Boundary Cues for 3D Object Shape Recovery

Kevin Karsch[1] Zicheng Liao[1] Jason Rock[1] Jonathan T. Barron[2] Derek Hoiem[1]
[1]University of Illinois at Urbana-Champaign [2]University of California, Berkeley
{karsch1, liao17, jjrock2, dhoiem}@illinois.edu barron@eecs.berkeley.edu

Abstract

Early work in computer vision considered a host of geometric cues for both shape reconstruction [11] and recognition [14]. However, since then, the vision community has focused heavily on shading cues for reconstruction [1], and moved towards data-driven approaches for recognition [6]. In this paper, we reconsider these perhaps overlooked "boundary" cues (such as self occlusions and folds in a surface), as well as many other established constraints for shape reconstruction. In a variety of user studies and quantitative tasks, we evaluate how well these cues inform shape reconstruction (relative to each other) in terms of both shape quality and shape recognition. Our findings suggest many new directions for future research in shape reconstruction, such as automatic boundary cue detection and relaxing assumptions in shape from shading (e.g. orthographic projection, Lambertian surfaces).

1. Introduction

3D object shape is a major cue to object category and function. Early approaches to object recognition [14] considered shape reconstruction as the first step. As data-driven approaches to recognition became popular, researchers began to represent shape implicitly through weighted image gradient features, rather than explicitly through reconstruction [13]. The best current approaches recognize objects with mixtures of gradient-based templates. Were the early researchers misguided to focus on explicit shape representation?

We have good reason to reconsider the importance of 3D shape. A study by Hoiem et al. [9] provides some evidence that gradient-based features are a limiting factor in object detection performance. Distinct architectures [6, 18] whose main commonality is gradient-based feature representations have very similar performance characteristics. The study also suggests that performance may be limited by heavy-tailed appearance distributions of object categories. For example, projected dog shapes may vary due to pose, viewpoint, and high intraclass variation. Because many examples are required to learn which boundaries are reliable (i.e., correspond to shape), dogs of unusual variety, pose, or viewpoint are poorly classified. Representations based on 3D shape would enable more sample-efficient category learning through viewpoint robustness and a reduced need to learn stable boundaries through statistics. Beyond interest in object categorization, ability to recover 3D shape is important for inferring object pose and affordance and for manipulation tasks.

The importance of shape is clear, but there are many mysteries to be solved before we can recover shape. What cues are important? What errors in 3D shape are important? How do we recover shape cues from an image? How do we encode and use 3D shape for recognition? In this paper, we focus on improving our understanding of the importance of boundary shape cues for 3D shape reconstruction and recognition. In particular, we consider boundaries due to object silhouette, self-occlusion (depth discontinuity) and folds (surface normal discontinuity). We also consider cues for whether boundaries are soft (extrema of curved surface) or sharp. We evaluate on a standard 3D shape dataset and a selection of PASCAL VOC object images. On the standard dataset, reconstructions using various cues are compared via metrics of surface normal and depth accuracy. On the VOC dataset, we evaluate reconstructions qualitatively and in terms of how well people and computers can categorize objects given the reconstructed shape.

Contributions. Our main contribution is to evaluate the importance of various boundary and shading cues for shape reconstruction and shape-based recognition. We extend Barron and Malik's shape from shading and silhouette method [1] to include interior occlusions with figure/ground labels, folds, and sharp/soft boundary labels. The standard evaluation is based on depth error, surface normal, shading, or reflectance on the MIT Intrinsic Image dataset. We also introduce perceptual and recognition-based measures of reconstruction quality for the PASCAL VOC dataset (Fig 1 shows one example of the types of reconstructions we evaluate, and the annotation required by our algorithm). These experiments are important because they tests reconstruction of typical objects, such as cats and boats, with complex shapes and materials in natural environments, and because it can provide insight into which errors matter. Furthermore, much work has gone into shape-based representations for recognition, focusing on the cues provided by the silhouette

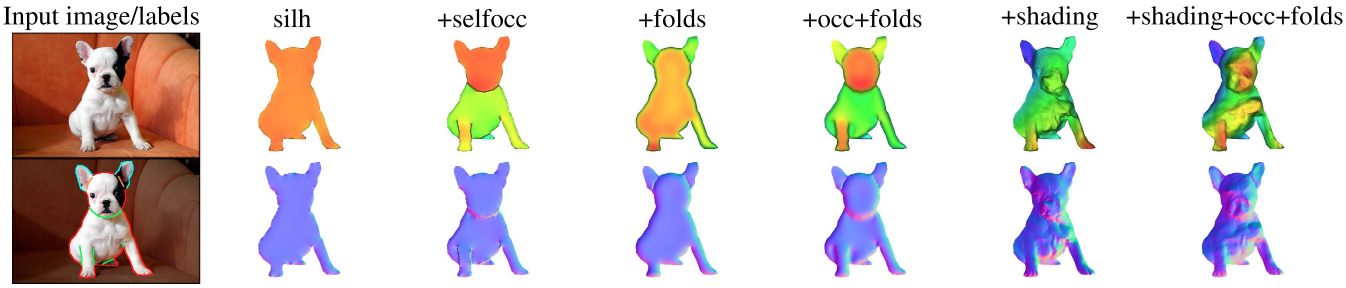


Figure 1. For a given input image, we hand-label geometric cues including: smooth silhouette contour (red), sharp silhouette contour (cyan), self occlusions (green), and folds (orange). We then use various combinations of these cues (as well as appearance-based cues) to obtain different shape reconstructions (see Sec 3). We evaluate these reconstructions in a variety of tasks in order to find which set(s) of cues may be most beneficial for reconstructing shapes.

(e.g. Ferrari et al. [7]). Our findings suggest a 3D representation that incorporates interior occlusions and folds might benefit such existing systems.

Limitations. Our study is a good step towards understanding shape reconstruction in the context of recognition, but we must leave several aspects of this complex problem unexplored. First, we assume boundary cues are provided. Eventually, we will want automatic recovery of shape cues and reconstruction algorithms that handle uncertainty. Second, cues such as ground contact points and object-level shape priors are useful but not investigated. Third, we assume an orthographic projection which can be a poor assumption for large objects, such as busses or trains. Finally, we recover depth maps, which provides a 2.5D reconstruction, rather than a full 3D reconstruction.

2. Cues for object reconstruction

We focus on reconstructing shape from geometric cues, revisiting early work on reconstructing shape from line drawings [11, 12]. Through human labeling, we collect information about an object's *silhouette*, *self-occlusions*, and *folds* in the surface. Since appearance can be a helpful factor in determining shape, we also investigate the benefit of shading cues using the shape-from-shading priors of Barron and Malik [1]. Figure 1 shows reconstructions using each of these cues.

To reconstruct shapes, we extend the continuous optimization framework of Barron and Malik by building in additional constraints on the surface. Following Barron and Malik's notation, we write Z for the surface (represented by a height field viewed orthographically), and $N : \mathbb{R} \to \mathbb{R}^3$ as the function that takes a height field to surface normals (component-wise; $N = (N^x, N^y, N^z)$). We use a coordinate system such that x and y vary in the image plane, and negative z is in the viewing direction.

Extending Barron and Malik's continuous optimization framework, we write our optimization problem as:

$$\begin{aligned} \underset{Z,R,L}{\text{minimize}} \quad & \delta_{sfc} f_{sfc}(Z) + \delta_{selfocc} f_{selfocc}(Z) \\ & +\delta_{fold} f_{fold}(Z) + \delta_{reg} f_{reg}(Z) \\ & +\delta_{sfs}(g(R) + h(L)) \\ \text{subject to} \quad & c_{sfs}(Z, R, L) = 0, \end{aligned} \tag{1}$$

where f_* and c_{sfs} are sub-objective and constraint functions, $g(R)$ and $h(L)$ are priors on reflectance and illumination, and δ_* are the weights that determine their influence. In the remainder of the section, we describe each of these functions/constraints.

Silhouette. The silhouette is rich with shape information, both perceptually and geometrically [10]. At the occluding contour of an object, the surface is tangent to all rays from the vantage point, unless however there is a discontinuity in surface normals across the visible and non-visible regions of the object (e.g. the edges of a cube). We treat these two cases separately, labeling parts of the silhouette as *smooth* if the surface normal should lie perpendicular to both the viewing direction and image silhouette, and *sharp* otherwise[1]. In the case of a smooth silhouette contour, the z-component of the normal is 0, and the x and y components are normal to the silhouette (i.e. perpendicular to the silhouette's tangent in 2D). Denoting (n^x, n^y) as normals of the silhouette contour, and C_{smooth} as the set of pixels labelled as the smooth part of the silhouette, we write the silhouette constraint as:

$$f_{sfc}(Z) = \sum_{i \in C_{smooth}} \sqrt{(N_i^x(Z) - n_i^x)^2 + (N_i^y(Z) - n_i^y)^2}. \tag{2}$$

This is the most typical constraint used in shape-from-contour algorithms (hence the notation f_{sfc}), and is identical to that used by Barron and Malik, with the notable exception that we only enforce the constraint when the silhouette is not sharp. If the silhouette is labelled sharp, there is no added constraint.

[1] It is also common notation to denote smooth boundaries as "limbs" and sharp boundaries as "edges" or "cuts"

Self-occlusions. Self-occlusions can be thought of in much the same way as the silhouette. The boundary of a self-occlusion implies a *discontinuity in depth*, and thus the surface along the foreground boundary should be constrained to be tangent to the viewing direction. Besides knowing a self occlusion boundary, it is also mandatory to know which side of the contour is in front of the other (figure and ground labels). With this information, we impose additional surface normal constraints along self occlusion boundaries ($C_{selfocc}$):

$$f_{selfocc}(Z) = \sum_{i \in C_{selfocc}} \sqrt{(N_i^x(Z) - n_i^x)^2 + (N_i^y(Z) - n_i^y)^2}. \tag{3}$$

Notice that there is no explicit constraint to force the height of the foreground to be greater than that of the background; however, by constraining the foreground normals to be pointing outward and perpendicular to the viewing direction, the correct effect is achieved. This is due in part because we enforce integrability of the surface (since height is directly optimized).

Folds. A fold in the surface denotes a *discontinuity in surface normals* across a contour along the object, e.g. edges where faces of a cube meet. Folds can be at any angle (e.g. folds on a cube are at 90°, but this is not always the case), and can be convex (surface normals pointing away from each other) or concave (surface normals pointing towards each other). Our labels consist of fold contours and also a flag denoting whether the given fold is convex or concave. We did not annotate exact fold orientation as this task is susceptible to human error and tedious.

We incorporate fold labels by adding another term to our objective function, developed using intuition from Malik and Maydan [12]. The idea is to constrain normals at pixels that lie across a fold to have convex or concave orientation (depending on the label), and to be oriented consistently in the direction of the fold. Let $\mathbf{u} = (\mathbf{u}_x, \mathbf{u}_y, 0)$ be a fold's tangent vector in the image plane, and N_i^ℓ, N_i^r as two corresponding normals across pixel i in the fold contour C. We write the constraint as

$$f_{fold}(Z) = \sum_{i \in C} \max(0, \epsilon - (N_i^\ell \times N_i^r) \cdot \mathbf{u}), \tag{4}$$

and set $\epsilon = \frac{1}{\sqrt{2}}$ (additional details can be found in the appendix).

Regularization priors. Because we only have constraints at a sparse set of points on the surface, we incorporate additional terms to guide the optimization to a plausible result. Following Barron and Malik, we impose one prior that prefers the flattest shape within the bas-relief family (f_f), and another that minimizes change in mean curvature (f_k):

$$f_f(Z) = -\sum_{i \in pixels} \log\left(N_i^z(Z)\right), \tag{5}$$

$$f_k(Z) = \sum_{i \in pixels} \sum_{j \in neighbors(i)} c\left(H(Z)_i - H(Z)_j\right), \tag{6}$$

$$f_{reg}(Z) = \lambda_f f_f(Z) + \lambda_k f_k(Z), \tag{7}$$

where $c(\cdot)$ is the negative log-likelihood of a Gaussian scale mixture, $H(\cdot)$ computes mean curvature, and the neighbors are in a 5x5 window around i. For all of our reconstructions, we set $\lambda_f = \lambda_k = 1$ (see [1] for implementation details).

Shading. We use the albedo and illumination priors of Barron and Malik to incorporate shading cues into our reconstructions. Summarizing these priors, we encourage albedo to be be piecewise smooth over space. Illumination is parameterized by second order spherical harmonics (9 coefficients per color channel), and is encouraged to match a Gaussian fit to real world spherical harmonics (regressed from an image based lighting dataset[2]). For brevity, we denote priors on reflectance as $g(R)$, and priors on illumination as $h(L)$, where R is log-diffuse reflectance (log-albedo) and L is the 27-dimensional RGB spherical harmonic coefficient vector. We refer the reader to [1, 2] for further details.

Jointly estimating shape along with albedo and illumination requires an additional constraint that forces a rendering of the surface to match the input image. Assuming Lambertian reflectance and disregarding occlusions, our rendering function is simply reflectance multiplied by shading (or in log space, log-reflectance plus log-shading). Denoting I as the log-input image, R as log-diffuse reflectance (log-albedo), and $S(Z, L)$ as the log-shaded surface Z under light L, we write the shape-from-shading constraint as:

$$c_{sfs}(Z, R, L) = R + S(Z, L) - I. \tag{8}$$

We emphasize that R, I, and $S(\cdot)$ are all in log-space, as is done in [1] which allows us to write the rendering constraint in additive fashion.

2.1. Optimization

To estimate a shape given a set of labels, we solve the optimization problem in Eq. 1 using the multiscale optimization technique introduced by Barron and Malik [1]. Notice that shading cues are only incorporated if $\delta_{sfs} > 0$; otherwise, our reconstructions rely purely on geometric information.

Setting the weights (δ). Throughout our experiments, we choose each weight to be binary for two reasons. For one, each term in the objective should have an equal weighting for a fair comparison, otherwise one cue may dominate others. Second, learning these weights requires a dataset of

[2] http://www.hdrlabs.com/sibl

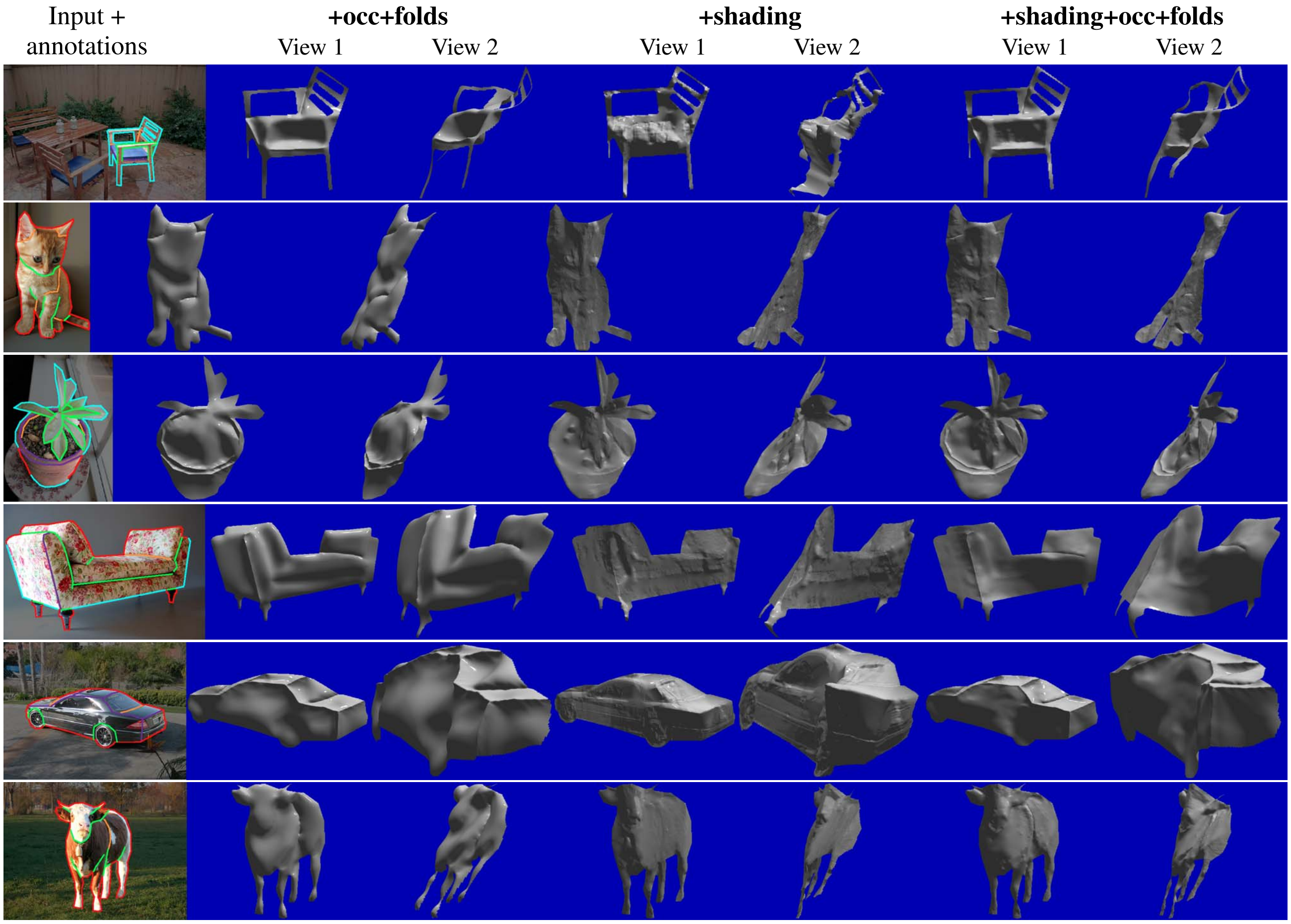


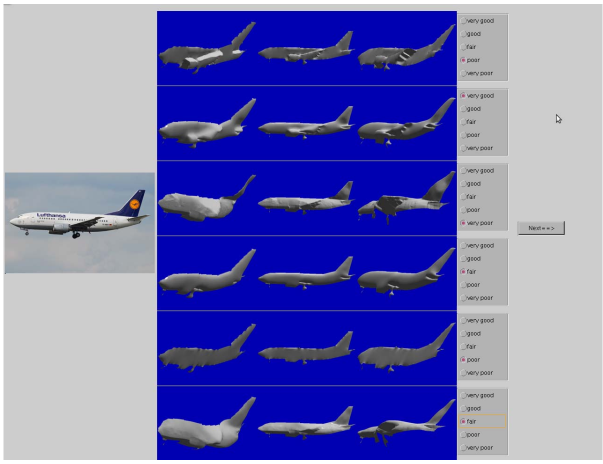
Figure 2. Several annotations and shape reconstructions used in our analyses. The annotated images (left) include: smooth silhouette contour (red), sharp silhouette contour (cyan), self occlusions (green), and folds (orange). In each row, we show the input image (with geometric labels), and the results of three reconstruction algorithms. For each algorithm, two views of the shape are shown (frontal on left, heavily rotated view on right). Notice that the reconstructed shapes look generally good frontally. Rotated views expose that shape estimates often err towards being too flat (especially with the cow or potted plant). This paper is the first that we know of to provide a rigorous analysis of shape reconstruction on typical objects in consumer photographs (e.g. outside of a lab setting).

ground truth shapes, and we have good reason to believe that weights learned from existing datasets (e.g. the MIT Intrinsic Image dataset [8]) will not generalize to shapes found in the VOC dataset (e.g. more geometric detail on VOC shapes). Furthermore, we ran the MIT-learned parameters on several of the VOC images, and noticed only slight perceptual differences in results.

3. Evaluation of shape and appearance cues

In this section, we examine each of the cues used in our shape reconstruction method, and hope to find a cue or set of cues that lead to better shape estimates (qualitatively, and in terms of recognition ability). Our objective function (Eq 1) allows us to easily produce shape reconstructions for various combinations of cues by turning "on" and "off" different cues; equivalently, setting the corresponding weights to 1 (on) or 0 (off). We use six different cue combinations to see which cue or set of cues contribute most to a better reconstruction. These six combinations are:

- **silh**: Priors on silhouette shape and surface smoothness; i.e. shape-from-contour constraints ($\delta_{sfc} = 1$).
- **+selfocc**: Silhouette and self occlusion constraints ($\delta_{sfc} = \delta_{selfocc} = 1$).
- **+folds**: Silhouette and fold constraints ($\delta_{sfc} = \delta_{folds} = 1$).
- **+occ+folds**: Silhouette, self occlusion and fold constraints ($\delta_{sfc} = \delta_{selfocc} = \delta_{folds} = 1$).
- **+shading**: Shape-from-shading as in [1]; includes silh ($\delta_{sfc} = \delta_{sfs} = 1$).
- **+shading+occ+folds**: SFS with self occlusion and fold constraints ($\delta_{sfc} = \delta_{sfs} = \delta_{selfocc} = \delta_{folds} = 1$).



○ very good ○ good ○ fair ● poor ○ very poor

● very good ○ good ○ fair ○ poor ○ very poor

○ very good ○ good ○ fair ○ poor ● very poor

○ very good ○ good ● fair ○ poor ○ very poor

○ very good ○ good ○ fair ● poor ○ very poor

○ very good ○ good ● fair ○ poor ○ very poor

Figure 3. User study interface for qualitative rating. Left: source image of the object. Middle: shape visualizations. Each row is the result of one algorithm (silh, +selfocc, etc) in random order visualized in three view angles: upper left, frontal, bottom right (from left to right). The participant rates each row as a whole.

We will refer to these as separate *algorithms* for the remainder of the paper, and Fig 1 shows an example reconstruction for each of these algorithms. Note that silh cues are present in each algorithm (hence the '+' prefix).

To find which cues are most critical for recovering shape, we evaluate each algorithm on a variety of tasks that measure *shape quality* and *shape recognition*. We first evaluate the performance of the six algorithms on the VOC 2012 dataset. We selected 17 of the VOC categories out of the 20 (we exclude "bicycle", "motorbike" and "person" since we found these objects difficult to label by hand). Each class has 10 examples. Since we do not have ground truth shape for VOC objects, we conduct two user studies to evaluate qualitative performance: *qualitative rating* and *shape-based recognition*. Next, we evaluated the different algorithms using existing automatic recognition techniques, and compare them to the results of using RGB features (alone) and RGB+depth features. Finally, we ran a quantitative comparison of depth and surface normals using the MIT depth dataset. The remainder of this section details our results for each of these tasks, split under headings concerning shape quality and shape recognition.

3.1. Shape quality

Our experiments examine shape quality perceived by people (through a user study) and computers (ground truth comparison). The goal of these experiments is to find a common set of cues, or shape reconstruction algorithm(s), that consistently report the best shape.

Qualitative rating on VOC. The qualitative rating portion of the user study collected subjects' ratings for each of the six shape reconstruction algorithms. We designed an interface (Fig. 3) that displays the visualization of the six shape estimation results side by side on the screen and allows participants to rate the quality of each shape estimation result from scale 1 (very poor) to 5 (very good). The 17 class × 10 instance results are shuffled and divided into 5 groups. Each participant rated an entire group. Additional images from the study are displayed in Fig 2.

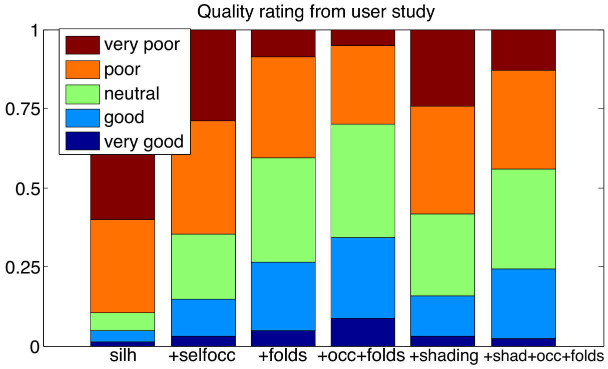


Figure 4. For each algorithm, we show the percentage of times a certain rating was assigned to it during the qualitative rating user study. +occ+folds had the highest average rating, followed closely by +folds and +shading+occ+folds. Notice however that there is still much room for improvement, since the best-rated method (+occ+folds) was only chosen as "good" or "very good" less than 30% of the time.

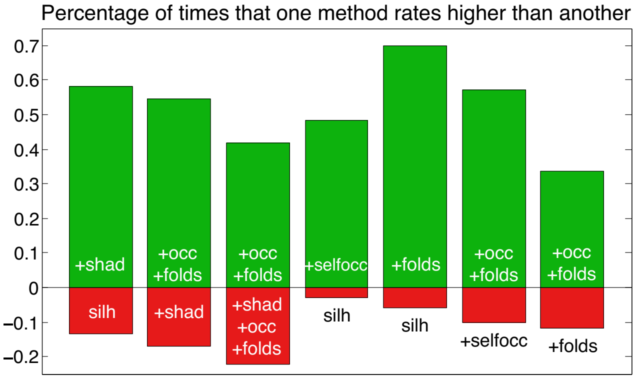


Figure 5. The percentage that one algorithm rates higher (green color) or lower (red color) than another. Each column shows the result of one algorithm pair. For example, for the left most column, +shading was rated above silh approximately 60% of the time, below silh about 10% of the time, and rated the same otherwise. Shading seems to help when accompanied by with a silhouette cues, but when additional boundary cues are present, shading tends to produce more artifacts than improvements. We also see a strong improvement from combining fold and occlusion contours.

Our results indicate that +occ+folds is the most appealing reconstruction method to humans, followed closely by +folds and +shading+occ+folds. Figure 4 shows the averaged rating score grouped by algorithm; where a higher average rating indicates a better shape. In every case, as intuition suggests, adding more geometric cues leads to a more preferable shape. For an algorithm-by-algorithm comparison, we plot the percentage of times that one algorithm was rated higher than another (Figure 5). Here, we see geometric cues (other than silh) were consistently preferred over shading cues; in one example, +occ+folds was rated higher than +shading+occ+folds about 40% of the time.

Ground truth comparison. Using ground truth shapes available from the MIT Intrinsic Image dataset [8], we analyze our shape reconstructions using established errors metrics. We report results for both a surface normal-based error metric, N-MSE [1], as well as for a depth-based error metric, Z-MAE [2]. N-MSE is computed as the mean squared error of the difference in normal orientation (measured in radians), and Z-MAE is the translation-invariant absolute error of of depth. Both metrics are averaged per-pixel, over the entire dataset of 20 objects. We also ran the same comparison, but substituted Barron and Malik's learned weights on the MIT dataset for our binary weights (δ_*); these results are in the N-MSE† and Z-MAE† columns:

	N-MSE	Z-MAE	N-MSE†	Z-MAE†
silh	0.573	25.533	0.521	25.637
+selfocc	0.565	25.198	0.498	25.342
+folds	0.496	25.562	0.501	25.400
+occ+folds	0.487	25.161	0.482	24.983
+shading	0.874	38.968	0.310	25.793
+shading+occ+folds	0.574	27.379	0.350	24.492

We observe that adding geometric cues generally increase quantitative performance. One notable exception is in the N-MSE† column, where +shading alone performs the best. This is almost certainly because all +shading reconstruction has been trained on the MIT dataset, whereas several parameters for the other algorithms have not been (e.g. $\delta_{selfocc}, \delta_{fold}$). Surprisingly, using binary weights (as in the N-MSE and Z-MAE columns) results in significantly worse +shading performance, but the geometric-based algorithms are largely unaffected. However, for non-MIT reconstructed shapes (e.g. VOC), using binary weights versus the learned weights weights gave perceptually similar results, possibly indicating that these metrics are sensitive to different criteria than human perception.

3.2. Shape for object recognition

We are also interested in how well our shapes convey the object that has been reconstructed. Here, we describe experiments that gauge this task both through a human recognition study and computer recognition algorithms.

In the second task, we asked users to identify the object class based on the reconstructed shape alone. Our hypothesis is that higher class-recognition indicates better shape quality. We also consider that object silhouette could be a dominating factor for recognition; to reduce this factor, we show a silhouette-masked view of each result first (Fig. 6 left); and then show the result without masking (Fig. 6 right). The task is evenly divided into 7 groups. Each group is assigned to one participant, who will go over all of the 170 objects in our test set. For each object, only one of the 7 results (generated by random permutation) is shown to one participant. The participants are also asked to rate their level of confidence on a scale from 1 (least confident) to 5 (most confident).

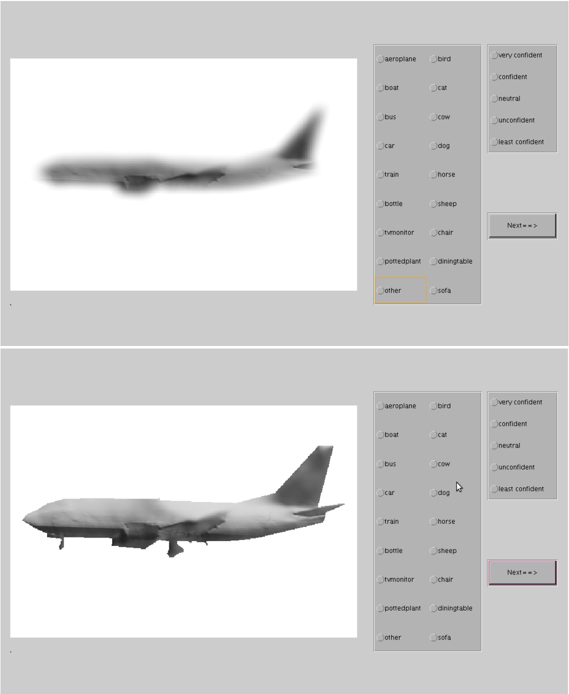


Figure 6. User study interface for shape-based recognition. Participants are asked to recognize the object category using shape alone, estimated with one of the six algorithms. For each trial, a "masked" view (top) is displayed first to deter silhouette-based recognition, followed by the unmasked view (bottom).

Figure 7 displays the recognition error rate for each algorithm. For each algorithm, the left bar shows the result from the masked view; the right bar shows that result from the unmasked view. In the masked view, +occ+folds yields the lowest recognition error, consistent with qualitative rating portion of our user study. In the unmasked view, +shading+occ+folds performs the best, closely followed by +occ+folds.

Automatic recognition. We evaluate the shapes by performing classification on the depth maps. Outside the image, the depth is set to 0. Since the heights inside objects are set to be fairly high, this ensures that there is a large edge at the contour. To provide some invariance to specifics of classification methods or features, we run classification using two methods. We use a PHOW feature from [5] and a the Pegasos SVM solver [15] with homogeneous kernel mapping [19] as a baseline classifier (all available from VLFeat [17]). It is motivated by a similar method in [16] used to classify objects in Kinect images. For another method, we use the RGB-D kernel match descriptors of [3, 4] for which code is available. Leave one out cross validation is used to determine the accuracy of classification

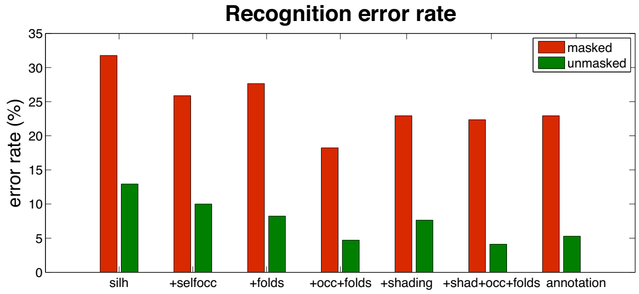


Figure 7. Recognition error rate (as judged by participants in our user study) for each algorithm using the silhouette-masked and unmasked (unaltered) images.

	RGB-D kernel [4]	VLFeat [17]
rgb	55.29	-
+occ+folds+rgb	**70.00**	-
+shading+occ+folds+rgb	62.35	-
+shading	48.24	41.76
+shading+occ+folds	52.94	42.94
silh	47.06	45.29
+selfocc	65.88	**54.12**
+folds	51.76	47.65
+occ+folds	65.88	51.76

Table 1. Average recognition accuracy for different sets of features using existing, automatic recognition methods. rgb implies that image appearance was used as a feature (row 1), and compared against rgb+depth (rows 2 and 3), as well as using depth alone (remaining rows). As one might expect, adding geometric features to the existing rgb information improves recognition accuracy, and shape tends to be more revealing than appearance alone. VLFeat offers only depth classification, hence the missing entries.

on each reconstruction as well as rgb, rgb+occ+folds, and rgb+shading+occ+folds for the kernel matching method to determine if shape and shading cues add information compared to RGB alone.

Table 1 shows classifications results for each of the metrics. Our classification accuracy results are slightly different than the human ratings, although there are some similar trends. +occ+folds still appears to be one of the best, though it is beaten in this case by +selfocc. Also expected, +shading+occ+folds outperforms +shading. The ordering of the remaining reconstructions is less consistent across the two classifiers, therefore it is difficult to draw any strong conclusions. It is interesting that +folds performed poorly but was rated highly by our test subjects; this likely implies that the features used do not make use of the information available from folds.

We also show the results of an RGB classifier using [4]. While state of the art classification on VOC2012 is roughly 70%, we see only 55% due to the constrained dataset (few examples per class). The shape reconstructions increase the accuracy of the result, but this could be partially due to the mask provided by the height which is not available in the RGB only method.

4. Conclusion

We demonstrate a simple and extensible technique for reconstructing shape from images, resurrecting highly informative cues from early vision work. Our method itself is an extension of Barron and Malik's [1] reconstruction framework, and we show how additional cues can be incorporated in this framework to create improved reconstructions.

Through our experiments, we have shown the necessity of considering cues that go beyond typical shape-from-shading constraints. In almost every task we assessed, using more geometric cues gives better results. For human-based tasks, shading cues seem to help when applied with to silhouette cues (+shading consistently outperforms silh), but adds little information once additional boundary cues are incorporated (+occ+folds performs similarly to +shading+occ+folds); see Figs 4 and 7. Interestingly, when the boundary is not available for viewing, +occ+folds performs better than +shading+occ+folds (Fig 7; masked errors), and shading cues seem to have an adverse effect on automatic recognition algorithms (Table 1). As far as we know, our experiments are the first to evaluate reconstruction methods on consumer photos (e.g. PASCAL VOC).

One interesting observation from our experiments is that our shading cues tend to confound boundary cues; e.g. +occ+folds outperforms +shading+occ+folds in each task except (unmasked) human recognition (Sec 3.2). It seems counterintuitive that incorporating shading information would degrade reconstructions, and we offer several possible causes. Foremost is the fact that we weight all terms equally, whereas learning these weights from ground truth will lead to better shading reconstructions (evidenced especially by our quantitative results on the MIT Intrinsic dataset in Sec 3.1). Second, this observation may be in part due to the inherent assumptions of existing shape-from-shading algorithms, including our own (e.g. 2.5D shape, orthographic camera, Lambertian reflectance, and smooth and infinitely distant illumination). Our tests use real images from PASCAL, and some contain significant perspective, as well as complex reflectance and illumination. Relaxing these assumptions, as well as developing and enforcing stronger shape priors, are difficult but interesting problems for future research.

Our evaluations show that self occlusion and fold cues are undoubtedly helpful, and most importantly, point in many directions for improving existing shape reconstruction algorithms. Extracting boundary cues, such as folds and self occlusions, automatically from photographs is a logical next step. It is also evident that shape-from-shading algorithms can be improved by incorporating additional geometric cues, and additional research should go into extending shape-from-shading to real world (rather than lab) images. In terms of reconstructing shapes, considering perspective projections (rather than orthographic) may help, as well as extending surface representations beyond 2.5D and into 3D. By exploring these directions, we believe significant steps can be taken in the longstanding vision goal of reconstructing shape in the wild.

Acknowledgements

This research was supported in part by the NSF GRFP (KK & JB), NSF Award IIS 0916014 and ONR MURI Awards N00014-10-10933/N00014-10-10934.

References

[1] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. In *ECCV*, 2012.

[2] J. T. Barron and J. Malik. Shape, albedo, and illumination from a single image of an unknown object. In *CVPR*, 2012.

[3] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. *NIPS*, 2010.

[4] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *Intelligent Robots and Systems (IROS)*, pages 821–826. IEEE, 2011.

[5] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *ICCV*. IEEE, 2007.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE TPAMI*, 32(9):1627–1645, 2010.

[7] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of Adjacent Contour Segments for Object Detection. *IEEE TPAMI*, 30(1):36–51, Jan. 2008.

[8] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. *ICCV*, 2009.

[9] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[10] J. Koenderink. What does the occluding contour tell us about solid shape. *Perception*, 1984.

[11] J. Malik. *Interpreting line drawings of curved objects*. PhD thesis, Stanford University, Stanford, CA, USA, 1986.

[12] J. Malik and D. Maydan. Recovering three-dimensional shape from a single image of curved objects. *IEEE TPAMI*, 11(6):555–566, June 1989.

[13] J. L. Mundy. Object recognition in the geometric era: A retrospective. In *Toward Category Level Object Recognition*, pages 3–29. Springer, 2006.

[14] L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963.

[15] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, pages 807–814. ACM, 2007.

[16] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *ICCV - Workshop on 3D Representation and Recognition*, 2011.

[17] A. Vedaldi and B. Fulkerson. Vlfeat – an open and portable library of computer vision algorithms. In *Proceedings of the 18th annual ACM International Conference on Multimedia*, 2010.

[18] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[19] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE TPAMI*, 34(3):480–492, 2012.

Appendix: Fold constraint implementation

Consider the (i)th point on the contour C, parametrized by position $\mathbf{p} = [\mathbf{p}_x,\, \mathbf{p}_y]$ and tangent vector $\mathbf{u} = [\mathbf{u}_x,\, \mathbf{u}_y]$, both on the image plane. The sign of the tangent vector is arbitrary. Let us define a vector perpendicular to each tangent vector: $\mathbf{v} = [-\mathbf{u}_y,\, \mathbf{u}_x]$. By default, this fold is convex — folded in the direction of negative Z. To construct a concave fold, we flip the sign of $\mathbf{v}$. With this parametrization, we can find the positions of the points to the left and right of the point in question relative to the contour:

$$\mathbf{p}^\ell = [\mathrm{round}\,(\mathbf{p}_x + \mathbf{v}_x)\,,\, \mathrm{round}\,(\mathbf{p}_y + \mathbf{v}_y)] \tag{9}$$

$$\mathbf{p}^r = [\mathrm{round}\,(\mathbf{p}_x - \mathbf{v}_x)\,,\, \mathrm{round}\,(\mathbf{p}_y - \mathbf{v}_y)] \tag{10}$$

Given a normal field N we compute the normal of the surface at these “left” and “right” points:

$$N^\ell = [N_x(\mathbf{p}^\ell_x, \mathbf{p}^\ell_y),\, N_y(\mathbf{p}^\ell_x, \mathbf{p}^\ell_y),\, N_z(\mathbf{p}^\ell_x, \mathbf{p}^\ell_y)] \tag{11}$$

$$N^r = [N_x(\mathbf{p}^r_x, \mathbf{p}^r_y),\, N_y(\mathbf{p}^r_x, \mathbf{p}^r_y),\, N_z(\mathbf{p}^r_x, \mathbf{p}^r_y)] \tag{12}$$

Consider c, the dot product of $[\mathbf{u}_x, \mathbf{u}_y, 0]$ with the cross-product of $\mathbf{n}^\ell$ and $\mathbf{n}^r$:

$$c = \mathbf{u}_x(N^\ell_y N^r_z - N^\ell_z N^r_y) + \mathbf{u}_y(N^\ell_z N^r_x - N^\ell_x N^r_z) \tag{13}$$

If $c = 1$, then the cross product of the surface normals on both sides of the contour is exactly equal to the tangent vector, and the surface is therefore convexly folded in the direction of the contour. If $c = -1$, then the surface is folded and concave. Of course, If the sign of the contour, and therefore of the $\mathbf{v}$ vector, is flipped, then $c = 1$ when the surface is concavely folded, etc. Intuitively, to force the surface to satisfy the fold constraint imposed by the contour, we should force c to be as close to 1 as possible. This is the insight used in edge constraint of the shape-from-contour algorithm in [12]. But constraining $c = 1$ is not appropriate for our purposes, as it ignores the fact that $\mathbf{u}$ and therefore $\mathbf{v}$ lie in an image plane, while the true tangent vector of the contour may not be parallel to the image plane. To account for such contours, we will therefore penalized c for being significantly smaller than 1. More concretely, we will minimize the following cost with respect to each contour pixel:

$$f_{\mathit{fold}}(N(Z)) = \sum_{i \in C} \max(0, \epsilon - c^{(i)}), \tag{14}$$

where $\epsilon = \frac{1}{\sqrt{2}}$. This is a sort of ϵ-insensitive hinge loss which allows for fold contours to be oriented as much as $45°$ out of the image plane. In practice, the value of ϵ effects how sharp the contours produced by the fold-constraint are — $\epsilon = 0$ is satisfied by a flat fronto-parallel plane, and $\epsilon = 1$ is only satisfied by a perfect fold whose crease is parallel with the image plane. In our experience, $\epsilon = \frac{1}{\sqrt{2}}$ produces folds that are roughly $90°$, and which look reasonable upon inspection.