in Proc. 7th IAPR-TC-15 Workshop Graph-based Representations in Pattern Rec. (GbR), 2009

Matching Hierarchies of Deformable Shapes

Nadia Payet and Sinisa Todorovic

Oregon State University, Corvallis, OR 97331, USA,
payetn@onid.orst.edu, sinisa@eecs.oregonstate.edu

Abstract. This paper presents an approach to matching parts of deformable shapes. Multiscale salient parts of the two shapes are first identified. Then, these parts are matched if their immediate properties are similar, the same holds recursively for their subparts, and the same holds for their neighbor parts. The shapes are represented by hierarchical attributed graphs whose node attributes encode the photometric and geometric properties of corresponding parts, and edge attributes capture the strength of neighbor and part-of interactions between the parts. Their matching is formulated as finding the subgraph isomorphism that minimizes a quadratic cost. The dimensionality of the matching space is dramatically reduced by convexifying the cost. Experimental evaluation on the benchmark MPEG-7 and Brown datasets demonstrates that the proposed approach is robust.

1 Introduction

This paper is about shape matching by using: (1) a new hierarchical shape representation, and (2) a new quadratic-assignment objective function that is efficiently optimized via convexification. Many psychophysical studies suggest that shape perception is the major route for acquiring knowledge about the visual world [1]. However, while humans are very efficient in recognizing shapes, this proves a challenging task for computer vision. This is mainly due to certain limitations in existing shape representations and matching criteria used, which typically cannot adequately address matching of deformable shapes. Two perceptually similar deformable shapes may have certain parts very different or even missing, whereas some other parts very similar. Therefore, accounting for shape parts in matching is important. However, it is not always clear how to define a shape part. The motivation behind the work described in this paper is to improve robustness of shape matching by using a rich hierarchical shape representation that will provide access to all shape parts existing at all scales, and by formulating a matching criterion that will account for these shape parts and their hierarchical properties.

We address the following problem: Given two shapes find correspondences between all their parts that are similar in terms of photometric, geometric, and structural properties, the same holds recursively for their subparts, and the same holds for their neighbor parts. To this end, a shape is represented by a hierarchical attributed graph whose node attributes encode the intrinsic properties of corresponding multiscale shape parts (e.g., intensity gradient, length, orientation), and edge attributes capture the strength of *neighbor* and *part-of* interactions between the parts. We formulate shape matching as finding the subgraph isomorphism that preserves the original graph connectivity and

minimizes a quadratic cost whose linear and quadratic terms account for differences between node and edge attributes, respectively. The cost is defined so as to be invariant to scale changes and in-plane rotation of the shapes. The search in the matching space of all shape-part pairs is accelerated by convexifying the quadratic cost, which also reduces the chances to get trapped in a local minimum. As our experiments demonstrate, the proposed approach is robust against large variations of individual shape parts and partial occlusion.

In the rest of this paper, Sec. 2 points out main contributions of our approach with respect to prior work, Sec. 3 describes our hierarchical representation of a shape, Sec. 4.1 specifies node and edge compatibilities and formulates our matching algorithm, Sec. 4.2 explains how to convexify and solve the quadratic program, and Sec. 5 presents experimental evaluation of our approach.

2 Our Contributions and Relationships to Prior Work

This section reviews prior work and points out our main contributions. Hierarchical shape representations are aimed at efficiently capturing both global and local properties of shapes, and thus facilitating their matching. Shortcomings of existing representations typically reduce the efficiency of matching algorithms. For example, the arc-tree [2, 3] trades off its accuracy and stability for lower complexity, since it is a binary tree, generated by recursively splitting the curve in two halves. Arc-trees are different for similar shapes with some part variations, which will be hard to match. Another example is the curvature scale-space [4, 5] that loses its descriptive power by pre-specifying the degree of image decimation (i.e., blurring and subsampling), while capturing salient curvature points of a contour at different degrees of smoothing. Also, building the articulation-invariant, part-based signatures of deformable shapes, presented in [6], is sensitive to the correct identification of the shape's landmark points and to the multidimensional scaling and estimating of the shortest path between these points. Other hierarchical shape descriptions include the Markov-tree graphical models [7], and the hierarchy of polygons [8] that are based on the restrictive assumptions about the number, size, and hierarchy depth of parts that a curve consists of. The aforementioned methods encode only geometric properties of shape parts, and their part-of relationships, yielding a strict tree. In contrast, we use a more general, hierarchical graph that encodes the strength of all ascendant-descendant and neighbor relationships between shape parts, as well as their geometric and photometric properties. The sensitivity of the graph structure to small shape variations is reduced, since we estimate the shape's salient points at multiple scales. Also, unlike in prior work, the number of nodes, depth, and branching factor in different parts of the hierarchical graph are data dependent.

Graph-based shape matching has been the focus of sustained research activity for more than three decades. Graph matching may be performed by: (i) exploiting spectral properties of the graphs' adjacency matrices [9, 10]; (ii) minimizing the graph edit-distance [11, 12]; (iii) finding a maximum clique of the association graph [13]; (iv) using the expectation-maximization of a statistical, generative model [14]. Regardless of a particular formulation, graph matching in general can be cast as a quadratic assignment problem, where a linear term in the objective function encodes node compatibility func-

tions, and a quadratic term encodes edge compatibility functions. Therefore, approaches to graph matching mainly focus on: (i) finding suitable definitions of the compatibility functions; and (ii) developing efficient algorithms for approximately solving the quadratic assignment problem (since it is NP-hard), including a suitable reformulation of the quadratic into linear assignment problem. However, most popular approximation algorithms (e.g., relaxation labeling, and loopy belief propagation) critically depend on a good initialization and may be easily trapped in a local minimum, while some (e.g., deterministic annealing schemes) can be used only for graphs with a small number of nodes. Graduated nonconvexity schemes [15], and successive convexification methods [16] have been used to convexify the objective function of graph matching, and thus alleviate these problems. Since it is difficult to convexify matching cost surfaces that are not explicit functions, these methods resort to restrictive assumptions about the functional form of a matching cost, or reformulate the quadratic objective function into a linear program. In this paper, we develop a convexification scheme that shrinks the pool of matching candidates for each individual node in the shape hierarchy, and thus renders the objective function amenable to solution by a convex quadratic solver.

3 Hierarchical Shape Representation

In this paper, a shape (also called contour or curve) is represented by a hierarchical graph. We first detect the contour's salient points at multiple scales, which in turn define the corresponding shape parts. Then, we derive a hierarchy of these shape parts, as illustrated in Fig. 1.

Multiscale part detection: A data-driven number of salient (or dominant) points along the contour are detected using the scale-invariant algorithm of [17]. This algorithm does not require any input parameters, and remains reliable even when the shape is rich in both fine and coarse details, unlike most existing approaches. The algorithm first determines, for each point along the curve, its curvature and the region of support, which jointly serve as a measure of the point's relative significance. Then, the dominant points are detected by the standard nonmaximum suppression. Each pair of subsequent dominant points along the shape define the corresponding shape part. The end points of each shape part define a straight line that is taken to approximate the part. We recursively apply the algorithm of [17] to each shape part whose associated line segment has a larger approximation error than a pre-set threshold. This threshold controls the resolution level (i.e., scale) at which we seek to represent the contour's fine details. How to compute this approximation error is explained later in this section. After the desired resolution level is reached, the shape parts obtained at different scales can be organized in a tree structure, where nodes and parent-child (directed) edges represent the shape parts and their part-of relationships. The number of nodes, depth, and branching factors of each node of this tree are all automatically determined by the shape at hand.

Transitive closure: Small, perceptually negligent shape variations (e.g., due to varying illumination in images) may lead to undesired, large structural changes in the shape tree (e.g., causing a tree node to split into multiple descendants at multiple levels). As in [18], we address these potential structural changes of the shape tree by adding new directed edges that connect every node with all of its descendants, resulting in a

transitive closure of the tree. Later, in matching, the transitive closures will allow that a search for a maximally matching node pair is conducted over all descendants under a visited ancestor node pair, rather than stopping the search if the ancestors' children do not match. This, in turn, will make matching more robust.

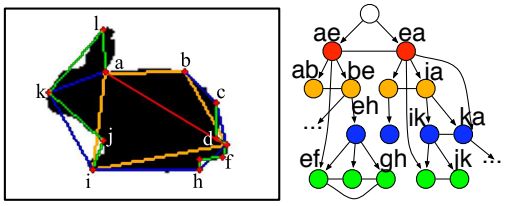

Fig. 1. An example contour: (left) Lines approximating the detected contour parts are marked with different colors. (right) The shape parts are organized in a hierarchical graph that encodes their part-of and neighbor relationships. Only a few ascendant-descendant and neighbor edges are depicted for clarity.

Neighbors: Like other strictly hierarchical representations, the transitive closure of the shape tree is capable of encoding only a limited description of spatial-layout properties of the shape parts. For example, it cannot distinguish different layouts of the same set of parts along the shape. In the literature, this problem has been usually addressed by associating a context descriptor with each part. In this paper, we instead augment the transitive closure with new, undirected edges, capturing the neighbor relationships between parts. This transforms the transitive closure of the shape tree into a more general graph that we call the shape hierarchy.

Node Attributes: Both nodes and edges of the shape hierarchy are attributed. Node attributes are vectors whose elements describe photometric and geometric properties of the corresponding shape part. The following estimates help us define the shape properties. We estimate the contour's mean intensity gradient, and use this vector to identify the contour's direction – namely, the sequence of points along the shape – by the right-hand rule. The principal axis of the entire contour is estimated as the principal axis of an ellipse fitted to all points of the shape. The attribute vector of a node (i.e., shape part) includes the following properties: (1) length as a percentage of the parent length; (2) angle between the principal axes of this shape part and its parent; (3) approximation error estimated as the total area between the shape part and its associated straight line, expressed as a percentage of the area of the fitted ellipse; (4) signed approximation error is similar to the approximation error except that the total area between the shape part and its approximating straight line is computed by accounting for the sign of the intensity gradient along the shape; and (5) curvature at the two end points of the shape part. All the properties are normalized to be in $[0, 1]$.

Edge Attributes: The attribute of an edge in the shape hierarchy encodes the strength of the corresponding part-of or neighbor relationship. Given a directed edge between a shape part and its descendant part, the attribute of this edge is defined as the percentage that the length of the descendant makes in the length of the shape part. Thus, the shorter

descendant or the longer ancestor, the smaller strength of their interaction. The attribute of an undirected edge between two shape parts can be either 1 or 0, where 1 means that the parts have one common end point, and 0 means that the parts are not neighbors.

4 Shape Matching

Given two shapes, our goal is to identify best matching shape parts and discard dissimilar parts, so that the total cost is minimized. This cost is defined as a function of geometric, photometric, and structural properties of the matched parts, their subparts, and their neighbor parts, as explained below.

4.1 Definition of the Objective Function of Matching

Let $H = (V, E, \psi, \phi)$ denote the shape hierarchy, where $V = \{v\}$ and $E = \{(v, u)\} \subseteq V \times V$ are the sets of nodes and edges, and ψ and ϕ are functions that assign attributes to nodes, $\psi : V \rightarrow [0, 1]^d$, and to edges, $\phi : E \rightarrow [0, 1]$. Given two shapes, H and H', the goal of the matching algorithm is to find the subgraph isomorphism, $f{:}U \rightarrow U'$, where $U \subseteq V$ and $U' \subseteq V'$, which minimizes the cost, C, defined as

$$C = \left[\beta \sum_{v \in V} c_1(v, f(v)) + (1 - \beta) \sum_{(v,u) \in E} c_2(v, f(v), u, f(u))\right], \qquad (1)$$

where c_1 is a non-negative cost function of matching nodes v and $v' = f(v)$, and c_2 is a non-negative cost function of matching edges (v, u) and (v', u'), and $\beta \in [0, 1]$ weights their relative significance to matching. To minimize C, we introduce a vector, X, indexed by all node pairs $(v, v') \in V \times V'$, whose each element $x_{vv'} \in [0, 1]$ encodes the confidence that pair (v, v') should be matched. Matching can then be reformulated as estimating X so that C is minimized. That is, we use the standard linearization and relaxation of (1) to obtain the following quadratic program (QP):

$$\begin{array}{l} \min_X \left[\beta A^T X + (1 - \beta) X^T B X\right], \\ \text{s.t. } \forall (v, v') \in V \times V', x_{vv'} \geq 0, \;\; \forall v' \in V', \sum_{v \in V} x_{vv'} = 1, \;\; \forall v \in V, \sum_{v' \in V'} x_{vv'} = 1, \end{array} \qquad (2)$$

where A is a vector of costs $a_{vv'}$ of matching nodes v and v', and B is a matrix of costs $b_{vv'uu'}$ of matching edges (v, u) and (v', u'). We define $a_{vv'} = \frac{1}{d}\|\psi(v) - \psi(v')\|_2$, where d is the dimensionality of the node attribute vector. Also, we define $b_{vv'uu'}$ so that matching edges of different types is prohibited, and matches between edges of the same type with similar properties are favored in (2): $b_{vv'uu'} = \infty$ if edges (v, u) and (v', u') are not of the same type; and $b_{vv'uu'} = |\phi(v, v') - \phi(u, u')| \in [0, 1]$ if edges (v, u) and (v', u') are of the same type.

The constraints in (2) are typically too restrictive, because of potentially large structural changes of V or E in H that may be caused by relatively small variations of certain shape parts. For example, suppose H and H' represent similar shapes. It may happen that node v in H corresponds to a subgraph consisting of nodes $\{v'_1, \ldots, v'_m\}$ in H', and vice versa. Therefore, a more general many-to-many matching formulation would be more appropriate for our purposes. The literature reports a number of heuristic approaches to many-to-many matching [19–21], which however are developed only

for weighted graphs, and thus cannot be used for our shape hierarchies that have attributes on both nodes and edges. To relax the constraints in (2), we first match H to H', which yields solution X_1. Then, we match H' to H, which yields solution X_2. The final solution, $\tilde{X}$, is estimated as an intersection of non-zero elements of X_1 and X_2. Formally, the constraints are relaxed as follows: (i) $\forall (v, v') \in V \times V', x_{vv'} \geq 0$; and (ii) $\forall v \in V, \sum_{v' \in V'} x_{vv'} = 1$ when matching H to H'; and $\forall v' \in V', \sum_{v \in V} x_{vv'} = 1$ when matching H' to H.

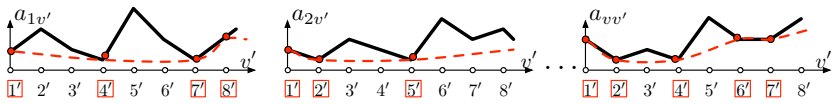

Fig. 2. Convexification of costs $\{a_{vv'}\}_{v' \in V'}$ for each node $v \in V$. Matching candidates of v that belong to the region of support of the lower convex hull, $v' \in \breve{V}'(v)$, are marked red.

4.2 Convexification of the Objective Function of Matching

The QP in (2) is in general non-convex, and defines a matching space of typically 10^4 possible node pairs in our experiments. In order to efficiently find a solution, we convexify the QP. This significantly reduces the number of matching candidates.

Given H and H' to be matched, for each node $v \in V$ of H, we identify those matching candidates $v' \in V'$ of H' that form the region of support of the lower convex hull of costs $\{a_{vv'}\}_{v' \in V'}$, as illustrated in Fig. 2. Let $\breve{V}'(v) \subset V'$ denote this region of support of the convex hull, and let $\tilde{V}'(v) \subset V'$ denote the set of true matches of node v that minimize the QP in (2) (i.e., the solution). Then, by definition, we have that $\tilde{V}'(v) \subseteq \breve{V}'(v)$, i.e., the true matches must be located in the region of support of the convex hull. It follows, that for each node $v \in V$, we can discard those matching candidates from V' that do not belong to $\breve{V}'(v)$. In our experiments, we typically obtain $|\breve{V}'(v)| \ll |V'|$, which leads to a dramatic reduction of the dimensionality of the original matching space, $|V \times V'|$.

In summary, we compute $\breve{A}$, $\breve{X}$, $\breve{B}$ from original A, X, B, respectively, by deleting all their elements $a_{vv'}$, $x_{vv'}$, $b_{vv'uu'}$ for which $v' \notin \breve{V}'(v)$. Then, we use the standard interior-reflective Newton method to solve the following program:

$$\begin{aligned} &\min_{\breve{X}} \left[\beta \breve{A}^T \breve{X} + (1-\beta) \breve{X}^T \breve{B} \breve{X} \right], \\ &\text{s.t. } \forall (v, v') \in V \times \breve{V}'(v), \;\; x_{vv'} \geq 0, \;\; \forall v \in V, \;\; \textstyle\sum_{v' \in \breve{V}'(v)} x_{vv'} = 1. \end{aligned} \tag{3}$$

5 Results

This section presents the experimental evaluation of our approach on the standard MPEG-7 and Brown shape datasets [12]. MPEG-7 has 1400 silhouette images showing 70 different object classes, with 20 images per object class, as illustrated in Fig. 3. MPEG-7

presents many challenges due to a large intra-class variability within each class, and small differences between certain classes. The Brown shape dataset has 11 examples from 9 different object categories, totaling 99 images. This dataset introduces additional challenges, since many of the shapes have missing parts (e.g., due to occlusion), and the images may contain clutter in addition to the silhouettes, as illustrated in Figs. 1, 4, 5. We use the standard evaluation on both datasets. For every silhouette in MPEG-7, we retrieve the 40 best matches, and count the number of those that are in the same class as the query image. The retrieval rate is defined as the ratio of the total number of correct hits obtained and the best possible number of correct hits. The latter number is $1400 \cdot 20$. Also, for each shape in the Brown dataset, we first retrieve the 10 best matches, then, check if they are in the same class as the query shape, and, finally, compute the retrieval rate, as explained above. Input to our algorithm consists of two parameters: the fine-resolution level (approximation error defined in Sec. 3) of representing the contour, and β. For silhouettes in both datasets, and the approximation error (defined in Sec. 3) equal to 1%, we obtain shape hierarchies with typically 50-100 nodes, maximum hierarchy depths of 5-7, and maximum branching factors of 4-6. For every query shape, the distances to other shapes are computed as the normalized total matching cost, D, between the query and these other shapes. If X is the solution of our quadratic programming, then $D{=}[\beta A^T X + (1{-}\beta) X^T B X]/[|V| + |V'|]$, where $|V|$ is the total number of nodes in one shape hierarchy. Matching two shape hierarchies takes about 5-10sec in MATLAB on a 3.1GHz, 1GB RAM PC.

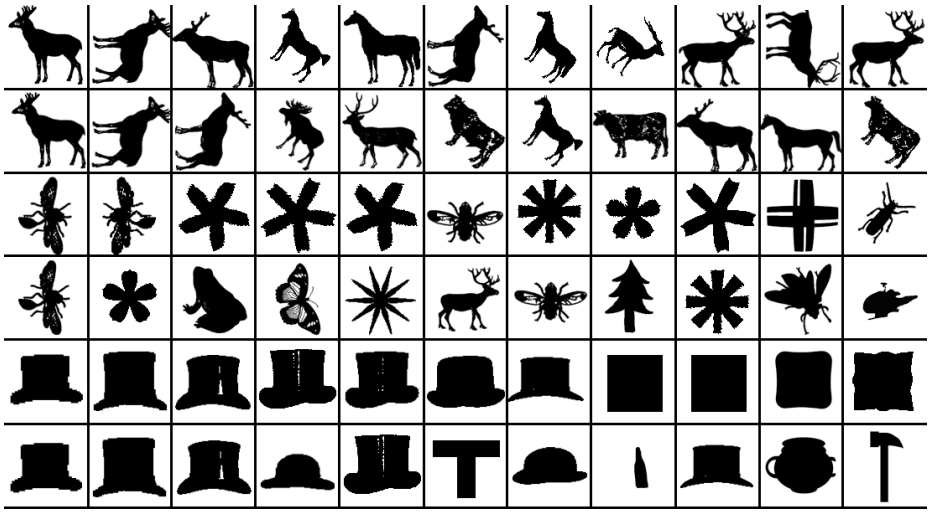

Fig. 3. MPEG-7 retrieval results on three query examples and comparison with [6]. For each query, we show 11 retrieved shapes with smallest to highest cost. (top) Results of [6]. (bottom) Our results. Note that for deer we make the first mistake in 6th retrieval, and then get confused with shapes whose parts are very similar to those of deer. Mistakes for other queries usually occur due to missing to capture fine details of the curves in the shape hierarchy in our implementation.

Fig. 4. The Brown dataset – each of the four columns shows one example pair of silhouettes, and each of the two rows shows shape parts at a specific scale that got matched; top row shows finer scale and bottom row shows coarser scale. As can be seen, silhouettes that belong to the same class may have large differences; despite the differences, corresponding parts got successfully matched (each match is marked with unique color).

Fig. 5. The Brown dataset – two example pairs of silhouettes, and their shape parts that got matched. The shapes belong to different classes, but the algorithm identifies their similar parts, as expected (each match is marked with unique color). The normalized total matching cost between the bunny and gen (left), or the fish and tool (right) is larger than the costs computed for the examples shown in Fig 4, since there are fewer similar than dissimilar parts. ($\beta = 0.4$)

Qualitative evaluation: Fig. 3 shows a few examples of our shape retrieval results on MPEG-7. From the figure, our approach makes errors mainly due to the non-optimal pre-setting of the fine-resolution level at which contours are represented by the shape hierarchy. Also, some object classes in the MPEG-7 are characterized by multiple disjoint contours, whereas our approach is aimed at matching only one single contour at a time. Next, Fig. 4 shows four example pairs of silhouettes from the same class, and their matched shape parts. Similar shape parts at multiple scales got successfully matched in all cases, as expected. Fig. 4 presents two example pairs of silhouettes that belong to different classes. As in the previous case, similar shape parts got successfully matched; however, since there are fewer similar than dissimilar parts the normalized total match-

ing cost in this case is larger. This helps discriminate between the shapes from different classes in the retrieval.

Quantitative evaluation: To evaluate the sensitivity of our approach to input parameter β, we compute the average retrieval rate on the Brown dataset as a function of input $\beta = 0.1 : 0.1 : 0.9$. The maximum retrieval rate of 99% is obtained for β=0.4, while for $\beta = \{0.3, 0.5, 0.6\}$ we obtain the rate of 98%. This suggests that both intrinsic properties of shape parts and their spatial relations are important for shape matching, and that our algorithm is relatively insensitive to small changes of β around 0.4. However, as any hierarchical approach, ours also seems to be sensitive to the right choice of the finest resolution at which the shape is represented. As mentioned above, different values of this input parameter may result in large variations of the number of nodes in the shape hierarchy, which, in turn, cause changes in computing the normalized total matching cost. If the right choice is selected separately for each class of MPEG-7, using validation data, then we obtain the retrieval rate of 88.3%. If this parameter is set to 1%, as stated above, for all classes, then our performance drops to 84.3%. This is comparable to the state of the art that achieves the rates of 85.40% in [6] and 87.70% in [3]. Table 1 summarizes our retrieval rates on the Brown dataset after first top 1 to 10 retrievals, for $\beta = 0.4$ and shape-resolution level fixed over all classes. Again, this retrieval improves if we select a suitable value for the resolution parameter for each class separately, using validation data.

Table 1. Retrieval results on the Brown dataset for $\beta = 0.4$

Approaches	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
[12]	99	99	99	98	98	97	96	95	93	82
[6]	99	99	99	98	98	97	97	98	94	79
[3]	99	99	99	99	99	99	99	97	93	86
Our method	99	99	98	98	98	97	96	94	93	82

6 Conclusion

Matching deformable shapes is difficult since they may be perceptually similar, but have certain parts very different or even missing. We have presented an approach aimed at robust matching of deformable shapes by identifying multiscale salient shape parts, and accounting for their intrinsic properties, and part-of and neighbor relationships. Experimental evaluation of the proposed hierarchical shape representation and efficient minimization via convexification of a quadratic matching cost has demonstrated that the approach robustly deals with large variations of or missing parts of perceptually similar shapes.

References

1. Biederman, I.: Recent psychophysical and neural research in shape recognition. In Osaka, N., Rentschler, I., Biederman, I., eds.: Object Recognition, Attention, and Action. Springer (2007) 71–88
2. Günther, O., Wong, E.: The arc tree: an approximation scheme to represent arbitrary curved shapes. Comput. Vision Graph. Image Process. **51**(3) (1990) 313–337
3. Felzenszwalb, P.F., Schwartz, J.D.: Hierarchical matching of deformable shapes. In: CVPR. (2007)
4. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planar curves. IEEE TPAMI **14**(8) (1992) 789–805
5. Ueda, N., Suzuki, S.: Learning visual models from shape contours using multiscale convex/concave structure matching. IEEE TPAMI **15**(4) (1993) 337–352
6. Ling, H., Jacobs, D.: Shape classification using the inner-distance. IEEE TPAMI **29**(2) (2007) 286–299
7. Fan, X., Qi, C., Liang, D., Huang, H.: Probabilistic contour extraction using hierarchical shape representation. In: ICCV. (2005) 302–308
8. Mcneill, G., Vijayakumar, S.: Hierarchical procrustes matching for shape retrieval. In: CVPR. (2006)
9. Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W.: Shock graphs and shape matching. Int. J. Comput. Vision **35**(1) (1999) 13–32
10. Shokoufandeh, A., Macrini, D., Dickinson, S., Siddiqi, K., Zucker, S.W.: Indexing hierarchical structures using graph spectra. IEEE TPAMI **27**(7) (2005) 1125–1140
11. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Rec. Letters **1**(4) (1983) 245–253
12. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Trans. Pattern Anal. Machine Intell. **26**(5) (2004) 550–571
13. Pelillo, M., Siddiqi, K., Zucker, S.W.: Matching hierarchical structures using association graphs. IEEE TPAMI **21**(11) (1999) 1105–1120
14. Tu, Z., Yuille, A.: Shape matching and recognition - using generative models and informative features. In: ECCV (3). (2004) 195–209
15. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE TPAMI **18**(4) (1996) 377–388
16. Jiang, H., Drew, M.S., Li, Z.N.: Matching by linear programming and successive convexification. IEEE TPAMI **29**(6) (2007) 959–975
17. Teh, C.H., Chin, R.T.: On the detection of dominant points on digital curves. IEEE Trans. Pattern Anal. Mach. Intell. **11**(8) (1989) 859–872
18. Torsello, A., Hancock, E.R.: Computing approximate tree edit distance using relaxation labeling. Pattern Recogn. Lett. **24**(8) (2003) 1089–1097
19. Pelillo, M., Siddiqi, K., Zucker, S.W.: Many-to-many matching of attributed trees using association graphs and game dynamics. In: Int. Workshop Visual Form, Springer LNCS. Volume 2059. (2001) 583–593
20. Demirci, M.F., Shokoufandeh, A., Keselman, Y., Bretzner, L., Dickinson, S.J.: Object recognition as many-to-many feature matching. Int. J. Computer Vision **69**(2) (2006) 203–222
21. Todorovic, S., Ahuja, N.: Region-based hierarchical image matching. Int. J. of Computer Vision **78**(1) (2008) 47–66