8 Acknowledgments

We would like to acknowledge Hugues Hoppe, Paul Lalonde, Erwin Coumans, Angjoo Kanazawa, Alec Jacobson, David Levine, and Christopher Batty for the insightful discussions. Gerard Pons-Moll is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180. Andrea Tagliasacchi is funded by NSERC Discovery grant RGPIN-2016-05786,, NSERC Collaborative Research and Development grant CRDPJ 537560-18, and NSERC Research Tool Instruments RTI-16-2018.

9 Supplementary material

9.1 Additional discussion on tracking techniques

In dense real-time tracking applications [44,41,52,53] there are examples of articulated models that use hybrid representations. Similarly to our work, they also provide constant-time occupancy queries without the need for acceleration data structures. However, these shape models consist of *simple rigid primitives*, or use an *artist-designed template*. In contrast, we learn a deformable shape model from data, and allow users to query the pose-corrected occupancy anywhere in space. In more detail:

- Schmidt et al. [44]: relies on a discretization of the SDF on grids and/or primitives to answer distance queries, only models piecewise *rigid* parts (compare our rigid (R) variant), and requires the construction of the parts SDF before tracking – a process that relies on user interaction.
- Tkach et al. [54]: the quality of approximation produced by our method is vastly superior to that achieved by sphere-meshes. Further, similarly to [44], the template is specified manually.
- Taylor et al. [52]: similarly to [44], the tracking template is specified manually, and the technique requires a *mixture* of mesh-based closest point queries and implicit function queries.
- Thiery et al. [53]: the approximation quality argument is analogous to that of [54], while to construct such models one need a consistently meshed motion sequence – an extremely stringent requirement in practice.

9.2 Ablation studies

Please see the animated results of *reconstruction* and *tracking* in the **supplementary video**. Please see the details of the dataset in the **supplementary data split files**. Note that, except Figure 9 where we use AMASS/Transitions due to its diversity of poses, we adopt AMASS/DFaust for all the other studies. Also note that due to computational limitations, we evaluate on *one* motion sequence only in Figure 10. We select a sequence that has a median reconstruction performance as a representative example.

$\mathcal{L}_{\text{occupancy}}$	mIoU↑	Chamfer L1↓	F%↑
Cross-Entropy	.959	.00006	98.01
L2	**.959**	**.00004**	**98.54**

Table 5: $\mathcal{L}_{\text{occupancy}}$

$\mathcal{L}_{\text{weights}}$	mIoU↑	Chamfer L1↓	F%↑
✗	.845	.00351	76.64
✓	**.959**	**.00004**	**98.54**

Table 6: $\mathcal{L}_{\text{weights}}$

Fig. 7: Ablation study of the loss used for fitting the occupancy function (L2 vs. binary cross-entropy), and the ablation study of the impact of the skinning weight loss in Eq. (10) on the right.

Losses ablation – Figure 7. One can view $\mathcal{O}(\mathbf{x}|\boldsymbol{\theta})$ as a binary classifier that aims to separate the interior of the shape from its exterior. Accordingly, one can use a binary cross-entropy loss for optimization, but our experiments suggest that an L2 loss perform slightly better. Hence, we employ the L2 loss for all of our experiments; see Table 5. We also validate the importance of the skinning weights loss in Table 6 and observe a big improvement when $\mathcal{L}_{\text{weights}}$ is included.

Model	mIoU↑	Chamfer L1↓	F%↑
R	.933	.00021	94.13
$D\backslash\Pi$	.926	.00023	92.23
D	**.959**	**.00004**	**98.54**

Table 7: Projection Π

D	1	2	4	8	16
mIoU↑	.955	.957	**.959**	.958	.957
Chamfer L1↓	.00130	.00004	.00004	.00199	.00004
F%↑	98.00	98.38	**98.54**	98.09	97.85

Table 8: Projection size D

Fig. 8: Ablation of our (per-part) linear subspace projection.

Linear subspace projection Π – Figure 8. Note that the rigid model (R) actually *outperforms* the deformable model (D) if one *removes* the learnt linear dimensionality reduction ($D\backslash\Pi$); see Table 7. This is a result only observed on the *test* set, while on the training set $D\backslash\Pi$ performs comparably. In other words, Π helps our model to achieve better *generalization* by enforcing a sparse representation of pose. In Table 8, we report the results of an ablation study on the dimensionality of the projection, which was the basis for the selection of D=4.

	$\boldsymbol{\theta}$	mIoU↑	ChamferL1↓	F%↑
D	$\{\mathbf{B}_b^{-1}\}$	.962	.00003	99.22
D	$\{\mathbf{B}_b^{-1}\mathbf{x}\}$	.959	.00003	98.86
D	$\{\mathbf{B}_b^{-1}\mathbf{t}_0\}$	**.965**	**.00002**	**99.42**

Table 9: $\boldsymbol{\theta}$ for D.

	MLP input	mIoU↑	ChamferL1↓	F%↑
U	$[\mathbf{x}, \{\mathbf{B}_b^{-1}\mathbf{t}_0\}]$	.520	.001057	26.83
U	$[\{\mathbf{B}_b^{-1}\mathbf{x}\}]$	.865	.00019	86.61
D	$[\{\mathbf{B}_b^{-1}\mathbf{x}\}, \{\mathbf{B}_b^{-1}\mathbf{t}_0\}]$	**.965**	**.00002**	**99.42**

Table 10: $[x, \boldsymbol{\theta}]$ for U model.

Fig. 9: Ablations of pose representations.

Analysis of pose representations – Figure 9. In Table 9, we ablate several representations for the pose $\boldsymbol{\theta}$ used by the deformable model. We start by

just using the *collection* of homogeneous transformations $\{\mathbf{B}_b^{-1}\}$. Note that the query point encoded in various coordinate frames is also an effective pose representation $\{\mathbf{B}_b^{-1}\mathbf{x}\}$, which has a much lower dimensionality. Finally, we notice that rather than using the query point, one can pick a fixed point to represent pose. While any fixed point can be used, we select the origin of the model $\mathbf{t}_0$ for simplicity, resulting in $\{\mathbf{B}_b^{-1}\mathbf{t}_0\}$. The resulting representation is *compact* and *effective*. Table 10 shows a similar analysis for the unstructured model (metrics for D provided for reference). Note how the performance of U can be significantly improved by providing the network with the encoding of the query point $\mathbf{x}$ in various coordinate frames – that is, the network is no longer required to "learn" the concept of changes of coordinates.

Model	24×**8**	24×**16**	24×**24**	24×**32**	24×**40**
U	.539	.538	.601	.642	.653
R	.913	.902	.931	.939	.946
D	**.917**	**.915**	**.946**	**.950**	**.952**

Table 11: IoU metric.

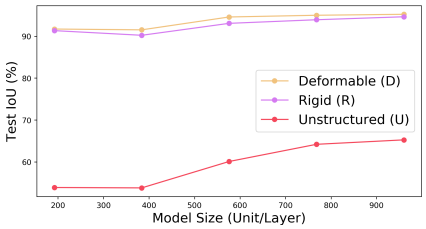

Fig. 10: We evaluate the performance of the model as we increase the number of units used for each of the 24 parts in the set $\{8, 16, 24, 32, 40\}$. The number of layers in each sub-network is held *fixed* to 4.

Analysis of model size – Figure 10. Both rigid (R) and deformable (D) models significantly outperform the results of the unstructured (U) model, as we increase the neural network's layer size and approach the network capacity employed by [9,39,37].

$p(\mathcal{O}\|\boldsymbol{\theta})$	$p(\boldsymbol{\theta})$	$\circledast$	mIoU↑	Chamfer L1↓	F%↑
D	✗	✗	.952	.00005	97.24
D	✗	✓	.948	.00005	97.50
D	✓	✗	.965	.00004	98.79
D	✓	✓	**.968**	**.00004**	**99.08**

Table 12: DFaust "easy" (00-01)

$p(\mathcal{O}\|\boldsymbol{\theta})$	$p(\boldsymbol{\theta})$	$\circledast$	mIoU↑	Chamfer L1↓	F%↑
D	✗	✗	.546	.01430	44.31
D	✗	✓	.891	.00032	86.15
D	✓	✗	.862	.00258	79.05
D	✓	✓	**.948**	**.00006**	**96.48**

Table 13: DFaust "hard" (02-09)

Fig. 11: Ablations for the tracking application

Tracking ablations – Figure 11. In the tracking application, we ablate with respect to the pose prior ($p(\boldsymbol{\theta})$) and the use of random perturbations to approximate the distance function via convolution ($\circledast$). First, note that the best results are achieved when both of these components are enabled, across *all* metrics. We compare the performance of our models on easy vs. hard sequences. Hard sequences more clearly illustrate the advantages of the algorithms proposed. We validate the usefulness of the pose prior in avoiding tracking failure (e.g. IoU : 44.31% → 86.15%). The use of random perturbations allow the optimization to converge more precisely (Chamfer: .00258 → .00006).

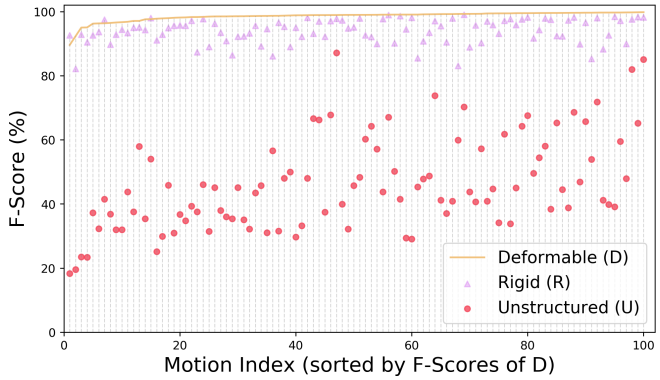

Fig. 12: Distribution of F-Score across the AMASS/DFaust dataset.

Metrics distribution on AMASS/DFaust. Rather than reporting aggregated statistics, we visualize the IoU errors of all of the 100 DFaust experiments, and sort them by the performance achieved by the deformable model (D). Note how the deformable model achieves consistent performance across the dataset. There are *only two sequences* where the rigid model performs better than the deformable model.