

# SHAPE FROM POINT FEATURES

Steve Gu, Ying Zheng and Carlo Tomasi

Email: { steve, yuanqi, tomasi }@cs.duke.edu  
Department of Computer Science  
Duke University, Durham, NC USA 27708

## ABSTRACT

We present a nonparametric and efficient method for shape localization that improves on the traditional sub-window search in capturing the fine geometry of an object from a small number of feature points. Our method implies that the discrete set of features capture more appearance and shape information than is commonly exploited. We use the  $\alpha$ -complex by Edelsbrunner *et al.* to build a filtration of simplicial complexes from a user-provided set of features. The optimal value of  $\alpha$  is determined automatically by a search for the densest complex connected component, resulting in a parameter-free algorithm. Given  $K$  features, localization occurs in  $O(K \log K)$  time. For VGA-resolution images, computation takes typically less than 10 milliseconds. We use our method for interactive object cut, with promising results.

*Index Terms*—alpha shapes, object segmentation

## 1. INTRODUCTION

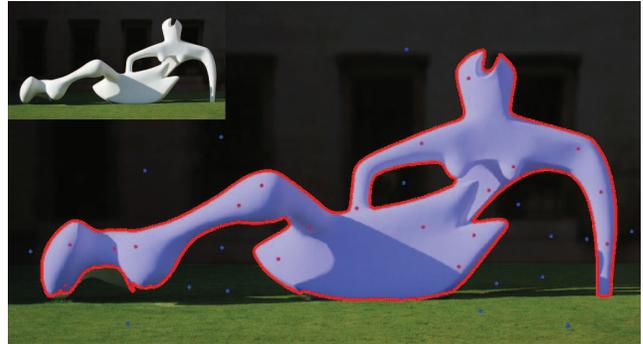
We present a method to find the detailed shape of an object of interest from a small number of feature points. These can come from user input – as in our experiments – or from the output of a trained classifier. In this paper, we assume each feature comes with its 2D location and a *score* that measures how likely it belongs to some object of interest.

### 1.1. Literature Review

Bag-of-feature models [1, 2, 3, 4] have been quite successful in tasks ranging from image matching [5] and object tracking [6, 7] to object recognition [1, 3].

Each feature receives a *score* – positive or negative – of the likelihood that a given feature belongs to the object of interest. Crucially, high positive feature scores tend to occur within an object, so the region that maximizes the sum of scores tallied in its interior is likely to correspond closely to the location of the object.

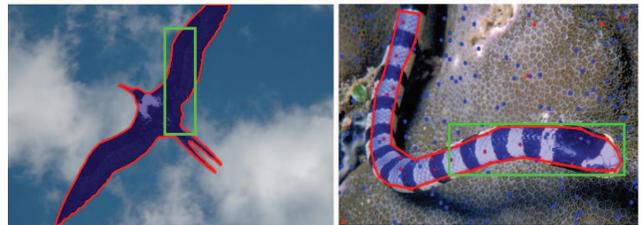
Typical methods [8, 9, 7] employ an axis-aligned rectangular window to cover the region of interest. However, rectangular windows only localize objects coarsely and may not work well for non-rectangular objects (Figure 2).



**Fig. 1.** The shape of a Henry Moore sculpture automatically localized from the red and blue dots in less than 0.01s, and with no parameters. Image from the Wikimedia Commons.

### 1.2. Our Contributions

In contrast, we search over a one-parameter family of shapes derived from the so-called  $\alpha$ -complex [10]. On one hand, this family allows general shapes, with possibly even holes. On the other hand, 2D alpha complexes can be enumerated quickly in  $O(K \log K)$  time for  $K$  points. This winning combination suggests that point features – together with their scores – encode more geometric information than is customarily exploited in the literature.



**Fig. 2.** Subwindow search (green rectangles) cannot localize shape in detail. Our method does better (red regions) given features are well distributed (red and blue dots on the right).

### 1.3. Paper Organization

Section 2 reviews the construct of  $\alpha$ -complexes. Section 3 presents an optimization procedure in the space of  $\alpha$ -complexes. Section 4 describes a simple, efficient, and easy to implement algorithm for shape from point features. Section 5 shows experiments, and Section 6 concludes.

## 2. ALPHA COMPONENTS

Let  $\Gamma \subset \mathbb{R}^2$  be a set of discrete, isolated points on the plane. The Delaunay triangulation  $\text{Del}(\Gamma)$  of  $\Gamma$  is a *simplicial complex* whose boundary is the convex hull of  $\Gamma$ . What makes  $\text{Del}(\Gamma)$  a simplicial complex is that it is composed of simplices – points, edges and triangles – that are glued together appropriately: If an edge is in  $\text{Del}(\Gamma)$ , so are its endpoints; if a triangle is in  $\text{Del}(\Gamma)$ , so are its edges. Conversely, if two triangles in the complex share an edge, that edge is in the complex, and if two edges in the complex meet at an endpoint, that endpoint is in the complex.

The  $\alpha$ -complex  $A_\alpha(\Gamma)$  is a one-parameter simplicial complex that lies between the discrete point set itself (which is a simplicial complex!) and the Delaunay triangulation:

$$\Gamma \subseteq A_\alpha(\Gamma) \subseteq \text{Del}(\Gamma)$$

That is, by varying  $\alpha$  from 0 to  $+\infty$ , the  $\alpha$ -complex  $A_\alpha(\Gamma)$  grows from a discrete point set to its Delaunay triangulation. In this process, the  $\alpha$ -complex  $A_\alpha(\Gamma)$  can take on fairly general shapes that can be convex, concave, or even with holes. Specifically, given a fixed  $\alpha$ ,  $A_\alpha(\Gamma)$  is constructed by retaining all the points, plus the Delaunay edges of length up to  $\alpha$ . If the three edges of a triangle are in  $A_\alpha(\Gamma)$ , we also add the triangle to the complex.

Let  $B_\alpha(x) = \{y \mid \|x - y\| \leq \alpha\}$  be a disk of radius  $\alpha$  centered at  $x$ . Then, it is known that:

$$\bigcup_{x \in \Gamma} B_\alpha(x)$$

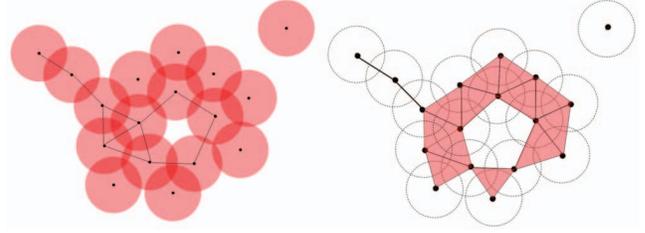
is homotopy-equivalent to  $A_\alpha(\Gamma)$ , meaning that the union of disks can be transformed to the  $\alpha$ -complex by a deformation that preserves the topology. See Figure 3 for an illustration.

The family of  $\alpha$ -complexes forms a nested set sequence called mathematical *filtration*. That is, for  $0 < \alpha_1 < \alpha_2 < \dots < \alpha_n < +\infty$ , we have:

$$A_{\alpha_1}(\Gamma) \subseteq A_{\alpha_2}(\Gamma) \subseteq \dots \subseteq A_{\alpha_n}(\Gamma) \quad (1)$$

This nesting leads to a way to enumerate all the elements of the entire complex quickly, as we will see in Section 4.

An  $\alpha$ -complex is not necessarily connected (See Figure 3). In our application, shapes are assumed to be connected, so we define the family  $S(\Gamma)$  of shapes that we consider for a particular input feature set  $\Gamma$  to be the set of all connected components of the  $\alpha$ -complex  $A_\alpha(\Gamma)$ , for  $\alpha$  ranging from 0 to  $+\infty$ . Each member of this family is called an  $\alpha$ -component.



**Fig. 3.** The union of disks on the left is homotopy-equivalent to the  $\alpha$ -complex on the right, which is composed of points, edges and triangles. This complex has a hole.

## 3. SHAPE FROM POINT FEATURES

The key idea is: instead of thinking that a shape encloses a feature, we envision the set of features as “generating” the shape. In this spirit, let  $f : \Gamma \rightarrow \mathbb{R}$  be a score function, and let  $\Gamma^+$  be the set of features with positive score. Furthermore, for any  $\alpha$ -component  $W$  in the family  $S(\Gamma^+)$  generated by  $\Gamma^+$ , let  $\tau(W)$  be the minimal edge length that unites  $W$  into a single connected component. In other words, removing edges that are at least  $\tau(W)$  long would break the component, but removing edges that are strictly longer than  $\tau(W)$  would not. With these definitions, we formulate the problem:

**Definition 1.** (*Shape from Features.*) *The shape induced by feature set  $\Gamma$  and score function  $f : \Gamma \rightarrow \mathbb{R}$  is:*

$$\hat{W} = \arg \max_{W \in S(\Gamma^+)} \underbrace{\frac{\sum_{x \in W \cap \Gamma} f(x)}{\tau(W)}}_{\rho(W)} \quad (2)$$

where  $\rho(W)$  is the density of  $\alpha$ -component  $W$ , if  $W$  contains at least one edge. If  $W$  is a single point, define  $\rho(W) = 0$ .  $W \cap \Gamma$  is the set of features living within the  $\alpha$ -component  $W$ .

Note that the  $\alpha$ -components are generated from  $\Gamma^+$ , but the density includes all features in  $W \cap \Gamma$ , regardless of the sign of their scores. In other words, features of positive scores generate a shape that is indirectly bounded by the presence of features of negative scores.

The denominator in the definition of  $\rho(W)$  is a measure of the size of the  $\alpha$ -component  $W$ , so that  $\rho(W)$  can be viewed as evidence of “objectness” per unit size. Ideally, size should be measured by a function of the area of  $W$ , but this would lead to an intractable problem. We found  $\tau(W)$  to be an adequate proxy.

## 4. THE ALGORITHM

Algorithm 1 outlines our method for finding the optimal shape from a feature set  $\Gamma$  with given score function  $f$ . Since  $\alpha$ -complexes form a filtration, they can be built up in order of

increasing value of  $\alpha$ , and the separate  $\alpha$  components can be maintained in a union-find data structure. Initially, each feature is a separate  $\alpha$ -component. At each step of the construction, the density of the current  $\alpha$ -component is updated. The notation  $d(e)$  in Algorithm 1 denotes the length of edge  $e$ , and  $s(\Delta)$  represents the sum of all scores in triangle  $\Delta$ .

One of two cases arises when a new edge  $e$  is inserted. If  $e$  connects two previously distinct components, the score summation and the density are updated for the new component. On the other hand, if  $e$  joins vertices that already belong to the same component, we check if one or two triangles need to be added to the component (an edge in a simplicial complex can belong to zero, one, or two triangles). If triangles are added, we update the score summation and the density value accordingly.

---

**Algorithm 1** Compute the densest  $\alpha$ -component for feature set  $\Gamma$  and score function  $f$ .

---

- 1: Compute the Delaunay triangulation of  $\Gamma^+$ ;
  - 2: Put all Delaunay edges into priority queue  $E$  ordered by increasing lengths;
  - 3: For each feature  $i$ , set score  $s[i] \leftarrow f(i)$  and density  $\rho[i] \leftarrow 0$ ;
  - 4: Set  $\rho_{\max} \leftarrow 0$ ;
  - 5: **while**  $E$  is not empty **do**
  - 6:   Remove the shortest  $e$  from  $E$  and let  $i, j$  be the component IDs of its two endpoints;
  - 7:   **if**  $i \neq j$  **then**
  - 8:     Merge  $i$  and  $j$  into component  $i$  in the union-find structure;
  - 9:     Update:  $s[i] \leftarrow s[i] + s[j]$ ;  $\rho[i] \leftarrow s[i]/d(e)$ ;  $\rho_{\max} = \max\{\rho_{\max}, \rho[i]\}$ ;
  - 10:   **else**
  - 11:     **for** any triangle  $\Delta$  closed by  $e$  whose edges are all in component  $i$  **do**
  - 12:       Update:  $s[i] \leftarrow s[i] + s(\Delta)$  and  $\rho[i] \leftarrow s[i]/d(e)$ ;
  - 13:        $\rho_{\max} = \max\{\rho_{\max}, \rho[i]\}$ ;
  - 13:     **end for**
  - 14:   **end if**
  - 15: **end while**
- 

#### 4.1. Analysis

Once the algorithm has found the best density measure, the corresponding alpha component can be found by running the algorithm again and returning the alpha component that achieves the best density measure. Computing the 2D Delaunay triangulation and the score summation within each triangle takes time  $O(K \log K)$ . Sorting the Delaunay edges takes  $O(K \log K)$  time because the Delaunay triangulation is a planar graph and the number of its edges is less than  $3K - 6$ . The time complexity for union-find on  $K$  points is  $O(K\alpha(K))$  where  $\alpha(K)$  is the inverse Ackermann function.



**Fig. 4.** From left to right: The user input seeds, the features classified as positive (red) or negative (blue), and the segmented object with background removed. The entire computation takes less than 0.01 seconds on a quad-core laptop.

So the overall complexity of finding the optimal  $\alpha$ -component is  $O(K \log K)$  for a set  $\Gamma$  of  $K$  features.

## 5. DEMONSTRATIONS ON OBJECT CUT

We apply our method to segmenting an object out of a color image. Similarly to GrabCut [11], we require the user to somehow indicate what is inside and outside the desired object. In our case, all that is required is a small set of points in each set. In our experiments, we mark 20 points in the object and 20 points outside. Compared to GrabCut, input for our method is much easier for the user to specify, and computation is much faster. Moreover, holes are allowed.

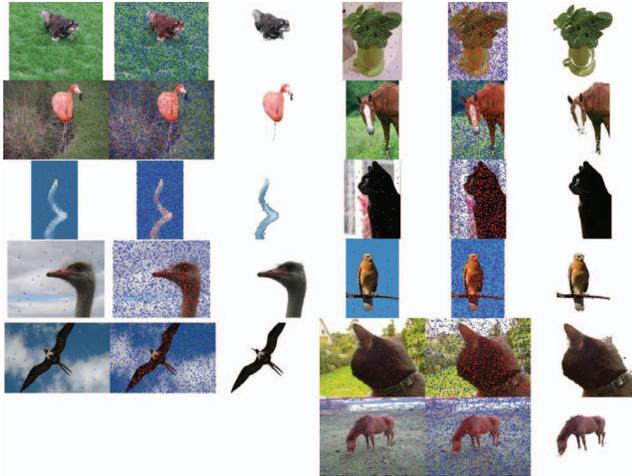
Let  $I$  be the given color image. We describe feature points by their R,G,B values. Let  $O$  be the set of input object color vectors and  $B$  be the set of background color vectors. We first classify each remaining image pixel into object, background, or neither by nearest neighbor: For each pixel  $p$  with 3D color vector  $u$ , we compute the ratio:

$$\zeta = \frac{\min_{v \in B} \|u - v\|}{\min_{v \in O} \|u - v\|}$$

We let  $f(p) = 1$  if  $\zeta > 3/2$  and  $f(p) = -1$  if  $\zeta < 2/3$ . Otherwise,  $f(p) = 0$ . Pixels with positive scores are treated as positive features  $\Gamma^+$ . We then apply Equation (2) to find the best  $\alpha$ -component using Algorithm 1. Since both the object and background contain only a few user selected features, computing the low dimensional nearest neighbor queries takes practically linear time with respect to the number of image pixels. Figure 4 illustrates the process, and Figures 5 and 6 show results. All images are chosen from the PASCAL data set [12].

### 5.1. Performance and Reproducibility

Our method for shape from features takes less than 0.01 seconds to find the optimal shape for an image of size  $400 \times 600$ . To the best of our knowledge, this is the fastest shape localization algorithm in the literature and is suitable for real time applications. The MATLAB and C++ code which implement the algorithm is available at [http://www.cs.duke.edu/~steve/shape\\_from\\_features.html](http://www.cs.duke.edu/~steve/shape_from_features.html)



**Fig. 5.** Each column shows the user input, the features classified as positive or negative, and the segmented object. No preprocessing (e.g., smooth) or postprocessing (e.g., hole filling) are applied. Best viewed when enlarged.

## 6. CONCLUSIONS

We present a nonparametric and efficient method for extracting flexible shapes from features in practically real time. Results on interactive object are very encouraging. We expect our method to be useful in other applications like supervised object detection, tracking, and computer graphics.

**Acknowledgement:** This work is supported by the Army Research Office under Grant No. W911NF-10-1-0387.

## 7. REFERENCES

- [1] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," *ICCV*, vol. 2, pp. 1470–1477, 2003.
- [2] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV Int'l W. on Statistical Learning in Computer Vision*, 2004.
- [3] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *IEEE CVPR*, 2005, vol. 2, pp. 524–531.
- [4] K. Grauman and T. Darrell, "The pyramid match kernel: Efficient learning with sets of features," *J. of Machine Learning Research*, vol. 8, pp. 725–760, 2007.
- [5] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999, pp. 1150–1157.
- [6] J. Shi and C. Tomasi, "Good features to track," in *IEEE CVPR*, 1994, pp. 593 – 600.
- [7] S. Gu, Y. Zheng, and C. Tomasi, "Efficient visual object tracking with online nearest neighbor classifier," in *ACCV*, 2010, pp. 271–282.



**Fig. 6.** More results on object cut. Results are left unedited with neither preprocessing nor postprocessing.

- [8] P. Viola, J. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *NIPS*, 2005.
- [9] C. Lampert, M. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE PAMI*, vol. 31, no. 12, pp. 2129–2142, 2009.
- [10] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–558, 1983.
- [11] C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.
- [12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html>.