Pattern Recognition Letters
journal homepage: www.elsevier.com

Shape-based object matching using interesting points and high-order graphs

Cong Yang[**], Christian Feinen, Oliver Tiebe, Kimiaki Shirahama, Marcin Grzegorzek

Pattern Recognition Group, University of Siegen, Germany

ABSTRACT

In shape-based object matching, it is important how to fuse similarities between points on a shape contour and the ones on another contour into the overall similarity. However, existing methods face two critical problems. Firstly, since most contour points are involved for possible matchings without taking into account the usefulness of each point, it causes high computational costs for point matching. Secondly, existing methods do not consider geometrical relations characterised by multiple points. In this paper, we propose a shape-based object matching method which is able to overcome these problems. To counteract the first problem mentioned, we devise a shape descriptor using a small number of interesting points which are generated by considering both curvatures and the overall shape trend. We also introduce a simple and highly discriminative point descriptor, namely Point Context, which represents the geometrical and topological location of each interesting point. For the second problem, we employ high-order graph matching which examines similarities for singleton, pairwise and triple relations of points. We validate the robustness and accuracy of our method through a series of experiments on six datasets.

1. Introduction

Object matching is a process for identifying a specific object in a digital image or video. As an expressive abstraction of the visual pattern of an object, shape is being used in many applications to understand or identify objects in images [1, 2]. Among many existing shape-based object matching methods [3], one of the most popular approaches is contour-based shape matching consisting of the following two processes. The first process is shape representation where the contour of a shape is represented using a set of descriptors, each of which describes the characteristics of a point on the contour (for short 'contour point'). The second process is shape matching which extracts correspondences among contour points in two shapes by comparing descriptors of those points. The overall similarity is computed by aggregating similarities between corresponding points in terms of their descriptors.

For the shape representation, there are three main challenges. The first challenge is how to extract efficient descriptors that are invariant to shape rotation, translation and scaling. The second one is how to extract shape descriptors that are robust to noise and distortions. This is known as the robustness requirement. The third challenge is how to generate descriptors with a low computation complexity. In order to solve these problems, one possible way is to use the simple geometry descriptors such as [4, 3], since they have low computational complexity and most of them are robust to noise and distortions. However, these descriptors have limited description power due to the lack of information. Another possibility is to use some rich descriptors [5, 6, 7, 8] which have higher description power. Although most of them are invariant to shape rotation, translation and scaling, they need to sample many contour points to precisely represent its characteristics. Hence, using rich descriptors incurs high computational costs for extraction and matching.

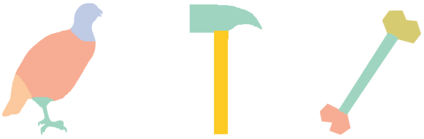
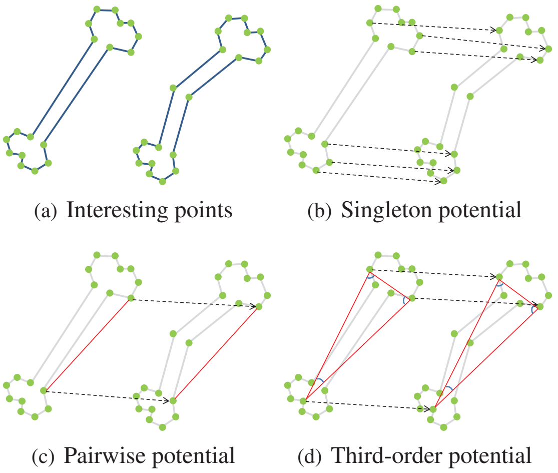


Fig. 1: Shapes are composed by different regions.

We assume that the main reason for the high computation costs is the lack of consideration of how useful each contour point is. Therefore, we represent the shape with only a small number of interesting points, each of which is defined as a contour point that represents a rigid region of a shape. More specifically, shapes are normally composed of different regions (Figure 1) and some regions are likely to be deformed when some changes occur in the shape. However, some regions are resilient

[**]Corresponding author: Tel.: +49-271-740-4088; fax: +49-271-740-3972; *e-mail:* cong.yang@uni-siegen.de (Cong Yang)

against shape deformations like the bird's head, bone ends or the handle of a hammer in Figure 1. We regard such a region as rigid, and model it as a region which is deviated from the overall shape trend. As interesting points are mainly detected in the rigid regions, they are robust to shape deformation. For example, in Figure 2(a), the two end regions of a bone are more stable than its middle part, and our method can ensure that the locations of those interesting points are more stable and similar in the same class rather than different classes. Based on the interesting points, we can dramatically reduce the computation costs for descriptor generation and matching.


(a) Interesting points (b) Singleton potential

(c) Pairwise potential (d) Third-order potential

Fig. 2: Different potentials for object matching.

Since we only use a limited number of interesting points to represent the whole shape, each interesting point should carry both global and local shape features. Therefore, for each interesting point, a simple and intuitive descriptor, namely Point Context, is proposed to capture its geometrical and topological features. Specifically, a point context is generated by considering both lengths and orientations from each interesting point to other contour points. This descriptor expresses the configuration of the entire shape relative to the interesting point. In addition, by normalising point contexts in terms of lengths and orientations, they become invariant to shape rotation and translation. Thanks to such point contexts integrated with interesting points, our method features a high discriminative power while keeping low computational complexity.

For the shape matching process, even using the proposed point contexts, it is inevitable that several interesting points are falsely matched (the right bone in Figure 2(a)). Moreover, some interesting points in the same object may have very similar geometrical locations because of their small distances or symmetry. To overcome this, one possible way is to avoid those points which are below a threshold of similarity value during the correspondence-based matching [9, 10, 11]. Another way is to jump over those similar points by adding dummy points in another object [5, 12]. However, even these methods only consider the relationship between single points. This could lead to ambiguous matching because many different points may have similar descriptors [13, 2].

Compared to the above-mentioned methods, we aim to consider the geometric relations among multiple points using high-order graph matching, which is an approach to match two graphs by extracting the correspondences of multiple nodes [14]. We adopt this approach by considering nodes as interesting points described by point contexts. As shown in Figure 2(b), singleton point matching is a well-known assignment problem where the interesting point is matched with one point in another shape. For the pairwise matching (Figure 2(c)), it finds consistent correspondences between two pairs of interesting points by taking into consideration both how well their descriptors match and how similar their pairwise geometric relations are. For the high-order matching (mostly third-order, see Figure 2(d)), it considers the cost of matching three correspondences. More specifically, a triple of interesting points in a shape are matched with the one in another shape. With this observation, we propose a high-order graph matching strategy for improving the extraction of correspondences between interesting points.

The main contributions of this article include (i) the introduction of a novel shape descriptor with robust interesting points and their point context descriptors, (ii) the implementation of a high-order graph matching algorithm that solves the shape matching problem and (iii) the design of potential functions for different orders. Our method is validated through a series of object retrieval experiments on five datasets demonstrating its robustness and accuracy.

2. Related Work

Shape descriptors generally look for effective and perceptually important shape features. Common simple descriptors like area, circularity, etc. can only discriminate shapes with large differences [3]. They are frequently used as filters or combined with other richer shape descriptors to enhance their discrimination power.

For some richer descriptors, the skeleton-based method [5] normally generates the shape descriptor through some skeletonisation and pruning methods to achieve a visually promising skeleton [15]. Shape Invariants [7] is a shape descriptor based on integral kernels. It describes a shape in an implicit form and is characterised by a series of isotropic kernels that provide desirable invariance properties. Shape Context [6] is a descriptor of contour points with histograms in which bins are uniformly divided log-polar space. Since those methods do not know which contour point is useful for matching, they need to use a large number of contour points to achieve accurate correspondences and alignments. Thus, all of the above-mentioned descriptors incur high computational complexity. In this paper, we propose to use only a limited number of interesting points. These and their associated point contexts can efficiently reduce the matching complexity while keeping the crucial shape geometry and topology.

Other descriptors like Discrete Curve Evolution [16] can simplify the original shape into a hierarchy of polygons, and vertexes of each polygon at a layer are defined as interest points. However, since these interest points highly rely on the parameters which control the degree of shape simplification, the method needs manual parameter tuning to achieve accurate matching. Liu et al. proposed some methods [17, 18] to detect the corner points by visual curvatures under different scales. Though those corner points are robust to fine-grained deformations and can be used as the interesting points, this method also requires a curvature threshold for point detection. Moreover, the proposed interesting point detection method is detected by considering both curvatures and overall shape trend.

For shape matching, Hausdorff distance [19] is a classical matching method where the distance of two point sets is normally calculated by both the maximal and minimal distance between point pairs. Hence, this method is sensitive to noise and slight variations. Belongie et al. proposed a correspondence-based shape matching method using shape contexts [6] wherein the matching of two shapes is done by matching their point histograms. Bai et al. proposed a skeleton-based shape matching method which uses the Hungarian algorithm to find the best match of skeleton endpoints in terms of their geodesic paths [5]. However, for all these matching methods, they only consider the singleton constraints among each corresponding points. In order to improve the accuracy, Leordeanu et al. proposed a spectral technique for matching problems using pairwise constraints [20] where the correspondence is established through preserving the structure similarity across two point sets. Nevertheless, this strategy could lead to substantial loss of performance since every pair of interesting points trivially defines a line which is repetitive and similar to each other. Therefore, instead of the singleton or pairwise, Zass et al. proposed to match interesting points using hyper-graphs [21] which are going beyond the pairwise. Specifically, each interesting point set is modelled by a hyper-graph where the relations between points are represented by hyper-edges. A match between the interesting point sets is then modelled as a hyper-graph matching problem. Due to the theoretical advance and empirical success, hyper-graph matching has attracted increasing attention and many methods have been proposed [22, 23, 24, 25] and the references therein. However, it is unclear which potential functions are suitable for our interesting point matching tasks. Moreover, it is also interesting to explore the performances of different hyper-graph matching algorithms. Thus, we propose the singleton, pairwise and third-order potential functions for the 2D interesting point matching problem. In addition, we compare the matching performance between the employed and other hyper-graph matching algorithms.

3. Shape Descriptor

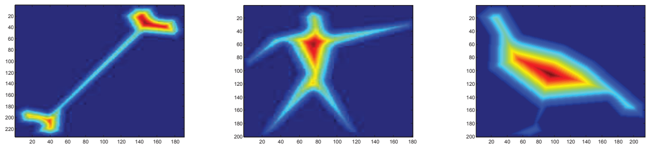
In this section, we first describe our method which generates robust interesting points along the shape boundary. After that, point context is introduced and analysed.

3.1. Interesting Points

We consider the problem of selecting a set of interesting points $\{\boldsymbol{p}_i\}_{i=1}^{N}$ from a given shape Ω. We assume that distinctive contours like the legs or the tail of an elephant are characterised by a high curvature towards the overall shape trend. Based on this idea, we compute the distance between each single contour point and its closest reference point. Here, a reference point is a point that is inside the shape and characterised with the highest distance to the contour. By arranging these values sequentially, a sequence $\boldsymbol{s}$ is generated where interest points characterised by high curvatures are detected as peaks.

Since noises on a contour could have adverse influence on interesting point detection, we first perform polygonisation to suppress noises without removing significant parts of the contour. For this purpose, the well-known *Douglas-Peucker* technique [26] is recursively applied to the object's contour. Then, the contour is converted into a polygon $\mathcal{P}$.

Having $\mathcal{P}$, the reference points $\boldsymbol{x}_i$ (residing inside the shape) can be detected by utilising a *fast marching method* (FMM) [27]. We employ FMM for reference point detection since it is convenient to use while delivering a high accuracy result. The approach is initialised with the whole polygon (interpreted as a set of points) leading to a distance map T. This map provides the distance from each pixel to the closet contour point on $\mathcal{P}$, as shown in Figure 3.



(a) Bone (b) Person (c) Bird

Fig. 3: The distance map T of a bone, a person and a bird.

Reference points are then discovered at maximum value locations in T. To find reference points that appropriately cover the whole shape, we iteratively mask T and detect regions from which reference points are extracted. This masking is carried out using the following dynamically adapted threshold $\mu^{(BG)}$:

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

where $\phi(\cdot)$ returns the maximum value inside T restricted to the area of Ω and $\psi(\cdot)$ indicates the standard derivation. Here, another option is to use $\phi(\cdot)$ mean instead of maximum as maximum is an unstable statistic compared to mean. However, it leads to less stability in interesting point detection and worse shape retrieval results in our preliminary experiments. The main reason is that reference points should be discovered at the maximum value in the most prominent regions in T. If $\mu^{(BG)}$ is assigned with the $\phi(\cdot)$ mean value, some irrelevant regions could be involved due to the low threshold $\mu^{(BG)}$. By masking T with $\mu^{(BG)}$ in Eq. 1, we can emphasise the most prominent regions for which reference points should be obtained while removing the other irrelevant regions. In particular, the former regions are clustered into disjoint regions $\boldsymbol{A}_i$. For each of these regions, a reference point is determined as a weighted centroid where the weight of a contour point is its value in T.

All input $T_i^{t=1}$ (where $i = [1, 2, \ldots, |C^{t-1}|]$) are multiplied with the original $T^{t=0}$. Consequently, $t = 0$ detected area is alleviated while new ones are stressed. This pixelwise multiplication, depicted as $T^{t=1} \circ T^{t=0}$. A new threshold $\mu^{(BG),t=1}$ is computed using $T^{t=1} \circ T^{t=0}$, so that a new region $A_i^{t=1}$ and the corresponding reference point $C^{t=1}$ (i.e. weighted centroid) is detected. The algorithm terminates if $C^t = \emptyset$ and the final set is generated by $\boldsymbol{R} = \bigcup_{j=0}^{t} C^{t=j}$.

Finally, we conduct the following two types of filtering to eliminate meaningless reference points. First, none of the reference points are allowed to be too close to the shape's centroid, which is ensured by the threshold $\mu^{(cog)}$. If only one reference point $\boldsymbol{x}_i \in \boldsymbol{R}$ is violating this constraint, all reference points are discarded except for the responsible one. Second, in presence of multiple reference points, the contour has to be split into subparts. During this separation, the algorithm monitors that each $\boldsymbol{x}_i$ is only assigned to one region. If one or two regions are assigned, the point will be removed. This strategy can ensure $\boldsymbol{x}_i$ only represents one region of the shape. Moreover, it is easier to sample the contour points sequentially in the next steps.

We now have $\boldsymbol{R}$ which is the set of reference points that passed the above-mentioned filtering. Using $\boldsymbol{R}$, we aim to compute the distance between each contour point on the shape and its closest reference point. This is done by FMM which uses $\boldsymbol{R}$ as seed points and outputs $T^{(final)}$. By sequentially aggregating values corresponding to contour points in $T^{(final)}$, we obtain a sequence $\boldsymbol{s}$ where each element represents the distance between a contour point and its nearest reference point. Based on $\boldsymbol{s}$, we can extract interesting points which have a high influence on the perceptual appearance of the shape. Figure 4 shows both the $T^{(final)}$ values of the contour and their signal plots. Please notice that the bone structure yields two reference points so that the contour has been separated into two parts (first and second column).

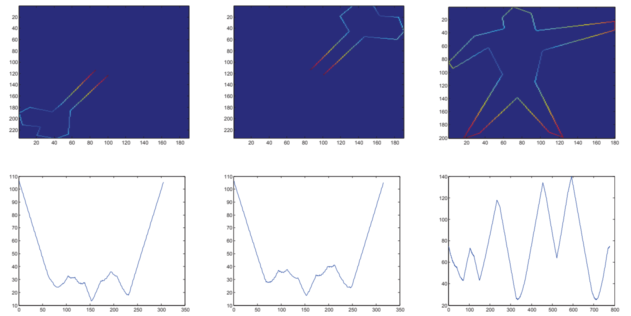


Fig. 4: Example of sequences representing distances between contour points and their closest reference points. The colours on the contour line encode the distance from a pixel to its closest reference point.

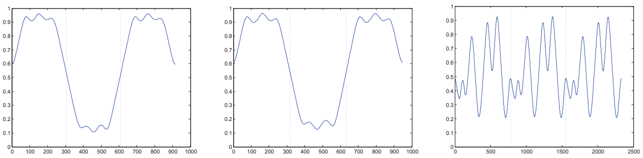


Fig. 5: Sequences obtained by smoothing sequences in Figure 4 (each smoothed sequence corresponds to the middle part of the sequence).

It is obvious that the noise-to-signal ratio degrades the peak detection with imprecise interesting point detection. Thus, $\boldsymbol{s}$ is smoothed using a Low Pass Filter (LPF) based on a Fast Fourier Transform (FFT). Figure 5 shows sequences smoothed by applying LPF to the ones in Figure 4. Here, the LPF is established with a Gaussian coefficient mask that is applied to the zero-shifted frequency domain.

Before LPF, the sequence is padded at the start and at the end to alleviate border artifacts. This padding is needed for shapes that are separated into several parts like the bone in Figure 4. For such a shape, without padding, contour points at boundaries of parts could be falsely detected as interesting.

Finally, we detect interesting points as peaks in the smoothed sequence $\hat{\boldsymbol{s}}$. Each peak is identified as a point where the first-order derivative ($\hat{\boldsymbol{s}}'$) of $\hat{\boldsymbol{s}}$ is zero, and the second-order ($\hat{\boldsymbol{s}}''$) is positive or negative. Please note that the computation of the second-order derivative only considers a signed binary version of the first-order derivative (sign($\hat{\boldsymbol{s}}'$), where sign($\cdot$) returns 0 if $\hat{s}'_i = 0$, 1 if $\hat{s}'_i > 0$ and -1 otherwise). As shown in Figure 6, the naive approach which locates interesting points based on $\hat{\boldsymbol{s}}'' = 0$ may cause several false-positives that multiple interesting points are located within a very short distance. To avoid such false-positives, a distance threshold $\mu^{(sdiff)}$ is applied to $\hat{\boldsymbol{s}}''$. If the distance between adjacent peaks is below $\mu^{(sdiff)}$ coupled with a low height difference (taken from $\hat{\boldsymbol{s}}$), the power of the LPF is dynamically decreased. With the lower smoothing power, the procedure is repeated with the original sequence $\boldsymbol{s}$ until the peak distance constraint is fulfilled. Finally, based on these validated peaks, we can extract interesting points which highly indicate shape characteristics.

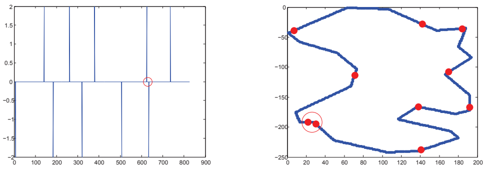


Fig. 6: A situation where the sigma of the Gaussian filter has not been chosen appropriately. It is obvious that these artifacts can be easily determined by analysing the second-order derivative. The red circles indicate the problem.

3.2. Point Context

Inspired by [28, 29], we propose the point context descriptor which represents each interesting point $\boldsymbol{p}_i, (i = 1, 2, ..., m)$ based on its geometrical and topological location. We consider the set of vectors originating from $\boldsymbol{p}_i$ to all other sample points on a shape contour. These vectors express the configuration of the entire shape relative to $\boldsymbol{p}_i$. Theoretically, instead of contour sample points, we can form vectors only using interesting points. However, we will not employ this strategy for the following reasons: (1) It will lose some coarse- and fine-grained features since the number of interesting points is limited. (2) It will reduce the robustness of point context since our method may fail to locate interesting points on some characteristic contours. This causes the dramatic change of the point context. (3) The major computation time derives from matching interesting points rather than the vector computation in point context extraction. Therefore, the difference of computation time between the two strategies can be ignored.

Let P denote a sequences of interesting points $P = \{\boldsymbol{p}_1, \cdots \boldsymbol{p}_m\}$ and Q denotes a finite number of contour sample points $Q = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_n\}$, $P \notin Q$. All points in P and Q are represented by their coordinate locations. Points in Q are ordered clockwise along the shape contour. For $\boldsymbol{p}_i$, we compute two vectors, one presenting the distance of $\boldsymbol{p}_i$ to each $\boldsymbol{q}_k \in Q(k = 1, \cdots, n)$, and the second representing the orientation of the vector from $\boldsymbol{p}_i$ to $\boldsymbol{q}_k$. A distance $D^{\boldsymbol{p}_i}(k)$ from $\boldsymbol{p}_i$ to $\boldsymbol{q}_k$ is defined as Euclidean distance in the log space

$$D^{\boldsymbol{p}_i}(k) = \log(1 + \|\overrightarrow{\boldsymbol{p}_i} - \overrightarrow{\boldsymbol{q}_k}\|^2) \quad . \tag{2}$$

In order to avoid the divergence of log, we add one to the Euclidean distance. An orientation $\Theta^{\boldsymbol{p}_i}(k)$ from $\boldsymbol{p}_i$ to $\boldsymbol{q}_k$ is defined as the orientation of vector $\overrightarrow{\boldsymbol{p}_i} - \overrightarrow{\boldsymbol{q}_k}$:

$$\Theta^{\boldsymbol{p}_i}(k) = \text{atan2}(\overrightarrow{\boldsymbol{p}_i} - \overrightarrow{\boldsymbol{q}_k}) \quad . \tag{3}$$

where atan2 stands for the four quadrant inverse tangent which can ensure $\Theta^{\boldsymbol{p}_i}(k) \in [-\pi, \pi]$. Together with the distances, a single interesting point $\boldsymbol{p}_i$ is encoded as two n-dimensional vectors $D^{\boldsymbol{p}_i}$ and $\Theta^{\boldsymbol{p}_i}$.

The proposed point descriptor is different from the methods in [13] and [6]. Firstly, we only consider the feature vectors on the basis of interesting points instead of uniformly or randomly selecting sample points. This strategy can reduce the mismatches and computational complexity conspicuously. Secondly, the proposed point descriptor is naturally translation and scaling invariant since the distance between point contexts is computed by normalising $D^{\boldsymbol{p}_i}(k)$ and $\Theta^{\boldsymbol{p}_i}(k)$ (see Eq. 8). In addition, we generate the point context features by the Euclidean distance and the four quadrant inverse tangent methods; their values remain the same even a shape is rotated. Thus, the proposed descriptor is also rotation invariant. On the contrary, approaches in [13, 6] are not intrinsically rotation invariant because each point is characterised by the tangent angle which is ineffective for some points for which no reliable tangent can be computed.

Finally, given an arbitrary shape Ω, its contour $\partial\Omega$ can be represented with the locations as well as the distance and orientation vectors of all contour interesting points:

$$\partial\Omega = \{\boldsymbol{p}_i, D^{\boldsymbol{p}_i}, \Theta^{\boldsymbol{p}_i}\} \quad . \tag{4}$$

4. Shape Matching Based on High-order Graph Matching

In this section, using interesting points described by point contexts, we firstly formulate shape matching as high-order graph matching consisting of potential functions with different orders. Then, we introduce the definition of each potential function. Finally, we explain a method which can efficiently find the optimal matching on high-order graphs.

4.1. Formulation

Let P_1 and P_2 denote sets of interesting points from two shapes S_1 and S_2 respectively. $\boldsymbol{p}_i$ and $\boldsymbol{p}'_j$ denote a single interesting point in P_1 and P_2 respectively. $P \triangleq P_1 \times P_2$ denotes the set of possible correspondences. We define the following boolean indicator:

$$x_a = \begin{cases} 1 & \text{if } a = (\boldsymbol{p}_i, \boldsymbol{p}'_j) \in P \text{ is a correspondence} \\ 0 & \text{otherwise} \end{cases} \quad . \tag{5}$$

In our definition, a basic constraint is that each point $\boldsymbol{p}_i$ in P_1 is mapped to at most one point $\boldsymbol{p}'_j$ in P_2, while for each point $\boldsymbol{p}'_j$ in P_2 there is at most one point $\boldsymbol{p}_i$ in P_1 mapped to it. Therefore, we have the set of constraints:

$$\begin{aligned} \zeta = \{\mathbf{x} \in \{0,1\}^{P_1 \times P_2} \mid \sum_{\boldsymbol{p}_i \in P_1} x_{\boldsymbol{p}_i, \boldsymbol{p}'_j} \leqslant 1, \sum_{\boldsymbol{p}'_j \in P_2} x_{\boldsymbol{p}_i, \boldsymbol{p}'_j} \leqslant 1, \\ \forall \boldsymbol{p}_i \in P_1 \text{ and } \forall \boldsymbol{p}'_j \in P_2\} \end{aligned} \quad . \tag{6}$$

Inspired by [24, 30], our high-order (degree 3) matching formulation is formulated as the following optimisation problem:

$$\begin{aligned} \min_{\mathbf{x} \in \zeta} \{E(\mathbf{x}|\theta) = \sum_{a \in P} \theta_a x_a + \sum_{(a,b) \in P \times P} \theta_{ab} x_a x_b + \\ \sum_{(a,b,c) \in P \times P \times P} \theta_{abc} x_a x_b x_c\} \end{aligned} \quad . \tag{7}$$

where θ is the whole set of matching costs that we consider and consists of the following three components: θ_a is the matching cost for each correspondence $a \in P$ (Figure 2(b)), θ_{ab} for a pair of correspondences $(a, b) \in P \times P$ (Figure 2(c)), and θ_{abc} for a triplet of correspondences $(a, b, c) \in P \times P \times P$ (Figure 2(d)). Since the matching constraint in Eq. 6 makes the optimisation problem in Eq. 7 difficult to solve, we will introduce a method that decomposes the problem in Eq. 7 into several sub-problems in Section 4.3.

4.2. Potential Functions

We only consider the first and third order terms for the following reasons. Firstly, although singleton potential causes mis-matching of interesting points due to the lack of their topological relations, they still offer the major contribution to examine overall shape characteristics. Secondly, since we already consider point contexts of interesting points in singleton potential, it is redundant to consider them for pairwise potentials. Thus, we adopt to define pairwise potentials based only on relative location relations of interesting points. However, our preliminary experiment showed that such pairwise potentials have low discriminative power, and many different pairs of points have similar descriptors. Hence, we have decided not to use pairwise potentials. Lastly, as discussed in [24], higher-order potentials make it possible to build more expressive features. This was also confirmed in our preliminary experiment, where triplets representing relative locations of three interesting points have a high discrimination power, even without considering their point contexts. This way, by assigning the similarity computation of point contexts only to singleton potential, we keep the computational cost of high-order graph matching as low as possible.

4.2.1. The Singleton Potential

We define the singleton potential θ_a for the correspondence $(\boldsymbol{p}_i, \boldsymbol{p}'_j)$ between two interesting points $\boldsymbol{p}_i$ and $\boldsymbol{p}'_j$, using their point contexts. We first compute the affinity vectors between the corresponding elements in their distance and orientation vectors:

$$A^k_D(\boldsymbol{p}_i, \boldsymbol{p}'_j) = \exp(-\frac{(D^{\boldsymbol{p}_i}(k) - D^{\boldsymbol{p}'_j}(k))^2}{(\max(D^{\boldsymbol{p}_i})\sigma)^2}) \quad . \tag{8}$$

$$A_{\Theta}^{k}(\boldsymbol{p}_i, \boldsymbol{p}'_j) = \exp(-\frac{(\Theta^{\boldsymbol{p}_i}(k) - \Theta^{\boldsymbol{p}'_j}(k))^2}{\delta^2}) \quad . \tag{9}$$

where k represents the dimension index of n-dimensional vectors $D^{\boldsymbol{p}_i}$ and $\Theta^{\boldsymbol{p}_i}$ (or $D^{\boldsymbol{p}'_j}$ and $\Theta^{\boldsymbol{p}'_j}$) where each dimension in $D^{\boldsymbol{p}_i}$ (or $D^{\boldsymbol{p}'_j}$) and $\Theta^{\boldsymbol{p}_i}$ (or $\Theta^{\boldsymbol{p}'_j}$) represent the distance and orientation of $\boldsymbol{p}_i$ (or $\boldsymbol{p}'_j$) to the k-th sample point, respectively. σ and δ are parameters to control the tolerance of distance and orientation differences, respectively. We set $\sigma = 0.2$ and $\delta = \pi/4$ in all experiments. We calculate the A_D^k and A_Θ^k for n sample points and get two n-dimensional vectors $A_D(\boldsymbol{p}_i, \boldsymbol{p}'_j)$ and $A_\Theta(\boldsymbol{p}_i, \boldsymbol{p}'_j)$. To make the value of $A_D(\boldsymbol{p}_i, \boldsymbol{p}'_j)$ invariant to scale changes, we divide each distance difference by the maximal distance in the first distance vector.

Since both A_D and A_Θ are normalised, we can simply add them to obtain the affinity vector:

$$A(\boldsymbol{p}_i, \boldsymbol{p}'_j) = A_D(\boldsymbol{p}_i, \boldsymbol{p}'_j) + A_\Theta(\boldsymbol{p}_i, \boldsymbol{p}'_j) \quad . \tag{10}$$

The overall similarity between $\boldsymbol{p}_i$ and $\boldsymbol{p}'_j$ can be calculated as the mean value of $A(\boldsymbol{p}_i, \boldsymbol{p}'_j)$. Consequently, the singleton potential for the correspondence $(\boldsymbol{p}_i, \boldsymbol{p}'_j)$ is defined as

$$\theta_a = \theta_{\boldsymbol{p}_i, \boldsymbol{p}'_j} = \frac{1}{n} \sum_{k=1}^{n} A(k) \quad . \tag{11}$$

4.2.2. The Third-Order Potential

We define a third-order potential using angles which are formed by a triplet of interesting points. Suppose that P_1 and P_2 are the set of interesting points for two shapes S_1 and S_2, respectively. For any two triplets, $(\boldsymbol{p}_i^1, \boldsymbol{p}_j^1, \boldsymbol{p}_k^1) \in P_1$ and $(\boldsymbol{p}_i^2, \boldsymbol{p}_j^2, \boldsymbol{p}_k^2) \in P_2$, the third-order potential for each possible triple matching $(\boldsymbol{p}_i^1, \boldsymbol{p}_j^1, \boldsymbol{p}_k^1) \rightarrow (\boldsymbol{p}_i^2, \boldsymbol{p}_j^2, \boldsymbol{p}_k^2)$ is defined with a truncated Gaussian kernel:

$$\theta_{abc} = \theta_{\boldsymbol{p}_i^1, \boldsymbol{p}_j^1, \boldsymbol{p}_k^1, \boldsymbol{p}_i^2, \boldsymbol{p}_j^2, \boldsymbol{p}_k^2} =$$

$$\begin{cases} \exp(-\gamma \| f_{i_1,j_1,k_1} - f_{i_2,j_2,k_2} \|^2) & \text{if } \| f_{i_1,j_1,k_1} - f_{i_2,j_2,k_2} \| \leqslant \vartheta \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where f_{i_1,j_1,k_1} (or f_{i_2,j_2,k_2}) is the three-dimensional vector which describes sine values of three angles formed by $(\boldsymbol{p}_i^1, \boldsymbol{p}_j^1, \boldsymbol{p}_k^1)$ (or $(\boldsymbol{p}_i^2, \boldsymbol{p}_j^2, \boldsymbol{p}_k^2)$). Points in such a triplet are ordered in a clockwise fashion where $\boldsymbol{p}_i^1$ or $\boldsymbol{p}_i^2$ are starting points. We use the truncated Gaussian kernel to scatter and reduce matching times since the number of possible triple matching is huge and it is not necessary to compute them completely. We set γ to 2 in our experiment. With Eq. 12, for each triplet in P_1, we find the triplets in P_2 in a neighbourhood of size ϑ.

Based on [24] and our preliminary experiments, we only sample 20 triangles per interesting points in P_1. There are several possible strategies to select triangles depending on user intentions. If the aim is matching with deformation allowance, the triangle should be selected at small scales. On the other hand, if one wants to capture the global property of a shape, the triangles should be big enough. According to this, we select the triangle based on the distribution of interesting points. If points are densely located in some regions (like the bone in Figure 2), more triangles are sampled in those regions. Otherwise, triangles are sampled randomly. Then, with the same strategy, we select the triangles of P_2, and compute their descriptors. We employ a kd-tree to store them efficiently.

4.3. Formulation Dual-Decomposition

In Eq. 7, we formulate interesting point matching as a high-order graph matching problem combining both extrinsic similarity and intrinsic embedding information (interesting point triangles). The matching is achieved by globally optimising Eq. 7 which includes the cost of the deformation as well as the cost of correspondences according to multiple cues. In order to obtain a globally optimal or near optimal solution while reducing the complexity without searching for all possible matching correspondences, we re-formulate Eq. 7 into sub-problems and reduce the high-order terms in Eq. 7 to quadratic terms.

Specifically, we first employ the dual-decomposition method [31] to re-formulate Eq. 7 into sub-problems that are easier to solve. We define a sub-problem for each type of potentials, that is, $E^1(\mathbf{x}|\theta^1) = \sum \theta_a x_a$ ($\theta^1 = \theta_a$) and $E^2(\mathbf{x}|\theta^2) = \sum \theta_{abc} x_a x_b x_c$ ($\theta^2 = \theta_{abc}$). Based on this definition, let v denote the index of one of V (= 2) sub-problems. Under this setting, we approximate the original problem in Eq. 7 as the following linear combination of sub-problems:

$$E(\mathbf{x}|\theta) = \sum_{v \in I} \rho_v E^v(x|\theta^v) \quad . \tag{13}$$

where ρ_v is the weight for each sub-problem and used to control its importance. In our case, using the heuristic method of Gradient Hill Climbing integrated with Simulated Annealing [32], we set $\rho_1 = 0.7$ and $\rho_2 = 0.3$ for the first and third-order sub-problems, respectively. This means that we put a higher priority on matching of overall shape characteristics based on the singleton potentials. Then, the original problem is solved by iteratively updating potentials θ^v and their interrelated correspondences (x_a or $x_a x_b x_c$) of each sub-problem v while fixing potentials and correspondences for the other sub-problems.

In order to solve the first-order sub-problem, we employ the Hungarian algorithm which is a method to solve the linear assignment problem [10]. For each interesting point $\boldsymbol{p}_i$ in S_1, the Hungarian algorithm can find its corresponding interesting point $\boldsymbol{p}'_j$ in S_2 based on their similarity value in Eq. 11. For the third-order sub-problem, we first employ the high-order reduction method [33] to reduce the high-order terms in Eq. 7 to quadratic terms. Then, the original problem in Eq. 7 can be solved by QPBO algorithm [34].

Essentially, Eq. 7 is a standard optimisation problem which could be solved by many existing methods like HGM [21] and RRWHM [22]. Similar to the employed dual-decomposition method, these methods are designed for high-order graphs. In Section 5.3, we compare the matching performance between the proposed method and other high-order matching approaches. Moreover, Monte-Carlo based methods [35] could also be employed to search the optimal correspondences, since random sampling is not burdensome as we only have a limited number of interesting points in each shape. In Section 6, we will discuss this extension as our future work.

5. Experimental Results

In this section we first evaluate the proposed interest point detector and point context on different datasets. After that, we evaluate and compare the performance of the proposed matching method with some traditional methods to illustrate our advantages. Lastly, we compare our method to the related ones on three datasets. The experiments in this paper are performed on a laptop with Inter Core i7 2.2GHz CPU, 8.00GB memory and 64-bit Windows 8.1 OS. All methods in our experiments are implemented in Matlab R2015a.

5.1. Evaluation of Interest Point Detection

To evaluate the effectiveness of our interest point detector, we use a standard dataset Kimia216 [5] consisting of 216 objects in 18 classes. Our evaluation is based on the retrieval framework, where each shape is used as a query, and the 10 most similar shapes are retrieved from the whole dataset. We evaluate the performance of a method by checking retrieval results using all 216 shapes as queries, and counting how many retrieved shapes belong to the same class of their queries. The similarity value between each shape and its query is calculated using the interesting point matching method in Section 4.2. Since the performance of interest point detection is tightly coupled with the end goal of matching, the similarity values are calculated and compared by using the same point descriptors and the matching algorithm. Built on this, the influence of point descriptor and matching algorithm can be minimised by keeping the only variable: point detection methods. Thus, our experiments are more targeted to the performance of interesting point detection methods.

We compare our interesting point detection method to the most related DCE [16] and corner detection [17]. DCE method detects interesting points by considering the polygon convex during the iterative polygon simplification. In Corner detection method, a contour point is described both by its visual curvatures and corresponding scales. In a certain scale, they consider the points of which their digital visual curvature is above a threshold DK_0 as interesting points. Based on the generated interesting points, we compare retrieval performances of our method and the other two methods using the same descriptors, shape context [6] and our proposed point context. As discussed in Section 2, DCE requires a stop parameter k to control polygon simplification. For a fair comparison, we tried several parameters from $k = 3$ to $k = 15$ and chose $k = 10$ which achieved the best retrieval result for comparison. In corner detection method, we employ their mentioned threshold $DK_0 = 17\pi/64(48^\circ)$.

Table 1 presents the performance comparison between our interesting point detection method and two other methods. The results in this table are collected by checking retrieval results using all the 216 shapes as queries. For example, the fourth position in the row of IP1 shows that from 216 retrieval results in the forth position, 184 shapes belong to the same class as their query shapes. We can observe that retrieval results based on the proposed interesting point detector perform better on both descriptors. The main reason is that the property of interesting points generated by the DCE method is highly related to the stop parameter k. Obviously, it is impractical to set an appropriate k manually on each object. Similar to DCE, corner detection method also detects the interesting points based on the visual curvature which is higher than the threshold DK_0. Since this threshold is not general for all the shapes in this dataset, some interesting points could be mis-detected. On the contrary, our method can generate stable points with no sensitive parameter.

Table 1: Experimental comparison on Kimia216 dataset. SC: Shape Context [6] descriptor, PC: The proposed point context descriptor, IP1: Interesting points detected by DCE method [16], IP2: Interesting points detected by visual curvature method [17], IP3: Interesting points detected by the proposed method.

SC	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
IP1	216	210	195	184	181	172	161	146	148	128
IP2	216	205	195	190	187	179	180	170	171	161
IP3	**216**	**212**	**206**	**197**	**191**	**190**	**186**	**186**	**183**	**171**

PC	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
IP1	216	211	205	196	192	191	186	178	177	175
IP2	216	210	205	203	194	188	179	170	160	155
IP3	**216**	**212**	**211**	**211**	**205**	**200**	**201**	**195**	**193**	**195**

5.2. Evaluation of Point Context

In order to evaluate the performance of the point context descriptor, we compare the retrieval performance using it to the ones employing other most related descriptors, Shape Context (SC) context [6] and PCCS [13]. For an interesting point, the shape context [6] method extracts descriptors as the diagram of the bins which are uniformed in log-polar space. The PCCS method considers interesting points as the contour partition points; then, the shape is represented as the fused shape context descriptor on interesting points and contour segment descriptors. In this experiment, we use MPEG400 dataset which consists of 400 objects categorised in 20 classes. The shapes in this dataset have large intra-class variations and inter-class similarities. Except for the descriptors, all performances are obtained using the same interest points based on our proposed method and are matched by the Hungarian algorithm.

Table 2: Experimental comparison of our point context descriptor to the Shape Context (SC) [6] and partition points-based descriptors (PCCS) [13] using the interesting points generated by our proposed method on MPEG400 dataset. The matching algorithm on all the three methods is the Hungarian algorithm.

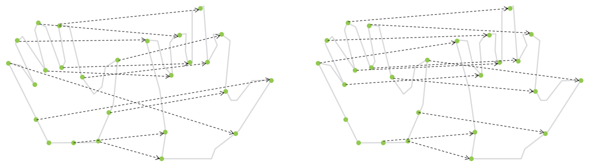
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
SC	370	343	310	302	277	272	265	264	239	240
PCCS	377	351	336	331	317	302	287	282	273	262
Our	**391**	**377**	**372**	**364**	**356**	**343**	**338**	**319**	**304**	**276**

Table 2 illustrates the experimental comparison between three descriptors and the point context descriptor achieves the best performance. The presentation format of Table 2 is the same as the one of Table 1, except for the fact that 400 query shapes are used in Table 2. It is clear that PCCS performs better than SC, since PCCS considers not only SCs on interesting points, but also the geometrical features on contour segments. However, compared to PCCS, point context performs better since it is generated by taking both distance and orientation features for measuring the distribution of relative positions from an interesting point to the sample points.

5.3. Evaluation of High-Order Matching

In this section, we quantitatively illustrate the performance of the proposed high-order graph matching method. First of all, we visually compare the performance of high-order graph matching to the traditional Hungarian method [10] and the state-of-art method in [13] using the same objects. In Figure 7, we match two hands with deformations on some fingers using only the first-order potentials (i.e. matching among single points by Hungarian algorithm) and its combination with the third-order potentials. As shown in Figure 7(a), there are some mismatched interesting points because of the similar points in both shapes. Moreover, the geometrical relations among the interesting points are not considered. Figure 7(b) shows that the proposed high-order matching method yields appropriate matching. Since there are more interesting points in the left hand than the right one, with our constraint in Section 4.1, some points in the left hand will be left out.

In Figure 7(b), we can clearly observe that an interesting point in the right hand (the bottom right corner) is not properly detected. This is because we detect interesting points by considering both curvatures and the overall shape tend. If a corner point and some contour points have similar distances to a reference point, the corner point could be removed during the sequence smoothing steps. Because of this, we can find that the interesting point in the left hand (the bottom right corner) could not be found as a proper corresponding. As a result, it is assigned to another point to meet the singleton and high-order constraints. It also influences the correspondences of its adjacent points. Thus, the proposed method has erroneous correspondences around the wrist area. This problem could be solved by enriching the type of interesting points and setting up a threshold to remove the ambiguous correspondences.



(a) Only the first-order (b) The first and third-order

Fig. 7: Object matching with different potentials. We can observe that the high-order potential performs better.

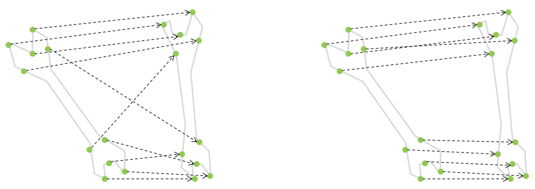


Fig. 8: Comparing the matching result of the proposed method (right) to the method in [13] (left).

In Figure 8, we compare the matching results in [13] to the proposed high-order matching approach. Depending on human perception, there are several mismatched points in Figure 8(a). The main reason is their symmetric silhouette which renders several points difficult to match based on single-point matching. In contrast, as shown in Figure 8(b), with the point context feature and our proposed matching approach, all points in the left tool are correctly matched to the right one.

Next, we quantitatively evaluate the improvement of matching accuracy by our high-order graph matching method. For this, we use the Kimia99 [36] database which contains images of 9 categories of objects, with 11 images per species for a total of 99 images. Table 3 depicts the comparison between incorrect correspondences found using only the first-order potentials (i.e. using the Hungarian algorithm) and those found using the higher-order methods. Using interesting points detected by our method, we first count the total number of correspondences between shapes in the same class. The values in the second column in Table 3 illustrate the total number of correspondences in each class. After that, we apply the Hungarian and different high-order matching methods independently and collect the incorrect correspondences between shapes in the same class. The values in the third and the last columns present the total numbers of incorrect correspondences using the Hungarian and the proposed method, respectively. The values in the fourth and the fifth columns are generated using the HGM [21] and the RRWHM [22].

We can observe that the RRWHM method achieves a performance close to our approach while the HGM has the lowest performance among the three high-order matching methods. The main reason is that the HGM method is unable to effectively incorporate with the matching constraints during its approximation stage. Different from HGM, the RRWHM method works with the mapping constraints in the approximation stage which can effectively reflect the one-to-one matching constraints during the random walks for higher-graph matching. Overall, Table 3 indicates that high-order matching methods can significantly improve the shape matching performance and reduce the number of incorrect matches.

Table 3: Experimental comparison of incorrect correspondences between the Hungarian and high-order matching methods in each class.

Class	Total	Hungarian	HGM	RRWHM	**Our**
animal	1070	270	261	166	**176**
bunny	852	115	84	4	**14**
dude	1034	157	178	83	**94**
fish	414	49	62	27	**28**
hand	1319	497	286	180	**180**
hat	712	193	140	78	**82**
key	947	1	0	0	**0**
plane	1120	450	292	213	**203**
tool	718	130	85	18	**26**

5.4. Performance Comparison to State-of-the-art Methods

Kimia99 Database: Table 4 shows the performance comparison between the proposed method and the most famous methods like Inner Distance (ID) [8], Shape Context (SC) [6] and Path Similarity (PS) [5]. These methods are commonly used for shape matching with different descriptors. We can clearly observe that the proposed method performs better than the ID and SC methods while being close to the best results from the PS method in this database. The main reason is that skeletons employed in the PS method are perfectly pruned with human interaction. In contrast, the whole process of our method can be performed without any human interaction.

Table 4: Experimental comparison of our method to Inner Distance (ID) [8], Shape Context (SC) [6] and Path Similarity (PS) [5] on Kimia99 dataset. The last column illustrates the overall feature generation and matching time in hours.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Time
ID	99	97	92	89	85	85	76	75	63	53	0.15
SC	99	97	91	88	84	83	76	76	68	62	0.69
PS	99	99	99	99	96	97	95	93	89	73	0.38
Our	**99**	**99**	**96**	**92**	**88**	**84**	**80**	**78**	**73**	**60**	**0.36**

Tetrapod120 Database: Tetrapod120 database is organised by ourselves, includes 120 visually similar tetrapod animals with 6 classes, such as camel, cattle, deer, dog, elephant and horse.With this dataset, we aim to evaluate the ability of matching methods for fine-grained shapes. Fine-grained shapes have the similar global topology, but they are mixed up with deformations in some local regions, like dogs and cats. In other words, we evaluate the ability of object matching methods for handling local shape deformations. As illustrated in Table 5, the proposed method achieves the best results. This indicates that the geometric relationship between interesting points is an important feature for distinguishing fine-grained objects.

Table 5: Experimental comparison of our methodology to Inner Distance (ID) [8], Shape Context (SC) [6] and Path Similarity (PS) [5] using Tetrapod120 dataset. The last column illustrates the overall feature generation and matching time in hours.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Time
ID	120	118	106	101	90	83	77	69	70	56	0.22
SC	100	80	70	53	53	51	40	28	27	27	0.96
PS	120	109	101	98	81	78	68	66	65	59	1.41
Our	**120**	**115**	**111**	**105**	**105**	**103**	**98**	**93**	**94**	**87**	**0.53**

MPEG7 Database: The total number of images in the MPEG7 [37] database is 1400: 70 classes of various shapes, each class with 20 images. We employ the so-called bulls-eye score [37] for evaluation. Given a query shape, we retrieve the 40 most similar shapes from the database and count the number of shapes belonging to the same class as the query. The bulls-eye score is the ratio of the total number of correctly matched shapes to the number of all the possible matches (which is 20 × 1400). Thus, the best score is 100 percent. However, as discussed in [37], the 100% bulls-eye score is impossible to achieve since some classes contain objects whose shape is significantly different so that it is not possible to group them into the same class using only their shapes.

For comparison, Table 6 lists several reported results and the results by our proposed method on MPEG7 dataset. We cluster existing methods into two groups: pairwise matching and context-based matching. In the first group, results are decided by the similarity measures for shape pairs. In the second group, results are generated by considering the underlying structure of the shape manifold [38] in which the obtained similarity scores are post-processed by analysing the shape similarities between all given shapes to increase the discriminability between different shape groups.

In the first group, the proposed matching method achieves a 80.28% bulls-eye score which is better than a traditional contour-based descriptor [6]. However, this approach performs not as well as Shape Tree [39] and HF [40], etc. The main reason is that we only used 240 sample points from shape contour (containing more than 1500 points) to generate point context features for interesting points and the performance can be improved by using more sample points. In the group of context-based methods, we employ a simple and fast Mutual *k*NN Graph method [41] based on the proposed similarity scores between all the shapes. This method captures the manifold structure by defining a neighbourhood for each shape. Our method, which achieves 96.43% bulls-eye score, outperforms most state-of-the-art methods. It is important to mention that Donoser et al. [42] proposed a generic framework for diffusion processes in the scope of retrieval applications which achieved 100% accuracy on MPEG7 dataset. However, as illustrated in Table 6, our performance comes close to 100% only using the simple Mutual *k*NN Graph method.

Table 6: Bulls-eye score on the MPEG7 Dataset. HG denotes the proposed high-order matching method.

Pairwise Matching	Score	Context-based	Score
Shape Contexts [6]	76.51%	INSC + CDM [43]	88.30%
Skeletal Context [44]	79.92%	IDSC + LP [45]	91.00%
Optimized CSS [46]	81.12%	SC + LP [45]	92.91%
Multiscale Rep. [47]	84.93%	IDSC + LCDP [48]	93.32%
Shape LAneRouge [49]	85.25%	SC + GM + Meta [50]	92.51%
Fixed Cor. [51]	85.40%	IDSC + MG [41]	93.40%
Inner Distance [8]	85.40%	IDSC + PS + LDCP [52]	95.60%
Symbolic Rep. [53]	85.92%	ASC + LDCP [54]	95.96%
Hier.Procrustes [55]	86.35%	HF + LCDP [40]	96.45%
Triangle Area [56]	87.23%	SC + DDGM + Co-T [38]	97.45%
Shape Tree [39]	87.70%	AIR [57]	93.67%
Height Functions [40]	89.66%	ASC + TN + TPG [58]	96.47%
HG	**80.28%**	**HG + Mutual Graph**	**96.43%**

5.5. Application Independent Experiments

In order to evaluate the proposed method in the case of unsegmented natural image settings, we apply two experiments using the ETHZ shape classes dataset [59]. This dataset features five diverse classes (apples, bottles, giraffes, mugs and swans) and contains a total of 255 images collected from the Internet. It is highly challenging, as the objects appear in a wide range of scales; there is considerable intraclass shape variation, and many images are severely cluttered. Based on this dataset, we generate two groups of shapes. In the first group, shapes are generated by the ground truth shapes. We also introduce noise into the shapes to mimic imperfect segmentations. In the second group, shapes are generated by a semi-supervised segmentation method introduced in [60]. In this method, the shape of a target object is detected and segmented by an object boundary detector which can best fit its situation using the global image appearance. Some sample images and their segmented shapes are shown in Figure 9.

In the first experiment, we apply the proposed point detection method to two shape groups and report the false positive (FP) and false negative (FN) rates of detecting interesting points in Table 7 (the upper table). We can observe that the FP rates in two groups are both promising (1.66% and 0.81%) while the FN rates are much higher (11.26% and 9.64%). The main reason is that our point detection method only takes the boundary points by mainly considering the overall shape tend. In this case, some corner points could be ignored which leads to the high FN rates. In the future, we will try to detect interesting points by considering both overall shape tend and boundary corner to reduce the FN rate of our method.

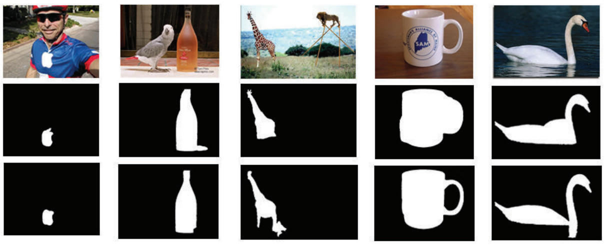


Fig. 9: Sample images and their correlated shapes from ETHZ [59] database. The second row illustrates the ground truth shapes with manually added noise. The third row shows the segmented shapes using the semi-supervised segmentation method in [60].

Table 7: Experimental results of interesting point detection (the upper table) and shape retrieval (the lower table) in the case of imperfect segmentations. Here, M and S illustrate the ground truth shapes with manually added noise and the shapes using the semi-supervised segmentation method [60], respectively.

	apples		bottles		giraffes		mugs		swans		**mean**	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN	**FP**	**FN**
M	2.98	16.45	2.26	1.04	0.38	12.35	2.15	16.96	0.52	9.52	**1.66**	**11.26**
S	0.94	9.63	0.87	3.56	0.83	12.40	0.69	12.75	0.74	9.88	**0.81**	**9.64**

	apples	bottles	giraffes	mugs	swans	**mean**
M	83.81	79.99	68.01	71.09	54.88	**71.56**
S	70.50	60.81	62.07	55.12	48.63	**59.43**

In the second experiment, we use two shape groups in a retrieval scenario. The mean accuracies within each class and the whole dataset are reported in Table 7 (the lower table). Though the mean accuracy in the shape group with manually added noise is higher than the semi-segmented shapes, the overall performances of our method are not promising in both shape groups. There are two reasons for this: The first reason is that the integrated high-order graph matching could mislead the partial matching since an imperfectly segmented shape could have a high number of interesting points which do not belong to the main object. With the higher-order constraints, those “fake” interesting points could be matched to the “real” one which influence the similarity value between shapes. Therefore, for the shape matching applications with imperfect segmentation, the high-order graph matching is not recommended. The second reason is due to the invariant properties of the proposed shape descriptor. Theoretically, for a single interesting point, our point context descriptor is rotation invariant. Considering a shape with multiple interesting points, the proposed shape descriptor in Eq. 4 is not completely invariant to rotation since the order of interesting points could be changed if a shape is rotated. In practice, this problem can be easily solved by some shape preprocessing methods [61]. However, in many applications, complete invariance impedes recognition performance [6]. Therefore, we apply the complete rotation invariance based on applications.

5.6. Computational Complexity

We now analyse the computational complexity of the proposed interesting point generation and matching approaches. (1) For interesting point generation, the time complexity is in the order of $O(N'^2)$, where N' is the number of boundary points in the shape Ω. This is because the worst case of *Douglas-Peucker* [26] is $O(N'^2)$. Moreover, the computational complexity of FMM [27] is $O(N')$. For the remaining sequence generation and filtering tasks, they can be finished in $O(N')$ time. Fusing those tasks together, the overall complexity is $O(N'^2)$. (2) For point context generation, the time complexity is $O(mn)$, where m and n are the number of interesting points and contour sample points, respectively. This is because point context features are generated by considering the distance and orientation relationships between each interesting point and contour sample points. In practice, since the m is a value independent of contour sample points and $m \ll n$, the computational complexity of this method is determined by the contour sample points. Therefore, the complexity for feature generation is $O(n)$. (3) For interesting point matching, we analyse the computational complexity by different potentials. The Hungarian algorithm is used for singleton potential; as introduced in [10], it can solve our point matching task in $O(N^3)$ time. For the third-order potential, assuming we select m_1 and m_2 interesting points from two shapes, there are $O(m_1^3 m_2^3)$ possible triplets, each represented by a high-order term in Eq. 7. This enables us to significantly reduce the complexity without searching for all possible matching correspondences.

Here we report the computation time based on Kimia216 dataset with the experimental environment introduced above. On average, the shape resolution in this dataset is 187 × 239. For interesting point generation, the mean time is 5.33 seconds on each shape. With interesting points, the mean time for point context generation is 0.01 seconds. Given two sets of interesting points, the mean matching time is 0.09 seconds. Please notice that our code is not optimised, and its faster implementation is possible by optimising loops and programming, etc.

6. Conclusion and Future Work

In this paper, a novel shape matching method based on the interesting point detector and high-order graph matching is presented. The main idea is to generate some interesting points on each shape contour for correspondence matching. For each interesting point, we also proposed the point context descriptor to capture its geometrical and topological locations. For shape matching, we employ a high-order graph matching method in order to involve not only the assignments of single interesting points, but also the geometrical relations of their triplets. The experiments on six databases demonstrate the significant improvements of shape matching. In the future, we will try to enrich the types of interesting points by fusing with other corner detection methods. Moreover, we will also consider to solve our high-order matching problem using Monte-Carlo based methods. Specifically, with the framework of particle filter, we can sample particles using the assignments between interesting points and their triangles. The individual importance weight is assigned to each particle according to the proposed singleton and third-order potentials. With the iterations of resampling, the optimal correspondences can be observed.

Acknowledgments

Research activities leading to this work have been supported by the China Scholarship Council (CSC) and the German Research Foundation (DFG) within the Research Training Group 1564 (GRK 1564).

References

[1] Tadeusiewicz R., et al. Notes on a linguistic description as the basis for automatic image understanding. *IJAMCS*, 19(1):143–150, 2009.

[2] Oliver van Kaick, et al. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[3] Dengsheng Zhang et al. Review of shape representation and description techniques. *PR*, 37(1):1–19, 2004.

[4] C. Yang, et al. Shape-based object retrieval and classification with supervised optimisation. In *ICPRAM*, pages 204–211. 2015.

[5] Xiang Bai et al. Path similarity skeleton graph matching. *PAMI*, 30(7):1282–1292, 2008.

[6] S. Belongie, et al. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[7] Byung-Woo Hong et al. Shape matching using multiscale integral invariants. *PAMI*, 37(1):151–160, 2015.

[8] Haibin Ling et al. Shape classification using the inner-distance. *PAMI*, 29(2):286–299, 2007.

[9] Ravindra K. Ahuja, et al. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., 1993.

[10] Harold W. Kuhn. The hungarian method for the assignment problem. *NRLQ*, 2:83–97, 1955.

[11] Yossi Rubner, et al. The earth mover's distance as a metric for image retrieval. *IJCV*, 40(2):99–121, 2000.

[12] L.J. Latecki, et al. Optimal subsequence bijection. In *ICMD*, pages 565–570. 2007.

[13] C. Feinen, et al. Shape matching using point context and contour segments. In *ACCV*. 2014.

[14] Lorenzo Torresani, et al. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, pages 596–609. 2008.

[15] Xiang Bai, et al. Skeleton pruning by contour partitioning with discrete curve evolution. *PAMI*, 29(3):449–462, 2007.

[16] L. J. Latecki et al. Convexity rule for shape decomposition based on discrete contour evolution. *CVIU*, 73:441–454, 1999.

[17] HaiRong Liu, et al. Visual curvature. In *CVPR*, pages 1–8, 2007.

[18] Hairong Liu, et al. A unified curvature definition for regular, polygonal, and digital planar curves. *IJCV*, 80(1):104–124, 2008.

[19] D.P. Huttenlocher, et al. Comparing images using the hausdorff distance. *PAMI*, 15(9):850–863, 1993.

[20] M. Leordeanu et al. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, volume 2, pages 1482–1489. 2005.

[21] R. Zass et al. Probabilistic graph and hypergraph matching. In *CVPR*, pages 1–8, 2008.

[22] Jungmin Lee, et al. Hyper-graph matching via reweighted random walks. In *CVPR*, pages 1633–1640, 2011.

[23] Ming-Ching Chang et al. Measuring 3d shape similarity by graph-based matching of the medial scaffolds. *CVIU*, 115(5):707–720, 2011.

[24] O. Duchenne, et al. A tensor-based algorithm for high-order graph matching. *PAMI*, 33(12):2383–2395, 2011.

[25] Hairong Liu, et al. Robust clustering as ensembles of affinity relations. In *NIPS*, pages 1414–1422, 2010.

[26] Konrad Ebisch. A correction to the douglas-peucker line generalization algorithm. *CG*, 28(8):995–997, 2002.

[27] M. Sabry Hassouna et al. A highly accurate solution to the eikonal equation on cartesian domains. *PAMI*, 29(9):1563–1574, 2007.

[28] Tianyang Ma et al. From partial shape matching through local deformation to robust global shape similarity for object detection. In *CVPR*, pages 1441–1448. 2011.

[29] ChengEn Lu, et al. Shape guided contour grouping with particle filters. In *ICCV*, pages 2288–2295. 2009.

[30] Yun Zeng, et al. Dense non-rigid surface registration using high-order graph matching. In *CVPR*, pages 382–389. 2010.

[31] N. Komodakis et al. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *CVPR*, pages 2985–2992. 2009.

[32] S. Kirkpatrick, et al. Optimization by simulated annealing. *Science*, pages 671–680, 1983.

[33] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, pages 2993–3000. 2009.

[34] V. Kolmogorov et al. Minimizing nonsubmodular functions with graph cuts-a review. *PAMI*, 29(7):1274–1279, 2007.

[35] Christian Robert et al. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[36] T. B. Sebastian, et al. Recognition of shapes by editing their shock graphs. *PAMI*, 26(5):550–571, 2004.

[37] L.J. Latecki, et al. Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, volume 1, pages 424–429. 2000.

[38] X. Bai, et al. Co-transduction for shape retrieval. *Image Processing*, 21(5):2747–2757, 2012.

[39] P.F. Felzenszwalb et al. Hierarchical matching of deformable shapes. In *CVPR*, pages 1–8. 2007.

[40] J. Wang, et al. Shape matching and classification using height functions. *PRL*, 33(2):134–143, 2012.

[41] Peter Kontschieder, et al. Beyond pairwise shape similarity analysis. In *ACCV*, pages 655–666. 2010.

[42] M. Donoser et al. Diffusion processes for retrieval revisited. In *CVPR*, pages 1320–1327. 2013.

[43] H. Jegou, et al. Accurate image search using the contextual dissimilarity measure. *PAMI*, 32(1):2–11, 2010.

[44] Jun Xie, et al. Shape matching and modeling using skeletal context. *PR*, 41(5):1773–1784, 2008.

[45] X. Bai, et al. Learning context-sensitive shape similarity by graph transduction. *PAMI*, 32(5):861–874, 2010.

[46] F. Mokhtarian et al. Curvature scale space representation: Theory, applications and mpeg-7 standardization. 2003.

[47] T. Adamek et al. A multiscale representation method for nonrigid shapes with a single closed contour. *CSVT*, 14(5):742–753, 2004.

[48] X. Yang, et al. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, pages 357–364, 2009.

[49] A. Peter, et al. Shape iane rough: Sliding wavelets for indexing and retrieval. In *CVPR*, pages 1–8, 2008.

[50] A. Egozi, et al. Improving shape retrieval by spectral matching and meta similarity. *Image Processing*, 19(5):1319–1327, 2010.

[51] B. J. Super. Retrieval from shape databases using chance probability functions and fixed correspondence. *IJPRAI*, 20(8):1117–1138, 2006.

[52] A. Temlyakov, et al. Two perceptually motivated strategies for shape classification. *CVPR*, pages 2289–2296, 2010.

[53] M. R. Daliri et al. Robust symbolic representation for shape recognition and retrieval. *PR*, 41(5):1782–1798, 2008.

[54] H. Ling, et al. Balancing deformability and discriminability for shape matching. In *ECCV*, pages 411–424, 2010.

[55] G. McNeill et al. Hierarchical procrustes matching for shape retrieval. In *CVPR*, volume 1, pages 885–894, 2006.

[56] N. Alajlan, et al. Geometry-based image retrieval in binary image databases. *PAMI*, 30(6):1003–1013, 2008.

[57] R. Gopalan, et al. Articulation-invariant representation of non-planar shapes. In *ECCV*, pages 286–299, 2010.

[58] X. Yang, et al. Affinity learning with diffusion on tensor product graph. *PAMI*, 35(1):28–38, 2013.

[59] Vittorio Ferrari, et al. From images to shape models for object detection. *IJCV*, 87(3):284–303, 2009.

[60] Jasper R. R. Uijlings et al. Situational object boundary detection. In *CVPR*, 2015.

[61] C. Yang, et al. Shape-based classification of environmental microorganisms. In *ICPR*, pages 3374–3379. 2014.