Temporally-Coherent Surface Reconstruction via Metric-Consistent Atlases

Supplementary Material

Jan Bednarik[1] Vladimir G. Kim[2] Siddhartha Chaudhuri[2] Shaifali Parashar[1]
Mathieu Salzmann[1] Pascal Fua[1] Noam Aigerman[2]

[1]EPFL, Lausanne, Switzerland [2]Adobe Research

1. Training and Evaluation Details

We provide details of the triplet sampling strategy used to train the cycle consistent point cloud deformation method [3] (CC) in Section 1.1, an analysis of the strategy used to evaluate the non-rigid ICP method [4] (nrICP) in Section 1.2, more information on the points sampling strategy used to evaluate all the atlas-based methods, i.e. AtlasNet [2] (AN), Differential Surface Representation [1] (DSR) and our method (OUR), in Section 1.3 and a time complexity analysis in Section 1.4.

1.1. Details of Training CC

The training of CC relies on sampling triplets of shapes from the given dataset. The authors argue that the best results were achieved when sampling triplets of shapes that are close to each other in the Chamfer distance (CD) sense. Specifically, given a randomly sampled shape A, two other shapes B, C are randomly sampled from the 20 nearest neighbors of A to complete the triplet. Let us refer to this sampling strategy as *knn*.

OUR itself relies on sampling shape pairs, and as shown in Section 4.3 of the main paper, better results are achieved when sampling the shape pairs from a time window δ of a given sequence (*neighbors*) rather sampling pairs randomly within a sequence (*random*).

For fair comparison, we experimented with training CC using all three strategies, *knn*, *neighbors* and *random*. Table 1 reports the results on the DFAUST dataset using the validation sequence `jumping_jacks` and one more randomly chosen sequence `jiggle_on_toes`. Since CC performs best by a large margin when trained using *random*, we use this strategy for all the experiments.

Table 1. **Comparison of different triplet sampling strategies to train CC**. The experiments were conducted using DFAUST.

sequence	sampling	$m_{sL2} \downarrow$	$m_r \downarrow$	$m_{\text{AUC}} \uparrow$
jumping_jacks	knn	105.57±217.32	6.43±18.26	77.06±20.18
	neighbors	95.13±179.81	6.21±17.42	75.84±20.78
	random	**32.74±31.65**	**0.70±1.68**	**91.47±6.25**
jiggle_on_toes	knn	71.73±203.46	4.36±16.88	87.77±20.92
	neighbors	47.69±99.55	2.15±8.48	88.31±12.20
	random	**26.26±69.02**	**0.91±5.71**	**94.86±10.98**

1.2. Analysis of the nrICP [4] Strategy

The nrICP method deforms a point cloud to best match another point cloud and thus can be used to find point-wise correspondences in an unsupervised way. Formally, let ν_{P_j} be the non-rigid ICP function which deforms an input point cloud P_i to best match P_j. Following the notation introduced in Section 4.1 of the main paper, let $\pi_{\mathcal{X}}$ be a mapping that projects the points from an input point cloud to their respective nearest neighbors in the target point cloud $\mathcal{X}$. The simplest way to use nrICP to find correspondences between a pair of point clouds (P_i, P_j) randomly drawn from the given sequence is to compute $\pi_{P_j} \circ \nu_{P_j}(P_i)$. Let us call this strategy *random*.

Non-rigid ICP tends to break when the deformation between the two point clouds is severe. However, as we are dealing with sequences depicting a deforming shape, one can compute the correspondences between a pair of point clouds (P_i, P_j) by first predicting the correspondences for consecutive pairs of point clouds where the deformation is minimal, i.e., $(P_i, P_{i+1}), (P_{i+1}, P_{i+1}), \ldots, (P_{j-1}, P_j)$, and finally propagating the correspondences from P_i to P_j. Formally, we compute $\pi_{P_j} \circ \nu_{P_j} \circ \cdots \circ \pi_{P_{i+2}} \circ \nu_{P_{i+2}} \circ \pi_{P_{i+1}} \circ \nu_{P_{i+1}}(P_i)$ and refer to this strategy as *propagate_simple*.

The drawback of *propagate_simple* is that every mapping π_{P_k} is onto and thus throughout the propagation, progressively more source points get mapped to the same target point, which causes a loss of spatial information and ultimately yields less precise correspondences. To overcome this problem, one can replace π_{P_k} with ρ_{P_k}, which performs a Hungarian matching of the input point cloud and the target point cloud P_k with the objective of minimizing the overall per-point-pair distance. Formally, we compute $\rho_{P_j} \circ \nu_{P_j} \circ \cdots \circ \rho_{P_{i+2}} \circ \nu_{P_{i+2}} \circ \rho_{P_{i+1}} \circ \nu_{P_{i+1}}(P_i)$ and call this strategy *propagate_bijective*.

Finally, an alternative option is not to perform any projection π_{P_k} or ρ_{P_k} as we propagate the correspondences from P_i to P_j, but instead to gradually deform the input point cloud P_i to best match each point cloud along the sequence between P_i and P_j. Formally, we compute $\nu_{P_j} \circ \ldots \nu_{P_{i+2}} \circ \nu_{P_{i+1}}(P_i)$ and refer to this strategy as *propagate_deform*.

Table 2 reports the results of all four aforementioned correspondence estimation strategies on the `crane` validation sequence from the AMA dataset. We found that *propagate_simple* suffers from the loss of spatial precision due to the onto mapping. While *propagate_bijective* overcomes this problem, the Hungarian matching introduces a strong drift along the sequence yielding even worse overall correspondences. The strategy *propagate_deform* performs the best out of all three propagation-based strategies, but is still outperformed by the simplest strategy *random*. Therefore, as *random* yields the highest correspondence accuracy, we use it to evaluate nrICP on all datasets.

Table 2. **Comparison of strategies used to establish correspondences with nrICP**. The experiments were conducted on the `crane` validation sequence from the AMA dataset.

strategy	$m_{sL2} \downarrow$	$m_r \downarrow$	$m_{\text{AUC}} \uparrow$
random	**172.55±167.76**	**7.83±12.56**	**41.61±19.29**
propagate_simple	211.11±147.43	9.38±10.32	23.01±18.31
propagate_bijective	213.87±169.00	10.55±13.99	25.31±17.80
propagate_deform	206.64±150.45	10.40±13.35	25.41±16.55

1.3. Point Sampling in Atlas Based Methods

The original AtlasNet work [2] argues that better reconstruction accuracy is achieved if the 2D points sampled from the UV domain Ω are spaced on a regular grid. As explained in Section 4.1 of the main paper, at evaluation time each atlas based method, i.e., AN, DSR and OUR, predicts $N = 3125$ points. Due to the unknown number of collapsed patches, which are discarded at runtime, it might not be possible to evenly split N points into P non-collapsed patches so that the points would form a regular grid in the UV space Ω.

Therefore, instead of using a regular grid, we distribute the given available number of points as regularly as possible in the 2D domain using a simulated annealing based algorithm. The points are initially distributed uniformly at random, and then their position is iteratively adjusted so that every point maximizes its distance to the nearest points. This procedure is summarized in Algorithm 1. The difference between random and as regular as possible 2D points sampling is demonstrated in Fig. 1.

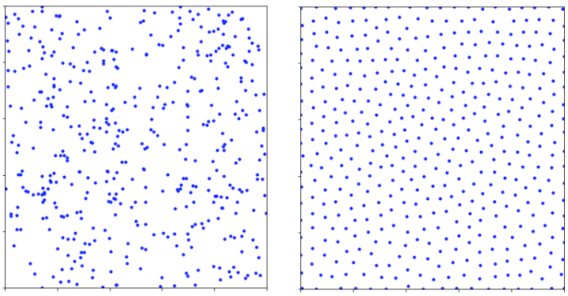
1.4. Time Complexity

The optimization of all the learning based methods was performed using an Nvidia Tesla V100 GPU, and processing a sequence of average length takes 16.1 hours for OUR, while AN, DSR and CC take 4.1, 16.4 and 9.7 hours, respectively. nrICP does not involve the optimization stage and can process ~ 1 sample per second.

Algorithm 1: As regular as possible 2D points.

Input: $M \in \mathbb{N}$ // Number of 2D points.
Output: $p_i \in \mathbb{R}^2, \forall 1 \leq i \leq M$ // 2D points.

/* Initialization */
step := $\frac{1}{4\sqrt{M}}$
decay := 0.994
$p_i \sim \mathcal{U}(\mathbf{0}, \mathbf{1}), \forall 1 \leq i \leq M$ // Random 2D points.
iter := 0
/* Main algorithm. */
while *iter* < 250 **do**
  for $i := 1$ **to** M **do**
    $d_i := \min_{j \neq i} ||p_i - p_j||$
    $\alpha_i \sim \mathcal{U}(0, 2\pi)$
    $p_i^{\text{new}} := p_i + \text{step} \cdot R(\alpha_i) \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ // R: rot. matrix
    $d_i^{\text{new}} := \min_{j \neq i} ||p_i^{\text{new}} - p_j^{\text{new}}||$
    if $d_i^{new} > d_i$ **then**
      $p_i := p_i^{\text{new}}$
  step := step · decay
  iter := iter +1



(a) (b)

Figure 1. **Comparison of (a) uniform and (b) as regular as possible 2D points sampling.**

2. Complete Results

We provide details of the search for the best value of the hyper-parameter δ in Section 2.1 and we list the complete per-sequence results of all the evaluated methods on all the datasets in Section 2.3. Furthermore, we refer the reader to the supplementary video[1] which contains the comparison of all methods on multiple sequences from all the datasets.

[1] https://www.youtube.com/watch?v=U89suo6MpIo

2.1. Tuning the Time-Window δ

As described in Section 3.3 of the main paper, OUR relies on sampling pairs of shapes from a time window denoted as δ. We tuned this hyper-paramater individually for every dataset using a respective validation sequence, and set it to the values yielding the best correspondence accuracy as measured by the metrics m_{sL2}, m_r and m_{AUC}. Table 3 lists the results of training OUR for $\delta \in [1, 6]$ and justifies the selection of $\delta = 1$ for ANIM, $\delta = 1$ for AMA and $\delta = 5$ for DFAUST.

Note that, as the ANIM and AMA datasets appear to have lower frame-rates than the DFAUST dataset, i.e., the surface undergoes larger motion from frame to frame, the correspondence error clearly decreases with the decreasing size of the time window δ, indicating that our method benefits from observing pairs of shapes which are similar enough to each other. On the other hand, as the DFAUST dataset in general exhibits small frame to frame changes, the search reveals that our method can benefit from observing pairs from larger time windows, since in this case the consecutive frames are nearly identical and decreasing δ makes $\mathcal{L}_{\text{mc}}$ less useful. Note, however, that using the value $\delta = 1$ for all the sequences shown in this paper still consistently outperforms all the competing methods.

Table 3. **Search for the best value of the hyper-parameter δ used by OUR on each dataset**.

dataset	neigh.	$m_{sL2} \downarrow$	$m_r \downarrow$	$m_{\text{AUC}} \uparrow$	CD $\downarrow$
ANIM (cat)	**1**	**9.80±14.36**	0.24±0.60	**98.27±0.82**	0.39±0.00
	2	10.27±15.03	0.24±0.54	98.09±1.01	0.39±0.00
	3	10.07±15.52	**0.23±0.56**	98.06±0.94	**0.38±0.00**
	4	17.10±37.51	0.78±3.27	94.49±4.60	0.38±0.01
	5	44.58±88.60	3.45±10.04	85.76±11.61	0.41±0.00
	6	11.45±16.33	0.30±0.66	97.78±1.03	0.39±0.00
AMA (crane)	**1**	**66.63±103.11**	**2.11±6.75**	**80.24±11.42**	**0.31±0.02**
	2	99.86±163.94	4.31±10.82	76.91±17.36	**0.31±0.02**
	3	91.09±138.11	3.68±9.22	74.42±16.49	0.32±0.01
	4	81.15±130.99	3.02±8.67	77.58±13.50	0.33±0.01
	5	106.34±166.29	4.61±10.85	74.10±17.69	0.34±0.02
	6	113.02±162.47	5.16±11.91	68.48±20.39	0.35±0.09
DFAUST (jumping_jacks)	1	32.71±46.68	0.92±3.15	91.77±4.53	0.51±0.09
	2	32.01±51.48	0.89±3.50	92.60±3.87	0.48±0.11
	3	29.39±33.80	0.73±2.25	93.30±2.86	0.50±0.09
	4	30.67±45.30	0.92±3.32	92.38±3.77	0.55±0.15
	5	**27.98±38.15**	**0.67±2.55**	**93.65±3.15**	**0.41±0.08**
	6	29.80±51.77	0.84±3.56	93.06±3.76	0.48±0.09

2.2. Impact of α_{mc} on the Visual Quality

As shown in Table 3 of the main paper, every dataset benefits from a different value of the hyper-parameter α_{mc} which balances metric consistency and Chamfer distance, while $\alpha_{\text{mc}} \in [0.1, 1.0]$ yields the best quantitative results. Here we show that this fact manifests in the qualitative results as well. The sequence `crane` from AMA is one case where setting $\alpha_{\text{mc}} = 1$ instead of 0.1 yields better quantitative results. However, both reconstructions are visually comparable, as shown in Fig. 2.

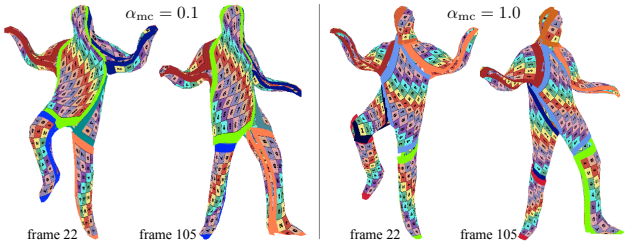


Figure 2. **Comparison of the reconstruction and correspondence quality when using $\alpha_{\text{mc}} = 0.1$ and $\alpha_{\text{mc}} = 1.0$.** The sample pair comes from the sequence `crane` of AMA. Note that these are two independent runs, therefore, the spatial distribution of the patches is arbitrary.

2.3. Evaluation on all Datasets and Stress Test

For brevity, Section 4.3 of the main paper only reports the mean results computed over all the sequences contained in the individual datasets. Here we report detailed results for each sequence separately. The results of all methods evaluated on the ANIM, AMA and DFAUST datasets are summarized in Tables 4, 5 and 6, respectively. Note that the average values reported in the last cell in each table are computed on all the test sequences, i.e., excluding the validation sequence `cat` in ANIM, `crane` in AMA and `jumping_jacks` in DFAUST.

Finally, Table 7 shows the results on the `horse_collapse` sequence used for the stress test of our method, as reported in Section 4.3 of the main paper, and an additional similar sequence `camel_collapse`. Both sequences come from the same work of [5] as the sequences `horse`, `camel` and `elephant` from the ANIM dataset, and thus we preprocess them in the same way, i.e., by scaling each sample so that the first frame of each sequence fits in a unit cube.

References

[1] Jan Bednarik, Shaifali Parashar, Erhan Gundogdu, Mathieu Salzmann, and Pascal Fua. Shape reconstruction by learning differentiable surface representations. In *CVPR*, 2020. 1

[2] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry. Atlasnet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018. 1, 2

[3] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Unsupervised cycle-consistent deformation for shape matching. *Computer Graphics Forum*, 2019. 1

[4] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G. Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences

Table 4. **Comparison of OUR to SotA methods on correspondence accuracy and reconstruction quality on the ANIM dataset**. Our method is the most accurate and also yields the same reconstruction quality as AN.

sequence	model	$m_{sL2} \downarrow$	$m_r \downarrow$	$m_{\text{AUC}} \uparrow$	CD $\downarrow$
cat	nrICP	77.96±94.55	5.72±9.65	70.39±15.02	-
	AN	14.27±18.96	0.41±1.04	97.07±1.18	**0.38±0.00**
	DSR	48.05±80.24	3.05±7.62	82.53±11.60	0.41±0.01
	CC	53.37±93.99	3.90±8.83	80.54±14.99	-
	OUR	**9.80±14.36**	**0.24±0.60**	**98.27±0.82**	0.39±0.00
horse	nrICP	69.94±76.34	4.91±7.11	72.62±13.23	-
	AN	17.52±27.33	0.66±3.01	96.60±1.24	**0.09±0.00**
	DSR	40.21±59.73	2.20±4.74	84.31±11.61	0.22±0.01
	CC	30.24±57.97	1.60±4.30	88.39±7.95	-
	OUR	**12.97±12.81**	**0.31±0.55**	**97.82±0.85**	0.10±0.00
camel	nrICP	78.48±108.22	7.45±12.29	73.59±14.99	**-**
	AN	16.29±16.93	0.86±1.65	96.90±1.27	0.10±0.00
	DSR	75.93±116.15	7.43±13.56	73.80±13.47	0.17±0.02
	CC	56.62±93.14	4.75±9.32	77.70±14.57	-
	OUR	**11.08±10.72**	**0.42±0.83**	**98.19±0.53**	**0.09±0.00**
elephant	nrICP	62.54±70.03	4.01±9.17	76.47±12.82	-
	AN	21.39±30.20	0.82±3.88	95.35±2.17	0.09±0.01
	DSR	23.16±26.39	0.68±1.82	93.77±3.66	0.19±0.00
	CC	14.65±11.27	0.27±0.63	97.78±0.75	-
	OUR	**11.73±9.47**	**0.16±0.33**	**98.30±0.45**	**0.08±0.00**
MEAN	nrICP	70.32±84.86	5.46±9.52	74.23±13.68	-
	AN	18.40±24.82	0.78±2.85	96.28±1.56	**0.09±0.00**
	DSR	46.43±67.42	3.44±6.71	83.96±9.58	0.19±0.01
	CC	33.84±54.13	2.21±4.75	87.96±7.76	-
	OUR	**11.93±11.00**	**0.30±0.57**	**98.10±0.61**	**0.09±0.00**

with multiview convolutional networks. *ACM Trans. Graph.*, 2017. 1

[5] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 2004. 3

Table 5. **Comparison of OUR to SotA methods on correspondence accuracy and reconstruction quality on the AMA dataset**. Our method is the most accurate and also yields reconstruction quality competitive with AN.

sequence	model	$m_{sL2}\downarrow$	$m_r\downarrow$	$m_{\text{AUC}}\uparrow$	CD $\downarrow$	sequence	model	$m_{sL2}\downarrow$	$m_r\downarrow$	$m_{\text{AUC}}\uparrow$	CD $\downarrow$
bouncing	nrICP	130.93±111.27	4.11±7.76	43.64±17.60	-	march_2	nrICP	187.52±182.98	9.33±14.65	39.92±23.08	-
	AN	59.86±50.14	0.92±3.19	77.84±7.56	**0.37±0.02**		AN	120.96±166.77	5.03±11.33	64.99±18.34	**0.30±0.02**
	DSR	51.09±34.89	0.56±0.96	81.57±7.00	0.35±0.01		DSR	143.05±144.18	5.76±10.75	47.77±20.80	0.96±0.15
	CC	45.21±28.94	**0.40±0.60**	85.56±6.59	-		CC	118.02±160.88	5.23±11.34	65.18±20.49	-
	OUR	**44.31±29.47**	0.41±0.71	**85.92±6.06**	0.40±0.03		**OUR**	**90.62±156.21**	**3.80±11.09**	**77.99±18.38**	0.36±0.03
crane	nrICP	172.55±167.76	7.83±12.56	41.61±19.29	-	samba	nrICP	100.50±91.01	3.58±6.17	58.47±23.58	-
	AN	77.64±99.96	2.13±6.27	74.08±12.38	**0.30±0.01**		AN	75.38±83.78	2.41±5.42	72.22±20.14	**0.22±0.01**
	DSR	128.18±168.36	5.64±11.18	64.07±23.11	0.29±0.01		DSR	114.61±124.73	5.06±9.48	57.38±26.40	0.25±0.01
	CC	76.80±96.52	2.17±5.85	73.49±18.45	-		CC	72.31±85.51	2.34±5.49	73.72±20.53	-
	OUR	**66.63±103.11**	**2.11±6.75**	**80.24±11.42**	0.31±0.02		**OUR**	**62.84±80.00**	**1.93±4.64**	**77.44±21.38**	0.27±0.02
handstand	nrICP	256.22±194.98	14.16±17.45	21.87±24.70	-	squat_1	nrICP	84.88±86.77	2.23±4.93	65.16±26.07	-
	AN	182.36±186.97	8.74±14.38	42.11±26.55	**0.35±0.03**		AN	46.91±41.54	0.71±1.59	82.91±12.40	0.28±0.01
	DSR	380.79±226.71	20.52±17.35	4.54±1.59	551.20±471.57		DSR	46.91±41.82	0.65±1.52	82.67±13.36	0.28±0.01
	CC	**89.79±142.39**	**3.11±10.10**	**71.53±16.92**	-		CC	**26.81±18.42**	**0.16±0.25**	**94.32±2.83**	-
	OUR	126.52±167.36	5.69±13.80	58.47±23.49	0.38±0.03		**OUR**	27.81±27.48	0.25±0.78	92.60±4.73	**0.27±0.00**
jumping	nrICP	206.43±172.56	10.38±14.50	29.02±18.96	-	squat_2	nrICP	90.50±85.60	2.29±4.77	61.18±25.81	-
	AN	116.86±148.11	4.86±11.78	61.05±18.89	**0.32±0.02**		AN	47.93±42.15	0.66±1.63	82.61±10.97	0.29±0.01
	DSR	114.90±151.03	4.92±11.16	64.81±19.44	0.33±0.02		DSR	121.50±119.10	3.75±6.51	51.09±27.88	4.77±0.71
	CC	77.53±111.13	2.29±6.71	75.31±17.45	-		CC	37.02±28.14	**0.32±0.58**	89.14±7.37	-
	OUR	**45.10±31.25**	**0.50±1.01**	**85.31±6.01**	0.37±0.03		**OUR**	**33.72±31.41**	**0.32±0.81**	**89.85±7.06**	**0.28±0.01**
march_1	nrICP	174.05±171.86	8.59±14.23	42.75±22.83	-	swing	nrICP	127.41±111.79	4.98±7.89	46.58±17.80	-
	AN	57.66±42.75	0.93±1.99	77.90±9.56	**0.29±0.01**		AN	73.26±59.32	1.86±4.30	69.01±13.40	**0.24±0.02**
	DSR	79.26±97.42	2.52±6.44	72.20±14.13	0.33±0.02		DSR	59.92±49.39	1.25±2.38	75.19±12.90	**0.24±0.01**
	CC	124.76±157.34	5.26±10.01	64.16±23.63	-		CC	79.78±149.09	3.13±12.27	74.73±19.23	-
	OUR	**34.85±24.72**	**0.29±0.56**	**90.66±4.66**	0.31±0.01		**OUR**	**48.28±40.07**	**0.80±1.67**	**82.39±8.68**	0.26±0.01
						MEAN	nrICP	150.94±134.31	6.63±10.26	45.40±22.27	-
							AN	86.80±91.28	2.90±6.18	70.07±15.31	**0.30±0.01**
							DSR	123.56±109.92	5.00±7.39	59.69±15.94	62.08±52.50
							CC	74.58±97.98	2.47±6.37	77.07±15.00	-
							OUR	**57.12±65.33**	**1.55±3.90**	**82.29±11.16**	0.32±0.02

Table 6. **Comparison of OUR to SotA methods on correspondence accuracy and reconstruction quality on the DFAUST dataset**. Our method is the most accurate and also yields reconstruction quality on par with AN.

sequence	model	$m_{sL2} \downarrow$	$m_r \downarrow$	$m_{\text{AUC}} \uparrow$	CD $\downarrow$	sequence	model	$m_{sL2} \downarrow$	$m_r \downarrow$	$m_{\text{AUC}} \uparrow$	CD $\downarrow$
chicken_wings	nrICP	72.20±135.09	5.15±15.70	80.89±13.79	-	one_leg_jump	nrICP	81.13±98.03	2.86±5.19	71.68±14.61	-
	AN	49.75±92.05	2.31±6.96	85.57±16.07	**0.37±0.08**		AN	30.65±23.71	0.58±0.98	**92.47±3.48**	0.43±0.08
	DSR	282.61±130.70	21.85±17.43	6.40±2.66	97.10±25.04		DSR	47.59±99.20	1.30±4.72	88.44±8.96	**0.36±0.07**
	CC	35.57±113.74	2.12±11.40	93.59±16.90	-		CC	**36.40±76.87**	**0.79±3.47**	91.23±8.64	-
	OUR	**18.51±20.37**	**0.38±1.34**	**96.53±1.98**	0.43±0.09		**OUR**	41.44±74.93	1.03±3.56	87.92±11.02	**0.36±0.07**
hips	nrICP	59.22±49.65	2.39±3.91	76.95±14.85	-	one_leg_loose	nrICP	54.98±64.48	1.74±3.31	81.73±11.45	-
	AN	21.16±16.45	0.33±0.60	96.06±1.58	0.27±0.05		AN	31.92±35.99	0.68±1.77	91.35±6.00	**0.38±0.07**
	DSR	17.33±14.65	0.29±0.62	97.11±1.10	0.28±0.04		DSR	23.53±24.66	0.41±0.80	94.93±3.59	0.43±0.09
	CC	41.57±138.23	2.19±12.00	93.30±16.63	-		CC	23.90±31.93	0.51±1.85	94.61±11.02	-
	OUR	**14.76±12.21**	**0.23±0.51**	**97.83±0.77**	**0.25±0.04**		**OUR**	**18.40±15.28**	**0.29±0.53**	**96.92±1.61**	0.41±0.07
jiggle_on_toes	nrICP	128.60±194.22	7.46±16.20	59.47±25.80	-	punching	nrICP	117.92±175.81	8.29±18.23	65.40±19.74	-
	AN	35.43±72.20	1.34±5.59	90.04±9.74	0.31±0.05		AN	43.01±64.98	1.70±6.07	87.19±9.49	0.36±0.06
	DSR	29.01±47.00	0.95±3.74	92.70±7.63	0.38±0.05		DSR	204.86±141.43	13.04±15.68	15.99±4.52	39.66±12.26
	CC	26.26±69.02	0.91±5.71	94.86±10.98	-		CC	**21.13±17.43**	**0.41±0.89**	**95.94±2.14**	-
	OUR	**16.45±14.24**	**0.28±0.69**	**97.38±1.31**	**0.30±0.05**		**OUR**	21.88±24.82	0.50±1.44	95.51±2.90	**0.32±0.06**
jumping jacks	nrICP	187.37±240.34	11.78±20.47	46.37±28.06	-	running_on_spot	nrICP	103.57±129.95	5.57±12.83	65.36±15.24	-
	AN	51.76±60.73	1.95±4.67	81.68±10.34	0.52±0.08		AN	39.77±46.21	1.18±4.33	88.93±4.76	0.49±0.09
	DSR	35.25±55.18	1.07±4.01	90.60±5.07	0.43±0.05		DSR	43.93±58.32	1.57±4.44	86.86±8.58	0.45±0.05
	CC	32.74±31.65	0.70±1.68	91.47±6.25	-		CC	26.64±20.92	0.52±1.25	94.38±3.04	-
	OUR	**27.98±38.15**	**0.67±2.55**	**93.65±3.15**	**0.41±0.08**		**OUR**	**21.28±18.99**	**0.39±1.39**	**96.07±1.64**	**0.32±0.05**
knees	nrICP	116.26±157.30	5.22±10.39	61.25±20.85	-	shake_arms	nrICP	88.38±158.81	5.40±14.90	75.74±18.18	-
	AN	43.33±81.94	1.02±3.72	89.00±5.32	**0.40±0.09**		AN	28.13±31.89	0.81±2.35	92.92±4.44	**0.34±0.06**
	DSR	24.76±22.94	0.43±0.79	94.25±2.69	0.47±0.11		DSR	25.99±27.03	0.77±1.85	93.67±3.30	0.50±0.12
	CC	30.38±86.52	1.14±7.12	94.47±13.35	-		CC	21.82±18.50	0.51±1.02	95.61±2.01	-
	OUR	**23.45±19.49**	**0.40±0.74**	**95.23±2.37**	0.52±0.12		**OUR**	**17.29±19.21**	**0.41±1.34**	**96.97±1.31**	0.36±0.08
light_hopping_loose	nrICP	41.15±29.98	1.29±2.24	87.08±6.84	-	shake_hips	nrICP	86.76±156.94	5.00±14.35	76.47±17.42	-
	AN	21.46±21.06	0.39±1.14	95.83±2.04	**0.28±0.05**		AN	28.27±28.74	0.71±1.77	92.46±5.48	0.29±0.05
	DSR	21.81±20.35	0.51±1.18	95.48±2.91	0.40±0.05		DSR	92.23±158.27	5.22±13.78	75.27±19.64	1.28±0.37
	CC	25.61±53.71	0.82±4.13	94.66±12.36	-		CC	49.93±160.68	3.21±15.05	91.60±19.67	-
	OUR	**16.12±12.56**	**0.27±0.54**	**97.56±0.96**	0.32±0.06		**OUR**	**17.60±16.05**	**0.29±0.89**	**96.90±1.73**	**0.28±0.05**
light_hopping_stiff	nrICP	33.16±24.91	0.93±1.80	91.11±3.93	-	shake_shoulders	nrICP	53.85±42.97	1.90±2.84	79.19±11.83	-
	AN	17.30±15.16	0.25±0.62	97.05±1.25	0.28±0.05		AN	22.48±17.33	0.39±0.68	95.63±1.78	**0.29±0.05**
	DSR	58.32±36.51	2.05±2.81	77.38±14.48	4.11±2.07		DSR	22.27±18.30	0.43±0.78	95.48±2.30	0.31±0.04
	CC	25.50±79.93	1.14±7.54	95.63±13.47	-		CC	19.67±14.47	**0.31±0.51**	96.63±1.08	-
	OUR	**12.21±9.95**	**0.16±0.30**	**98.40±0.39**	**0.27±0.05**		**OUR**	**18.08±14.37**	0.32±0.58	**96.97±1.19**	0.30±0.05
						MEAN	nrICP	79.78±118.46	4.09±10.17	74.79±15.90	-
							AN	31.74±43.46	0.90±2.95	91.88±5.84	**0.34±0.06**
							DSR	68.79±61.04	3.76±5.19	78.00±6.25	11.21±2.89
							CC	29.57±65.26	1.12±5.26	94.35±9.82	-
							OUR	**19.81±22.19**	**0.38±1.17**	**96.17±2.31**	**0.34±0.06**

Table 7. **Comparison of OUR to SotA methods on correspondence accuracy and reconstruction quality on a collapsing rubber horse used to stress test our method and on an additional similar sequence depicting a collapsing camel**. Our method is the most accurate and also yields the same reconstruction quality as AN.

sequence	model	$m_{sL2} \downarrow$	$m_r \downarrow$	$m_{\text{AUC}} \uparrow$	CD $\downarrow$
horse_collapse	nrICP	54.32±46.49	3.88±5.47	78.36±13.66	-
	AN	62.38±79.58	4.91±9.05	74.81±18.23	**0.13±0.01**
	DSR	49.00±60.40	3.25±6.18	81.73±14.42	0.18±0.02
	CC	56.51±78.80	3.89±7.49	77.97±19.67	-
	OUR	**23.82±39.39**	**1.11±3.48**	**93.32±6.48**	**0.13±0.01**
camel_collapse	nrICP	40.68±36.05	2.76±3.69	86.60±9.77	-
	AN	43.78±61.53	3.05±6.11	85.05±12.47	0.16±0.01
	DSR	67.16±96.21	5.12±8.99	75.66±19.57	0.25±0.02
	CC	349.08±371.70	33.96±37.24	48.85±48.25	-
	OUR	**19.25±28.05**	**0.81±2.09**	**95.72±3.89**	**0.15±0.01**
MEAN	nrICP	47.50±41.27	3.32±4.58	82.48±11.72	-
	AN	53.08±70.56	3.98±7.58	79.93±15.35	**0.14±0.01**
	DSR	58.08±78.31	4.19±7.59	78.70±17.00	0.21±0.02
	CC	202.80±225.25	18.93±22.37	63.41±33.96	-
	OUR	**21.54±33.72**	**0.96±2.79**	**94.52±5.19**	**0.14±0.01**