Shapecollage: occlusion-aware, example-based shape interpretation

Forrester Cole, Phillip Isola, William T. Freeman, Frédo Durand, and Edward H. Adelson

Massachusetts Institute of Technology
{fcole,phillipi,billf,fredo,adelson}@csail.mit.edu

Abstract. This paper presents an example-based method to interpret a 3D shape from a single image depicting that shape. A major difficulty in applying an example-based approach to shape interpretation is the combinatorial explosion of shape possibilities that occur at occluding contours. Our key technical contribution is a new shape patch representation and corresponding pairwise compatibility terms that allow for flexible matching of overlapping patches, avoiding the combinatorial explosion by allowing patches to explain only the parts of the image they best fit. We infer the best set of localized shape patches over a graph of keypoints at multiple scales to produce a discontinuous shape representation we term a *shape collage*. To reconstruct a smooth result, we fit a surface to the collage using the predicted confidence of each shape patch. We demonstrate the method on shapes depicted in line drawing, diffuse and glossy shading, and textured styles.

1 Introduction

A long-standing goal of computer vision is to recover 3D shape from a single image. Early researchers developed techniques for specific domains such as line drawings of polyhedral objects [1], shape-from-shading for Lambertian objects [2], and shape-from-texture for simple textured objects [3]. While based on solid mathematical and physical foundations, these techniques have proven difficult to generalize beyond limited cases: even the seemingly simple line drawing and shaded images of Figure 1 confound all existing techniques.

Recently, machine learning methods (e.g., [4, 5]) have been proposed to allow generalization by learning the relationship between shape and training images. However, the flexibility and computational resources required for learning-based vision depend critically on the choice of shape representation. Occlusion boundaries, for example, lie between uncorrelated regions and cause a combinatorial explosion in possibilities if not explicitly included in the representation.

In this paper, we describe a fully data-driven approach that reconstructs an input shape by piecing together patches from a training set. We make this approach practical by using a new, irregular, multi-scale representation we term a *shape collage*, coupled with a new patch representation and compatibility measurement between patches that addresses the combinatorial explosion at occlusion boundaries and generalizes patches across scale, rotation, and translation.

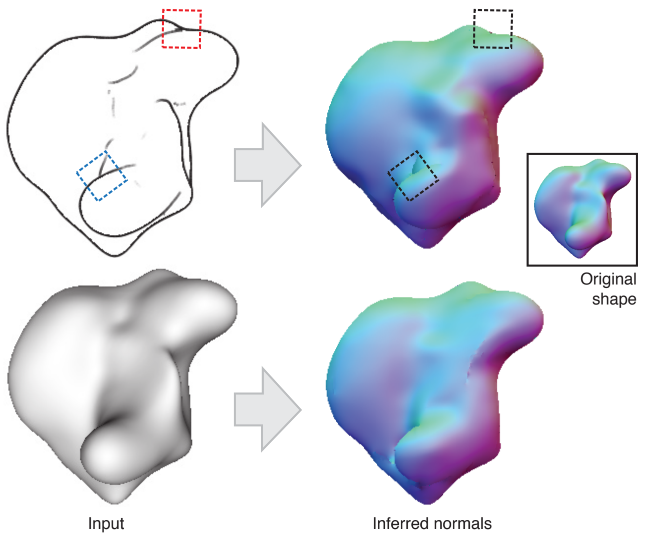

Fig. 1. A line drawing and diffuse rendering of a 3D shape (left) are interpreted as normal maps by our system (right). Each rendering produces a different plausible interpretation. Our system uses compatibility between local depth layers to interpret self-occlusion (blue box) and near-occlusion (red box).

Given compatibility between patches, we use belief propagation to produce the most likely collage and then fit a smooth surface to produce the final shape interpretation (Figure 1).

We demonstrate our approach on synthetic renderings of blobby shapes in a range of rendering styles, including line drawings and glossy, textured images. Interpreting line drawings of curved shapes has been a long-standing unsolved problem in computer vision [6, 7]. Much progress has been made for drawings and wire frame representations of 3D shapes with flat facets (e.g., [8]), but progress on the interpretation of generic line drawings has stalled. Interpretation of shaded renderings has been addressed more frequently in the computer vision community, but is still not a solved problem [2].

Our contributions are in the design of the shape patches and the measurement of compatibility between them, the irregular, multi-scale relationships between patches, and the selection of a diverse set of shape candidates for each patch. Besides shape, our representation can also estimate factors such as the visual style of the input.

1.1 Related Work

The shape-from-a-single-image problem is perhaps the oldest in computer vision, dating at least to Roberts [1]. Because of the difficulty of the problem, it has over time been divided into subproblems such as shape-from-shading (see [2] for a recent survey), shape-from-line-drawing (e.g., [6–8]), and shape-from-texture (e.g., [3]). These parametric methods have difficulty capturing many of the cases that occur in the visual world, while learning-based methods can successfully capture such visual complexity. Saxena, et al [5], use a learning-based approach to find a parametric model of a 3D scene and also explicitly treat occlusion boundaries. While their system can process complex input due to its learned model, their parametric scene description is tailored for boxy shapes such as buildings and does not contain features other than shape. By contrast, our rich, example-based model extends to smooth shapes and can predict visual style as well as shape.

For line drawing recognition, Saund [9] applied belief propagation in a Markov Random Field (MRF) to label contours and edges of pre-parsed sketch images, although without any explicit representation of shape. Ouyang and Davis [10] combined learned local evidence for symbol sketches with MRF inference for sketch recogntion, focussing on chemical sketches.

Our work is a spiritual successor to the work of Freeman, et al [11], "learning low-level vision." That system uses a network of multi-scale patches and a set of candidates at each patch to infer the most likely explanation for the stimulus image. Hassner and Basri [4] also use a patch-based approach to reconstruct depth, while adding the ability to synthesize new exemplar images on-the-fly. Unlike both systems, we use an irregular network of patches centered on interest points in the image, and directly tackle the problem of occlusion boundaries. Because our training data is synthetic we could also create new exemplars on-the-fly, but have not explored that option in this work.

Example-based image priors were also used by Fitzgibbon et al [12] in the context of image interpolation between measurements from multiple cameras. The benefits of learning from large, synthetically generated datasets were shown emphatically in the recent work of Shotton et al [13].

2 Overview

Our approach finds interest points in the image, selects candidate interpretations for each point, then defines a Markov random field tailored to those interests and performs inference to determine the most likely shape configuration (Figure 2).

Shape is represented by patches that contain the rendered appearance of the patch, a normal map, a depth map, a map of occluding contours, and an ownership mask that defines for which pixels the patch provides a shape explanation (Figure 3). Patches are placed and oriented using image keypoints computed at multiple scales. As described in Section 3, we found that current approaches for detecting keypoints do not provide a good set of points for stimuli such as line drawings, and designed our own method based on disk sampling of an interest map. We propose a simple set of multi-scale graph relationships for these keypoints.

The local appearance at each keypoint determines the selection of candidate patches (Section 4). There may be thousands of possible matches for the more

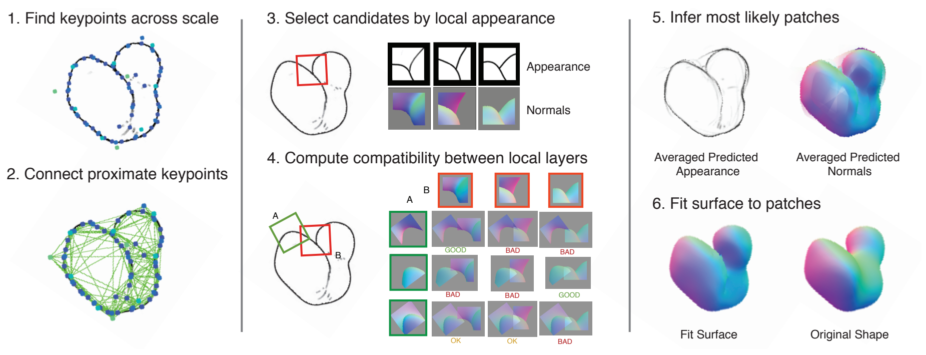
1. Find keypoints across scale
2. Connect proximate keypoints
3. Select candidates by local appearance
Appearance
Normals
4. Compute compatibility between local layers
A
B
GOOD
BAD
BAD
BAD
BAD
GOOD
OK
OK
BAD
5. Infer most likely patches
Averaged Predicted Appearance
Averaged Predicted Normals
6. Fit surface to patches
Fit Surface
Original Shape

Fig. 2. The major steps of our approach. First, keypoints are extracted at multiple scales (1), then neighboring keypoints in both image space and scale are connected (2). Separately, a set of candidate shapes is selected for each keypoint based on local appearance (3), and the compatibility of each candidate is evaluated against the candidates at each connected keypoint (4). Loopy belief propagation is used to find the most likely candidates, which form a *shape collage* (5). Finally, a smooth surface is fit to the collage (6).

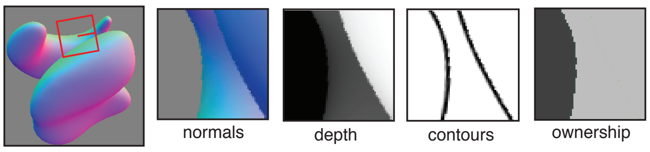

Fig. 3. Our shape patch representation. Given a square, oriented training patch (left, red box) the normal map is extracted and rotated to match the keypoint orientation. The depth map and occluding contours are also stored along with an ownership mask.

common patches, far more than our inference algorithm can support. It is critical to choose a diverse-enough subset of candidates, for which we use similarity of local shape descriptors.

The patch candidates are scored based on similarity of local appearance, normal and depth layer compatibility with adjacent patches, and prior probability (Section 5). The most likely shape collage given these scores is found using loopy belief propagation and a thin-plate spline surface is fit to this shape collage.

3 Keypoints and Graph

The first step of shape interpretation is to extract interest-driven keypoints at multiple scales and connect them into a graph. Each keypoint will become the center of a patch. A keypoint has four parameters: image position, scale, and orientation. All four are found by the detection stage and remain fixed throughout the interpretation process.

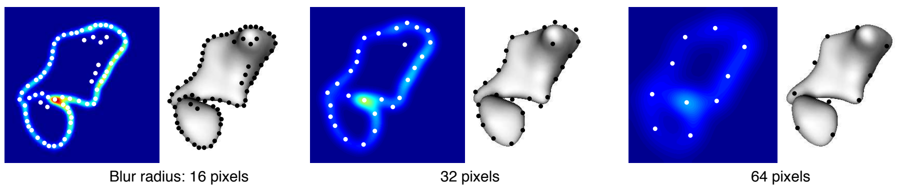

Fig. 4. Keypoint detection at three scales (blur at 16, 32, 64 pixels, image: 300^2 pixels). Left of pair: level of "interest" in the image, computed by the trace of the structure tensor. Right of pair: original image with keypoints overlaid. Keypoints are placed to cut out the "most interesting" pixel until a threshold is reached.

Our criteria for a keypoint detector are: *coverage*, all the "interesting" parts of the image should be covered by at least one keypoint; *sparsity*, keypoints should not be redundant; and *repeatability*, similar image patches should have a similar distribution of keypoints.

Standard methods such as Harris corner detection [14] and SIFT [15] are not directly suitable for our needs. Corner detection, for example, fails the coverage criterion, since we want keypoints along all linear features in addition to corners. The SIFT keypoint detector also fails the coverage criterion: some "interesting" image features do not correspond to maxima of the blob detector in scale space, and so do not produce a SIFT keypoint.

By contrast, a naive method such as a regular or randomized grid fails the repeatability criterion. Repeatability is important for two reasons: it reduces the number of required training examples, and increases the effectiveness of local appearance matching (Section 5) by aligning image and patch features.

3.1 Detection

Our approach to keypoint detection is to compute an interest map over the image, then iteratively place keypoints at the highest point of this map that is not already near another keypoint. Intuitively, this approach uses a cookie-cutter to greedily stamp out keypoints at the most interesting points in the image.

The interest map $I(\mathbf{p})$ is defined as the sum of the eigenvalues (i.e., the trace) of the 2x2 structure tensor of the test image. Prior to computing the structure tensor, the image is blurred to the desired level of scale. For a keypoint of radius r, the blur is defined as a Gaussian of $\sigma = r/3$.

The stamp shape for a keypoint at $\mathbf{p}$ is a radially symmetric function of the distance s from $\mathbf{p}$, with a smooth rolloff:

$$\texttt{stamp}_{\mathbf{p}}(s) = \begin{cases} I(\mathbf{p}) & : s < 0.25r \\ I(\mathbf{p})G(s - 0.25r) & : s > 0.25r \end{cases} \tag{1}$$

where G is a Gaussian of $\sigma = r/3$. The stamp function is subtracted from the interest map for each keypoint. Detection stops when the maximum remaining value in the interest map falls below a 1% of the original maximum value.

The orientation of a keypoint is found using the SIFT orientation detector, which defines orientation using the histogram of gradients inside the keypoint radius [15].

This detection method satisfies our three criteria: the iterative procedure ensures that the keypoints are comprehensive and cover all interesting areas in the image. The stamp function ensures that the keypoints are sparse. The keypoint selection is repeatable since the trace of the structure tensor is rotation and translation invariant.

3.2 Graph Connections

We treat each keypoint as a vertex in an MRF graph. The edges of the graph are defined by the positions and scales of the keypoints. The edge weights correspond to the compatibility between shape patches. To compute compatibility, we need sufficient overlap between patches. We therefore only connect two keypoints if

$$\|\mathbf{p}_1 - \mathbf{p}_2\| < (r_1 + r_2) * 0.8 \tag{2}$$

where $\mathbf{p}_i$ are the keypoint positions and r_i are the radii.

Additionally, two keypoints are only connected if they are close enough in scale. In general, large scale patches will fit the test image more coarsely than small scale patches. The tolerable margin of error for large patches may be wider than an entire patch at a small scale, making the compatibility score between very large and very small patches useless. Therefore, we only connect patches where

$$|\log_2 r_1 - \log_2 r_2| <= 1 \tag{3}$$

or in other words, where the radii differ by a power of two or less.

4 Selecting shape candidates

We formulate shape interpretation as a discrete labeling problem on the keypoint graph, where the labels correspond to candidate shape patches.

Ideally, every patch in the training set would be a candidate at every vertex in our graph. However, it is computationally intractable to include all $\sim 10^6$ training patches at each vertex. We must carefully prune a subset of the training set for each vertex in our graph. Proper selection of this subset is critical to successful interpretation because it defines the space of possible shapes that the MRF inference can explore.

Our approach is as follows. Given a keypoint and its associated test image patch, we first select the subset of the training patches that are similar in appearance to the test patch. This subset may still include several thousand candidates, especially for common or ambiguous patches. Out of this large subset of candidates, we choose a constant, small number of patches with diverse shapes. This approach reduces the number of labels to a manageable number while maintaining a sufficiently wide space of candidate shapes.

4.1 Matching appearance

Our appearance matching is based on comparison of standard 128-dimension SIFT descriptors to provide invariance to scale and rotation. The SIFT descriptor is computed at the scale and orientation of the keypoint. A training descriptor is said to match the test descriptor if the Euclidean distance between the descriptors is less than a fixed threshold t, where $t = 150$ in our experiments. Each descriptor dimension varied from $0 - 255$.

4.2 Choosing diverse shapes

After the appearance matching step, we have a set of candidates that look "close enough." We want the most diverse set of shapes that lie in this subset.

When creating the training set, we compute a shape descriptor for each patch by resampling the normal map to 32x32 pixels, applying PCA to find to dominant directions of variation in the 32^2D space, then keeping the principal components that account for 95% of the variance. This produces descriptors with $20 - 30$ components, depending on the training set.

At interpretation time, we pick a diverse subset of k candidates from the "close enough" set of N candidates. We pick the first candidate uniformly at random from the set of N. For the nth pick, we choose the candidate with the maximum minimum distance in descriptor space from the previous $n - 1$ selections. In other words, we choose the candidate farthest away from the closest previous pick.

4.3 Null candidates

In some cases the candidate selection process fails to find any shape patches suitable for a given keypoint. To handle these cases, we include a null or dummy label at each keypoint that provides no shape explanation but allows the inference to "bail out" if it cannot find an acceptable solution. The penalty for choosing this label should intuitively be the maximum error we are willing to tolerate in the interpretation. A negative aspect of allowing null candidates is that the shape interpretation can become disconnected. To keep the shape in one piece, we disallow null candidates for the finest scale keypoints.

5 Occlusion-aware compatibility scoring

Once a set of candidates is selected for each keypoint, we compute the likelihoods of each candidate and the compatibilites between candidates to set up the MRF inference step. Each candidate is scored by appearance match with the test image and the shape compatibility with the candidates at each neighboring keypoint. Shape compatibility scoring is involved because each patch may explain only part of its assigned area, and the context of the test image can affect the compatibility of two candidates even when their ownership masks do not overlap.

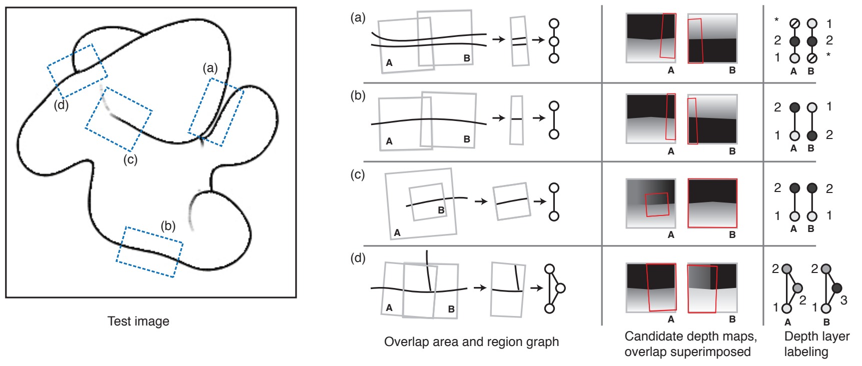
(a)
(b)
(c)
(d)
Test image
A
B
Overlap area and region graph
Candidate depth maps, overlap superimposed
Depth layer labeling

Fig. 5. Depth layer labeling. Left: test image with challenging areas outlined. Middle: contours of test image (black) define regions in the overlap area and a graph of connections between them. Right: the depth map of each candidate shape is used to assign an ordinal layer to each overlap region. The compatibility of the candidates is the compatibility of the graph labelings weighted by the size of the overlap regions. In (a), the two shapes are compatible because the overlap graph allows for a gap; (b), the two shapes are incompatible because they compete to label both sides of the single contour; (c), the large scale patch has a single depth layer in the entire patch, but two layers inside the overlap region; (d), at the T-junction the two labelings are slightly incompatible, but the incompatible labeling has small weight.

Consider Figure 5a: in order to avoid a combinatorial explosion in the necessary size of the training set, it must be possible for two separate shape patches to explain two nearby contours. Simultaneously, two separate patches should not claim opposite sides of the same contour (Figure 5b). The imperfect alignment of shape patches to image features makes exact matching of pixels unreliable. Instead, we propose a scoring mechanism based on the regions of the test image inside the overlap area (Figure 5, middle and right).

5.1 Overlap regions and graph

The regions of the overlap area O between two patches are found by applying image segmentation to the test image pixels restricted to the overlap area. As detailed below, our scoring metric is robust to oversegmentation and we only need a very basic segmentation algorithm. We currently edge detect, binarize, then flood fill (`bwlabel` in MATLAB) the overlap area to find regions.

Once we have regions O_i of the overlap area we construct a graph with a node for each O_i and edges between O_i that abut in the test image (Figure 5, middle). We say a candidate patch "owns" a node in this graph if more than 50% of the corresponding region's pixels are covered by the patch's ownership mask. Labeling is only compared between nodes that are either owned by both patches or that border nodes owned by both patches.

Each candidate patch assigns a labeling to O_i by constructing a local depth layer representation inside the overlap area. The local depth layers are constructed as follows: the depth map of the patch is masked by the overlap area and divided into regions R_i by the patch's occluding contours. The depth values inside each R_i are averaged, then the average values are sorted. The index j of R_j in the sorted list is the local depth layer. Each O_i takes the mean of the local depth layer pixels inside its region, rounded to the nearest integer.

5.2 Depth layer compatibility

Because the depth layer representation is ordinal, not metric, scoring the compatibility of two patches requires a fitting operation performed on the overlap regions O_i, as follows.

First we select a subset of the overlap regions O_i in which to compute compatibility. For depth layers, the important regions are those either owned by a shape patch or adjacent to an owned region, and other regions in the graph may be ignored (e.g., Figure 5a). Let OA_i be the regions owned by or bordering a region owned by patch A, and OB_i be the same for patch B. Then the subset of interest is:

$$\hat{O}_i = OA_i \cap OB_i \tag{4}$$

Next we define the label sets $\hat{A}_i$ and $\hat{B}_i$ using the layer labels assigned by the two candidates to $\hat{O}_i$, but compressed so that they have no gaps (i.e., [1 3] becomes [1 2]).

Finally, we perform a linear fit of $\hat{A}_i$ against $\hat{B}_i$ and vice versa. Define $\mathbf{b}$ as the label set to fit against and $\mathbf{a}$ as the other set. Define $\mathbf{w}$ as the vector of weights corresponding to the relative areas of $\hat{O}_i$. Then we solve

$$[\mathbf{aw}, \mathbf{w}]\mathbf{x} = \mathbf{b} \tag{5}$$

for the two-element parameter vector $\mathbf{x}$ in the least-squares sense. The best fit labels $\hat{\mathbf{a}}$ are found by multiplying through with $\mathbf{x}$. The score of the fitting is then

$$S_{ab} = \sum_{1..n}^{i} |\hat{a}_i - b_i| w_i \tag{6}$$

The final compatibility score is the max of the fitting scores in both directions.

5.3 Normal map compatibility

Normal map compatibility is defined as the mean L1 distance between the 3D normals of each patch. Because the ownership masks (areas where normals are defined) of the two patches will not in general align perfectly, we only compute the normal map compatibility inside the overlap regions O_i that are directly owned by both patches. Pixels inside an owned region but outside the ownership mask receive a value of $(0, 0, 0)$ for purposes of comparison.

5.4 Local appearance score

Scoring local appearance is straightforward. The basic idea is to define the score as the average difference in pixel value between the test image and the candidate patch, for pixels where ownership is nonzero. However, naive masking has the undesirable effect of benefiting patches with vanishingly small masks, since they have fewer chances to make errors. We compromise by computing the average error as usual, but rolling off to a constant "not explained" error if the area of the ownership mask is too small ($< 15\%$ of the patch area).

In addition, the average difference of pixel values is very vulnerable to slight misalignments in the matching patches, and misalignment of patches is the rule rather than the exception. To make our metric robust to (slight) misalignment, we first blur both the test and candidate patches with a Gaussian of $\sigma = r/3$, where r is the radius of the patch. See [16] for details of the rolloff and blur.

5.5 Prior probability score

The candidate selection process ensures that a diverse set of shapes are present at each keypoint. It is important to give greater weight to the more common shape candidates than the unusual ones to avoid inferring a possible – but very implausible – shape. Given a set of candidate patches C, we compute the prior probability of a patch P_i using kernel density estimation on the training set patches near P_i in shape descriptor space. The density is estimated by filtering the space of descriptors with an N-dimensional Gaussian with $\sigma = r/9$, where r is the distance between the farthest two patches in C. The density provides an unnormalized probability p_i, which we convert to a likelihood score:

$$l_i = -\texttt{log}\left(\frac{p_i}{\sum_{1..|C|}^{j} p_j}\right) \tag{7}$$

This prior estimate is simple but considerably improves the quality of results, especially for ambiguous stimuli such as line drawings.

6 Implementation and Results

Our approach produces an estimate of the surface normal at each point and an estimate of the rendering style of the test image. We present results on synthetic data rendered in six styles: line drawings with occluding and suggestive contours [17], diffuse shading with frontal illumination, glossy shading with frontal illumination, isotropic solid texture with no lighting, texture with diffuse shading, and texture with glossy shading.

6.1 Training and test set

The training data for our approach consists of many synthetic renderings of abstract blobs. The blobs are constructed by finding isosurfaces of filtered, 3D

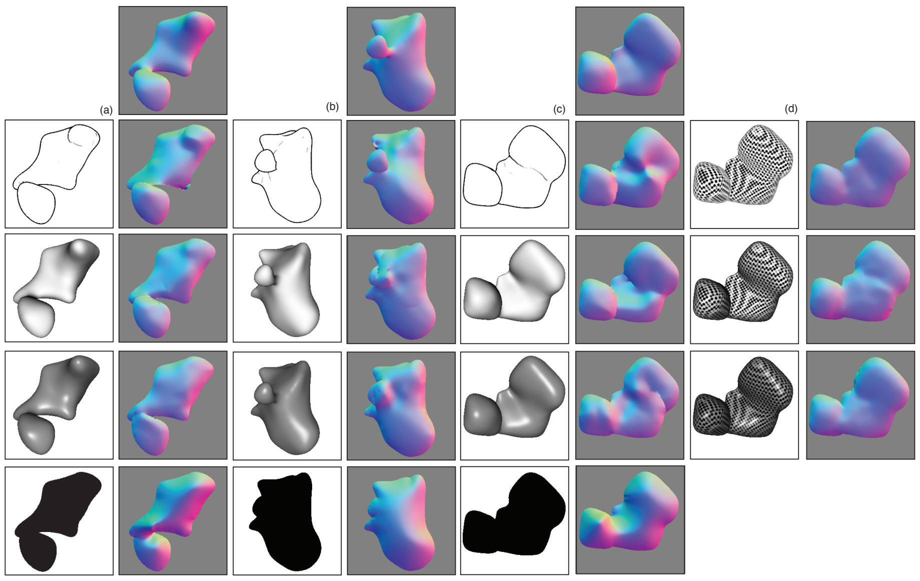

Fig. 6. Examples of normal map estimation for varying rendering styles. Top: ground truth normal map. (a,b,c): line drawing, diffuse and glossy shading with solid material, with interpreted normal map. (d): same as (c) but with texture only, diffuse and glossy shading with texture. Bottom: normal interpolation from shape boundaries.

white noise. We generate these shapes procedurally so that we can produce an arbitrary amount of training data. For each shape (and set of random camera positions), we render a depth map, a normal map, an image containing occlusion boundaries, and one image for each of the six rendering styles.

The training set itself is constructed by detecting keypoints in each rendered image (see Section 3) at four scale levels: radius 16, 32, 64, and 128 pixels. The input images were 300x300 pixels. We use a training set of $N = 96$ shapes, $k = 20$ cameras per shape, and 100-200 keypoints per rendered image, and six styles for a total of approximately 1.2m patches. The entire training set is <1GB, a modest size compared to other data-driven vision systems (e.g., [18]).

The test set is 10 shapes with the same blobby characteristics as the training set shapes. All shapes and renderings are available in [16].

6.2 Shape interpretation

Given a full set of candidate patches and likelihood scores between them, finding the most probable shape collage is a straightforward MRF inference task. We use min-sum loopy belief propagation for this purpose.

To produce a complete surface from the most likely shape collage, we fit a thin-plate spline to the most likely patches, taking care to allow the spline to split at occluding contours. An example collage and fit surface is shown in

Figure 2. To measure and visualize our results, we mask out background pixels using ground truth. Additional details of our inference and fitting steps may be found in [16].

6.3 Normal map estimation

The principal goal of our method is to estimate a normal map for the stimulus image. Figure 6 shows example normal maps computed by our system. Table 1 shows the average errors for our test set against ground truth.

For a baseline comparison we propose the following simple method, similar to that proposed by Tappen [19]: clamp the normals to ground truth values at the occluding contour, and smoothly interpolate the normals in the interior of the shape. This method produces smooth blobs (Figure 6, bottom). Note that while the interpolation method is very simple, it still is given ground truth normals at shape boundaries, whereas our method is not.

Overall our method produces accurate interpretations of the simuli, with average angular error between 20° and 26° for the shaded and textured stimuli, and 32° for line drawings (Table 1). In some cases, particularly the line drawings, the input stimulus is ambiguous, and our system proposes a plausible interpretation that is different from the original shape (e.g., Figure 6c). We also tested with training sets containing only patches of the same style as the input and found a small improvement in accuracy.

The running time for a single stimulus is approximately 20 minutes on a modern, multicore PC. The implementation is written entirely in MATLAB.

6.4 Style Prediction

The inferred patches can also be used trivially to approximate the stimulus image and thus make an estimate of the rendering style (Figure 7). We simply take the original appearance of each training patch and average it with the appearance of its neighbors. To make an estimate of the stimulus style, we simply find the style used by the majority of the selected shape patches. In our experiments the styles are sufficiently distinct that this estimate is very accurate.

Table 1. Average per image RMS error in degrees for normal estimates for each style, trained with all styles and only with the target style. Boundary interpolation does not vary between styles. Front-facing error is deflection from a front-facing normal.

	line	diffuse	glossy	only tex.	tex. + diff.	tex. + gloss	all
Training set w/ all styles	32	20	26	23	21	23	24
Train. set w/ only target style	30	20	25	23	21	23	24
Boundary interpolation	-	-	-	-	-	-	39
Front-facing	-	-	-	-	-	-	46

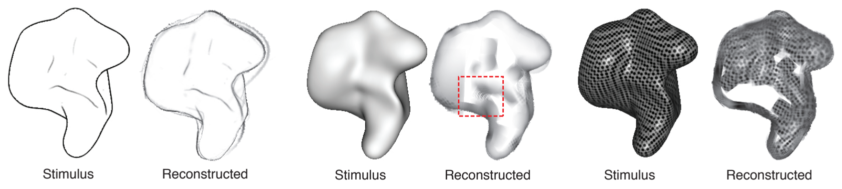

Fig. 7. Reconstructing the stimulus from patch appearance. Reconstructed patches are usually faithful to the original rendering, though are sometimes confused (boxed). The set of styles used for reconstruction gives an estimate of the stimulus style.

6.5 Discussion

So far we have experimented with synthetic datasets. The renderings were chosen to span a range of styles that challenge parametric interpretation systems. Suggestive contours, for example, have been shown to mimic lines drawn by artists [20], but are often disconnected and noisy. Disconnected lines violate the major assumption of shape-from-line systems (e.g., [6]) that the line drawing be a complete, connected graph. The glossy, textured stimuli violate the Lambertian assumptions of even recent work on shape-from-shading (e.g., [21]). Our system can interpret all six styles with no modification and a single training set.

There are several limitations to the current method that must be overcome in order to process real-world stimuli. Most importantly, we need a rich training set of photographs and associated 3D shape. Such data may become more commonplace as 3D acquisition hardware becomes more robust and easy to use. Also, realistic computer graphics renderings may provide an accurate enough approximation of real photographs to provide training data (similar to [22]).

Some aspects of the algorithm itself would also require extension. The appearance matching step (Section 4.1), for example, matches based on the appearance of the entire patch. This approach works well when the background is empty, such as our stimuli, but can become confused by noise or distracting elements. An extended method could match appearance based on the ownership masks of training set patches. The appearance scoring metric (Section 5.4) would also need to be adapted to more general stimuli.

7 Conclusion

We have presented an approach to infer a normal map from a single image of a 3D shape. The treatment of occlusion between patches and the irregular, interest-driven patch placement that we introduce dramatically reduce the complexity of example-based shape interpretation, allowing our method to interpret images with multiple layers of depth and self-occlusion using a moderately sized training set. Because it is data-driven, our method can interpret ambiguous stimuli such as sparse line drawings and complex stimuli such as glossy, textured surfaces, and it uses the same machinery in all cases. To our knowledge, this property is unique among current vision systems.

Acknowledgements

We thank Ce Liu and Jon Barron for helpful comments. This material is based upon work supported by the National Science Foundation under Grant No. 1111415, by ONR MURI grant N00014-09-1-1051, by NIH grant R01-EY019262 to E. Adelson, and a grant from NTT Research Labs to E. Adelson. P. Isola is supported by an NSF graduate research fellowship.

References

1. Roberts, L.: Machine perception of three-dimensional solids. dspace.mit.edu (1963)
2. Durou, J., Falcone, M., Sagona, M.: Num. methods for shape-from-shading: A new survey with benchmarks. Comp. vis. and img. understanding **109** (2008) 22–43
3. Malik, J., Rosenholtz, R.: Computing local surface orientation and shape from texture for curved surfaces. IJCV **23** (1997) 149–168
4. Hassner, T., Basri, R.: Example based 3d reconstruction from single 2d images. CVPR Workshops: Beyond Patches (2006) 1–8
5. Saxena, A., Sun, M., Ng, A.: Make3d: Learning 3d scene structure from a single still image. PAMI **31** (2009) 824–840
6. Malik, J.: Interpreting line drawings of curved objects. IJCV (1987)
7. Wang, Y., Chen, Y., Liu, J., Tang, X.: 3d reconstruction of curved objects from single 2d line drawings. CVPR (2009) 1834–1841
8. Xue, T., Liu, J., Tang, X.: Symmetric piecewise planar object reconstruction from a single image. CVPR (2011) 2577–2584
9. Saund, E.: Logic and mrf circuitry for labeling occluding and thinline visual contours. In: Neural Information Processing Systems (NIPS). (2005)
10. Ouyang, T., Davis, R.: Learning from neighboring strokes: combining appearance and context for multi-domain sketch recognition. In: NIPS. (2009)
11. Freeman, W., Pasztor, E., Carmichael, O.: Learning low-level vision. International Journal of Computer Vision **40** (2000) 25–47
12. Fitzgibbon, A.W., Wexler, Y., Zisserman, A.: Image-based rendering using image-based priors. In: Intl. Conf. Computer Vision (ICCV). (2003)
13. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from a single depth image. In: CVPR. (2011)
14. Harris, C., Stephens, M.: A combined corner and edge detector. Alvey vision conference **15** (1988) 50
15. Lowe, D.: Object recognition from local scale-invariant features. ICCV (1999)
16. : Supplemental Material: http://people.csail.mit.edu/fcole/shapecollage. (2012)
17. DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., Santella, A.: Suggestive contours for conveying shape. ACM Trans. Graph. **22** (2003) 848–855
18. Hays, J., Efros, A.: Im2gps: estimating geographic information from a single image. CVPR (2008) 1–8
19. Tappen, M.: Recovering shape from a single image of a mirrored surface from curvature constraints. CVPR (2011) 2545–2552
20. Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S.: Where do people draw lines? SIGGRAPH (2008)
21. Barron, J.T., Malik, J.: High-frequency shape and albedo from shading using natural image statistics. In: CVPR. (2011)
22. Kaneva, B., Torralba, A., Freeman, W.: Evaluation of image features using a photorealistic virtual world. ICCV (2011)