scientific reports

OPEN

Scene flow based deep network for hand reconstruction using depth images

Adnan Anwer[1], Jameel Malik[1], Khawar Khurshid[2]✉, Ahmed Elhayek[3], Ramy Battrawy[4], Didier Stricker[4,5], Syed Sarib Naveed[2] & Muhammad Usman Safder[2]

Accurate 3D hand reconstruction remains a challenging computer vision problem with numerous applications. Existing approaches predominantly focus on single-frame hand reconstruction, thereby neglecting crucial temporal information essential for stable hand tracking. A novel pipeline for 3D hand reconstruction from consecutive multi-view depth images, termed HandFlowNet, is presented in this work. The proposed methodology converts multi-view depth images into a single point cloud, from which temporal information between consecutive frames is deduced through estimated scene flow of hand mesh vertices. The scene flow estimator establishes one-to-one correspondences between point sets from sequential depth frames. This scene flow is subsequently applied as an offset to initially estimated hand mesh vertices from the previous frame to determine the current frame's hand mesh vertices. These vertices are further refined using a graph convolutional network that incorporates predicted local and global features of the current frame. Through extensive evaluations, HandFlowNet is demonstrated to achieve state-of-the-art performance on public real hand benchmarks including DexYCB and HO3D.

The reconstruction of hands in three-dimensional space constitutes a critical component across numerous applications, including virtual and augmented reality environments where seamless, jitter-free interaction between hands and virtual objects is imperative. Additional applications encompass human-computer interaction, sign language production, user identification, gaming, gesture recognition, and motion capture[1–6]. Despite significant advances in this domain, achieving accurate and temporally stable three-dimensional hand reconstruction remains a formidable challenge within the computer vision community.

Recent investigations have demonstrated considerable progress in three-dimensional hand reconstruction from static imagery[7–10]. Nevertheless, the accurate estimation of temporally stable hand shape and pose in real-time continues to pose significant difficulties. These challenges stem from multiple factors, including the high dimensionality inherent in hand mesh representation, the substantial variation in hand shapes and poses across individuals, severe occlusion during natural movements, and inherent noise in depth data acquisition. These complications are further exacerbated in video sequences, where temporal consistency becomes paramount for the production of smooth and realistic hand motion.

The majority of existing methodologies primarily address hand reconstruction from individual frames[11–14]. However, temporal information is fundamentally essential for achieving smooth reconstruction of hand movements across sequential frames. Several investigations[15–21] have incorporated temporal data within given sequences to enhance hand reconstruction accuracy. For instance,[16] introduce a spatial-temporal graph structure applied to two-dimensional skeleton sequences, which is subsequently processed through graph convolutional networks to estimate three-dimensional poses from predicted two-dimensional pose sequences.[17] propose a self-supervised neural network that integrates optical flow features with estimated two-dimensional joint heat maps to determine three-dimensional hand shape and pose. While these approaches leverage temporal information, they frequently fail to capture the fine-grained motion cues essential for consistent and accurate hand reconstruction.

In contrast to previous approaches, our methodology for hand shape and pose reconstruction is guided by two key observations. First, estimating the three-dimensional movement of hand mesh vertices provides substantially richer geometric and motion information compared to two-dimensional optical flow or joint-based temporal feature fusion. Although joint-based methods effectively capture high-level motion patterns,

[1]National University of Sciences and Technology (NUST), Islamabad, Pakistan. [2]Namal University, Mianwali, Pakistan. [3]University of Prince Mugrin (UPM), Madinah, Saudi Arabia. [4]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany. [5]University of Kaiserslautern-Landau (RPTU), Landau, Germany. ✉email: khawar.khurshid@namal.edu.pk

they often sacrifice fine-grained spatial details that are critical for accurate and consistent hand reconstruction. Our approach estimates the three-dimensional motion of hand points-scene flow-which captures precise point-level displacements between consecutive frames. This explicit motion modeling is particularly crucial for hand reconstruction, where minute, intricate movements of the hand mesh significantly impact the accuracy of shape and pose estimation. Drawing inspiration from advances in LIDAR-based scene flow[22–24], we employ scene flow to address temporal inconsistencies and errors in single-frame reconstructions. By tracking the three-dimensional motion of hand vertices and joints, our method achieves more accurate and coherent results compared to static or joint-based methodologies.

Second, the point cloud representation of depth sequences offers superior suitability for our task. While this representation circumvents the limitations associated with voxelization or two-dimensional projections, it is not the primary focus of our investigation. Rather, we leverage the point cloud representation to enable efficient processing of three-dimensional spatial information, which complements our scene flow-based approach. Inspired by the work of[25] and the BiPointFlow-Net framework[26], we adopt the point set representation, enabling our network to efficiently process the 2.5D information contained within depth maps.

To our knowledge, we present the first scene flow-based hand reconstruction pipeline, designated HandFlowNet, which generates accurate three-dimensional hand shapes and poses given consecutive pairs of multi-view depth frames (see Fig. 1 for an overview). Our method leverages scene flow to explicitly model the three-dimensional motion of hand vertices and joints, enabling temporally consistent and accurate hand reconstruction. Through the combination of a transformer-based hand shape estimator with a graph convolution-based refinement network, HandFlowNet effectively addresses the challenges of temporal inconsistency and noisy depth data. This novel integration of scene flow estimation and multi-view point cloud processing establishes a new direction for three-dimensional hand reconstruction research.

For the initial frame, a novel transformer-based neural hand shape estimator predicts an accurate hand mesh and pose from the corresponding input multi-view point set. Subsequently, our method estimates the three-dimensional motion of hand points between consecutive frames utilizing a neural scene flow estimator. This estimator processes consecutive pairs of multi-view point sets as input and establishes a one-to-one mapping between the two point sets to compute dense scene flow. Dense scene flow refers to the three-dimensional motion vectors of all points in the input point set, while sparse scene flow captures the three-dimensional motion of hand joints and mesh vertices. The sparse scene flow is derived from the dense scene flow through application of the k-nearest neighbors algorithm ("Sparse scene flow estimation").

To estimate the hand shape of the current frame, the sparse scene flow is added to the hand shape of the previous frame. However, to mitigate potential error accumulation caused by the scene flow estimation network, a graph convolution-based refinement network is introduced. This refinement network utilizes global and local features computed from the point set of the current frame to produce a more accurate and refined hand shape ("Method overview").

In summary, we propose a novel approach for three-dimensional hand reconstruction from consecutive pairs of multi-view depth frames based on the three-dimensional scene flow of hands. The principal contributions of this paper are as follows:

- A novel hands scene flow estimation (SFE) approach which simultaneously and accurately estimates dense and sparse three-dimensional movement of hand joints and hand mesh vertices, respectively, from one frame to another ("Hand scene flow").
- A new transformer-based neural network for three-dimensional hand pose and shape estimation (HSE) utilizing multi-view point sets generated from multi-view depth maps ("Reference shape estimation").
- A new graph convolution-based hand shape refinement network (RM) which employs global and local features from point sets to further enhance hand pose and mesh estimation accuracy ("Hand shape refinement").
- Extensive experiments on two challenging real hands benchmarks (DexYCB[27] and HO3D[28]) demonstrate that HandFlowNet achieves superior or comparable results relative to existing methods (" Evaluation of hand shape estimation" and "Evaluation of hand pose estimation").

Related work

Hand pose and shape estimation from depth images and point sets is a critical problem in computer vision, influencing various applications. Over time, numerous approaches have been explored to address this challenge, alongside related methods for scene flow estimation from point sets.

Point cloud processing through deep learning

In the domain of point cloud processing, significant advances were made with the introduction of the PointNet [29] and PointNet++ [30] architectures, which brought groundbreaking improvements to the field. PointNet++ [30] particularly enhanced the preservation of geometric structures through the incorporation of ball query techniques and hierarchical clustering within a novel Set Abstraction Layer. This improvement allowed for better retention of the underlying geometric information present in point cloud data.

The advent of transformer architectures has led to numerous adaptations for three-dimensional point cloud processing tasks. Notably, [31] introduced Point Transformer, which applies vector attention-based feature aggregation to local neighborhoods of points, enabling more effective feature extraction. Similarly[32], achieved comparable performance with their Point Cloud Transformer through the application of Laplacian matrices for feature enhancement. Our proposed network architecture integrates these advancements by implementing a feature extractor comprising successive Set Abstraction layers [30] and Point Transformer blocks [31].

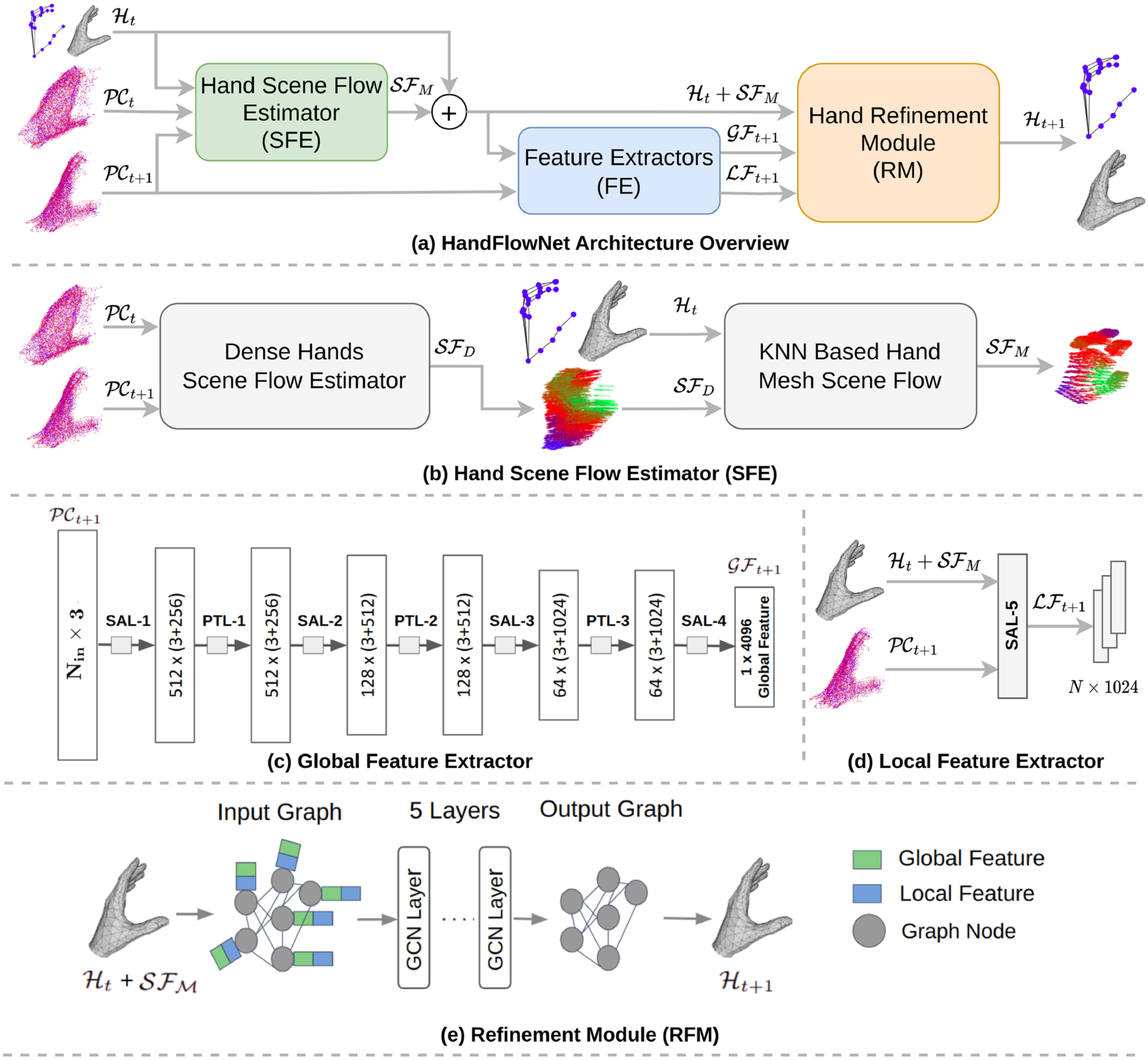


Fig. 1. HandFlowNet architecture. Given an estimated hand shape H_t and a pair of consecutive point sets (PC_t, PC_{t+1}), the Hand Scene Flow Estimator (SFE) module generates scene flow (SF_M) of hand mesh vertices and joints. For the first frame, H_t is extracted from PC_t through the reference hand shape estimator (HSE) module. For subsequent frames, the hand shape of the previous frame is fed back into SFE for tracking purposes. SF_M is added to H_t to obtain an initial estimate of hand shape for the next frame. To compensate for error propagation due to scene flow uncertainty in subsequent frames, $H_t + SF_M$ is refined by the hand refinement module (RM) using the local and global features (LF_{t+1} and GF_{t+1}, respectively) extracted from PC_{t+1}.

Scene flow estimation from point sets

The concept of three-dimensional scene flow estimation was initially formulated by [33] as a correspondence problem, focusing on the determination of three-dimensional motion vector fields for points on surfaces within a scene. Early investigations in this field predominantly utilized RGB [34–43] and RGB-D [44–49] stereo image pairs to derive three-dimensional motion vectors from two-dimensional optical flow estimates. More recently, the field has expanded to include scene flow estimation from monocular RGB images [50–54], demonstrating the versatility of this approach.

With the increasing prevalence of depth cameras and significant breakthroughs in point cloud processing algorithms [29–32,55], a trend has emerged towards the direct estimation of three-dimensional scene flow from cloud pairs [23,56,57] in an end-to-end manner. FlowNet3D [58] pioneered this approach by introducing a PointNet [29]-based flow embedding layer to address the correspondence problem between point sets. Subsequently, HPLFlownet [59] enhanced this methodology through the incorporation of multi-scale learning and bilateral convolutions [60,61]. PointPWCNet [62] achieved further significant improvements by implementing a coarse-to-fine estimation strategy within a pyramid-type network architecture.

Bi-PointFlowNet [22] introduced an innovative approach featuring bidirectional flow embedding layers that facilitate learning in both forward and backward flow directions, substantially improving the accuracy of scene flow estimation. In our research, we utilize a fine-tuned Bi-PointFlowNet as an off-the-shelf scene flow estimator for our experimental validations.

Hand reconstruction from depth images

Methods for hand shape reconstruction can be broadly categorized based on input type: two-dimensional depth image-based techniques [63–75] and three-dimensional data-based approaches [11,76–79]. Due to the inherent lack of three-dimensional geometric information, two-dimensional depth image-based techniques suffer from perspective distortions [9]. Consequently, three-dimensional data-based approaches utilizing voxels or point clouds are better suited for accurate hand shape and pose estimation tasks.

Among three-dimensional data-based methods, voxel-based approaches [9,78,79] employ three-dimensional convolutional neural networks, which, while effective, are computationally expensive. In contrast, point cloud processing networks [30,31] offer comparable or superior accuracy with significantly improved computational efficiency. [25,80] proposed PointNet [29]-based networks for direct regression of hand joints utilizing point-wise probability distributions, establishing a foundation for this approach. SHPR-Net [81] improved upon this work through the integration of an auxiliary semantic segmentation module, while the Point-to-pose method [82] further enhanced performance through residual permutation layers [83].

Recent advancements include HandFoldingNet [84], which introduced a multi-scale feature-guided folding mechanism for three-dimensional hand pose estimation, and HandOccNet [85], which improved shape estimation by transferring features from non-occluded to occluded regions through feature injecting transformer (FIT) and self-enhancing transformer (SET) blocks. HFL-Net [86] enhanced hand shape reconstruction by incorporating features extracted from occluding objects, while DiffHand [87] pioneered a diffusion-based framework with a novel cross-modality decoder for accurate modeling of vertex connectivity.

The Anchor-to-Joint regression network (A2J) [88] utilizes two-dimensional CNN-based local regressors to learn global-local spatial context for improved hand joint estimation. Adaptive Weighting Regression (AWR) [89] combines pixel data from dense representations to estimate hand joint coordinates, effectively leveraging the strengths of both detection and regression approaches. [90] introduced virtual view selection and fusion modules specifically designed for pose estimation tasks. The Image-Point cloud Network (IPNet) [75] combines two-dimensional CNNs for efficient initial estimation with iterative refinement in the point cloud domain to more effectively leverage three-dimensional geometry.

Graph convolutional neural networks (GCNN) [91–96] have recently demonstrated significant benefits for the refinement of hand pose and shape estimates. Our proposed HandFlowNet also incorporates a GCNN-based refinement module to enhance the accuracy of hand reconstruction.

Hand reconstruction using temporal information

Temporal information inherent in video sequences has been successfully utilized in various hand-related tasks, including pose estimation [28,97,98], shape estimation [99], gesture recognition [15,100–103], and hand tracking [97,104], typically using RGB image sequences as input data. Self-supervised and semi-supervised networks [18,105,106] frequently employ cyclic-consistency losses as temporal guides to enhance training efficiency.

Notable contributions in this area include the work of Cai et al. [16], who proposed a spatial-temporal graph convolutional neural network to simultaneously exploit geometric and temporal consistency in hand pose estimation. Wen et al. [15] introduced a hierarchical temporal transformer specifically designed for exploiting temporal consistency across frames. Chen et al. [17] implemented temporal cycle-consistency learning techniques [107] applied to two-dimensional optical flow for estimating hand poses throughout video sequences.

H2ONet [108] significantly enhances three-dimensional hand mesh reconstruction quality by exploiting non-occluded information from multiple frames through finger-level occlusion-aware feature fusion. Zhang et al. [109] proposed the Pyramidal Mesh Alignment Feedback (PyMAF) method for generating well-aligned meshes across temporal sequences. VIBE [19] employs a self-attention-based adversarial network to generate kinematically plausible three-dimensional pose and shape estimations from video without requiring in-the-wild ground-truth three-dimensional labels. TCMR [21] enhances temporal consistency through effective fusion of features from past and future frames using an intelligent feature integration architecture. Deformer [20] improves accuracy through a dynamic fusion transformer that learns spatial-temporal features specifically designed to handle occlusions.

In contrast to these approaches, our method utilizes three-dimensional scene flow estimated from temporally consecutive point clouds as a temporal indicator to directly predict three-dimensional hand shape and pose simultaneously, without requiring the entire video sequence. To the best of our knowledge, this represents the first attempt to incorporate three-dimensional scene flow as temporal information for simultaneous hand shape and pose estimation.

Multi-view hand reconstruction

The fusion of multiple hand views represents a challenging multi-view stereo (MVS) problem. Several approaches address this challenge by extracting two-dimensional features from each view and subsequently merging them in feature space [110–114]. Alternative methodologies extract three-dimensional features directly from point clouds [115–118] using three-dimensional convolutional neural networks or point cloud processing networks.

POEM [119] represents a state-of-the-art network that fuses two-dimensional features in three-dimensional space through sophisticated feature aggregation and structure-conscious vertex query mechanisms. [120] implemented a cross-view fusion method for multi-view aggregation in a self-supervised manner. HaMuCo [121] integrated single-view estimators through a cross-view interaction network. SiMA-Hand [122] improved accuracy in occluded regions using Single-to-Multi-view Adaptation through feature fusion at image, joint, and vertex levels. MLP Hand [123] employs a multi-view feature infusion network for tri-axis modeled per-bone reconstruction of hands, achieving superior real-time performance.

In contrast to these complex fusion approaches, our HandFlowNet implements a more straightforward yet effective methodology by directly merging point clouds into a global frame of reference using camera calibration matrices, simplifying the reconstruction process while maintaining high accuracy.

Method overview

Figure 1a illustrates our proposed approach for 3D hand reconstruction. Given a pair of temporally consecutive depth image frames, our objective is to accurately estimate the 3D hand shape and pose of the second frame $H_{t+1} \in R^{3\times(V+J)}$, where $V = 778$ represents MANO[124] hand mesh vertices and $J = 21$ denotes MANO hand joints. The input depth images D_t and D_{t+1} are transformed into 3D point sets $PC_t, PC_{t+1} \in R^{3\times N}$, where N is the number of 3D points per point set (see "Point set representation of depth image"). The hand scene flow estimator (SFE) computes the scene flow of hand SF_M, which represents the 3D movement of hand mesh vertices and joints from PC_t to PC_{t+1} (see "Hand scene flow").

For the initial frame in an image sequence, a reference hand shape and pose H_t is estimated using a reference hand shape estimator (HSE) based on a neural transformer network (see "Reference shape estimation"). The hand shape and pose H_{t+1} corresponding to the subsequent input point set PC_{t+1} is obtained by adding the scene flow of hand SF_M to the reference hand shape H_t is given in Eq. (1):

$$H_{t+1} = H_t + SF_M \tag{1}$$

It is important to note that the reference hand shape estimator is exclusively utilized for generating hand shape and pose for the first frame in a given image sequence. For tracking during subsequent frames, H_{t+1} is recursively fed back to the input of the Scene Flow Estimator via a multiplexer. To mitigate error accumulation inherent in the Scene Flow Estimator, the hand shape and pose estimated from Equation 1 is further refined using a hand refinement module (RM) (see "Hand shape refinement"). In this module, local and global features extracted from the input point set PC_{t+1} using neural transformer networks (see "Feature extractors") are concatenated with the scene flow based hand shape and pose estimate. This concatenated representation forms an input graph which is processed through GCNN to produce the final refined hand shape and pose (see "GCNN-based shape refinement").

The proposed HandFlowNet approach

In this section, the architecture and methodology of the proposed HandFlowNet pipeline are detailed. The approach effectively harnesses the temporal information embedded within the 3D motion of hand points across consecutive frames, represented as scene flow. For the first time, the scene flow of hand mesh vertices and joints is leveraged to accurately reconstruct both 3D hand shape and pose, thereby offering a novel perspective in hand motion modeling.

The transformation of depth images into single-view and multi-view point set representations is discussed in "Point set representation of depth image". The detailed architecture of the proposed hand mesh scene flow estimator is elaborated in "Hand scene flow". "Hand shape refinement" describes the hand refinement module, which integrates transformer and graph convolution-based deep networks to enhance hand pose and shape predictions. Finally, in "Reference shape estimation", a neural transformer-based reference hand shape estimator is introduced, facilitating robust shape reconstruction.

Point set representation of depth image

The input depth images D_t and D_{t+1} are transformed into corresponding point set representations PC_t and PC_{t+1} for each camera viewpoint by utilizing the camera intrinsic parameters. Subsequently, all viewpoints are merged using the camera extrinsic parameters to construct unified multi-view point sets. The transformation from a depth image D_t to its point set representation PC_t is performed in five distinct steps, as illustrated in Fig. 2.

First, the raw depth image is multiplied by a hand mask to suppress all pixels that do not belong to the hand region. *Second*, the camera intrinsic parameters are employed to project the non-zero pixels of the hand region into 3D camera coordinates. Due to inaccuracies in the hand mask, outliers appear in the hand point set, as indicated in orange in the top right and bottom left images of Fig. 2. In the *third* step, these outliers are removed using the DBSCAN density-based clustering algorithm [125]. The point set is partitioned into clusters based on two constraints: (i) the maximum distance between any two points within the same cluster does not exceed 5 mm, and (ii) each cluster must contain at least 5 points. The largest cluster is retained, while smaller clusters are discarded. *Fourth*, the point sets are transformed from camera 3D coordinates to world 3D coordinates using the camera's extrinsic parameters. Finally, in the *fifth* step, all single-view point sets are concatenated to obtain a unified multi-view point set per frame, as all individual point sets have already been transformed into world 3D coordinates.

The final multi-view point set (as depicted in Fig. 2) varies in point count depending on the hand posture and the degree of occlusion. In most cases, the number of points is sufficiently large; to ensure consistency, 4096 points are randomly subsampled from each set. In rare cases where a set contains fewer than 4096 points, points from the existing set are randomly selected and duplicated to reach this number, ensuring uniformity. For the DexYCB dataset, multi-view point sets containing fewer than 1024 points are discarded, as they indicate frames in which the hand is not visible.

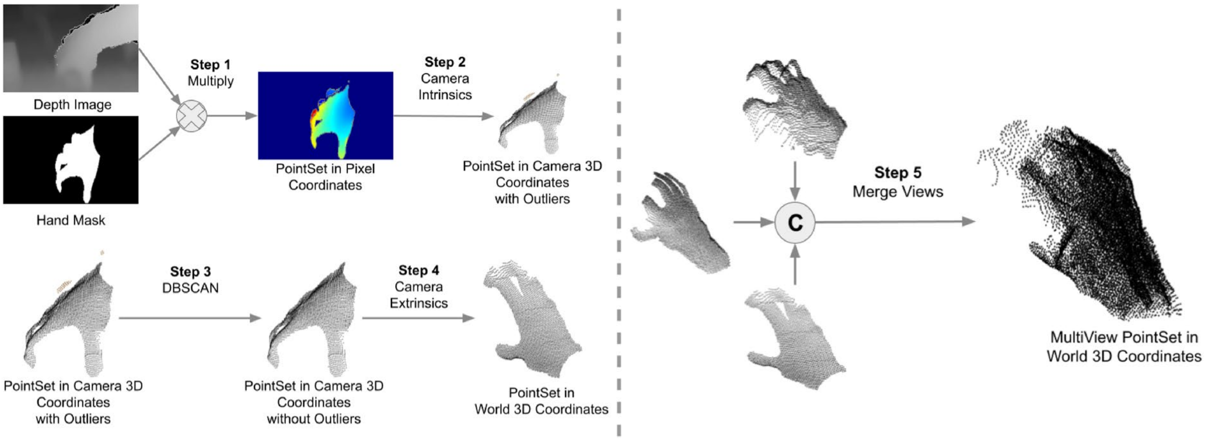

Depth Image
Hand Mask
Step 1
Multiply
PointSet in Pixel Coordinates
Step 2
Camera Intrinsics
PointSet in Camera 3D Coordinates with Outliers
Step 3
DBSCAN
PointSet in Camera 3D Coordinates with Outliers
PointSet in Camera 3D Coordinates without Outliers
Step 4
Camera Extrinsics
PointSet in World 3D Coordinates
C
Step 5
Merge Views
MultiView PointSet in World 3D Coordinates

Fig. 2. Multi-view point sets. The input depth image is first refined by applying a hand mask and subsequently transformed into camera 3D coordinates utilizing the camera's intrinsic parameters. The resulting point set inherently contains outliers (illustrated in orange), which are removed through the DBSCAN clustering algorithm [125]. Subsequently, the point set is projected from camera 3D coordinates to world 3D coordinates using the camera's extrinsic parameters. Once the point sets for all views are obtained, they are concatenated to construct a comprehensive multi-view point set representation.

Hand scene flow

As mentioned earlier, our main idea is to exploit the scene flow information between point sets to implicitly include the temporal information in a sequence for the estimation of hand shape. DexYCB and HO3D datasets do not provide scene flow ground truth. For this reason, we generate the sparse scene flow of hand mesh vertices by exploiting the movement of hand mesh vertices between previous and current frames. In the following subsections, we explain our method for annotating DexYCB dataset sequences for dense and sparse scene flow, and deep network architectures for scene flow estimation.

Our scene flow estimator comprises three primary stages. Initially, we generate ground truth scene flow annotations, as detailed in "Scene flow annotation". This process begins by computing *sparse scene flow* at the vertices of the hand mesh, derived from the differences in their 3D positions across consecutive frames. These sparse annotations are subsequently propagated to the entire observed hand surface, yielding *dense scene flow* for all points in the point cloud using a *k*-nearest neighbors (k-NN) approach. Next, we train a deep neural network to estimate dense scene flow directly from pairs of point sets, as described in "Dense scene flow estimation". Finally, for tracking the hand mesh and joints, we extract *sparse scene flow* at key locations from the dense predictions using a *k*-nearest neighbors (k-NN) approach, as outlined in "Sparse scene flow estimation".

The deep neural network, combined with the subsequent k-NN algorithm, constitutes the Hand Scene Flow Estimator (SFE), as illustrated in Fig. 1a.

Scene flow annotation

To obtain scene flow ground truth between given point sets, accurate point-to-point correspondences between 3D points are required. Once these correspondences are established, the scene flow ground truth SF_P of any 3D point P is defined as a vector directed from the location of this point in the previous frame $\mathbf{P_t}$ to its location in the current frame $\mathbf{P_{t+1}}$ is given in Eq. (2),

$$\mathrm{SF_P} = \mathbf{P_{t+1}} - \mathbf{P_t} \tag{2}$$

The DexYCB dataset provides comprehensive 3D hand shape annotations, enabling the derivation of sparse scene flow for hand mesh vertices, denoted as SF_M (illustrated in Fig. 4a). This sparse scene flow SF_M, comprising scene flow vectors at the locations of hand mesh vertices, is obtained through the Eq. (3):

$$SF_M = H_{t+1} - H_t \tag{3}$$

where H_t and H_{t+1} represent the known 3D hand shape annotations of two consecutive frames provided by the DexYCB dataset. The underlying assumption is that the scene flow within a local patch (or a group of 3D points) representing a rigid body (i.e., the hand) maintains relative consistency. Based on this fundamental assumption, dense scene flow annotations SF_D (depicted in Fig. 4b), which comprise scene flow vectors at the locations of all 3D points in point set PC_t, can be effectively deduced from the known SF_M annotations.

When the hand mesh vertices and joints H_t are overlaid on the point set PC_t, a mixed point set is formed, containing points from both H_t and PC_t. Within this integrated point set, the scene flow SF_M at the locations of H_t is already known, while the scene flow SF_D remains to be determined. In such a mixed point set scenario, the unknown scene flow SF_D can be reliably estimated from the known scene flow SF_M of neighboring points through the application of the k-nearest neighbors (k-NN) algorithm. Consequently, the scene flow SF_{D_i} of

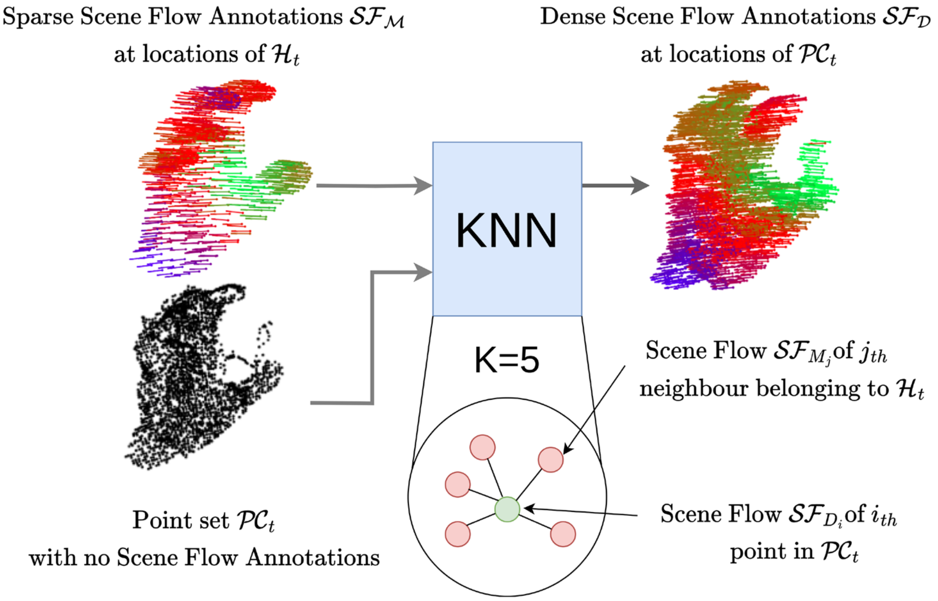

Sparse Scene Flow Annotations $\mathcal{SF_M}$ at locations of $\mathcal{H}_t$
Dense Scene Flow Annotations $\mathcal{SF_D}$ at locations of $\mathcal{PC}_t$
KNN
K=5
Scene Flow $\mathcal{SF}_{M_j}$of j_{th} neighbour belonging to $\mathcal{H}_t$
Point set $\mathcal{PC}_t$ with no Scene Flow Annotations
Scene Flow $\mathcal{SF}_{D_i}$of i_{th} point in $\mathcal{PC}_t$

Fig. 3. Dense scene flow annotations. SF_D of point set PC_t with no scene flow annotations can be estimated by averaging the sparse scene flow annotations SF_M of the k-nearest vertices of each point of PC_t in hand mesh H_t. This figure is best viewed in color.

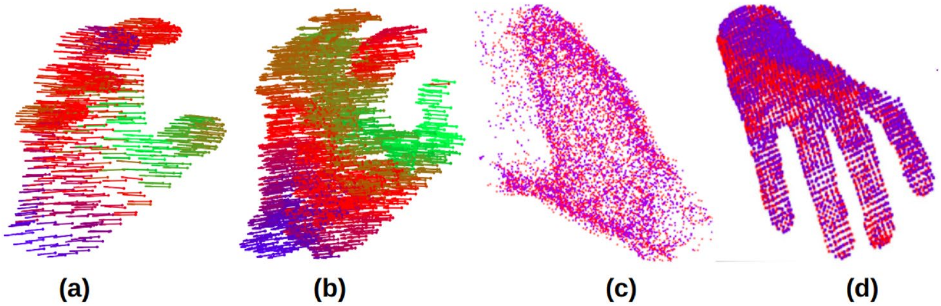


Fig. 4. Scene flow annotations for DexYCB[27]. **(a,b)** Depict sparse (SF_M) and dense (SF_D) scene flow annotations, respectively. Here, the color signifies normalized scene flow magnitude with blue as lowest and green as highest value. **(c,d)** Represent the point set PC_{t+1} (blue) overlaid on $PC_t + SF_D$ (red). This figure is best viewed in color.

the i_{th} point in PC_t is computed as the mean of scene flow vectors of its k-nearest vertices belonging to H_t, as given in Eq. (4):

$$SF_{D_i} = \frac{1}{k}\sum_{j=1}^{k} SF_{M_j} \tag{4}$$

where SF_{M_j} represents the scene flow vector of the j_{th} neighbor belonging to H_t, as illustrated in Fig. 3. The quality of the estimated SF_D is systematically verified by overlaying the point set PC_{t+1} with the transformed point set $PC_t + SF_D$. The resulting overlay visualizations are presented in Fig. 4c, d. In these visualizations, the blue color represents PC_{t+1} while red depicts $PC_t + SF_D$. The high degree of alignment observed between these two point sets serves as a robust indicator of the quality and accuracy of the dense scene flow annotations SF_D derived through the proposed methodology.

Dense scene flow estimation

Given input point sets PC_t and PC_{t+1}, a set of 3D motion vectors (i.e., the dense scene flow SF_D) corresponding to each 3D point in PC_t is estimated. Scene flow estimation represents a widely investigated research area, with numerous state-of-the-art deep learning architectures designed specifically for point cloud based scene flow estimation that achieve remarkable accuracy[22,23,126].

Bi-PointFlowNet[22] is employed as an off-the-shelf dense scene flow estimator in this work. This particular architecture is selected primarily for two compelling reasons: (i) it achieves superior performance on the challenging real-world KITTI benchmark[127], and (ii) it demonstrates enhanced cross-dataset generalization capabilities, as evidenced by the direct application of the model trained on the synthetic FlyingThings3D dataset[128] to KITTI evaluations without requiring fine-tuning or normalization procedures. The Bi-PointFlowNet architecture implements a sophisticated coarse-to-fine scene flow estimation approach within a pyramid-like structural framework, drawing inspiration from the seminal PointPWCNet design[62]. Additional architectural details of this network are provided comprehensively in the supplementary material for interested readers. Bi-PointFlowNet, originally trained on FlyingThings3D, exhibited suboptimal performance on our hand datasets due to inherent scale disparities between the training and target domains. To address this limitation, we adapted the network architecture for hand-specific applications through systematic input scaling and subsequent fine-tuning on dense scene flow annotations (detailed in "Scene flow annotation"). The scaling methodology and fine-tuning procedures are comprehensively analyzed in "Ablation studies" and "Evaluation of hand scene flow estimation", respectively.

When the input point sets are processed through Bi-PointFlowNet, the network generates estimated dense scene flow vectors SF_D at the precise location of each 3D point in the point set PC_t. A representative example of SF_D estimation resembles the visualization presented in Fig. 4b, providing a clear illustration of the estimated flow field.

Sparse scene flow estimation

Since the primary objective of this work is to track the 3D hand shape and pose using scene flow, an accurate estimation of the sparse scene flow SF_M of hand mesh vertices and joints is essential (as illustrated in Fig. 4a). Figure 1b illustrates the methodological approach for estimating sparse scene flow SF_M from the previously computed dense scene flow SF_D. When the hand mesh vertices and joints H_t are overlaid on the input point set PC_t, a heterogeneous mixed point set is formed, consisting of points from both H_t and PC_t. Within this integrated point set, the scene flow SF_D at locations corresponding to PC_t is known, whereas the scene flow SF_M at locations corresponding to H_t remains undetermined.

Given such a mixed point set configuration, the unknown scene flow SF_M can be systematically estimated from the known scene flow SF_D of neighboring points through the application of the k-nearest neighbors (k-NN) algorithm. Specifically, the scene flow vector SF_{M_i} of the i_{th} point in H_t is computed as the arithmetic mean of dense scene flow vectors SF_{D_j} of the k-nearest neighbors of this i_{th} point belonging to PC_t, as given in Eq. (5):

$$SF_{M_i} = \frac{1}{k}\sum_{j=1}^{k} SF_{D_j} \tag{5}$$

This bidirectional approach to scene flow estimation-from dense to sparse and vice versa-creates a robust framework for accurate hand tracking that effectively leverages the complementary strengths of both representation modalities.

Hand shape refinement

The proposed GCNN-based refinement module addresses inaccuracies in initial hand shape and pose estimates through a dual-branch architecture that exploits both global contextual information and local geometric features extracted from the input point cloud. The module employs two parallel processing pathways: one dedicated to shape refinement and another to pose refinement. Each branch integrates the coarse initial estimate with global contextual features and local geometric descriptors, subsequently applying Graph Convolutional Neural Networks (GCNNs) that influence the inherent topological structure of hand anatomy to predict precise vertex-wise and joint-wise corrections. This architectural design ensures the preservation of both global structural coherence and fine-grained local geometric details, thereby enabling accurate and anatomically plausible hand reconstructions.

The hand shape and pose of the subsequent frame H_{t+1} can be computed from the estimated reference hand shape H_t and sparse scene flow of the hand SF_M by utilizing Eq. (3). However, inaccuracies in the estimation of SF_M may propagate and exponentially amplify during hand tracking across successive frames. For this critical reason, the initially estimated hand shape and pose (i.e., $H_{t+1} = H_t + SF_M$) necessitates refinement based on discriminative features extracted from the point set PC_{t+1}. The proposed Refinement Module (RM) operates through a two-stage process: first, it extracts comprehensive local and global features from the point set PC_{t+1}, and subsequently, it employs these multi-scale features within GCNN to estimate the final refined hand shape and pose H_{t+1} with enhanced precision.

Feature extractors

Given an input point set and a scene flow based hand shape and pose estimate, the primary objective of the feature extractors is to derive complementary local and global features LF_{t+1} and GF_{t+1}, respectively. The framework incorporates two specialized feature extractors: a global feature extractor for capturing holistic contextual information and a local feature extractor for preserving fine-grained geometric details. These feature extractors are architected upon a synergistic combination of Set Abstraction Layers (SAL) and Point Transformer Layers (PTL), drawing inspiration from the well-established PointNet++[30] and Point Transformer[31] architectures.

Set abstraction layer (SAL): this specialized layer learns hierarchical features from the input point set through a systematic three-step process. First, the input point set is down-sampled into a more compact representation

Sr.	Name	$\mathbf{N_{out}}$	$\mathbf{N_s}$	$\mathbf{F_{in}}$	MLP	$\mathbf{F_{out}}$
1	SAL-1	512	4	3	128, 128, 246	256
2	SAL-2	512	4	256+3	256, 256, 512	512
3	SAL-3	256	4	512+3	512, 512, 1024	1024
4	SAL-4	1	4	1024+3	1024, 2048, 4096	4096
5	SAL-5	778	4	3	32, 64, 128, 256, 512, 1024	1024

Table 1. Feature extractor—SAL hyper-parameters.

Sr.	Name	$\mathbf{M_{dim}}$	$\mathbf{F_{dim}}$	$\mathbf{N_s}$
1	PTL-1	64	64	4
2	PTL-2	128	128	4
3	PTL-3	256	256	4

Table 2. Feature extractor—PTL hyper-parameters.

using the iterative farthest point sampling (FPS)[129] algorithm, which ensures maximum spatial coverage. Second, a grouping operation is performed wherein the local neighbors of each point within the down-sampled point set are collected from the original input point set utilizing the k-Nearest Neighbors (k-NN) algorithm. Finally, a shared PointNET[29] architecture is applied to each constructed group to estimate the respective output hierarchical feature corresponding to each point of the down-sampled point set. Comprehensive architectural details of this layer are provided in the supplementary material for thoroughness. Point transformer layer (PTL): this layer applies vector attention mechanisms to the features extracted by the preceding Set Abstraction Layer. The fundamental purpose of this transformer-based layer is to adaptively enhance the contribution of 3D point features based on their relative spatial configuration within a local neighborhood, thereby capturing long-range dependencies in the point cloud data. Detailed architectural specifications of this layer are elaborated in the supplementary material for completeness.

Global feature extraction: the global feature extractor component (see Fig. 1c) is constructed using four Set Abstraction Layers (SAL) and three Point Transformer Layers (PTL). The precise hyper-parameters selected for these complementary layers are meticulously documented in Tables 1 and 2.

While SAL-1, SAL-2, and SAL-3 generate per-point features without aggregation, SAL-4 specifically returns the final aggregated global feature vector that encapsulates the holistic geometric properties of the input point cloud.

Local feature extraction: as depicted in Fig. 1d, the local feature extraction component is implemented through a single specialized Set Abstraction Layer (SAL-5), with its hyper-parameters enumerated in Table 1. SAL-5 exhibits a distinctive characteristic compared to other SAL layers in that it deliberately omits the Farthest Point Sampling (FPS)[129] operation. Instead of generating a sub-sampled point set through FPS, it directly utilizes the scene flow based hand shape and pose estimate (i.e., $H_t + SF_M$) as query points and systematically computes the features in the vicinity of these anatomically meaningful points. Through this dual-feature extraction approach, both global contextual information and local geometric details are effectively captured, providing a comprehensive feature representation for the subsequent hand shape refinement process. The global features provide the structural understanding of the entire hand, while local features preserve the fine-grained details necessary for accurate articulation modeling.

GCNN-based shape refinement

The core of the shape refinement module is GCNN. The initial hand shape estimate is obtained by combining H_t and SF_M. However, this estimation of H_{t+1} inherently contains uncertainties due to the prediction errors associated with both H_t and SF_M. The primary objective of the GCNN is to address these uncertainties and eliminate slight discrepancies through alignment correction.

Graph convolutional neural networks[91–93] have demonstrated significant effectiveness in solving refinement problems, particularly in the context of 3D shape and pose estimation. Our proposed GCNN architecture takes as input the scene flow-based hand shape and pose estimate (i.e., $H_t + SF_M$), along with global and local features extracted from the feature extractors. It then outputs the final refined hand shape and pose, denoted as H_{t+1}. The architecture of the proposed graph convolutional neural network (GCNN) (see Fig. 1e) is inspired by the works of[94] and[95]. The formulation of the GCNN layer follows the methodology introduced by[96] is given in Eq. 6:

$$\mathbf{G}_{\widetilde{\mathbf{H}}} = \mathrm{F}(\hat{\mathbf{A}}\mathbf{G}_{\mathbf{H}}\mathrm{W}), \tag{6}$$

where $\mathbf{G}_{\mathbf{H}}$ represents the input hand shape or pose graph, $\mathbf{G}_{\widetilde{\mathbf{H}}}$ denotes the output refined hand shape or pose graph, $\mathrm{F}(\cdot)$ is the activation function, W corresponds to the learnable weights matrix, and $\hat{\mathbf{A}}$ is the row-normalized graph adjacency matrix.

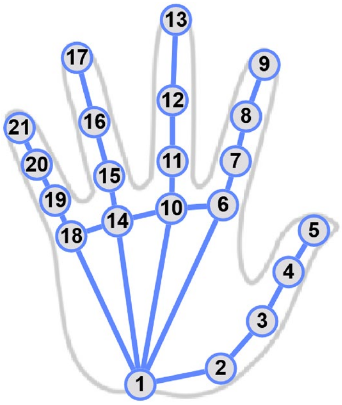


Fig. 5. Kinematic joint adjacency template. Illustration of the 21-joint hand skeleton topology employed in the pose refinement GCNN branch. Edges represent kinematic adjacency relationships as defined by the anatomical joint hierarchy, which forms the structural basis for the adjacency matrix $\hat{\mathbf{A}}$ utilized in the pose refinement network.

Layer	Input dim	Output dim	Operation
Input	3 + 1024 (global) + 1024 (local)	2051	Concatenation
GraphLinear	2051	1024	Linear
GraphResBlock 1	1024	512	Residual GCN
GraphResBlock 2–5	512	512	Residual GCN (4×)
GraphResBlock 6	512	64	Residual GCN
GraphResBlock 7	64	32	Residual GCN
GroupNorm + ReLU	32	32	Normalization + activation
GraphLinear	32	3	Linear (regress coordinates)

Table 3. GCNN architecture for shape and pose refinement. Both branches employ identical layer architectures, differing solely in their respective adjacency matrices: the shape branch utilizes the MANO mesh adjacency, while the pose branch employs the kinematic joint adjacency (illustrated in Fig. 5). Each node receives input comprising concatenated 3D coordinates, a 1024-dimensional global feature, and a 1024-dimensional local feature.

Two separate GCNNs are employed for refining the hand shape and hand pose, respectively. For the hand shape refinement, the adjacency matrix is derived from the standard MANO mesh[124]. In contrast, for the hand pose refinement, the adjacency matrix is constructed based on the skeleton definition illustrated in Fig. 5 The input graphs are constructed using the 3D locations of hand vertices and joints as graph nodes. Global and local features are appended to each node of the input graphs, which are then fed into the first graph convolutional layer of the corresponding GCNN.

The proposed architecture consists of five graph convolutional layers in both GCNNs. The first four layers have an output node feature dimension of 512, while the final layer produces the refined hand shape and pose H_{t+1}. This multi-layer design ensures a progressive refinement of the hand mesh, resulting in accurate and realistic hand shape and pose estimates. Each GCNN branch processes node-specific feature vectors constructed by concatenating 3D spatial coordinates with both global and local feature representations. Specifically, each node receives a 2051-dimensional input vector comprising 3D coordinates (3D), a global contextual feature (1024D), and a local geometric feature (1024D). The shape refinement branch operates on the adjacency matrix derived from the MANO mesh topology, while the pose refinement branch utilizes a kinematic skeleton adjacency matrix (detailed in Fig.5). Both adjacency matrices undergo row-normalization according to Eq. (6) to ensure numerical stability and convergence. The complete architectural specifications for both branches are presented in Table 3. Notably, the two branches maintain independent parameter sets to enable specialized learning for their respective tasks. This dual-branch design draws inspiration from the architectural frameworks proposed by Litany et al.[94] and Kolotouros et al.[95].

Reference shape estimation

To track 3D hand shape and pose throughout a sequence, a reference or initial hand shape and pose for the first frame is required. This initial hand shape is estimated using a reference hand shape estimator (HSE), which consists of a global feature extractor (see “Feature extractors”) and a three-layer multilayer perceptron (MLP). Initially, a global feature vector is extracted from the input point set using the global feature extractor, as illustrated in Fig. 1c. Subsequently, this global feature vector is utilized to regress the hand shape and pose through an MLP. The MLP consists of three fully connected layers, each employing a ReLU activation function and batch normalization (BN)[130]. The significance of these architectural choices is demonstrated in the ablation study (see Table 4).

HandFlowNet training

Following the generation of multi-view point sets (see "Point set representation of depth image"), the training of HandFlowNet is performed in a sequential manner. Initially, the reference hand shape estimator and the scene flow estimator are trained separately. Subsequently, the entire HandFlowNet pipeline is fine-tuned in an end-to-end fashion. Training is conducted using a single NVIDIA RTX 3090 GPU, with the Adam optimizer[131] employed for optimization.

Training of reference hand shape estimator (HSE)

The reference hand shape estimator is trained for 100 epochs using the hand shape and pose annotations provided by the DexYCB dataset[27]. A batch size of 96 is utilized, with an initial learning rate set to 0.01, which is progressively reduced by a factor of 0.314 at every 10th epoch. The loss function is defined as a weighted combination of mean squared error and mean absolute error, formulated as given in Eq. (7):

$$L = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \left[(\hat{p}_i - p_i)^2 + |\hat{p}_i - p_i| \right] \tag{7}$$

where p_i and $\hat{p}_i$ represent the ground truth and predicted locations of the i^{th} 3D point in H_t, respectively.

Prior to training, both the input point set (PC_t) and the corresponding ground truth hand shape and pose annotations (H_t) undergo mean normalization, where the mean value of the input point set ($\overline{PC_t}$) is subtracted from each point. To further normalize the scale, all points in PC_t and H_t are multiplied by a factor of 10 to ensure values lie within the range $[-1, +1]$. This normalization is performed as given in Eq. 8:

$$p_i := (p_i - \overline{PC}_t) \times 10 \tag{8}$$

where p_i denotes any 3D point location in PC_t or H_t.

Training of hand mesh scene flow estimator (SFE)

The scene flow estimator is initialized with Bi-PointFlowNet[22], which is pre-trained on the FlyingThings3D dataset[128]. For adaptation to the DexYCB dataset, fine-tuning is performed for 20 epochs with a learning rate of 0.001 and a batch size of 20. The dense scene flow annotations, generated using the method described in "Scene flow annotation", serve as the supervision signal. The loss function employed follows the mean squared error formulation as defined in[22].

Training of HandFlowNet

For training the complete HandFlowNet architecture, the weights of the reference hand shape estimator, scene flow estimator, and global feature extractor remain fixed, while the local feature extractor and the graph convolutional neural network are optimized. HandFlowNet is trained for 30 epochs with an initial learning rate of 0.001 and a batch size of 96. The loss function employed is the same as in Eq. (7).

To ensure consistency in feature representation, the input point sets (PC_t, PC_{t+1}) and the target hand shape and pose annotations (H_{t+1}) are normalized in the same manner as described earlier. Specifically, the mean of the second point set ($\overline{PC}_{t+1}$) is subtracted, followed by scaling of all values to the range $[-1, +1]$ as given in Eq. (9):

$$p_i := (p_i - \overline{PC}_{t+1}) \times 10 \tag{9}$$

where p_i represents any 3D point in PC_t, PC_{t+1}, or H_{t+1}.

Sr.	BN	Act.	Layers	GF_t Size	MPVPE (mm)
1	No	ReLU	3	1024	33.25
2	Yes	ReLU	2	1024	10.91
3	Yes	ReLU	3	1024	10.52
4	Yes	ReLU	4	1024	10.57
5	Yes	ReLU	5	1024	10.79
6	Yes	Leaky ReLU	3	1024	10.65
7	Yes	ReLU	3	2048	10.40
8	**Yes**	**ReLU**	**3**	**4096**	**9.76**
9	Yes	ReLU	3	8192	12.40

Table 4. Ablation study on the architecture of HSE.

For the HO3D dataset, fine-tuning is performed on the model pre-trained on DexYCB for an additional five epochs to adapt to the dataset characteristics.

Experiments

We conduct both qualitative and quantitative evaluations of our approach on multi-view point sets derived from the DexYCB dataset[27] and the HO3D dataset (see "Point set representation of depth image"). Additionally, a series of ablation studies are performed to validate the effectiveness of key design choices in our method.

Datasets and evaluation metrics

DexYCB dataset: DexYCB is a real-world hand dataset comprising 100 RGB-D sequences captured from eight different viewpoints. It provides annotated hand masks along with pose and shape parameters in the form of MANO representations[132]. The hand shape and pose annotations (H_t) are obtained using Manopth[133]. Following the approach of[119], only sequences containing the right hand are utilized. Furthermore, the validation and test splits of subject S0 are merged to form a single validation set. In our multi-view point set representation (see "Point set representation of depth image"), the DexYCB dataset comprises 25,387, 1,412, and 4,951 samples in the training, validation, and test splits, respectively.

HO3D dataset: we employ version 3 of the HO3D dataset, which consists of multiple RGB-D sequences captured from varying viewpoints. Similar to DexYCB, this dataset includes annotated hand masks and hand pose/shape information represented using MANO parameters[132]. The hand shape and pose annotations (H_t) are extracted using Manopth[133]. Following the methodology of[119], sequences captured from five viewpoints are used, adhering to their train/test split strategy.

In the multi-view point set representation (see "Point set representation of depth image"), the HO3D dataset comprises 9,087 and 2,706 samples in the training and test splits, respectively. Consistent with POEM[119], the test split is also used for validation, and results should be interpreted accordingly. The HO3D test split is used for validation to enable direct and fair comparison with state-of-the-art methods, ensuring consistent benchmarking across the literature. To enhance the diversity of training samples, data augmentation is performed by applying 3D rotations. Specifically, point clouds are rotated with roll and pitch angles selected from the set {0, 90, 180, 270} degrees, thereby increasing the number of training samples by a factor of 16.

Evaluation metrics: the evaluation of scene flow estimation is conducted using the 3D Endpoint Error (EPE3D, mm), which computes the average Euclidean distance between predicted and ground truth scene flow vectors for both dense scene flow SF_D and sparse hand scene flow SF_M. For hand shape estimation, the Mean Per-Vertex Position Error (MPVPE, mm) is used, measuring the average Euclidean distance between the predicted and ground truth 3D hand mesh vertices. Similarly, for hand pose estimation, the Mean Per-Joint Position Error (MPJPE, mm) is employed, which computes the average Euclidean distance between predicted and ground truth hand joints.

Additionally, root-relative versions of these metrics, MPVPE-RR (mm) and MPJPE-RR (mm), are evaluated by subtracting the root location (palm center) from both the predicted and ground truth hand shape/pose before computing MPVPE and MPJPE, respectively.

To assess temporal consistency, the acceleration error Accel-RR (mm/s²)[19] is reported. This metric quantifies the difference between the predicted and ground-truth root-relative 3D hand joint accelerations, capturing the smoothness of motion predictions.

Ablation studies

The design choices of HandFlowNet are substantiated by training its modules on a 10% subset (i.e., subject-01 only) of the multi-view DexYcb dataset, following the training methodology outlined in "HandFlowNet training". These investigations focus exclusively on hand shape recovery. Consequently, the MPVPE (mm) metric is employed for method assessment.

Reference hand shape estimator (HSE): the reference hand shape estimator is trained with various hyper-parameter configurations of a 3-layer MLP, as outlined in Table 4. Optimal performance is achieved with an MLP incorporating batch normalization and Leaky ReLU. Increasing the network depth does not yield substantial improvements; therefore, the MLP configuration consists of three fully connected layers with batch normalization and Leaky ReLU. After finalizing the MLP parameters, experiments are conducted on the global feature extractor using different global feature vector sizes (1024, 2048, 4096, and 8192). As indicated in Table 4, a 4096-dimensional global feature vector provides the best MPVPE (mm).

Hand mesh scene flow estimator (SFE): the Scene Flow Estimator is evaluated using the Bi-PointFlowNet backbone, pre-trained on the FlyingThings3D dataset, with validation performed on the multi-view validation split. The EPE3D (mm) of dense scene flow (SF_D) depends on the scale of the input point sets. To address this, a scaling factor (S) is introduced and applied to the input point sets. An empirically determined optimal value of $S = 15$ is selected. Since the Scene Flow Estimator derives sparse scene flow of hands (SF_M) from the dense scene flow (SF_D) using k-NN, the EPE3D (mm) of SF_M is influenced by the choice of k. Based on the ablation study results presented in Table 5, an optimal value of $k = 15$ is selected using the knee point rule of thumb for the k-NN algorithm.

Refinement module (RM): the effectiveness of global and local feature vectors in the refinement module (RM) is assessed through three experiments: (i) utilizing only the global feature, (ii) utilizing only the local feature, and (iii) utilizing both global and local features. As shown in Table 6, quantitative improvements are observed when both global and local feature vectors are incorporated. Additionally, superior performance is obtained by employing the pre-trained global feature extractor used in the reference hand shape estimator, rather than training a new one from scratch. Consequently, the global feature extractor weights are shared between the

k	SF_M EPE3D (mm)	k	SF_M EPE3D (mm)	k	SF_M EPE3D (mm)
1	4.33	17	4.14	40	4.05
3	4.27	19	4.13	45	4.04
5	4.24	21	4.12	50	4.03
7	4.22	23	4.11	55	4.02
9	4.20	25	4.10	60	4.01
11	4.18	27	4.09	65	4.00
13	4.17	30	4.08	70	3.99
15	**4.15**	35	4.07	75	3.98

Table 5. Ablation study on SFE for k with S $=$ 15.

Sr.	Features	MPVPE (mm)
1	GF_t only	10.97
2	LF_t only	10.75
3	GF_t only with shared *GFE*	10.75
4	**Both$GF_t$and$LF_t$with sharedGFE**	**10.50**

Table 6. Ablation study on RM. Both global and local features play a crucial role in refinement. Identical weights of the global feature extractor (*GFE*) are utilized for both PC_t and PC_{t+1}. Significant values are in bold.

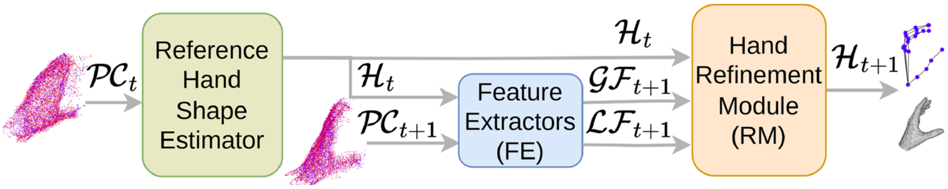


Fig. 6. Ablation study. We assess the role of scene flow by conducting an experiment without it. The initial hand estimate (H_t) is predicted using the hand shape estimator (HSE). Local $(LFt + 1)$ and global (GF_{t+1}) features are extracted by the feature extraction (FE) module, and the final hand shape H_{t+1} is predicted using the refinement module (RM).

Sr.	Model type	MPVPE	MPJPE	Accel-RR
1	Without scene flow	12.20	12.03	9.35
2	With scene flow	8.95	9.15	6.26

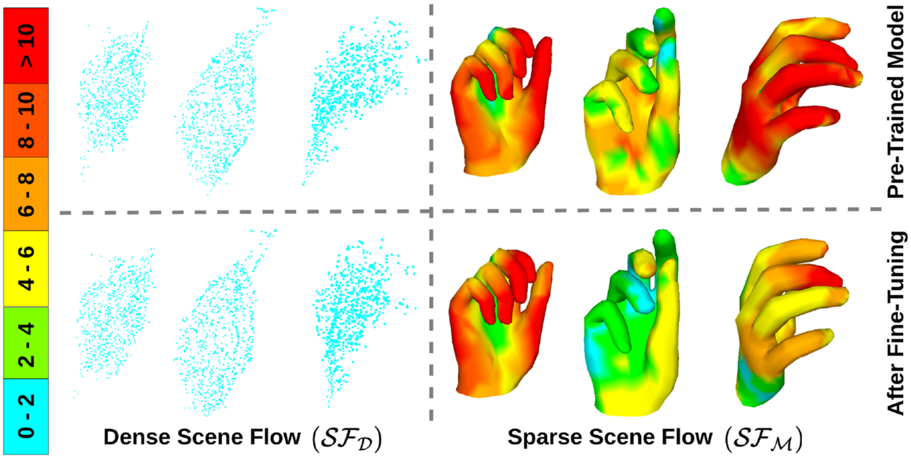
Table 7. Ablation study showing the significance of scene flow estimation on the DexYcb dataset.

reference hand shape estimator and the refinement module. For the ablation study on the Refinement Module, the global feature vector size is fixed at 1024.

Significance of scene flow: the essential role of scene flow is evaluated through an ablation study, in which the scene flow module is removed, resulting in a model that consists only of the Feature Extraction Module and the Refinement Module, as illustrated in Fig. 6. In the “Without scene flow” setting, the scene flow estimation module is omitted, and the model is retrained to predict hand shape and pose directly from the input point set of the current frame. The objective is to predict the hand shape and pose H_{t+1} based on the point set PC_{t+1} and the initial hand shape and pose H_t provided by the reference hand shape estimator. Rather than directly enhancing hand reconstruction, scene flow functions as a mechanism for tracking hand shape and pose over time by providing explicit 3D motion information between consecutive frames. This temporal information plays a crucial role in propagating hand shape and pose across frames, facilitating error correction in single-frame predictions, and maintaining consistency in reconstruction. In comparison to the proposed HandFlowNet (Fig. 1a), the results presented in Table 7 emphasize the significance of scene flow as temporal information. Furthermore, the inclusion of Accel-RR results in Table 7 demonstrates the substantial improvement in temporal stability achieved through the integration of scene flow into the pipeline.

Model weights	SF_D EPE3D (mm)	SF_M EPE3D (mm)
Pre-trained	4.51	4.62
Fine-tuned	$\mathbf{2.39}$	$\mathbf{3.04}$
Improvement	47%	34.20%

Table 8. Effectiveness of fine-tuning SFE on DexYCB. Significant values are in bold.



> 10
8 - 10
6 - 8
4 - 6
2 - 4
0 - 2
Pre-Trained Model
After Fine-Tuning
Dense Scene Flow ($\mathcal{SF_D}$)
Sparse Scene Flow ($\mathcal{SF_M}$)

Fig. 7. Scene flow error diagram. It shows the point-wise EPE3D (mm) according to the color map. Significant improvement can be observed in SF_M after fine-tuning without causing any performance degradation in SF_D estimation. This figure is best viewed in color.

Evaluation of hand scene flow estimation

The performance of the Scene Flow Estimator is evaluated on the DexYCB dataset in this subsection. The accuracy of hand shape reconstruction depends on the estimation of hand scene flow, making robust and accurate scene flow estimation essential for the proposed approach. To enhance performance, transfer learning is applied by fine-tuning the Scene Flow Estimator, which has been pre-trained on the FlyingThings3D dataset, using the multi-view training split (see “HandFlowNet training”).

Bi-PointFlowNet, pre-trained on FlyingThings3D, initially demonstrated suboptimal performance on hand datasets, achieving an EPE3D of 512.78 mm due to inherent scale disparities between domains. Domain adaptation through systematic input scaling by a factor of 15 (detailed in “Ablation studies”) substantially reduced the EPE3D to 4.51 mm. Subsequent fine-tuning on our dense scene flow annotations (described in “Scene flow annotation”) further improved performance, yielding an EPE3D of 2.39 mm.

A quantitative comparison between the pre-trained and fine-tuned Scene Flow Estimator, evaluated on the multi-view validation split, is presented in Table 8. The results indicate that fine-tuning improves the EPE3D (mm) of Scene Flow Estimator for both SF_D and SF_M by 47% and 34.20%, respectively, representing a significant margin of improvement.

Figure 7 visualizes the point-wise EPE3D (mm) of the estimated dense scene flow SF_D and sparse hand scene flow SF_M using a color map. Points with errors exceeding 10 mm are marked in red, while the remaining errors are colored according to the color scale. The qualitative results in Fig. 7 indicate that the estimation of dense scene flow SF_D is both robust and accurate, as the majority of points fall within the 0–2 mm error range. This observation aligns with the overall qualitative average EPE3D (mm) value of 2.39 mm, as reported in Table 8. Such accuracy is expected because the dense scene flow ground truth annotations are not derived from real measurements but are estimated from the sparse scene flow ground truth (see “Scene flow annotation”). Figure 7 further reveals that sparse hand scene flow SF_M estimation exhibits higher variation, with increased errors observed near the fingers. This behavior is attributed to occlusions, inaccuracies in depth measurement, and errors in mask annotations. Consequently, despite an overall qualitative average EPE3D (mm) value of 3.04 mm, scene flow errors exceeding 10 mm are expected near the fingertips. These errors may result in inaccuracies in hand pose and shape estimation by HandFlowNet. A detailed discussion on this issue is provided in “Evaluation of hand shape estimation”, where it is demonstrated that the Refinement Module effectively compensates for this error. The qualitative results in Fig. 7 further illustrate that a significant performance gain is achieved in sparse hand scene flow SF_M estimation following fine-tuning.

Evaluation of hand shape estimation

The HandFlowNet pipeline is evaluated on the DexYCB and HO3D datasets for the task of 3D hand shape estimation. The models are trained following the methodology outlined in “HandFlowNet training”. A quantitative comparison of the proposed method with the state-of-the-art (SoTA) approaches is presented in

Sr.	Method	Input data	Multiview	Temporal	MPVPE↓	MPJPE↓	MPVPE-RR↓	MPJPE-RR↓	Accel-RR↓
1	MobRecon[99]	RGB	–	–	13.05	14.20	13.10	14.20	–
2	HFL Net[86]	RGB	–	–	–	12.56	–	–	–
3	HandOccNet[85]	RGB	–	Yes	13.09	14.04	–	–	–
4	H2ONet[108]	RGB	–	Yes	12.70	13.70	–	–	–
5	VIBE[19]	RGB	–	Yes	–	16.95	–	16.95	36.40
6	Deformer[20]	RGB	–	Yes	–	–	–	13.64	31.70
7	TCMR[21]	RGB	–	Yes	–	16.03	–	16.03	34.30
8	PyMAF-X[109]	RGB	–	Yes	–	–	–	38.10	*17.10*
9	POEM[119]	RGB	Yes	–	**6.13**	**6.06**	**7.21**	**7.30**	–
10	SiMA-Hand[122]	RGB	Yes	–	12.78	13.25	–	–	–
11	DiffHand[87]	RGB+D	–	–	12.1	–	–	–	–
12	A2J[10]	D	–	–	–	23.93	–	–	–
13	IPNet[75]	D+Point	–	–	–	*8.03*	–	–	–
14	HandFlowNet (ours)	Point	Yes	Yes	*9.45*	9.67	*8.61*	*8.80*	**7.25**

Table 9. Quantitative comparison on DexYCB Dataset (best results are in bold while second best are italics).

Sr.	Method	Input data	Multiview	Temporal	MPVPE↓	MPJPE↓	MPVPE-RR↓	MPJPE-RR↓	Accel-RR↓
1	THOR-Net[11]	RGB	–	–	23.70	25.60	–	–	–
2	METRO[14]	RGB	–	–	–	–	–	38.70	17.40
3	PyMAF-X[109]	RGB	–	Yes	–	–	–	29.60	**9.30**
4	HandOccNet[85]	RGB	–	Yes	–	–	–	24.9	–
5	POEM[119]	RGB	Yes	–	17.20	17.28	21.45	21.94	–
6	MLP Hand[123]	RGB	Yes	–	18.69	18.70	23.28	23.76	–
7	ShapeGraFormer[134]	D	–	–	19.40	19.90	–	–	–
8	HandFlowNet (ours)	Point	Yes	Yes	**13.79**	**14.41**	**12.73**	**13.35**	*9.65*

Table 10. Quantitative comparison on HO3D Dataset (best results are in bold, while second best are italics).

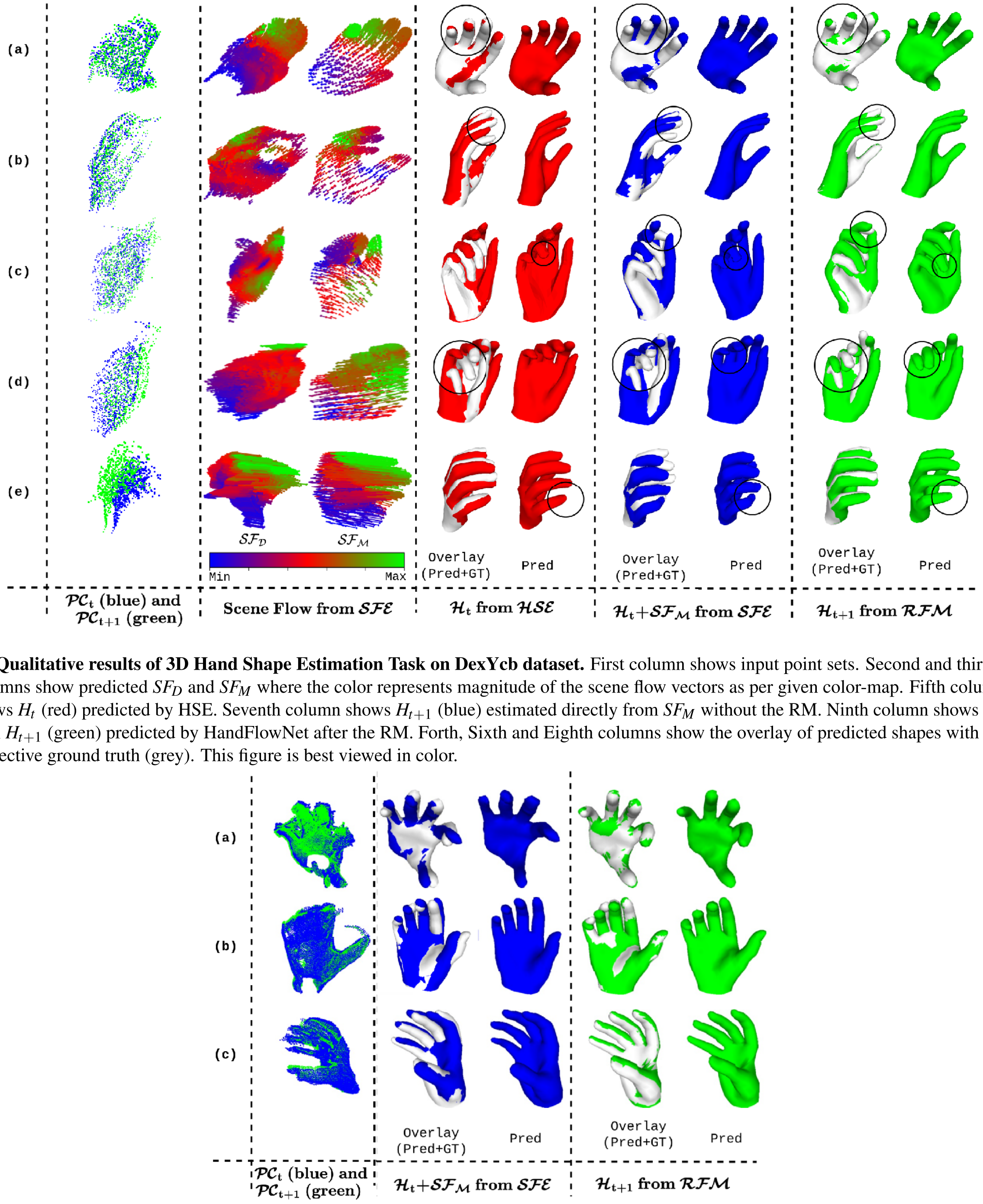
Tables 9 and 10. The Mean Per-Vertex Position Error (MPVPE (mm)) values demonstrate that the HandFlowNet approach achieves competitive performance in hand shape estimation on both DexYCB and HO3D benchmarks.

The statistical distribution of MPVPE across the DexYCB dataset is illustrated in Fig. 12. The MPVPE exhibits a standard deviation of 6.18 mm and variance of 38.21 mm on the test partition.

Figure 8aillustrates the quality of hand shapes reconstructed by the reference hand shape estimator and HandFlowNet for five selected frames from DexYCB. The first column presents input point sets (PC_t in blue and PC_{t+1} in green). The second and third columns visualize the scene flow vectors of dense scene flow (SF_D) and sparse hand scene flow (SF_M), with colors representing the magnitudes of scene flow vectors according to the provided color map.

Columns four and five display the outputs of the reference hand shape estimator, where the predicted hand shape is shown in red in column five, while column four overlays the reference hand shape estimator output on the ground truth. Similarly, the hand shape predicted by HandFlowNet is presented in green in column nine, whereas column eight overlays the HandFlowNet output on the ground truth. The effectiveness of the Scene Flow Estimator in estimating the hand shape (H_{t+1}) without the Refinement Module is demonstrated in the sixth and seventh columns. The seventh column shows the scene flow-based hand shape ($H_{t+1} = H_t + SF_M$) in blue, while the eighth column presents the ground truth overlaid on this estimate.

Since the initial hand shape and scene flow are estimated through deep neural networks, they remain susceptible to errors. However, the HandFlowNet pipeline effectively mitigates these errors from the reference hand shape estimator (HSE) and the scene flow estimator (SFE) to produce accurate hand shapes. The samples in Fig. 8a illustrate this capability. Figure 8a(b), a(d) depict cases where deviations in the initial hand shape reconstructed by HSE are passed to the input of the Refinement Module (RM). Despite these deviations, RM successfully corrects the errors, producing accurate outputs. Figure 8a(c), a(e) present cases where variations in the initial hand shape (red) due to HSE, combined with additional deformations and artifacts from SFE errors (blue), are observed. In particular, Fig. 8a(e) highlights a case involving significant hand movement, as indicated by the large scene flow vector magnitudes. Despite this challenging scenario, the HandFlowNet pipeline accurately predicts the hand shape and pose with high precision. The final outputs indicate that RM effectively eliminates these errors, resulting in a smooth hand shape. Additionally, Fig. 8a(a) presents a case where the initial hand shape is accurate, but translation errors and artifacts from SFE remain present. RM successfully addresses these errors as well.


(a)
(b)
(c)
(d)
(e)
$\mathcal{SF}_\mathcal{D}$
$\mathcal{SF}_\mathcal{M}$
Min
Max
Overlay (Pred+GT)
Pred
Overlay (Pred+GT)
Pred
Overlay (Pred+GT)
Pred
$\mathcal{PC}_t$ (blue) and $\mathcal{PC}_{t+1}$ (green)
Scene Flow from $\mathcal{SFE}$
$\mathcal{H}_t$ from $\mathcal{HSE}$
$\mathcal{H}_t$+$\mathcal{SF}_\mathcal{M}$ from $\mathcal{SFE}$
$\mathcal{H}_{t+1}$ from $\mathcal{RFM}$
(a)
(b)
(c)
Overlay (Pred+GT)
Pred
Overlay (Pred+GT)
Pred
$\mathcal{PC}_t$ (blue) and $\mathcal{PC}_{t+1}$ (green)
$\mathcal{H}_t$+$\mathcal{SF}_\mathcal{M}$ from $\mathcal{SFE}$
$\mathcal{H}_{t+1}$ from $\mathcal{RFM}$

(a) Qualitative results of 3D Hand Shape Estimation Task on DexYcb dataset. First column shows input point sets. Second and third columns show predicted SF_D and SF_M where the color represents magnitude of the scene flow vectors as per given color-map. Fifth column shows H_t (red) predicted by HSE. Seventh column shows H_{t+1} (blue) estimated directly from SF_M without the RM. Ninth column shows the final H_{t+1} (green) predicted by HandFlowNet after the RM. Forth, Sixth and Eighth columns show the overlay of predicted shapes with respective ground truth (grey). This figure is best viewed in color.

(b) Qualitative results on 3D Hand Shape Estimation Task on HO3D dataset. First column shows input point sets. Third column shows H_{t+1} (blue) estimated directly from SF_M without the RM. Fifth column shows the final H_{t+1} (green) predicted by HandFlowNet after the RM. Second and Fourth columns show the overlay of predicted shapes with respective ground truth (grey).

Fig. 8. Qualitative results of 3D hand shape estimation task.

The qualitative performance shown in Fig. 8a demonstrates the robustness of the proposed approach. Rather than directly enhancing hand reconstruction, the approach utilizes scene flow to ensure temporal consistency by tracking hand shape and pose across frames. Sparse scene flow estimates the 3D motion of hand mesh vertices and joints, enabling error correction in single-frame predictions through temporal propagation. For instance, as depicted in Fig. 8a(d), the Refinement Module corrects inaccuracies, such as the pinky finger's position, by integrating scene flow with features from the current frame. Even in challenging scenarios where both HSE and SFE introduce errors, as shown in Fig. 8a(d), the Refinement Module demonstrates its robustness by effectively addressing these inaccuracies. Similarly, Fig. 9 illustrates the effectiveness of our method for 3D hand pose

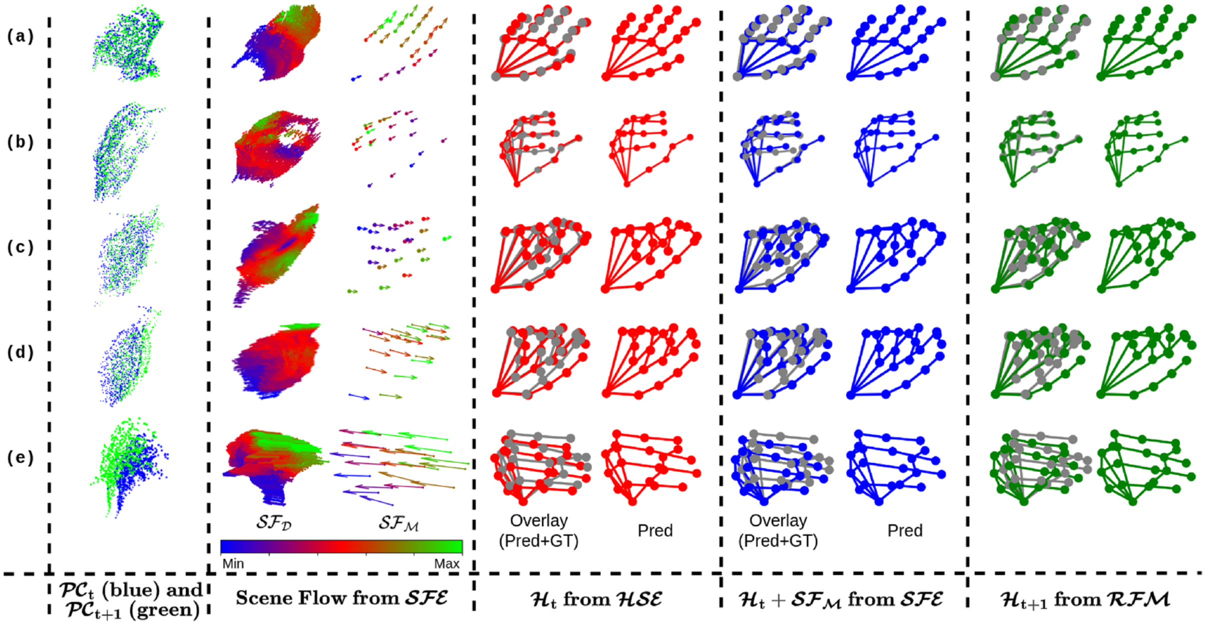

(a)
(b)
(c)
(d)
(e)
$\mathcal{SF}_\mathcal{D}$
$\mathcal{SF}_\mathcal{M}$
Min
Max
Overlay (Pred+GT)
Pred
Overlay (Pred+GT)
Pred
$\mathcal{PC}_t$ (blue) and $\mathcal{PC}_{t+1}$ (green)
Scene Flow from $\mathcal{SFE}$
$\mathcal{H}_t$ from $\mathcal{HSE}$
$\mathcal{H}_t + \mathcal{SF}_\mathcal{M}$ from $\mathcal{SFE}$
$\mathcal{H}_{t+1}$ from $\mathcal{RFM}$

Fig. 9. Qualitative results of 3D hand pose estimation task on DexYcb dataset. First column shows input point sets. Second and third columns show predicted SF_D and SF_M where the color represents magnitude of the scene flow vectors as per given color-map. Fifth column shows H_t (red) predicted by HSE. Seventh column shows H_{t+1} (blue) estimated directly from SF_M without the RM. Ninth column shows the final H_{t+1} (green) predicted by HandFlowNet after the RM. Fourth, Sixth and Eighth columns show the overlay of predicted shapes with respective ground truth (grey). This figure is best viewed in color.

estimation. As shown in Fig. 9, our HandFlowNet accurately estimates joint positions across various hand poses and interactions. Rows (a) through (e) represent different input scenarios, showcasing the model's ability to handle diverse hand configurations. For example, in challenging cases like row (c) with partially visible fingers and row (e) with more complex articulations, the Refinement Module significantly improves the final joint predictions (green) compared to those without refinement (blue), bringing them closer to ground truth. The scene flow visualization (SF_D and SF_M in columns 2-3) demonstrates how our approach effectively captures the motion patterns that guide accurate pose estimation.

The results indicate that the HandFlowNet pipeline generates accurate hand shapes utilizing the estimated scene flow and the refinement network. Fig. 8b further demonstrates similar qualitative performance on sample frames from the HO3D dataset.

Evaluation of hand pose estimation

In this subsection, the HandFlowNet approach is evaluated on the DexYCB and HO3D datasets for the 3D hand pose estimation task. The models are trained following the methodology outlined in "HandFlowNet training". The quantitative results of the proposed method, in comparison with state-of-the-art (SoTA) approaches, are presented in Tables 9 and 10. The Mean Per-Joint Position Error (MPJPE (mm)) values demonstrate that the HandFlowNet method achieves competitive performance relative to SoTA techniques.

Figure 10b illustrates the percentage of samples within varying error thresholds, compared against depth-based methods, specifically IPNet [75] and [10]. Our approach achieves the lowest MPJPE across nearly all thresholds. Additionally, Fig. 10a depicts the error distribution across individual finger joints, which is analyzed in greater detail in "Discussion".

Furthermore, the statistical distribution of MPJPE across the DexYCB dataset is illustrated in Fig. 12. The MPJPE exhibits a standard deviation of 6.24 mm and variance of 38.93 mm^2 on the test partition.

The visualization of 3D hand poses estimated by the reference hand shape estimator and HandFlowNet for sample frames on DexYcb is shown in Fig. 8a. The first column presents the input point sets. The second and third columns illustrate the estimated dense scene flow (SF_D) and sparse hand scene flow (SF_M) using vector representations. Columns four and five depict the estimations from the reference hand shape estimator. The hand pose predicted by the reference hand shape estimator is displayed in red in column five, while column four presents the estimator's output overlaid on the ground truth. Similarly, the hand pose predicted by HandFlowNet is displayed in green in column nine, while column eight overlays the HandFlowNet output on the ground truth. The sixth and seventh columns highlight the effectiveness of the Scene Flow Estimator in estimating hand pose without the Refinement Module. In the seventh column, the scene flow-based hand pose ($H_{t+1} = H_t + SF_M$) is shown in blue, while the eighth column overlays the ground truth on this estimate. The qualitative results illustrated in Fig. 8a indicate that the proposed approach effectively recovers accurate 3D hand poses by utilizing hand joint scene flow estimations.

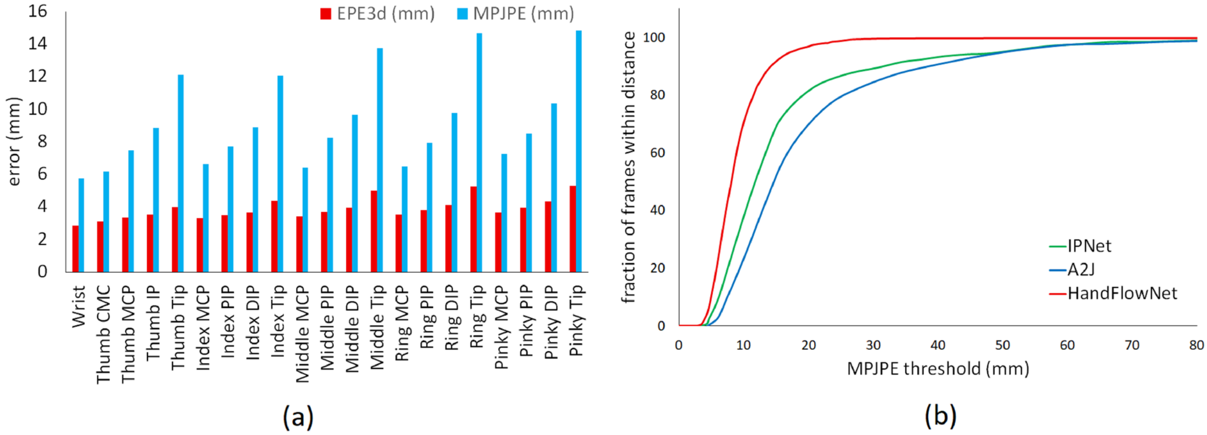


Fig. 10. (**a**) Joint-wise error distribution on DexYcb dataset. (**b**) Distribution of samples across error thresholds on the DexYCB dataset..

Module	Inference time (ms)	Parameters (M)
Hand shape estimator (HSE)	5	37
Scene flow estimator (SFE)	19	8
Refinement module (RFM)	15	19
HandFlowNet (Total)	39	64

Table 11. Runtime performance (batch size 1) and parameter count for each module of HandFlowNet.

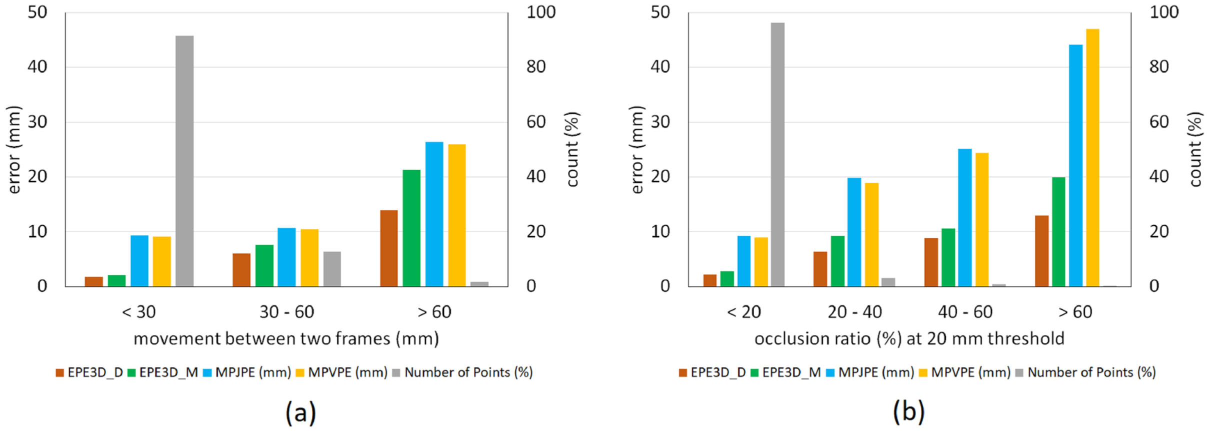


Fig. 11. Error distribution on DexYcb dataset across different ranges of (**a**) hand movement (**b**) input point cloud occlusion ratios.

Runtime and model parameters

The proposed HandFlowNet processes a single forward pass in 39 ms with a batch size of 1 on an NVIDIA RTX 3090 GPU. This efficiency enables real-time deployment, achieving a throughput of 25 frames per second (FPS). The model architecture consists of 64 million parameters. Individually, HSE, SFE, and RFM comprise 37, 8, and 19 million parameters, respectively, with inference times of 5, 19, and 15 ms per frame (batch size 1) on identical hardware as specified in Table 11 (Fig. 11).

Discussion

The quantitative results presented in Tables 9 and 10, coupled with the qualitative findings illustrated in Figs.8,9, and 10b, demonstrate the competitive performance of HandFlowNet-a novel scene flow-based hand tracking methodology.

HandFlowNet utilizes an estimated initial hand shape and dense scene flow to predict 3D hand configurations in subsequent frames. Consequently, the overall system accuracy is contingent upon the precision of both

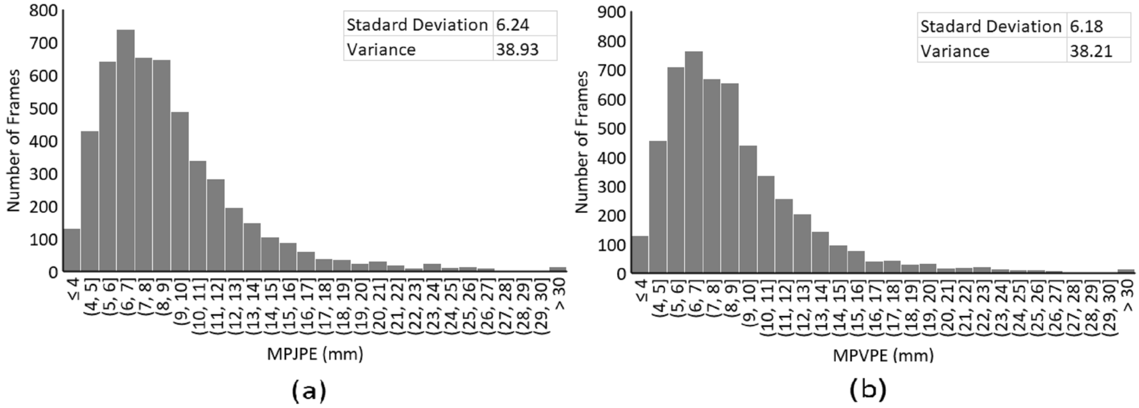


Fig. 12. Statistical distribution (histogram), standard deviation and variance of errors on DexYcb dataset (**a**) MPJPE (**b**) MPVPE.

estimation components. We introduce a comprehensive strategy for generating scene flow annotations, facilitating the optimization of scene flow estimation methods specifically tailored to hand dynamics. As evidenced in Table 8, this domain-specific optimization proves crucial for achieving competitive performance. The results reported in Tables 9 and 10 are based on initial hand shape estimates from a reference estimator. Further performance gains remain achievable through the integration of more sophisticated hand shape estimation networks or the incorporation of hand calibration frames. Nevertheless, we acknowledge that the effectiveness of our GCNN-based refinement module becomes limited when initial errors from either the hand shape or scene flow estimators are substantial, particularly under conditions of severe occlusion or rapid hand motion. To address this limitation, future work could investigate multi-stage GCNN refinement strategies to enhance robustness against error propagation and further improve tracking accuracy.

While video-level and recurrent models such as VIBE [19] and TCMR [21] employ sophisticated temporal priors through recurrent or transformer architectures, they typically rely on parametric models optimized for global motion patterns. In contrast, our scene flow-based approach provides direct, dense temporal correspondence, making it particularly well-suited for capturing the fine-grained, highly articulated motions characteristic of hand dynamics without dependence on parametric model constraints. Future research could explore the integration of dense correspondence with recurrent temporal modeling to further enhance performance.

We conducted a comprehensive analysis to assess the impact of rapid movements on both the scene flow estimator and HandFlowNet, as illustrated in Fig. 10a. The analysis categorizes samples into three average point displacement bins (< 30 mm, 30–60 mm, > 60 mm) between consecutive frames. The primary y-axis displays EPE3D_D, EPE3D_M, MPJPE, and MPVPE error metrics per bin, while the secondary y-axis indicates the percentage of test samples within each bin. Scene flow estimator errors (EPE3D_D, EPE3D_M) exhibit increasing trends with displacement magnitude, particularly for rapid movements ((>60 mm)60 mm). Notably, HandFlowNet errors (MPJPE, MPVPE) remain stable across the first two bins, indicating that the GCNN-based refinement module effectively mitigates propagated errors. However, for rapid movements, errors increase proportionally with scene flow estimation errors.

To analyze the effect of occlusions systematically, we examined performance across four occlusion ratio bins (< 20%, 20–40%, 40–60%, > 80%), as presented in Fig. 10b. The occlusion ratio represents the percentage of hand mesh ground truth vertices lacking a neighboring input point cloud point within 20 mm. Both the scene flow estimator and HandFlowNet exhibit performance degradation proportional to the occlusion percentage, highlighting the fundamental challenge posed by incomplete observations.

We further evaluated error distribution across different hand components by analyzing mean errors (EPE3D_M, MPJPE) against each joint location. Performance degradation is most pronounced near the fingertips, as also observed qualitatively in Fig. 7. This degradation is likely attributable to occlusions near the fingertips, stemming from depth camera limitations in capturing fine-scale structures and the inherent geometric complexity of fingertip regions.

Collectively, these analyses (Figs.10 and7) indicate that the scene flow estimator's performance is fundamentally constrained in frames with rapid movement or heavy occlusion, which consequently limits HandFlowNet's overall accuracy.

As discussed in "Dense scene flow estimation", the selection of a point cloud-based network for dense scene flow estimation in HandFlowNet is motivated by the pipeline's multiview fusion strategy, which operates at the point cloud level using camera projection matrices. While Bi-PointFlowNet serves as an effective foundation, significant potential exists for improvement through the integration of advanced scene flow models such as RAFT-3D[47] or FlowStep3D [56]. Future research may explore these alternatives to enhance the robustness and performance of HandFlowNet, particularly under challenging conditions of rapid movement and heavy occlusion.

The current scene flow estimation network could be further adapted to hand-specific tasks by incorporating geometric or kinematic information into the model architecture. Additionally, the refinement module may benefit from a pyramid architecture for feature aggregation and multi-stage GCNN refinement strategies.

For consistency and fair comparison with state-of-the-art methods such as POEM [119], we utilized only right-hand samples. However, HandFlowNet can be readily extended to both hands by incorporating a right/left hand detector. For left-hand cases, a two-axis mirroring operation can convert the left hand to a right-hand representation prior to processing, with the output mirrored back post-processing to restore the original configuration.

It is important to note that the DBSCAN-based artifact removal step presented in " Point set representation of depth image" can present a vulnerability under heavy occlusion conditions, potentially removing valid hand regions along with outliers. This reduction in input point set quality underscores the need for more robust preprocessing methods that better preserve essential hand structures under challenging conditions.

Finally, we evaluated the proposed pipeline on single-view point sets, obtaining MPVPE and MPJPE values of 15.55 mm and 15.84 mm, respectively, on the DexYCB dataset. The observed performance limitation, compared to the multiview case in Table 9, is attributable to substantial occlusions, geometric inconsistencies in the 3D point cloud under single-view conditions, and the limitations of DBSCAN-based preprocessing. Given that this study primarily addresses the multiview scenario, extending the scene flow-based technique to single-view settings through more adaptive algorithms and enhanced data fusion strategies represents a promising avenue for future research.

Conclusion and future work

In this work, a novel HandFlowNet method has been proposed for the reconstruction of 3D hand shape and pose from point sets by effectively leveraging scene flow information inherently present in an image sequence. The proposed architecture is designed using neural transformers and graph convolutional networks. A new approach has been developed for generating annotations of both dense and sparse hand scene flow, utilizing the available hand shape annotations in hand datasets. Experimental evaluations demonstrate that hand scene flow can be accurately estimated from a pair of consecutive image frames. The estimated scene flow, along with the extracted features from the subsequent frame, can be effectively processed within a graph neural network to generate an accurate reconstruction of the hand shape in the next frame. The results achieved on the DexYCB benchmark confirm that the proposed approach establishes a new state-of-the-art for 3D hand pose and shape reconstruction.

While our method demonstrates competitive performance, several challenges persist. As previously analyzed, HandFlowNet's accuracy is constrained by rapid hand movements and severe occlusions, particularly in single-view scenarios where incomplete observations and aggressive artifact removal algorithms can compromise input data quality. Performance degradation is most pronounced in fingertip regions, likely attributable to inherent sensor limitations in capturing fine-scale structures and the prevalence of occlusions in these anatomically complex areas.

Although graph neural networks inherently capture the structural properties of hand shape, the explicit integration of a parametric hand model within the pipeline could be explored as a potential future direction. Such an enhancement may contribute to the generation of more stable and refined hand shapes and poses. Furthermore, the development of a multi-frame scene flow estimation network could further improve not only the accuracy of hand scene flow estimation but also the precision of reconstructed hand shapes and poses.

Future research will investigate the integration of advanced scene flow models, incorporation of geometric or kinematic priors, and enhancement of the refinement module through pyramid and multi-stage architectures. Developing robust preprocessing methodologies to better address occlusions and outliers, alongside adaptive algorithms for single-view scenarios, represents promising research directions. Additionally, fusing dense correspondence with recurrent or transformer-based temporal modeling could further enhance the capture of fine-grained, highly articulated hand dynamics.

Data availability

The datasets utilized in this study, namely the DexYCB dataset and the HO3D dataset, are publicly accessible through their respective platforms. The DexYCB dataset is available at https://dex-ycb.github.io/, while the HO3D dataset can be accessed via https://paperswithcode.com/dataset/ho-3d. To ensure reproducibility and facilitate future research, the implementation code will be made publicly available on GitHub upon manuscript acceptance.

Received: 23 April 2025; Accepted: 19 August 2025
Published online: 24 September 2025

References

1. Fan, Z. et al. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12943–12954. https://doi.org/10.1109/CVPR52729.2023.01244 (2023).
2. Rastgoo, R., Kiani, K., Escalera, S. & Sabokrou, M. Sign language production: A review. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 3446–3456. https://doi.org/10.1109/CVPRW53098.2021.00384 (2021).
3. Brahmbhatt, S., Tang, C., Twigg, C. D., Kemp, C. C. & Hays, J. Contactpose: A dataset of grasps with object contact and hand pose. In *European Conference on Computer Vision (ECCV)* (2020).
4. Zhou, Y. et al. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).
5. Chen, X., Wang, B. & Shum, H.-Y. Hand avatar: Free-pose hand animation and rendering from monocular video. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8683–8693. https://doi.org/10.1109/CVPR52729.2023.00839 (2023).

6. Malik, J. et al. Deepairsig: End-to-end deep learning based in-air signature verification. *IEEE Access* **8**, 195832–195843. https://doi.org/10.1109/ACCESS.2020.3033848 (2020).
7. Karunratanakul, K., Prokudin, S., Hilliges, O. & Tang, S. Harp: Personalized hand reconstruction from a monocular rgb video. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12802–12813. https://doi.org/10.1109/CVPR52729.2023.01231 (2023).
8. Wang, C., Zhu, F. & Wen, S. Memahand: Exploiting mesh-mano interaction for single image two-hand reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
9. Moon, G., Yong Chang, J. & Mu Lee, K. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
10. Jiang, C. *et al.* A2j-transformer: Anchor-to-joint transformer network for 3D interacting hand pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
11. Aboukhadra, A. T., Malik, J., Elhayek, A., Robertini, N. & Stricker, D. Thor-net: End-to-end graformer-based realistic two hands and object reconstruction with self-supervision. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 1001–1010. https://doi.org/10.1109/WACV56688.2023.00106 (2023).
12. Chen, Z., Hasson, Y., Schmid, C. & Laptev, I. Alignsdf: Pose-aligned signed distance fields for hand-object reconstruction. In *European Conference on Computer Vision (ECCV)* (2022).
13. Tse, T. H. E., Kim, K. I., Leonardis, A. & Chang, H. J. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).
14. Lin, K., Wang, L. & Liu, Z. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).
15. Wen, Y. *et al.* Hierarchical temporal transformer for 3d hand pose estimation and action recognition from egocentric rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
16. Cai, Y. et al. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).
17. Chen, L., Lin, S.-Y., Xie, Y., Lin, Y.-Y. & Xie, X. Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).
18. Tu, Z. et al. Consistent 3D hand reconstruction in video via self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 9469–9485. https://doi.org/10.1109/TPAMI.2023.3247907 (2023).
19. Kocabas, M., Athanasiou, N. & Black, M. J. Vibe: Video inference for human body pose and shape estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5252–5262. https://doi.org/10.1109/CVPR42600.2020.00530 (2020).
20. Fu, Q., Liu, X., Xu, R., Niebles, J. C. & Kitani, K. M. Deformer: Dynamic fusion transformer for robust hand pose estimation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 23543–23554. https://doi.org/10.1109/ICCV51070.2023.02157 (2023).
21. Choi, H., Moon, G., Chang, J. Y. & Lee, K. M. Beyond static features for temporally consistent 3d human pose and shape from a video. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1964–1973. https://doi.org/10.1109/CVPR46437.2021.00200 (2021).
22. Cheng, W. & Ko, J. H. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *European Conference on Computer Vision (ECCV)* (2022).
23. Wang, G. et al. What matters for 3D scene flow network. In *European Conference on Computer Vision (ECCV)* (2022).
24. Li, R., Lin, G., He, T., Liu, F. & Shen, C. Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).
25. Ge, L., Ren, Z. & Yuan, J. Point-to-point regression pointnet for 3D hand pose estimation. In *European Conference on Computer Vision (ECCV)* (2018).
26. Cheng, W. & Ko, J. H. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *European Conference on Computer Vision (ECCV)* (2022).
27. Chao, Y.-W. et al. Dexycb: A benchmark for capturing hand grasping of objects (2021). arXiv:2104.04631.
28. Hampali, S., Rad, M., Oberweger, M. & Lepetit, V. Honnotate: A method for 3D annotation of hand and object poses (2020). arXiv:1907.01481.
29. Qi, C. R., Su, H., Mo, K. & Guibas, L. J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
30. Qi, C. R., Yi, L., Su, H. & Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Conference on Neural Information Processing Systems (NeurIPS)* (2017).
31. Zhao, H., Jiang, L., Jia, J., Torr, P. H. & Koltun, V. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).
32. Guo, M.-H. et al. Pct: Point cloud transformer. *Comput. Visual Med.* **7**, 187–199 (2021).
33. Vedula, S., Rander, P., Collins, R. & Kanade, T. Three-dimensional scene flow. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 475–480. https://doi.org/10.1109/TPAMI.2005.63 (2005).
34. Huguet, F. & Devernay, F. A variational method for scene flow estimation from stereo sequences. *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)* https://doi.org/10.1109/ICCV.2007.4409000 (2007).
35. Valgaerts, L. et al. Joint estimation of motion, structure and geometry from stereo sequences. *Eur. Conf. Comput. Vis. (ECCV)* https://doi.org/10.1007/978-3-642-15561-1_41 (2010).
36. Pons, J.-P., Keriven, R. & Faugeras, O. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *Int. J. Comput. Vis. (IJCV)* **72**, 179–193. https://doi.org/10.1007/s11263-006-8671-5 (2007).
37. Wedel, A. et al. Efficient dense scene flow from sparse or dense stereo data. *Eur. Conf. Comput. Vis. (ECCV)* https://doi.org/10.1007/978-3-540-88682-2_56 (2008).
38. Čech, J., Sanchez-Riera, J. & Horaud, R. Scene flow estimation by growing correspondence seeds. *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)* https://doi.org/10.1109/CVPR.2011.5995442 (2011).
39. Menze, M. & Geiger, A. Object scene flow for autonomous vehicles. *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)* https://doi.org/10.1109/CVPR.2015.7298925 (2015).
40. Mayer, N. et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)* https://doi.org/10.1109/CVPR.2016.438 (2016).
41. Vogel, C., Schindler, K. & Roth, S. Piecewise rigid scene flow. *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)* https://doi.org/10.1109/ICCV.2013.174 (2013).
42. Vogel, C., Schindler, K. & Roth, S. 3D scene flow estimation with a piecewise rigid scene model. *Int. J. Comput. Vis. (IJCV)* **115**, 1–28. https://doi.org/10.1007/s11263-015-0806-0 (2015).
43. Ma, W.-C., Wang, S., Hu, R., Xiong, Y. & Urtasun, R. Deep rigid instance scene flow. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3609–3617. https://doi.org/10.1109/CVPR.2019.00373 (2019).
44. Hadfield, S. & Bowden, R. Kinecting the dots: Particle based scene flow from depth sensors. In *2011 International Conference on Computer Vision*. 2290–2295. https://doi.org/10.1109/ICCV.2011.6126509 (2011).

45. Herbst, E., Ren, X. & Fox, D. Rgb-d flow: Dense 3-D motion estimation using color and depth. In *2013 IEEE International Conference on Robotics and Automation*. 2276–2282. https://doi.org/10.1109/ICRA.2013.6630885 (2013).
46. Jaimez, M., Souiai, M., Gonzalez-Jimenez, J. & Cremers, D. A primal-dual framework for real-time dense rgb-d scene flow. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 98–104. https://doi.org/10.1109/ICRA.2015.7138986 (2015).
47. Teed, Z. & Deng, J. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).
48. Liu, H. et al. Camliflow: Bidirectional camera-lidar fusion for joint optical flow and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).
49. Peng, C. et al. Delflow: Dense efficient learning of scene flow for large-scale point clouds. arXiv preprint arXiv:2308.04383 (2023).
50. Hur, J. & Roth, S. Self-supervised monocular scene flow estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7394–7403. https://doi.org/10.1109/CVPR42600.2020.00742 (2020).
51. Hur, J. & Roth, S. Self-supervised multi-frame monocular scene flow. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2683–2693. https://doi.org/10.1109/CVPR46437.2021.00271 (2021).
52. Yang, G. & Ramanan, D. Upgrading optical flow to 3d scene flow through optical expansion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1331–1340. https://doi.org/10.1109/CVPR42600.2020.00141 (2020).
53. Yang, G. & Ramanan, D. Learning to segment rigid motions from two frames. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1266–1275. https://doi.org/10.1109/CVPR46437.2021.00132 (2021).
54. Wang, G., Tian, X., Ding, R. & Wang, H. Unsupervised learning of 3D scene flow from monocular camera. In *IEEE International Conference on Robotics and Automation (ICRA)* (2021).
55. Wu, W., Qi, Z. & Fuxin, L. Pointconv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
56. Kittenplon, Y., Eldar, Y. C. & Raviv, D. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).
57. Puy, G., Boulch, A. & Marlet, R. Flot: Scene flow on point clouds guided by optimal transport. In *European Conference on Computer Vision (ECCV)* (2020).
58. Liu, X., Qi, C. R. & Guibas, L. J. Flownet3d: Learning scene flow in 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
59. Gu, X., Wang, Y., Wu, C., Lee, Y. J. & Wang, P. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
60. Su, H. *et al.* Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
61. Jampani, V., Kiefel, M. & Gehler, P. V. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
62. Wu, W., Wang, Z., Li, Z., Liu, W. & Fuxin, L. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. arXiv preprint arXiv:1911.12408 (2019).
63. Sridhar, S. et al. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision (ECCV)* (2016).
64. Guo, H. et al. Region ensemble network: Improving convolutional network for hand pose estimation. In *IEEE International Conference on Image Processing (ICIP)* (2017).
65. Mueller, F. et al. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2017).
66. Sinha, A., Unmesh, A., Huang, Q. & Ramani, K. Surfnet: Generating 3D shape surfaces using deep residual networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
67. Yuan, S. *et al.* Depth-based 3D hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
68. Wang, G., Chen, X., Guo, H. & Zhang, C. Region ensemble network: Towards good practices for deep 3d hand pose estimation. *J. Vis. Commun. Image Represent.* **55**, 404–414. https://doi.org/10.1016/j.jvcir.2018.04.005 (2018).
69. Malik, J.et al. Deephps: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth. In *International Conference on 3D Vision (3DV)* (2018).
70. Malik, J., Elhayek, A., Nunnari, F. & Stricker, D. Simple and effective deep hand shape and pose regression from a single depth image. *Comput. Graph.* **85**, 85–91 (2019).
71. Malik, J., Elhayek, A. & Stricker, D. Whsp-net: A weakly-supervised approach for 3D hand shape and pose recovery from a single depth image. *Sensors* **19**, 3784 (2019).
72. Mueller, F. et al. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Trans. Graph. (ToG)* **38**, 1–13. https://doi.org/10.1145/3306346.3322958 (2019).
73. Boukhayma, A., Bem, R. D. & Torr, P. H. 3D hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
74. Deng, X. et al. Weakly supervised learning for single depth-based hand shape recovery. *IEEE Trans. Image Process.* **30**, 532–545. https://doi.org/10.1109/TIP.2020.3037479 (2021).
75. Ren, P. et al. Two heads are better than one: Image-point cloud network for depth-based 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2023).
76. Ge, L., Liang, H., Yuan, J. & Thalmann, D. 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
77. Moon, G., Chang, J. Y. & Lee, K. M. V2v-posenet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
78. Malik, J. et al. Handvoxnet: Deep voxel-based network for 3D hand shape and pose estimation from a single depth map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).
79. Malik, J. et al. Handvoxnet++: 3D hand shape and pose estimation using voxel-based neural networks. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **44**, 8962–8974 (2021).
80. Ge, L., Cai, Y., Weng, J. & Yuan, J. Hand pointnet: 3D hand pose estimation using point sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
81. Chen, X., Wang, G., Zhang, C., Kim, T.-K. & Ji, X. Shpr-net: Deep semantic hand pose regression from point clouds. *IEEE Access* **6**, 43425–43439 (2018).
82. Li, S. & Lee, D. 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 (CVPR)* (2019).
83. Ravanbakhsh, S., Schneider, J. & Poczos, B. Deep learning with sets and point clouds. arXiv preprint arXiv:1611.04500 (2016).
84. Cheng, W., Park, J. H. & Ko, J. H. Handfoldingnet: A 3D hand pose estimation network using multiscale-feature guided folding of a 2D hand skeleton. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).
85. Park, J., Oh, Y., Moon, G., Choi, H. & Lee, K. M. Handoccnet: Occlusion-robust 3D hand mesh estimation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

86. Lin, Z., Ding, C., Yao, H., Kuang, Z. & Huang, S. Harmonious feature learning for interactive hand-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
87. Li, L., Zhuo, L., Zhang, B., Bo, L. & Chen, C. Diffhand: End-to-end hand mesh reconstruction via diffusion models. arXiv preprint arXiv:2305.13705 (2023).
88. Xiong, F. et al. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).
89. Huang, W., Ren, P., Wang, J., Qi, Q. & Sun, H. AWR: Adaptive weighting regression for 3D hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020).
90. Cheng, J. et al. Efficient virtual view selection for 3D hand pose estimation. *Proc. AAAI Conf. Artif. Intell.* https://doi.org/10.1609/aaai.v36i1.19919 (2022).
91. Kolotouros, N., Pavlakos, G. & Daniilidis, K. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
92. Ranjan, A., Bolkart, T., Sanyal, S. & Black, M. J. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)* (2018).
93. Cheng, S., Tzimiropoulos, G., Shen, J. & Pantic, M. Faster, better and more detailed: 3D face reconstruction with graph convolutional networks. In *Asian Conference on Computer Vision (ACCV)* (2020).
94. Litany, O., Bronstein, A., Bronstein, M. & Makadia, A. Deformable shape completion with graph convolutional autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
95. Kolotouros, N., Pavlakos, G. & Daniilidis, K. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
96. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *Proceedings of the International Conference on Learning Representations (ICLR)* (2017).
97. Mueller, F. et al. Ganerated hands for real-time 3D hand tracking from monocular rgb. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
98. Hasson, Y., Varol, G., Schmid, C. & Laptev, I. Towards unconstrained joint hand-object reconstruction from rgb videos. In *2021 International Conference on 3D Vision (3DV)*. 659–668. https://doi.org/10.1109/3DV53792.2021.00075 (2021).
99. Chen, X. et al. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).
100. Tekin, B., Bogo, F. & Pollefeys, M. H+ o: Unified egocentric recognition of 3D hand-object poses and interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
101. Yang, S., Liu, J., Lu, S., Er, M. H. & Kot, A. C. Collaborative learning of gesture recognition and 3D hand pose estimation with multi-order feature analysis. *Eur. Conf. Comput. Vis. (ECCV)* https://doi.org/10.1007/978-3-030-58580-8_45 (2020).
102. Wang, J. et al. Rgb2hands: Real-time tracking of 3d hand interactions from monocular rgb video. *ACM Trans. Graph. (ToG)* **39**, 1–16. https://doi.org/10.1145/3414685.3417852 (2020).
103. Kwon, T., Tekin, B., Stühmer, J., Bogo, F. & Pollefeys, M. H2O: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).
104. Han, S. et al. Megatrack: Monochrome egocentric articulated hand-tracking for virtual reality. *ACM Trans. Graph. (ToG)* **39**, 87–91. https://doi.org/10.1145/3386569.3392452 (2020).
105. Liu, S., Jiang, H., Xu, J., Liu, S. & Wang, X. Semi-supervised 3D hand-object poses estimation with interactions in time. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14682–14692. https://doi.org/10.1109/CVPR46437.2021.01445 (2021).
106. Baek, S., Kim, K. I. & Kim, T.-K. Weakly-supervised domain adaptation via GAN and mesh model for estimating 3D hand poses interacting objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).
107. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P. & Zisserman, A. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
108. Xu, H., Wang, T., Tang, X. & Fu, C.-W. H2onet: Hand-occlusion-and-orientation-aware network for real-time 3D hand mesh reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
109. Zhang, H. et al. Pymaf-x: Towards well-aligned full-body model regression from monocular images. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 12287–12303. https://doi.org/10.1109/TPAMI.2023.3271691 (2023).
110. He, Y., Yan, R., Fragkiadaki, K. & Yu, S.-I. Epipolar transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).
111. Han, S. et al. Umetrack: Unified multi-view end-to-end hand tracking for VR. In *ACM Special Interest Group on Computer Graphics (SIGGRAPH)* (2022).
112. Remelli, E., Han, S., Honari, S., Fua, P. & Wang, R. Lightweight multi-view 3D pose estimation through camera-disentangled representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).
113. Zhang, J., Cai, Y., Yan, S., Feng, J. et al. Direct multi-view multi-person 3D pose estimation. In *Conference on Neural Information Processing Systems (NeurIPS)* (2021).
114. Liu, Y., Wang, T., Zhang, X. & Sun, J. Petr: Position embedding transformation for multi-view 3D object detection. In *European Conference on Computer Vision (ECCV)* (2022).
115. Ji, M., Gall, J., Zheng, H., Liu, Y. & Fang, L. Surfacenet: An end-to-end 3D neural network for multiview stereopsis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2017).
116. Kar, A., Häne, C. & Malik, J. Learning a multi-view stereo machine. In *Conference on Neural Information Processing Systems (NeurIPS)* (2017).
117. Yao, Y., Luo, Z., Li, S., Fang, T. & Quan, L. Mvsnet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)* (2018).
118. Chen, R., Han, S., Xu, J. & Su, H. Point-based multi-view stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).
119. Yang, L. *et al.* Poem: Reconstructing hand in a point embedded multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
120. Ren, P. et al. Mining multi-view information: A strong self-supervised framework for depth-based 3D hand pose and mesh estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20523–20533. https://doi.org/10.1109/CVPR52688.2022.01990 (2022).
121. Zheng, X., Wen, C., Xue, Z., Ren, P. & Wang, J. Hamuco: Hand pose estimation via multiview collaborative self-supervised learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 20706–20716. https://doi.org/10.1109/ICCV51070.2023.01898 (2023).
122. Wang, Y., Xu, H., Heng, P. A. & Fu, C.-W. Sima-hand: Boosting 3D hand-mesh reconstruction by single-to-multi-view adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. https://doi.org/10.48550/arXiv.2402.01389 (2024).
123. Yang, J. et al. Mlphand: Real time multi-view 3D hand mesh reconstruction via mlp modeling. arXiv preprint arXiv:2406.16137https://doi.org/10.1007/978-3-031-72904-1_2 (2024).
124. Romero, J., Tzionas, D. & Black, M. J. Embodied hands: Modeling and capturing hands and bodies together. In *ACM Special Interest Group on Computer Graphics (SIGGRAPH)* (2017).

125. Schubert, E., Sander, J., Ester, M., Kriegel, H. P. & Xu, X. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst. (TODS)* **42**, 1–21 (2017).
126. Fu, J., Xiang, Z., Qiao, C. & Bai, T. Pt-flownet: Scene flow estimation on point clouds with point transformer. *IEEE Robot. Autom. Lett.* **8**, 2566–2573 (2023).
127. Menze, M., Heipke, C. & Geiger, A. Object scene flow. *ISPRS J. Photogram. Remote Sens.* **140**, 60–76 (2018).
128. Mayer, N. et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
129. Eldar, Y., Lindenbaum, M., Porat, M. & Zeevi, Y. Y. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.* **6**, 1305–1315 (1997).
130. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)* (2015).
131. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)* (2015).
132. Romero, J., Tzionas, D. & Black, M. J. Embodied hands: Modeling and capturing hands and bodies together. arXiv preprint arXiv:2201.02610 (2022).
133. Hasson, Y. et al. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
134. Aboukhadra, A. T., Malik, J., Robertini, N., Elhayek, A. & Stricker, D. Shapegraformer: Graformer-based network for hand-object reconstruction from a single depth map. arXiv preprint arXiv:2310.11811https://doi.org/10.48550/arXiv.2310.11811 (2023).

Author contributions

M.A.A. and J.N.M. conceived the study, designed the methodology, validated the results, performed the formal analysis, and curated the data. K.K. contributed to the conceptualization, methodology, validation, formal analysis. He also reviewed and edited the manuscript, supervised the project, and handled project administration and funding acquisition. A.E. and R.B. contributed to the methodology, experimentation, validation, formal analysis, investigation, and manuscript review and editing. D.S. provided conceptualization and resources, reviewed and edited the manuscript, and contributed to supervision, project administration, and funding acquisition. S.S.N. and M.U.S. contributed to the methodology, experimentation, validation, formal analysis, investigation, drafting of the original manuscript, and manuscript review and editing.

Funding

This work was partially funded by the Federal Ministry of Education and Research Germany under the project COPPER (01IW24009) and partially by the European Union under Horizon Europe, grant number 101092889, project SHARESPACE.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025