

Extended Investigations on Skeleton Graph Matching for Object Recognition

Jens Hedrich¹, Cong Yang², Christian Feinen², Simone Schäfer¹, Dietrich Paulus¹, and Marcin Grzegorzek²

Abstract Shape similarity estimation of objects is a key component in many computer vision systems. In order to compare two shapes, salient features of a query and target shape are selected and compared with each other, based on a predefined similarity measure. The challenge is to find a meaningful similarity measure that captures most of the original shape properties. One well performing approach called Path Similarity Skeleton Graph Matching has been introduced by Bai and Latecki. Their idea is to represent and match the objects shape by its interior through geodesic paths between skeleton end nodes. Thus it is enabled to robustly match deformable objects. However, insight knowledge about how a similarity measure works is of great importance to understand the matching procedure. In this paper we experimentally evaluate our reimplementations of the Path Similarity Skeleton Graph Matching Algorithm on three 2D shape databases. Furthermore, we outline in detail the strengths and limitations of the described methods. Additionally, we explain how the limitations of the existing algorithm can be overcome.

Key words: Skeleton, Skeleton Graph, Graph Matching, Shape Recognition

1 Introduction

Sensory devices have become increasingly affordable. The processing power as well as storage space have been drastically improved in last decades. The amount of image data is growing rapidly. On the one hand, recording and consumption of such data has been getting easier. On the other hand, complexity of searching and reasoning complicates the access to data [17]. Compared to well-known data types like plain text documents, images are much more sophisticated to manage. For example, searching in semi-structured data like text documents are less complex since searching conditions can be controlled by syntactic means. However, queries based

University of Koblenz-Landau, Germany · University of Siegen, Germany

on images have to be implemented by semantic aspects which are not explicitly known previously. Thus, similarity measures are an ongoing research topic, which is an important contribution to various applications, like multimedia retrieval and object recognition [12]. In order to compare two shapes, salient features of both have to be extracted and compared with each other. Afterwards, shape features of the query object are compared to those of the target. Therefore, an appropriate and predefined similarity measure has to be selected. One challenge is to find a meaningful similarity measure that captures most of the original object's properties. Unlike other similarity measures, the proposed method in this paper establishes correspondences between *skeleton and nodes*. The basic concept is proposed in [2], including a comprehensive evaluation which shows promising results for comparing 2D objects based on skeletons.

We start by discussing the related work and providing a brief explanation of the Path Similarity Skeleton Graph Matching algorithm proposed by Bai et al. [2] (Section 2). In Section 3, we outline some limitations and issues which can appear within the matching process. Section 4 presents the recognition performance of the reimplemented algorithm on three 2D shape databases. Finally, we conclude our work in Section 5.

2 Related Work

Algorithms to analyse objects by shape can be typically categorized by the data representation, namely (i) point set representation, (ii) boundary representation, and (iii) medial representation. The *point set representation* is an unorganized point set. One-to-one correspondences between two point sets are established based on meaningful descriptors [4]. The goal is to find corresponding pairs of points in both shapes that have the highest similarity. The *boundary representations* represent an object by its hull; e.g. *snakes* [7] are used to match objects. Thereby, one idea is to measure the matching energy which is needed to match two contours [18]. The *medial representation* describes the object by its interior, e.g. *skeletons* (also called *medial axes* or *symmetry axes*) are the most propagated medial object representations. They include essential topology and geometrical information [10]. In comparison to boundary representation approaches, skeleton-based methods show their advantages in matching of deformable objects. They are more robust to overlaps, deformations or misplaced object parts [14]. However, skeletons are sensitive to noisy input data, which increases the complexity of the skeletal structure and subsequently the matching complexity. Therefore, the skeleton's quality depends on pre- and post-processing methods, e.g., contour sampling or skeleton pruning. The advantage is that the matching can be reduced to the graph matching problem. Matching single salient skeleton points, e.g. junction nodes or end nodes, is a further approach. In [8, 9] shock trees – a variant of skeletons – are used for shape-comparison based on an edit-distance algorithm. The edit distance is computed by traversing the rooted shock tree while edit operations are being applied to the traversed edges. All edit

operations are associated to a predefined edit cost. The idea is to transform one skeleton branch into in opponent branch. Thus, the similarity between two shapes is measured by summing up the cost of deformations between the shock transitions. Many matching approaches enforce one-to-one correspondences, but noisy image data deal with the problem that one-to-one matchings are not always possible. In [5, 6] the authors present a technique for many-to-many matching of medial axis graphs. The idea is to embed the nodes of two graphs into a fixed-dimension Euclidian space and using the Earth Movers Distance to enable a many-to-many matching between nodes of the graphs.

2.1 Path Similarity Skeleton Graph Matching

Following contents provide a description of the Path Similarity Skeleton Graph Matching algorithm, proposed by Bai and Latecki in [2]. Our aim is to provide a deep understanding of the concept in order to explain the investigated limitations of this algorithm in Section 3. The algorithm can be divided into three major parts: (i) getting a compact skeleton representation, (ii) computing the matching costs between the end points, (iii) repeating the latter procedure for all combinations of end nodes. The final matching of the skeleton end nodes is performed by applying the Hungarian algorithm.

2.1.1 Skeleton Representation

A key concept is the use of information about skeleton paths. The skeleton representation incorporates the main contour information. The example of the two bird shapes in Fig. 1a-b will guide through the explanations. A skeleton path $p(v_m, v_n)$ is defined as the shortest path between a pair of end nodes v_m and v_n passing the skeletal structure (see Fig. 1c). The skeleton paths are found by constructing a weighted skeleton graph. The edge weight is defined by the length of the corresponding skeleton branch. Hence, it is possible to apply a shortest path algorithm (e.g. Dijkstra's algorithm). Additionally, information about the object contour is included. Thus, a skeleton path $p(v_m, v_n)$ is sampled with M equidistant points. Afterwards, all radii of maximal disks, measured at each sampled point t , are noted within a path vector (cf. Eq. 1 and Fig. 1d):

$$R_{m,n} = (R_{m,n}(t))_{t=1,2,\dots,M} = (r_1, r_2, \dots, r_M), \quad (1)$$

In [2] the distance of each sampled skeleton point t to its feature point is approximated by a distance transform $DT(t)$, in our case a Euclidian distance map is computed. Afterwards, the distance is normalized to make the method invariant to scale. Finally, the distances are approximated and normalized as follows:

$$R_{m,n} = \frac{DT(t)}{\frac{1}{N_0} \sum_{i=1}^{N_0} DT(s_i)}, \quad (2)$$

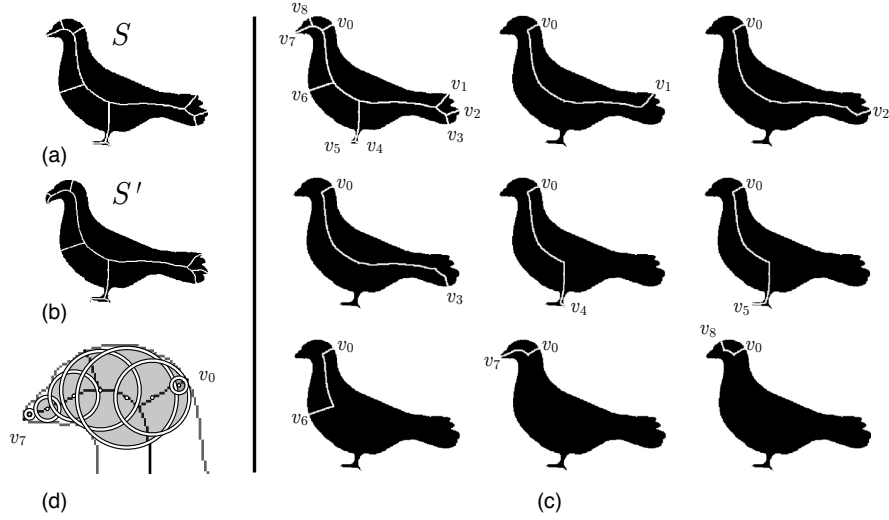


Fig. 1: (a-b) Example for two skeletons S and S' to be matched. (c) The complete skeleton (top left) and all skeleton paths emanating of one example end node. (d) Sampling of a skeleton path. The sampling points are indicated with white dots. The distance to their feature points is indicated by the black circles. For the skeleton path representation, the normalized distance of the skeleton points to their feature points is measured and noted in the skeleton path vector.

where N_0 is the number of pixels in the original shape and $s_i (i = 1, 2, \dots, N_0)$ varies over all N_0 pixels within the shape. The ordered list of M distance values is obtained for each skeleton path. All distance values are noted in the path vector (cf. Eq. 1).

2.1.2 Dissimilarity between End Nodes

In order to compute the matching costs for two end nodes, a similarity measure is necessary. Therefore, the dissimilarity of two skeleton paths is given by the *path distance* shown in (Eq. 3) (where r_i and r'_i are radii of maximal disks of the path vectors R and R' , l and l' are the length of the skeleton paths $p(u, v)$ and $p(u', v')$). The influence of the path length is weighted by the factor $\alpha \in \mathbb{R}^+$. To make the approach invariant to scale, the lengths are normalized. The assumption is that similar skeleton paths have consecutive skeleton points with similar radii of maximal, inscribed discs.

$$pd(p(u, v), p(u', v')) = \sum_{i=1}^M \frac{(r_i - r'_i)^2}{r_i + r'_i} + \alpha \frac{(l - l')^2}{l + l'} \quad (3)$$

All path distances for one pair of end nodes are combined in one *path distance matrix* (**PDM**) (Eq. 4). Two skeleton graphs G with $K + 1$ end nodes and G' with $N + 1$ end nodes are matched with $K \leq N$. The end nodes v_i and v'_j of G and G' are ordered by traversing the object contours in clockwise direction. This leads to an

ordered list of end nodes: $\{v_{i0}, v_{i1}, \dots, v_{iK}\}$ for G and $\{v'_{j0}, v'_{j1}, \dots, v'_{jN}\}$ for G' .

$$\mathbf{PDM}(v_{i0}, v'_{j0}) = \begin{pmatrix} pd(p(v_{i0}, v_{i1}), p(v'_{j0}, v'_{j1})) & \cdots & pd(p(v_{i0}, v_{i1}), p(v'_{j0}, v'_{jN})) \\ \vdots & \vdots & \vdots \\ pd(p(v_{i0}, v_{iK}), p(v'_{j0}, v'_{j1})) & \cdots & pd(p(v_{i0}, v_{iK}), p(v'_{j0}, v'_{jN})) \end{pmatrix} \quad (4)$$

To estimate a dissimilarity value for a pair of end nodes v_i and v'_j , the **PDM** is applied to the *Optimal Subsequence Bijection (OSB)* [11]. By using this approach, the problem of estimating the similarity of two end nodes is reduced to elastic matching of time series. One main advantage of using *OSB* is that outliers within the path distance matrix can be easily skipped. The **PDM** is computed for every combination of end nodes in two skeleton graphs and afterwards applied to the *OSB* function: $c(v_i, v'_j) = OSB(\mathbf{PDM}(v_i, v'_j))$. Subsequently, the resulting cost matrix C (cf. eq. 5) is used as input for the Hungarian algorithm. Hence, the matching problem is reduced to the classic assignment problem in a bipartite graph.

$$C(G, G') = \begin{pmatrix} c(v_0, v'_0) & c(v_0, v'_1) & \cdots & c(v_0, v'_N) \\ c(v_1, v'_0) & c(v_1, v'_1) & \cdots & c(v_1, v'_N) \\ \vdots & \vdots & \ddots & \vdots \\ c(v_K, v'_0) & c(v_K, v'_1) & \cdots & c(v_K, v'_N) \end{pmatrix} \quad (5)$$

3 Investigations on the Path Similarity Skeleton Graph Matching Algorithm

The following investigations are based on a reimplementation of the Path Similarity Skeleton Graph Matching algorithm proposed in [2]. Deduced from our experience, we detected three major limitations, which can occur in special matching cases: flipped images, 1-to-1 matching of the end nodes, and spurious skeleton branches.

3.1 Flipped Images

The cheapest path through a given path distance matrix (**PDM**) is estimated by the *OSB*-Function. It is assumed that the cheapest path for two corresponding end nodes goes from the upper left corner to the lower right corner. By traversing the matrix with the *OSB* function, it is not allowed to go backwards, neither in the rows nor in the columns. In the case of matching two similar shapes which head towards different directions, the correct matching costs cannot be estimated with the *OSB* function (cf. Fig. 2a). In these cases, the actual cheapest path through the matrix is flipped and goes from the upper right corner to the lower left corner (cf. Eq. 6). This means that the *OSB* function is not able to estimate a reliable indicator for the similarity.

$$\mathbf{PDM}(v_{i0}, v'_{j0}) = \begin{pmatrix} \dots & 7 & 5 & 8 & 0 \\ \dots & 2 & 3 & 0 & 8 \\ \dots & 5 & 0 & 3 & 5 \\ \dots & 0 & 5 & 2 & 7 \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad (6)$$

As a solution, we apply the *OSB* function twice: once for the original image, and once with one image flipped horizontally. From the resulting two match lists, the one with lower matching costs will be chosen as the real matching. In the most cases this works quite well. The minimal matching cost value is taken as similarity value. However, this strategy can fail in the case of strong dissimilarities between two shapes. Only, a more complex strategy which compares contour partitions in detail can overcome this failure. For example, the shapes in Fig. 2a are oriented in opposite directions, but the matching costs for the second run with one of the images flipped leads to lower matching costs than the first run. Thus, in the algorithm the two shapes are assumed to be oriented in the same direction, which in the end leads to an unsatisfying matching.

3.2 1-to-1 Matching

1-to-1 matching of end nodes is another issue which we identified. It is not always possible to assign a correct matching partner to each end node. For example Fig. 2b depicts an acceptable matching, but both skeletons have one additional end node that has no matching partner. Within the Hungarian algorithm each end node has to be assigned to one partner, even though they do not correspond. This is not only a limitation of this particular algorithm, but a problem of all matching algorithms that reduce the matching problem to a 1-to-1 matching in a bipartite graph. Using a different approach could be a solution to this problem. For example, the Earth Mover's Distance (EMD) as used in [13] also allows partial matchings. This could be a solution to better deal with noisy skeleton data.

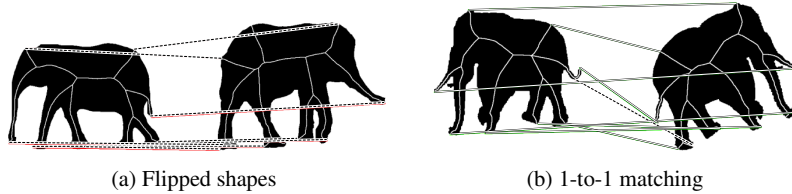


Fig. 2: a) If one shape is flipped, it is likely that the method fails to determine. b) 1-to-1 matching is not always possible. In this example, all matchings have been found correctly, but the two remaining end nodes with no matching partner in the other skeleton are matched.

3.3 Spurious Skeleton Branches

The Path Similarity Skeleton Graph Matching algorithm requires perfectly regularized skeletons, where each skeleton branch represents a significant visual part of the shape. Since each end node has to be assigned to an 1-to-1 matching partner, spurious skeleton branches have profound negative impact on the matching result. Fig. 3a illustrates wrong assignments between two elephant shapes, which are caused by a spurious branch in the tail of the left elephant (see Fig. 3b). After manually removing this branch the number of correct correspondences has significantly increased (cf. Fig. 3c). Thus, one has to make sure that the input skeletons do not contain spurious branches.

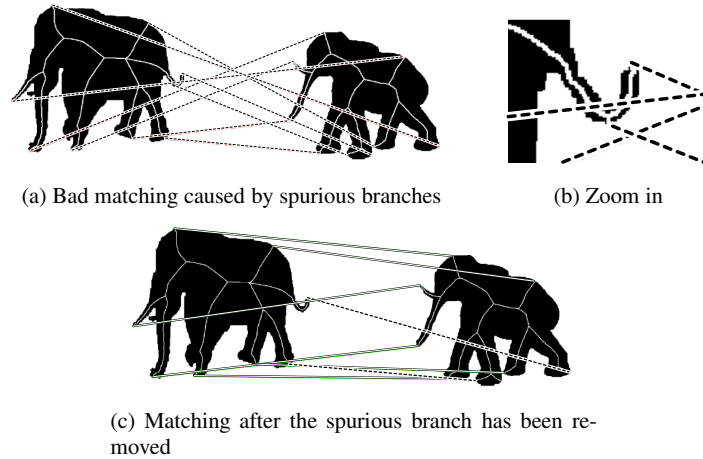


Fig. 3: One spurious branch can have a high impact on the matching result

4 Recognition Performance

Based on our reimplementation, we used several shape databases like Aslan and Tari [1], kimia-99 [15] and kimia-216 [16] to evaluate the recognition performance. The skeleton of each shape is computed by the *Discrete Curve Evolution* (DCE) algorithm [3]¹, with the parameters $\rho = 4$, $T_1 = 1$ and $number_vertice = 15$. The parameters for the similarity measurement were $M = 50$ and $\alpha = 40$. All shapes from the databases has been used as a query. In order to rate the retrieval, the average precision is computed and for each query the maximal number of shapes within the retrieval class is returned. For each query on the Aslan and Tari database, three shapes are returned, which leads to a average precision of 0.93. For each query on the kimia-99 database 9 shapes are returned, which results in an average precision of

¹ available at <https://sites.google.com/site/xiangbai/BaiSkeletonPruningDCE.zip>

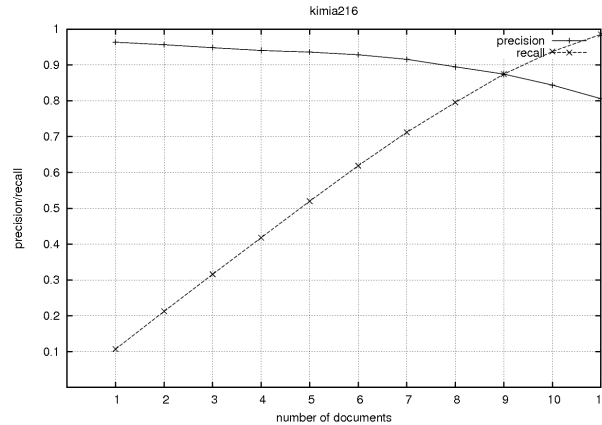


Fig. 4: Average precision and recall development in the kimia216 database with increasing number of result documents.

0.84. In the kimia-216 database 11 shapes are returned for each query, the average precision is 0.81 (cf. Fig. 4). Table 1 summarizes the number of all correct shapes for the first eleven retrieval results from the kima216 database in comparison to the values listed in the original paper. Obviously, the results are not as good as in the original paper. It is assumed that the input skeletons play a significant role. Several skeletons used in the experiments contain spurious branches, which had a profound impact on the matching quality and led to distorted overall similarity values. This effect has been observed in many of the query results. However, the originally used parameters for the DCE algorithm are not reported.




























Table 1: Summary of correct shapes in the 1st, 2nd,.. retrieval result.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
Original paper	216	216	215	216	213	210	210	207	205	191	177
Our results	210	208	203	202	200	192	186	167	161	130	96

To verify this assumption, further experiments were performed on the more problematic classes of the Kima-216 database. This time, the skeletons in the database were pruned manually so that each skeleton branch represents a significant visual part of the original shape. As the significant parts of shapes of the same class should be quite similar, the skeletons get more comparable. Using the manually pruned skeletons leads to better results in the performed queries. For example, the average precision for the queries from the 'bird'-class went up from 0.69 to 0.81, the average precision for the queries from the 'camel' class went up from 0.63 to 0.73 (see table 2). Additionally, it can be observed that the average precision value for the Kimia-99 and Kimia-216 database is worse than for the Aslan and Tari database. The reason for this partly lies in the composition of data in both databases. In the Aslan and Tari database, the algorithm's performance for non-rigid shapes is mainly evaluated. Parts of the shapes are bent and the shapes are similar to each other within

a class. The main challenge in the Kimia-99 database is that several shapes are occluded.

Table 2: Example queries on the Kimia-216 database. In the left column, the query shape is shown. From left to right, the eight most similar shapes in the database are shown. The similarity to the query drops from left to right.

Query	1st	2nd	3rd	4th	5th	6th	7th	8th
								
								
								

5 Conclusion

In this paper we reimplemented the Path Similarity Skeleton Graph Matching algorithm [2] and performed a reevaluation on three shape databases (Aslan and Tari, kimia-99 and kimia-216). Additionally, we reported the limitations of the algorithm in detail for the first time. A fundamental understanding is necessary to understand upcoming issues during a matching process. The algorithm showed its advantages when dealing with non-rigid objects and articulated joints. An average precision of 0.93 (0.98 with manually pruned skeletons, respectively) for the Aslan and Tari database shows that shape deformations do not affect the skeleton's topology and the use of path radii has no impact on the matching results. The experiments with the Kimia-99 and Kimia-216 database also showed acceptable results with an average precision of 0.84 and 0.81, respectively. In this paper, we were not able to reproduce the excellent recognition results of the original paper, due to the lack of the not reported parameters for the skeletonization method. However, problems in the recognition performance occurred when shapes of different classes were similar. In addition, overlaps have a negative impact on the recognition results. A severe limitation of the algorithm is its requirement of optimal skeletons. The experiments showed that spurious branches in one of the skeletons lead to distorted matching results in several cases. Consequently, this affects the object recognition performance when using the described matching algorithm in the retrieval system. In the future, we will investigate strategies how to deal with flipped images and pruning algorithms for skeletons to reduce the number of spurious branches. Furthermore, this matching algorithm could be used as an initial similarity measurement in a hierarchical and more complex object recognition system. Another challenge is the rapidly growing amount of 3D data. An extension of the Path Similarity Skeleton Graph Matching method towards the third dimension would be a great contribution to the object recognition community.

References

1. Cagri Aslan and Sibel Tari. An axis-based representation for recognition. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1339–1346, 2005.
2. Xiang Bai and L.J. Latecki. Path similarity skeleton graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1282–1292, 2008.
3. Xiang Bai, Longin Jan Latecki, and Wen-Yu Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:449–462, 2007.
4. Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.
5. M. Fatih Demirci, Ali Shokoufandeh, Yakov Keselman, Lars Bretzner, and Sven J. Dickinson. Object recognition as many-to-many feature matching. *Int. J. Comput. Vision*, 69:203–222, 2006.
6. M. Fatih Demirci, Ali Shokoufandeh, and Sven Dickinson. Skeletal shape abstraction from examples. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 31(5):944–952, 2009.
7. Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
8. Philip Klein, Tirthapura Srikanta, Daniel Sharvit, and Ben Kimia. A tree-edit-distance algorithm for comparing simple, closed shapes. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 696–704, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematic.
9. Philip N. Klein, Thomas B. Sebastian, and Benjamin B. Kimia. Shape matching using edit-distance: an implementation. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 781–790, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematic.
10. J. Komala Lakshmi and M. Punithavalli. A survey on skeletons in digital image processing. In *Proceedings of the International Conference on Digital Image Processing*, pages 260–269, 2009.
11. Longin Jan Latecki, Qiang Wang, Suzan Koknar-Tezel, and Vasileios Megalooikonomou. Optimal subsequence bijection. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 565–570, Washington, DC, USA, 2007. IEEE Computer Society.
12. Tadeusiewicz R. *How Intelligent Should Be System for Image Analysis? Preface to book: Kwasnicka H., Jain L.C. (Eds.): Innovations in Intelligent Image Analysis. Studies in Computational Intelligence*, page VX. Springer-Verlag, <http://www.springer.com>, 2011.
13. Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40:99–121, 2000.
14. Thomas B. Sebastian and Benjamin B. Kimia. Curves vs. skeletons in object recognition. *Signal Processing*, 85(2):247–263, 2005.
15. Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of shapes by editing shock graphs. In *IEEE International Conference on Computer Vision*, pages 755–762, 2001.
16. Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004.
17. R. Tadeusiewicz. What does it means ”automatic understanding of the images”? In *Proceedings of the 2007 IEEE International Workshop on Imaging Systems and Techniques*, pages 1–3, may 2007.
18. Laurent Younes. Computable elastic distances between shapes. *SIAM J. Appl. Math.*, 58:565–586, 1998.