DispVoxNets: Non-Rigid Point Set Alignment with Supervised Learning Proxies

Soshi Shimada$^{1,2}$ Vladislav Golyanik$^3$ Edgar Tretschk$^3$ Didier Stricker$^{1,2}$ Christian Theobalt$^3$

$^1$University of Kaiserslautern $^2$DFKI $^3$MPI for Informatics, SIC

Abstract

We introduce a supervised-learning framework for non-rigid point set alignment of a new kind — Displacements on Voxels Networks (DispVoxNets) — which abstracts away from the point set representation and regresses 3D displacement fields on regularly sampled proxy 3D voxel grids. Thanks to recently released collections of deformable objects with known intra-state correspondences, DispVoxNets learn a deformation model and further priors (e.g., weak point topology preservation) for different object categories such as cloths, human bodies and faces. DispVoxNets cope with large deformations, noise and clustered outliers more robustly than the state-of-the-art. At test time, our approach runs orders of magnitude faster than previous techniques. All properties of DispVoxNets are ascertained numerically and qualitatively in extensive experiments and comparisons to several previous methods.

1. Introduction

Point sets are raw shape representations which can implicitly encode surfaces and volumetric structures with inhomogeneous sampling densities. Many 3D vision techniques generate point sets which need to be subsequently aligned for various tasks such as shape recognition, appearance transfer and shape completion, among others.

The objective of non-rigid point set registration (NRPSR) is the recovery of a general displacement field aligning template and reference point sets, as well as correspondences between those. In contrast to rigid or affine alignment, where all template points transform according to a single shared transformation, in NRPSR, every point of the template has an individual transformation. Nonetheless, real structures do not evolve arbitrarily and often preserve the point topology.

1.1. Motivation and Contributions

On the one hand, existing general-purpose NRPSR techniques struggle to align point clouds differing by large non-linear deformations or articulations (e.g., significantly different facial expressions or body poses) and cause overregularisation, flattening and structure distortions [3, 9, 40, 31]. On the other hand, specialised methods exploit additional engineered (often class-specific) priors to align articulated and highly varying structures [39, 43, 13, 54, 56, 18]. In contrast, we are interested in a general-purpose method supporting large deformations and articulations (such as those shown in Fig. 1), which is robust to noise and clustered outliers and which can adapt to various object classes.

It is desirable but challenging to combine all these properties into a single technique. We address this difficult problem with supervised learning on collections of deformable objects with known intra-state correspondences. Even though deep learning is broadly and successfully applied to various tasks in computer vision, its applications to NRPSR have not been demonstrated in the literature so far (see Sec. 2). One of the reasons is the varying cardinalities...
Figure 2: Overview of our approach. The objective is to non-rigidly align a template $Y$ to a reference $X$. In the displacement estimation stage, we first convert the point sets to a voxel representation (P2V). DispVoxNets then regress per-voxel displacements that we apply to $Y$ (V2P), see Table 1 for more details on the network architecture. The subsequent stage refines the displacements with the unsupervised point projection loss ($L_{PP}$). Trilinear weights are applied to the displacements for interpolation and are used to compute weighted gradients in the backward pass, see Fig. 3 and our supplement for more details on the trilinear interpolation.

2. Related Work

Methods with Global Regularisers. When correspondences between points are given, an optimal rigid transformation between the point sets can be estimated in a closed form [27]. Iterative Closest Point (ICP) alternates between estimating the correspondences based on the nearest neighbour rule and local transformations until convergence [3, 7]. ICP is a simple and widely-used point set alignment technique, with multiple policies available to improve its convergence properties, runtime and robustness to noise [23, 48, 21, 12]. In practice, conditions for a successful alignment with ICP (an accurate initialisation and no disturbing effects such as noise) are often not satisfied. Extensions of ICP for the non-rigid case employ thin splines for topology regularisation [9] or Markov random fields linked by a non-linear potential function [24].

Probabilistic approaches operate with multiply-linked point associations. Robust Point Matching (RPM) with a thin-plate splines (TPS) deformation model [9] uses a combination of soft-assign [14] and deterministic annealing for non-rigid alignment. As the transformation approaches the optimal solution, the correspondences become more and more certain. In [61], point set alignment is formulated as a graph matching problem which aims to maximise the

1 introduced by Chui et al. [9] as a baseline ICP modification
ber of matched edges in the graphs. The otherwise NP-hard combinatorial graph matching problem is approximated as a constrained optimisation problem with continuous variables by relaxation labeling. In Gaussian Mixture Model (GMM) Registration (GMR) [31], NRPSR is interpreted as minimising the distance between two mixtures of Gaussians with a TPS mapping. The method was shown to be more tolerant to outliers and more statistically robust than TPS-RPM and non-rigid ICP. Myronenko and Song [40] interpret NRPSR as fitting a GMM (template points) to data (reference points) and regularise displacement fields using motion coherence theory. The resulting Coherent Point Drift (CPD) was shown to handle noisy and partially overlapping data with unprecedented accuracy. Zhou et al. [62] investigate the advantages of the Student’s-t Mixture Model over a GMM. The method of Ma et al. [38] alternates between correspondence estimation with shape context descriptors and transformation estimation by minimising the $\ell_2$ distance between two densities. Their method demands deformations to lie in the reproducing kernel Hilbert space. 

Recently, physics-based alignment approaches were discovered [11, 15, 1, 30, 19]. Deng et al. [11] minimise a distance metric between Schrödinger distance transforms performed on the point sets. Their method has shown an improved recall, i.e., the portion of correctly recovered correspondences. Ali et al. [1] align point sets as systems of particles with masses deforming under simulated gravitational forces. Gravitational simulation combined with smoothed particle hydrodynamics regularisation place this approach among the most resilient to large amounts of uniform noise in the data, and, at the same time, most computationally expensive techniques. In contrast, the proposed approach executes in just a few seconds and is robust to large amounts of noise due to our training policy with noise augmentation.

Large Deformations and Articulations. If point sets differ by large deformations and articulations, global topology regularisers of the methods discussed so far often overconstrain local deformations. Several extended versions of ICP address the case of articulated bodies with the primary applications to human hands and bodies [39, 43, 54]. Ge et al. [13] extend CPD with a local linear embedding which accounts for multiple non-coherent motions and local deformations. The method assumes a uniform sampling density and its accuracy promptly decays with an increasing level of noise. Some methods align articulated bodies with problem-specific segmented templates [16]. In contrast to all these techniques, our DispVoxNets can be trained for an arbitrary object class and are not restricted to a single template. Furthermore, our approach is resilient to sampling densities, large amounts of outliers and missing data. It grasps the intrinsic class-specific deformation model on multiple scales (global and localised deformations) directly from data. A more in-depth overview and comparison of NRPSR methods can be found in [55, 63].

Voxel-Based Methods. Voxel-based methods have been an active research domain of 3D reconstruction over the past decades [50, 6, 4, 44, 33, 37, 49, 58]. Their core idea is to discretise the target volume of space in which the reconstruction is performed. With the renaissance of deep learning in the modern era [34, 20, 53, 29, 26], there have been multiple attempts to adapt voxel-based techniques to learning-based 3D reconstruction [8, 46]. These methods have been criticised for a high training complexity due to expensive 3D convolutions and discretisation artefacts due to a low grid resolution. In contrast to previous works, we use a voxel-based proxy to regress displacements instead of deformed shapes. Note that in many related tasks, deformations are parametrised by lower-resolution data structures such as deformation graphs [52, 41, 60]. To alleviate discretisation artefacts and enable superresolution of displacements, we apply point projection and trilinear interpolation.

Learning Deformation Models. Recently, the first supervised learning methods trained for a deformation model were proposed for monocular non-rigid 3D reconstruction [17, 45, 51, 57]. Their main idea is to train a deep neural network (DNN) for feasible deformation modes from collections of deforming objects with known correspondences between non-rigid states. Implicitly, a single shape at rest (a thin surface) is assumed which is deformed upon 2D observations. Next, several works include a free-form deformation component for monocular rigid 3D reconstruction with an object-class template [35, 28]. Hanocka et al. [25] align meshes in a voxelised representation with an unsupervised learning approach. They learn a shape-aware deformation prior from shape datasets and can handle incomplete data. 

Our method is inspired by the concept of a learned deformation model. In NRPSR, both the reference and template address the case of articulated bodies with the primary applications to human hands and bodies [39, 43, 54]. Ge et al. [13] extend CPD with a local linear embedding which accounts for multiple non-coherent motions and local deformations. The method assumes a uniform sampling density and its accuracy promptly decays with an increasing level of noise. Some methods align articulated bodies with problem-specific segmented templates [16]. In contrast to all these techniques, our DispVoxNets can be trained for an arbitrary object class and are not restricted to a single template. Furthermore, our approach is resilient to sampling densities, large amounts of outliers and missing data. It grasps the intrinsic class-specific deformation model on multiple scales (global and localised deformations) directly from data. A more in-depth overview and comparison of NRPSR methods can be found in [55, 63].
the cardinalities of \(X\) are two point sets: the reference \(\mathbf{X}\) and \(\mathbf{Y}\). Notations and Assumptions. The inputs of our algorithm are two point sets: the reference \(\mathbf{X} = (x_1, \ldots, x_N)^T \in \mathbb{R}^{N \times D}\) and the template \(\mathbf{Y} = (y_1, \ldots, y_M)^T \in \mathbb{R}^{M \times D}\) which has to be non-rigidly matched to \(\mathbf{X}\). \(N\) and \(M\) are the cardinalities of \(\mathbf{X}\) and \(\mathbf{Y}\) respectively, and \(D\) denotes the point set dimensionality. We assume the general case when \(M \neq N\) and \(D = 3\) in all experiments, although our method is directly applicable to \(D = 2\) and generalisable to \(D > 3\) if training data is available and a voxel grid is feasible in this dimension. Our objective is to find the displacement function (a vector field) \(v: \mathbb{R}^D \times \mathbb{R}^3 \rightarrow \mathbb{R}^3\) so that \(\mathbf{Y} + v(\mathbf{Y}, \mathbf{X})\) matches \(\mathbf{X}\) as close as possible.

3. The Proposed Approach

We propose a neural network-based method for NRPSR that takes a template and a reference point set and returns a displacement vector for each template point, see Fig. 2 for an overview. As described in Sec. 2, existing methods show lower accuracy in the presence of large deformations between point sets. We expect that neural-network-based methods are able to deal with such challenging cases since they learn class-specific priors implicitly during training. In NRPSR, the numbers of points in the template and reference are generally different. This inconsistency of the input dimensionality is problematic because we need to fix the number of neurons before training. To resolve this issue, we convert the point sets into a regular voxel-grid representation at the beginning of the pipeline, which makes our approach invariant with respect to the number and order of input points. Furthermore, due to the nature of convolutional layers, we expect a network with 3D convolutions to be robust to noises and outliers. Even though handling 3D voxel data is computationally demanding, modern hardware supports sufficiently fine-grained voxel grids which our approach relies on.

Table 1: U-Net-style architecture of DispVoxNet. The concatenation column contains the layer IDs whose outputs are concatenated and used as an input to the current layer. We use a negative slope for LeakyRelU of 0.01.

![Diagram](image.png)

**Figure 3:** Overview of the forward and backward pass of trilinear interpolation on a voxel grid. Numbers ①, ②, ③ and ④ indicate the sequence of steps performed in every training iteration, see Sec. 3.1 for more details.

**Notations and Assumptions.** The inputs of our algorithm are two point sets: the reference \(\mathbf{X} = (x_1, \ldots, x_N)^T \in \mathbb{R}^{N \times D}\) and the template \(\mathbf{Y} = (y_1, \ldots, y_M)^T \in \mathbb{R}^{M \times D}\) which has to be non-rigidly matched to \(\mathbf{X}\). \(N\) and \(M\) are the cardinalities of \(\mathbf{X}\) and \(\mathbf{Y}\) respectively, and \(D\) denotes the point set dimensionality. We assume the general case when \(M \neq N\) and \(D = 3\) in all experiments, although our method is directly applicable to \(D = 2\) and generalisable to \(D > 3\) if training data is available and a voxel grid is feasible in this dimension. Our objective is to find the displacement function (a vector field) \(v: \mathbb{R}^D \times \mathbb{R}^3 \rightarrow \mathbb{R}^3\) so that \(\mathbf{Y} + v(\mathbf{Y}, \mathbf{X})\) matches \(\mathbf{X}\) as close as possible.

There is no universal criterion for optimal matching and it varies from scenario to scenario. We demand 1) that \(v\) results in realistic class-specific object deformations so that the global alignment is recovered along with fine local deformations, and 2) that the template deformation preserves the point topology as far as possible. The first requirement remains very challenging for current general NRPSR methods. Either the shapes are globally matched while fine details are disregarded or the main deformation component is neglected which can lead to distorted registrations. The problem becomes even more ill-posed due to noise in the point sets. Some methods apply multiscale matching or parameter adjustment schemes [21, 18]. Even though a relaxation of the global topology-preserving constraint can lead to a finer local alignment, there is an increased chance of arriving at a local minimum and fitting to noise.

Let \(\mathbf{V}_X\) and \(\mathbf{V}_Y\) be the voxel grids, i.e., the voxel-based proxies on which DispVoxNets regress deformations. Without loss of generality, we assume that \(\mathbf{V}_X\) and \(\mathbf{V}_Y\) are cubic and both have equal dimensions \(Q = 64\). We propose to learn \(v\) as described next.

3.1. Architecture

Our method is composed of displacement estimation and refinement stages. Each stage contains a DispVoxNet, i.e., a 3D-convolutational neural network based on a U-Net architecture [47], which we also denote by \(D_{vn}\). See Figs. 2–3 and Table 1 for details on the network architecture.

### Displacement Estimation (DE) Stage

We first discretise both \(\mathbf{Y}\) and \(\mathbf{X}\) on \(\mathbf{V}_Y\) and \(\mathbf{V}_X\), respectively (\(P2V\) in Fig. 2). During the conversion, the point-to-voxel correspondences are stored in an affinity table. As voxel grids sample the space uniformly, each point of \(\mathbf{X}\) and \(\mathbf{Y}\) is mapped to one of the \(Q^3\) voxels in \(\mathbf{V}_X\) and \(\mathbf{V}_Y\), respec-
tively. In $V_X$ and $V_Y$, we represent $X$ and $Y$ as binary voxel occupancy indicators, i.e., if at least one point falls into a voxel, the voxel’s occupancy is set to 1; otherwise, it equals to 0. Starting from $V_X$ and $V_Y$, DispVoxNet regresses per-voxel displacements of the dimension $Q^3 \times 3$. During training, we penalise the discrepancy between the inferred voxel displacements and the ground truth displacements $Z$ using a mean squared error normalised by the number of voxels. $Z$ is obtained by converting the ground truth point correspondences to the voxel-based representation of dimensions $Q^3 \times 3$ compatible with our architecture. The point displacement loss is given by:

$$L_{\text{Disp}}(Z, V_Y, V_X) = \frac{1}{Q^3} \|Z - D_{\text{vox}}(V_Y, V_X)\|_2^2. \quad (1)$$

Using the affinity table between $Y$ and $V_Y$, we determine each point’s displacement by applying trilinear interpolation on the eight nearest displacements in the voxel grid ($V2P$ in Fig. 2 and see supplement). After adding the displacements to $Y$, we observe that the resulting output after a single DispVoxNet bears some roughness. The refinement stage described in the following alleviates this problem.

**Refinement Stage.** Since the DE stage accounts for global deformations but misses some fine details, the unresolved residual displacements at the refinement stage are small. Recall that DispVoxNet is exposed to scenarios with small displacements during the training of the DE stage, since our datasets also contain similar (but not equal) states. Thus, assuming small displacements, we design a refinement stage as a combination of a pre-trained DispVoxNet and an additional unsupervised loss. Eventually, the refinement stage resolves the remaining small displacements and smooths the displacement field. To summarise, at the beginning of the refinement stage, the already deformed template point set is converted into the voxel representation $V_Y$. From $V_Y$ and $V_X$, a pre-trained DispVoxNet learns to regress refined per-voxel displacements.

To apply the inferred voxel displacements to a template point at the end of the refinement stage (see Fig. 3), (I) we compute the trilinear interpolation of the eight nearest displacements of $y_i, i \in \{1, \ldots, M\}$, and calculate a weighted consensus displacement for $y_i$. (II) The weights and indices of the eight nearest voxels are saved in an affinity table. To further increase the accuracy, (III) we introduce the unsupervised, differentiable point projection (PP) loss between the final output $Y + v(Y, X)$ and $X$. The PP loss penalises the Euclidean distances between a point $y^i$ in $Y + v(Y, X)$ and its closest point $x_{y^i}$ in $X$:

$$L_{\text{PP}}(Y + v(Y, X), X) = \frac{1}{M} \sum_{i=1}^{M} \|y^i - x_{y^i}\|_2. \quad (2)$$

We employ a $k$-d tree to determine $x_{y^i}$ for all $y^i$ in $Y + v(Y, X)$, see Fig. 4 for a schematic visualisation.

Since the training is performed through backpropagation, we need to ensure the differentiability of all network stages. Our approach contains conversions from voxel to point-set representations and vice versa that are not fully differentiable. Thanks to the affinity table, we know the correspondences between points and voxels at the refinement stage. Therefore, (IV) gradients back-propagated from the PP loss can be distributed into the corresponding voxels in the voxel grid as shown in Fig. 3. As eight displacements contribute to the displacement of a point due to trilinear interpolation in the forward pass, the gradient of the point is back-propagated to the eight nearest voxels in the voxel grid according to the trilinear weights from the forward pass.

Two consecutive DispVoxNets in the DE and refinement stages implement a hierarchy with two granularity levels for the regression of point displacements. Combining more stages does not significantly improve the result, i.e., we find two DispVoxNets necessary and sufficient.

### 3.2. Training Details

We use Adam [32] optimiser with a learning rate of $3 \cdot 10^{-4}$. As the number of points varies between the training pairs, we set the batch size to 1. We train the stages in two consecutive phases, starting with the DE stage using the displacement loss until convergence. This allows the network to regress rough displacements between $Y$ and $X$. Then, another instance of DispVoxNet in the refinement stage is trained using only the PP loss. We initialise it with the weights from DispVoxNet of the DE stage. Since the PP loss considers only one nearest neighbour, we need to ensure that each output point from the DE stage is already close to its corresponding point in $X$. Thus, we freeze the weights of DispVoxNet in the DE stage when training the refinement stage. See our supplement for training statistics.

To enhance robustness of the network to noise and missing points, $0 – 30\%$ of the points are removed at random from $Y$ and $X$, and uniform noise is added to both point sets. The number of added points ranges from $0\%$ to $100\%$.
of the number of points in the respective point set. The amount of noise per sample is determined randomly. When computing the PP loss, added noise is not considered.

4. Experiments

Our method is implemented in PyTorch [42]. The evaluation system consists of two Intel(R) Xeon(R) E5-2687W v3 running at 3.10GHz and a NVIDIA GTX 1080Ti GPU. We compare DispVoxNets with four methods with publicly available code, i.e., point-to-point non-rigid ICP (NR-ICP) [9], GMR [31], CPD [40] and CPD with Fast Gaussian Transform (FGT) [40]. FGT is a technique for the fast evaluation of Gaussians at multiple points [22].

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Ours</th>
<th>NR-ICP [9]</th>
<th>CPD [40]</th>
<th>GMR [31]</th>
</tr>
</thead>
<tbody>
<tr>
<td>thin plate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>0.0103</td>
<td>0.0402</td>
<td>0.0083 / 0.0192</td>
<td>0.2189</td>
</tr>
<tr>
<td>σ</td>
<td>0.0059</td>
<td>0.0273</td>
<td>0.0102 / 0.0083</td>
<td>1.021</td>
</tr>
<tr>
<td>FLAME [36]</td>
<td>e</td>
<td>0.0063</td>
<td>0.0588</td>
<td>0.0043 / 0.0094</td>
</tr>
<tr>
<td>σ</td>
<td>0.0009</td>
<td>0.0454</td>
<td>0.0008 / 0.0005</td>
<td>0.0007</td>
</tr>
<tr>
<td>DFAUST [5]</td>
<td>e</td>
<td>0.0166</td>
<td>0.0585</td>
<td>0.0683 / 0.0721</td>
</tr>
<tr>
<td>σ</td>
<td>0.0020</td>
<td>0.0215</td>
<td>0.0317 / 0.0256</td>
<td>0.8494</td>
</tr>
<tr>
<td>cloth [2]</td>
<td>e</td>
<td>0.0080</td>
<td>0.0225</td>
<td>0.0149 / 0.0173</td>
</tr>
<tr>
<td>σ</td>
<td>0.0021</td>
<td>0.0075</td>
<td>0.0066 / 0.0033</td>
<td>1.021</td>
</tr>
</tbody>
</table>

Table 2: Comparison of registration errors for all tested methods. For CPD, we also report results with FGT (right-hand values).

<table>
<thead>
<tr>
<th>Ablation Study</th>
<th>DE</th>
<th>DE + Ref. (nearest voxel)</th>
<th>Full: DE + Ref. (trilinear)</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>0.0100</td>
<td>0.0088</td>
<td>0.0069</td>
</tr>
<tr>
<td>σ</td>
<td>0.0021</td>
<td>0.0075</td>
<td>0.0016</td>
</tr>
</tbody>
</table>

Table 3: Ablation study highlighting the importance of the refinement stage and trilinear interpolation compared to a nearest voxel lookup.

Ablation Study. We conduct an ablation study on the cloth dataset to test the influence of each component of our architecture. The tested cases include 1) only DE stage, 2) DE and refinement stages with nearest voxel lookup (the naïve alternative to trilinear interpolation) and 3) our entire architecture with DE and refinement stages, plus trilinear interpolation. Quantitative results are shown in Table 3. The full model with trilinear interpolation reduces the error by more than 30% over the DE only setting.

4.2. Noiseless Data

We first evaluate the registration accuracy of our method and several baselines on noiseless data. Table 2 and Fig. 1 summarise the results. Our approach significantly outperforms other methods on the DFAUST and cloth datasets which contain articulated motion and large non-linear deformations between the template and reference. On thin plate, DispVoxNets perform on par with CPD (CPD with FGT) and show a lower e in three cases out of four. On FLAME, which contains localised and small deformations, our approach achieves e of the same order of magnitude as CPD and GMR. CPD, GMR and DispVoxNets outperform NR-ICP in all cases. The experiment confirms the advantage of our approach in aligning point sets with large global non-linear deformations (additionally, see the supplement).

4.3. Deteriorated Data

The experiments with deteriorated data follow the evaluation protocol of Sec. 4.2. We introduce clustered outliers and uniform noise to the data.

Structured Outliers. We evaluate the robustness of our method to added structured outliers and missing data. We either add a sphere-like object to the inputs or arbitrarily remove a chunk from one of the point sets. As summarised in Tables 4–5, our approach shows the highest accuracy among all methods on thin plate, DFAUST and cloth, even though DispVoxNets were not trained with clustered outliers and have not been exposed to sphere-like structures.
Tables 4–5 report results for both the cases with modified references and templates, and Fig. 6 compares results qualitatively. DispVoxNets are less influenced by the outliers and produce visually accurate alignments of regions with correspondences in both point sets. CPD, GMR and NR-ICP suffer from various effects, i.e., their alignments are severely influenced by outliers and the target regions are corrupted in many cases. We hypothesise that convolutional layers in DispVoxNet learn to extract informative features from the input points set and ignore noise. Furthermore, the network learns a class-specific deformation model which further enhances the robustness to outliers.

Uniform Noises. Next, we augment templates with uniform noise and repeat the experiment. Fig. 5 reports metrics for different amounts of added noise. Note that CPD, GMR and NR-ICP fail multiple times, and we define the success criterion in this experiment as $e < (4 \times \text{median})$ followed by $e < 4.0$. DispVoxNets show stable accuracy across different noise ratios and datasets, while the error of other approaches increases significantly (up to 100 times) with an increasing amount of noise. Only our approach is agnostic to large amount of noise, despite CPD explicitly modeling a uniform noise component. For a qualitative comparison, see the fifth and sixth rows in Fig. 6 as well as the supplement.

4.4. Runtime Analysis

We prepare five point set pairs out of $DFAUST$ dataset where the number of points varies from 1.5k to 10k. The runtime plots for different number of points are shown in Fig. 7. The numbers of points in both point sets are kept equal in this experiment. For 10k points, GMR, NR-ICP, CPD and CPD (FGT) take about 2 hours, 1.5 hours, 15 minutes, and 2 minutes per registration, respectively. Our approach requires only 1.5 seconds, which suggests its potential for applications with real-time constraints.

4.5. Real Scans

In this section, we demonstrate the generalisability of DispVoxNets to real-world scans. We test on 3D head point sets from the data collection of Dai et al. [10]. Since facial expression do not vary much in it, we use a reference from $FLAME$ [36] and a template from [10]. Registration results can be seen in Fig. 8. Although some distortion in the output shape is recognisable, DispVoxNets transfer the reference facial expression to the template. Even though the network has only seen $FLAME$ at training time, it is able to align two point sets of different cardinalities and origins.
5. Conclusions

We propose a DNN architecture with DispVoxNets — the first NRPSR approach with a supervised learning proxy — which regresses 3D displacements on regularly sampled voxel grids. Thanks to two consecutive DispVoxNets with trilinear interpolation and point projection, our approach outperforms other NRPSR methods by a large margin in scenarios with large deformations, articulations, clustered outliers and large amount of noise, while not relying on engineered class-specific priors. The runtime of DispVoxNets is around one second whereas other methods can take a few hours per registration, which suggests that our approach can be used in interactive applications. We show a high degree of generalisability of our architecture to real scans.

We believe that the direction revealed in this paper has a high potential for further investigation. Even though DispVoxNets represent a promising step towards accurate non-rigid point set alignment, its accuracy is limited by the resolution of the voxel grid and composition of training datasets. In future work, our method could be extended for operation on non-uniform voxel grids, and other types of losses could be investigated. Embedding of further alignment cues such as point normals, curvature, colours as well as sparse prior matches is a logical next step. We also expect to see extensions of DispVoxNets for point sets with small overlaps and adaptations of the proposed architecture for learning-based depth map fusion systems.
References


[36] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 36(6), 2017. 2, 6, 7, 8


[40] A. Myronenko and X. Song. Point-set registration: Coherent point drift. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. 1, 2, 3, 6, 7


