Human Pose Estimation using Automated Multi-Camera Calibration

Israt Sharmin Dola[1], Inbae Jeong[2], Youjin Jang[3], and Moein Younesi Heravi[4]

[1,2] Dept. of Mechanical Engineering, North Dakota State Univ., Fargo, USA

[3,4] Dept. of Civil, Construction and Environmental Engineering, North Dakota State Univ., Fargo, USA

israt.dola@ndsu.edu, inbae.jeong@ndsu.edu, y.jang@ndsu.edu, moein.younesiheravi@ndsu.edu

Abstract

3D human joint estimation is essential for enabling effective human-robot interaction in construction automation, facilitating precise monitoring of worker movements to enhance safety, ergonomics, and operational efficiency. Vision-based systems utilizing multi-camera setups offer diverse perspectives to address challenges such as occlusions, projection ambiguities, and sensor noise. However, these systems depend heavily on accurate camera calibration to align views and synchronize measurements. Traditional manual calibration methods are time-consuming, labor-intensive, and prone to human error, making them unsuitable for dynamic construction sites where frequent camera repositioning is required due to shifting conditions and tasks. This study proposes a novel framework that leverages external marker detection for real-time, automated camera calibration, eliminating the need for manual intervention. This approach significantly reduces setup time, minimizes errors, and ensures reliable performance in rapidly changing environments. Furthermore, the framework integrates an Extended Kalman Filter (EKF) to fuse 2D joint locations from multiple cameras, effectively handling sensor noise and the nonlinear nature of human motion. By combining marker-based calibration with EKF-based fusion, the proposed framework delivers a robust and automated solution for 3D human joint estimation, enhancing safety, efficiency, and adaptability in construction automation applications.

Keywords – 3D Human Joint Estimation; Multi Camera System; Automated Camera Calibration; Extended Kalman Filter

1 Introduction

Construction automation has emerged as a transformative approach to improving efficiency, safety, and ergonomics on job sites. Central to this advancement is the ability to accurately recognize human activities and estimate 3D poses, enabling seamless interaction between workers and automated systems. Modern construction sites increasingly feature human-robot collaboration, where workers and autonomous machines must operate in close proximity to accomplish complex tasks. However, these environments present significant hazards, including heavy machinery, work at heights, uneven terrain, and dynamic interactions between workers and automated systems [1], [2], [3]. In such settings, precise monitoring of worker movements is essential not only for ensuring safe interactions but also for optimizing task coordination and preventing errors. By reconstructing human movements through 3D joint estimation, construction managers can identify unsafe behaviors, assess ergonomic risks, and refine workflow strategies. This capability enhances safety protocols, task efficiency, and human-robot collaboration, reducing the likelihood of injuries from improper posture, repetitive strain, or equipment misuse [4].

Various 3D joint estimation approaches leverage advanced sensing, filtering, and modeling techniques. Among these, vision-based methods are widely adopted, utilizing camera data to track joint positions in three dimensions. RGB-D and 3D cameras help overcome the dimensional limitations of 2D cameras by providing depth information [5], [6], [7]. However, such systems often struggle with resolution constraints and sensitivity to lighting variations [8], [9], [10]. Recent monocular 3D pose estimation techniques have advanced significantly by leveraging Convolutional Neural Networks (CNNs) to regress 3D joint locations from single RGB images [11]. These approaches have expanded the application of vision-based motion capture to unconstrained environments, but real-world deployment remains challenging due to occlusion, motion blur, and variable lighting conditions, necessitating further improvements.

While monocular methods offer promising results, they often struggle with depth ambiguities. Multi-camera setups address this limitation by capturing multiple viewpoints of a subject, enabling robust triangulation of joint positions. However, such setups require accurate camera calibration to ensure proper alignment across views. Once calibrated, their performance can be enhanced using data fusion techniques such as the Kalman Filter (KF) and Extended Kalman Filter (EKF) to mitigate sensor noise and missing data, while particle filters are particularly effective in handling nonlinearities and occlusions [12], [13], [14]. Additionally, deep learning-based approaches, including Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), and sequence-to-sequence models, further improve motion prediction and temporal consistency in pose estimation [15], [16], [17].

Despite these advancements, significant challenges remain, particularly in dynamic and complex environments like construction. A major limitation is the reliance on accurate camera calibration, which is essential for precise depth perception and spatial alignment. Misaligned views due to improper calibration compromise depth calculations, reducing 3D estimations accuracy. Traditional calibration methods, such as the Direct Linear Transformation and two-step approaches, provide high precision in estimating intrinsic and extrinsic camera parameters. However, they are manual, time-consuming, and best suited for controlled environments, making them impractical for dynamic and unpredictable construction settings [18], [19].

To address these limitations, researchers have explored alternative approaches. Active vision-based methods leverage controlled camera movements to simplify mathematical modeling, but their dependency on precise motion control makes them challenging to implement in real-world applications. Similarly, advancements in computational techniques, such as neural networks combined with global optimization algorithms, have shown promise for automated calibration. These methods effectively model nonlinear relationships in calibration, but their reliance on large training datasets, intensive computation, and careful initialization can limit their applicability in time-sensitive or resource-constrained settings [20], [21].

While these techniques perform well in controlled environments, they encounter significant challenges in dynamic settings where frequent recalibration is required. On construction sites, cameras must be regularly repositioned to accommodate shifting tasks, occlusions, changing lighting conditions, and complex human motion patterns. Manual recalibration in such scenarios is time-consuming and inefficient, emphasizing the need for a real-time, automated calibration system that can maintain accuracy without disrupting ongoing operations.

In this study, we propose an automated calibration framework that integrates a real-time, self-updating multi-camera calibration process for 3D joint estimation. Unlike traditional methods, where camera calibration is performed separately before pose estimation, our framework simultaneously calibrates camera positions, orientations, and fields of view (FoV) while estimating 3D joint positions, enhancing both efficiency and adaptability. To achieve this, we employ an EKF-based multi-view fusion approach, where 2D joint detections from multiple cameras are integrated to generate robust 3D pose estimates, effectively reducing sensor noise and motion uncertainties. Additionally, we introduce an external marker-based calibration technique that automatically updates camera parameters throughout the estimation process, eliminating the need for manual recalibration, which is often time-consuming and prone to errors. This approach minimizes setup errors, enhances real-time adaptability, and ensures consistent calibration even in dynamic and complex applications such as construction automation.

2 Methodology

This study aims to simultaneously estimate the 3D joint positions of a human and automatically calibrate a multi-camera system. To achieve this, a state vector, X_t is defined at a given time t representing all the unknown variables to be estimated: human 3D joint positions and camera parameters (position, orientation, and FoV).

The 3D joint positions are denoted as $H_{t,j}$, where $j \in \{1, \dots n\}$, and n represents the total number of joints in the human body. Each $H_{t,j}$ specifies the global 3D coordinates of the j-th joint in a Cartesian coordinate system. The camera parameters are expressed as $C_{t,i}$, where $i \in \{1, \dots m\}$, and m denotes the total number of cameras. Each $C_{t,i}$ includes both the extrinsic parameters (3D position and orientation) and intrinsic parameters (horizontal and vertical fields of view) of the i-th camera.

The state vector, X_t is defined as:

$$X_t = \left[H_{t,1}, H_{t,2} \cdots H_{t,n}, C_{t,1}, C_{t,2} \cdots C_{t,m}\right]^T \qquad (1)$$

Here, each joint position $H_{t,j}$ is a 3D vector representing the global coordinates $(x_{h,j}, y_{h,j}, z_{h,j})$ of the j-th joint:

$$H_{t,j} = [x_{h,j} \quad y_{h,j} \quad z_{h,j}]^T \qquad (2)$$

Similarly, the camera parameter $C_{t,i}$ represents the 3D position $(x_{c,i}, y_{c,i}, z_{c,i})$, orientation defined by roll $\varphi_{c,i}$, pitch $\varphi_{c,i}$, yaw $\psi_{c,i}$, as well as the horizontal $(F_{h_{c,i}})$ and vertical $(F_{v_{c,i}})$ FoV:

$$C_{t,i} = \left[x_{c,i}, y_{c,i}, z_{c,i}, \varphi_{c,i}, \varphi_{c,i}, \psi_{c,i}, F_{h_{c,i}}, F_{v_{c,i}}\right]^T \quad (3)$$

This formulation serves as the foundation for simultaneously refining 3D joint positions and calibrating the multi-camera system. The proposed methodology, illustrated in Figure 1, comprises three main steps: 2D joint estimation, marker detection and alignment, and EKF-based integration. 2D joint estimation extracts pixel coordinates of human joints, while marker detection establishes a global reference frame through ArUco marker corners. These measurements feed into the EKF, which refines the state vector, X_t to estimate 3D joint positions and camera parameters, ensuring precise alignment. The following sections provide a detailed explanation of each step.

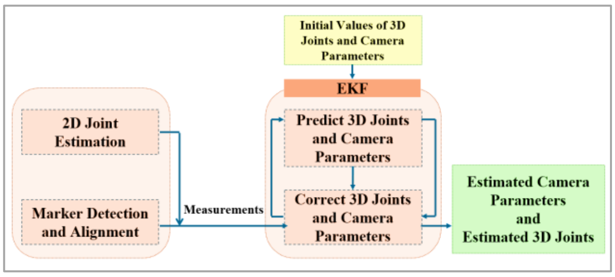
Initial Values of 3D Joints and Camera Parameters
EKF
2D Joint Estimation
Marker Detection and Alignment
Measurements
Predict 3D Joints and Camera Parameters
Correct 3D Joints and Camera Parameters
Estimated Camera Parameters and Estimated 3D Joints

Figure 1. Overall framework

2.1 2D Joint Estimation

The performance of the 3D joint estimation largely depends on accurate 2D joint detection. This study employs MediaPipe's Pose module; a deep learning-based framework developed by Google for high-accuracy, real-time application [22]. It utilizes the lightweight BlazePose model, which features a detector-tracker architecture that reduces latency by tracking previously detected keypoints rather than re-detecting them in each frame. BlazePose outputs 33 keypoints [23]; but this research selects 15 primary keypoints, including major body joints (shoulders, elbows, wrists, hips, knees, and ankles) and key facial landmarks, while excluding keypoints that are not essential for pose estimation [2]. This selection provides a comprehensive 2D representation of the human body, as shown in Figure 2.

Before 2D joint estimation, images from all cameras are synchronized to ensure temporal alignment across views for consistent detection. Each frame is processed individually, with the model outputting pixel coordinates and a confidence score (0–1) for each keypoint. To enhance accuracy, only keypoints with confidence scores above 0.8 are retained, reducing false detections from occlusions or challenging poses. The extracted 2D joint data is structured into arrays for each camera view, enabling synchronized multi-camera processing. This serves as input for the fusion algorithm, facilitating precise 3D joint reconstruction. Leveraging MediaPipe's efficient architecture, this approach achieves low latency, making it ideal for real-time multi-view analysis.

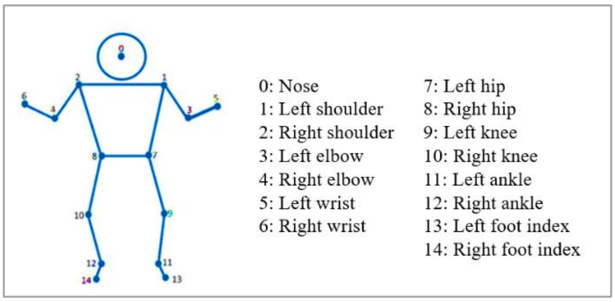

Figure 2. Defined 2D human joints

2.2 Marker Detection and Alignment

In this study, ArUco markers are chosen for their unique binary patterns, which enable easy identification and robust tracking. Unlike checkerboards or circular grids, ArUco markers are resilient to moderate lighting variations and can be reliably detected from diverse angles and distances, making them ideal for multi-camera setups [24].

For this phase, an ArUco marker with a predefined size is used as the reference for automatic camera calibration. The bottom-left corner of the marker is defined as the origin of the global coordinate system, $(0,0,0)$. Based on the marker's size, s, the 3D coordinates of the other corners are defined as $(s,0,0)$, $(0,s,0)$ and $(s,s,0)$. These fixed coordinates serve as reference points for aligning all camera views within a unified coordinate system.

The process begins with detecting the precise corner coordinates of the marker in the camera image planes. These detected coordinates serve as observed measurements for the fusion algorithm, which subsequently estimates both extrinsic and intrinsic camera parameters. The extrinsic parameters, represented by the 3×4 transformation matrix $[R|t]$, define each camera's orientation, R relative to the marker and its position, t in 3D space. Meanwhile, the intrinsic parameters such as FoV, influence how 3D points are projected onto the image plane. This simultaneous estimation ensures that all camera views are accurately aligned within a unified global coordinate system, reducing projection errors and improving multi-view consistency.

2.3 Estimation of Human Pose and Camera Parameter

EKF serves as the core algorithm for simultaneously estimating the 3D human joint positions and the intrinsic and extrinsic parameters of the multi-camera system by integrating synchronized 2D joint detections and marker-based measurements to iteratively refine the state vector X_t (Equation (1)), which encapsulates both the human joint positions and the camera parameters.

The EKF follows a prediction-correction cycle. In the prediction phase, the system propagates the state and uncertainty forward using a process model, where each joint position evolves independently over time, modeled as:

$$x_{t+1,j} = x_{t,j} + \eta_t \tag{4}$$

where $\eta_{t,j} \sim \mathcal{N}(0, Q_t)$ represents Gaussian process noise, with Q_t as a covariance matrix capturing uncertainties for each joint and camera. The prediction step maintains a Gaussian assumption centered around the current estimate while updating the state distribution over time. Unlike methods relying on pre-known 3D joint positions or calibrated camera parameters, our approach uses the EKF framework to estimate both the joint positions and camera parameters dynamically. The EKF estimates and refines these parameters through its fusion of multi-camera observations and marker-based measurements, ensuring adaptability to dynamic environments.

In the correction phase, 2D joint detections and marker corner coordinates are incorporated to refine the state vector. The observation model predicts the expected pixel coordinates (u_e, v_e) for both joints and marker corners based on the current state estimate. Figure 3 illustrates how a point (joint or marker corner) is transformed from the world coordinate system to the camera coordinate system and finally to the pixel coordinate system.

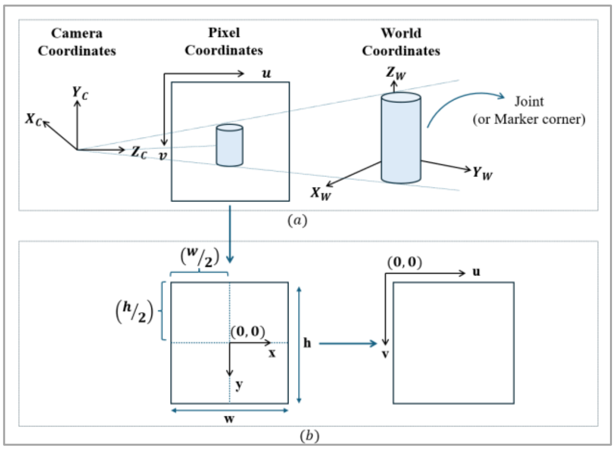

Figure 3. Imaging geometry for joint or marker corner (a) projection joint or marker onto image plane (b) joint or marker in pixel coordinates

To map global joint positions to each camera i's local frame, the EKF employs a transformation matrix T_i and a rotation matrix R_i which is defined as:

$$v_{h,j|C_i} = R_i^{-1} \cdot T_i^{-1} \cdot \mathrm{v}_{h,\mathrm{j|G}} \tag{5}$$

where $\mathrm{v}_{h,\mathrm{j|G}}$ and $\mathrm{v}_{h,\mathrm{j|C_i}}$ represent the j-th joint location in the global frame and in the camera i's frame respectively. They are defined as:

$$\mathrm{v}_{h,\mathrm{j|G}} = \left[x_{h,j}, y_{h,j}, z_{h,j}, 1\right]^T \tag{6}$$

$$v_{h,j|C_i} = \left[x_{h,cam.j}, y_{h,cam,j}, z_{h,cam,j}, 1\right]^T \tag{7}$$

Similarly, each corner of the marker is mapped to each camera i's local frame:

$$v_{m,k|C_i} = R_i^{-1} \cdot T_i^{-1} \cdot \mathrm{v}_{m,\mathrm{k|G}} \tag{8}$$

where $\mathrm{v}_{m,\mathrm{k|G}}$ and $\mathrm{v}_{m,\mathrm{k|C_i}}$ represent the marker's k-th corner location in the global frame and in the camera i's frame respectively. They are defined as:

$$\mathrm{v}_{m,\mathrm{k|G}} = \left[x_{m,k}, y_{m,k}, z_{m,k}, 1\right]^T \tag{9}$$

$$v_{m,k|C_i} = \left[x_{m,cam.k}, y_{m,cam,k}, z_{m,cam,k}, 1\right]^T \tag{10}$$

The transformation matrix T_i, where $(x_{c,i}, y_{c,i}, z_{c,i})$ represents the translation of camera i in the global coordinate system. is defined as:

$$T_i = \begin{bmatrix} 1 & 0 & 0 & x_{c,i} \\ 0 & 1 & 0 & y_{c,i} \\ 0 & 0 & 1 & z_{c.i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

The rotation matrix R_i accounts for the camera's orientation, combining rotations along the x, y, and z axes:

$$R_i = R_z(\theta_{c,i}) R_y(\phi_{c,i}) R_x(\psi_{c,i}) \tag{12}$$

where $\theta_{c,i}$, $\phi_{c,i}$, and $\psi_{c,i}$ are the roll, pitch, and yaw angles of camera i, respectively. These transformations map global 3D positions to the local coordinate system of each camera, resulting in the camera-frame coordinates $(x_{cam}, y_{cam}, z_{cam})$.

The transformed 3D coordinates are then projected onto the image plane to compute the expected pixel coordinates (u_e, v_e) normalized by the image width (w) and height (h) to ensure values range between 0 and 1. Using the horizontal and vertical FoV $(FoV_h$ and $FoV_v)$ the equations for the expected pixel coordinates are:

$$u_e = \frac{1}{2}\left(1 + \frac{x_{cam}}{z_{cam} \tan\left(\frac{FoV_h}{2}\right)}\right) \tag{13}$$

$$v_e = \frac{1}{2}\left(1 + \frac{y_{cam}}{z_{cam} \tan\left(\frac{FoV_v}{2}\right)}\right) \tag{14}$$

In this formulation, we use the pinhole camera model, detailed in figure 4, where FoV inherently defines the camera's intrinsic properties seamlessly integrating with our projection model. This approach eliminates the need for focal length as a separate parameter, making the formulation more compact and efficient while maintaining accuracy in projection calculations. This formulation is applicable to both human joints and marker corners, denoted as $\left(u_{e_{h,j}}, v_{e_{h,j}}\right)$ for the j th human joint and $\left(u_{e_{m,k}}, v_{e_{m,k}}\right)$ for the k-th marker corner.

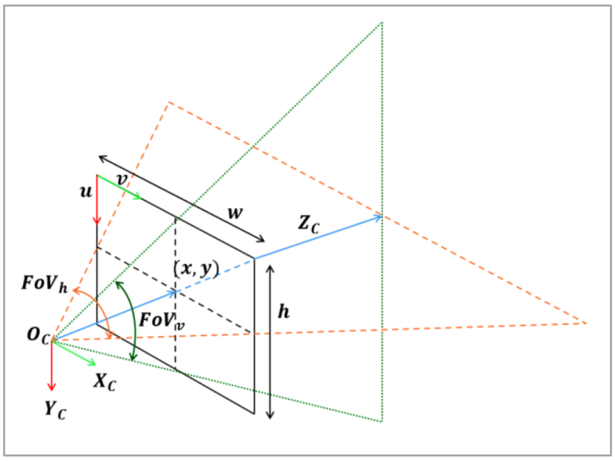

Figure 4. Expected measurements from planar projection

For n cameras, the expected measurement vector $z_{t,e}$ combines the expected 2D pixel coordinates of the detected human joints, $z_{t,e,h}$ and marker corners, $z_{t,e,m}$ from each camera i. It is defined as:

$$z_{t,e} = \left[z_{t,e,h}, z_{t,e,m}\right]^T \tag{15}$$

$$z_{t,e,h} = \left[u_{e,h,i,1}, v_{e,h,i,1}, \ldots, u_{e,h,i,n}, v_{e,h,i,n}\right]^T \tag{16}$$

$$z_{t,e,m} = \left[u_{e,m,i,1}, v_{e,m,i,1}, \ldots u_{e,m,i,k}, v_{e,m,i,k}\right]^T \tag{17}$$

Here $u_{e,h,i,j}$ and $v_{e,h,i,j}$ are the expected pixel coordinates of joint j in camera i's image plane, and $u_{e,m,i,k}$ and $v_{e,m,i,k}$ are the pixel coordinates of marker k's corners.

Once the actual measurements $z_{t,a}$, obtained directly from sensors as noisy 2D pixel coordinates of detected joints and marker corners, are available, the measurement residual,y_t is computed:

$$y_t = z_{t,a} - z_{t,e} \tag{18}$$

where $z_{t,a}$ is derived from the outputs of the 2D joint detection and marker detection algorithms steps discussed earlier in the methodology, while $z_{t,e}$ represents the expected measurements predicted by the observation model.

To incorporate this residual into the state update, the Kalman gain K_t is computed as:

$$K_t = P_t H_t^T (H_t P_t H_t^T + R_t)^{-1} \tag{19}$$

where H_t is the Jacobian of the observation model relative to the state vector, capturing the sensitivity of the measurements to changes in state. R_t is the measurement noise covariance matrix which captures the uncertainty in each camera's observation and P_t is the predicted state covariance matrix, representing uncertainty prior to incorporating the measurements.

In the measurement update step, the EKF integrates the observed measurements $z_{t,a}$. The updated state x_{t+1} is computed as:

$$x_{t+1} = x_{t+1}^{-} + K_t y_t \tag{20}$$

and the updated state covariance, P_{t+1} is:

$$P_{t+1} = (I - K_t H_t) P_{t+1}^{-} \tag{21}$$

where x_{t+1}^{-} and P_{t+1}^{-} the predicted state and covariance before the measurement update.

This EKF-based filtering algorithm effectively combines predicted and observed data, yielding precise and robust 3D joint estimations and camera parameter calibrations in dynamic, multi-camera environments.

3 Experimental Setup

To evaluate the proposed EKF-based multi-camera calibration and 3D joint estimation system, the implementation was carried out in the Webots simulation environment. Webots was chosen for its ability to replicate real-world scenarios in a controlled setting, providing access to ground truth data, enabling repetitive and reproducible experiments, and allowing systematic testing of algorithms under various configurations.

The simulation was designed to emulate a real-world multi-camera setup with four static cameras and a 5 cm ArUco marker. A pedestrian proto model simulated a human subject, with predefined joint positions replicating realistic human motion. Planar projection was utilized to simulate the 2D image planes of the cameras, enabling accurate marker and joint detections essential for subsequent calibration and estimation steps. The key parameters of the simulation environment are as follows:

- Image Resolution: 640 × 640 pixels.
- Marker Size: 5 cm (edge length).
- Camera Configuration: Planar layout with four static cameras.
- Subject: Webots pedestrian proto model.
- Number of Trials: 30.

The evaluation focused on quantifying the accuracy of the estimated 3D joint positions, camera positions, camera orientations, and camera FoVs. The Root Mean Square (RMS) error was used as the primary metric for all evaluations:

$$RMS = \sqrt{\frac{1}{N \cdot K} \sum_{i=1}^{N} \sum\nolimits_{k=1}^{K} \left\| x_{i,k}^{tr} - x_{ik}^{est} \right\|^2} \qquad (22)$$

where, N is the total number of measurements, K is the total number of cameras, $x_{i,k}^{tr}$ and x_{ik}^{est} are the true and estimated values of the quantity being evaluated. x can represent joint positions, camera positions, camera orientations, or FoVs.

By simulating a multi-camera system with realistic projections, human motion, and marker detections, the experimental setup successfully replicated the challenges of real-world environments.

4 Results

This section presents the proposed methodology's findings in the Webots simulation environment. Figure 5 illustrates the successful detection of joints in the pedestrian model and all four corners of the ArUco marker in the image planes from all four cameras. The joint detection provides 2D coordinates for estimating 3D joint positions, while marker corner detection establishes a global reference for camera calibration by defining a consistent coordinate system.

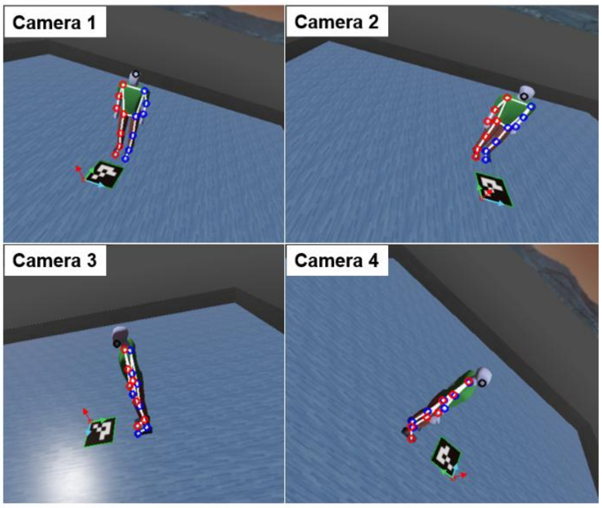

Figure 5. Detection of joints and ArUco marker corners across all camera views

The performance of the proposed methodology was evaluated by analyzing the error trends for joint positions, camera positions, camera orientations, and camera FoV. The results showed an RMS error of 0.18m for joint positions and $0.32m$ for camera positions, while the RMS errors for camera orientations and FoV were approximately 11° and 9°, respectively. Figure. 6 illustrates the error trends for joint positions, camera positions, camera orientations, and camera FoV over time. The graphs show a steady reduction in errors as the EKF iteratively refines the state estimates using multi-camera observations and marker-based measurements.

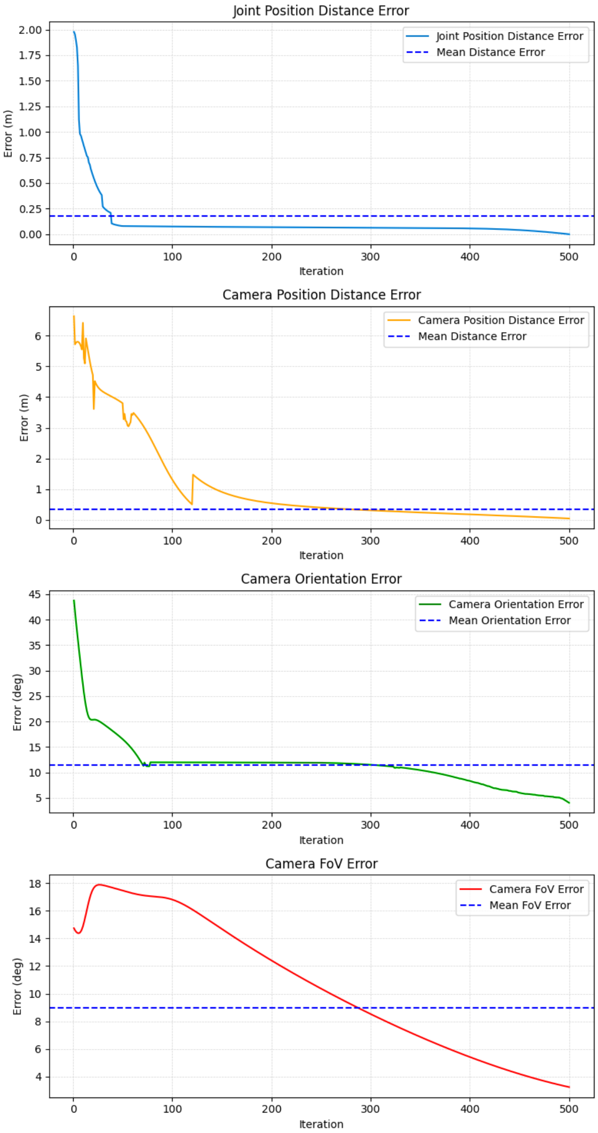

Figure 6. Error trends of joint positions and camera parameters

These RMS errors indicate that the proposed system achieved automatic camera calibration for 3D joint estimation. The consistently small deviations across trials highlight the robustness of the methodology in handling challenges such as sensor noise, nonlinearities, and multi-camera synchronization. Combined with the error trends shown in Figure 6, these findings validate the reliability and accuracy of the framework in achieving its objectives in a controlled simulation environment.

5 Conclusion

This study proposed a novel framework for automatic camera calibration for 3D joint estimation using a multi-camera system. By integrating an EKF for multi-camera data fusion and an external marker-based calibration method, the framework eliminates the need for manual calibration. Simulations conducted in Webots demonstrated its capability to accurately estimate 3D joint positions and maintain reliable calibration with four cameras. Results showed that the automated calibration method reduced reliance on manual interventions, highlighting the framework's potential for real-world applications where rapid deployment and adaptability are critical.

Despite promising results, certain limitations present opportunities for future research. Simulations, while providing controlled and reproducible results, may not fully capture real-world complexities such as dynamic lighting, occlusions, and irregular motion patterns. Future work should validate the framework in real-world settings to assess its robustness. While our approach models intrinsics via FoV, camera parameter calibration remains a challenge, as real-world systems often require calibration of focal length, principal point, and distortion coefficients. Additionally, the reliance on external markers may pose challenges in cluttered environments where occlusions impact visibility. Optimizing marker placement, increasing marker count, or exploring marker-free calibration could enhance performance. Scalability is another concern, as real-time EKF updates and multi-camera processing demand high computational resources. Leveraging parallel processing or Edge AI acceleration could improve efficiency for real-time deployment. Lastly, while EKF performed well, exploring alternative filtering techniques could further improve accuracy or reduce computational overhead.

With these advancements, the proposed framework has the potential to evolve into a robust and adaptable tool for enhancing worker safety and efficiency in high-risk, dynamic environments. This progress would pave the way for more intelligent and scalable systems in construction automation, furthering the integration of AI and robotics in real-world applications.

Acknowledgement

The authors express their gratitude for the support provided by the North Dakota Economic Diversification Research Fund and the National Science Foundation (NSF) EPSCoR RII Track 2 Program under the grant number OIA-2119691. The findings and opinions expressed in this article are those of the authors only and do not necessarily reflect the views of the sponsors.

References

[1] I. Jeong, Y. Jang, J. Park, and Y. K. Cho, "Motion Planning of Mobile Robots for Autonomous Navigation on Uneven Ground Surfaces," Journal of Computing in Civil Engineering, vol. 35, no. 3, May 2021, ASCE, doi: 10.1061/(asce)cp.1943-5487.0000963.

[2] Y. Jang, I. Jeong, M. Younesi Heravi, S. Sarkar, H. Shin, and Y. Ahn, "Multi-Camera-Based Human Activity Recognition for Human–Robot Collaboration in Construction," Sensors, vol. 23, no. 15, Aug. 2023, MDPI, doi: 10.3390/s23156997.

[3] K. Kim and Y. K. Cho, "Effective inertial sensor quantity and locations on a body for deep learning-based worker's motion recognition," Automous Construction, vol. 113, May 2020, Elsevier, doi: 10.1016/j.autcon.2020.103126.

[4] F. Hoenigsberger et al., "Machine Learning and Knowledge Extraction to Support Work Safety for Smart Forest Operations," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Science and Business Media Deutschland GmbH, 2022, pp. 362–375. doi: 10.1007/978-3-031-14463-9_23.

[5] J. Wang, L. Liu, W. Xu, K. Sarkar, and C. Theobalt, "Estimating Egocentric 3D Human Pose in Global Space." 2021. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 11500–11509. IEEE. doi: 10.1109/ICCV48922.2021.01130.

[6] D. Tome et al., "SelfPose: 3D Egocentric Pose Estimation From a Headset Mounted Camera," IEEE Trans Pattern Anal Mach Intell, vol. 45, no. 6, pp. 6794–6806, Jun. 2020, doi: 10.1109/TPAMI.2020.3029700.

[7] J. Shotton et al., "Real-Time Human Pose Recognition in Parts from Single Depth Images," 2011. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), Colorado Springs, CO, USA, pp. 1297–1304. IEEE. doi.org/10.1109/CVPR.2011.5995316.

[8] Zequn Zhang, Yuanning Liu, Ao Li, and Minghui Wang, A Novel Method for User-Defined Human Posture Recognition Using Kinect. Proceedings of the 7th International Congress on Image and Signal

Processing (CISP 2014), Dalian, China, pp. 736–740. IEEE, 2014. doi.org/10.1109/CISP.2014.7003875.

[9] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," Pattern Recognit, vol. 108, Dec. Elsevier. 2020, doi: 10.1016/j.patcog.2020.107561.

[10] A. Jalal, Y. H. Kim, Y. J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," Pattern Recognit, vol. 61, pp. 295–308, Jan. 2017, Elsevier. doi: 10.1016/j.patcog.2016.08.003.

[11] D. Mehta et al., "Monocular 3D human pose estimation in the wild using improved CNN supervision," In Proceedings of the 2017 International Conference on 3D Vision (3DV), pp. 506–516. IEEE. doi: 10.1109/3DV.2017.00064.

[12] M. Gholami, A. Rezaei, H. Rhodin, R. Ward, and Z. J. Wang, "TriPose: A Weakly-Supervised 3D Human Pose Estimation via Triangulation from Video," arXiv preprint, arXiv:2105.06599. [Online]. Available: http://arxiv.org/abs/2105.06599.

[13] L. Chen, H. Ai, R. Chen, Z. Zhuang, and S. Liu, "Cross-view tracking for multi-human 3D pose estimation at over 100 FPS." In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 3276–3285. IEEE. doi:10.1109/CVPR42600.2020.00334.

[14] X. Li, Z. Fan, Y. Liu, Y. Li, and Q. Dai, "3D pose detection of closely interactive humans using multi-view cameras," Sensors (Switzerland), 19(12), 2831. MDPI. doi: 10.3390/s19122831.

[15] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. F. Moura, "Adversarial Geometry-Aware Human Motion Prediction," 2018. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 786–803. Springer. .

[16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," 2016. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5308–5317. IEEE. doi: 10.1109/CVPR.2016.573.

[17] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," 2017. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), pp. 4674–4683. IEEE. doi: 10.1109/CVPR.2017.497.

[18] Z. Qi, L. Xiao, S. Fu, T. Li, G. Jiang, and X. Long, "Two-step camera calibration method based on the SPGD algorithm," Applied Optics, 51(26), 6421–6428. Optica Publishing Group. doi: 10.1364/ao.51.006421.

[19] L. Song, W. Wu, J. Guo, and X. Li, "Survey on camera calibration technique," 2013. In Proceedings of the 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2013), pp. 389–392. IEEE. doi: 10.1109/IHMSC.2013.240.

[20] D. Xianzhi, "Research on Camera Calibration Technology Based on Deep Neural Network in Mine Environment," 2020. In Proceedings of the 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL 2020), pp. 375–379. IEEE. doi: 10.1109/CVIDL51233.2020.00-68.

[21] Wang Qi, Fu Li, and Liu Zhenzhong, "Review on Camera Calibration," 2010. In Proceedings of the 2010 Chinese Control and Decision Conference (CCDC), Xuzhou, China, pp. 3354–3358. IEEE. doi.org/10.1109/CCDC.2010.5498574.

[22] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," 2019. arXiv preprint, arXiv:1906.08172. [Online]. Available: http://arxiv.org/abs/1906.08172.

[23] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device Real-time Body Pose tracking," Jun. 2020, arXiv preprint, arXiv:2006.10204. [Online]. Available: http://arxiv.org/abs/2006.10204.

[24] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial Markers for Pose Estimation: Overview, Applications and Experimental Comparison of the ARTag, AprilTag, ArUco and STag Markers," Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 101, no. 4, Apr. 2021, Springer. doi: 10.1007/s10846-020-01307-9.