Supplemental Results for
"Automating Image Morphing using Structural Similarity on a Halfway Domain"

1 Occlusion handling using the halfway parameterization

Figure 1 shows an additional result of self occlusion during a morph. In cases where regions from one image are not represented in the other image, blending artifacts can occur. We also show the halfway domain grid visualization, which demonstrates the warp that is required in order to get the images in alignment. Finally, figure 2 shows a complex occlusion example where one object occludes entire regions of another object. Our basic method cannot directly handle this case (top row) and we must resort to segmentation to address the problem (bottom row).

2 Correspondece estimation

To better motivate our SSIM-inspired approach, we compared the result of our method with that of using SSD on both intensity and on RGB, as shown in figure 3. For the first three examples, our method is clearly superior since SSD does not work correctly on situations where regions of different intensities/colors should be matched. In the last example, where colors are similar, SSD performs reasonably well, though still not as well as our approach. While our approach works well for many cases, it certainly needs the aid of at least a few manual control points in the majority of cases. This is especially clear in figure 17 of our submission, where we show how the results improve as more control points are added. In the these results, one problematic case is the chin mismatch between the two kids in the first example, which requires at least one additional control point in order to be matched properly.

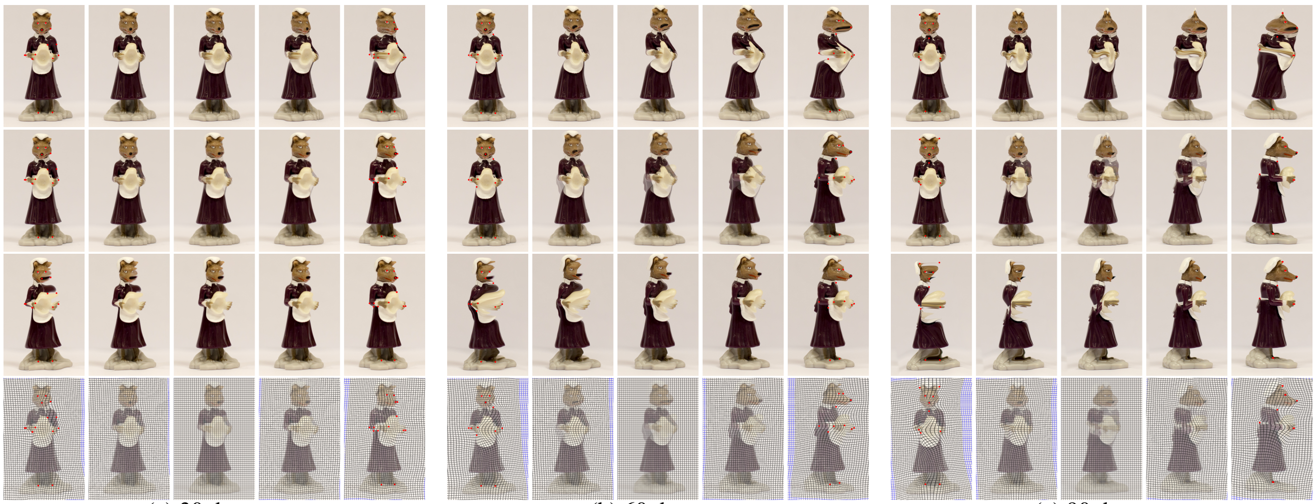
3 Image extension and direct pixel evaluation

Figure 4 shows additional results of three animation sequences comparing rasterization of the pixel grid and our direct pixel evaluation. The direct pixel evaluation performed in the entire intermediate image domain enables Poisson image extension, leading to improved results near the boundaries.

We have extended our morphing technique to show parametric grids mapped over the blended image during the morph (figure 5). These grid lines, which are defined over the halfway parameterization, are generated within our iterative-search-based direct pixel evaluation algorithm. The grid is drawn in blue where it extends beyond the halfway domain. The nice continuity between the black and blue grid lines shows that our extrapolation scheme behaves well.

4 Visualization of motion paths

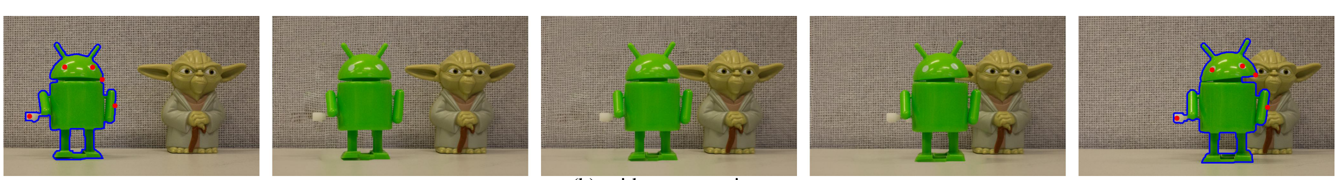
Finally, figure 6 shows a visualization of the vector fields v and w of our computed motion paths. It more clearly shows in what directions different regions of the input images move during the morph sequence.



(a) 30 degrees (b) 60 degrees (c) 90 degrees

Figure 1: *Different degrees of rotation. For each, the top row uses color only from the first (left-most) input image (no blending). The second row uses the blended color result of our algorithm. The third row uses the color from the second (right-most) input image. Finally, the last row overlays the halfway image domain grid on the results of the second row.*

(a) without segmentation



(b) with segmentation

Figure 2: *A complex occlusion example that requires segmentation. User-drawn correspondence points and segmentation boundaries are shown over the input images I_0 (left) and I_1 (right).*

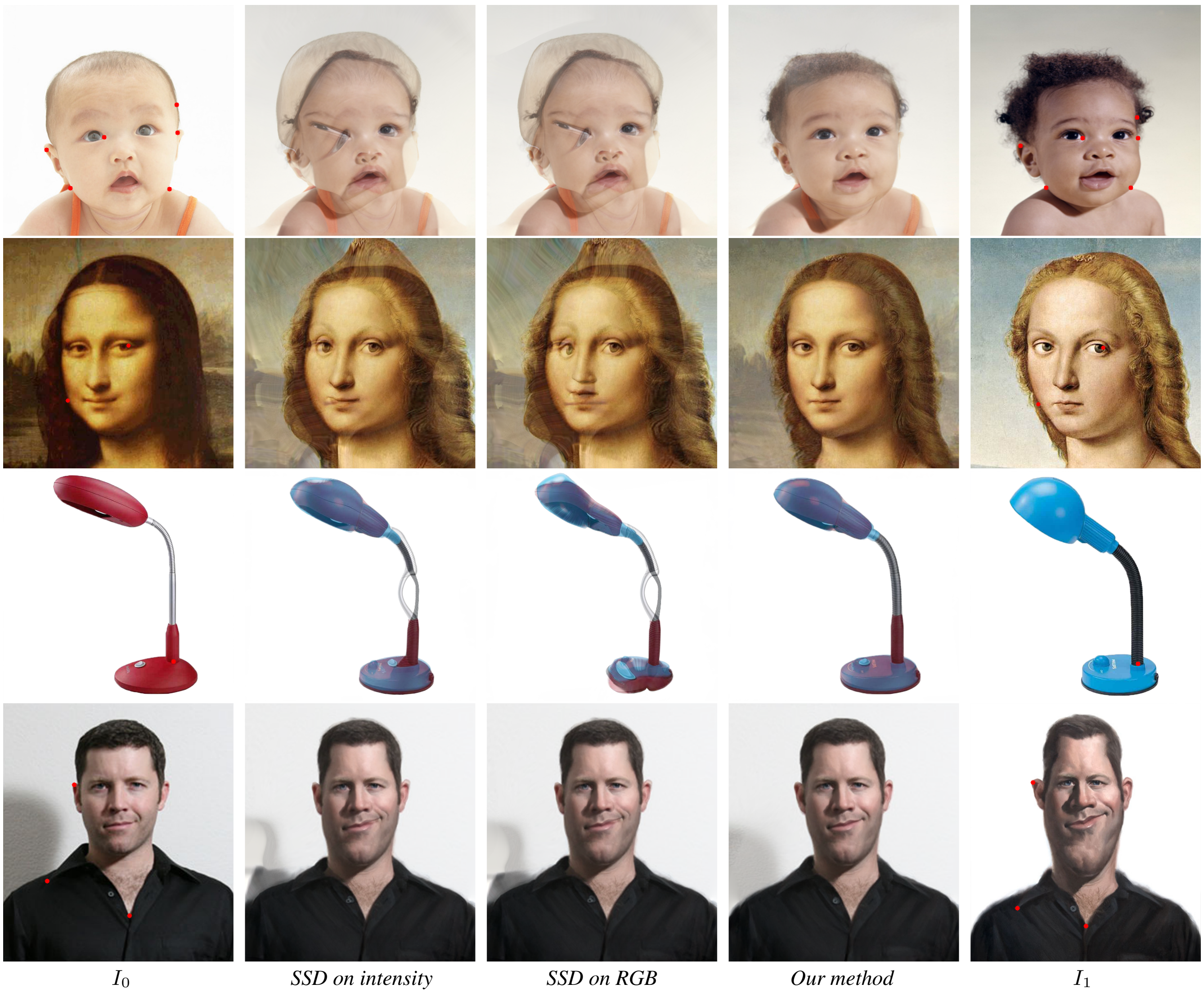
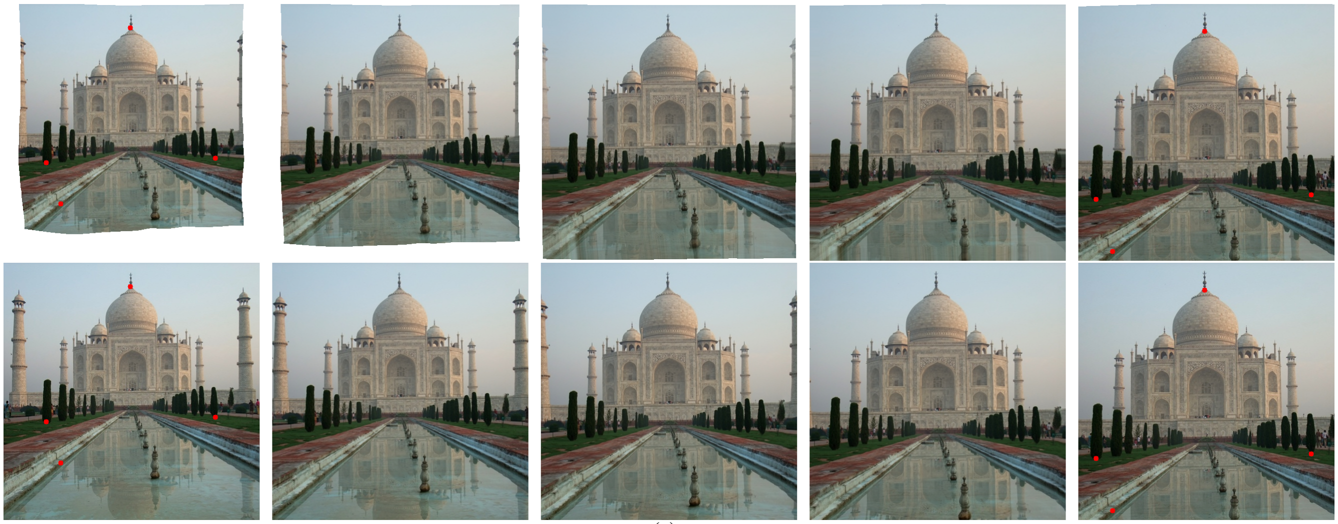
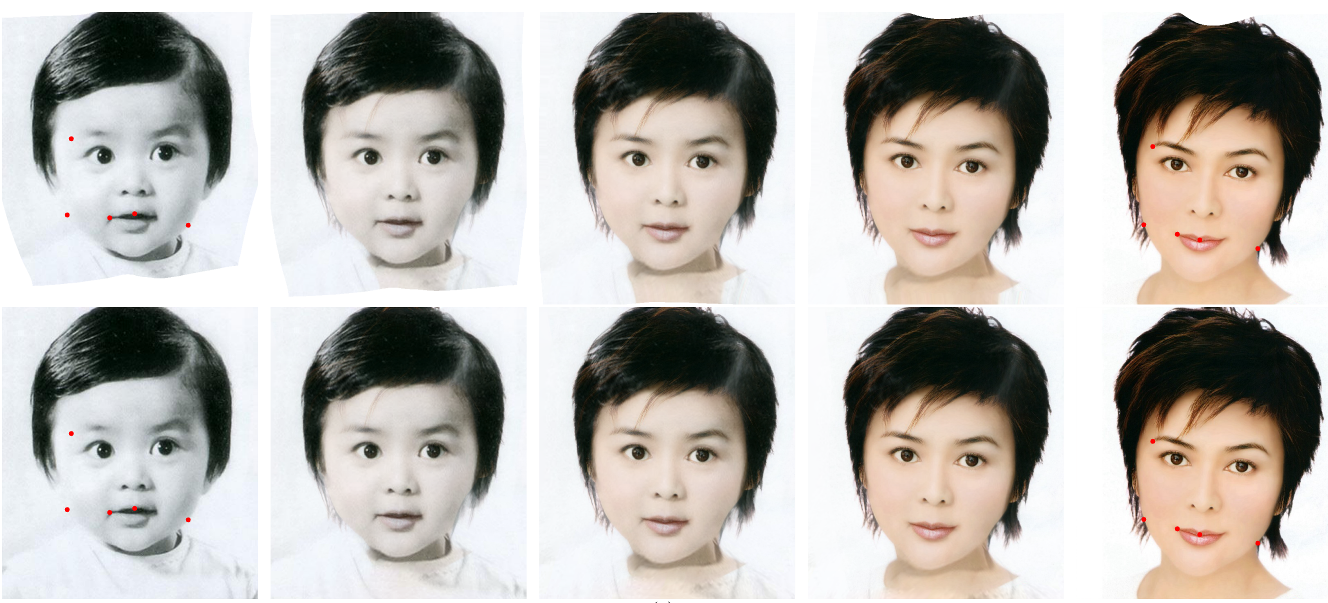


Figure 3: *Comparison between using our technique and SSD on intensity and RGB values.*



(a)

(b)



(c)

Figure 4: *For each of three examples, comparison of morph sequences obtained using rasterization (top row) and using direct pixel evaluation (bottom row).*

Figure 5: *Visualization of morph deformation by overlaying the halfway domain grid.*

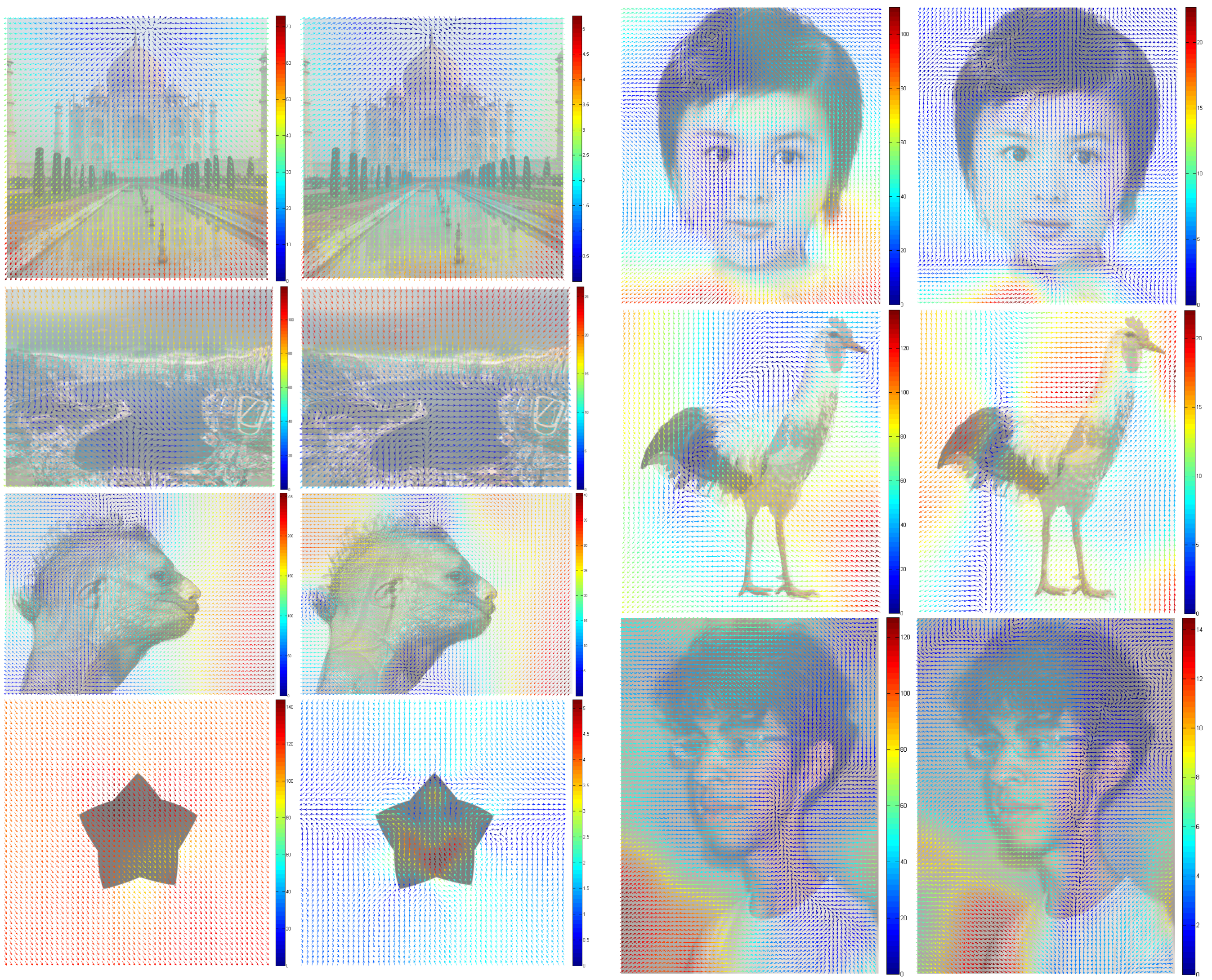


Figure 6: *Visualization of the vector fields v (left) and w (right). The orientation of each vector glyph shows the direction of the field, while its color shows the field's magnitude.*