Error Estimation for Single-Image Human Body Mesh Reconstruction

by

Hamoon Jafarian

A thesis submitted to the
School of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Masters of Science in **Computer Science**

Faculty of Science
University of Ontario Institute of Technology (Ontario Tech University)
Oshawa, Ontario, Canada
May 2023

THESIS EXAMINATION INFORMATION

Submitted by: **Hamoon Jafarian**

Master of Science in Computer Science

Thesis Title: Error Estimation for Single-Image Human Body Mesh Reconstruction

An oral defense of this thesis took place on May 18^{th}, 2023 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Khalil El-Khatib
Research Supervisor	Dr. Faisal Qureshi
Examining Committee Member	Dr. Andrew Hogue
Thesis Examiner	Dr. Peter Lewis

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Human pose and shape estimation methods continue to suffer in situations where one or more parts of the body are occluded. More importantly, these methods cannot express when their predicted pose is incorrect. This has serious consequences when these methods are used in human-robot interaction scenarios, where we need methods that can evaluate their predictions and flag situations where they might be wrong. This work studies this problem. We propose a method that combines information from OpenPose and SPIN—two popular human pose and shape estimation methods—to highlight regions on the predicted mesh that are least reliable. We have evaluated the proposed approach on 3DPW, 3DOH, and Human3.6M datasets, and the results demonstrate our model's effectiveness in identifying inaccurate regions of the human body mesh.

Keywords: human mesh recovery; human pose and shape estimation; OpenPose; SPIN; error estimation;

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I authorize the Ontario Tech University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize the Ontario Tech University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Hamoon Jafarian

Statement of Contributions

I hereby certify that I am the sole author of this thesis and I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Faisal Qureshi, for his guidance, encouragement, and support throughout my research journey. His invaluable insights, constructive feedback, and unwavering dedication have been instrumental in shaping this thesis.

I would also like to extend my appreciation to my lab colleagues for their invaluable support and collaboration. Their valuable input, discussions, and feedback have greatly enriched my research experience and enabled me to broaden my horizons.

Finally, I would like to thank my family for their support, and encouragement. Their constant encouragement and belief in me have been a constant source of motivation and inspiration, and I am forever grateful for their unwavering support.

Contents

List of Tables

List of Figures

List of Symbols

c	Line intersect
ED	Estimation difference
f_{MC}	Mesh classifier function
f_{WJC}	Worst joint classifier function
gt	Ground truth
i	Image index
k	Joint index
K	Total number of joints
k_{worst}	Index of the least reliable joint
m	Pixel location/Line gradient
MPJPE	Mean per joint position error
n	Pixel location
OP	OpenPose model
r	Pearson Correlation Coefficient
SE	SPIN model error
x	2D coordinates
X	3D coordinates
y_{mesh}	Mesh label
$\mathtt{R}$	The set of all real numbers
$\|a - b\|$	Euclidean distance between a and b
$a \in A$	Element a is in set A

Chapter 1

Introduction

The applications of Human Body Mesh Recovery (HMR) are diverse and numerous. HMR, for example, is useful for Human-Robot Interaction (HRI) scenarios, where accurate 3D mesh representation is essential for ensuring safe interactions [12]. Drones, self-driving cars, and human-robot collaborative manufacturing systems are some examples where a three-dimensional understanding of the environment and humans is critical for reliable operation [13]. Additionally, the animation and movie industries can benefit significantly from HMR by simplifying the process of character motion capture (MOCAP) and reducing the costs involved [14]. Other areas such as part and foreground segmentation, computer-assisted coaching, and virtual try-on can also leverage the capabilities of 3D mesh recovery to enhance their outcomes [6]. Figure 1.1 illustrates some of these applications.

The task of estimating a human body mesh from a single RGB image is an active area of research that has garnered significant interest in the field of computer vision. Kolotoures and colleagues [15] proposed SPIN that achieves impressive results on single-image human body mesh recovery. SPIN represents a significant improvement in human pose and shape estimation over prior methods, and it is now a widely adopted baseline in the field. A number of recent methods attempt to recover human body mesh in the

Figure 1.1: HMR Applications. From left to right: Personal service robot [1], Self-driving cars [2], Dynamic simulations during swimming [3]

presence of occlusions [8, 9, 16]. None of these methods, however, provide a confidence score for the recovered mesh. The ability to tell whether or not the recovered mesh is correct or to identify parts of the mesh that may be inaccurate is particularly relevant for human-robot interaction scenarios. A robot, for example, can choose to halt its operation if it deems that the recovered mesh is not reliable. Alternatively, a robot may adjust its viewpoint to achieve a better reconstruction if it identifies one or more parts of the mesh to be unreliable.

Here we tackle the problem of estimating the error in the reconstructed human body meshes. We propose a method that fuses information from SPIN and OpenPose [4] to highlight regions of the recovered mesh that *may* be inaccurate (Figure 1.2). OpenPose estimates human joints' keypoints, and it is able to identify joints that are not visible in the image. The proposed method leverages the observation that SPIN and OpenPose agree when the person is visible in the image; whereas, these two methods disagree when the person is partially occluded. We have used *sensitivity analysis* to quantify the disagreement between SPIN and OpenPose models under occluded settings. The differences between the joints' keypoints estimated by OpenPose and those constructed by projecting the human body mesh recovered by SPIN are fed into two multi-linear perceptron networks to compute an error estimate for each region of the mesh.

Previous models, such as SPIN, mostly focus on estimating the mesh and do not provide further information regarding the quality of the estimated mesh or its reliability.

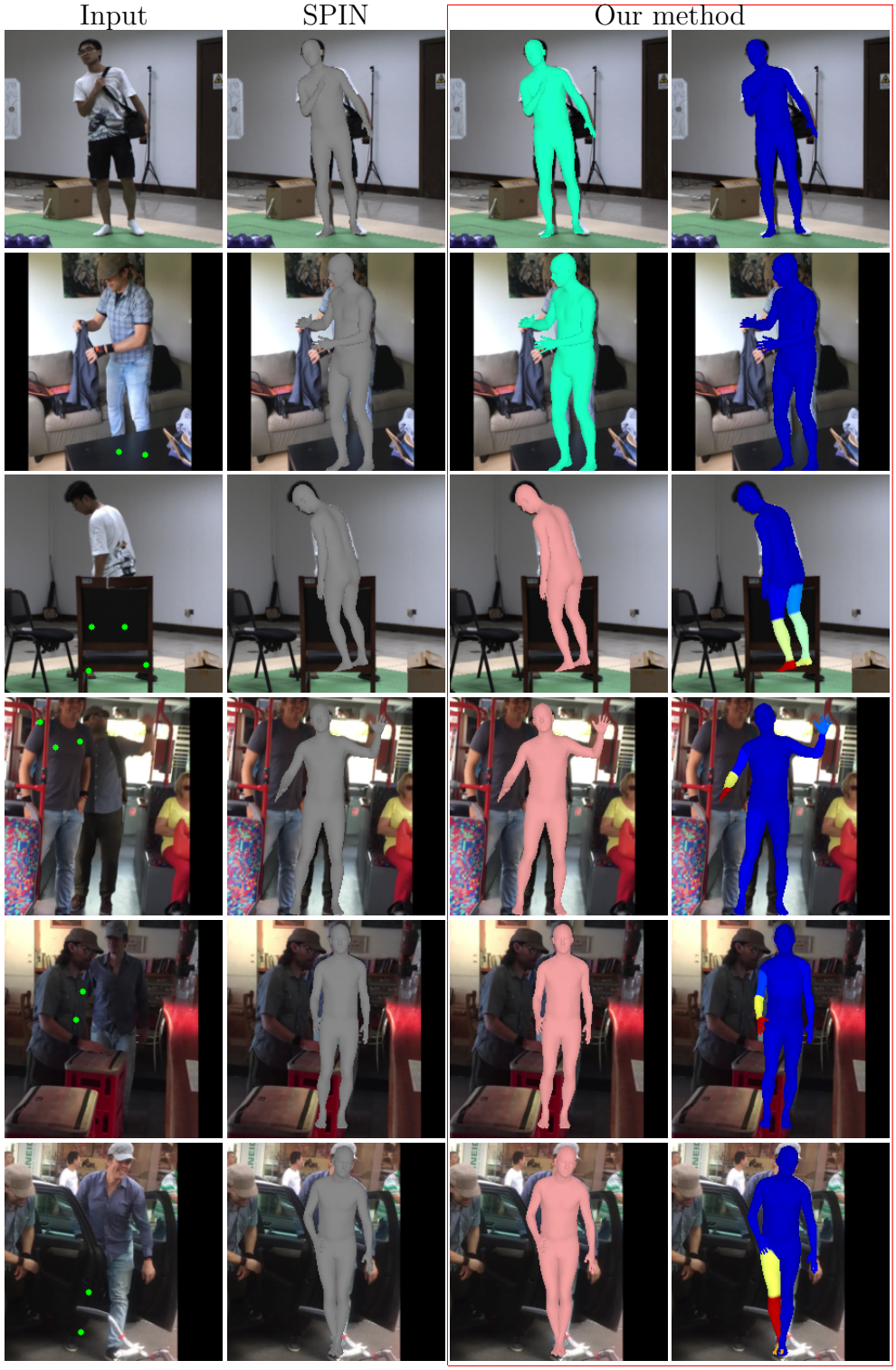

Figure 1.2: Human Body Mesh Recovery, the SPIN Model vs. Our Method. The input images are in the first column; the recovered meshes using SPIN are shown in the second column. Our model incorporates an error estimation for the estimated mesh, enabling us to classify meshes as either accurate or inaccurate (third column) and identify the least reliable regions of the mesh (fourth column) as well.

However, our proposed method offers an error estimation for the mesh. Our model is able to classify accurate and inaccurate meshes and highlight the least reliable parts of the mesh. In Figure 1.2 (last column from right), regions shown in red depict mesh parts with the lowest reliability. Note that these regions correspond to the parts of the human body that are not visible in the image. For instance, in the last row, the subject's right foot is occluded, and the SPIN model generates an inaccurate mesh estimate. However, there is no more information about the quality of the estimated mesh. Our model detects a fault and classifies the mesh as inaccurate (pink mesh). Moreover, our model highlights the right foot as the least reliable part of the mesh.

1.1 Contribution of This Work

To the best of our knowledge, this work represents the first attempt at estimating error in single-image 3D human body mesh reconstructions. The contributions of the work presented here are:

- Location-based and joint-based occlusion sensitivity analysis to quantify the relationship between the disagreement of OpenPose and SPIN joint location estimates and the "true" error.

- A mesh classifier that identifies whether or not the recovered mesh is reliable.

- A worst joint classifier that selects the least reliable joint.

This work represents a significant step towards improving the safety and reliability of those human-robot interactions that rely upon accurate reconstructions of human body mesh by providing additional information about the confidence and reliability of the estimated mesh.

1.2 Thesis Outline

The remainder of this thesis is organized as follows.

In **Chapter 2**, we review the previous related research and discuss the benefits and drawbacks of recent works. Then, we state how our work will contribute to solving the active problems in Human Mesh Recovery.

Chapter 3 is dedicated to sensitivity analysis. We employ different approaches to investigate the SPIN and OpenPose models' behavior against occlusion. The objective is to reveal the distinctive performance of each model in occluded cases.

In **Chapter 4**, we demonstrate the available potential of error estimation based on the Openpose and SPIN models fusion. Then, we describe our proposed methods for error estimation in detail.

Chapter 5 illustrates the performance of our model. We present quantitative and qualitative evaluations of our model. Moreover, we show some examples of real-world applications of our proposed model.

Chapter 6 summarizes our research. Achievements and future work are discussed afterward. We restate the steps we went through to generate an error estimation for the SPIN model.

1.3 Software, Open Data, and Source Code

1.3.1 Software

We chose Python as the programming language for implementation. Python is free, and also supports a wide range of open-source packages, making it useful for mathematical applications as well as image processing, computer vision, and machine learning.

Dealing with large datasets and deep neural networks is the main focus of this project. These tasks require considerable computational power. Hence, most of our computations

are performed on GPUs (Graphics Processing Units), and we utilize CPU (Central Processing Unit) hyper-threading to pre-process and render datasets efficiently. The following open-sourced Python packages were used in this research.

- **PyTorch** is an open-source deep learning platform Python package that provides support for tensor computation with strong GPU acceleration, and neural networks on a tape-based autograd system. `https://github.com/pytorch/pytorch`
- **OpenCV** (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library that can take advantage of multi-core processing and hardware acceleration. `https://opencv.org/`
- **Pyrender** is a pure Python library for physically-based rendering and visualization. It comes packaged with both an intuitive scene viewer and a headache-free offscreen renderer with support for GPU-accelerated rendering on headless servers, which makes it perfect for machine-learning applications. `https://pyrender.readthedocs.io/en/latest/`
- **Trimesh** is a pure Python library for loading and using triangular meshes. The goal of the library is to provide a full-featured and well-tested Trimesh object which allows for easy manipulation and analysis. `https://trimsh.org/index.html`
- **Scikit-image** is a collection of algorithms for image processing written by an active community of volunteers. `https://scikit-image.org`
- **NumPy** is an open-source Python package, that adds support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. `http://www.numpy.org`

1.3.2 Open Data

We evaluate our proposed models on the following publicly available standard datasets.

- **3DPW** [17]. The "3D Poses in the Wild dataset" is the first dataset in the wild with accurate 3D poses for evaluation. `https://virtualhumans.mpi-inf.mpg.de/3DPW`

- **Human3.6M** [18]. A dataset with 3.6 million 3D human poses and corresponding images including 11 professional actors (6 male, 5 female) and 17 scenarios (discussion, smoking, taking a photo, talking on the phone...). `http://vision.imar.ro/human3.6m`

- **3DOH50K** [8]. This dataset contains more than 51600 images, where all images were captured from real scenes with 6 viewpoints. `https://www.yangangwang.com`

1.3.3 Source Code

The Python implementation of our models, evaluation metrics and pre-trained models can be accessed through the following link.

`https://github.com/Hamoon1987/meshConfidence`

Chapter 2

Related Works

In this chapter, we provide a comprehensive review of the related research in Human Mesh Reconstruction. We categorize the literature into three main categories: 2D keypoint estimation, 3D pose and shape estimation, and occlusion handling. For each category, we investigate the various techniques employed by researchers, highlighting the advantages and shortcomings of the models. We also identify the most commonly used datasets in this field based on our review of previous works. Finally, We discuss how our work adds to the previous research.

2.1 2D Keypoint Estimation

2D keypoint estimation aims to localize body joints within an image. Joints' keypoint estimation comes in two flavors: regression-based methods [19, 20, 21] and detection-based methods [22, 23]. Regression-based approaches aim to predict the exact positions of body joints by directly regressing their coordinates. On the other hand, detection-based methods employ a two-stage process to estimate the joint positions. In the first stage, the model predicts the probability that each pixel in the image corresponds to a joint. In the second stage, the model refines the joint positions based on spatial information. When dealing with multi-person scenarios, top-down approaches typically achieve higher

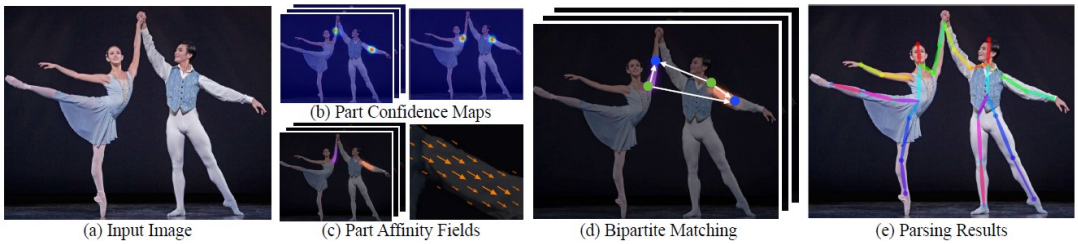

Figure 2.1: Overview of the OpenPose Framework. A two-branch CNN is employed to receive the input image (a) and generate two heatmaps shown in (b) part confidence and (c) affinity field. Afterwards, in the bipartite stage, a score is calculated for each pair of the detected body parts. Finally, in the last step (d), full body poses are estimated [4].

accuracy compared to bottom-up methods. However, bottom-up methods are often faster and more suitable for real-world applications that require real-time performance [24]. In top-down approaches, a human detector is used to locate individuals in the image, followed by a single-person pose estimator to estimate the pose of each individual. On the other hand, in bottom-up approaches, the first step involves detecting and localizing the joints of all persons in the image. The second step then focuses on associating the detected joints with their corresponding person instance.

Pishcgulin et al. [25] proposed DeepCut, which jointly solves detection and pose estimation tasks utilizing a CNN-based body part detector. Insafutdinov et al. [26] improved the DeepCut model by replacing the manually calculated features with an extremely deep part detector based on ResNet. This detector generates body part proposals, which are refined to obtain accurate joint locations. Cao et al. [4] introduced Part Affinity Fields (PAFs) that encode the position and orientation of human body parts and propose OpenPose, an accurate, fast, and robust model for multi-person joints' keypoints estimation. As a part of our model, we incorporate the OpenPose model. OpenPose uses a bottom-up approach to detect keypoints, first detecting all body parts in the image and then associating them with specific individuals.

Figure 2.1 demonstrates the overall OpenPose model's framework. In the first step, 2D part confidence maps (one for each part) and 2D vector fields (one for each limb) are predicted using a feed-forward network. Each confidence map depicts the expectation

that a specific body part will appear at each pixel point. 2D vectors, called affinity fields, encode the direction that points from one part of the limb to the other. PAFs are used in the next stage to assign a score to each candidate limb (part pair). Finally, a novel optimization scheme is employed to assemble the full-body poses of multiple people.

2.2 3D Pose and Shape Estimation

Broadly speaking, 3D pose and shape estimation methods are divided into two classes: optimization-based methods that deform a canonical pose to match the image [27, 28, 29] and regression-based methods that directly estimate the mesh from the image [10, 30, 31]. Optimization-based methods achieve good results; however, these are slow and require careful initialization. Conversely, regression-based methods are difficult to train to attain high-quality meshes [6]. The HMR model proposed by Kanazawa et al. [10] is widely known as the regression-based methods' backbone. Hence, we have provided a detailed review of the paper in Appendix A. Kolotoures et al. [13] present the SPIN model that forms a strong collaboration between the two paradigms to benefit from both approaches. The SPIN model employs optimization to provide explicit 3D supervision to train a regressor to construct high-quality meshes. The SPIN model has become a widely adopted baseline method in this field of study, and we also incorporate it in our model.

Figure 2.2 illustrates the overview of the SPIN model. In the training stage, images are passed through a deep neural network to predict the initial SMPL [5] parameters (Θ_{reg}). Then, an optimization-based method called SMPLify [27] is utilized to improve the initial prediction (Θ_{opt}). This process enables us to access a more effective loss function $L_{3D} = \|\Theta_{reg} - \Theta_{opt}\|$. Previously, a weak supervision based on the difference between the estimated 2D joint coordinates and the ground truth was used; however, using SMPLify, a 3D supervision is provided, which is much more powerful. This creates

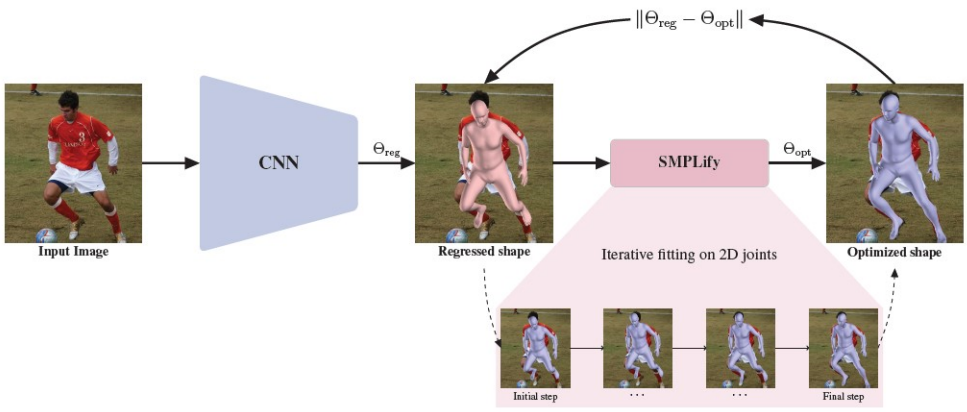

Figure 2.2: Overview of the SPIN Framework. Within the training loop, initial SMPL parameters [5] are predicted by the CNN. Afterward, SMPLify, an iterative optimization algorithm, is employed to improve the initial estimate. Now, we have more effective 3D supervision to enhance the CNN. Moreover, a better initial prediction would lead to more efficient optimization. This would form a self-improving loop which results in a more accurate model [6].

a self-improving loop, resulting in more accurate 3D mesh generation. In each iteration, the regressor improves since more efficient supervision is provided. Also, the optimizer with a better initialization Θ_{reg} would provide a more accurate Θ_{opt}.

Hybrid models achieve state-of-the-art performance. These benefit from the 3D pose and shape estimation models' ability to capture the realistic body structure and combine it with the higher accuracy of keypoint estimation models. Li et al. [7] proposed a model called HybrIK that consists of two main components: a regression model that estimates the body structure and a 3D keypoint predictor that calculates the final position of the joints. HybrIK uses an inverse kinematic solution to link the two components and enable them to be trained simultaneously. Specifically, the regression model provides information on the body structure, including bone length and the 1D twist of each joint. The 3D keypoint predictor then determines the final position of each joint. The HybrIK component calculates the swing of each joint based on this information, producing an output of Θ, which is the input of the SMPL model for mesh generation. However, the paper does not investigate the model's performance when dealing with occlusions. Figure 2.3 depicts the general overview of this model.

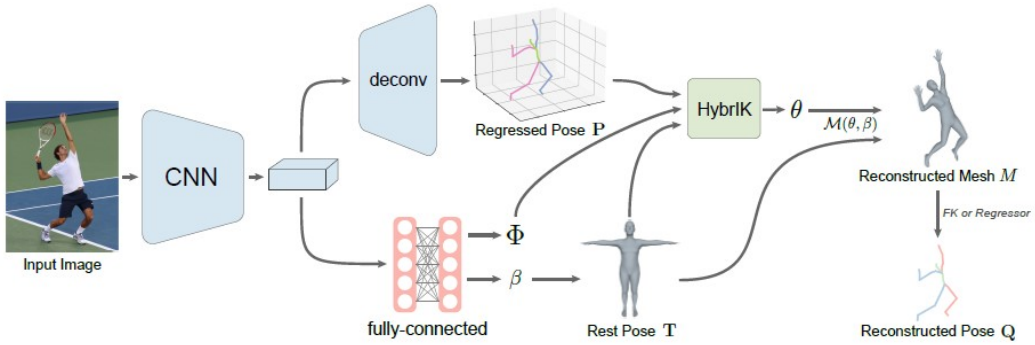

Figure 2.3: Overview of the HybrIK Framework. Following feature extraction from the input image, the proposed method employs the deconvolutional layers to estimate the 3D joint positions. Additionally, fully connected layers are utilized to estimate the joints' shape parameters and twist angles. These outputs are then used in the HybrIK process to determine the relevant SMPL parameters required to generate the 3D mesh [7].

Iqbal et al. [32] propose a similar approach. KAMA method [32] integrates a 3D heatmap-based keypoint estimation module and a body mesh regression module. The 3D keypoint module estimates the 3D joint locations by generating a 3D heatmap, which encodes the likelihood of the presence of each keypoint at each voxel in a 3D space. The body mesh regression module predicts the body shape and mesh articulation from the estimated 3D keypoints. The two modules are jointly trained using a multi-task loss function, which considers both the keypoint estimation and body mesh reconstruction tasks.

2.3 Occlusion Handling

Inspired by random erasing [33] and synthetic occlusion [34] techniques exploited in classification and object detection tasks, some researchers suggest that data augmentation could be a suitable solution against occlusion. In this scenario, the images are occluded throughout the training process, and the model is taught to perform better against occlusion [35, 36]. Others modified the model architecture to improve the model's robustness against occlusion [8, 9, 16].

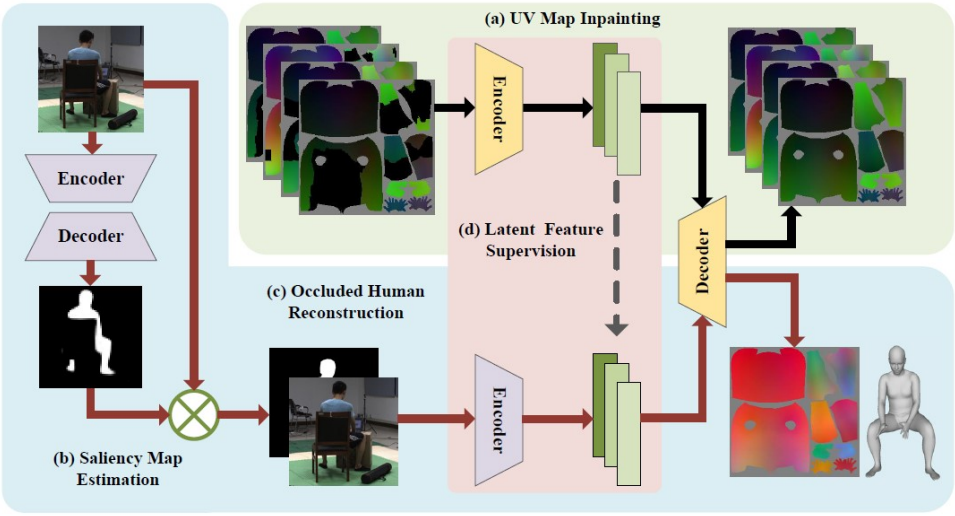

Figure 2.4: Model Framework. The proposed network consists of two branches. During training, the UV map inpainting branch (a) is trained first, followed by the color image encoder branch (c), which takes in the occluded color image concatenated with its saliency map (b). The corresponding partial UV map is encoded by a fixed inpainting network and used to supervise the color image encoder in the latent space (d). During inference, a single color image is passed through the saliency map sub-network (b) and the occluded human reconstruction sub-network (c). Finally, the output mesh is resampled directly from the UV position map [8].

Zhang et al. [8] use a partial UV map model to convert the occluded human body mesh reconstruction to an image inpainting problem. Figure 2.4 illustrates the process. First, partial UV maps of input images are calculated through a separate process using the available ground truth 3D mesh. Then, we train an encoder-decoder as shown in part (a) in Figure 2.4 using the obtained partial UV maps from the previous step. Afterward, the input image is concatenated with its saliency map and fed to the encoder as shown in Figure 2.4 with the blue background. We use the latent features of the previous step to supervise the bottom branch. In other words, in this method, we try to teach the model to receive the input image and generate latent features similar to those generated from the UV map input in the top branch. Since the decoder is the same in both branches, we can now expect to predict accurate partial UV maps from the input image using only the bottom branch.

Wang et al.[37] also exploit a UV inpainting module in their three-staged model.

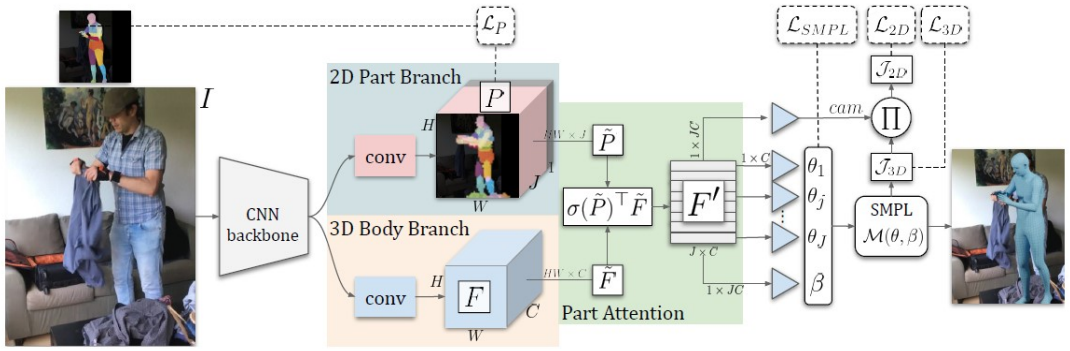

Figure 2.5: PARE Model Architecture. Features are extracted from the input image in two separate branches. In part attention (green box), the 2D part branch works as a soft attention mask to emphasize the essential features of the 3D body branch output. Hence, when parts are visible, the model focuses on the part, and in case of occlusion, the model looks for other helpful clues, including other visible parts. [9].

Combining a dense map prediction, inverse kinematic, and a UV inpainting module, their framework leverages the best of non-parametric and model-based methods and is also robust to partial occlusion.

Georgakis et al. [38] develop a prior-informed regressor that knows the hierarchical structure of the human body, and the experiments show that this method improves the model performance against occluded cases. Kocabas et al. [9] implemented the soft attention mechanism for the HMR problem, resulting in a considerable improvement of the model's robustness against occlusion. The developed part attention regressor (PARE) learns to rely on visible body parts to reason about the occluded parts. Figure 2.5 shows the PARE's overall framework. The model comprises two branches: the top branch generates part attentions, while the bottom branch estimates the 3D body parameters. Each part attention contains high-value pixels at the corresponding part location. When a part is occluded, the attention leverages other informative regions in the image. The two branches are combined using a fusion block (green box) that aggregates features in the 3D body branch utilizing the 2D part branch as a soft attention mask. The proposed method achieves state-of-the-art results on several benchmark datasets, demonstrating its effectiveness and robustness in handling complex real-world scenarios.

Dataset	3DPW	H36M	3DOH
Sarandi et al. [35]	-	✓	-
Kocabas et al. [9]	✓	✓	✓
Li et al. [7]	✓	✓	-
Wang et al. [37]	-	✓	✓
Georgakis et al. [38]	-	✓	-
Zhang et al. [8]	✓	✓	✓
Yang et al. [39]	✓	-	✓
Our work	✓	✓	✓

Table 2.1: Common Datasets in Previous Research.

2.4 Datasets

We chose 3DPW [17], Human3.6M [18], and 3DOH50K [8] datasets which are the most common ones in this field as shown in Table 2.1. Human3.6M and 3DPW datasets are employed in both the training and testing stages while 3DOH is only used to test the model.

Human3.6M: It is an indoor benchmark for 3D human pose estimation. It includes multiple subjects performing actions like Eating, Sitting and Walking. Following typical protocols, e.g., [10], we use subjects S9 and S11.

3DPW: It is a very recent dataset, captured mostly in outdoor conditions, using IMU sensors to compute pose and shape ground truth. We use this dataset for both the training and testing stages.

3DOH50K: This dataset was formed to compensate the fact that the most of existing 3D human datasets often overlook the occlusions generated by the interactions between the human and objects. It contains images of human activities in occlusion scenarios. All images are captured from real scenes with six views. 3DOH50K is the first real 3D human dataset for the problem of occlusion.

Type	Method	Year	Ref.
2D Keypoint Estimation			
Regression-based	Part detection & contextual information comb.	2019	[20]
Regression-based	Distribution-aware coordinate representation	2020	[21]
Detection-based	Two-stage normalization scheme	2017	[22]
Detection-based	Adversarial posene (Discriminator)	2017	[23]
Top-down	Multi-stage pose network	2019	[40]
Top-down	Pose graph convolutional network	2020	[41]
Bottom-up	Part affinity fields (OpenPose)	2017	[4]
Bottom-up	Disentangled keypoint regression	2021	[42]
3D Pose and Shape Estimation			
Optimization-based	Multiple scene constraints	2018	[28]
Optimization-based	3D keypoint aware mesh articulation (KAMA)	2021	[32]
Optimization-based	Pose and shape refinement (Skeleton2Mesh)	2021	[43]
Regression-based	SMPL optimization in the loop (SPIN)	2019	[15]
Regression-based	Analytical-neural inverse kinematics (HybrIK)	2021	[7]
Regression-based	Transformer (FastMETRO)	2022	[44]
Occlusion Handling			
Data augmentation	Keypoint masking	2018	[36]
Data augmentation	Artificial and synthetic occlusion	2018	[35]
Architecture altering	Hierarchical kinematic mesh recovery	2020	[38]
Architecture altering	Partial UV map	2020	[8]
Architecture altering	Deep UV Prior (Pose2UV)	2022	[45]
Architecture altering	Synthetic Occlusion-Aware Data (LASOR)	2022	[39]
Architecture altering	Model-based and nonparametric methods comb.	2022	[37]
Architecture altering	Contextual Normalization (CoNorm)	2022	[16]
Architecture altering	Part Attention Regressor (PARE)	2021	[9]

Table 2.2: Related Works Summary.

2.4.1 Related Works Summary

Table 2.2 shows a summary of related works. Previous research in the field of 3D human body estimation has been primarily concerned with enhancing the accuracy of estimated meshes or keypoint predictions, as well as improving the model's capacity to handle occlusions. However, we take a unique approach by concentrating on providing confidence levels for the reconstructed mesh. We strive to generate a reliability distribution that can enable machines to operate safely around humans, even if the mesh estimation is imperfect. To the best of our knowledge, this research represents the first attempt to address this particular issue.

Chapter 3

Occlusion Sensitivity Analysis

We experiment with two approaches to visualize and understand the effects of partial occlusions of the human body on the performance of SPIN and OpenPose models. The first approach captures the sensitivity (of both methods) to occluded regions for a given image. The second approach, on the other hand, shows the sensitivity to an occluded joint over the entire dataset.

3.1 Location-based Sensitivity Analysis

The first approach is inspired by [46, 9], where a square occluder is pasted onto different pixel locations in the image. Both the size and the stride of the occluder can be changed. Similar to [46], we use a grey colored square. To ensure an effective investigation of occlusion effects, it is crucial to choose an appropriate square size. Small square sizes do not affect the model's performance and are not suitable for sensitivity analysis. Large square sizes, on the other hand, would lead to high errors regardless of the occluder's position and do not properly distinguish the relative importance of different regions. In this study, we opted for a square occluder size of 20 by 20 pixels, striking a balance that allows for an accurate assessment of occlusion effects while maintaining the relative significance of different regions in the analysis. The occluded images are passed to SPIN

and OpenPose models and the errors are recorded. The performance of both models is measured using the Mean per Joint Position Error (MPJPE) that is defined as the mean value of the Euclidean distance between the ground truth and the predicted locations of all the joints. SPIN model recovers an SMPL mesh M. Using a pre-trained regressor W, it is possible to estimate 3D joint locations $X = WM$, where $X \in \mathtt{R}^{K\times 3}$. $K = 14$ refers to the number of joints. MPJPE for SPIN model is

$$\text{MPJPE}^{\text{SPIN}}_{(m,n)} = \underset{k}{\text{Mean}} \, \|X_{(m,n)} - X^{\text{gt}}\|. \tag{3.1}$$

Here $X_{(m,n)}$ denotes 3D joint locations when occluder is centered at location (m, n). X^{gt} denotes ground truth 3D joint locations. For the OpenPose model, which estimates 2D joint locations $x \in \mathtt{R}^{K\times 2}$,

$$\text{MPJPE}^{\text{OP}}_{(m,n)} = \underset{k}{\text{Mean}} \, \|x_{(m,n)} - x^{\text{gt}}\|, \tag{3.2}$$

where $x_{(m,n)}$ and x^{gt} are 2d joint estimates when occluder is centered at (m, n) and ground truth 2d locations, respectively.

Figure 3.1 plots MPJPE scores for both models using a heatmap. The figure shows how partial occlusions affect the performance of the two methods as measured by MPJPE.

We could investigate deeper by providing per-joint error heatmaps. In this case, after occluding a part of the image, only the error for a specific joint is calculated and projected on the input image. Figure 3.2 illustrates the results. It could be observed that we have higher sensitivity on the body and less in the background. Moreover, if a joint is visible, occluding it causes a high error for that joint (right knee, head). Also, it is shown that when a joint is occluded, the model looks at other related joints to leverage useful information (right ankle). In the second example in Figure 3.2, the OpenPose model can not detect the right wrist and the right elbow. That's why we see a high error regardless of the occluder's position. However, occluding the other person's head in the image helps

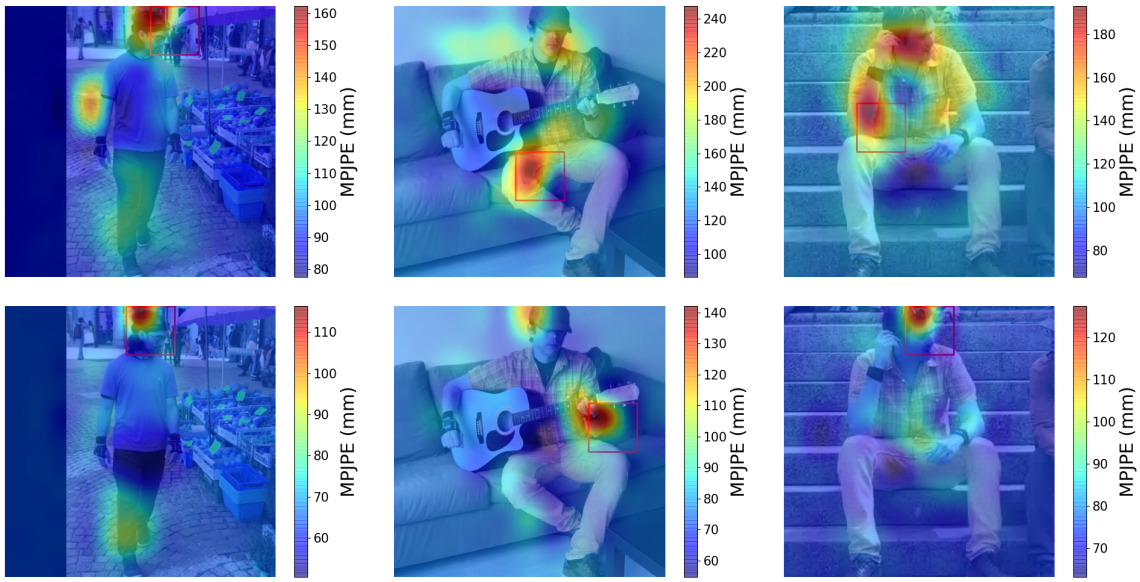

Figure 3.1: Locations-based Occlusion Sensitivity Analysis. A 40×40 square (gray) occluder is moved across the image and MPJPE values are computed (for each) for SPIN (top-row) and OpenPose (bottom-row) models. The heatmaps highlight locations in the image that strongly affects the performance of the two models. Both models are sensitive to occlusions in regions shown in red. Image size is 224×224 and the stride is selected to be 20. The figure demonstrates that although there are certain similarities, the SPIN and OpenPose models exhibit distinct responses to occlusion, leading to different regions of highest sensitivity highlighted by the red squares. The purpose of this image is not to compare the accuracy of the models, but rather to illustrate the models' diverse responses to occlusion.

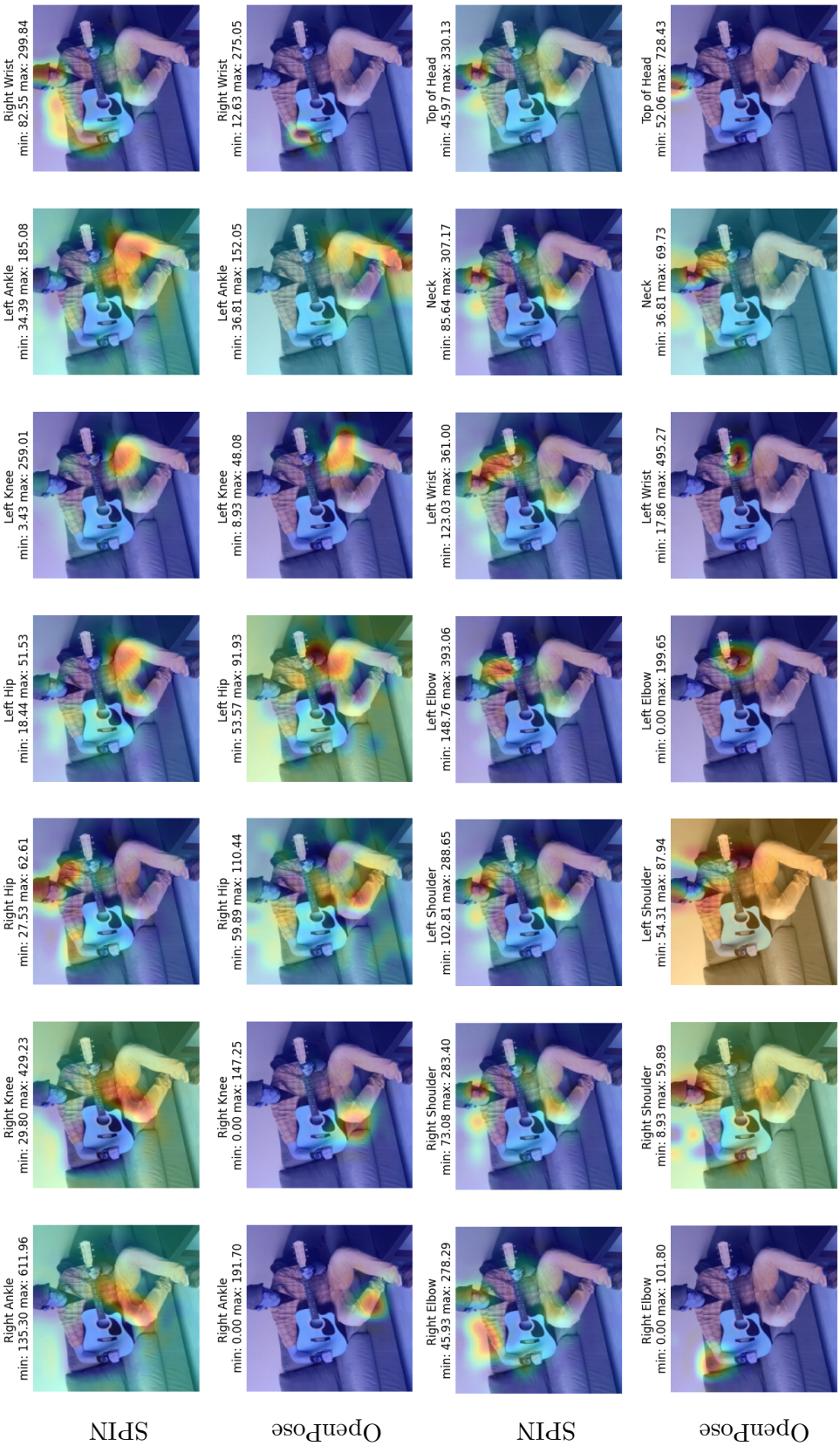

Figure 3.2: Locations-based Per-joint Occlusion Sensitivity Analysis. A 40×40 square (gray) occluder is moved across the image and errors are computed for each joint. The heatmaps highlight locations in the image that strongly affects the performance of the two models. Both models are sensitive to occlusions in regions shown in red. The image size is 224×224 and the stride is selected to be 20.

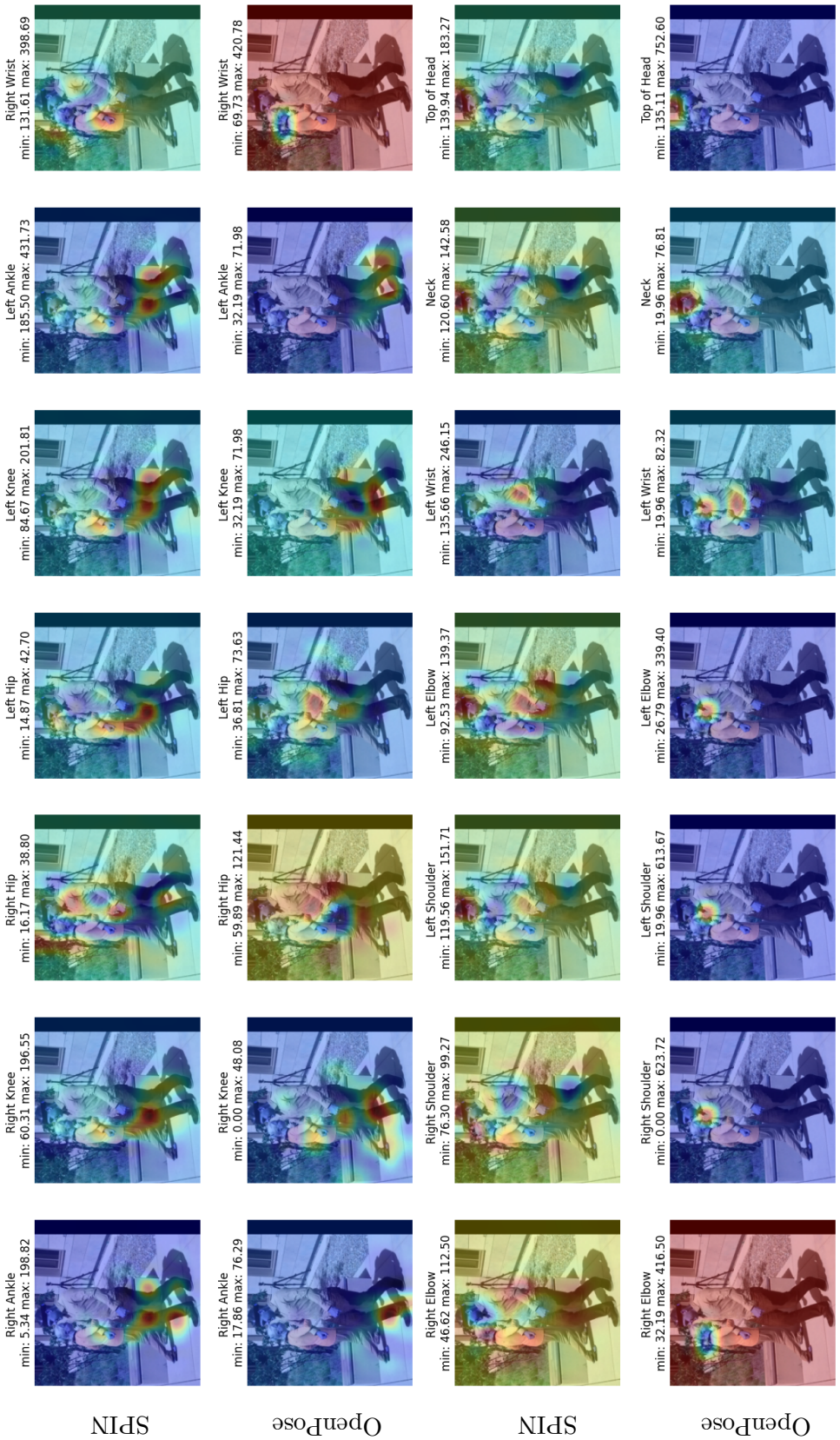

Figure 3.2: Locations-based Per-joint Occlusion Sensitivity Analysis (continued).

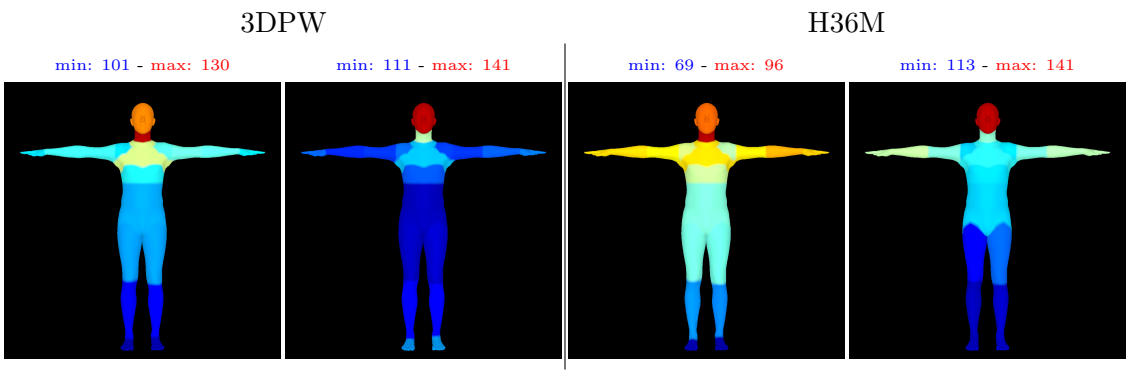

Figure 3.3: Joints-based Occlusion Sensitivity Analysis. For every image in 3DPW and H36M datasets, a square occluder is pasted over each joint in turn and MPJPE values are computed for SPIN (left) and OpenPose (right) models. MPJPE errors for each joint are visualized by highlighting the vertices (of the mesh) that correspond to each joint. These figures illustrate the contrasting behavior of the OpenPose and SPIN models when faced with occlusion. As shown in this figure, the error values and the most sensitive regions to occlusion differ between these models. This figure is best viewed in color.

the model to detect the joints and improve accuracy.

3.2 Joint-based Sensitivity Analysis

For the second approach, the square occluder is used to hide specific joints through the entire dataset. Where as the first approach captures the occlusions sensitivity to particular image locations, the second approach finds occlusions sensitivity to different joints. In this case

$$\text{MPJPE}_k^{\text{SPIN}} = \underset{i}{\text{Mean}}\ \underset{k}{\text{Mean}}\ \|X_{i,k} - X_i^{\text{gt}}\|, \tag{3.3}$$

where i indices over images, k indices over images, $X_{i,k}$ denotes 3D joints' locations estimations for image i when occluder is centered on joint k. X_i^{gt} is ground truth 3D joint locations for image i. Similarly,

$$\text{MPJPE}_k^{\text{OP}} = \underset{i}{\text{Mean}}\ \underset{k}{\text{Mean}}\ \|x_{i,k} - x_i^{\text{gt}}\|. \tag{3.4}$$

Here $x_{i,k}$ refers to OpenPose joint estimates for image i when the occluder is centered

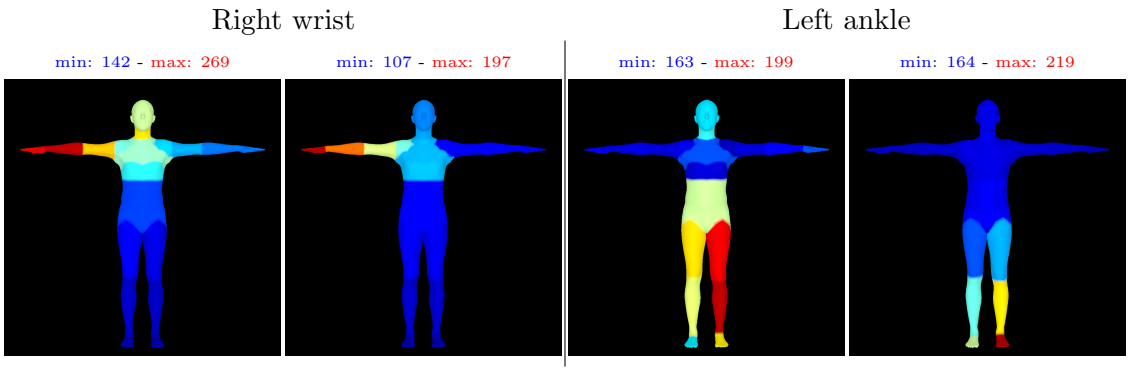

Figure 3.4: Joints-based Per-joint Occlusion Sensitivity Analysis. For every image in 3DPW datasets, a square occluder is pasted over each joint in turn and error values for the right wrist and the left ankle are computed for SPIN (left) and OpenPose (right) models. Results illustrate which parts of the body, models are looking at to estimate the right wrist and the left ankle position. Although both models depend on the neighboring body parts to estimate a joint position, the SPIN model exhibits a broader range of dependencies compared to the OpenPose model.

at joint k and x_i^{gt} denotes ground truth 2D joints for image i. Figure 3.3 visualizes MPJPE values for both methods on an SMPL mesh. Every vertex of a joint is associated with one or more joints, and each vertex is assigned a color using MPJPE_k values, where k belongs to the set of joints associated with this vertex. These colors visualize the sensitivity of the two methods to an occluded joint. The results indicate that the SPIN model is more dependent on other parts of the body compared to the OpenPose.

Furthermore, we investigate the impact of occluding different joints throughout the dataset on specific body parts. This would enable us to determine where the model is looking to estimate the position of that particular joint. Similar to the previous procedure, we occlude one joint at a time throughout the dataset, but this time instead of MPJPE we only calculate the error for a particular joint. Figure 3.4 illustrates some examples. Our results indicate that when the SPIN model tries to estimate the right wrist position, it is sensitive to the whole right arm and the neck. While predicting the left ankle position, the model is more dependent on the whole left foot rather than just the ankle. The left ankle is affected by the right foot occlusion as well. Comparing the SPIN model's results with the OpenPose outputs suggests that the SPIN model relies

more on neighboring body parts when estimating the position of a joint.

To conclude, the sensitivity analysis reveals that the SPIN and OpenPose models react differently to occlusion. In most cases, each model highlights different regions of the image as more sensitive. Furthermore, the different error values suggest that the models are estimating different joint positions when they make an error. This finding can be leveraged to calculate an error estimation for the reconstructed mesh.

Chapter 4

Method

In this chapter, we present our approach to estimating errors for single-image human body mesh reconstruction. We begin by demonstrating a correlation between the SPIN model error and the difference between the joint position estimations of SPIN and OpenPose. Afterward, we introduce three different approaches, namely Raw ED, Linear regression, and classifiers to extract the confidence feature.

4.1 Pearson Correlation Coefficient

When SPIN and OpenPose models correctly estimate a joint position, the estimated coordinates are close to each other and adjacent to the ground truth. However, based on the sensitivity analysis, when the models' estimated positions are inaccurate, we expect the joint position estimations to be dissimilar. Hence, the distance between the models' outputs

$$\mathrm{ED}_i = \|x_i^{\mathrm{SPIN}} - x_i^{\mathrm{OP}}\|, \tag{4.1}$$

can be considered as a proxy for confidence in the recovered human body mesh. Here x_i^{OP} are 2D joint estimates for OpenPose and x_i^{SPIN} are *projected* 2D joint estimates for SPIN. $\mathrm{ED}_i \in \mathtt{R}^K$ and i refers to the image. The related equations for projection are

explained in Appendix B.

To investigate the hypothesis that ED is a useful proxy for confidence in the recovered mesh, we calculate the correlation between the ED and the SPIN model's error

$$\mathrm{SE}_i = \|x_i^{\mathrm{SPIN}} - x_i^{\mathrm{gt}}\|. \tag{4.2}$$

The Pearson correlation coefficient (Appendix C) of joint k which is shown by r_k is calculated using

$$r_k = \mathrm{Corr}([\mathrm{ED}_{0,k}, ..., \mathrm{ED}_{n,k}], [\mathrm{SE}_{0,k}, ..., \mathrm{SE}_{n,k}]), \tag{4.3}$$

where n stands for the number of images in the dataset. Since the OpenPose model provides 2D estimates, it can only be compared to the 2D projection of the SPIN model output. Hence, SE only captures the 2D error of the SPIN model. Additionally, the OpePose model does not provide any estimations for the undetected joints, which forces us to ignore those points for calculating the correlation. That is to say, in our investigation of the correlation between estimation difference (ED) and real SPIN error (SE) for each joint, there are instances where OpenPose fails to provide estimated coordinates. In such cases, it becomes impossible to calculate ED, and as a result, we need to exclude that specific image (point) from the correlation calculation for that particular joint. By accounting for these exceptions and omitting the corresponding data points, we ensure a reliable and accurate assessment of the correlation between ED and SE for the joints under investigation. The computed correlation coefficient for the 3DPW test dataset for each joint is presented in Figure 4.1. The average coefficient $\overline{r} = 0.67$ indicates a strong correlation between ED and SE. This suggests that the differences in the estimated joint positions by SPIN and OpenPose models capture the error of SPIN model with respect to the ground truth. We leverage this information and explore three techniques that use ED to estimate confidence for the recovered mesh.

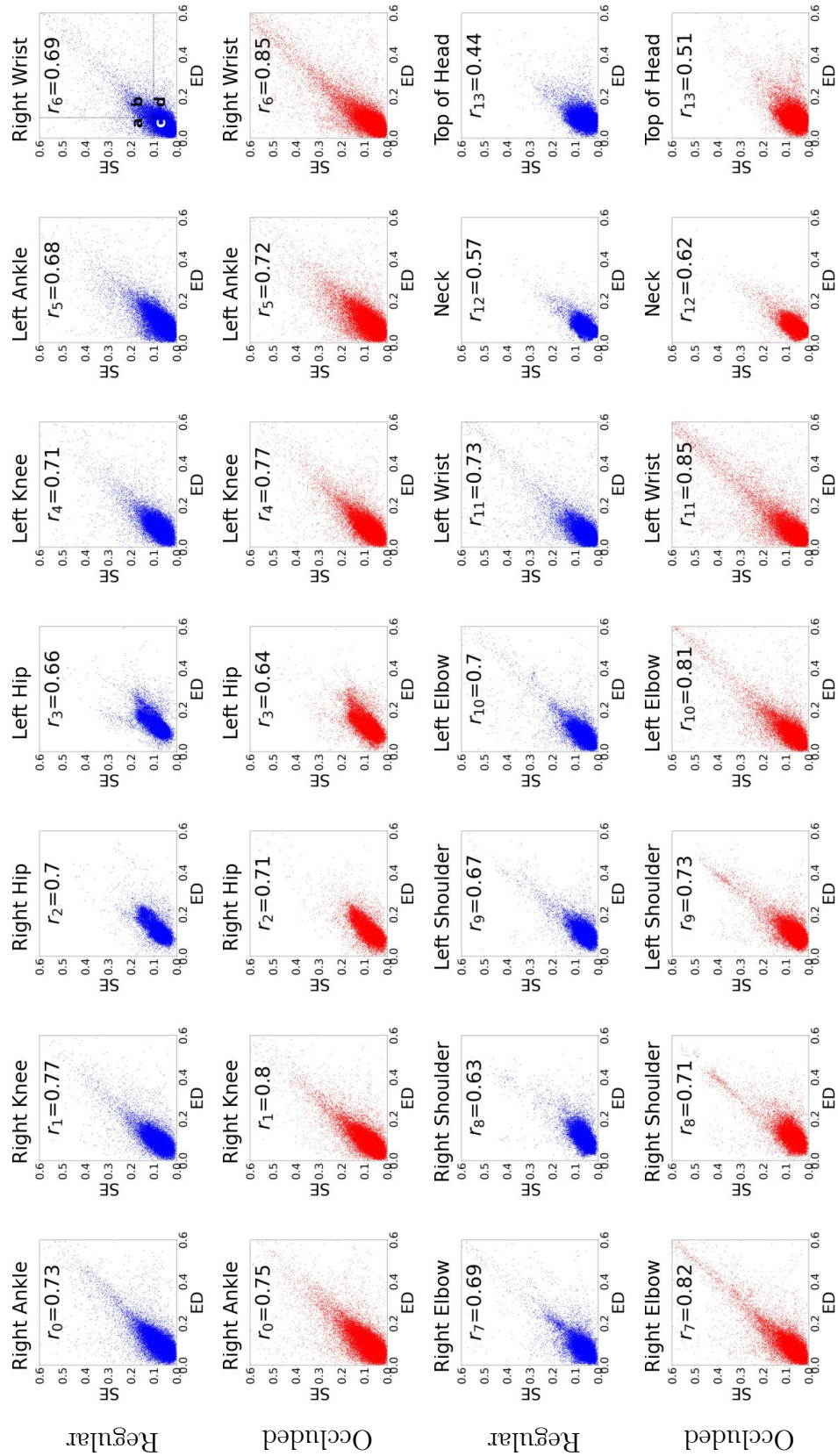

Figure 4.1: Pearson Correlation Coefficient. We calculate the correlation coefficient for each joint throughout the 3DPW dataset. The average value in the absence of occlusions is $\overline{r} = 0.67$. This value jumps to 0.735 for the occluded version of the 3DPW dataset. These values suggest a positive correlation between ED and SE. Four regions (a, b, c and d) are indicated in the top-right plot. Region a denotes False Positive scenarios, i.e., the estimated joint location is inaccurate, however, the proposed model has failed to identify it. Region d denotes False Negative scenarios where the estimated joint location is erroneously labelled inaccurate. Combining information from multiple joints helps deal with these scenarios.

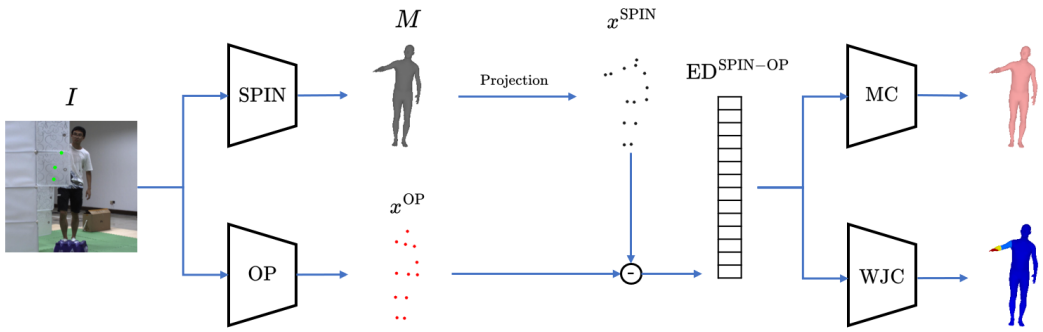

Figure 4.2: Overview of Our Proposed Framework. The input image I is passed through the SPIN and OpenPose models. Then, the estimated SPIN mesh (M) is regressed and projected into 2D joint coordinates. Comparing the results with the OpenPose predicted 2D joint positions, Estimation Difference (ED) is obtained. Afterward, ED is employed to train the Mesh Classifier (MC) and Worst Joint Classifier (WJC) that decide the SPIN mesh quality and detect the least reliable parts of the mesh, respectively.

4.2 Model's Framework

Figure 4.2 illustrates the proposed method for assigning an error estimate to different regions of the reconstructed human body mesh. It comprises three steps: 1) SPIN model is used to estimate "2D" joint locations, 2) OpenPose model is used to recover 2D joint locations, 3) The difference between the 2D joint estimates for SPIN and OpenPose is used to assign a confidence score to the mesh. For the third stage, we propose three methods, which are discussed in the following section.

4.2.1 Using Raw ED Values

For a given image, ED is a K-dimensional vector that stores the differences between joints' location estimates from SPIN and OpenPose models. We can use these values to decide whether or not the mesh is "good" as follows

$$y_{\text{mesh}} = \begin{cases} \text{good} & \text{if } \max \text{ ED} \leq \text{threshold} \\ \text{bad} & \text{otherwise.} \end{cases} \tag{4.4}$$

We can use a similar argument to identify the worst joint:

$$k_{\text{worst}} = \arg\max_{k} \text{ ED}. \tag{4.5}$$

4.2.2 Using Linear Regression

Plots shown in Figure 4.1 suggest a positive correlation between SE and ED (for all 14 joints), which suggests that it is possible to estimate SE given ED for a given joint. We are interested in estimating SE, since it represents the true SPIN error as computed using ground truth data. We do not have ground truth data at inference time, so instead we estimate SE using ED, which we can easily compute using SPIN and OpenPose models. Therefore, we fit a linear regressor

$$\text{SE}_k = (m_k)(\text{ED}_k) + c_k \tag{4.6}$$

that predicts $\text{SE}_{\cdot,k}$ given observation $\text{ED}_{\cdot,k}$, where $k \in [1, K]$. Given a new image, 1) compute ED, 2) use the trained linear regressor in Eq. 4.6 to estimate $\text{SE} \in \mathtt{R}^K$, and 3) use the estimated SE to decide whether or not mesh is "good" or to identify "good" and "bad" joints using the approach discussed in the previous section. Just substitute SE in place of ED.

4.2.3 Classifiers

The previous two approaches of using ED to classify recovered human body meshes and joints treat each joint separately. We now propose an approach that looks at all K joints simultaneously to classify the mesh and identify the worst joint. Specifically, we use two multi-linear perceptron networks that use ED to classify mesh and identify the worst joint, respectively.

The Mesh Classifier (MC) network is a binary classifier containing three hidden linear

layers that contain 10, 8, and 6 neurons respectively with ReLU activation functions. Input to MC is ED and it outputs whether or not the recovered mesh is reliable, i.e., all parts of the human body are visible in the image. MC network is trained using binary cross-entropy. The ground truth data for training MC is constructed using SE scores—if $\text{SE}_{.,k} \geq$ threshold for any k then the mesh is deemed unreliable, where $\text{SE}_{.,k}$ is the SE score for joint k. Under this regime

$$y_{\text{mesh}} = f_{\text{MC}}(\text{ED}). \tag{4.7}$$

The Worst Joint Classifier (WJC) network is a K-class classification network. It comprises three hidden layers containing 28, 56 and 28 neurons, respectively. Hidden layers use ReLU activation functions. ED is fed into WJC, and WJC is trained using cross-entropy. The ground truth data for training WJC is constructed from SE. We simply encode SE using one-hot-encoded form with 1 at $\arg\max_k$ SE and 0 elsewhere. Using WJC,

$$k_{\text{worst}} = f_{\text{WJC}}(\text{ED}). \tag{4.8}$$

In Chapter 5 we show that implementing the classifiers leads to the best results, hence the third approach as demonstrated in Figure 4.2 would be the default version of our proposed model. Moreover, the process of error projection from the joints to the mesh is described in Appendix D.

Chapter 5

Experiments and Results

Having developed the model, the purpose of this chapter is to evaluate its performance through experiments. Quantitative and qualitative results are presented, and an ablation study is conducted to compare the performance of different approaches. Finally, in order to better illustrate the application of our model in real world situations, we test our model for video input.

We use 3DPW [17] and Human3.6M [18] (S9 and S11) datasets for model training and testing. In addition, we use 3DOH [8] dataset for testing only. The threshold used in Section 4 is set at 10 mm, i.e., if the difference between an estimated joint location and the ground truth joint location is higher than 10 mm, the mesh recovered by the SPIN model is labelled inaccurate. We also created occluded versions of the three datasets where a randomly selected joint is occluded using a square occluder in each image.

5.1 Correlation Analysis

Figure 4.1 (rows 1 and 3) shows scatter plots of SE vs ED for every joint for the un-occluded 3DPW dataset. These plots also show Pearson correlation coefficient for each joint, which suggests that ED is positively correlated with SE. This is good news, since it suggests that in the absence of SE, which is not available at inference time, we can use

Figure 5.1: Samples with Positive and Negative Effects on the Correlation. The right wrist is occluded in the first input image, making both Openpose and SPIN models misestimate the right wrist's position. However, these wrong estimations are adjacent. The green dot shows the ground truth position and the red dot represents the OpenPose estimation of the right wrist. In the second case, OpenPose is confused by the other person's right wrist and makes a wrong estimation, while the SPIN model accurately estimates the right wrist. These are two samples that negatively affect the correlation between ED and SE. In the second row, two cases with a positive effect on the correlation are demonstrated. The first pair represents a case where both models perform accurately. The last pair indicates a situation where both models are inaccurate, but the position estimations for the right wrist are different.

ED to compute an error estimate for the recovered mesh.

Consider the ED vs. SE plot for right-wrist joint in Figure 4.1 (first row, right most figure). The plot identifies four regions labeled (a), (b), (c) and (d). Points in the regions (a) and (d) negatively affect the correlation while points in (b) and (c) regions have a positive effect.

Points in region (a) suggest that there are several situations where both models are inaccurate, but that they agree with each other. Thus, we conclude that when the OpenPose and SPIN estimates are close to each other, it does not necessarily mean that the recovered human mesh is accurate. Rather, it may be that joint estimates from both models are close to each other but far from the ground truth locations. Figure 5.1

(input/output pair on the left) depicts such a case. Here both models are in agreement with each other, however, both models fail to detect the right wrist due to self-occlusion and the presence of other people. Points in the region (d) represent cases where although the estimated values of SPIN and OpenPose model are different, the SPIN model is accurate. In other words, in some cases, a measurable difference in OpenPose and SPIN outputs does not indicate an inaccurate mesh reconstruction by the SPIN model. The right input/output pair in Figure 5.1 shows an example of such a case. The SPIN model is successful in estimating the right wrist of the person, however, OpenPose model makes a mistake and selects the other person's hand position as the correct location for the right wrist.

The points in region (c) represent cases where both SPIN and OpenPose accurately estimate the joint position. On the other hand, when both models fail to estimate a joint position accurately and predict different coordinates, region (b) is formed. This is the most important case since it would be the most common under heavy occlusion. The fact that models estimate different positions for the same joint contributes greatly to the observed correlation. In the second row of Figure 5.1, samples for cases (c) and (b) are depicted, respectively.

Despite the points in regions (a) and (d), the average Pearson correlation coefficient for all joints is $\overline{r} = 0.67$, indicating a strong correlation between ED and SE for all the joints. This confirms our intuition that ED is a good proxy for SE.

We performed a similar analysis as shown in Figure 4.1 (rows 2 and 4) for occluded dataset, where a square occluder is pasted on a randomly selected joint. The average Pearson correlation coefficient obtained under these settings is $\overline{r} = 0.735$, which is even higher than the value computed for the unonccluded case. This suggests two things: 1) that the proposed model is robust to occlusions and 2) ED is even more positively correlated with SE. Table 5.1 shows the Pearson correlation coefficient for different test datasets, and it shows Pearson correlation coefficient is higher for occluded datasets. In

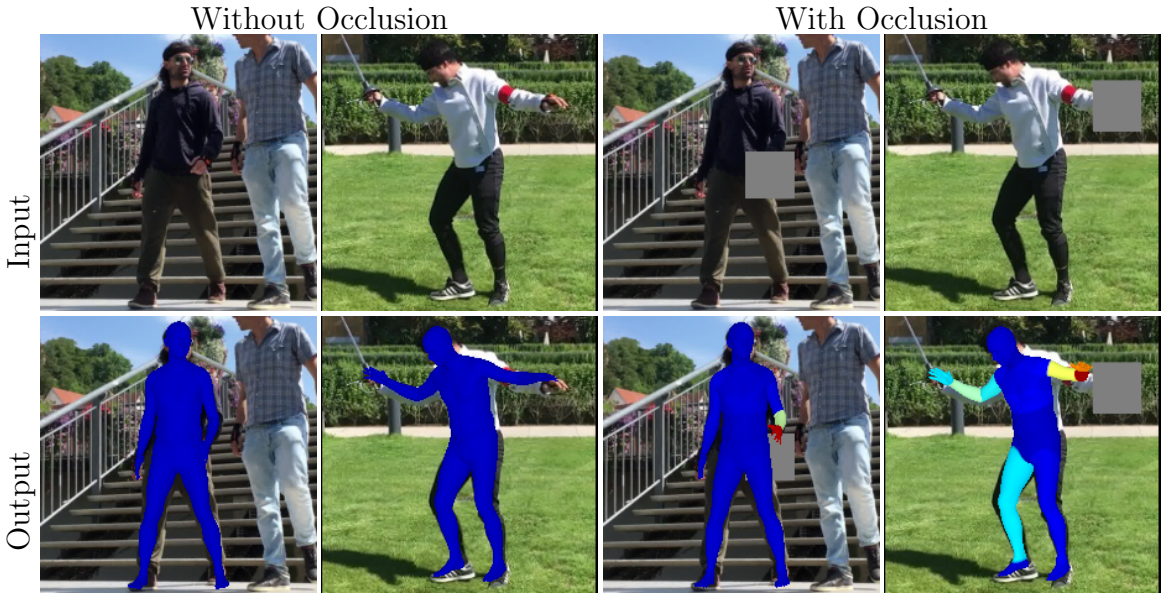

Figure 5.2: Occluded Samples. The error distribution on the estimated mesh changes when part of the human is occluded. For example, when a squared occluder is pasted onto the left hand, the model successfully identifies that is the least reliable region of the mesh (red regions on the mesh).

addition, 3DPW and 3DOH datasets have higher coefficient values since these exhibit higher occlusion levels.

Figure 5.2 illustrates two instances of the model's behavior towards occlusion. Our model predicts that the recovered mesh is correct when there are no occlusions, however, the model correctly identifies the left wrist region of the recovered mesh to be unreliable when a square occluder is used to hide this joint in the input image.

5.2 Quantitative Results

We exploit the positive correlation between ED and SE to estimate the error in the human body mesh recovered by SPIN. The proposed method also highlights the least reliable region of the recovered mesh. Table 5.1 lists our model's performance at identifying an inaccurate mesh. Additionally, this table also includes model's performance at identifying the least reliable joint. There is no baseline, since, to the best of our knowledge, ours

Dataset	PCC		Mesh		WJ-R1		WJ-R3	
Regular/Occluded	R	O	R	O	R	O	R	O
3DPW	0.67	0.735	79.2%	86.2%	42.2%	45.4%	70.6%	73.8%
3DOH	0.665	0.707	81.6%	88%	37.3%	40.5%	64.2%	66.8%
H36M-P1	0.492	0.545	71.9%	82.1%	42.4%	43.9%	76.4%	75.1%

Table 5.1: Model Evaluation. Pearson Coefficient Correlation (PCC), model accuracy in separating accurate and faulty meshes (Mesh), and model performance on detecting the least reliable joints, i.e., worst joints (WJ), are presented in this table. Model is allowed a single guess for Rank 1 (R1) and it is allowed three guesses for Rank 3 (R3).

is the first attempt at performing error estimation for single-image human body mesh reconstruction scenarios. For example, while the model was never trained on 3DOH dataset, it is able to identify an inaccurate mesh with 88% accuracy. The model is also able to identify the least reliable joint 40.5% accuracy. This number jumps to 66.8% when the model is allowed three guesses for the least reliable joint. These numbers are considerably higher than randomly selecting the least reliable joint. A similar trend is visible for 3DPW and H36M-P1 datasets.

5.3 Qualitative Results

Consider Figure 5.3 that presents some qualitative results. The first four rows show cases where the proposed model performed correctly. Here MC denotes output from the mesh classifier and MC-GT denotes the ground truth. WJC highlights the least reliable joint(s) and WJC-GT shows the least reliable joint ground truth. The bottom two rows show failure cases. Here, while the model correctly predicts that the recovered mesh is unreliable, it is unable to identify the least reliable joint correctly.

5.4 Video Analysis

Figure 5.4 shows an application of our method on video data. Here the top row shows input frames, the second row shows whether or not the recovered mesh is reliable, and the

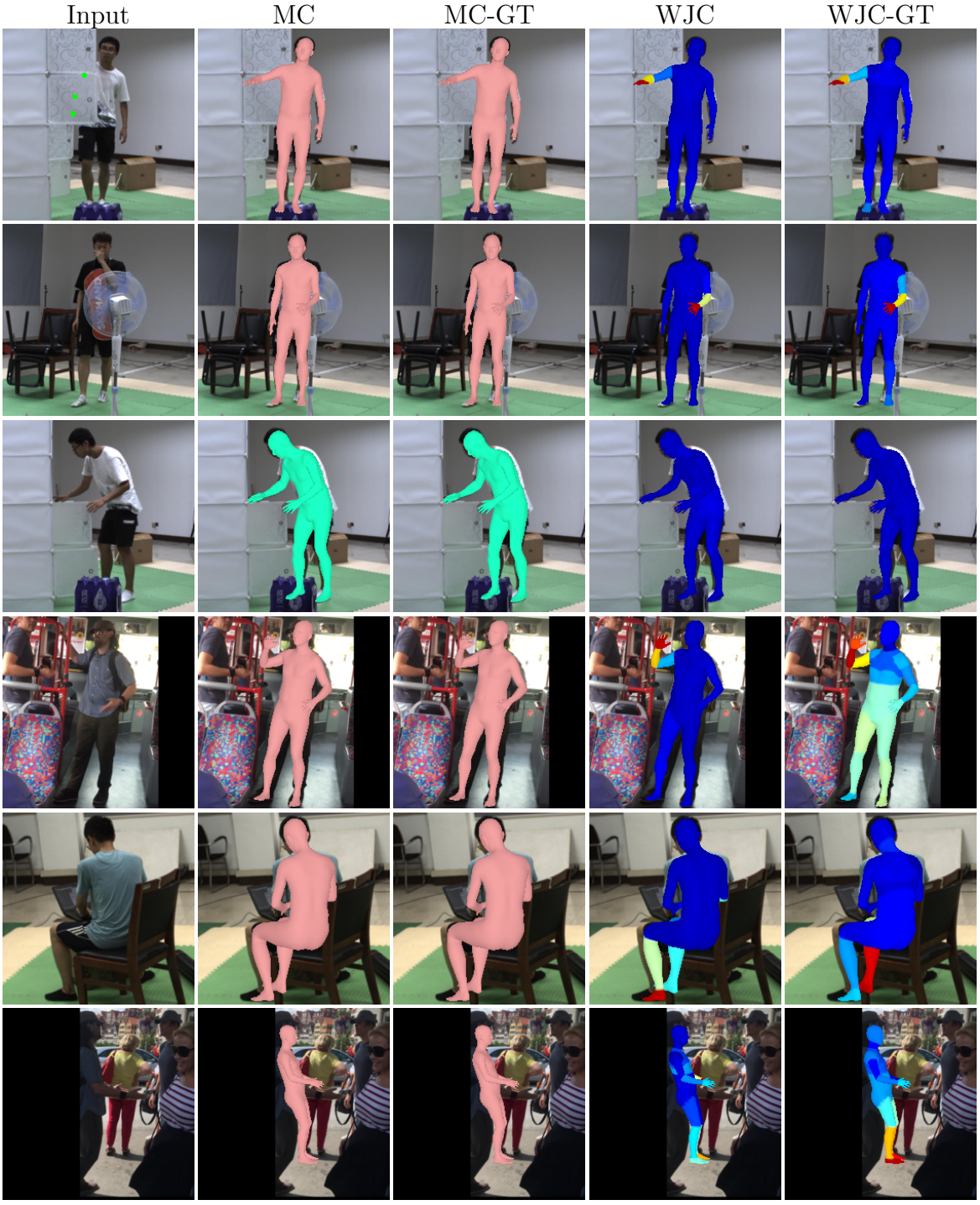

Figure 5.3: Qualitative Results. Input images are shown in the left column. The next two columns contain the mesh classifier output and the ground truth. Unreliable meshes are shown in light pink. The fourth column highlights the least reliable joints. Red regions on the mesh correspond to the least reliable joints. The last column shows the ground truth for the least reliable joints.

last row includes a visualization of the least reliable joint. The meshes shown in the last row in Figure 5.4 are rotated to better see the least reliable joints. Our model correctly handles occlusions due to other objects in the scene (top three rows) and self-occlusions (last three rows). Check the third row where the model correctly predicts that the left foot is the least reliable region of the recovered mesh since it is not visible in the image (it is occluded by the table). The decision to decide if the recovered mesh is "reliable" when only left foot is not visible in the image is application specific. For example, say a robot is simply navigating around this person then perhaps it is okay to deem the recovered mesh to be reliable. However, if this same robot is carrying out a task that involves the left foot of this person then it is best to consider this mesh unreliable. In the second case, due to the dancer's fast movements, frame get blurry and the SPIN model fails to accurately estimate the hands. Our model, as shown in Figure5.4 detects the inaccurate mesh and in most cases distinguishes the hands as the least reliable part of the mesh. finally, in the third example, as soon as the left hand moves behind the body, our model tags the mesh as inaccurate and highlights the left hand as well.

Furthermore, we investigate the model's performance against artificial occlusion, and the results are demonstrated in Figure 5.5. In the first case, a fixed square occluder is placed on the top of the frames. As the person jumps and his head is occluded, SPIN model starts generating inaccurate meshes and our model detects the unreliability in the estimated mesh. In the second case, the occluder is moved over the image, and results illustrate that in most cases, our model not only detects the inaccurate meshes but also distinguishes the least reliable parts.

5.5 Ablation Study

We now compare the performance of the three approaches discussed in Section 4. All three approaches leverage the positive correlation between ED and SE. Table 5.2 shows

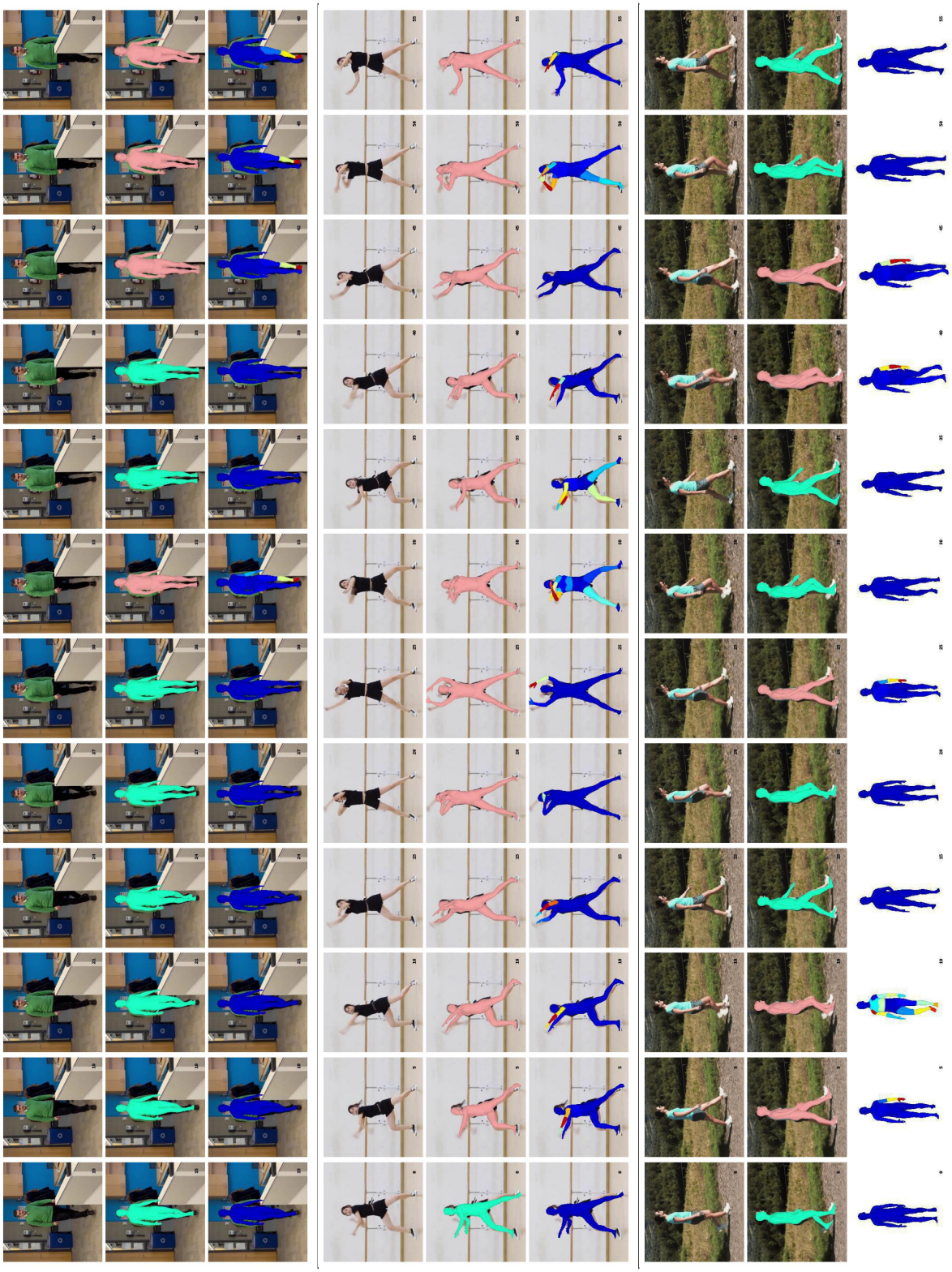

Figure 5.4: Video Analysis. The first row shows the input video frames. The second row shows mesh reliability classification results (MC). Light pink indicates an unreliable mesh. The third row shows the least reliable joints (WJC). The red regions on the mesh highlight the least reliable joint.

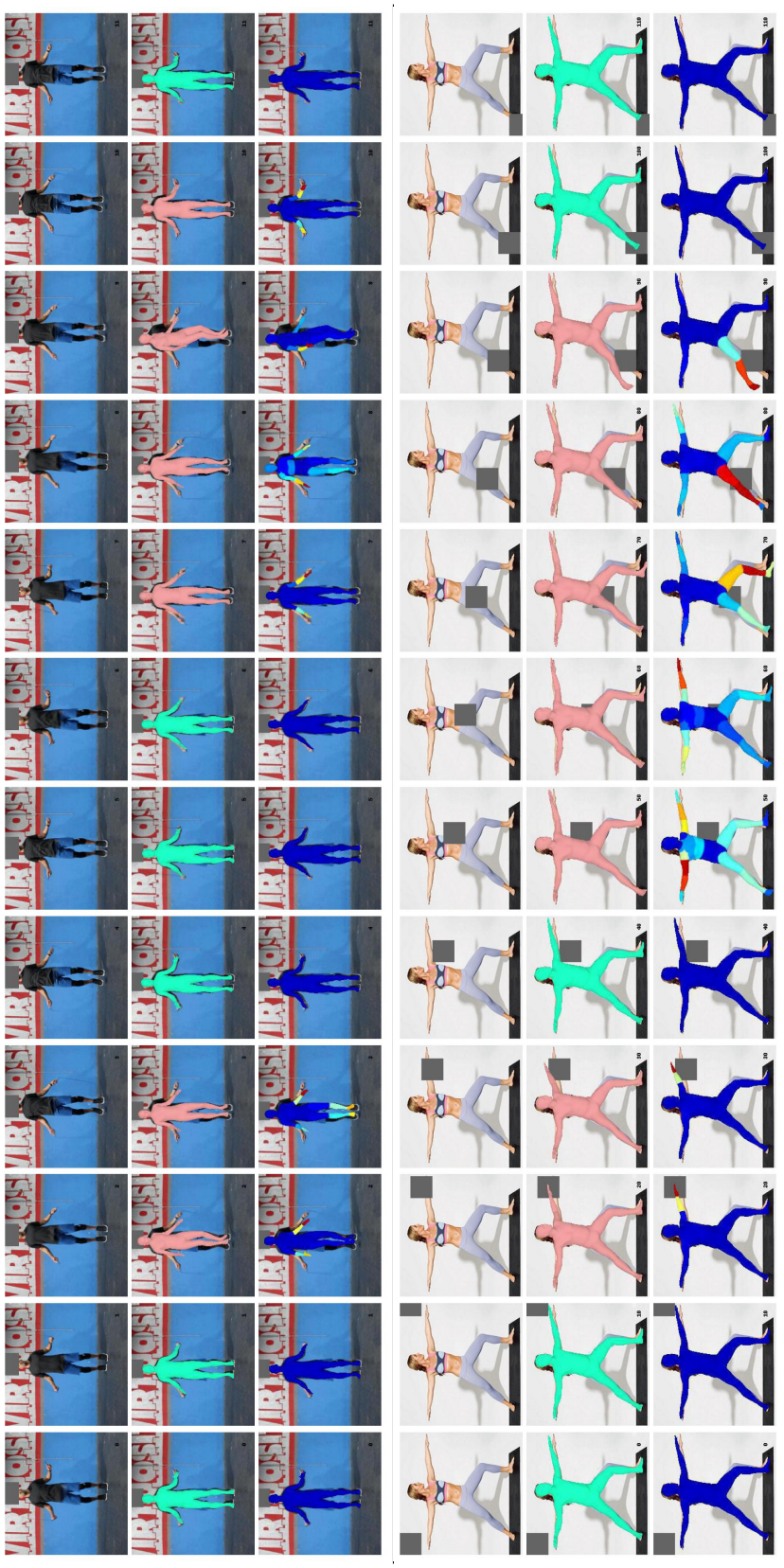

Figure 5.5: Occluded Video Analysis. The first row shows the input video frames. The second row shows mesh reliability classification results (MC). Light pink indicates an unreliable mesh. The third row shows the least reliable joints (WJC). The red regions on the mesh highlight the least reliable joint.

Datasets		Metric	ED	L. Regressor	Classifier
3DPW	R	Mesh	71.2	75.3	79.2
		WJ-R1	27.8	38.2	42.2
		WJ-R3	61.5	68.8	70.6
	O	Mesh	82.7	81.2	86.2
		WJ-R1	30.7	42.17	45.4
		WJ-R3	65.5	72.2	73.8
3DOH	R	Mesh	80.8	82.9	81.6
		WJ-R1	22	30.4	37.3
		WJ-R3	54.5	70.2	64.2
	O	Mesh	88.6	88	88
		WJ-R1	22.5	31.7	40.5
		WJ-R3	55.2	67.5	66.8
H36M-P1	R	Mesh	66.5	67.8	71.9
		WJ-R1	17.8	29	42.4
		WJ-R3	58.6	66.5	76.4
	O	Mesh	79.9	78.2	82.1
		WJ-R1	22.9	36	43.9
		WJ-R3	64.1	69.5	75.1

Table 5.2: Ablation Study. Comparing the method that uses raw ED values (column 3), linear regressor (column 4), and classifier based method (column 5) for classifying unreliable meshes and identifying the least reliable joints. Mesh refers to mesh reliability classification results, WJ-R1 refers to the results for identifying the worst joint (least reliable) when a single guess is allowed, and WJ-R3 refers to results for identifying the worst joint in three guesses.

the results obtained for each approach on the three datasets in both unoccluded and occluded cases. The results confirm that the classifier-based approach that combines ED information from different joints outperforms the other two methods. Method that uses raw ED values posts the worst performance. What is interesting to note is that using a classifier dramatically increases the performance of identifying the least reliable joint, both when the model is allowed a single guess and when it is allowed three guesses. This is mainly due to the fact that the classifier's approach considers all elements of the estimation difference (ED) simultaneously. By doing so, the classifier is able to effectively utilize all the available information regarding joint estimation differences. For mesh classification, however, the improvement obtained by using a classifier-based approach over using the method that relies on raw ED values is not nearly as significant.

Chapter 6

Conclusion

This work develops a method for estimating the error in the human body meshes reconstructed by the SPIN model. The model is not only able to decide whether or not a mesh is unreliable, but it is also able to highlight the least reliable, i.e., having the highest error, regions on the mesh. The proposed model uses the disagreement between joint location estimates between OpenPose and SPIN model to compute error values for the recovered mesh. Pearson correlation coefficient studies on 3DPW dataset show this disagreement is a good proxy for the "true" error. Evaluations on 3DPW, 3DPH, and H36M-P1 confirm that the model is able to estimate error in the SPIN based single-image human body mesh reconstructions in the presence of occlusions. Furthermore, it is able to correctly estimate the error in SPIN meshes even when OpenPose estimates are incorrect. The model is also able to identify the least reliable joints. The ability to estimate the error in the recovered meshes is particularly important when these meshes are used in human-robot interaction scenarios. To the best of our knowledge, ours is the first method to estimate the error in single-image 3D human body mesh reconstruction.

6.1 Future Works

Improving the current research in human body mesh reconstruction can be achieved by estimating the 3D error of the reconstructed mesh. By using a pair of 3D models, we can estimate the error for each vertex of the mesh. The response of different models towards occlusion could be utilized to make an error estimation. Our research presents a comprehensive framework for error estimation in 3D mesh generation. Within this framework, it is feasible to incorporate other mesh generation methods to estimate the error. Notably, models like ICON [47], ECON [48], and Vid2Avatar [49], specifically designed for 3D clothed human mesh recovery, are well-suited for integration. By exploring their behavior under occlusion and assessing the accuracy and compatibility of different model combinations, we can further enhance the capabilities of our framework. Additionally, even models utilizing multiple camera inputs can be compared and incorporated into our proposed approach to generate more precise error estimations.

Furthermore, adding temporal features by considering multiple frames in the estimation process could lead to improved accuracy. The proposed model's application in safe Human-Robot Interaction is a rewarding subject to study. The reliability of the estimation can inform the robot to adjust its position or stop working to avoid any potential risks.

Moreover, recent advancements in shape and pose estimation techniques have allowed for the accurate estimation of body shape that reflects an individual's weight group and body type. However, the generation of realistic, undressed body shapes raises ethical and privacy concerns as it may not be acceptable to most individuals. As a result, there is a need for further research to explore the ethical and privacy-related issues associated with this field.

Bibliography

[1] A. Zacharaki, I. Kostavelis, A. Gasteratos, and I. Dokas, "Safety bounds in human robot interaction: A survey," *Safety science*, vol. 127, p. 104667, 2020.

[2] B. Marr, "5 ways self-driving cars could make our world (and our lives) better." `https://www.forbes.com/sites/bernardmarr/2020/07/17/5-ways-self-driving-cars-could-make-our-world-and-our-lives-better/?sh=fa53e5042a33`.

[3] "Digital human." `https://research.csiro.au/digitalhuman/`.

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299, 2017.

[5] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[6] Y. Tian, H. Zhang, Y. Liu, and L. Wang, "Recovering 3d human mesh from monocular images: A survey," *arXiv preprint arXiv:2203.01923*, 2022.

[7] J. Li, C. Xu, Z. Chen, S. Bian, L. Yang, and C. Lu, "Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3383–3393, 2021.

[8] T. Zhang, B. Huang, and Y. Wang, "Object-occluded human shape and pose estimation from a single color image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7376–7385, 2020.

[9] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black, "Pare: Part attention regressor for 3d human body estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11127–11137, 2021.

[10] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7122–7131, 2018.

[11] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] A. Martínez-González, M. Villamizar, O. Canévet, and J.-M. Odobez, "Real-time convolutional networks for depth-based human pose estimation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 41–47, IEEE, 2018.

[13] H. Liu and L. Wang, "Human motion prediction for human-robot collaboration," *Journal of Manufacturing Systems*, vol. 44, pp. 287–294, 2017.

[14] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki, "Self-supervised learning of motion capture," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2252–2261, 2019.

[16] R. Khirodkar, S. Tripathi, and K. Kitani, "Occluded human mesh recovery," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1715–1725, 2022.

[17] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 601–617, 2018.

[18] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

[19] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2602–2611, 2017.

[20] D. C. Luvizon, H. Tabia, and D. Picard, "Human pose regression by combining indirect part detection and contextual information," *Computers & Graphics*, vol. 85, pp. 15–22, 2019.

[21] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7093–7102, 2020.

[22] K. Sun, C. Lan, J. Xing, W. Zeng, D. Liu, and J. Wang, "Human pose estimation using global and local normalization," in *Proceedings of the IEEE international conference on computer vision*, pp. 5599–5607, 2017.

[23] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1212–1221, 2017.

[24] M. B. Gamra and M. A. Akhloufi, "A review of deep learning techniques for 2d and 3d human pose estimation," *Image and Vision Computing*, vol. 114, p. 104282, 2021.

[25] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4929–4937, 2016.

[26] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," in *European conference on computer vision*, pp. 34–50, Springer, 2016.

[27] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *European conference on computer vision*, pp. 561–578, Springer, 2016.

[28] A. Zanfir, E. Marinoiu, and C. Sminchisescu, "Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2148–2157, 2018.

[29] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3d and 2d human representations," in *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pp. 6050–6059, 2017.

[30] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model based human pose and shape estimation," in *2018 international conference on 3D vision (3DV)*, pp. 484–494, IEEE, 2018.

[31] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3d human pose and shape from a single color image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 459–468, 2018.

[32] U. Iqbal, K. Xie, Y. Guo, J. Kautz, and P. Molchanov, "Kama: 3d keypoint aware body mesh articulation," in *2021 International Conference on 3D Vision (3DV)*, pp. 689–699, IEEE, 2021.

[33] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 13001–13008, 2020.

[34] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 364–380, 2018.

[35] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe, "How robust is 3d human pose estimation to occlusion?," *arXiv preprint arXiv:1808.09316*, 2018.

[36] L. Ke, M.-C. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," in *Proceedings of the european conference on computer vision (ECCV)*, pp. 713–728, 2018.

[37] Z. Wang, J. Yang, and C. Fowlkes, "The best of both worlds: combining model-based and nonparametric approaches for 3d human body estimation," in *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2318–2327, 2022.

[38] G. Georgakis, R. Li, S. Karanam, T. Chen, J. Košecká, and Z. Wu, "Hierarchical kinematic human mesh recovery," in *European Conference on Computer Vision*, pp. 768–784, Springer, 2020.

[39] K. Yang, R. Gu, M. Wang, M. Toyoura, and G. Xu, "Lasor: Learning accurate 3d human pose and shape via synthetic occlusion-aware data and neural mesh rendering," *IEEE Transactions on Image Processing*, vol. 31, pp. 1938–1948, 2022.

[40] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on multi-stage networks for human pose estimation," *arXiv preprint arXiv:1901.00148*, 2019.

[41] Y. Bin, Z.-M. Chen, X.-S. Wei, X. Chen, C. Gao, and N. Sang, "Structure-aware human pose estimation with graph convolutional networks," *Pattern Recognition*, vol. 106, p. 107410, 2020.

[42] Z. Geng, K. Sun, B. Xiao, Z. Zhang, and J. Wang, "Bottom-up human pose estimation via disentangled keypoint regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14676–14686, 2021.

[43] Z. Yu, J. Wang, J. Xu, B. Ni, C. Zhao, M. Wang, and W. Zhang, "Skeleton2mesh: Kinematics prior injected unsupervised human mesh recovery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8619–8629, 2021.

[44] J. Cho, K. Youwang, and T.-H. Oh, "Cross-attention of disentangled modalities for 3d human mesh recovery with transformers," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pp. 342–359, Springer, 2022.

[45] B. Huang, T. Zhang, and Y. Wang, "Pose2uv: Single-shot multiperson mesh recovery with deep uv prior," *IEEE Transactions on Image Processing*, vol. 31, pp. 4679–4692, 2022.

[46] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[47] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black, "Icon: implicit clothed humans obtained from normals," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13286–13296, IEEE, 2022.

[48] Y. Xiu, J. Yang, X. Cao, D. Tzionas, and M. J. Black, "ECON: Explicit Clothed humans Optimized via Normal integration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[49] C. Guo, T. Jiang, X. Chen, J. Song, and O. Hilliges, "Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition," in *Computer Vision and Pattern Recognition (CVPR)*, 2023.

[50] W. Kirch, "Pearson's correlation coefficient," *Encyclopedia of public health*, vol. 1, pp. 1090–1091, 2008.

[51] E. I. Obilor and E. C. Amadi, "Test for significance of pearson's correlation coefficient," *International Journal of Innovative Mathematics, Statistics & Energy Policies*, vol. 6, no. 1, pp. 11–23, 2018.

Appendix A

End-to-end Recovery of Human Shape and Pose

A.1 Introduction

This paper [10] presents an end-to-end method for reconstructing 3D human body mesh from a single RGB image. Most of the past research in this field concentrated on recovering 3D joint locations. Those works focused on recovering the 3D mesh had a multi-stage framework that could have been more optimal. 3D parameters were estimated based on the previous stage of 2D keypoint prediction. HMR was the first model that proposed an approach to map 3D parameters directly from image pixels.

The main challenge is that few 3D annotated datasets exist for in-the-wild images. The root problem with employing 2D datasets is that 2D to 3D mapping is always accompanied by ambiguity. Since many different 3D poses are projected to the same 2D pose. Moreover, camera parameters estimation is challenging and causes confusion between the person's size and camera distance. These difficulties lead to unrealistic 3D meshes where body angles are impossible or the recovered mesh is too small or too big. Kanazawa et al. tackled these problems by proposing a model trained on the available

2D annotated datasets while dealing with ambiguities, utilizing a discriminator to reject the unrealistic results. The discriminator is developed based on the large-scale dataset of 3D meshes of people with various poses and shapes.

A.2 Model

The overall framework is illustrated in Figure A.1. Image features are extracted using the pre-trained Resnet network. Then, the extracted features are fed into a regressor which aims to predict the 3D parameters, including camera, shape, and pose parameters (85-dimensional vector). The generated 3D parameters are the SMPL model input responsible for reconstructing the human body mesh. Afterward, the mesh is projected into 2D keypoints, and the results are compared to the available 2D ground truth. The regressor's output also passes through the discriminator. The goal of the discriminator is to reject unrealistic inputs. Constraints such as logical height range, weight range, bone ratios, and joint angles are checked through the discriminator. Simply put, the discriminator works as a weak 3D supervision in the absence of real 3D data. The overall loss function is

$$L = \lambda(L_{reproj} + L_{3D}) + L_{adv}, \tag{A.1}$$

where L_{reproj} and L_{3D} are the regular 2D and 3D losses and L_{adv} stands for the adversarial prior (discriminator). Through the training stage, to minimize the loss, the regressor learns to predict 3D parameters that generate a realistic mesh and reduce the 2D and 3D reprojection error.

A.2.1 Iterative 3D Regression with Feedback

The regressor's objective is to estimate a proper Θ, the 85-dimensional feature vector. However, this is a challenging task. Hence, an iterative error feedback technique is employed. The extracted image features ϕ and an initial estimate Θ_0 are concatenated

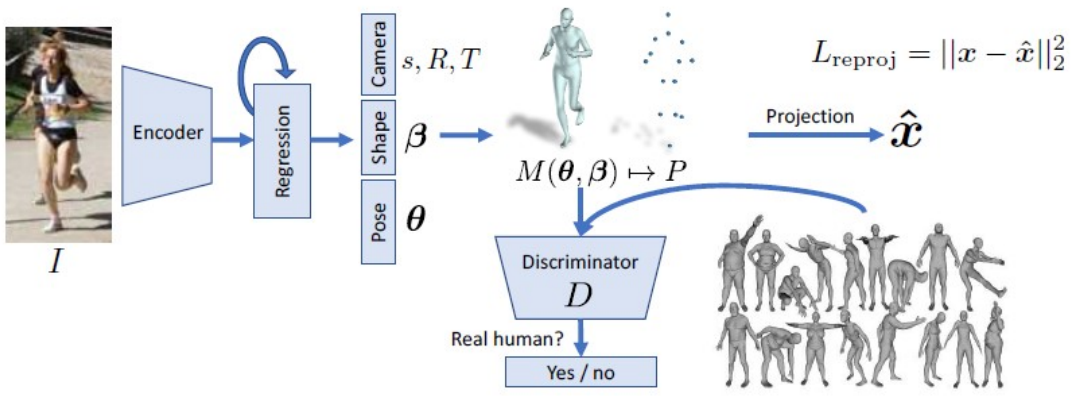

Figure A.1: Overview of the HMR Framework. A convolutional encoder is used to process an image, which is then transmitted to an iterative 3D regression module that calculates the hidden 3D representation of the person in the image in a way that minimizes the error in projecting the joints. The 3D parameters are also sent to a discriminator called D, which determines whether these parameters are derived from a genuine human shape and pose [10].

and fed to the regressor. The output is $\Delta\Theta_t$. Using the output, the updated $\Theta_{t+1} = \Theta_t + \Delta\Theta_t$ will be the next input of the regressor. This loop is repeated thrice, and the final Θ_T is obtained. The reprojection and 3D loss are calculated as follows:

$$L_{reproj} = \sum_i \|\nu_i(x_i - \hat{x}_i)\|, \tag{A.2}$$

$$L_{3D} = L_{3Djoints} + L_{3Dsmpl}, \tag{A.3}$$

$$L_{joints} = \|(X_i - \hat{X}_i)\|, \tag{A.4}$$

$$L_{smpl} = \|\Theta_i - \hat{\Theta}_i\|, \tag{A.5}$$

where, $\nu_i \in [0, 1]$ is the visibility of each joint, x_i and $\hat{x}_i$ are the 2D ground truth and predicted coordinates of the joints, X_i and $\hat{X}_i$ are the 3D ground truth and predicted coordinates of the joints. The adversarial loss is applied in each iteration to achieve the best results while L_{reproj} and $L3D$ are just for the final estimate Θ_T.

Figure A.2: Without the use of both the discriminator and direct 3D supervision, the network generates unrealistic results or "monsters," as depicted in the examples. Despite the abnormal pose and shape of the generated images, their 2D projection error is very precise [10].

A.2.2 Factorized Adversarial Prior

Relying solely on 2D and 3D projection loss leads to low-quality mesh recovery due to the existing ambiguity in a 2D image to 3D mesh problem. To tackle this issue, discriminators are employed, increasing the loss when the generated mesh is unrealistic. There is one discriminator for each joint, one for all joints together, and one for the shape parameters. While developing the discriminator, the goal is to get outputs close to 1 when we use valid Θ from the mesh bank and low values when the input is generated by our model from an image, as shown below:

$$\min \quad L(D_i) = \mathbb{E}_{\Theta \sim P_{data}}[(D_i(\Theta) - 1)^2] + \mathbb{E}_{\Theta \sim PE}[D_i(E(I))^2]. \tag{A.6}$$

During the HMR training, we want high loss values when the discriminator detects unusual input values (close to 0) and low values when the input Θ is in its realistic range (close to 1). Therefore, the discriminator loss is calculated as in

$$\min \quad L_{adv}(E) = \sum_i \mathbb{E}_{\Theta \sim PE}[(D_i(E(I)) - 1)^2], \tag{A.7}$$

where, D_i is the discriminator and E is our encoder. Figure A.2 shows the importance of the discriminator. Since there are not enough 3D datasets, without a discriminator model performance deteriorates considerably.

A.2.3 Conclusion

HMR presented a state-of-the-art performance on an appropriate estimation of 3D body parameters and part segmentation even from in-the-wild data resources and reconstructed a better mesh than existing optimization-based approaches. The presented framework has been widely used as the backbone of many other architectures, including the SPIN [15] model.

Appendix B

3D to 2D Projection

Any image is a 2D representation of the outside 3D world. Although some information is lost due to the dimension reduction, we can map 3D points (X) to 2D image points (x). The general camera geometry is illustrated in Figure B.1. Using homogeneous coordinates, we have:

$$x = PX. \tag{B.1}$$

The projection matrix P is defined as

$$P = K[R|t], \tag{B.2}$$

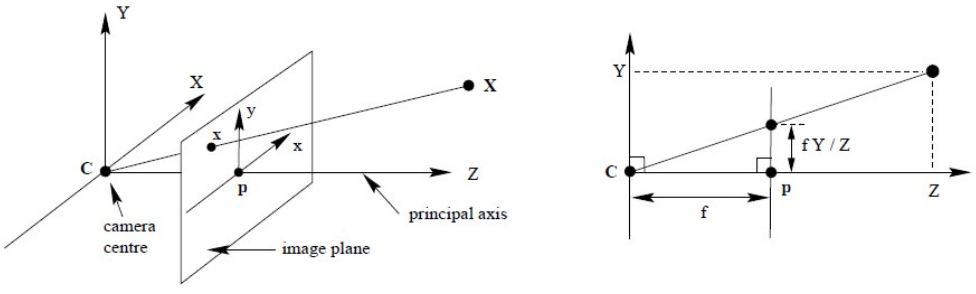

Figure B.1: Pinhole Camera Geometry. C is the camera center, and p is the principal point. The camera center is here placed at the coordinate origin. Note the image plane is placed in front of the camera center. [11].

where K is the intrinsic matrix and $[R|t]$ is called the extrinsic matrix. In some cases, these matrices are available, and we can directly calculate the projection matrix. Otherwise, we need the camera rotation matrix R and the camera translation t. Moreover, to form the intrinsic matrix, we need the focal length f and the principal point offset:

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{B.3}$$

In most cases, there is no principal point offset, and the camera is not rotated. Also, the focal length is known. The camera translation is the only unknown value. In our research, camera translation is predicted by the SPIN model. Therefore, we are able to project the 3D point into a 2D image environment.

Appendix C

Pearson Coefficient Correlation

Pearson's correlation coefficient (r) measures the linear relationship between two variables. Correlation analysis typically begins with a graphical representation of the relationship between data pairs, such as a scatter diagram. The correlation coefficient ranges from -1 to +1. Positive correlation coefficient values suggest a propensity for one variable to rise or decrease in tandem with another one. Negative correlation coefficient values suggest a tendency for an increase in one variable to be connected with a fall in the other variable and vice versa. Correlation coefficient values close to zero suggest a weak linear relationship between two variables, whereas those close to -1 or +1 indicate a strong linear relationship between two variables. [50].

A correlation coefficient of 0 implies no correlation (zero relationship). Further, correlation coefficients lower than 0.40 (whether negative or positive 0.40) are said to be low, between 0.40 and 0.60 are moderate, and above 0.60 are high [51]. Finally, given the bivariate set $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, Pearson's Correlation Coefficient r is defined as:

$$r = \frac{N \sum xy - (\sum x)(\sum y)}{\sqrt{[N \sum x^2 - (\sum x)^2][N \sum y^2 - (\sum y)^2]}}, \tag{C.1}$$

where,

r = Pearson's Correlation Coefficient

N = Number of pairs of values or scores

$\sum xy$ = Sum of the products of x and y

$\sum x$ = Sum of the x values (or x scores)

$\sum y$ = Sum of the y values (or y scores)

$\sum x^2$ = Sum of squares of x values (or x scores)

$\sum y^2$ = Sum of squares of y values (or y scores)

$(\sum x)^2$ = Square of the sum of x values (or x scores)

$(\sum y)^2$ = Square of the sum of y values (or y scores)

Appendix D

Error Projection

As outlined in Section 4, in order to identify the least reliable areas of the reconstructed mesh, we calculate the error for each joint and then project this estimation onto the mesh. Figure D.1 shows the part segmentation and the available joints. We have 24 body parts and 14 joints. Table D.1 provides a clear mapping between the joints and the mesh parts, allowing us to calculate the error for each body part based on the average error of the corresponding joints. For instance, the error of part 1 (right up leg) is the average error of joint 1 (right knee) and joint 2 (right hip).

Part	0	1	2	3	4	5
Joint	6	1, 2	9, 10	4, 5	5	5
Part	6	7	8	9	10	11
Joint	9, 8	9, 8, 12	9	8	0	13
Part	12	13	14	15	16	17
Joint	7, 8	11	0, 1	6	10, 11	6, 7
Part	18	19	20	21	22	23
Joint	12	0	2, 3	3, 4	11	2, 3

Table D.1: Mesh Parts and Joints Association.

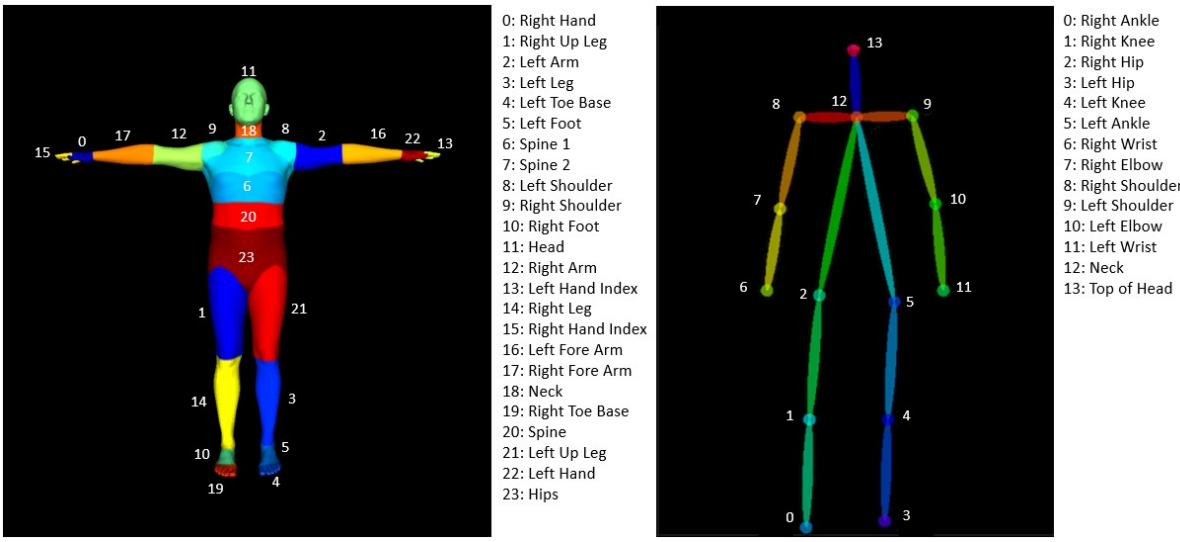

Figure D.1: Name and Number of the Mesh Parts and Joints.