SHAPE-BASED OBJECT RETRIEVAL BY CONTOUR SEGMENT MATCHING

Cong Yang⋆ *Oliver Tiebe*⋆ *Pit Pietsch*† *Christian Feinen*⋆ *Udo Kelter*† *Marcin Grzegorzek*⋆

⋆Research Group for Pattern Recognition, University of Siegen
†Software Engineering and Databases Group, University of Siegen
Corresponding Email: marcin.grzegorzek@uni-siegen.de

ABSTRACT

In this paper we introduce an approach for object retrieval that uses contour segment matching for shape similarity computation. The object contour is partitioned into segments by skeleton endpoints. Each contour segment is represented by a rotation and scale invariant, 12-dimensional feature vector. The similarity of two objects is determined by matching their contour segments using the Hungarian algorithm. Our method is insensitive to object deformation and outperforms existing shape-based object retrieval algorithms. The most significant scientific contributions of this paper include (i) the introduction of a new feature extraction technique for contour segments as well as (ii) a new similarity measure for contour segments cleverly modelling the human perception and easily adapting to concrete application domains, and (iii) the impressive robustness of the method in an object retrieval scenario.

Index Terms— Object Retrieval, Shape Similarity, Contour Matching.

1. INTRODUCTION

Shape is a very important object property being perceptually unique due to the fact that it is both complex and structured. Shapes are perceived veridically and are the only perceptual attributes of objects that allow unambiguous classification [1]. Estimating similarities of object shapes belongs to the most common unconscious human activity. Humans process shapes using a huge knowledge database of prior experiences and taking into account the surrounding environment. For instance, a horse and a cat become, for humans, less similar to each other, if a dog suddenly appears in the scene and chases away a stork. Moreover, humans unconsciously consider both the outer contour and the topology of an object for categorisation. However, it is really difficult to imitate the amazing human shape interpretation and abstraction capabilities with computer-based algorithms.

The contribution of this article reacts to this problem. We introduce a contour segment descriptor together with a corresponding similarity measure that does not only consider two contour segments isolated from the environment. The similarity to neighbouring object contour segments is also taken into account. Further, the similarity function is able to integrate background knowledge about a certain context into the calculation process. For this, all dimensions of the feature space describing contour segments are weighted by coefficients that are automatically learnt based on a small subset of images from a certain application domain.

The rest of the paper is structured as follows: related work (Section 2), shape representation (Section 3), object matching (Section 4), experiments and results (Section 5), and conclusion (Section 6).

The research activity leading to this article has been supported by the German Research Foundation within the Research Training Group 1564 “Imaging New Modalities” and the China Scholarship Council.

2. RELATED WORK

Shape Modelling by Contours and Skeletons: Bai et al. [2] combine the object contour and skeleton properties for shape classification. They extract contour segments using the Discrete Curve Evolution [3]. However, their approach works in a supervised pattern recognition mode and multiple training examples of an object class are necessary for modelling. Zeng et al. [4] combine properties of skeletons and boundaries for general shape decomposition. Unfortunately, this method is rotation variant and highly sensitive to shape deformation. In [5], Bai et al. introduce a shape-based algorithm for detecting and recognising non-rigid objects from natural images. The skeleton is used to capture the main structure of an object. Each branch on the skeleton models the object boundary information, however, the real combination of skeleton and contour properties has not been foreseen in this algorithm.

Contour-Based Shape Modelling: Nguyen et al. [6] propose a shape-based local binary descriptor for object detection that has been tested in the task of detecting humans from static images. In [7], an algorithm for partial shape matching with mildly non-rigid deformations using Markov chains and the Monte Carlo method is introduced. Shotton et al. [8] present a categorical object detection scheme that uses only local contour-based features and is realised in a partly supervised learning framework. In [9], an approach for contour-based object detection using a contour model for a class of objects is described. The model is hierarchically decomposed into fragments and a global similarity measure is applied for object detection. Yang et al. [10] formulate the contour-based object detection as a matching problem between model contour parts and image edge fragments. They treat this problem as the task of finding dominant sets in weighted graphs. Though insensitive to noise and outliers, the approach is not rotation invariant.

Skeleton-Based Shape Modelling: Compared to contour matching methods, skeleton matching approaches feature lower sensitivity to occlusion, limb growth, and articulation [11]. However, they are computationally more complex [12] and still have not yet been fully successfully applied to real images. Baseski et al. [13] present a tree-edit-based shape matching method that uses a recent coarse skeleton representation. Their dissimilarity measure gives a better understanding within group versus between group separation which mimics the asymmetric nature of human similarity judgements. To the best of our knowledge, the best performing skeleton-based object matching algorithm has been proposed by Bai et al. [14]. Their main idea is to match skeleton graphs by comparing the geodesic paths between skeleton endpoints. Unfortunately, the performance of this method is limited to the presence of large protrusions, since they require skipping a large number of skeleton endpoints.

3. OBJECT REPRESENTATION

To represent the shape and the topology of an object, its outer contour and its skeleton are determined first, whereas, for skeletonisation, contour partitioning with Discrete Curve Evolution (DCE) [3] is employed. The representation model based on skeletons follows [3], the object contour representation is described below.

First, the object contour is divided into N Contour Segments (CS) by the skeleton endpoints. For each CS a 12-dimensional meaningful feature vector $\boldsymbol{c}'_n$ is extracted, whereas its first element is equal to the number of contour segments resulting from the whole object ($c'_{n,1} = N$). Euclidean distance of contour segment endpoints and the total number of pixels in a CS determine $c'_{n,2}$ and $c'_{n,3}$, respectively. These two features are able to express how much a contour segment differs from a straight line. In order to distinguish contour segments of the type presented in Figure 1a from those of the type depicted in Figure 1b, the area between the straight line connecting the CS endpoints and the contour segment itself (marked as grey in Figure 1) is used as the fourth feature ($c'_{n,4}$).

Before computing remaining features, each CS is transformed into a normalised vertical orientation (i.e., so that its endpoints are vertically aligned) to ensure rotation invariance of the object contour representation (see Figure 2). From the two possible results of such a normalising transform, the CS with the majority of points lying on the right

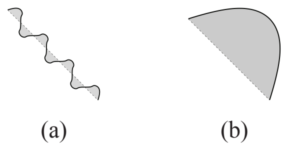

Fig. 1: Since $c'_{n,2}$ and $c'_{n,3}$ are equal for contour segments (a) and (b), a fourth feature $c'_{n,4}$ corresponding to the area depicted in grey is introduced.

side of the straight line connecting its endpoints is selected for further processing. For computing further features $c'_{n,5}, c'_{n,6}, \dots, c'_{n,12}$, we use the bounding box of the whole CS as well as the three equally high sub-boxes shown in Figure 2:

$$\begin{array}{ll} c'_{n,5} = \frac{h_n}{w_n} & c'_{n,6} = \frac{h_{n,1}}{w_{n,1}} \\ c'_{n,7} = \frac{h_{n,2}}{w_{n,2}} & c'_{n,8} = \frac{h_{n,3}}{w_{n,3}} \\ c'_{n,9} = \frac{w_{n,3}h_{n,3}}{w_{n,1}h_{n,1}} & c'_{n,10} = \frac{w_{n,2}h_{n,2}}{w_{n,1}h_{n,1}} \\ c'_{n,11} = \alpha_n & c'_{n,12} = \beta_n \end{array} \quad . \tag{1}$$

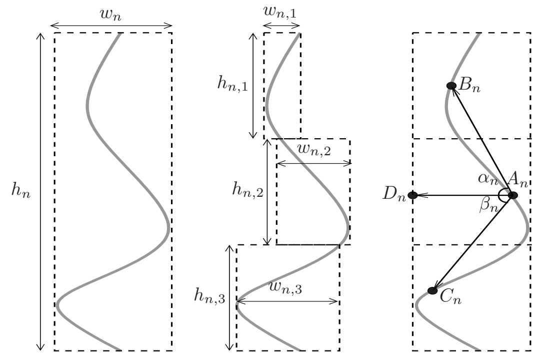

Fig. 2: CS bounding box and equally high sub-boxes ($h_{n,1} = h_{n,2} = h_{n,3}$) used for feature extraction: $A_n \to D_n$ is oriented vertically; B_n, A_n, and C_n are centre pixels of the top, middle, and bottom contour sub-segments, respectively.

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

$$\boldsymbol{c}_n = \frac{\boldsymbol{c}'_n}{w_n + h_n} = (c_{n,1}, c_{n,2}, \dots, c_{n,12})^{\mathrm{T}} \quad . \tag{2}$$

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

$$C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_n \dots, \boldsymbol{c}_N\} \quad . \tag{3}$$

For simplification, in the following we do not differentiate between object contours and their representations C as well as between contour segments and the feature vectors describing them $\boldsymbol{c}_n$.

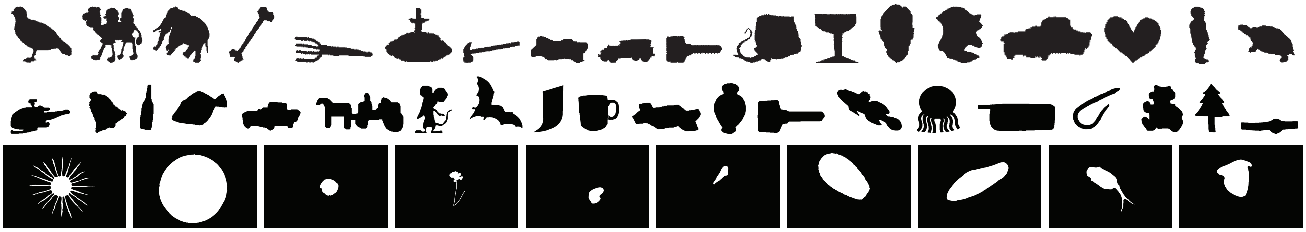

Fig. 3: Example shapes from the experimental datasets: Kimia-216 (1st row), MPEG-400 (2nd row), and EM-200 (3rd row).

4. OBJECT MATCHING

The matching of two objects including the computation of their similarity is performed separately for their skeleton and contour representations. While the skeleton-based matching follows [14], the matching and similarity computation based on the contour segments is described below.

Similarity of Object Contours: First, we arrange the contour segments of both objects in a clockwise way so that the objects can be represented by ordered lists of feature vectors:

$$\begin{array}{l} \boldsymbol{C}^{\star} = (\boldsymbol{c}_1^{\star}, \boldsymbol{c}_2^{\star}, \ldots, \boldsymbol{c}_n^{\star}, \ldots, \boldsymbol{c}_N^{\star}) \\ \boldsymbol{C}^{\diamond} = (\boldsymbol{c}_1^{\diamond}, \boldsymbol{c}_2^{\diamond}, \ldots, \boldsymbol{c}_k^{\diamond}, \ldots, \boldsymbol{c}_K^{\diamond}) \end{array} \quad . \tag{4}$$

To simplify further explanations, we assume that $N \leq K$. Now, we introduce a dissimilarity measure for contour segments belonging to different objects $\boldsymbol{C}^{\star}$ and $\boldsymbol{C}^{\diamond}$:

$$d(\boldsymbol{c}_n^{\star}, \boldsymbol{c}_k^{\diamond}) = \frac{1}{M} \sum_{m=1}^{M} \frac{\sigma_m |c_{n,m}^{\star} - c_{k,m}^{\diamond}|}{\sum\limits_{j=1}^{K} |c_{n,m}^{\star} - c_{j,m}^{\diamond}|} \quad , \tag{5}$$

where $M = 12$ is the dimensionality of the feature space and σ_m is the weight for each feature achieved in an optimisation process explained at the end of this section. As one can see, the dissimilarity value between $\boldsymbol{c}_n^{\star}$ and $\boldsymbol{c}_k^{\diamond}$ does not only depend on these two contour segments. All CS of $\boldsymbol{C}^{\diamond}$ are taken into consideration, whereby (5) does not fulfil the symmetry property $d(\boldsymbol{c}_n^{\star}, \boldsymbol{c}_k^{\diamond}) \neq d(\boldsymbol{c}_k^{\diamond}, \boldsymbol{c}_n^{\star})$. However, it behaves equally to human perception. If the dissimilarity of $\boldsymbol{c}_n^{\star}$ to the neighbours of $\boldsymbol{c}_k^{\diamond}$ in $\boldsymbol{C}^{\diamond}$ decreases, $d(\boldsymbol{c}_n^{\star}, \boldsymbol{c}_k^{\diamond})$ increases. The values of the dissimilarity function (5) belong to the range $d(\boldsymbol{c}_n^{\star}, \boldsymbol{c}_k^{\diamond}) \in [0, 1]$ which enables their easy conversion to similarity values:

$$s(\boldsymbol{c}_n^{\star}, \boldsymbol{c}_k^{\diamond}) = 1 - d(\boldsymbol{c}_n^{\star}, \boldsymbol{c}_k^{\diamond}) \quad . \tag{6}$$

Using (6) we generate a matrix of similarities between all CS in $\boldsymbol{C}^{\star}$ and in $\boldsymbol{C}^{\diamond}$:

$$\boldsymbol{S}(\boldsymbol{C}^{\star}, \boldsymbol{C}^{\diamond}) = \begin{pmatrix} s(\boldsymbol{c}_1^{\star}, \boldsymbol{c}_1^{\diamond}) & \cdots & s(\boldsymbol{c}_1^{\star}, \boldsymbol{c}_K^{\diamond}) \\ \vdots & \vdots & \vdots \\ s(\boldsymbol{c}_N^{\star}, \boldsymbol{c}_1^{\diamond}) & \cdots & s(\boldsymbol{c}_N^{\star}, \boldsymbol{c}_K^{\diamond}) \end{pmatrix} . \tag{7}$$

In order to find an optimum match of contour segments from $\boldsymbol{C}^{\star}$ to CS from $\boldsymbol{C}^{\diamond}$, we finally apply the Hungarian algorithm [15] for the matrix expressed in (7). The resulting similarity values of the matched contour segments can be denoted as $s_1, s_2, \ldots, s_N$ and the global similarity between the object contours $\boldsymbol{C}^{\star}$ and $\boldsymbol{C}^{\diamond}$ is calculated as follows:

$$s_{\text{contours}}(\boldsymbol{C}^{\star}, \boldsymbol{C}^{\diamond}) = \frac{1}{N} \sum_{n=1}^{N} s_n \quad . \tag{8}$$

Fusion of Skeleton and Contour Similarities: The similarities for two objects determined separately for their skeletons $s_{\text{skeletons}}$ and contours s_{contours} are simply averaged to get the combined object similarity value:

$$s_{\text{objects}} = \frac{s_{\text{contours}} + s_{\text{skeletons}}}{2} \quad . \tag{9}$$

As mentioned above, $s_{\text{skeletons}}$ is calculated according to [14].

Context Adaptation: As one can see in (5), the dissimilarity value for two CS depends on the weights $\sigma_1, \ldots, \sigma_{M=12}$. The weight of each feature expresses its importance for the overall similarity of two CS. Setting the weights for a particular dataset gives us the opportunity to adapt our algorithm to the application domain (context). In order to automatically estimate these weights, we apply the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [16]. We start the algorithm with a configuration of equally distributed weights for all features and find the optimum values for a certain dataset in an iterative process. In the practical realisation only a subset of each dataset is used for this optimisation, of course.

5. EXPERIMENTS AND RESULTS

Datasets: To evaluate our methodology, we have performed experiments in an object retrieval scenario using three different datasets: (i) Kimia-216 [18] consisting of 216 objects categories in 18 classes (first row in Figure 3); (ii) MPEG-400, a subset of the MPEG-7 shape collection [19], consisting of 400 objects categorised in 20 classes (second row in Figure 3); and (iii) EM-200 [20] containing 200 objects categorised in 10 classes (third row in Figure 3).

Retrieval Results for Kimia-216	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
PSSGM [14]	216	216	215	216	213	210	210	207	205	191	177
Revised PSSGM [17]	205	208	202	199	200	192	184	167	161	130	96
Our Method (Contours only)	216	215	206	204	200	186	172	163	130	124	107
Our Method (Contours and Skeletons)	216	216	214	213	213	211	204	193	184	175	149

Retrieval Results for MPEG-400	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
PSSGM [14]	380	371	361	351	344	339	332	320	330	309	305
Our Method (Contours only)	375	348	333	325	317	311	300	295	276	275	259
Our Method (Contours and Skeletons)	383	373	364	356	349	343	336	320	330	312	309

Retrieval Results for EM-200	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
Our Method (Contours only)	196	190	190	187	184	182	184	179	173	173	164

Table 1: Experimental comparison of our methodology to the most powerful related algorithm using the Kimia-216 and the MPEG-400 datasets as well as the proof of applicability of our approach to real world problems using the EM-200 dataset. Results are summarised as the number of shapes from the same class among the first top 1-11 shapes.

Application Independent Experiments: For Kimia-216 and MPEG-400, we have run our algorithm in two configuration modes and compared it to the Path Similarity Skeleton Graph Matching (PSSGM) [14] which has been, to the best of our knowledge, outperforming all existing techniques for shape retrieval. In the first configuration, only contours have been used for object description, whereby an isolated evaluation of the new feature space introduced in Section 3) together with the similarity measure proposed in Section 4 has been possible. In the second mode, properties of both, contours and skeletons, have been extracted for object representation. In both cases, skeletons computed according to [14] have been used for partitioning the shapes into multiple contour segments. For each of the shapes used as a query, we have checked whether the retrieved results are correct, i.e., belong to the same class as the query. In order to enable quantitative comparison, we have kept the experimental convention proposed in [14] and considered the 11 best matches for each query. Results achieved for the Kimia-216 and the MPEG-400 datasets can be found in Table 1, whereas in [17] the PSSGM algorithm has been re-evaluated without any preliminary assumptions regarding the object skeletonisation.

The results of the application independent experiments allow us to draw three main conclusions. First, our contour segment descriptor together with the object contour similarity measure is very robust, since it leads to very good shape retrieval results without using any additional discriminative properties of the object. Second, the combination of object contour and skeleton properties significantly improves the effectivity of the shape retrieval methodology. Third, our new method combining contour and skeleton properties for object retrieval outperforms all existing related algorithms.

Environmental Microorganism Classification: EMs and their species are very important indicators to evaluate environmental quality, but their manual classification is very time-consuming [20, 21]. Thus, automatic analysis techniques for microscopic images of EMs would be very appreciated by environmental scientists. We have tested our methodology for this application using the EM-200 dataset. Since some EM-200 objects can hardly be skeletonised (e.g., the first two objects in the third row of Figure 3), we have used the whole microorganism contours for object description without dividing them into segments. Similar to the contour segments, we have normalised the orientation of each object by rotating it so that (i) the straight line connecting two maximally distant contour points has become vertical and (ii) the majority of contour points has lain on the right side of this line.

The impressive results for the EM-200 dataset (see Table 1) confirm the high descriptive power of our new feature extraction technique for contours and prove the applicability of our shape retrieval algorithm to real-world applications.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a method for 2D shape similarity measure based on contour segment matching and, after fusion with a state-of-the-art skeleton-based matching [14], use it for object retrieval. The most innovative part of our approach is the robust comparison and matching of contour segments. The algorithm can easily adapt to a concrete application domain by learning weights assigned to different dimensions of the feature space used for contour description. Its superior performance has been proven in a meaningful experimental setup.

In the future we will investigate possibilities of fusing the discriminative properties of skeletons and contours in an earlier stage of the processing pipeline. Moreover, we will extend the matching algorithm to objects acquired by a depth sensor which requires, for example, an evolution of the CS descriptor to 3D.

7. REFERENCES

[1] Z. Pizlo, *3D Shape - Its Unique Place in Visual Perception*, MIT Press, 2010.

[2] Xiang Bai, Wenyu Liu, and Zhuowen Tu, "Integrating contour and skeleton for shape classification," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 2009, pp. 360–367.

[3] Xiang Bai, L.J. Latecki, and Wen yu Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *PAMI*, vol. 29, no. 3, pp. 449–462, 2007.

[4] Jingting Zeng, Rolf Lakaemper, Xingwei Yang, and Xin Li, "2d shape decomposition based on combined skeleton-boundary features," in *Proceedings of the 4th International Symposium on Advances in Visual Computing, Part II*, Berlin, Heidelberg, 2008, ISVC '08, pp. 682–691, Springer-Verlag.

[5] Xiang Bai, Xinggang Wang, L.J. Latecki, Wenyu Liu, and Zhuowen Tu, "Active skeleton for non-rigid object detection," in *ICCV*, 2009, pp. 575–582.

[6] Duc Thanh Nguyen, Philip O. Ogunbona, and Wanqing Li, "A novel shape-based non-redundant local binary pattern descriptor for object detection," *Pattern Recognition*, vol. 46, no. 5, pp. 1485–1500, 2013.

[7] Yu Cao, Zhiqi Zhang, I. Czogiel, I. Dryden, and Song Wang, "2d nonrigid partial shape matching using mcmc and contour subdivision," in *CVPR*, 2011, pp. 2345–2352.

[8] J. Shotton, A. Blake, and R. Cipolla, "Contour-based learning for object detection," in *ICCV*, 2005, vol. 1, pp. 503–510.

[9] ChengEn Lu, Nagesh Adluru, Haibin Ling, Guangxi Zhu, and Longin Jan Latecki, "Contour based object detection using part bundles," *CVIU*, vol. 114, no. 7, pp. 827–834, 2010.

[10] Xingwei Yang, Hairong Liu, and Longin Jan Latecki, "Contour-based object detection as dominant set computation," *Pattern Recognition*, vol. 45, no. 5, pp. 1927–1936, 2012.

[11] Wooi-Boon Goh, "Strategies for shape matching using skeletons," *CVIU*, vol. 110, no. 3, pp. 326–345, 2008.

[12] T.B. Sebastian and B.B. Kimia, "Curves vs skeletons in object recognition," in *ICIP*, 2001, vol. 3, pp. 22–25.

[13] E. Baseski, A. Erdem, and S. Tari, "Dissimilarity between two skeletal trees in a context," *Pattern Recognition*, vol. 42, no. 3, pp. 370–385, 2009.

[14] Xiang Bai and L.J. Latecki, "Path similarity skeleton graph matching," *PAMI*, vol. 30, no. 7, pp. 1282–1292, 2008.

[15] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, 2002.

[16] N. Hansen, S.D. Muller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES).," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[17] J. Hedrich, C. Yang, C. Feinen, S. Schäfer, D. Paulus, and M. Grzegorzek, "Extended Investigations on Skeleton Graph Matching for Object Recognition," in *8th International Conference on Computer Recognition Systems*, R. Burduk, K. Jackowski, M. Kurzynski, M. Wozniak, and A. Zolnierek, Eds., Milkow, Poland, May 2013, pp. 371–381, Springer LNCS.

[18] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia, "Recognition of shapes by editing their shock graphs," *PAMI*, vol. 26, no. 5, pp. 550–571, may 2004.

[19] L.J. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *CVPR*, 2000, vol. 1, pp. 424–429 vol.1.

[20] C. Li, K. Shirahama, M. Grzegorzek, F. Ma, and B. Zhou, "Classification of environmental microorganisms in microscopic images using shape features and support vector machines," in *ICIP*. September 2013, pp. 2435–2439, IEEE Computer Society.

[21] C. Li, K. Shirahama, J. Czajkowska, M. Grzegorzek, F. Ma, and B. Zhou, "A multi-stage approach for automatic classification of environmental microorganisms," in *IPCV*. July 2013, p. 364370, CSREA Press.