A Clustering-Based Ensemble Technique for Shape Decomposition

Sergej Lewin, Xiaoyi Jiang, and Achim Clausing

Department of Mathematics and Computer Science
University of Münster

Einsteinstrasse 62, 49149 Münster, Germany
{slewin,xjiang,achim.clausing}@uni-muenster.de

Abstract. Ensemble techniques have been very successful in pattern recognition. In this work we investigate ensemble solution for shape decomposition. A clustering-based approach is proposed to determine a final decomposition from an ensemble of input decompositions. A recently published performance evaluation framework consisting of a benchmark database with manual ground truth together with evaluation measures is used to demonstrate the benefit of the proposed ensemble technique.

1 Introduction

Ensemble techniques have been very successful in pattern recognition. In addition to the classification problem [1], the fundamental fusion approach has been introduced to many other domains including strings [2], graphs [3], clusterings [4], and segmentations [5]. In this work we investigate ensemble solution of shape decomposition.

The shape decomposition methods from the literature have their respective strengths and weaknesses, and their performance varies for different shapes. There exists no dominating approach that outperforms the others for all shapes. The variety of decomposition results is exemplarily demonstrated in Fig. 1(a) with overlaid five decompositions (produced by five different algorithms). Some cuts cross each other while there are cuts, which are almost congruent.

Combination (fusion) is a useful technique for taking advantages of the strengths and compensating the weaknesses of different approaches. In this particular case, a simple union of all cuts from different decompositions is obviously not meaningful. Instead, it suggests itself that some cuts should be removed and almost congruent cuts should be unified. In this paper we present an ensemble technique for combining shape decompositions. Although there exist quite a number of shape decompositions algorithms (see [6–8] and the references therein), multiple decomposition combination has not been studied before.

The remainder of this paper is organized as follows. Section 2 describes our decomposition ensemble technique. A performance evaluation framework from [9] applied in this work is summarized in Section 3, followed by the evaluation results in Section 4. Finally, some discussions conclude this paper.

Fig. 1. (a) Overlaid five decompositions; (b) representative cuts of clusters; (c) combined decomposition with threshold 3 (majority voting); (d) combined decomposition with threshold 2

2 Combined Shape Decomposition Approach

Our combination approach consists of two steps. First, the set of all cuts from the input decompositions is clustered into groups of similar cuts and each cluster is represented by one single cut by averaging all cuts within it. Then, a selection procedure is performed to determine the cuts of the result decomposition. An outline of the decomposition ensemble algorithm is given in Fig. 2. It is invariant to transformations like scaling and rotation (although the baseline decomposition algorithms may be effected by such transformations).

2.1 Cut Clustering

The set of all cuts from the input decompositions can be partitioned into groups of similar ones by clustering. In case of agglomerative hierarchical clustering, for instance, each cut builds its own cluster at the very beginning. Then, the clusters are merged until the distance between any two clusters is larger than a threshold t_1 . Depending on how the distance between clusters is computed, different types of hierarchical clustering can be distinguished. In our work the so-called unweighted pair group method with arithmetic mean is used. This distance between two clusters X and Y of size $|X|$ and $|Y|$, respectively, is defined by:

$$D(X, Y) = \frac{1}{|X| \cdot |Y|} \sum_{x \in X} \sum_{y \in Y} dist(x, y)$$

where $dist(x, y)$ is the distance between two cuts x and y , which can be computed based on distances between cut endpoints along shape boundary. The distance between two boundary points is defined by the minimal arc length of boundary curve between these points. Then, the distance between two cuts x (with endpoints a and b) and y (with endpoints c and d), respectively, can be simply defined as a sum of distances between corresponding endpoints:

$$dist(x, y) = \min\{pdist(a, c) + pdist(b, d), pdist(a, d) + pdist(b, c)\}$$

where $pdist$ is the distance along shape boundary.

Since after clustering each cluster contains similar cuts, we can determine one representative cut for a cluster by averaging all cuts in it. In order to guarantee

Input:	n decompositions of a shape;
Output:	combined decomposition;
Parameter:	thresholds t_1 and t_2 ;

Cluster the cuts of all n input decompositions (t_1 needed);
 Determine a representative cut for each cluster;
 Compute the weight for each representative cut;
 Select representative cuts based on the weights (t_2 needed);
 Resolve intersecting selected cuts;

Fig. 2. Outline of ensemble decomposition algorithm

that the endpoints of the average cut lie on the shape boundary, the averaging is performed based on boundary parametrization. Assuming that $\gamma : [0, 1] \rightarrow \mathbb{R}^2$ is a boundary curve. The parameter $p \in [0, 1]$ describes the position of curve point $\gamma(p)$ relative to start point $\gamma(0)$. The endpoints of the average cut are not computed by averaging the positions of left (right) endpoints of the cuts in a cluster geometrically, but in the parameter space. For averaging the left endpoints $x_1, x_2, \dots, x_k$ of k representative cuts, for instance, let's assume their corresponding parameter values be $p_1, p_2, \dots, p_k$, then the average endpoint is $\gamma(\frac{1}{k} \sum_{i=1}^k p_i)$. Fig. 1(b) shows the representative cuts of the clusters computed from the cuts in Fig. 1(a).

2.2 Selection of Cuts

After clustering and computing representative cuts, the cuts for the combined decomposition should be selected. This selection step should reject perceptually unreasonable cuts and resolve intersections of cuts. This can be performed based on weights derived from clusters. Each representative cut represents one cluster consisting of one or more cuts. Since all involved shape decomposition methods try to mimic human being's behavior, it can be assumed that most of the cuts created are perceptually reasonable. Therefore, the more methods create one cut, the larger is the likelihood that this cut is perceptually reasonable and should thus be kept. At the same time, it can be surely assumed that the likelihood of a perceptually unreasonable cut, which is coevally produced by two different methods, is relatively small. According to these considerations, the representative cut of a cluster can be weighted by the number of input shape decompositions, whose cuts are contained in the cluster. Note that by this definition the weight is usually smaller than the number of cuts in a cluster, since two cuts from a decomposition can be clustered into the same cluster.

Given the weight of representative cuts, the selection of cuts can be performed by using thresholding. As a first thresholding scheme, the *majority voting* can be realized. In this case the threshold t_2 is equal to the half of the number of input shape decompositions to be combined. All representative cuts, which

appear in at least 50% of shape decompositions, are selected. Alternatively, a smaller threshold t_2 can be used.

After the cut selection it can happen that some of the remaining cuts intersect each other. Such intersection can be resolved based on the same weights by simply removing the lower weighted cut of two crossing cuts.

The decomposition ensemble algorithm described above is summarized in Fig. 2. Fig. 1(c) and 1(d) show two combined decompositions created from the representative cuts in Fig. 1(b). The first one corresponds to the majority voting scheme, while the second one rejects all cuts created by one decomposition only. As expected, the majority voting rule results in a coarser shape decomposition and a smaller threshold retains more representative cuts for a finer shape decomposition. In this particular case, the second combined decomposition contains all important cuts. In general, however, the lower the threshold value is, the higher likelihood is that some perceptually unreasonable cuts would be selected for the combined decomposition.

Our ensemble approach introduces two extra thresholding parameters t_1 and t_2 . They can be optimized using training data, similarly as we have to deal with the parameters of the involved shape decomposition methods. This issue will be further discussed in Section 4.1.

3 Performance Evaluation Framework

In computer vision there exist today a lot of ground truth data and quantitative evaluations. However, shape decomposition is an exception. In this domain performance evaluation still remains qualitative. This is even true in recent publications (e.g. [10]). Typically, statements like "As we can see, ..., our method decomposes shapes into parts with high visual naturalness comparable to [x]" [10] are made¹.

Recently, a supervised quantitative performance evaluation framework was proposed [9], which consists of a benchmark database with manually specified ground truth (GT) and dissimilarity measures. A decomposition generated by an algorithm is compared with the GT decomposition of the same shape and quantitative dissimilarity measures are computed for performance comparison. We apply this framework to demonstrate the benefit of our ensemble technique. In the following its main components are briefly summarized.

3.1 Dissimilarity Measures

A dissimilarity measure is a distance function: $\gamma : D_S \times D_S \rightarrow \mathbb{R}$, where D_S denotes the set of all valid decompositions of a given shape S . It is reasonable to require that this distance function be semi-metric, i.e. it fulfils the following three conditions: $\forall x, y \in D_S$, 1) non-negativity: $\gamma(x, y) \geq 0$; 2) identity: $\gamma(x, y) = 0$ if and only if $x = y$; 3) symmetry: $\gamma(x, y) = \gamma(y, x)$. Several variants of distance

¹ Note that the quantitative comparison presented in [10] concerns the number of computed parts in the sense of data reduction, which is not related to the perceptual quality of decomposition.

measures have been discussed in [9]. In this work we apply the following two measures for experimental work.

Hamming Distance. A shape can be regarded as a non-rectangular image and a shape decomposition as a region-based segmentation accordingly. Thus, comparison of shape decompositions becomes that of comparing image segmentations. There exist a number of distance measures for this purpose [11]. One such measure is the Hamming distance (D_H). Considering two decompositions $A = \{A_1, \dots, A_n\}$ and $B = \{B_1, \dots, B_m\}$ of the same shape S , the directional Hamming distance between A and B is defined by: $D_H(A \rightarrow B) = \sum_i |A_i \setminus B_{k_i}|$, where B_{k_i} , $k_i = \arg \max_j |A_i \cap B_j|$, is the best fitting of A_i in B . The reversed distance $D_H(B \rightarrow A)$ can be similarly computed. Finally, the overall Hamming distance is given by:

$$D_H(A, B) = \frac{D_H(A \rightarrow B) + D_H(B \rightarrow A)}{2|S|}$$

Obviously, $H_D \in [0, 1]$ is semi-metric.

Jaccard Measure. If we interpret a shape as a set of pixels and a decomposition as a clustering, then the distance measures for comparing clusterings can be applied to quantify the difference between two shape decompositions. Several such measures have been proposed in the literature [11]. Particularly intuitive are those measures based on counting pairs. Given two decompositions A and B , we consider all pairs of pixels and count: a) n_{11} : #pairs of pixels, which are in the same part of A and in the same part of B ; b) n_{10} : #pairs of pixels, which are in the same part of A and in different parts of B ; c) n_{01} : #pairs of pixels, which are in different parts of B and in the same part of A ; d) n_{00} : #pairs of pixels, which are in different parts of both A and of B . Several distance measures for comparing clusterings are based on these four counts. For instance, the Jaccard measure is defined by:

$$J(A, B) = \frac{n_{10} + n_{01}}{n_{10} + n_{01} + n_{11}}$$

Obviously, $J \in [0, 1]$ is semi-metric.

3.2 Benchmark Database

To perceptually evaluate shape decomposition methods, i.e. how well they mimic decompositions created by human beings, we need a database with manually specified GT. In [9] it is proposed to use the benchmark database from a large-scale psychological study [12], which fits the requirement perfectly and is publicly available. It consists of human-generated decompositions of 88 different shapes. Each shape is decomposed by 38 human subjects on average. Fig. 3 (upper row) shows overlayed decompositions (represented by cuts) of five shapes. As can be seen, a high degree of decomposition consistency was achieved among human subjects. Nevertheless, there are a lot of outliers among the cuts. Some shapes, e.g. the fourth shape in Fig. 3, are decomposed less consistently.

Fig. 3. Top: human-generated decompositions. Bottom: corresponding majority-voted decompositions.

When using this database, the question arises how to handle the one-to-many situation (one decomposition D generated by an algorithm vs. multiple GT decompositions). The standard practice is to average the dissimilarity measures computed between D and each GT decomposition. An alternative solution is as follows. In the ideal case, the automatic decomposition should mimic as well as possible the shape decomposition performed by the majority of the involved human beings. Thus, it makes sense to determine a "majority-voted" decomposition from the decompositions created by human beings. The ensemble technique presented in this paper is a suitable tool for this task and thus will be used in our experimental testing. Fig. 3 (bottom row) shows majority-voted decompositions computed by our method.

4 Experimental Validation

For validation purpose we have performed a study using five shape decomposition methods and the quantitative performance evaluation framework.

4.1 Shape Decomposition Algorithms

In order to involve as many shape decomposition approaches as possible, we contacted several authors for their code. Unfortunately, only very few authors responded and finally, only one working implementation [6] (with source code) was provided. Thus, we implemented four other shape decomposition methods. In total, we have the following five algorithms for our study: 1) **ACD**: approximate convexity-based approach [6]; 2) **DCE**: evolution-based approach [13]; 3) **IFD**: flow-based approach [14]; 4) **MD**: morphology-based approach [7]; 5) **SD**: skeleton-based approach [8].

In order to treat all approaches equally, their parameters were individually adapted to the benchmark database. All five algorithms depend on a single parameter, which essentially specifies the fineness of shape decomposition. Its optimal value was trained by computing the average Hamming distances and the average Jaccard distances for different parameter values and selecting the best one. Taking **ACD** as an example, Table 1 shows the Hamming and Jaccard

Table 1. Distances for different values of the single parameter θ of **ACD**

θ	0.04	0.05	0.06	0.07	0.08
Hamming	0.142	0.132	0.128	0.132	0.133
Jaccard	0.350	0.329	0.323	0.330	0.332

distances for different values of its single parameter θ . In both cases an optimal parameter value $\theta = 0.06$ is found and used for performance evaluation.

Our ensemble approach introduces two extra thresholding parameters t_1 and t_2 . In our experiments the majority voting scheme for t_2 turns out to clearly outperform lower threshold values. For this reason this option is chosen for the experimental testing. For t_1 the same training procedure was done to find its optimal parameter value.

4.2 Validation Results

The first part of Table 2 shows the average dissimilarity measures over all shapes of the benchmark database for the five individual shape decomposition methods (sorted by their performance). For each shape the computed decomposition is either compared to all corresponding GT instances and the average value is reported (left half of the table). Alternatively, the "majority-voted" decomposition determined by our ensemble technique is used to compute the measures (right half of the table). It is important to mention that the performance listed in Table 2 and the resulting ranking of the involved shape decomposition algorithms must be treated with care. Only **ACD** was provided by the algorithm developer. The other four algorithms were implemented by ourselves and we may not reach the optimal performance, partly due to the missing details in the algorithm description. However, this fact is not harmful for our study because our goal is to demonstrate improved performance by means of ensemble solution.

The performance of combining four (five variants in total) and all five shape decomposition algorithms are listed in the second part of Table 2. All combinations outperform the individual baseline algorithms. In the former case the combination **ACD/IFD/MD/SD** dominates because the worst-performer **DCE** (in our implementation) is not involved. It is remarkable that the four test configurations (all GT instances vs. majority-voted, Hamming vs. Jaccard) produced virtually the same ranking list.

Table 2 reveals that a comparison against the "majority-voted" decomposition consistently leads to higher performance measures than against all corresponding GT instances. This fact may be credited to the majority voting rule used in the evaluation.

Fig. 4 shows the decompositions of four selected shapes. The results of the quantitative comparison coincide well with visual comparison of decompositions. It is also confirmed that the ensemble technique outperforms the baseline algorithms. This superiority of combination is attributed to the fact that they can compensate the absence of some important cuts. For example, the cut separating the front wheel of tricycle in Fig. 4 (fourth shape) is not created by **IFD** and **SD**, but contained in the combined decomposition due to the support of other

Table 2. Average dissimilarity measures over the benchmark database

	all GT instances		majority-voted	
	Hamming	Jaccard	Hamming	Jaccard
ACD	0.128	0.323	0.092	0.251
IFD	0.145	0.350	0.112	0.267
MD	0.151	0.371	0.126	0.328
SD	0.163	0.402	0.131	0.335
DCE	0.208	0.497	0.188	0.466
ACD/IFD/MD/SD	0.114	0.302	0.069	0.190
ACD/IFD/MD/DCE	0.117	0.305	0.074	0.201
ACD/IFD/SD/DCE	0.118	0.311	0.069	0.188
ACD/MD/SD/DCE	0.117	0.305	0.076	0.206
IFD/MD/SD/DCE	0.121	0.317	0.076	0.206
ACD/IFD/MD/SD/DCE	0.111	0.288	0.069	0.186

Fig. 4. Decompositions of four shapes generated by involved algorithms

decompositions. In addition, a lot of perceptually unreasonable cuts are rejected. For example, **DCE** produces a lot of unimportant cuts, which are not contained in the combined decomposition.

5 Conclusion

Although there exist quite a number of shape decompositions algorithms, multiple decomposition combination has not been studied before. In this paper we presented a clustering-based ensemble solution for shape decomposition. A recently published performance evaluation framework consisting of a benchmark database with manual ground truth together with evaluation measures was used to demonstrate the benefit of the proposed ensemble technique. We will make the source code for our decomposition ensemble method publicly available.

The proposed ensemble technique is useful in its right to improve the decomposition performance. In addition, it can also be adopted to solve the parameter

selection problem as done in [15, 16]. The parameter space is explored *without* the need of ground truth and combined towards high-quality results.

Acknowledgement. The authors thank J.-M. Lien for providing his program [6] for our tests.

References

1. Tulyakov, S., Jaeger, S., Govindaraju, V., Doermann, D.S.: Review of Classifier Combination Methods. In: Marinai, S., Fujisawa, H. (eds.) *Machine Learning in Document Analysis and Recognition*. SCI, vol. 90, pp. 361–386. Springer, Heidelberg (2008)
2. Jiang, X., Wentker, J., Ferrer, M.: Generalized median string computation by means of string embedding in vector spaces. *Pattern Recognition Letters* 33, 842–852 (2012)
3. Jiang, X., Munger, A., Bunke, H.: On median graphs: Properties, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 1144–1151 (2001)
4. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. *Int. Journal of Pattern Recognition and Artificial Intelligence* 25, 337–372 (2011)
5. Franek, L., Abdala, D.D., Vega-Pons, S., Jiang, X.: Image Segmentation Fusion Using General Ensemble Clustering Methods. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) *ACCV 2010, Part IV*. LNCS, vol. 6495, pp. 373–384. Springer, Heidelberg (2011)
6. Lien, J.M., Amato, N.M.: Approximate convex decomposition of polygons. *Comput. Geom. Theory Appl.* 35, 100–123 (2006)
7. Kim, D.H., Yun, I.D., Lee, S.U.: A new shape decomposition scheme for graph-based representation. *Pattern Recognition* 38, 673–689 (2005)
8. Zeng, J., Lakaemper, R., Yang, X., Li, X.: 2D Shape Decomposition Based on Combined Skeleton-Boundary Features. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) *ISVC 2008, Part II*. LNCS, vol. 5359, pp. 682–691. Springer, Heidelberg (2008)
9. Lewin, S., Jiang, X., Clausing, A.: Framework for quantitative performance evaluation of shape decomposition algorithms. In: *Proc. of ICPR* (2012)
10. Ren, Z., Yuan, J., Li, C., Liu, W.: Minimum near-convex decomposition for robust shape representation. In: *Proc. of ICCV*, pp. 303–310 (2011)
11. Jiang, X., Marti, C., Irniger, C., Bunke, H.: Distance measures for image segmentation evaluation. *EURASIP Journal on Adv. in Signal Processing*, 1–10 (2006)
12. DeWinter, J., Wagemans, J.: Segmentation of object outlines into parts: A large-scale integrative study. *Cognition* 99, 275–325 (2006)
13. Latecki, L.J., Lakamper, R.: Convexity rule for shape decomposition based on discrete contour evolution. *CVIU* 73, 441–454 (1999)
14. Dey, T., Giesen, J., Goswami, S.: Shape Segmentation and Matching with Flow Discretization. In: Dehne, F., Sack, J.-R., Smid, M. (eds.) *WADS 2003*. LNCS, vol. 2748, pp. 25–36. Springer, Heidelberg (2003)
15. Wattuya, P., Jiang, X.: A Class of Generalized Median Contour Problem with Exact Solution. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) *SSPR 2006 and SPR 2006*. LNCS, vol. 4109, pp. 109–117. Springer, Heidelberg (2006)
16. Wattuya, P., Jiang, X.: Ensemble Combination for Solving the Parameter Selection Problem in Image Segmentation. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) *S+SSPR 2008*. LNCS, vol. 5342, pp. 392–401. Springer, Heidelberg (2008)