Object Matching with Hierarchical Skeletons

Cong Yang*, Oliver Tiebe, Kimiaki Shirahama and Marcin Grzegorzek*

Pattern Recognition Group, Institute for Vision and Graphics

University of Siegen, Siegen, D-57076, Germany

Abstract

The skeleton of an object provides an intuitive and effective abstraction which facilitates object matching and recognition. However, without any human interaction, traditional skeleton-based descriptors and matching algorithms are not stable for deformable objects. Specifically, some fine-grained topological and geometrical features would be discarded if the skeleton was incomplete or only represented significant visual parts of an object. Moreover, the performance of skeleton-based matching highly depends on the quality and completeness of skeletons. In this paper, we propose a novel object representation and matching algorithm based on hierarchical skeletons which capture the shape topology and geometry through multiple levels of skeletons. For object representation, we reuse the pruned skeleton branches to represent the coarse- and fine-grained shape topological and geometrical features. Moreover, this can improve the stability of skeleton pruning without human interaction. We also propose an object matching method which considers both global shape properties and fine-grained deformations by defining singleton and pairwise potentials for similarity computation between hierarchical skeletons. Our experiments attest our hierarchical skeleton-based method a significantly better performance than most existing shape-based object matching methods on six datasets, achieving a 99.21% bulls-eye score on the MPEG7 shape dataset.

Keywords: Skeletonisation, Hierarchical Skeleton, Skeleton Evolution, Object Representation, Object Matching

*Corresponding authors: Tel.: +49 271 740 3972; fax: +49 271 740 1 3972

Email address: cong.yang@uni-siegen.de (Cong Yang)

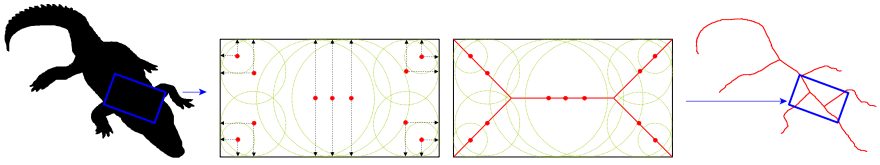
[1]C. Yang, O. Tiebe, K. Shirahama and M. Grzegorzek are with the Research Group for Pattern Recognition, Institute for Vision and Graphics, University of Siegen, Siegen, D-57076, Germany.

2010 MSC: 00-01, 99-00

1. Introduction

Shape is an expressive abstraction of the visual pattern of an object. While there are many different approaches [1] using shape for object matching, nearly all of them face the same challenge: object deformation. As shown in Figure 1, the shapes of the same object are visually different depending on its deformations. To overcome this, on the one hand, various robust shape descriptors [2, 3, 4, 5, 6, 7, 8, 9, 10, 11] are designed to capture both local and global geometric properties. On the other hand, some holistic [12, 13] and elastic [8, 14, 2] matching algorithms are proposed to handle the ambiguous correspondences. Among the above-mentioned research efforts, skeleton is an important shape descriptor for deformable object matching since it integrates both geometrical and topological features of an object.

Figure 1: An illustration of shapes which significantly vary depending on deformations.



(a) Shape (b) Maximum disks (c) Collection of centres (d) Skeleton

Figure 2: An overview of the skeletonisation process to convert a given shape (a) into a skeleton (d). (b) and (c) visually illustrate the skeleton extraction process, where the skeleton (red line) of a shape (rectangle) is generated by collecting the centres (red dots) of all discs (green dotted circles) that touch the boundary of the shape on two or more different locations (dotted arrows).

Figure 2 shows an overview of the skeletonisation process to convert a given shape (a) into a skeleton (d). Specifically, a skeleton is defined as a connected set of medial

lines along the limbs of its shape [15]. From a technical point of view, such a skeleton is extracted by continuously collecting centre points of maximal tangent disks touching the object boundary on two or more locations, as shown in Figure 2 (b) and (c). The centre point of a maximal tangent disk is referred to as a skeleton point. The sequence of connected skeleton points is called a skeleton branch. A skeleton point having only one adjacent point is an endpoint (the skeleton endpoint). A skeleton point having three or more adjacent points is a junction point. The skeletons described above usually lead to a better performance than contour or other shape descriptors in the presence of partial occlusion and articulation of parts [16]. This is because skeletons have a notion of both the interior and exterior of the shape [16], and are useful for finding the intuitive correspondence of deformable shapes.

However, a skeleton is sensitive to the deformation of an object's boundary because little variation or noise of the boundary often generates redundant skeleton branches that may seriously disturb the topology of the skeleton [17, 18, 19]. Furthermore, a large number of skeleton branches may cause the overfitting problem and high computation complexity. Though skeleton pruning [12, 17] approaches can remove the inaccurate or redundant branches while preserving the essential topology, they normally require manual intervention to produce visually pleasing skeletons. Moreover, the performance of skeleton-based matching highly depends on the quality and completeness of skeletons.

To overcome these problems, we propose a hierarchical skeleton-based object matching method. A hierarchical skeleton is a set of skeletons that represent an object at different levels. More specifically, during the skeleton pruning process, we store all the pruned branches until the skeleton is pruned to the simplest form. These branches are reused to construct the hierarchical skeleton which is favourable for the following reasons: First, it does not need any manual intervention since we consider a set of skeletons rather than a single one. Second, a hierarchical skeleton captures geometric and topological features at different levels along with skeleton pruning. Fine levels feature the small object deformation while skeletons at coarse levels capture global shape deformations. This enables us to develop an object matching algorithm that allows more deformations on finer levels while preserving important global geometrical

and topological properties. This design is based on the fact that objects (e.g. the four objects on the right side of the arrow in Figure 3) reconstructed with the same skeleton topology are still perceptually similar to the original (the triangle on the left side of the arrow in Figure 3) even though there are some fine-grained noises and deformations.

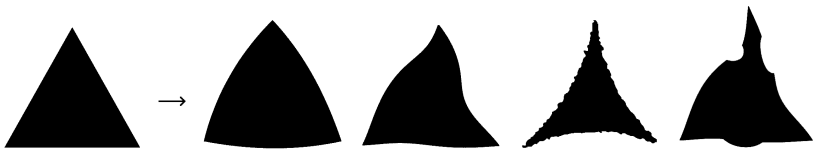


Figure 3: Examples of shapes that are perceptually similar to the original one, irrespective of fine-grained noises and deformations.

The third advantage of a hierarchical skeleton is that it can also provide additional information for improving the object matching accuracy. In particular, by looking into the skeleton pruning process, transitions of pruned skeletons from the same category are more similar than those from different ones. This is because skeletons from the same category have more similar branches and these branches on each level have similar effects on the possible skeleton reconstruction. We call this phenomenon skeleton evolution. In Section 5.3, we show that adopting skeleton evolutions improves the performance of object matching.

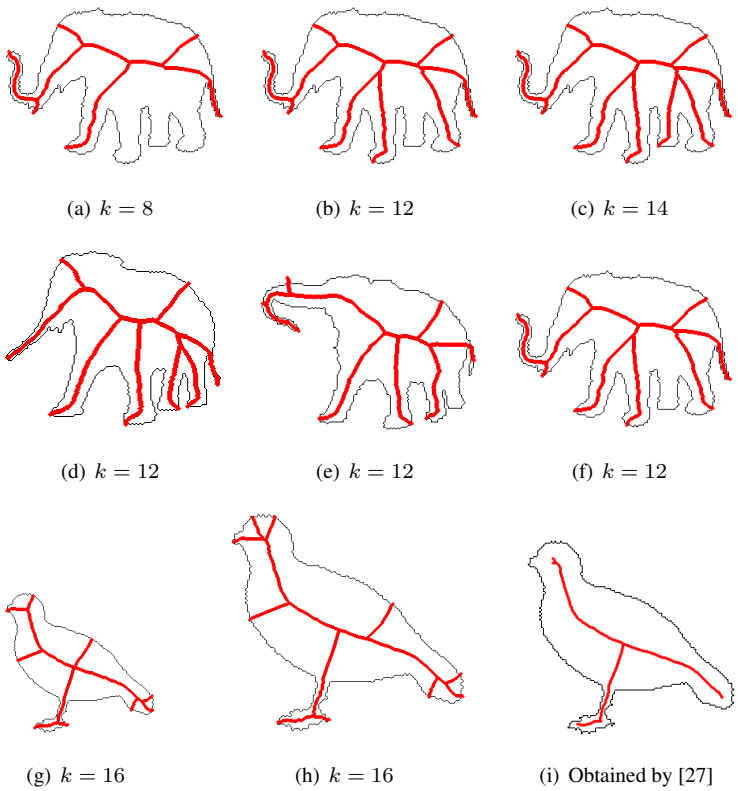
As the fourth advantage, the hierarchical skeleton is obtained along with the skeleton pruning process, requiring no extra computational cost. Lastly, by limiting levels of hierachical skeletons, we can filter out skeleton branches which represent shape properties irrelevant to matching. This alleviates the overfitting problem.

2. Related Work

Several skeletonisation methods have been developed to generate proper skeletons [20, 21, 22, 23]. One typical approach is to continuously collect the centre points of maximal tangent disks that touch the object boundary in two or more locations. However, all of the obtained skeletons are sensitive to small changes and noises in the object boundary [24, 25]. The intrinsic reason is that a small protrusion on the

boundary may result in a large skeleton branch. To solve this problem, Choi and Telea, etc [26, 27, 28, 29] proposed algorithms to detect the skeleton in a distance map of the boundary points. Figure 4(i) shows a skeleton obtained by the method in [28]. Although these methods can preserve some visual parts of a shape, some significant parts are missing. Therefore, they cannot guarantee the completeness of a skeleton. To overcome this, Bai and Latecki present significance measures for skeleton pruning associated with Discrete Curve Evolution (DCE) [12] or Bending Potential Ratio (BPR) [17]. Both methods decide whether or not a skeletal branch should be pruned by evaluating the contribution of its corresponding boundary segment to the overall shape. However, these methods require manual intervention to stop the evaluation and produce visually pleasing skeletons. For example, in Figure 4, DCE [12] requires a proper stop parameter k to calibrate the pruning power. However, different stop parameters for the same object (the first row in Figure 4) or the same parameter for different objects (the second row in Figure 4) lead to visually different skeletons in which some important parts are missing (legs in Figure 4(a), 4(b), 4(e), 4(f)). Furthermore, even if we find the best stop parameter, skeletons of the same object sometimes differ if the scale is changed (Figure 4(g) and 4(h)). This is because the vanishing of shape parts is unavoidable when the resolution decreases [30]. Therefore, fixing k for skeleton pruning is not a proper solution for all objects. In contrast, our hierarchical skeleton is a collection of skeletons obtained by all the stop parameters. This not only eliminates the necessity of manually tuning a stop parameter, but also preserves both the coarse-grained global and fine-grained local properties of a shape.

For skeleton matching, most methods [31, 32, 33, 34, 35, 36, 37, 38] only consider one skeleton for a shape. However, the matching performance relies on the quality of a skeleton since it is essential to find the correct corresponding elements. Some methods [18, 25, 39] promote the matching performance by fusing additional shape descriptors. Though the global matching accuracy could be improved, it requires additional time complexity for feature generation and parameter optimisation. Moreover, the contribution of the skeleton-based part remains the same. In contrast, we represent a hierarchical skeleton as a set of multiple skeletons. Although it includes both high- and low-quality skeletons, our matching method can appropriately determine the



(a) $k = 8$ (b) $k = 12$ (c) $k = 14$

(d) $k = 12$ (e) $k = 12$ (f) $k = 12$

(g) $k = 16$ (h) $k = 16$ (i) Obtained by [27]

Figure 4: Skeletons on the same elephant shape with different DCE stop parameters k (first row), skeletons on different shapes with the same DCE stop parameter k (second row), and skeletons on the shape with different scales and the same stop parameter ((g) and (h) in the third row), and a pruned skeleton obtained by the method in [27] ((i) in the third row).

priority of each skeleton.

Although some researchers proposed approaches to organise skeletons into a hierarchy or a graph, they have several problems in handling object deformation and fine-grained shape characteristics. Pizer [32] and Ogniewicz [40] proposed the shape descriptors in the form of a hierarchy by inducing shapes into components with multi-resolution and symmetric axis transform approaches. The scale value and parent-child relationship are induced under a successive reduction of resolution. Macrini [33, 34, 35] introduced a graph-based medial shape abstraction called bone graph [41]. Here, edges are designed to allow a richer specification of relational information, includ-

ing how and where two medial parts meet. Although the approaches described above are robust to noise in the object boundary, several drawbacks remain: (1) They may fail to find an appropriate matching solution because of the heuristics for finding the skeleton tree root [42]. (2) Some fine-grained shape features could be lost as they smooth the shape boundaries before extracting the skeletons. In particular, curvature flow smoothing would remove the fine characteristic of a shape. As a result, the topology and geometry of the skeleton would be altered. (3) The main skeleton branches based on these approaches are shortened which may lose important fine-grained shape information and seriously compromise the structure of the skeletons. Thus, the shortening of branches may make branches of significant visual parts indistinguishable from branches attributed to noise.

Unlike the above-mentioned approaches, our hierarchical skeleton is constructed by collecting skeletons that are examined in skeleton pruning steps. This offers the following advantages: First, we do not need to search the root level since all hierarchical levels are ordered along with skeleton pruning steps. Moreover, the proposed method fully uses the boundary information for object matching since they are hierarchically preserved by our descriptor. Lastly, in our method, skeletons on each level are not shortened which can reduce the ambiguous correspondences and improve the accuracy of single skeleton matching.

3. Hierarchical Skeleton Extraction

The extraction of a hierarchical skeleton takes advantage of skeleton pruning which iteratively removes skeleton branches of visually insignificant shape parts based on the boundary abstraction method, DCE [12]. Figure 5 illustrates an overview of the pruning process: (1) Given a planar shape D (Figure 5(a)), the Max-Disk Model [26] is used to generate the initial skeleton $S^n(D)$ (Figure 5(b)) as a set of centre points of circles that are in contact with the shape boundary. That is, $s \in S^n(D)$ is the centre of such a circle, and contact points of s on the shape boundary are called generating points. n indicates the first iteration index of DCE and will be iteratively decremented until 3. k denotes one of these steps. (2) The boundary of D is regarded as the initial

polygon P^n and will be simplified into the polygon P^k (blue solid line in Figure 5(c)) using the polygon simplification method described below. (3) With P^k, $S^n(D)$ is pruned by removing all skeleton points $s \in S^n(D)$ so that the generating points (the points of tangency between the shape contour and the max discs) of s are contained in the same contour segment. A contour segment is defined as a part of the shape boundary which is approximated by the straight line (polygon partition) between two neighbouring vertices of P^k (red stars in Figure 5(c)). Each pruned point s results from a contour segment with respect to the polygon partition and therefore, s can be considered as an unimportant skeleton point and can be removed.

In order to describe the aforementioned processes more intuitively, we enlarge the bird head (the part in the red-dotted rectangles) in Figure 5 as an example. Specifically, the initial head skeleton and the head contour are illustrated together in (e). With vertices of P^k (the red points), the head contour is divided into three contour segments (marked with three different colours in (f)). Here, a skeleton point s (the green point in (f)) is generated by a max-disc (the green circle) which has at least two touching points (generating points) to the shape contour. Since all the touching points of s are located in the same contour segment (with the pink colour), s is an unimportant skeleton point and can be removed. We iteratively apply this removal process until all unimportant skeleton points are erased. As a result, the skeleton in (f) is pruned and illustrated in (g).

The skeleton pruning described above is based on a simplified polygon P^k. Below, we explain how DCE generates P^k. As shown in Figure 6, a pair of consecutive line segments s_1, s_2 is replaced by a single line segment that connects the endpoints of $s_1 \cup s_2$. DCE produces a sequence of simpler polygons $P = P^n, P^{n-1}, ..., P^3$ so that P^{n-k} is obtained by removing a single vertex v from P^{n-k+1}. Here, v is regarded as having the smallest shape contribution based on the following measure K:

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)} \quad . \tag{1}$$

where $\beta(s_1, s_2)$ is the angle of the corner consisting of s_1 and s_2. l is the length function normalised with respect to the total length of lines constituting the polygon. Based on K, while the the value of $K(s_1, s_2)$ becomes higher, the contribution of

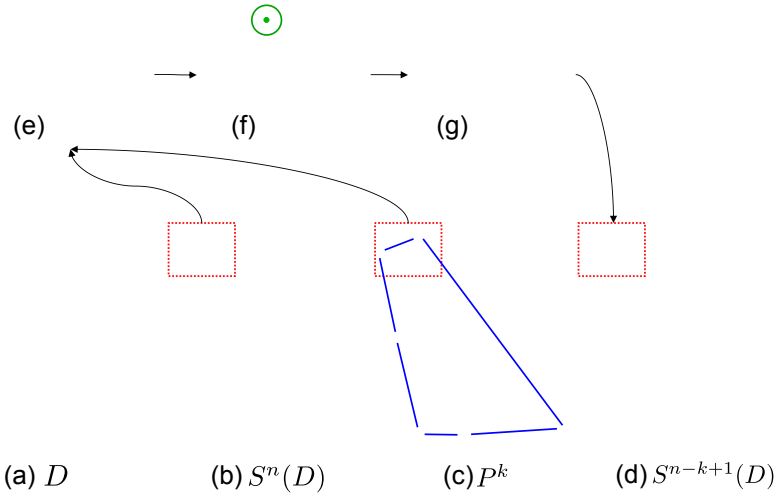
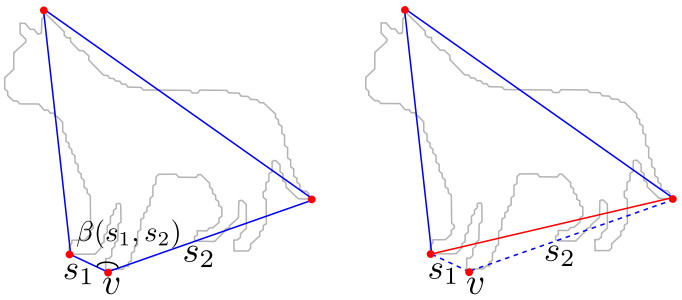


Figure 5: Illustration of original shape D, initial skeleton $S^n(D)$, simplified polygon P^k and the pruned skeleton $S^{n-k+1}(D)$ generated by [12] with $k = 10$.

$s_1 \cup s_2$ to the polygon is larger. A few stages of polygon simplification are illustrated in Figure 7.

By reordering polygons from simple to complex in the polygon simplification steps (i.e., $P = P^3, P^4, \cdots, P^n$), we construct a hierarchical skeleton $S(D) = S^3, S^4, \cdots, S^n$. As pruned skeletons are a by-product of the polygon simplification process, the total calculation cost stays the same to the skeleton pruning process. In order to reduce the computational cost of skeleton matching, we build $S(D)$ only using skeletons collected from the T_{min}th to T_{max}th iterations. In this experiment, we select T_{max} as the start step number based on the trade-off between computation time and matching results, $T_{max} > T_{min}$. With this, the DCE iteration can directly start from T_{max} and stops at T_{min}. An example of a hierarchical skeleton is shown in Figure 8, and we only use the skeletons with $t \in [3, 15]$ for matching. Setting T_{min} = 3 is based on the constraint that skeletons should have at least three endpoints.



(a) (b)

Figure 6: Polygon simplification. As vertex v has the smallest contribution with Eq. 1, its consecutive line segments s_1, s_2 are replaced by a single line segment (red line in (b)).

4. Matching

In this section, we first introduce the formulation of a matching algorithm based on the properties of hierarchical skeletons. Then, we define two potential functions which are used to capture the shape similarity from different properties of hierarchical skeletons.

4.1. Formulation of Hierarchical Skeleton Matching

Let D_1 and D_2 be two planar shapes, $S(D_1)$ and $S(D_2)$ denote the full hierarchical skeletons for D_1 and D_2, respectively. Let us denote by $S_1 \subseteq S(D_1)$ or $S_2 \subseteq S(D_2)$ the set of skeletons from the levels $[T_{min}, T_{max}]$ that are chosen for object matching. In order to calculate the distance between S_1 and S_2, we define p as a set of correspondences between skeletons in S_1 and those in S_2. Based on this, we formulate hierarchical skeleton matching as follows:

$$d(S_1, S_2) = \sum_{a \in p} g(\theta_a) x_a + \sum_{(a,b) \in p \times p} g(\theta_{ab}) x_a x_b \quad . \tag{2}$$

where θ_a is the matching cost for each correspondence $a \in p$ (the singleton potential, dotted arrows in Figure 9) that expresses the property of skeleton-based object matching. θ_{ab} is the matching cost of a pair of correspondences $(a, b) \in p \times p$ (the pairwise potential, skeleton pairs connected by solid arrows in Figure 9) that represents the difference between skeleton evolutions in two hierarchical skeletons. Here, the skeleton

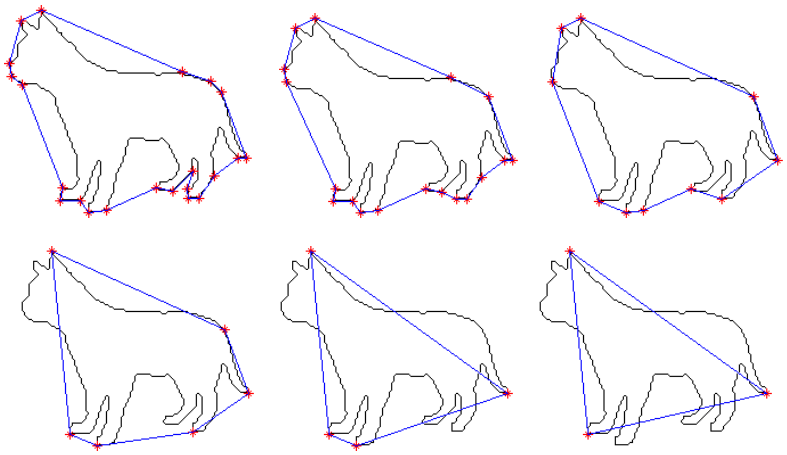


Figure 7: Polygon P (in blue line) and vertex v (in red asterisk) changes along with a few stages of DCE.

evolution of a hierarchical skeleton is characterised as changes between skeleton pairs on two different levels. The rationale behind this is that there is no predictable pattern in how skeletons change in different levels. In some levels, skeletons are even the same due to the overlapped removal of skeleton points along with the polygon simplification. However, as shown in Figure 8, the overall trend is that skeletons are gradually becoming more complex along with the DCE steps. Built on this observation, we collect the skeleton changes by considering all skeleton pairs in a hierarchical skeleton. With this strategy, even the skeleton changes are zero within some pairs, we can still get sufficient skeleton evolution information to distinguish two hierarchical skeletons since we calculate the overall accumulation of the skeleton changes. $g(\theta_a) = |\theta_a|^\alpha$ (or $g(\theta_{ab}) = |\theta_{ab}|^\alpha$) is the power function term to alleviate effects by abnormally large matching costs θ_a (or θ_{ab}) [43]. Based on our preliminary experiment, we set α to 0.18. This value is obtained by employing a combination of two heuristic optimisation methods: Gradient Hill Climbing [44] integrated with Simulated Annealing [45]. x_a is the boolean indicator variable:

$$x_a = \begin{cases} 1 & \text{if } a = (i,j) \in p \text{ and } i = j \\ 0 & \text{otherwise} \end{cases} . \tag{3}$$

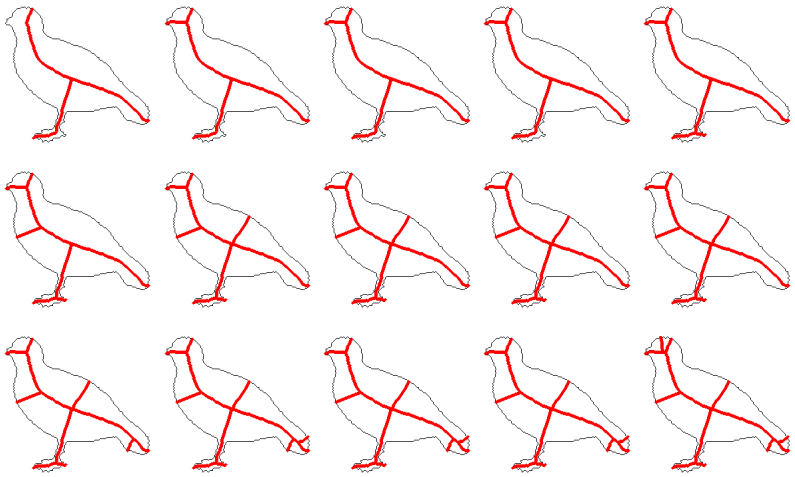


Figure 8: A hierarchical skeleton with DCE steps from $t = 3$ to 17. We call the changes of these skeletons as skeleton evolution.

where i and j denote the skeleton level, $i, j \in [T_{min}, T_{max}]$. In the following discussion, we denote the skeleton on the ith level of S_1 and on the jth level of S_2 by S_1^i and S_2^j, respectively.

As shown in Eq. 3, we set the constraint condition $i = j$ for $x_a = 1$. This constraint ensures that each skeleton in S_1 is mapped to the skeleton on the same level in S_2 (presented by dash arrows in Figure 9). The rationale behind this is as follows: Firstly, it is very important for our proposed method to reduce the computation complexity since one computation of the similarity between two skeletons takes a long time, so repeating this computation $(T_{max} - T_{min} + 1)^2$ times requires prohibitive computational cost. Secondly, a hierarchical skeleton is organized from simple to complex, and the skeleton on one level is included in the ones on higher levels. In other words, even if one hierarchical level does not offer the best correspondences for endpoints in two skeletons, they could be found on the higher levels. Thirdly, experimental results in Figure 15 illustrate that the performance of finding the global optimum matching operates far less efficiently than our proposed method due to the overfitting problem. Another possible singleton potential is to match skeletons on the levels that are adjacent to the current level. However, as illustrated in Figure 8, many adjacent skeletons

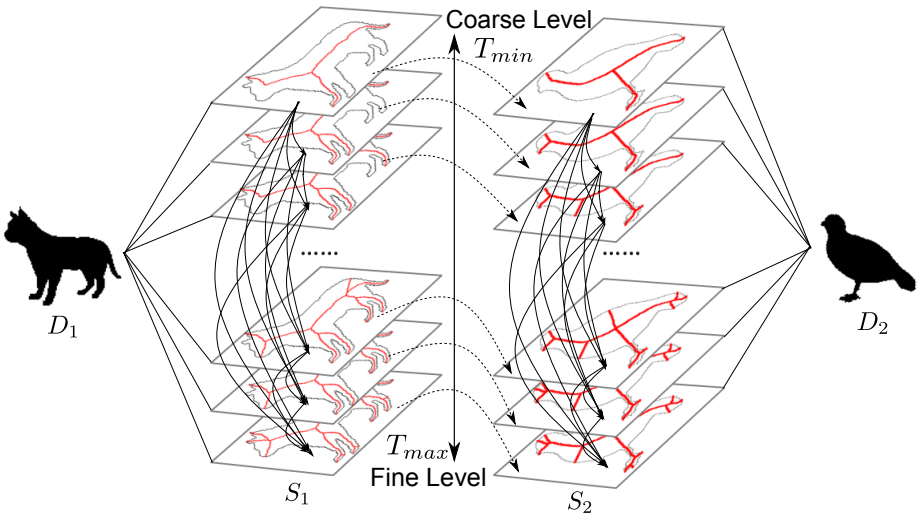


Figure 9: Illustration of the matching algorithm for hierarchical skeletons. All skeletons in S_1 and S_2 are ordered from coarse level (T_{min}) to fine level (T_{max}).

have the same skeleton structure. Moreover, the new added endpoints in the adjacent skeletons could be jumped by the skeleton matching algorithm which we employed for the singleton potential. Thus, we think that matching skeletons on the adjacent levels will not offer significant performance improvement, but it incurs a significant increase of computational cost. Therefore, matching skeletons on the same level is reasonable in terms of both the computational cost and performance.

4.2. Potential Functions

Below, we introduce the singleton potential θ_a and pairwise potential θ_{ab} in Eq. 2.

4.2.1. The Singleton Potential

For each correspondence $(i, j), i, j \in [T_{min}, T_{max}]$, we consider the skeleton graph information to define its singleton potential as in [31]. The idea is to find the best matching between endpoints in two skeletons. The skeleton graphs S_1^i and S_2^j are matched by comparing the geodesic paths between their skeleton endpoints. Then, all the dissimilarity costs between their endpoints are represented as a distance matrix $M(S_1^i, S_2^j)$. The total dissimilarity $c(S_1^i, S_2^j)$ between S_1^i and S_2^j is computed by

searching correspondences between skeleton endpoints with the Hungarian algorithm on $M(S_1^i, S_2^j)$, so that endpoints in S_1^i and S_2^j are matched with the minimal cost. The singleton potential for the correspondence (i, j) is defined as

$$\theta_a = \sigma \cdot c(S_1^i, S_2^j) \quad . \tag{4}$$

where σ is a weighting factor obtained using the arithmetic progression $[\frac{1}{T_{max}-T_{min}+1}, \frac{2}{T_{max}-T_{min}+1}, \cdots, 1]$. In other words, we assign higher weights to coarse-level skeletons to preserve important global shape properties while charging lower weights on fine-level skeletons to allow small local deformations. Although the arithmetic progression uses the same weight for skeletons on a certain level without considering their characteristics, it practically works well. A possible improvement is to estimate optimal weights depending on a given pair of hierarchical skeletons using distance metric learning [46]. In Section 6, we will discuss this extension as our future work.

4.2.2. The Pairwise Potential

The pairwise potential is calculated by comparing skeletons (S_1^i and S_1^j) (solid arrows on S_1 in Figure 9) on two levels in $S_1(D)$ to those (S_2^i and S_2^j) (solid arrows on S_2 in Figure 9) on the respective levels in $S_2(D)$:

$$\theta_{ab} = \tfrac{1}{2} m_{i,j} \quad . \tag{5}$$

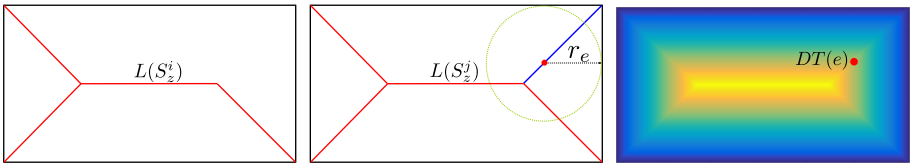
where $i, j \in [T_{min}, T_{max}]$ and $m_{i,j}$ are denoted as the similarity between the pair of S_1^i and S_1^j and the pair of S_2^i and S_2^j:

$$m_{i,j} = \frac{|f(S_1^i, S_1^j) - f(S_2^i, S_2^j)|}{f(S_1^i, S_1^j) + f(S_2^i, S_2^j)} \tag{6}$$

In Eq. 6, $f(S_z^i, S_z^j)(z \in \{1, 2\})$ represents the change from S_z^i to S_z^j. We calculate $f(S_z^i, S_z^j)$ based on the radius and length of new skeleton points between S_z^i and S_z^j (Figure 10(a) and 10(b)). Let N_0 denote the number of skeleton points of the new skeleton branches (e.g. the blue line in Figure 10(b)) from S_z^i to S_z^j. The dissimilarity $f(S_z^i, S_z^j)$ is defined as

$$f(S_z^i, S_z^j) = \sum_{e=1}^{N_0} r_e + \eta \frac{(L(S_z^i) - L(S_z^j))^2}{L(S_z^i) + L(S_z^j)} \quad . \tag{7}$$

where $L(S_z^i)$ and $L(S_z^j)$ denote the length of skeleton S_z^i and that of S_z^j, respectively. In order to make our representation robust to scale changes, the new branch lengths are normalized. r_e denotes the radius (e.g. the dotted arrow in Figure 10(b)) of the tangent disk of a skeleton point that touches the shape boundary in two or more locations. η is the weight factor. This parameter is used for controlling the numeric consistency between the term of radii and the term of skeleton lengths. Without η, the sum of radii will be much larger than the distance between two skeletons and finally dominate $f(S_z^i, S_z^j)$ in Eq. 7. We experimentally set $\eta = 1.2$ [2]. In Eq. 7, the difference between skeleton lengths primarily captures their coarse-grained dissimilarity, while their fine-grained difference is magnified by taking the sum of radii from skeleton points. By fusing two terms, the distinctiveness between two skeletons is more obvious and robust.



(a) S_z^i (red) (b) S_z^j (red and blue) (c) DT (colour map)

Figure 10: Radius and skeleton length. Skeleton S_z^j has one new branch (blue line) compared to skeleton S_z^i. $L(S_z^i)$ is calculated by the length of red lines in (a) while $L(S_z^j)$ is calculated by the length of the red lines plus the length of the blue line in (b). For a skeleton point (red point), r_e is the radius (dotted arrow) of its tangent disk (green dotted circle). r_e is approximately equal to the normalised value of $DT(e)$ in the distance transform matrix DT.

It should be noted that the length ratio pairwise potential have likewise been used extensively [47, 14], however, the sequence of radii is not fully involved. We consider the radius to better represent the reconstruction area between skeletons on different hierarchical levels. For this purpose, we first perform the Distance Transform to compute a matrix DT in which each pixel in the original shape D is characterised by the distance to its closest boundary (Figure 10(c)). Then, for each skeleton point with index e,

[2] In the appendix, we illustrate how performance changes in a figure by varying η.

we select $DT(e)$ as the radius r_e. To make the proposed method invariant to the scale, we normalize r_e in the following way:

$$r_e = \frac{DT(e)}{\frac{1}{N_1}\sum_{t'=1}^{N_1} DT(p_{t'})} \quad . \tag{8}$$

where $p_{t'}(t' = 1, 2, \cdots, N_1)$ varies over all N_1 pixels in D.

Although flexible matchings for measuring skeleton evolution between S_1 and S_2 are possible using tree or graph matching algorithms [31, 48], we decide not to use them for the following reasons: First, tree or graph matching algorithms can flexibly match $S_1^i \in S_1$ and $S_2^j \in S_2$ where i and j do not have to be the same. However, this kind of matching requires examining similarities for many pairs of S_1^i and S_2^j, and what is worse, even computing the similarity for one pair constitutes a high computational cost. Secondly, tree or graph matching methods normally consider the correspondences between endpoints in S_1^i and S_2^j. However, as can be seen in Figure 8, a skeleton on one level is characterised by adding a trivial endpoint to the skeleton on a higher level. Such an endpoint is not useful for describing the transitions of skeletons (i.e. skeleton evolution). Compared to endpoints, we consider changes of length and radius to be a better representation of skeleton evolution, and they cannot be used directly in usual tree or graph matching algorithms.

5. Experiments

In this section, the proposed object representation method is first assessed formed on the singleton potential. Secondly, the usability of skeleton evolution is approved depending on the pairwise potential. Thirdly, the proposed matching algorithm with both potentials is evaluated. Lastly, the implementation and computational complexity of the proposed method is introduced and analysed. The experiments in this paper are performed on two platforms: Laptop and cluster. Hierarchical skeletons are generated on a laptop with Inter Core i7 2.2GHz CPU, 8.00GB memory and 64-bit Windows 8.1 OS. Shape retrieval experiments are accomplished on *Horus*, a cluster provided by the University of Siegen, which includes 136 nodes, each consisting of 2 Intel Xeon X5650 with 2,66 GHz and 48 GB of DDR3 Memory with 1333 MHz. With this cluster, our massive experiments using various datasets and different comparison methods can

be finished efficiently. All methods in our experiments are implemented in Matlab (R2015a on the laptop, and R2014b on *Horus*).

Our evaluation is built on a retrieval framework where shapes in the database are ranked based on their similarity to a query shape. To evaluate the retrieval performance, we use the following measure:

$$y = \frac{1}{100} \sum_{n=1}^{Q} R_n (1 - \frac{n-1}{Q}) \quad . \tag{9}$$

where Q denotes the number of shapes which belong to the same class as the query shape. R_n denotes the number of retrieved shapes that are in the same class as the query in the top-ranked n shapes. The evaluation measure in Eq. 9 is necessary for us to evaluate the retrieval performance accurately using both the number of correct matches and the ranking positions.

5.1. Evaluation of Hierarchical Skeleton-based Representation

This section's premise is to quantitatively assess the influence of skeleton pruning for shape matching as well as the effectiveness of hierarchical skeletons. For the first purpose, we perform skeleton-based object matching on Kimia216 [49] database which contains 216 images from 18 classes. Figure 11 shows two example shapes in each of these 18 classes. We employ a popular skeleton graph matching algorithm proposed by Bai [31]. Table 1 depicts the matching performance of this method where skeletons are generated by DCE-based approach with a fixed and manually tuned stop parameter k. We use each shape as a query and retrieve the 12 most similar shapes among the whole dataset. The final value in each position is counter values that are obtained by checking retrieval results using all the 216 shapes as queries. For example, the fourth position in the row of $k = 3$ shows that from 216 retrieval results in this position, 186 shapes are relevant to the query shapes. Scores in the last column are calculated with Eq. 9. As shown in Table 1, results obtained by the fixed k are much worse than the result reported in [12], as a consequence of the incompleteness of skeletons to represent shapes (e.g. body parts are missing in Figure 4). Like this, the matching performance heavily relies on the quality of skeletons and especially the stop parameter k.

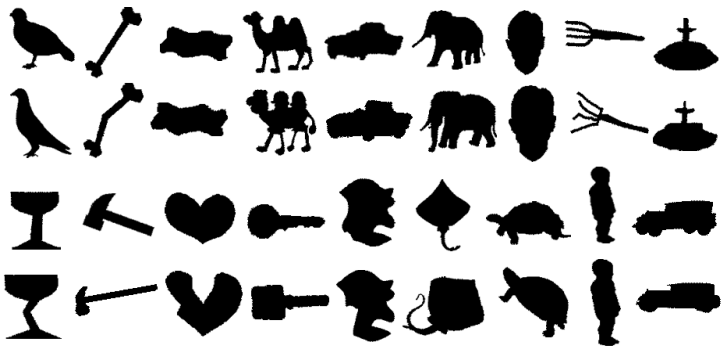


Figure 11: Sample shapes from Kimia216 [49] database.

We use MPEG7 [50] and Animal2000 [51] databases to evaluate the performance of our hierarchical skeleton representation. The total number of images in the MPEG7 [50] database is 1400: 70 classes of various shapes, each class with 20 images (Figure 12). We employ the so-called bulls-eye score [50] for evaluation since this score is generally used by other methods for comparing the retrieval performance on MPEG7 dataset. 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 [50], the 100% bulls-eye score is not possible since some classes contain objects whose shapes are significantly different, making it impossible to group them into the same class using only their shapes. Similar to the Kimia216 dataset, we use the same skeleton pruning algorithm [12] and matching [31] algorithms.

Animal2000 database [51] has 2000 images where of 20 categories, each one consists of 100 images (Figure 13). Since shapes in Animal2000 are obtained from objects in real images, each class is characterised by a large intra-class variation of shapes. In particular, some important shape parts (e.g. legs) are missing, and some shapes have a noisy inside or outside (holes and patches).

Tables 2 and 3 show the performance comparison between single and hierarchical skeleton matching on MPEG7 and Animal2000, respectively. The performance in Table 2 is evaluated using bulls-eye scores, while the one in Table 3 is measured based on

Table 1: Experimental comparison of path similarity skeleton graph matching [31] with different stop parameters k for contour partitioning with DCE. Results are summarised as the number of shapes from the same class among the first top 1-12 shapes. Last row represents the reported results in [31] where k is manually tuned for each shape.

k	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	Score
3	216	203	196	186	182	154	157	132	117	102	91	77	11.3567
4	216	205	190	182	171	157	148	136	126	117	104	102	11.3208
5	216	203	191	187	184	171	165	154	138	136	117	108	11.7908
6	216	205	188	191	178	181	173	148	148	122	115	98	11.8342
7	216	208	204	197	190	181	156	155	135	127	102	92	12.0067
8	216	205	198	195	197	193	189	176	155	126	120	93	12.3783
9	216	209	199	203	193	191	186	177	162	150	131	118	12.4567
10	**216**	**210**	**209**	**204**	**196**	**197**	**176**	**173**	**164**	**152**	**150**	**110**	**12.5817**
11	216	207	204	196	193	189	177	165	155	141	132	92	12.3550
12	216	209	202	197	188	185	162	163	157	140	137	93	12.2375
13	216	205	206	195	192	186	177	167	152	146	120	89	12.3100
14	216	208	204	199	198	184	185	166	153	158	139	103	12.4917
15	216	212	206	201	197	192	187	168	150	131	95	95	12.4608
[31]	216	216	215	216	213	210	210	207	205	191	177	160	13.6983

Eq. 9. For single skeleton matching, we present scores obtained by different stop parameters k. [3,15] in both Tables 2 and 3 illustrates the hierarchical levels $[T_{min}, T_{max}]$ for calculating the singleton potential (without fusing pairwise potentials). For a fair comparison, we use the same skeleton-based matching algorithm proposed by Bai [31] for single and hierarchical skeletons. Tables 2 and 3 clearly indicate that without any human intervention (i.e. manual turning of a stop parameter), the proposed hierarchical skeletons perform better than the traditional single skeletons. In the following discussion, the best scores marked with bold font in Tables 1, 2 and 3 will be used for the comparison of our proposed matching method.

Table 2: Bulls-eye score on MPEG7 dataset using single skeletons with different stop parameters k and hierarchical skeletons within the level [3,15]. Hierarchical skeleton matching outperforms the best single skeleton matching (marked with bold font).

k	3	4	5	6	7	8	9
bulls-eye	0.7125	0.7165	0.7325	0.7421	0.7405	0.7430	0.7516
k	10	**11**	12	13	14	15	[3,15]
bulls-eye	0.7451	**0.7547**	0.7452	0.7455	0.7464	0.7432	0.7884

Table 3: Scores are computed by Eq. 9 on Animal2000 using the single skeletons with different stop parameters k and the proposed hierarchical skeletons within the level [3,15]. Hierarchical skeleton matching outperforms the best single skeleton matching (marked with bold font).

k	3	4	5	6	7	8	9
bulls-eye	231.55	225.98	234.49	228.09	230.73	225.93	232.61
k	10	11	**12**	13	14	15	[3,15]
bulls-eye	234.39	233.97	**241.33**	233.60	239.77	233.67	346.61

5.2. Evaluation of Hierarchical Skeleton Evolutions

In this section, we visually prove that skeleton evolutions for the same category are more similar than those for different ones. To do so, we randomly select nine objects for three different classes (three objects in each class) in Kimia216 dataset [49]. We generate the hierarchical skeleton of each object with k from 3 to 28. As it is composed by single skeletons from coarse to fine levels, we treat it as a sequence. The first class involves bird02, bird07 and bird10 sequences. The second class involves camel12, camel14 and camel17. The third class involves face01, face02, face03. Here, skeletons with $k = 3$ and $k = 28$ are the start and end frames of the sequence, respectively. In each frame in a sequence, we calculate the dissimilarity to the start frame using Eq. 7. This dissimilarity calculation is applied to all nine sequences. We note that shapes in the same class would have different skeleton evolutions due to the scale difference. However, as illustrated in Eq. 7 and Eq. 8, both radii and skeleton lengths have been normalised, the proposed dissimilarity measure for skeleton evolution is scale invariant.

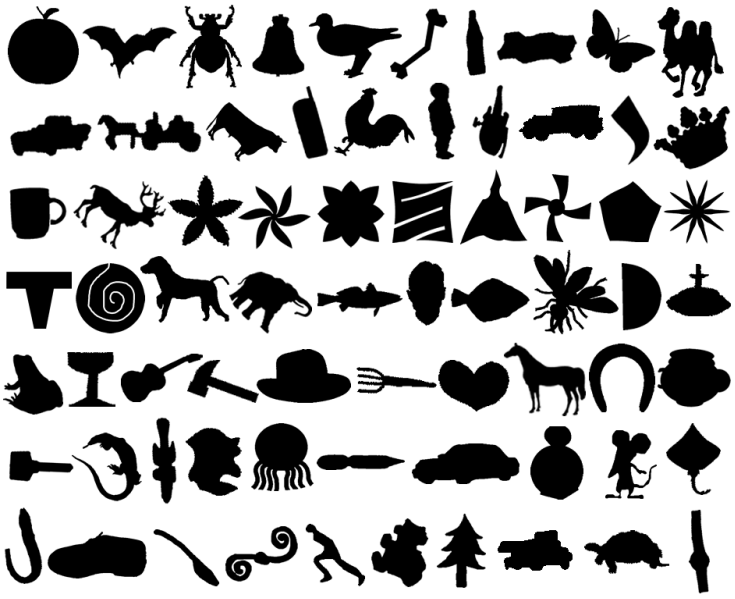


Figure 12: Sample shapes from MPEG7 [50] database.

For each sequence, we plot the dissimilarity between each frame and the start frame in Figure 14. Horizontal and vertical axes represent frame IDs (hierarchical levels) and dissimilarities to the start frame, respectively. For the sequences belonging to the same class, we plot them with the same colour. In Figure 14(a), the first (birds, red) and the second (camels, green) classes are printed. In Figure 14(b), the first and the third (faces, green) classes are printed. For all nine objects, we can clearly observe that 1) the monotonicity of the dissimilarity is evident throughout the sequence as well as 2) dissimilarity changes in the same class are more similar than the ones in different classes. This proves the capacity of skeleton evolutions for distinguishing topologically different shapes.

Moreover, we evaluate the performance improvement using skeleton evolution for object retrieval on Kimia216 dataset. Based on Eq. 9, we calculate matching scores of four methods along with the DCE steps. As shown in Figure 15, these methods are matching with both singleton and pairwise potentials (red bar), matching with singleton potential (blue bar), the globally optimum matching (green bar) and the single skeleton

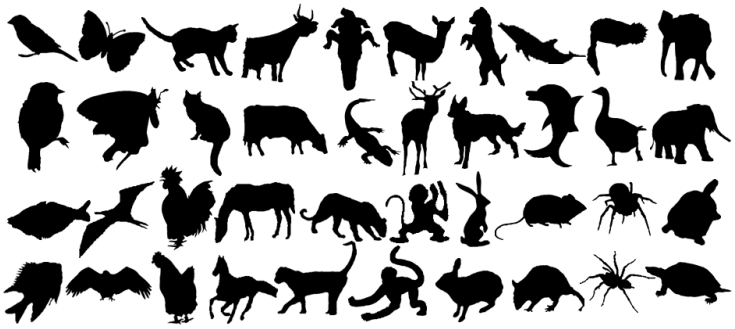
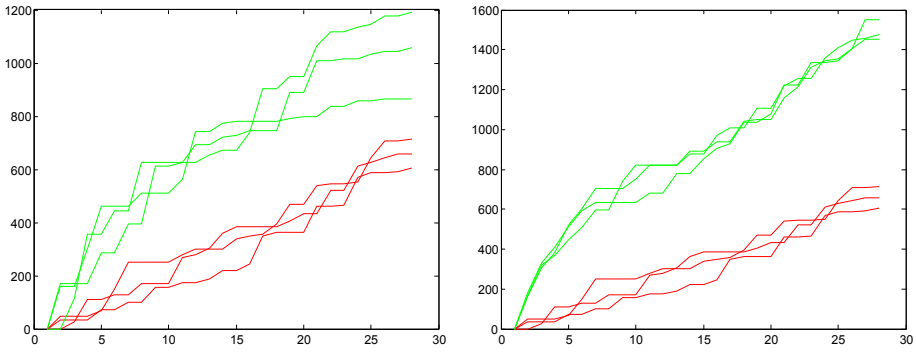


Figure 13: Sample shapes from Animal2000 [51] database.

matching with PS [31] (black bar). For example, on DCE step 5, the score of the red bar is calculated with the hierarchical skeletons from level 3 to 5 using both singleton and pairwise potentials while the blue bar uses only the singleton potential. The score of the green bar is calculated with the best matched skeletons among 3 to 5 levels while the black bar uses the skeletons only on level 5. Figure 15 shows that the globally optimum matching obtains the lowest score among all DCE steps. The main reason for this is the overfitting between skeletons with little endpoints and skeletons with plentiful endpoints. For the PS [31] method, the score is increasing from step 3 until step 10. After that, the score decreases. For the singleton potential and proposed method, scores are gradually increasing. Even the singleton potential performs better than the PS method in the majority of steps. With the benefits from both the singleton and pairwise potentials, the proposed method achieves the highest score. This validates the effectiveness of skeleton evolutions for object matching. In order to improve the matching performance, we can use more levels for constructing hierarchical skeletons. It will, however, incur more time for skeleton matching. Therefore, we need to choose a proper level range of hierarchical skeletons to balance the scores and the matching time. As shown in Figure 15, the matching performance of our method is gradually increased from level 3 to 13 and then becomes stable from level 13 to 15. With this observation, we have selected 13 or its adjacent levels as the general level range for all other experiments.



(a) (b)

Figure 14: Dissimilarities of each frame to the first one in skeleton evolution sequences. (a) Dissimilarities in birds (red) and camels (green) sequences and (b) dissimilarities in birds (red) and faces (green) sequences.

5.3. Hierarchical Skeleton-based Matching

In this section, we evaluate the generality of the proposed method using six datasets: Kimia216 [49], Kimia99 [49], Tari56 [52], Tetrapod, MPEG7 [50] and Animal2000 [51].

Kimia216 Database: Table 4 shows the performance comparison between the proposed method, path similarity (PS1) [31] with the fixed k which achieves the best score in Table 1, path similarity (PS2) with manually tuned k for each shape, inner distance [6] and shape context [2]. As shown in the upper block of Table 4, our method performs better than PS1 with the best stop parameter and close to PS2 with manually tuned k for each shape.

Kimia99 Database: Kimia99 database [49] has images of 9 categories of objects, with 11 images per species for a total of 99 images (Figure 16). In Table 4, we compare our result to the other methods (best score from $k = 12$ with PS [31]). The proposed algorithm significantly outperforms the other methods with a 1.85% improvement over PS1 (with Eq. 9) and close to the PS2 with manually tuned k for each shape.

Tari56 Database: Tari56 database is used for testing the performance on non-rigid objects. It includes 14 classes of articulated shapes with 4 shapes in each class (Figure 17). Although this database was introduced in [52], no result on the whole database is presented. Moreover, there is no detailed explanation of retrieval results on this

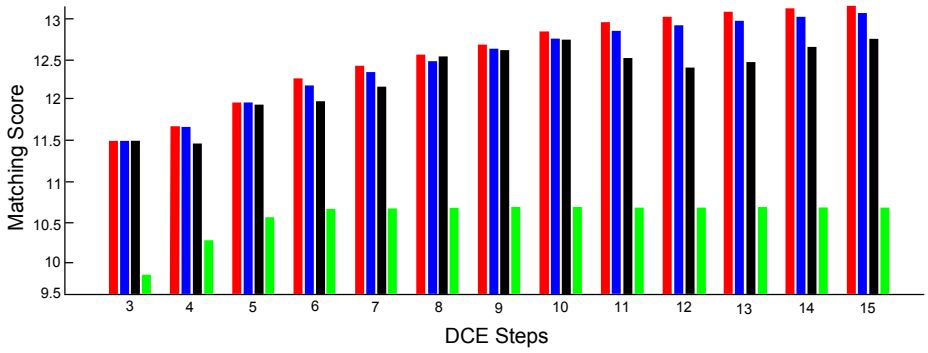


Figure 15: Score comparison between different matching methods: the proposed method (red), the singleton potential (blue), the global optimum matching method (green) and the PS method (black). Horizontal axis represents the hierarchical skeleton levels and vertical axis represents the retrieval score with Eq. 9.

Figure 16: Sample shapes from Kimia99 [49] database. This dataset is different from Kimia216.

dataset using Inner Distance [6], Path Similarity [31] and Shape Context [2] methods. In this paper, we use the source code of Inner Distance and Shape Context methods, implement the Path Similarity method by ourselves, and apply all these methods to Tari56 database. Since there are some parameters involved by Inner Distance and Shape Context methods, we optimise them by a combination of two heuristic optimisation methods: Gradient Hill Climbing [44] integrated with Simulated Annealing [45]. The retrieval results are reported in Table 5 (the result of PS is obtained by the best stop parameter $k = 10$). We can observe that the proposed method performs well (1.96% better than the PS method) in the presence of non-rigid deformations. This is because hierarchical skeletons can capture non-rigid features of shapes by multiple skeletons, which have higher discrimination power than single skeletons used in the other methods.

Tetrapod Database: Tetrapod database includes 120 visually similar tetrapod animals

Table 4: Experimental comparison of our method to state-of-the-art methods on Kimia216 and Kimia99 datasets. PS1 denotes the retrieval results with fixed k which achieves the highest score with Eq. 9. PS2 denotes the reported results in [31] with manually tuned k for each shape.

Kimia216	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
ID [6]	216	198	189	176	167	156	136	130	122	118	108
SC [2]	204	199	192	187	185	181	175	166	160	163	155
PS1	216	210	209	204	196	197	176	173	164	152	150
PS2 [31]	216	216	215	216	213	210	210	207	205	191	177
Ours	**216**	**216**	**213**	**212**	**209**	**197**	**196**	**192**	**193**	**172**	**169**

Kimia99	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
ID [6]	99	97	92	89	85	85	76	75	63	53
SC [2]	99	97	91	88	84	83	76	76	68	62
PS1	99	97	97	97	96	92	93	81	71	68
PS2 [31]	99	99	99	99	96	97	95	93	89	73
Ours	**99**	**99**	**99**	**96**	**94**	**95**	**91**	**89**	**85**	**77**

with 6 classes, such as camel, cattle, deer, dog, elephant and horse (Figure 18). We aim to evaluate the ability of matching methods for fine-grained shapes where some species are really difficult to distinguish (e.g. horses and dogs). The main skeleton structures among all these objects are quite similar. Therefore, in order to improve the accuracy for object matching, fine-grained skeleton branches should be considered. As shown in Table 6, the proposed method achieved the best results among all the other methods (8.96% better than the best score from ID [6] with Eq. 9). Although we only use 15 DCE levels for the experiment, there are still some levels unused but with essential shape features. Therefore, we may improve the overall performance by integrating edit distance [14] for measuring fine-grained deformations using these unused hierarchical levels.

MPEG7 Database: For comparison, Table 7 lists several reported results and the results by our proposed method on MPEG7 dataset. We cluster them into two groups: pairwise matching and context-based matching. In the first group, results are decided

Table 5: Results comparison on Tari56 dataset.

Tari56	1st	2nd	3rd	4th
ID [6]	56	46	37	28
SC [2]	52	17	10	10
PS [31]	56	49	44	40
Ours	**56**	**51**	**50**	**33**

Table 6: Experimental comparison on Tetrapod dataset.

Tetrapod	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
ID [6]	120	118	106	101	90	83	77	69	70	56
SC [2]	100	80	70	53	53	51	40	28	27	27
PS [31]	120	109	101	98	81	78	68	66	65	59
Ours	**120**	**118**	**106**	**100**	**95**	**90**	**84**	**71**	**83**	**81**
	11th	12th	13th	14th	15th	16th	17th	18th	19th	20th
ID [6]	57	45	38	29	41	35	26	27	30	21
SC [2]	29	27	25	32	32	23	31	26	20	28
PS [31]	59	49	50	42	43	35	39	31	33	36
Ours	**68**	**73**	**67**	**77**	**68**	**67**	**60**	**51**	**56**	**43**

by the similarity measures for shape pairs. In the second group, results are generated by considering the underlying structure of the shape manifold [53] 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. The proposed Hierarchical Skeleton (HS) with both singleton and pairwise potentials achieves a 81.62% bulls-eye score that is significantly better than those of traditional skeleton-based methods [3, 12]. This is because a single skeleton has limited ability to capture geometric properties at different levels of resolution. However, this approach performs not as well as Shape Tree [8] and Height Functions (HF) [9], etc. The main reason for this is the maximal DCE levels for our experiment being limited to 15 while the performance can be improved by using more DCE levels.

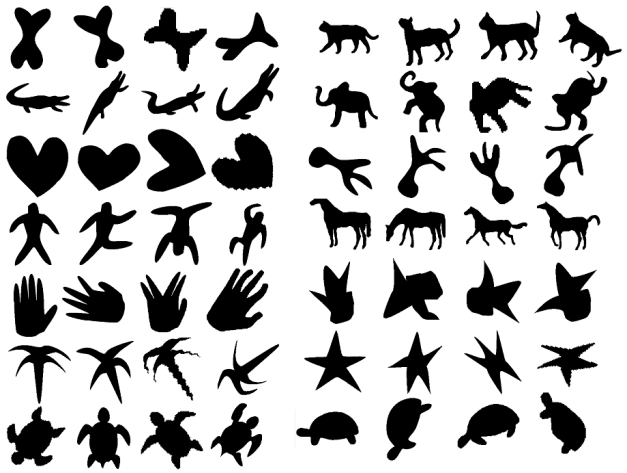


Figure 17: Full shapes from Tari56 [52] database.

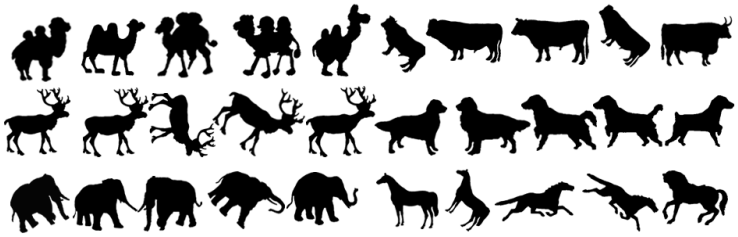


Figure 18: Sample shapes from Tetrapod database.

In the group of context-based methods, we employ a simple and fast Mutual kNN Graph method [54] based on the HS similarity scores between all the shapes. This method captures the manifold structure by defining a neighbourhood for each shape. Our method, which achieves 99.21% bulls-eye score, outperforms most state-of-the-art methods. It is important to mention that Donoser et al. [55] proposed a generic framework for diffusion processes in the scope of retrieval applications which achieved a 100% accuracy on MPEG7 dataset. However, as illustrated in Table 7, our performance is very close to the 100% only using the Mutual kNN Graph method.

Animal2000 Database: As shown in Table 8, our method performs better than the Shape Context [2] and Path Similarity [12] methods. However, there is still some space to improve our method. In particular, the shape noise has a bad influence on the

Table 7: Bulls-eye score on the MPEG7 Dataset. HS denotes the hierarchical skeleton method.

Pairwise Matching	Score	Context-based	Score
Shape Contexts [2]	76.51%	INSC + CDM [56]	88.30%
Skeletal Context [3]	79.92%	IDSC + LP [57]	91.00%
Optimized CSS [4]	81.12%	SC + LP [57]	92.91%
Multiscale Rep. [5]	84.93%	IDSC + LCDP [58]	93.32%
Shape LAneRouge [59]	85.25%	SC + GM + Meta [60]	92.51%
Fixed Cor. [61]	85.40%	IDSC + Mutual Graph [54]	93.40%
Inner Distance [6]	85.40%	IDSC + PS + LDCP [62]	95.60%
Symbolic Rep. [63]	85.92%	ASC + LDCP [64]	95.96%
Hier.Procrustes [7]	86.35%	HF + LCDP [9]	96.45%
Triangle Area [65]	87.23%	SC + IDSC + Co-T [53]	97.72%
Shape Tree [8]	87.70%	SC + DDGM + Co-T [53]	97.45%
HF [9]	89.66%	AIR [66]	93.67%
IP [67]	80.28%	IP+HG [67]	96.43%
Path Similarity [12]	75.16%	ASC + TN + TPG [68]	96.47%
HS	**81.62%**	**HS + Mutual Graph**	**99.21%**

skeleton generation and pruning. This is why the retrieval score of Inner-Distance [6], which is robust to shape noise, is higher than our method's score. In addition, some shapes in the same class are significantly different (Figure 13). Therefore, it is not possible to group them into the same class only using shape distance methods. In the future, we will use this dataset for evaluating the tasks with a supervised learning process, e.g. shape-based object classification using hierarchical skeletons.

Table 8: Retrieval score comparison on Animal2000 dataset.

Method	Inner Dis. [6]	Shape Context [2]	Path Similarity [12]	**Our**
Score	452.66	193.79	241.33	**348.73**

5.4. Implementation and Computational Complexity

Here, we briefly describe all the implementation steps: first, we compute the initial skeletons with the method in [26]. After that, we generate hierarchical skeletons using the skeleton pruning method in Section 3. Then, we match the hierarchical skeletons by two potentials. For the singleton potential, we employ the skeleton matching method in which Dijkstra's shortest path algorithm [69] is employed to build the skeleton graph [31]. For the pairwise potential, skeleton lengths and skeleton point radius are calculated to obtain the distance between two skeletons on different hierarchical levels. Finally, the total costs between hierarchical skeletons are computed with the proposed method.

We now analyse the computational complexity of the proposed hierarchical skeleton generation and matching approaches. (1) For initial skeleton generation, the time complexity is in the order of $O(8n')$, where n' is the number of points in the planar shape D. This is because in [26], to determine whether a pixel point in D is a skeleton point, the corresponding nearest contour point for each of the 8 neighbouring points is determined. (2) For hierarchical skeleton generation, the time complexity is $O(v \log v)$, where v is the number of the vertices on the original polygon. This is because, as introduced in [12], the skeleton pruning based on DCE has a complexity of $O(v \log v)$. (3) For hierarchical skeleton matching, we analyse the computational complexity by different potentials. Assuming that k_1 and k_2 are the number of all nodes (skeleton endpoints and junction points) in two single skeletons, the time complexity for computing the singleton potential is $O(k_1^2 k_2^2)$ [31]. As we employ $T_{max} - T_{min} + 1$ hierarchical levels for the singleton matching, the total complexity is $O(T_{max} - T_{min} + 1) \times O(k_1^2 k_2^2)$. Since $T_{max} - T_{min} + 1$ is constant, the total complexity for the singleton potential is $O(k_1^2 k_2^2)$. For the pairwise potential, the time for computing DT is $O(n')$ by the approach proposed in [70]. With DT, skeleton evolution is computed by considering all possible pairs of skeletons on two different levels, the time complexity is $O(C_2^{T_{max}-T_{min}+1}) = O((T_{max}-T_{min}+1)(T_{max}-T_{min})/2)$. Considering these two parts, the pairwise potential runs in $O(n')$ + $O(C_2^{T_{max}-T_{min}+1})$ time. Since $(T_{max} - T_{min} + 1)$ and $(T_{max} - T_{min})$ are constant and small, the total computational complexity of the pairwise potential is $O(n')$.

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 initial skeleton generation, the mean time is 0.07 seconds on each shape. With an initial skeleton, the mean time for hierarchical skeleton generation is 0.97 seconds. Given two hierarchical skeletons with levels $[3, 15]$, the mean matching time is 11.29 seconds. Please notice that our code is not optimised, and its faster implementation is possible by optimising loops, settings and programming language, etc. Thus, there are still plenty of opportunities to reduce the running time.

6. Conclusion and Future Work

In this paper, a novel object matching method based on hierarchical skeletons is presented. The main idea is to reuse skeleton branches pruned in the skeleton pruning process since they contain some fine-grained geometric and topological information of original shapes. Based on this, we proposed a hierarchical skeleton as a shape representation to organise multiple skeletons obtained by the skeleton pruning process. We also developed a matching method which considers similarities for both single skeletons and skeleton pairs (skeleton evolution) in a hierarchical skeleton. The experiments on six datasets demonstrate that our method is significantly superior to most conventional shape descriptors and the single skeleton-based methods. In the future, we will focus on optimising the singleton and pairwise potentials. For the singleton potential, we will check the distance metric learning [46, 71] approach to obtain the optimal weights. Specifically, given a collection of dissimilarity values between skeletons from each level among two hierarchical skeletons, appropriate projections can be extracted by analysing the relations between dissimilarity values and their hierarchical levels. Essentially, these projects are used in such a way that the distances of similar-skeleton and those of dissimilar-skeleton are preserved and enlarged, respectively. With these projections, the optimal weight on each hierarchical level can be estimated. For the pairwise potential, we will evaluate the strategy of matching skeletons on the levels that are adjacent to the current level. In addition, we will also try to fuse some local features to improve the accuracy of object retrieval on fine-grained shapes.

Appendix: In this appendix, we demonstrate how performance changes by varying η. Specifically, we conduct shape retrieval on Kimia216 dataset [49] using different η values. In order to eliminate the influence of other factors, the rest of parameters are fixed. This experiment demonstrates a means of how a proper η value is obtained. As shown in Figure A1, $\eta = 1.2$ leads the best performance and we use this for the whole experiments in Section 5.

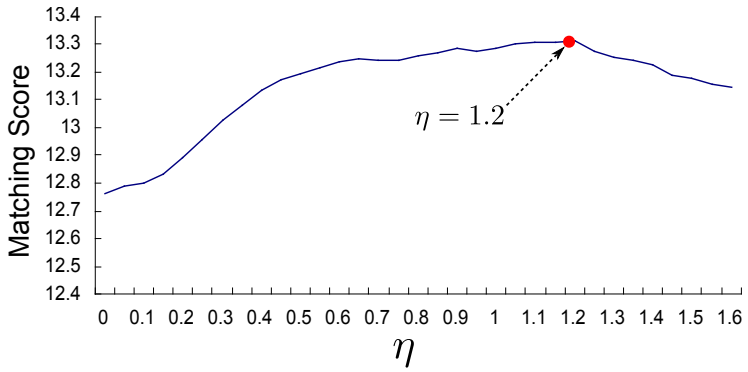


Figure A1: Transition of shape retrieval performances where the horizontal and vertical axes represent η values and retrieval scores computed based on Eq. 9.

Acknowledgements: 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] M. Yang, K. K. Idiyo, R. Joseph, A survey of shape feature extraction techniques, Pattern Recognition (2008) 43–90.

[2] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.

[3] J. Xie, P.-A. Heng, M. Shah, Shape matching and modeling using skeletal context, Pattern Recognition 41 (5) (2008) 1756–1767.

[4] F. Mokhtarian, M. Bober, Curvature scale space representation: Theory, applications and mpeg-7 standardization, in: Computational Imaging and Vision, Kluwer Academic Pub, United States, 2003.

[5] T. Adamek, N. O'Connor, A multiscale representation method for nonrigid shapes with a single closed contour, IEEE Transactions on Circuits and Systems for Video Technology 14 (5) (2004) 742–753.

[6] H. Ling, D. Jacobs, Shape classification using the inner-distance, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2) (2007) 286–299.

[7] G. McNeill, S. Vijayakumar, Hierarchical procrustes matching for shape retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 885–894.

[8] P. F. Felzenszwalb, J. Schwartz, Hierarchical matching of deformable shapes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[9] J. Wang, X. Bai, X. You, W. Liu, L. J. Latecki, Shape matching and classification using height functions, Pattern Recognition Letters 33 (2) (2012) 134–143.

[10] B.-W. Hong, S. Soatto, Shape matching using multiscale integral invariants, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (1) (2015) 151–160.

[11] M. Baust, L. Demaret, M. Storath, N. Navab, A. Weinmann, Total variation regularization of shape signals, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2075–2083.

[12] X. Bai, L. Latecki, W. yu Liu, Skeleton pruning by contour partitioning with discrete curve evolution, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 449–462.

[13] W. Zhang, P. Srinivasan, J. Shi, Discriminative image warping with attribute flow, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2393–2400.

[14] A. Torsello, E. R. Hancock, A skeletal measure of 2d shape similarity, Computer Vision and Image Understanding 95 (1) (2004) 1–29.

[15] E. R. Davies, Machine Vision: Theory, Algorithms, Practicalities, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[16] T. B. Sebastian, B. B. Kimia, Curves vs. skeletons in object recognition, Signal Processing 85 (2) (2005) 247–263.

[17] W. Shen, X. Bai, R. Hu, H. Wang, L. J. Latecki, Skeleton growing and pruning with bending potential ratio, Pattern Recognition 44 (2) (2011) 196–209.

[18] J. Hu, X. Peng, C. Fu, A novel description based on skeleton and contour for shape matching, in: International Symposium on High-Power Laser Systems and Applications, 2014, pp. 1–9.

[19] H. Chatbri, K. Kameyama, P. Kwan, A comparative study using contours and skeletons as shape representations for binary image matching, Pattern Recognition Letters (2015) pages to appear.

[20] L. Lam, S.-W. Lee, C. Suen, Thinning methodologies-a comprehensive survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (9) (1992) 869–885.

[21] N. Mayya, V. T. Rajan, Voronoi diagrams of polygons: A framework for shape representation, in: IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 638–643.

[22] L. Gorelick, M. Galun, E. Sharon, R. Basri, A. Brandt, Shape representation and classification using the poisson equation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 1991–2005.

[23] P. Dimitrov, J. N. Damon, K. Siddiqi, Flux invariants for shape, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 835–841.

[24] R. Ogniewicz, O. Kbler, Hierarchic voronoi skeletons, Pattern Recognition 28 (3) (1995) 343–359.

[25] B. H. Shekar, B. Pilar, Shape representation and classification through pattern spectrum and local binary pattern - a decision level fusion approach, in: International Conference on Signal and Image Processing, 2014, pp. 218–224.

[26] W. Choi, K. Lam, W. Siu, Extraction of the euclidean skeleton based on a connectivity criterion, Pattern Recognition 36 (3) (2003) 721–729.

[27] H. J. Chang, Y. Demiris, Unsupervised learning of complex articulated kinematic structures combining motion and skeleton information, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3138–3146.

[28] J. Kustra, A. Jalba, A. Telea, Probabilistic view-based 3d curve skeleton computation on the gpu, in: International Conference on Computer Vision Theory and Applications, 2013, pp. 237–246.

[29] A. Telea, J. J. van Wijk, An augmented fast marching method for computing skeletons and centerlines, in: Proceedings of the Symposium on Data Visualisation, Eurographics Association, 2002, pp. 251–ff.

[30] G. Borgefors, G. Ramella, G. Sanniti di Baja, Hierarchical decomposition of multiscale skeletons, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1296–1312.

[31] X. Bai, L. Latecki, Path similarity skeleton graph matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7) (2008) 1282–1292.

[32] S. M. Pizer, O. R. William, S. H. Bloomberg, Hierarchical shape description via the multiresolution symmetric axis transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (4) (1987) 505–511.

[33] D. Macrini, S. Dickinson, D. Fleet, K. Siddiqi, Bone graphs: Medial shape parsing and abstraction, Computer Vision and Image Understanding 115 (7) (2011) 1044–1061.

[34] D. Macrini, S. Dickinson, D. Fleet, K. Siddiqi, Object categorization using bone graphs, Computer Vision and Image Understanding 115 (8) (2011) 1187–1206.

[35] D. Macrini, K. Siddiqi, S. Dickinson, From skeletons to bone graphs: Medial abstraction for object recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[36] C. Yang, M. Grzegorzek, E. Lukasik, Representing the evolving temporal envelope of musical instruments sounds using computer vision methods, in: Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2015, pp. 76–80.

[37] G. Bal, J. Diebold, E. Chambers, E. Gasparovic, R. Hu, K. Leonard, M. Shaker, C. Wenk, Skeleton-based recognition of shapes in images via longest path matching, in: Research in Shape Modeling, Vol. 1, 2015, pp. 81–99.

[38] W. Shen, Y. Wang, X. Bai, H. Wang, L. J. Latecki, Shape clustering: Common structure discovery, Pattern Recognition 46 (2) (2013) 539–550.

[39] C. Yang, O. Tiebe, P. Pietsch, C. Feinen, U. Kelter, M. Grzegorzek, Shape-based object retrieval by contour segment matching, in: IEEE International Conference on Image Processing, 2014, pp. 2202–2206.

[40] R. L. Ogniewicz, Skeleton-space: A multiscale shape description combining region and boundary information, in: IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 746–751.

[41] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S. Zucker, Shock graphs and shape matching, International Journal of Computer Vision 35 (1) (1999) 13–32.

[42] M. Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, L. Bretzner, Many-to-many feature matching using spherical coding of directed graphs, in: European Conference on Computer Vision, 2004, pp. 322–335.

[43] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (9) (2012) 1704–1716.

[44] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall Press, 2009.

[45] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science (1983) 671–680.

[46] E. P. Xing, A. Y. Ng, M. I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Advances in Neural Information Processing Systems, 2003, pp. 505–512.

[47] P. Dimitrov, C. Phillips, K. Siddiqi, Robust and efficient skeletal graphs, in: IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 417–423.

[48] T. Liu, D. Geiger, Approximate tree matching and shape similarity, in: IEEE International Conference on Computer Vision, 1999, pp. 456–462.

[49] T. B. Sebastian, P. Klein, B. Kimia, Recognition of shapes by editing their shock graphs, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 550–571.

[50] L. J. Latecki, R. Lakamper, T. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 424–429.

[51] X. Bai, W. Liu, Z. Tu, Integrating contour and skeleton for shape classification, in: IEEE International Conference on Computer Vision, 2009, pp. 360–367.

[52] C. Asian, S. Tari, An axis-based representation for recognition, in: IEEE International Conference on Computer Vision, Vol. 2, 2005, pp. 1339–1346.

[53] X. Bai, B. Wang, C. Yao, W. Liu, Z. Tu, Co-transduction for shape retrieval, IEEE Transactions on Image Processing 21 (5) (2012) 2747–2757.

[54] P. Kontschieder, M. Donoser, H. Bischof, Beyond pairwise shape similarity analysis, in: Asian Conference on Computer Vision, 2010, pp. 655–666.

[55] M. Donoser, H. Bischof, Diffusion processes for retrieval revisited, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1320–1327.

[56] H. Jegou, C. Schmid, H. Harzallah, J. Verbeek, Accurate image search using the contextual dissimilarity measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 2–11.

[57] X. Bai, X. Yang, L. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (5) (2010) 861–874.

[58] X. Yang, S. Koknar-Tezel, L. Latecki, Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 357–364.

[59] A. Peter, A. Rangarajan, J. Ho, Shape iane rough: Sliding wavelets for indexing and retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[60] A. Egozi, Y. Keller, H. Guterman, Improving shape retrieval by spectral matching and meta similarity, IEEE Transactions on Image Processing 19 (5) (2010) 1319–1327.

[61] B. J. Super, Retrieval from shape databases using chance probability functions and fixed correspondence, International Journal of Pattern Recognition and Artificial Intelligence 20 (8) (2006) 1117–1138.

[62] A. Temlyakov, B. Munsell, J. Waggoner, S. Wang, Two perceptually motivated strategies for shape classification, IEEE Conference on Computer Vision and Pattern Recognition (2010) 2289–2296.

[63] M. R. Daliri, V. Torre, Robust symbolic representation for shape recognition and retrieval, Pattern Recognition 41 (5) (2008) 1782–1798.

[64] H. Ling, X. Yang, L. Latecki, Balancing deformability and discriminability for shape matching, in: European Conference on Computer Vision, 2010, pp. 411–424.

[65] N. Alajlan, M. Kamel, G. Freeman, Geometry-based image retrieval in binary image databases, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (6) (2008) 1003–1013.

[66] R. Gopalan, P. Turaga, R. Chellappa, Articulation-invariant representation of non-planar shapes, in: European Conference on Computer Vision, 2010, pp. 286–299.

[67] C. Yang, C. Feinen, O. Tiebe, K. Shirahama, M. Grzegorzek, Shape-based object matching using point context, in: International Conference on Multimedia Retrieval, 2015, pp. 519–522.

[68] X. Yang, L. Prasad, L. J. Latecki, Affinity learning with diffusion on tensor product graph, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 28–38.

[69] R. A. Valenzano, S. J. Arfaee, J. T. Thayer, R. Stern, N. R. Sturtevant, Using alternative suboptimality bounds in heuristic search, in: International Conference on Automated Planning and Scheduling, 2013, pp. 233–241.

[70] J. Maurer, C.R., R. Qi, V. Raghavan, A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2) (2003) 265–270.

[71] G. Zhang, J. Kato, Y. Wang, K. Mase, Adaptive metric learning in local distance comparison for people re-identification, in: Asian Conference on Pattern Recognition, 2013, pp. 196–200.