

Shape Matching Using Point Context and Contour Segments

Christian Feinen, Cong Yang, Oliver Tiebe, and Marcin Grzegorzek^(✉)

Research Group for Pattern Recognition, Department ETI, University of Siegen,
Hoelderlinstr. 3, 57076 Siegen, Germany
marcin.grzegorzek@uni-siegen.de

Abstract. This paper proposes a novel method to generate robust contour partition points and applies them to produce point context and contour segment features for shape matching. The main idea is to match object shapes by matching contour partition points and contour segments. In contrast to typical shape context method, we do not consider the topological graph structure since our approach is only considering a small number of partition points rather than full contour points. The experimental results demonstrate that our method is able to produce correct results in the presence of articulations, stretching, and contour deformations. The most significant scientific contributions of this paper include (i) the introduction of a novel partition point extraction technique for point context and contour segments as well as (ii) a new fused similarity measure for object matching and recognition, and (iii) the impressive robustness of the method in an object retrieval scenario as well as in a real application for environmental microorganism recognition.

1 Introduction

Shape is a powerful feature for recognition as psychophysical studies [1] have shown. As an important feature carrier, shape contour is widely used in object detection and matching methods [2, 3]. This is because shape contour is invariant to lighting conditions and variations in object colour and texture. More importantly, it can efficiently represent image structures with large spatial extents [3].

Since shape contour is sensitive to object deformation, developing robust contour-based shape descriptors invariant to shape rotation, translation, scaling, and transformation is still a challenge task. Although skeleton-based descriptors contain both shape features and topological structures of original objects, they require heavy calculation for skeleton pruning [4]. Moreover, the following matching process is highly dependent on the quality of branch pruning. Recently, researchers have tried to employ contour segment [2, 5, 6] for shape matching since contour segments are invariant to shape deformation. However, in order to generate applicable contour segments, appropriate partition points are required. Traditionally, skeleton endpoints can be used as partition points since they are stable and covering most visual parts of original shapes. Nevertheless, as we discussed previously, it heavily relies on skeleton pruning.

The main goal of this paper is to present a method that extracts the exact contour partition points and its related robust descriptors for shape matching. Our proposed method is easy to implement and can be computed efficiently. To achieve this, we propose a novel method for generating contour partition points which is robust and stable for shape deformation. Based on these points, we generate point context and contour segments as shape descriptor. For object matching, we apply the point context-based matching and the contour segments-based matching independently. By fusing two matching results with an integrated Gradient Hill Climbing [7] and Simulated Annealing [8] method, we calculate the shape distance and apply it for a shape retrieval scenario. Finally, we employ a modified mutual k NN graph on the distance matrix to improve the accuracy of shape retrieval. The advantages of our proposed approach include (i) low computational complexity as well as (ii) the inclusion of only a limited number of sampling points instead of all contour points with point context, and (iii) the impressive robustness of the method in an object retrieval scenario.

2 Related Work

Contour-Based Shape Matching: For contour-based approaches, many methods achieve state-of-the-art performance by only utilising contour information. For example, Shotton et al. [9] present a categorical object detection scheme that uses only local contour-based features and is realised in a partly supervised learning framework. Nguyen et al. [10] propose a shape-based local binary descriptor for object detection that has been tested in the task of detecting humans from static images. In [11], an algorithm for partial shape matching with mildly non-rigid deformations using Markov chains and the Monte Carlo method is introduced. Based on contour points, Belongie et al. proposed shape context [12] descriptor for object matching. Shape context appears to be one of the best performing shape descriptor and definitely the most popular one. With the same basic geometric units of shape context, Ma and Latecki [6] proposed a shape descriptor that is particularly suitable for shape matching of edge fragments in images to model contours of target objects. Inspired by the Bag of Words model, Wang et al. [13] introduced the Bag of contour fragments as a higher level shape descriptor. Combining it with a linear SVM the authors were able to achieve impressive results on existing Datasets.

Skeleton-Based Shape Matching: Though skeleton (or medial axis) can be generated by shape contour with maximal disk model [14], comparing to contour matching methods, skeleton matching approaches feature lower sensitivity to occlusion, limb growth and articulation [15]. However, they are computationally more complex [16] and still have not yet been fully successfully applied to real images. Baseski et al. [17] present a tree-edit-based shape matching method that uses a recent coarse skeleton representation. Their dissimilarity measure gives a better understanding within group versus between group separation which mimics the asymmetric nature of human similarity judgements. To the best of our knowledge, the best performing skeleton-based object matching algorithm

has been proposed by Bai et al. [18]. Their main idea is to match skeleton graphs by comparing the geodesic paths between skeleton endpoints.

Shape Matching by Skeletons and Contours: Besides the utilisation of only contour or skeleton features for shape matching, some fused shape descriptors with contours and skeletons have been proposed. Bai et al. [5] combine the object contour and skeleton properties for shape classification. They extract contour segments using the Discrete Curve Evolution [4]. However, their approach works in a supervised pattern recognition mode and multiple training examples of an object class are necessary for modelling. Zeng et al. [19] combine properties of skeletons and boundaries for general shape decomposition. Unfortunately, this method is rotation variant and highly sensitive to shape deformation. Yang et al. [2] proposed an approach for object retrieval that uses contour segment and skeleton matching for shape similarity computation. They employ skeleton endpoints for contour partition. For contour segment, they proposed a new feature extraction technique for contour segments. Though this method achieved excellent results, however, it still highly relies on skeleton pruning.

3 Shape Representation

Partition Point: Here we describe the contour partitioning approach employed for separating the shape of an arbitrary object Ω into a set of sub-parts. This subset as well as the partition points p_i are required to perform the matching described below. The general idea is to detect contour regions with a high curvature toward the overall shape trend. Therefore, either a single or multiple *reference points* (x_i) inside the shape are selected to compute distances between all contour points $\partial\Omega$ and their closest x_i . By sampling the contour clockwise and arranging these values linearly, a signal s is generated that can be used to detect peaks in its second-order derivative. By backtracking these peaks, the desired partition points are obtained.

Therefore, first the contour is extracted and subsequently converted to a polygon \mathcal{P} . For the purpose of polygonisation the well-known *Douglas-Peucker* technique is recursively applied to $\partial\Omega$. Here, only a small tolerance value $\epsilon^{(DP)}$ is taken which drastically reduces the noise-to-signal ratio without removing significant characteristics of the shape. Figure 1 illustrates this on the example image of the bird by using two different tolerance values.

Next, the reference points x_i have to be located inside the shape in order to sample the contour. They are detected by utilising a high accurate *fast marching*



Fig. 1. The original image and two versions created by different tolerance values.

method proposed by the authors Hassouna et al. in [20]. The result is a distance map whose values are generated based on a given initialisation (seed point(s)). In this context, \mathcal{P} is taken as initialisation with the result of a distance or rather a time-crossing map T . The values in T can also be interpreted as contour contraction or expansion, respectively. This is illustrated in Fig. 2.

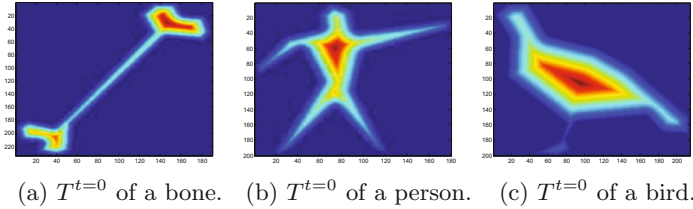


Fig. 2. The figures shows the time-crossing maps of a bone, a person and a bird. They can also be interpreted as distance maps initialised with the objects' contours.

The actual reference points are then discovered at maximum value locations in T . Therefore, a detection procedure is localising iteratively candidate regions and validates them in terms of being a new reference point. This is done based on a dynamically adapting threshold $\mu^{(BG)}$, that is applied to T in order to remove background values to emphasise these candidate regions:

$$\mu^{(BG)} = \phi(T(\Omega)) - 2 \cdot \psi(T(\Omega)), \quad (1)$$

where $\phi(\cdot)$ is a function returning the maximum value with $\{v^* \mid v^* \in T, \forall v \in T : v^* > v\}$ inside Ω . In contrast to this, $\psi(\cdot)$ indicates the standard derivation. Once the background pixels are removed, the remaining ones are clustered to disjoint regions \mathbf{A}_i which are subsequently used to determine *weighted centroids*, respectively. The first iteration ($t = 0$) is then completed with storing all $\mathbf{A}_i^{t=0}$ inside a binary mask \mathbf{B} having the same size as the input image \mathbf{I} and the weighted centroids are collected in a set, depicted by $\mathcal{C}^{t=0}$. The next iterations are processing the same way, but differ as follows: In each further iteration t , the algorithm loops only over the reference points found previously (\mathcal{C}^{t-1}), where each of them is used as seed point for the fast marching method. The resulting T_i^t (where $i = [1, 2, \dots, \otimes(\mathcal{C}^{t-1})]$ with $\otimes(\cdot)$ being a function that returns the number of elements) is then applied (here multiplied) to $T^{t=0}$. The effect of $T^t \circ T^{t=0}$ is illustrated in Fig. 3 based on the example of a glass. As shown, the method would not return a reference point for the lower part by only using $T^{t=0}$.

On the basis $T^t \circ T^{t=0}$ the threshold $\mu^{(BG)}$ is adapted and new candidate regions \mathbf{A}_i^t are located. Subsequently, the centroids (\mathbf{x}_i^t) are calculated for all \mathbf{A}_i^t . If one of these \mathbf{x}_i^t is residing in an already processed area, which means $\mathbf{B}(\mathbf{x}_i^t) = 1$, the point is rejected, otherwise $\mathbf{x}_i^t \in \mathcal{C}^t$ and \mathbf{B} is updated. It terminates if the current iteration returns an empty set of newly detected reference points ($\mathcal{C}^t = \emptyset$). The final set, in turn, is generated by $\mathbf{P} = \bigcup_{j=0}^t \mathcal{C}^{t=j}$. Finally, all reference points have to pass two further constraints. First, none of the reference points

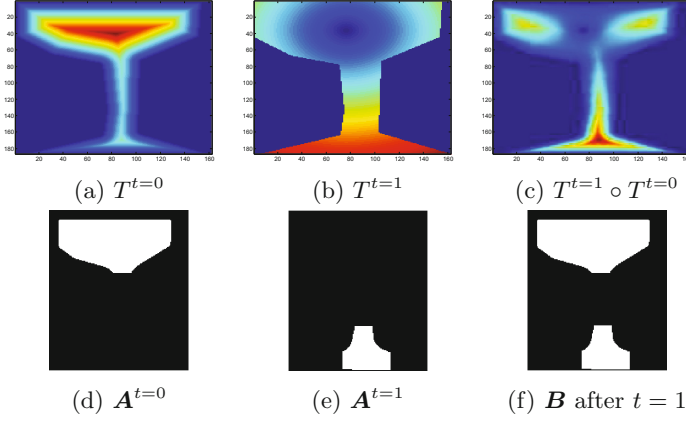


Fig. 3. The figure illustrates the working principle to detect reference points iteratively. As shown, the lower point inside the glass would not be found by only using $T^{t=0}$.

are allowed to be too close to the centroid of the shape, which is ensured with the threshold $\mu^{(cog)}$. If only one \mathbf{x}_i is violating this constraint, all reference points are discarded except of the responsible one. Second, in presence of multiple reference points, the contour has to be split into sub-parts. During that separation the algorithm monitors that each \mathbf{x}_i has only one assignment. If two contour parts get assigned to one \mathbf{x}_i , the point will be removed. Once the validation is completed, the fast marching approach has to be invoked a last time initialised by the final set of reference points. From the resulting time-crossing map $T^{(final)}$ only the values corresponding to the object's contour are extracted. Afterwards, they are arranged in a vector \mathbf{s} where the number of elements equals the number of pixels belonging to $\partial\Omega$. Actually, this data is responsible for the final contour partitioning. For this purpose, \mathbf{s} is regarded as 1D signal in the following content. The goal is now to derive the second-order derivative of \mathbf{s} to determine maximum and minimum points. By backtracking these turning points, contour pixel can be selected as partition points \mathbf{p}_i for the object's contour. Figure 4 shows both, the $T^{(final)}$ values corresponding to the contour as well as their signal plots. Since the bone structure yields two reference points, the contour has to be separated into two parts, which are shown in the first and the second column.

It is obvious that the noise-to-signal ratio exacerbates the peak detection with the result of an imprecise and noisy contour partitioning set. Thus, all subsequent matching methods are faced with unnecessarily challenging input data. In order to alleviate artifacts and to increase the robustness of the proposed approach, the signal is smoothed at first. Therefore, the *Fast Fourier Transform (FFT)* is employed with the intention to implement a *low pass filter (LPF)*. The FFT is applied, since it is capable to transform a periodic signal that is defined in the time or spatial domain toward its frequency spectrum as vice-versa. Therefore, the signal is approximated as a sum of sine and cosine functions, where the coefficients describe the power of the corresponding frequency in the input signal.

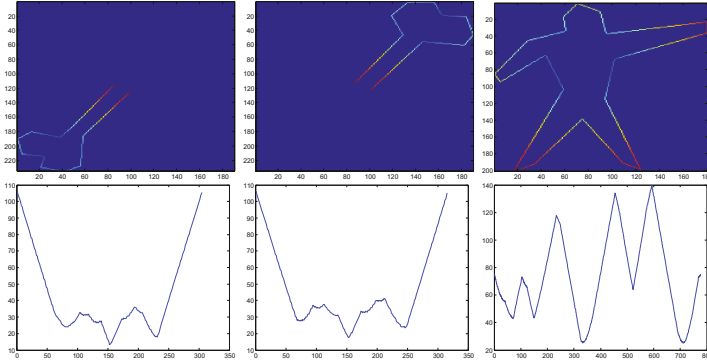


Fig. 4. The figure shows the 1D signal that has been derived from the contour depicted on the image above. The colours on the contour line encode the distance from a pixel to its closest reference point.

Since noise (and only detail information) usually manifests in high frequencies, thus, one can easily suppress noisy information without losing characteristic properties of the signal. This is shown in Fig. 5, where the signals of Fig. 4 are plotted once again, but this time after applying a LPF to them. Here, the LPF is established by a Gaussian filter that is directly applied to the zero-shifted frequency domain.

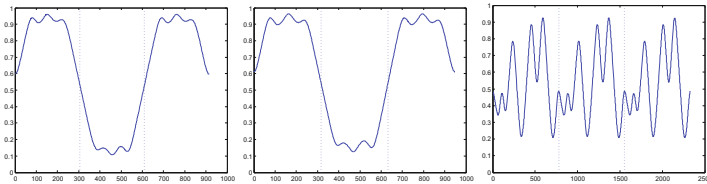


Fig. 5. From left to right, one can observe the smoothed signal of the lower left, upper right part of the bone as well as of the person. These smoothed versions are achieved by applying all LPF established in the frequency domain. It has to be noted, that the signals have been padded at start and end to alleviate border artifacts.

Two things have to be noted here: First, before the signal is passed to the LPF, it has to be padded at the start and at the end to alleviate border artifacts. Second, the padding process reacts in different ways depending on the number of reference points. This means, in the case of only a single reference, the signal is doubled at its start and its end, since the contour was not separated and thus forms a circle. If there are more than one x_i (with $i > 1$), the contour parts should not be repeated in the signal, since it would lead to a fake peak. Hence, the start and the end are only extended by flipping s . Finally, the first (\hat{s}') and the second-order (\hat{s}'') derivative of the smoothed (and discretised) signal

\hat{s} is calculated to determine all zero locations in \hat{s}' with positive or negative curvature in \hat{s}'' . Figure 6 shows the second-order derivatives for the signal plots in Fig. 5. Please note, that the computation of the second-order derivative only considers a signed binary version of the first-order derivative ($\text{sign}(\hat{s}')$, where $\text{sign}(\cdot)$ returns 0 if $\hat{s}'_i = 0$, the value 1 if $\hat{s}'_i > 0$ and -1 otherwise).

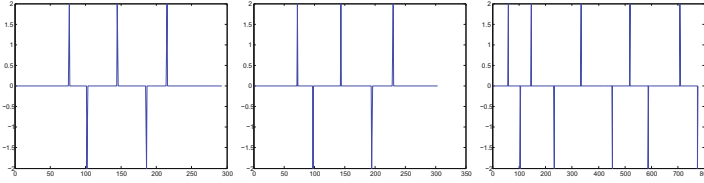


Fig. 6. Second-order derivatives that correspond to the smoothed versions of the signals shown in Fig. 4. Please note, that for the computation of the second-order derivative only a signed binary version of the first-order derivative is taken.

Theoretically, the desired partition points p_i can now be easily detected by localising all positions in \hat{s}'' which are not equal to zero. Practically, this naive approach might still encompass limited amount of false-positives generated by noise. The problem is, that it is not possible to find one Gaussian filter that is fitting for all object instances caused by the varying degree of deformation. In order to tackle this problem, the second-order derivative is analysed in terms of neighbours whose distance is too close to each other. Figure 7 demonstrates such a situation, where the red circle indicates the problem. To detect such cases in \hat{s}'' , a threshold, $\mu^{(sdiff)}$, is applied, defined as the half of the standard derivation of all distances between adjacent peaks in \hat{s}'' .

If the distance between adjacent peaks is below $\mu^{(sdiff)}$, the difference of their heights (taken from \hat{s}) is compared to a threshold $\mu^{(hdiff)}$ which has to be chosen close to zero. If this threshold is also greater than the height difference, the sigma of the Gaussian filter is decreased by 0.1 and the smoothing procedure is repeated with the original s . Experiments showed, that it only requires one to

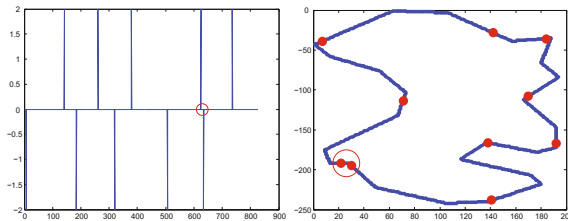


Fig. 7. The figure illustrates a situation, where the sigma of the Gaussian filter has not been chosen appropriately. It is obvious, that these artifacts can be easily determined by analysing the second-order derivative. The red circles indicate the problem.

two iterations if the algorithm starts with a $\sigma = 1$. Finally, based on this validated set of peaks, the corresponding contour partition points are determined. Figure 8 shows the final results of the bone, the person and the bird.

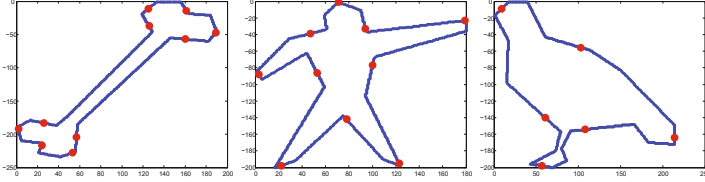


Fig. 8. Final contour partitioning set of the bone, bird and the person.

Point Context: For each partition point $\mathbf{p}_i, i = 1, 2, \dots, N$, we propose a point descriptor, *Point Context*, which has the same basic geometric unit as shape context [12]. Considering the set of vectors originating from a partition point to all other points on a shape, these vectors express the configuration of the entire shape relative to the reference point. However, different from the proposed matching method from [12], we only consider the feature vectors from partition points instead of roughly uniform spacing for selecting sample points. Moreover, during object shape matching, we calculate the shape distance by corresponding matrix between pairs of partition points without any modelling transformation process. We employ a diagram of log-polar histogram bins used in computing the point context. More specifically, as has been defined in [12], for a partition point \mathbf{p}_i on the shape contour, we compute a coarse histogram p_i of the relative coordinates of the remaining contour points,

$$p_{i,m} = \#\{\mathbf{q} \neq \mathbf{p}_i : (\mathbf{q} - \mathbf{p}_i) \in \text{bin}(m)\}, \quad (2)$$

where $\#\{\cdot\}$ is the function that extracts feature values for log-polar histogram and $\mathbf{q} = \partial\Omega \setminus \{\mathbf{p}_i\}$. We use five bins for $\log r$ and 12 bins for θ . This histogram is defined to be the point context of \mathbf{p}_i . To achieve scale invariance we normalise all radial distances by the mean distance between the contour point pairs in the shape. Thereafter, for each partition point \mathbf{p}_i , it can be represented as a 60 dimension feature vector:

$$\mathbf{p}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,60})^T. \quad (3)$$

The whole partition points can be represented as a set of feature vectors describing its points context:

$$\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_i, \dots, \mathbf{p}_N\}. \quad (4)$$

Contour Segment: Based on partition points, the object contour is divided into N Contour Segments (CS). For each CS a 10-dimensional meaningful feature vector \mathbf{c}'_n is extracted, whereas its first element is equal to Euclidean distance

of contour segment two partition points $c'_{n,1}$. The total number of pixels in a CS determines $c'_{n,2}$. These two features are able to express how much a CS differs from a straight line. In order to distinguish contour segments of the type presented in [2], the area between the straight line connecting the CS endpoints and the contour segment itself is used as the third feature ($c'_{n,3}$). Moreover, the ratio between $c'_{n,1}$ and $c'_{n,2}$ is applied as $c'_{n,4}$ to evaluate the path variation.

Before computing remaining features, each CS is transformed into a normalised vertical orientation (i.e., so that its endpoints are vertically aligned) to ensure rotation invariance of the object contour representation (see Fig. 9). From the two possible results of such a normalising transformation, the CS with the majority of points lying on the right side of the straight line connecting its endpoints is selected for further processing. For computing further features $c'_{n,5}, c'_{n,6}, \dots, c'_{n,10}$, we use the bounding box of the whole CS as well as the three equally high sub-boxes shown in Fig. 9:

$$\begin{aligned} c'_{n,5} &= \frac{h_n}{w_n} & c'_{n,6} &= \frac{h_{n,1}}{w_{n,1}} & c'_{n,7} &= \frac{h_{n,2}}{w_{n,2}} \\ c'_{n,8} &= \frac{h_{n,3}}{w_{n,3}} & c'_{n,9} &= \frac{w_{n,3}h_{n,3}}{w_{n,1}h_{n,1}} & c'_{n,10} &= \frac{w_{n,2}h_{n,2}}{w_{n,1}h_{n,1}}. \end{aligned} \quad (5)$$

Finally, we divide the elements of the feature vector by a half of the bounding box perimeter for scale invariance:

$$\mathbf{c}_n = \frac{\mathbf{c}'_n}{w_n + h_n} = (c_{n,1}, \dots, c_{n,10})^T \quad (6)$$

The object contour can now be represented as a set of feature vectors describing its contour segments:

$$\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n, \dots, \mathbf{c}_N\}. \quad (7)$$

Finally, we fuse the partition point vectors and the contour segment vectors as the descriptor for the whole object shape contour $\partial\Omega$:

$$\partial\Omega = \{\mathbf{P}, \mathbf{C}\}. \quad (8)$$

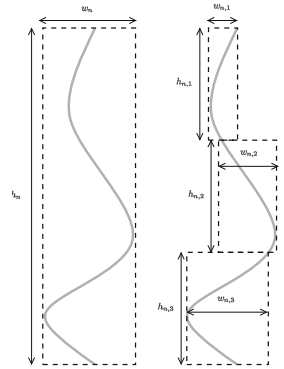


Fig. 9. CS bounding box and equally high sub-boxes ($h_{n,1} = h_{n,2} = h_{n,3}$) used for feature extraction.

4 Shape Matching

The matching of two objects including the computation of their similarity is performed separately for their partition points and contour segments representations. After matching, we estimated shape distances as the weighted sum of two terms: partition points dissimilarity, contour segments dissimilarity. For query-by-example shape retrieval, we built our approach by considering the underlying structure of the shape manifold, which is estimated from the shape distance scores between all the shapes within a database.

Point Context-Based Matching: Let $\partial\Omega$ and $\partial\Omega'$ denote two shape contours to be matched, and let \mathbf{p}_n and \mathbf{p}'_k be some partition points in \mathbf{P} and \mathbf{P}' , respectively. Let the numbers of the partition points in $\partial\Omega$ and $\partial\Omega'$ be N and K , respectively, and $N \leq K$. As point contexts are distributions represented as histograms, the dissimilarity $d(\mathbf{p}_n, \mathbf{p}'_k)$ between \mathbf{p}_n and \mathbf{p}'_k is estimated based on the χ^2 test statistic:

$$d(\mathbf{p}_n, \mathbf{p}'_k) = \frac{1}{2} \sum_{m=1}^M \frac{[p_{n,m} - p_{k,m}]^2}{p_{n,m} + p_{k,m}}. \quad (9)$$

where $M = 60$ is the dimensionality of the feature space. $h_n(m)$ and $h_k(m)$ denote the M -bin normalised histogram at \mathbf{p}_n and \mathbf{p}'_k , respectively. For two shapes contour, $\partial\Omega$ and $\partial\Omega'$, we use the lists of feature vectors for the clockwise ordered partition points. We compute all the dissimilarity between partition points and obtain a matrix:

$$D(\mathbf{P}, \mathbf{P}') = \begin{pmatrix} d(\mathbf{p}_1, \mathbf{p}'_1) & d(\mathbf{p}_1, \mathbf{p}'_2) & \dots & d(\mathbf{p}_1, \mathbf{p}'_K) \\ d(\mathbf{p}_2, \mathbf{p}'_1) & d(\mathbf{p}_2, \mathbf{p}'_2) & \dots & d(\mathbf{p}_2, \mathbf{p}'_K) \\ \vdots & \vdots & \ddots & \vdots \\ d(\mathbf{p}_N, \mathbf{p}'_1) & d(\mathbf{p}_N, \mathbf{p}'_2) & \dots & d(\mathbf{p}_N, \mathbf{p}'_K) \end{pmatrix}. \quad (10)$$

Based on (10), we find the best matched points from \mathbf{P} to \mathbf{P}' by Hungarian algorithm [12]. For each partition point \mathbf{p}_i in \mathbf{P} , the Hungarian algorithm can find its corresponding partition point \mathbf{p}'_j in \mathbf{P}' . Since $\partial\Omega$ and $\partial\Omega'$ may have different numbers of partition points, the total dissimilarity value should include the penalty for end nodes that didn't find any partner. To achieve this, we simply add additional rows with a constant value const to (10) so that $D(\mathbf{P}, \mathbf{P}')$ becomes a square matrix. The constant value const is the average of all the other values in $D(\mathbf{P}, \mathbf{P}')$.

The resulting dissimilarity values of the matched partition points can be denoted as d_1, d_2, \dots, d_N and the global partition point-based dissimilarity between \mathbf{P} and \mathbf{P}' is calculated as follows:

$$d_{\text{points}}(\mathbf{P}, \mathbf{P}') = \frac{1}{N} \sum_{n=1}^N d_n. \quad (11)$$

Contour Segment-Based Matching: Similar to point context-based matching, Let $\partial\Omega$ and $\partial\Omega'$ denote two shape contours to be matched, and let \mathbf{c}_n and \mathbf{c}'_k be some contour segments in \mathbf{C} and \mathbf{C}' , respectively. The numbers of contour segments is equal to the number of partition points in $\partial\Omega$ and $\partial\Omega'$, be N and K , respectively, and $N \leq K$.

We introduce a matching cost measure for contour segments \mathbf{c}_n and \mathbf{c}'_k belonging to different shape contour $\partial\Omega$ and $\partial\Omega'$:

$$d(\mathbf{c}_n, \mathbf{c}'_k) = \frac{1}{M} \sum_{m=1}^M \frac{\sigma_m |c_{n,m} - c'_{k,m}|}{\sum_{j=1}^K |c_{n,m} - c'_{j,m}|}, \quad (12)$$

where $M = 10$ is the dimensionality of the feature space and σ_m is the weight for each feature achieved in an optimisation process explained in [2]. As one can see, the dissimilarity value between \mathbf{c}_n and \mathbf{c}'_k does not only depend on these two contour segments. All CS of \mathcal{C} are taken into consideration, whereby (12) does not fulfil the symmetry property $d(\mathbf{c}_n, \mathbf{c}'_k) \neq d(\mathbf{c}'_k, \mathbf{c}_n)$. However, it behaves in accordance to human perception. If the matching cost of \mathbf{c}_n to the neighbours of \mathbf{c}'_k in \mathcal{C}' decreases, $d(\mathbf{c}_n, \mathbf{c}'_k)$ increases.

We arrange the contour segments of both objects in a clockwise way so that the objects can be represented by ordered lists of feature vectors:

$$\begin{aligned} \mathcal{C} &= (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n, \dots, \mathbf{c}_N) \\ \mathcal{C}' &= (\mathbf{c}'_1, \mathbf{c}'_2, \dots, \mathbf{c}'_k, \dots, \mathbf{c}'_K) \end{aligned} \quad (13)$$

Using (12) we generate a matrix of matching cost between all CS in \mathcal{C} and in \mathcal{C}' :

$$D(\mathcal{C}, \mathcal{C}') = \begin{pmatrix} d(\mathbf{c}_1, \mathbf{c}'_1) & d(\mathbf{c}_1, \mathbf{c}'_2) & \dots & d(\mathbf{c}_1, \mathbf{c}'_K) \\ d(\mathbf{c}_2, \mathbf{c}'_1) & d(\mathbf{c}_2, \mathbf{c}'_2) & \dots & d(\mathbf{c}_2, \mathbf{c}'_K) \\ \vdots & \vdots & \ddots & \vdots \\ d(\mathbf{c}_N, \mathbf{c}'_1) & d(\mathbf{c}_N, \mathbf{c}'_2) & \dots & d(\mathbf{c}_N, \mathbf{c}'_K) \end{pmatrix}. \quad (14)$$

The next steps are the same as partition points. In order to find an optimum match of contour segments from \mathcal{C} to CS from \mathcal{C}' , we apply the Hungarian algorithm for the matrix expressed in (14). Then we can find the resulting dissimilarity values of the matched contour segments denoted as d_1, d_2, \dots, d_N and the global dissimilarity $d_{\text{contours}}(\mathcal{C}, \mathcal{C}')$ is calculated by the same method in (11).

Shape Distance and Retrieval: In this paper, we defined the distance between two shapes with a fused scheme between partition points and contour segments:

$$d(\Omega, \Omega') = \alpha d_{\text{points}}(\mathbf{P}, \mathbf{P}') + (1 - \alpha) d_{\text{contours}}(\mathcal{C}, \mathcal{C}'). \quad (15)$$

The weight α in (15) has been optimised by an integrated Gradient Hill Climbing [7] and Simulated Annealing [8] method on the Kimia216 dataset. By experiment, the optimised weight is $\alpha = 0.7$ and we employ it for all experiments in Sect. 5.

For the task of shape retrieval, given a set of H shapes, a shape matching algorithm (15) can be applied to obtain a $H \times H$ distance matrix to describe the pairwise relations between all shapes in terms of a dissimilarity measure. Such an approach ignores the fact that also distances to all other shapes contain important information about the overall shape manifold. Therefore, in this paper, we employ a method proposed by [21] that allows to improve retrieval results by analysing the underlying structure of the shape manifold. This method captures the manifold structure of the data by defining a neighbourhood for each data point in terms of a modified version of a mutual k NN graph which yields improved performance on all experiment databases in Sect. 5.

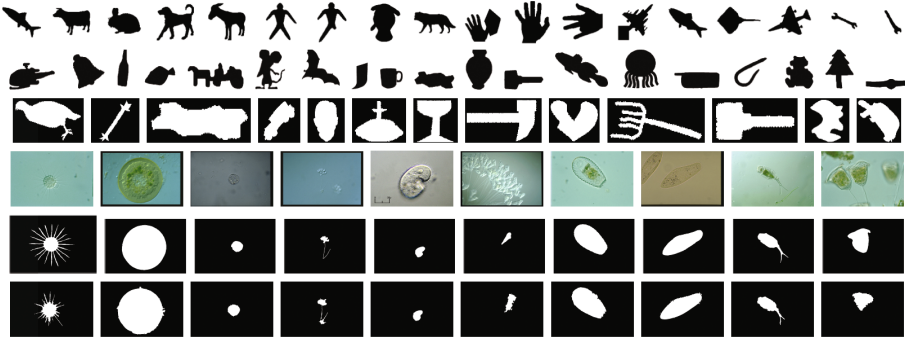


Fig. 10. Example shapes from the experimental datasets: Kimia-99 (first row), MPEG-400 (second row), Kimia-216 (third row) and EM-200 (fourth row with original, fifth row with manually- and the sixth row with semi-automatically segmented images).

5 Experiments and Results

Correspondence Matching: In Fig. 11 (left), we match a dog to another dog with legs and backside deformation. Since the articulations of the query dog is similar to the deformed one, our matching process finds the correct correspondences between partition points as well as contour segments. Figure 11 (right) shows the correspondence between two tools. Based on human perception there are two mismatched partition points. This is due to their symmetric silhouette and there are many similar contour segments and partition points in one object which could affect the performance of matching approach. However, in terms of the geometry, the correspondences are correct since the matched partition points and contour segments have the same geometrical location among their objects. Therefore, the final similarity between two objects is not affected. Figure 12 are taken from Kimia-99’s person dataset, whereas the silhouette on the right side is manually modified. It demonstrates that the proposed method is able to establish a correct correspondence if parts are substantially deformed. The obtained correspondence in Kimia-99 [22] illustrates that our matching process has strong performance in the presence of occlusion.

Application Independent Experiments: We tested our algorithm on two shape databases provided by Kimia-216 [18] (the fourth row in Fig. 10) and MPEG-400 [2] (the third row in Fig. 10). We compare our method to the Path Similarity Skeleton Graph Matching (PSSGM) [18] and the Contour Segments-based Matching [2] in which contour segments are generated by the endpoints of skeletons. To compare our partition point extraction technique against existing methods, we also tested our algorithm with partition points gained from Discrete Curve Evolution [23]. To get the best results we set the stop parameter at 4. For each of the shapes used as a query, we have checked whether the retrieved results are correct, i.e., belong to the same class as the query. In order to enable

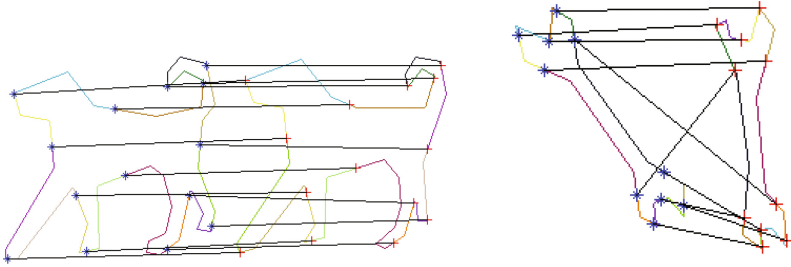


Fig. 11. The corresponding partition points (linked with lines) and contour segments (linked with colours) between two dogs (left) and tools (right) from Kimia99 dataset.

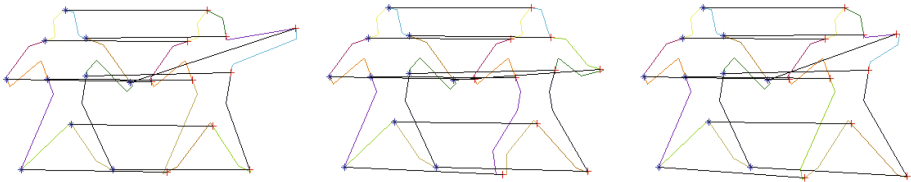


Fig. 12. Example correspondence between articulating persons.

quantitative comparison, we have kept the experimental convention proposed in [18] and considered the 10 best matches for each query. Results achieved for the Kimia-216 and the MPEG-400 datasets can be found in Table 1, whereas in [24] the PSSGM algorithm had been re-evaluated without any preliminary assumptions regarding the object skeletonisation. For these two databases, it is important to observe that the fused point context and contour segment outperforms skeleton-based methods [18] and contour segments-based methods [2] while the complexity of feature generation and matching has been reduced.

Environmental Microorganism Retrieval: EMs and their species are very important indicators to evaluate environmental quality, but their manual recognition is very time-consuming [25]. Thus, automatic analysis techniques for microscopic images of EMs would be very appreciated by environmental scientists. We have tested our methodology for this application using a real dataset acquired in environmental laboratories of the University of Science and Technology Beijing. We segmented the images manually and semi-automatically with the method introduced in [26]. Examples of EM original images as well as manually segmented and semi-automatically segmented EM images can be seen in Fig. 10. The impressive results for the EM-200 [27] dataset (see Table 1) confirm the high descriptive power of our new feature extraction technique for contours and prove the applicability of our shape retrieval algorithm to real-world applications.

Table 1. Experimental comparison of our methodology to the most related algorithm using the MPEG-400, Kimia216 datasets as well as the proof of applicability of our approach to real world problems using the EM-200 dataset. Results are summarised as the number of shapes from the same class among the first top 1–10 shapes.

Object Retrieval for MPEG-400										
Algorithm	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
PSSGM [24]	380	371	361	351	344	339	332	320	330	309
Contours [2]	375	348	333	325	317	311	300	295	276	275
DCE [23] + CS [2]	375	368	346	337	338	323	308	297	286	276
Our method	389	374	368	368	358	347	344	339	346	330

Object Retrieval for Kimia-216										
Algorithm	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
PSSGM [24]	205	208	202	199	200	192	184	167	161	130
Contours [2]	216	215	206	204	200	186	172	163	130	124
DCE [23] + CS [2]	216	204	197	185	175	162	154	154	142	131
Our method	216	214	207	204	201	204	191	188	192	185

Object Retrieval for EM-200										
Algorithm	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Manually	200	183	175	167	166	156	154	146	156	153
Semi-Automatically	200	180	174	163	160	160	150	143	153	147

6 Conclusion and Future Work

In this paper, we proposed a method for 2D shape similarity measure based on contour segment matching and, after fusion with a partition point matching, use it for object retrieval. The most innovative part of our approach is the robust and stable partition point generation and the comparison and matching of contour segments. The algorithm can be easily implemented to real applications. In the future, we will investigate possibilities of weighting properties on contour segment feature, which could be applied for dataset adopting with a supervised optimisation scheme.

Acknowledgement. Research activities leading to this work have been supported by the China Scholarship Council, the German Research Foundation (DFG) within the Research Training Group 1564. We greatly thank M.Sc Chen Li from the University of Siegen for providing us with the Environmental Microorganism image dataset for experiments. We are also very grateful to Dr. Kimiaki Shirahama from University of Siegen for his guiding on significant technologies.

References

1. Biederman, I., Ju, G.: Surface versus edge-based determinants of visual recognition. *Cogn. Psychol.* **20**, 38–64 (1988)
2. Yang, C., Tiede, O., Pietsch, P., Feinen, C., Kelter, U., Grzegorzec, M.: Shape-based object retrieval by contour segment matching. In: *International Conference on Image Processing (ICIP)*. IEEE Computer Society (2014, to appear)
3. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**, 1270–1281 (2008)
4. Bai, X., Latecki, L., Yu Liu, W.: Skeleton pruning by contour partitioning with discrete curve evolution. *PAMI* **29**, 449–462 (2007)
5. Bai, X., Liu, W., Tu, Z.: Integrating contour and skeleton for shape classification. In: *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 360–367 (2009)
6. Ma, T., Latecki, L.: From partial shape matching through local deformation to robust global shape similarity for object detection. In: *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1441–1448 (2011)
7. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 3rd edn. Prentice Hall Press, Englewood Cliffs (2009)
8. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**, 671–680 (1983)
9. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. *ICCV* **1**, 503–510 (2005)
10. Nguyen, D.T., Ogunbona, P.O., Li, W.: A novel shape-based non-redundant local binary pattern descriptor for object detection. *Pattern Recognit.* **46**, 1485–1500 (2013)
11. Cao, Y., Zhang, Z., Czogiel, I., Dryden, I., Wang, S.: 2d nonrigid partial shape matching using mcmc and contour subdivision. In: *CVPR*, pp. 2345–2352 (2011)
12. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 509–522 (2002)
13. Wang, X., Feng, B., Bai, X., Liu, W., Latecki, L.J.: Bag of contour fragments for robust shape classification, pp. 2116–2125 (2014)
14. Yang, X., Bai, X., Yang, X., Zeng, L.: An efficient quick algorithm for computing stable skeletons. In: *2nd International Congress on Image and Signal Processing, CISP 2009*, pp. 1–5 (2009)
15. Goh, W.B.: Strategies for shape matching using skeletons. *CVIU* **110**, 326–345 (2008)
16. Sebastian, T., Kimia, B.: Curves vs skeletons in object recognition. *ICIP* **3**, 22–25 (2001)
17. Baseski, E., Erdem, A., Tari, S.: Dissimilarity between two skeletal trees in a context. *Pattern Recognit.* **42**, 370–385 (2009)
18. Bai, X., Latecki, L.: Path similarity skeleton graph matching. *PAMI* **30**, 1282–1292 (2008)
19. Zeng, J., Lakaemper, R., Yang, X., Li, X.: 2d shape decomposition based on combined skeleton-boundary features. In: *Bebis, G., et al. (eds.) Advances in Visual Computing*. LNCS, vol. 5359. Springer, Heidelberg (2008)
20. Hassouna, M.S., Farag, A.A.: MultiStencils fast marching methods: a highly accurate solution to the eikonal equation on cartesian domains. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 1563–1574 (2007)

21. Kontschieder, P., Donoser, M., Bischof, H.: Beyond pairwise shape similarity analysis. In: Zha, H., Taniguchi, R., Maybank, S. (eds.) *Computer Vision - ACCV 2009*. LNCS, pp. 655–666. Springer, Heidelberg (2010)
22. Sebastian, T., Klein, P., Kimia, B.: Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 550–571 (2004)
23. Latecki, L., Lakamper, R.: Convexity rule for shape decomposition based on discrete contour evolution. *Comput. Vis. Image Underst.* **73**, 441–454 (1999)
24. Hedrich, J., Yang, C., Feinen, C., Schaefer, S., Paulus, D., Grzegorzec, M.: Extended investigations on skeleton graph matching for object recognition. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierok, A. (eds.) *CORES 2013. AISC*, vol. 226, pp. 371–381. Springer, Heidelberg (2013)
25. Li, C., Shirahama, K., Grzegorzec, M., Ma, F., Zhou, B.: Classification of environmental microorganisms in microscopic images using shape features and support vector machines. In: *ICIP*, pp. 2435–2439. IEEE Computer Society (2013)
26. Li, C., Shirahama, K., Czajkowska, J., Grzegorzec, M., Ma, F., Zhou, B.: A multi-stage approach for automatic classification of environmental microorganisms. In: *IPCV*, pp. 364–370. CSREA Press (2013)
27. Yang, C., Li, C., Tiebe, O., Shirahama, K., Grzegorzec, M.: Shape-based classification of environmental microorganisms. In: *International Conference on Pattern Recognition (ICPR)*, pp. 3374–3379. IEEE Computer Society (2014)