RECOVERING GEOMETRIC INFORMATION WITH LEARNED TEXTURE PERTURBATIONS: SUPPLEMENTAL MATERIAL

Anonymous authors
Paper under double-blind review

A DATASET GENERATION

A.1 TOPOLOGICAL CONSIDERATIONS

There are some edge cases that require additional topological consideration. In particular, the collar, sleeves, and waist are areas where a ray cast to an inferred cloth vertex can intersect with a back-facing triangle on the inside of the ground truth shirt. We aim to define texture coordinates on inferred cloth vertices so that barycentric interpolation can be used to find the texture coordinates of a ground truth vertex for 3D reconstruction. However, mixing texture coordinates from the inside and outside of the shirt in a single triangle causes dramatic interpolation error. In fact, as shown in Figure 2, large errors may occur for any triangle that mixes texture coordinates from geodesically far-away regions. Thus, we omit such triangles from consideration by omitting a vertex from any edge that connects two geodesically far-away regions.

As a further improvement to our method, one can treat the inside and outside of the shirt as separate meshes, applying texture sliding twice and training two separate networks; moreover, one may take a patch-based approach, applying TS and training a TSNN for each (slightly overlapping) patch of the shirt.

A.2 SMOOTHNESS CONSIDERATIONS

When training a neural network, more predictable results are obtained when the inferred cloth vertex data is smoother. Thus, there exists tradeoffs between smoothness and accuracy when assigning texture coordinates. An edge that connects two geodesically far-away regions introduces a jump discontinuity in the texture coordinates leading to high frequencies in the ground truth data that place increased demands on the network. Although subdivision adds degrees of freedom along such edges to better sample the high frequency, it is often better to delete such edges entirely by removing one of the edge's vertices. Recall that any vertex not assigned a ground truth texture coordinate is instead defined via smoothness considerations (see Section 4.2) reducing demands on the network.

B 3D RECONSTRUCTION

There are a couple of issues with finding the texture coordinates of the ground truth vertices on an inferred cloth mesh whether it be TS or TSNN data. Firstly, there could be seams in the texture in which case smoothing would be needed near the seam as discussed above in order to avoid degrading the data. A patch-based approach can be used to alleviate any such seams. Secondly, seams, smoothing, and non-linearity along the lines of Figure 2 may all contribute to more than one inferred cloth triangle containing the texture coordinates of a ground truth vertex. This ambiguity can be treated similarly to how correspondence uncertainties are addressed in standard multi-view stereo algorithms. The straightforward approach is to consider each distinct possibility for each camera in all possible combinations and choose the set of rays that have the least disagreement for triangulation; furthermore, one may also consider the 3D reconstruction of neighboring vertices, material deformation, etc. Overall, reliance on multi-view stereo does require careful attention when utilizing our method. As such, we provide a few more examples of 3D reconstruction for examples from the test set in order to demonstrate the efficacy of our approach. See Figure 1.

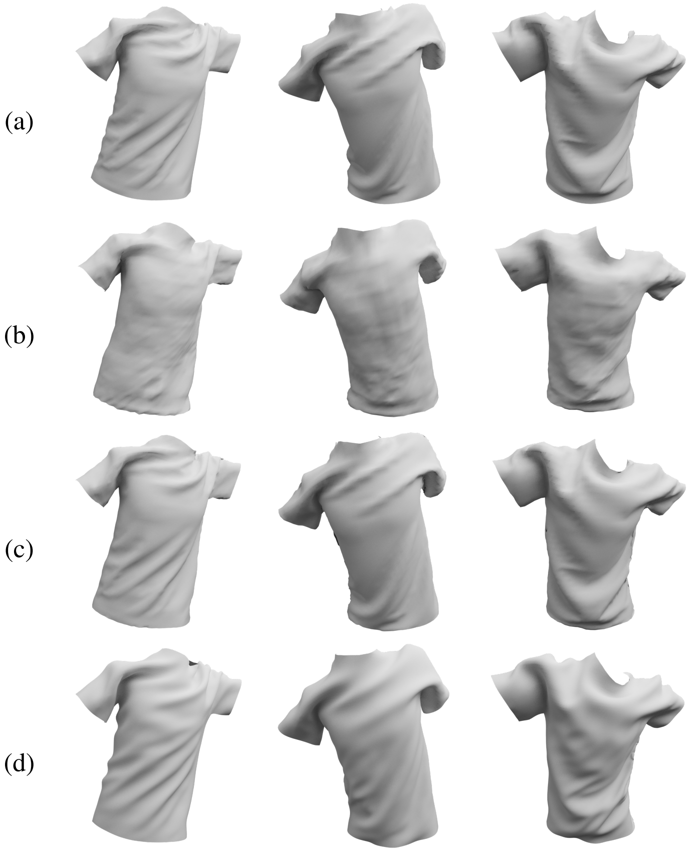


Figure 1: Comparisons of the ground truth cloth (a) and inferred cloth (b) to the 3D reconstructions obtained using texture sliding (c) and the TSNN (d) for three test set examples. Note that the postprocess in Geng et al. (2020) is only applied to (d).

Instead of applying a standard smoothing algorithm to the somewhat noisy results of the 3D reconstructions of the TSNN data, we used the postprocess from Geng et al. (2020). This choice was made because of our desire to use neural networks to branch the gap between physical simulations and real-world material behavior. In order to quantify the impact of the postprocess from Geng et al. (2020) on the final results, Figure 2 shows the results obtained when applying the postprocess directly to the inferred cloth as compared to applying it to TS and TSNN data.

C Novel View Interpolation

Figure 3 quantifies the results in Figure 11 for the inferred cloth $C_N(\theta_k)$, texture sliding $C'_N(\theta_k, v)$, and the results of the TSNN $\hat{C}'_N(\theta_k, v)$. In Figure 4, we repeat these comparisons, except using bilinear interpolation between four camera views.

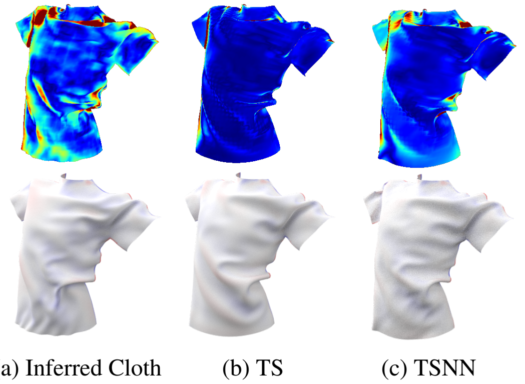

(a) Inferred Cloth (b) TS (c) TSNN

Figure 2: Per-pixel errors (top) and local compression/extension energies (bottom) comparing the postprocess from Geng et al. (2020) applied to (a) the inferred cloth, (b) the 3D reconstruction from TS data, and (c) the 3D reconstruction from TSNN data.

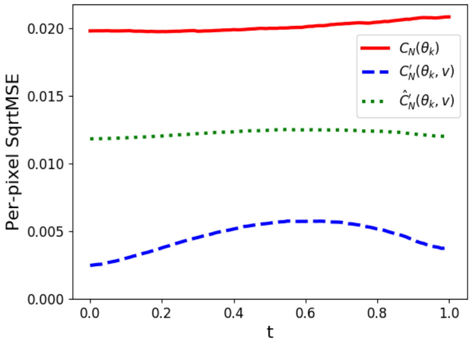
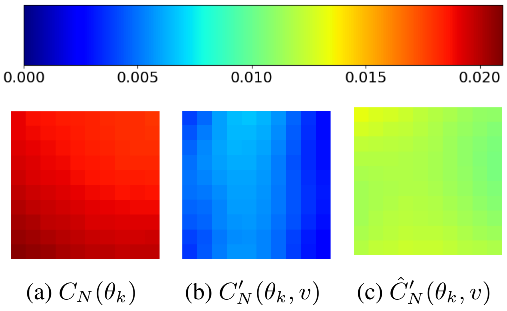


Figure 3: Per-pixel SqrtMSE for interpolating between two cameras (using a test set example). Note that the inferred cloth does not use any view based information, but that our error metric does depend on the view.



(a) $C_N(\theta_k)$ (b) $C'_N(\theta_k, v)$ (c) $\hat{C}'_N(\theta_k, v)$

Figure 4: Per-pixel SqrtMSE for interpolating between four cameras (one at each corner of the square). The pose θ_k is the same as in Figure 3, which plots the values along the bottom edge of the square.

Interpolating between two cameras, each with TS or TSNN data, has the effect of following a straight-line path. However, by choosing the camera array and subsequent interpolation carefully one can interpolate along curved paths. For example in Figure 5, one can interpolate between the 12 cameras (represented by blue dots) in order to follow the curved camera path.

D Error Analysis (for Patches)

In this section, we consider each step of the ray intersection algorithm, carefully illustrating the sources of error. This is done for a single patch consisting of the entire front half of the shirt in order to ensure continuous and unique texture coordinates. Additionally, this section highlights our patch-based approach, noting that we would utilize this approach on a number of overlapping patches and blend the final results together. In fact, when considering only a single patch, we modify our nodes from the inferred cloth to only include that patch, ignoring the rest of the vertices and triangles in the mesh. Similarly, the ground truth cloth is assumed to only consider the data for that patch. Note that any existing network that predicts cloth vertex positions can be adapted to this patch-based approach as a postprocess applied to their training examples, and that one may readily apply the predicted texture separately to each patch.

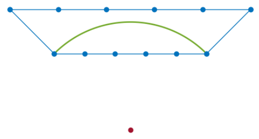


Figure 5: Let the red dot represent the center of the cloth mesh. One can interpolate between the 12 cameras (blue dots) on the trapezoid in order to follow the curved camera path.

Along the lines of Section 4.1 and Appendix A, a ray between the camera aperture and each inferred cloth vertex of the patch is intersected with the ground truth cloth, in order to find the ground truth texture coordinates to assign to the inferred cloth vertex. Recall that the inferred cloth vertex remains unassigned when occluded; however, we modify our definition of occlusion to only consider the inferred cloth patch under consideration. This allows, for example, one to reconstruct the back half of the shirt with cameras from the front, since the front half of the shirt would not be considered and not be occluded by the back half. Since we only consider the front half of both the inferred and ground truth cloth, one also does not compute ground truth texture coordinates to be assigned to the inferred vertex when the ray does not intersect the front half patch of the ground truth cloth. Separating the front and back of the shirt guarantees the inferred cloth patch is assigned texture coordinates from a continuous texture. This leads to a sub-mesh of assigned texture coordinates T_N. As usual, we remove any edge (by deleting an inferred vertex) which connects geodesically far away regions as indicated by differing texture coordinate values. See Figure 6.

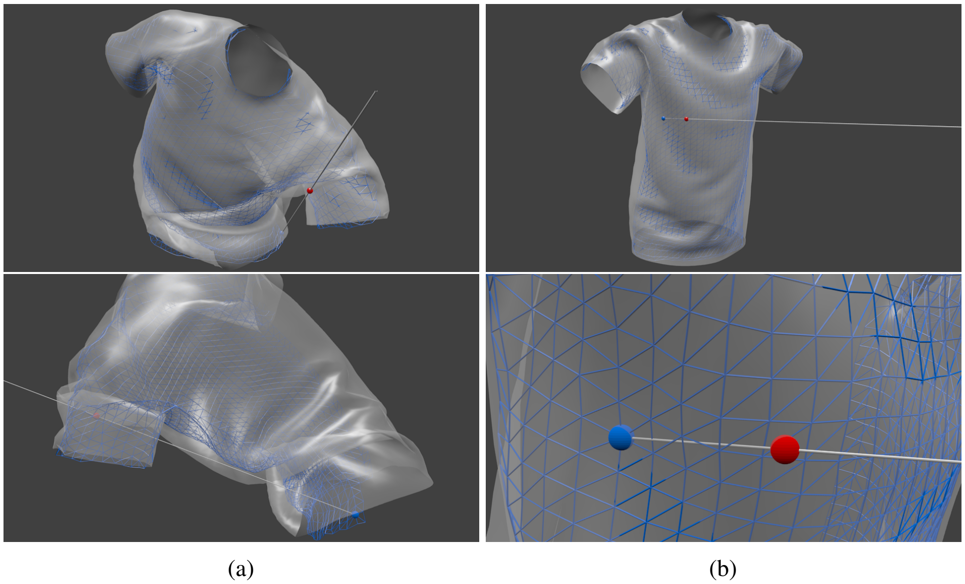


Figure 6: The maximum texture coordinate displacement before (a) and after (b) removing vertices which connect geodesically far away regions. The inferred cloth vertices are drawn in blue, and the ground truth ray intersection points are drawn in red. The wireframe of the inferred cloth is in blue, and the ground truth cloth is in white.

D.1 Texture Coordinates

To quantify the worst case texture sliding scenarios, we first consider T_N for every pose θ_k and camera view v_p used in training. The edge with the largest difference in texture coordinates (as a proxy for geodesic distances) is shown in Figure 7. We do the same for Euclidean distances along every edge to connected vertices in Figure 8.

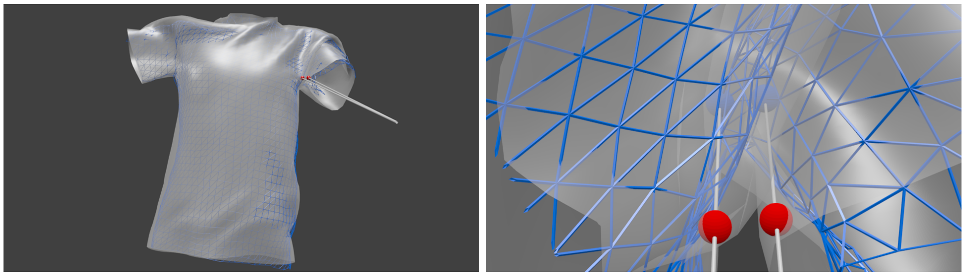


Figure 7: The edge over the entire training set with the largest change in texture coordinates.

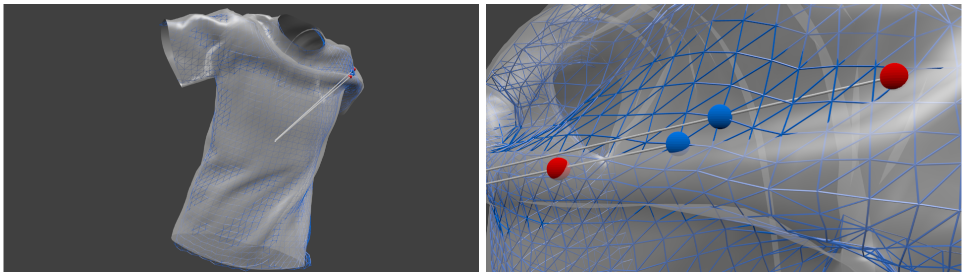


Figure 8: The edge over the entire training set with the largest ground truth intersection Euclidean distance.

D.2 Texture Coordinate Displacements

In order to fill in unassigned vertices for the patch under consideration, we show the most extreme behavior of texture sliding over all (θ_k, v_p). First, we compute the maximal value of $||d_{v_p}(\theta_k)||$ among all (θ_k, v_p) pairs, i.e. where maximal texture sliding occurs in our training set. See Figure 6b. We also compute $\Delta d_{v_p}(\theta_k)$ along each assigned edge in order to ascertain the biggest jump (indicating high frequency) that would be seen by the TSNN. The edge with the maximal $||\Delta d_{v_p}(\theta_k)||$ over all (θ_k, v_p) pairs is shown in Figure 9. Note that one may obtain better smoothness when training the TSNN by not assigning vertices where $d_{v_p}(\theta_k)$ is too large or one of the vertices of an edge where $\Delta d_{v_p}(\theta_k)$ is too large.

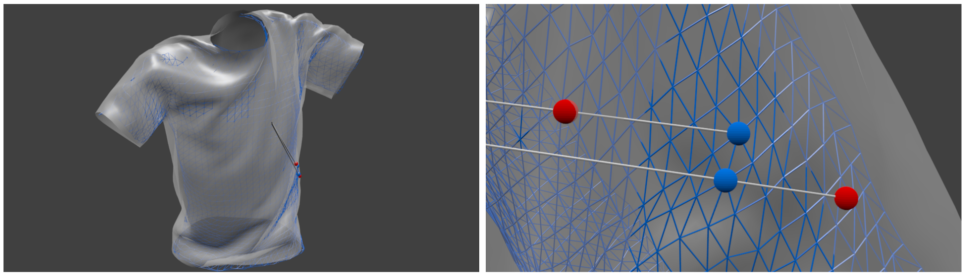


Figure 9: The edge over the entire training set with the largest change in texture coordinate displacements.

D.3 Smoothness Considerations

As long as extrapolation is done smoothly to assign texture coordinates to the remaining vertices, there should be no new extrema in $\Delta T_N(\theta_k, v_p)$ and $\Delta d_{v_p}(\theta_k)$. After applying smoothing, we verify that the largest $\Delta T_N(\theta_k, v_p)$ and $\Delta d_{v_p}(\theta_k)$ are the same as before.

E TSNN – Additional Experiments

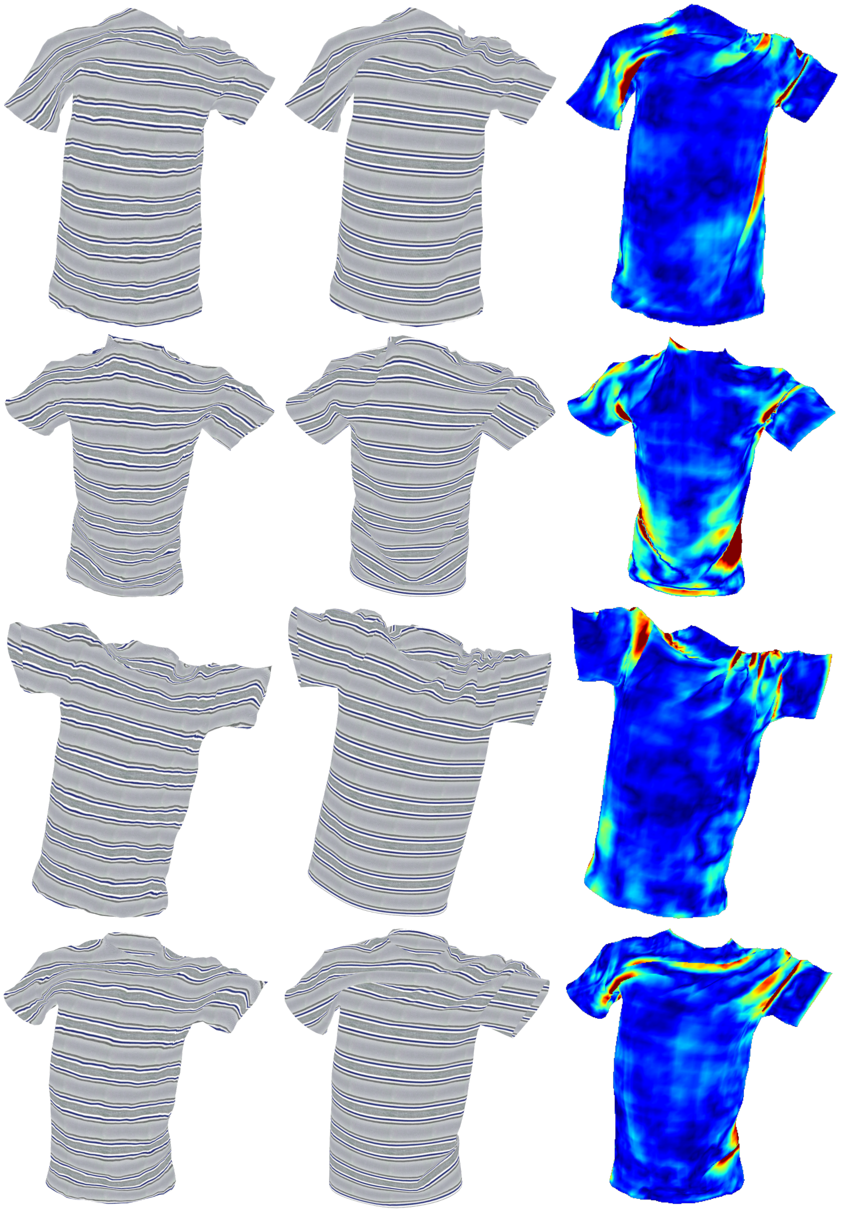
Additional TSNN inference results are shown in Figure 10. Table 1 shows additional TSNN results after applying a displacement threshold to the TS dataset. Subdivision has a larger impact on a coarse mesh; since the cloth mesh is relatively fine prior to subdivision, we do not see significant improvements in the TSNN results with the addition of subdivision when generating the ground truth data. In addition, in Table 2 we decompose the TSNN errors based on whether vertices were assigned via our ray intersection method or extrapolation. Results indicate that training separate networks for smooth and wrinkled regions of the cloth may be a promising avenue for future work.

Network	SqrtMSE ($\times 10^{-3}$)
TSNN	15.058 ± 6.5256
TSNN + subdivision	14.926 ± 6.5918

Table 1: Comparisons of per-pixel SqrtMSE for the test set after applying a threshold to the ground truth TS displacements.

	Original	Threshold
Ray Intersection	11.670 ± 3.2160	10.958 ± 2.9056
Extrapolation	39.564 ± 27.087	95.200 ± 48.406
Combination	14.279 ± 4.5970	15.405 ± 5.5923

Table 2: Breakdown of the TSNN errors ($\times 10^{-3}$) in Table 2 from the paper and Table 1 based on whether each pixel contains vertices assigned via ray intersection, diffusion, or a combination of both.



(a) $\hat{C}'_N$ (b) C'_N (c) $Error(\hat{C}'_N, C_G)$

Figure 10: Additional test set example predictions. The per-pixel errors are shown in (c) (blue $= 0$, red ≥ 0.04).

REFERENCES

Zhenglin Geng, Daniel Johnson, and Ronald Fedkiw. Coercing machine learning to output physically accurate results. *Journal of Computational Physics*, 406:109099, 2020.