3D Human Reconstruction on the Multi-View, In-The-Wild, YOUth data

Vladyslav Kalyuzhnyy
7050763

Jun 2023

Supervised by
Dr. Ronald Poppe
Metehan Doyran MSc

Department of Information and Computing Sciences
Utrecht University
Artificial Intelligence

Abstract

The main goal of this research thesis is to retrieve three dimensional human body models, of the parent and infant, depicted in the private YOUth dataset, from multiple uncalibrated cameras. The previous research in this area is primarily reliant on ground-truth annotations of two dimensional poses across the multi-view data or the prior knowledge of the camera parameters. To this end, we develop a mechanism which bridges two dimensional pose estimation methods with camera calibration and three dimensional human reconstruction models. To reliably achieve our goal, we study the mechanisms of top-down and bottom-up two dimensional pose estimation methods, as well as, one-stage and two-stage three dimensional human reconstruction strategies. To link the data between these different models, we develop a pipeline which identifies the same individual across sequential frames and different points of view, ensuring to accommodate for missing, or redundant, information. We quantify the quality of the reconstruction based on the estimated two dimensional pose data. The study of the qualitative results show the implications of challenges, such as occlusions and two dimensional pose detection ambiguities, which cannot be accounted for in the absence of ground-truth pose annotations or ground-truth camera parameters.

Project Code: https://github.com/Vlad-2299/3DHumanReconstruction_YOUth.git

Contents

1 Introduction

In recent years, the computer vision field has witnessed remarkable advancements in the reconstruction of three-dimensional (3D) scenes from two-dimensional (2D) data. In particular, the reconstruction of humans, in 3D, has been a highly studied area in the domain. The resulting 3D human models can be used in a vast set of different applications, including social behavior understanding, sports broadcasting, gaming and medical diagnostics. A typical 3D human reconstruction pipeline encompasses several steps in order to obtain its final result. These steps include camera calibration, feature extraction, tracking and the estimation of the 3D human structure. When building such a pipeline, one needs to account for the type of scenario that is being reconstructed. Several works allow for a consistent reconstruction when only a single person is in scene [1, 2]. However, more robust works can achieve competitive results by reconstructing multiple people from a single frame [3, 4]. Nonetheless, depth ambiguities and occlusions, both spatial and temporal, can heavily impact the overall quality of the reconstruction. These challenges introduce errors and ambiguities in feature extraction, as well as body scale estimation, and motion information. To mitigate such challenges, multi-view approaches [5, 6] have been proposed. By possessing multi-view information, obtained from a set of synchronized cameras, developers can achieve a more consistent and robust 3D reconstruction, often taking advantage of known camera parameters. In spite of that, when working with *in-the-wild* data, accurate reconstruction cannot be guaranteed, unless the camera parameters are accurately estimated, as they contain crucial information about the characteristics and position, relative to the scene, of the camera. Thus, the multi-view methods that allow for 3D human reconstruction *in-the-wild* [7, 8], need to initially estimate the missing camera parameters.

Following these preliminaries, this research project has the end goal of performing a consistent 3D human reconstruction on the private, non-annotated, YOUth dataset[1]. YOUth consists of thousands of videos depicting children, of various ages, interacting with one of their parents for a period of time. The playful interaction takes place in a relatively small, static room, with a set of a few toys and is monitored from four uncalibrated cameras and a microphone, where, in some views, the zoom setting is applied throughout the video. For this research, only the video information will be utilized for the reconstruction, outlining the multi-person, multi-view and *in-the-wild* scenario. Essentially, our goal is to reconstruct two close-range interacting agents, by utilizing the temporal information from four different cameras.

1.1 Research Objectives

This research project targets the study of previously developed techniques for 3D human pose estimation. More specifically, our goal is to understand the different approaches employed, given one's available setup and investigate which strategies would best fit to our data. Ultimately, after developing a robust 3D human reconstruction system, we intend to deliver the obtained reconstructions for further research, which will enable the study of the interactive behavior between the two agents. Along these lines, we will dive into the challenges caused by severe occlusions and search for mitigating solutions.

1.2 Research Questions

The current state of the art heavily balances towards monocular 3D human reconstruction systems. Since synchronized multi-view data is more scarce than monocular data, multi-view 3D human reconstruction systems often assume that the multi-view data was collected in a controlled environment. These prior assumptions often require the user to supply the 3D reconstruction system with 2D human poses or camera parameters. The systems which estimate the camera parameters either only allow the reconstruction on one individual, or require the user to pass 2D human poses, along with the identification of the person in order to establish feature correspondences among the different views. Given that the aim of the project is to perform a consistent 3D reconstruction on two people, the main research question is the following: *How accurately can we perform a multi-view 3D human reconstruction, between two close range interacting agents, without ground truth annotations?*

This research question can be divided into the following sub-questions:

- Can we establish an efficient and reliable feature correspondence mechanism to track individuals among sequential frames and different views?
- During prolonged temporal occlusions, will the reconstruction be improved if we discard the deficient view?

[1] https://www.uu.nl/en/research/youth-cohort-study

1.3 Research Outlook

As a consequence of the step-wise approach that one inherently adopts when trying to solve the 3D human reconstruction problem, the following sections are structured in the following manner. After studying some of the most relevant works in 3D human reconstruction, we start by introducing the concept of 2D pose estimation, branching to the contrasting differences of the two proposed approaches, top-down and bottom-up. Following this, we introduce the different training routines that enable the model to lift the previously detected 2D poses, in the case of two-step monocular approaches, into the 3D space. We then turn our attention to automatic camera calibration strategies and multi-view triangulation approaches, discussing different strategies that enable the model to perform 3D reconstruction from 2D data. Afterwards, we introduce the details of the parametric model that will enable the reconstruction of the human mesh, parameterizing its shape and pose.

2 Literature Review

This section presents a discussion about the literature concerning 3D human reconstruction. Even though the studied methods are aimed at the same target, the majority of these methods employ contrasting techniques to achieve their end goal. From studying the literature, our objective is to understand such contrasting implementations and determine to which scenarios they fit best, as often, the choice of intermediate implementations is ruled by specific challenges/optimization that the authors are trying to overcome/implement.

The following Table 1 gives an overview of the relevant studies conducted in 3D human reconstruction, including the current state-of-the-art (SOTA) models evaluated on *Human3.6M* [9], *MPI-INF-3DHP* [10], *MuPoTS-3D* [11], *3DPW* [12] and *HumanEva-I* [13] datasets (for a description of the mentioned datasets please refer to Section 2.6). From each column, respectively, one can recognize which methods allow the 3D modeling of multiple people in one single frame; which methods are employed in a multi-camera setup; which methods don't require the camera parameters to be known beforehand; does the method take into account temporal information; is the 3D pose estimation technique reliant on a prior 2D keypoint estimation to regress 3D positions of each joint; and finally, does the method fit a parametric human body, or does it simply estimate its skeleton.

Method	Multi-Person	Multi-View	In-The-Wild	Temporal	Two-Stage	Model-Based
Wang *et al.* [8]	✓	✓	✓	✓	✓	✓
Iqbal *et al.* [7]	✓	✓	✓	×	✓	×
QuickPose [5]	✓	✓	×	✓	✓	×
MetaPose [14]	×	✓	✓	×	✓	×
Dong *et al.* [6]	✓	✓	×	×	✓	×
Kanazawa *et al.* [15]	×	×	✓	×	×	✓
CLIFF [16] [2]	✓	×	✓	×	×	✓
SPEC [17]	✓	×	✓	×	×	✓
PARE [18]	✓	×	✓	×	×	✓
Ugrinovic *et al.* [19]	✓	×	✓	×	×	✓
Liu *et al.* [1] [3]	×	×	✓	✓	✓	×
P-STMO [2] [4]	×	×	✓	✓	✓	×
Zanfir *et al.* [20]	✓	×	✓	✓	✓	✓
VideoPose3D [21]	×	×	✓	✓	✓	×
Shan *et al.* [22]	×	×	✓	✓	✓	×
SMPLify [23]	×	×	✓	×	✓	✓
Chun *et al.* [3] [5]	✓	×	×	✓	✓	×
Cheng *et al.* [4] [6]	✓	×	✓	✓	✓	×
HybrIK [24]	×	×	✓	×	✓	✓
Yang *et al.* [25]	×	×	✓	×	×	×
Iskakov *et al.* [26]	×	✓	×	×	✓	×
Moon *et al.* [27]	✓	×	✓	×	✓	×
HoloPose [28]	✓	×	✓	×	✓	×
Li *et al.* [29]	×	✓	✓	×	×	✓
Zanfir *et al.* [30]	✓	×	✓	×	✓	✓

Table 1: Overview of relevant studies for 3D human pose estimation

From the works listed above, the ones that best satisfy the criteria of our data were prioritized. Such criteria concern the multi-view, multi-person paradigm, in an *in-the-wild* scenario. For this reason, the first quintet

[2]SOTA on the 3DPW dataset. Leaderboard: https://paperswithcode.com/sota/3d-human-pose-estimation-on-3dpw
[3]SOTA on the HumanEva-I dataset. Leaderboard: https://paperswithcode.com/sota/3d-human-pose-estimation-on-humaneva-i
[4]SOTA on the MPI-INF-3DHP dataset. Leaderboard:
https://paperswithcode.com/sota/3d-human-pose-estimation-on-mpi-inf-3dhp
[5]SOTA on the Human3.6M dataset. Leaderboard: https://paperswithcode.com/sota/3d-human-pose-estimation-on-human36m
[6]SOTA on the MuPoTS-3D dataset. Leaderboard:
https://paperswithcode.com/sota/3d-multi-person-pose-estimation-root-relative

of works outlines the methods that can be deployed in a setup where multiple cameras are available. In one hand, when good quality synchronized and multi-perspective data of the same scene is available, one naturally overcomes the inherent challenges of occlusion and depth ambiguity which one faces with monocular data. On the other hand, the challenge of establishing view-view correspondences arises. Usually, to overcome such an obstacle, researchers first extract 2D poses and then perform the 3D reconstruction through triangulation. The works that best address this issue, according to our criteria, are those of Wang *et al.* (DMMR) [8] and Iqbal *et al.* [7].

Unfortunately, the number of studies that address the multi-view 3D reconstruction of multiple humans *in-the-wild* is very limited. Thus, the remaining listed works only concern single-view scenarios. The second quintet in Table (1), references the methods that directly estimate 3D pose and shape information from a single image, without requiring paired 2D pose annotations. Such methods benefit from the fact that they decouple themselves from prior 2D pose detectors, thus they don't rely on the quality of 2D joint detection. However, they are inherently less robust to domain shifts, as the feature extraction step is more prone to fail in scenarios that it has not seen in the training data, as it uses image information, such as head and limb orientation, to estimate 3D poses. The work of Kanazawa *et al.* [15] often serves as a baseline for other one-stage approaches, such as the ones mentioned in the second group of the table. Due to the nature of the problem, capturing the human pose from a sequence of images might yield a noisy reconstruction. To this end, some works adopt temporal inferences, and capture long-term information, to robustly estimate consistent poses from videos. All the works present in the third quintet of the table benefit from the temporal information available in videos. Finally, without following any hierarchical logic, the last ten works serve only as complementary information to the references in future sections.

2.1 2D Pose Estimation

Early classic methods, as DeepPose [31], formulate the pose estimation as a joint regression problem, by utilizing a 7-layered convolutional deep neural network which learned the natural topology and interactions between joints. With the advances of Convolutional Neural Network (CNN) [32] architectures, researchers were able to increase their 3D reconstruction performance for both single-person [33, 34] and multi-person [35–37] scenarios. For either case, during the following years, different network architectures have been applied to solve the task. Although 2D human pose estimation is naturally a regression problem, historically, these methods haven't been as accurate as heatmap-based techniques, therefore received less attention [38]. The core idea of regression-based architectures is to directly map the input to the output joints coordinates, which is flexible and efficient for various human pose estimation tasks and real time applications. Regression-based architectures are more efficient, but less accurate, than the heatmap-based architectures [39]. Overall, the quality of a pose estimation system is dictated by its robustness to occlusions and severe deformations, success on rare and novel poses, and by its invariance to changes in appearance due to factors like clothing and lighting [33].

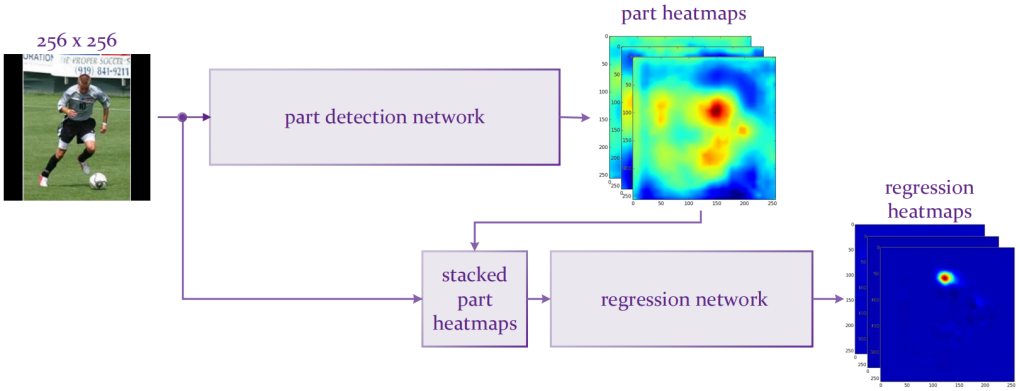

Figure 1: Architecture proposed by Bulat *et al.* [34] that represents the core idea of the most used representation for human keypoints, the heatmap [37]

Bulat *et al.* [34] proposed a CNN cascade, heatmap-based architecture, which serves as a baseline for more recent heatmap-based approaches (see Figure 2.1). The general concept of the proposed architecture can be divided into two parts. The first, a part detection network, which is trained to detect individual human body parts, outputting a set of N part heatmaps. The second, a regression subnetwork that jointly regresses the part heatmaps, stacked along with the input image, to confidence maps representing the location of the specific body part. In general, it is more efficient to use a single heatmap for a single keypoint, as it eases the management of the challenges caused by occlusions and close-range interactions between multiple people [33].

In recent years, transformer-based networks [40, 41] have embraced the task with relative success, leveraging from the advances of Vision Transformer (ViT) [42]. Contrary to the CNN-based models, the attention layers in the transformer-based architecture enable the model to capture long-range relationships efficiently, and also reveal the dependencies that are taken into account when predicting the location of each keypoint. The detected dependencies can provide further explainability of how the model handles special cases, such as occlusions. But nevertheless, these methods are all heatmap-based.

2.1.1 Single-Person

As one can expect, single person approaches are inherently more simplistic than their multi-person counterparts, as they are able to assume more prior knowledge, such as the number of keypoints to be regressed. Conventionally, these systems either require a pre-processing step which crops the image, containing one single person, or the input image must portray a single individual. On the other hand, systems such as Stacked Hourglass Networks, proposed by Newell *et al.* [33], utilize the center annotations, provided with all images, during training. For this reason, the network only estimates the pose of the person in the direct center. In addition to the center

annotation, the network makes use of keypoint visibility annotations, that reflect on the joints which are not visible in the image, but with apparent position. Due to such training routine, the network is able to deal with partial occlusions. Nonetheless, assessing the performance of how the model deals with occlusions can be a challenge, as it often falls into two distinct categories. The fist consists of cases where the body keypoint is not visible, but its position is apparent and annotations are provided. The second consists of cases which have no information about where a particular joint might be, thus if no ground truth annotation is provided, it is impossible to assess the quality of the prediction. To this end, the network has to make strong assumptions of keypoint location [33].

2.1.2 Multi-Person

When inferring the pose of multiple people in a single image, one is faced with a set of new challenges, which where not present in the single person scenario. First, each image may contain an unknown number of people at different positions or scales. Second, close range interacting people induce complex spatial interference that makes the association of parts more difficult. Such examples are occlusions, contact, limb articulations, or loose clothing. Third, real-time performance becomes challenging, as the run-time complexity tends to grow with the number of people present in the image [35]. Nonetheless, multi-person pose estimation techniques have seen rapid advances that try to overcome these challenges, being mainly divided into top-down and bottom-up approaches.

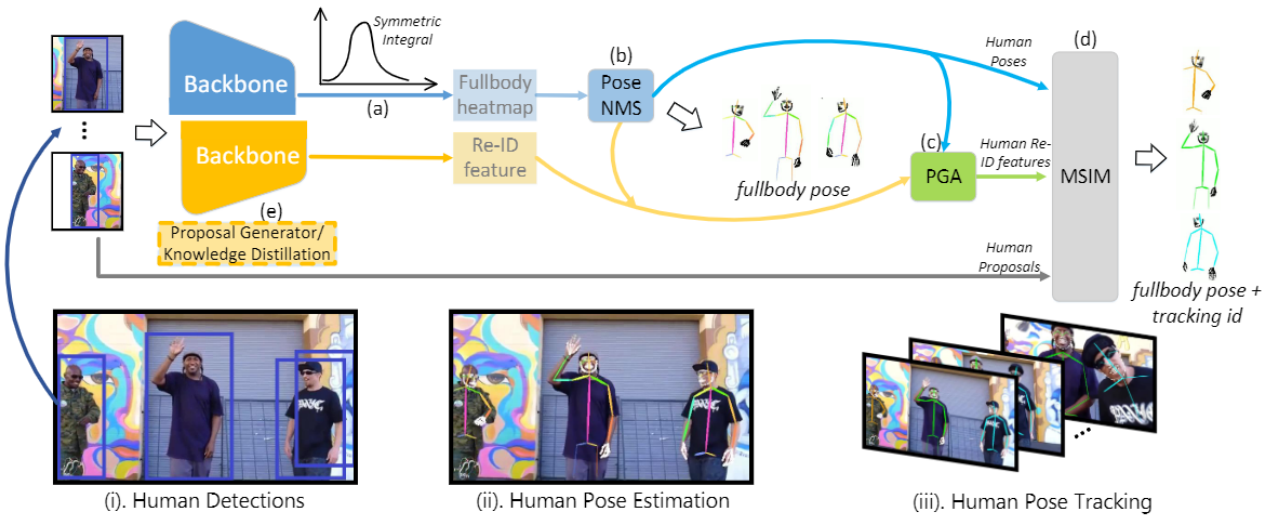

Figure 2: Overview of the AlphaPose pipeline [37]. (i) the full picture is split into a different number of cropped images, resulted from the human detection step. (ii) the localization of keypoints is formulated with the symmetric integral keypoints regression and human pose is estimated. (iii) the pose-guided alignment (PGA) module is applied on the predicted human re-identification feature to obtain pose-aligned human re-identification features.

2.1.2.1 Top-Down Top-down approaches [37, 43–45], interpret the process as a two-stage pipeline. Initially, a detector is employed to predict, and crop, bounding boxes for each person in the image, which are in turn fed to the proceeding pose estimation network. Due to the human detection step, top-down approaches can leverage from prior knowledge about human appearance and pose patterns, which allows them to achieve higher accuracy in keypoint detection. Furthermore, given the human detections over a period of time, tracking algorithms can be utilized.

The commonly adapted, top-down, multi-person, pose estimation method, AlphaPose [37], embraces the challenge by first, given an input image, obtaining human detections using and off-the-shelf object detector such as YoloV3 [7] or EfficientDet [47]. Given the fact that the detection stage and the pose estimation stage are separate, and since the second stage is dependent of the first, in order to alleviate the missing detection problem, AlphaPose lowers the detection confidence to provide more candidates for the subsequent step. The redundant poses are

[7] https://pjreddie.com/darknet/yolo/

then eliminated by a parametric pose Non-Maximum-Suppression (NMS), which works as a pose distance metric to compare pose similarity. In parallel, the cropped human detection images are also forwarded to a tracking network, which obtains its re-identification features, and matches it according to the features in previous frames.

The Mask R-CNN [45], which can be extended for human pose estimation, by adapting the segmentation system to keypoints, implements an additional branch into the pose estimation pipeline. In parallel, with the human detection step, they predict, in a pixel-to-pixel manner, the segmentation mask for each detected individual. Following detection and segmentation, they perform the keypoint detection step, where only a single pixel is labeled as the respective human joint. The results of this method are shown in Figure 3.

Figure 3: Keypoint detection results on the Coco test set using Mask R-CNN (ResNet-50-FPN) for pose estimation, with person segmentations mask predicted by the same model [45]

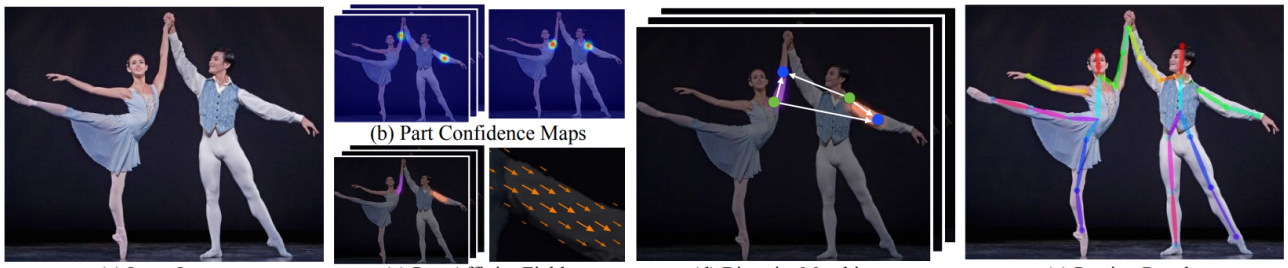

Figure 4: Overview of the OpenPose pipeline. (a) input image that is fed into a CNN to jointly perdict (b) confidence maps for each body part detection and (c) PAFs for part association. (d) corresponds to the matching step that associates body part candidates. (e) resulting assembly of the previous step into a full body pose [35]

2.1.2.2 Bottom-Up Bottom-up approaches [35, 48], as their top-down counterparts, interpret the process as a two-stage pipeline. First, they localize all human body keypoints in an input image, and then employ a clustering technique to group them into each person. Due to this strategy, bottom-up approaches are more robust to occlusions and achieve higher accuracy in dense crowds, when comparing to top-down approaches.

Cao *et al.* [35] proposed Part Affinity Fields (PAFs) that model the association between human body keypoints. In other words, PAFs represent a set of flow fields that encode unstructured pairwise relationships between body parts of a variable number of people, as can be seen in Figure 4. Additionally, Cao *et al.* adopt two concurrent networks that estimate face landmarks and hand keypoints.

Other approaches, such as DensePose [49], perform the association task by implementing human body part segmentation. More specifically, they associate the detected keypoints by segmenting individual limbs/body parts in order to group the respective body joints. However, due to the inherent ambiguity and variability in human

body shape and pose, as well as loose clothing, consistent segmentation of body parts becomes a challenge, which inherently impacts keypoint association. Kocabas *et al.* [48], proposed a similar approach to the top-down Mask R-CNN for pose estimation, but in a bottom-up fashion, where they first perform human keypoint estimation, followed by simultaneous human detection and segmentation.

2.1.2.3 Combined After the advances made in top-down and bottom-up human pose estimation techniques, a few researchers proposed a combination of the both methods. Before deep learning, earlier methods, such as Hua *et al.* [50], introduced a data driven belief propagation Monte Carlo algorithm [51], integrating both top-down and bottom-up reasoning mechanisms. Alternative methods propose Gaussian mixture modeling, or different classifiers for joint and skeleton location [52,53]. More recently, leveraging the advancements of deep learning, the proposed methods that make use of both top-down and bottom-up information, such as Hu and Ramanan [54] employ a hierarchical rectified Gaussian model to incorporate top-down feedback with bottom-up CNNs. Cai *et al.* [55], incorporate spatial and temporal consistencies to alleviate the challenges caused by depth ambiguities and severe self-occlusions, by introducing a spatial-temporal graph convolutional network (GCN).

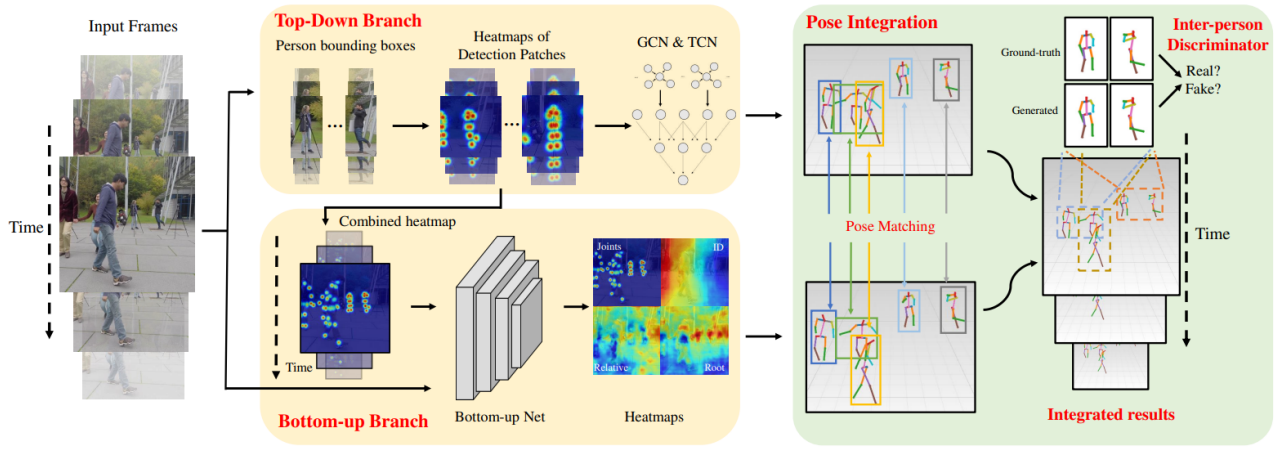

Figure 5: Framework that employs a top-down branch to estimate fine-grained instance-wise 3D pose, and a bottom-up branch to generate global-aware camera-centric 3D pose, proposed by Cheng *et al.* [4]

Cheng *et al.* [4] proposed three main networks for human pose estimation (see Figure 5). The first, a top-down network, estimates human joints from all persons, depicted in the processed frame. The network begins by estimating enlarged bounding boxes around the detected humans, producing heatmaps for all joints inside it, and estimating the ID for each joint to group them into the corresponding person. After estimating the 2D pose heatmaps, a directed GCN is used to refine the potentially incomplete poses caused by occlusions or partially out-of-bounding box body parts. The second network aims to surpass the limitation of the first, which is the lack of global awareness of other persons, since top-down methods perform estimation only inside the bounding box. Thus, they propose a bottom-up network that processes multiple persons simultaneously, easing the estimation of poses in camera-centric coordinates. Acknowledging that bottom-up methods suffer from human scale variations, they concatenate the heatmaps, obtained from the previous network, with the original input frame as the input to the second network. The third and final network takes the output from the previous two networks as input, and finds the corresponding poses.

2.2 3D Human Pose Estimation

Several recent works leverage the advances of deep neural networks in order to directly infer 3D body parameters (more information at Section 2.5) from image features. However, most of the existing methods first estimate 2D joint locations, and, from these, estimate the 3D body parameters. In general, two-stage pose estimation techniques outperform their end-to-end counterparts, as they benefit from intermediate supervision and can better adapt to domain shifts [15]. Additionally, training such a model in an end-to-end manner aggregates new challenges, such as the lack of large-scale ground truth 3D annotations for *in-the-wild* images. Therefore, in order to overcome these shortcomings, researchers have turned to the degree of supervision of their methods, proposing different techniques and network architectures.

2.2.1 Fully-supervised Training

Fully-supervised methods aim to learn a mapping from 2D pose to 3D pose information, given pairs of 2D-3D correspondences as supervision. Therefore, the methods that take this path, directly predict 3D poses from images [56]. However, this is a highly challenging approach, starting from the fact that it is very difficult to acquire large amounts of training images with accurate 3D pose annotations. Leveraging the skewed available data, some methods employ augmentation techniques to further train their networks [57], or incorporate additional data with 2D pose annotations [58], which are easier to obtain. Adversarial losses during training or testing have also been proposed to improve the performance of the models [25]. Overall, full supervision yields the lowest 3D pose estimation errors, but does not allow for *in-the-wild* deployment.

2.2.2 Semi-supervised Training

Semi-supervised methods require only a small subset of training data with 3D annotations, using the remaining samples as unlabeled data. To this end, these methods assume that multiple views of the same 2D pose are available and use multi-view constrains for supervision during the training phase [59]. To achieve this intermediate supervision for 3D pose prediction, it is required to input ground-truth 2D pose annotations, or multi-view imagery with extrinsic camera parameters (for a description of camera parameters see Section 2.3) [21], these are known as two-stage methods. However, while demonstrating impressive results, the main limiting factor is the need of ground-truth 3D data [7], as can be seen by the evaluation tests performed by VideoPose3D, where their semi-supervised approach becomes more effective as it decouples itself from 3D pose annotations [21].

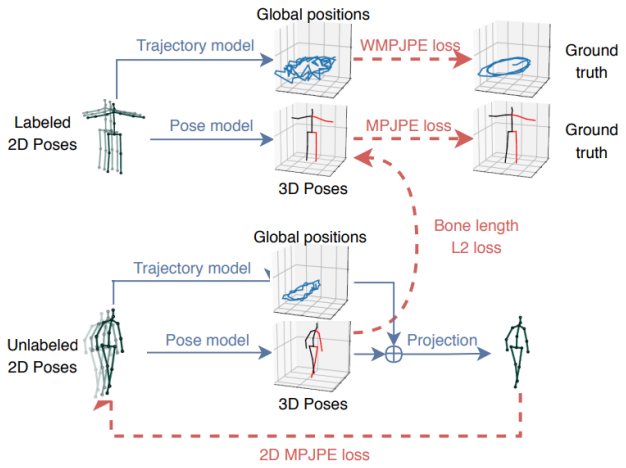

Figure 6: Semi-supervised training with a 3D pose model that takes a sequence of 2D poses as input. 3D trajectories are regressed of the person and soft-constrains are added to match the mean bone lengths of the unlabeled predictions to the labeled ones. Proposed by VideoPose3D [21]

2.2.3 Weakly-supervised Training

Weakly-supervised methods do not require paired 2D-3D annotated data. Essentially, these enable the training with only 2D annotated images. The approaches to obtain 3D pose estimations vary upon the techniques proposed by the researchers. Iqbal *et al.* [7] propose a framework that can be deployed in *in-the-wild* scenarios, where multiple views are available. They approach this task by training an end-to-end framework, using multi-view consistency and employing an object function that can only be minimized when the predictions of the trained model are consistent and plausible across all camera views. Alternatively, Tome *et al.* [60] propose a probabilistic 3D pose model that reasons jointly about 2D keypoint estimation and 3D pose reconstruction. To account for human size variance, the data was normalized such that the sum of the squared limb lengths on the human skeleton is one. Essentially, this approach employs a multi-stage CNN architecture which uses the knowledge of plausible 3D joint locations to refine the search for better 2D joint locations, from a single image, with known camera parameters, as input.

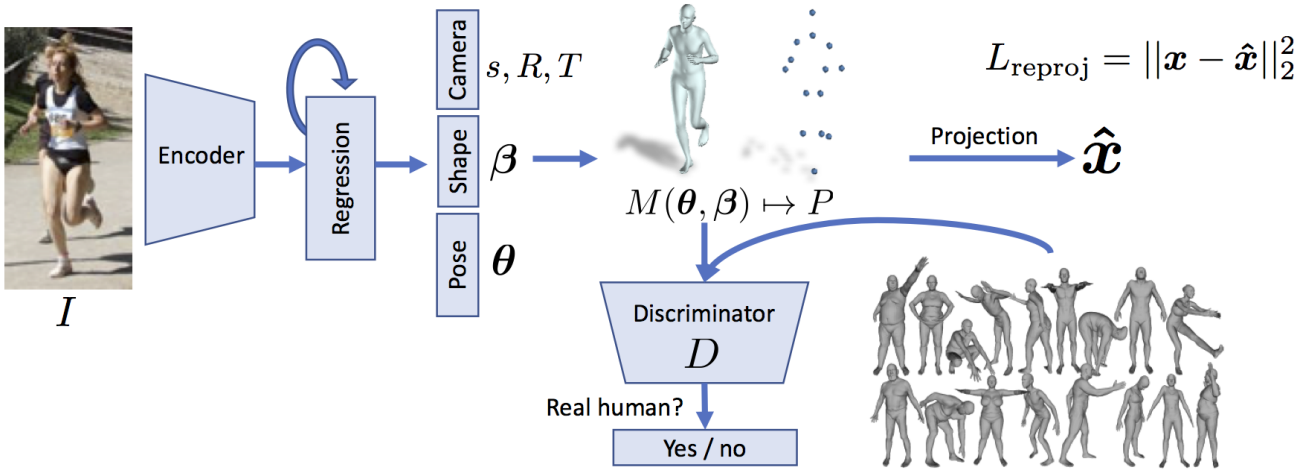

Figure 7: Overview of the framework proposed by Kanazawa *et al.* [15]

In the monocular *in-the-wild* paradigm, Kanazawa *et al.* (HMR) [15] introduce an end-to-end **single-stage** adversarial learning framework (HMR), that reconstructs a full 3D human body from a single RGB image, with no intermediate supervision. The model was trained on images with ground truth 2D joint annotations and it assumes that a pool of 3D meshes of human bodies, of varying shape and pose, is available. The proposed framework begins by extracting convolutional features of the input image, which in turn are sent to a iterative 3D regression module. During regression, the module infers the 3D human body and camera such that its 3D joints project onto the annotated 2D joints. Afterwards, the infered parameters are sent to an adversarial discriminator network wich determines if the 3D parameters are real parameters, based on the available pool of meshes.

HMR's top-down architecture (see Figure 7) was effectively adopted, by many researchers, as baseline [16–19, 61], where each contributed differently for the improvement of results. For instance, Li *et al.* [16] propose Carry Location Information in Full Frames (CLIFF), inferring global rotations, of multiple people, relative to the camera, by feeding and supervising the model with global-location-aware information, in the original camera coordinate system, along with the encoded image, and not the cropped images of the human detection, like was previously done by HMR. Essentially, CLIFF takes into account the different global rotation, of each human body, relative to the original camera. This is accomplished via the CLIFF Annotator model that generates pseudo ground truth SMPL (see Section 2.5) parameter annotations, on an existing 2D dataset, which in turn are used to train the CLIFF model.

2.3 Camera Calibration

2D projections lack the z component of the 3D coordinate space. The key factor in this research is how to recover the lost z component in order to perform 3D reconstruction (x, y, z), from 2D images (x, y). Therefore, camera parameters have to be determined in order to proceed with this task. The computation of the internal (intrinsic) camera calibration parameters can occur simultaneously with the estimation of the external (extrinsic) pose of the camera with respect to a known calibration target [62]. Camera calibration requires estimation of intrinsic parameters such as focal length and principal point of a single camera, as well as extrinsic parameters, namely rotation and translation. However, several factors might affect the camera parameters. In one hand, it is necessary to re-calibrate the intrinsic camera parameters if any mechanical damage or replacement is done. Additionally, zoom modifies the focal length of the lens, thus it directly influences the intrinsic camera parameters. On the other hand, changing the camera's position or orientation will require the re-calibration of the extrinsic parameters.

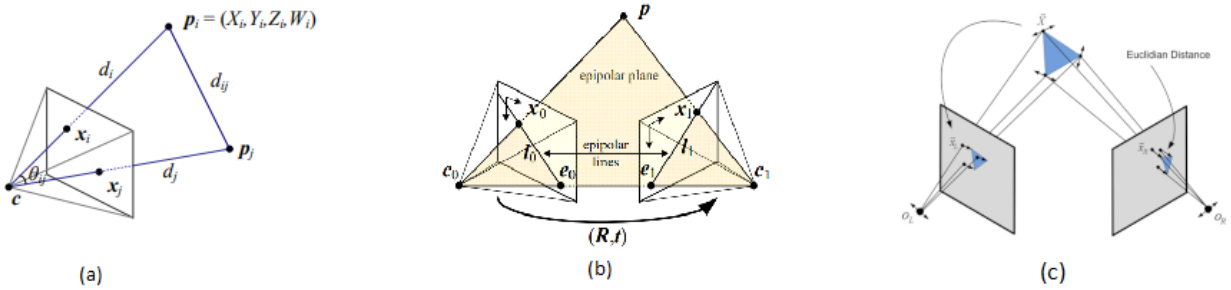

Figure 8: Overview of 2D-3D keypoint mapping with known camera parameters, in monocular (a) and multi-view scenarios via epipolar geometry (b) and bundle adjustment (c) [62]

Most camera calibration methods require the aid of specific calibration patterns, such as a chessboard, since its scale is known and has high contrast features. However, this traditional approach does not work in already recorded videos where no calibration patterns are in sight. Early auto-calibration methods [63] are based on the detection of scene keypoints via the implementation of detection algorithms, such as the Scale Invariant Feature Transform (SIFT) algorithm [64], however these usually have to deal with a large number of parameters, which makes them slow to run, and rarely obtain the most reliable results [65]. Thus, researches have turned to tracking human body pose in order to obtain the extrinsic parameters.

Any approach that uses human body pose, in order to learn the extrinsic camera parameters, has to adopt a pose tracking approach. Given a single-person scenario, identifying the location of the left elbow is relatively easy among the different perspectives. However, when multiple people are present, identifying one single left elbow among the different perspectives becomes a challenge. Iqbal *et al.* [7] adopt a previously proposed pose tracking mechanism [44] that is based on optical flow. Wang *et al.* [8] adopt a CNN termed Omni-Scale Network (OSNet) [66], for *omni-scale* feature learning. Essentially, OSNet learns to detect and match features among the different views, regardless of their scale.

In order to accurately estimate the extrinsic camera parameters, using the human body as a target, researchers often opt for a multi-camera setup. The availability of different perspectives allows to recover epipolar geometry, *structure from motion*, (see Figure 8 (b)), by detecting and matching human body keypoints. A popular optimization-based technique, known in computer vision as *bundle adjustment* (see Figure 8 (c)), recovers *structure from motion* by performing robust non-linear minimization of the measured re-projection errors [62]. Essentially, keypoint correspondences are triangulated using the direct linear transform method [67] (epipolar geometry). Thereafter, in order to minimize re-projection error, a refinement stage optimizes the camera pose (bundle adjustment). Such refinements are often done by a neural network which is initialized with the information acquired in the triangulation step, as was done by Usman *et al.* [14].

Wang *et al.* [8] propose a physics-geometry consistent denoising framework, and a robust latent motion prior,

to remove noisy keypoint detections, and recover the extrinsic camera parameters. They first estimate and track the 2D poses, then obtain the fundamental matrix from multi-view 2D poses, in the first frame, using epipolar geometry with known intrinsic parameters. Essentially, the fundamental matrix tells how keypoints in each view are related to epipolar lines in the other view, and epipolar geometry between two views is the geometry of the intersection of the image planes, having as baseline the line joining the camera centers, as is shown in Figure 8 (b). After this initial, and inaccurate, estimation of the camera parameters, the physics-geometry consistency is applied to reduce noises in the 2D poses from each view. To achieve this, first a set of optical rays is utilized, which come from the optical center of the camera and pass through corresponding 2D joint coordinates. This results in a physical constraint, and enforces the rays from each view to be co-planar. This technique is similar to the bundle adjustment approach. Finally, they simultaneously optimize multi-person motions and camera parameters, by adopting a Variant Auto-Encoder (VAE) [68], using bidirectional Gated Recurrent Unit (GRU) [69] as backbone, which is used to optimize long sequences of frames. This motion prior contains both local kinematics and global dynamics, and can be trained on short motion clips.

2.4 Multi-View Triangulation

As we have seen so far, dealing with occlusions is a geometrically ill-posed problem. Single view methods infer the occluded human parts in a data-driven manner, often making strong assumptions about scenes and human body pose in order to predict the position of the occluded keypoint. Multi-view methods help to deal with this problem by providing multiple perspectives of the scene, essentially offering supplementary information about the pose of each individual. Additionally, when training with multi-view data, one drastically reduces the amounts of information where 2D joint positions are not fully annotated, due to hard occlusions. However, possessing this additional data aggregates other challenges. After estimating the 2D poses for all the N views, one can apply a triangulation strategy, using the known extrinsic camera parameters, and associate each individual among the different views. In this section we will focus on the triangulation methods that aggregate information from multiple views, in order to infer 3D joint coordinates.

2.4.1 Algebraic Triangulation Approach

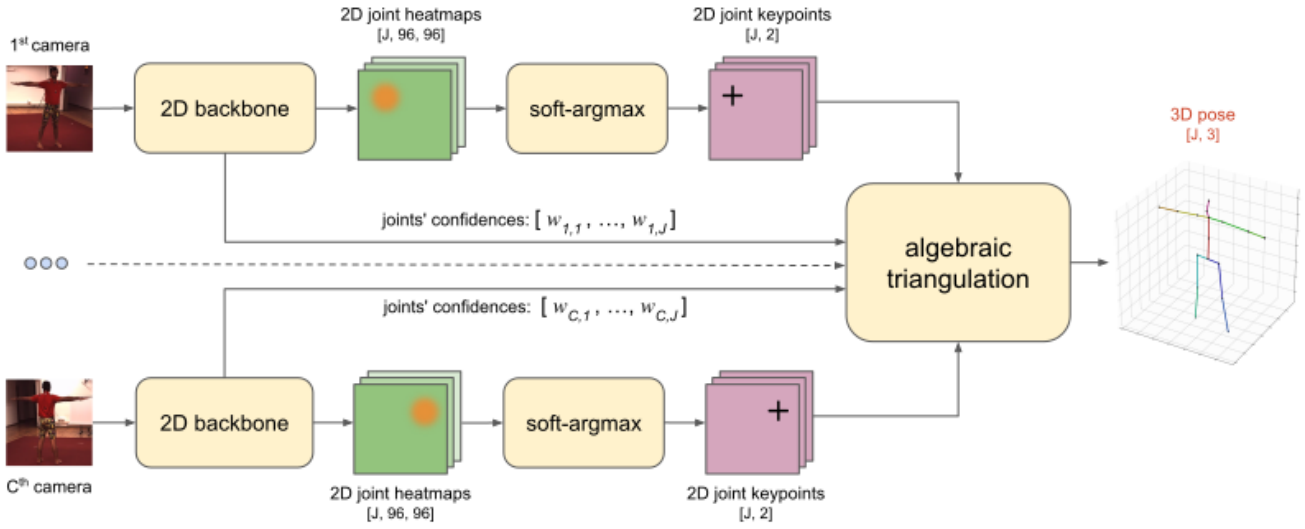

Figure 9: Outline of the algebraic triangulation approach, proposed by [26]

Algebraic methods take as input a set of RGB images with known camera parameters, compute 2D joint heatmaps and infer the 2D joint positions by applying soft-argmax function. The 2D positions of the keypoints are in turn passed to the triangulation module, along with joint confidences, that outputs the triangulated 3D pose using the same introduced logic in the previous Section 2.3 (see Figure 9). Robust triangulation algorithms assume that the joint coordinates from each view are dependent of each other, since in some views the position of

2D joints cannot be reliably estimated, *e.g.* due to occlusions. Iskakov *et al.* [26] address this task by implementing additional weights to the coefficients of the matrix of each view, which are controlled by the neural network branch that is learned jointly with the 2D joint detector. Essentially, the additional weight dictates the contribution that each camera view has on the reconstruction of the specific joint. Thus, the learned weight reflects the confidence of each join detection, which decreases the contribution of the joint that has the least confidence.

2.4.2 Volumetric Triangulation Approach

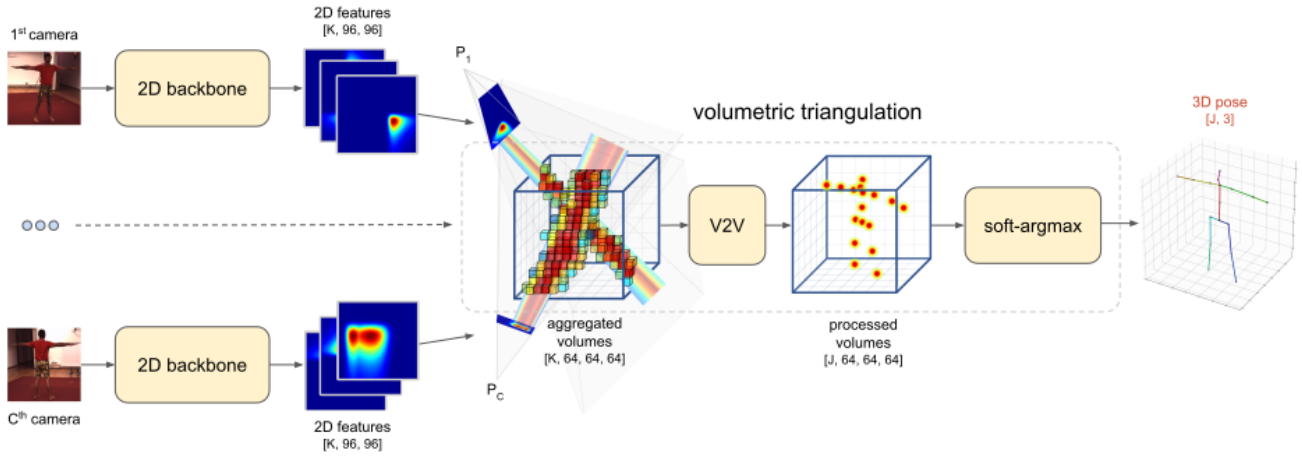

Figure 10: Outline of the volumetric triangulation approach, proposed by [26]

The volumetric approach, contrary to the algebraic approach, processes the images from the different cameras jointly, which enables the model to add a 3D human pose prior and filter out the cameras with wrong projection matrices. The core idea of this approach is to *fill* a 3D cube by projecting the output of the 2D detection along the projection rays inside the 3D cube. The projections, from each view, are then aggregated and processed. Iskakov *et al.* [26] explores three different aggregation methods. The first consists of the raw summation of the voxel data, which causes each view to contribute equally to the reconstruction. Second is the summation of the voxel data, with normalized confidence weights, which dictate the contribution from each camera. For the third method, the softmax of each voxel is computed across all cameras, producing the volumetric coefficient distribution, similarly to the second method. Thereafter, the voxel maps from each view are summed with the volumetric coefficients.

Overall, the volumetric triangulation outperforms the algebraic approach. However, this solution is based on single person reconstruction, and requires that at least two camera views observe the pelvis. Additionally, the algebraic triangulation approach is preferred over the volumetric, as it shares common steps with estimating the extrinsic camera parameters.

2.5 3D Human Body Representation

When inferring the 3D human body pose and shape, researchers often encode the 3D mesh of a human body using the Skinned Multi-person Linear Model (SMPL) [70]. By employing standard skinning methods, SMPL is highly compatible with existing graphics software and rendering engines. Essentially, these traditional methods model how vertices are related to an underlying skeleton structure. A skinned body model defines the vertices of a template T and a rest pose, joint positions J and blend weights W, given the pose of the skeleton θ. Vertex locations of the mesh are computed using linear blending of the verticies based on rotation of different joints. To achieve this, the SMPL model's parameters were trained to minimize vertex reconstruction error on the *multi-pose* and *multi-shape* datasets. Each dataset contains meshes with the same topology as the SMPL model. The *multi-pose* dataset consists of 1786 meshes, also called "registrations", of 40 individuals, where 20 females span 291 registrations and 20 males span 895 registrations. The *multi-shape* dataset consists of a total of 1700 registrations for males and 2100 for females. Since the model decomposes shape and pose, these are trained separately.

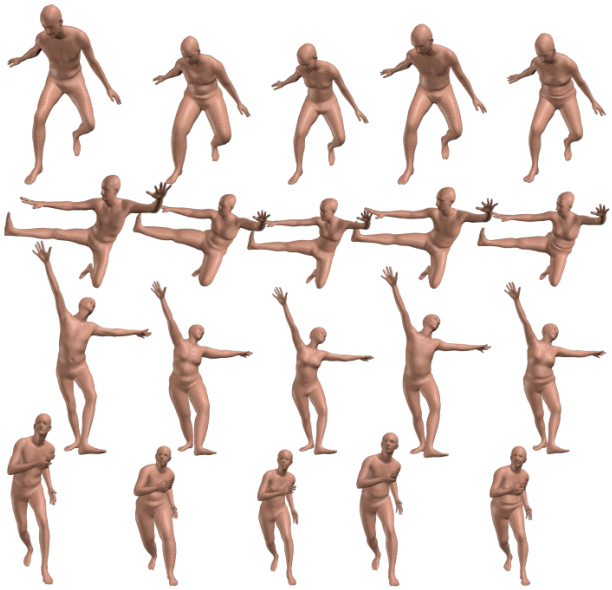

Figure 11: Overview of SMPL mesh results according to variation of shape β (*left-to-right*) and pose θ parameters (*top-to-bottom*)

SMPL provides a differentiable function, which takes dozens of parameters as input, $(\Theta = \theta, \beta)$, and returns a posed 3D mesh of 6890 vertices (V) and 24 joints (K). The pose parameters $(\theta \in \mathbb{R}^{K\times 3})$ consist of the global rotation of the root joint (*pelvis*), with respect to the camera coordinate system, and 23 local rotations of other articulated joints, relative to their parents along the kinematic tree. The shape $(\beta \in \mathbb{R}^{10})$ is parameterized by the first 10 coefficients of a Principal Component Analysis (PCA) of the shape space. The impact that each learned parameter has on the resulting mesh can be seen in Figure 11.

Currently, three SMPL models are available for public use. Since SMPL infers the shape parameters from 2D joints, the full estimation of the 3D shape becomes highly ambiguous when reconstructing the male or female body. To this end, besides the neutral model, which does not take sex distinctive features into account, SMPL offers a separate pre-trained model for the male and female body type. Nonetheless, SMPL will fail to represent accurate body shapes of *e.g.* pregnant women, babies, children, body builders, amputees, etc [71]. Therefore, to account for such body shapes, different models have to be learned. Unfortunately, such variations of the SMPL model are only available for infants younger than 10 months [72].

Overall, the methods that construct a full 3D mesh of the human body by estimating several parameters are known as **model-based** methods, and fall into two distinct categories. Optimization-based approaches estimate the body pose and shape by deploying an iterative fitting process, which tunes the parameters to reduce the error between its 2D projection and 2D observations, *e.g.* 2D joint locations, as is done by Wang *et al.* [8] in the multi-view scenario. However, the optimization problem is non-convex [15] and is likely to fall into local minima, for such a reason, Wang *et al.* introduced the previously mentioned VAE model. On the other hand, without robust solutions to the mentioned challenges, the spotlight is shifted towards learning-based approaches, which use neural networks to regress the model parameters directly, such as the method proposed by Kanazawa *et al.* [15].

2.6 Datasets

Deep learning-based 3D human pose estimation methods perform best when trained on large amounts of quality labeled data. However, different datasets provide different 3D skeleton formats, labeling different anatomical keyopoints. Thus, the methods that utilize these 3D pose annotations for training either stick to one type of skeleton, or apply a conversion mechanism to keep the skeleton consistent. Meanwhile, weakly-supervised methods, that lift 2D poses to 3D poses, in the case of a two-step approach, require only consistency when detecting 2D keypoints. However, the model-based methods that we targeted in this research do not suffer from such discrepancies, as they give less importance to pose accuracy and emphasize on the parametric 3D mesh. For our research, the most relevant datasets are the following:

Human3.6M

Commonly used dataset for training fully-supervised methods, as the data was collected in controlled indoor settings, using a calibrated multi-camera motion capture system. Contains a total of 3.6 million 3D human poses and corresponding images, depicting 11 professional actors (6 male and 5 female) and contains 17 different action scenarios, such as taking a photo or smoking. Everything was recorded from four different view angles [9].

MPI-INF-3DHP

Consists of more than 1.3 million annotated frames. It records 8 actors performing 8 different activities from 14 camera views [10]. This dataset is collected both indoors and outdoors with a multi camera marker-less MoCap system, which causes the 3D annotations to have some noise [15]

MSCOCO

Large-scale *in-the-wild* 2D human pose dataset[8]. This dataset utilizes the same 2D human pose format as the *Human3.6M* dataset.

3DPW

3DPW is an *in-the-wild* dataset with ground-truth SMPL parameters [12].

Halpe-FullBody

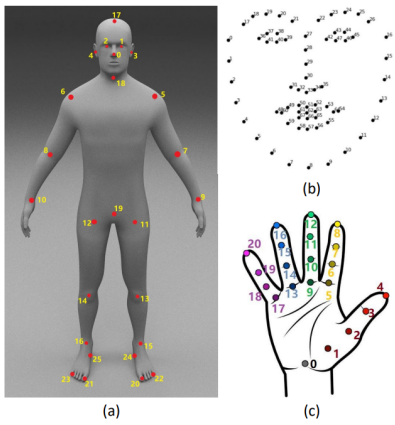

Figure 12: Halpe dataset kepoint format: (a) body and foot, (b) face, (c) hand

Halpe is a dataset introduced by *AlphaPose* [37]. The goal of it's development was to facilitate the research on whole body human pose estimation. Each person is annotated in total with 136 kepoints, including 20 for body, 6 for feet, 42 for hands and 68 for face. For training, the researcher can select which keypoints the model will learn, *e. g.* those of the body and feet, totaling at 26 keypoints.

[8] https://cocodataset.org

3 Methodology

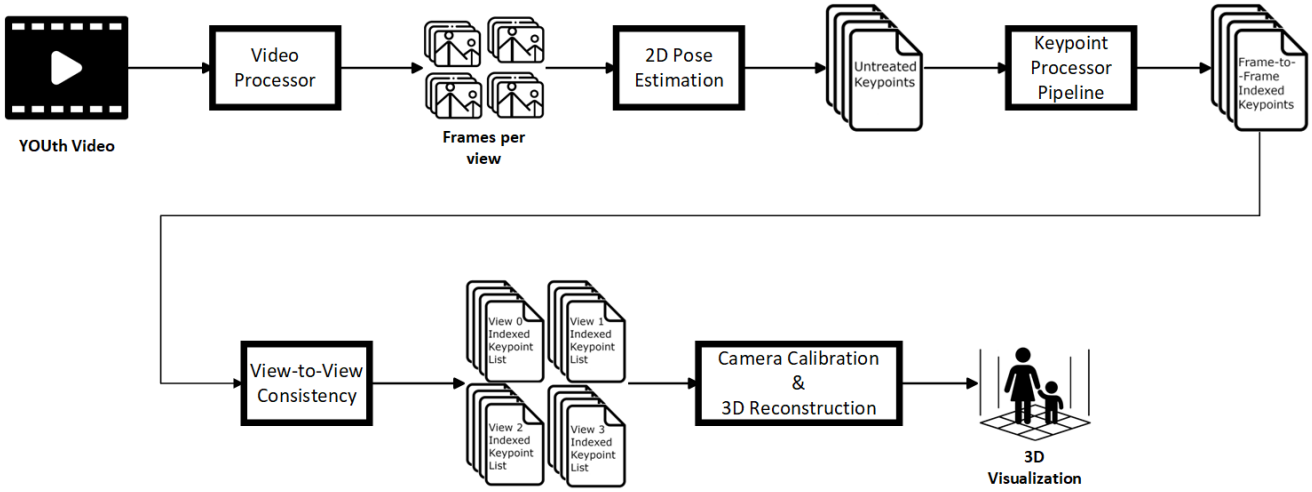

Figure 13: Simplified overview of the step-wise data process of the developed framework.

This section underlines, and motivates, the structure of the developed framework. Figure 13 illustrates the flow of data throughout the system, as well as which systems are responsible for each step. Based on the literature study conducted in Section 2, we elect the systems which are most fit to our data and goal. Bearing in mind that our multi-view, multi-person data has no ground truth pose annotations, nor information about the camera parameters, we opt for a model-based two-stage framework.

3.1 Structure of the Framework

The ultimate goal of the framework is to perform a consistent 3D human reconstruction on the YOUth data. Contrary to many 3D human reconstruction multi-view systems, as input, our system only takes video data and follows a specific set of prior assumptions. The established assumptions are guided by the nature of the YOUth data. Our framework assumes that a maximum of two individuals are captured, per different view, and that no camera panning or zooming is applied during the processed video.

After specifying the video to process, as well as the initial and final time stamps, the system will break down the video into frames, per each view (see Video Processor 3.2). Once all the frames are saved, an off-the-shelf 2D keypoint regression model is employed to extract the joint locations of all the people in each frame (see 2D Pose Estimation 3.3). After processing a batch of frames, the extracted keypoints are saved into a *json* file. However, the data obtained from the previous step is prone to be noisy and dynamically inconsistent. In other words, the number of detections, per each frame, is volatile and does not always represent the same individual which was captured in the previous frame. To this end, we developed a keypoint processor pipeline which removes unreliable detections and interpolates missing data. Additionally, the pipeline ensures that the detections are frame-to-frame and view-to-view consistent (see Keypoint Processor Pipeline 3.4). With the prior knowledge that the same two people are in scene, we annex the 2D poses of the same individual throughout the frame sequence. Off-the-shelf 2D pose estimators cannot perform this step as they expect a variable number of people to be detected and do not keep track of the same individual throughout the video. Nonetheless, complementary re-identification models can be employed to aid this step. However, off-the-shelf re-identification models also suffer from the assumption that a variable number of people might be in scene, and that new individuals might be detected throughout the frame sequence, often failing to re-identify the same individual across a different point of view.

Once all keypoints are processed, these are fed to a dynamic multi-person, multi-view, *in-the-wild* mesh recovery system (see Camera Calibration and 3D Reconstruction 3.5). Once again, with the prior knowledge of the data that we want to process, we can interfere on the reconstruction of each human body mesh, and adjust its characteristics accordingly to the body of the represented person, making the reconstruction more realistic.

Finally, after estimating the meshes, for each frame, as well as the camera parameters, these can be visualized in an independent visualization program (see Section A.1).

3.2 Video Processor

As our initial step, we employ the multimedia framework *FFmpeg* [73] to crop the original four view video into separate single view videos, preserving video quality and discarding sound data. Based on the original four view video, throughout the framework we declare that the top left view represents view 0, the top right represents view 1, bottom left represents view 2 and bottom right represents view 3. For each of the single view videos, using *FFmpeg*, we extract and store 30 frames per second (FPS), always ensuring to preserve image quality, since low quality images are more prone to generate noisy data in future steps. *FFmpeg* was used since it can be applied for both tasks in this step.

In order to avoid overloading the volatile memory and the graphics processing unit of the computer in the next step, the data was divided into different sub-folders, referenced as frame batches. Given each view, frames are stored in their respective folder. Inside each view folder, the frame sequence is divided in batches of 100 frames. By doing this operation, the future step (2D Pose Estimation 3.3) will treat each batch as independent information, consequently losing all temporal data at each new batch. We underline that this has no negative consequences for the processed data, and that after the 2D pose estimation the data is treated as a whole, and not in batches.

3.3 2D Pose Estimation

Considering that camera parameters can be approximately estimated by triangulating tracked features from different views, we employ a top-down human pose estimator, *AlphaPose* [37]. Unlike bottom-up mechanisms, which detect individual keypoints, with *AlphaPose* we are able to perform keypoint tracking by utilizing the human bounding box detection and deploying, in parallel, an off-the-shelf human tracking mechanism, saving computation time. Additionally, from the studied top-down pose estimators, *AlphaPose* was the one to reveal better results, on top yielding poses in the same format as required for the SMPL human body model. Opting for another model, with a different 2D pose format, would induce ambiguities in establishing the keypoint position in the 3D human body mesh, since these are annotated differently.

While executing *AlphaPose*, at each frame, for each human detection, a dictionary object is created to specify the name of the frame, the detected keypoint and human bounding box coordinates, as well as personal tracking identification (ID). In addition, with the intent of not losing any temporal information, for frames where no detections are captured, we create a dummy detection dictionary, containing the name of the frame and a track ID flag of 0.

Since top-down architectures begin by estimating the location of each person, we opt for the *YOLOX* [74] model as human detector. Following the human detection, we elect the pre-trained *FastPose* model as joint regressor, with a *ResNet50* [75] backbone, which follows the same skeleton structure as the SMPL model (see Figure 14). Most importantly, we opted for this model due to its scalability for capturing facial and hand keypoints, which can be useful for further analysis of the YOUth data. The employed joint regression model estimates 26 keypoint locations for each detected person, as well as the confidence score of each keypoint. The confidence score will be used as a quality measure of the detected keypoint in future steps, since occluded keypoints are expected to yield a lower confidence score. In parallel, we employ *Torchreid* [76] as the pose tracking module, which aims to re-identify people across different camera views and across different frames within the same view. At each frame, *Torchreid* assigns a unique ID number, starting at 1, for each new person in scene. If the person is recognized by *Torchreid* from previous frames, the same ID number will be used for the current detection. However, the temporal memory of *Torchreid* is only preserved while processing the same frame batch. Finally, given the number of data batches in each view folder, created in the previous step (Video Processor 3.2), a single file is created containing all the information of all the detections in the respective frame batch.

3.4 Keypoint Processor Pipeline

Given the raw output data generated in the previous step (2D Pose Estimation 3.3), the pipeline aims to output, for each view and for each frame, a single *json* file that contains the parent's keypoints in the first list index and the child's keypoints in the second index. Noting that we have no prior knowledge about the location of the parent nor the child, this pipeline takes a step-wise approach in order to disambiguate the identification

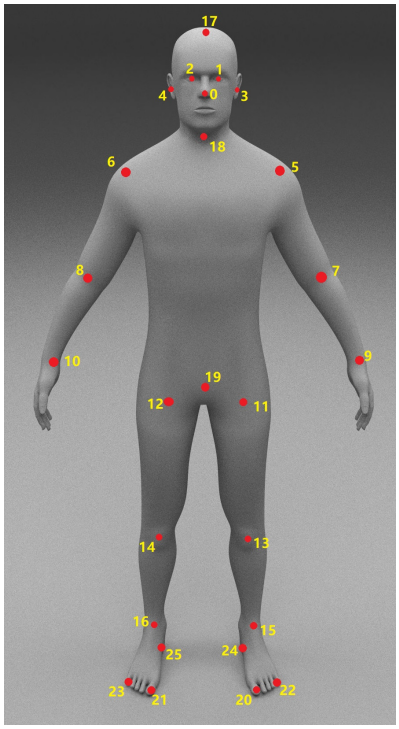

Figure 14: Halp human body skeleton format.

of both individuals. Initially, the pipeline assumes that each frame should contain two detections. However, this is not the case with the raw output data. We need to account for noisy detections, such as the detection of the doll toy, or situations where two detections are generated for the same person. Therefore, the initial Step 3.4.1 ensures to remove excessive detections, and the following Step 3.4.2 ensures to match and identify the detections of each individual. Additionally, temporal and spatial occlusions are a frequent challenge in our data. It is often encountered that only a single individual was captured in the 2D pose estimation step. In order to deal with the challenges of temporal occlusions, Step 3.4.3 ensures that we don't have incomplete data, and interpolates the missing keypoints of the missing individual. Finally, in order to combine the information among the different views, Step 3.4.4 distinguishes which detections belong to which individual across the four views.

3.4.1 Discard Extra Detections

On one hand, we discard initial and final frames where detections are missing. Given the fact that one can only interpolate values given an initial and final range, the first and last frames, which contain less than two detections, are discarded. If such is the case for one view, the remaining views get padded in order to preserve temporal consistency. Therefore, the initial and final frame is preserved only if, among the four views, we have a minimum of two detections.

On the other hand, we discard detections in frames which contain more than two detections. The disambiguation process falls into two paradigms. The first checks if two of the track IDs in the current frame match the history of the previous frame's track IDs. If such is not the case, we calculate the keypoint (k) overlap between the current frame (f) detections (N) and the previous frame ($f-1$) detections (M) (see Equation 1). Given all ($N \times M$) overlap values, we match n to m if the average overlap value of all 26 keypoints is the closest to zero. In addition, this step applies a unique logic for the initial frame's detections. Here, we discard the detections that have the lowest bounding box perimeter. Dispite being prone to errors, we perform this step in order to avoid anchoring the detection of the doll toy to the detections of one of the individuals. If we did not apply this logic, the detection of the doll would stay consistently anchored throughout the entire frame sequence, in which it was detected. On the other hand, this logic might ignore the detection of the child, if two parent detections are

captured. However, due to the high volatility of poses, due to movement, we are able to fix this error in future steps.

$$overlap_{f_n, f-1_m} = \frac{\sum_{k=1}^{26} |k_f - k_{f-1}|}{26} \tag{1}$$

3.4.2 Frame-to-Frame Consistency

This step in the pipeline aims to anchor the detections of the same individual to the same list index. Given the unstructured detection list, with a maximum of two detections per each frame, we iterate over all the frames and compare the similarity between the current frame detections, and the previous frame detections. If a previous frame doesn't contain any detections, the pipeline will trace-back to the last frame with at least one detection. In order to consistently identify the correct index for the current frame detections, we first need to verify how many detections are present in the current and previous frame. Additionally, due to the high ambiguity in track ID assignment, our main similarity metric between two detections is keypoint overlap, which is calculated using Equation 1.

There are four main matching processes that this algorithm takes into account:

1. The first matching paradigm is selected when both the previous and current frame have two detections. For this scenario the algorithm starts by identifying if the current detection's IDs match with the previous detections IDs, indexing the current detections to the same index as the previous. If this is not the case for at least one current detection, the algorithm identifies the best index for both current detections, given the overlap values with the previous two detections. If both IDs don't match, the algorithm calculates the keypoint overlap between both current detections, the overlap between the first previous detection with the first current detection and the previous second detection with the current second detection. By calculating these three overlap values we are able to identify scenarios in which the current frame has two detections, but for the same person. The double detection of the same person happens when the overlap between the two current detections is lesser than both overlap values with the previous detection or if the indices in the current frame are swapped in comparison to the previous frame. Thus, in order to disambiguate this situation, we calculate the bounding box overlap of all the detections, and index them accordingly. If the keypoint overlap between both current detections is not lesser than the other two overlap values, we assign the current detections to the indices that best match the previous detection.

2. The second matching paradigm is selected when the current frame has two detections, and the previous only one. We begin by identifying the indices of the previous missing detection, which has a null value. Thereafter, the algorithm verifies if the previous detection ID value matches at least one of the current detections ID values. If not, we traverse back until we find a previous frame with two detections. This procedure ensures that if noisy detections get assigned, we are able to recover the correct index order. In practice, this happens when we have one detection of an individual, and one detection of the toy doll. However, if the ID value matches at least one of the current detections, we apply the same logic that we used on the previous paradigm but we start by calculating the three overlap values.

3. The third matching paradigm accounts for scenarios in which we have one current detection, and one previous detection. For this stage, the algorithm does not begin by matching the ID value with the previous frame, instead, we trace back to the frame in which the missing person in the previous frame is present. Thereafter, we calculate the keypoint overlap between the current and previous detection, and current detection with the recovered missing detection, matching the current detection to the index with best overlap value.

4. The final matching paradigm accounts for scenarios in which we have one current detection and two previous detections. This process matches the current frame to the index of the previous detection which best overlaps the current detection.

Given the structure of this disambiguation algorithm, we are able to approximate the number of times *Torchreid* wrongly identifies an individual and the number of times *AlphaPose* generates two detections for a single individual. Once the algorithm detects that the current first detection ID is present in the second detection ID history list, or vice-versa, we increment the number of *Torchreid* errors. Meanwhile, *AlphaPose* errors are incremented

once the algorithm detects that the both current detections have a better overlapping value than the overlapping values between the current and previous detections.

3.4.3 Interpolate Missing Detections

After indexing the detection data, we can distinguish the frames in which the detection step failed for a particular individual. Due to the loss of information, this stage aims to linearly interpolate the unknown keypoint values that fall between two existing detections. Given the keypoint value lists of detection n at frame f and the detection m at frame $f + I$, with I corresponding to the number of frames in which the individual detections are missing. For each keypoint value we calculate the difference between k_f and k_{f+I}, and the resulting stride value is the quotient between the keypoint difference and I. Thus, for each missing detection ($i \in I$), we calculate the keypoint value by adding k_n to the stride value times i. We note that this step will interpolate all the keypoints between any given range, regardless of the I value. However, interpolated frame ranges are stored to be taken into account in future steps, allowing to discontinue the data from the view in which I exceeds a maximum threshold.

3.4.4 View-to-View Consistency

This step aims to match the indexed data, resulted from the previous steps, across the four views. Previously, we treated the data from each view independently. Now, our goal is to merge the data from the different views, by associating the detections in one view, to the detections in another view. In essence, our goal is to identify the same individual across the data obtained from different cameras. To do so, we deploy and fine-tune, an instance segmentation Mask R-CNN *Detectron2* model[9]. We opted for *Detectron2*, which encapsulates the Mask R-CNN model, since it has built in support to fine tune the model on custom COCO datasets. To this end, we perform the fine-tuning on the COCO Human Instance Segmentation dataset[10], changing the output layer from 18 possible classes to 1. By changing the number of possible outputs, we inhibit the model of regressing the bounding boxes and the segmentation instances for the unnecessary classes, saving computation time. During literature review, while we were evaluating the advantages of one-stage frameworks, we noticed that works such as of Ugrinovic *et al.* [19] used *Detectron2* as their semantic segmentation model. Initially, we deployed *Detectron2* with the same goal as Ugrinovic *et al.*, however, with the advances of the research and the preference for two-stage frameworks, we utilized the same model, but for different purposes.

In order to save computation time, and to avoid calculating the instance segmentations on all available frames, we deploy the instance segmentation model on the first and last frames of each frame batch, which resulted from the video processor step (Section 3.2). For videos in which Step 3.4.1 deleted more than 3 batches (300 frames), we include the middle frame of each remaining batch, and for cases where 5 (500 frames) or more batches got deleted, we add 5 frames per batch. Essentially, this step counts how many available frames there are for the reconstruction, and populates the *Detectron2* data file accordingly, ensuring that we don't have nor too many nor too less frames to calculate the instance segmentation on the human bodies. In parallel, for all the stored *Detectron2* frames, we store the bounding boxes of all the detections generated by the 2D pose estimation (Section 3.3). In addition to storing the original video frames, we create a duplicate frame list where we follow a color transfer algorithm [77]. In essence, we normalize the colors of the remaining views, given view 0. For a list of four frames, same frame per four views, we convert all frames to the CIELAB (L*a*b*) color space, where a small change in an amount of color value produces a relatively equal change in color importance. Thereafter, given the source frame of view 0, and a target frame of any remaining views, we compute the mean and standard deviation for each of the L*a*b* channels of both frames, which represents the distribution of colors in the frames. In order to normalize the color statistics of the target frame, given the source frame, we subtract the mean of each channel in the target image, multiply the quotient between the standard deviation in the source and target frame , and add the mean of the corresponding channel of the source image, ensuring to preserve the 0 to 255 pixel value range for each color channel. This is done to account for the different color pallets in each camera. Nonetheless, we execute the instance segmentation model on the original frames, in order not to lose any data quality. Once in execution, *Detectron2* begins by predicting bounding boxes around each detected human, and for each bounding box we obtain the segmentation instances, which highlight the pixels that contain the detected human body. After execution, for each frame in each view, we store the segmentation instances together with the bounding box information. With this data, our initial goal is to follow the same indexation principle as the list of *AlphaPose*

[9] https://detectron2.readthedocs.io/en/latest/
[10] https://cocodataset.org/#explore

detections, from Step 3.4.2. This will discard all the redundant *Detectron2* detections, keeping only the most representative segmentation of each individual. To achieve this we compare and match the bounding box lists of both detection models, following Equation 1. Once correctly indexed, detection masks are generated on the normalized images, which contain only the values of the pixels which were segmented. Finally, for each mask, a color histogram of 16 color bins is calculated and normalized. We execute *Detectron2* on the original frame and not on the cropped bounding box, which resulted from *AlphaPose*, due to the fact that we cannot guarantee that only a single person is depicted in the cropped frame.

To avoid repeated histogram comparisons, in each view, for each individual, we select the frame that has the best bounding box overlap score between the two bounding box lists. In essence, the algorithm finds the most representative segmentation of the individual, given the *AlphaPose* bounding box. The similarity of a pair of histograms is measured using the Chi-square distance, which quantifies the difference between observed and expected frequencies in the histograms. Thus, the lesser the Chi-square distance, the more similar the histograms are. Given that this metric calculates the difference, we ensure to calculate, normalize, and average the difference between histogram one and two and histogram two and one, guaranteeing symmetry in the operation. Given the four views, two detections per view, and the normalization of the Chi-square distance, we perform a total of 48 ($4^3 \times 2^2$) comparisons. Here, we underline that for each pair of views, we compute two commutative similarity values which indicate the similarity of the histograms of the detections in the same list index, and the histograms of the detections in opposite list indices. Thus, if the first item is lesser than the second, we know that the detections are view-to-view consistent. Once all the comparisons are performed, the algorithm will elect the views which are not consistent with the remaining, by following the iterative logic illustrated in Figure 15. The disambiguation process begins by generating two value matrices, the first containing the similarity values between the detections in the same index and the second matrix stores the similarity values between the detections in different indices (see Tables (a) in Figure 15). Given the sum of the scores in each column, staring at view 1, the algorithm iteratively decides if the the detections of the same individual are interchanged, among the two views. For instance, if the algorithm deems that the detections in view 2 are interchanged in relation to the other views, the row values of that same view will be also interchanged, as is the case of iteration (b) to (c) in Figure 15. As a result, a boolean variable is stored to indicate if all the detections of the individuals should be interchanged, or not.

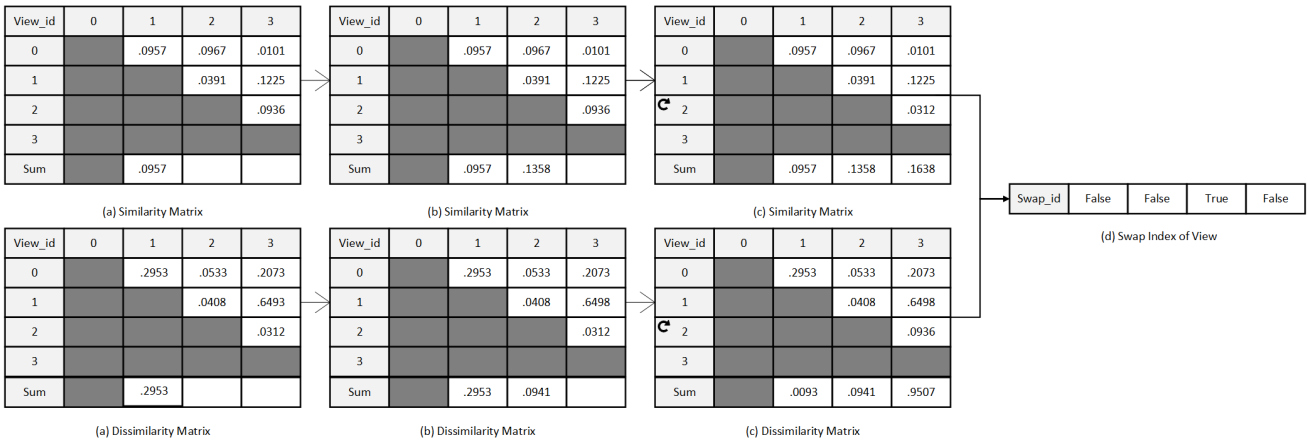

Figure 15: Illustrative example of the person inter-view identification process. The values concern the practical case of view-to-view person identification of video *B45358*

Once the *AlphaPose* detections are consistently indexed at each view, we can merge the information from all the views and identify which detections belong to the infant and which detections belong to the parent. Assuming that the parent's body size is larger than the child's, for each frame we calculate and average, among the four views, the bounding box perimeter of each individual. Consequently, the average of the computed values will determine which detections belong to which individual. This step utilizes the prior assumption that two people

are always present in scene and that no camera panning or zooming is applied.

3.5 Camera Calibration and 3D Human Reconstruction

As input, this step takes the data returned from the keypoint processor pipeline (Section 3.4). At this stage, the 2D keypoint data is assumed to consistently contain two detections per frame. Meaning that missing data was interpolated and redundant detections were removed. Moreover, it is assumed that, in the four different frame sequences of each view, for each frame detection list, the first index contains the 2D keypoints of the parent, and the second list index contains the 2D keypoints of the infant. Having the 2D keypoint information, of each individual, per different views, we utilize the works of Wang *et al.* [8] *DMMR* to calculate the position of the cameras and keypoints in the 3D world.

Given our multi-view, *in-the-wild* data, and the literature review conducted in the previous section, the other candidate to perform the computations of this step would be the works of Iqbal *et al.* [7]. However, despite not benefiting from temporal information, the possibility of using the candidate method was discontinued due to the fact that it yields reconstructions of the human body skeleton, and not a human body mesh. Additionally, the method proposed by Iqbal *et al.* is not publicly available, making it out of reach for this research project. On the other hand, the elected *DMMR* method, on top of benefiting from temporal information, by utilizing the VAE mechanism, described in Section 2.3, is able to calculate both the camera parameters, and the parameters required to fit the parametric human body model, given 2D keypoints. By fitting a human body model, we are able to yield realistic reconstructions of the individuals, which capture body, limb and head orientation.

In essence, this step is composed of two phases. The first phase targets the optimization of the camera parameters, over the entire sequence of frames. The first phase is performed for each different YOUth video, since it is assumed that the cameras are not consistently in the same position, or zoom setting, throughout all the videos in the database. However, it is also assumed that the camera parameters are consistent throughout the frame sequence. Thus, the second phase takes the previously computed camera parameters, and fixes them throughout the new reconstruction. Essentially, the main goal of the second phase is to calculate the 3D poses of both individuals. Given that the second phase takes fixed camera parameters, our target is to minimize keypoint re-projection error.

Once the framework reaches this stage, *DMMR's* data folder is automatically updated with all the information from all the four views. However, during the data population process, one can define the maximum number of consecutive frames in which the 2D pose estimation (Section 3.3) failed to generate a detection for one of the individuals. If the defined threshold is met, the framework discards the deficient view during the frames in which the detections are incomplete. This mechanism was developed with the goal of minimizing keypoint re-projection error. Essentially, it enables *DMMR* to use a variable number of camera information during the reconstruction of the entire frame sequence, without losing any temporal information. We underline that if the missing data overlaps between two views, for the shortest intersection of ranges, we utilize the interpolated poses in order to preserve a third view. If the mechanism detects a case in which two views require to be discarded at the same frame ranges, an error message will be displayed, halting the splitting of the data. *DMMR* utilizes both the 2D keypoints data and the original frames in order to fit the parametric human body model, *SMPL Neutral*. Original frames are required for this step since a sub-mechanism within *DMMR*, *SPIN* [78], utilizes the frame information to perform iterative fitting on the 2D joints, essentially optimizing the regressed shape and orientation of the *SMPL* model. Unfortunately, the *SMPL Neural* model is not trained to be fitted to the body proportions of a 10 month old child. The only possible alternative for this scenario would be to employ the *SMIL* model [72]. Even then, *SMIL* is not the most representative model for our scenario since it was fitted for the body proportions of preterm infants, and requires the input of a depth channel, which is captured from RGBD cameras. On top of that, we elected the *SMPL* model due to its compatibility with mechanisms such as *SPIN*.

Before fitting the 3D *SMPL* model, one can specify a pair of parameters which will guide the reconstruction process. The first boolean parameter determines if the camera parameters should be optimized to the data, or stay unchanged. The second parameter reflects on the scale of the *SMPL* child model. Given that both individuals have significant body scale differences, the *SMPL* model of the parent has a constant scale of 100%, and the child as a pre-defined scale value of 45%. The child scale value was defined based on the international growth charts of children aged between 0 and 59 months, released by the World Health Organization (WHO) [79]. The research indicates that, on average, the body scale of a 10 month old child is approximately 25% to 30% compared to that of an adult. However, these values are nuanced for our reconstruction, since we aim to scale down an adult sized

human body. Therefore, in order to avoid reconstructing a miniature adult body, the scale parameter of the child is fixed at 45%.

A set of camera parameters was pre-computed on the YOUth data and was saved as an initial guess for all the upcoming reconstruction procedures. In order to do so, we manually selected four pairs of intrinsic and extrinsic camera parameters from the MHHI [80] dataset, ensuring that the MHHI cameras follow the same resolution scheme as the cameras that captured the YOUth data. Thereafter, we utilized *DMMR* to adjust these camera parameters on the demonstration video of the YOUth data. Once fitted, we save these camera parameters to be utilized as initial guesses for future reconstructions. When optimizing the camera parameters, the re-projection error is taken into account. For each frame, the camera parameters are jointly optimized with the triangulated keypoints in order to reduce re-projection error between the estimated keypoints and the input 2D keypoints. Additionally, during the triangulation of keypoints, input 2D keypoint confidence score, which resulted from the 2D pose estimation step (Section 3.3), is taken into account. This will prioritize the keypoint location at the view in which it has the highest confidence score. For instance, if spatial occlusions are present in one view, the keypoints of the occluded body part are roughly guessed by the 2D pose estimator, yielding a lower confidence score value. Consequently, this mechanism will prioritize the triangulated keypoint location over the remaining views, where the keypoint at hands is visibly detected. However, by enabling this feature, we introduce high levels of ambiguity, while calculating the 3D human poses, for the cases which only a single view captures a specific keypoint or keypoints. If this happens, the reconstructed pose will only be representative for the view in which the keypoints are detected, failing to accurately triangulate the remaining keypoints across the view.

Given one's aim with this step, the reconstruction phase is ruled by the value of the camera optimization parameter. After executing the first phase, by optimizing the camera parameters, the current method returns the optimized camera parameters, over the entire frame sequence, the reconstructed body meshes and the re-projection errors of each keypoint in each frame. Consequently, to proceed to the second phase of the reconstruction, the user has to update the camera parameters, within *DMMR's* data folder, with the outputted camera parameters from the first phase. On top of that, the camera optimization flag is required to be turned off in order for the second phase to be fully initialized. Once this is done, the second phase will return the optimized body meshes and the respective re-projection error values.

4 Experiments and Results

This section dives into the evaluation of the developed system. After introducing the data used for the evaluation (see Section 4.1) we motivate the choice on the selected metrics (see Section 4.2). Bearing in mind that we do not possess any ground truth annotations, we evaluate the 3D human reconstruction step, based on the data generated by Step 3.4. Additionally, we aim to study the relationship between the quality of the reconstruction and keypoint confidence score. Given the relation between these two variables, before performing a reconstruction, we target the prediction of the quality of the reconstruction, based only on keypoint confidence scores. Furthermore, after introducing the results in Section 4.3, we interpret the issues in the reconstruction.

4.1 Data

The evaluation of the developed system is performed on a total of 19 distinct YOUth videos. For each video, clips of 35 to 30 seconds were selected. We disregard sections of the video in which we observe camera movement. Additionally, we aim to capture different interactions, behavior and movement patterns among the parent and infant. Nonetheless, the reconstruction is performed on a variable number of frames. Due to the nature of Step 3.4.1, we often encounter initial, or final frames, in which a pair of detections is not found. On top of benchmarking our system, our goal with these experiments is to understand what negatively impacts the reconstruction and which scenarios should be avoided. More specifically, we aim to understand the implications of strong spatial occlusions, and close range interactions, on the reconstruction. Fairly short video clips were selected due to the high computation demands of the camera calibration and 3D human reconstruction procedure (Section 3.5).

As further insight, processing a 1050 frame sequence (35 seconds) of a YOUth video requires approximately two hours[11]. The first step of the framework (Video Processor 3.2) takes approximately 3 minutes to split the data into frames. The proceeding step (2D Pose Estimation 3.3) outputs the 2D poses after processing the data for roughly 12 minutes. Once the 2D data reaches the next phase (Keypoint Processor Pipeline 3.4), it requires approximately 2 minutes to be processed. Regarding the pipeline, the most time consuming step is the generation of detection masks for all the selected frames in step View-to-View Consistency 3.4.4. Respectively, phase one and phase two of step Camera Calibration and 3D Human Reconstruction 3.5, take roughly 60 and 40 minutes to complete. The difference in computation time is justified by the camera optimization procedure in phase one, which slows down the overall reconstruction.

4.2 Metrics

In the absence of ground truth 2D or 3D human joint annotations, we perform a qualitative and quantitative validation of the system. To quantify our results, we use *AlphaPose's* 2D keypoint location and confidence score, together with the estimated 3D keypoint location. Each 3D keypoint is projected back to the image plane of each camera. With the re-projected 2D keypoints, and the 2D keypoints estimated by *AlphaPose* (see Figure 16), we measure the **re-projection error**, using Euclidean distance. Under this evaluation scheme, we underline that the performance of *AlphaPose*, when regressing 2D human keypoints, is not optimal. We frequently observe that inaccurate 2D poses are generated during occlusions or close-range interactions. Therefore, we state that re-projection error values do not directly reflect on the quality of the 3D reconstruction, compared to the YOUth video.

By observing that occlusions often lead to low keypoint confidence values, which lead to inaccurate poses, we now aim to study the relationship between 2D keypoint confidence and re-projection error. To study the relationship between the pair of variables, we measure linear correlation using the **Pearson correlation coefficient**. The coefficient ranges from -1 to 1, and reflects the strength and direction of the relationship between the two variables. Essentially, a coefficient value close to 1 means that there is a strong positive correlation between the two variables, thus if one variable's value increases, the other variable's value also tends to increase. On the other hand, a coefficient value close to -1 means that there is a strong negative correlation between the two variables, which means that when a variable's value increases, the other tends to decrease. Finally, if the resulting coefficient value is close to 0, we can conclude that both variables are not related to one another. Once we have an understanding of how keypoint confidence relates to re-projection error, we can estimate the quality of the reconstruction, given the confidence score.

[11] We run the experiment on one MSI GeForce RTX 2080 Ti

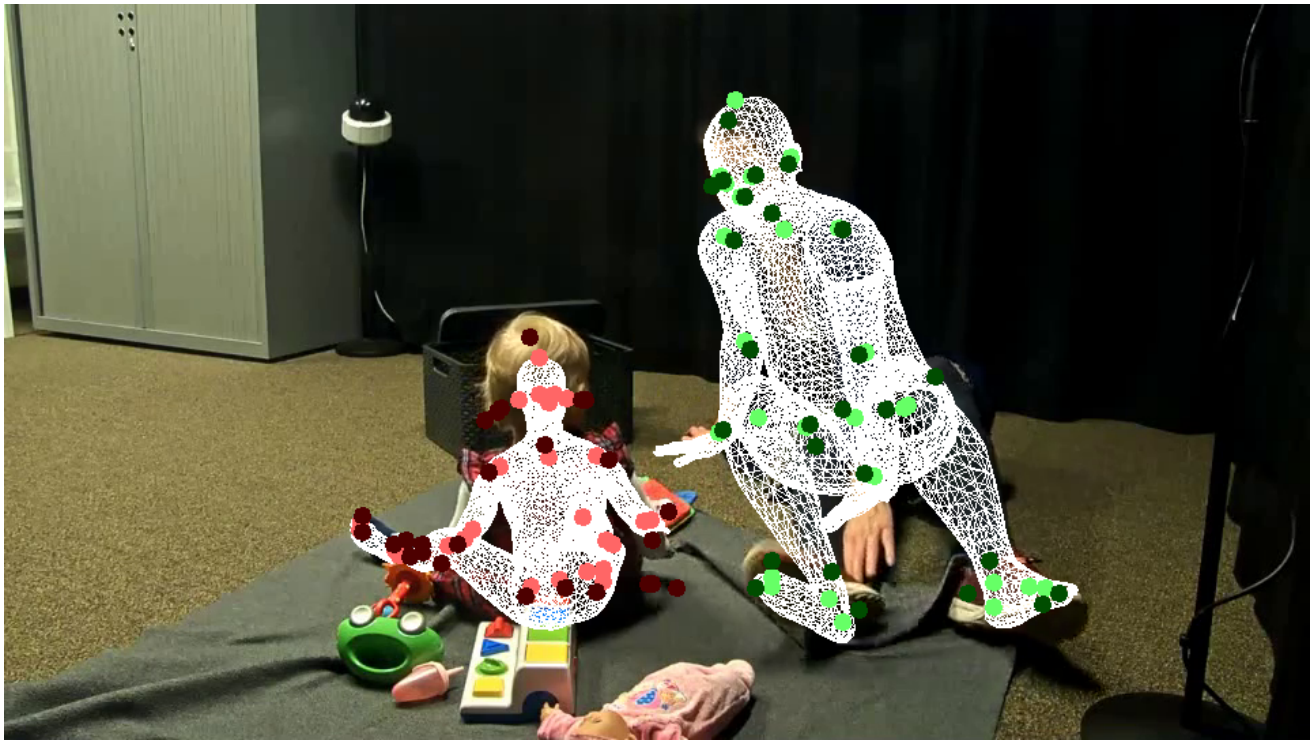

Figure 16: 2D overlap between keypoints predicted by AlphaPose (dark color) and estimated 3D keypoints (light color). In white we outline the vertices of the predicted *SMPL* mesh.

4.3 Results

While processing each YOUth video, we notice that the quality of the 3D reconstruction is not consistent for each video. Given the fact that each video is unique in its own characteristics, we begin by analyzing the overall keypoint confidence value, obtained in Step 4.3.1. Afterwards, we analyze the re-projection error during phase one of Step 3.5. Consequently, we analyze and compare the results obtained from phase two of Step 3.5, with the results obtained from phase one. With this comparison we aim to verify how phase two benefits from the computation of the camera parameters in phase one. Additionally, we hope to motivate, by analyzing the re-projection error, why discarding incomplete information in one view is more beneficial to the reconstruction quality. To conclude our result analysis, we measure how re-projection error relates to 2D keypoint confidence. More specifically, we aim to study how the different aspects in the captured YOUth data relate to defective reconstructions (see Section 4.3.5).

4.3.1 Keypoint Confidence Scores

Table 2 contains the averaged keypoint confidence values for each individual. For the tables which contain the averaged values, over all views, per individual, please reference to the Appendix Section A.2 to find their complete versions. The following are the averaged keypoint confidence output scores returned by *AlphaPose*, during the 2D pose estimation (Section 3.3). By overviewing their overall outcome on the conducted evaluation, we identify which human body joints are frequently occluded in the YOUth data. Similarly, we determine which of the depicted individuals is more prone to reconstruction defects.

By observing the averaged confidence values, we identify that the regression of keypoints for the parent is more challenging than the regression of the keypoints of the infant. To gain further insight of the reasons which lead to such a challenge, we analyze the averaged values per individual keypoints for both individuals. The following histogram plots concern the 2D detection confidence values of the parent (Figure 17) and the child (Figure 18). Despite representing different individuals, common patterns can be observed. We notice that the confidence

YOUth	**2D Keypoint Confidence**		**Error Count**
Video Name	Parent	Infant	ReID + AP
B33718 00:13:40 - 00:14:15	0.712	0.658	290
B33892 00:10:00 - 00:10:35	0.685	0.618	21
B35985 00:10:50 - 00:11:25	0.661	0.649	227
B38777 00:11:17 - 00:11:52	0.719	0.564	80
B40508 00:06:55 - 00:07:30	0.729	0.646	660
B44801 00:02:55 - 00:03:25	0.577	0.615	117
B45111 00:11:26 - 00:12:01	0.668	0.661	72
B45358 00:01:25 - 00:02:00	0.718	0.700	4
B47859 00:07:35 - 00:08:05	0.688	0.685	6
B51848 00:15:35 - 00:16:05	0.705	0.662	199
B64396 00:14:00 - 00:14:35	0.659	0.662	656
B64612 00:01:53 - 00:02:23	0.637	0.687	81
B67411 00:14:35 - 00:15:10	0.749	0.778	8
B70410 00:01:20 - 00:01:55	0.720	0.727	9
B83755 00:10:30 - 00:11:05	0.733	0.678	7
B86218 00:06:00 - 00:06:35	0.717	0.751	12
B89136 00:06:25 - 00:07:00	0.671	0.759	457
B93177 00:14:25 - 00:15:00	0.730	0.726	12
B97605 00:06:45 - 00:07:20	0.745	0.540	7
Average	34.483	42.494	117.474

Table 2: Average 2D pose estimation keypoint confidence score of each individual, given the four views. Error count reflects the number of total *AlphaPose* (AP) and *Torchreid* (ReID) errors, of each video

values concerning the lower body (knees, heels, and toes) are diminished, in relation to the detection confidence of the upper body parts. By observing the YOUth data, we can reason about these results. Very frequently, the parent is captured sitting on their calves, cross legged, crouching, among many other self-occluding poses. On top of being depicted in unconventional sitting poses, we note that the colors of the lower body of the parent are often of a darker tone, which frequently overlaps with the background color, increasing the difficulty for the 2D pose estimator model to accurately regress the body joints. On the contrary, these negative features are not as frequently present during the detection of the upper body joints. On top of that, the data prioritizes the capture of the face of both individuals, over the depiction of the entire body.

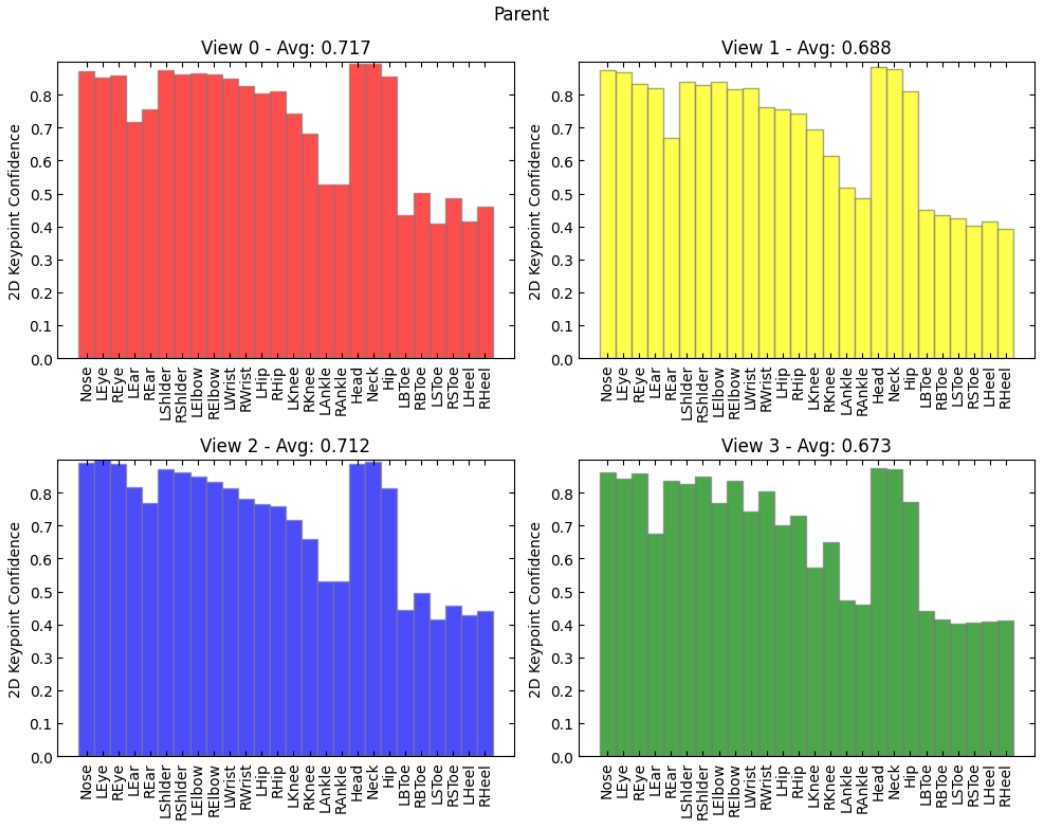

Figure 17: Histogram plot of the parent's individual keypoint confidence scores. The values are the view averages among all 19 videos.

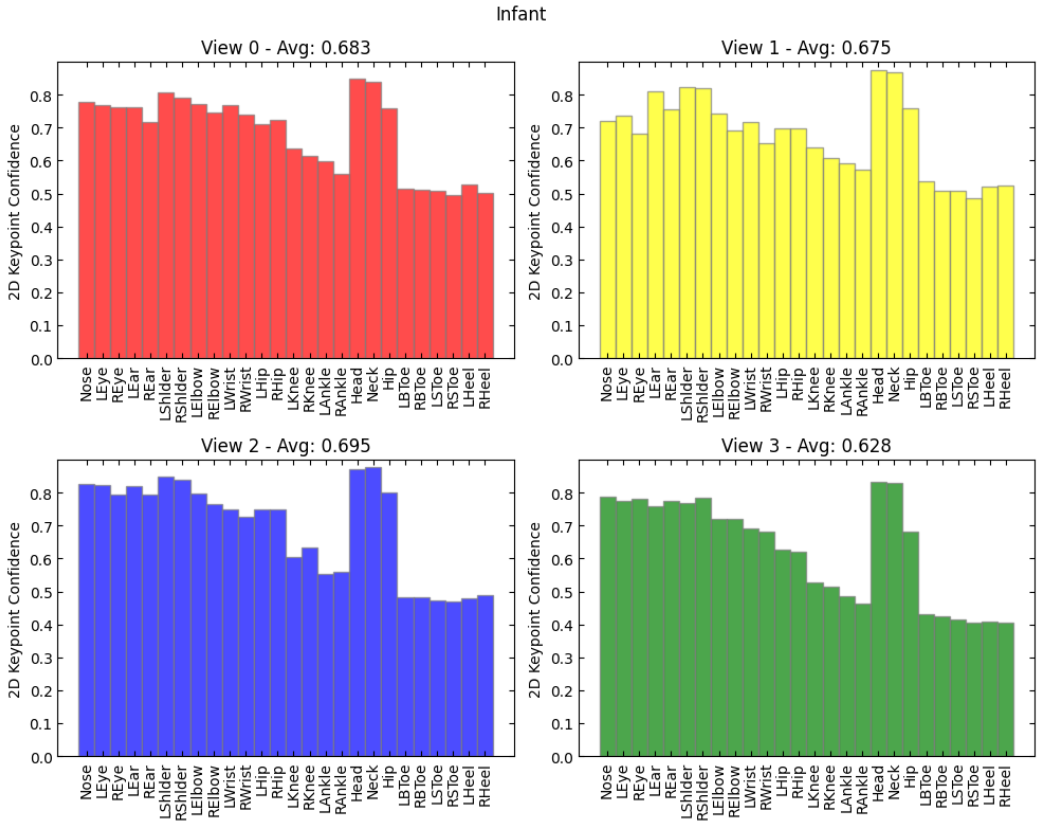

Figure 18: Histogram plot of the infant's individual keypoint confidence scores. The values are the view averages among all 19 videos.

4.3.2 Re-projection Error - Camera Optimization

The previous Section 4.3.1 quantified the performance of the 2D pose estimation method (Section 3.3) on the YOUth data. Consequently, the current section quantifies how phase one of Step 3.5 performs with the *AlphaPose* data, processed by the keypoint processor pipeline (Step 3.4). To conduct the evaluation, we compute the re-projection error of the 3D estimated keypoints in relation to the 2D pose data. The overall results concerning the detections of the parent and infant are listed in Table 9.

The following histogram plots quantify the re-projection error of individual keypoints for the parent's detection (Figure 19), and the infant's detection (Figure 20), at each view. For this scenario, we notice that the different views contain significant deviations of averaged values. In general, view 0 yields the smallest re-projection error. This is reasonable given the inclined top-down view of camera 0, which is less prone to occlusions between both individuals. Contrary to the remaining views, view 0 captures a general overview of the entire room, without being too close to the depicted individuals. Thus, we can conclude that view 0 is the most informative view for the reconstruction. Meanwhile, views 1 and 3 present the highest values in re-projection error. This is due to their close range proximity, and diminished field of view, in regard of the captured individuals. We conclude that, overall, the lower body parts are often occluded in the YOUth data. This phenomena is more problematic for the reconstruction of the parent, than for the reconstruction of the infant, given their natural anatomical differences.

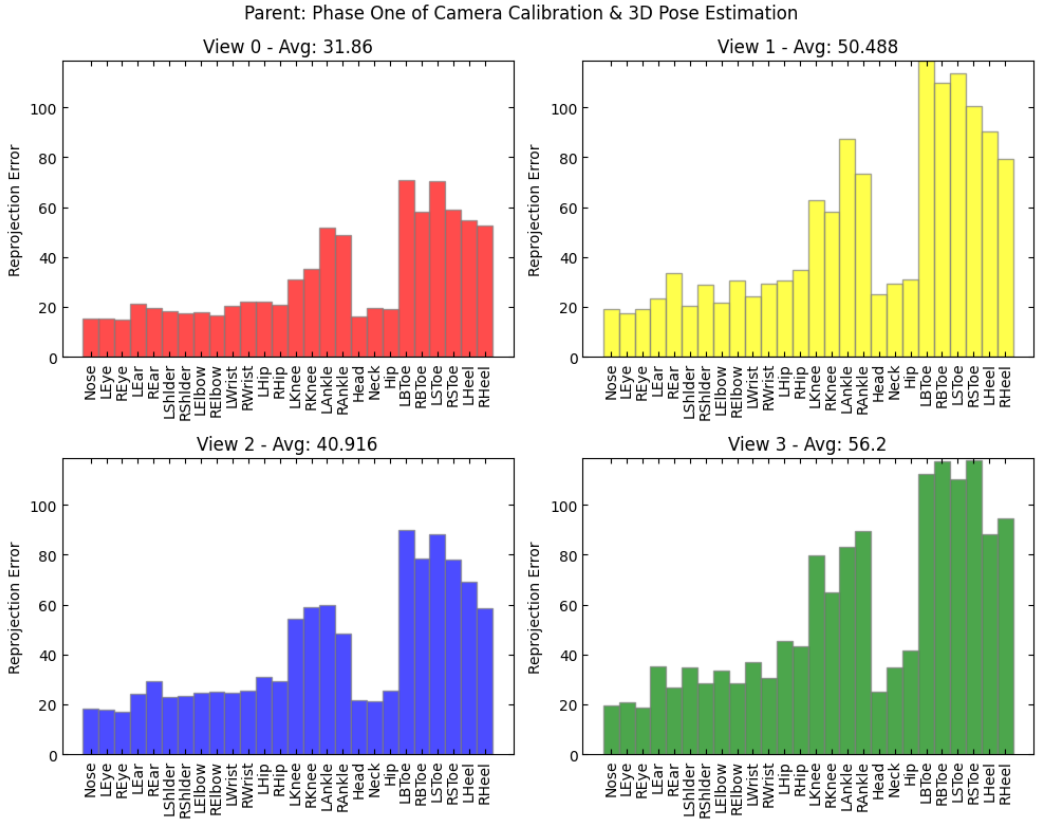

Figure 19: Histogram plot of average parent keypoint re-projection error values, per each view of all 19 videos, during phase one of Step 3.5

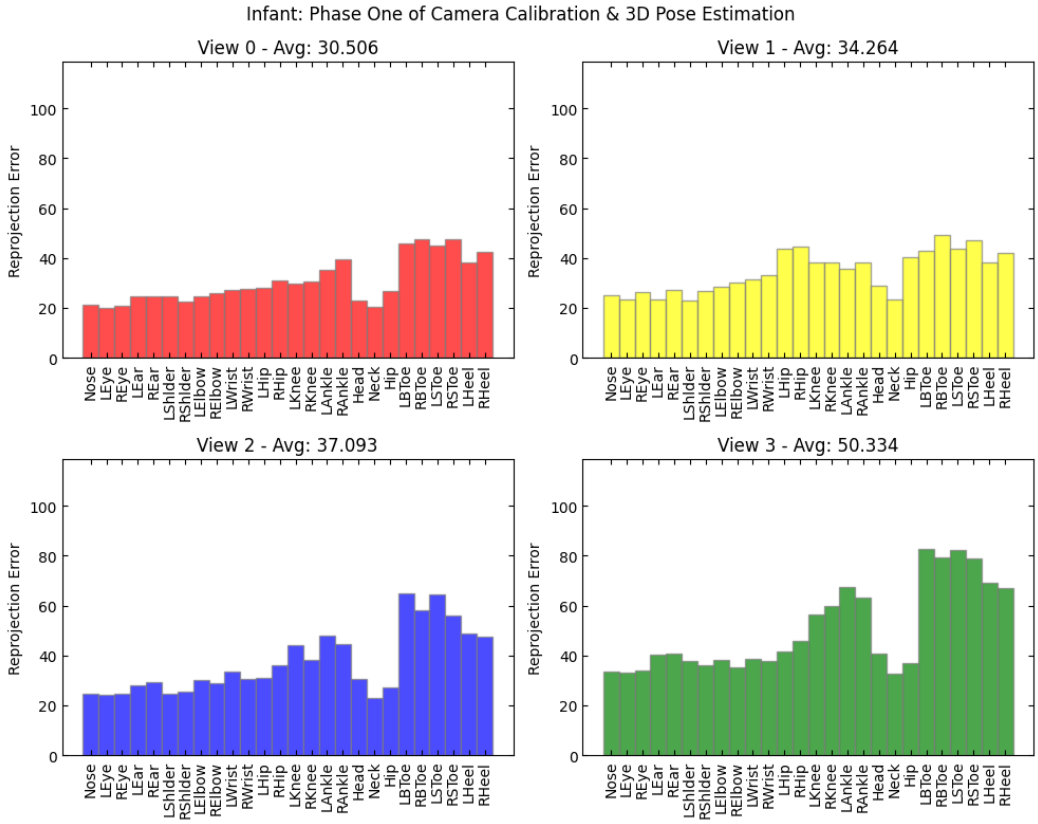

Figure 20: Histogram plot of average infant keypoint re-projection error values, per each view of all 19 videos, during phase one of Step 3.5

4.3.3 Re-projection Error - Fixed Camera Parameters

Table 11 contains the re-projection error values of phase two of Step 3.5. Having the results from step one as a baseline, we calculate the difference in re-projection error between both phases. By evaluating the difference in re-projection error, and by visualizing the 3D human body meshes, we reach the conclusion that the videos which have a high positive difference in re-projection error contain the reconstuction of meshes in unrealistic poses, for an example see Figure 23. On the other hand, accurate poses were achieved for the videos which are represented by a negative difference in re-projection error. This means that the reconstruction was improved in comparison to the first phase. We can also notice that the detections of the parent are more ambiguous than the detections of the child, given the difference values.

Similarly to the analysis of results of phase one, for phase two we compute the histogram plots of the averaged keypoint re-projection errors at each view, for both the parent (Figure 21) and the infant (Figure 22). Overall, we observe that the re-projection errors were not improved. However, this is explained by the fact that in phase two, the cameras cannot be adjusted to account for the defective poses in the faulty detections. This observation indicates that if we feed defective data to phase one of Step 3.5, the cameras will be adjusted to account for the defects in the data. More specifically, after executing phase two on phase's one camera parameters, we are unable to discriminate the videos in which the keypoint processor pipeline failed to correctly index the detections of each individual. To detect such defects, we are required to, beforehand, know the approximated camera parameters. An example of this scenario is given the next Section 5. Nonetheless, the defective data can be clearly observed in the mesh visualization system (Section A.1). By doing so, we state that the keypoint processor pipeline (Section 3.4) failed to correctly index the individuals, for video *B35985*, *B45358* and *B51848*. The erroneous scenarios are described in Section 4.3.6.

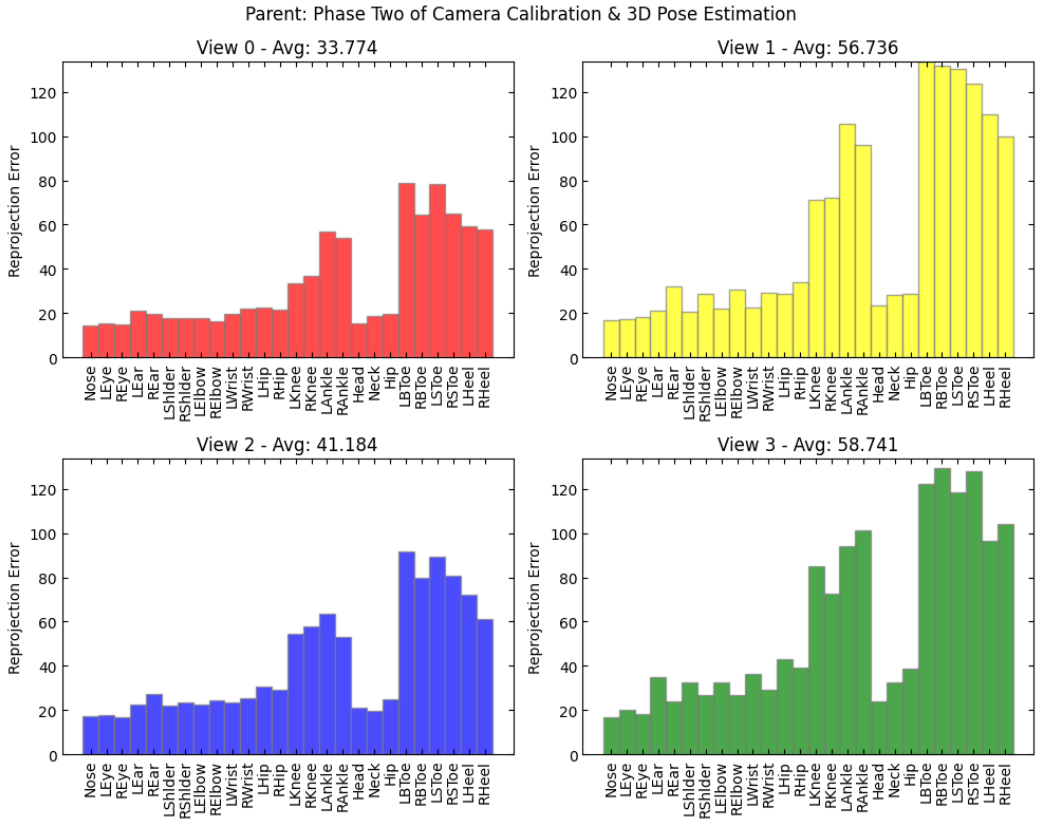

Figure 21: Histogram plot of average parent keypoint re-projection error values, per each view of all 19 videos, during phase two of Step 3.5

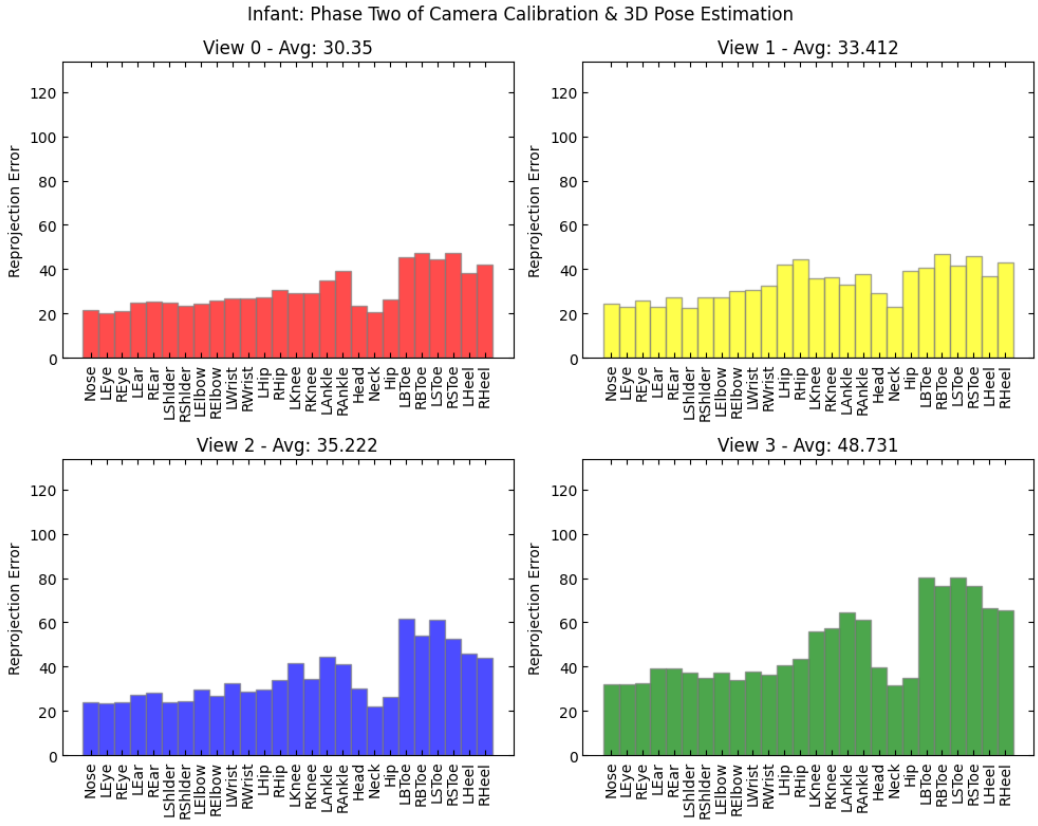

Figure 22: Histogram plot of average infant keypoint re-projection error values, per each view of all 19 videos, during phase two of Step (3.5).

The high ambiguity about the location of one keypoint, among the four views, leads to errors in the reconstruction. More specifically, given the high uncertainty of the location of the lower body keypoints, the resulting body models might suffer from pose instability throughout the 3D visualization, in addition to reconstructing body meshes in unrealistic positions (see Figure 23 for an example). However, these erroneous poses are only reconstructed during a short period of frames, since they suffer from high amounts of jitter throughout the frame sequence. Having insight about the pose information from Step 3.3, we can underline that in the initial frame of video *B45111*, the left leg of the parent was only captured in view 1, while being undetected in the remaining views (see Figure 23). For a visualization of the difference in re-projection error between both phases, for video *B45111*, see Figure 24.

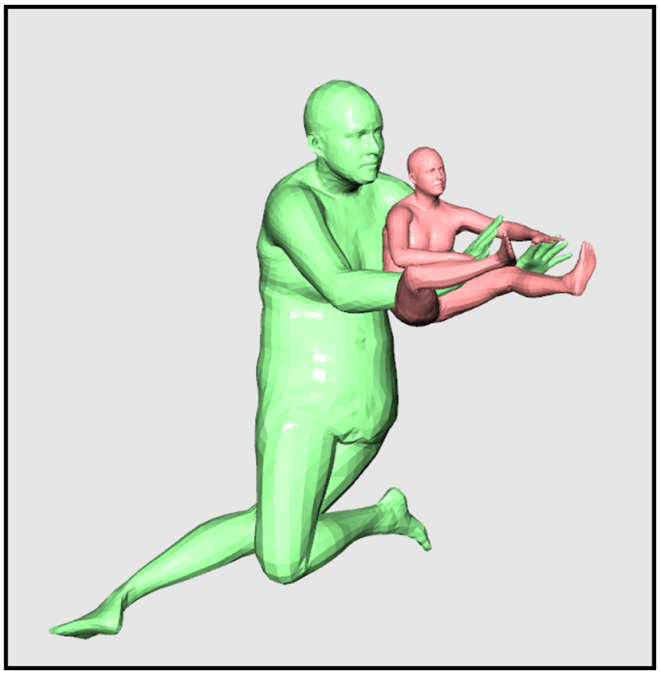

Figure 23: Depiction of the reconstruction of the first frame of video *B45111*. The captured scenario depicts the mesh results during high ambiguity in estimating the 3D position of the parent's legs. The green colored mesh represents the parent, and the red mesh represents the infant.

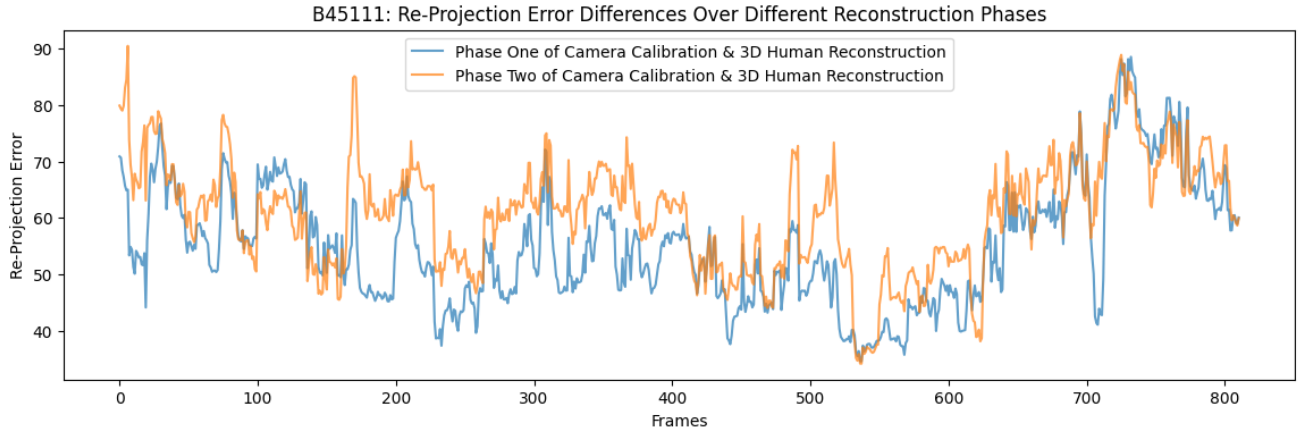

Figure 24: Plot of re-projection errors, throughout the frame sequence, between phase one and phase two of Camera Calibration and 3D Human Reconstruction 3.5, for video *B45111*.

4.3.4 Re-projection Error with Variable Number of Views

We now target our attention on the view discarding feature, described at Step 3.5. By ignoring the views with insufficient detections, during specific frame ranges, we aim to quantify and compare the resulting re-projection error. Utilizing the same 19 videos, we now enable the mechanism to capture frame ranges in which one of the individuals was out of detection for more than 3 seconds (90 frames). From our data, 3 videos were impacted. The updated re-projection values for the selected videos can be seen in Table 3. Regarding the discarded frame sequences, view 3 of video *B33892* was discarded two times, between frames $[258-456]$ and $[764-938]$. This was caused by the missing detections of the infant (see Figure 38 and Figure 39). View number 1 of video *B40508* was ignored during frames $[648-835]$, due to the missing detections of the infant (see Figure 40 and Figure 41). Finally, concerning the missing detections of the infant in video *B64396*, view 3 was ignored during frames $[157-319]$, view 0 during frames $[580-745]$ and, once again, view 3 was discarded during frames $[746-847]$, having its original missing frame range padded in order to account for the overlapping missing detections in view 0 (see Figure 42 and Figure 44). From this study, we conclude that temporal occlusions affect more the infant than the parent. This is to be expected given the fact that the larger body proportions of the parent often occlude the smaller body of the infant.

Updated Re-Projection Error - P2			**Difference**	
Video Name	Parent	Infant	Parent	Infant
B33892	25.179	18.102	-2.072	-2.869
B40508	43.137	40.578	-2.095	-4.558
B64396	54.995	56.275	3.511	2.945

Table 3: New re-projection error values for videos in which one individual was undetected for more than 3 seconds.

From the average re-projection errors listed above (Table 3), we verify that discarding one view from the reconstruction is often beneficial to the quantitative results. However, as we see with the reconstruction outcome of video *B64396*, performing the reconstruction with only three views during 267 frames (8.9 seconds) negatively impacts the re-projection error. The following histogram plots (Figure 25 and Figure 26) quantify the updated re-projection error averages, over all the previously computed results. For each view, we notice that the re-projection error did not suffer significant changes.

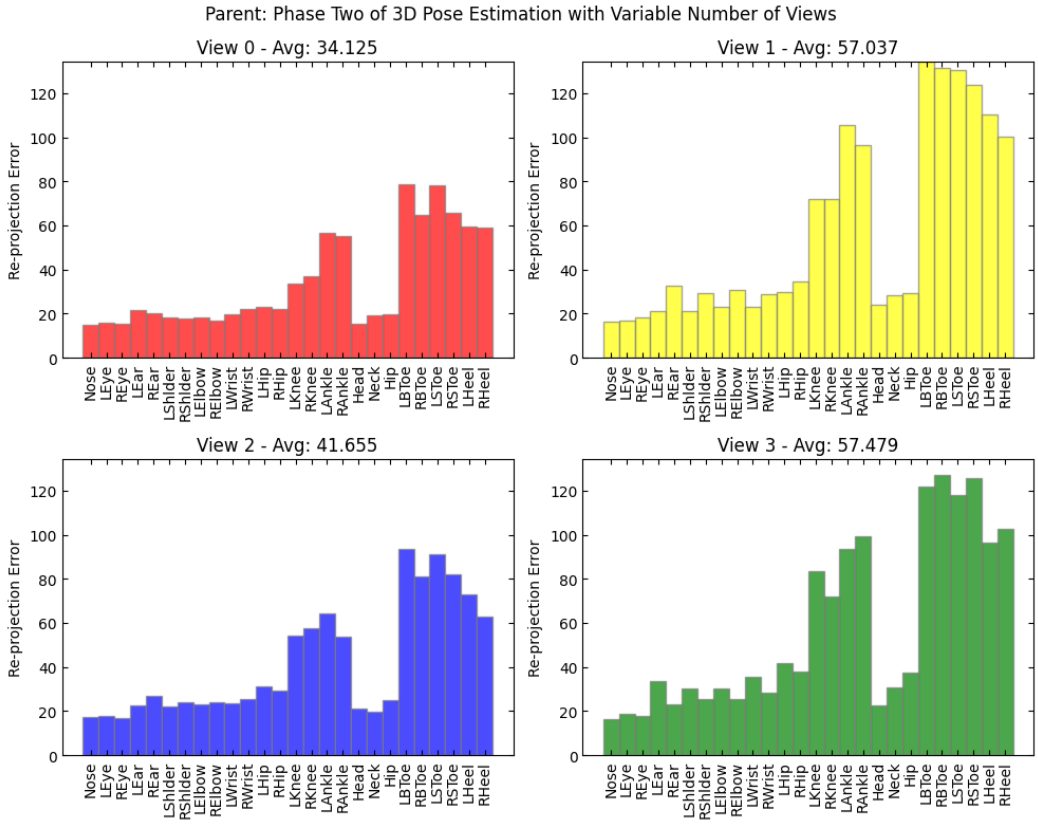

Figure 25: Histogram plot of the updated average keypoint re-projection error values, across all the 19 videos, for the parent detection.

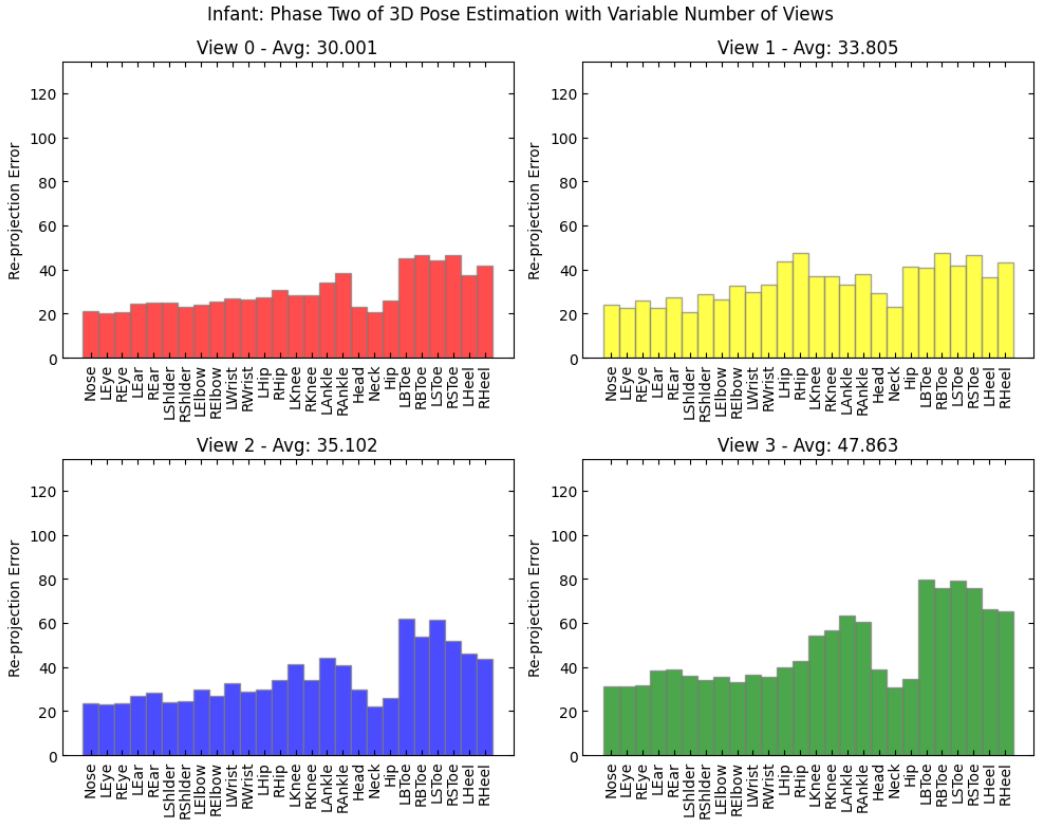

Figure 26: Histogram plot the updated average keypoint re-projection error values, across all the 19 videos, for the infant detection.

4.3.5 Keypoint Confidence and Re-Projection Error Analysis

To underline what we have previously observed, we calculate the linear Pearson correlation coefficient between re-projection error and 2D keypoint confidence values. Based on the plots below (Figure 27 and Figure 30) we confirm that uncertain keypoint detections lead to an increase in re-projection error, since both variables show strong negative correlation. These results relate to the fact that the 3D Human Reconstruction step prioritizes the location of keypoints which have a higher confidence value.

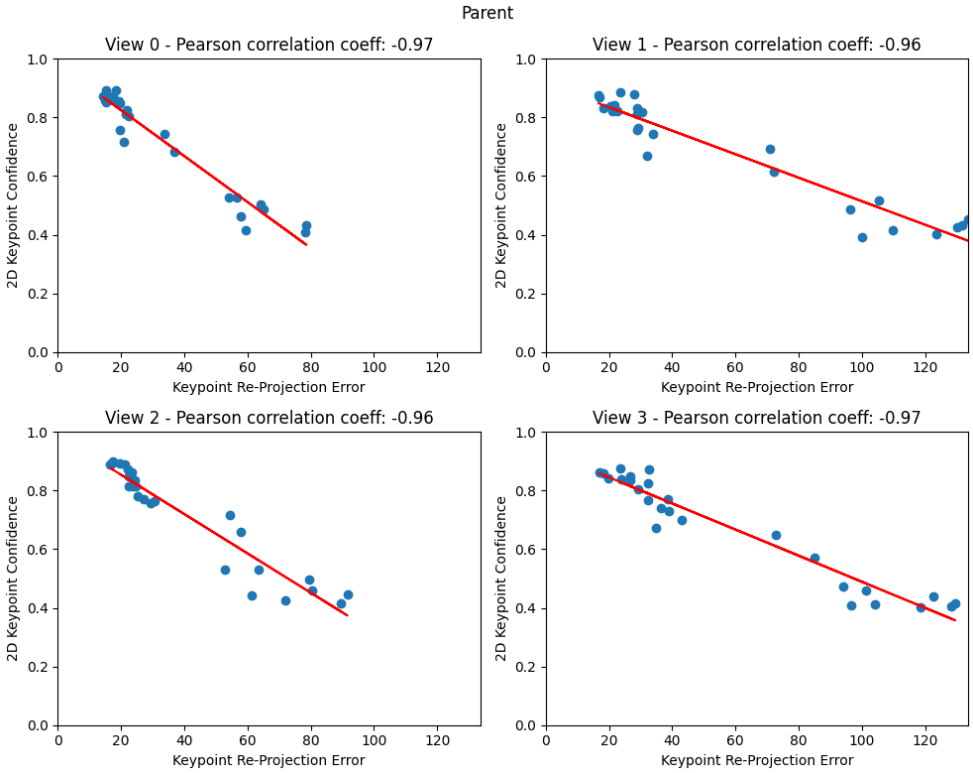

Figure 27: Linear correlation coefficient value between confidence score and re-projection error, of keypoints which belong to the parent detection among all 19 videos.

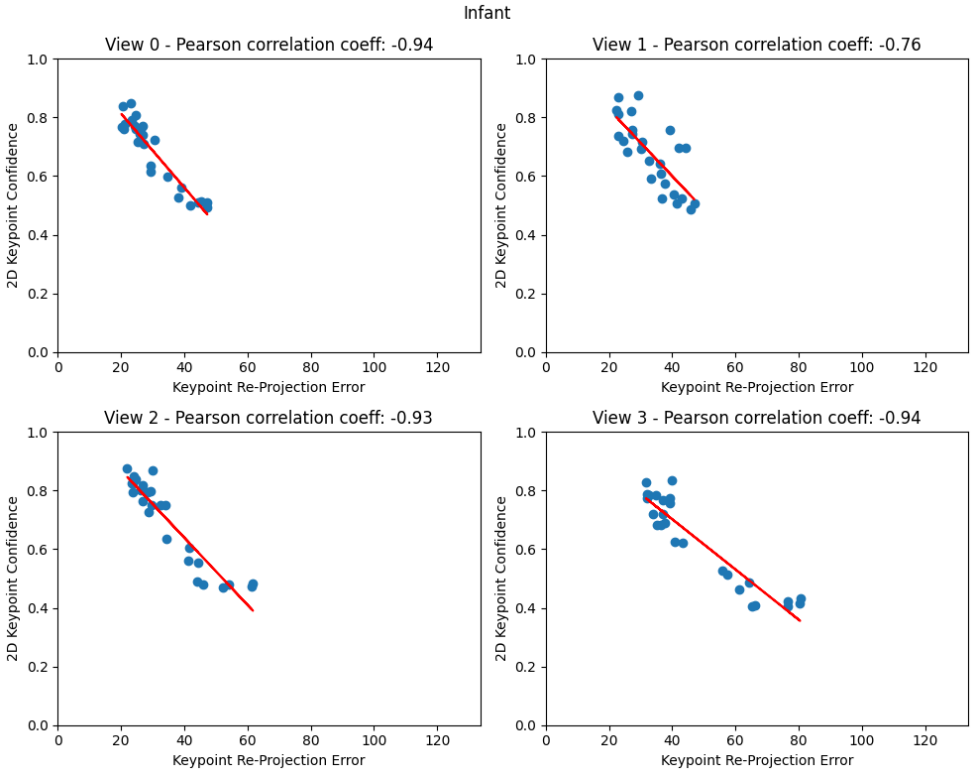

Figure 28: Linear correlation coefficient value between confidence score and re-projection error, of keypoints which belong to the infant detection among all 19 videos.

4.3.6 Pipeline Error Analysis

After performing the quantitative evaluation, we now aim to bring the qualitative errors to surface. These correspond to the calculations performed prior to step Camera Calibration and 3D Human Reconstruction 3.5. Having previously identified the videos with qualitative errors (*B35985*, *B45358* and *B51848*), we now aim to understand the video characteristics which introduced unsolvable nuances for the keypoint processor pipeline (Section 3.4).

B35985

We begin by noting that video *B35985* has a different camera layout, compared to the remaining videos. Additionally, view 0, the source view for the color transfer algorithm, is depicted in a yellowish color pallet. This is undesirable since we are normalizing the colors of the remaining views to the least informative video coloring. However, we detect that the error originates from both an *AlphaPose* and *ReID* error. Halfway of the frame sequence, in view 0 we notice that the horizontally extended legs of the parent are semi occluded, and the torso of the child is depicted in close range to the parent. This leads the human detector model to identify only the torso of the parent and the torso of the infant with the legs of the parent. The problematic pose is thereafter associated with both the parent and the infant, which introduces ambiguities in discriminating to whom the pose belongs. For the following frames, the keypoint processor pipeline mistakenly switched the detections of the parent with the detections of the infant. For the remaining views, the pipeline correctly indexed the resulting detections.

B45358

For video *B45358*, the view-to-view consistency algorithm (Section 3.4.4) failed to correctly identify which detections belonged to the parent and which detections belonged to the infant, in regard of view 2. The inaccurate decision originated from the segmentation of the infant's upper body, in the parent's detection of view 2. In other

words, the segmentation mask which concerned the detection of the parent, contained both individuals. This overlap of depictions caused the histogram comparing mechanism to identify the faulty detection as a detection of the infant and not the parent (for the representative disambiguation values see Figure 15). Nonetheless, the detections were correctly indexed, given the frame-to-frame consistency mechanism (Section 3.4.2).

Figure 29: Illustrative example of the impact of incorrect indexing on the reconstruction of video *B45358*.

B51848

For video *B51848* we notice that, once again, the fusion of errors of *AlphaPose* and *ReID* caused the detections of the individuals to be interchanged during the frame-to-frame consistency procedure (Section 3.4.2). After observing these results, we underline that close-range interactions, during spatial occlusions, lead to identification errors between both individuals, which lead to reconstruction flaws.

5 Ablation Study

This section aims to quantify how different parameters, and camera optimization strategies, interfere in the overall results of the 3D reconstruction. Two additional experiments are conducted in order to gain further insight about the variation in re-projection error, when the detections in one view are exchanged or completely discarded from the reconstruction. We conduct a total of 6 different ablation experiments on the same YOUth demonstration video. As baseline, we perform the original two phase scheme of step Camera Calibration and 3D Human Reconstruction 3.5, with the parent's scale fixed at 100% and the child's scale at 45%. The results of the baseline experiment concern the second phase of the reconstruction and serve as benchmark for the remaining experiments.

Demo 00:03:45 - 00:04:20	**Parent Re-Proj. Error**				**Infant Re-Proj. Error**			
Experiment	V0	V1	V2	V3	V0	V1	V2	V3
Baseline	30.758	22.920	47.063	40.979	13.822	30.177	23.200	20.599

Table 4: Baseline experiment re-projection error values, per each view, for each individual

5.1 Child Scale

The following experiments demonstrate the influence of the child's scale parameter. For the first experiment, we perform the second phase of Step 3.5 with the infant's scale defined as the same scale as the parent's, 100%. The second experiment follows the same principles of the first, but with the child's scale defined at 75%.

Demo 00:03:45 - 00:04:20	**Parent Re-Proj. Error**				**Infant Re-Proj. Error**			
Child Body Scale	V0	V1	V2	V3	V0	V1	V2	V3
100%	30.793	23.118	46.991	41.336	16.970	41.510	27.136	28.966
75%	30.736	22.878	47.067	40.997	14.597	32.609	24.104	24.311

Table 5: Ablation study re-projection error results on the demonstration video.

Based on the values in Table 5 and the benchmark results, we observe that the baseline scale of the infant yields superior quantitative results, in comparison to the two experiments. Figure 30 illustrates the qualitative impact on the reconstruction of the child, given the different scale values. We underline that high scale values for the body mesh of the child yield unrealistic, thus, undesirable results.

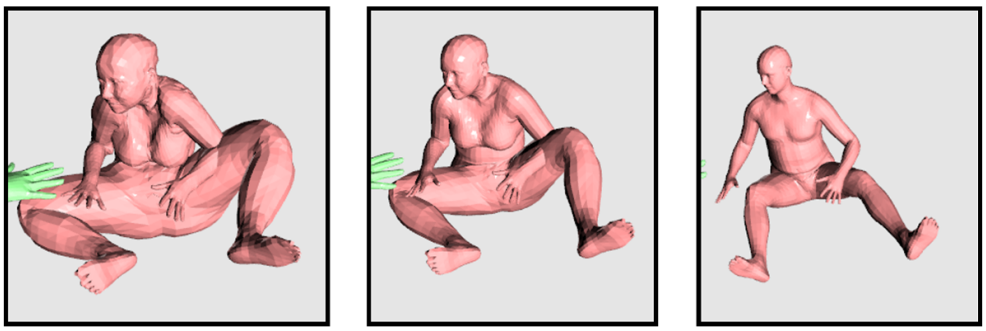

Figure 30: Qualitative illustration of the different scales used to reconstruct the body mesh of the infant.

5.2 Person-Specific Camera Calibration

For the following studies, we perform the camera calibration phase on individual agent's 2D keypoints. The fist experiment optimizes the camera parameters on the keypoints of the parent and the second on the keypoints of the infant.

Demo 00:03:45 - 00:04:20	**Parent Re-Proj. Error**				**Infant Re-Proj. Error**			
Camera Calibration	V0	V1	V2	V3	V0	V1	V2	V3
Parent	30.355	22.360	46.440	40.566	17.644	38.231	29.606	22.579
Infant	47.175	25.520	56.467	43.876	12.512	29.222	22.388	18.940

Table 6: Ablation study re-projection error results on the demonstration video.

Given the re-projection error values of both experiments, we note that, during phase two of Step 3.5, the reconstruction is optimal to the individual with which we optimized the cameras on. However, the second individual, absent during the camera calibration phase, is reconstructed with higher uncertainty, suffering from increased amounts of jitter during the reconstruction visualization. These studies confirm that, in order to retrieve the most optimal reconstruction results, we are required of using both individual's keypoints during the camera optimization phase.

5.3 Exchange and Removal of Detections

The final pair of experiments aim to quantify the change in re-projection error during wrong keypoint indexation and during the reconstruction with only three cameras. We perform both of the experiments with the same camera parameters as we performed the baseline study. The first experiment exchanges the keypoints of the individuals in view 2 and the second experiment discards all the information from view 2.

Demo 00:03:45 - 00:04:20	**Parent Re-Proj. Error**				**Infant Re-Proj. Error**			
View 2 Keypoints	V0	V1	V2	V3	V0	V1	V2	V3
Exchanged	40.033	34.262	194.555	46.222	34.240	54.301	160.980	25.570
Discarded	31.923	19.995	–	41.664	12.225	27.935	–	20.086

Table 7: Ablation study re-projection error results on the demonstration video.

Based on the re-projection errors listed in Table 7, we confirm that wrongly indexed views are only quantifiable, by their re-projection error, when we have a correct estimation of the camera parameters. Meanwhile, for the second experiment, the re-projection errors of the present view, don't suffer significant changes when compared to the baseline experiment. However, by discarding view 2 from the reconstruction, we notice an increased amount of pose uncertainty during the mesh visualization. The most impacted body parts are the parent's legs. This is to be expected given the established observations in Section 4.3.3. By losing all the information coming from one view, the uncertainty in predicting the 3D position of low confidence 2D keypoints is increased.

6 Conclusion and Future Work

This research project had the ultimate goal of enabling the 3D human reconstruction on the non-annotated YOUth data. In order to accomplish this milestone, we steered our research according to the most optimal methods and strategies, which aligned with the criteria of our data, developing features which filled the gaps in the field of *in-the-wild* multi-view 3D human reconstruction. To understand how the developed methodology helped us to achieve our main goal, we first have to seek the answer for our sub-questions:

Can we establish an efficient and reliable feature correspondence mechanism to track individuals among sequential frames and different views?

Based on the overall qualitative and quantitative results, it can be concluded that the developed framework often overcame the natural ambiguities in the 2D pose information, yielding accurate 3D representations of the depicted individuals. After the validation of the qualitative results, from the 19 processed videos, we have observed that only 3 videos caused the pipeline to fail during the 2D pose disambiguation process. As was described in Section 4.3.6, we identified that the origin of the wrong indexation derived from *AlphaPose*, *Torchreid* and *Detectron2* ambiguities. Despite being unable to solve for these prior model errors, we conclude that the developed mechanism reliably corresponded the features among different frames and different views.

During prolonged temporal occlusions, will the reconstruction be improved if we discard the deficient view?

To answer this sub-question, we first have to understand the YOUth data and how accurately it is represented by the reconstruction. As we have observed, the accuracy of the 2D pose estimation model was highly influenced by the quality of the captured video. In other words, ambiguous depictions and frequent occlusions led to an overall decrease in keypoint regression accuracy. As a consequence, discarding one deficient view from the reconstruction is only beneficial if the remaining views accurately capture the position of all human keypoints. Unfortunately, this is not the case with the YOUth data, as occlusions and color ambiguities tend to be present in more than one view. This is to be expected during depictions of close-range interactions and when clothing of a similar color tone as the background is being worn. Therefore, we conclude that to achieve optimal reconstruction quality, we are required of utilizing all the information available.

Given the answers to our sub-questions, now we seek to reply to our main question:

How accurately can we perform a multi-view 3D human reconstruction, between two close range interacting agents, without ground-truth annotations?

Despite the inability to quantify the performance of our framework, we have established that ambiguous depictions lead to ambiguous reconstructions. Therefore, we conclude that the accuracy of the reconstructed individuals is proportionally impacted by how accurately all the body features of both individuals are captured in the data. Nonetheless, the biggest limitation of our research was the reconstruction of an infant body, with an adult sized model. Although we were able to realistically approximate the scale of the adult model to the body size of the infant, we failed to capture its true body proportions.

In it's current state, the pipeline discards the information from all views if at least two detections aren't unanimously present in the initial or final frame. This can have negative implications on the analysis of the data since the initial or final sequence of frames of the selected YOUth video is discarded from the reconstruction. Therefore, one could perform modifications to pipeline's Step 3.4.1 in order to overcome this loss of data. One could allow the pipeline to discard only the information from the deficient view, and maintain the remaining three, if all contain at least two detections. Eventually, when the remaining view matches the detections of the remaining views, one can start the tracking of all the data coming from the previously deficient view. This range requires to be flagged and transmitted to Step 3.5, where the data population process identifies the range, and populates the data accordingly.

As was previously mentioned, we limit the reconstruction for YOUth videos which do not contain any camera movement (Section 4.1). This limitation not only discards valuable information from the reconstruction, but also requires the user to account for any camera changes when selecting a video to process. Therefore, we suggest the integration of a new mechanism between Step 3.2 and Step 3.3, in order to account for camera pan and zoom change. This can be identified with optical flow based techniques, as described in the works of Makkapati *et*

al. [81]. The goal of the additional step is to encompass the data in movement, by identifying it's initial and final frame. Once captured, these ranges should be saved and transmitted to Step 3.5, for the data population procedure. These ranges should be flagged not to discard the view, but to allow for camera optimization.

Despite the improvements that can still be implemented to the framework, we hope to have opened a gateway for further research of the YOUth data. By observing the *SMPL* parameters one can gain insight about each individual's location, pose, shape and orientation. With further processing of these values, one can estimate, for example, how frequently the parent touches the infant. With the current mesh visualization system, we are limited to the observation of the interactions between both individuals. For further analysis of the data, we underline that the reconstruction should not be limited to the depicted individuals. To better understand behavior patterns, one can model, in 3D, each toy in the fixed toy set. One possible solution to accomplish this would be to model, beforehand, a mesh of each toy. Thereafter, in parallel with the 2D human pose estimation step, one could deploy separate toy identification models to identify the position of each toy in the scene. After the identification, and during the triangulation procedure, one could develop a mechanism to distinguish human keypoints from the feature coordinates of the toy, eventually calculating its location in the 3D scene.

References

[1] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, and W. Yang, "Exploiting temporal contexts with strided transformer for 3d human pose estimation," 2022.

[2] W. Shan, Z. Liu, X. Zhang, S. Wang, S. Ma, and W. Gao, "P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation," 2022. [Online]. Available: https://arxiv.org/abs/2203.07628

[3] S. Chun, S. Park, and J. Y. Chang, "Learnable human mesh triangulation for 3d human pose and shape estimation," 2022. [Online]. Available: https://arxiv.org/abs/2208.11251

[4] Y. Cheng, B. Wang, B. Yang, and R. T. Tan, "Monocular 3d multi-person pose estimation by integrating top-down and bottom-up networks," 2021. [Online]. Available: https://arxiv.org/abs/2104.01797

[5] Z. Zhou, Q. Shuai, Y. Wang, Q. Fang, X. Ji, F. Li, H. Bao, and X. Zhou, "Quickpose: Real-time multi-view multi-person pose estimation in crowded scenes," New York, NY, USA, 2022. [Online]. Available: https://doi.org/10.1145/3528233.3530746

[6] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou, "Fast and robust multi-person 3d pose estimation from multiple views," 2019.

[7] U. Iqbal, P. Molchanov, and J. Kautz, "Weakly-supervised 3d human pose learning via multi-view images in the wild," 2020. [Online]. Available: https://arxiv.org/abs/2003.07581

[8] B. Huang, Y. Shu, T. Zhang, and Y. Wang, "Dynamic multi-person mesh recovery from uncalibrated multi-view cameras," 2021.

[9] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," pp. 1325–1339, 2014.

[10] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," IEEE, 2017.

[11] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," IEEE, sep 2018. [Online]. Available: http://gvv.mpi-inf.mpg.de/projects/SingleShotMultiPerson

[12] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," 2018.

[13] M. Black, "Humaneva dataset," 2010. [Online]. Available: http://humaneva.is.tue.mpg.de/

[14] B. Usman, A. Tagliasacchi, K. Saenko, and A. Sud, "Metapose: Fast 3d pose from multiple views without 3d supervision," 2021. [Online]. Available: https://arxiv.org/abs/2108.04869

[15] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," 2017. [Online]. Available: https://arxiv.org/abs/1712.06584

[16] Z. Li, J. Liu, Z. Zhang, S. Xu, and Y. Yan, "Cliff: Carrying location information in full frames into human pose and shape estimation," 2022. [Online]. Available: https://arxiv.org/abs/2208.00571

[17] M. Kocabas, C.-H. P. Huang, J. Tesch, L. Müller, O. Hilliges, and M. J. Black, "Spec: Seeing people in the wild with an estimated camera," 2021. [Online]. Available: https://arxiv.org/abs/2110.00620

[18] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black, "Pare: Part attention regressor for 3d human body estimation," 2021. [Online]. Available: https://arxiv.org/abs/2104.08527

[19] N. Ugrinovic, A. Ruiz, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Body size and depth disambiguation in multi-person reconstruction from single images," Dec 2021. [Online]. Available: https://arxiv.org/abs/2111.01884

[20] 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," pp. 2148–2157, 2018.

[21] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," 2018. [Online]. Available: https://arxiv.org/abs/1811.11742

[22] W. Shan, Z. Liu, X. Zhang, S. Wang, S. Ma, and W. Gao, "P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation," 2022.

[23] 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," 2016. [Online]. Available: https://arxiv.org/abs/1607.08128

[24] 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," 2020. [Online]. Available: https://arxiv.org/abs/2011.14672

[25] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," 2018. [Online]. Available: https://arxiv.org/abs/1803.09722

[26] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," 2019. [Online]. Available: https://arxiv.org/abs/1905.05754

[27] G. Moon, J. Y. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," 2019. [Online]. Available: https://arxiv.org/abs/1907.11346

[28] R. A. Güler and I. Kokkinos, "Holopose: Holistic 3d human reconstruction in-the-wild," pp. 10 876–10 886, 2019.

[29] Z. Li, M. Oskarsson, and A. Heyden, "3d human pose and shape estimation through collaborative learning and multi-view model-fitting," pp. 1887–1896, 2021.

[30] A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. Sminchisescu, "Deep network for the integrated 3d sensing of multiple people in natural images," 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/6a6610feab86a1f294dbbf5855c74af9-Paper.pdf

[31] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," jun 2014.

[32] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015. [Online]. Available: https://arxiv.org/abs/1511.08458

[33] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016. [Online]. Available: https://arxiv.org/abs/1603.06937

[34] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," 2016.

[35] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," 2018. [Online]. Available: https://arxiv.org/abs/1812.08008

[36] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," 2015. [Online]. Available: https://arxiv.org/abs/1511.06645

[37] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," 2022. [Online]. Available: https://arxiv.org/abs/2211.03375

[38] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," 2017. [Online]. Available: https://arxiv.org/abs/1711.08229

[39] J. Li, S. Bian, A. Zeng, C. Wang, B. Pang, W. Liu, and C. Lu, "Human pose regression with residual log-likelihood estimation," 2021. [Online]. Available: https://arxiv.org/abs/2107.11291

[40] S. Yang, Z. Quan, M. Nie, and W. Yang, "Transpose: Keypoint localization via transformer," 2020. [Online]. Available: https://arxiv.org/abs/2012.14214

[41] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang, "Hrformer: High-resolution transformer for dense prediction," 2021. [Online]. Available: https://arxiv.org/abs/2110.09408

[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[43] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," 2017. [Online]. Available: https://arxiv.org/abs/1711.07319

[44] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," September 2018.

[45] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," pp. 2980–2988, 2017.

[46] V. Meel, "Yolov3: Real-time object detection algorithm (guide)," Aug 2022. [Online]. Available: https://viso.ai/deep-learning/yolov3-overview/

[47] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," 2019. [Online]. Available: https://arxiv.org/abs/1911.09070

[48] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," 2018. [Online]. Available: https://arxiv.org/abs/1807.04067

[49] R. A. Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," 2018. [Online]. Available: https://arxiv.org/abs/1802.00434

[50] G. Hua, M.-H. Yang, and Y. Wu, "Learning to estimate human pose with data driven belief propagation," pp. 747–754 vol. 2, 2005.

[51] W. Krauth, "Introduction to monte carlo algorithms," 1996. [Online]. Available: https://arxiv.org/abs/cond-mat/9612186

[52] P. Kuo and D. Makris, "Integration of bottom-up/top-down approaches for 2d pose estimation using probabilistic gaussian modelling," pp. 242–255, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314210001864

[53] S. Wang, H. Ai, T. Yamashita, and S. Lao, "Combined top-down/bottom-up human articulated pose estimation using adaboost learning," pp. 3670–3673, 2010.

[54] P. Hu and D. Ramanan, "Bottom-up and top-down reasoning with hierarchical rectified gaussians," pp. 5600–5609, 2016.

[55] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," pp. 2272–2281, 2019.

[56] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," 2014.

[57] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen, "Synthesizing training images for boosting human 3d pose estimation," 2016. [Online]. Available: https://arxiv.org/abs/1604.02703

[58] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," 2016. [Online]. Available: https://arxiv.org/abs/1611.07828

[59] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua, "Learning monocular 3d human pose estimation from multi-view images," 2018. [Online]. Available: https://arxiv.org/abs/1803.04775

[60] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image," jul 2017.

[61] W. Jiang, "Coherent reconstruction of multiple humans from a single image," 2020. [Online]. Available: https://jiangwenpl.github.io/multiperson/

[62] R. Szeliski, *Computer vision: Algorithms and applications*. Springer Nature Switzerland, 2022.

[63] Hartley, "An algorithm for self calibration from several views," pp. 908–912, 1994.

[64] D. Lowe, 1999. [Online]. Available: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[65] A. M. Truong, W. Philips, N. Deligiannis, L. Abrahamyan, and J. Guan, "Automatic multi-camera extrinsic parameter calibration based on pedestrian torsors," 2019.

[66] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," October 2019.

[67] H. Hristov, "The direct linear transform," Nov 2022. [Online]. Available: https://www.baeldung.com/cs/direct-linear-transform

[68] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013. [Online]. Available: https://arxiv.org/abs/1312.6114

[69] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014. [Online]. Available: https://arxiv.org/abs/1406.1078

[70] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," pp. 248:1–248:16, Oct. 2015.

[71] J. Williams. [Online]. Available: https://smpl.is.tue.mpg.de/index.html

[72] N. Hesse, S. Pujades, J. Romero, and M. Black, "Skinned multi-infant linear body model," 2021.

[73] P. Mahol, "Ffmpeg documentation," May 2023. [Online]. Available: https://ffmpeg.org/ffmpeg.html

[74] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[76] K. Zhou and T. Xiang, "Torchreid: A library for deep learning person re-identification in pytorch," 2019.

[77] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," pp. 34–41, 10 2001.

[78] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," 2019.

[79] E. Graber, "Growth charts - who child growth standards," Sep 2010.

[80] X. Li, Z. Fan, Y. Liu, Y. Li, and Q. Dai, "3d pose detection of closely interactive humans using multi-view cameras," p. 2831, 06 2019.

[81] V. V. Makkapati, "Robust camera pan and zoom change detection using optical flow," 2007.

A Appendix

A.1 Mesh Visualization

With the intent of performing a 3D reconstruction of the scene depicted in the input video, we created a mesh visualization program. After executing the previous step 3.5, the resulting meshes can be sequentially visualized, together with the predicted cameras. Once in execution, a set of keyboard keys can be pressed in order to interact with the reconstruction. Given the four cameras, one can set the scene view to match the perspective of any of the used cameras for the reconstruction. By pressing the keyboard key *1*, *2*, *3* or *4* we respectively set the view to match the perspective of camera *0* (red), *1* (yellow), *2* (blue) or *3* (green). By pressing the keyboard key *Space* the visualized meshes will be updated along the frame sequence. Additionally, the key *a* increments one frame and key *b* decrements one frame, updating the visualization. See Figure 31 for an illustrative example of the layout of the cameras in the reconstruction scene.

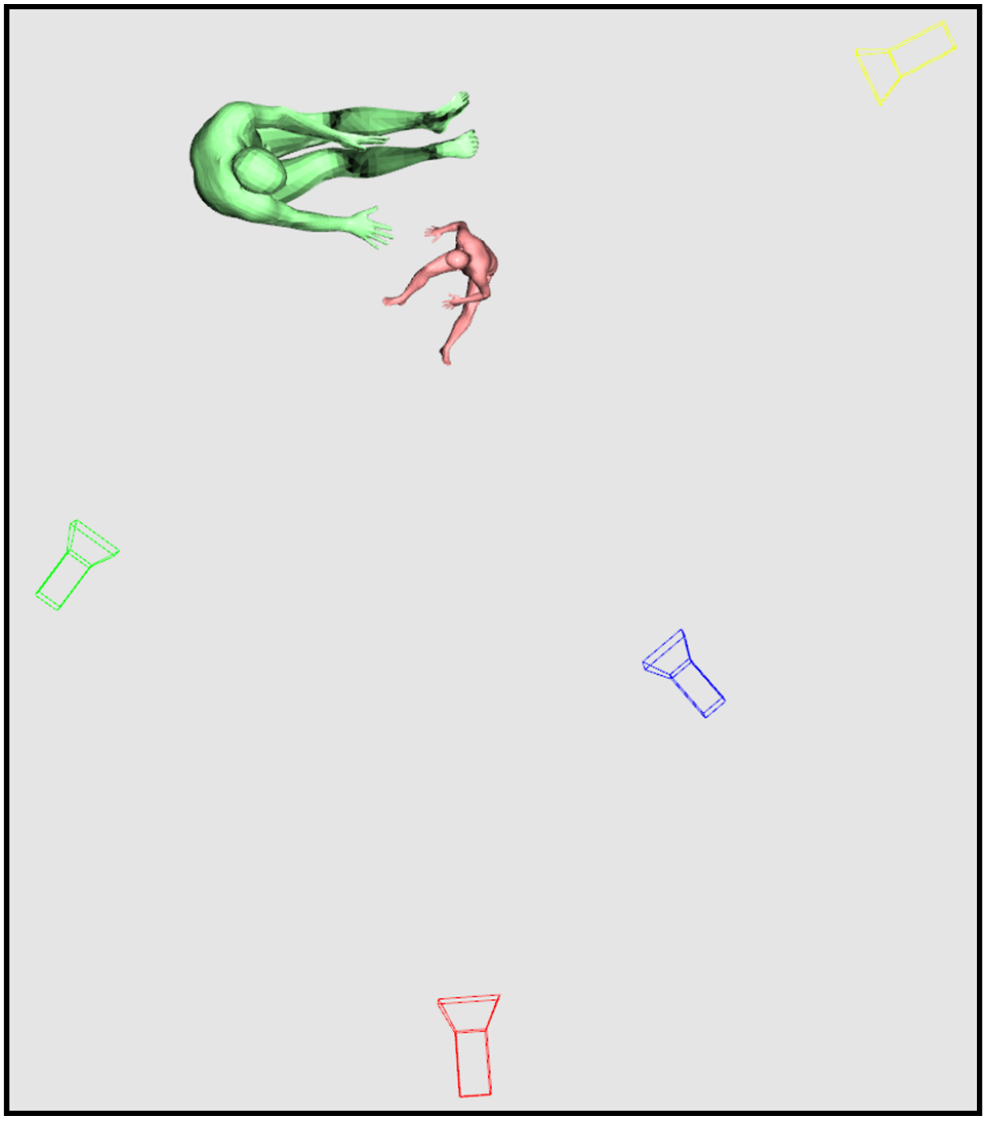

Figure 31: Bird eye view of the initial frame of the reconstructed demonstration video.

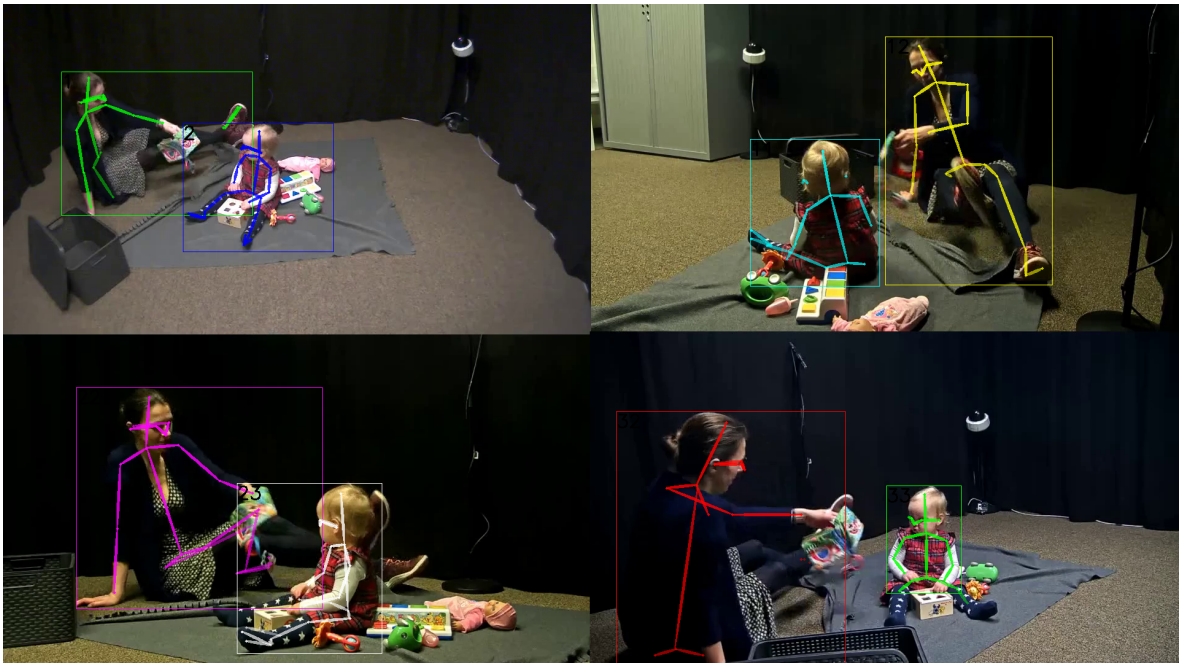

Figure 32: Illustrative example of the data used to perform the reconstruction depicted in Figure 31

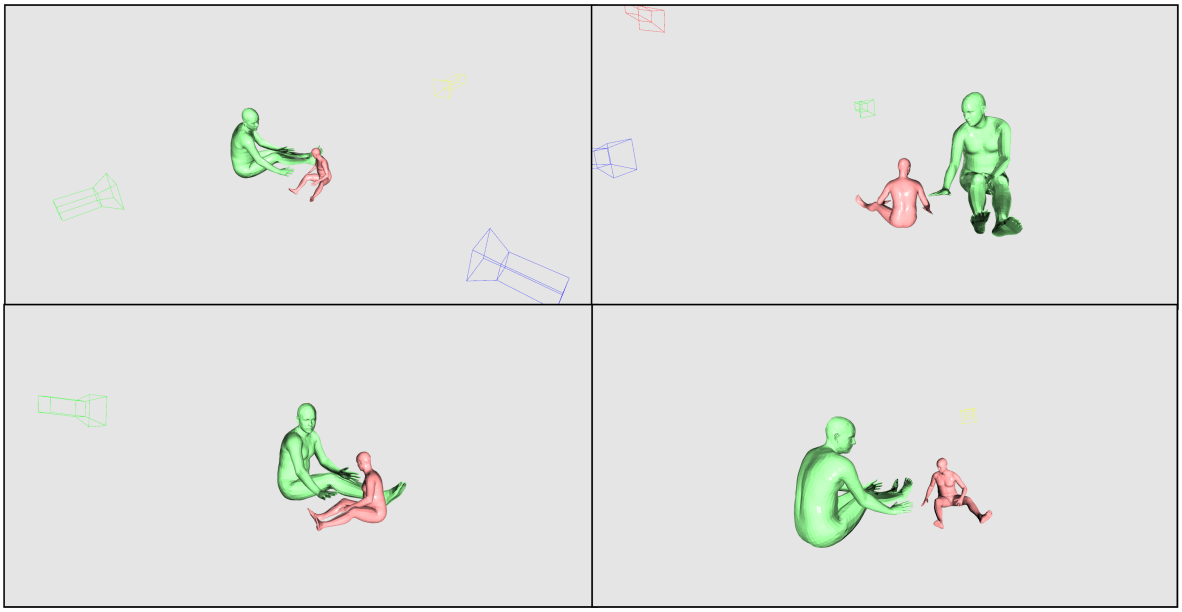

Figure 33: Reconstructed data of Figure 32, according to each camera view.

Figure 34: Reconstruction visualization of video *B33892*, when the parent picks up the infant.

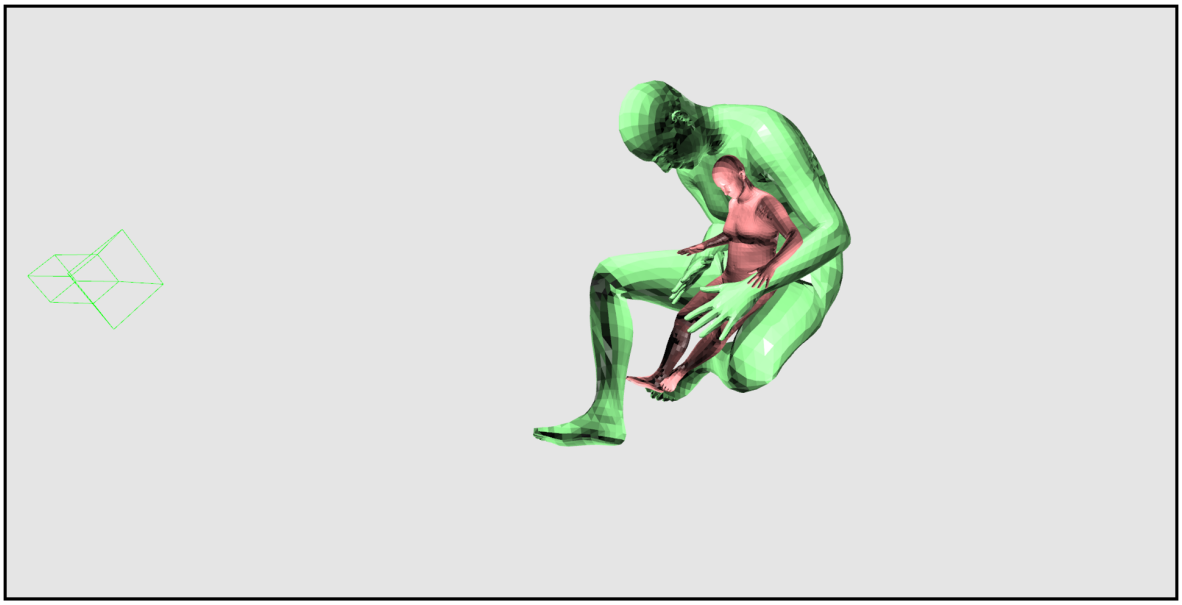

Figure 35: Reconstruction visualization of video *B89136*, while the infant is on the lap of the parent, and both interact with the same toy.

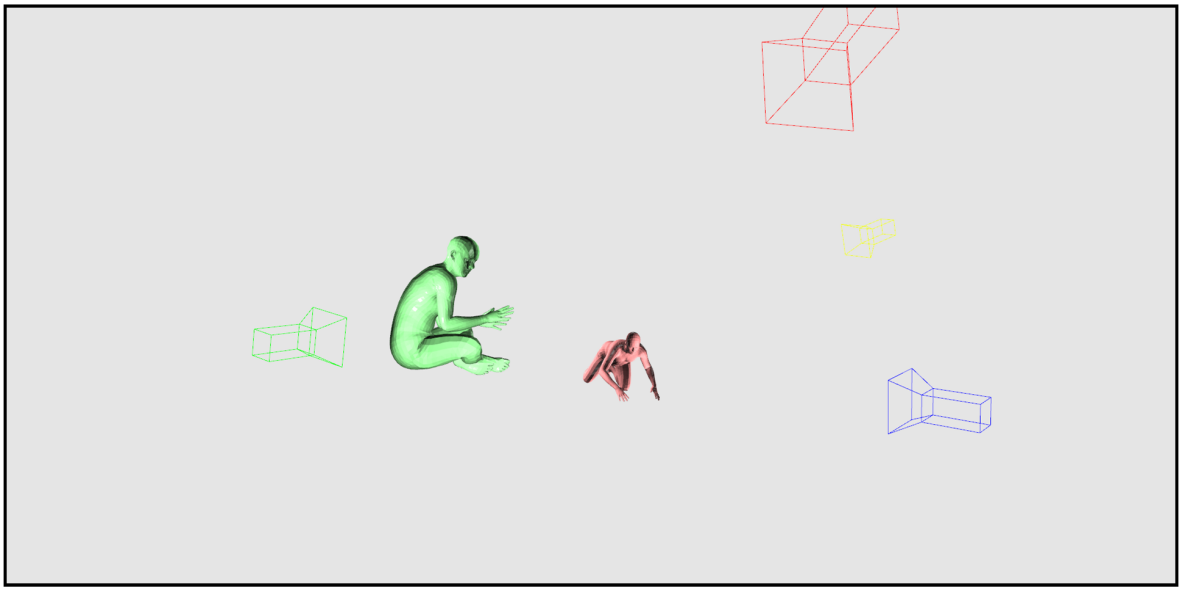

Figure 36: Reconstruction visualization of video *B47859*, during the moment in which the infant crawls away from the parent.

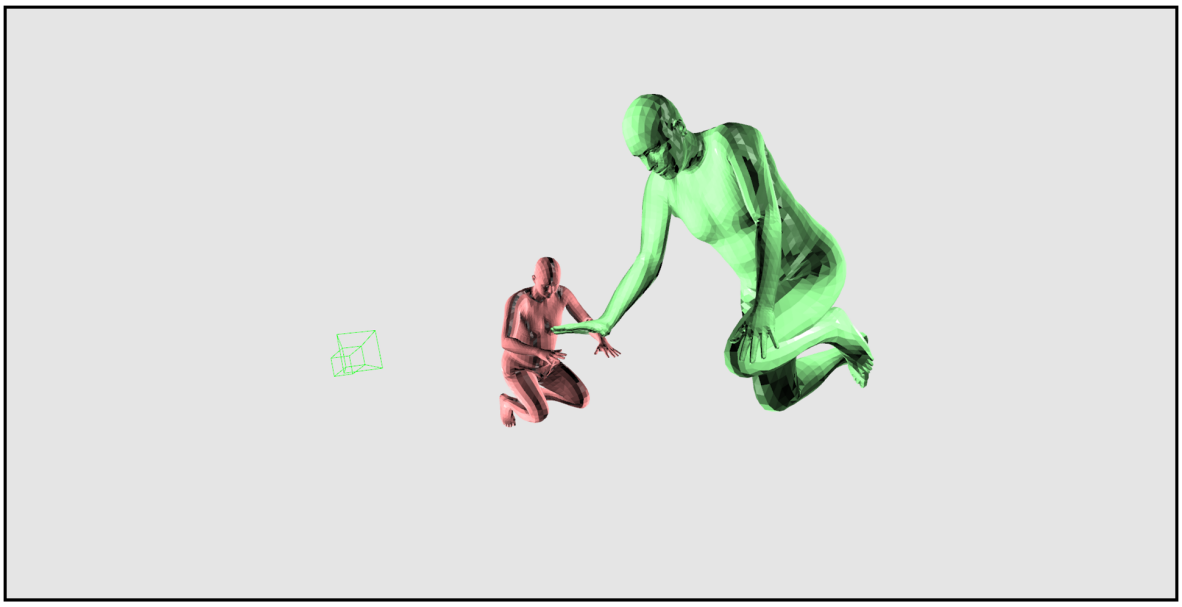

Figure 37: Reconstruction visualization of video *B83755*, while both individuals interact with the same toy.

A.2 Auxiliary Tables

Keypoint Confidence Values

YOUth	Parent Keypoint Conf.				Infant Keypoint Conf.				Error Count	
Video Name	V0	V1	V2	V3	V0	V1	V2	V3	ReID	AP
B33718 00:13:40 - 00:14:15	0.687	0.695	0.720	0.748	0.744	0.665	0.607	0.619	202	88
B33892 00:10:00 - 00:10:35	0.644	0.723	0.632	0.740	0.646	0.609	0.668	0.548	16	5
B35985 00:10:50 - 00:11:25	0.646	0.684	0.741	0.695	0.657	0.713	0.610	0.495	119	108
B38777 00:11:17 - 00:11:52	0.648	0.680	0.708	0.841	0.636	0.428	0.561	0.630	43	37
B40508 00:06:55 - 00:07:30	0.731	0.808	0.724	0.651	0.628	0.614	0.674	0.668	479	181
B44801 00:02:55 - 00:03:25	0.613	0.687	0.595	0.412	0.613	0.592	0.740	0.515	95	19
B45111 00:11:26 - 00:12:01	0.777	0.616	0.711	0.568	0.643	0.794	0.690	0.515	34	38
B45358 00:01:25 - 00:02:00	0.642	0.703	0.727	0.798	0.743	0.719	0.715	0.621	0	4
B47859 00:07:35 - 00:08:05	0.715	0.659	0.683	0.696	0.517	0.771	0.682	0.770	0	6
B51848 00:15:35 - 00:16:05	0.851	0.515	0.771	0.684	0.584	0.823	0.602	0.638	136	63
B64396 00:14:00 - 00:14:35	0.719	0.598	0.680	0.638	0.706	0.548	0.739	0.609	430	226
B64612 00:01:53 - 00:02:23	0.673	0.693	0.605	0.577	0.783	0.544	0.739	0.680	63	18
B67411 00:14:35 - 00:15:10	0.789	0.630	0.780	0.797	0.791	0.843	0.805	0.673	0	8
B70410 00:01:20 - 00:01:55	0.795	0.678	0.684	0.723	0.737	0.762	0.770	0.634	0	9
B83755 00:10:30 - 00:11:05	0.736	0.741	0.705	0.751	0.652	0.682	0.709	0.668	0	7
B86218 00:06:00 - 00:06:35	0.759	0.745	0.757	0.606	0.789	0.582	0.801	0.831	0	12
B89136 00:06:25 - 00:07:00	0.702	0.726	0.728	0.529	0.820	0.762	0.746	0.710	315	142
B93177 00:14:25 - 00:15:00	0.757	0.719	0.805	0.637	0.744	0.741	0.763	0.656	0	12
B97605 00:06:45 - 00:07:20	0.746	0.748	0.774	0.689	0.503	0.635	0.576	0.444	0	7
Average	0.712	0.685	0.692	0.719	0.658	0.618	0.619	0.564	117.474	52.105

Table 8: Average 2D pose estimation keypoint confidence score of each individual, in each view. Error count reflects the number of total *AlphaPose* (AP) and *Torchreid* (ReID) errors, of each video

Re-projection Error - Camera Optimization

Avg Re-Projection Error - P1		
Video Name	Parent	Infant
B33718	50.163	40.810
B33892	28.157	21.746
B35985	84.868	98.602
B38777	28.882	31.740
B40508	45.802	45.641
B44801	50.194	56.587
B45111	70.625	39.586
B45358	54.388	53.982
B47859	24.640	23.902
B51848	66.910	57.428
B64396	51.780	54.194
B64612	42.282	27.566
B67411	25.003	19.097
B70410	50.594	28.838
B83755	34.581	27.524
B86218	35.762	14.752
B89136	36.135	19.556
B93177	37.203	18.893
B97605	34.483	42.494
Average	44.866	38.211

Table 9: Average keypoint re-projection error, during phase one (P1) of Step 3.5

YOUth	Parent Re-Proj. Error				Infant Re-Proj. Error			
Video Name	V0	V1	V2	V3	V0	V1	V2	V3
B33718 00:13:40 - 00:14:15	31.406	56.801	52.129	60.314	17.593	26.771	56.484	62.393
B33892 00:10:00 - 00:10:35	28.904	29.742	26.292	27.688	17.812	22.011	19.889	27.270
B35985 00:10:50 - 00:11:25	118.354	96.719	58.718	65.679	111.741	44.804	133.278	104.586
B38777 00:11:17 - 00:11:52	33.863	29.131	28.701	23.833	18.883	44.44	35.489	28.147
B40508 00:06:55 - 00:07:30	31.646	25.880	48.443	77.241	34.332	38.932	45.329	63.972
B44801 00:02:55 - 00:03:25	52.946	31.841	42.094	73.894	27.072	29.752	20.725	148.800
B45111 00:11:26 - 00:12:01	34.828	69.376	96.183	82.112	76.904	16.202	40.782	24.456
B45358 00:01:25 - 00:02:00	36.601	67.932	51.054	61.963	45.807	34.423	91.063	44.634
B47859 00:07:35 - 00:08:05	17.332	29.261	22.220	29.746	35.017	17.891	25.265	17.434
B51848 00:15:35 - 00:16:05	22.573	134.157	38.319	72.593	44.410	58.379	40.851	86.074
B64396 00:14:00 - 00:14:35	26.888	94.308	35.163	50.762	22.477	123.679	21.682	48.936
B64612 00:01:53 - 00:02:23	25.728	43.311	51.094	48.996	12.113	46.443	26.823	24.883
B67411 00:14:35 - 00:15:10	15.842	35.774	20.651	27.745	14.203	11.251	14.826	36.107
B70410 00:01:20 - 00:01:55	18.483	53.290	70.218	60.386	15.143	18.379	20.045	61.785
B83755 00:10:30 - 00:11:05	27.441	35.238	33.482	42.163	24.206	19.606	25.997	40.285
B86218 00:06:00 - 00:06:35	19.340	18.665	25.250	79.792	7.317	29.410	11.058	11.225
B89136 00:06:25 - 00:07:00	21.402	32.483	22.786	67.871	10.757	26.970	17.986	22.512
B93177 00:14:25 - 00:15:00	21.134	39.678	27.209	60.790	13.146	15.840	17.564	29.022
B97605 00:06:45 - 00:07:20	20.623	35.688	27.400	54.221	30.682	25.831	39.637	73.827
Average	50.163	28.157	84.868	28.882	40.810	21.746	98.602	31.740

Table 10: Average keypoint re-projection error, during phase one of Step 3.5

Re-projection Error - Fixed Camera Parameters

Avg Re-Projection Error - P2			**Difference**	
Video Name	Parent	Infant	Parent	Infant
B33718	47.271	39.220	-2.892	-1.590
B33892	27.251	20.971	-0.906	-0.775
B35985	83.912	95.938	-0.956	- 2.664
B38777	53.33	29.062	24.448	-2.678
B40508	45.232	45.136	-0.570	-0.505
B44801	54.065	55.146	3.871	-1.441
B45111	82.954	38.699	12.329	-0.887
B45358	51.644	53.313	-2.744	-0.669
B47859	24.197	23.187	-0.443	-0.715
B51848	72.410	56.214	5.500	1.214
B64396	51.484	53.330	-0.296	-0.864
B64612	63.690	25.942	21.408	-1.624
B67411	22.918	18.312	-2.085	-0.785
B70410	53.700	28.338	3.106	-0.500
B83755	33.678	26.741	-0.903	-0.783
B86218	33.596	14.153	-2.166	-0.599
B89136	35.798	19.407	-0.337	-0.149
B93177	23.849	17.938	-13.354	-0.955
B97605	43.585	40.596	9.102	-1.898
Average	47.609	36.842	2.665	-1.036

Table 11: Average keypoint re-projection error, during phase two of Step 3.5

YOUth	Parent Re-Proj. Error				Infant Re-Proj. Error			
Video Name	V0	V1	V2	V3	V0	V1	V2	V3
B33718 00:13:40 - 00:14:15	28.772	53.056	40.017	67.239	15.767	23.809	55.126	62.176
B33892 00:10:00 - 00:10:35	28.636	28.271	25.969	26.130	17.069	21.032	18.782	27.001
B35985 00:10:50 - 00:11:25	113.992	92.388	62.966	66.302	120.471	45.800	125.798	91.682
B38777 00:11:17 - 00:11:52	57.726	76.756	53.213	25.626	17.419	43.445	31.308	24.077
B40508 00:13:45 - 00:14:20	30.988	25.875	47.526	76.539	34.732	39.199	44.427	62.185
B44801 00:02:55 - 00:03:25	56.637	47.319	42.143	70.160	27.071	29.752	20.724	148.800
B45111 00:11:26 - 00:12:01	34.749	79.621	104.533	112.912	76.925	15.453	38.419	23.997
B45358 00:01:25 - 00:02:00	38.993	62.800	54.332	50.449	44.330	35.067	88.321	45.546
B47859 00:07:35 - 00:08:05	17.143	29.131	21.181	29.333	34.501	17.404	24.377	16.464
B51848 00:15:35 - 00:16:05	22.562	154.095	37.201	75.785	43.408	56.595	40.502	84.352
B64396 00:14:00 - 00:14:35	28.002	89.805	34.874	53.254	22.540	122.681	20.212	47.886
B64612 00:01:53 - 00:02:23	43.715	74.389	48.944	87.713	11.080	44.041	25.044	23.604
B67411 00:14:35 - 00:15:10	14.198	34.999	18.369	24.106	13.833	10.482	14.032	34.903
B70410 00:01:20 - 00:01:55	19.139	70.429	68.301	56.930	15.058	17.754	19.336	61.204
B83755 00:10:30 - 00:11:05	26.933	33.530	33.221	41.027	23.672	19.367	24.998	38.926
B86218 00:06:00 - 00:06:35	18.764	17.316	23.026	75.278	6.825	29.701	9.940	10.145
B89136 00:06:25 - 00:07:00	21.230	32.141	22.212	67.609	10.689	26.723	17.880	22.337
B93177 00:14:25 - 00:15:00	13.409	23.425	14.579	43.983	12.775	15.009	16.227	27.742
B97605 00:06:45 - 00:07:20	26.121	52.635	29.883	65.699	30.642	23.661	35.953	72.129
Average	47.271	27.251	83.912	53.330	39.220	20.971	95.938	29.062

Table 12: Average keypoint re-projection error, during phase two of Step 3.5

A.3 Auxiliary Plots

Missing Data - B333892

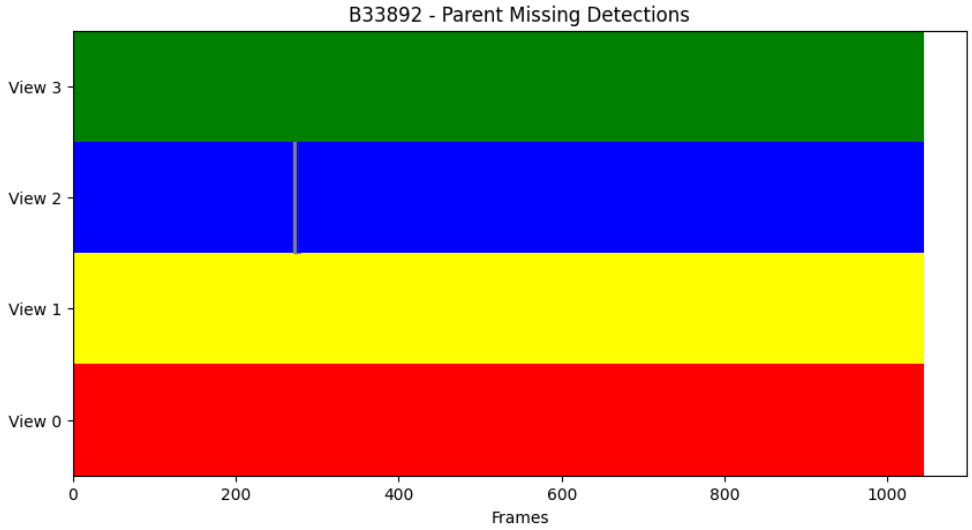

Figure 38: Frame-by-frame sequence of the captured (colored) and uncaptured (gray) detections of the parent in video *B333892*

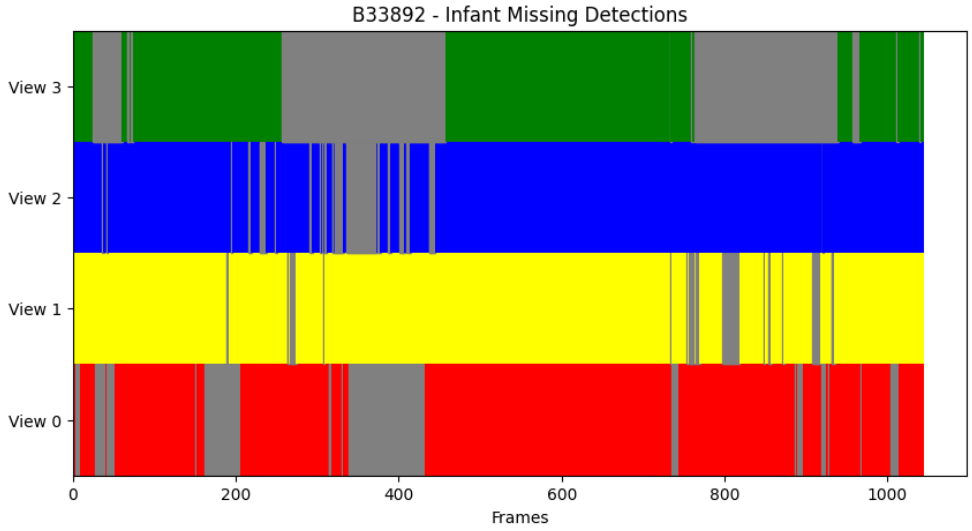

Figure 39: Frame-by-frame sequence of the captured (colored) and uncaptured (gray) detections of the infant in video *B333892*

Missing Data - B40508

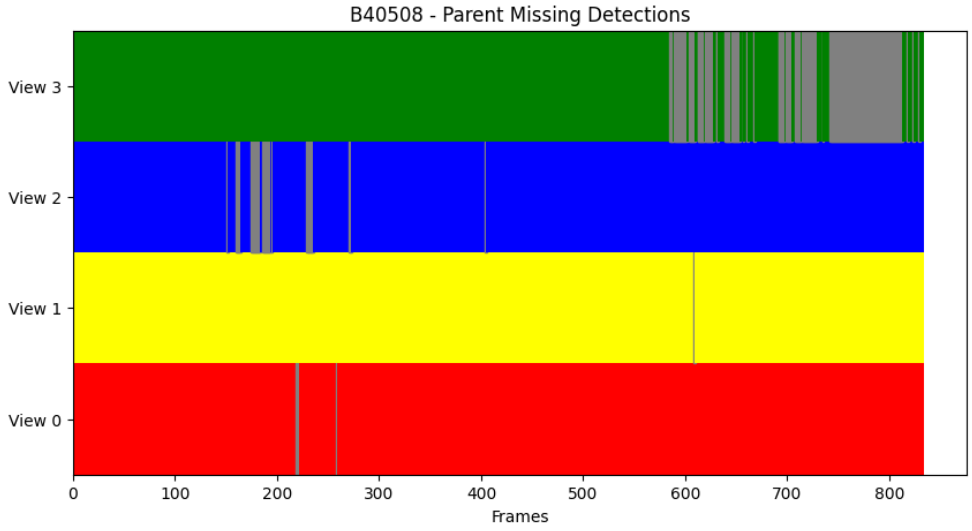

Figure 40: Frame-by-frame sequence of the captured (colored) and uncaptured (gray) detections of the parent in video *B40508*

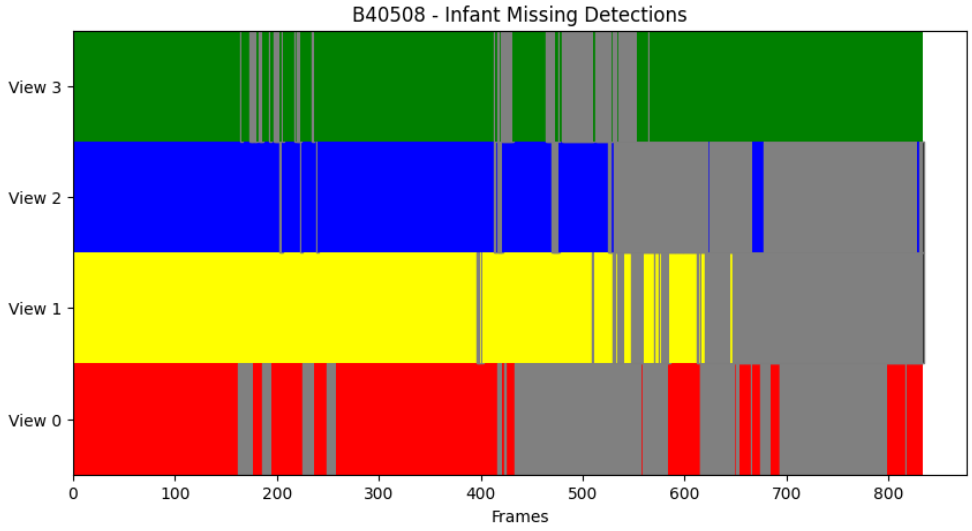

Figure 41: Frame-by-frame sequence of the captured (colored) and uncaptured (gray) detections of the infant in video *B40508*

Missing Data - B64396

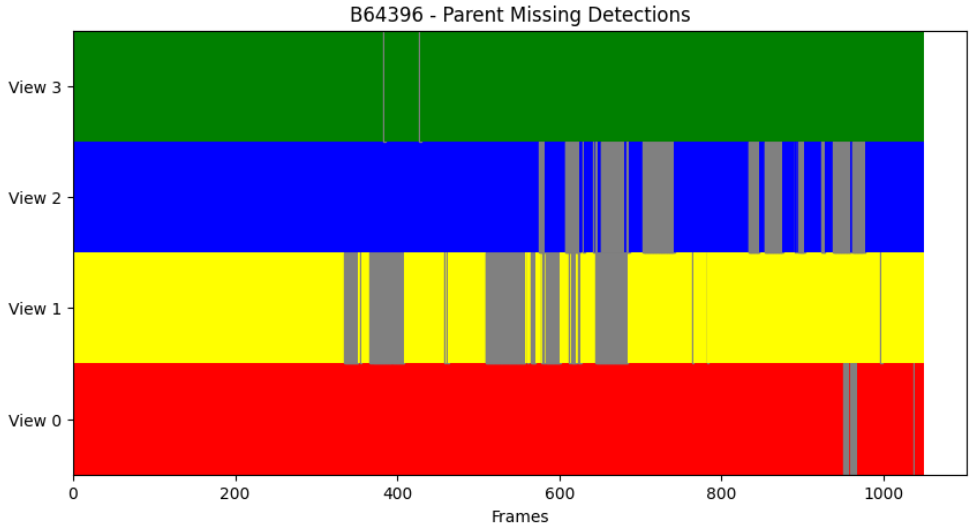

Figure 42: Frame-by-frame sequence of the captured (colored) and uncaptured (gray) detections of the parent in video *B64396*

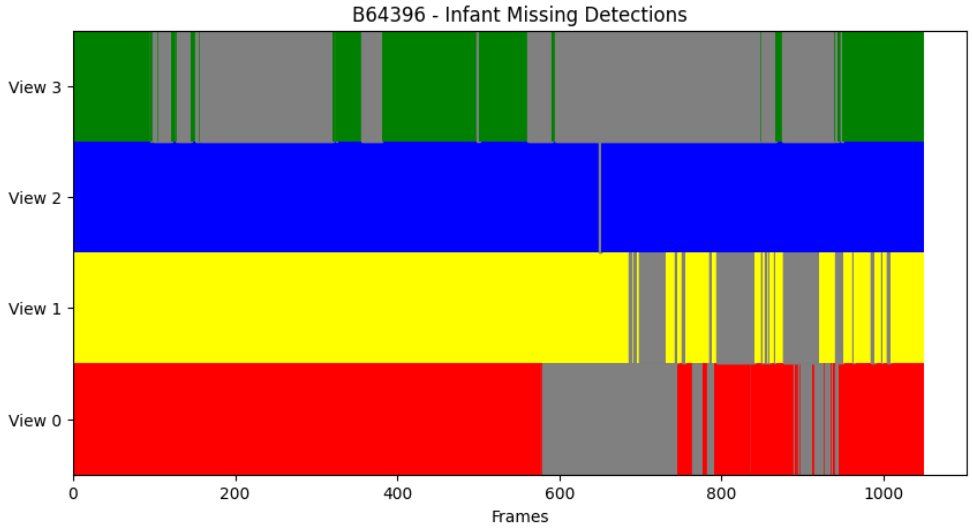

Figure 43: Frame-by-frame sequence of the captured (colored) and uncaptured (gray) detections of the infant in video *B64396*

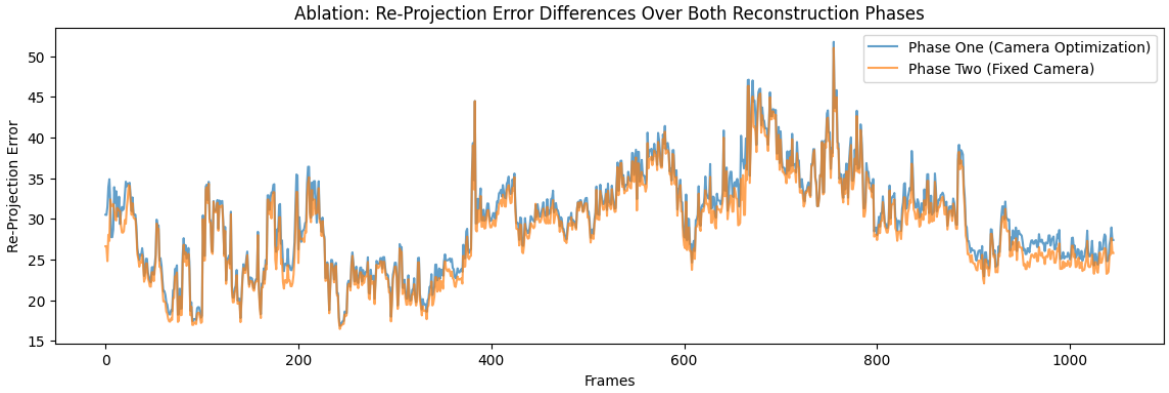

Figure 44: Re-projection error over time during phase one and phase two of the demonstration video reconstruction