Supplemental Material

Illustrations of several sub-optimal algorithm variations

The proposed method uses non-local matches and an iterative process. In Figure 1 we show the results of using only local matches with iteration, or using non-local matches but with no iteration. In either case, the results are inferior to that of the full non-local, iterative algorithm, whose final result is illustrated in Figure 2(e) of the paper.

Illustrations of several sub-optimal algorithm variations

Figure 2 illustrates what happens to the depth of an object as it is deformed to match image contours. A sphere is deformed to fit the contour of a football, which preserves it circular cross-section.

Challenges in directly computing occlusion contours

Figure 3 illustrates the noisy nature of unprocessed occlusion contours as computed from a geometric mesh, and the potentially complex ways that occlusion contours can split and join. The HMM algorithm avoids having to explicitly compute such contours for the model.

Illustration of the iterative deformation

Figure 4 shows the iterative correspond-and-deform process.

Teddy bear model reconstruction from multiple views

Multiple views can be used simultaneously or in sequence, as shown in Figure 5. To use the views simultaneously we would need to register them accurately with respect to one another. Misalignments can lead to artifacts when working simultaneously with two views, and thus we adopt a sequential strategy instead. In the given example, we deform the template using the contours from view A, then view B, and once again for view A. The image teddy differs from the template teddy in pose as well as the shape of the nose, ears, and feet. These differences are successfully recovered from the contour information with the help of a few user-specified correspondences. These are necessary mostly to correctly position occluded parts in the side view. Since the template is not symmetric, we cannot use symmetry for correct modeling of occluded parts in this case.

Details of symmetry preservation

A symmetry plane is first precomputed on the template using any of the recently developed symmetry detection methods for geometric models. When the template deforms, the global symmetry is no longer preserved. However, we expect symmetry to be locally preserved, e.g., both hands of a human have equal finger length, even if the two arms are positioned differently. Using the symmetry plane we compute a mapping for any vertex of the undeformed template mesh to its corresponding symmetric position.

Our method first deforms the model using the iterative correspond-and-deform procedure, disregarding symmetry. Users are then asked to specify the occluded regions they want repaired. Given this input, the method computes new local shape descriptors, in our case mean-value coordinates, for each vertex. First for vertices outside the specified regions it computes a new descriptor using the deformed mesh. For vertices inside these regions it computes the symmetric descriptor by first mapping the vertex and its neighboring vertices to their associated symmetric locations on the deformed model and then computing the descriptor based on those. The algorithm then reapplies the deformation using the precomputed matches with vertices in the selected regions using their symmetric descriptors and vertices everywhere else using the newly deformed ones.

Another example of using symmetry constraints

In the absence of symmetry constraints, the lion in Figure 6 the algorithm arbitrarily selects one of the front legs to match to the front-leg contour, and similarly for the back legs. This highlights the importance of consistency enforcement in the matching stage and showcasing the method's ability to preserve unmatched template geometry. With the additional enforcement of symmetry, both left and right legs are effectively matched to the singly drawn leg contours.

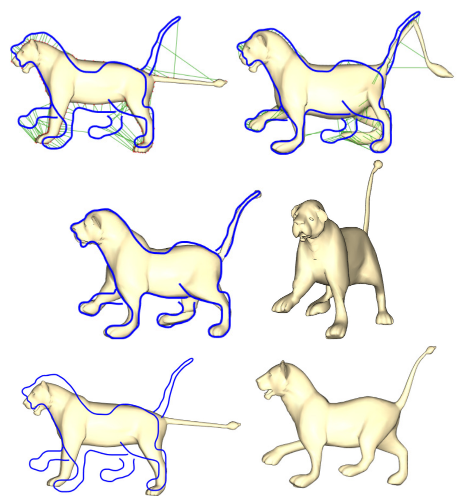
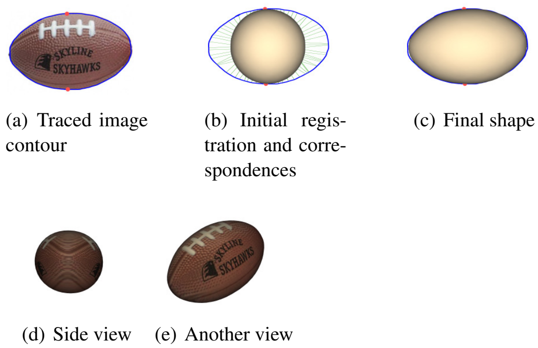


Figure 1: *Sub-optimal algorithm variations applied to the cartoon lion (Figure 2) Top: Iterative deformation using only local matches, showing the first-iteration correspondences (left) and the resulting final deformation (right). Middle: Non-iterative deformation using only initial correspondences (Figure 2(d)). Bottom: Unconstrained initial alignment produces a solution with a left-right mismatch for the hind legs.*



(a) Traced image contour

(b) Initial registration and correspondences

(c) Final shape

(d) Side view

(e) Another view

Figure 2: *Football*

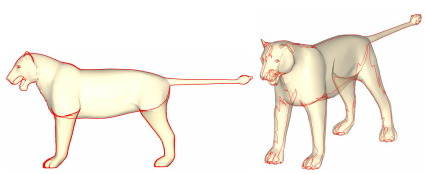


Figure 3: *Occlusion contours for the caricature lion in side view (left) rotated to visualize contour structure (right).*

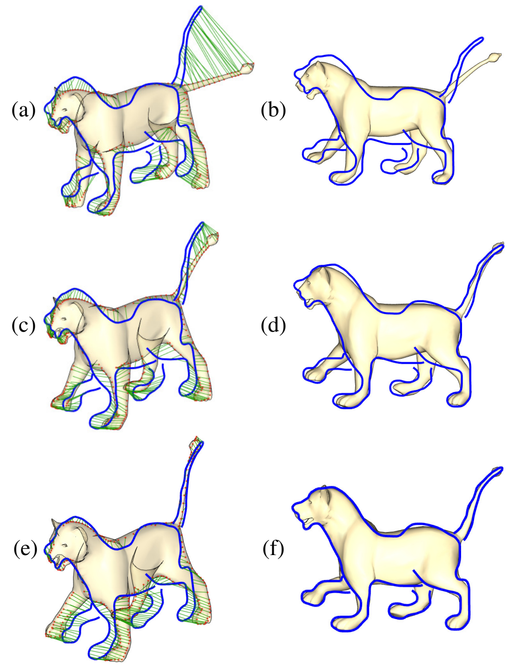


Figure 4: *The iterative correspond-and-deform process. The matches are shown in semi-transparent 3/4 view in order to showcase the impact of continuity. (a) initial correspondences; (b) deformed model after 5 iterations; (c) correspondences after 5 iterations; (d) deformed model after 10 iterations; (e) correspondences after 10 iterations; (f) final fit to contours after 15 iterations.*

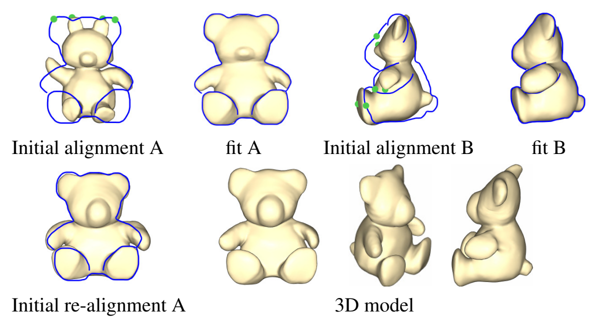
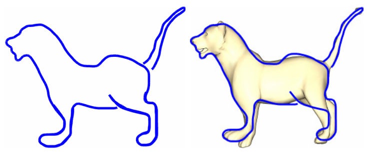
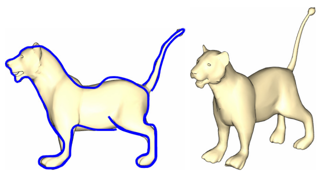


Figure 5: *Teddy bear model illustrating modeling from multi-view sketches.*



(a) Input contour (b) Fit (no symmetry)



(c) Final result

Figure 6: *Modeling a lion from a side view (two legs) contour. Symmetry is used to achieve the final result.*