

4DComplete: Non-Rigid Motion Estimation Beyond the Observable Surface

Yang Li¹ Hikari Takehara² Takafumi Taketomi² Bo Zheng² Matthias Nießner³

¹The University of Tokyo ²Tokyo Research Center, Huawei ³Technical University Munich

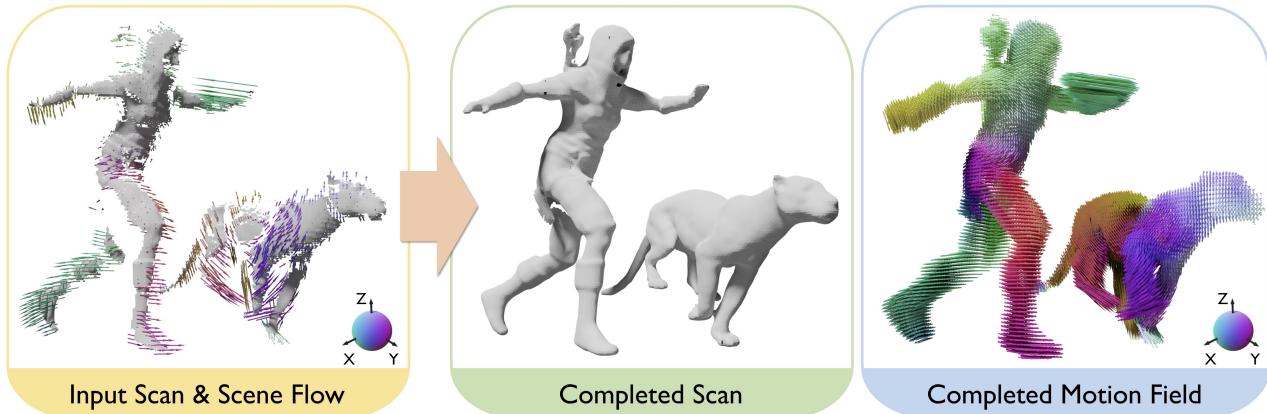


Figure 1: Given an input partial scan and inter-frame scene flow from a non-rigidly deforming scene (left), our method jointly recovers missing geometry (middle) and a volumetric motion field (right). The colors show the motion directions on the unit sphere (bottom right corner) where the vector's length corresponds to the magnitude of the motion.

Abstract

Tracking non-rigidly deforming scenes using range sensors has numerous applications including computer vision, AR/VR, and robotics. However, due to occlusions and physical limitations of range sensors, existing methods only handle the visible surface, thus causing discontinuities and incompleteness in the motion field. To this end, we introduce 4DComplete, a novel data-driven approach that estimates the non-rigid motion for the unobserved geometry. 4DComplete takes as input a partial shape and motion observation, extracts 4D time-space embedding, and jointly infers the missing geometry and motion field using a sparse fully-convolutional network. For network training, we constructed a large-scale synthetic dataset called DeformingThings4D, which consists of 1,972 animation sequences spanning 31 different animals or humanoid categories with dense 4D annotation. Experiments show that 4DComplete 1) reconstructs high-resolution volumetric shape and motion field from a partial observation, 2) learns an entangled 4D feature representation that benefits both shape and motion estimation, 3) yields more accurate and natural deformation than classic non-rigid priors such as As-Rigid-As-Possible (ARAP) deformation, and 4) generalizes well to unseen objects in real-world sequences.

1. Introduction

Understanding the motion of non-rigidly deforming scenes using a single range sensor lies at the core of many computer vision, AR/VR, and robotics applications. In this context, one fundamental limitation is that a single-view range sensor cannot capture data in occluded regions, leading to incomplete observations of a 3D environment. As a result, existing non-rigid motion tracking methods are restricted to the observable part of the scene. However, the ability to infer complete motion from a partial observation is indispensable for many high-level tasks. For instance, as a nursing robot, to safely care for an elderly person (e.g., predict the person’s action and react accordingly), it needs to understand both the complete body shape and how the whole body moves even if the person is always partially occluded.

In order to address these challenges, we pose the question *how can we infer the motion of the unobserved geometry in a non-rigidly deforming scene?* Existing works such as DynamicFusion [38] and VolumeDeform [26] propose to propagate deformations from the visible surface to the invisible space through a latent deformation graph. Hidden deformations are then determined by optimizing hand-crafted deformation priors such as As-Rigid-As-Possible [47] or Embedded Deformation [49], which enforces that graph vertices locally move in an approximately rigid manner.

Such deformation priors have several limitations: 1) they require heavy parameter tuning; 2) they do not always reflect natural deformations; and 3) they often assume a continuous surface. As a result, these priors are mostly used as regularizers for local deformations, but struggle with larger hidden regions. One promising avenue towards solving this problem is to leverage data-driven priors that learn to infer the missing geometry. Very recently, deep learning approaches for 3D shape or scene completion and other generative tasks involving a single depth image or room-scale scans have shown promising results [12, 46, 11, 7, 10]. However, these works primarily focus on static environments.

In this paper, we make the first effort to combine geometry completion with non-rigid motion tracking. We argue that the shape and motion of non-rigidly deforming objects are highly entangled data modalities: on one hand, the ability to infer the geometry of unobserved object parts provides valuable information for motion estimation. On the other hand, motion is considered as the shape’s evolution in the time axis, as similarity in motion patterns are a strong indicator for structural connectivity. To leverage these synergies, we propose 4DComplete, which jointly recovers the missing geometry and predicts motion for both seen and unseen regions. We build 4DComplete on a sparse, fully-convolutional neural network, which facilitates the joint estimation of shape and motion at high resolutions. In addition, we introduce DeformingThings4D, a new large-scale synthetic dataset which captures a variety of non-rigidly deforming objects including humanoids and animals. Our dataset provides holistic 4D ground truth with color, optical/scene flow, depth, signed distance representations, and volumetric motion fields.

In summary, we propose the following contributions:

- We introduce 4DComplete, the first method that jointly recovers the shape and motion field from partial observations.
- We demonstrate that these two tasks help each other, resulting in strong 4D feature representations outperforming existing baselines by a significant margin.
- We provide a large-scale non-rigid 4D dataset for training and benchmarking. The dataset consists of 1,972 animation sequences, and 122,365 frames. Dataset is available at: <https://github.com/rabbity1/DeformingThings4D>.

2. Related Work

2.1. Non-Rigid Tracking Using Depth Sensors

Many methods for non-rigid tracking use variations of the Non-rigid Iterative Closest Point (N-ICP) algorithm [1, 41, 29, 59], where the point-to-point or point-to-plane distance of correspondences points are iteratively

minimized. To prevent uncontrolled deformations and resolve motion ambiguities, the N-ICP optimization usually employs deformation regularizers such as As-Rigid-As-Possible (ARAP) [47] or embedded deformation [49]. One of the first real-time methods to jointly track and reconstruct non-rigid surfaces was DynamicFusion [38]. VolumeDeform [26] extends the ideas of DynamicFusion by adding sparse SIFT feature matches to improve tracking robustness. Using deep learning, DeepDeform [5] replaces the classical feature matching by CNN-based correspondence matching. Li et al. [30] goes one step further and differentiates through the N-ICP algorithm thus obtaining a dense feature matching term. A similar direction is taken by Neural Non-Rigid Tracking [4]; however, their focus lies in an end-to-end robust correspondence estimation. To handle topology changes, KillingFusion [45] directly estimates the motion field given a pair of signed distance fields (SDF). Optical/scene flow [15, 51, 50, 53, 32, 55, 33, 22, 34] is a closely related technique. They have been used to generate initial guess for non-rigid tracking in [17, 18, 4, 16, 54]. Among these works, FlowNet3D [32] is one of the first methods that directly estimates scene flow from two sets of point clouds. While existing methods mainly focus on the visible surface of a scene, we take one step further to model the deformation of the hidden surface.

2.2. Shape and Scene completion

Completing partial 3D scans is an active research area in geometry processing. Traditional methods, such as Poisson Surface Reconstruction [28], locally optimize for a surface to fit observed points and work well for small missing regions. Zheng et al. [57] predict the unobserved voxels by reasoning physics and Halimi et al. [24] complete partial human scans by deforming human templates. More recently, we have seen 3D CNNs with promising results for geometry completion for depth scans [46, 12, 11, 10]. These works operate either on a single depth image of a scene as with SSCNet [46], or scene completion on room- and building floor-scale scans, as shown by ScanComplete [11] and SGNN [10]. An alternative line of research for shape completion uses implicit scene representations [37, 39, 42, 40, 31, 7, 27]; however, while these approaches achieve stunning results for fitting and interpolation of objects/scenes, they still struggle to generalize across object categories with high geometric variety. While these existing works mainly focus on static scenes, we investigate how to leverage the synergies of shape completion in the dynamic 4D domain.

2.3. Non-Rigid 4D Datasets

Collecting large scale 4D datasets for deforming objects is a non-trivial task, in particular when the goal is to obtain a sufficiently large number of objects. Non-rigid

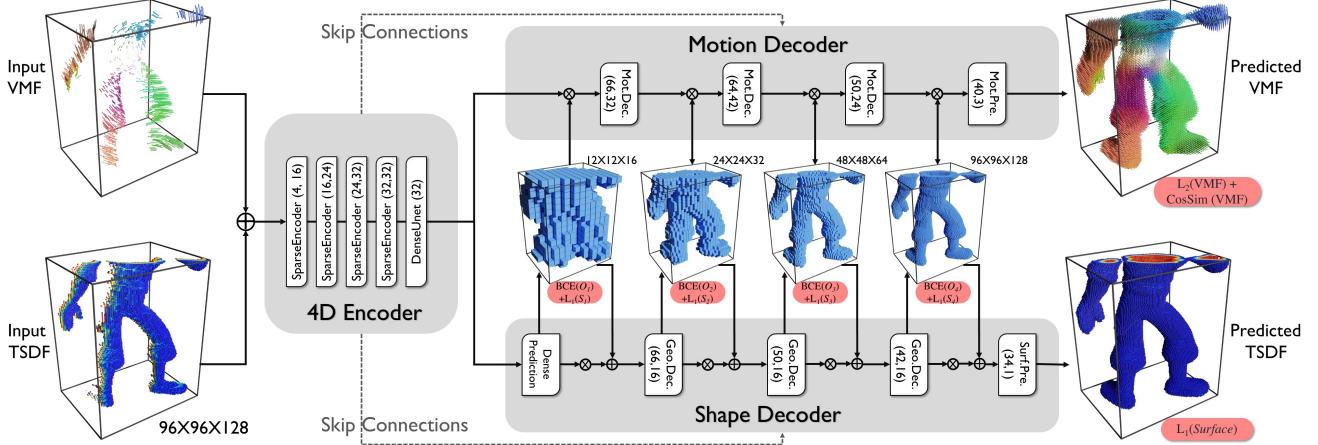


Figure 2: The network architecture of 4DComplete (Pink capsule: training loss; \oplus : concatenation; \otimes : filter by geometry, number in the brackets: (n_{in}, n_{out}) feature dimension). The input partial TSDF and VMF are concatenated together and fed into the 4D encoder. The two decoders predict the complete TSDF and VMF in parallel. There are 4 hierarchical levels. The shape decoder predicts the geometry in each hierarchical level and passes the predicted geometry to the corresponding layer in the motion decoder and the following layer in the shape decoder. Our method is trained on cropped volumes of spatial dimension $96 \times 96 \times 128$, which covers about 70 percent of an object. The fully-convolutional nature of our approach enables testing on whole objects of arbitrary sizes.

datasets [14, 3, 2, 52, 23, 56, 26, 45, 5, 58, 35, 60] have been widely used, but they are either relatively small, limited to specific scene types, or suffer from occlusion and sensor noise; hence, they are not directly suited for our 4D completion task. Notably, obtaining dense motion field ground truth from real-world 3D scans is quite challenging as it requires costly per-point correspondence annotations. This is one of the reasons why we have seen many synthetic datasets in the context of dense optical flow methods [36, 6, 34, 44]; Among them, Sintel [6] and Monka [36] are composed of rendered animations of deforming objects. However, these sequences are relatively short and do not provide complete 3D shapes and motion fields. In order to facilitate learning data-driven deformation priors, we introduce a much larger synthetic dataset with over 1,972 animation sequences, spanning a large diversity of objects ranging from humanoids to various animal species (cf. Sec. 4).

3. Method: 4DComplete

Given a single-view depth map observation of a 3D scene, and the scene flow that is computed between the current frame and its next frame, the goal of 4DComplete is to recover the hidden geometry and its motion field.

Input. We use a 3D volumetric grid to represent both shape and motion. The input shape is represented as a truncated signed distance field (TSDF), as a sparse set of voxel locations within truncation and their corresponding distance values. The TSDF is computed from a single depth map using volumetric fusion [9]; i.e., every voxel is projected into

the current depth map and their distance values are updated accordingly. To represent the input motion of the visible surface, we pre-compute the 3D motion vector (in \mathbb{R}^3) for each occupied voxel, resulting in a volumetric motion field (VMF) representation. We concatenate the TSDF and VMF and feed it as input to a neural network.

Scene Flow Field \Leftrightarrow Volumetric Motion Field. We use FlowNet3D [32] to predict the motion for the visible surface, which estimates the scene flow field (SFF) between two sets of point clouds. Because a 3D point does not necessarily lie on a regular 3D grid position, we convert between the SFF and the VMF as follows: given a point cloud $\{p_i | i = 1, \dots, N\}$, where $p_i \in \mathbb{R}^3$ are XYZ coordinates of individual point, the SFF is defined as $\{\mathcal{SFF}_i | i = 1, \dots, N\}$, where $\mathcal{SFF}_i \in \mathbb{R}^3$ are the 3D translational motion vectors of the points. Similarly, given a set 3D voxel positions $\{v_j | j = 1, \dots, M\}$, the VMF is defined as $\{\mathcal{VMF}_i | i = 1, \dots, M\}$, where $\mathcal{VMF}_i \in \mathbb{R}^3$ are the 3D translational motion vectors of the voxels. To convert from SFF to VMF, we use the inverse-distance weighted interpolation as defined in [43]:

$$\mathcal{VMF}_j = \sum_{p_i \in knn(v_j)} \frac{\mathcal{SFF}_i \cdot dist(p_i, v_j)^{-1}}{\sum_{p_i \in knn(v_j)} dist(p_i, v_j)^{-1}} \quad (1)$$

where $knn()$ is the function to find K-Nearest-Neighbors. We set the number of neighbors to $K = 3$, and $dist(,)$ computes the euclidean distance between two positions. To

convert from VMF to SFF, we do tri-linear interpolation:

$$SFF_j = \sum_{v_j \in knn(p_i)} \mathcal{VMF}_j \cdot w(p_i, v_j) \quad (2)$$

where $w(,)$ computes the linear-interpolation weights, and $K = 8$ represents the neighboring 8 corners of the cube that a point lies in.

Network Architecture. To allow for high-resolution outputs of the shape and motion field, we leverage sparse convolutions [21, 20, 8] for our neural network architecture, which makes our architecture computationally efficient in processing 3D volumetric data by operating only on the surface geometry. Hence, our method only processes the surface region and ignores the truncated region. Fig. 2 shows an overview of our network architecture. The network consists of a shared 4D encoder and two decoders to estimate shape and motion in parallel. The input sparse tensor is first fed into the *4D Encoder*, which encodes the data using a series of sparse convolutions where each set reduces the spatial dimensions by a factor of two. The two decoders are designed in a coarse-to-fine architecture with 4 hierarchical levels. We use skip connections between the 4D encoder and the 2 decoders to connect feature maps of the same spatial resolution. Since the shape decoder usually generates a larger set of sparse locations than the input, we use zero feature vector for the locations that do not exist in the input volume.

Message passing between two branches. At a hierarchical level k , the shape decoder predicts the voxels' occupancy O_k and TSDF value S_k . We filter voxels with $\text{sigmoid}(O_k(v)) > 0.5$ as the input geometry for the next hierarchical level. Within each hierarchical level, the shape decoder feeds the predicted geometry to the parallel motion decoder to inform where the motion should be estimated. In return, the motion feature is filtered by the sparse geometry and shared to the shape decoder.

Shape Loss. The shape decoder's final output is a sparse TSDF from which a mesh can be extracted by Marching Cubes. Following [10], we apply an l_1 loss on the log-transformed TSDF values. Using the log-transformation on the TSDF values helps to shift the losses attention more towards the surface points as larger values far away from the surface get smaller, thus encouraging more accurate prediction near the surface geometry. We additionally employ proxy losses at each hierarchy level for outputs O_k and S_k , using binary cross-entropy with target occupancies and l_1 with target TSDF values, respectively.

Motion Loss. The output of our sparse neural network is facilitated by the motion decoder, which estimates the completed volumetric motion field $\{\mathcal{VMF}_i | i = 1, \dots, M\}$. The

ground truth motion field at the predicted sparse locations is given by $\{\mathcal{VMF}_{i,gt} | i = 1, \dots, M\}$. We formulate the loss for the motion field on the final predicted sparse locations using the l_2 loss: $\sum_{i=1}^M \|\mathcal{VMF}_i - \mathcal{VMF}_{i,gt}\|_2^2$. In addition, we apply the cosine similarity loss: $\sum_{i=1}^M (1 - \frac{\mathcal{VMF}_i \cdot \mathcal{VMF}_{i,gt}}{\|\mathcal{VMF}_i\| \cdot \|\mathcal{VMF}_{i,gt}\|})$ on the normalized motion vectors to encourage the directions of the motion to be consistent with the ground truth.

Progressive Growing. We train our network in a progressively growing fashion following the ideas of [10]. There are four hierarchy levels, we progressively introduce higher resolution geometry decoder after every 2000 training iterations. To facilitate motion decoder learning, instead of using the predicted geometry of shape decoder, we fed ground-truth geometry to motion decoder during the beginning 10K iterations.

Training. We use our newly-constructed DeformingThings4D dataset (c.f. Sec. 4) to train our network. At training time, we consider cropped views of scans for efficiency (see Fig. 2); we use random crops of size $[96 \times 96 \times 128]$ voxels for the finest level. We crop the volumes at 1 meter intervals out of each train object and discard empty volume. The resolution drops by a factor of 2, resulting resolution of $[48 \times 48 \times 64]$, $[24 \times 24 \times 32]$, and $[12 \times 12 \times 16]$ for each hierarchical level. The fully-convolutional design of our approach enables testing on whole objects of arbitrary sizes at testing time. To learn viewpoint-invariant motion representation, we apply random rigid rotation transformations on the 3D motion vectors as data augmentation during training. The randomness is drawn from the Haar distribution [48], which yields uniform distribution on SO3. We train our network using the Adam optimizer with a learning rate of 0.001 and a batch size of 8.

4. DeformingThings4D Dataset

Training our network requires a sufficient amount of non-rigidly deforming target sequences with ground truth 4D correspondences at the voxel level (i.e., motion and shape). In order to provide such data, we construct a synthetic non-rigid dataset, DeformingThings4D, which consists of a large number of animated characters including humanoids and animals with skin mesh, texture, and skeleton. We obtained the characters from Adobe Mixamo¹ where humanoid motion data was collected using a motion capture system. Animals' skin and motion are designed by CG experts. Generally, these objects are animated by using “rigging” and “skinning” to blend the skeletal movement to the surface skin mesh. Fig. 3 shows examples of characters in the dataset and the statistics of our dataset.

¹<https://mixamo.com>



Figure 3: DeformingThings4D dataset. Left: examples of animated characters. Right: dataset statistics. In total, we collected 147 different characters spanning 31 categories, with a total of 1,972 animations and 122,365 frames.

4.1. Data Generation

Given a animated 3D mesh, we generate per-frame RGB-D maps, inter-frame scene flow, signed distance field and volumetric motion field; see Fig. 4. We perform data generation with Blender² scripts.

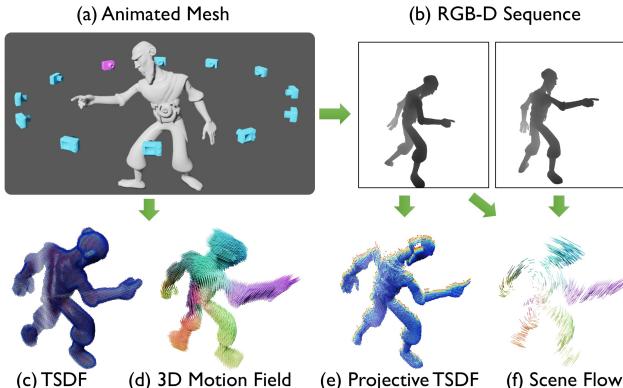
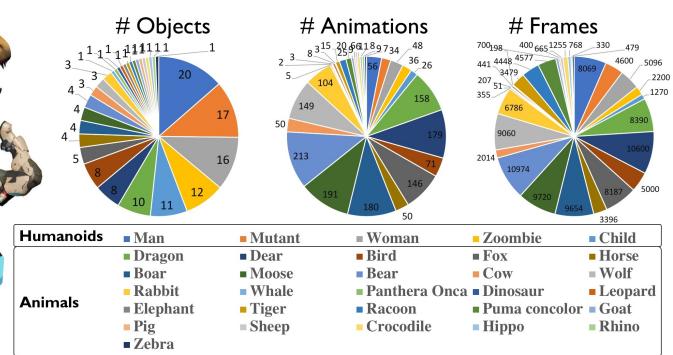


Figure 4: Data Generation Process: given an animated 3D mesh (a), virtual cameras are sampled on a sphere. One of the cameras is selected as the input view, for which depth maps (b) are rendered. Depth frames are used to compute the projective TSDF (e) and inter-frame scene flow (f). The ground-truth complete TSDF (c) is computed by integrating the depth images from all virtual cameras. The complete 3D motion field (d) is obtained by blending the mesh vertices’ motion to nearby occupied voxels.

RGB-D Map. To render depth maps, we uniformly sample 42 camera viewpoints on a sphere that is centered by the target character’s mesh. The mesh-to-camera distance ranges within $0.5 - 2.5m$. We render all the depth maps using the intrinsic parameters of the Azure Kinect camera. We store per-pixel depth in millimeters and render the color channel using Blender’s built-in Eevee engine with a principled BSDF shader.

²<https://www.blender.org/>



Inter-frame Scene Flow Field (SFF). The mesh animations run at 25 frames per second. We track the mesh vertices’ 3D displacements between a pair of temporally adjacent frames and project the 3D displacements to the camera’s pixel coordinates as scene flow. The flow vector for a pixel is computed by interpolating the 3 vertices’ motion on a triangle face where the pixel’s casted ray is first received. We generate scene flow ground truth for all observable pixels in the source frame even if the pixels are occluded in the target frame. To simulate the different magnitudes of deformation we sub-sample the sequences using the frame jumps: {1, 3, 7, 12}.

Signed Distance Field (SDF). In order to generate the ground truth SDF, we volumetrically fuse the depth maps from all virtual cameras into a dense regular grid [9], where each voxel stores a truncated signed distance value. We repeat this process independently for four hierarchy levels, with voxel sizes of $1.0cm^3$, $2.0cm^3$, $4.0cm^3$, and $8.0cm^3$. From the input depth map, we compute the projective SDF with voxel sizes of $1.0cm^3$ as network input while setting the truncation to $3 \times$ the voxel size. TSDF values are stored in voxel-distance metrics, which facilitates testing on volumes with arbitrarily sampled voxel size.

Volumetric Motion Field (VMF). We compute the motion ground-truth for all voxels near the mesh surface, i.e., within $3 \times$ voxel truncation. For each valid voxel, we first find its K -nearest-neighbor vertices on the mesh surface and then use Dual Quaternion Blending (DQB) to bind the motion of the KNN vertices to the voxel position. Empirically, we set $K = 3$. We follow the same procedure for the SDF volume and we repeat this process independently for all four resolutions, i.e., with voxel size of $1.0cm^3$, $2.0cm^3$, $4.0cm^3$, and $8.0cm^3$.

Method	Training dataset	DeformingThings4D			DeepDeform Dataset [5]			KITTI SFLow [19]		
		EPE↓	ACC(5)↑	ACC(10)↑	EPE↓	ACC(5)↑	ACC(10)↑	EPE↓	ACC(5)↑	ACC(10)↑
FlowNet3D	FlyingThings3D [36]	7.36	69.43%	80.04%	21.07	24.62 %	45.09%	16.88	38.49%	67.17%
	DeformingThings4D (Ours)	3.74	82.02%	91.63%	13.08	27.78 %	61.26%	17.01	36.89%	71.67%

Table 1: Scene flow estimation results on the DeformingThings4D, DeepDeform [5], and KITTI [19] datasets. Metrics are end-point-error (EPE) in centimeters, and Accuracy (<5cm or 5%, 10cm or 10%) for motion.

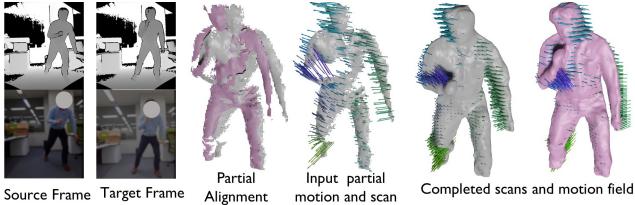


Figure 5: Testing on a pair of real-world RGB-D images.

5. Results

5.1. Evaluation Metrics

Motion Estimation Evaluation Metric. Following [32], we use 3D end-point-error (EPE) and motion accuracy (ACC) as our motion evaluation metrics. The 3D EPE measures the average euclidean distance between the estimated motion vector to the ground truth motion vector. The ACC score measures the portion of estimated motion vectors that are below a specified end-point-error among all the points. We report two ACC metrics with two different thresholds. Note that throughout the experiments we convert all VMF to SFF (using Eqn. 2) before doing motion evaluation.

Shape Completion Evaluation Metric. We use the following metrics to evaluate the reconstructed geometry, Volumetric IoU (IoU), Chamfer Distance (CD) in centimeters, Surface Normal Consistency (SNC), Point to Plane distance (P2P), and ℓ_1 error of SDF value.

5.2. Benchmarking Scene Flow

We compare our DeformingThings4D dataset with the FlyingThings3D [36], which is a large-scale dynamic motion dataset consists of flying rigid objects. We train FlowNet3D [32] with the two datasets and evaluate it on the test sets of DeformingThings4D, the DeepDeform [5], and the KITTI [19] scene flow benchmark. The results are shown in Tab. 1. DeepDeform [5] is a very challenging real-world benchmark for non-rigid motion. The FlowNet3D model trained on our dataset significantly reduces the scene flow error on the real-world DeepDeform benchmark (from 21.07 to 13.08). KITTI dataset captures street scenes with mainly rigid cars moving around which is more close to the flying things scenario. Our dataset still shows comparable results to FlyingThings3D on KITTI.

5.3. Motion Prediction for the Hidden Surface

This section evaluates the motion estimation of the hidden surface. We conduct the following experiment: the complete mesh shape, a subset of mesh vertices that is visible from a given camera viewpoint, and the ground truth scene flow for the visible vertices are given, and the goal is to estimate the motion of the hidden vertices of the mesh. We evaluate the following methods:

- **Rigid Fitting.** This method assumes that the shape undergoes rigid motion. It finds a single rigid transform in $SE(3)$ for the entire shape that best explains the surface motion.

- **As-Rigid-As-Possible (ARAP) Deformation.** ARAP [47] is widely used as a deformation prior in non-rigid reconstruction [38, 26, 59]. It assumes that locally, a point is transformed with a rigid transformation. Such rigid constraints are imposed upon nearby vertices that are connected by edges. ARAP deformation finds for each mesh vertex a local fan-rotation $R \in SO(3)$ and a global translation vector $t \in \mathbb{R}^3$ that best explains the scene flow motion with the local rigidity constraints.

- **Motion Complete (Ours).** Given the complete shape and the partial motion on the visible surface, this method predicts the VMF for the complete shape and converts it to SFF to get the motion on mesh vertices' positions. This method is trained only on humanoid motions and evaluates on an animal motion subset (we aim to confirm how the model generalizes across domains).

- **Motion Complete + Post Processing (PP) (Ours).** We found that the motion prediction of our Motion Complete model is sometimes noisy. We employ optimization-based post-processing to alleviate the noise: the predicted motion filed on the mesh surface is jointly optimized with ARAP prior that enforces that nearby vertices have similar motions.

Tab. 2 reports the motion estimation results for the occluded surface. The testing sequence includes one humanoid sequence and 6 animal sequences with different animations. Note that our method is trained only on the humanoids dataset. Among the baselines, rigid fitting yields significantly larger errors on most sequences, which indicate that the sequences undergo large non-rigid motion. Our Motion Complete overall achieves lower end-point-error than the ARAP on most sequences. Motion Complete + PP further improves the numbers. Fig. 6 shows the qualitative results of surface deformation for the “Deer” and “Dairy

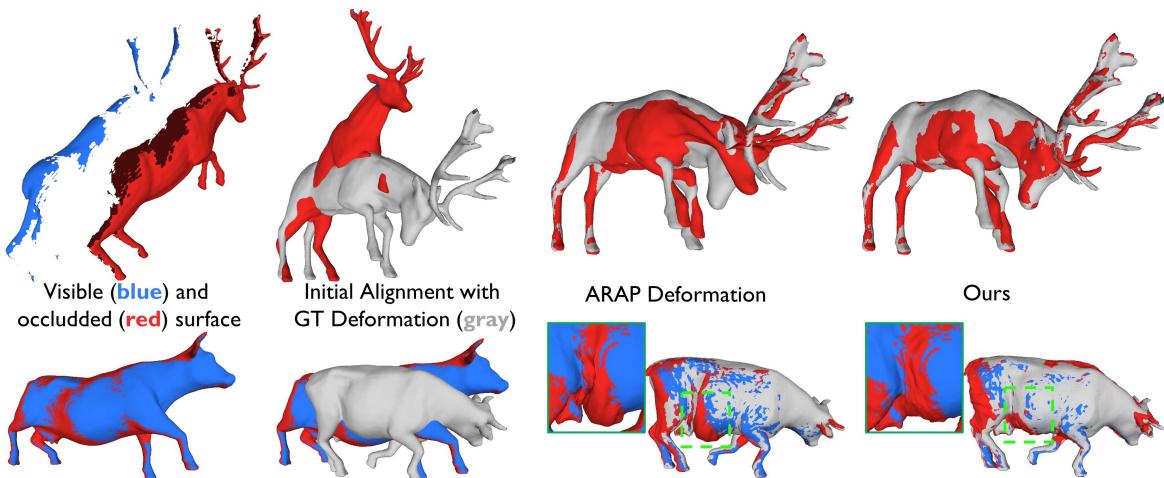


Figure 6: Surface deformation for the “Deer” and “Dairy Cow” sequence. The complete shape and the motion of the visible surface (**blue**) are given, and the goal is to estimate the deformation of the hidden surface (**red**). The gray mesh shows the ground truth deformation (**gray**), which is not available/used for registration. ARAP leads to severe distortion on the neck and head of the deer; The dairy cow’s stomach is undergoing a contraction movement. ARAP can not evenly distribute such deformation, leading to unnatural surface folding at the stomach. Our method yields natural deformations for both sequences. Note that our method is trained only on humanoid motions.

Methods	Humanoids (Samba Dance)	Dairy Cow (Attack)	Moose Bull (Walk)	Fox (Jump)	Dear Stag (Attack)	Panthera Onca (Run)	Avg.
Rigid Fitting	15.30	17.67	2.98	18.78	16.96	22.61	94.23
ARAP Deformation [47]	3.24	2.73	1.27	5.71	4.99	13.78	31.72
Motion Complete (Ours)	2.32	2.90	1.34	4.88	4.08	8.56	24.08
Motion Complete + PP (Ours)	1.81	2.24	1.21	4.20	2.05	7.18	18.69

Table 2: Quantitative evaluation for the motion estimation results of the unobserved surface. The Metric is 3D End-Point-Error (EPE) in centimeter. Note that our method is trained only on humanoid motions.

Cow” sequence. The deformed surfaces are achieved by using the estimated motion to warp the source model. Our method yields more plausible deformation than ARAP deformation for the occluded surface.

5.4. Ablation of Shape and Motion Estimation

This experiment examines how the two tasks, geometry completion and motion estimation, influence each other. To get the scene flow of the visible surface, we re-train FlowNet3D [32] using our scene flow dataset. FlowNet3D predicts the SFF given a pair of point clouds with a subsampled size of 2048. We convert the sparse SFF to VMF using Eqn. 1 as network input. The voxel position in the VMF is consistent with the input projective TSDF.

As defined in Fig. 2, we alternatively remove the shape completion head or the motion estimation head to examine the synergy of the two tasks. Tab. 3 reports the motion prediction results for the visible surface. Though only evaluating on the visible surface, the model trained with the added supervision of the geometry completion task show improvement over the model trained only on motion pre-

diction. This demonstrates that complete the missing shape is beneficial for non-rigid motion estimation. Tab. 4 reports the geometry completion results in our synthetic DeformingThings4D dataset. The whole model shows improvement over the model that is trained for geometry completion only. This result validates the idea that in a dynamic scene it is beneficial to understand the motion in order to achieve better geometric completion.

5.5. Shape Completion Results

We show qualitative shape completion results of our approach. IF-Nets [7] is a state-of-the-art method that performs single depth image reconstruction from point clouds. At the core of IF-Nets is an implicit function that maps a 3D coordinate to an occupancy score using a multi-layer perceptron. We train both methods on the humanoids dataset and evaluate the completion performance on unseen sequences. Fig. 7 shows shape completion from real-world RGB-D images. Tab. 5 shows quantitative evaluation on VolumeDeform [26] sequences. In particular, for large scenes, our approach effectively captures both global and

Method	DeformingThings4D			DeepDeform Dataset [5]		
	EPE↓	ACC(5)↑	ACC (10)↑	EPE↓	ACC(5)↑	ACC(10)↑
Ours (w/o shape completion)	3.82	79.02%	90.55%	13.75	26.89%	63.42%
Ours (w/ shape completion)	3.56	85.02%	91.59%	13.15	28.57%	63.66%

Table 3: Scene Flow estimation results on our DeformingThings4D dataset and DeepDeform [5] dataset. All scores are reported only for the visible surface points. Metrics are end-point-error (EPE) in centimeter, and accuracy ($<5\text{cm}$ or 5%, 10cm or 10%).

Method	CD↓	IoU ↑	SNC ↑	L1 ↓
Ours (w/o motion)	2.66	74.98%	0.779	0.531
Ours (w/ motion)	2.57	75.72%	0.812	0.503

Table 4: Surface prediction error on the test set of DeformingThings4D. The metrics are Volumetric IoU (IoU), Chamfer Distance (CD) in centimeters, Surface Normal Consistency (SNC), and ℓ_1 score of SDF.

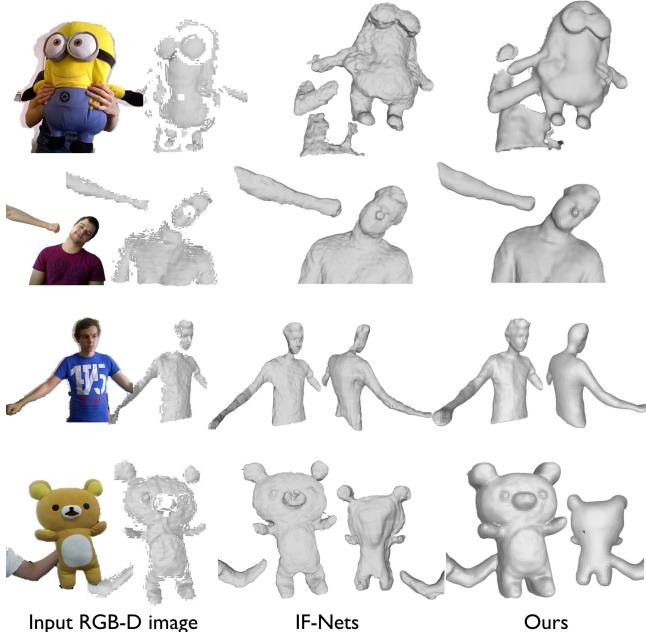


Figure 7: Shape completion results on real-world RGB-D images. The top 3 rows are images from VolumeDeform [26], and the last row is from [30].

local structures, as shown in Fig. 8.

Methods	IF-Nets	Ours (w/o motion)	Ours
P2P (cm)↓	2.231	1.983	1.876
SNC ↑	0.757	0.899	0.908
time (s) ↓	14.26	3.19	3.45
memory (MB) ↓	19,437	1,103	1,379

Table 5: Quantitative results on VolumeDeform examples. The surface ground-truth is provided by VolumeDeform. The metrics are the point-to-plane (P2P) distance, and surface normal consistency (SNC). We also report the average inference time and memory on a Tesla V100.

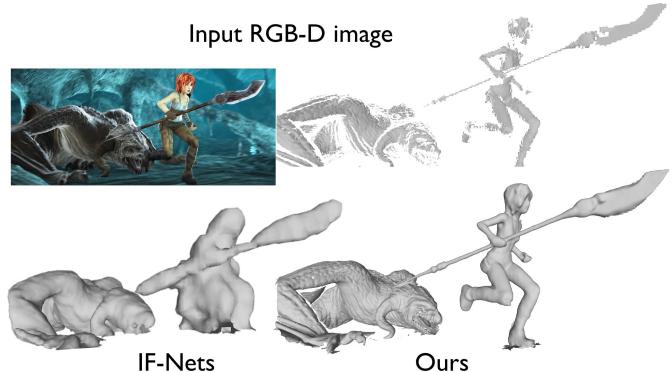


Figure 8: Shape completion on a large scene. The image is from MPI Sintel [6] dataset. The maximum depth is set to 10 meters in this sintel scene. With fixed volume size (256^3), IF-Net [7] loses the ability to model details for large scenes.

6. Conclusion & Limitations

In this work, we present the first method that jointly estimates the invisible shape and deformation from partial depth frame observation. We show that shape completion and motion estimation are mutually complementary tasks, with joint learning benefiting each. We believe that our method and new dataset open a new research avenue on generic non-rigid 4D reconstruction. Our approach maintains several limitations: 1) estimating the uncertainty of hidden motion is necessary but not handled by our approach. A probabilistic approach (e.g., Huang et al. [25]) would be promising for modeling motion uncertainty. 2) our method does not predict surface colors. The differentiable volumetric rendering approach of [13] is a potential solution to learn colored deformable objects. 3) DeformingThings4D largely contains articulated objects such as humans and animals. We are planning to expand the dataset with examples of loose clothing or plants that are deformed by external forces.

7. Acknowledgements

This work was conducted during Yang Li’s internship at Tokyo Research Center, Huawei. Matthias Nießner is supported by a TUM-IAS Rudolf Mößbauer Fellowship and the ERC Starting Grant Scan2CAD (804724). We thank Angela Dai for the voice-over of the video and also thank for setting up the dataset.

References

- [1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007. 2
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 3
- [3] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, 2014. 3
- [4] Aljaž Božič, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. *arXiv preprint arXiv:2006.13240*, 2020. 2
- [5] Aljaž Božič, Michael Zollhöfer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7002–7012, 2020. 2, 3, 6, 8
- [6] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 611–625, 2012. 3, 8
- [7] Julian Chibane, Thieno Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6970–6981, 2020. 2, 7, 8
- [8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019. 4
- [9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 3, 5
- [10] Angela Dai, Christian Diller, and Matthias Nießner. Sgnn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 849–858, 2020. 2, 4
- [11] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2018. 2
- [12] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5868–5877, 2017. 2
- [13] Angela Dai, Yawar Siddiqui, Justus Thies, Julien Valentin, and Matthias Nießner. Spsg: Self-supervised photometric scene generation from rgb-d scans. *arXiv preprint arXiv:2006.14660*, 2020. 8
- [14] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *ACM SIGGRAPH 2008 papers*, pages 1–10, 2008. 3
- [15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 2
- [16] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6):1–16, 2017. 2
- [17] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):114, 2016. 2
- [18] Wei Gao and Russ Tedrake. Surfelwarp: Efficient non-volumetric single view dynamic reconstruction. *arXiv preprint arXiv:1904.13073*, 2019. 2
- [19] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 6
- [20] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232, 2018. 4
- [21] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 4
- [22] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3254–3263, 2019. 2
- [23] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3083–3091, 2015. 3
- [24] Oshri Halimi, Ido Imanuel, Or Litany, Giovanni Trappolini, Emanuele Rodolà, Leonidas Guibas, and Ron Kimmel. The whole is greater than the sum of its nonrigid parts. *arXiv preprint arXiv:2001.09650*, 2020. 2
- [25] Jiahui Huang, Shi-Sheng Huang, Haoxuan Song, and Shi-Min Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. *arXiv preprint arXiv:2012.05551*, 2020. 8

- [26] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 362–379, 2016. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)
- [27] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6001–6010, 2020. [2](#)
- [28] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. [2](#)
- [29] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008. [2](#)
- [30] Yang Li, Aljaz Bozic, Tianwei Zhang, Yanli Ji, Tatsuya Harada, and Matthias Nießner. Learning to optimize non-rigid tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4910–4918, 2020. [2](#), [8](#)
- [31] Zhe Li, Tao Yu, Chuanyu Pan, Zerong Zheng, and Yebin Liu. Robust 3d self-portraits in seconds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1344–1353, 2020. [2](#)
- [32] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–537, 2019. [2](#), [3](#), [6](#), [7](#)
- [33] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteoronet: Deep learning on dynamic 3d point cloud sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9246–9255, 2019. [2](#)
- [34] Zhaoyang Lv, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James M Rehg, and Jan Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 468–484, 2018. [2](#), [3](#)
- [35] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. [3](#)
- [36] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. [3](#), [6](#)
- [37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. [2](#)
- [38] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352, 2015. [1](#), [2](#), [6](#)
- [39] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5379–5389, 2019. [2](#)
- [40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. [2](#)
- [41] Mark Pauly, Niloy J Mitra, Joachim Giesen, Markus H Gross, and Leonidas J Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, pages 23–32, 2005. [2](#)
- [42] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *arXiv preprint arXiv:2003.04618*, 2020. [2](#)
- [43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. [3](#)
- [44] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2213–2222, 2017. [3](#)
- [45] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1386–1395, 2017. [2](#), [3](#)
- [46] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1754, 2017. [2](#)
- [47] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, volume 4, pages 109–116, 2007. [1](#), [2](#), [6](#), [7](#)
- [48] Gilbert W Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980. [4](#)
- [49] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 26(3):80, 2007. [1](#), [2](#)
- [50] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. [2](#)
- [51] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. *arXiv preprint arXiv:2003.12039*, 2020. [2](#)

- [52] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers*, pages 1–9, 2008. 3
- [53] Shenlong Wang, Sean Ryan Fanello, Christoph Rhemann, Shahram Izadi, and Pushmeet Kohli. The global patch colider. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 127–135, 2016. 2
- [54] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen. Flownet3d++: Geometric losses for deep scene flow estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 91–98, 2020. 2
- [55] Wenzuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*, 2019. 2
- [56] Genzhi Ye, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt. Performance capture of interacting characters with handheld kinects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 828–841, 2012. 3
- [57] Bo Zheng, Yibiao Zhao, Joey C Yu, Katsushi Ikeuchi, and Song-Chun Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3127–3134, 2013. 2
- [58] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3
- [59] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (ToG)*, 33(4):156, 2014. 2, 6
- [60] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6365–6373, 2017. 3