2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506135

REAL-TIME 3D HAND-OBJECT POSE ESTIMATION FOR MOBILE DEVICES

Yue Yin, Chris McCarthy, Dana Rezazadegan

Swinburne University of Technology
Department of Computer Science and Software Engineering
Melbourne, Australia

ABSTRACT

Interest in 3D hand pose estimation is rapidly growing, offering the potential for real-time hand gesture recognition in a range of interactive VR/AR applications, and beyond. Most current 3D hand pose estimation models rely on dedicated depth-sensing cameras and/or specialised hardware support to handle both the high computation and memory requirements. However, such requirements hinder the practical application of such models on mobile devices or in other embedded computing contexts. To address this, we propose a lightweight model for hand and object pose estimation specifically targeting mobile applications. Using RGB images only, we show how our approach achieves real-time performance, comparable accuracy, and an 81% model size reduction compared with state-of-the-art methods, thereby supporting the feasibility of the model for deployment on mobile platforms.

Index Terms— 3D hand pose estimation, mobile computing, gesture recognition, virtual reality, augmented reality

1. INTRODUCTION

3D human hand pose estimation is an area of growing interest in computer vision, with application across a wide variety of human-computer interaction scenarios including virtual and augmented reality (VR/AR) [1, 2]. Previous work has predominantly focused on accurately detecting 3D hand pose using either standard RGB images, or depth-sensing RGB-D cameras [3]. In particular, breakthrough results have been recently achieved using regression-based methods to locate key points of the hand [3], Generative Adversarial Networks (GAN) [4] and 3D Convolutional Neural Networks (CNNs) with applied residual blocks [5] on a large amount of datasets (see [6] for a recent review).

Despite such advances, only limited success has been achieved applying such methods in mobile and/or embedded settings. Specifically, dependence on specialised GPU hardware, high memory capacity to cater for typically large model sizes, as well as dedicated sensors for depth sensing [7] or hand-motion sensing [6] all limit the general applicability of these methods in computationally constrained settings.

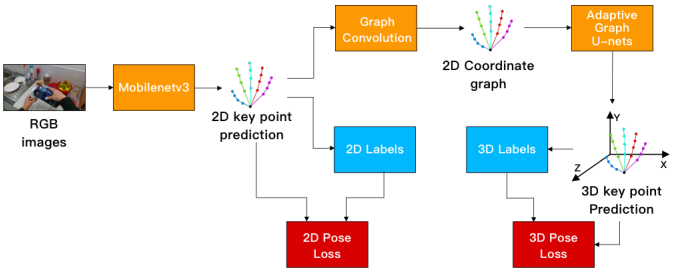

Fig. 1. Our proposed light-weight network architecture for real-time hand-object pose estimation.

To address this, we propose a hand pose estimation method specifically designed for deployment on mobile and embedded platforms. To achieve real-time performance, we consider the trade-off between accuracy, model size and overall performance with a focus on the practical needs of real-time mobile deployment. Inspired by recent methods such as HOPE-Net [8], we further consider the close relationship between human hand posture and hand-held objects that serve to constrain the space of possible hand poses. Taking HOPE-Net as a starting point, we compute hand and object pose estimation simultaneously but propose novel adaptations of the method to achieve real-time performance with comparable accuracy, and 81% reduction in the model size. Through a range of experiments, we test our model on different settings and parameter adjustments, including detection accuracy, frame rate, model size, and GPU memory usage. Our paper provides the following novel contributions:

- A novel RGB-based lightweight 3D hand-object pose estimation framework
- New results establishing the feasibility of real-time hand-object pose estimation for low-compute contexts.

2. BACKGROUND AND RELATED WORK

2.1. Hand Pose Estimation

Prior to the emergence of deep learning, a majority of works on 3D hand pose estimation employed traditional computer

vision methods such as random forests and other variants [9, 7]. However, such methods have since been superseded by data-driven deep learning based methods and thus we focus on these here. Moreover, in light of our focus on mobile device deployment, we only focus on methods designed for use with RGB images

RGB-based 3D pose estimation is particularly challenging, though recent work has made significant gains. Zimmermann et al use a regression network called PosePrior to predict the location of the joints and perform normalization [10]. From this, they find a 3D rotation matrix to align key points with the y-axis of the canonical frame. Dibra use a combination of depth images and RGB images to complete 2.5D to 3D conversion [11]. They then compare the differences of these generated 3D models to calculate the loss of the algorithm. Simon et al. propose a *multiview bootstrapping* approach to complete hand pose estimation. Multiview geometry is used to traingualte noisy key piont detections which are then reprojected and used as new labels for retraining, thereby continuously improving the accuracy of the detector [12].

2.2. Hand-object pose estimation

The strong relationship between the posture of the hand and the object it interacts with has motivated consideration of joint hand and object pose estimation. Oikonomidis et al. use the context of hand-object interaction to complete the task of multi-view hand pose restoration [13]. Choi et al. generate a 3D model of the interaction between the hand and the unknown object, while the depth information was also used [14]. Sridhar et al. introduce a 3D articulated Gaussian mixture alignment strategy to track the posture of the hand and the object together in real time [15]. Tekin et al. propose a 3D YOLO-based model to complete the 3D pose estimation of the hand and object directly from a separate RGB image [16]. More recently, Huang et al. propose HOT-net, utilising the structural correlation between 3D hand joints and object angles to conduct 3D hand pose estimation [17].

2.2.1. HOPE-Net and Graph Convolutional Networks

Our approach is most closely related to HOPE-Net [8], which applies an adaptive Graph U-Net to convert 2D hand key points into 3D key points. A ResNet architecture is used for obtaining 2D key point information from decoding RGB images, while an adaptive graph convolutional network is used to modify the initial prediction of the 2D coordinates based on the image features.

Graph Convolutional Networks (GCNs) were initially proposed by Kipf et al. [18], allowing deep learning methods to be applied to graph data. Gao et al. [19] introduced the Graph U-Net architecture with particular pooling and unpooling layers. In HOPE-Net, an adaptive Graph U-Net architecture takes the processed 2D key points as input and generates 3D key points through graph pooling and graph unpooling methods to complete the 3D hand pose estimation.

HOPE-Net and other state-of-the-art RGB-based methods achieve impressive accuracy but those are not directly applicable to mobile devices (or embedded computing contexts generally) due to excessive computational and memory requirements. In the following section, we outline an approach for such applications.

3. METHODOLOGY

Using HOPE-Net as a starting point, we outline a lightweight and real-time hand-object pose estimation method. The overview of our approach is shown in Fig 1. We discuss each component of the system below.

3.1. 3D Hand and object key-point detection

We use the commonly used encoder-decoder paradigm in the model structure design. We encode the inputs into a lower intermediate representation before decoding the inputs through spatial upsampling. Inspired by HOPE-Net, we used the combination of encoder and graph convolution to complete the 2D hand key point detection. However, our network and layer structure is completely different than HOPE-Net.

Unlike HOPE-Net which uses Resnet for encoding, we apply Mobilenetv3 [20]. This lightweight network structure can quickly convert the input image into a vector with multiple features and make preliminary predictions of 2D hand key points. Through the experiments, in order to obtain enough features to ensure the accuracy of the prediction results, we modify the number of layers in Mobilenetv3 such that it can decode the input image into a feature vector with the size of 1280. We also enhance the architecture with a fully-connected layer to complete the preliminary prediction of the key points of the 2D hand. We add the initial coordinate information of the key points to the vector, then get the graph data of each key point with spatial information and feature information. Finally, we perform graph convolution operations on the obtained graph data to use adjacency information, as HOPE-Net does [8].

We apply an Adaptive Graph U-Net to predict 3D hand key points based on 2D hand key points, while the overall structure of this network is similar to the classic Graph Unet [19]. Using graph convolution to process 3D key point detection makes it easier to find the relationship between the key points of the hand which improves the prediction accuracy, and reduces the required calculations.

3.2. Activation Functions

Our choice of activation function is also different than HOPE-Net. We used Hswish[20] instead of ReLu as used by HOPE-Net as the activation function. The Hswish function is an

approximation function of the swish function, which enables mobile devices to avoid sigmoid operations that require high computational complexity. Compared with ReLu, the proposed activation function can improve the accuracy of the model at a very small cost. In terms of model size, there is no significant difference between our model using Hswish and the model using ReLu.

3.3. Loss Function

Our model uses the Mean Square Error (MSE) as the loss function:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

where n is the number of points, y_i is the ground truth key point coordinate, and $\hat{y}_i$ the predicted location.

We calculate the MSE loss separately for the 2D and 3D key point detection parts, and then sum both for the total loss such that:

$$Loss = \alpha MSE_{\text{2D}} + MSE_{\text{3D}} \quad (2)$$

, where α is a scale factor. We use α to account for the different units of measurement (i.e., MSE_{2D} is measured in pixels, while MSE_{3D} is in millimeters), and is calculable directly from the DPI of the RGB image.

4. EXPERIMENTAL RESULTS

4.1. Experiment Setup and Metrics

4.1.1. First-Person Hand Action Dataset

The First-Person Hand Action Dataset contains first-person view videos of the hand interacting with various objects [21]. These hand movements are collected by a Mo-cap system, providing the 3D position of 21 joints of the hand. To facilitate comparison of model performance with HOPE-Net, we adopt the same subset and object processing method as described in [8]. A bounding box composed of eight 3D coordinate points is used to describe the objects interacting with the hand. This subset includes 21501 frame images, 11019 to be used for training and 10482 to be used for evaluation.

4.1.2. Implementation Details

We implemented our model using the PyTorch deep learning framework. We train the whole model for 2000 epochs, with a batch size of 32 and initial learning rate of 0.001. We use exponential decay in our training process, with the learning rate decaying to 1% of the original learning rate at the end of training. All images were resized to 224×224 pixels before being passed to the modified Mobilenetv3 model.

4.1.3. Metrics

To thoroughly examine performance with respect to our intended real-time mobile application, we include various performance indicators to evaluate and compare our proposed model. In order to have a fair comparison, we compared the performance of our proposed model against HOPE-Net, as it was the only closely related existing work in the domain of to hand-object pose estimation. In section 4.2, 4.3 and 4.4. we present our results compared to HOPE-Net, which has been also summarised in Table 1. Specifically, we compare pose estimation accuracy, the average frame rate at run-time, GPU memory usage, and the size of the model. We describe these in more detail below.

Model Size: We use the storage space occupied by the model (Kb) to measure the size of the model

Accuracy: Similar to [16], We evaluate the accuracy of our model using the well established Percentage of Correct Keypoint (PCK) metric for both 2D and 3D coordinates. In this metric, a keypoint is considered correct if the average distance to the ground truth position is less than a threshold.

GPU usage: We record the use of computing resources by monitoring GPU usage during model testing in real time. We record the current GPU memory usage once every 1 second, and calculate the average of all recorded valuess at the end of the model test.

4.2. Model size comparison

Deployment on mobile devices emphasises the importance of model size reduction in order to account for the limited storage space which is typically available. We compared the storage space occupied by our model and HOPE-Net's model, in Table 1. Notably, HOPE-Net [8] uses ResNet50, a network with substantially more layers than Mobilenetv3 utilised by our approach. ResNet50 decodes each input image into a vector with 2048 features, whereas a feature vector of size 1280 is obtained by our model. To compensate this, and to avoid the difference in network size which is caused by the difference in the number of generated features, we multiply the depth of our decoding network by a coefficient to increase the number of feature vectors of the decoded picture to be the same as HOPE-Net. In Table 1 we include both versions of our model in the comparison of model size.

Through comparison, we see that both versions of our model provide substantial reductions in size compared to HOPE-Net, with our unmodified version utilising only 18% of HOPE-Net's storage space. Such a light model will be easier to deploy on memory-constrained devices with limited storage space.

4.3. GPU Utilisation and frame rate Comparison

To fairly compare the memory requirements of different models at runtime, we fixed the pose estimation frame rate of all

Table 1. Comparison of model size and GPU Utilisation (memory) when reaching a frame rate of 30 frames per second(fps) on the First-Person Hand Action dataset. Model size comparison results for our model are reported for both the un-modified version (Ours) and modified version in which the same sized feature vector as HOPE-Net is used (Ours(2048D))

Model	Size(KB)	GPU Usage(Mib)
HOPE-Net	93665	1758
Ours	17561	1290
Ours(2048D)	43448	1582

Table 2. Comparison of loss distance and frame rate on the First-Person Hand Action dataset. GPU usage is fixed at 1.7G. The average loss distance is the average of the distances between all ground truth keypoint and predicted coordinates.

Model	Average loss distance(mm)	Frame rate(fps)
HOPE-Net	50	31
Ours	67	60

models to 30 frames per second (fps).

To this end, we used a GPU that can provide 16GB of video memory for testing. Table 2 demonstrates the average memory requirement data per second, for different models at an output frame rate of 30 fps.

By comparing the data, it can be seen that GPU utiliation when the frame rate was set to 30 fps was 35% less for our model compared to HOPE-Net Also, GPU usage remained highly stable, and thus usage error is within 1 Mib(mebibyte). This result can be further optimised through pruning operations.

To examine the stability of relative GPU usage between the two, we further explored different frame rate settings. Our results indicated no change in this relative GPU usage reduction compared with HOPE-Net.

Finally, we explored the impact of fixing the GPU usage at 1.7GB across all models on the frame rate. Table 2 reports the result of this, where it can be seen that our approach achieves nearly double the frame rate of the HOPE-Net, suggesting a substantial speed up in frame processing for the same GPU usage compared with HOPE-Net.

4.4. Accuracy Comparison

In order to understand how changes in the computing environment effect detection accuracy, we compared the PCK performance of the model in a low-computing environment (1GB graphics card) and a high-computing environment (6GB graphics card). Based on the results (Fig. 2), we found that the increase in GPU capacity can significantly improve the detection accuracy of our model. As the PCK threshold increases, our model approaches the accuracy of Hope-Net.

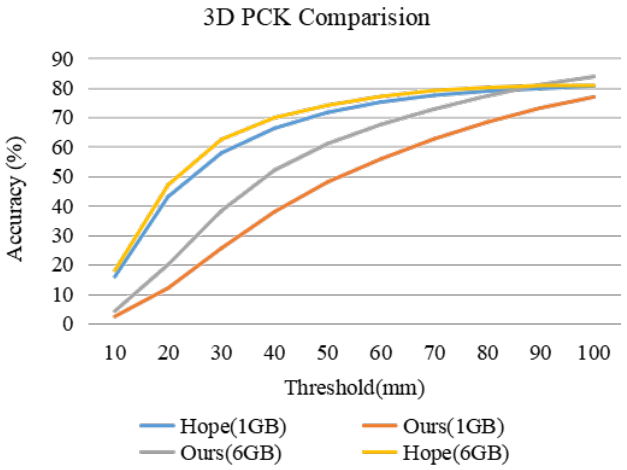

Fig. 2. Comparison of model accuracy using Percentage of Correct Keypoint (PCK). Comparisons include results achieved under low-computing (1GB GPU) and high-computing (6GB GPU) conditions.

This suggests that for tasks with relatively relaxed detection error requirements, our model can achieve comparable accuracy with higher efficiency compared to HOPE-Net.

5. SUMMARY AND CONCLUSION

In this paper we introduced a lightweight model for real-time 3D hand pose estimation using RGB images, targeting deployment on mobile devices and embedded computing platforms. Overall, our results supported the feasibility of real-time 3D hand pose estimation, using only RGB images, on mobile and embedded computing devices. Moreover, our results confirmed that adaptive graph convolutional networks can be applied to the light-weight neural networks such as Mobilenetv3, achieving comparable accuracy and significantly faster and computationally more efficient performance. The proposed model had a remarkable size reduction making it more flexible to be deployed on various types of equipment. However, our method has some limitations. While accuracy is still comparable, restrictions on the depth of the 2D decoder impose substantial reduction to the achievable accuracy. Hence, our approach is best suited to applications where high accuracy is not critical (e.g. consumer level VR/AR applications, and where temporal filtering can also be applied). Future work will consider the use of data augmentation to improve pose estimation results and accuracy without increasing computational overhead.

6. REFERENCES

[1] Y. Jang, S. Noh, H. J. Chang, T. Kim, and W. Woo, "3d finger cape: Clicking action and position estimation

under self-occlusions in egocentric viewpoint," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 501–510, 2015.

[2] T. Lee and T. Hollerer, "Multithreaded hybrid feature tracking for markerless augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 355–368, 2009.

[3] Ayan Sinha, Chiho Choi, and Karthik Ramani, "Deephand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4150–4158.

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim, "Augmented skeleton space transfer for depth-based hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8330–8339.

[5] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee, "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*, 2018, pp. 5079–5088.

[6] Bardia Doosti, "Hand pose estimation: A survey," *arXiv preprint arXiv:1903.01013*, 2019.

[7] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*. Ieee, 2011, pp. 1297–1304.

[8] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall, "Hope-net: A graph-based model for hand-object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6608–6617.

[9] Chi Xu and Li Cheng, "Efficient hand pose estimation from a single depth image," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3456–3462.

[10] Christian Zimmermann and Thomas Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4903–4911.

[11] Endri Dibra, Silvan Melchior, Ali Balkis, Thomas Wolf, Cengiz Oztireli, and Markus Gross, "Monocular rgb hand pose inference from unsupervised refinable nets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1075–1085.

[12] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1145–1153.

[13] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros, "Efficient model-based 3d tracking of hand articulations using kinect.," in *BmVC*, 2011, vol. 1, p. 3.

[14] Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani, "Robust hand pose estimation during the interaction with an unknown object," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3123–3132.

[15] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt, "Real-time joint tracking of a hand manipulating an object from rgb-d input," in *European Conference on Computer Vision*. Springer, 2016, pp. 294–310.

[16] Bugra Tekin, Federica Bogo, and Marc Pollefeys, "H+ o: Unified egocentric recognition of 3d hand-object poses and interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4511–4520.

[17] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan, "Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3136–3145.

[18] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[19] Hongyang Gao and Shuiwang Ji, "Graph u-nets," *arXiv preprint arXiv:1905.05178*, 2019.

[20] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al., "Searching for mobilenetv3," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1314–1324.

[21] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim, "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*, 2018, pp. 409–419.