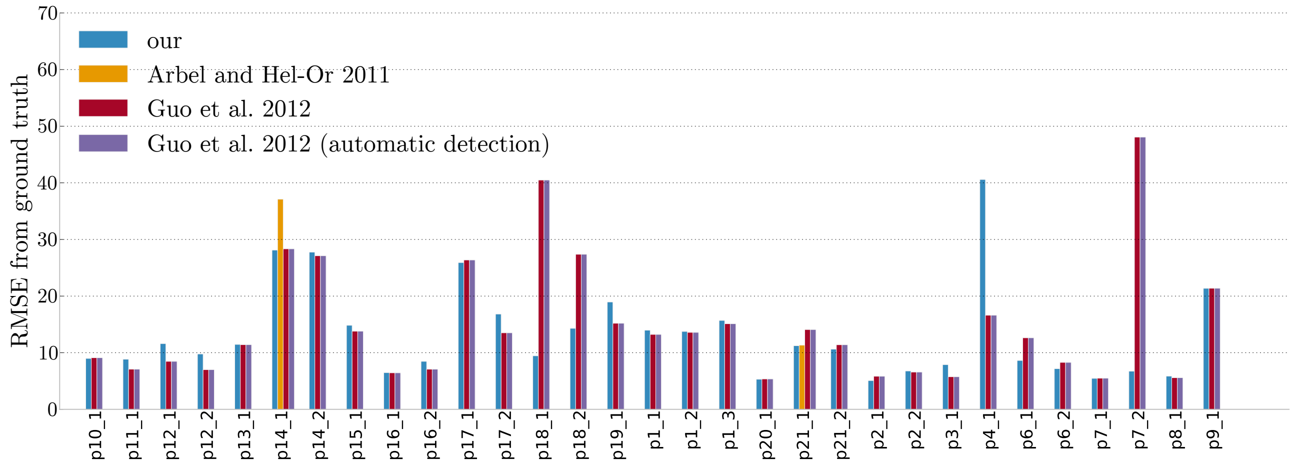
Learning to Remove Soft Shadows
Supplementary Material

Appendix 1: Per-Image Quantitative Comparisons

Here we present more quantitative evaluation of our and related methods as an addition to Table I in the main paper.



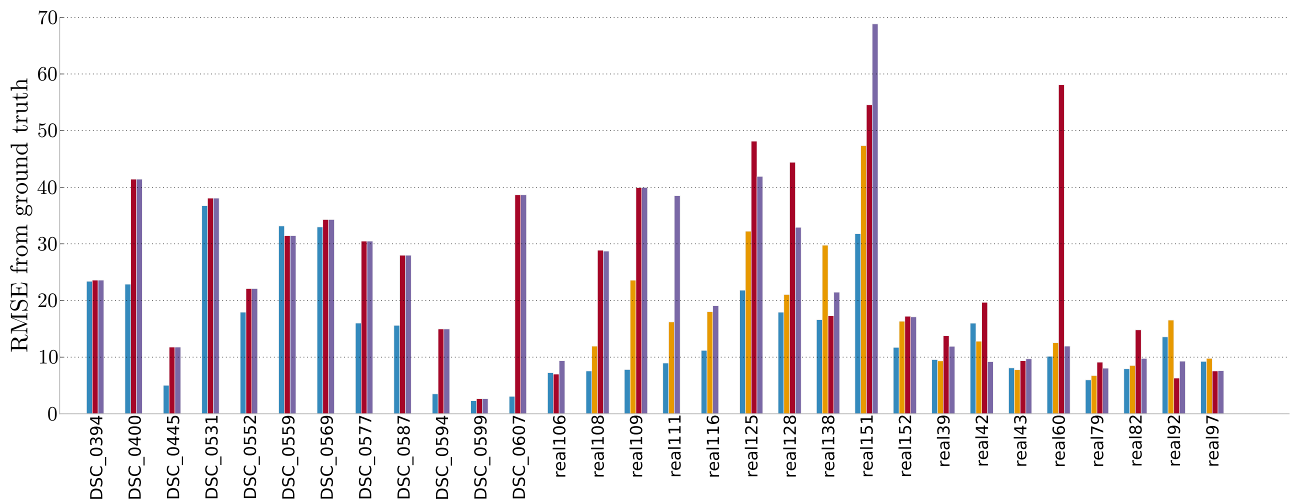


Figure 1: RMSE between results of different shadow removal methods and the ground truth shadow-free images. Ushadowing results for "real151" and "p4_1" can be found in figures 8 and 17 in the main paper respectively.

Appendix 2: User Study Interface

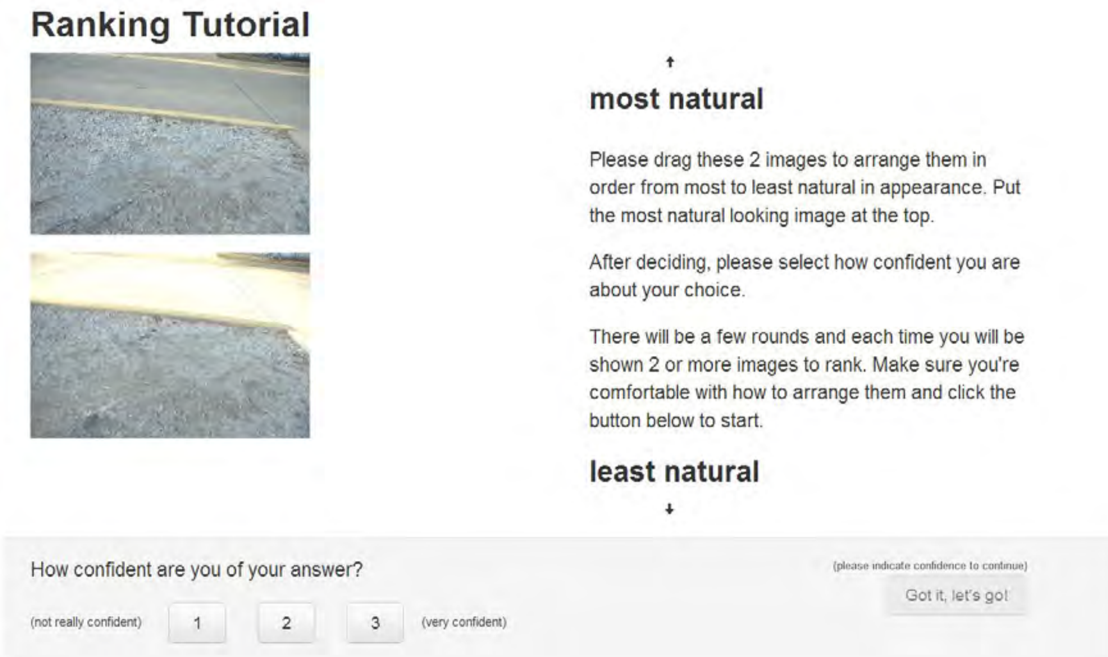


Figure 2: Screen capture of Task 1 of the user study. The participants were asked to decide which of the presented images seemed more natural and arrange the presented results by click-and-drag.

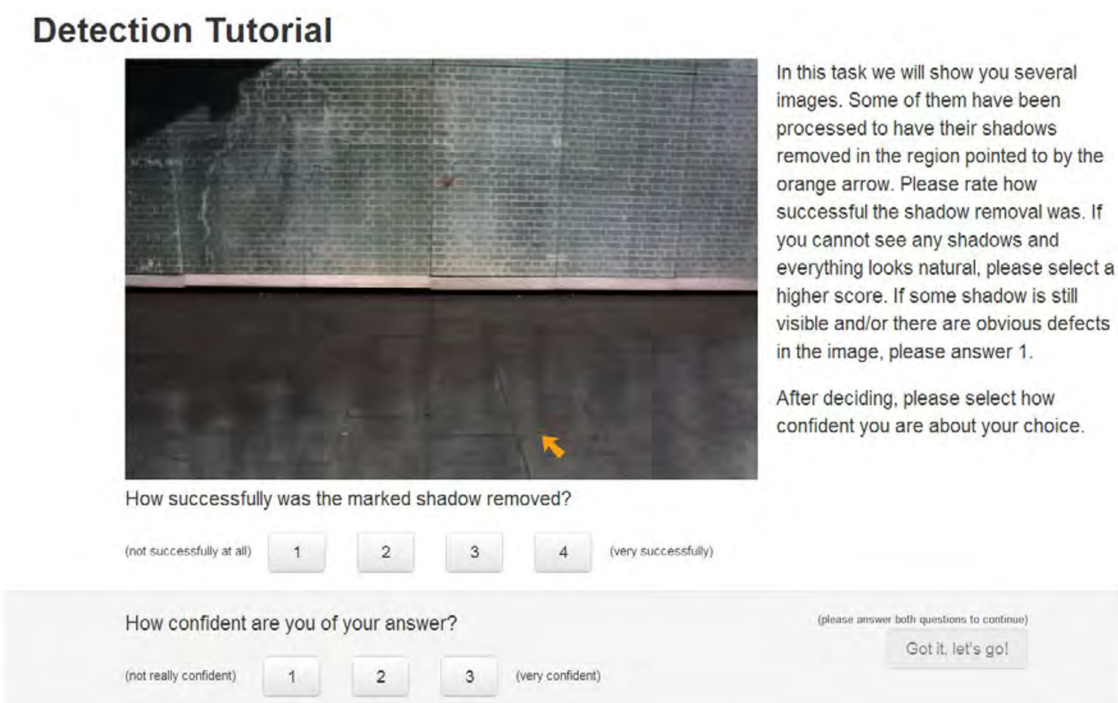


Figure 3: Screen capture of Task 2 in the user study. The participants were presented with results of shadow removal using either our method, Guo *et al.*'s or Arbel and Hel-Or's and asked to indicate how successful the shadow removal was in the marked region.

Appendix 3: Additional User Study Data

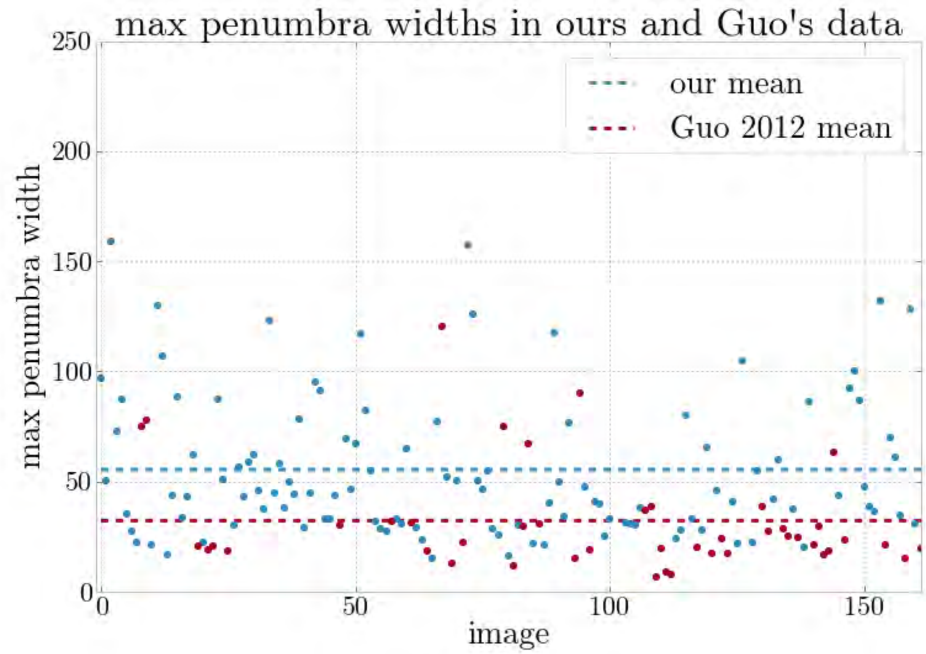


Figure 4: Maximum penumbra widths for images in ours and Guo *et al.*'s datasets. Note that the average penumbra in our dataset is significantly wider than in Guo *et al.*'s (meaning softer shadows).

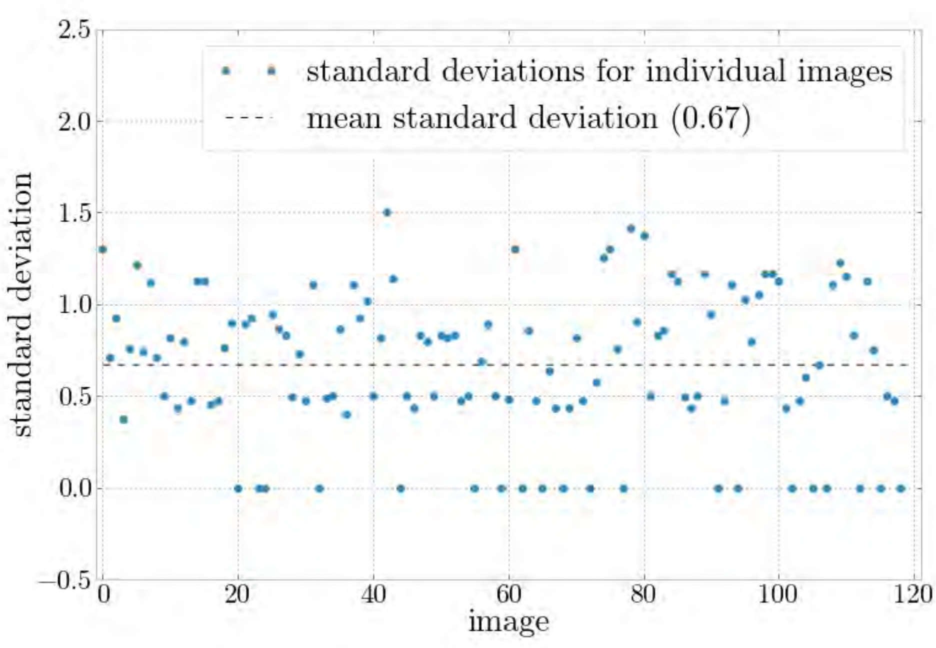


Figure 5: We have measured the standard deviations of scores that the study participants assigned to individual images. The users seemed to mostly agree in their evaluations of the results.

Appendix 4: Algorithm Design and Exploration

Comparison of Different Initialization Techniques

Figure 6: Comparison of results using different inpainting approaches. Note that in the case of texture-guided inpainting there are often out-of-shadow parts of the image that the algorithm chooses as the source. This is a failure of the distance function used as described below, however, we were not able to find a better solution.

Impact of the Training Set Size

We have evaluated how the performance of our algorithm varies with the amount of training data supplied. First, we have kept the number of patches sampled per image constant and set the number of training images rendered to 1000, 5000, 10000 and 15000. We have empirically determined that using up to 10000 images provides a good balance between the training time and performance, while beyond the trade-off becomes less attractive. Secondly, we have used similar procedure to arrive at the appropriate number of patches sampled per image.

Feature Vector

Here we present a short exploration of the relative importance of features in our feature vector. While feature importance is a very informative measure, it would not be feasible to compute using our full dataset. Instead below we present "feature frequency", a number of times each feature was used as a split in the entire forest.

We have also explored other features, such as explicit pixel-wise gradient orientation and magnitude, and the angular position of the patch within the masked region. However, none of them offered substantial improvements in performance.

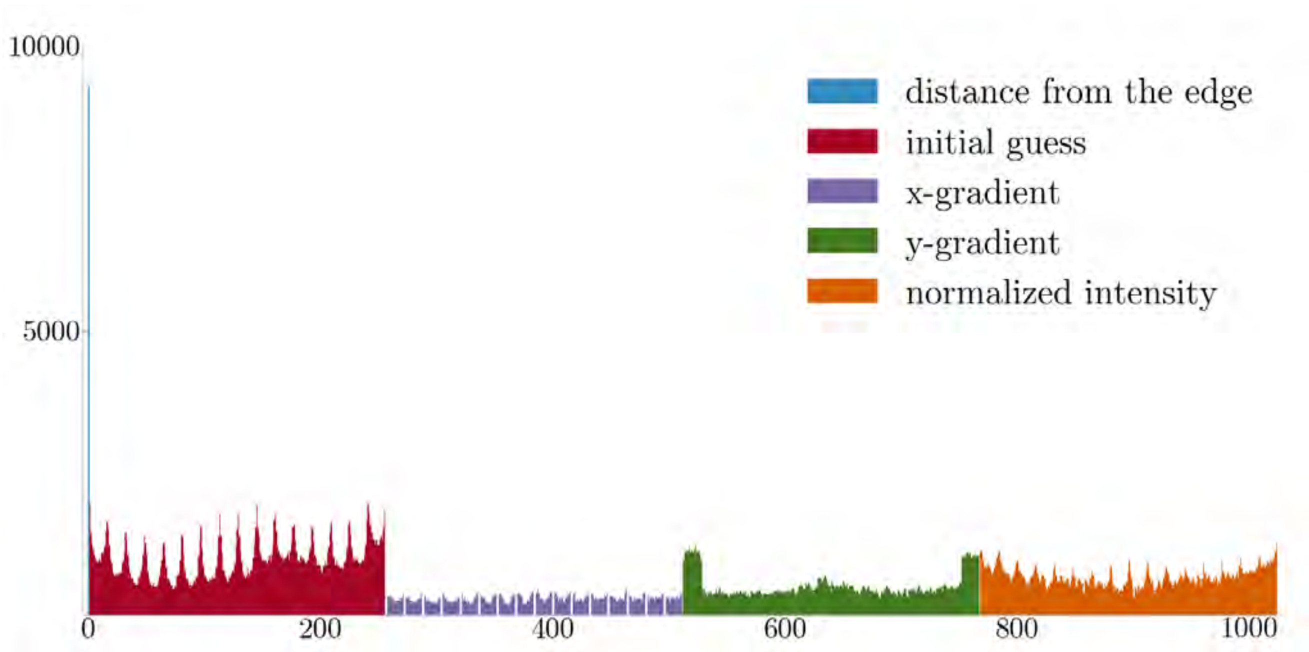
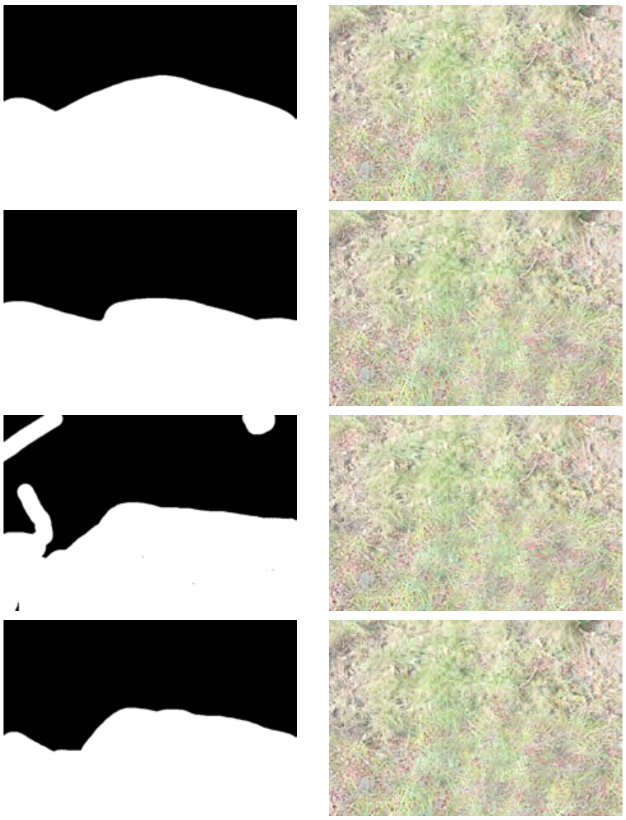


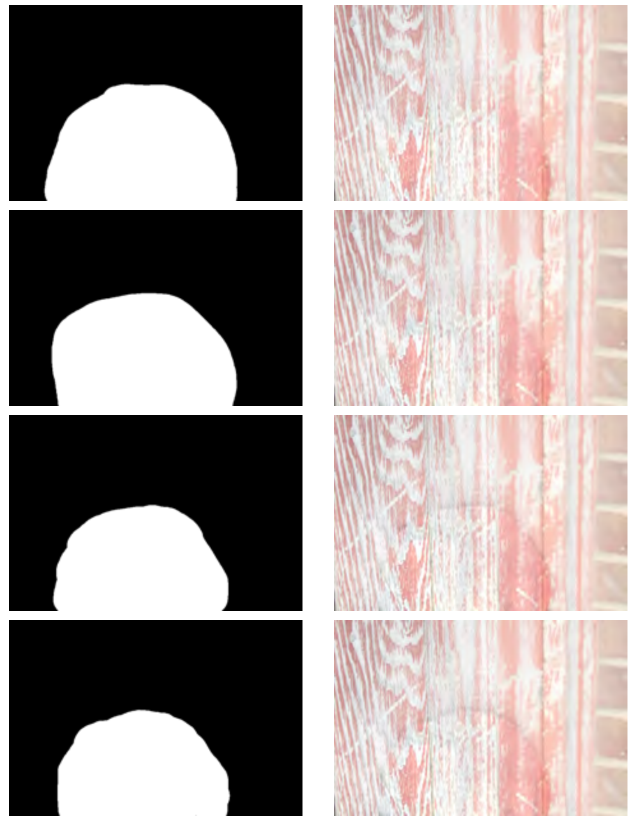
Figure 7: Number of times each part of our feature vector was used as a split dimension at a node. Note the spike at position 0 indicating that the distance from the mask edge is an important feature.

Appendix 5: Impact of User Input on Results

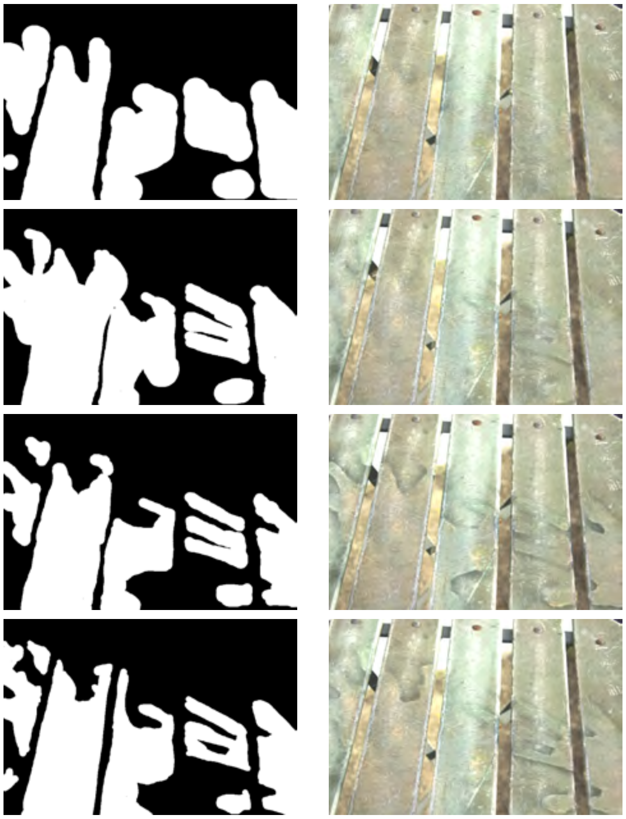
The results below show unshadowing results obtained by three different people for five scenes, as described in Section 6 in the paper. We conclude that small differences in the provided masks do not contribute significantly to the quality of final results as long as all the penumbrae are selected. The exception is the bottom row in most of the scenes, which shows penumbrae parts being left untouched, which is caused by this user's tendency to underestimate the extent of the shadow. Recall that we intentionally constrain the unselected regions to be considered out-of-shadow and therefore unquestioningly trust the users' judgment. This design decision could be revisited *e.g.* by automatically dilating the masks by a certain amount.



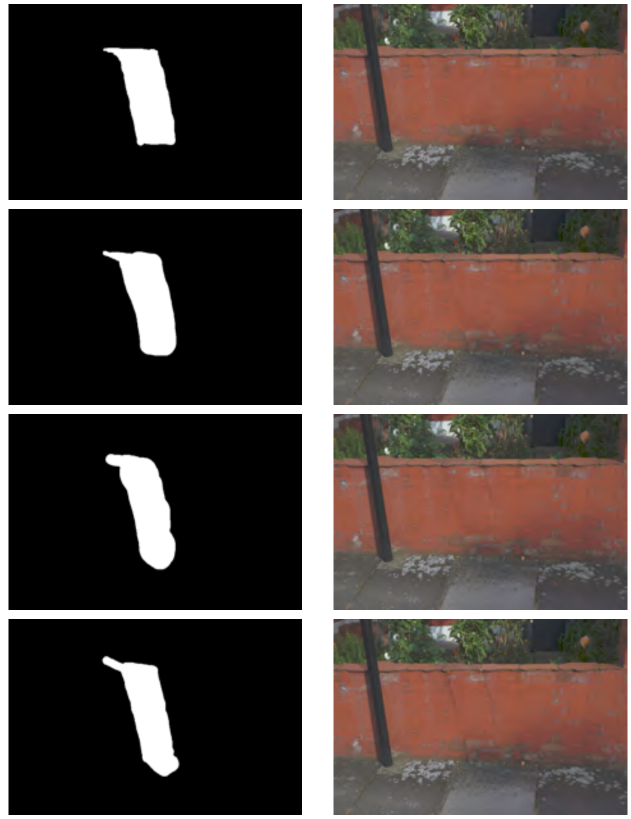
real22



real26



real138



real168

real249

Figure 8: Unshadowed results ae robust to variations in the user-provided binary mask.

Appendix 6: Converting RAW images to linear PNGs

As explained in Section 7 of the paper, we have used linear PNG photographs to evaluate our method. Specifically, we have captured 14-bit RAW images using Canon 550D camera and used the `dcraw` and `pnmtopng` command line tools to obtain 8-bit linear PNG images as follows:

```
dcraw -c -q 3 -b 2.0 -W -w -g 1 1 input.CR2 | pnmtopng -gamma 1 -force > output.png
```