DeepDeform: Learning Non-rigid RGB-D Reconstruction with Semi-supervised Data

Aljaž Božič 1
Michael Zollhöfer 2
Christian Theobalt 3
Matthias Nießner 1

1Technical University of Munich 2Stanford University 3Max Planck Institute for Informatics

Figure 1: We propose a semi-supervised strategy combining self-supervision with sparse annotations to build a large-scale RGB-D dataset of non-rigidly deforming scenes (400 scenes, 390,000 frames, 5,533 densely aligned frame pairs). With this data, we propose a new method for non-rigid matching, which we integrate into a non-rigid reconstruction approach.

Abstract

Applying data-driven approaches to non-rigid 3D reconstruction has been difficult, which we believe can be attributed to the lack of a large-scale training corpus. Unfortunately, this method fails for important cases such as highly non-rigid deformations. We first address this problem of lack of data by introducing a novel semi-supervised strategy to obtain dense inter-frame correspondences from a sparse set of annotations. This way, we obtain a large dataset of 400 scenes, over 390,000 RGB-D frames, and 5,533 densely aligned frame pairs; in addition, we provide a test set along with several metrics for evaluation.

Based on this corpus, we introduce a data-driven non-rigid feature matching approach, which we integrate into an optimization-based reconstruction pipeline. Here, we propose a new neural network that operates on RGB-D frames, while maintaining robustness under large non-rigid deformations and producing accurate predictions. Our approach significantly outperforms existing non-rigid reconstruction methods that do not use learned data terms, as well as learning-based approaches that only use self-supervision.

1. Introduction

Non-rigid 3D reconstruction, i.e., the dense, space-time coherent capture of non-rigidly deforming surfaces in full temporal correspondence, is key towards obtaining 3D abstractions of the moving real world. The wide availability of commodity RGB-D sensors, such as the Microsoft Kinect or Intel Realsense, has led to tremendous progress on static scene reconstruction methods. However, robust and high-quality reconstruction of non-rigidly moving scenes with one depth camera is still challenging. Applications for real-time non-rigid reconstruction range from augmented (AR) and virtual reality (VR) up to building realistic 3D holograms for fully immersive teleconferencing systems. The seminal DynamicFusion [28] approach was the first to show dynamic non-rigid reconstruction in real-time. Extensions primarily differ in the used energy formulation. Some methods use hand-crafted data terms based on dense geometry [28, 36, 37], dense color and geometry [14, 47], and sparse feature constraints [17]. Other approaches leverage multi-camera RGB-D setups [9, 8] for higher robustness. However, there are very few reconstruction methods that use learning-based data terms for general real-world scenes rather than specific scenarios [48], and that are trained to be robust under real-world appearance variation and diffi-
cult motions. One reason for this is the lack of a large-scale training corpus. One recent approach [33] proposes self-supervision for ground truth generation, i.e., they employ DynamicFusion [28] for reconstruction and train a non-rigid correspondence descriptor on the computed inter-frame correspondences. However, we show that existing non-rigid reconstruction methods are not robust enough to handle realistic non-rigid sequences; therefore this tracking-based approach does not scale to training data generation for real world scenes. Unfortunately, this means that self-supervision exactly fails for many challenging scenarios such as highly non-rigid deformations and fast scene motion. By design, this approach trained with self-supervision cannot be better than the employed tracker. We propose to employ semi-supervised training data by combining self-supervision with sparse user annotations to obtain dense inter-frame correspondences. The annotated sparse point correspondences guide non-rigid reconstruction; this allows us to handle even challenging motions. The result is a large dataset of 400 scenes, over 390,000 RGB-D frames, and 5,533 densely aligned frame pairs. Based on this novel training corpus, we develop a new non-rigid correspondence matching approach (see Sec. 3) that finds accurate matches between RGB-D frames and is robust to difficult real world deformations. We further propose a re-weighting scheme that gives more weight to corner cases and challenging deformations during training. Given a keypoint in a source frame, our approach predicts a probability heatmap of the corresponding location in the target frame. Finally, we integrate our learned data term into a non-rigid reconstruction pipeline that combines learned heatmap matches with a dense RGB-D reconstruction objective. In addition, we introduce a new benchmark and metric for evaluating RGB-D based non-rigid 3D correspondence matching and reconstruction. We extensively compare our new data-driven approach to existing hand-crafted features. We also integrate the learned features into a non-rigid reconstruction framework, leading to significant improvement over state of the art. In sum, our contributions are:

- A semi-supervised labeling approach for dense non-rigid correspondence learning, resulting in a dataset featuring 400 annotated dynamic RGB-D sequences and 5,533 densely aligned frame pairs.

- A novel data-driven non-rigid correspondence matching strategy that leads to more robust correspondence estimation compared to the state-of-the-art hand-crafted and learned descriptors, especially in the case of extreme deformations.

- A non-rigid reconstruction approach for general scenes that combines learned and geometric data-terms and handles significantly faster and more complex motions than the state-of-the-art.

Figure 2: Our large-scale dataset contains a large variety of dynamic sequences with segmentation masks and point correspondences between different RGB-D frames.

2. Related Work

Our approach is related to several research areas, such as volumetric 3D scene reconstruction, non-rigid object tracking, and learned correspondence matching. We focus our discussion on the most related RGB-D based techniques. For a detailed discussion, we refer to the recent survey [59].

Volumetric Scene Reconstruction Reconstructing static environments with a single RGB-D sensor has had a long history in vision and graphics, including KinectFusion [27, 18], which employs a uniform voxel grid to represent the scene as a truncated signed distance function (TSDF) [4], as well as many extensions to large-scale scenes [49, 2, 40, 29]. These techniques track the 6-DoF camera motion by solving a geometric model-to-frame alignment problem using a fast data-parallel variant of the point-to-plane Iterative Closest Point (ICP) algorithm [30]. Globally consistent reconstruction based on Bundle Adjustment [57, 3] was for a long time only possible offline; data-parallel solvers now enable real-time frame rates [6]. An alternative to TSDFs are point-based scene representations [20, 23, 22]. Recent techniques also employ non-rigid registration to robustly handle loop closures [50, 58].

Non-Rigid Reconstruction The reconstruction of general non-rigidly deforming objects based on real-time scan data has a long tradition [44]. One class of methods uses pre-defined templates, e.g., human templates, to capture pose and time-varying shape of clothed humans from RGB-D [54] or stereo camera data [51]. First template-less approaches had slow offline runtimes and only worked for slow and simple motions. The approach of [10] solves a global optimization problem to reconstruct the canonical
shape of a non-rigidly deforming object given an RGB-D video sequence as input, but does not recover the time-dependent non-rigid motion across the entire sequence. The first approach to demonstrate truly dynamic reconstruction of non-rigid deformation and rest shape in real-time was DynamicFusion [28]. Since this seminal work, many extensions have been proposed. VolumeDeform [17] improves tracking quality based on sparse feature alignment. In addition, they parameterize the deformation field based on a dense volumetric grid instead of a sparse deformation graph. The KillingFusion [36] and SobolevFusion [37] approaches allow for topology changes, but do not recover dense space-time correspondence along the complete input sequence. Other approaches jointly optimize for geometry, albedo, and motion [14] to obtain higher robustness and better quality. The approach of Wang et al. [47] employs global optimization to minimize surface tracking errors. In contrast to these methods using a single RGB-D camera, other techniques use multiple color [7, 43] or depth cameras [52, 45, 9, 8], which enables high-quality reconstruction at the cost of more complex hardware. We propose a new learning-based correspondence matching and reconstruction approach that outperforms existing techniques.

Learning Rigid Correspondence Matching Historically, correspondence matching for the task of rigid registration has been based on hand-crafted geometry descriptors [19, 11, 42, 32, 31]. If color information is available in addition to depth, SIFT [24] or SURF [1] can be used to establish a sparse set of feature matches between RGB-D frames. More recently, 2D descriptors for feature matching in static scenes have been learned directly from large-scale training corpora [34, 35, 15, 53, 60]. The Matchnet [15] approach employs end-to-end training of a CNN to extract and match patch-based features in 2D image data. Descriptors for the rigid registration of static scenes can be learned and matched directly in 3D space with the 3DMatch [56] architecture. Visual descriptors for dense correspondence estimation can be learned in a self-supervised manner by employing a dense reconstruction approach to automatically label correspondences in RGB-D recordings [33]. Descriptor learning and matching for static scenes has been well-studied, but is lacking in the challenging non-rigid scenario. While class-specific dense matching of non-rigid scenes has been learned for specific object classes [48, 12], none of these techniques can handle arbitrary deforming non-rigid objects. We believe one reason for this is the lack of a large-scale training corpus. In this work, we propose such a corpus and demonstrate how non-rigid matching between depth images can be learned end-to-end.

RGB-D Datasets While we have seen a plethora of RGB-D datasets for static scenes, such as NYU [26], SUN RGB-D [38], and ScanNet [5], the largest of which have thousands of scans, non-rigid RGB-D datasets remain in their infancy. While these datasets can be used to pretrain networks for the task of non-rigid correspondence matching, they do not capture the invariants that are useful for the much harder non-rigid setting, and thus lead to sub-par accuracy. Current non-rigid reconstruction datasets are far too small and often limited to specific scene types [7, 52], which is not sufficient to provide the required training data for supervised learning. The datasets that provide real-world depth recordings [13, 17] do not come with ground truth reconstructions, which makes objectively benchmarking different approaches challenging. Other datasets that are commonly used for evaluation do not provide real-world depth data, e.g., [7, 43]. In this work, we introduce the first large-scale dataset for non-rigid matching based on semi-supervised labeling and provide a benchmark enabling objective comparison of different approaches.

3. Data-driven Non-Rigid Matching

Our goal is to find matches between a source and a target RGB-D frame. To this end, we propose a network architecture for RGB-D matching based on a Siamese network [21] with two towers. Input to the network are two local patches of size 224 × 224 pixels each (with 3 color channels and 3-dimensional points in camera coordinate space). We assume that the source patch is centered at a feature point.

Heatmap The goal is to predict a probability heatmap \( \mathcal{H} \) in the target frame that gives the likelihood of the location of the source point. First, we compute a sigmoid-heatmap:

\[
\mathcal{H}_{\text{sg}} = \sigma_{\text{sg}} \left( \text{H} \left( D_{\text{out}} \right) \right)
\]

It is computed based on a sigmoid activation \( \sigma_{\text{sg}} \) to map responses to \([0, 1]\). Here, \( D_{\text{out}} \) is the output feature map of the last layer of the decoder and \( \text{H} \) is a convolutional layer converting feature space into heatmap values. This is equivalent to independent binary classification problems per pixel. Second, we also compute a softmax-heatmap:

\[
\mathcal{H}_{\text{sm}} = \sigma_{\text{sm}} \left( \text{H} \left( D_{\text{out}} \right) \right)
\]

Here we use a softmax activation \( \sigma_{\text{sm}} \) to make the complete heatmap a probability distribution, i.e., it sums to one. As ground truth for the heatmap prediction we could take an image with zero values everywhere except at the ground truth pixel position that we set to one. To prevent the trained network from predicting only zero values, we apply a Gaussian kernel \( G_{e_{\text{sm}}} \) around the ground truth pixel. It sets the ground truth pixel’s value to one and decays the neighboring pixel values to zero with standard deviation of 7 pixels, resulting in the ground truth heatmap \( \mathcal{H}_{\text{gt}} \). We also add larger weight to pixels close to the ground truth pixel, defining the
Figure 3: We devise an end-to-end architecture for RGB-D matching based on a Siamese network to find matches between a source and a target frame. Our network is based on two towers that share the encoder and have a decoder that predicts two probability heatmaps in the target frame that encode the likelihood of the location of the source point. Our network also predicts a depth value for the matched point and a visibility score that measures if the source point is visible in target frame.

Pixel weight as \( w_H(x) = 1 + 10 \cdot G_{x,y}(x) \). The heatmap loss is then computed as:

\[
\mathcal{L}_H = \Phi_{bce}(w_H(\mathcal{H}_{sg} - \mathcal{H}_{gt})) + \lambda_{nll} \Phi_{nll}(w_H(\mathcal{H}_{sm} - \mathcal{H}_{gt})) .
\]

Here, \( \Phi_{bce}(\bullet) \) denotes the binary cross entropy loss and \( \Phi_{nll}(\bullet) \) the negative log-likelihood loss, and we empirically determined a weight \( \lambda_{nll} = 10 \). From these two probability heatmaps, a single one is computed as \( \mathcal{H} = \mathcal{H}_{sg} \otimes \mathcal{H}_{sm} \), where \( \otimes \) is the Hadamard product.

**Depth** In addition to heatmap prediction, our network also predicts the matched point’s depth value in the target camera’s coordinate system. Inspired by [25], we predict the depth densely, predicting the same depth value for every pixel in the output image:

\[
D = \exp(D_{out}) .
\]

Here, \( D \) is a convolutional layer converting feature space into depth values, and the exponential is applied to guarantee positive depth predictions. Ground truth for depth prediction \( D_{gt} \) is the depth of the ground truth match, repeated for the whole image. Since we want to encourage depth prediction to focus on the matched pixel, we again use pixel weighting, this time in the form of \( w_D(x) = G_{x,y}(x) \), setting the center pixel’s weight to 1 and decaying the weights to 0. Using the weighted version of mean squared error \( \Phi_{mse}(\bullet) \) we employ the following loss for depth prediction:

\[
\mathcal{L}_D = \lambda_d \sum_i \Phi_{mse}(w_D(D - D_{gt})) .
\]

**Visibility** Furthermore, we also predict a visibility score \( \in [0, 1] \) that measures whether the source point is visible (high value) or occluded (low value) in the target frame:

\[
\mathcal{V} = \sigma_{sg}(V(B_{out})).
\]

Here, \( B_{out} \) is the output feature map of the bottleneck layer, \( V \) is a convolutional layer, and \( \sigma_{sg} \) a sigmoid activation. The visibility loss takes the following form:

\[
\mathcal{L}_V = \sum_i \Phi_{bce}(\mathcal{V} - \mathcal{V}_{gt}) .
\]

In the end, we train the network using a weighted combination of all presented loss functions:

\[
\mathcal{L} = \mathcal{L}_H + \lambda_D \mathcal{L}_D + \lambda_V \mathcal{L}_V .
\]

In all experiments we use the constant and empirically determined weights \( \lambda_D = 100 \) and \( \lambda_V = 1 \). An overview of the network architecture is given in Fig. 3. More network and training details can be found in the supplemental.

4. Non-Rigid Reconstruction Pipeline

We integrate the learned non-rigid matching algorithm into a non-rigid RGB-D reconstruction framework that efficiently tracks dense, space-time coherent, non-rigid deformations on the GPU and also provides an efficient volumetric fusion backend. A canonical model of the scene is reconstructed from data, in parallel to tracking the non-rigid deformations, and stored based on a truncated signed distance field (TSDF) represented by a uniform voxel grid. New observations are fused into the grid based on an exponentially moving average. The non-rigid scene motion is tracked based on the following tracking energy:

\[
E_{total}(T) = E_{data}(T) + \lambda_{learned}E_{learned}(T) + \lambda_{reg}E_{reg}(T) .
\]
The weights $\lambda_{\text{learned}} = 1$ and $\lambda_{\text{reg}} = 1$ are empirically determined and balance the different terms.

### 4.1. Deformation Model

To parameterize scene motion, similar to [41], we employ a coarse deformation graph $\mathcal{G}$ with $K$ deformation nodes $g_i \in \mathbb{R}^3$. The graph models the deformation of space based on the convex combination of local per-node transformations that are parameterized by rotation parameters $\theta_i \in \mathbb{R}^3$ in Lie algebra space and a translation vector $t_i \in \mathbb{R}^3$. In total, this leads to $2K$ free variables to describe scene motion that we jointly refer to as $\mathcal{T}$. This enables us to decouple the number of free variables from the complexity of the reconstructed scene. The deformation nodes are connected based on proximity, for details we refer to the original embedded deformation paper [41].

### 4.2. Optimization Terms

For data term $E_{\text{data}}(T)$, similar to [28, 17], we employ dense point-to-point and point-to-plane alignment constraints between the input depth map and the current reconstruction. For regularizer $E_{\text{reg}}$, we employ the as-rigid-as-possible (ARAP) constraint [39] to enforce locally rigid motion. In addition, we integrate a sparse feature alignment term based on our learned correspondences (see Sec. 3). For each node $g_i$ of the current deformation graph, we predict a probability heatmap $H_i$ that gives the likelihood of its 2D uv-position in the current input depth map, using the initial depth map as the reference frame. Furthermore, we also back-project the pixel with the maximum heatmap response into a 3D point $p_i \in \mathbb{R}^3$, using its depth. We aim to align the graph nodes with the maximum in the corresponding heatmap using the following alignment constraint:

$$E_{\text{learned}}(T) = \sum_{g_i \in \mathcal{V}} \left( 1 - H_i(\pi(g_i + t_i))) \right)^2 + \lambda_{\text{point}} \sum_{g_i \in \mathcal{V}} (g_i + t_i - p_i)^2.$$  

Here, $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the projection from 3D camera space to 2D screen space. The heatmap $H_i$ is normalized to a maximum of 1. We empirically set $\lambda_{\text{point}} = 10$. In order to handle outliers, especially in the case of occluded correspondences, we make use of the predicted visibility score and the predicted depth value of the match. We filter out all heatmap correspondences with visibility score $< 0.5$. We compare the predicted depth with the queried depth from the target frame’s depth map at the pixel with the maximum heatmap response and invalidate any correspondences with a depth difference $> 0.15$ meters.

### 4.3. Energy Optimization

We efficiently tackle the underlying optimization problem using a data-parallel Gauss-Newton solver to find the deformation graph $\mathcal{G}^*$ that best explains the data:

$$\mathcal{G}^* = \arg\min E_{\text{total}}(\mathcal{G}).$$

In the Gauss-Newton solver, we solve the underlying sequence of linear problems using data-parallel preconditioned conjugate gradient (PCG). For implementation details, we refer to the supplemental document.

### 5. Semi-supervised Data Acquisition

In the following, we provide the details of our semi-supervised non-rigid data collection process that is used for training the non-rigid matching and for the evaluation of non-rigid reconstruction algorithms. The high-level overview of the data acquisition pipeline is shown in Fig. 1.

#### 5.1. Data Acquisition

In order to obtain RGB-D scans of non-rigidly moving objects, we use a Structure Sensor mounted on an iPad. The depth stream is recorded at a resolution of $640 \times 480$ and 30 frames per second; the RGB stream is captured with the iPad camera at a resolution of $1296 \times 968$ pixels that is calibrated with respect to the range sensor. Regarding the scanning instructions, we follow the ScanNet [5] pipeline. However, in our case, we focus on scenes with one up to several non-rigidly moving objects in addition to a static background. In total, we recorded 400 scenes with over 390,000 RGB-D frames.

#### 5.2. Data Annotation

We crowd sourced sparse ground truth correspondence annotations and segmentation masks for our novel data set. To this end, we employed a web-based annotation tool. The annotation was divided into two tasks. Firstly, we select up to 10 frames per sequence. All dynamic objects that are found in these frames are given unique instance ids (the same instance in the whole sequence) and their masks are annotated in each frame. To accelerate mask segmentation, we use a hierarchy of superpixels as candidate brush sizes. Secondly, among the annotated frames up to 10 frame pairs are selected, and the sparse correspondences between all dynamic objects are annotated. Expert annotators were instructed to annotate correspondences uniformly over the complete object, labeling about 20 point matches per frame pair. Furthermore, in parts of the source image that are occluded in the target image occlusion points were uniformly selected to collect data samples for visibility detection. The dynamic object segmentation task takes on average about 1 min per frame, while the correspondence labeling task takes on average about 2 min per frame.

#### 5.3. Dense Data Alignment

Using the annotated object masks and the sparse matches, dense non-rigid alignment of the frame pairs is
performed. We follow a similar approach as for non-rigid reconstruction (see Sec. 4), based on the sparse deformation graph of [41]. The deformation graph is defined on the source frame’s depth map, covering only the dynamic object by using the source object mask. The final energy function that is optimized to align the source RGB-D frame to the target RGB-D frame is:

\[ E_{\text{total}}(T) = E_{\text{data}}(T) + \lambda_{\text{photo}} E_{\text{photo}}(T) + \lambda_{\text{silh}} E_{\text{silh}}(T) + \lambda_{\text{sparse}} E_{\text{sparse}}(T) + \lambda_{\text{reg}} E_{\text{reg}}(T). \]

Here, \( E_{\text{photo}}(T) \) encourages the color gradient values from the source frame to match the target frame, \( E_{\text{silh}} \) penalizes deformation of the object outside of the target frame’s object mask, and \( E_{\text{sparse}} \) enforces annotated sparse matches to be satisfied. The weights \( \lambda_{\text{photo}} = 0.001, \lambda_{\text{silh}} = 0.0001, \lambda_{\text{sparse}} = 100.0 \) and \( \lambda_{\text{reg}} = 10.0 \) have been empirically determined. Details about the different optimization terms and a qualitative comparison of their effects can be found in the supplemental document. In order to cope with simple apparent topology changes that are very common while capturing natural non-rigid motion, such as the hand touching the body in one frame and moving away in another frame, we execute non-rigid alignment in both directions and compute the final non-rigid alignment using forward-backward motion interpolation, similar to [55]. At the end, a quick manual review step is performed, in which, if necessary, any incorrectly aligned mesh parts are removed. The review step takes about 30 seconds per frame. Examples of dense alignment results and the employed review interface can be found in the accompanying video.

### 6. Experiments

We provide a train-val-test split with 340 sequences in the training set, 30 in the test set, and 30 in the validation set. We made sure that there is no overlap between captured environments between training and validation/test scenes.

#### 6.1. Non-Rigid Matching Evaluation

For a given set of pixels (and corresponding 3D points) in the source image, the task is to find the corresponding pixel (and 3D point) in the target image. We evaluate the average 2D pixel and 3D point error (in meters), and compute the matching accuracy (ratio of matches closer than 20 pixels or 0.05 meters from the ground truth correspondences). We compare our non-rigid matching approach to several handcrafted feature matching strategies, that are based on depth or color based descriptors, and to the learned 3Dmatch [56] descriptor, see Tab. 1. Specifically, we compare to the handcrafted geometry descriptors, such as Unique Signatures of Histograms (SHOT) [42] and the Fast Point Feature Histograms (FPFH) [31]. We also compare to color-based descriptors, e.g., SIFT [24] and SURF [1], that can be used to establish a sparse set of matches been RGB-D frames. Finally, we train a learned descriptor from [56], patch-based random forest matcher from [46] and optical flow prediction network from [16] on our training sequences. Our method consistently outperforms all the baselines.

<table>
<thead>
<tr>
<th>Method</th>
<th>2D-err</th>
<th>3D-err</th>
<th>2D-acc</th>
<th>3D-acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT [24]</td>
<td>1.8840</td>
<td>0.5302</td>
<td>16.20</td>
<td>14.08</td>
</tr>
<tr>
<td>SURF [1]</td>
<td>1.2572</td>
<td>0.4768</td>
<td>22.13</td>
<td>19.82</td>
</tr>
<tr>
<td>SHOT [42]</td>
<td>1.1734</td>
<td>0.4328</td>
<td>13.43</td>
<td>11.51</td>
</tr>
<tr>
<td>FPFH [31]</td>
<td>1.1666</td>
<td>0.3931</td>
<td>10.85</td>
<td>9.43</td>
</tr>
<tr>
<td>3Dmatch [56]</td>
<td>1.3302</td>
<td>0.2734</td>
<td>10.72</td>
<td>7.24</td>
</tr>
<tr>
<td>GPC [46]</td>
<td>1.6584</td>
<td>0.2314</td>
<td>31.93</td>
<td>28.16</td>
</tr>
<tr>
<td>FlowNet-2.0 [16]</td>
<td>2.3724</td>
<td>0.1188</td>
<td>68.68</td>
<td>63.66</td>
</tr>
<tr>
<td>Ours-12.5%</td>
<td>7.828</td>
<td>0.2689</td>
<td>27.17</td>
<td>23.28</td>
</tr>
<tr>
<td>Ours-25.0%</td>
<td>5.288</td>
<td>0.1971</td>
<td>40.32</td>
<td>35.81</td>
</tr>
<tr>
<td>Ours-50.0%</td>
<td>4.543</td>
<td>0.1561</td>
<td>50.70</td>
<td>46.57</td>
</tr>
<tr>
<td>Ours-Rigid</td>
<td>5.743</td>
<td>0.2707</td>
<td>40.93</td>
<td>36.64</td>
</tr>
<tr>
<td>Ours-SelfSupervised</td>
<td>3.34</td>
<td>0.1219</td>
<td>60.87</td>
<td>55.70</td>
</tr>
<tr>
<td>Ours-Sparse</td>
<td>3.41</td>
<td>0.1068</td>
<td>58.53</td>
<td>52.24</td>
</tr>
<tr>
<td>Ours-NoWeighting</td>
<td>2.3724</td>
<td>0.0837</td>
<td>73.13</td>
<td>68.46</td>
</tr>
<tr>
<td>Ours</td>
<td>19.756</td>
<td>0.073</td>
<td>77.10</td>
<td>72.48</td>
</tr>
</tbody>
</table>

Table 1: We outperform all baseline matching methods by a considerable margin. 2D/3D errors are average pixel/point errors, and 2D/3D accuracy is the percentage of pixels/points with distance of at most 20 pixels/0.05 meters.

6.2. Non-Rigid Reconstruction Results

We integrated our learned matching strategy into a non-rigid reconstruction pipeline. Our learned data term significantly improves reconstruction quality, both qualitatively and quantitatively. To be able to perform a quantitative comparison on our test sequences, we used our re-implementation of [28], and the code or results provided from the authors of [17, 36, 37, 14]. We also replaced our data-driven correspondence matching module with the descriptor learning network from [56], trained on our data, and used it in combination with 3D Harris keypoints. The quantitative evaluation is shown in Tab. 2. The evaluation metrics measure deformation error (a 3D distance between the annotated and computed correspondence positions) and geometry error (comparing depth values inside the object mask to the reconstructed geometry). Deformation error is the more important metric, since it also measures tangential drift within the surface. To be able to know which dynamic object to reconstruct if multiple are present, we always provide the initial ground truth segmentation mask of the selected object. All approaches in Tab. 2 were evaluated on all 30 test sequences to provide a comparison on different kinds of objects and deformable motions. [14] provided results on two challenging test sequences, their average deformation and geometry error are 21.05 cm and 14.87 cm respectively, while our approach achieves average errors of 3.63 cm and 0.48 cm. Our approach outperforms the state of the art by a large margin. The methods [36] and [37] do not
Table 2: Comparison with state-of-the-art approaches. Our learned correspondences significantly improve both tracking and reconstruction quality. We also provide ablation studies on training data type and different network parts.

<table>
<thead>
<tr>
<th>Method</th>
<th>Def. error (cm)</th>
<th>Geo. error (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynamicFusion re-impl.</td>
<td>28.31</td>
<td>1.08</td>
</tr>
<tr>
<td>VolumeDeform [17]</td>
<td>21.27</td>
<td>7.78</td>
</tr>
<tr>
<td>DynamicFusion + 3Dmatch</td>
<td>6.64</td>
<td>1.59</td>
</tr>
<tr>
<td>Ours-Rigid</td>
<td>12.21</td>
<td>2.30</td>
</tr>
<tr>
<td>Ours-Sparse</td>
<td>8.24</td>
<td>0.77</td>
</tr>
<tr>
<td>Ours-SelfSupervised</td>
<td>5.47</td>
<td>0.54</td>
</tr>
<tr>
<td>Ours-Rigid</td>
<td>3.94</td>
<td>0.43</td>
</tr>
<tr>
<td>Ours-Occlusion</td>
<td>3.70</td>
<td>0.42</td>
</tr>
<tr>
<td>Ours-Occlusion+Depth</td>
<td><strong>3.28</strong></td>
<td><strong>0.41</strong></td>
</tr>
</tbody>
</table>

6.3. Ablation Study

We evaluated different components of our network and their effect on the reconstruction quality, see Tab. 2. Since some sequences include motions in which large parts of the reconstructed object are occluded, as can be observed in Fig. 6, using visibility detection for correspondence pruning makes our method more robust. Furthermore, since depth measurements and heatmap predictions can both be sometimes noisy, adding correspondence filtering with depth prediction further improves the reconstruction results.

6.4. Data Generation Evaluation

To show the importance of our semi-supervised strategy for constructing the training corpus, we evaluate how different training corpora influence the performance of data-driven reconstruction methods. Aside from our training data, which has been generated using dense semi-supervised frame alignment, we used a publicly available rigid dataset of indoor scenes (from [5]), self-supervised alignment of our sequences (as in [33]), and only manually annotated sparse samples from our dataset. We provide comparison on both non-rigid matching in Tab. 1 and non-rigid reconstruction in Tab. 2. Using only rigid data does not generalize to non-rigid sequences. While sparse matches already improve network performance, there is not enough data for reliable correspondence prediction on every part of the observed object. In addition, annotated sparse matches are usually matches on image parts that are easy for humans to match, and there are much less accurate correspondences in areas with uniform color. In the self-supervised setting, the network gets dense correspondence information, which improves the method’s reconstruction performance compared to using only sparse features. However, without semi-supervised densely aligned frame pairs, we can only generate matches for the simple deformations, where the DynamicFusion approach can successfully track the motion. Therefore, the performance of the network trained only on self-supervised data degrades considerably on more extreme deformations, as can be seen in Fig. 7. Dense alignment of too far away frames is needed for accurate network prediction also in the case of extreme deformations. Since the majority of the dense aligned matches is still moving rigidly, it turned out to be beneficial to sample more deformable samples during training. In order to estimate which parts of the scene are more deformable, we employed sparsely annotated matches and ran RANSAC in combination with a Procrustes algorithm to estimate the average rigid pose. The more the motion of each sampled match differs from the average rigid motion, the more often we sample it during network training using a multinomial distribution of the non-rigid displacement weights. This
strategy improved the network performance, as is shown in Tab. 1, compared to training on non-weighted samples. Finally, we demonstrate how much data is needed to achieve robust correspondence prediction performance; i.e., using less training data considerably degrades matching accuracy, as summarized in Tab. 1, where we trained networks using only 12.5%, 25.0%, and 50.0% of the training data.

6.5. Limitations

While learned correspondences make tracking of fast motion more robust, there is still room for improvement when reconstructing dynamic objects. One pressing issue is that background clutter might be accidentally fused with the object when the object is close to the background. In this case, the reconstructed shape would slowly grow and we might also start reconstructing the background. This can cause wrong deformation graph connectivity and lead to tracking failures. A potential future avenue is to subtract and ignore the background; e.g., we could use our annotated object masks to develop a data-driven method.

7. Conclusion

We have proposed a neural network architecture for matching correspondences in non-rigid sequences that operates on RGB-D frames and demonstrated that our learned descriptors outperform existing hand-crafted ones. In addition, we introduced the first large-scale dataset that is composed of 400 scenes, over 390,000 RGB-D frames, and 5,533 densely aligned frame pairs. The dataset is obtained with a semi-supervised strategy by combining self-supervision with sparse annotations to obtain dense inter-frame correspondences. We also provide a test set along with several metrics for evaluating non-rigid matching and non-rigid reconstruction. We believe that our dataset is a first step towards enabling learning-based non-rigid matching and our benchmark will help to quantitatively and objectively compare different approaches.

Acknowledgements

We would like to thank the expert annotators Sathy Ashok, Omar Hedayat, Haya Irfan, Azza Jenane, Soh Yee Lee, Suzana Spasova, and Weile Weng for their efforts in building the dataset. We further thank Edgar Tretisch for his help with data collection, Xiaochen Fan for his valuable contributions during his research visit, and Armen Avetisyan for numerous helpful discussions. This work was supported by the Max Planck Center for Visual Computing and Communications (MPC-VCC), a TUM-IAS Rudolf Mößbauer Fellowship, the ERC Starting Grant Scan2CAD (804724), the ERC Consolidator Grant 4DRepLy (770784), and the German Research Foundation (DFG) Grant Making Machine Learning on Static and Dynamic 3D Data Practical.
References


[38] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015. 3


[45] Ruizhe Wang, Lingyu Wei, Etienne Vouga, Qi-Xing Huang, Duygu Ceylan, Gérard G. Medioni, and Hao Li. Capturing dynamic textured surfaces of moving targets. CoRR, abs/1604.02801, 2016. 3


[48] Lingyu Wei, Qiixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In Computer Vision and Pattern Recognition (CVPR), 2016. 1, 3


