Incorporating Temporal Information into 3D Hand Pose Estimation Using Scene Flow

Niklas Hermes[1,2], Alexander Bigalke[1] and Mattias P. Heinrich[1]
[1]*Institute of Medical Informatics, Lübeck University, Germany*
[2]*Gestigon GmbH, Lübeck, Germany*

Keywords: Deep Learning, Hand Pose Estimation, 3D Point Clouds, Tracking, Temporal Information, Scene Flow.

Abstract: In this paper we present a novel approach that uses 3D point cloud sequences to integrate temporal information and spatial constraints into existing 3D hand pose estimation methods in order to establish an improved prediction of 3D hand poses. We utilize scene flow to match correspondences between two point sets and present a method that optimizes and harnesses existing scene flow networks for the application of 3D hand pose estimation. For increased generalizability, we propose a module that learns to recognize spatial hand pose associations to transform existing poses into a low-dimensional pose space. In a comprehensive evaluation on the public dataset NYU, we show the benefits of our individual modules and provide insights into the generalization capabilities and the behaviour of our method with noisy data. Furthermore, we demonstrate that our method reduces the error of existing state-of-the-art 3D hand pose estimation methods by up to 7.6%. With a speed of over 40 fps our method is real-time capable and can be integrated into existing 3D hand pose estimation methods with little computational overhead.

1 INTRODUCTION

3D human hand pose estimation is becoming increasingly important especially in the fields of virtual or augmented reality, as it enables an intuitive way for a human-machine interaction and provides a realistic user experience (Buckingham, 2021). In this way, interaction possibilities such as the use of gesture language (Nielsen et al., 2004) or object manipulation (Buchmann et al., 2004) can be brought into applications. The use of 3D point clouds has the general advantage of being more independent from the camera setup, for example the data of multiple depth sensors can be fused in a dense 3D point cloud (Hu et al., 2021) and thus be used.

Earlier approaches attached gloves or additional sensors to the user's hand to obtain precise data for estimating the hand pose (Wang and Popović, 2009; Ma et al., 2011) with the disadvantage that this also entails a restriction in the movement of a user. This led to the development of purely image-based methods for hand pose estimation, which advanced significantly with the progress of deep learning (Chatzis et al., 2020). A successful but error prone approach is to predict the pose of a hand at an individual point in time (Zhang et al., 2020; Rezaei et al., 2023). While performant, the largest remaining challenge for these methods is to deal with frequently occurring self-occlusions (Barsoum, 2016), which make it difficult to determine the hand pose precisely and negatively influence human-machine interaction.

The integration of temporal information is one way to remedy this problem, as it enables to propagate information from previous frames in which the presently occluded parts were visible. Different approaches to determine temporal correspondences in a scene already exist and are currently being actively researched. Scene flow methods analyze the motion of a scene given by a sequence of 3D point clouds and enable a more comprehensive understanding of the dynamics in the scene (Gu et al., 2019; Mittal et al., 2020). While these methods can accurately estimate point-wise correspondences, it remains unclear and has not been investigated yet how 3D pose estimation can benefit from scene flow.

We strongly believe that incorporating temporal information to pose estimation methods in the form of scene flow is a key to overcome the occlusion problem and achieve accurate pose predictions. Therefore we propose, to our knowledge, the first network that simultaneously estimates skeleton scene flow and 3D hand poses. Given the frame-wise predictions of an

Hermes, N., Bigalke, A. and Heinrich, M.
Incorporating Temporal Information into 3D Hand Pose Estimation Using Scene Flow.
DOI: 10.5220/0012361400003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 4: VISAPP*, pages 286-294
ISBN: 978-989-758-679-8; ISSN: 2184-4321

arbitrary pose estimator, the key idea is to analyze the motion of the bone structure between consecutive points in time with a scene flow module to improve the pose predictions of the estimator. Particularly noteworthy is the independence of our method in the choice of networks used to determine the scene flow and to estimate the initial hand pose. Thus, it is agnostic to a specific implementation choice, makes use of pretrained models and can continue to benefit from the future development of both fields by exchanging them against newer networks.

In the following, we briefly summarize the main contributions of this work.

- Proposal of a new network that combines the information in a 3D point cloud sequence to improve existing per-frame pose predictions. Once trained, the network can be used out of the box to improve new per-frame pose prediction methods.
- Proposal of a new module that learns spatial pose relations and transfers poses into a lower-dimensional space to increase the generalization capabilities of a network.
- Introduction of a method that optimizes existing scene flow prediction networks to track pose keypoints based on the flow.
- Exclusive evaluation on a publicly available hand dataset to demonstrate the accuracy of our method and to obtain new insights in 3D hand pose estimation.

Besides the convincing qualitative results of our method, we demonstrate its generalization capabilities in a comprehensive evaluation and show that we achieve significant improvements for 3D hand pose estimation.

2 RELATED WORK

In this section we briefly summarize previous work in the area of incorporating temporal information into pose estimation, which is most related to this work.

3D Pose from Single Frames. Besides the methods that employ depth maps (Moon et al., 2018; Xiong et al., 2019), there are also some existing methods that use 3D point clouds to estimate the hand pose frame-by-frame. Some approaches use PointNet (Qi et al., 2016) layers in a hierarchical structure to capture local and global features to regress the hand pose (Ge et al., 2018b; Ge et al., 2018a). Another method (Hermes et al., 2022) extends the hand point cloud by support points to then extract features through graph CNNs (Wang et al., 2018) to determine the hand pose. A further approach to extract features in point clouds are Residual Permutation Equivariant Layers (Li and Lee, 2019). In their work, point-wise poses are predicted and then the final pose is determined by voting.

3D Hand Tracking. An existing 3D hand tracking approach (Chen et al., 2022) regresses the hand position for a current hand point cloud by updating the previous pose estimate. The previous pose is used to determine a global hand pose based on which a transformation of all point clouds in a canonical space is performed. This, in turn, is used to extract features and thus determine the current pose. Since this is a tracking method, the main idea is to transform an already found previous pose to the currently available data in the best possible way, which is different from our goal to create an improved pose estimate based on previous data.

Optical Flow-Based Pose Estimation. Other existing methods use the information extracted by optical flow to achieve an improved pose estimate (Alldieck et al., 2017; Liu et al., 2021). The computation of optical flow is a well-established technique to estimate the motion of pixels between consecutive frames in an image sequence. By analyzing the displacement of pixels, valuable information about the movement and velocity of objects within the scene can be obtained. An option is to use graph geodesic distances to establish correspondences between the 3D positions of human joints (Schwarz et al., 2012). This spatial information is combined with the temporal information that is extracted from intensity images using optical flow in order to refine the detected landmarks and thus to create a pose estimate by fitting a skeleton body model. A two-way improvement of pose and optical flow estimation represents another method (Arko et al., 2022). First, the pose is used to fine-tune the optical flow estimation to better fit the human pose. With this optimized flow an improvement of the pose can be achieved in return. Since optical flow can only be estimated on 2D grid data, the application of these methodologies to 3D point clouds is not directly possible.

Scene Flow Estimation. One way to analyze the motion and dynamics of a scene given by 3D point clouds is to estimate the scene flow. Scene flow methods (Liu et al., 2019; Li et al., 2021; Wang et al., 2021) predict the correspondences of the points of two 3D point clouds captured at two successive points in time. However, it has never been used in 3D hand pose estimation and it is uncertain until now how the pose estimation can benefit from this information.

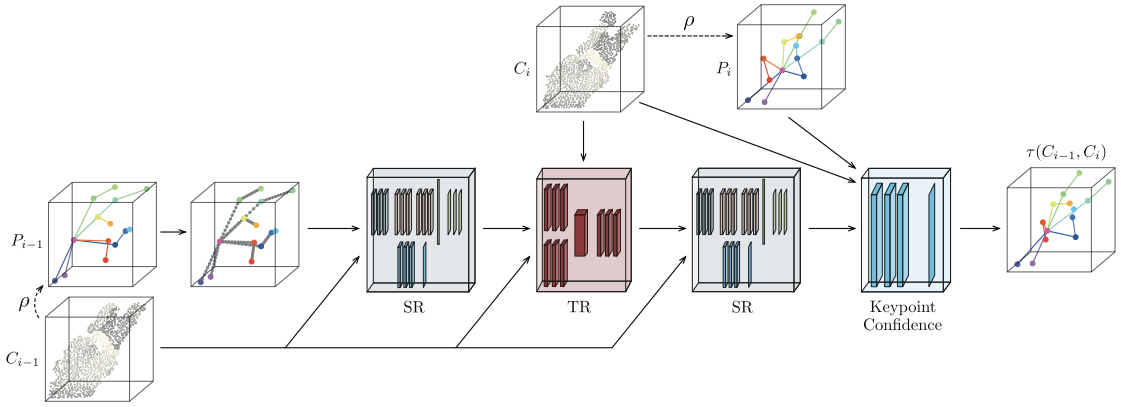

Figure 1: Overview of the presented network. The previous estimate P_{i-1} is extend by bone points and fed into a Spatial Refinement module (SR) with the previous cloud. The result is fed into a Temporal Refinement module (TR) together with both the current and previous point clouds C_i and C_{i-1}. After a consecutive spatial refinement, the final pose estimate is generated in a weighting module.

3 METHOD

Pose tracking deals with the detection of specific keypoints, given a sequence of temporally consecutive visual data. In the 3D case, we assume the consecutive acquisition of the object to be tracked at $N \in \mathbb{N}$ time points $t = \{t_1, ..., t_N\}$ which leads to a sequence of point clouds $\mathcal{C} = \{C_1, ..., C_N\}$ with $C_i \in \mathbb{R}^{L_{C_i} \times 3}$. Furthermore, we define t_N as the current point in time, thus the times t_i with $i < N$ have already passed. The objective is to find the 3D positions of the keypoints $K_N \in \mathbb{R}^{L_K \times 3}$ at the current time t_N, using the information of the whole sequence:

$$K_N = \varphi(\mathcal{C}). \tag{1}$$

As we want to improve the estimation of an existing pose estimator, we further assume that an already trained network ρ is available, which computes a prediction of the keypoints, given a single point cloud $\rho(C_i) = P_i$ with $P \in \mathbb{R}^{L_K \times 3}$. Our network $\tau(C_{N-1}, C_N)$ processes two consecutive frames at a time, allowing the entire sequence to be processed by executing the model sequentially.

Figure 1 shows a general overview of our model. First, the previous pose estimate P_{i-1} is extended by points that are supposed to represent the bone structure by constructing a connection graph and defining equidistant points on the connections. The pose data is then converted into a more plausible shape in the *Spatial Refinement* (SR) module. The constrained pose is transferred to the current point in time with additional input of the current data within the *Temporal Refinement* (TR) module. After penalizing unrealistic poses again in another SR module, the pose to be output is determined by confidence-based weighting of the improved pose and the existing estimate. Conceptually, our network thus consists mainly of two modules, which are intended to establish spatial as well as temporal correspondences to improve a given pose estimate. Both modules will therefore be described in more detail hereafter.

3.1 Spatial Correlation Regression

Many of the existing hand pose estimation methods learn to determine keypoints without directly constraining the correlation between anatomically important points. To address this issue, we present a *Spatial Refinement* (SR) module which estimates an anatomically reasonable pose, given any point cloud C_i with an existing prediction P_i. Conceptually, the module is based on the principle of Principal Component Analysis (PCA), extended by a confidence measure in order to keep already good predictions. An overview is shown in Fig. 2. The upper branch of the visualization shows the generation of the anatomically reasonable pose estimate. First, the local features of the point cloud are extracted at the specific keypoints. As in PointNet++ (Qi et al., 2017), the neighborhood in the point cloud is determined for each keypoint and the features of the local regions are extracted using successive PointNet (Qi et al., 2016) convolutions. In a next step, all keypoints are themselves defined as neighborhoods for each individual keypoint and translated into new features through further successive convolutions. This ensures to also capture global features of the predicted pose. A subsequent abstraction layer outputs a single feature vector of the pose which is used within multiple linear layers to estimate weights for a given set of principal components. Mul-

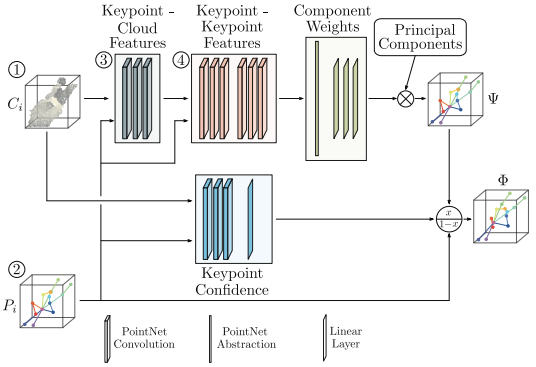

Figure 2: Overview of the Spatial Refinement module. 1.) Given a point cloud C_i and 2.) the corresponding keypoint predictions P_i, 3.) cloud features are generated using each keypoints closest neighbors in the point cloud. 4.) The resulting features are extended by a new dimension that contains all other keypoint features, resulting in keypoint-to-keypoint features after multiple convolutions. In a last step these features are converted into weights which are multiplied with the principal components to output a suitable pose Ψ. This pose is linearly interpolated with the existing prediction using the weights from the Keypoint Confidence module.

tiplying these weights with a set of the most important principal components results in an anatomically plausible pose estimate Ψ. By using only a subset of the principal components the resulting pose is transferred into a lower-dimensional pose space to avoid unnatural hand pose outliers.

The lower branch of the visualization shows the generation of confidence values for the existing prediction. As in the upper branch, the nearest neighbors in the point cloud are determined for each keypoint in order to extract local features through subsequent convolutions. The local features are clamped into confidence values between 0.01 and 1 by a linear layer. At this point, we prevent confidence values of 0 to avoid getting stuck in local extrema. The confidence values are used for linear interpolation between the initially predicted keypoints P_i and the refined pose Ψ, resulting in a final pose estimate Φ.

By transferring the existing pose into a lower dimensional space learned by PCA, a restriction of the possible keypoint positions explicitly accompanies it. In the definition of the loss function to be optimized, we further restrict the space of possible poses and require that the bone structure of the resulting pose is anatomically correct and that the 3D positions of the keypoints match the ground truth.

Let $\hat{P} \in \mathbb{R}^{N_K \times 3}$ be the ground truth 3D positions of the keypoints and further $B = \{(h,j) \mid h,\ j \in 1,...,N_K \text{ with } h \neq j\}$ a set of index tuples for spanning a bone vector with $P_{B_i^h} - P_{B_i^j}$. Then we define

$$\mathcal{L}_{bones}(P,\hat{P}) = \frac{1}{|B|}\sum_{i=1}^{|B|} \|(P_{B_i^h} - P_{B_i^j}) - (\hat{P}_{B_i^h} - \hat{P}_{B_i^j})\|_2^2 \tag{2}$$

as a bone loss function which enforces that the bone vectors of the resulting pose correspond to the ground truth without taking into account the correct position in space. We further define

$$\mathcal{L}_{position}(P,\hat{P}) = \frac{1}{N_K}\sum_{i=1}^{N_K} \|P_i - \hat{P}_i\|_2^2 \tag{3}$$

as position loss function that ensures that the 3D positions of a given prediction P are correct. This results in the following loss function to be minimised for the network:

$$\begin{aligned}\mathcal{L}_{spatial}(\Psi,\Phi,\hat{P}) = \alpha\mathcal{L}_{bones}(\Psi,\hat{P}) + \beta\mathcal{L}_{position}(\Psi,\hat{P}) \\ + \beta\mathcal{L}_{position}(\Phi,\hat{P}),\end{aligned} \tag{4}$$

with the hyperparameters $\alpha,\beta \in \mathbb{R}$, that weigh the bone loss against the spatially constrained pose estimate Ψ and the interpolated resulting pose Φ.

3.2 Temporal Correlation Regression

One of the biggest difficulties with pose estimation is the occlusion of important parts of the object. This issue is particularly prominent in the determination of hand poses and constitutes a major challenge. We present the *Temporal Refinement* (TR) module to tackle this problem. If an important part of the object is occluded at the current time t_N, then we assume that the part was visible at a previous point in time. The aim of this module is to transfer the information from previous frames to the current time in order to compensate for the missing information of the hidden structure and to ensure a more precise determination of the pose. An overview of the module is visualized in Fig. 3. Given two successively captured point clouds C_{i-1} and C_i, as well as their corresponding pose predictions P_{i-1} and P_i, the module outputs an improved pose estimate Θ that contains the temporal information of both points in time. This is based on a scene flow network, which determines the correspondences between the previous pose and the current prediction, subject to a small error. To better represent the skeletal structure of the object, we extend the previous pose P_{i-1} by points that lie on the bone vectors. Subsequently, the additional points are combined with the point cloud and the scene flow is determined. The scene flow can then be used to compute a new pose estimate by adding P_{i-1} and the corresponding flow

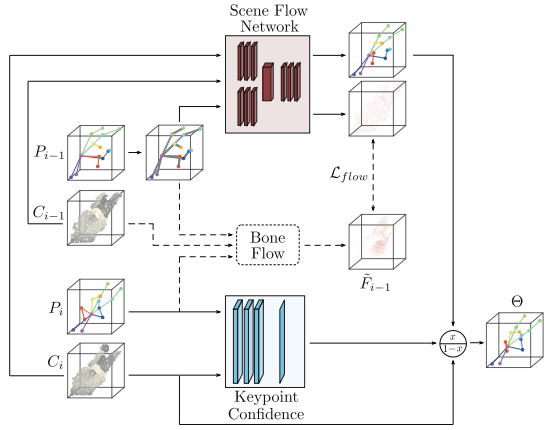

Figure 3: Structure of the Temporal Refinement module that receives two successively captured point clouds C_{i-1} and C_i as input and the corresponding pose predictions P_{i-1} and P_i. The predictions of the earlier captured cloud are extended by points that lie on bone vectors and fed into the scene flow network, together with both clouds. The scene flow network in turn outputs the flow of the earlier point cloud and a new pose estimate which is the addition of the flow and the prediction P_{i-1}. The final pose estimate Θ is received by linear interpolation between the new pose estimate and prediction P_i, using the confidence values of the Keypoint Confidence module. During training, a refined bone flow $\tilde{F}_{i-1}$ is computed that integrates the movement of the bones of the object to be tracked. This process is visualized with dashed lines.

subset. To prevent existing correct estimates from being discarded, we use estimated confidence values as in the Spatial Refinement module to linearly interpolate between the existing prediction P_i and the new pose estimate and receive a temporal refined pose estimate Θ.

Bone flow. A crucial part of the module is that the scene flow network is optimized to recognize the movement of the skeleton in particular. We therefore propose a bone flow refinement process during the training, which is visualized by the dashed lines. An existing method already optimizes optical flow using 2D body pose predictions and computing a pixel-wise flow of a stick-man, which is integrated into the ground truth optical flow (Arko et al., 2022). Since we do not use camera data throughout the method and thus no projected 2D predictions are available, the method is not readily applicable. We therefore propose to extend the same methodology for 3D poses.

We assume that the predictions P_i can be used to construct a bone-like graph consisting of bone vectors. For each of these vectors we specify an anatomically suitable third keypoint to construct a coordinate system using the cross product. For two predictions P_{i-1} and P_i we obtain the sets of bone vectors $\mathcal{B}_{i-1}$ and $\mathcal{B}_i$ as well as the corresponding coordinate systems. Let the point cloud C_{i-1} and the corresponding ground truth scene flow $\hat{F}_{i-1}$ be given. The following process produces a bone-motion specific flow $\tilde{F}_{i-1}$:

1. For each point of the cloud C_{i-1} find the closest bone vector in $\mathcal{B}_{i-1}$.
2. Translate each point to the current point in time i by transferring it into the coordinate system that belongs to the closest bone vector. Afterwards transfer it back to world coordinate system but using the coordinate system of the corresponding bone vector from B_i.
3. Compute the bone flow as vectors from the points towards their translated positions.
4. Linearly interpolate between the bone flow and the ground truth flow $\hat{F}_{i-1}$, where the distances of the points to the closest bone vectors are used to compute weights for interpolation.

To accommodate this revised flow in training, a suitable loss function must be selected. The requirements for a loss function to be optimized during training are that the flow network outputs the improved flow, as well as that the 3D positions of the keypoints of the resulting pose Θ are correctly determined. We therefore define the loss as:

$$\mathcal{L}_{temporal}(\Theta, F, \hat{P}_i, \tilde{F}) = \mathcal{L}_{position}(\Theta, \hat{P}_i) + \lambda \mathcal{L}_{flow}(F, \tilde{F}), \tag{5}$$

with the adjusting scalar $\lambda \in \mathbb{R}$, the network's pose estimate Θ, the estimated flow F, the ground truth keypoint positions $\hat{P}_i$ and the refined bone flow $\tilde{F}$. We further define the flow loss as:

$$\mathcal{L}_{flow}(F, \tilde{F}) = \frac{1}{|F|} \sum_{i=1}^{|F|} \frac{1}{3} \|F_i - \tilde{F}_i\|_1 \tag{6}$$

The data terms Eq. (4) and Eq. (5) of the presented modules lead to a convex optimization model to be minimized during training.

4 EVALUATION

In this section, we provide a comprehensive analysis of our method by conducting a quantitative evaluation.

Dataset. To the best of our knowledge, the most widely used datasets for evaluating per-frame 3D hand pose estimation methods are NYU (Tompson et al., 2014), MSRA (Sun et al., 2015), and ICVL (Tang et al., 2014). Since the MSRA dataset is less informative due to some erroneous annotations (Ge et al., 2018b) and a saturation of average joint errors

in the ICVL dataset (Wan et al., 2018), this evaluation is limited to the NYU dataset. Moreover, the NYU dataset contains more complex movements, making it the most challenging of the three datasets (Chen et al., 2018). It provides depth data of hands and ground truth 3D positions of 42 keypoints of the hand. However, we stick to the common practice for hand pose estimation methods and use only 14 keypoints (Wan et al., 2018; Ge et al., 2018b; Hermes et al., 2022). For training, 72K frames in 34 sequences are available and the evaluation is carried out on 8K frames in 14 sequences. It should also be mentioned that we do not use the depth data at any time, but only point clouds with 1024 points extracted from it.

Implementation Details. For training of the modules we constantly use the adjusting scalars $\alpha = \beta = \lambda = 1$, as well as an Adam (Kingma and Ba, 2014) optimizer with an initial learning rate of 0.001. The Spatial Refinement module was always trained independently with initial learning rate 0.0001. Both modules were trained using a batch size of 16 for a total of 40 epochs. During the tests we focus on a single network for the calculation of the scene flow, in order to be able to exclude the factor of different quality grades in the scene flow determination. Therefore we follow an existing work (Hermes et al., 2023) and select a pre-trained FLOT (Puy et al., 2020) network to determine the scene flow, due to its ability to handle deformations well. Furthermore, we refer to their approach and use the 42 keypoint positions to calculate a ground truth scene flow. All trainings and evaluations were run on a PC with a GeForce® GTX 1080 Ti GPU.

Metrics. As a measure of error, we consistently use the widely used *Average End Point Error* (EPE) to ensure comparability with previous work.

4.1 Impact of Scene Flow Refinement

In a first ablation study we investigate the influence of the proposed bone flow refinement process. We performed the following experiments on the NYU dataset, using only a subset of the training set. As per-frame pose prediction we used the results of the support points-based method (Hermes et al., 2022), that achieves an EPE of 9.44 mm.

Two successive point clouds with their corresponding pose predictions were used as input to improve the 3D keypoint positions of the temporally more recent acquisition. Table 1 provides an overview of the results. The trained FLOT network was used as baseline to determine the scene flow of the concatenation of the point cloud and the predictions of the previous frame. The addition of the previous pose with the scene flow results in the new hand pose. This methodology achieves an EPE of 9.92 mm and thus even degrades the per-frame prediction by 4.6%.

Table 1: Results of various experiments with the presented bone flow refinement process. Training and testing was performed on the NYU dataset using the support points-based hand pose predictions. The trained scene flow network was chosen as baseline without further optimization.

Method	**EPE** [mm]
per-frame	9.48
common flow	$9.92_{\uparrow 4.6\%}$
refined flow	$9.65_{\uparrow 1.8\%}$
refined flow + distance weights	$9.32_{\downarrow 1.7\%}$
refined flow + learned weights	$\mathbf{9.07}_{\downarrow 4.3\%}$

Optimizing the scene flow network with the bone flow refinement process improves the baseline results by 0.27 mm, but still results in higher errors as the per-frame predictions.

Assuming specific keypoints are estimated well, only the improved keypoints should influence the resulting pose. A weighted averaging between existing and improved pose addresses this issue. One potential weighting method is based on the Euclidean distance of the keypoints to the point cloud, serving as a measure of occlusion. This methodology achieves an error of 9.32 mm and thus improves the per-frame prediction by a small amount. Finally, we learned the weights during training rather than determining them algorithmically, yielding the lowest error and improving the existing per-frame prediction by 4.3%.

The experiments have thus shown that the presented bone flow refinement process improves the usual scene flow for tracking a pose. Furthermore, they show that scene flow is well suited for tracking certain keypoints, especially occluded keypoints with missing information. If enough information is available for the position of a keypoint, per-frame pose prediction is superior.

4.2 Hand Pose Estimation on NYU Dataset

This experiment shows the different contributions of the presented modules of our network. We refer to per-frame predictions of different methods for a quantitative evaluation and analyze the behavior of our network in the presence of noisy pose estimates as inputs. Furthermore, we check the behavior of our proposed method when only little training data is available.

The different modules as well as the whole network were trained using the predictions determined by the support points-based network. Evaluation was additionally performed on the pose predictions of P2P

Table 2: EPE [mm] evaluation on the NYU dataset. The training is based on the predictions from the support points-based method, while evaluation was performed with the information of five frames using the predictions of various hand pose estimation methods. The best results are highlighted.

	Support	**P2P**	**V2V**
Per-frame	9.48	9.05	8.42
Ours-spatial	$9.38_{\downarrow 1.1\%}$	$9.05_{\downarrow 0\%}$	$8.44_{\uparrow 0.2\%}$
Ours-temporal	$\mathbf{8.76}_{\downarrow 7.6\%}$	$8.65_{\downarrow 4.4\%}$	$8.24_{\downarrow 2.1\%}$
Ours-full	$8.81_{\downarrow 7.1\%}$	$\mathbf{8.61}_{\downarrow 4.9\%}$	$\mathbf{8.07}_{\downarrow 4.2\%}$

(Ge et al., 2018b) and V2V (Moon et al., 2018), both of which yield state of the art per-frame results. The EPE is listed in Tab. 2, based on consecutive execution of our network to process the information of five frames.

An individual analysis of the SR module shows that limiting to a spatial consideration does not significantly improve the pose estimation. Bringing in temporal information has a greater impact and improves the per-frame predictions of the three considered methods. The combination of temporal information and spatial correlations (Ours-full) continues to improve the results for P2P slightly and noticeably for V2V, but marginally worsens the results of the support-based prediction. Since the estimated poses of P2P and V2V were not part of the training, we infer from the results that the SR module noticeably improves the already good generalization ability of the TR module. Additionally, it can be assumed that the network adapted slightly too much to the predictions in the training data during the training.

With p-values far below 0.05, a paired t-test proves that the difference between the EPE means of the per-frame predictions and the results of our network is statistically significant. Furthermore, a runtime of over 40 fps is achieved for the processing of two point clouds, including the time for the scene flow network, which demonstrates high computationally efficiency and real-time capability.

Evaluation on Corrupted Input Poses. In real scenarios, external influences can always cause noise in the pose detection. Since the incoming pose predictions are an important ingredient for our method, we perform an evaluation with corrupted incoming poses to investigate the behavior of our method with noisy data. For this purpose, we applied Gaussian noise with varying variance to the pose predictions of the method based on support points. We used the same network as in the previous experiment for further evaluations, which was trained with non-noisy pose predictions as input. The results of the evaluation are shown in Fig. 4.

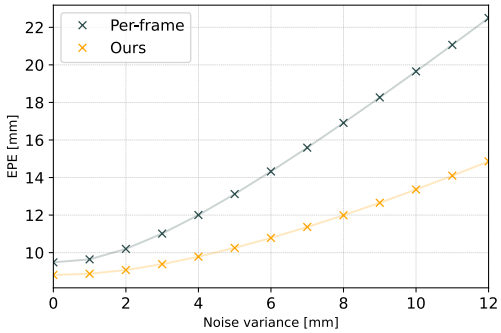

Figure 4: Error evolution of our method on ingoing pose predictions corrupted with Gaussian noise of increasing variance. The upper curve shows the EPE of the per-frame predictions under the effect of the noise and the lower curve shows the results of our network with the noisy pose predictions as input.

The upper curve shows the evolution of the EPE of the noisy per-frame predictions as the variance of the noise increases, and thus serves as a baseline for this experiment. The lower curve shows the EPE of our method, based on the respective noisy pose predictions as input. Both error curves increase linearly with the increase of the variance of the noise, but the slope of the curve of our method is significantly lower and thus achieves a smaller increase of the EPE as the baseline. These results show that our proposed network can counteract the error of noisy poses.

4.3 Comparison with HandTrackNet

The already existing HandTrackNet method (Chen et al., 2022) tracks hand poses using 3D point clouds. Given the previous hand keypoint positions and the current 3D hand point cloud, both inputs are transformed into a canonical space, whereas the transformation is calculated using an initial hand pose and the current 3D positions of the palm keypoints. Subsequently, features are extracted on the basis of which the previous pose is placed in the current hand point cloud. The aim of this process is to transfer a previous pose into a current 3D hand point cloud in the best possible way, whereas our method improves an existing pose estimate based on previous data. As these objectives differ from each other, a direct comparison is not readily possible. Nevertheless, we explore the differences in these two approaches in a joint evaluation.

For the evaluation, the already pre-trained HandTrackNet was used and tested on the NYU (Tompson et al., 2014) dataset. Since the procedure requires the position of 6 palm keypoints, we extended the 14 keypoints by the 6 required palm keypoints that are also part of the ground truth in the NYU dataset. The er-

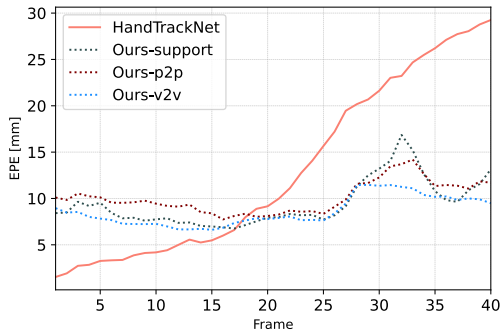

Figure 5: EPE average over the six exemplary sequences of 40 frames. The pre-trained HandTrackNet was initialized with the ground truth pose. Our trained network was evaluated with the pose estimations predicted by the support point-based method, P2P and V2V.

ror calculation was carried out on the 14 keypoints as before.

As a first experiment, similar to the evaluation setup in Sec. 4.2 we had HandTrackNet estimate the hand poses based on five previous frames, using the pose predictions of the support points-based method as initial pose. This resulted in an EPE of 9.67 mm which is worse than the per-frame predictions with an error of 9.48 mm. Since the method only tracks a pose and does not improve it, this result was to be expected, because the initial pose was already faulty and the tracking accumulated the errors.

In a second experiment, we initialized the HandTrackNet on six exemplary sequences of 40 frames each with a ground truth pose and tracked them continuously. On the same sequences, we evaluated our from Sec. 4.2 trained network, using the different pose estimators. The averaged EPE of the six evaluated sequences is depicted in Fig. 5, illustrating that while HandTrackNet initially exhibits lower error due to initialization with ground truth data, its errors accumulate over time, leading to a continuous increase of the EPE. In contrast, our method can quickly recover even after erroneous frames since new keypoint predictions are consistently incorporated into the process.

5 CONCLUSION

We propose a method for incorporating temporal information into hand pose estimation from a sequence of 3D point clouds. To the best of our knowledge, we are the first to utilize scene flow for an improved pose estimation. In this context, we introduce a process that optimizes arbitrary scene flow networks for the application of pose estimation by focusing on the motion of the bone structure. This procedure is supplemented by a module that recognizes the spatial pose correlations and transfers existing poses into a lower-dimensional pose space in order to avoid unnatural outliers and to improve the generalization capability. In a comprehensive evaluation we demonstrate that the proposed method significantly improves existing frame-by-frame methods for 3D hand pose estimation. We further show that the network is able to reduce the negative influence of noisy pose predictions. As a future perspective, we are convinced that the information gained through scene flow will provide an even greater advantage when directly integrated into the pose estimation model.

REFERENCES

Alldieck, T., Kassubeck, M., Wandt, B., Rosenhahn, B., and Magnor, M. (2017). Optical flow-based 3d human motion estimation from monocular video. In Roth, V. and Vetter, T., editors, *Pattern Recognition*, pages 347–360, Cham. Springer International Publishing.

Arko, A. R., Little, J. J., and Yi, K. M. (2022). Bootstrapping human optical flow and pose.

Barsoum, E. (2016). Articulated hand pose estimation review. *CoRR*, abs/1604.06195.

Buchmann, V., Violich, S., Billinghurst, M., and Cockburn, A. (2004). Fingartips: Gesture based direct manipulation in augmented reality. In *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, GRAPHITE '04, page 212–221, New York, NY, USA. Association for Computing Machinery.

Buckingham, G. (2021). Hand tracking for immersive virtual reality: opportunities and challenges. *Frontiers in Virtual Reality*, 2:728461.

Chatzis, T., Stergioulas, A., Konstantinidis, D., Dimitropoulos, K., and Daras, P. (2020). A comprehensive study on deep learning-based 3d hand pose estimation methods. *Applied Sciences*, 10(19):6850.

Chen, J., Yan, M., Zhang, J., Xu, Y., Li, X., Weng, Y., Yi, L., Song, S., and Wang, H. (2022). Tracking and reconstructing hand object interactions from point cloud sequences in the wild. *arXiv preprint arXiv:2209.12009*.

Chen, X., Wang, G., Zhang, C., Kim, T.-K., and Ji, X. (2018). Shpr-net: Deep semantic hand pose regression from point clouds. *IEEE Access*, 6:43425–43439.

Ge, L., Cai, Y., Weng, J., and Yuan, J. (2018a). Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ge, L., Ren, Z., and Yuan, J. (2018b). Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Gu, X., Wang, Y., Wu, C., Lee, Y. J., and Wang, P. (2019). Hplflownet: Hierarchical permutohedral lat-

tice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hermes, N., Bigalke, A., and Heinrich, M. P. (2023). Point cloud-based scene flow estimation on realistically deformable objects: A benchmark of deep learning-based methods. *Journal of Visual Communication and Image Representation*, page 103893.

Hermes, N., Hansen, L., Bigalke, A., and Heinrich, M. P. (2022). Support point sets for improving contactless interaction in geometric learning for hand pose estimation. pages 89–94.

Hu, T., Lin, G., Han, Z., and Zwicker, M. (2021). Learning to generate dense point clouds with textures on multiple categories. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2169–2178, Los Alamitos, CA, USA. IEEE Computer Society.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Li, R., Lin, G., He, T., Liu, F., and Shen, C. (2021). Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 364–373, Los Alamitos, CA, USA. IEEE Computer Society.

Li, S. and Lee, D. (2019). Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11927–11936.

Liu, X., Qi, C. R., and Guibas, L. J. (2019). Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, X., Yu, S.-y., Flierman, N. A., Loyola, S., Kamermans, M., Hoogland, T. M., and De Zeeuw, C. I. (2021). Optiflex: Multi-frame animal pose estimation combining deep learning with optical flow. *Frontiers in Cellular Neuroscience*, 15.

Ma, Y., Mao, Z.-H., Jia, W., Li, C., Yang, J., and Sun, M. (2011). Magnetic hand tracking for human-computer interface. *IEEE Transactions on Magnetics*, 47(5):970–973.

Mittal, H., Okorn, B., and Held, D. (2020). Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Moon, G., Chang, J., and Lee, K. M. (2018). V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nielsen, M., Störring, M., Moeslund, T. B., and Granum, E. (2004). A procedure for developing intuitive and ergonomic gesture interfaces for hci. In Camurri, A. and Volpe, G., editors, *Gesture-Based Communication in Human-Computer Interaction*, pages 409–420, Berlin, Heidelberg. Springer Berlin Heidelberg.

Puy, G., Boulch, A., and Marlet, R. (2020). FLOT: Scene Flow on Point Clouds Guided by Optimal Transport. In *European Conference on Computer Vision*.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016). Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*.

Rezaei, M., Rastgoo, R., and Athitsos, V. (2023). Trihornnet: A model for accurate depth-based 3d hand pose estimation. *Expert Systems with Applications*, page 119922.

Schwarz, L. A., Mkhitaryan, A., Mateus, D., and Navab, N. (2012). Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing*, 30(3):217–226. Best of Automatic Face and Gesture Recognition 2011.

Sun, X., Wei, Y., Liang, S., Tang, X., and Sun, J. (2015). Cascaded hand pose regression. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 824–832.

Tang, D., Chang, H. J., Tejani, A., and Kim, T.-K. (2014). Latent regression forest: Structured estimation of 3d articulated hand posture. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793.

Tompson, J., Stein, M., Lecun, Y., and Perlin, K. (2014). Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33.

Wan, C., Probst, T., Gool, L. V., and Yao, A. (2018). Dense 3d regression for hand pose estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5147–5156.

Wang, H., Pang, J., Lodhi, M. A., Tian, Y., and Tian, D. (2021). Festa: Flow estimation via spatial-temporal attention for scene point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14173–14182.

Wang, R. Y. and Popović, J. (2009). Real-time hand-tracking with a color glove. *ACM Trans. Graph.*, 28(3).

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2018). Dynamic graph CNN for learning on point clouds. *CoRR*, abs/1801.07829.

Xiong, F., Zhang, B., Xiao, Y., Cao, Z., Yu, T., Zhou Tianyi, J., and Yuan, J. (2019). A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV)*.

Zhang, Z., Xie, S., Chen, M., and Zhu, H. (2020). Handaugment: A simple data augmentation method for depth-based 3d hand pose estimation. *arXiv*, pages arXiv–2001.