

# Shading Annotations in the Wild (Supplementary Material)

Balazs Kovacs

Sean Bell

Noah Snavely

Kavita Bala

Cornell University

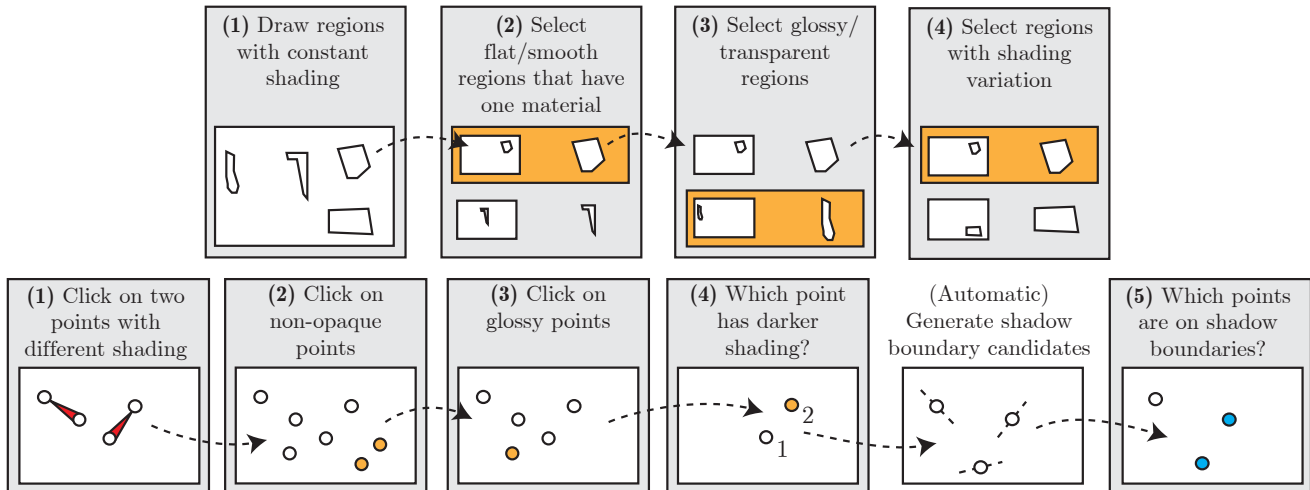


Figure 1: Overview of our crowd-sourcing pipelines. **First row:** Region annotation pipeline, as described in Section 2.2. **Second row:** Point annotation pipeline, as described in Section 2.3.

## 1. Overview

In this supplementary material, we include additional details and results which could not be included in the main paper due to space restrictions. We present:

- The SAW crowd-sourcing pipeline (Sec. 2)
- Training parameters (Sec. 3)
- Dataset statistics (Sec. 4)
- Smooth shading heatmap predictions (Sec. 5)
- Comparison of [Bell *et al.* 2014] [2] and [Zhou *et al.* 2015] [4] intrinsic image decomposition algorithms (Sec. 6)
- Intrinsic image decompositions with and without our smooth shading prior (Sec. 7)

## 2. Crowd-sourcing Pipeline

In this section we present detailed description of all crowd-sourcing tasks we used to collect our constant shading regions and point annotations. See Figure 1 for an overview of our pipelines.

### 2.1. Glossy Annotations

Our goal is to curate a dataset that is generally useful to tasks that use shading estimation. One specifically important task is intrinsic image decomposition. Since the majority of intrinsic image decomposition algorithms make a Lambertian assumption, and disregard specular surfaces, we want to eliminate points that have a substantial glossy component. Unfortunately, the IIW [2] dataset did not take this into account. We address this limitation of IIW by annotating all old and new points in the IIW dataset with a “glossy” label.

### 2.2. Region Annotation Pipeline

To collect constant shading regions, we built a four step crowd-sourcing pipeline. First, a handful of selected workers were asked to draw polygons around constant shading regions in a scene (step 1, see Figure 2). To get higher quality data, we also told workers that these regions have to be opaque, non-glossy, and on a flat surface without bumps. Further, the regions had to have a single type of material (*e.g.* wood or plastic), and we did not allow fabrics, since they usually have wrinkles or bumps. Next, we

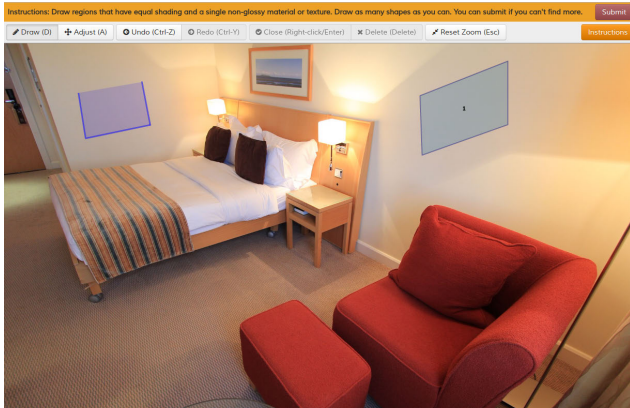


Figure 2: *Region pipeline step 1: Draw regions with constant shading.* Workers were asked to draw polygons around constant shading regions. In the example, a worker has finished drawing the blue region (1) on the right and drew three sides of the region on the left.

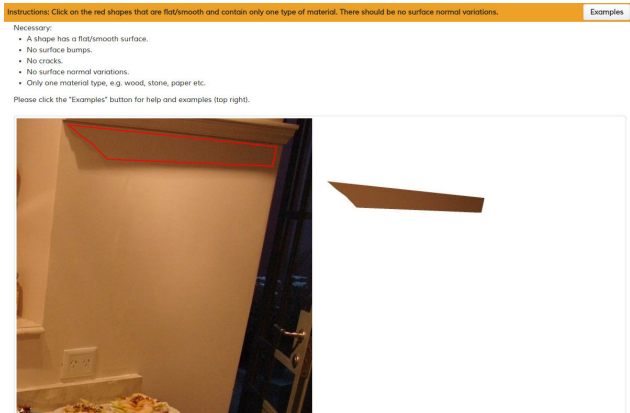


Figure 3: *Region pipeline step 2: Select flat/smooth regions that have one type of material.* We collected 5 votes for each region.

filtered these regions with 3 tasks (steps 2, 3 and 4, see Figures 3, 4, 5): (1) click on regions that are flat/smooth and contain only one type of material (we kept these), (2) click on regions that are glossy/transparent (we discarded these), and (3) click on regions that have shading variation (we discarded these). In each filtering task, workers were shown a list of regions, and they had to select the ones that met the specified criteria.

### 2.3. Point Annotation Pipeline

Our point annotation pipeline has five crowd-sourcing steps. Figures 6 through 10 show illustrative examples from each step of the pipeline. First, workers were asked to create shading comparisons by clicking on pairs of points that have different shading (step 1, see Figure 6). Next, we filter out

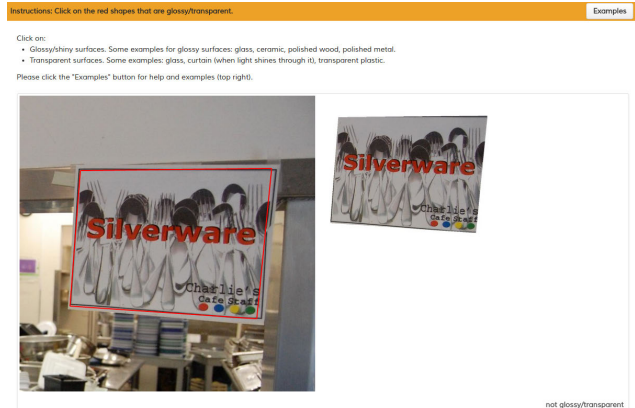


Figure 4: *Region pipeline step 3: Select glossy/transparent regions.* We collected 5 votes for each region.

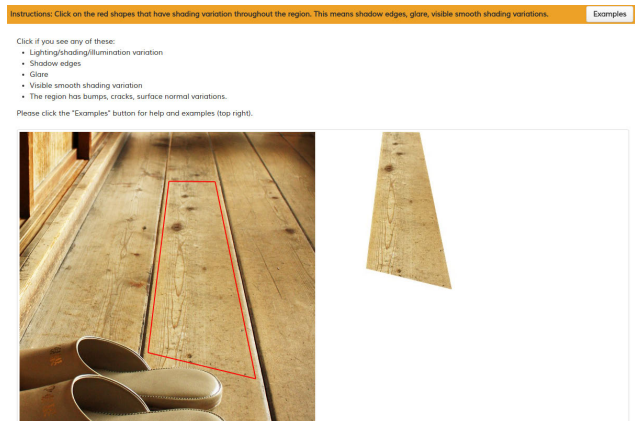


Figure 5: *Region pipeline step 4: Select regions with shading variation.* We collected 5 votes for each region.

non-opaque or glossy points through two tasks (steps 2 and 3, see Figures 7, 8). For each remaining point pair (i.e., pairs where neither point was discarded), workers were asked which of the two points has darker shading (step 4, see Figure 9). Finally, after generating candidate shadow boundary points based on these comparisons, we asked workers to choose points that are truly on sharp shadow boundaries and not on normal/depth discontinuities (step 5, see Figure 10).

After collecting the data, we found a minor mistake in our tutorial for the shading comparison task (step 4, see Figure 9). The mistake was that we inverted which point had darker pixel color in the image for an example. See Figure 11 for this particular point pair shown in the tutorial example. We believe this did not have a noticeable effect on data quality, because (1) the mistake was not in comparing shading, but in the observation of pixel colors; (2) workers had to successfully pass a series of tests (sample tasks with known ground truth) in the tutorial to start work on our tasks; (3) we used 6 votes for each point pair.

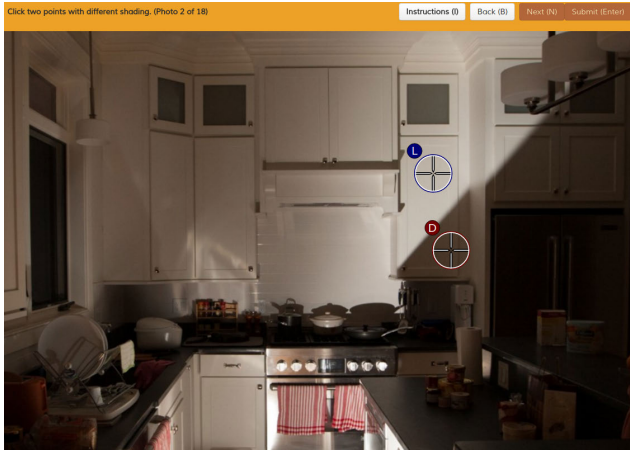


Figure 6: *Point pipeline step 1: Click on two points with different shading.* Workers were asked to create point pairs where the first point has darker shading than the second. The red “D” cross represents the darker shading point and “L” cross represents the lighter shading point in the image.

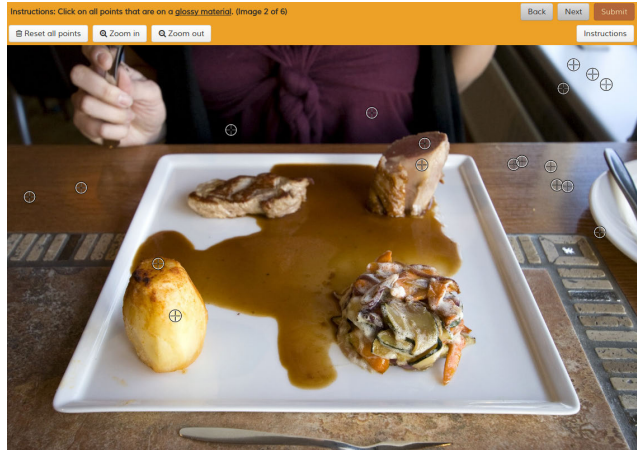


Figure 8: *Point pipeline step 3: Click on glossy points.* Workers were asked to select points that are glossy. We discard such glossy points for the remainder of the pipeline. We collected 5 votes for each point. Candidate points are represented with crosses.

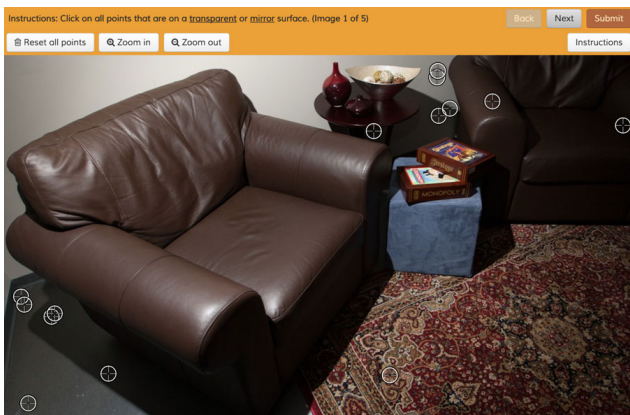


Figure 7: *Point pipeline step 2: Click on non-opaque points.* Workers were asked to select points that are not opaque. We discarded these points before proceeding to the next step of the pipeline. We collected 5 votes for each point. Candidate points are represented with crosses.

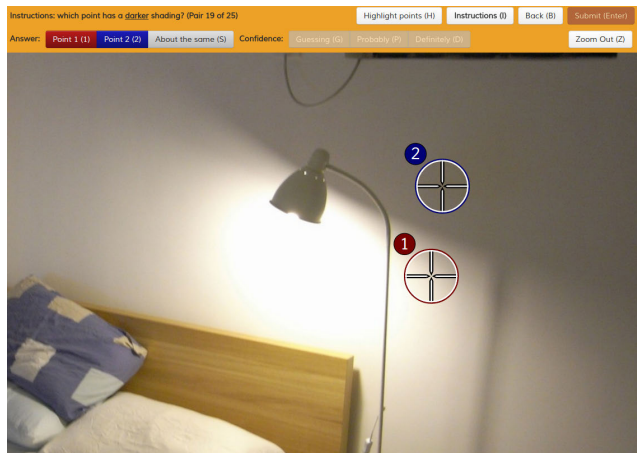


Figure 9: *Point pipeline step 4: Choose which point has darker shading.* Workers were shown point pairs (shading comparisons) and were asked to select the point with darker shading. We collected 5 additional votes for each point pair. The two points are represented with a red (point 1) and blue (point 2) cross.

### 3. Training Parameters

We used the Caffe [3] deep learning framework for our experiments. The training parameters are the following:

- Solver: SGD.
- Learning rate: 0.001.
- Learning rate policy is “step” with step size 5000 and gamma 0.5.
- Momentum: 0.9.
- Weight decay: 0.0005.
- Gradient accumulation: every 2 iterations (this is

called `iter_size` in Caffe [3]).

- Final iteration: 17500.
- Class balance: 2:1:1 (S : NS-ND : NS-SB).
- Batch size: 20 images.
- Points sampled per image: 60.

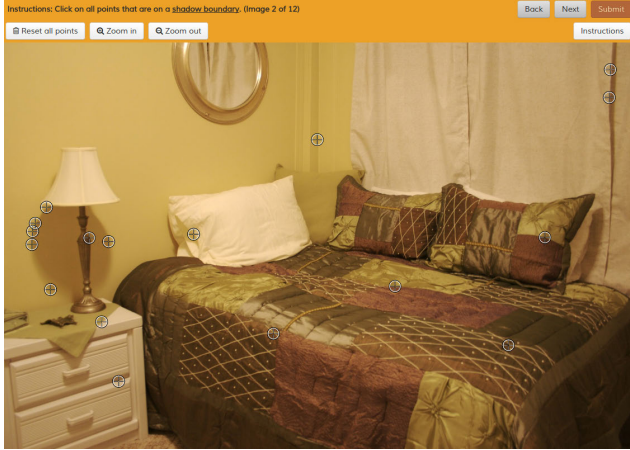


Figure 10: *Point pipeline step 5: Click on points that are on a shadow boundary.* We generate candidate shadow boundary points based on image gradients between valid non-equal shading point pairs. Workers were asked to select points that are on a sharp shadow boundary. We collected 5 votes for each point. Candidate points are represented with crosses.

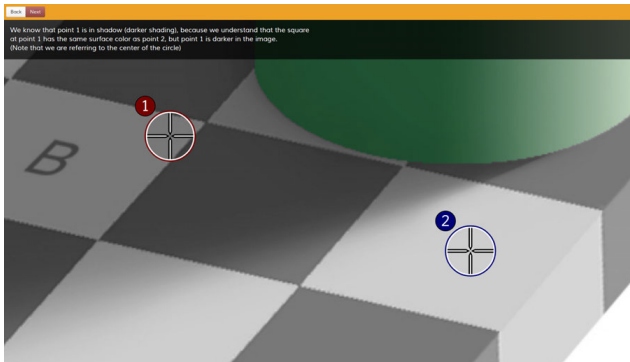


Figure 11: The corrected tutorial example.



Figure 12: Probability-color mapping for heatmaps. The heatmaps themselves are on the next few pages.

## 4. Dataset Statistics

In this section, we present additional statistics about our dataset. In Figure 13-(a) we show a joint plot of the log average color gradient magnitude over each constant shading region and the log normalized area (1 means that the region covers the entire image). The gradient magnitude is correlated with how textured the region is. Textured regions are valuable because constant shading cannot be easily predicted based on simple pixel intensity measurements.

As Figure 13-(b) shows, most images have 0-4 constant shading regions. This is expected, because constant shading is not very common in arbitrary indoor images.

Finally, based on the statistics of shadow boundary points (see Figure 13-(c)), we can say that the majority of images have 0-5 shadow boundary points. Similarly to constant shading, sharp shadow boundaries are relatively rare in images.

On the other hand, the variety of the annotations is more important than having many redundant annotations in a few images. Our dataset contains 15,407 shadow boundary points and 23,947 constant shading regions over thousands of photos.

## 5. Smooth Shading Heatmap Predictions

We show 204 examples of smooth shading heatmap predictions randomly sampled from the test set below. See Figure 12 for the probability-color mapping for all heatmaps. We release the full set of predictions online along with our trained model.

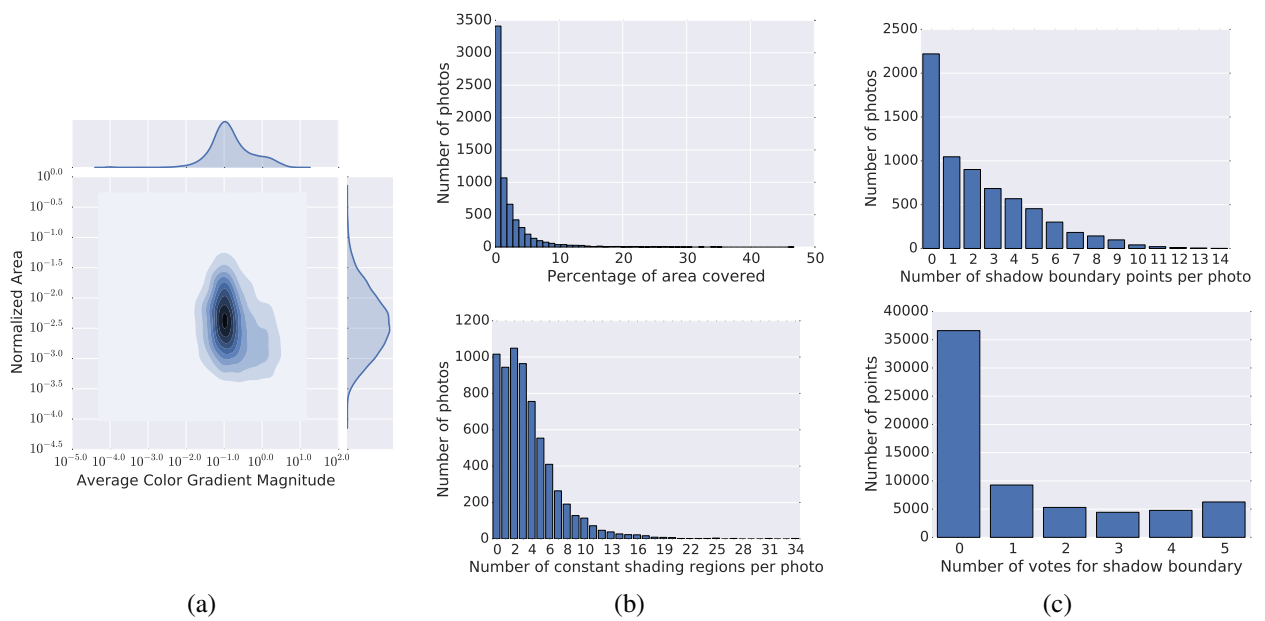
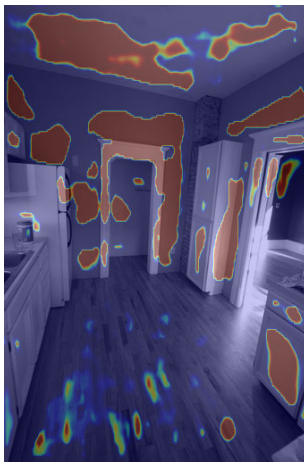
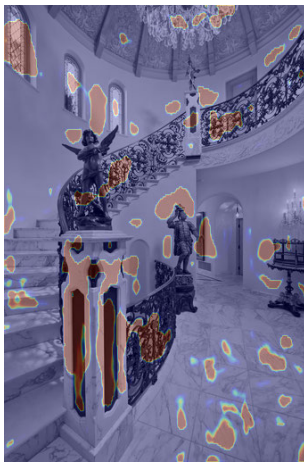
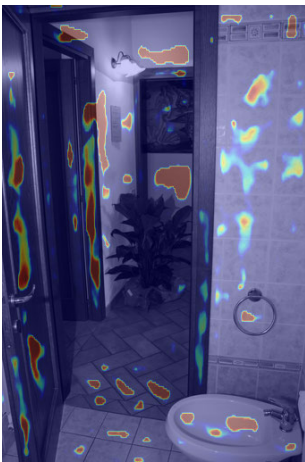
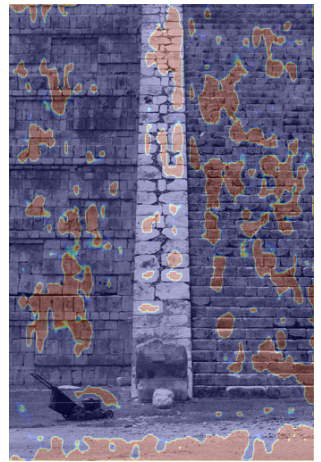
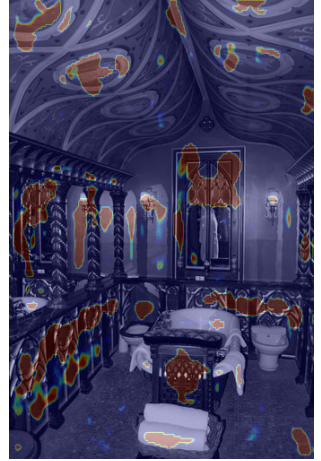
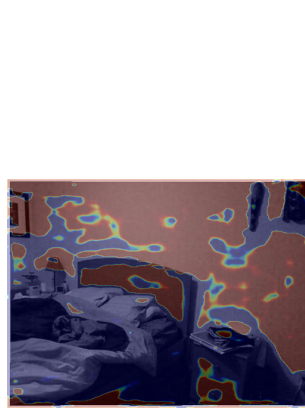
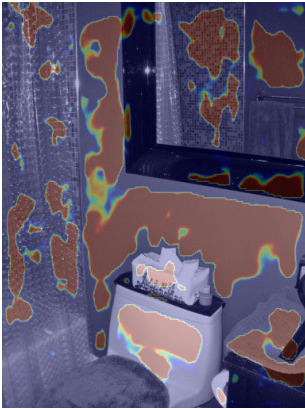
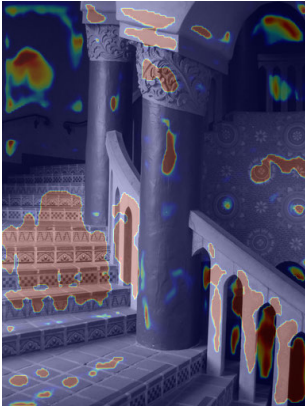
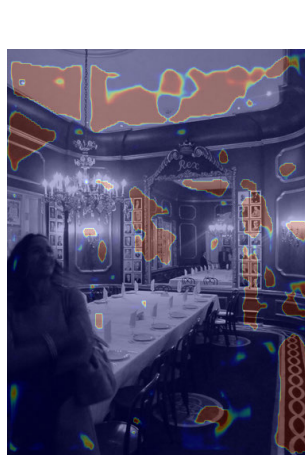
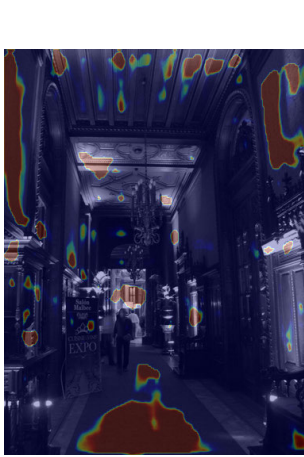
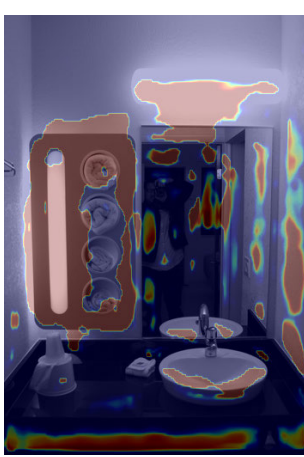
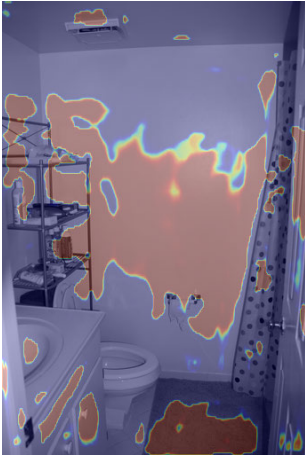
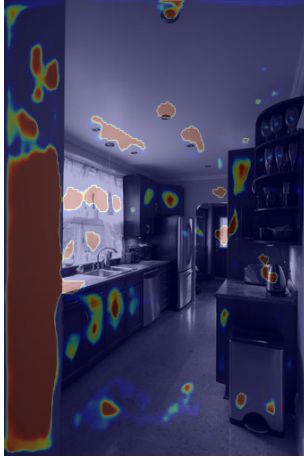
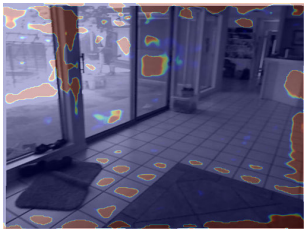
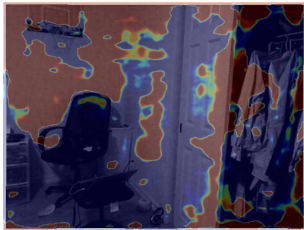
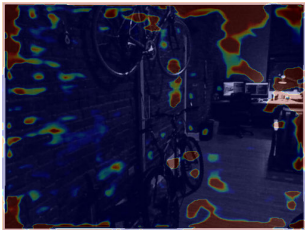
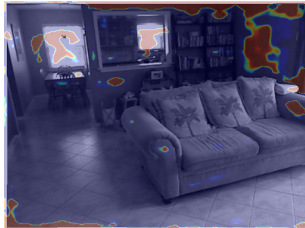
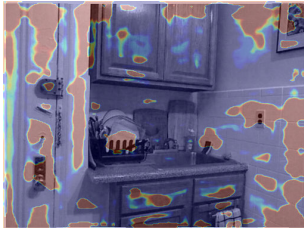
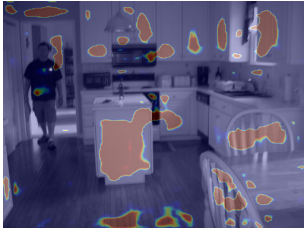
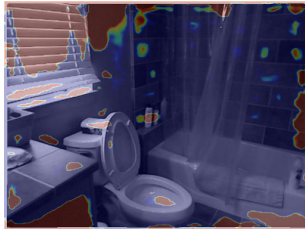
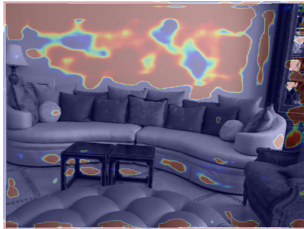
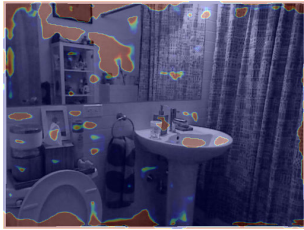
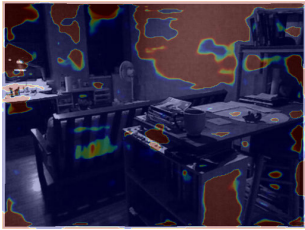
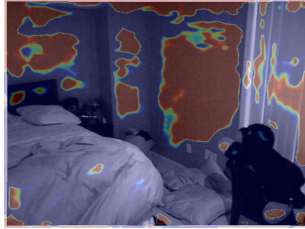
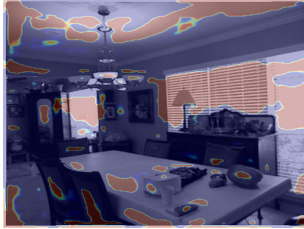
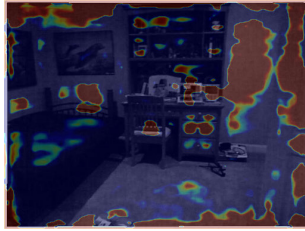
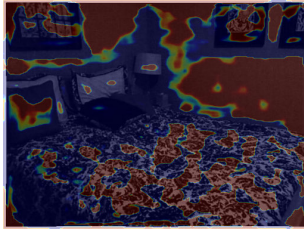
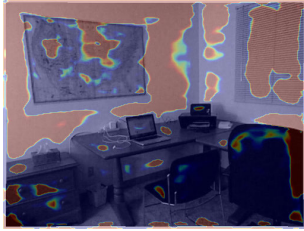
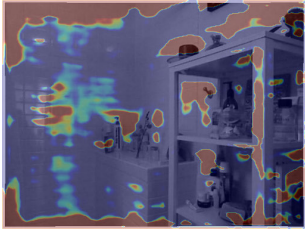


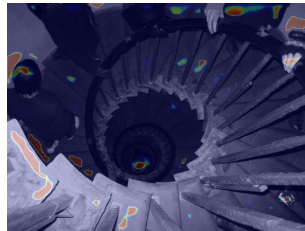
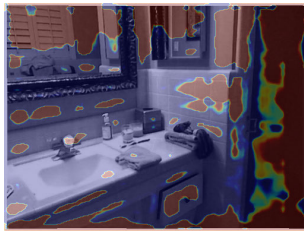
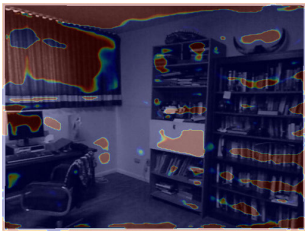
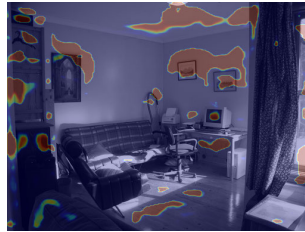
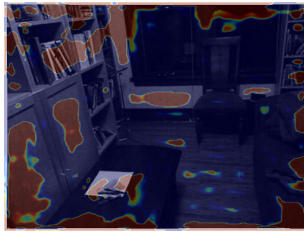
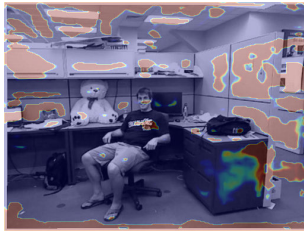
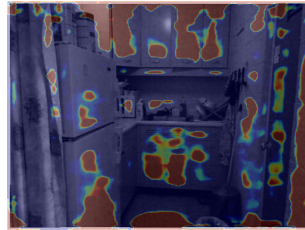
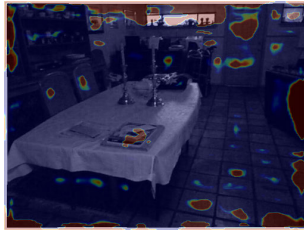
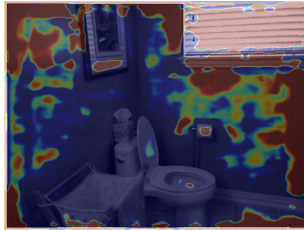
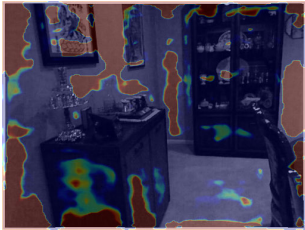
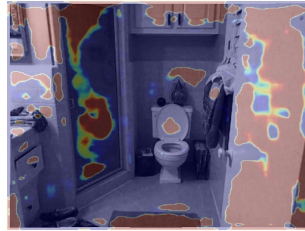
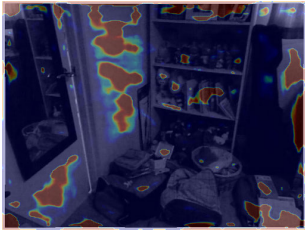
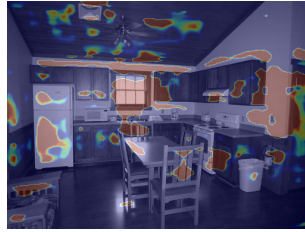
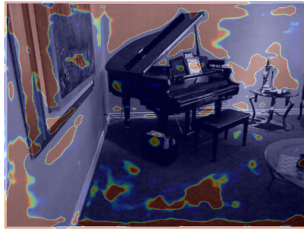
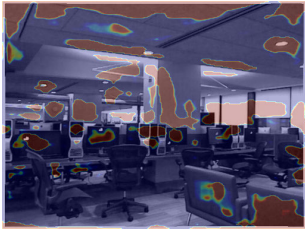
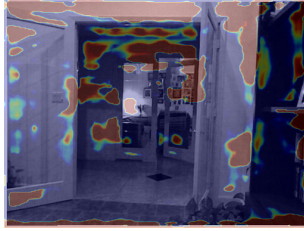
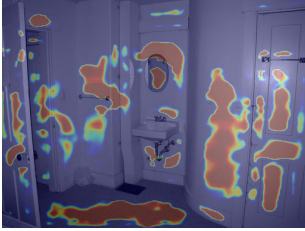
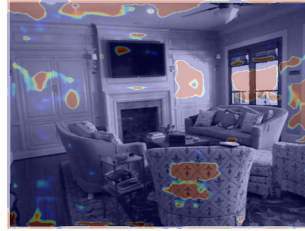
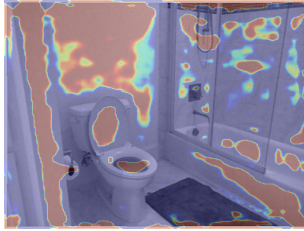
Figure 13: **(a)** Joint plot of the log average color gradient magnitude over each constant shading region and the log normalized area (1 means that the region covers the entire image). **(b)** Histogram of percentage of image area covered per photo (top) and number of constant shading regions per photo (bottom). **(c)** Statistics of shadow boundary points.

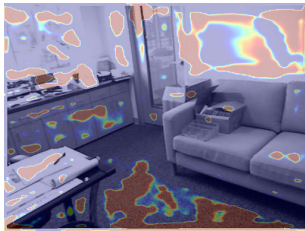
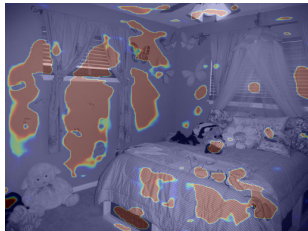
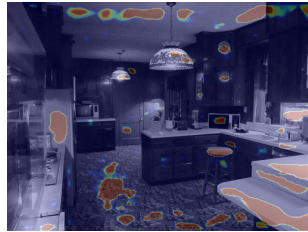
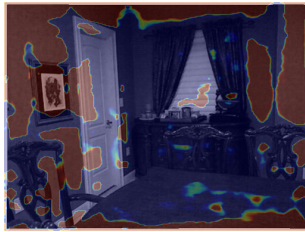
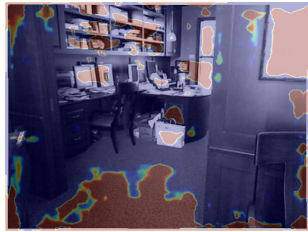
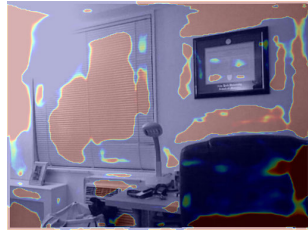
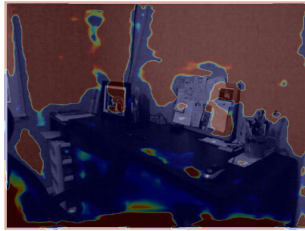
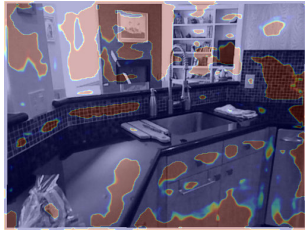
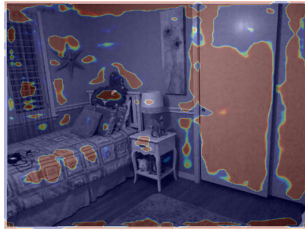
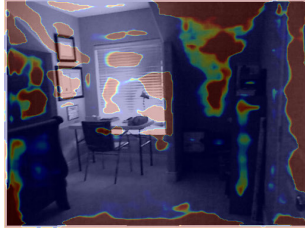
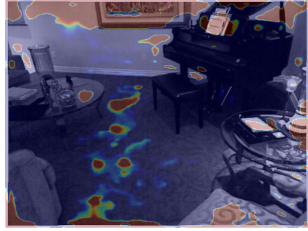
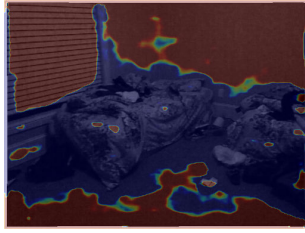
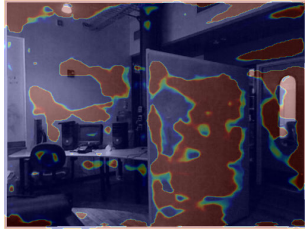
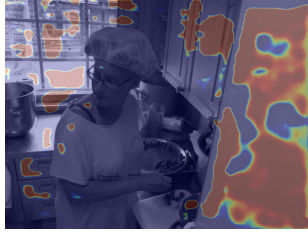


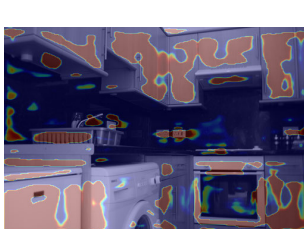
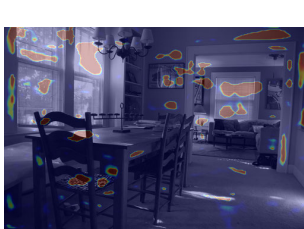
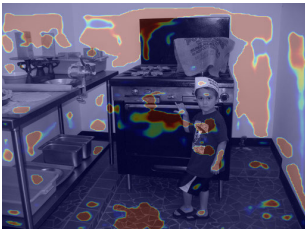
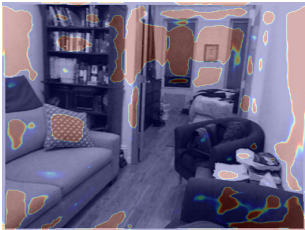
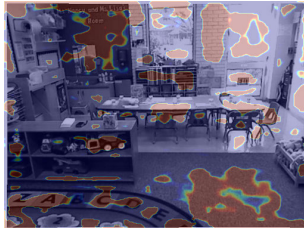
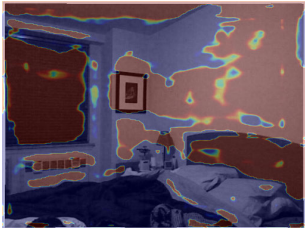
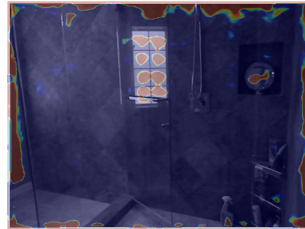
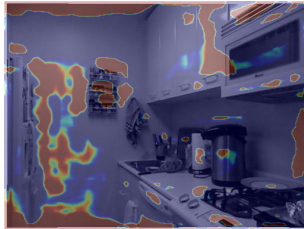
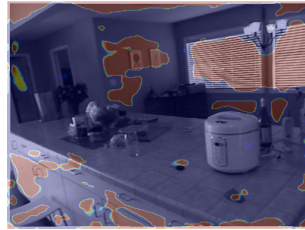
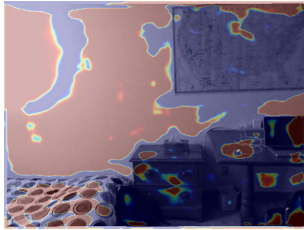
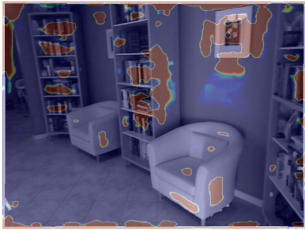
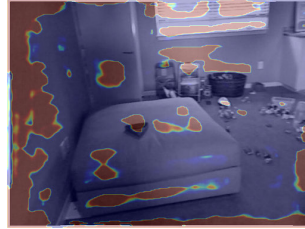
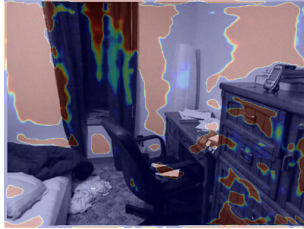
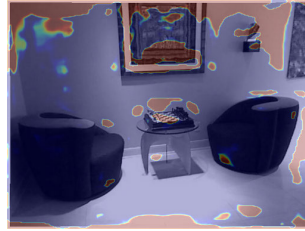
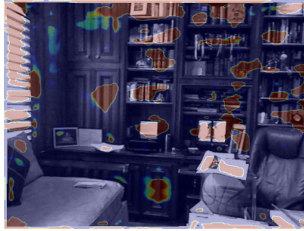
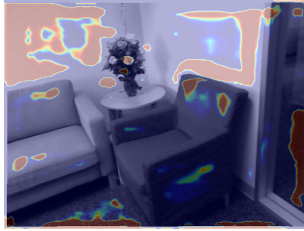
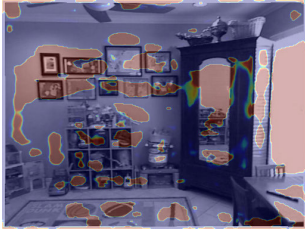




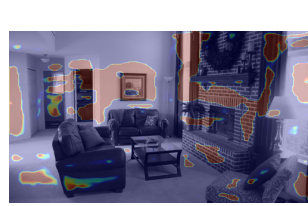
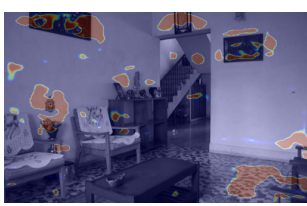
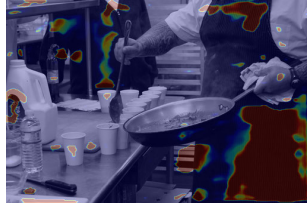
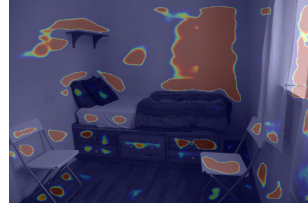
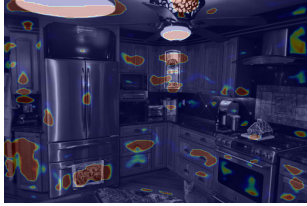
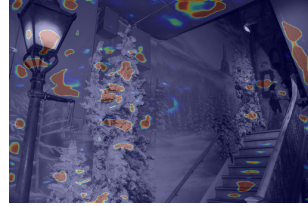
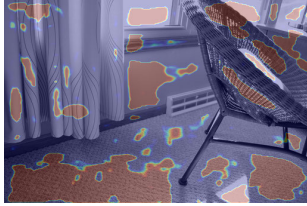
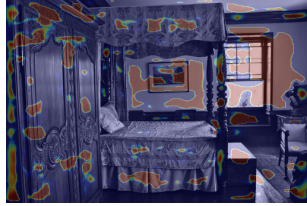
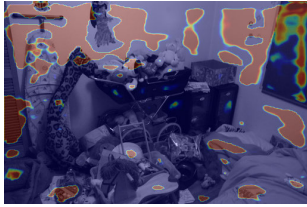
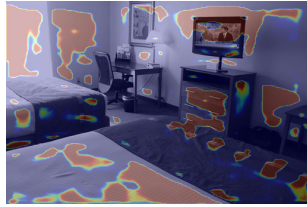
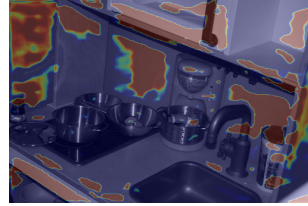
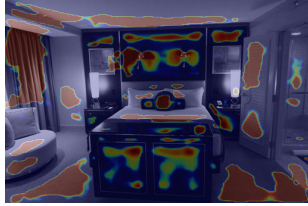
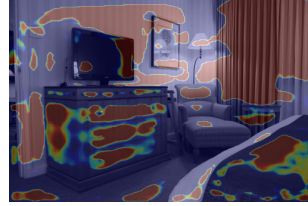
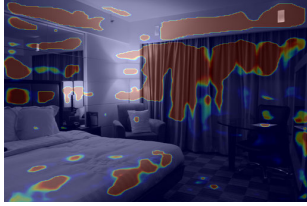
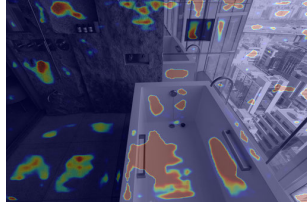
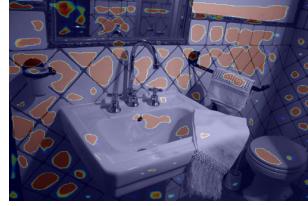
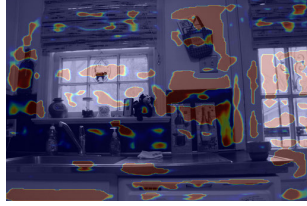
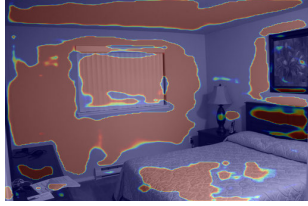
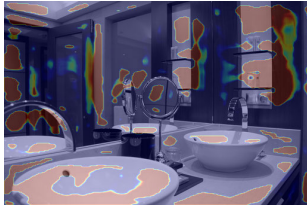
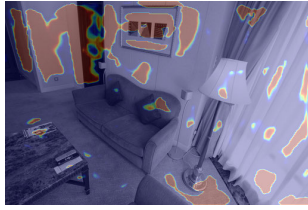












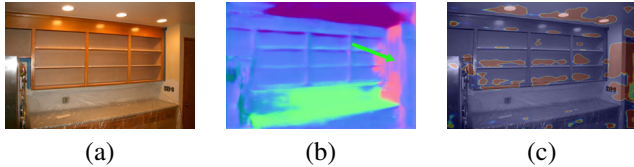


Figure 14: Connection between our smooth shading predictions and normal predictions of [1]. **(a)** Input image. **(b)** Normals predicted by [1]. Note the incorrectly predicted wall edge at the right of the image (indicated by green arrow). **(c)** Smooth shading heatmap predicted by our method. Note the smooth shading bleeding over a normal discontinuity at the same area where the normal prediction was incorrect.

### 5.1. Heatmap Discussion

Most predicted smooth shading regions are reasonably good with high precision. However, our network sometimes makes mistakes. For example, predictions may “bleed over” normal or depth discontinuities as shown in Figure 14-(c). To understand this behavior, we dug deeper into the prediction of the normals from Bansal *et al.* [1], whose network we fine-tuned. We find that indeed the original network often incorrectly predicts normals in such cases (Figure 14-(b)), and this error propagates through to our network as well. Future improvements in normal prediction can improve our heat map predictions, though this is orthogonal to our current submission.

Another occasional mistake is when the normal prediction misses thin structures. This might occur because we resize the input images to a relatively low resolution ( $224 \times 224$ ). These problems can be mitigated in future work with fully convolutional training on higher resolution inputs.

## 6. [Zhou *et al.* 2015] vs. [Bell *et al.* 2014]

In Section 5.2 of the paper we show that the baseline based on [Bell *et al.* 2014] outperforms the baseline from [Zhou *et al.* 2015] on the task of predicting smooth shading pixels. Even though the decomposed reflectance layers of [Zhou *et al.* 2015] are higher quality than [Bell *et al.* 2014] based on the WHDR metric, [Zhou *et al.* 2015] makes certain mistakes in the shading layer which [Bell *et al.* 2014] does not, and consequently Zhou *et al.* [4] performs worse in our smooth shading prediction experiment. We further demonstrate this by showing decompositions of both algorithms on 20 randomly picked images below (to avoid bias). For each image, we point out the most important differences with green arrows. In two cases, we deemed [Zhou *et al.* 2015]’s shading layer to be better than [Bell *et al.* 2014] and showed the mistakes of [Bell *et al.* 2014] with red arrows. Generally, the shading channel of [Zhou *et al.* 2015]

is too high contrast in many cases and follows image intensities. However, a typical case where [Zhou *et al.* 2015] performs better is at depth discontinuities or shadows on surfaces with constant reflectance thanks to the learned reflectance prior. An exciting future research direction is to take into account both reflectance and shading errors to further improve Zhou *et al.*’s algorithm.



(a) Input

(b)  $\mathcal{R}$  [Bell *et al.* 2014]

(c)  $S$  [Bell *et al.* 2014]

(d)  $\mathcal{R}$  [Zhou *et al.* 2015]

(e)  $S$  [Zhou *et al.* 2015]



(a) Input

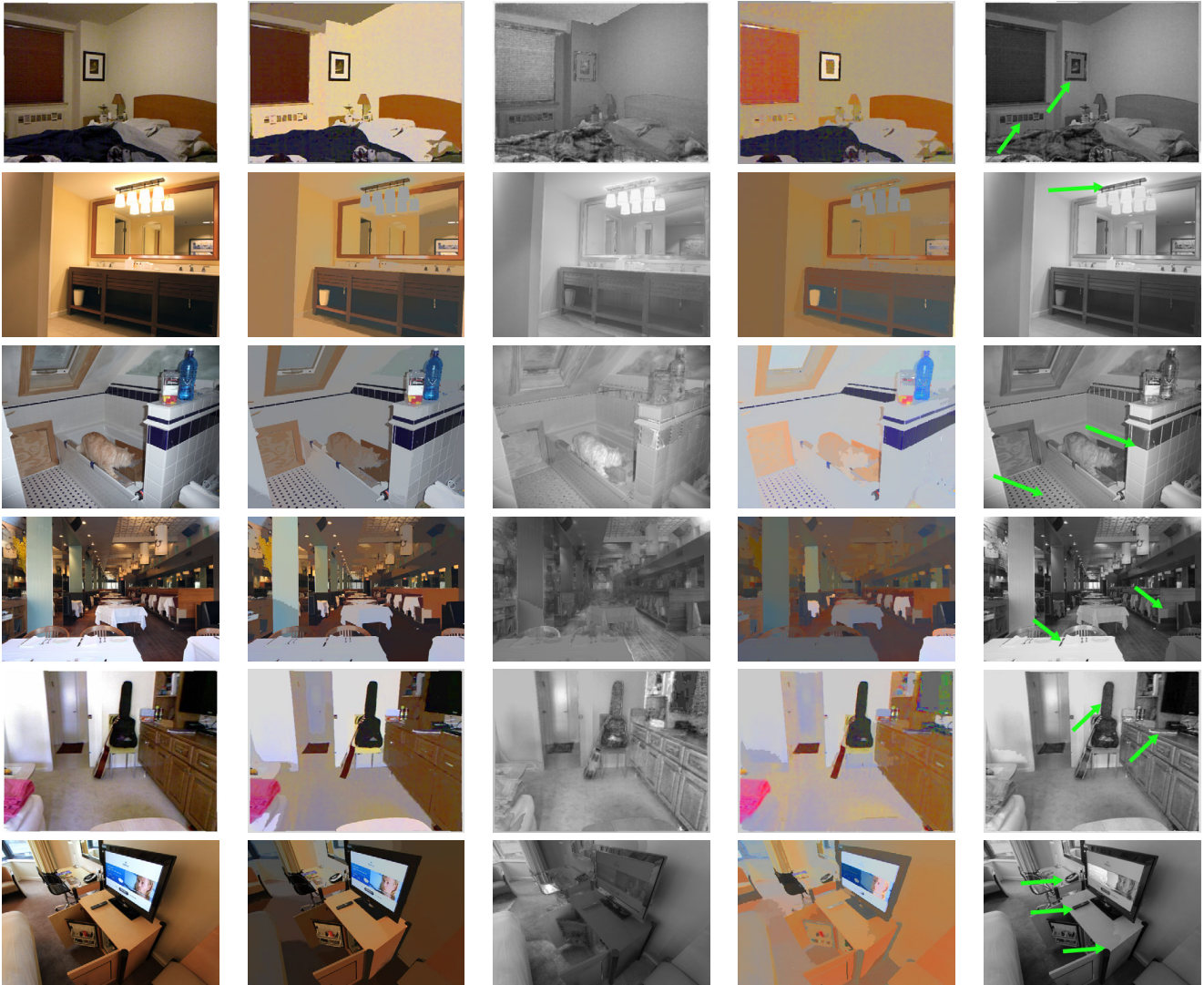
(b)  $\mathcal{R}$  [Bell et al. 2014]

(c)  $S$  [Bell et al. 2014]

(d)  $\mathcal{R}$  [Zhou et al. 2015]

(e)  $S$  [Zhou et al. 2015]





(a) Input

(b)  $\mathcal{R}$  [Bell *et al.* 2014]

(c)  $\mathcal{S}$  [Bell *et al.* 2014]

(d)  $\mathcal{R}$  [Zhou *et al.* 2015]

(e)  $\mathcal{S}$  [Zhou *et al.* 2015]

## 7. Intrinsic Images Decompositions with and without Our Smooth Shading Prior

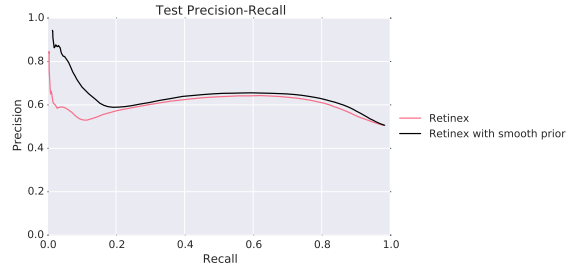


Figure 15: *PR curves over all test images of our simple Retinex implementation with and without smooth-shading prior. We can see that the smooth-shading prior improves the shading decomposition performance.*

decomposition algorithms are inferior to the results of baseline algorithms presented in the paper.

### 7.2. Qualitative Evaluation

We show intrinsic image decompositions with our smooth shading prior for 20 photos from the test set below. We also show 3 failure cases in Figure 16. These preliminary results show improvement in many cases and that our method is a promising new way to think about the intrinsic images problem. However, more work has to be done to incorporate our prior into state-of-the-art intrinsic image algorithms.

### 7.1. Quantitative Evaluation

We generated the PR curves for the two algorithms we used in Figure 8 of the paper on the whole test set (see Figure 15). We can see that the smooth-shading prior helps the algorithm to achieve better shading decomposition performance. Note that we did not choose the optimal  $t$  threshold using cross-validation; therefore, the results of these two



(a) Input

(b) Smooth  $\mathcal{S}$  ( $\mathcal{S}$ )

(c)  $\mathcal{R}$  w/o prior

(d)  $\mathcal{S}$  w/o prior

(e)  $\mathcal{R}$  with prior

(f)  $\mathcal{S}$  with prior



(a) Input

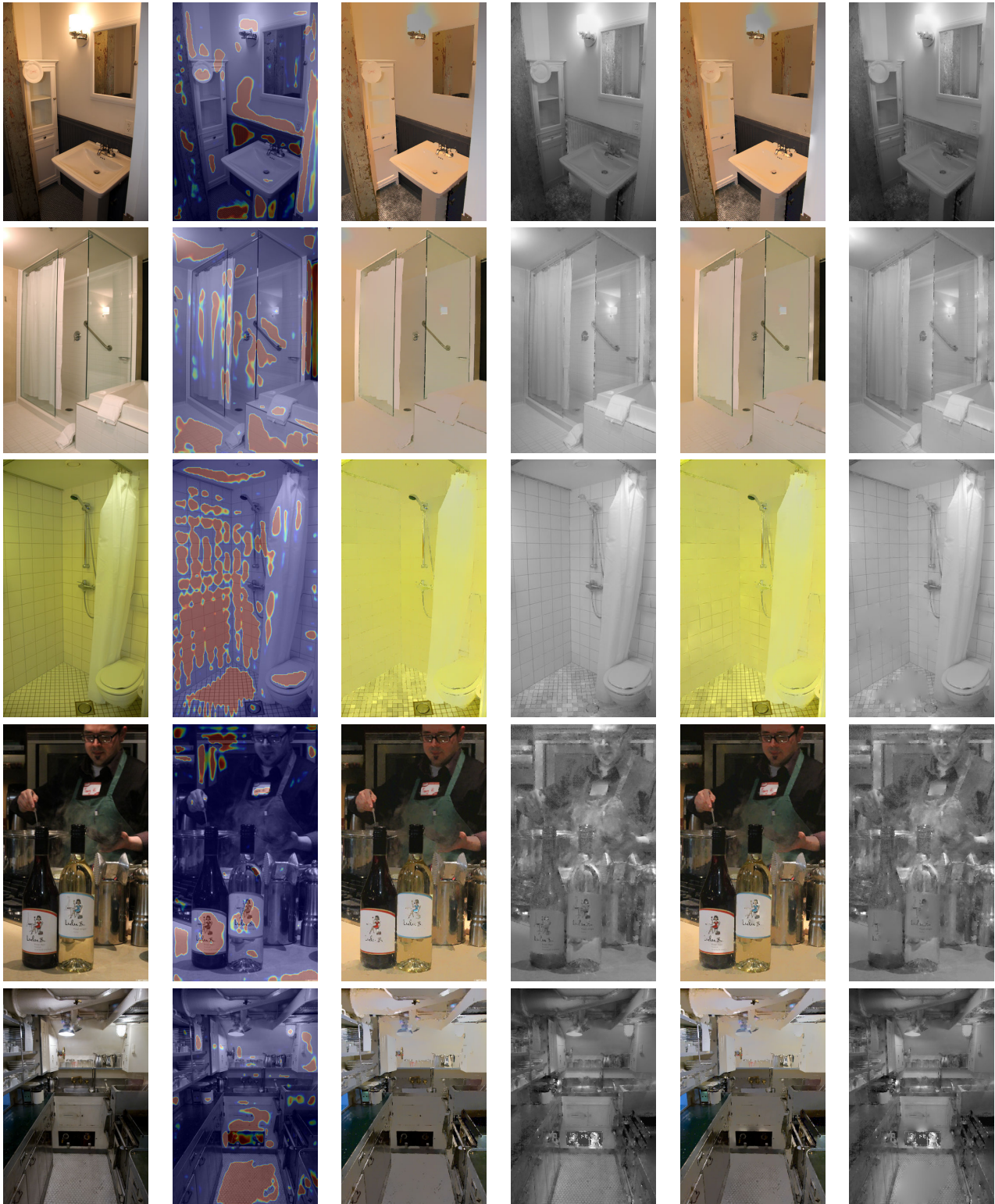
(b) Smooth  $\mathcal{S}$  ( $\mathcal{S}$ )

(c)  $\mathcal{R}$  w/o prior

(d)  $\mathcal{S}$  w/o prior

(e)  $\mathcal{R}$  with prior

(f)  $\mathcal{S}$  with prior



(a) Input

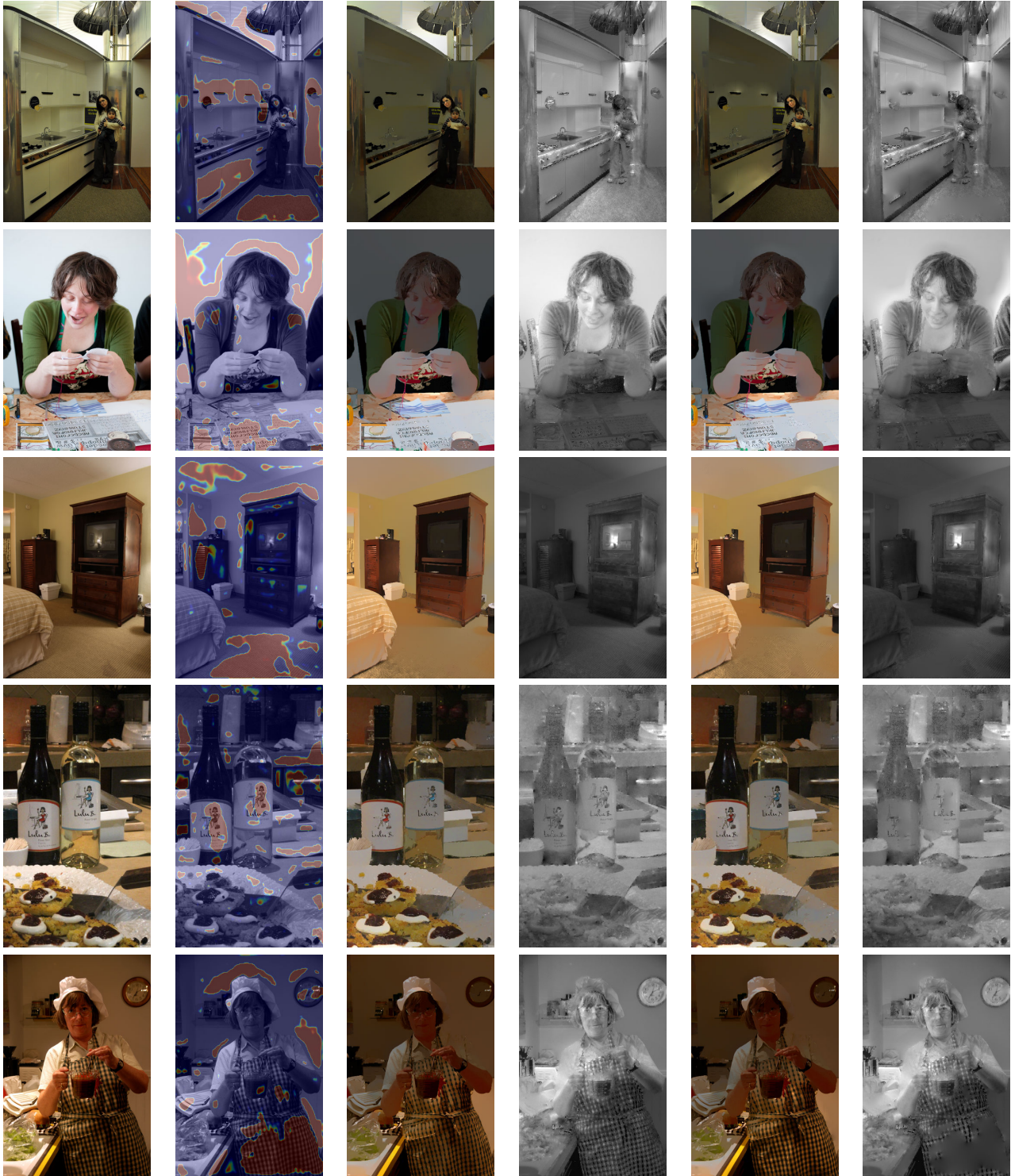
(b) Smooth  $\mathcal{S}$  ( $\mathcal{S}$ )

(c)  $\mathcal{R}$  w/o prior

(d)  $\mathcal{S}$  w/o prior

(e)  $\mathcal{R}$  with prior

(f)  $\mathcal{S}$  with prior



(a) Input

(b) Smooth  $\mathcal{S}$  ( $\mathcal{S}$ )

(c)  $\mathcal{R}$  w/o prior

(d)  $\mathcal{S}$  w/o prior

(e)  $\mathcal{R}$  with prior

(f)  $\mathcal{S}$  with prior

## References

- [1] A. Bansal, B. Russell, and A. Gupta. Marr Revisited: 2D-3D model alignment via surface normal prediction. In *Proc. Computer Vision and Pattern Recognition*, 2016. 14
- [2] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014. 1
- [3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 3
- [4] T. Zhou, P. Krahenbuhl, and A. A. Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *Proc. International Conference on Computer Vision*, pages 3469–3477, 2015. 1, 14



(a) Input

(b) Smooth  $\mathcal{S}$  ( $S$ )

(c)  $\mathcal{R}$  w/o prior

(d)  $S$  w/o prior

(e)  $\mathcal{R}$  with prior

(f)  $S$  with prior

Figure 16: *Failure cases*. In some cases, our algorithm predicts smooth shading incorrectly and this leads to incorrect intrinsic image decompositions. The mistakes for each row are as follows: **First row:** We predict that the window shade region has smooth shading. **Second row:** The bookshelf has normal discontinuities and should not be predicted as smooth shading. **Third row:** The normal discontinuity of the door-frames on the left and right is ignored.