2025 IEEE International Conference on Consumer Electronics (ICCE)

Human Pose-Guided Joint Optimization for Camera Auto-Calibration System with Unknown Intrinsics

Nathaniel Rensly, Zheng-Kai Chen, Hsuan-Cheng Chu, and Huang-Chia Shih
Department of Electrical Engineering, Yuan Ze University, Taiwan
$\{$*s1103749; s1103742; s1103725*$\}$*@mail.yzu.edu.tw hcshih@saturn.yzu.edu.tw*

***Abstract*—Camera calibration is pivotal in computer vision and image processing, which involves acquiring internal and external camera parameters. However, the currently widely used calibration techniques often require calibrators, proving to be cumbersome and time-intensive. To address this challenge, we introduce a high-precision, multi-camera relative pose estimation approach that uses human joints as reference points. Traditional calibration methods typically rely on checkerboards or triangular patterns as reference coordinates, which limits their effectiveness in environments where such patterns are unavailable. Our method eliminates the need for specialized calibration tools, requiring only a person to stand in the scene for camera calibration. Furthermore, we replace conventional feature-matching techniques with a novel neural network, SuperGlue, which utilizes graph-based and attention-based mechanisms to improve accuracy and speed. This approach significantly reduces the time needed for calibration compared to Zhang's method.**

***Index Terms*—Pose Estimation, Camera Calibration, Feature Matching, Multi-Camera Systems**

I. Introduction

Camera calibration is a critical process in multi-camera systems, as it enables the accurate estimation of both intrinsic parameters, such as focal length and principal point, and extrinsic parameters, including rotation and translation. In dynamic environments, recalibration is often necessary when camera positions are altered, particularly for the extrinsic parameters. The accuracy of this process largely depends on the ability to detect and match common points across camera views. However, limited overlapping fields of view in multi-camera setups complicate the detection and alignment of these points, presenting a significant challenge in calibration.

Traditional calibration methods, such as checkerboard-based approach proposed by Zhang [1] and radial distortion correction presented by Tsai [2], are widely adopted but are constrained by several factors. Zhang's method requires a visible calibration pattern, limiting its applicability in uncontrolled environments, while Tsai's method addresses only radial distortion, making it less effective in scenarios with severe distortion. Moreover, both methods depend on predefined calibration patterns, limiting their flexibility in diverse settings.

To address these limitations, a wide-baseline multi-camera calibration method based on pose estimation has been proposed. This approach leverages human joints as point correspondences across multiple camera views, thereby eliminating the need for traditional calibration patterns.

Our main contributions are as follows: First, during the pose estimation model selection phase, we evaluated MediaPipe, OpenPose, and HigherHRNet [3] on the COCO dataset, calculating results for each method and selecting the model with the highest accuracy based on this comparative analysis. Second, in the matching stage, we assessed SuperGlue [4] and Disk [5], conducting calculations and evaluations to select the descriptor with higher dimensionality for improved precision. Third, in the optimization stage, we enhanced the original method—initially based on a single optimizer—by integrating a custom-developed loss function and optimization strategy. Finally, we utilized PyTorch CUDA to resolve performance bottlenecks in feature matching, particularly in the SuperGlue algorithm and calibration computations. This enabled iterative training, leading to progressively refined and optimized parameter values.

II. Related Work

Camera self-calibration methods use key points from various perspectives to estimate parameters, with techniques such as two-step approach for improved robustness [6], and the use of the absolute conic for parameter derivation from multiple planes [7]. Other methods such as transformation-based approach [8], pedestrian observation technique [9], and minor rotation algorithm [10], demonstrate versatility but often require well-structured scenes and remain sensitive to environmental changes.

Calibrator-based methods typically achieve higher precision by using structured patterns, such as Zhang's checkerboard technique [1], Abdel-Aziz *et al.* proposed a linear system approach [11], and Tsai *et al.* presented a method with radial distortion correction [2], though these can be computationally demanding. In multi-camera setups, methods like Tang's cube chessboard calibration [12] and Shen's sphere-based approach [13] show effectiveness but require customized patterns and precise fitting. Point cloud techniques, such as those by Huang [14] and Liang [15], extend applications to complex surfaces but assume a wide shared field of view, limiting their applicability.

Our method overcomes these limitations by utilizing the flexibility of the human body, reducing dependence on structured patterns and broad shared fields of view.

III. Methodology

The 2D pose estimation pipeline starts with image capture using two synchronized cameras. The subject stands at a distance that allows both cameras to capture their entire body

979-8-3315-2116-5

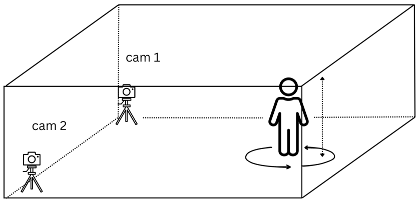

Fig. 1: Setup: Subject's whole body is within the field of view of the cameras

as shown in Fig. 1. They perform a slow, controlled spin with arms extended to ensure full body coverage within the shared camera view.

This rotational movement helps capture a wide range of body angles. Collecting more images improves the accuracy of camera calibration by enabling the selection of high-confidence image pairs, which, in turn, leads to more precise calculations of the camera parameters. However, increasing the number of images also adds computational complexity, as more data needs to be processed and analyzed. In the pose detection phase, images are processed using HigherHRNet, a 2D pose detector that identifies key body joints without camera calibration. To address reduced accuracy from self-occlusion, only joints visible to both cameras and exceeding a confidence threshold are selected for further processing.

A. Mathematical Model of Camera Calibration

In this section, we describe the fundamental mathematical models and algorithms that underlie our camera calibration and pose estimation system. The key components involve both intrinsic and extrinsic camera parameters, as well as reprojection error minimization techniques that have been adapted from previous works, such as Zhang's method [1] as well as Liu et al.'s auto calibration method [16].

The camera is modeled using the *pinhole camera model*, where the relationship between a 3D world point (X, Y, Z) and its corresponding 2D image point (x, y) is represented through projection. This projection is governed by the intrinsic and extrinsic parameters of the camera in the form of the projection matrix (P). The intrinsic parameters (e.g., focal length, principal point) are related to the internal characteristics of the camera, while the extrinsic parameters (e.g., rotation, translation) represent the camera's orientation and position in the world coordinate system.

The relationship from the 3D world coordinates (X, Y, Z) to the 2D image point (x, y) is defined by the following equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{1}$$

where R is the 3×3 rotation matrix, t is the 3×1 translation vector, and K is the intrinsic parameter matrix, as defined in Equation (2). The rotation matrix and translation vector together represent the extrinsic parameters, while the intrinsic matrix defines the camera's internal characteristics. These parameters describe the camera's orientation and position relative to the world.

$$K = \begin{bmatrix} f_x & s & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where f_x and f_y denote the focal lengths, s is the skew, and (C_x, C_y) is the principal point of the image. However, considering how modern cameras have well-aligned pixels, the skew factor is almost zero and can be ignored.

$$Z_{\text{cam}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix}. \tag{3}$$

The relationship between the 3D camera coordinates$(X_{\text{cam}}, Y_{\text{cam}}, Z_{\text{cam}})$ and the 2D image coordinates (x, y) is then described by the projection equation above.

B. Human Joint Detection

Efficiently acquiring stable keypoints in an image is crucial for reliable feature matching. Early in our research, we experimented with MediaPipe, a framework that detects around 33 human keypoints. While it provided more keypoints, we found that focusing on a smaller set of joints simplified the process and improved speed, aligning better with our goal of fast and efficient camera calibration.

In this work, we use the HigherHRNet model for joint detection. Assume p_i denotes a specific joint, with its 2D confidence scores represented as $c_i^{(0)}$ and $c_i^{(1)}$ for camera 0 and camera 1, respectively. Define a confidence threshold C, such that the condition for selecting p_i as the optimized target point is:

$$c_i^{(0)} \geq C \quad \text{and} \quad c_i^{(1)} \geq C. \tag{4}$$

After applying this threshold, a set of joint pairs with high confidence scores from both cameras can be identified for further processing.

C. Keypoints Feature Matching

In human joint detection tasks, accurately matching joint keypoints across multiple views is critical for solving geometric problems such as estimating the fundamental matrix. The fundamental matrix, which describes the relationship between corresponding points in two images, requires multiple sets of homonymous points to be identified between the image pairs. To achieve this, we utilize SuperGlue [17], a state-of-the-art deep learning-based feature matching algorithm, to establish reliable correspondences.

SuperGlue's architecture consists of three main components: the input layer, the GNN, and the optimal matching layer, which are obtained from the SuperPoint descriptor network. SuperGlue uses 256-dimensional descriptors, which provide richer information for matching tasks. In comparison, we experimented with DISK's feature matching algorithm [5], but its descriptors are only 128-dimensional, making SuperGlue more effective in matching accuracy due to its larger descriptor size.

After detecting the high-confidence keypoints using HigherHRNet, we match the joint keypoints with feature points extracted by SuperGlue. While SuperGlue is typically used for general scene feature matching, we adapt it to match the spatial locations of human joints. By identifying feature points from SuperGlue that are spatially close to the detected joints from HigherHRNet through descriptors, we create a subset of feature points that share similar positions with the joint keypoints. This relationship is described by Equation (5), in which the difference between the HigherHRNet-detected joint k^i and the descriptor keypoint k_d^i cannot exceed a tolerance range $\mathcal{T}$.

$$\left|k^i - k_d^i\right| \leq \mathcal{T}. \tag{5}$$

D. Camera Calibration

The camera calibration process in our approach involves a sequence of steps designed to accurately estimate the camera's intrinsic and extrinsic parameters. This process starts with collecting the joints and proceeds through a series of computations and optimizations to refine the camera models.

Calculating Fundamental and Essential Matrix: Using the 8-point algorithm, we compute the fundamental matrix F. This matrix captures the geometric relationship between the camera views. However, because of the scale invariance of the fundamental matrix F, we need at least 8 pairs of homonymous points to be taken and linearly transformed into a system of homogeneous linear equations. As shown in Equation (6).

$$P_1 \cdot F \cdot P_2^T = 0, \tag{6}$$

where P_1 and P_2 represent the (x, y) coordinates of the points from the first and second cameras, respectively, and F is an $(n \times 9)$ matrix. We will also leverage Singular Value Decomposition (SVD) for this process. In which SVD allows us to decompose the matrix F into three matrices U, Σ, and V^T, such that:

$$F = U \cdot \Sigma \cdot V^T. \tag{7}$$

In this decomposition, U and V are orthogonal matrices, and Σ is a diagonal matrix containing the singular values of F.

To obtain a refined estimate of the fundamental matrix F, we can enforce the rank-2 constraint, which states that the essential matrix should have only two non-zero singular values. This can be achieved by setting the smallest singular value in Σ to zero, resulting in a new matrix Σ' that maintains the structure of the original F:

$$\Sigma' = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{8}$$

we can then reconstruct the corrected fundamental matrix as:

$$F' = U \cdot \Sigma' \cdot V^T. \tag{9}$$

Whilst the essential matrix E is derived from F using the intrinsic camera matrices K_1 and K_2 as shown in Equation (10). For our method, we start with factory default K.

$$E = K_1 \cdot F \cdot K_2^T. \tag{10}$$

Decomposition and Triangulation: From the essential matrix E, we extract four possible combinations of rotation R and translation t using Singular Value Decomposition (SVD). This is done because the essential matrix encodes the relative pose (rotation and translation) between two cameras up to scale. R can be one of two rotations derived from U and V, denoted as R_1 and R_2. These rotations are computed using the skew-symmetric matrix W:

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{11}$$

and two possible rotations are given by:

$$R_1 = UWV^T, \quad R_2 = UW^TV^T, \tag{12}$$

t denotes the translation vector, which can either be the positive or negative direction of the third column of matrix U. Thus, we have four possible combinations: (R_1, t_1), (R_1, t_2), (R_2, t_1), (R_2, t_2).

After computing these four combinations, we perform triangulation on the matched points to reconstruct their 3D positions. Triangulation determines the 3D coordinates by projecting the 2D points into 3D space. By evaluating the depth information, we select the combination that yields positive depths, ensuring that the reconstructed points are in front of the cameras. Given the large number of pairs, each pair generates a corresponding set of rotation (R) and translation (t) matrices. We then perform a selection process to identify the optimal R and t that best fit the majority of the pairs by calculating the reconstruction error. The pair with the lowest error is selected for further optimization.

Optimization of Intrinsic Parameters: After obtaining initial extrinsic parameters, we optimize intrinsic parameters. We start with factory default values of K based on Equation (3) and use N pairs of reliable camera-view joints for optimization. With current intrinsic and extrinsic parameters, the 3D points of the human body are triangulated and reconstructed. Each camera's intrinsic parameters are optimized in the following steps:

1. The 3D human body joints are reprojected to the pixel coordinate system to form projection points with Equation (14).

2. We construct a loss function to optimize intrinsic parameters through the difference between the reprojected point and the 2D detected joint, as shown in the Equation below:

$$\begin{aligned} L = \frac{1}{N} \sum_{i=1}^{N} (&\|Z_{\text{cam}} \times x_i - (f_x \times X_{\text{cam}} + C_x \times Z_{\text{cam}})\| + \\ &\|Z_{\text{cam}} \times y_i - (f_y \times Y_{\text{cam}} + C_y \times Z_{\text{cam}})\|). \end{aligned} \tag{13}$$

3. Optimization: Use the Adam optimizer to minimize this loss function and refine the intrinsic parameters.

Algorithm 1 Joint Optimization

```
Inputs:
  $K1$: Intrinsic parameters of Camera 1
  $K2$: Intrinsic parameters of Camera 2
  $Rt1$: Extrinsic parameters of Camera 1 (anchor)
  $Rt2$: Extrinsic parameters of Camera 2 (to be optimized)
  Combinations: 4 possible combinations of $Rt2$
  Adam: PyTorch Adam optimizer
  Iterations: Iterations for optimization loop
Step 1: Set $Rt1$ as anchor (R = Identity, t = Zero)
  $Rt1.R \leftarrow I$ {Identity matrix}
  $Rt1.t \leftarrow 0$ {Zero vector}
Step 2: Initialize $Rt2$
  Best_$Rt2$ ← Select best combination (low error)
Step 3: Initialize optimizer (Adam) with $K1$, $K2$, $Rt2$
Step 4: Optimization Loop
for iteration in $1 \rightarrow$ MaxIterations do
    Step 4.1: Optimize both intrinsic parameters ($K1$, $K2$)
    Compute loss (reprojection error) using current $K1$, $K2$, $Rt2$
    Update $K1$, $K2$, $Rt2$ using Adam optimizer
    Step 4.2: Fix $Rt1$ (anchor), optimize only $Rt2$
    Fix $Rt1$ ($R = I$, $t = 0$)
    Compute loss (reprojection error) using fixed $Rt1$
    Optimize $K1$, $K2$, $Rt2$
    Update $Rt2$, $K1$, $K2$ using Adam optimizer
    Step 4.3: Repeat until convergence
end for
Step 5: Return optimized $K_1$, $K_2$, and $Rt2$
```

Joint Optimization: To overcome the challenge of local optima in non-linear systems, we employ a joint optimization approach using an iterative solution. This process alternates between:

1. Fixing intrinsic parameters and solving for extrinsic parameters.

2. Fixing extrinsic parameters and solving for intrinsic parameters.

However, our approach differs from previous works in that we keep the extrinsic parameters of the reference camera fixed throughout the optimization process. Specifically, for the reference (anchor) camera, which in our case is camera 1, the rotation matrix R is set as the identity matrix, and the translation vector t is the zero vector.

Using these updated camera parameters, we perform triangulation once again to reconstruct the 3D points of the human body. These new 3D points are subsequently reprojected onto each camera view to evaluate the accuracy of the calibration. The reprojection relationship is presented in Equation (14) below:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z_{\text{cam}}} K[R|t] \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \\ 1 \end{bmatrix} . \tag{14}$$

We iterate through these steps while keeping K_1 and K_2 fixed to further optimize R and t using the Adam optimizer, and vice versa. Adjustments to the parameters will ultimately reduce the Average Reprojection Error (ARE).

TABLE I: Human Pose Estimation Models

	AP.5	AP.75	AP(M)	AP(L)
HigherHRNet	0.350	0.351	**0.346**	**0.588**
MediaPipe	0.091	0.048	0.085	0.059
OpenPose	0.471	0.442	0.310	0.497

E. *Average Reprojection Error*

To evaluate the calibration accuracy, the reprojection error is calculated. This error is determined by measuring the Euclidean distance between the reprojected 2D points and the actual 2D feature points detected in the images [16]. Specifically, the error is computed using the formula:

$$E = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x'_i - x_i)^2 + (\ y'_i - y_i)^2}, \tag{15}$$

where x'_i and y'_i are the coordinates of the i^{th} joint detected by the feature detection algorithm, and x_i and y_i are the coordinates obtained from the reprojection process. A lower average error indicates a higher accuracy of the calibration.

IV. Experiments

In this section, images are taken from two distinct viewpoints to detect human joint positions, match corresponding points, and iteratively optimize extrinsic parameters.

A. *Pose Estimation Model Selection Phase*

First, we applied MediaPipe, OpenPose, and HigherHRNet on the COCO dataset, each model produces four different AP Scores: AP.5 and AP.75, which measure precision at IoU thresholds of 50 percent and 75 percent, respectively, and AP (M) and AP (L), which evaluate the model's precision performance on medium and large objects. Following the analysis in Table. 1, we selected HigherHRNet as the relatively more efficient and accurate model.

In our evaluation, we observed that OpenPose did not perform optimally on the COCO test Average Precision (AP) score. A significant factor contributing to this underperformance is that OpenPose uses a model with 33 keypoints, whereas the COCO dataset standard defines only 17 keypoints.

B. *Matching Step*

In the model matching step, we compared two models: SuperGlue and Disk. Our tests revealed that Disk operates at a faster computation speed, which can be attributed to its 128-dimensional descriptor compared to SuperGlue's 256-dimensional descriptor. The lower dimensionality of Disk means that it requires significantly less computation per feature point.

The higher dimensionality of SuperGlue allows it to detect feature points with greater precision, as it has a broader and more detailed detection range for each point. As shown in Fig. 2, when applied to the same dataset, the feature matching lines produced by Disk show low confidence and exhibit pairing errors, which are not observed with SuperGlue. Although SuperGlue requires slightly more computation time,

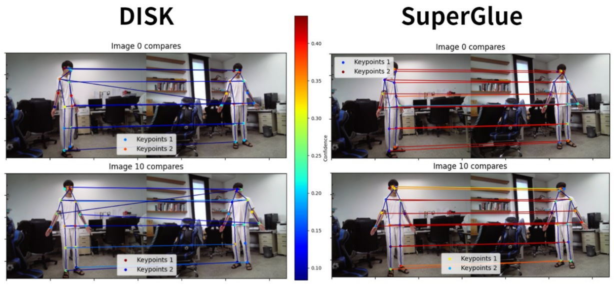

Fig. 2: Disk and SuperGlue Comparison (Red: High Confidence, Blue: Low Confidence).

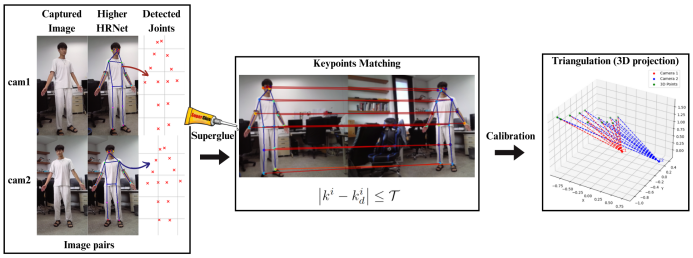

Fig. 3: Overall Calibration Pipeline including initial pose estimation, matching via feature descriptors, and optimization stages for refined calibration results.

it accurately matches each feature point while maintaining high accuracy, making it a more reliable option.

C. Optimization

Eliminating the need for calibration patterns in camera calibration, allows for a more flexible setup. Consequently, we do not require a specific environment for capturing images.

For this experiment, the cameras are placed to capture images of a human body as it enters into the scene (in our case a laboratory setting). The setup consists of two cameras placed at various angles to ensure sufficient overlap in their fields of view. This positioning allows both cameras to capture the human body simultaneously, capturing 1,000 images for each camera. This ensures that both cameras acquire corresponding calibration data.

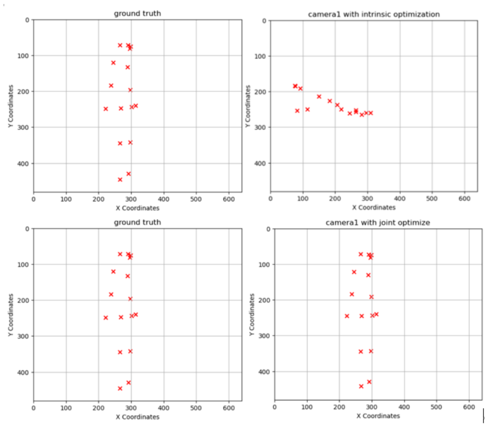

Fig. 4: Reprojection to 2D Plane: The left column shows the ground truth. The top right shows the results before optimization, bottom right are results after optimization.

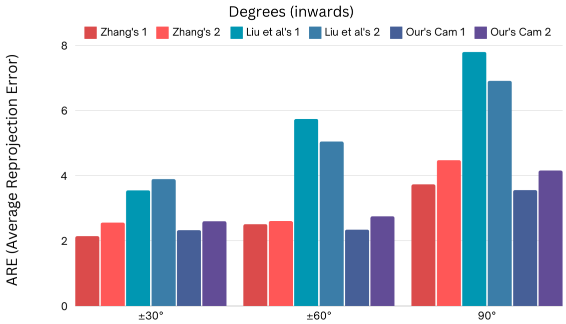

Fig. 5: Comparison of the ARE values with other standard methods

After acquiring the images, the human joint positions are detected using HigherHRNet model. Subsequently, the SuperGlue algorithm is applied to match the joints across pairs of images. The detailed arrangement of the cameras and the captured image are presented in a pipeline shown in Fig. 3.

Final Results and Analysis: As mentioned before, We set the first camera's R an identity rotation matrix (i.e., no rotation) and a translation vector of zero. Consequently, the optimization process focused solely on determining the rotation matrix $\mathbf{R}_2$ and translation vector $\mathbf{t}_2$ for the second camera, which describe the relative pose between the first camera and the second camera.

Upon completing the camera calibration, we obtained the extrinsic parameters of the second camera. Using these parameters along with the intrinsic parameters of both cameras, we performed triangulation to recover the 3D points. We then reprojected these points onto the 2D image plane for comparison with the original 2D points, allowing us to compute the reprojection error, as illustrated in Fig. 4.

Fig. 5 presents a bar chart comparing Zhang's method, Liu et al.'s method from three different angles with our proposed method. The results indicate that the differences between our's and Zhang's method are minimal, with the Average Reprojection Error (ARE) differences within 0.5. However, our method is comparatively simpler as it does not require additional calibration using a checkerboard pattern. As with Liu et al's we observe a noticeable difference with the ARE.

After joint optimization, we obtained a reprojected 2D image with a low reprojection error. With this improved accuracy, we proceeded with the 3D reconstruction. The results are illustrated in Fig. 6, where the left panel shows the human joints detected by the HigherHRNet model, and the right panel presents the corresponding 3D reconstruction of the human body.

As demonstrated, we did not include the human face in our 3D reconstruction process. By focusing on a limited set

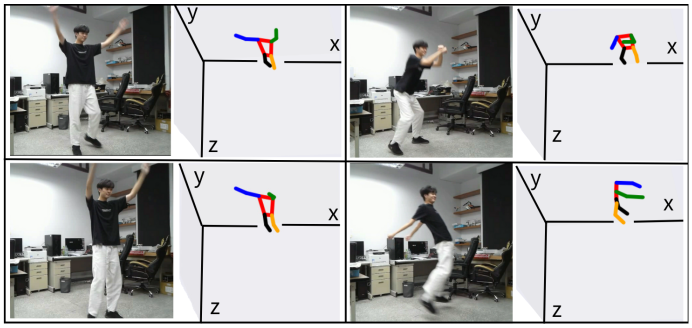

Fig. 6: 3D reconstruction

of joints, the algorithm can more effectively and reliably infer spatial relationships and geometrical structures. This approach reduces the complexity of the correspondence problem and minimizes potential matching errors, ultimately leading to a more accurate 3D reconstruction of the human body.

V. Conclusion

In this paper, we proposed a novel approach for multi-camera calibration that uses human joints as reference points, eliminating the need for traditional calibration patterns. We leveraged human joints and their match corresponding points across camera views. These matched pairs are used to compute and derive the necessary parameters. We refine the parameters by adopting our optimization strategies. This method reduces calibration complexity and time while maintaining accuracy, offering a flexible solution for various environments, without the need of a calibrator.

VI. Acknowledgment

This research work was partially supported by the National Science and Technology Council, grant number 113-2221-E-155-028.

References

[1] Z. Zhang, “A Flexible New Technique for Camera Calibration.” *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 22, pp. 1330-1334, 2000.

[2] R. Y. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses.” *IEEE J. Robotics Autom*, vol. 3, pp. 323-344, 1987.

[3] Cheng, Bowen, B. Xiao, J. Wang, H. Shi, T. S. Huang and L. Zhang. “HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation.” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5385-5394, 2019.

[4] P. E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, “SuperGlue: Learning Feature Matching With Graph Neural Networks.” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019, pp. 4937-4946.

[5] M. J. Tyszkiewicz, P. Fua, E. Trulls, *DISK: Learning local features with policy gradient*, 2020.

[6] C. Lei, Z. Y. Hu, F. C. Wu, and H. T. Tsui, A novel camera self-calibration technique based on the Krupa equation. *in Chinese Journal of Computers*, pp. 587-598, 2003.

[7] M. Pollefeys and L. Van Gool, Self-Calibration from the absolute conic on the plane at infinity. *In Lecture Notes in Computer Science*, pp. 175-182, 1997.

[8] O. D. Faugeras, Q. T. Luong and S. J. Maybank, “Camera Self-Calibration: Theory and Experiments.” *European Conference on Computer Vision* , 1992.

[9] A. M. Ali, Camera auto-calibration for complex scenes, *International Conference on Machine Vision*, 2021.

[10] R. Hassanpour and M. V. Atalay, “Camera auto-calibration using a sequence of 2D images with small rotations” *Pattern Recognition Letters*, vol. 25, pp. 987-997, 2004.

[11] Y. I. Abdel-Aziz, H. M. Karara. “Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry.” *Photogrammetric Engineering and Remote Sensing*, vol. 81, pp. 103-107, 2015.

[12] M. Tang,F. Da, and S. Gai, “Multi-cameras calibration approach based on fringe projection.”, vol. 37, pp. 2149–2155, 2016.

[13] E. Shen, and R. Hornsey, “Multi-Camera Network Calibration With a Non-Planar Target.” *IEEE Sensors Journal*, vol. 11, pp. 2356-2364, 2011.

[14] Y. Huang, F. Da, and H. Tao, “An Automatic Registration Algorithm for Point Cloud Based on Feature Extraction.” *Chinese Journal of Lasers* , vol. 42, 0308002, 2015.

[15] L. Liang, M. Wei, A. Szymczak, A. Petrella, H. Xie, J. Qin, J. Wang and F. L. Wang. “Nonrigid iterative closest points for registration of 3D biomedical surfaces.” *Optics and Lasers in Engineering*, vol. 100, pp. 141-154, 2018.

[16] K. Liu, L. Chen, L. Xie, J. Yin, S. Gan, Y. Yan, and E. Yin. “Auto calibrationof multi-camera system for human pose estimation.” *IETComput. Vis*. 16(7), pp. 607–618, 2022.

[17] D. F. Wang, L. Chen, Z. Gong, T. Liu, and X. Guo. “Human Joints Auto-Calibration Method Based on SuperGlue.” *2023 42nd Chinese Control Conference (CCC)* , 2023, pp. 7735-7739.

[18] Y. Xu, Y. J. Li, X. Weng, and K. Kitani, “Wide-Baseline Multi-Camera Calibration using Person Re-Identification.” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13129-13138, 2021.

[19] S. Huang, M. Gong, and D. Tao, “A Coarse-Fine Network for Keypoint Localization.” *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3047-3056.

[20] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. “Cascaded Pyramid Network for Multi-person Pose Estimation.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7103-7112.

[21] K. Sun, B. Xiao, D. Liu and J. Wang. “Deep High-Resolution Representation Learning for Human Pose Estimation.” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5686-5696.

[22] C. Zhe, G. Hidalgo, T. Simon, S. E. Wei and Y. Sheikh. “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 172-186, 2018.

[23] P. Leonid, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B.Schiele, “DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4929-4937.

[24] M. Kocabas, S. Karagoz, and E. Akbas, “MultiPoseNet: Fast Multi-Person Pose Estimation using Pose Residual Network.” *ArXiv, In: Proc. European Conference on Computer Vision*, 2018.

[25] Besl, Paul J. and Neil D. McKay. “A Method for Registration of 3-D Shapes.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239-256, 1992.

[26] D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperPoint: Self-Supervised Interest Point Detection and Description.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3371-3372, 2017.