized cases representing sets of similar shapes. Each case brings along a measure of its generalization - the maximum permissible distance from this case. Our novel matching algorithm was developed in order to run in conformity with this generalization measure. That means, we aimed to directly involve this measure in the matching process as the threshold for similarity. For that purpose we have introduced into the algorithm a novel normalized similarity measure which is based on the mean and maximum Euclidean distances between corresponding points. Point correspondences had to be established between case points and edge pixels of the supposed object in the image. Points of the case without corresponding edge pixels were marked as outlier and reduce the calculated similarity. The resulting similarity measure is normalized and runs between zero and one. The algorithm was successfully tested on a set of images including spores of fungal strains.

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


8. Jänichen, S. and Perner P., Learning General Cases of 2-dimensional Forms, Artificial Intelligence in Medicine, Elsevier, ISSN 0933-3657, To Appear 2005