Applied Intelligence (2022) 52:16667–16682
https://doi.org/10.1007/s10489-022-03390-x

A graph-based approach for absolute 3D hand pose estimation using a single RGB image

Ikram Kourbane[1] · **Yakup Genc**[1]

Accepted: 15 February 2022 / Published online: 25 March 2022

Abstract
Monocular RGB-based 3D hand pose estimation is n is crucial for a wide range of augmented reality and human-computer interaction applications. However, this task is highly challenging due to occlusion, scale, and depth ambiguities. Most existing methods mainly focus on estimating a scale-normalized root-relative 3D pose from the cropped hand image. In this work, we propose a multi-stage GCN-based (Graph Convolutional Networks) approach to estimate the absolute 3D hand pose from a single RGB image. We exploit both the cropped hand and the global scene image, which provides clues about the hand scale and location in the camera space. Our network consists of three main stages: 2D key-points, 3D root-relative, and 3D absolute pose estimation. To achieve better performance, we propose a new loss function. It separates the extracted image features based on 3D joint locations to simplify the regression task. Extensive experiments on five public datasets show that our efficient model estimates accurate global 3D hand poses and performs favorably against several baselines and state-of-the-art methods. Also, we validate the proposed approach on a newly created dataset. It contains RGB hand images with accurate 3D pose annotations and high lighting and poses variations.

Keywords 3D hand pose estimation · Graph convolutional networks · Loss function · Multi-stage learning · Monocular RGB image · Global coordinates

1 Introduction

The hand is a primary and intuitive tool that allows humans to interact with the world. Thus, hand pose estimation is one of the fundamental and long-standing tasks in computer vision. It plays an essential role in many applications, such as augmented reality (AR) [1], virtual reality (VR) [2], gesture recognition [3], and human-computer interaction [4]. Early works in hand tracking and pose estimation use magnetic sensing devices such as gloves [5] and visual markers [6]. But these solutions are expensive and require complex calibration and setup [7]. To this end, current commercial systems are moving to vision-based 3D hand pose estimation methods since they are natural and increase comfort [8]. However, despite years of studies [9–11], this task remains challenging due to self-occlusion or object-occlusion, similarity among fingers, fast motion and the high degree of freedom of the hand.

With the fast advancement of deep learning techniques and the widespread availability of commodity depth sensors, RGB-D-based 3D hand pose estimation methods achieve reliable results [12–14]. However, this hardware requires high-power consumption, and it is limited to indoor environments restricting many applications [15–17]. Other approaches alleviate occlusion problems using multiple RGB cameras located at different viewpoints [18–20]. These techniques are difficult to set up and cannot operate in unconstrained scenes (e.g. community videos). As a result, 3D hand pose estimation from a single RGB image has gained considerable attention since it is more accessible and does not pose any setup overhead. However, this task is more challenging than the aforementioned approaches because it exhibits depth ambiguity. Recently, several learning-based methods tackle this problem by designing effective models trained on large-scale annotated datasets. We can divide them mainly into two major approaches: generative and discriminative. The former one adopts variational autoencoders

✉ Ikram Kourbane
ikourbane@gtu.edu.tr

Yakup Genc
yakup.genc@gtu.edu.tr

[1] Computer engineering, Gebze Technical University, Gebze, Kocaeli, Turkey

(VAEs) [21] and/or generative adversarial networks (GANs) [22] to model the 3D hand pose distribution [23–25]. The methods of the second category use Convolutional Neural Networks (CNNs) [26] that extract high-level features from the cropped hand image to estimate the 3D pose [15, 27–30]. Inspired by the natural graph representation of the hand, recent discriminative studies use graph convolution networks (GCNs) [31] to model kinematic relationships between the joints [32–34].

Most existing methods estimate a scale-normalized root-relative pose. In particular, they predict the positions of hand joints relative to a reference point located on the hand (e.g., the wrist). The global joint coordinates in the world coordinate system are unknown. They assume that the depth of the root joint and the global hand scale are provided at test time [15, 23]. In contrast, absolute hand pose estimation finds the hand joint positions in camera-centered coordinates [16, 17, 29]. This extends the application scope to AR/VR, where headsets require camera distance-aware 3D hand pose estimation approaches to perform hand-object interactions. Also, the absolute pose is crucial in multi-hand images because a root-relative pose may represent different global poses. Thus, estimating the absolute pose is essential to recover the spatial layout of the whole scene. However, this task is too challenging. One difficulty comes from the fact that it presents an ill-posed problem due to the irreversible geometry. In particular, a 2D point in the image plane can correspond to multiple 3D points in world space. Besides, recovering the global scale of the hand is challenging due to depth ambiguity. Finally, the absolute pose resides in a larger space than the root-relative pose, which changes the underlying optimization process.

We observe that the input of the previous approaches is the cropped hand image. They do not exploit the global scene image that contains valuable information, such as the size of the hand that provides a clue about its depth (e.g. big hands means small depth and vice-versa). Also, the interaction of hands with other commonly seen objects, specifically other body parts such as shoulders and heads, can help to learn the hand position. Motivated by this observation, we propose a GCN-based method that consists of three stages: the first one extracts high-level features from the original scene and the cropped hand images to estimate the global and the local 2D keypoints. The second stage combines the extracted image features with the predicted 2D local hand pose to estimate a root-relative non-scale normalized 3D hand pose. Finally, we estimate the absolute 3D hand pose using the previous stage outputs (Fig. 1). Besides, we propose a feature matching (FM) loss function to simplify the regression task and improve the performance. Specifically, we correlate the extracted image features to the root-relative 3D joints to make them sufficiently separable.

We conduct extensive experiments on five benchmarks [15, 17, 35–37]. Quantitative and qualitative results demonstrate that the proposed method achieves competitive results against several state-of-the-art methods with efficient running time. It estimates accurate 2D keypoints and 3D hand poses in challenging scenarios, such as complex texture backgrounds, self-occluded hands and object-occluded hands, Also, our approach estimate promising 3D global poses for both hands in the outdoor. Finally, we extend our experiments to our newly built dataset. It includes 60 thousand samples collected using the leap motion sensor [38] with different lighting conditions and high pose variation. Each sample contains the hand bounding box, the 2D keypoint, the 3D pose, and the corresponding RGB image. We can summarize our contributions as follows:

- We propose a multi-stage GCN-based approach that exploits the global scene and the cropped hand images to estimate an accurate absolute 3D hand pose from a single RGB image.
- We propose a feature matching loss to separate the extracted features of the hand image and simplify the 3D pose regression task.
- To evaluate the proposed approach, we conduct extensive experiments on several synthetic and real-world datasets. The reported quantitative and qualitative results demonstrate that our method outperforms state-of-the-art and estimates accurate 3D hand poses.

The rest of this paper is organized as follows: Section 2 reviews the related studies of our work. Section 3 explains the proposed method in-depth and describes the most important modules of our framework. Section 4 describes the experimental setting. Section 5 presents experimental results on five datasets and compares the proposed approach against the state-of-the-art methods. Finally, Section 6 presents the conclusion of the study and the direction for future work.

2 Related work

We classify 3D hand pose estimation methods based on the input modality into three categories depth-based, multiview RGB-based, and monocular RGB-based. In the past few years, numerous studies adopted deep learning techniques to handle various types of inputs for depth methods, such as 2D pixels [13], a set of 3D points [12] or voxels [14]. In the second category, several studies use many RGB cameras on different angles to alleviate the occlusion problem [18, 19]. RGB cameras are more available than the two-mentioned categories and work in all environments. Thus, there are several attempts to solve the monocular RGB-based 3D

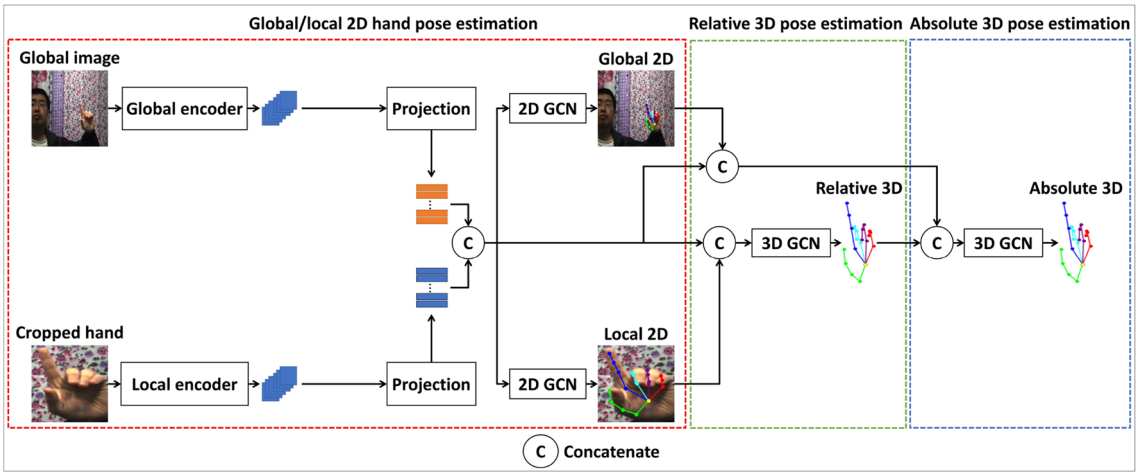

Fig. 1 Schematic overview of our multi-stage approach for monocular RGB-based absolute 3D hand pose estimation. We use ResNet-10 backbones for image feature extraction and GCNs for 2D/3D pose regression

hand pose estimation problem. Traditional methods [9–11] search the closest configuration that fits the hand model via an optimization process. These methods require powerful prior knowledge about many dynamic and physics hypotheses, which leads to poor performances. Learning-based algorithms outperform conventional techniques and estimate reliable 3D poses. We classify them in our study into two main groups:

2.1 Root-relative 3D hand pose estimation

[15] is the first learning-based study that estimates a root-relative 3D hand pose from a monocular RGB image. It employs synthetic data since they are more practical to generate. Their predicted 3D poses are relative to a canonical frame. After this work, several learning-based studies have been proposed to improve 3D hand pose estimation accuracy. [28] leverage depth information to supervise the training by having a pair of RGB-RGBD images. Other approaches achieve reliable 3D poses by regressing the hand mesh and estimating the 3D pose from it [30, 39, 40]. However, these methods require additional supervision since the datasets must include the meshes and the 3D poses. The graph-like structure of the hand inspires [32–34] to apply GCNs [31] on the 3D hand pose estimation task achieving accurate results.

Deep generative-based approaches also exhibit competitive performances and estimate reliable 3D poses. Spurr et al. [23] uses VAEs [21] to learn a shared latent space across RGB and depth modalities to remedy the missing depth ambiguity. However, these methods model black-box latent representations that synthesize a single 3D pose for a given RGB image. To address this issue, [24, 25] disentangle the diverse factors that affect the hand visualization, including camera viewpoint, scene context and background.

2.2 Absolute 3D hand pose estimation

Although absolute 3D hand pose estimation has more application in AR/VR and multi-hand scenes, it is less studied in the literature compared to the root-relative 3D pose. We can explain this by the fact that estimating the global coordinates is too challenging due to irreversible geometry and scale ambiguities. To the best of our knowledge, there are only three studies [16, 17, 29] that work around this problem. Mueller et al. [16] employs GANs [22] to translate synthetic labeled data to realistic images. This data augmentation reduces the domain gap and estimates a reliable scale-normalized root-relative 3D hand pose. Next, it recovers the absolute 3D hand pose using kinematic hand model fitting that optimizes different loss functions: a 3D loss term, an angle constraint loss and a temporal smoothness loss. However, optimizing loss functions with dissimilar objectives is prone to errors, e.g: failures in the previous frames may affect the temporal smoothness loss of the current one. [29] reconstructs the global 3D hand pose from the latent 2.5D heatmaps. In particular, it estimates the 2D key points and depth map to ensure the scale and translation invariance. This method assumes that the intrinsic camera parameters are known. To recover the hand scale, they conduct statistics about the dataset in a post-processing step (e.g average bone lengths). Recently, [17] estimates the global pose for the two hands using four main stages: hand segmentation, 2D canonical pose estimation, 3D canonical hand pose estimation, and 3D global hand pose estimation. It uses the spherical coordinate system to overcome the depth and rotation ambiguities encountered in the cartesian system. To eliminate the scale ambiguity, it fixes the distance between the finger MetaCarpoPhalangeal joints (MCP) and the palm. Our approach estimates the absolute 3D pose using a GCN-based framework that learns the hand scale and location without using pre-calculated statistics about the datasets or requiring the camera intrinsic parameters.

3 Methodology

3.1 Architecture

Estimating an absolute 3D hand pose from a single RGB image is challenging due to depth and scale ambiguities. Most existing 3D hand pose estimation methods crop the hand area in an input image with a ground truth bounding box or the bounding box that is predicted from a hand detection model. The cropped hand image is fed into the 3D pose estimation module to predict a scale-normalized root-relative pose. As their models take a single cropped image, estimating the absolute camera-centered coordinate of each keypoint is difficult. To handle this issue, we exploit the original scene image that includes relevant information that can constrain the problem, such as the size of the hand that provides a clue about its distance from the camera (depth). More specifically, a small hand indicates being far from the camera (large depth value) and vice-versa (Fig. 2). Also, the scale of the hand relative to the other objects in the scene helps to reduce the depth and scale ambiguities. In particular, when the hand pose, scale, and location change from an image to another, other objects in the background remain motionless. Their scale invariance gives information about the distance between the hand and the camera (z coordinate).

Our framework involves three training stages: global/local 2D pose, root-relative 3D pose and absolute 3D pose estimation (Fig. 1). In the first stage, we use two customized ResNet-10 networks (Table 1) as a backbone to extract image feature maps from the original scene image (global encoder) and the cropped hand image (local encoder). We feed their feature maps to projection layers (Convolution layer) that reduce the dimensionality of the feature maps from (256, 7, 7) and (256, 4, 4) to (21, 7, 7) and (21, 4, 4) for the global and local image, respectively.We flatten the last two dimensions of the feature maps from $(21, W, H)$ to $(21, W \times H)$, where W and H are the width and the height of the feature maps, respectively. The reason for this transformation is to get the correct input format for the GCN-based regressors, where each joint has a feature vector of size $W \times H$. We concatenate the feature vectors from the global and the local hand image in the second dimension as shown in (1). The resulting tensor of size (21, 65) is given to global and local GCNs to estimate 2D keypoint locations in the original scene and the cropped hand images.

$$output = concat([features_{local}, features_{global}], dim = 1) \quad (1)$$

Where: the shape of $features_{local}$ and $features_{global}$ is (21, 16) and (21, 49), respectively.

Recent works demonstrate that 2D key points are crucial for the 3D pose estimation task [41–43]. Thus, the second stage feeds the concatenation of the estimated local 2D poses and the global/local image feature vectors to the root-relative 3D regressor. Instead of solving quadratic equations or putting assumptions about the hand, our absolute 3D regressor learns depth from the cropped hand image and the global scene features. Meanwhile, it exploits the predicted root-relative 3D pose and the global 2D key points to estimate (x, y) coordinates in the 3D space.

3.2 Hand pose regression using graph convolution network

3.2.1 Revising GCNs

Let $G = (V, E)$ denote a graph, where V and E represent a set of M nodes and L edges, respectively. Let $A \in [0, 1]^{M \times M}$ be the adjacency matrix of G. Let I be the identity matrix and $\tilde{A} = A + I$ is the new self-loop adjacency matrix. Since the graph may include low/high-degree nodes a normalization process is required to avoid vanishing/exploding gradients. [31] addressed this problem by scaling rows and columns of $\tilde{A}$ in (2).

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \quad (2)$$

Where $\tilde{D}$ is the degree matrix of $\tilde{A}$. GCNs propagate information between nodes to update their representations. The adjacency matrix serves as a mask to select

Fig. 2 Illustration of two samples with the same relative 3D pose. Unlike the cropped hand image, the global scene preserves the original size of the hand that gives a clue about the distance from the camera

Table 1 The architecture of the customized ResNet-10 network

Layer	In	Out	Kernel	Stride	Padding
Conv2D	3	64	3×3	2	1
BN	64	64	-	-	-
ReLU	64	64	-	-	-
Max-pool	64	64	3	2	1
Res-Block	64	64	3×3	1	1
Res-Block	64	128	3×3	2	1
Res-Block	128	256	3×3	2	1
Res-Block	256	256	3×3	2	1

the aggregated nodes. The propagation rule in GCN is computed in (3):

$$H^{k+1} = \delta(\hat{A}H^k W^k) \tag{3}$$

Where: $\delta = ReLU$ is the activation function, H^k and W^k are the node features and the weights in the k^{th} layer, respectively.

3.2.2 Joint relationships representation for hand pose estimation

Unlike CNNs that only handle fixed structures, GCNs treat irregular ones in a non-euclidian space. Recent 3D hand pose estimation methods adapt GCNs since the hand joints have a basic graph data structure with proper geometry details. Inspired by these studies [32–34], we apply Chebyshev spectral GCN [31] to estimate the 3D hand joint coordinates from a single RGB image. There are two joint relationships representation approaches (adjacency matrices) in 3D hand pose estimation. The first technique uses a predefined hand skeletal-based adjacency matrix to model kinematic dependencies between joints. However, this approach misses potential relationships of physically disconnected joints. Also, it remains the same for all datasets. [33] addresses this limitation by replacing the skeletal relationships with global learnable joint relationships constraints. More specifically, the GCN network contains a weight matrix of size 21×21 that is updated during training, and the values are fed as input to the GC layers. The final weights in the matrix define the adjacency matrix learned from the dataset. In our 2D/3D GCN-based regressor, we use this propagation rule to estimate 2D local, 2D global, 3D root-relative pose, and 3D absolute pose. Our GCNs contains two graph convolution layers, where the number of features is set to 128. The output layer yields 2 and 3 features for 2D and 3D hand pose estimation, respectively. Since we concatenate outputs of different ranges from the previous stages, we use layer normalization before feeding the inputs to the GCNs.

3.3 Loss functions

Our network consists of three branches trained in a multi-stage manner.Our model is supervised by the proposed FM loss and the 2D/3D regression losses. Let $P_{absolute} = (X, Y, Z)$ be the absolute 3D pose of a hand joint. Also, let the global 2D pose p_{global} be the projection of $P_{absolute}$ on the image plane using intrinsic and extrinsic camera parameters K, where: $p_{global} = K * P$. To create the local 2D pose p_{local}, we crop p_{global} using the minimum and the maximum values of the x and y axis, respectively. To get the root-relative 3D pose, we subtract P_{palm} from $P_{absolute}$: $P_{relative} = P_{absolute} - P_{palm}$, where P_{palm} is the absolute 3D position of the root joint.

3.3.1 Features matching

We observe that current hand pose estimators do not separate the extracted feature maps before the regression task. Thus, we propose a new loss function that correlates the feature maps and 3D root-relative poses to make them linearly separable (Fig. 3). We feed the projected feature maps of the cropped hand image into a pairwise distance function, where we calculate the Euclidian distance between each feature vector pair in the (21, 16) tensor. Meanwhile, we calculate the Euclidian distance between each joint pair of the ground truth root-relative 3D hand pose. We minimize the distance between the two matrices to enforce agreements between image and 3D features in an intermediate space. This correlation simplifies the regression task to produce an accurate root-relative 3D hand pose. We use the pairwise distance function defined in the *nn* package of Pytorch ($torch.nn.PairwiseDistance$). We name our features matching loss as $FM(.)$.

3.3.2 Optimization

In this section, we explain our loss terms for each training stage. In the first stage, (4) and (5) calculates the L2 loss between the predicted and the ground truth global/local 2D

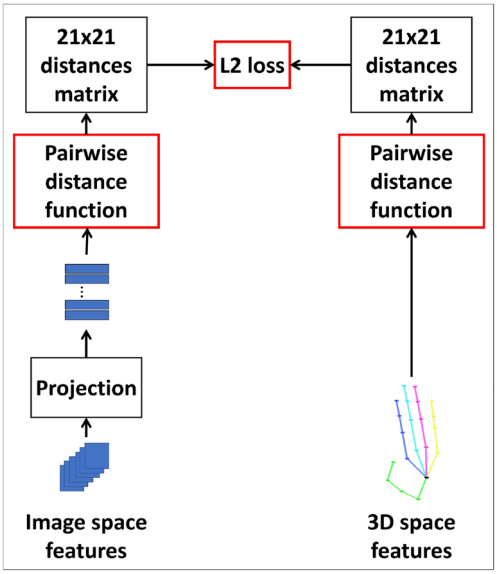

Fig. 3 Our feature matching loss measures the distance between the pairwise distance matrices of the cropped hand image feature maps and root-relative 3D hand pose

poses, respectively.

$$\mathcal{L}_{global} = ||\hat{p}_{global} - p_{global}||_2 \tag{4}$$

$$\mathcal{L}_{local} = ||\hat{p}_{local} - p_{local}||_2 \tag{5}$$

Besides, we employ the FM loss between the projected features of the cropped hand image (P_L) and the ground truth relative 3D pose ($P_{relative}$).

$$L_{FM} = FM(P_L, P_{relative}) \tag{6}$$

The overall loss of the first stage is the sum of all losses (7).

$$\mathcal{L}_{stage_1} = \mathcal{L}_{global} + \mathcal{L}_{local} + L_{FM} \tag{7}$$

In the second stage, we calculate the loss between the estimated $\hat{P}_{relative}$ and the ground truth $P_{relative}$ root-relative 3D hand pose using (8).

$$\mathcal{L}_{stage_2} = ||\hat{P}_{relative} - P_{relative}||_2 \tag{8}$$

In the final stage, we measure the loss between the estimated $\hat{P}_{absolute}$ and the ground truth P_{stage3} absolute 3D hand pose using (9).

$$\mathcal{L}_{stage_3} = ||\hat{P}_{absolute} - P_{absolute}||_2 \tag{9}$$

4 Experimental setting

4.1 Datasets

We conduct our experiments using different datasets: Stereo Hand Pose Tracking Benchmark (STB) [36], Multiview 3D Hand Pose dataset (MHP) [35], Rendered Hand Pose (RHD) [15] and First-Person Hand Action (FPHA) [37]. We also create the GTHD dataset to extend our experiments. The hand is represented by 21 joints: four key points (mcp, pip, dip and tip) per finger (thumb, index, middle, ring and little) and the hand palm (wrist).

The STB dataset is widely used to train and validate RGB-based 3D hand pose estimation methods. It contains 640 × 480 hand images with different illumination and background conditions. It includes 18K samples with the corresponding 3D poses covering random and counting gestures. To obtain the 2D keypoints, we perform a projection operation from the 3D space using the given camera intrinsic and extrinsic parameters.

MHP is a large-scale dataset that holds 21 hand motion videos collected from multiple views and divided into 60K, 15K, and 12760 images for the training set, validation set, and test set, respectively. Each sample includes RGB hand images, bounding boxes, 2D hand keypoints and the 3D pose for different hand sizes and colors.

RHD is one of the most challenging datasets since it presents low-resolution noisy synthetic images (320 × 320) containing diverse hand textures with heavily occluded fingers. The images are rendered from varying camera angles and collected from twenty characters performing 39 actions in front of random backgrounds. RHD contains 41,258 and 2,728 samples for training and test set, respectively. Each instance includes the RGB image, 2D key-points, 3D pose, depth map and segmentation mask.

FPHA presents first-person videos of complex hand actions corresponding to daily human activities (open, close and put) performed on different objects (milk, juice bottle, liquid soap, and salt). This dataset can be used for various tasks since it includes 3D hand pose labels, 6D object pose, and action categories. FPHA is a large and diverse dataset including 105,459 annotated frames obtained from 1175 videos performed by six actors.

To enrich our experiments, we build our new dataset (GTHD) using an RGB camera and a Leap Motion sensor [38]. To get the correct pose with its corresponding image, we synchronize the two sensors in time. The RGB camera provides an image with a resolution of 640 × 480 pixels. The leap motion controller senses the fingers to give the 3D joint locations. Thus, a projection process from 3D space to the 2D image plane is necessary. We achieve this goal in two steps. In the first one, we use OpenCV to estimate specific intrinsic parameters of the camera. The second step estimates the extrinsic parameters between the leap motion controller and the camera. To find the rotation and translation matrices, we manually mark one joint in a set of hand images and solve the *PnP* problem by computing the 3D-2D correspondences [44]. We capture the images in an indoor environment using six distinct backgrounds. To obtain a vast variety in the pose space, three subjects

with different skin tones and hand shapes perform several gestures, such as grasping, pointing, and counting. The motion of the hand was natural and unrestricted. To ensure sufficient lighting variations, we collect images at different times during the day. This allows various sampling of sunlight in the space. Our dataset contains 60 thousand RGB images with the corresponding hand bounding boxes, 2D keypoints, and 3D poses. To crop the hand from the image, we find the min_x, min_y, max_x and max_y of the 2D points. The top left corner (min_x, min_y) and the bottom right corner (max_x, max_y) are subtracted/added to a threshold of 20, ensuring that the cropped hand image includes all the joints. We randomly split the GTHD dataset into a training set (75%), a validation set (10%), and a test set (15%).

4.2 Metrics

To evaluate the accuracy of the proposed method and compare it against the state-of-the-art, we report the three most common metrics in 3D hand pose estimation:

- EPE: End-Point Error measures the average Euclidean distance between the ground-truth and the estimated key points. The distances are expressed in millimeters (mm) and pixels (px) for 3D and 2D hand pose estimation, respectively.
- PCK: considers the predicted joints correct if the distance to the ground truth joint is within a given threshold. It is widely used to evaluate RGB-based 3D hand pose estimation approaches between 20-50mm on STB, MHP and RHD datasets.
- AUC: instead of computing PCK for a single threshold, we show the area under the curve on PCK for different error thresholds. AUC metric gives a more accurate evaluation of the model performances.

4.3 Implementation details

We implement the proposed method using Pytorch v1.8 [45], CUDA v10.1 and cuDNN v7.6.4. We use a batch size of 128, and we train our model for 150, 75 and 50 epochs per stage, respectively. We resize the original scene and the cropped hand images to 224×224 and 128×128, respectively. We keep the original aspect ratio of the cropped hand images during resizing. We initialize the weights of the ResNet-10 network using a normal distribution, where the $mean = 0$ and $std = 0.02$. We use the Adam optimizer [46] with a learning rate of 0.001 decayed using a Cosine learning rate scheduler. We initialize the weights of the GCN networks using Xavier [47] where we set the gain to the square root of 2. We use NVIDIA's Apex mixed-precision training of 16-bits to speed up the training. Our whole training process takes two days on average to converge using an NVIDIA TITAN X GPU.

5 Experimental results

5.1 Ablation studies

To evaluate the proposed approach and find the best design choice, we conduct extensive experiments on the aforementioned datasets. We investigate the impact of the original scene (global) and the cropped hand (local) image on the 3D hand pose estimation accuracy. We also analyze the effect of the proposed FM loss. We report qualitative and quantitative results of different baselines, and we select the best model to compare against the state-of-the-art methods.

5.1.1 Analysis of the global and local modules

To validate the proposed approach, we perform *Baseline A* which employs only the cropped hand image as input and does not exploit any knowledge about the scene (global image). In particular, we remove the global feature extractor and the global 2D pose estimator modules from the framework in Fig. 1. We can see from Table 2 and Fig. 4 that the proposed method (*Full*) outperforms *Baseline A* that provides lower scores in both AUC and EPE metrics on all the datasets. We can explain this by that feeding the hand in its original scene gives a clue about its position in the camera coordinate system.

To support our claim, we conduct three additional experiments. Initially, to investigate the impact of the hand scale in the original scene, we perform *Baseline B*, where the inputs are the cropped hand image and the hand bounding boxes. We can see from Table 3 that *Baseline B* outperforms *Baseline A*, confirming the benefits of the hand scale information. Meanwhile, it shows that other motionless objects in the scene help to reduce the depth ambiguities since it reports inferior performance compared to our *Full* model. To verify this claim, we perform *Baseline C* that examines the impact of the face on the performance of the *Full* model. More specifically, we use the face detector [48] to remove the face pixels from the scene image, and then we apply our proposed architecture in Fig. 1. We conduct this experiment using STB and GTHD datasets since they include fontal faces with backgrounds and poses variation. Our ablation studies show that when the face is removed from the image, the results degrade. The third experiment is *Baseline D* which investigates the impact of the background on MHP and GTHD datasets. In particular, it replaces the background pixel values with white color and performs the proposed architecture. Experimental results show the benefit of the background in improving the estimation accuracy. These results can be explained by that while the hand scale and position change, static objects provide clues about the varied scale and depth.

Finally, we show the impact of local and global modules that use the cropped hand and the original scene image,

Table 2 Ablation studies of 3D hand pose estimation on STB [36], MHP [35], RHD [15] and GTHD datasets with EPE [mm] and AUC metrics

	Method	$MeanEPE^{\downarrow}$	$AUC^{\uparrow}_{0-50}$	$AUC^{\uparrow}_{20-50}$
STB	Full	**6.21**	**0.796**	**0.998**
	Full w/o FM loss	9.62	0.716	0.982
	Full w/o global	15.07	0.612	0.948
MHP	Full	**10.23**	**0.771**	**0.954**
	Full w/o FM loss	14.42	0.712	0.906
	Full w/o global	21.36	0.617	0.824
RHD	Full	**12.09**	**0.748**	**0.915**
	Full w/o FM loss	16.73	0.679	0.887
	Full w/o global	29.57	0.503	0.835
$GTHD$	Full	**9.03**	**0.783**	**0.972**
	Full w/o FM loss	13.23	0.720	0.924
	Full w/o global	19.71	0.625	0.861

↑: higher is better, ↓: lower is better

respectively. As seen from Table 3, the local module is more crucial than the global for the root-relative 3D hand pose estimation stage. We can explain this by that the network focuses on hand pixels and will not be affected by the background. In contrast, for the absolute 3D hand pose estimation stage, other objects in the scene and the original scale of the hand provide clues about its distance to the camera. We note that combining local and global modules yields the most effective baseline in both root-relative and global 3D pose estimation stages.

5.1.2 Effect of the feature matching loss

To investigate the impact of the proposed FM loss, we report quantitative and qualitative results of *Baseline Full w/o FM* that performs the proposed approach using only L2 loss between the ground truth and the predicted joint locations in 2D and 3D spaces. We can see from Table 2 and Fig. 4 that the FM loss improves the model performance and achieves more accurate results. To explain this, we visualize the extracted features of the cropped hand image w/wo FM loss.

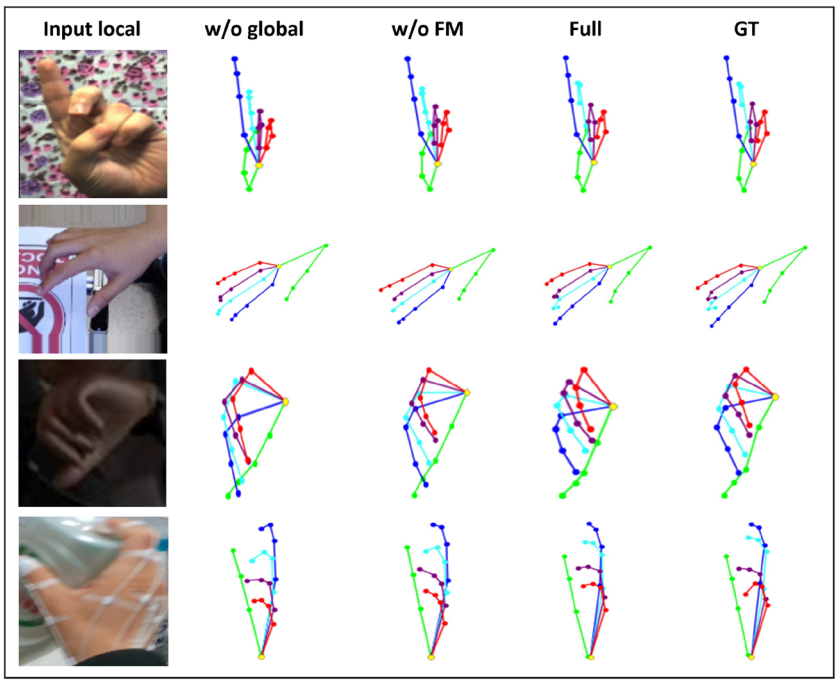

Fig. 4 Ablation studies of the proposed approach. From left to right, we show the input hand image, our model trained without the global image, our model trained without FM loss, our full model prediction and the corresponding ground truth 3D joint skeleton

Table 3 Ablation studies of the root-relative and the absolute pose estimation stages on STB [36], MHP [35] and GTHD datasets with EPE [mm] metric

	Method	STB	MHP	GTHD
Absolute pose	Full	**6.21**	**10.23**	**9.03**
	Baseline A: Full w/o global module	15.07	21.36	19.71
	Baseline B: Baseline A + hand bounding box	14.56	20.47	18.54
	Baseline C: Full w/o face pixels	11.08	-	12.81
	Baseline D: Full w/o background pixels	-	17.56	16.02
	Baseline E: Full w/o local module	9.53	15.34	14.96
Root-relative pose	Baseline A: Full w/o global	10.43	13.78	12.76
	Baseline E: Full w/o local	17.50	20.16	18.43

Specifically, we project the feature vectors of each joint into the 2D space using t-SNE [49]. We can see from Fig. 5 that the FM loss separates the extracted image features and more organized joint clusters appear in STB, MHP and GTHD datasets. Having large margins between the joints in the intermediate space facilitates the regression and improves

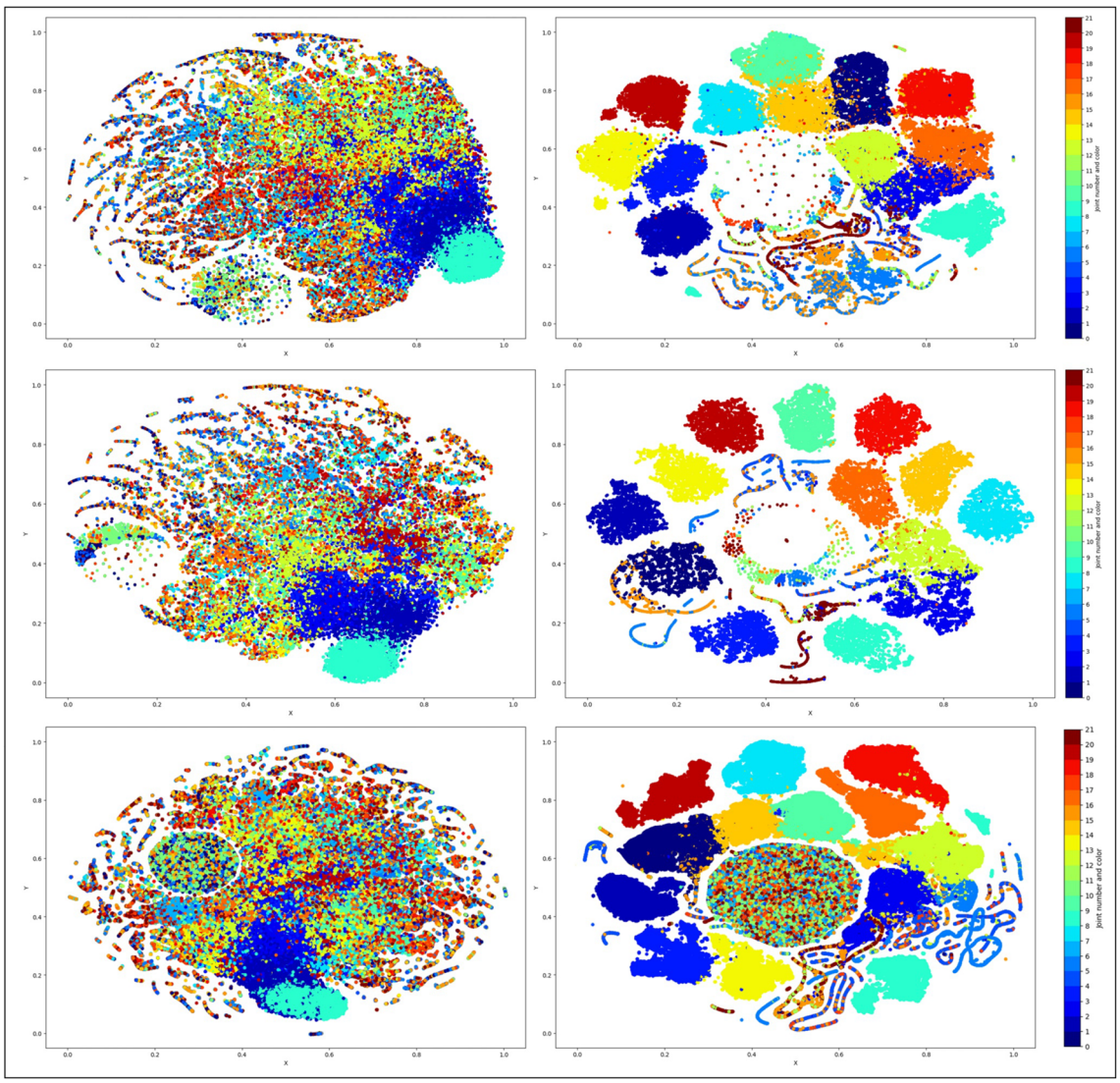

Fig. 5 The impact of the proposed FM loss on the STB [36], MHP [35] and GTHD datasets. Left and right images represent the extracted image features w/ and w/o the FM loss

the hand pose estimation accuracy. In contrast, *Baseline Full w/o FM* exhibits overlapped features, which makes the optimization task more challenging.

5.2 Comparison with the state-of-the-art methods

We compare the proposed approach against recent state-of-the-art monocular RGB-based 3D hand pose estimation methods [15, 16, 23–25, 29]. We validate the proposed method under the same evaluation standards of the selected algorithms [15]. We use the same preprocessing and data split for training and test. To ensure a fair comparison on STB and MHP datasets, we aligned the root key points (wrist) of the ground-truth and estimated poses before calculating the metrics of all the methods as done in [16]. Since the RHD and FPHA datasets provide 3D pose annotations for complete hand skeleton, we report the performance of the proposed approach by adding the ground-truth depth of the root joint and the global scale of the hand as done in [29].

Initially, we compare our approach against [15, 23–25] with the EPE metric. Experimental results in Table 4 demonstrate the superiority of the proposed method that reports the lowest error on the four datasets STB, MHP, RHD and FPHA. More specifically, it archives improvements up to 1.06 Mean EPE in STB (14.5%), 2.89 Mean EPE in MHP (22.02%), 5.02 Mean EPE in RHD (29.33%) and 0.17 Mean EPE in FPHA (2.49%). We note that our method exhibits an impressive gain on the RHD dataset, which has more occluded fingers and complex backgrounds. We also report the quantitative results of the 2D hand pose estimation task with the EPE [pixel] metric on STB and RHD datasets. We can see from Table 5 that our approach reports the lowest error outperforming [15, 29].

Table 4 Comparison with the state-of-the-art methods on STB [36], MHP [35], RHD [15] and FPHA [37] datasets using Mean EPE [mm]

Method	RHD	STB	MHP	FPHA
Zimmermann et al [15]	30.42	8.68	–	–
Spurr et al [23]	19.73	8.56	–	–
Yang et al [24]	19.95	8.66	–	–
Gu et al [25]	17.11	7.27	–	–
Chen et al [50]	18	10.05	13.12	–
Chen et al [51]	–	11.3	16.2	–
Boukhayma et al [30]	–	9.76	–	–
Hernando et al [37]	–	–	–	11.25
Tekin et al [52]	–	–	–	16.15
Doosti et al [33]	–	–	–	6.81
Ours	**12.09**	**6.21**	**10.23**	**6.64**

Table 5 Comparison with the state-of-the-art methods on RHD and STB datasets using Mean EPE [pixel]

Methods	RHD	STB
Zimmermann et al [15]	9.14	5
Iqbal et al [29]	3.57	-
Ours	**2.83**	**1.49**

Figure 6 shows the 3D PCK results comparison on the STB dataset of our approach and several state-of-the-art methods [9–11, 15, 16, 23–25, 27, 32]. Our method outperforms [9–11, 15, 16, 23, 24, 27] and achieves competitive performances compared to [25, 32].

In Fig. 7, we present the 3D PCK results comparison of the MHP dataset. Our method performs favorably against three single-view-based state-of-the-art methods [28, 50, 51]. However, it exhibits inferior performance compared to [20] that uses a multi-view-based approach. More specifically, it uses images collected from cameras located at different angles to reduce the depth ambiguity.

Also, we report the 3D PCK curves on the RHD dataset, which have varying backgrounds and viewpoints. Figure 8 shows that our method surpasses all the stated methods [15, 23, 24, 28] with a significant margin on all the PCK thresholds. We note that STB and MHP datasets contain fewer hand poses and background variations. As such, the

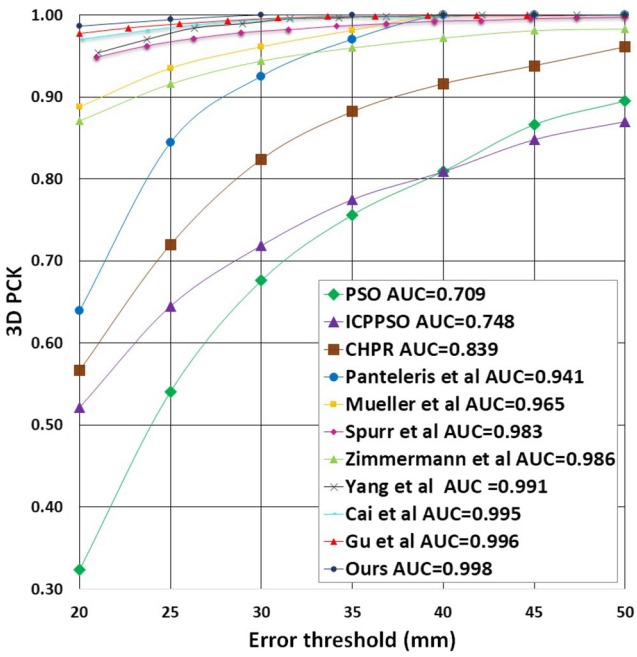

Fig. 6 Comparison with the state-of-the-art methods [9–11, 15, 16, 23–25, 27, 32] on the STB dataset [36] using 3D PCK

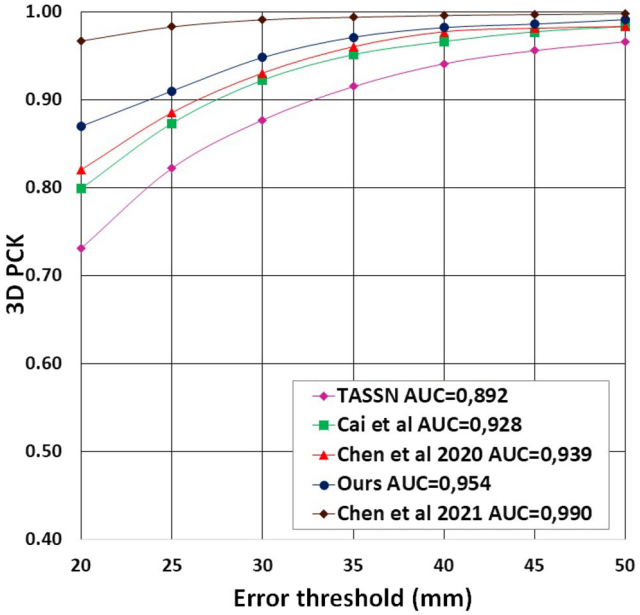

Fig. 7 Comparison with the state-of-the-art methods [20, 28, 50, 51] on the MHP dataset [35] using 3D PCK

advantages of our approach are more apparent on the RHD dataset.

Finally, to confirm the scalability of the proposed approach, we report the 3D PCK curve on the FPHA dataset [37] that includes hands holding objects of different sizes and shapes.

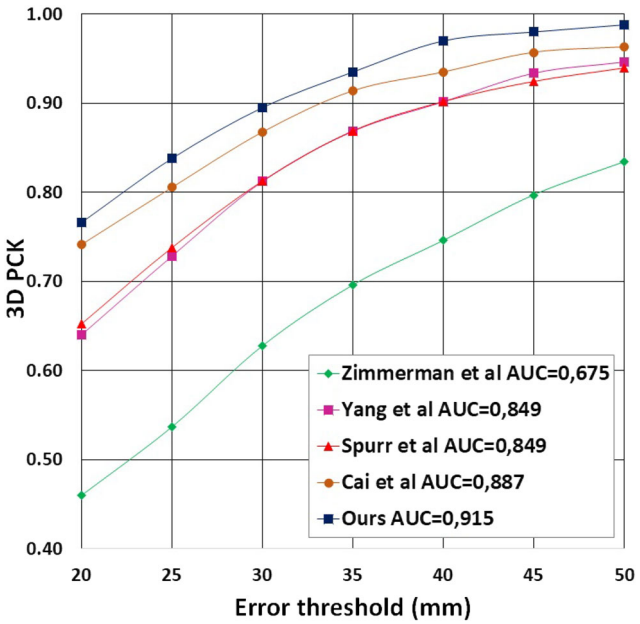

Fig. 8 Comparison with the state-of-the-art methods [15, 23, 24, 28] on the RHD dataset [15] using 3D PCK

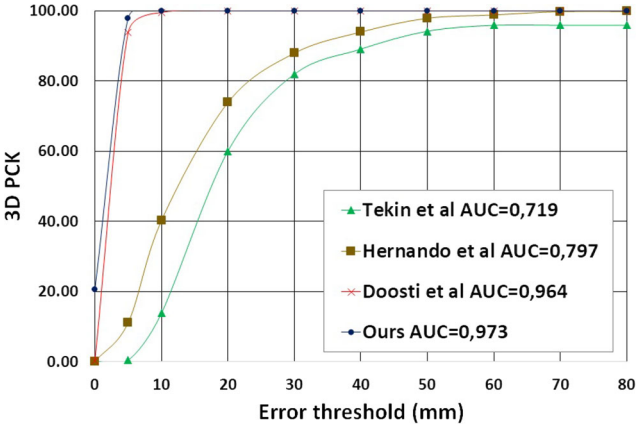

Fig. 9 Comparison with the state-of-the-art methods [33, 37, 52] on the FPHA dataset [37] using 3D PCK

Fig. 9 shows that our method performs favorably against the three reported state-of-the-art approaches [33, 37, 52].

5.3 Qualitative results

In addition to the quantitative evaluation, we report the qualitative results on different datasets. We visually evaluate the effectiveness of our model to estimate reliable poses in all the intermediate stages, including 2D local, 2D global and the root-relative 3D pose. Figure 10 shows some randomly selected test images on STB, MHP, RHD, FPHA and GTHD datasets. The proposed method can robustly estimate the 2D global and local hand poses that confirm the advantage of multi-task learning of the two branches. Besides, the root-relative 3D hand poses are accurate even in complex texture backgrounds and hand poses. That is mainly due to the proposed FM loss that enhances the extracted features to learn better representations.

We also report qualitative results for the global 3D hand pose estimation task. Seen from Fig. 11 that the proposed approach estimates reliable 3D hand poses on STB, MHP and GTHD datasets even in self-occluded hands and diverse lighting conditions. We can explain this by that combining the cropped hand and global scene reduces depth and scale ambiguities.

5.4 Computation complexity

The proposed approach is computationally efficient to state-of-the-art methods [34] as the running time on NVIDIA TITAN X GPU is 12ms. We can explain this by that the network consists of a lightweight feature extractor (ResNet10) and four GCN-based modules, which are significantly faster than CNNs [33]. While the 2D global pose estimator

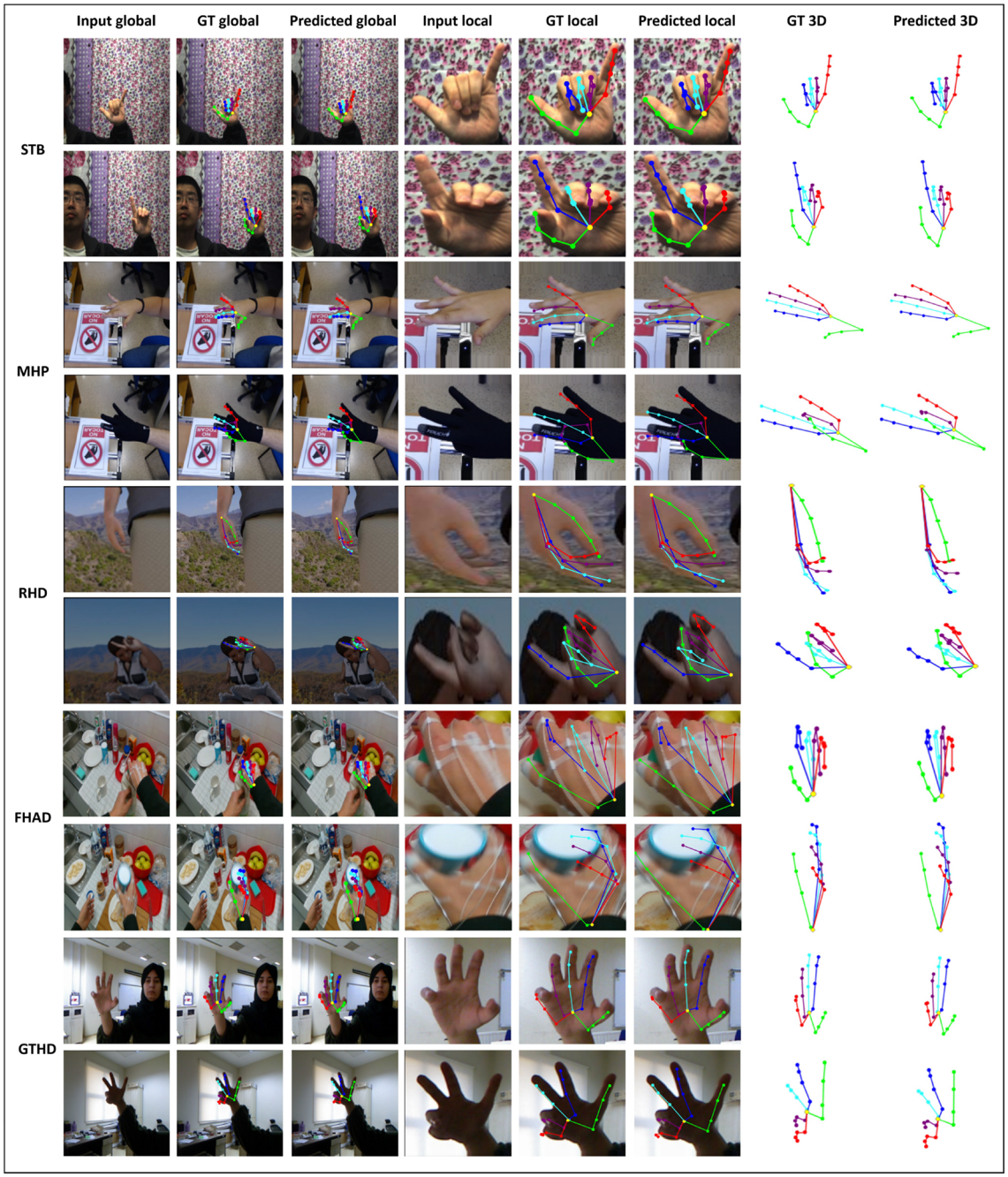

Fig. 10 The intermediate outputs of the proposed method on STB [36], MHP [35], RHD [15], FPHA [37] and GTHD datasets

module notably contributes to improving the performance, it necessitates only an additional 1.5ms per image, which does not speed down the inference time.

We train and test our model using two different backbones, specifically, ResNet-50 and our customized ResNet-10. Table 6 shows the used architecture, the number of parameters and the running time. Also, we report the EPE metric to evaluate the 3D hand pose estimation performance on the STB dataset. We can see that the model that uses our customized ResNet-10 as a feature extractor achieves competitive results in an efficient running time. In contrast, ResNet-50- based baseline significantly speeds down the computational time without noticeable improvements. Thus, we reduce the number of feature maps in each convolution

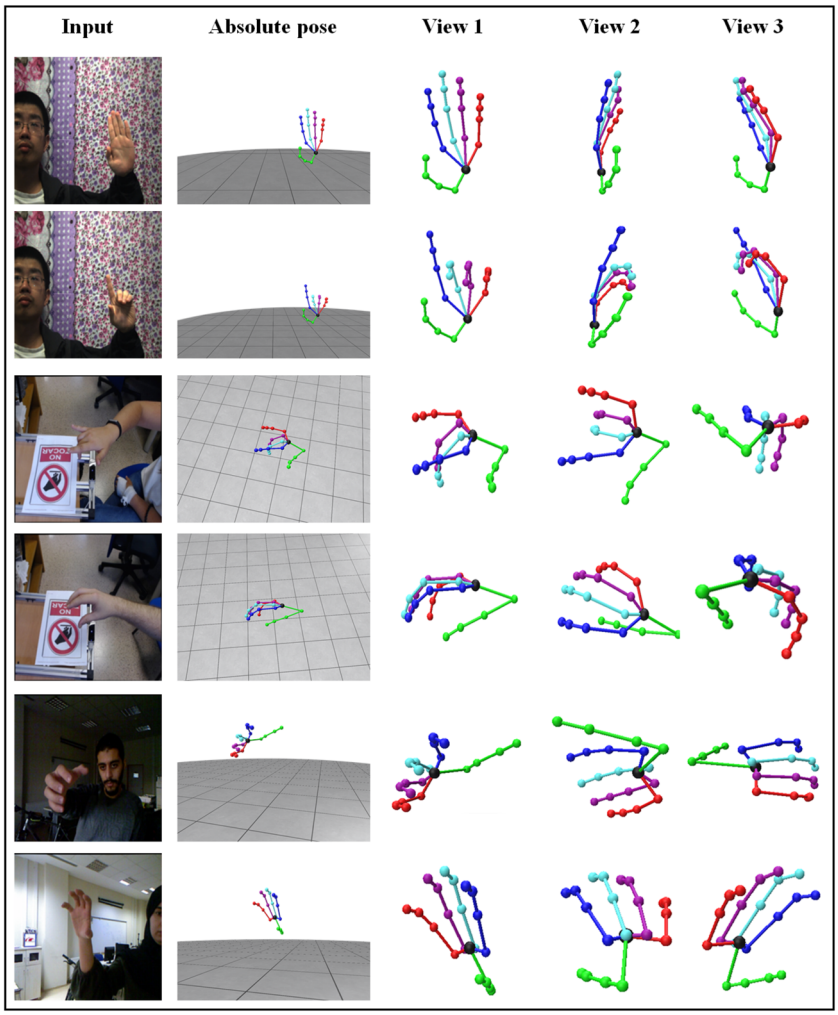

Fig. 11 Absolute 3D hand pose estimation results on STB [36], MHP [35] and GTHD datasets. We show the input image and the recovered 3D hand skeleton visualized from three different viewpoints

layer to improve the model efficiency without affecting the performance.

5.5 Absolute 3D hand pose estimation for two hands in the outdoor

In addition to the mentioned indoor datasets, we show the effectiveness of our method in an outdoor environment. We use the EgoHand dataset [17] that includes synthetic images with the corresponding absolute 3D annotations of the two hands. It contains 50K and 5K samples for training and test, respectively. It also has segmentation masks and the 2D key points for the two hands from an egocentric view. Since we have left and right hands, we apply small changes in our architecture. In particular, we share the local image encoder in Fig. 1 between the cropped left and right images. We keep the global image encoder to get original scene image features. Since the two hands have different articulations, we horizontally flip the right-cropped hand and its corresponding 2D and 3D ground truth data before feeding it to the shared image encoder to remain consistent for the model. Also, we share all the GCNs for the local 2D, the root-relative and the absolute 3D pose estimation. Our model yields competitive performances (AUC=0.914)

Table 6 Running time and performance comparison between ResNet-50 and our customized ResNet-10

Backbone	Number of parameters	Running time	EPE
ResNet-50	43.6M	55ms	**6.07**
Customized ResNet-10	**3.8M**	**12ms**	6.21

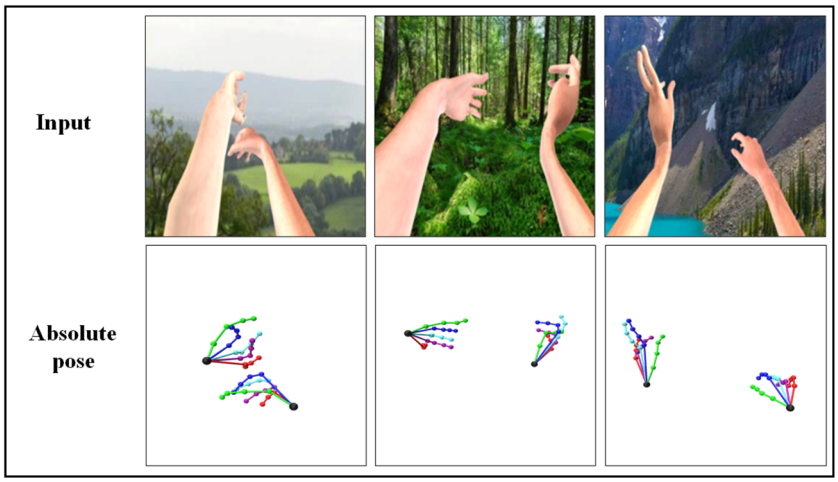

Fig. 12 Absolute 3D hand pose estimation results on the EgoHand dataset [17]. We show the input image and the estimated poses of the two hands

and estimates promising results for two hands in outdoor images (Fig. 12). In contrast, root-relative 3D hand pose estimation methods cannot work for multi-hand scenes since the predicted pose are relative to a single point (wrist) located in one hand.

6 Conclusion

We propose a multi-stage GCN-based method to estimate the absolute 3D hand pose from a single RGB image. We demonstrate the benefit of combining the cropped hand and original scene images in inferring accurate global coordinates. Besides, we present a new loss function that separates the extracted image features based on the 3D joint location to facilitate the regression task. Our experiments show that our approach improves the performances in some datasets noticeably and in others slightly. We can explain this by the fact that the advantage of the global feature noticeably appears in datasets containing hand samples with different depths. In contrast, fixing the distance between the hand and the camera decreases the model effectiveness. Also, our ablation studies demonstrate that using uniform backgrounds degrade the results. In future work, we will improve our approach using temporal 3D hand pose estimation, where the model recalls information about the hand motion from previous frame estimates, which are stored in a memory bank.

Funding No funds, grants, or other support was received.

Declarations

Competing Interests

Conflicts of Interest The authors declare that they have no conflict of interest.

References

1. Piumsomboon T, Clark A, Billinghurst M, Cockburn A (2013) User-defined gestures for augmented reality. In: IFIP conference on human-computer interaction. Springer, pp 282–299
2. Caggianese G, Capece N, Erra U, Gallo L, Rinaldi M (2020) Freehand-steering locomotion techniques for immersive virtual environments: A comparative evaluation. Int J Hum–Comput Interact 36(18):1734–1755
3. Zhou Y, Jiang G, Lin Y (2016) A novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recogn 49:102–114
4. Sridhar S, Feit AM, Theobalt C, Oulasvirta A (2015) Investigating the dexterity of multi-finger input for mid-air text entry. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems, pp 3643–3652
5. Bianchi M, Salaris P, Bicchi A (2013) Synergy-based hand pose sensing: Optimal glove design. Int J Robot Res 32(4):407–424
6. Usabiaga J, Erol A, Bebis G, Boyle R, Twombly X (2009) Global hand pose estimation by multiple camera ellipse tracking. Mach Vis Appl 21(1):1–15
7. Liang H, Yuan J, Thalmann D, Zhang Z (2013) Model-based hand pose estimation via spatial-temporal hand parsing and 3d fingertip localization. Vis Comput 29(6):837–848
8. Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X (2007) Vision-based hand pose estimation: A review. Comput Vis Image Underst 108(1-2):52–73
9. Oikonomidis I, Kyriazis N, Argyros AA (2011) Efficient model-based 3d tracking of hand articulations using kinect. In: BmVC, vol 1, p 3
10. Qian C, Sun X, Wei Y, Tang X, Sun J (2014) Realtime and robust hand tracking from depth. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1106–1113
11. Zhang J, Jiao J, Chen M, Qu L, Xu X, Yang Q (2016) 3d hand pose tracking and estimation using stereo matching. arXiv:1610.07214
12. Li S, Lee D (2019) Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 11927–11936
13. Oberweger M, Lepetit V (2017) Deepprior++: Improving fast and accurate 3d hand pose estimation. In: Proceedings of the IEEE international conference on computer vision workshops, pp 585–594

14. Moon G, Yong Chang J, Mu Lee K (2018) 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 conference on computer vision and pattern recognition, pp 5079–5088
15. Zimmermann C, Brox T (2017) Learning to estimate 3d hand pose from single rgb images. In: Proceedings of the IEEE international conference on computer vision, pp 4903–4911
16. Mueller F, Bernard F, Sotnychenko O, Mehta D, Sridhar S, Casas D, Theobalt C (2018) Ganerated hands for real-time 3d hand tracking from monocular rgb. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 49–59
17. Lin F, Wilhelm C, Martinez T (2021) Two-hand global 3d pose estimation using monocular rgb. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 2373–2381
18. Simon T, Joo H, Matthews I, Sheikh Y (2017) Hand keypoint detection in single images using multiview bootstrapping. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1145–1153
19. Wang R, Paris S, Popović J (2011) 6d hands: markerless hand-tracking for computer aided design. In: Proceedings of the 24th annual ACM symposium on User interface software and technology, pp 549–558
20. Chen L, Lin S-Y, Xie Y, Lin Y-Y, Xie X (2021) Mvhm: A large-scale multi-view hand mesh benchmark for accurate 3d hand pose estimation. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 836–845
21. Kingma DP, Welling M (2014) Auto-encoding variational bayes
22. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
23. Spurr A, Song J, Park S, Hilliges O (2018) Cross-modal deep variational hand pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 89–98
24. Yang L, Yao A (2019) Disentangling latent hands for image synthesis and pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9877–9886
25. Gu J, Wang Z, Ouyang W, Li J, Zhuo L et al (2020) 3d hand pose estimation with disentangled cross-modal latent space. In: The IEEE Winter conference on applications of computer vision, pp 391–400
26. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
27. Panteleris P, Oikonomidis I, Argyros A (2018) Using a single rgb frame for real time 3d hand pose estimation in the wild. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp 436–445
28. Cai Y, Ge L, Cai J, Yuan J (2018) Weakly-supervised 3d hand pose estimation from monocular rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 666–682
29. Iqbal U, Molchanov P, Breuel Juergen Gall T, Kautz J (2018) Hand pose estimation via latent 2.5 d heatmap regression. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 118–134
30. Boukhayma A, de Bem R, Torr PHS (2019) 3d hand shape and pose from images in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 10843–10852
31. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR)
32. Cai Y, Ge L, Liu J, Cai J, Cham T-J, Yuan J, Thalmann NM (2019) Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 2272–2281
33. Doosti B, Naha S, Mirbagheri M, Crandall DJ (2020) Hope-net: A graph-based model for hand-object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6608–6617
34. Ge L, Ren Z, Li Y, Xue Z, Wang Y, Cai J, Yuan J (2019) 3d hand shape and pose estimation from a single rgb image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 10833–10842
35. Gomez-Donoso F, Orts-Escolano S, Cazorla M (2019) Large-scale multiview 3d hand pose dataset. Image Vis Comput 81:25–33
36. Zhang J, Jiao J, Chen M, Qu L, Xu X, Yang Q (2017) A hand pose tracking benchmark from stereo matching. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, pp 982–986
37. Garcia-Hernando G, Yuan S, Baek S, Kim T-K (2018) First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 409–419
38. Potter LE, Araullo J, Carter L (2013) The leap motion controller: a view on sign language. In: Proceedings of the 25th Australian computer-human interaction conference: augmentation, application, innovation, collaboration, pp 175–178
39. Baek S, Kim KI, Kim T-K (2019) Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1067–1076
40. Zhang X, Li Q, Mo H, Zhang W, Zheng W (2019) End-to-end hand mesh recovery from a monocular rgb image. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2354–2364
41. Zhao L, Peng X, Tian Y, Kapadia M, Metaxas DN (2019) Semantic graph convolutional networks for 3d human pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3425–3435
42. Martinez J, Hossain R, Romero J, Little JJ (2017) A simple yet effective baseline for 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2640–2649
43. Fang H, Xu Y, Wang W, Liu X, Zhu S-C (2018) Learning pose grammar to encode human body configuration for 3d pose estimation. AAAI
44. Beardsley P, Murray D, Zisserman A (1992) Camera calibration using multiple images. In: European Conference on Computer Vision. Springer, pp 312–320
45. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch
46. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. International Conference on Learning Representations ICLR
47. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256

48. Deng J, Guo J, Yuxiang Z, Yu J, Kotsia I, Zafeiriou S (2019) Retinaface: Single-stage dense face localisation in the wild. arxiv
49. Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(11)
50. Chen L, Lin S-Y, Xie Y, Lin Y-Y, Fan W, Xie X (2020) Dggan: Depth-image guided generative adversarial networks for disentangling rgb and depth images in 3d hand pose estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 411–419
51. Chen L, Lin S-Y, Xie Y, Lin Y-Y, Xie X (2021) Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 1050–1059
52. Tekin B, Bogo F, Pollefeys M (2019) 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, pp 4511–4520

Ikram Kourbane received her B.Sc. and M.S degrees in Computer Engineering from the University of Batna (Algeria) in 2013 and 2015, respectively. She earned her Ph.D. degree in Computer Engineering at Gebze Technical University (Turkey). Her research interests are in the areas of computer vision and machine learning.

Dr. Yakup Genc received his PhD in Computer Science from the University of Illinois at Urbana-Champaign. Right after graduation, he joined Siemens Corporate Research (SCR) in September 1999. As research scientist, project manager, program manager and group manager, he developed technology and research strategy in the areas of computer vision, augmented reality and machine learning. His tenure at SCR produced numerous publications and patents. Since September 2012, as a member of the faculty of the Computer Engineering department at the Gebze Technical University, he continuous to conduct research in fields of computer vision, augmented reality, autonomous vehicles, machine learning and deep learning while maintaining close ties with the industry for practical applications of his research.