



Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrecShape recognition by bag of skeleton-associated contour parts[☆]Wei Shen^a, Yuan Jiang^a, Wenjing Gao^a, Dan Zeng^{a,*}, Xinggong Wang^b^aKey Lab of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200072, China^bSchool of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Shape recognition
Skeleton-associated contour parts
Bag of features

ABSTRACT

Contour and skeleton are two complementary representations for shape recognition. However combining them in a principal way is nontrivial, as they are generally abstracted by different structures (closed string vs graph), respectively. This paper aims at addressing the shape recognition problem by combining contour and skeleton according to the correspondence between them. The correspondence provides a straightforward way to associate skeletal information with a shape contour. More specifically, we propose a new shape descriptor, named **Skeleton-associated Shape Context (SSC)**, which captures the features of a contour fragment associated with skeletal information. Benefited from the association, the proposed shape descriptor provides the complementary geometric information from both contour and skeleton parts, including the spatial distribution and the thickness change along the shape part. To form a meaningful shape feature vector for an overall shape, the Bag of Features framework is applied to the SSC descriptors extracted from it. Finally, the shape feature vector is fed into a linear SVM classifier to recognize the shape. The encouraging experimental results demonstrate that the proposed way to combine contour and skeleton is effective for shape recognition, which achieves the state-of-the-art performances on several standard shape benchmarks.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Shape is a significant cue in human perception for object recognition. The objects shown in Fig. 1 have lost their brightness, color and texture information and are only represented by their silhouettes, however it's not intractable for human to recognize their categories. This simple demonstration indicates that shape is stable to the variations in object color and texture and light conditions. Due to such advantages, recognizing objects by their shapes has been a long standing problem in the literature. Shape recognition is usually considered as a classification problem that is given a testing shape, to determine its category label based on a set of training shapes as well as their category label. The main challenges in shape recognition are the large intra-class variations induced by deformation, articulation and occlusion. Therefore, the main focus of the research efforts have been made in the last decade [5,6,8,27,42,43] is how to form an informative and discriminative shape representation.

Generally, the existing main stream shape representations can be classified into two classes: contour based [8,21,27] and

skeleton based [1,3,34,40,41,46]. The former one delivers the information that how the spatial distribution of the boundary points varies along the object contour. Therefore, it captures more informative shape information and is stable to affine transformation. However, it is sensitive to non-ridge deformation and articulation; On the contrary, the latter one provides the information that how thickness of the object changes along the skeleton. Therefore, it is invariant to non-ridge deformation and articulation, although it only carries more rough geometric features of the object. Consequently, such two representations are complementary. Nevertheless, very few works have tried to combine these two representations for shape recognition. The reason might be that combining the data of different structures is not trivial, as the contour is always abstracted by a closed string while the skeleton is abstracted either by a graph or a tree. Consequently, the matching methods [13,17,19,28–31] for these two data abstraction are different. ICS [5] is the first work to explicitly discuss how to combine contour and skeleton to improve the performance of shape recognition. However, the combination proposed in this work is just a weighted sum of the outputs of two generative models trained individually on contour features and skeleton features respectively. Therefore, how to combine contour and skeleton into a shape representation in a principled way is still an open problem.

In this paper, our goal is to address the above combination issue to explore the complementarity between contour and

[☆] This paper has been recommended for acceptance by Chennai Guest Editor.

* Corresponding author. Tel.: +86 13122300551.

E-mail address: dzeng_shu@outlook.com (D. Zeng).



Fig. 1. Human biological vision system is able to recognize these object without any appearance information (brightness, color and texture).

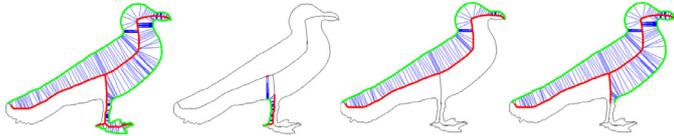


Fig. 2. Some corresponding contour and skeleton parts, marked in green and red, respectively. The corresponding contour and skeleton points are linked by blue lines.

skeleton to improve the performance of shape recognition. The main obstacle of the combination is the data structures of contour and skeleton are different (closed string vs graph). A contour is usually described by the features of its parts (contour fragments) [21,42]. As the correspondence between contour points and skeleton points can be obtained easily, for each contour point, we can associate the geometric information of its corresponding skeleton point with it. In this way, we can record the change of the object thickness, i.e., the skeleton radius, along each contour fragment. Such association actually leads to the combination of contour and skeleton on part level (Fig. 2 shows some corresponding contour and skeleton parts. Note that, a contour fragment may correspond to more than one skeleton segments, such as the second example in Fig. 2). Therefore, combing contour and skeleton on part level is a feasible way.

With the extra information provided by skeleton, inspired by the well known descriptor Shape Context (SC) [8], we propose to encode the features of a contour point into a 3D tensor, in which the three dimensions describe the Euclidean distances, orientations and thickness differences between the contour points and others in the fragment, respectively. Intuitively, the proposed new descriptor extends SC by including the extra information, object thickness, provided by skeleton. Therefore, it is more informative; Essentially, this new descriptor is formed by concatenating the SC descriptors of the sub-parts of the contour fragment separated according to thickness information. Such sub-parts based representation capture fine level geometric information, so it is more discriminative. Fig. 3 illustrates the new descriptor for a contour point in a contour fragment, in which the sub-parts of the contour fragment are marked by different colors and the sub-part and its SC descriptor are marked by the same color. This new shape descriptor is termed as Skeleton-associated Shape Context (SSC), as it associates the skeletal information with the contour descriptor.

Following the framework of the recent work Bag of Contour Fragments (BCF) [45], we can obtain a shape feature vector of an overall shape by encoding and then pooling the SSC descriptors extracted from it. We term our method as Bag of Skeleton-associated Contour Parts (BSCP), as it associates skeletal information with contour fragments and encodes the shape features from shape part level. Fig. 4 shows the pipeline of building a shape feature vector by BSCP. Given a shape, firstly a normalization step is performed to align the shape according to its major axis (Fig. 4(b)), as the Spatial Pyramid Matching (SPM) [25] step (Fig. 4(g)) is not rotation invariant. Then, the skeleton of the shape is extracted and the contour of the shape is decomposed into contour fragments (Fig. 4(c)). Each contour point is associated with an object thickness value, i.e., the radius of its corresponding skeleton point. A shape part is then described by the contour fragment associated with the ob-

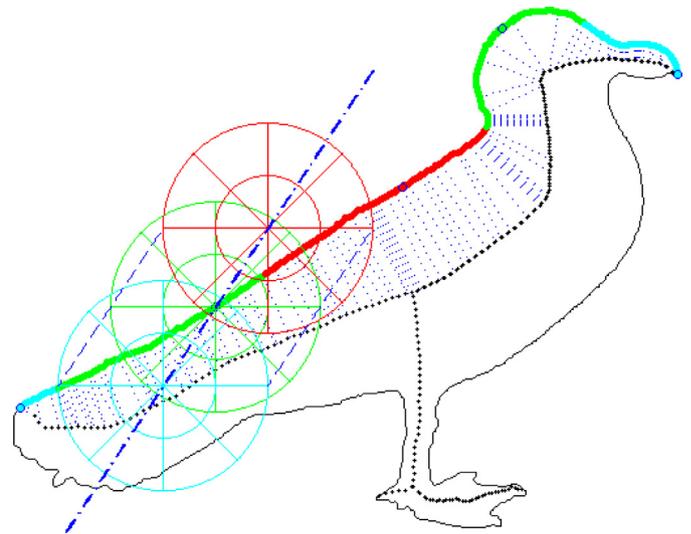


Fig. 3. The Skeleton-associated Shape Context descriptor of a contour point in a contour fragment, which is a 3D tensor to describe the Euclidean distances, orientations and thickness differences between the contour point and others in the fragment. It equals to the concatenated shape context descriptors [8] computed on sub-parts (marked by different colors) separated according to the object thickness differences between the contour point and others in the fragment.

ject thickness values provided by its corresponding skeleton segments (Fig. 4(d)). After that, each shape part is represented by concatenating the SSC descriptors extracted on its reference points (Fig. 4(e)), and then encoded into shape codes (Fig. 4(f)). To encode shape parts, we adopt local-constrained linear coding (LLC) [44] scheme, as it has been proved to be efficient and effective for image classification. Finally, the shape codes are pooled into a compact shape feature vector by SPM (Fig. 4(h)). The obtained shape feature vectors can be fed into any discriminative models, such as SVM and Random Forest, to perform shape classification. Using such discriminative models for shape recognition is more efficient than traditional shape classification methods, as the latter require time consuming matching and ranking steps.

Our contributions can be summarized in three aspects. First, we propose a natural way to associate a shape contour with skeletal information. Second, we propose a new shape descriptor which encodes the shape features from a contour fragment associated with skeletal information. Last, our method, Bag of Skeleton-associated Contour Parts achieves the state-of-the-arts on several shape benchmarks.

The remainder of this paper is organized as follows. Section 2 reviews the works related to shape recognition. Section 3 introduces the proposed shape descriptor as well as our framework for shape recognition. Experimental results and analysis on several shape benchmarks are shown in Section 4. Finally, we draw the conclusion in Section 5.

Our preliminary work [39] also combines contour and skeleton for shape recognition, while the difference to this paper is obvious. Rather than simply concatenating the contour and skeleton features on mid-level, this paper associates skeletal information with a shape contour on low-level by making full use of the natural correspondence between a contour and its skeleton.

2. Related work

There have been a rich body of works concerning shape recognition in recent years [7–10,16,20,22,27,42,43]. In the early age, the exemplar-based strategy has been widely used, such as [8,27]. Generally, there are two key steps in this strategy. The first one is extracting informative and robust shape descriptors. For example,

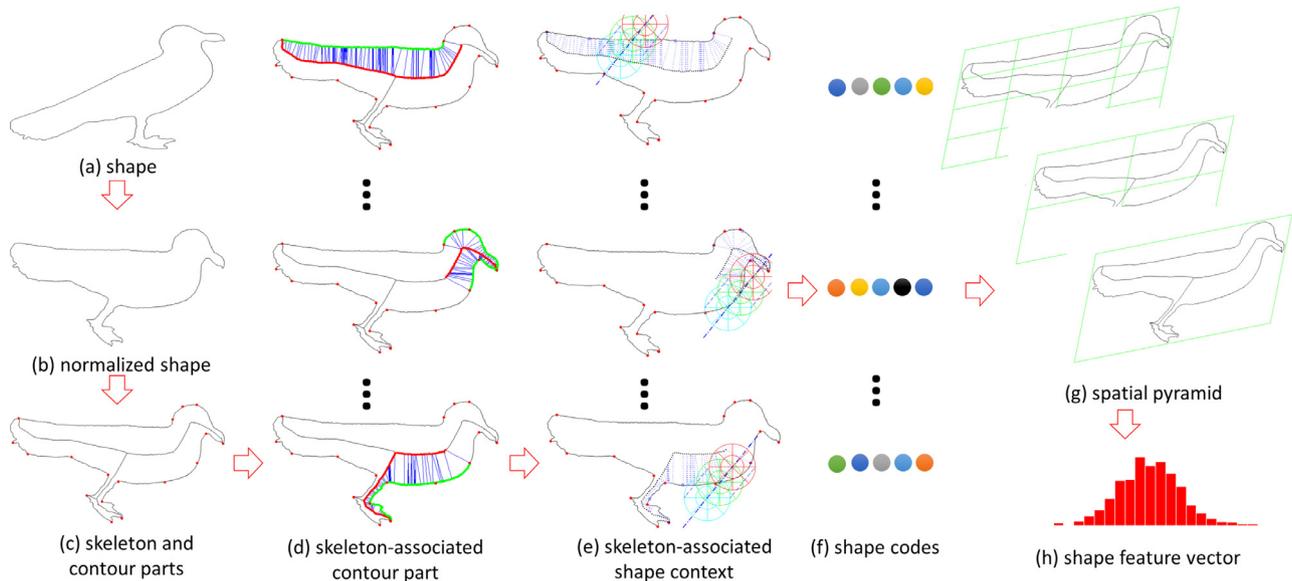


Fig. 4. The pipeline of building a shape feature vector by bag of skeleton-associated contour parts.

Belongie et al. [8] introduce a shape descriptor named shape context (SC) which describes the relative spatial distribution (distance and orientation) of landmark points sampled on the object contour around feature points. Ling and Jacobs [27] use inner distance to extend shape context to capture articulation. As for skeleton based shape descriptors, the reliability of them is ensured by effective skeletonization [12,33] or skeleton pruning [4,35] methods to a large extent. Among them, the shock graph and its variants [32,34,41] are most popular, which are abstracted from skeletons by designed shape grammar. The second one is finding the correspondences between two sets of the shape descriptors by matching algorithms such as Hungarian, thin plate spline (TPS) and dynamic programming (DP). A testing shape is classified into the class of its nearest neighbor ranked by the matching costs. The exemplar-based strategy requires a large number of training data to capture the large intra-class variances of shapes. However, when the size of training set become quite large, it's intractable to search the nearest neighbor due to the high time cost caused by pairwise matching.

Generative models are also used for shape recognition. Sun and Super [42] propose a Bayesian model, which use the normalized contour fragments as the input features for shape classification. Wang et al. [43] model shapes of one class by a skeletal prototype tree learned by skeleton graph matching. Then a Bayesian inference is used to compute the similarity between a testing skeleton and each skeletal prototype tree. Bai et al. [5] propose to integrate contour and skeleton by a Gaussian mixture model, in which contour fragments and skeleton paths are used as the input features. Unlike their method, ours encodes the contour and skeleton features into one shape descriptor according to the association between contour and skeleton. Therefore, we avoid the intractable step to finetune the weight between contour and skeleton models.

Recently, researchers begin to apply the powerful discriminative models to shape classification. Daliri and Torre [15,16] transform the contour into a string based representation according to a certain order of the corresponding contour points found during contour matching. Then they apply SVM to the kernel space built from the pairwise distances between strings to obtain classification results. Edem and Tari [20] transform a skeleton into a similarity vector, in which each element is the similarity between the skeleton and a skeletal prototype of one shape category. Then they apply linear SVM to the similarity vector to determine the category

of the skeleton. Wang et al. [45] utilize LLC strategy to extract the mid-level representation BCF from contour fragments and they also use linear SVM for classification. Such coding based methods are used for 2D and 3D shape retrieval [2,6]. Shen et al. [39] propose a skeleton based mid-level representation named **Bag of Skeleton Paths (BSP)**, and concatenate the BCF and BSP for shape recognition. The weights between BCF and BSP are automatically learned by SVM. This method implicitly combines contour and skeleton according to the weights learned by SVM, while this paper explicitly combines contour and skeleton by using the correspondence between them, which is a more natural combination way.

3. Methodology

In this section, we will introduce our method for shape recognition, including the steps of shape normalization, SSC descriptor and shape classification by BSCP.

3.1. Shape normalization

As the SPM strategy assumes that the parts of shapes falling in the same subregion are similar, it is not rotation invariant. To apply SPM to shape classification, a normalization step is required to align shapes roughly. One straightforward solution is to align each shape with its major axis. Here, we use principal component analysis (PCA) to compute the orientation of the major axis of each shape. Formally, given a shape $F \subset \mathbb{R}^2$, we apply PCA to the point set $\{p_i = (x_i, y_i) | p_i \in F\}_{i=1}^N$. First, the $N \times N$ covariance matrix Σ is computed by $\Sigma = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x}_i)(y_i - \bar{y}_i)$, where $\bar{x}_i = \sum_{i=1}^N x_i/N$ and $\bar{y}_i = \sum_{i=1}^N y_i/N$. Then, the two eigenvectors \mathbf{v}_1 and \mathbf{v}_2 of Σ form the columns of the $N \times N$ matrix V , and the two eigenvalues of Σ are $(\lambda_1, \lambda_2)^T = \mathbf{diag}(V^T \Sigma V)$. The orientation of the major axis of the shape F is the orientation of the eigenvector whose corresponding eigenvalue is bigger. All shapes are rotated to ensure their estimated major axes are aligned with the horizontal line, such as the example given in Fig. 4(b).

3.2. Skeleton-associated shape context

In this section, we show how to compute the SSC descriptor for a given contour point step by step.

3.2.1. Skeleton-associated contour

For a given shape F , let $\mathcal{C}(F)$ and $\mathcal{S}(F)$ denote its contour and skeleton, respectively. The skeleton $\mathcal{S}(F)$ can be obtained by the method introduced in [37], which does not require parameter tuning for skeleton computation. Our goal is to find the corresponding skeleton point of each contour point and assign a object thickness value to it. To describe our method clearly, here we first briefly review some skeleton related definitions. According to the definition of skeleton [11], a skeleton is a set of the centers of the maximal discs of a shape. A maximal disc has at least two points of tangency on the contour, which are called **Generating Points (GPs)**.

Formally, for a skeleton point $p(p \in \mathcal{S}(F))$, let $\mathcal{R}(p)$ be the radius of the maximal disc of the shape F centered at p and $\mathcal{G}(p)$ be the set of GPs of p . On the discrete domain, $\mathcal{R}(p)$ can be approached by the **Distance Transform (DT)** value of p to the contour $\mathcal{C}(F)$:

$$\mathcal{R}(p) = \min_{q \in \mathcal{C}(F)} \|p - q\|_2, \tag{1}$$

where $\|\cdot\|_2$ is the ℓ_2 -Norm. $\mathcal{G}(p)$ can be obtained approximatively by

$$\mathcal{G}(p) = \{q^* | \exists p_n \in \mathcal{N}(p), q^* = \arg \min_{q \in \mathcal{C}(F)} \|q - p_n\|_2\}, \tag{2}$$

where $\mathcal{N}(p)$ denotes the eight neighbors of p . Note that, $\mathcal{G}(p) \subset \mathcal{C}(F)$. Now we have a one-to-many correspondence between a skeleton point p and a set of contour point $\mathcal{G}(p)$. For each contour point $q \in \mathcal{G}(p)$, we associate the object thickness value $\mathcal{R}(p)$ with it, and use the notation $\mathcal{C}(\cdot)$ to denote the corresponding function mapping it to the skeleton point p , i.e., $p = \mathcal{C}(q)$, if $q \in \mathcal{G}(p)$. Now considering the overall shape, let $\mathcal{G}(\mathcal{S}(F))$ be the set of all the GPs of $\mathcal{S}(F)$:

$$\mathcal{G}(\mathcal{S}(F)) = \bigcup_{p \in \mathcal{S}(F)} \mathcal{G}(p). \tag{3}$$

Note that $\mathcal{G}(\mathcal{S}(F)) \subseteq \mathcal{C}(F)$, so the function $\mathcal{C}(\cdot)$ can not be applied to all the contour points. However, we can define a unified function $\mathcal{R}_a(\cdot)$ to compute the associated object thickness value for each contour point q :

$$\mathcal{R}_a(q) = \mathcal{R}(\mathcal{C}(q_a)), \tag{4}$$

where $q_a = \arg \min_{q_g \in \mathcal{G}(\mathcal{S}(F))} l(q, q_g)$ and $l(\cdot, \cdot)$ is denoted by the minimum contour curve length between two contour points. Eq. 4 means that for each contour point q , we search its closest contour point $q_a \in \mathcal{G}(\mathcal{S}(F))$ along the contour (if $q \in \mathcal{G}(\mathcal{S}(F))$, then $q_a = q$), and assign the associated object thickness value of q_a to q .

3.2.2. Shape descriptor computation

Part-based methods [5,21,42,45] have been widely used for shape recognition, as shape parts are the basic meaningful elements of a shape. We want to build a discriminative and informative shape representation based on shape parts. The shape parts can be obtained by any contour decomposition methods, such like **Discrete Contour Evolution (DCE)** [23]. Given a shape contour $\mathcal{C}(F)$, we apply DCE to obtain its critical points $\{u_i\}_{i=1}^T$, where T is the number of the critical points. We build a shape part set $\mathcal{P}_{\mathcal{C}(F)}$, which consists of the contour fragments between any pairs of critical points u_i, u_j . Let c_{ij} denote the contour fragment from u_i to u_j (anticlockwise direction), then we have

$$\mathcal{P}_{\mathcal{C}(F)} = \{c_{ij} | i \neq j, i, j \in \{1, \dots, T\}\}. \tag{5}$$

Note that we do not force u_i and u_j to be adjacent points in the critical point set, and c_{ij} and c_{ji} are two different parts. Also we have $\mathcal{C}(F) = c_{ij} \cup c_{ji}$. Using the method described in the previous section, any contour part can be transformed into a skeleton-associated contour part. In the reminder of this paper, unless otherwise specified, we treat these two concepts equally.

Now we propose how to compute the SSC descriptor at a reference contour point of a skeleton-associated contour part. Each point q on a skeleton-associated contour part can be represented by a triplet $(x, y, \mathcal{R}_a(q))$, where (x, y) is the relative coordinate and $\mathcal{R}_a(q)$ is the associated object thickness value.¹ From this view, the point q actually lies in a 3D space. Given a contour part, we uniformly sample n points on it, then for a given reference contour point r_i , we describe its descriptor by the distribution of relative differences to the n sampled points on Euclidean distance, orientation and associated object thickness value. We compute a coarse histogram h_i for r_i :

$$h_i(j) = \#\{q \neq r_i : (q - r_i) \in \text{bin}(j)\}, j \in \{1, \dots, M\} \tag{6}$$

Here, $(q - r_i) = (\rho_i, \theta_i, \log(\mathcal{R}_a(q)) - \log(\mathcal{R}_a(r_i)))$, where ρ_i and θ_i are the Euclidean distance between q and r_i and the orientation angle of the ray from r_i to q defined on log-polar space, respectively. We use M bins that are uniform in such a 3D space, which follows the strategy used in SC [8] to make the descriptor more sensitive to nearby sample points than those farther away. The histogram h_i is defined to be the SSC of r_i .

Finally, we concatenate the SSC descriptors of the reference points on a contour part c_{ij} to form the descriptor vector $\mathbf{f}_{ij} \in \mathbb{R}^D$ for c_{ij} : $\mathbf{f}_{ij} = (h_i; i = 1, \dots, n)^T$, where n is the number of the reference points and $D = n \times M$.

3.3. Bag of skeleton-associated contour parts

In this section, we introduce how to perform shape classification by BSCP.

3.3.1. Contour parts encoding

Encoding a skeleton-associated contour part $\mathbf{f} \in \mathbb{R}^D$ is transforming it into a new space \mathcal{B} by a given codebook with K entries, $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_K) \in \mathbb{R}^{D \times K}$. In the new space, the contour part \mathbf{f} is represented by a shape code $\mathbf{c} \in \mathbb{R}^K$.

Codebook construction is usually achieved by unsupervised learning, such as k-means. Given a set of contour parts randomly sampled from all the shapes in a dataset as well as their flipped mirrors, we apply k-means algorithm to cluster them into K clusters and construct a codebook $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_K)$. Each cluster center forms an entry of the codebook \mathbf{b}_i .

To encode a contour part \mathbf{f} , we adopt LLC scheme [44], as it has been proved to be effective for image classification. Encoding is usually achieved by minimizing the reconstruction error. LLC additionally incorporates locality constraint, which solves the following constrained least square fitting problem:

$$\min_{\mathbf{c}_{\pi_k}} \|\mathbf{f} - \mathbf{B}_{\pi_k} \mathbf{c}_{\pi_k}\|, \text{ s.t. } \mathbf{1}^T \mathbf{c}_{\pi_k} = 1, \tag{7}$$

where \mathbf{B}_{π_k} is the local bases formed by the k nearest neighbors of \mathbf{f} and $\mathbf{c}_{\pi_k} \in \mathbb{R}^k$ is the reconstruction coefficients. Such a locality constrain leads to several favorable properties such as local smooth sparsity and better reconstruction. The code of \mathbf{f} encoded by the codebook \mathbf{B} , i.e. $\mathbf{c} \in \mathbb{R}^K$, can be easily converted from \mathbf{c}_{π_k} by setting the corresponding entries of \mathbf{c} are equal to \mathbf{c}_{π_k} 's and others are zero.

Note that, the SSC descriptors of a contour part and its flipped mirror are different, as shown in Fig. 5. To make our shape code invariant to the flip transformation, for a contour part, we propose to add the shape code of its flipped mirror to its in an element-wise manner (as shown in Fig. 5). In this way, the shape codes of a contour part and its flipped mirror are the same. The available encoding of contour parts and their flipped mirrors are ensured

¹ To ensure scale invariant, this value should be normalized by dividing by the mean value of the points on the contour part.

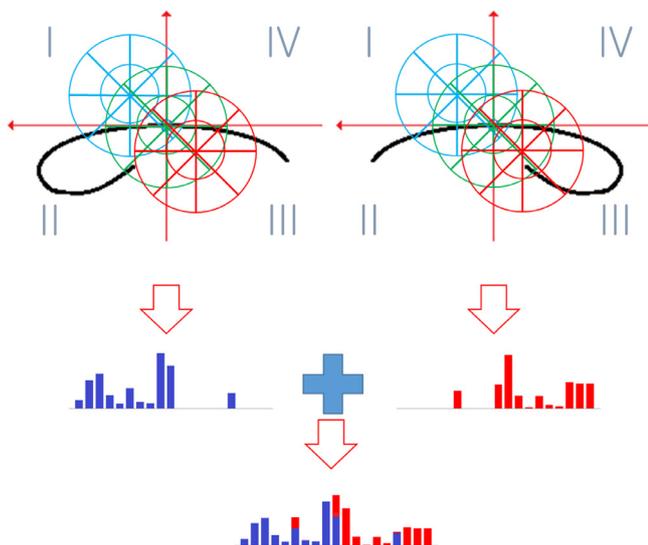


Fig. 5. The shape codes of a contour part and its flipped mirror are added in an element-wise manner to form the final shape code for it, which is invariant to flip transformation.

by the sufficient samples used for codebook building (recall that our codebook is generated by clustering a set of contour parts randomly sampled from all the shapes in a dataset as well as their flipped mirrors).

3.3.2. Shape code pooling

Given a shape F , its skeleton-associated contour parts are encoded into shape codes $\{\mathbf{c}_i\}_{i=1}^n$, where n is the number of the contour parts in F . Now we describe how to obtain a compact shape feature vector by pooling the shape codes. SPM is usually used to incorporate spatial layout information when pooling the image codes. It usually divides an image into $2^l \times 2^l$ ($l = 0, 1, 2$) subregions and then the features in each subregion are pooled respectively. For the aligned shapes belong to one category, the contour parts falls in the same subregions should be similar. Here, the position of a contour part is defined as its median point. More specifically, we divide a shape F into $2^l \times 2^l$ ($l = 0, 1, 2$) subregions, i.e. 21 subregions totally. Let $\mathbf{c}^z \in \mathbb{R}^K$ denote the shape code of a contour part at position z , to obtain a shape feature vector $\mathbf{g}(F)$, for each subregion SR_i , $i \in \{1, 2, \dots, 21\}$, we perform max pooling on it as follow:

$$\mathbf{g}_i(F) = \max(\mathbf{c}^z | z \in SR_i), \quad (8)$$

where the “max” function is performed in an element-wise manner, i.e. for each codeword, we take the max value of all shape codes in a subregion. Max pooling is robust to noise and has been successfully applied to image classification. $\mathbf{g}_i(F)$ is a K dimensional feature vector of the subregion SR_i . The BSCP vector $\mathbf{g}(F)$ is a concatenation of the feature vectors of all subregions:

$$\mathbf{g}(F) = (\mathbf{g}_1^T(F), \mathbf{g}_2^T(F), \dots, \mathbf{g}_{21}^T(F))^T. \quad (9)$$

Finally, $\mathbf{g}(F)$ is normalized by its ℓ_2 -norm: $\mathbf{g}(F) = \mathbf{g}(F) / \|\mathbf{g}(F)\|_2$.

3.4. Shape classification by BSCP

Given a training set $\{(\mathbf{g}_i, y_i)\}_{i=1}^M$ consisting of M shapes from L classes, where \mathbf{g}_i and $y_i \in \{1, 2, \dots, L\}$ are the BSCP vector and the class label of i th shapes respectively, we train a multi-class linear SVM [14] as the classifier:

$$\min_{\mathbf{w}_1, \dots, \mathbf{w}_L} \sum_{j=1}^M \|\mathbf{w}_j\|^2 + \alpha \sum_i \max(0, 1 + \mathbf{w}_i^T \mathbf{g}_i - \mathbf{w}_{y_i}^T \mathbf{g}_i), \quad (10)$$

where $l_i = \arg \max_{l \in \{1, 2, \dots, L\}, l \neq y_i} \mathbf{w}_l^T \mathbf{g}_i$; and α is a parameter to balance the weight between the regularization term (left part) and the multi-class hinge-loss term (right part). For a testing shape vector \mathbf{g} , its class label is given by

$$\hat{y} = \arg \max_{l \in \{1, 2, \dots, L\}} \mathbf{w}_l^T \mathbf{g}. \quad (11)$$

Here we adopt linear SVM, as the proposed BSCP feature vector is a high dimensional sparse vector, computed by LLC coding. The ℓ_2 normalization in LLC makes the inner product of any vector with itself to be one, which is desirable for linear kernels [44]. Using classifiers with nonlinear kernel, such as kernel SVM and random forest, instead leads to performance decrease.

4. Experimental results

In this section, we evaluate our method on several shape benchmarks in comparison to the state-of-the-arts. We also investigate the effects of two important parameters introduced in our method on classification accuracy: the number of object thickness difference bins for computing SSC N_{td} and codebook size K .

4.1. Experimental setup

For each contour part, we form a descriptor vector for it by concatenating the SSC descriptors computed on 5 reference points. Unless otherwise specified, we set the number of bins for computing SSC to 300 (5 Euclidean distance bins, 12 orientation bins, 5 object thickness difference bins). Thus the dimension of a descriptor vector for a contour part is 1500. The number of Euclidean distance bins and the number of orientation bins are set to the default values used in SC [8]. Hence, we will discuss the effects of the number of object thickness difference bins on classification accuracy individually. When learning the codebook, the number of cluster centers (codebook size) is set to 2500 by default. We also study the performances of BSCP by varying the codebook size. To encode a contour part, we adopt the approximated LLC with 5 nearest neighbors. When pooling, a shape is divided into 1×1 , 2×2 and 4×4 , in total 21 regions. The weight between the regularization term and the multi-class hinge-loss term in the multi-class linear SVM formulation is set to 10. Default parameter settings reported in [37] are adopted to extract skeletons.

All the experiments were carried out on a workstation (3.1 GHz 32-core CPU, 128G RAM and Ubuntu14.04 64-bit OS). It takes about 25 ms to compute our SSC descriptor for one contour fragment, and 1.1 s to encode the BSCP feature vector for one shape. The whole training process takes about 8 h (including feature computation and codebook learning), the testing process for one shape takes 17.5 ms (excluding feature computation).

We evaluate our method on several shape classification benchmark datasets, including the MPEG-7 dataset [24], the Animal dataset [5], and the ETH-80 dataset [26]. To avoid the biases caused by randomness, such a procedure is repeated 10 times. Average classification accuracy and standard derivation are reported to evaluate the performance of different shape classification methods. In each round, we randomly select half of shapes in each class to train and use the rest shapes to evaluate for every dataset except the ETH-80 dataset. On the ETH-80 dataset, following the previous methods [15,16,26,27,45], we use all shapes except the current one for training and use the current one for testing (Leave-one-out setting [18]). Experimental results and analysis are given in the rest of this section.

4.2. Animal dataset

We firstly test our method on the Animal dataset which is introduced in [5]. This dataset contains 2000 shapes divided into

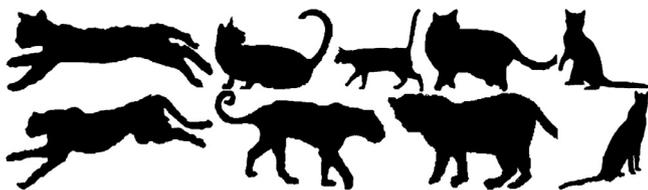


Fig. 6. Shapes of two classes from Animal dataset [5]. The first row shows 5 shapes of the Cat class, with large intra-class variations caused by view point change and various gestures of the cats. Moreover, leopards on the second row are similar to those cats on the first row, which makes recognition of these two kinds of shapes much more difficult.

Table 1
Classification accuracy comparison on Animal dataset [5].

Algorithm	Classification accuracy (%)
Skeleton paths [5]	67.90
Contour segments [5]	71.70
IDSC [27]	73.60
ICS [5]	78.40
BCF [45]	83.40 ± 1.30
Bioinformatic [10]	83.70
ShapeVocabulary [6]	84.30 ± 1.01
BCF+BSP [39]	85.50 ± 0.88
Contextual BOW [9]	86.00
BSCP	89.04 ± 0.95

20 kinds of animals, including cat, spider, leopard, etc. It is the most challenging shape dataset due to the large intra-class variations caused by view point change and various gestures of animals (as shown in Fig. 6). We randomly choose 50 shapes per class for training and leave the rest 50 shapes for testing. The comparison between BSCP and other shape classification methods is demonstrated in Table 1.

As shown in Table 1, the proposed method achieves a classification accuracy at 89.04% which significantly outperforms the previous state-of-the-art method, Contextual BOW [9], by over 3.0%. This result proves that the introduction of the object thickness information extracted from skeletons indeed help shape recognition. Our method also performs much better than BCF+BSP [39], evidencing that our method which associates a shape contour with skeletal information in such a principal way is more effective than the previous method, which combines contour and skeleton implicitly according to the weights learned by SVM. The comparison between our method and BCF [45], directly shows that SSC descriptor can capture not only the geometric information of the object contour but also the object thickness information for a shape. The combination of such two kinds of complementary information leads to an improvement on resisting interference caused by intra-class variations.

4.3. MPEG-7 dataset

Then we evaluate our method on the MPEG-7 dataset [24], which is the most well-known dataset for shape analysis in the field of computer vision (see Fig. 7). 1400 images of the dataset



Fig. 7. Typical shapes of some classes from MPEG-7 dataset [24].

Table 2
Classification accuracy comparison on MPEG-7 dataset [24].

Algorithm	Classification accuracy (%)
Skeleton Paths [5]	86.70
Contour Segments [5]	90.90
Bioinformatic [10]	96.10
ICS [5]	96.60
BCF [45]	97.16 ± 0.79
BCF+BSP [39]	98.35 ± 0.63
BSCP	98.41 ± 0.44

are divided into 70 classes with high shape variability, in each of which there are 20 different shapes. Average classification accuracy and standard derivation of classification accuracies are reported in Table 2.

As shown in Table 2, our method achieves the best performance on the MPEG-7 dataset. BCF [45] has already obtained good result, since it applies the Bag of Features framework to obtain the mid-level model of shape representation, which is more robust and accurate. BCF+BSP [39] combines skeleton and contour information in a simple but effective way, and performs better than BCF, which proves that both skeleton and contour features are important in shape classification. However, with adopting SSC descriptor to combine contour and skeleton information, our method achieves better result than BCF+BSP on this dataset. The improvement on this dataset is not so significant as the one on the Animal dataset, the reason is the accuracies of the state-of-the-arts on this dataset have already approached to 100%.

4.4. ETH-80 dataset

The ETH-80 dataset [26] contains 80 objects, which are divided into 8 categories. There are 41 3D color photographs taken from different viewpoints for each object. We use the segmentation masks provided by the dataset to evaluate our method. The result is shown in Table 3.

Compared with other methods, ours achieves the classification accuracy of 93.05%, outperforming the previous state-of-the-art approach in [45] by over 1.5%.

4.5. Parameter discussion

In this section, we investigate the effects of three important parameters on shape classification accuracy.

The number of object thickness difference bins for computing SSC. Since the proposal of the shape descriptor SSC is an important contribution, it is necessary to study how different settings of the descriptor effect the performance on shape classification.

As an extension of the Shape Context, SSC has one more dimension to describe the thickness differences, the number of

Table 3
Classification accuracy comparison on ETH-80 dataset [26].

Algorithm	Classification accuracy
Color histogram [26]	64.86
PCA gray [26]	82.99
PCA masks [26]	83.41
SC+DP [26]	86.40
IDSC+DP [27]	88.11
Robust symbolic [15]	90.28
Kernel-edit [16]	91.33
BCF [45]	91.49
Bioinformatic [10]	91.50
BSCP	93.05

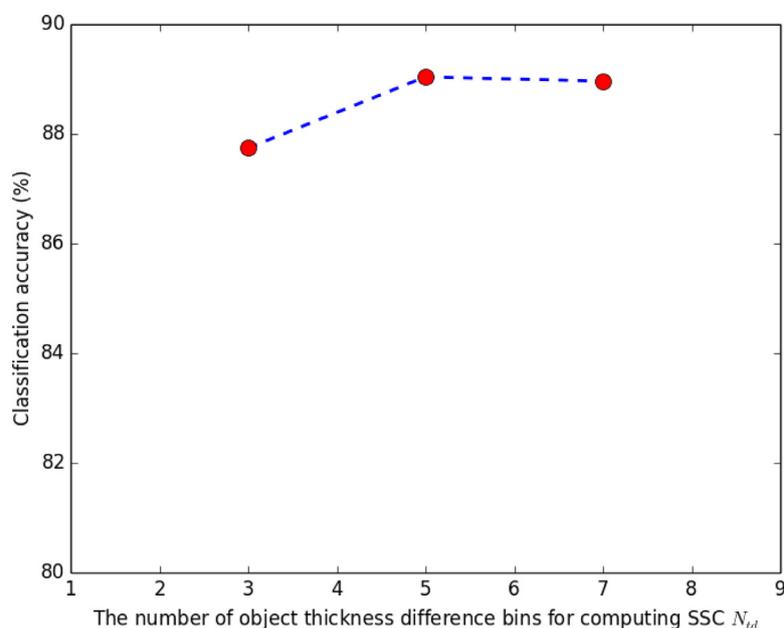


Fig. 8. Classification accuracies on Animal dataset [5] by varying the number of object thickness difference bins for computing SSC N_{td} .

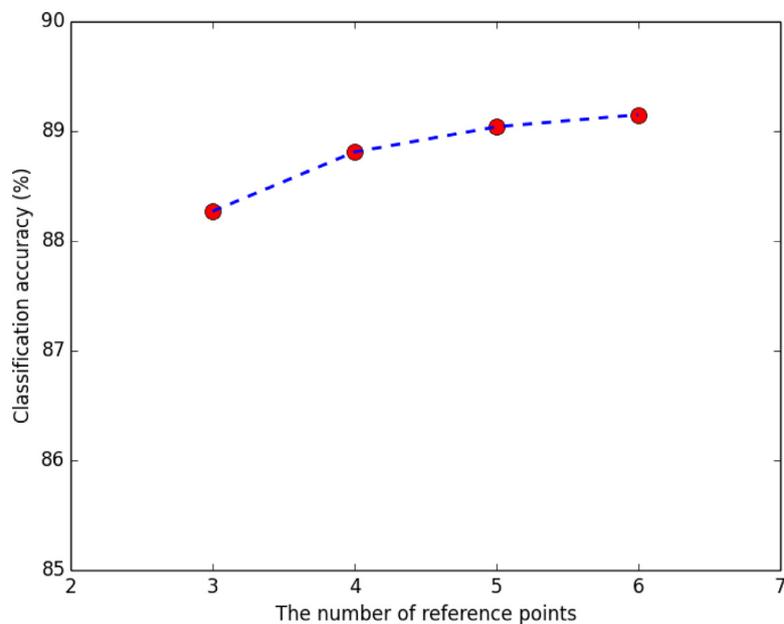


Fig. 9. Classification accuracies on Animal dataset [5] by varying the number of reference points for computing SSC.

object thickness difference bins N_{td} . To investigate the influence of this parameter, we set N_{td} to different values to observe the performance change on the Animal dataset, while other parameters are set to the default values. The result is reported in Fig. 8.

Observed that our method achieves the best performance when N_{td} is set to 5. $N_{td} = 3$ (or $N_{td} = 1$) leads to performance decrease. The reason may be that SSC with small N_{td} can only give a coarse representation of the thickness information, while losing most of the information a skeleton provides. Although $N_{td} = 7$ leads to a result close to the best one, it will result in significant increase in SSC descriptor computation, codebook learning and feature encoding. $N_{td} = 5$, which is selected by us, is thought to be the best trade-off between accuracy and efficiency. We use it as the default value in our experiments, and gain the state-of-the-art performances on several datasets (see Tables 1–3).

The number of reference points for computing SSC. We also show how performance changes by varying the number of reference points when computing our SSC descriptor in Fig. 9. Unsurprisingly, with the increase of the number of reference points, the classification accuracy is improved, as more shape details are considered. However, using more reference points leads to a significantly time consuming shape feature computation process. To balance the performance and computational cost, we choose 5 reference points.

Codebook size. In this experiment, we adopt codebooks with different sizes, including 500, 1000, 1500, 2000, 2500 and 3000, to classify shapes on the Animal dataset. Other parameters are fixed to their default values. The classification accuracies of BSCP by using different codebook sizes are shown in the Fig. 10. As the codebook size increases, shape classification accuracy improves generally, which was also reported in [45].

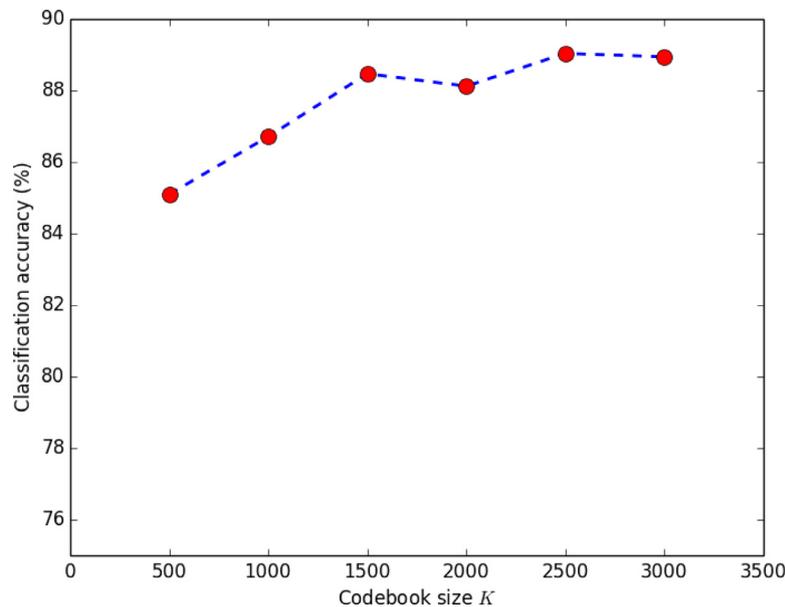


Fig. 10. Classification accuracies on Animal dataset [5] by varying codebook size K .

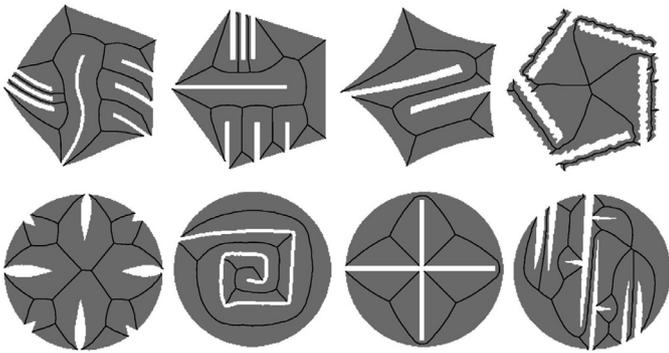


Fig. 11. Each row represents four shape examples from one kind of “device” class in the MPEG-7 dataset. The skeleton of each shape is visualized by black curves. The envelope contour of the shapes in each row are similar, while their skeletons are totally different.

4.6. Limitation

Our SSC descriptor relies on the quality of the extracted skeleton. It also requires that the object can be well represented by its skeleton. Some objects in the MPEG-7 dataset, such as the “device” classes shown in Fig. 11, are not suitable to be represented by skeletons. In this case, our SSC descriptor does not perform well. We have applied our SSC descriptor to the shape retrieval framework of “Shape Vocabulary” [6] and test it on the MPEG-7 dataset. Unfortunately, we do not see the performance increase. This may be another reason why our method does not achieve an obvious classification improvement on the MPEG-7 dataset, as shown in Table 2.

5. Conclusion

In this paper, we present a novel shape representation called BSCP, which combines contour and skeleton in a principal way. This is achieved through the adoption of a novel low-level shape descriptor, the SSC, which is able to make full use of the natural correspondence between a contour and its skeleton. Both the normalization step and SPM are adopted to ensure that our method is effective and accurate, without losing the invariance to rotation.

We have tested BSCP in many benchmarks, and the results lead to a conclusion that our method has achieved the state-of-the-art performance. Parameter discussion is also done as a reference for other researchers. In the future, we will further study how to apply BSCP to recognize objects in natural images, which requires reliable object contour detection [38] and symmetry detection [36].

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61303095, in part by Research Fund for the Doctoral Program of Higher Education of China under Grant 20133108120017, in part by Innovation Program of Shanghai Municipal Education Commission under Grant 14YZ018, in part by Innovation Program of Shanghai University under Grant SDCX2013012 and in part by Cultivation Fund for the Young Faculty of Higher Education of Shanghai under Grant ZZSD13005.

References

- [1] C. Aslan, A. Erdem, E. Erdem, S. Tari, Disconnected skeleton: shape at its absolute scale, *IEEE Trans. Patt. Anal. Mach. Intell.* 30 (12) (2008) 2188–2203.
- [2] X. Bai, S. Bai, Z. Zhu, L.J. Latecki, 3d shape matching via two layer coding, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (12) (2015) 2361–2373.
- [3] X. Bai, L. Latecki, Path similarity skeleton graph matching, *IEEE Trans. Patt. Anal. Mach. Intell.* 30 (7) (2008) 1282–1292.
- [4] X. Bai, L.J. Latecki, W. Liu, Skeleton pruning by contour partitioning with discrete curve evolution, *IEEE Trans. Patt. Anal. Mach. Intell.* 29 (3) (2007) 449–462.
- [5] X. Bai, W. Liu, Z. Tu, Integrating contour and skeleton for shape classification, in: *ICCV Workshops, 2009*, pp. 360–367.
- [6] X. Bai, C. Rao, X. Wang, Shape vocabulary: a robust and efficient shape representation for shape matching, *IEEE Trans. Image Process.* 23 (9) (2014) 3935–3949.
- [7] E. Baseski, A. Erdem, S. Tari, Dissimilarity between two skeletal trees in a context, *Patt. Recog.* 42 (3) (2009) 370–385.
- [8] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Trans. Patt. Anal. Mach. Intell.* 24 (4) (2002) 509–522.
- [9] R. Bharath, C. Xiang, T.H. Lee, Shape classification using invariant features and contextual information in the bag-of-words model, *Patt. Recog.* 48 (3) (2015) 894–906.
- [10] M. Bicego, P. Lovato, A bioinformatics approach to 2d shape classification, *Comput. Vision Image Understanding* (2015).
- [11] H. Blum, Biological shape and visual science, *J. Theor. Biol.* 38 (1973) 205–287.
- [12] G. Borgefors, I. Nyström, G.S. di Baja, Computing skeletons in three dimensions, *Patt. Recog.* 32 (7) (1999) 1225–1236.
- [13] T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to Algorithms*, second ed., MIT Press, 2001.

- [14] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, *J. Mach. Learn. Res.* 2 (2001) 265–292.
- [15] M.R. Daliri, V. Torre, Robust symbolic representation for shape recognition and retrieval, *Patt. Recog.* 41 (5) (2008) 1782–1798.
- [16] M.R. Daliri, V. Torre, Shape recognition based on kernel-edit distance, *Comput. Vis. Image Understanding* 114 (10) (2010) 1097–1103.
- [17] M. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, S. Dickinson, Object recognition as many-to-many feature matching, *Int. J. Comput. Vis.* 69 (2) (2006) 203–222.
- [18] P.A. Devijver, J. Kittler, *Pattern Recognition: A Statistical Approach*, London, GB: Prentice-Hall, 1982.
- [19] J. Duchon, *Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces*, Berlin: Springer-Verlag, 1977.
- [20] A. Erdem, S. Tari, A similarity-based approach for shape classification using aslan skeletons, *Patt. Recog. Lett.* 31 (13) (2010) 2024–2032.
- [21] P.F. Felzenszwalb, J. Schwartz, Hierarchical matching of deformable shapes, in: *CVPR*, 2007.
- [22] C. Grigorescu, N. Petkov, Distance sets for shape filters and shape recognition, *IEEE Trans. Image Process.* 12 (10) (2003) 1274–1286.
- [23] L.J. Latecki, R. Lakämper, Convexity rule for shape decomposition based on discrete contour evolution, *Comput. Vis. Image Understanding* 73 (3) (1999) 441–454.
- [24] L.J. Latecki, R. Lakämper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: *CVPR*, 2000, pp. 1424–1429.
- [25] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: *CVPR*, 2006, pp. 2169–2178.
- [26] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, 16–22 June 2003, Madison, WI, USA, 2003, pp. 409–415.
- [27] H. Ling, D.W. Jacobs, Shape classification using the inner-distance, *IEEE Trans. Patt. Anal. Mach. Intell.* 29 (2) (2007) 286–299.
- [28] J. Ma, W. Qiu, J. Zhao, Y. Ma, A.L. Yuille, Z. Tu, Robust l_2 estimation of transformation for non-rigid registration, *IEEE Trans. Signal Process.* 63 (5) (2015) 1115–1129.
- [29] J. Ma, J. Zhao, J. Tian, A.L. Yuille, Z. Tu, Robust point matching via vector field consensus, *IEEE Trans. Image Process.* 23 (4) (2014) 1706–1721.
- [30] J. Ma, J. Zhao, A.L. Yuille, Non-rigid point set registration by preserving global and local structures, *IEEE Trans. Image Process.* 25 (1) (2016) 53–64.
- [31] J. Ma, H. Zhou, J. Zhao, J. Tian, Robust feature matching for remote sensing image registration via locally linear transforming, *IEEE Trans. Geosci. Remote Sensing* 53 (12) (2015) 6469–6481.
- [32] D. Macrini, S.J. Dickinson, D.J. Fleet, K. Siddiqi, Object categorization using bone graphs, *Comput. Vis. Image Understanding* 115 (8) (2011) 1187–1206.
- [33] P.K. Saha, G. Borgefors, G.S. di Baja, A survey on skeletonization algorithms and their applications, *Patt. Recog. Lett.* (2015).
- [34] T. Sebastian, P. Klein, B. Kimia, Recognition of shapes by editing their shock graphs, *IEEE Trans. Patt. Anal. Mach. Intell.* 26 (5) (2004) 550–571.
- [35] W. Shen, X. Bai, R. Hu, H. Wang, L.J. Latecki, Skeleton growing and pruning with bending potential ratio, *Patt. Recog.* 44 (2) (2011) 196–209.
- [36] W. Shen, X. Bai, Z. Hu, Z. Zhang, Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images, *Patt. Recog.* 52 (4) (2016) 306–316.
- [37] W. Shen, X. Bai, X. Yang, L.J. Latecki, Skeleton pruning as trade-off between skeleton simplicity and reconstruction error, *SCIENCE CHINA Informat. Sci.* 56 (4) (2013) 1–14.
- [38] W. Shen, X. Wang, Y. Wang, X. Bai, Z. Zhang, Deepcontour: a deep convolutional feature learned by positive-sharing loss for contour detection, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, Boston, MA, USA, June 7–12, 2015, 2015, pp. 3982–3991.
- [39] W. Shen, X. Wang, C. Yao, X. Bai, Shape recognition by combining contour and skeleton into a mid-level representation, in: *Proceedings of Pattern Recognition - 6th Chinese Conference, CCPR 2014*, Changsha, China, November 17–19, 2014., 2014, pp. 391–400.
- [40] W. Shen, Y. Wang, X. Bai, H. Wang, L.J. Latecki, Shape clustering: Common structure discovery, *Patt. Recog.* 46 (2) (2013) 539–550.
- [41] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S. Zucker, Shock graphs and shape matching, *Int'l J. Comput. Vis.* 35 (1) (1999) 13–32.
- [42] K.B. Sun, B.J. Super, Classification of contour shapes using class segment sets, in: *CVPR*, 2005, pp. 727–733.
- [43] B. Wang, W. Shen, W. Liu, X. You, X. Bai, Shape classification using tree - unions, in: *ICPR*, 2010, pp. 983–986.
- [44] J. Wang, J. Yang, K. Yu, F. Lv, T.S. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: *CVPR*, 2010, pp. 3360–3367.
- [45] X. Wang, B. Feng, X. Bai, W. Liu, L.J. Latecki, Bag of contour fragments for robust shape classification, *Patt. Recog.* 47 (6) (2014) 2116–2125.
- [46] J. Xie, P. Heng, M. Shah, Shape matching and modeling using skeletal context, *Patt. Recog.* 41 (5) (2008) 1756–1767.