Multiple hand pose estimation from Single RGB Images

Harryn Oh, Ajay Mahendrakumaran, Luke Jackson,
Takuya Boehringer, Vladimiros Karin, Lucas Marrie

***Abstract*—This report presents a novel framework aimed at reconstructing and estimating the 3D hand poses of both hands from a single RGB image. Conventional methods for 3D hand pose estimation have typically been tailored for scenarios involving a single hand and struggle to adapt to situations with two hands due to occlusion and spatial constraints. Our focus lies specifically in addressing the challenge of estimating hand poses when both hands are present, whether they are interacting or not and for cases where there are more than two hands. We aim to surpass previous efforts in this regard, even in cases where both hands belong to the same side (e.g., two left hands in the image). However, the scarcity of datasets featuring images with two hands has highlighted the necessity of creating such datasets ourselves. To facilitate our research, we draw upon existing models such as InterNet [1], which is a model that specialises in 3D single and interacting hand pose estimation. Our code is available [1]**

***Index Terms*—Dataset Creation, Multi-hand Pose Estimation, 3D Reconstruction, InterHand, InterNet**

I. Introduction

Estimating hand pose and shape from single RGB images has garnered significant attention in recent years due to its wide range of potential applications, including augmented reality/virtual reality (AR/VR) systems and robotics. Past research has made considerable strides in the field, particularly in the realm of isolated hand pose estimation. Techniques have been developed to address challenging scenarios such as hand pose estimation from egocentric camera views or interactions with objects. However, a notable gap remains in the literature: the absence of robust methods for simultaneously estimating the poses of both hands from a single RGB image.

In real-world scenarios, human interactions with objects and tasks often involve the coordinated use of both hands. While existing methods have tackled single-hand pose estimation effectively, extending these techniques to handle multiple hands presents unique challenges. Traditionally, solutions to this problem have relied on depth sensors, multi-view camera systems, or optimization over tracked motion sequences. While effective, these approaches are often impractical due to their high cost, energy consumption, or complexity. Another problem that arises is that interacting hands often cause self-occlusions.

Thus, there exists a pressing need for the development of efficient and accurate techniques for multiple hand pose estimation from single RGB images. Previous works have tackled this by taking model fitting-based approaches but very recently there have been studies on convolutional neural network (CNN)-based learning approaches [2]. CNN-based approaches have been achieving successful results in addressing occlusions when provided with the context of an egocentric view or objects under interaction. However, for challenges such as self-occlusion - more specifically interacting hands - there is a limitation of data. By overcoming this challenge, researchers can unlock new possibilities for applications in diverse fields such as human-computer interaction, sign language recognition, and gesture-based interfaces. A good example of this is InterHand (shown in Figure 1) [1], which consists of a large-scale dataset InterHand2.6M and a network InterNet [1] which is able to perform single and interacting hand pose estimation from a single RGB image.

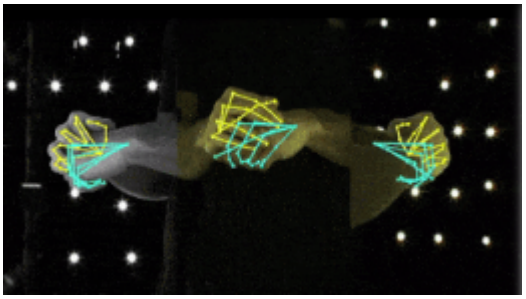


Fig. 1. Hand pose estimation from the InterNet model on the InterHand2.6M [1]

In this paper, we extend this field by developing the pipeline to incorporate a hand segmentation module which initially detects the hands within the frame and would apply a mask to allow for easier hand detection and pose estimation, most importantly for multiple hand pose estimation (two or more).

II. Related Work

The development of systems for 3D hand pose estimation from monoscopic RGB images, especially those handling multiple hands, marks a significant milestone in computer vision, with profound applications in human-computer interaction and augmented reality. This review segment delves into the progression of methodologies, challenges, and innovations tailored to the estimation of multiple hand poses.

Segmenting multiple hands from monoscopic RGB images presents unique challenges, primarily due to the increased likelihood of occlusion and the complexity of differentiating between hands. Despite these challenges, advances in segmentation technologies, such as those pioneered by Zimmerman and Brox [3] and later by Panteleris et al. [4] using YOLO

[1]https://github.com/harryn0502/Hand-Pose-Estimation

v2, have laid the groundwork for distinguishing multiple hands within an image. Tools like OpenPose, which have set benchmarks for body part segmentation, demonstrate potential for extension to multiple-hand scenarios by differentiating between left and right hands and identifying individual hand boundaries with high precision [5].

Detecting joints in scenarios featuring multiple hands from monoscopic RGB images demands architectures capable of discerning fine details amidst the complexities of overlapping and interacting hands. The task is challenging due to the need to differentiate between and accurately predict joint positions for each hand, considering their potential occlusion and proximity to one another. The Graph Convolutional Neural Networks (Graph CNNs) approach by Ge et al. [6] introduces a methodology that could be adapted for multi-hand scenarios. Graph CNNs process data structured in graph form, making them suitable for analyzing the spatial relationships inherent in images of multiple hands. By treating each hand as a separate graph entity, it's conceivable to extend this approach to simultaneously predict joint positions for multiple hands, thereby overcoming the challenges of inter-hand occlusion and interaction. Similarly, the Hourglass Network architecture, utilized by Yang et al. [7] and further explored by Zhang et al. [8] among others, presents a framework for multi-resolution analysis that can be crucial for multi-hand joint detection. The Hourglass Network's design, which captures and processes images at multiple scales, could be instrumental in identifying and distinguishing between joints of closely situated or overlapping hands. By refining predictions through successive stages, these networks are adept at capturing both global posture and fine joint details necessary for accurate multi-hand pose estimation.

The task of fitting hand models in scenarios with multiple hands involves sophisticated optimization techniques to accurately map 2D detections to 3D hand poses. The MANO hand model [9], known for its flexibility in representing various hand shapes and poses, stands out as a particularly promising tool for extending 3D hand pose estimation methods to multiple hands. The ability of MANO to adjust for individual hand variations could be leveraged to model each hand's unique pose and shape parameters in a scene, facilitating a more nuanced representation of multiple hands.

Systems designed for estimating the poses of multiple hands must overcome significant challenges, including hand-hand occlusions, complex inter-hand interactions, and the computational demands of processing multiple hand models simultaneously. Future research will benefit from focusing on these areas, developing more robust segmentation techniques, enhancing joint detection algorithms for multi-hand scenarios, and exploring efficient model fitting approaches that can handle the dynamics of multiple hands.

In conclusion, while substantial progress has been made in the field of 3D hand pose estimation from monoscopic RGB images, the specific challenges associated with multiple hand estimation underscore the need for continued innovation. Advancements in deep learning, model development, and dataset creation specifically tailored to multiple hands will be crucial in advancing this field further.

III. Research Hypothesis

Estimating multiple hand poses from a single RGB image presents unique challenges due to factors such as occlusion, complex backgrounds, and diverse hand shapes. One of the closest approaches to estimating multiple hand poses from a single RGB image is the method proposed by Meng et al. [10], which involves hand de-occlusion and the removal of hands from the original image to create new images for a single hand pose estimator. However, this method primarily focuses on interactions between a single left and a single right hand. In contrast, our goal is to enhance this method to estimate multiple hand poses, whether they involve multiples of left hands, right hands, or a combination of both interacting together. The central hypothesis of our research is that by isolating each hand into an individual image as input for a single hand pose estimator, we can estimate multiple hand poses—not only for a single left and right hand interacting together but also for scenarios involving only left or only right hands.

Our research objectives to achieve our goal are summarised as follows:

- To develop a robust Mask R-CNN model that can segment multiple interacting hand poses in RGB images.
- To create a comprehensive synthetic dataset featuring multiple left or right hands interacting in RGB images.
- To conduct extensive experiments with our model and synthetic dataset to demonstrate that our method successfully estimate multiple hands.

If the hypothesis holds true, our approach is expected to yield a method with the following capabilities:

- **Multiple Hand Detection:** The method should successfully segment and estimate poses for multiple hands within a single image frame.
- **Applicability** The method should be able to compatible with most of the existing hand pose estimation without the need of re-training process.

Due to the quality and diversity of the available dataset and the computational resources required for training large neural networks the our proposed model may not perform as expected, however it can provide the general method that will support for further research.

IV. Experiment Design

In this section, we introduce our experiment design for multiple hand pose estimation from single RGB images.

A. Methodology

As shown in Figure 2, our method employs a three-step pipeline. Initially, it begins with a **HandSeg** (Hand Segmentation) module that utilizes a deep learning network, specifically the Mask Region-based Convolutional Neural Network (Mask R-CNN), which is one of the highly successful models for instance segmentation tasks in computer vision. Our HandSeg module receives an RGB image and generates masks for each individual hand, presented as a black and white image where the hands are white, and everything else is black. In the second

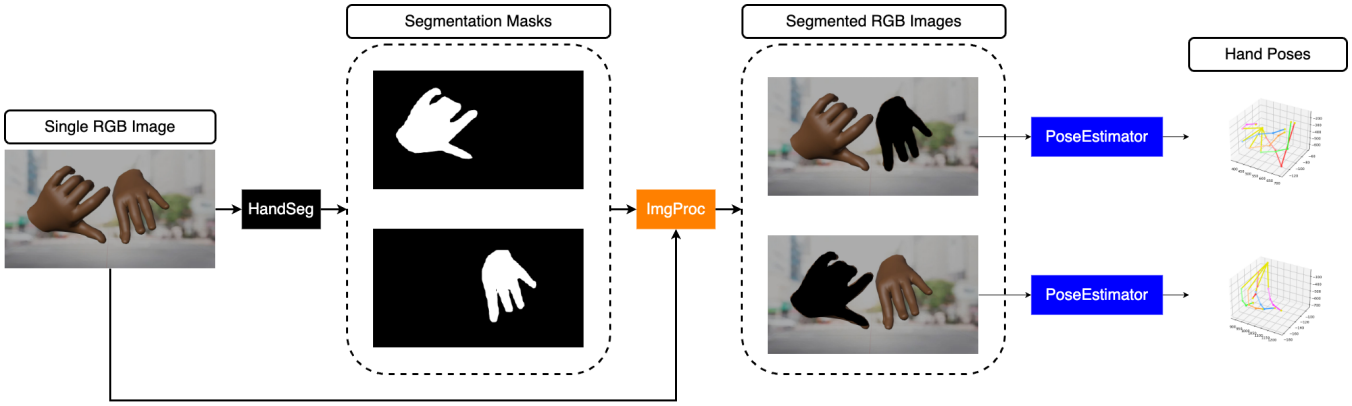


Fig. 2. Overview of the three-step pipeline of the Multiple Hand Pose Estimator. The HandSeg (Hand Segmentation) module takes a single RGB image and generates an individual segmentation mask for each hands. The ImgProc (Image Processing) module combines each segmentation mask with the original input single RGB image to generate a new set of segmented RGB images. Finally, the PoseEstimator (Hand Pose Estimator) module estimates the pose of each hand from the segmented RGB images.

stage, our **ImgProc** (Image Processing) module takes these individual masks, along with the original input image, and applies a multiplication operation between the mask and the original image's pixels to create a new set of RGB images that isolate each individual hand against a black background. In the final stage, our model inputs each RGB image from this set into a hand pose estimator. Our method is compatible with most existing hand pose estimators, as it generates a new set of RGB images through the ImgProc module.

B. Dataset Description and Data Collection

For hand pose estimation, we need large-scale datasets that can be either real or synthetic as long as there is estimated ground-truth joint locations. With the scarcity of two handed datasets with annotated ground-truth to train and test our approach, a synthetic dataset with enough variation and perfect annotation were introduced for advancing research on our method. Although our approach is designed to estimate 3D hand poses in a frame for a single hand, the new dataset is tasked to estimate the pose when there are two hands in the frame.

The dataset crafted for evaluating two-hand 3D pose estimation in a third-person view (shown in Figure 3) was generated utilizing 3D viewport rendering capabilities offered by Blender[2]. While Blender enables the extraction of segmentation masks for hand parts if desired, our primary objective was to leverage this dataset for testing purposes, benefiting from its annotated 2D and 3D joint locations. Employing the MANO[3] hand model code provided by Romero et al. [11], we dynamically generated random hand poses while recording the corresponding MANO hand parameters. Subsequently, utilizing the resultant random meshes, we placed them within Blender scenes against various backgrounds and captured renders from diverse camera angles. It's noteworthy that we maintained the rotations of the hand models consistent with the inputted MANO hand parameters, ensuring a standardized setup across the dataset.

[2]www.blender.org
[3]mano.is.tue.mpg.de

Fig. 3. This is example images taken from the synthetic dataset, with the three different backgrounds: City, Outside and Office. Taken from different angles: Front, Top and Angle. With different skin tones to change visibility.

Later this was scaled to have more hands in the frame to test mainly on occlusion and "stress-test" if the segmentation module can detect multiple hands. Shown in Figure 4. These set of images mainly tested hand detection, specifically to figure out how many hands are in the frame.

There were two synthetic datasets created. One was hand-made, and the other was programatically generated.

The second dataset was generated programatically using BlenderProc2 [12], a procedural Blender pipeline created specifically for realistic image generation in large numbers. To generate the dataset, a base *dataset_canvas.blend* Blender scene file was created. There, a set of modifiable camera and light positions, represented as armatures, were placed. The dataset featured a textured base mesh, extracted from the *dataset_canvas.blend* file, intended to resemble a red brick

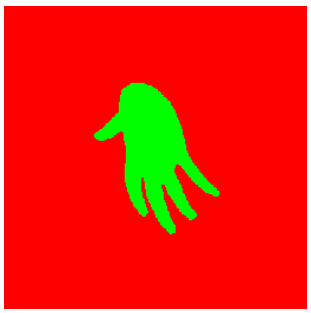
Fig. 4. This is images taken from the second iteration of the dataset, which involves having more hands in the image, with varying colours and sizes as well as rotations.

background. The code for dataset generation only imported built-in Python modules, with the exception to *Chumpy*[4], a library used by MANO, which was used by the code that generated MANO hand models. This was to lighten the load of installation to another system if another user wanted to generate the dataset for themselves. In the process of generating the dataset, the "Canvas_Camera" and "Canvas_Light" are loaded from *dataset_canvas.blend* and used as the camera and light. The camera positions, light positions and generated hand models are iterated over, saving each frame. As the number of images generated could reach up to thousands, constants were added to the top of the Python script run by BlenderProc2, limiting the number of generated images down to 64. To generate preview images on weaker machines, a 'fast' option was included, which changed the rendering settings to generate lower quality, lower resolution images much faster.

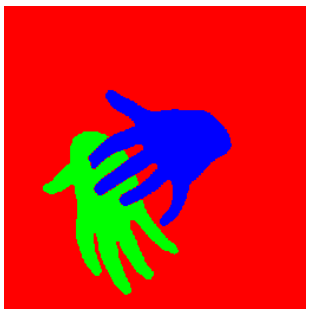
To train a reliable and accurate hand segmentation model that can distinguish and segment individual hands from others, we utilized the Amodal InterHand (AIH) dataset [10]. This dataset comprises approximately 3 million RGB images, each featuring interacting left and right hand poses with corresponding segmentation mask images. The dataset includes both "generated" and "rendered" hand poses. The term "generated hand poses" refers to a new dataset created by a simple copy-and-paste process from the Interhand2.6M dataset. This process preserves the detailed and realistic appearance information of the hand. On the other hand, "rendered hand poses" refer to textured 3D interacting hand meshes created using a rendering tool. Each label was annotated either by humans or a machine. The dataset has two types of segmentation masks: Amodal (x-ray vision to one of the hands) and visible (camera vision to both hands), as illustrated in Figure 5.

[4]https://github.com/mattloper/chumpy

(a) RGB image



(b) Amodal Segmentation mask



(c) Visible Segmentation mask

Fig. 5. AIH Dataset

However, we found out the generated hand pose images does not contain corresponding visible hand mask images. Since our goal for the the segmentation is to segment each individual hand that is only visible to the camera, we only had options to use the rendered hand pose RGB images with visible hand segmentation masks.

C. Training

We trained a hand segmentation model using the AIH dataset with the help of Detectron2 [13], a machine learning framework capable of training Mask R-CNN models. The training was conducted in a cloud linux environment provided by Kaggle [5], which only allowed for a single NVIDIA Tesla P100.

Due to the limited computational resources available for our research, we opted to select a subset of the dataset. This subset comprised 30,000 RGB images, each with segmentation label images where individual hand segmentation was annotated by a human.

We divided the 30,000 images from the dataset into two parts: 24,000 for training and 6,000 for validation. The training was carried out over 1000 iterations with a base learning rate of 0.001 and a batch size of 64. We performed a validation every 200 steps during training to test the performance of the hand segmentation model with unseen new data. This way of validation during training allow us check the performance of the hand segmentation without any biases.

D. Metrics + Benchmarks

In assessing the performance of our hand pose estimation model, we have implemented a robust evaluation framework that focuses on the accuracy of hand detection within images and the precision of hand type identification (distinguishing between left and right hands). We've scrutinized the effectiveness of model by conducting tests model outputs, each representing a different level of segmentation processing:

- **No Segmentation:** The first output variant operates without any segmentation, by directly putting the image through the pose estimator. This version serves as a baseline, allowing us to evaluate the model's performance compared to the Interhand pose estimator without any augementations.
- **Background Removal:** The second output involves a preprocessing step where everything in the image, except for the hand being pose-estimated, is masked out. This variant aims to determine the impact of removing background noise and distractions on the model's hand detection and type identification accuracy.
- **Excluding Other Hands:** The third output variant focuses on scenarios where multiple hands may be present in the image. In this version, all hands except the primary subject are masked out, leaving the main hand and the background intact. This setup tests the model's ability to concentrate on a single hand of interest amidst potential distractors.

V. Analysis of Results

Our analysis consists of two main segments. Initially, we undertook a quantitative analysis, adhering to the metrics and benchmarks outlined in our experimental design. Subsequently, we engaged in a qualitative review to assess the precision of the masking applied by the pose estimator model.

As outlined in the experimental design, our evaluations were conducted across three distinct model outputs: without segmentation, with background removal, and with other hands masked out but keeping the main hand and background. These tests aim to understand how different levels of segmentation affect the model's performance in real-world scenarios.

A. Quantitative Analysis

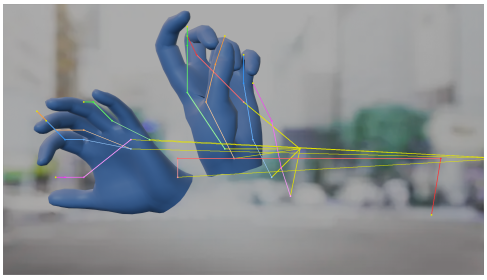
In the quantitative analysis section of our study, we evaluated the performance of our hand pose estimation model using our custom dataset. We selected 53 distinct images, each containing somewhere between 0 to 4 left or right hands. The diversity in hand position, orientation, and multiplicity made the dataset an especially challenging input for our model ensuring we highly scrutinise the model through the testing.

We perform two primary quantitative evaluations on our model outputs each corresponding to distinct aspects of the model:

Detected Masks Test: The initial part of our analysis delves into the model's proficiency in identifying hands within images, aiming to assess the effectiveness of our segmentation module in contrast with the raw capabilities of the Interhand model. As illustrated by the findings in Table I, our segmentation module demonstrates nearly double the efficiency in accurately detecting the number of hands present within an image. This marked improvement stems from the inherent limitations of the Interhand pose estimator, which can detect a maximum of two hands, even though up to four may be present in the input. Interestingly, about one-third of the tests without segmentation failed due to over-detections rather than under-detections. In contrast, all instances of failure in the tests with segmentation were attributed to over-detections, suggesting that our segmentation module might be overly sensitive.

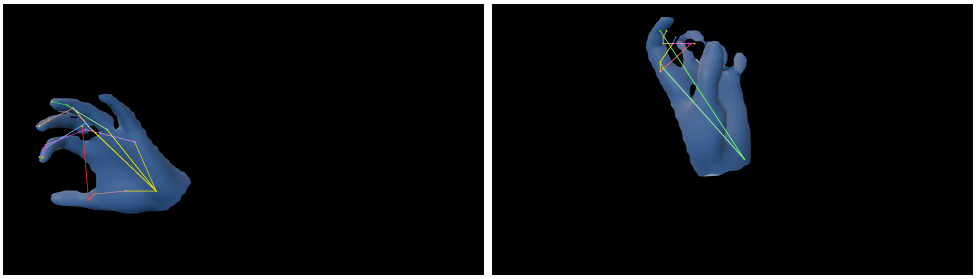
Hand Type Test: The second portion of our analysis assesses the model's capability to accurately classify each detected hand as either left or right. This evaluation aims to understand how the preprocessing masking component influences the subsequent pose estimator module's accuracy. According to the findings presented in Table II, omitting segmentation significantly affects the model's ability to discern between left and right hands, primarily due to its limitation in recognizing more than one hand of each type in an image. Additionally, we observed that selectively masking out extraneous hands—rather than eliminating the entire background—enhances hand type identification. This improvement is likely due to the preservation of additional contextual information in the image, which compensates for any data loss caused by imprecise masking. The criteria for a successful test involve correctly identifying the type of all hands present in an image. Given that our dataset includes images with up to four hands, this stringent requirement contributes to a notably low pass rate for such complex inputs. This difficulty in achieving a

[5]https://www.kaggle.com/code/harrynoh/detectron2-hand-segmentation/output



(a) No segmentation

(b) Excluding Other Hands



(c) Background removal

Fig. 6. Example results for two left hands. 6c

high pass rate underscores the challenge posed by images with multiple hands, highlighting areas for potential refinement in our model's hand type identification capabilities.

TABLE I
DETECTED MASKS TEST

Model output type	Tests passed /53
without segmentation	21
with background removal	37
with other hands masked out	37

TABLE II
HAND TYPE TEST

Model output type	Tests passed /53
without segmentation	4
with background removal	13
with other hands masked out	17

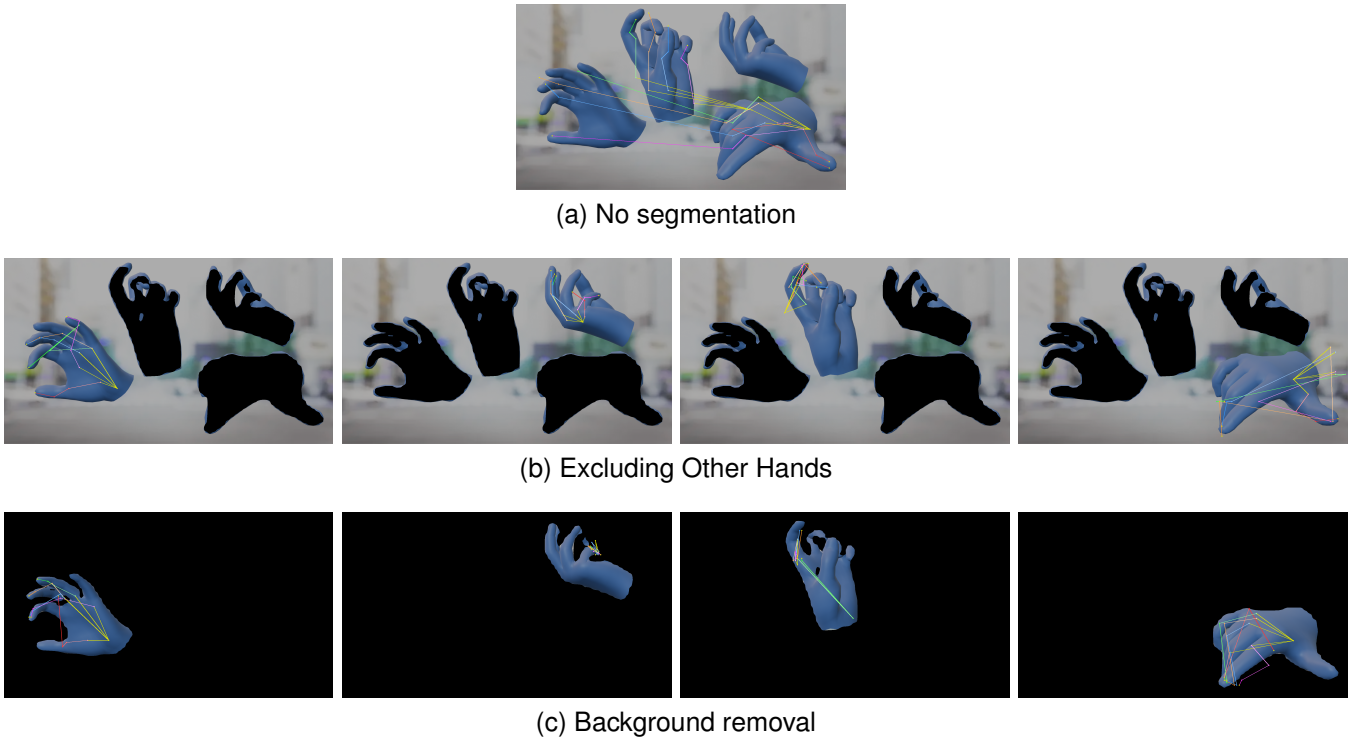
B. Qualitative analysis

For the qualitative analysis of our study, we use the same dataset to get the visualized version of the results data used in the qualitative analysis. There is clearly a performance benefit to using our system as opposed to using no segmentation and attempting to use a standard 3D pose estimator, as shown in the difference between the results of subfigures 6a and 7a and the other subfigures in figures 6 and 7. However, there is also a clear loss in adaptability between these figures as the system struggles to estimate more than two hands. Interestingly, both figures demonstrate that using background removal demonstrates slightly better results than excluding other hands. This is not a surprising result, given that the segmentation has been very effective and can therefore remove the unnecessary noise in the background without affecting the hand shapes. This would not be as certain if the hands had been physically interacting.

VI. DISCUSSION AND LIMITATIONS

A. Hand Segmentation

Our approach of incorporating a hand segmentation module successfully eliminated hand type dependencies. This allowed the existing hand pose estimator, previously limited to single hands or interacting pairs, to handle multiple hands. However, introducing hand segmentation creates a new dependency – the overall pose estimation heavily relies on the accuracy of hand



(a) No segmentation

(b) Excluding Other Hands

(c) Background removal

Fig. 7. Example results for two left hands and two right hands.

segmentation. We observed that segmentation failures, either missing hands entirely or inaccurately separating multiple hands, resulted in masked RGB images (generated by the ImgProc module) that still presented challenges for the final pose estimator.

Furthermore, computational resource limitations during training impacted the hand segmentation model's accuracy. We solely relied on the AIH dataset (containing only 30,000 samples) for training, which restricted the model's ability to handle diverse scenarios. Additionally, while we initially planned to utilize a synthetic dataset for both training the hand segmentation model and project evaluation, difficulties arose in creating ground truth data for hand poses within the synthetic dataset. Time constraints ultimately led us to utilize only the AIH dataset for hand segmentation training.

B. Dataset Creation

Initially, the plan for the synthetic dataset would have consisted of multiple animated scenes, featuring full human models engaging in various interactions. Those would have been rendered with different lighting angles and from multiple camera angles in order to render occluded hands. However, due to limited time, not all goals were reached fully.

The second synthetic dataset did not include backgrounds. The hands rendered were not textured. In contrast to the handmade synthetic dataset, they were not coloured either. The process of developing the code for dataset generation included learning and understanding not only how to use a given library, but also using Blender, as the initial *dataset_canvas.blend* file from which all features were loaded and used had to be configured using the Blender application.

The most underlying issue we came across was the compatibility of our inputs with those of the outputs from the pose estimator as the values didn't match up with what was originally inputted to generate the hand models. This led to not being able to get solid quantitative results and forced us to find a different route to test the approach of our method.

1) ***BlenderProc2****:* The second synthetic dataset used BlenderProc2 [12]. BlenderProc2 had multiple setbacks, such as an odd and restrictive API and system, which resulted in multiple workarounds and slowed down development progress. BlenderProc2 saved segmentation and depth rendered images as *.exr* files, which could only be viewed through Blender and was not compatible with the rest of experiment's framework, which needed *.png* images.

Finally, BlenderProc2 was limited to a specific version, *Blender 3.5*, which caused compatibility problems, as the creation of the initial canvas scene in *dataset_canvas.blend* required downgrading to an earlier version of Blender. Opening a *.blend* file saved in *Blender 4.0* using *Blender 3.5* resulted in a crash.

2) ***Alternatives****:* In the process of research, alternative options were briefly explored, which could have been used, but were abandoned. Two examples are DART [14] and *manotorch*[6] [15]. Both are to be elaborated in the conclusion, under *Future Works*.

For example, DART's GUI tools were tailored for Unity, which provided a roadblock towards use with Blender, as we desired a simple, straightforward pipeline. On the other hand,

[6]https://github.com/lixiny/manotorch

the hand models generated by DART were in the form of *.obj*, *.mtl* and *.png* file pairs, which were loadable through BlenderProc2.

VII. Conclusion

We have presented a novel framework for estimating the pose of multiple arbitrary left or right hands via segmentation and removal, as well as a new synthetic dataset generator able to create images suitable for training and validation of such systems. The system has demonstrated the ability to create segmentation masks for multiple hands in a given input image, regardless of whether the image comprises right hands, left hands, or both hand sides, in addition to estimating the pose of those segmented hands.

A. Future Works

There are multiple limitations that can be addressed in future works. Those include both the development of frameworks as well as changes to the methodology that we used in our paper.

Limitations of the system are present in the areas of hand segmentation, and more prominently output and quantitative evaluation of pose estimation, meaning there is scope for further improvement of this method in these aspects. Some examples are training the hand segmentation network on larger and more varied datasets, and adjustment of the pose estimator for validation, both against the synthetic dataset and on others. Finally, there is potential for the synthetic dataset to be used in training of a future dedicated pose estimator, which could be developed separately, or for use within our wider framework.

1) Synthetic Dataset: The synthetic dataset itself could be improved in multiple ways.

First, an alternative, more compatible code base for the MANO model could be used. One example is *manotorch*[7], developed by Yang et al. [15], a pyTorch layer that uses a set of shape and pose parameters in order to output a model that contains a set of 21 joints in 3-dimension form. In future works, this could be used in training the Hand Pose Estimator in our methodology, as it uses a set of 42 3-dimension parameters, representing 21 joints each for two hands.

Next, an alternative framework using MANO hand model parameters could be used. One example is DART [14], a framework that extends MANO, introducing a detailed, textured hand model, which could be customised with varying skin colours and textures. This provides the ability to handle racial bias in hand pose prediction and could be used to generate a synthetic dataset that is visually closer to real-life hands and improve performance for real-life data.

Finally, there is a demand for a framework that could stitch bodies from SMPL-X[8], developed by Pavlakos et al. [16], together with hands from DART. This would make our initial goals with our synthetic dataset easier to achieve.

[7]https://github.com/lixiny/manotorch

[8]https://smpl-x.is.tue.mpg.de/

Finally, there is a demand for a framework that could stitch bodies from Meshcapade[9], using SMPL-X[10], developed by Pavlakos et al. [16], together with hands from DART. This would make our initial goals with our synthetic dataset easier to achieve.

References

[1] Gyeongsik Moon, Shoou-i Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. *InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image*. 2020. arXiv: 2008.09309 [cs.CV].

[2] Francisco Gomez-Donoso, Sergio Orts, and Miguel Cazorla. "Robust Hand Pose Regression Using Convolutional Neural Networks". In: Jan. 2018, pp. 591–602. ISBN: 978-3-319-70832-4. DOI: 10.1007/978-3-319-70833-1_48.

[3] 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.

[4] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. "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. 2018, pp. 436–445.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime multi-person 2d pose estimation using part affinity fields". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7291–7299.

[6] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. "3d hand shape and pose estimation from a single rgb image". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10833–10842.

[7] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. "Aligning latent spaces for 3d hand pose estimation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 2335–2343.

[8] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. "End-to-end hand mesh recovery from a monocular rgb image". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2354–2364.

[9] Javier Romero, Dimitrios Tzionas, and Michael J Black. "Embodied hands: Modeling and capturing hands and bodies together". In: *arXiv preprint arXiv:2201.02610* (2022).

[10] Hao Meng, Sheng Jin, Wentao Liu, Chen Qian, Mengxiang Lin, Wanli Ouyang, and Ping Luo. *3D Interacting Hand Pose Estimation by Hand De-occlusion and Removal*. 2022. arXiv: 2207.11061 [cs.CV].

[9]https://meshcapade.wiki/SMPL

[10]https://smpl-x.is.tue.mpg.de/

[11] Javier Romero, Dimitrios Tzionas, and Michael J. Black. “Embodied Hands: Modeling and Capturing Hands and Bodies Together”. In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*. 245:1–245:17 36.6 (Nov. 2017).

[12] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. “BlenderProc2: A Procedural Pipeline for Photorealistic Rendering”. In: *Journal of Open Source Software* 8.82 (2023), p. 4901. DOI: 10.21105/joss.04901. URL: https://doi.org/10.21105/joss.04901.

[13] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[14] Daiheng Gao, Yuliang Xiu, Kailin Li, Lixin Yang, Feng Wang, Peng Zhang, Bang Zhang, Cewu Lu, and Ping Tan. “DART: Articulated Hand Model with Diverse Accessories and Rich Textures”. In: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2022.

[15] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. “CPF: Learning a Contact Potential Field to Model the Hand-Object Interaction”. In: *ICCV*. 2021.

[16] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. “Expressive Body Capture: 3D Hands, Face, and Body from a Single Image”. In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10975–10985.