Capturing Multi-Person Interactions from Videos

Emilia Szymanska
Robotics, Systems and Control
eszymanska@student.ethz.ch

Yuanchen Yuan
Robotics, Systems and Control
yuayuan@student.ethz.ch

Johannes Gaber
Robotics, Systems and Control
jgaber@student.ethz.ch

Piotr Libera
Robotics, Systems and Control
plibera@student.ethz.ch

Dr. Yan Zhang
project supervisor
yan.zhang@inf.ethz.ch

Abstract

This project addresses the problem of markerless motion capture of multiple persons in different scenes. The goal is to estimate the 3D poses of each person in multi-view videos depicting multiple persons performing various tasks and interacting with each other. To solve this problem, we created a pipeline that generates sequences of 3D LISST body parameters that correspond to individual persons. In our pipeline, we used OpenPose to recover 2D poses from individual frames and match them across views and over time with a greedy matching algorithm available in mv3dpose github repository. Additionally, we included a track stitching algorithm and used motion priors to achieve more accurate and smooth tracking even with many persons in the scene. In this report, we present the details of our approach and the performance of our pipeline on the CMU Panoptic and EgoBody datasets. We analyze our solution with ablation studies and present the shortcomings of additional possible elements of the pipeline that we did not use, such as visual similarity matching.
Code: github.com/emilia-szymanska/3D_pose_estimation

1. Introduction

Markerless motion capture is the task of capturing the movement of persons in the scene without the use of any physical markers. The goal of this task is to track human motion by only using multiple synchronized and calibrated standard 2D cameras that capture the scene. Accurate and robust solutions to this problem have many applications, e.g. in the entertainment industry (where marker-based motion capture is currently widely used), biomechanics research, or sports analysis. The ability to estimate and track the 3D poses of multiple persons in the scene can enable recovery and analysis of their interactions without the need to interfere with the analyzed persons and their environment.

To create a working solution to the problem of markerless motion capture, several challenges need to be overcome even if there is only one person in a very simple scene. These challenges include the initial keypoint and pose detection in individual frames, and 3D pose estimation that accounts for inaccuracies of keypoint estimation and their incompleteness due to occlusions of some body parts. Hence even in this simplified case, accurate motion capture require the use of body priors. The goal of this project, however, is to recover interactions of multiple persons in the scene, which introduces multiple additional challenges. Since multiple poses can be detected in each frame, each 2D pose needs to be matched and associated with poses detected in frames from other views. Only with a correct association, keypoints in detected poses can be triangulated to obtain 3D poses of persons in the scene. Furthermore, those poses need to be tracked over time, since all persons can dynamically move in the scene and interact with each other. The presence of many persons and objects in the scene introduces many more occlusions and ambiguities at various stages of any markerless motion capture pipeline. In section 2, we discuss different approaches that have been recently proposed to solve these challenges. Even though a lot of progress has been made in the field of markerless motion capture, the state-of-the-art methods still suffer from improper pose estimation and pose matching in challenging scenes, such as scenes with many persons of similar appearance.

In our pipeline, we build on recent solutions for pose detection and 3D pose estimation. We use OpenPose [7] to estimate 2D poses in individual frames from each view. This is a framework widely used as a first step of 3D pose estimation and tracking. However, it does not provide tracking of poses over consecutive frames. For pose matching, we use the algorithm presented in *mv3dpose* [9]. This algorithm greedily matches 2D poses across views based on keypoint positions and epipolar geometry, and 3D poses based on their position after each 3D pose has been triangulated. We build on those two elements to create our 3D pose estimation pipeline, which contains additional elements that aim to address the issues of recent 3D pose estimation pipelines: track stitching, LISST [33] integration, and motion priors. Hence, the main contributions of our project are:

- **Track stitching** – we improved the performance of matching 3D poses over time by adding a track stitching algorithm. With this algorithm, we addressed issues of the basic mv3dpose algorithm, that struggled with correct tracking in longer sequences, especially with multiple persons interacting with each other in the scene.

- **LISST integration** – we integrated the 3D pose estimation pipeline with LISST pose and shape priors.

- **Motion priors** – to improve the smoothness of movement in consecutive 3D body poses of each person we used motion priors and included them in the recovery of LISST parameters.

- **Visual similarity matching** – we additionally analyzed the method of pose matching with additional visual cues with several approaches, based on both feature detection and machine learning methods. However, the approaches we considered did not perform well with large viewpoint changes across views and similar appearance of some persons in the scene.

2. Related Work

3D human pose estimation. Estimating 2D (and further on 3D) human poses from videos is one of the highly challenging topics in computer vision research. State-of-the-art solutions still do not match the ground truth data and each year more and more improvements appear. One of the first trials to recognize human movements in images [1] modeled human bodies with 3D cylinders and applied a Kalman filter to estimate model parameters over the frames. Ten years later, Felzenszwalb and Huttenlocher [2] made an attempt to use pictorial structures to estimate human poses. They perceived a human body as a collection of parts connected in a deformable configuration and represented the overall appearance of each part with the use of a patchwork of local features. Another approach focused on sequential prediction taking advantage of convolutional neural networks [3]. It allowed for capturing spatial relationships between body parts, and by extension refining the estimates for the part locations. Neural networks (especially convolution-based ones) proved to yield satisfying results, therefore other researchers based their methods on them as well [4–6].

One of the most successful and currently widely used solutions for 2D human pose estimations is OpenPose [7]. This framework uses part affinity fields to model the spatial relationships between body parts and later assign them to multiple people in a frame. As for now, however, this solution provides only 2D poses and does not perform tracking over frames from the same video. Yet, due to its effectiveness, open-source access, and in-progress improvements, it was selected as the starting point for the 3D pose estimation in this project.

3D pose estimation can be based on 2D pose estimation from multi-view videos from calibrated cameras. Some methods use only geometric cues to obtain 3D joint estimates [8–10], but an approach integrating the appearance features into the geometric-based pipeline yields promising results [11].

Multi-person matching and tracking. One of the challenges in the project was correctly identifying, which poses from different views correspond to the same person. As OpenPose can return noisy data, the geometrical similarity may not be enough to define the matches. There exist multiple techniques dealing with matching images taken from different viewpoints, at different distances, and with varying illumination. There are two main trends: feature-based matching and machine learning approach. The first one focuses on extracting keypoints from images and then performing a matching based on the distance between features. Some of the commonly known algorithms for keypoint extracting (and later matching) are Scale-Invariant Feature Transform (SIFT) [19], Speeded Up Robust Features (SURF) [12], Binary Robust Independent Elementary Features (BRIEF) [13], ORB (Oriented FAST and Rotated BRIEF) [18], and Features from Accelerated Segment Test (FAST) [14]. Machine learning, however, has gained popularity over the years in this area. Deep learning has been used for identifying the embedding space in which similar images are close to each other [15], learning local image descriptors [17], or estimating the image homography [16].

Tracking multiple objects is commonly formulated as a data association problem, where a track-by-detection method can be applied to re-identify target objects across different frames. Detection tracking lies in implementing visual and motion association elements, which can be addressed with robust appearance models such as color histograms [22] and deep network features [24], and linear [21, 22] and non-linear [23] motion models for motion consistency. Incorporating tracking across multiple different views introduces another layer of complexity to multi-object tracking as data association is required between cameras as well. Researched solutions focus on both overlapping [29, 30]and disjoint [27, 28] camera views, with the general approach being to generate tracklets that contain geometric and/or appearance features over a period of time and to associate the tracklets into complete tracks for each detected person. Depending on the complexity and robustness of the tracked data, tracklets can be associated through machine learning approaches [24, 32] or with algorithmic methods such as combinatorial optimization [31], parametrized quadratic optimization [25], or even the simple greedy algorithm [9]. In this project, we decided to build upon the approach presented in [9], because it was open source and had some room for improvement in terms of track generation and probable appearance cue integration.

LISST integration. To recover sequences of LISST parameters, LISST already provides an optimizer framework [33]. It comes with basic smoothening functions and a pose- and shape-prior. To leverage the quality of recovering, Zhang et. al. [36] suggests to train a convolutional autoencoder with perfect motions, to implement a soothening loss in latent space that works as a motion prior. The results look promising. Chen et. al. [38] propose a very similar approach, using a variational autoencoder. Due to the availability of a codebase from Zhang et. al. and the closer proximity to LISST, we choose this approach.

3. Method

We provide an overview of our approach in Fig 1. Based on multiple videos of a scene from different views, our pipeline extracts a sequence of LISST [33] parameters for each person in the scene, that represents a smooth and natural-looking motion. We use OpenPose [7] to extract 2D keypoints per frame of humans from the input videos. We then match these 2D keypoints from different views to the right person and gain 3D positions of the keypoints via triangulation based on the camera parameters. The 3D sequences are prone to have missing or wrong points due to occlusion and tracking errors. Therefore we calculate the final LISST parameters based on pose, shape, and motion priors as well as applying velocity and distance smoothing on the output motions. We use the pose and shape priors provided by LISST [33] and train our own motion priors via convolutional autoencoder. With this approach, we can fill missing data with plausible motions and achieve natural-looking results.

3.1. 3D pose extraction

In accordance with the *mv3dpose* pipeline presented in [9], 3D human poses are first estimated from video data and then greedily matched together to form tracks that illustrate the movements of each individual throughout the videos. To estimate 3D poses, we first obtain the 2D poses of all detected persons in each frame at every view using OpenPose [7]. Each person is then discretized through 2D pose association, where at each frame, every detected 2D human pose from the first camera view is chosen to be a person candidate. The 2D poses from other views are then greedily matched to the existing person candidates through bipartite matching. The cost for 2D pose matching is derived from projecting joint positions along epipolar lines from one view to another and then comparing the location similarity between the projected and original joint. After all 2D poses for one frame have been associated, the 3D human poses are estimated through triangulation and then associated with the poses from the previous frame through once again bipartite matching. The cost for pose assignment is calculated from the mean Euclidean distance between common joint positions of two poses, and the two poses are assigned to one track if their pose distance falls below a certain threshold.

We discovered that for videos with many occlusions and/or unreliable OpenPose outputs, the original *mv3dpose* proposed pipeline would have trouble successfully tracking persons which resulted in many track fragments instead of complete tracks that ran from the beginning frame to the end frame. To mitigate this issue, we developed an additional step to the 3D pose extraction part of the pipeline, where we attempt to stitch the track fragments together based on time step differences and joint distances. Algorithm 1 summarizes the stitching approach we implemented.

Algorithm 1 Track stitching algorithm

```
Initialize empty list of new tracks
Initialize track ID counter
for each track in tracks do
    if track starts at frame 0 then
        Add to new tracks list
    else
        Compute pose distance between track and all tracks in new tracks list
        if distance is below threshold then
            if track frames are not already encompassed by existing track then
                Merge track with closest existing track in new tracks list
            end if
        else
            Add track as a new track in new tracks list
        end if
    end if
end for
Save all tracks in new tracks list
```

After stitching, tracks shorter than 2 seconds are removed. The final stitched 3D tracks are then processed into LISST parameters.

3.2. Visual similarity matching

The geometry-based approach to matching people across the views and frames may not be enough to correctly perform the tracking. To compensate for the geometrical

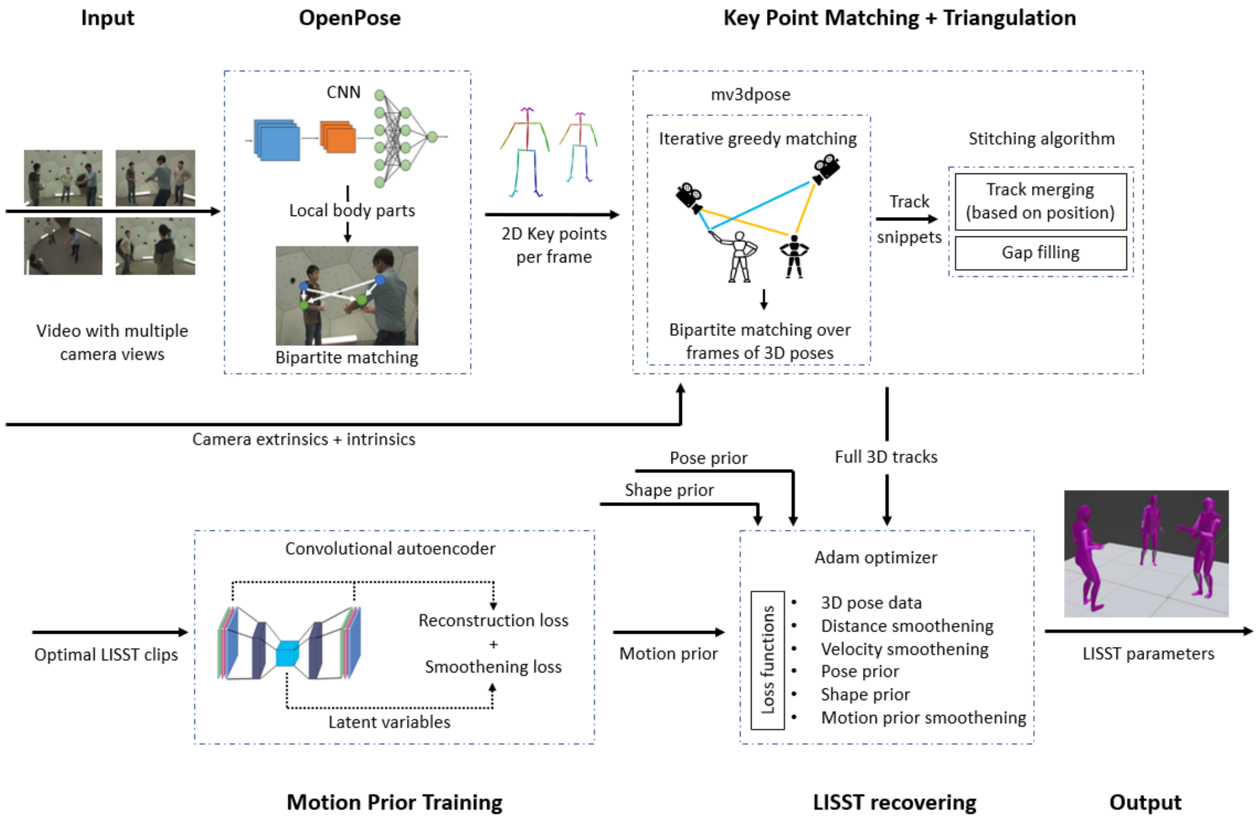

Figure 1. Pipeline flowchart of the implemented method.

method imperfections, an appearance similarity matching would need to be implemented. The idea was to confirm the geometrical matches between poses from different views corresponding to the same person. Three approaches were selected for testing.

The first two methods implement the idea of feature matching. It focuses on defining a set of interest points (called features) in two images and then performing a brute force matching between them so that the features which are closest with respect to each other are marked as corresponding ones.

The first descriptor used in the feature matching was the Oriented FAST and Rotated BRIEF (ORB) descriptor [18]. It uses the intensity-weighted centroid of the keypoint patch and extracts the orientation as the direction of this centroid with reference to the keypoint. It is a binary descriptor, hence it allows for a very fast matching, but also adds the feature of being rotation invariant. Brute force matching is performed with the Hamming distance criterion.

The second descriptor, Scale Invariant Feature Transform (SIFT) [19] was selected due to its scale invariance, feature locality, and effectiveness. It is robust to up to 50^o viewpoint changes and non-affine illumination shifts. Its features are local extrema in both space and scale of the Difference of Gaussian images. To avoid false matches, the ratio of closest-distance keypoint to second-closest distance keypoint is taken into consideration – it has to be lower than 0.75 for the match to not be rejected.

The third approach, however, utilizes a neural network Contrastive Language–Image Pre-training (CLIP) [20] created by the OpenAI development team. Although the model is mainly used for predicting the most relevant text description for an image out of the provided candidates, the idea was to encode the cropped images into vector space and afterward find high density regions corresponding to fairly similar areas in images. It performs well in finding resembling image pairs, therefore it was decided to perform trials on images from different viewpoints.

To assess all three methods, two sample images of the same time frame but different camera ids were extracted. Then, OpenPose detections were applied and the bounding boxes for each detected person were created (the minimum and maximum x and y positions were determined from the joint locations). These bounding boxes were used for cropping people from the images to later perform the similarity checks.

3.3. LISST parameter recovering

The final step is to convert the tracks, which contain the 3D positions of the OpenPose skeleton, into tracks of LISST parameters. These should be complete and natural looking. Based on the ETH Digital-Humans course assignment 3 [33], we formulate the conversion as an optimization problem. The following challenges will be taken into account:

- Different joint locations and number of joints due to different base skeletons of OpenPose and LISST.
- Incomplete tracks and missing joints due to occlusion, tracking errors, mismatching, etc.
- Outliers
- Jittering due to frame-by-frame pose extraction.

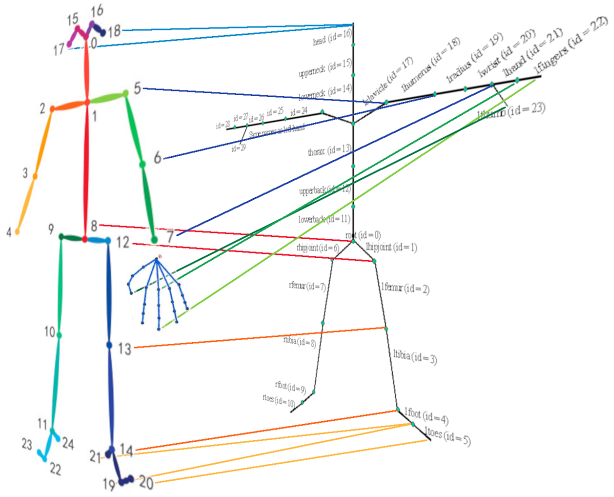

Figure 2. Symbolic mapping from OpenPose to LISST.

As shown in Fig 2, we assume a basic similarity between the two skeletons. When two joints in OpenPose are the basis of only one joint in LISST, we take the mean position (see head). Some joint positions in LISST do not have a correspondence from OpenPose (see spine). An exact mapping and interpolation/extrapolation of the missing joints would be very complicated and probably highly ill-posed in case of more missing joints. We tackle the problem of missing joint positions and offsets between the skeletons with pose and shape priors and smoothing during the optimization process.

We assume a sequence of T frames with B people and $J = 31$ joints in the LISST parameterization. From the mapping we receive a sequence of joint locations $J_{data} \in \mathbb{R}^{T\times B\times J\times 3}$ with a boolean mask $J_{mask} \in \{0,1\}^{T\times B\times J}$, indicating if the joint locations are valid or missing.

As in assignment 3, we introduce the optimization variables bone length $l_{bone,latent} \in \mathbb{R}^{B\times 12}$ and local joint rotation $J_{rot,canonical,latent} \in \mathbb{R}^{T\cdot B\times J\times z_d}$ in latent space, the root location $r_{locs} \in \mathbb{R}^{T\times B\times 3}$ and the transformed rotation of the bodies $r_{rot} \in \mathbb{R}^{T\times B\times 6}$. The variables will get initialized with zero. Due to the normalization of the priors, these are the most valid poses and shapes. We use the pose and shape priors from assignment 3 and can therefore use the according decoder to get l_{bone} and $J_{rot,canonical}$ and via forward kinematics receive the joint locations $J_l \in \mathbb{R}^{T\times B\times J\times 3}$ and rotations $J_{rot} \in \mathbb{R}^{T\times B\times J\times 3\times 3}$

$$J_l, J_{rot} = fk(r_{locs}, J_{rot,canonical}, r_{rot}, l_{bone}) \tag{1}$$

in world coordinates.

These can be used to apply loss functions:

- Data loss to ensure, that the LISST skeleton follows the OpenPose skeleton:

$$l_{data} = \frac{||J_{mask} \circ (J_{data} - J_l)||_1}{3 \cdot ||J_{mask}||_1} \tag{2}$$

- Motions are continuous. Body parts cannot move very far between one frame and the next one. To avoid jittering, we penalize big movements with a distance smoothing loss. Empirically we figured, for some body parts (e.g. hands, arms), sudden movements are way more common than for others (e.g. hips, legs). Therefore we weigh each loss per joint with $w_d \in \mathbb{R}^J$:

$$l_{sd} = \frac{1}{3(T-1)BJ} \sum_{i=1}^{T-1} \sum_{j=1}^{B} \sum_{k=1}^{J} \sum_{l=1}^{3} w_{d,k} |(J_l)_{i+1,j,k,l} - (J_l)_{i,j,k,l}| \tag{3}$$

- Due to low fps rates or fast-moving body parts, the previous loss term can also have negative effects through also penalizing real movements. We assume that movements that go in the same direction over multiple frames are not outliers. We therefore introduce a velocity loss, that only penalizes new directions of movement. We obtain the velocity matrix $V \in \mathbb{R}^{(T-1)\times B\times J\times 3}$ for all time steps t with $V_t = (J_l)_{t+1} - (J_l)_t$. Then we get the velocity smoothing loss with:

$$l_{sv} = \frac{1}{3(T-2)BJ} \sum_{i=1}^{T-2} \sum_{j=1}^{B} \sum_{k=1}^{J} \sum_{l=1}^{3} w_{d,k} |V_{i+1,j,k,l} - V_{i,j,k,l}| \tag{4}$$

- During training, the latent variables of the pose and shape priors are getting normalized around zero. Therefore we can use the squared mean over all entries of $J_{rot,canonical,latent}$ and $l_[bone, latent$ as the corresponding loss.

Additionally, motion priors can help not only to validate static poses and shapes, but also validate natural-looking motions over multiple frames. Based on Zhang et. al. [36], we use a convolutional autoencoder to learn the latent variables of LISST clips. As shown in Fig 1, while training we apply a smoothing loss, to ensure that motions that appear after each other have similar latent variables. We trained the autoencoder on 20.000 clips with 40 frames and normalized in orientation and position. In our LISST recovering optimizer, in each iteration, we split our sequence into 40 frames clips as well, normalize, and use the encoder part to transform the clips to the corresponding latent variables. A smoothing loss term applied to the latent variables ensures valid motions.

4. Experiments

4.1. Datasets

We used two datasets for our project: the CMU Panoptic Dataset [34] and the EgoBody Dataset [35]. Both datasets contain multiview videos of multiple persons in an indoor scene interacting with each other. With these datasets, we could therefore test the robustness of our pipeline against occlusions by other persons, as well as objects such as tables.

The CMU Panoptic dataset contains 65 sequences presenting one or multiple persons. Each sequence is recorded with 31 HD and 480 VGA cameras. In our experiments, we used up to 5 views from HD cameras per sequence. The videos from those cameras are already synchronized, and HD cameras record at a resolution of 1920x1080 at 30 FPS. We used several selected sequences from this dataset: dance, build, haggle, and pizza. The dance sequence allowed us to test the tracking of a single person performing very dynamic and complex movements. Other sequences allowed us to test pose matching and tracking of multiple persons. In particular, the pizza and build sequence enabled us to test the pipeline with a recording where multiple persons walk around in the scene and interact with objects creating many occlusions for various views.

The EgoBody Dataset contains 125 sequences of pairs of people performing various tasks and interacting with each other. We primarily used two sequences from this dataset for our experiments with 3 views per sequence.

Due to hardware limitations, we performed the 2D pose estimation for individual frames with OpenPose separately from the rest of the pipeline for our experiments. Before running the experiments, we precomputed 2D poses with hand detection for all frames. Due to memory limitations, we run OpenPose at a reduced resolution of 192x180, which might have negatively influenced its performance and the overall results of our pipeline, especially in more challenging scenes.

4.2. Evaluation

To demonstrate the validity of our approach for multi-person multi-view 3D pose extraction, we performed both visual and quantitative evaluation. Quantitatively, we evaluated the validity of our stitching algorithm and the overall accuracy of our 3D pose extraction pipeline.

Dataset	Before	After	Actual
Dance	5	1	1
Build	2	2	2
Haggle	5	3	3
Pizza	28	10	7
EgoBody1	2	2	2
Egobody2	2	2	2

Table 1. Number of tracks before and after stitching.

4.2.1 Stitching Algorithm

As the purpose of stitching was to piece together broken tracks to better match the actual number of tracks in a video, we compared the number of output tracks before and after stitching to the correct number of tracks, where each track represents the presence of one person in a video. Additional tracks are counted for every new person that enters the scene. Tab. 1 summarizes the performance of our pipeline in terms of track detection before and after stitching. For datasets Build, Egobody1, and Egobody2, the pipeline successfully detected the correct number of tracks without stitching. For datasets with fragmented tracks, our stitching algorithm was able to piece together the correct number of tracks for Dance and Build and improve the track detection for Pizza.

4.2.2 Mean Per Joint Position Error (MPJPE)

To demonstrate the correctness of the tracks pieced together through stitching and to give an overall evaluation of the accuracy of our 3D pose extraction pipeline, we calculated the MPJPE between our estimated LISST 3D poses and the ground truth 3D poses. Tab. 2 shows the average MPJPE and the deviation of MPJPEs for every track in the Build, Haggle, and Pizza datasets. Our pipeline averages around 6-7 cm of error on Build and Haggle and struggles more with Pizza, as Pizza contains more people which presents the opportunities for more occlusions and tracking confusion.

4.2.3 LISST re-projection

To visualize our LISST 3D pose accuracies, we wrote a script, that uses the camera parameters to re-project our final LISST 3D poses back onto their respective videos and compares our estimated joint positions to the original joint positions of the people. From the demo video here: Video-Link [39], one can see that the overlaid poses closely match

Dataset	Average MPJPE [cm]	Track MPJPE [cm]	Number of tracks
Build	6.98	6.22 - 7.75	2
Haggle	6.22	5.18 - 6.81	3
Pizza	13.01	5.55 - 33.30	7

Table 2. MPJPE for CMU datasets.

the movements of their respective persons in most frames for both the CMU Panoptic [34] and Egobody [35] datasets.

4.2.4 Difficulties

As mentioned previously, areas where the 3D poses do not successfully match their original humans, shown for example in Fig. 3, could originate from occlusions and OpenPose inaccuracies. For rapid or obstructed human movement, OpenPose may incorrectly identify joints or merge joints from one human to another if they are overlapping in a view. Additionally, noisy or missing joint data can also cause failures in matching and tracking, which ultimately result in missing or misplaced 3D joint positions.

The 3D poses tend to drift over time and sometimes the tracking algorithm merges poses that are detected closely together due to one or more of the previously discussed errors. The merging is necessary since otherwise too many 3D tracks are created, however the merging criterion should be adjusted so that the situation presented in Fig. 3b would not be repeated. If the final 3D output contains track fragments that have drifted too far away from their original track, then stitching may not be able to merge these fragments or may merge them to incorrect persons, which was the case with the Pizza dataset.

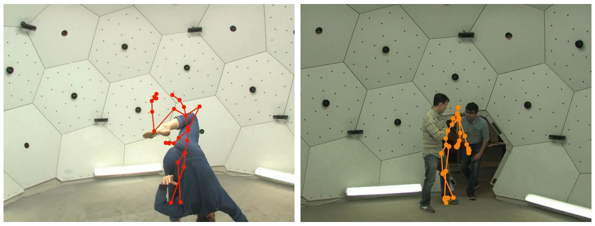
(a) (b)

Figure 3. (a) Error due to occlusion. (b) Error due to person assignment confusion.

4.2.5 Motion Prior

To evaluate the effect of the motion prior smoothening, we used visual comparison. In this video, we compare the raw 3D reconstruction to a reconstruction with motion prior applied. No other smoothing, pose- or shape priors got applied, apart from infilling of missing data: Video-Link [40] The motion prior alone already shows significant improvement of the raw data, although there still remains potential for further improvement. Training to near convergence of the motion prior took more than 24hrs on a Tesla V100 GPU. Tuning parameters of the autoencoder like clip-length, loss-weights and encoder/decoder structure did not take place due to time constraints.

4.3. LISST output

To visualize the final output of the pipeline, the sequence of LISST parameters, we use Blender [37]. A video of the of the resulting sequences of the 4 different scenes from the data-sets Egobody and Panotic is shown here: Video-Link [41]. In general, we can say that our pipeline delivers smooth and natural-looking motions that visually match the original video very accurately. Via priors and smoothening we can account for occlusion and tracking errors to a high degree. However, the pipeline has its limit when the original video contains very fast, sudden, or uncommon movements and large occlusions remain influencial.

4.4. Visual similarity matching

To assess the effectiveness of the selected methods for visual similarity matching, the first frames of the videos from two cameras (of ids 1 and 2 in the *pizza* dataset) were taken into consideration. Openpose detections transformed into bounding boxes – later used for image cropping – are presented in Fig. 4.

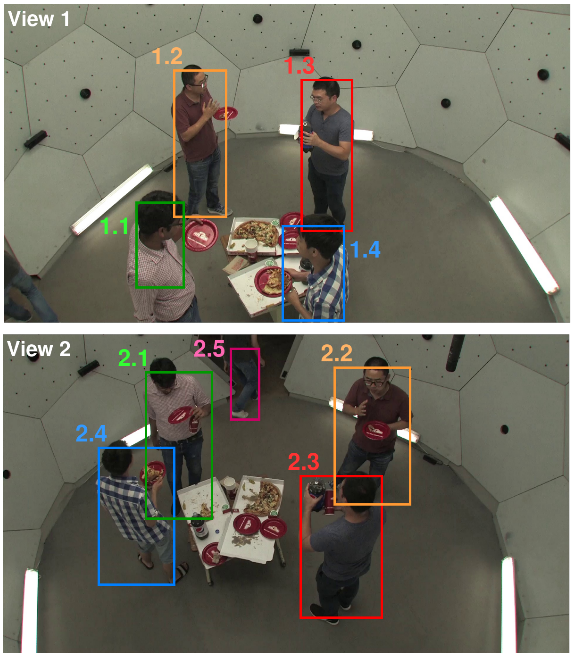

Figure 4. Two views of the same time frame selected for visual similarity matching between detected people.

	2.1	2.2	2.3	2.4	2.5
1.1	67	54	45	67	0
1.2	33	29	22	37	0
1.3	20	23	16	29	0
1.4	107	85	60	94	0

Table 3. Number of feature matches between two images with the use of ORB descriptor.

	2.1	2.2	2.3	2.4	2.5
1.1	4	2	2	2	1
1.2	4	2	4	3	4
1.3	4	8	4	4	0
1.4	20	13	7	24	11

Table 4. Number of feature matches between two images with the use of SIFT descriptor.

	2.1	2.2	2.3	2.4	2.5
1.1	75%	65%	72%	77%	75%
1.2	87%	86%	89%	83%	79%
1.3	86%	85%	91%	80%	78%
1.4	72%	66%	63%	69%	59%

Table 5. Similarity between images in accordance with the CLIP model.

Two approaches were used to match the people from different views using the appearance similarity – feature matching and similarity measure with a neural network model. The number of feature matches obtained with an ORB and SIFT descriptors are presented respectively in Tab. 3, 4. CLIP model, however, returns the similarity score in percentages, as shown in Tab. 5.

The results were not satisfactory enough to integrate any of these methods into the main pipeline. The number of matches for feature matching should be the highest for the corresponding images, and the similarity score should be maximum for the corresponding images, and these conditions are not met in the analyzed case.

The main challenge in the appearance similarity matching in this project were severe viewpoint changes. Although SIFT handles well the viewpoint changes even up to 50^o, ORB and CLIP are only partly view-point invariant. Positions and orientations of the cameras highly vary, which affects the perceived scale and illumination of people in each lens.

5. Conclusions

Summary. In this project, we developed a 3D human pose estimation pipeline that integrates with LISST. We based the tracking on OpenPose, matched key-points via mv3dpose and developed an algorithm to achieve one 3D joint track per person. We then leveraged and optimizer to create accurate, smooth and natural looking sequences of LISST parameters, that can be visualized with LISST. Our autoencoder based motion priors together with given pose- and shape-priors ensure good results even when significant occlusions are happening and are further fine-tuned by body-part-weighted velocity- and distance-smoothing functions.

Limitations and Future Work. The solution on which we based our 3D pose extraction did not always perform accurately; adjustments needed to be introduced, such as a stitching algorithm to obtain continuous tracks. There are known limitations to *mv3dpose*. For instance, the camera order highly affects the 3D pose computation, and the tracking & matching components are purely geometry-based, hence not robust to noisy data. Additionally, an exhaustive search of the optimal parameters set should be performed – it may be a matter of insufficient smoothing, too short/long minimum track length, or the poor track matching criterion.

The imperfections of the approach implemented in *mv3dpose* could be compensated by visual similarity tracking. Although the attempts presented in this project were not successful, another trial with an artificial neural network (pre-trained for the purpose of viewpoint invariance) could be performed. It could also be beneficial to account for incorrect OpenPose detections – if the poses are not correctly identified, then the person cropping can result in data loss, and therefore the methods cannot be accurate anymore.

Furthermore, OpenPose itself doesn't reach its full potential. Currently, it only tracks keypoints frame by frame and does not leverage information from other frames. Feeding back this information into a CNN used for detection or a Kalman-filter for predictable movements could potentially produce more accurate key-points from the beginning. Limited resources prevented effective tuning, which likely would have improved the outcome.

Triangulation also poses the issue of information loss. As triangulation requires at least two known 2D positions of a joint, frames containing joints that only appear in one view will be limited in the information available for person association and tracking. A possible solution is to first track people in each view over all frames and match them afterwards, keeping the 2D positions that got matched to one person. During the LISST recovering process, these can be used for loss functions via re-projecting the current LISST parameters, which would mitigate multi-camera occlusions.

6. Contributions of team members

- Emilia Szymanska – mv3dpose adjustments, stitching algorithm, visual similarity matching;
- Yuanchen Yuan – mv3dpose adjustments, stitching algorithm, pipeline evaluation;
- Johannes Gaber – LISST parameter recovering, motion prior;
- Piotr Libera – 2D pose estimation, experiments and pipeline evaluation.

References

[1] K. Rohr, Towards Model-Based Recognition of Human Movements in Image Sequences. CVGIP: Image Understanding, Volume 59, Issue 1, 1994, Pages 94-115, ISSN 1049-9660, https://doi.org/10.1006/ciun.1994.1006. 2

[2] Felzenszwalb, P. F., Huttenlocher, D. P. (2005). Pictorial structures for object recognition. International Journal of Computer Vision, 61(1), 55-79. 2

[3] Wei, S. E., Ramakrishna, V., Kanade, T., Sheikh, Y. (2016, June). Convolutional pose machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4724-4732). 2

[4] Newell, A., Yang, K., Deng, J. (2016, October). Stacked hourglass networks for human pose estimation. In European Conference on Computer Vision (pp. 483-499). Springer, Cham. 2

[5] Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., & Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4929-4937). 2

[6] Zhang, F., Zhu, X., Ye, M. (2019). Fast Human Pose Estimation. 3512-3521. 10.1109/CVPR.2019.00363. 2

[7] Cao, Z., Simon, T., Wei, S. E., Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7291-7299). 1, 2, 3

[8] A. Perez-Yus and A. Agudo, ”Matching and Recovering 3D People from Multiple Views,” 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2022, pp. 1184-1193, doi: 10.1109/WACV51458.2022.00125. 2

[9] Tanke, J. and Gall, J. Iterative Greedy Matching for 3D Human Pose Tracking from Multiple Views. German Conference on Pattern Recognition, 2019. 1, 2, 3

[10] Slembrouck, M., Luong, H., Gerlo, J., Schutte, K., Cauwelaert, D., Clercq, D., Vanwanseele, B., Veelaert, P., Philips, W. (2020). Multiview 3D Markerless Human Pose Estimation from OpenPose Skeletons. 10.1007/978-3-030-40605-9_15. 2

[11] Dong, J., Jiang, W., Huang, Q., Bao, H., Zhou, X. (2019). Fast and Robust Multi-Person 3D Pose Estimation from Multiple Views. 2

[12] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L. (2008). Speeded-Up Robust Features (SURF). Computer vision and image understanding, 110(3), 346-359. 2

[13] Calonder, M., Lepetit, V., Strecha, C., Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. Eur. Conf. Comput. Vis.. 6314. 778-792. 10.1007/978-3-642-15561-1_56. 2

[14] Viswanathan, D. (2011). Features from Accelerated Segment Test (FAST). 2

[15] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R. (1994). Signature verification using a ”Siamese” time delay neural network. In Advances in neural information processing systems (pp. 737-744). 2

[16] DeTone, D., Malisiewicz, T., Rabinovich, A. (2016). Deep image homography estimation. arXiv preprint arXiv:1606.03798. 2

[17] Zagoruyko, S., Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4353-4361). 2

[18] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, ”ORB: An efficient alternative to SIFT or SURF,” 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544. 2, 4

[19] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94 2, 4

[20] Radford, Alec and Kim, Jong Wook and Hallacy, Chris and Ramesh, Aditya and Goh, Gabriel and Agarwal, Sandhini and Sastry, Girish and Askell, Amanda and Mishkin, Pamela and Clark, Jack and Krueger, Gretchen and Sutskever, Ilya. Learning Transferable Visual Models From Natural Language Supervision. Proceedings of the 38th International Conference on Machine Learning, 2021, vol. 139, pp. 8748–8763. 4

[21] Xiang, Y., Alahi, A., and Savarese, S. Learning to track: Online multi-object tracking by decision making. ICCV 2015. 2

[22] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. 2015. GMMCP Tracker: Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking. In CVPR. 2

[23] Bo Yang and Ram Nevatia. 2012. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In CVPR. 2

[24] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. 2019. Spatial-temporal relation networks for multi-object tracking. In ICCV 2, 3

[25] Yonatan Tariku Tesfaye, Eyasu Zemene, Andrea Prati, Marcello Pelillo, and Mubarak Shah. 2019. Multi-target tracking in multiple non-overlapping cameras using fast-constrained dominant sets. IJCV 127, 9 (2019), 1303–1320. 3

[26] Ergys Ristani and Carlo Tomasi. 2018. Features for Multi-Target Multi-Camera Tracking and Re-Identification. In CVPR

[27] Y. Cai, K. Huang, and T. Tan, “Human appearance matching across multiple non-overlapping cameras,” in International Conference on Pattern Recognition (ICPR), 2008, pp. 1–4. 2

[28] Y. Wang, S. Velipasalar, and M. C. Gursoy, “Distributed widearea multi-object tracking with non-overlapping camera views,” Multimedia Tools and Applications, vol. 73, no. 1, pp. 7–39, 2014. 2

[29] S. Khan and M. Shah, “Consistent labeling of tracked objects in 13 multiple cameras with overlapping fields of view,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 25, no. 10, pp. 1355–1360, 2003 2

[30] B. Moller, T. Pl ¨ otz, and G. A. Fink, “Calibration-free camera hand-over for fast and reliable person tracking in multi-camera setups,” in International Conference on Pattern Recognition (ICPR), 2008, pp. 1–4. 2

[31] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Songchun Zhu. 2016. Multi-View People Tracking via Hierarchical Trajectory Composition. In CVPR. 3

[32] Yiyang Gan, Ruize Han, Liqiang Yin, Wei Feng, and Song Wang. 2021. Selfsupervised Multi-view Multi-Human Association and Tracking. In Proceedings of the 29th ACM International Conference on Multimedia (MM ’21), October 20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 9 pages 3

[33] Stelian Coros, Siyu Tang. 2023. Digital Humans - Data-driven inverse kinematics for human motion capture. https://github.com/Digital-Humans-23/a3 2, 3, 5

[34] Joo, Hanbyul and Simon, Tomas and Li, Xulong and Liu, Hao and Tan, Lei and Gui, Lin and Banerjee, Sean and Godisart, Timothy Scott and Nabbe, Bart and Matthews, Iain and Kanade, Takeo and Nobuhara, Shohei and Sheikh, Yaser. ”Panoptic Studio: A Massively Multiview System for Social Interaction Capture,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. 6, 7

[35] Zhang, Siwei and Ma, Qianli and Zhang, Yan and Qian, Zhiyin and Kwon, Taein and Pollefeys, Marc and Bogo, Federica and Tang, Siyu. ”EgoBody: Human Body Shape and Motion of Interacting People from Head-Mounted Devices,” European conference on computer vision (ECCV), October 2022. 6, 7

[36] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys and Siyu Tang. 2021. Learning Motion Priors for 4D Human Body Capture in 3D Scenes. https://arxiv.org/pdf/2108.10399.pdf 3, 5

[37] https://www.blender.org/ 7

[38] Xin Chen, Zhuo Su, Lingbo Yang, Pei Cheng, Lan Xu, Bin Fu, and Gang Yu. 2022. Learning Variational Motion Prior for Video-based Motion Capture. https://arxiv.org/pdf/2210.15134.pdf 3

[39] https://youtu.be/66z3thohJAM 6

[40] https://youtu.be/dmSxEZZvpp0 7

[41] https://youtu.be/j7S1wKx4u4s 7