Assignment #4

Prof. Hanbyul Joo
M1522.001000, Computer Vision

Assigned: June 2, 2025
Due: June 15, 2025, 11:59 PM

0 Instruction

In this assignment, you will implement 3D human keypoint reconstruction from multi-view images.

- **Submission Platform**: All homework must be submitted electronically on eTL.
- **Collaboration Policy**: Discussions with peers are encouraged, but you must solve the problems and write up the solutions independently.
- **Individual Assignment**: Each student is required to submit their own individual report and code.
- **Plagiarism Policy**: Do **not** copy code or reports from others. Any form of plagiarism may result in a score of zero.
- **Coding Requirements**:
  - Use **Python 3** for all programming tasks.
  - Use Colab to get support from TA regarding installation issues.
- **Reporting**:
  - Answers must be **clear, unambiguous**, and **supported by experimental results** (e.g., images, plots, brief quantitative analysis).
  - Only PDF submissions compiled using LaTeX (e.g., via Overleaf or other LaTeX tools) will be accepted. No specific template will be provided.
  - The maximum length of the report is 4 pages.
  - We recommend you to use screen captures from the provided visualizer to report your implementation results.
- **Notebook Submission**:
  - You must also submit the **Jupyter Notebook (.ipynb)** file.
  - Adding new cells is allowed.
  - Make sure that **all outputs are preserved**.
  - All images in the report should be **reproducible** from the notebook.
  - TAs will primarily check if your reported result is **fully reproducible** using your submitted notebook.
- **Questions**: We will only respond to questions posted on the eTL Q&A board. While TAs will do their best to respond promptly, replies may be delayed due to conference schedules. Please make sure to start the assignment as early as possible.

- **Structure**: Submit the zip file as the following folder and file structure.

```
{YOUR_NAME}_{YOUR_STUDENT_ID_NUMBER}.zip
├─ {YOUR_NAME}_{YOUR_STUDENT_ID_NUMBER}.pdf (your report)
├─ {YOUR_NAME}_{YOUR_STUDENT_ID_NUMBER}_HW4.ipynb (colab notebook)
├─ utils.py
└─ visualizer
   └─ panoptic_visualizer.py
```

Figure 1: 2D pose detections and score maps. (Column 1) Example views out of 480 views of Panoptic Studio with proposals by the pose detector (Column 2-7) Heat maps for each node on each view. Note that the body pose detector distinguishes left-right limbs. These are visualization from Panoptic Studio paper for your better understanding, and different from provided test dataset.

1 Preliminary: Human Keypoint Estimation

A human keypoint refers to a specific, anatomically meaningful location on the human body that is used to represent body posture or movement. In computer vision and pose estimation tasks, keypoints are used to describe the position of body parts such as:

- Joints: e.g., elbows, knees, wrists, shoulders, hips, ankles
- Facial landmarks: e.g., eyes, nose, mouth corners
- Body extremities: e.g., fingertips, toes

Each keypoint is typically represented as a 2D coordinate (x, y) in an image, or 3D (x, y, z) in a 3D space, and the full set of keypoints gives a structured representation of a person's pose.

Human pose estimation is the task of detecting and localizing human keypoints from images or videos, to reconstruct the pose of a person. Many human pose estimation methods use the concept of heatmap in their algorithms. A heatmap, is a 2D spatial map with the same or downsampled shape as the original image. Each pixel of the heatmap represents the confidence that a particular joint is located at that position. At training time, the ground truth keypoint is converted into gaussian blob, centered on the keypoint's location with high values (near 1) at the center and tapering off toward the edges, creating a heatmap. The model is trained to predict this heatmap from the original image. At inference time, model outputs heatmap with each pixel containing the confidence score. The pixel with the highest value of the confidence score is estimated to be the keypoint location.

In this homework, we will use 2D human pose estimation to generate 2D human body keypoints from image, and use the keypoints in multiple view to generate human keypoints in 3D. We will use data from multi-view capture system, and use the calibrated extrinsic and intrinsic parameter of each camera.

2 Single Person 3D Keypoint Reconstruction (20 Points)

2.1 Two-view Triangulation (10 Points).

We will implement triangulation with two views. Human pose estimation outputs the 2D coordinates and the confidence score of the keypoints in each image. You will use these outputs along with the corresponding camera matrices to triangulate 3D human keypoints. For each keypoint, check the confidence score and proceed to triangulation if the scores in both views are above 0.3. If not, skip the triangulation for that keypoint.

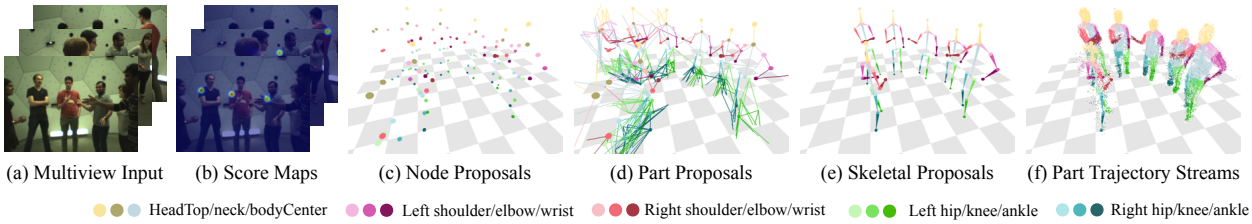

Figure 2: Several levels of proposals generated by our method. (a) Images from upto 36 views. (originally 480 views) (b) Per-joint detection score maps. (c) Node proposals generated after non-maxima suppression. (d) Part proposals by connecting a pair of node proposals. (e) Skeletal proposals generated by piecing together part proposals. (f) Labeled 3D patch trajectory stream showing associations with each part trajectory. In (c-f), color means joint or part labels shown below the figure. Our scope is reproducing results shown in (c-e).

2.2 N-view Triangulation (10 Points).

Using a similar approach as above, implement triangulation with N views. Inputs are the N camera matrices and N outputs of human pose estimation with human keypoints and confidence scores. For each keypoint, check the confidence score of the keypoint in each view, and select the views with score higher than 0.3. Use only the selected views to triangulate. If there exists no more than 2 view that satisfy the condition, skip the triangulation for that keypoint.

3 Two person 3D Keypoint Reconstruction (15 Points)

3.1 Brute Approach on Two-view (10 Points).

We will implement triangulation of two people's keypoint in two views. Human pose estimation does not know the identity of each person in the two views, so the correspondence of each keypoint in the two views is not known. We will try the naive approach, by calculating the reprojection error for all possible combination of the correspondence. Reprojection error is calculated via the Euclidean distance between the original human pose estimation output's 2D coordinate and the reprojected point. We will use the average of the reprojection of keypoint in each view, and average it again on each keypoint, and again with the two people. In this case, we fix the numbering of each person in the first view, and then the possible number combination of the correspondence in the second image is 2. Out of the two possible combination, choose the one with the lower reprojection error. Same as in the single person triangulation, check to see if the confidence score is above 0.3 in both views for each keypoint.

3.2 Brute Approach on N-view (5 Points).

We increase the view to N views. Since there are two people and N views, possible number of combination are 2^{N-1}. With large number of camera, the time to calculate the reprojection error on all possiblities becomes extensively long. Instead of running through all the possible combination, we randomly sample 10000 cases and report the combination with the lowest reprojection error as the final output. For each keypoint, as in the single person triangulation, use only the views with confidence score above 0.3 to triangulate the point. Use the same view again to calculate the reprojection error too.

4 Multi Person 3D Keypoint Reconstruction (65 Points)

Preliminary: Panoptic Studio

In the remaining section of this assignment, we turn our attention to building a more robust pipeline for multiview 3D human pose reconstruction. In the first half of the homework, we observed that naive triangulation approaches often struggle in multi-person scenarios. For example, simply grouping 2D detections from each view and triangulating them directly can lead to incorrect associations, especially when joints from

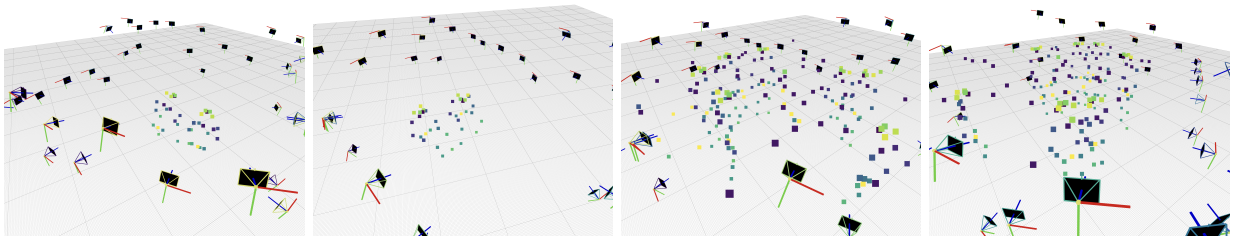

Figure 3: Expected output of node proposals in our attached viewer. First and seconds are from *two_person_1* dataset and others are from *labphoto_3* dataset.

different people are mistakenly combined. We also experimented with brute-force matching—evaluating all possible combinations of 2D detections across views and selecting the best-matching group per person. While conceptually simple, this method quickly becomes computationally infeasible as the number of views and people increases, and it is sensitive to spurious or missing detections. These challenges highlight the difficulty of enforcing consistent global assignments when relying purely on view-wise detections.

To address these limitations, we will reproduce the bottom-up, proposal-based 3D skeletal reconstruction method proposed in the **Panoptic Studio**, illustrated in Fig. 2. Rather than grouping detections person-by-person, this method first fuses 2D joint heatmaps from all views into 3D voxel grids, generating node score maps for each joint type. Local maxima in these score maps are extracted via 3D Non-Maximum Suppression to obtain 3D joint candidates (node proposals) (Sec. 4.1). Next, part proposals—pairs of node candidates representing bones—are scored by projecting them into all camera views and checking how consistently the two endpoints appear together in 2D detections of the same person (Sec. 4.2). Using these node and part proposals, the algorithm then assembles full 3D skeletons by applying dynamic programming over a predefined kinematic tree structure. It selects the best-scoring combination of parts to form a plausible skeleton and repeats this process greedily to extract multiple individuals (Sec. 4.3). Additionally, the final joint positions are refined by minimizing reprojection error using these correspondences (Sec. 4.4).

The original Panoptic Studio system extends this pipeline to track identities across time for full 4D reconstruction, while we focus only on the **per-timestep (static) reconstruction** stage in this assignment.

4.1 Node Proposals: 3D Joint Voting (20 Points)

As aforementioned, we provide heatmap and keypoint of detected person in each view. Each 2D skeleton detection i in a camera view c is denoted by $\mathbf{s}_i^c \in \mathbb{R}^{19\times 2}$, and is composed of 19 anatomical landmarks or *nodes*, also referred to as joints[1]. The position of the j-th node of the i-th person detection is denoted by $\mathbf{s}_{ij}^c \in \mathbb{R}^2$. We denote the heatmap(score map) representing the per-pixel detection confidence pf each node s_{ij}^c as $h_{ij}^c(\mathbf{z}) \in [0,1]$, where $\mathbf{z} \in \mathbb{R}^2$ indexes 2D image space. We also compute a merged score map by taking the maximum across all person detections at each pixel, $h_j^c(\mathbf{z}) = \max_i h_{ij}^c(\mathbf{z})$. Merged score maps of example views are shown in Fig. 1.

To combine 2D node score maps from multiple views into 3D, we generate a 3D score map for each node using a spatial voting method. We first index the 3D working space into a voxel grid (2cm), and compute the *node-likelihood* score of each voxel by projecting the center of the voxel to all views and taking the average of the 2D scores at the projected locations. The 3D score map $H_j(\mathbf{Z})$ for a node j at the 3D position $\mathbf{Z} \in \mathbb{R}^3$ is defined as

$$H_j(\mathbf{Z}) = \frac{1}{|V(\mathbf{Z})|} \sum_{c \in V(\mathbf{Z})} h_j^c\left(\mathcal{P}_c(\mathbf{Z})\right), \tag{1}$$

where $\mathcal{P}_c(\cdot) \in \mathbb{R}^2$ denotes projection into camera c, $V(\mathbf{Z})$ is the set of cameras where the 3D location $\mathbf{Z}$

[1]We modify the skeleton hierarchy of COCO to have an explicit torso bone, by taking the center of the two hip nodes as a body center node (19th joint).

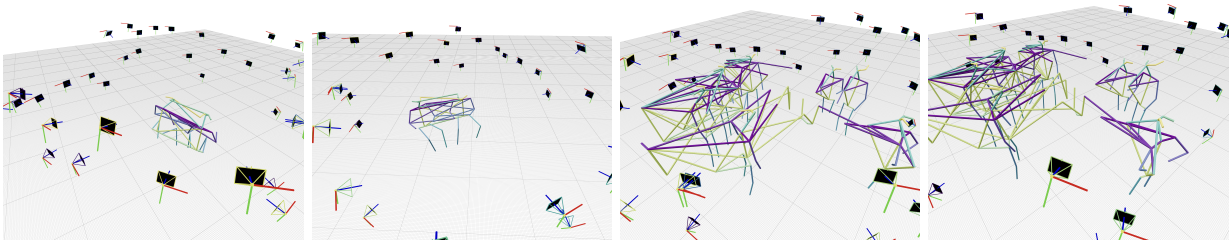

Figure 4: Expected output of part proposals in our attached viewer. First and seconds are from *two_person_1* dataset and others are from *labphoto_3* dataset.

is visible, and $|V(\mathbf{Z})|$ is the cardinality of the set. Note that the 3D score map for each node is computed separately, producing fifteen 3D score maps at each time instant.

From the 3D score map for each node at each time instance, we perform Non-Maxima Suppression (NMS), and keep all the candidates above a fixed threshold τ (we use τ=0.09 here.). We provide basic NMS function, but feel free to change it. The expected results are shown in the Figure 3. Each node proposal, denoted as $\mathbf{N}_j^k$ for the k-th proposal for node j, is a putative candidate for the j-th anatomical landmark of a participant.

4.2 Part Proposals: 3D Bone Voting (20 Points)

Given the generated node proposals, we infer part proposals by estimating connectivity between each pair of nodes that make up a possible body part. The 2D Human Keypoint Estimator uses appearance information during the inference, and, thus, the result tends to preserve connectivity information (e.g., left knee is connected to the left foot of the same person). Our approach fuses them by voting 2D connectivity into 3D. More specifically, we define a connectivity score between a pair of node proposals by projecting them onto all views, and checking in how many views they are actually connected, i.e., both nodes belong to the same person detection. Formally, the connectivity score of a part $\mathbf{P}_{uv}^k$ between two node proposals $(\mathbf{N}_u^{k_u}, \mathbf{N}_v^{k_v})$, where $(u, v) \in \mathbf{B}$, is defined as

$$\Phi(\mathbf{P}_{uv}^k) = \frac{1}{|V(\mathbf{P}_{uv}^k)|} \sum_{c \in V(\mathbf{P}_{uv}^k)} \max_i \phi_{iuv}^c \left(\mathcal{P}_c(\mathbf{N}_u^{k_u}), \mathcal{P}_c(\mathbf{N}_v^{k_v})\right),$$

$$\phi_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v) = w_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v) \delta_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v)$$

where

$$w_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v) = \frac{1}{2}\left(h_{iu}^c(\mathbf{z}_u) + h_{iv}^c(\mathbf{z}_v)\right), \text{and}$$

$$\delta_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v) = \begin{cases} 1 & \text{if } h_{iu}^c(\mathbf{z}_u) > \tau \text{ and } h_{iv}^c(\mathbf{z}_v) > \tau \\ 0 & \text{otherwise.} \end{cases}$$

Here, $\mathcal{P}_c(\mathbf{N}_u^{k_u})$ and $\mathcal{P}_c(\mathbf{N}_v^{k_v})$ are the projections of the two nodes of $\mathbf{P}_{uv}^k$ in view c, and $V(\mathbf{P}_{uv}^k)$ is the set of cameras where the 3D part is visible. Intuitively, the part score Φ represents the average connectivity score across all views from all potentially corresponding 2D person detections. Because we do not know the correspondence from 3D parts to 2D person detections, we take the maximum score across all possible detections i in each view. Assuming that the projected part corresponds to the i-th person detection in camera c, the part connectivity score ϕ_{iuv}^c is defined as the average score of the projected nodes, denoted by $w_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v)$. The delta function δ_{iuv}^c additionally ensures that ϕ_{iuv}^c is nonzero only if both projected node locations have a sufficiently high score for the same detection i (i.e., both nodes are detected as part of a single person). The expected results are shown in Figure 4.

4.3 Skeletal Proposals: Using Dynamic Programming (20 Points)

Now, we will generate skeleton proposals by piecing together the part proposals. Since each skeleton is a tree structure, this can be computed efficiently using Dynamic Programming (DP)—but only for a single

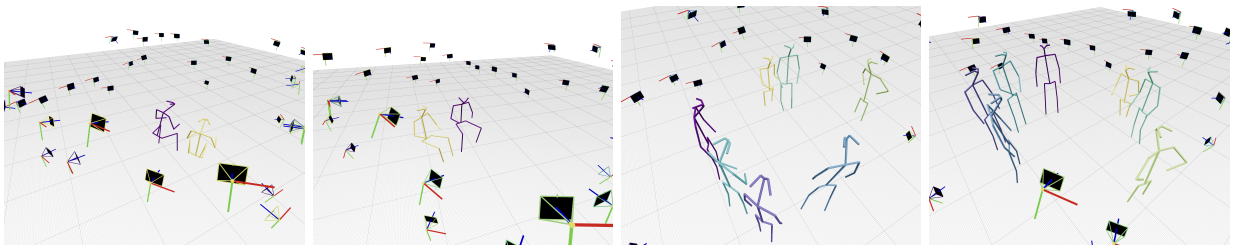

Figure 5: Expected output of skeletal proposals in our attached viewer. First and seconds are from *two_person_1* dataset and others are from *labphoto_3* dataset.

person. Therefore, we use DP to greedily find 3D skeletons $\mathbf{S}^k$ which maximize the sum of part scores,

$$\Theta(\mathbf{S}^k) = \max_{(k_1,\cdots,k_J)} \sum_{(u,v)\in\mathbf{B}} \Phi\left(\mathbf{P}^k_{uv}\right).$$

A skeleton $\mathbf{S}^k$ is given by the mapping $k \mapsto (k_1, \cdots, k_J)$, where the J-tuple $(k_1, \cdots, k_J)$ determines the assignment of node proposals $\mathbf{N}_j^{k_j}$ for each joint j in the body. After picking the highest scoring skeleton $\Theta(\mathbf{S}^k)$, the assigned nodes $(k_1, \cdots, k_J)$ are removed from the pool of available node proposals and we run DP again to find the next highest scoring skeleton, and so on until all possible skeletons are found.

One option here would be to threshold the skeleton scores $\Theta(\mathbf{S}^k)$ at some minimum value to determine valid detections. However, we can do better: each 3D skeleton should be supported by 2D detections, and each 2D detection can correspond to only a single 3D skeleton. Specifically, we place each 3D node $\mathbf{N}_j^k$ in skeleton $\mathbf{S}^k$ in correspondence with the closest 2D joint detection in each view. For each 3D node $\mathbf{N}_j^k$, we create a set of correspondences $\mathcal{C}_j^k$ with elements (c, i) such that the distance $\|\mathcal{P}_c(\mathbf{N}_j^k) - \mathbf{s}_{ij}^c\|_2$ is the minimum across all detections i in view c and smaller than δ=10px. Once a 2D correspondence is established, we remove it from the set of available 2D detections, and, as above, this is performed greedily in order of decreasing skeleton score $\Theta(\mathbf{S}^k)$. Skeletons where the head node has fewer than two correspondences are discarded, i.e., if $|\mathcal{C}_j^k| < 2$ for j the head. The expected results are shown in Figure 5.

4.4 Skeletal Proposals: Refining Skeletons (5 Points)

We additionally use the set of correspondences $\mathcal{C}_j^k$ to refine the 3D node locations by minimizing reprojection error. This overcomes the discretization error introduced by the voxel grid resolution. The final 3D node location $\hat{\mathbf{N}}_j^k$ is then

$$\hat{\mathbf{N}}_j^k = \arg\min_{\mathbf{Z}} \sum_{(c,i)\in\mathcal{C}_j^k} \left\|\mathcal{P}_c(\mathbf{Z}) - \mathbf{s}_{ij}^c\right\|_2.$$

You may use any optimizer (e.g., Adam or SGD) with any settings of your choice. However, please ensure that the optimization does not take an excessive amount of time (e.g., more than 15 minutes), as this step is intended to be a lightweight refinement. For reference, we show some expected outputs from the provided data in Fig. 6.

References

- Joo, Hanbyul, et al. "Panoptic studio: A massively multiview system for social motion capture." Proceedings of the IEEE international conference on computer vision. 2015.

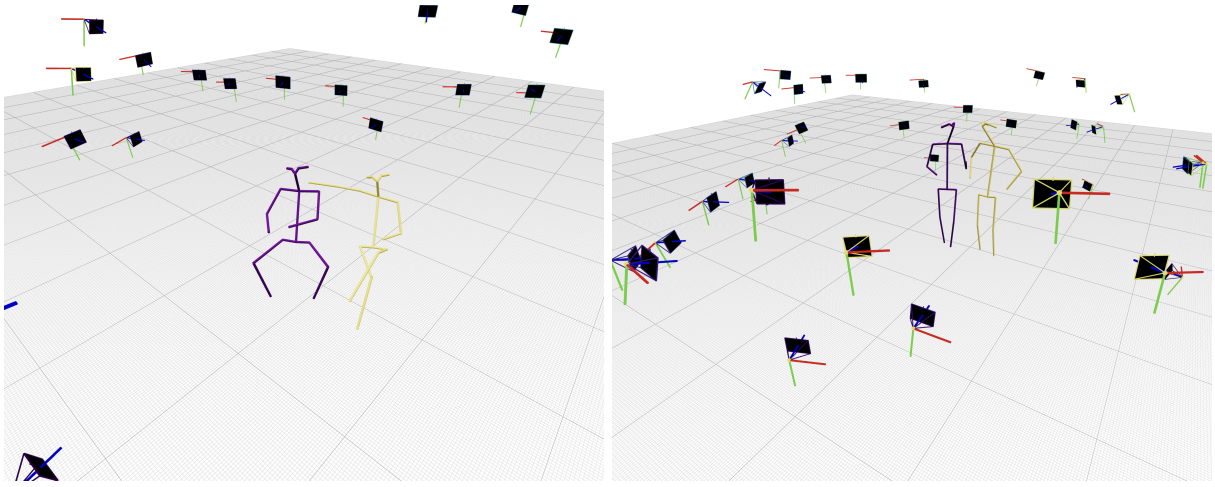

Figure 6: Expected skeleton output in our attached viewer. First and seconds are from *two_person_3* dataset and *two_person_4* dataset respectively.