Supplementary material for 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

1 Local appearance score

As the paper explains, the basic idea is to define the local appearance score as the average difference in pixel value between the test image and the candidate patch, for pixels where ownership is nonzero.

In general the average difference of pixel values is very vulnerable to slight misalignments in the matching patches. In our patch-based framework, misalignment is the rule rather than the exception. To make our metric robust to (slight) misalignment, we first blur both the test and candidate patches by a Gaussian G with $\sigma = r/9$. The score is then

$$S_l = \texttt{mean}(|G(test) - G(cand)| * G(ownership)) \tag{1}$$

where the ownership mask is blurred with the same blur kernel to create a soft weighting mask.

The score should only be computed for the 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 k_{l0} if the area of the ownership mask A is too small:

$$S_l = \texttt{rolloff}(A)S_l + (1 - \texttt{rolloff}(A))k_{l0} \tag{2}$$

where

$$\texttt{rolloff}(A) = \begin{cases} 1 & : A > 0.15 \\ 0.5(1 - \cos(\frac{A-0.05}{0.1}\pi)) & : A < 0.15 \end{cases} \tag{3}$$

The constant k_{l0} is also the error we assign to a null shape candidate (paper Section 4.3)

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 loopy belief propagation for this purpose.

In order to produce a complete surface from the most likely collage, we fit a thin-plate spline to the most likely patches, taking care to allow the spline to split at occluding contours.

2.1 Inference

To find the most likely shape interpretation we use standard min-sum loopy belief propagation [1] on the MRF defined by the keypoint graph. We use $n_c = 20$ labels at each keypoint, where the label values simply index the shape candidates.

The data term $\mathbf{d}$ is defined for each of the n_v keypoints and is a vector of length n_c where:

$$d_i = L_i k_l + P_i k_p \tag{4}$$

where L_i is the local appearance score, P_i is the prior probability score, and k_l and k_p are constants weighing the contribution of each term. Empirically we use $k_l = 2$ and $k_p = 0.05$.

The compatibility or smoothness term $\mathbf{C}$ is defined for each of the n_e edges and is a $n_c \times n_c$ matrix where:

$$C_{i,j} = D_{i,j} k_d + S_{i,j} k_s \tag{5}$$

where $D_{i,j}$ is the depth layer compatibility score, $S_{i,j}$ is the shape compatibility score, and k_d and k_s are constant weighting terms. Empirically we use $k_d = 1$, $k_s = 0.5$.

For null candidates, we use the data term $d_{null} = k_{l0} k_l$, where k_{l0} is the maximum appearance error (Section 1) and k_l is the appearance weighting term. We currently set the compatibility terms $C_{i,null} = 0$ and $C_{null,null} = 0$.

Because LBP is not guaranteed to converge to the globally most likely solution, we run the optimization multiple times with randomized starting conditions. We randomize the initial messages within the range $[0, \mathtt{max}(\mathbf{d}) + \mathtt{max}(\mathbf{C})]$. We also apply a dampening coefficient $\alpha = 0.2$ to the message updates.

2.2 Surface Fitting

Given the most likely shape candidates we fit a surface that interpolates them as closely as possible. We formulate the surface as a linear system over the depths at each pixel. The patch normals are soft constraints, and can be overridden by other normals or the smoothness term. Solving for depth, rather than normals, circumvents the problem of enforcing integrability of the result. In this scheme, we use depth gradients to represent normals, and thin-plate bending energy for smoothness. The system is defined as $(\mathrm{WA})\mathbf{x} = (\mathrm{W}\mathbf{b})$, where A is an $m \times n$ matrix ($m > n$), W is a diagonal $m \times m$ weight matrix, $\mathbf{b}$ is an m-element vector of constraint values, and $\mathbf{x}$ is an n-element vector of depth values. We solve for the best-fitting $\mathbf{x}$ in the L2 sense.

For notation, let $p \in (1..n)$ represent the pixel in question, $p_{1,0}$ represent the pixel immediately to its right, $p_{0,1}$ represent the pixel immediately above, etc.

Since each pixel can be covered by multiple patches, we have multiple normal vector estimates per pixel. Each pixel receives one gradient constraint from each estimate. To constrain the gradient at p to the gradient given by patch i for this pixel, $\{u_p^{(i)}, v_p^{(i)}\}$, we add the forward difference equations to the system:

$$\begin{aligned} \mathbf{x}_{p_{1,0}} - \mathbf{x}_p &= u_p^{(i)} \\ \mathbf{x}_{p_{0,1}} - \mathbf{x}_p &= v_p^{(i)} \end{aligned} \tag{6}$$

We weight each constraint according to the fit of patch i in the MAP configuration, as well as the size of this patch and the pixel p's location within the patch. Thus, the weight of a gradient constraint from patch i at pixel p is given by,

$$w_p^{(i)} = \frac{1}{a_i} \exp(-E_i) \frac{4}{\sqrt{2\pi a_i}} \exp(-\frac{8\|\mathbf{x} - \mathbf{c}_i\|^2}{a_i}) \tag{7}$$

$$E_i = d_i + \sum_j C_{i,j} \tag{8}$$

where a_i is the pixel area of patch i, and the second exponential represents a Gaussian mask about the center, $\mathbf{c}_i$, of the patch. The patch fit term, $\exp(-E_i)$, is proportional to the conditional probability, according to our MRF, of the patch given image data and given all other patches in the MAP configuration.

The smoothness of the solution is maintained by minimizing the thin-plate bending energy (second derivative of depth) across the shape by adding the following equations at each p:

$$\begin{aligned} \mathbf{x}_{p_{-1,0}} - 2\mathbf{x}_p + \mathbf{x}_{p_{1,0}} &= 0 \\ \mathbf{x}_{p_{0,-1}} - 2\mathbf{x}_p + \mathbf{x}_{p_{0,1}} &= 0 \\ 2(\mathbf{x}_p - \mathbf{x}_{p_{1,0}} - \mathbf{x}_{p_{0,1}} + \mathbf{x}_{p_{1,1}}) &= 0 \end{aligned} \tag{9}$$

In order to allow the surface to be discontinuous at occluding contours, we weight the smoothness constraints based on inferred occluding contours. We construct an image of occluding contours by taking the weighted average of the occluding contour channels of our MAP patches, using the same weights as in 7. We then filter to remove noise. In addition to using this image to weight smoothness, we completely exclude all constraints along the ridges of this image (ridges found by thresholding and morphologically thinning). Normals along these ridges are assigned by copying over estimated normals from adjacent to the ridge (for each ridge pixel, the neighbor with minimum depth value is selected for copying).

To measure and visualize our results, we mask out background pixels using ground truth.

3 Training Data

The training data consists of 1920 sets of 9 images each. The nine images include three types of shape information: depth map, normal map, occluding contours; and six types of renderings: lines (including suggestive contours), diffuse shading, glossy shading, texture only, texture with diffuse, texture with glossy. See Figure 1 for several examples. The texture we use is a solid texture consisting of regular, stacked black spheres on a white background.

4 Full results

In Figures 2 and 3, we display the inferred appearance and normal map for all 120 of our test image (10 test shapes x 6 styles x 2 training sets).

References

1. P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on DOI - 10.1109/CVPR.2004.1315041*, 1:I–261– I–268 Vol.1, 2004.

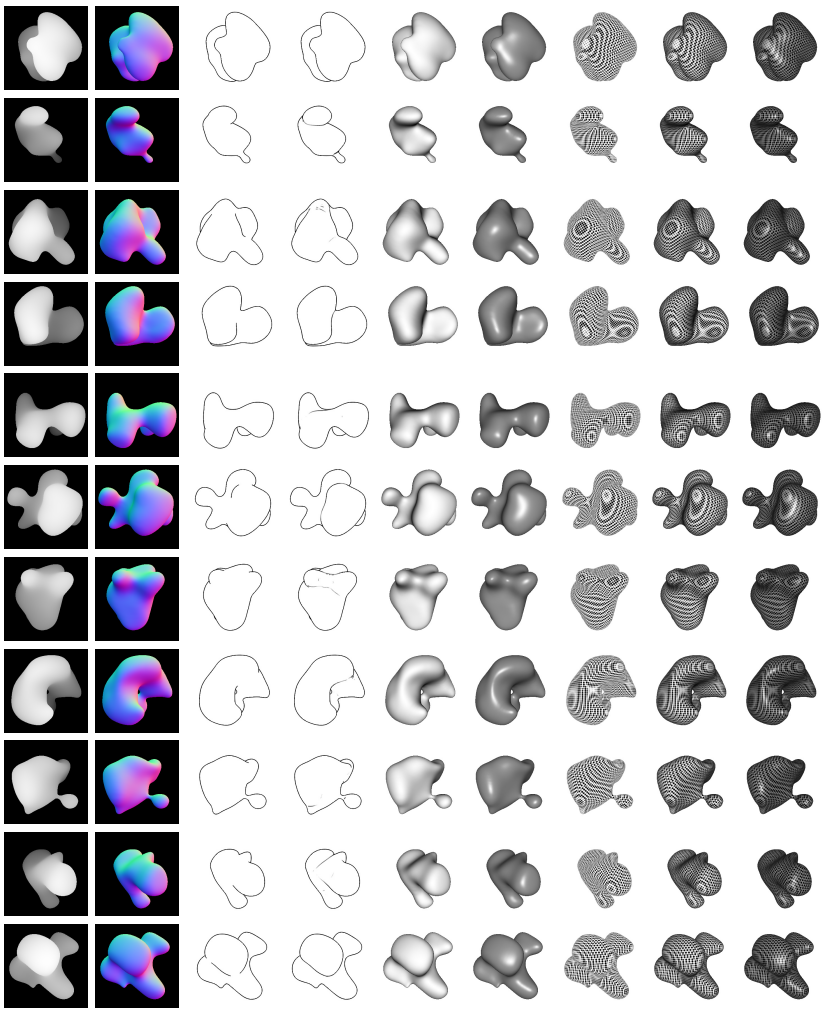

Fig. 1. Example training images.

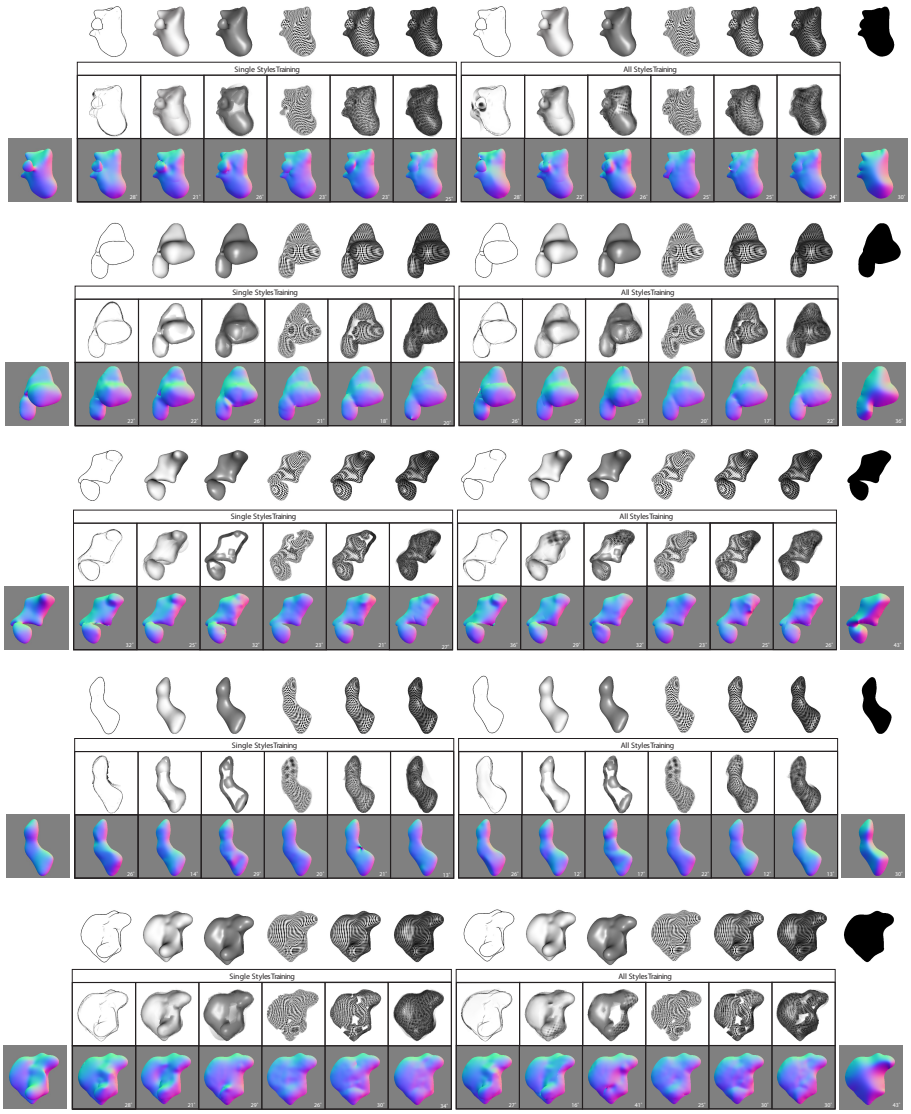

Fig. 2. Full test set results, part 1 (NOTE: the images are small but full resolution. Try zooming in a PDF reader). Left-most column is original shape. Right-most column is inflation from shape boundary. Middle columns show image to be interpreted (row 1 in each shape block), interpreted appearance (row 2), and interpreted normal map (row 3) for shapes rendered in six styles and under two types of training – 'same style' renderings and 'all styles' of renderings. Number in lower-right-hand corner of normal maps gives root mean squared angular errors between interpreted normals and normals of original shape.

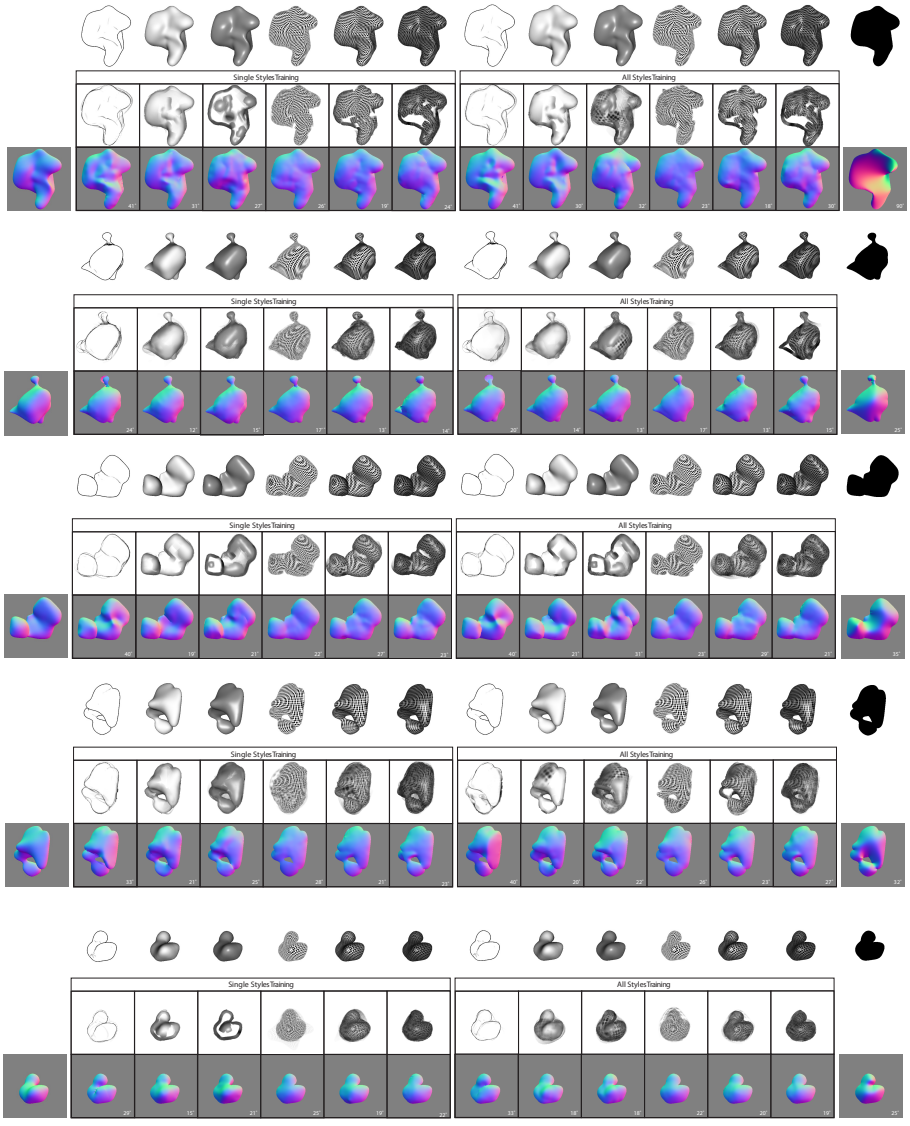

Fig. 3. Full test set results, part 2. See caption in Figure 2