

Weak Convex Decomposition by Lines-of-sight

Shmuel Asafi Avi Goren Daniel Cohen-Or

School of Computer Science,
Tel Aviv University

Abstract

We define the convexity rank of a set of points to be the portion of mutually visible pairs of points out of the total number of pairs. Based on this definition of weak convexity, we introduce a spectral method that decomposes a given shape into weakly convex regions. The decomposition is applied without explicitly measuring the convexity rank. The method merely amounts to a spectral clustering of a matrix representing the all-pairs line of sight. Our method can be directly applied on an oriented point cloud and does not require any topological information, nor explicit concavity or convexity measures. We demonstrate the efficiency of our algorithm on a large number of examples and compare them qualitatively with competitive approaches.

1. Introduction

High-level analysis of shapes has been receiving much attention recently. The general effort is to learn structural and semantic shape information from the low-level shape geometric properties (e.g., [FCODS08] [MY^Y*10] [WXL*11]). A fundamental problem in shape analysis is shape segmentation [Sha08]. Although the problem received a lot of attention recently (e.g., [GF08] [HKG11] [KHS10] [ZZWC12]) the problem remains challenging. Moreover, most of these methods assume complete and watertight shapes, represented by a triangular mesh.

A closely related problem to shape segmentation is the problem of convex decomposition. The two problems are related by the "minima rule" that states that humans perceive parts as being separated by lines of minimum curvature [HR84], which implies that a shape can be seen as being composed of approximately convex parts. This observation motivated the problem of approximate convex decomposition of shapes as means for meaningful segmentation of other related problems [KJS07] [LA07] [AMSF08] [RYLL11].

In this work we introduce a conceptually simple means for decomposing a shape into approximately convex parts. We define a set of points to be in a weakly convex position if most of its subsets are in a convex position, and define the convexity rank of a set of points to be the portion of mutually visible pairs of points out of the total number of pairs. Based on this definition, we compute the binary visibility of

all pairs of points and apply spectral clustering to yield the weakly convex decomposition. Our method avoids any complex geometric operations that measure concavity/convexity of regions or parts. It is directly drawn from the fundamental definition of convexity which implies that a region is convex if all pairs of points in that region have an inner line-of-sight. Consequently, loosely speaking, we seek a decomposition into regions which maximizes the convexity rank of its regions, without explicitly measuring convexities or concavities by complex geometric algorithms. Aside for its simplicity, a unique advantage of our method is that it can be applied on an incomplete shape, with large missing parts, possibly represented as a point cloud (see Figure 1).

2. Related Work

The convex decomposition of a shape can be useful as means of breaking a hard problem into a set of simplified problems; a complex geometry into an assembly of simple parts. However, an exact convex decomposition is NP-hard [?] and overly strict and conservative to be useful for common shapes. Approximate convex decomposition [LA07] [AMSF08] [RYLL11] is more effective since common shapes consist of an assembly of parts which are not strictly convex. Clearly, approximate convex decomposition is easier to compute than an exact one.

Yet, an approximate convex decomposition is not an easy task, and often hard to control. Kraevoy and Sheffer [KJS07] use a greedy region growing approach, where the distance

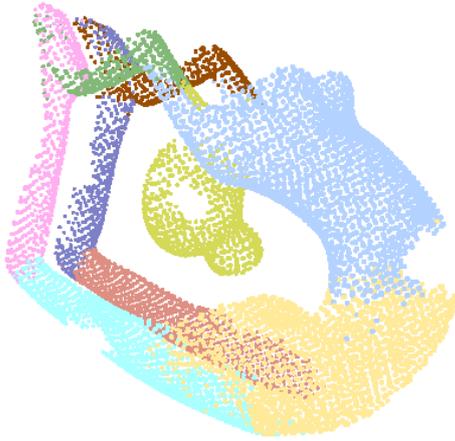


Figure 1: An approximate convex decomposition applied on an incomplete point cloud.

between the growing surface part and its convex hull serves as the stopping criterion. The approach taken by Lien and Amato [LA07] is based on iteratively identifying concave regions which guide the partitioning of the object into finer parts. Through the elimination of these concavities the resulting parts are more convex. These methods are hard to control and often generate redundant parts. Ren et al. [RYLL11] improve the decomposition by an optimization process that minimizes the number of parts.

A different strategy for an approximate convex decomposition is introduced by Attene et al. [AMSF08]. They assume that the shape has been decomposed into a tetrahedral mesh, where each tetrahedron is convex by definition. In a bottom up scheme they aggregate tetrahedra into a hierarchy of approximate convex polyhedra. The strength of their approach stems from the fact that unlike the above methods which consider the surface only, here they treat the convexity problem by considering the volume. The method that we present here considers the volume for computing the convex decomposition, however its complexity is defined by the number of surface point pairs for which we compute mutual visibility.

All methods for convex decomposition consider as input a complete watertight mesh. This limits their applicability to analyze shapes that were already consolidated, reconstructed or properly modeled. There is, of course, an interest and need in methods that analyze imperfect shapes. For example, in [TZCO09] the medial skeleton of a shape is computed from a rather incomplete point cloud. Lipman et al. [LCDF10] analyze the symmetry of a shape, possibly incomplete, represented by a point cloud. Analyzing point clouds requires using techniques which do not rely on common surface-based tools like geodesic paths or the Laplace operator. Our method is based solely on the visibility of

pairs of points, without considering any notion related to surfaces, thus allowing the convex analysis of incomplete point clouds.

Other work on shape segmentation, rather than convex decomposition, also have strong relation to our own work. Liu and Zhang [?] present a segmentation method based on spectral clustering as is our method. However, their affinity matrix is based on local surface properties, where ours is based on visibility, which is a global volumetric property. Shapira et al. [?] and Liu et al. [?] both use visibility to segment a shape, but they do so by reducing it to a function of the surface, rather than using it directly for clustering of the volume.

3. Weak Convexity

The convex decomposition method that we present is built upon a new definition of weak convexity which is based on a computation of inner visibility between points on the surface of the shape. Two points on the surface of a shape are mutually visible and are said to be in a line-of-sight (LoS), if the straight line-segment between them does not leave the inner volume of the shape. A point is always said to be in LoS with itself. Two surface points or any clique of points that are in LoS are also said to be in a convex position. Note that the visibility is not computed for the continuous surface of the shape, but rather on a uniformly-sampled finite set of surface points. We denote the set of sampled surface points by S ; we denote the LoS set of mutually visible pairs by $LoS(S)$.

We define $LoS(A)$ as the set of all pairs of points in $A \subseteq S$ that are in a line-of-sight; that is, $(i, j) \in LoS(A)$ if and only if the surface points $i \in A$ and $j \in A$ are mutually visible. For two subsets of surface points $A, B \subseteq S$ we denote $LoS(A, B)$ to be the set of pairs $(i, j), i \in A, j \in B$ which are mutually visible. Notice that this definition does not imply any assumptions about the shape; the shape might be incomplete, not watertight, noisy, and may actually be consisted of several geometrically separated shapes. However, to compute the LoS, we assume that the surface orientation (the in/out direction) is known at each point (more details in Section 6).

Figure 2 illustrates the concept of line-of-sight. It is clear how lines of sight relate to the property of convexity. As can be seen, the points that are in LoS with any specific point must not all be on a continuous portion of the surface. Note that our definition of weakly convex regions is identical for both 2D and 3D (see Figure 3).

Our method searches for clusters of surface points that are mutually visible, meaning that they are in a convex position with respect to each other. When the points in such a cluster are also geometrically connected (forming a continuous surface), they define the convex envelope of a partial volume of the shape. We call a cluster in which all pairs of points are in a convex position, a perfectly convex cluster.

Decomposing a shape into clusters of perfectly mutually

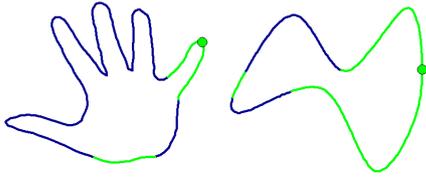


Figure 2: The regions in green have a line-of-sight with the marked point, while the blue regions do not.



Figure 3: Our method handles 2D and 3D objects under a unified framework. Both 2D and 3D shapes are clustered automatically into six clusters: five fingers and palm.

visible points is directly related to the clique cover problem, which is NP-hard. Also, a decomposition to perfectly convex clusters results (for most non trivial shapes) in a very large number of clusters, where each cluster has no clear meaning to the human perception. We aim to decompose shapes into few weakly (*approximately*) convex segments where each segment is more likely to be a meaningful and coherent part of the shape. To accomplish this, we define *the convexity rank* of a set of points to be the portion of mutually visible pairs of points out of the total number of pairs:

$$CR(A) = \frac{|LoS(A)|}{\binom{|A|}{2}}, \quad (1)$$

where A is a subset of $|A|$ points from S , and $|LoS(A)|$ is the number of mutually visible pairs in A .

Note that our weak convexity measure equally holds for 2D and 3D. Furthermore, its definition is fairly simple in contrast to alternative measures (e.g., [Kar10]), and it avoids computing convex hulls to estimate convexity (more in Section 6). Moreover, in convex hull based methods, the computation of the convex hull itself is performed for each decomposition solution examined; while visibility is calculated only once for the entire shape. Figure 4 demonstrates convexity rank associated with various shapes. It can be noticed that the convexity rank is intuitive, and can easily be interpreted visually.

4. Spectral Convex Decomposition

The problem of convex decomposition can now be formulated using the definition of weak convexity rank. For a given

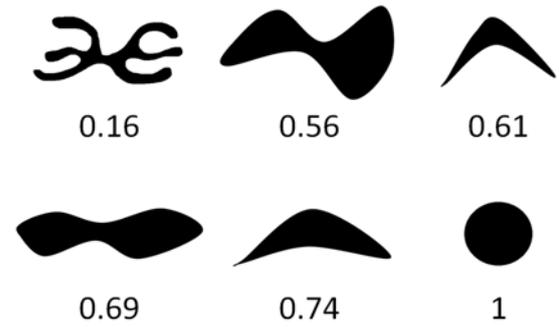


Figure 4: The convexity rank associated with various shapes.

number of clusters k , we wish to find a partitioning of the data for which each cluster has a high convexity rank and the visibility between clusters is low. In other words, we aim at a high intra-cluster number of LoS pairs and a low inter-cluster number of such pairs. This can be formulated by a minimization of the normalized cut (Ncut) functional:

$$S_1, \dots, S_k = \arg \min \sum_{i=1}^k \frac{|LoS(S_i, \bar{S}_i)|}{|LoS(S_i, S)|}, \quad (2)$$

where \bar{S}_i is the complement of S_i and S is the entire shape. The Ncut problem is NP-complete and a known approximation is the solution of the spectral clustering algorithm [SM00]. Figure 5 illustrates the convex decomposition of two shapes. The convexity rank of the original shapes is low, but our decomposition process successfully decomposes the shapes into approximately convex parts, which are also semantically meaningful. Note that the LoS relation is relatively low between different clusters.

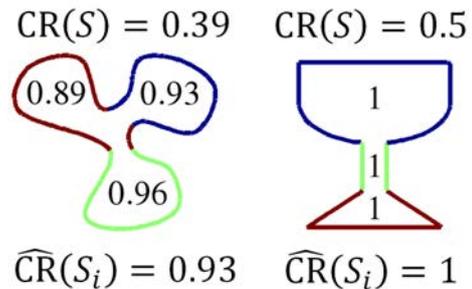


Figure 5: Two shapes with a low convexity rank (denoted by $CR(S)$) are decomposed into three regions with higher convexity ranks. The average of the ranks are denoted by $\widehat{CR}(S_i)$. Note that the shape on the right is decomposed into three strictly convex parts, while the shape on the left into three weakly convex regions.

The spectral clustering algorithm takes as input an affin-

ity matrix $A_{n \times n}$, that describes the similarity relations between n objects, and a parameter k ; it outputs a clustering of the objects into k clusters. Each entry in the affinity matrix specifies how similar two objects are to each other; when $A_{i,j} = 0$ the objects i and j are considered not similar at all, and higher values of $A_{i,j}$ indicate greater similarity.

The algorithm then requires normalizing the affinity matrix. Two methods of normalization are common: Laplacian and stochastic. In the Laplacian formulation the normalization is:

$$W = I - \Delta^{-1/2} A \Delta^{-1/2},$$

In the stochastic formulation each row is normalized to sum to 1:

$$W = \Delta^{-1} A,$$

where Δ is the diagonal degree matrix in both cases.

Determining the clustering is then achieved in two steps. First, eigen-analysis of the normalized affinity matrix W is performed and each object is assigned coordinates in a spectral space created using the eigenvectors. Then, a spatial clustering algorithm, such as K-means, is applied in the spectral space resulting in a k -way clustering of the objects.

The rationale behind this process is that groups of objects that are all mutually similar will be located next to each other in the spectral space. [MS01] have shown the relation of spectral clustering to a random walk process on a graph.

In our method we define a binary affinity matrix M as follows:

$$M_{i,j} = \begin{cases} 1 & (i,j) \in LoS(S) \\ 0 & otherwise \end{cases} \quad (3)$$

Hence we refer to M as the LoS matrix, and it acts as an affinity matrix in the sense that surface points that are visible to each other are likely to belong to the same convex segment, and can thus be interpreted as similar. Furthermore, according to [NLCK05], distances in the embedded space highly correspond to mean exit time between clusters; in our setting, a cluster is an approximately convex segment, and an exit from that segment occurs when the random walk process hits a line-of-sight that leads to a surface point outside that segment. Figure 6 illustrates the interpretation of lines-of-sight as a random walk within convex segments.

Figure 7 illustrates the LoS matrix, the spectral space and the clustering result. We do not pose any guarantees on the approximation to the optimal solution, but our empirical results are visually satisfactory and have high average convex ranks.

Note, that the spectral clustering does not guarantee that the resulting segments are continuous; this is perfectly reasonable since our definition and measure of weak convex-

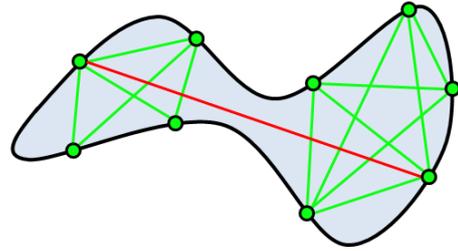
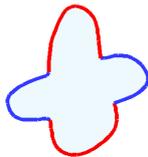


Figure 6: Most lines-of-sight are within a convex segment (green lines). Lines-of-sight that lead to another segment, also called an exit, are less common (red line). Therefore, a random walk along lines-of-sight is more likely to remain within a convex segment.

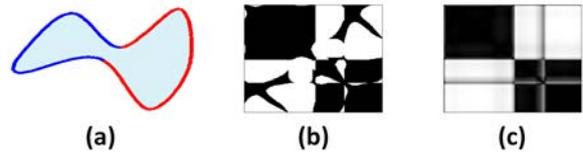


Figure 7: (a) A shape of 200 surface points decomposed to two convex segments. (b) The LoS Matrix sorted by clusters (black = visible LoS). (c) The distance matrix of points in the spectral space (black = short distance); illustrating that the distance between points in the spectral space highly correspond to the likelihood of belonging to the same convex segment. Notice that the concave bottleneck areas correspond to the “noisy” points in the LoS and distance matrices.

ity does not incorporate any information about continuous connectivity, but only about lines-of-sight which are not limited to the surface at all.

5. Implementation

A notable feature of our method is that it does not have strong requirements from the input shape. All that is required is a scheme for testing visibility between a pair of points. If the shape is represented by a triangular mesh then the LoS is no more than a common ray-surface intersection test. To accelerate this ray casting test we employ a voxel grid to subdivide the space. If the shape is represented by a point cloud we use splats to represent the points. Our splats are realized by triangles centered on the original points with an orientation that agrees with the normal direction associated with these points (see Figure 8). The size of a splat is approximated by the local density of the point cloud. However, the cloud of splats does not form a coherent surface and the ray-shape intersection test yields a noisy all-pair matrix M containing both false-positive and false-negative entries. Nevertheless, when the level of noise is moderate, the spectral

analysis is robust enough to reduce the sensitivity to noise and yield reasonable results (see Figure 9(a)). The density and uniformity of faces or splats is also important. Empirically, shapes with over 2000 triangles or points prove to be sufficiently dense and uniform for the success of our method. Simplified meshes with under 2000 triangles are a priori up-sampled uniformly to 3000 triangles.

The spectral convex decomposition method is also oblivious to the completeness of the shape and it can handle shapes with large missing parts (see Figure 9(b,c)). The robustness to missing parts is a direct implication of the fact that our definition of weak convexity considers a set of points and not a watertight surface or a continuous surface in general.

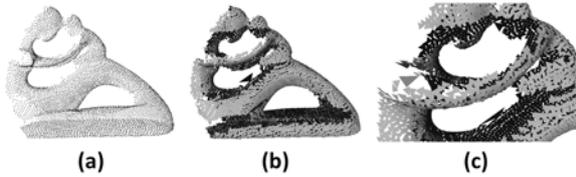


Figure 8: (a) Point cloud with approximate surface orientation. (b) Splat cloud constructed from the point cloud. (c) Zooming into a portion of the splat cloud. Notice that the splat cloud has gaps where the density of the point cloud is low or completely missing.

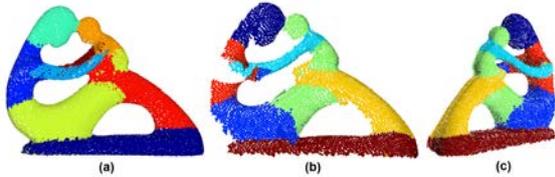


Figure 9: (a) Full splatted point decomposition. (b) Partial point cloud splatted and decomposed. (c) Same decomposed partial model from acquisition point of view.

Computation of all the entries of an LoS matrix for a large model is extremely time consuming and thus impractical, with a complexity of $O(N^2 \log N)$, where N is the number of faces or splats. Therefore, instead of calculating and using a full LoS matrix, we use a sparsely sub-sampled LoS. The sub-sampled LoS is composed of two matrices, a positive-LoS matrix containing entries for pairs of faces that are in light-of-sight; and a negative-LoS matrix containing entries for pairs of faces that are not in line-of-sight. Notice that the positive and negative matrices do not complement each other, because each contain only a small subset of all possible pairs. The combination of positive and negative LoS matrices provides a quick estimate of the convexity measure from a sparse set of pairs in line-of-sight and not in line-of-sight pairs.

We sub-sample the LoS by checking only a small number

(r) of lines-of-sight per each face/splat. The lines-of-sight are sampled by shooting a fixed array of r rays from the center of each face/splat, oriented around the surface normal. The rays are arranged uniformly inside a cone with an opening angle of 60 degrees. This method is very similar to the one used by Shapira et. al in [?] to calculate the Shape Diameter Function. Each ray hits one opposite face (unless there's a hole in the shape in that direction), contributing two (symmetric) entries to the positive-LoS matrix. The faces that lay behind and are occluded by the face that was hit, are not in line-of-sight with the source ray; thus contribute entries to the negative-LoS matrix. In our experiments we use $r = 30$ rays.

According to [?], the running-time of this procedure might be as low as a few seconds for a shape of 30,000 face with $r = 30$, when computed with GPU acceleration. In our implementation, we use only Matlab operations, thus running time is at least one order longer. The sparsity of the LoS matrix also speeds up spectral clustering calculation, and allows scalability. Calculation of the sparse LoS matrices takes about 200 seconds for a 10,000 face shape with $r = 30$, and spectral clustering about 50 seconds, on a 2.0Ghz computer with 2GB RAM.

The fixed value of parameter r was determined by exploring results for different values on a small set of shapes. We experimented with values in the range of [10,200]; since higher values would yield unreasonable running time. Results are quite “unstable” for $r = 10$ and become stable for values greater than about 20. We have chosen $r = 30$ as a compromise between high stability and low running time. We have also compared a few results against those obtained with a full LoS matrix (which takes hours to compute for relatively simple shapes); but did not notice significant artefacts introduced by our approximation.

Finally, to produce a visually aesthetic result on mesh, the boundaries between parts should be smooth and seam lines are encouraged to run along concave regions. We use a conventional graph cut method applied to the KNN graph of the point set that minimizes an energy functional with two cost terms:

$$E(l) = \sum_x \text{Data}(x, l(x)) + \lambda \sum_{\substack{l(x) \neq l(y) \\ (x,y) \in E}} \text{Smooth}(x, y) \quad (4)$$

The data terms aims at a faithful labeling of the graph vertices and the smoothness term strives at employing certain conditions on edges dividing different labels. In our setting the data term is built upon a Gaussian kernel of the Euclidean distance from points to cluster centers in the spectral embedded space: where x is a point in the embedding, l is the assigned label for x , \hat{x}_l is the center of the cluster associated with label l and σ_l^2 is the intra cluster variance. The smoothness term rewards boundaries along edges with negative dihedral angles.

As with any clustering method, automatically determining

the number of weakly convex parts in the decomposition is a difficult task. To control the number of weakly convex parts, we simply exhaustively search for the best decomposition into 2 to 13 parts.

We evaluate the quality of a decomposition of shape S into S_1, \dots, S_k by:

$$\psi(S_1, \dots, S_k) = \frac{1}{|S|^2} \sum_i (|LoS(S_i)| + \alpha |\overline{LoS}(S_i, \overline{S}_i)|), \quad (5)$$

where $|LoS(S_i)|$ is the count of intra visible LoS pairs in S_i ; $|\overline{LoS}(S_i, \overline{S}_i)|$ is the count of inter occluded pairs, where one point is in S_i and the other in another part; α controls the intra/inter factor. In all our experiment we used $\alpha = 1$.

The ψ function in Equation 5 is maximized by increasing visibility within a part S_i and decreasing visibility between different parts. The weight α allows controlling the balance between cluster intra visibility and cluster inter lack of visibility. Measuring the partition with ψ avoids rewarding for over-segmentation. On the other hand, parts with high ψ are further partitioned.

6. Evaluation and Discussion

Quantitative Evaluation. We tested our method on various objects as shown in Figures 10 and 15. An approximate convex decomposition does not have ground truth to serve as means to evaluate the method with a quantitative measure, nor benchmarks like those used for evaluating segmentation techniques. We do not present the reduction of the convexity ranks since such reduction is trivial and any naive decompositions also leads to a reduction. Instead, we evaluate the decomposition using the ψ function defined in 5.

As we show, our method succeeds in decomposing the shapes into parts with a high convexity rank and low visibility between parts. See Figure 10 where we display under each shape the ψ value of the input shape as a single part (which is equivalent to the shape's convexity rank), and the ψ value of the resulting decomposition.

Qualitative Comparison. To qualitatively evaluate the performance of our method, we compare it to other approaches. Alternative methods produce results with different qualities since they use different measures of convexity. A common convexity measure used in other methods (e.g., [AMSF08][†]) is:

$$Convexity(S) = \frac{vol(S)}{vol(CH(S))},$$

where $vol()$ measures the volume of a 3D part, or the area of a 2D part, and $CH(S)$ is the convex hull of S . We argue that

[†] [AMSF08] actually use a scaled version of this measure, however the discussion still holds

this measure is less intuitive and less sensitive to concavities. An illustrative example is shown in Figure 12. As can be seen, using the above measure, the spoon is about as convex as the fork, which is counter intuitive. Our convexity rank, on the other hand, provides a measure indicating that the spoon is significantly more convex than the fork. These examples illustrate the strength of using a fine-grained measure, like visibility to rank the convexity.

Lien and Amato [LA06] and Ren et al. [RYLL11] made a similar observation and suggested a different concavity measure. They measured the maximal outer route from a surface point to the convex hull of the shape. This measure is local and does not take a global view as shown in Figure 13. Our measure is global and intuitive. To stress this point, we display in Figure 10 a decomposition of a vase, where its non-convex body is split into two more convex parts. Note that the split does not occur at a sharp feature, nor at a particularly concave region.

Furthermore, a notable property of our method is demonstrated in Figure 11. Previous approximate convex decomposition methods attempt to find a disjoint partitioning of the inner volumes of the shapes. Our method attempts to find a disjoint decomposition of the shape's surface, which often provides a more intuitive decomposition.

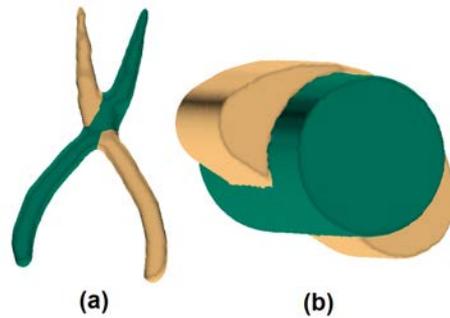


Figure 11: Unlike other methods, our method can decompose a shape into parts that are overlapping in volume, but with disjoint surfaces.

Limitations. As shown in Figure 13(c), our convexity measure, which is global, is not sensitive to sharp local and small concavities in a large convex context. For example, fingers of a human; they typically will not be segmented. Additionally, our method does not handle containers well. The inherent lack of inner visibility makes it hard to perform a meaningful convex decomposition. The container parallel sheets are treated independently by the graph-cut, which leads to incompatibility across the parallel surfaces (see Figure 14).

Conclusions. In summary, we have presented a spectral method for decomposing a shape into weakly convex parts. The decomposition has a simple formulation as it is merely a spectral clustering of the all-pairs (inner) visibility. The

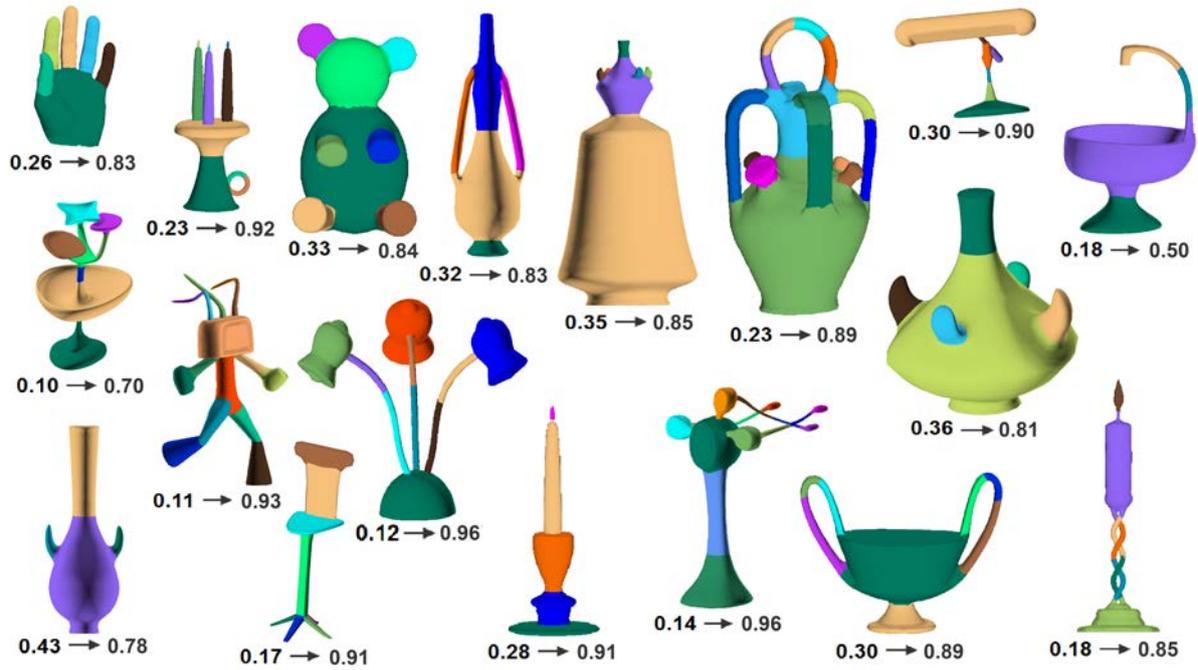


Figure 10: For each shape we display its ψ before and after decomposition. Notice the lower left shape where our method splits the non-convex body of the vase into two weakly convex regions. Note that the split occurs at a featureless point, and not at a sharp feature or a particularly concave region.

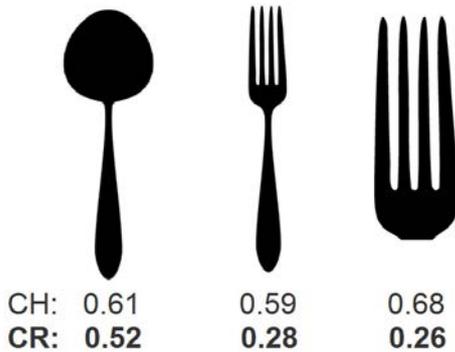


Figure 12: Convexity Measures. In the top row is the shape vs. convex hull area ratio. On the bottom is our convexity rank.

clusters then represent regions with high mutual visibility, or in other words, weakly convex parts. This approach eludes complex geometric algorithms by using efficient spectral clustering tools.

The simplicity of the formulation allows good control over the decomposition where the convexity rank of the parts can be refined. Furthermore, the all-pair visibility does not require strong assumptions about the input shape and the

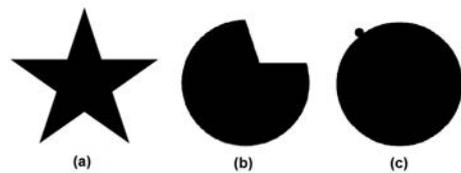


Figure 13: In both (a) and (b) the maximal route from a surface point to the convex hull of the shape is the same, but shape (a) seems more concave than shape (b). In shape (c) however, our measure gives a high convexity value while the alternative measure relates to the obvious sharp concavity better.

spectral convex decomposition method can handle shapes that are incomplete and possibly represented as a point cloud.

In the future we would like, as an immediate application of a weakly convex decomposition, to approximate a shape with a set of bounding boxes. Enclosing each weakly convex part with a minimal bounding box will lead to a tight and yet conservative bounding volume. We would also like to study the gap between shape segmentation and convex decomposition. We believe that the latter can form a good initial decomposition to improve segmentation methods, espe-

cially for manmade objects. We are also interested in studying the problem of concave parts, where instead of analyzing the inner visibility of the part, we consider the complement, and detect convex regions in the complement.

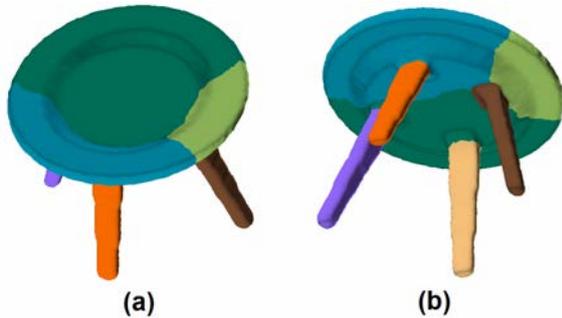


Figure 14: Handling Sheets of parallel surfaces. In some cases the method may produce unappealing results as shown in (a) and (b). There is no compatibility between the upper and lower side of the table.

References

- [AMSF08] ATTENE M., MORTARA M., SPAGNUOLO M., FALCIDIENO B.: Hierarchical convex approximation of 3d shapes for fast region selection. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1323–1332.
- [FCODS08] FU H., COHEN-OR D., DROR G., SHEFFER A.: Upright orientation of man-made objects. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 42:1–42:7. URL: <http://doi.acm.org/10.1145/1399504.1360641>, doi:10.1145/1399504.1360641.
- [GF08] GOLOVINSKIY A., FUNKHOUSER T.: Randomized cuts for 3d mesh analysis. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 145.
- [HKG11] HUANG Q., KOLTUN V., GUIBAS L.: Joint shape segmentation with linear programming. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 125.
- [HR84] HOFFMAN D., RICHARDS W.: Parts of recognition. *Cognition*, 18 (1984), 65–96.
- [Kar10] KARASEV R.: A measure of non-convexity in the plane and the minkowski sum. *Discrete & Computational Geometry* 44, 3 (2010), 608–621.
- [KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 102.
- [KJS07] KREAVOY V., JULIUS D., SHEFFER A.: Model composition from interchangeable components. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on* (2007), IEEE, pp. 129–138.
- [LA06] LIEN J., AMATO N.: Approximate convex decomposition of polygons. *Computational Geometry* 35, 1 (2006), 100–123.
- [LA07] LIEN J., AMATO N.: Approximate convex decomposition of polyhedra. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling* (2007), ACM, pp. 121–131.
- [LCDF10] LIPMAN Y., CHEN X., DAUBECHIES I., FUNKHOUSER T.: Symmetry factored embedding and distance. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 103.
- [MS01] MEILA M., SHI J.: A random walks view of spectral segmentation.
- [MYY*10] MITRA N., YANG Y., YAN D., LI W., AGRAWALA M.: Illustrating how mechanical assemblies work. *ACM Transactions on Graphics-TOG* 29, 4 (2010), 58.
- [NLCK05] NADLER B., LAFON S., COIFMAN R., KEVREKIDIS I.: Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. *Arxiv preprint math/0506090* (2005).
- [RYLL11] REN Z., YUAN J., LI C., LIU W.: Minimum near-convex decomposition for robust shape representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 303–310.
- [Sha08] SHAMIR A.: A survey on mesh segmentation techniques. In *Computer graphics forum* (2008), vol. 27, Wiley Online Library, pp. 1539–1556.
- [SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, 8 (2000), 888–905.
- [TZCO09] TAGLIASACCHI A., ZHANG H., COHEN-OR D.: Curve skeleton extraction from incomplete point cloud. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 71.
- [WXL*11] WANG Y., XU K., LI J., ZHANG H., SHAMIR A., LIU L., CHENG Z., XIONG Y.: Symmetry hierarchy of man-made objects. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 287–296.
- [ZZWC12] ZHANG J., ZHENG J., WU C., CAI J.: Variational mesh decomposition. *ACM Transactions on Graphics (TOG)* 31, 3 (2012), 21.

