Multi-object triangulation and 3D footprint tracking for multi-camera systems

Gabriel Donna Altoé*, Miquelly Nicolini Lucas*, Luiz Carlos Cosmi Filho*, and Raquel Frizera Vassallo*
*Electrical Engineering Department
Federal University of Espírito Santo, Vitória, Brazil
{gabriel.altoe, miquelly.lucas, luiz.cosmi}@edu.ufes.br, raquel.vassallo@ufes.br

***Abstract*—This paper presents a method for 3D footprint triangulation and tracking of multiple objects from different classes using a multi-camera system with overlapping views. The proposed method is designed to operate in complex environments such as Intelligent Spaces, where localization of heterogeneous entities (e.g., humans, robots, and furniture) is essential. Our approach integrates an object detection model (YOLO neural network) and a tracking algorithm (BoT-SORT). To establish cross-view correspondences, we exploit epipolar geometry by computing the normalized cross-distance between detection pairs and introducing a temporal consistency mechanism that favors previously matched pairs. Then a greedy matching algorithm, with filtering heuristics, provides robust association across camera views. From the valid matches, a graph is constructed, and connected components are triangulated to estimate the 3D footprint positions of each object. Our system requires only camera calibration and supports a flexible number of views, making it suitable for deployment in real-world multi-camera setups. The method is first evaluated by comparing reconstruction results against ArUco marker position estimation. Then a set of experiments demonstrates the effectiveness of the algorithm in tracking multiple entities of the same class, as well as different object classes. Due to its modular and generalizable structure, the framework supports any detector and tracker combination, while the proposed method is applicable to a wide range of scenarios.**

I. Introduction

A wide range of real-life applications, such as surveillance, industrial automation, and behavior analysis, demands tracking objects in the workspace. An accurate 3D tracking allows monitoring the movement of objects in space and time, which is crucial for tasks including anomaly detection, activity recognition, and motion planning.

For example, the so called “Intelligent Spaces” are the type of application that demands these requirements. Defined as physical environments equipped with a network of sensors, actuators, and computational services designed to meet the needs of users within the environment [1], in such spaces, understanding the precise position and trajectory of individuals or objects is essential to ensure safety, optimize workflows, and enable context-aware interactions.

Besides that, multi-camera systems provide robust spatial understanding by capturing scenes from multiple viewpoints, overcoming occlusions, limited fields of view, and depth ambiguity common in single-camera setups [2]. If proper calibration and synchronization is provided, accurate 3D triangulation and consistent tracking can be performed, making them ideal for understading complex environments like Intelligent Spaces. However, challenges remain in data association, computational cost, and real-time processing.

Several studies have tackled these challenges using homography or epipolar geometry. Almonfrey *et al.* [3] use homographies in a multi-camera setup to reproject bounding boxes and match them via IoU (Intersection over Union). Lee *et al.* [4] apply epipolar geometry in a stereo system to triangulate 3D positions, using distance ratio and frame grouping to reduce mismatches. Recently Yang *et al.* [5] propose a unified multi-camera framework that generates robust 3D trajectories by combining multi-view and multi-frame associations. Other works focus exclusively on multi-view multi-object tracking — such as using re-identification models and clustering algorithms to perform cross-view association [6].

Despite these advances, existing methods present limitations that restrict their applicability in more general scenarios. For instance, homography-based approaches [3] cannot estimate positions above the ground plane. Additionally, some methods rely on a fixed stereo pair [4], limiting flexibility in dynamic camera setups. Moreover, most existing solutions are designed specifically for person tracking [3]–[6] or do not incorporate 3D spatial reconstruction [6].

Therefore, this work proposes a method for 3D footprint triangulation and tracking of multi-objects of different classes (e.g.: robots, person, and chairs) using a multi-camera system. Our goal is to present a more general solution capable of tracking and triangulating any type of object using a variable number of cameras, suitable for Intelligent Spaces. Our approach relies on an object detection model [7] to identify objects of interest and a single-view tracking algorithm [8] to maintain temporal consistency. Given the high availability of object detection neural networks (pre-trained models in many datasets [9]–[12]) and the ease of re-training, our method can be applied in different contexts and provide good results.

II. Proposal

For better understanding, our method is divided into three main stages: object detection and tracking in images (Section II-A), geometry-based cross-view matching (Section II-B), and 3D triangulation/tracking (Section II-C).

A. Object Detection and Tracking

Our approach relies solely on bounding boxes and their corresponding identifiers. To obtain this information, it is necessary to use neural networks for object detection and algorithms for object tracking in image sequences. In this work, we employ the YOLO neural network [7] for detection and the BoT-SORT tracking algorithm [8] to maintain object identifiers across frames. However, the proposed method is not limited to these specific models; any object detector or tracking algorithm capable of producing consistent bounding boxes and track IDs can be used within the same framework.

Since the system involves multiple cameras, it is necessary to detect and track objects across multiple image streams concurrently. This requires processing each video feed independently while ensuring temporal and spatial consistency among detections from different viewpoints. Figure 1, shows detections in an overlapping scene with four views.

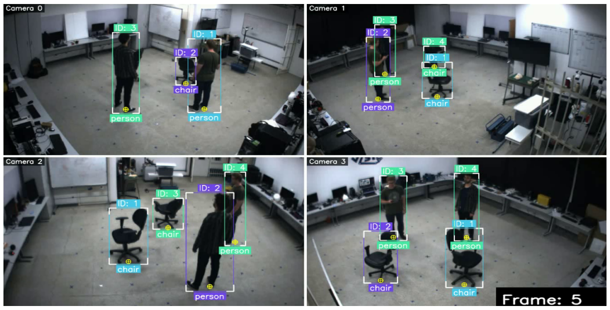


Fig. 1. Views of four cameras with detections in a overlapping scene.

B. Geometry-based Cross-view Matching

After object detection and tracking in each video feed, we perform a matching between detected objects from different views. Since the camera system is pre-calibrated, we compute the fundamental matrices $\mathcal{F}_{ij}$ between all camera pairs (i, j), with $i \neq j$. These matrices allow us to map bounding box points between views through epipolar geometry.

To establish cross-view correspondences, we evaluate all same-class detection pairs across different camera views. For each pair, we compute the normalized cross-distance d'_{cross}, which measures the geometric consistency between two detections based on their proximity to the corresponding epipolar lines in each image, normalized by their bounding box dimensions as proposed by Yang *et al.* [5]. Figure 2, illustrate an example of computation of objects distances to epipolar lines between two candidate pairs.

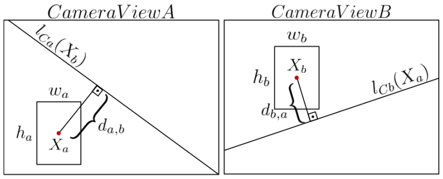


Fig. 2. Computation of object distance to epipolar lines.

Where $d_{a,b}$ denotes the minimum distance from point X_a to the epipolar line $l_{Ca}(X_b)$ and $d_{b,a}$ denotes the minimum distance from point X_b to the epipolar line $l_{Cb}(X_a)$. Then, we normalize by the height and width of each object to get d'_{cross}, as defined in Eq. (1).

$$d'_{cross}(X_a, X_b) = \frac{d_{a,b}}{|w_a + h_a|} + \frac{d_{b,a}}{|w_b + h_b|} \tag{1}$$

We then introduce a temporal consistency mechanism to favor matches that were already established in the previous frame. We denote by $\mathcal{M}_{\text{prev}}$ the set (or cache) of all detection-pair keys that were matched in the previous frames. For each new candidate pair of detections (D_a, D_b), we check whether it exists in a cached set of previous matches and apply a small bonus to its score, in this case, reducing the cross-distance.

$$d_{\text{score}} = \begin{cases} d'_{\text{cross}} - \text{bonus}, & \text{if } (D_a, D_b) \in \mathcal{M}_{\text{prev}}, \\ d'_{\text{cross}}, & \text{otherwise.} \end{cases} \tag{2}$$

Candidate pairs are sorted by their distance score d_{score}, filtered by a minimal threshold T_{min}, and selected using a greedy matching algorithm. Additionally, ambiguous matches — those whose scores differ by less than a drift threshold T_{drif} — are discarded to ensure robustness. Finally, accepted matches are stored in the cache for future frames, and memory usage is managed by periodically clearing old entries.

C. Triangulation and 3D Tracking

The geometry-based cross-view matching provides a list of valid correspondences. Then, we define a graph, $G = (V, E)$, where V is the set of nodes representing all objects detected in all camera views, and E is the set of edges corresponding to the matches found between objects (view Figure 3).

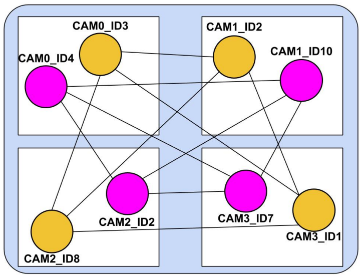


Fig. 3. Multi-camera detection association graph made using the geometry-based cross-view matching algorithm.

Then, we reconstruct each connected component of the graph as a 3D point. To infer the 3D positions of the objects from their 2D estimated bounding box, we use a linear algebraic triangulation approach as defined by Harley *et al.* [13]. The reference 2D point used for triangulation is the bounding box footprint, bottom-center point.

Once the 3D coordinates are estimated, we apply a tracking algorithm to propagate all targets' identity into the next frame in 3D space. We approximate the inter-frame displacements of each object with a linear constant velocity model. The state $\boldsymbol{x}_{id}$ of each target is modeled as follows:

$$\boldsymbol{x}_{id} = [x_{id}, y_{id}, z_{id}, \dot{x_{id}}, \dot{y_{id}}, \dot{z_{id}}] \tag{3}$$

Where x_{id}, y_{id} and z_{id} are the object 3D coordinates, and $\dot{x_{id}}$, $\dot{y_{id}}$ and $\dot{z_{id}}$ are the velocities. When a detection is associated to a target, the estimated 3D point is used to update the target state where the velocity components are estimated via a Kalman filter [14]. For data association, a Hungarian algorithm [15] is used, and the assignment cost matrix is computed as the euclidean distance between the 3D point and all the existing 3D target points.

III. Experiments and Results

To perform 3D footprint tracking and reconstruction of multiple objects from different classes using a multi-camera system, we employed a setup with four cameras mounted in the upper corners of a room and oriented toward the center. As a result, all cameras captured the same scene from different viewpoints, ensuring full coverage of the environment.

A video showcasing the two experiments described in Section III-B and Section III-C, along with several additional experiments, is available at this link.

Furthermore, the code of this project is available at this link.

A. Evaluating system precision

The first experiment aimed to evaluate the reconstruction accuracy of our method in comparison to a baseline. For this purpose, ArUco markers were evenly distributed across the floor ($z = 0$) in a 7×7 grid, with 0.5m spacing between adjacent markers. The 3D position of each ArUco marker in the grid was recovered using its detections in each camera and the triangulation algorithm cited in section II-C.

To simulate human presence at each reference point, a person stood at the exact location of each marker, reproducing the grid layout. The 3D position of the person's feet—approximated by the bottom-center point of the bounding box—was then triangulated using our proposed method. This setup enabled a direct and controlled comparison between the positions provided by the ArUco markers and the the ones estimated by our triangulation algorithm.

Table I reports the mean (μ) and standard deviation (σ) of the triangulation error for both methods under different camera configurations. Additionally, Figure 4 provides a qualitative visual comparison of the results. Overall, the results demonstrate that our method achieves satisfactory triangulation error for the proposed task, and closely matches the ArUco-based triangulation, indicating its effectiveness.

Method	Num. Cameras	μ (m)	σ (m)	Min (m)	Max (m)
ArUco	2	0.0663	0.0461	0.0056	0.2696
ArUco	3	0.0640	0.0457	0.0015	0.2696
ArUco	4	0.0577	0.0403	0.0147	0.2431
Person	2	0.1736	0.0535	0.0194	0.3167
Person	3	0.1581	0.0435	0.0416	0.2680
Person	4	0.1489	0.0434	0.0248	0.2316

TABLE I
Triangulation error of each method.

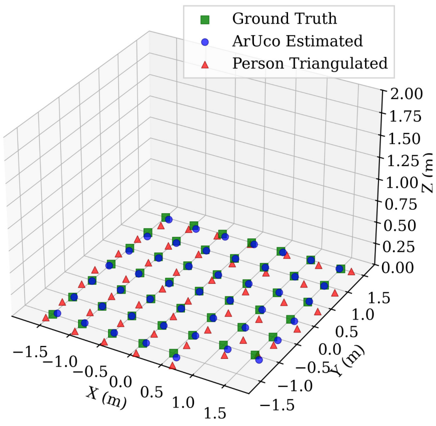
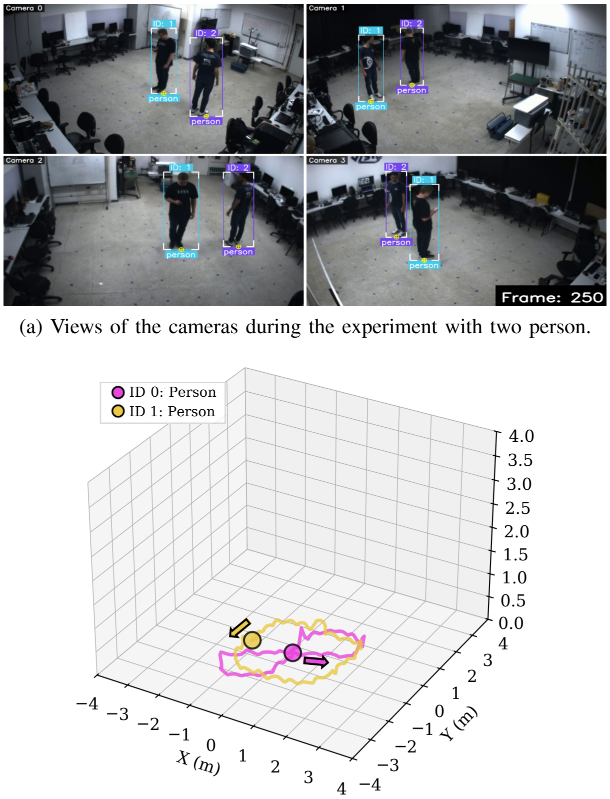


Fig. 4. Comparison between our method and triangulation with ArUco.

B. One-class results

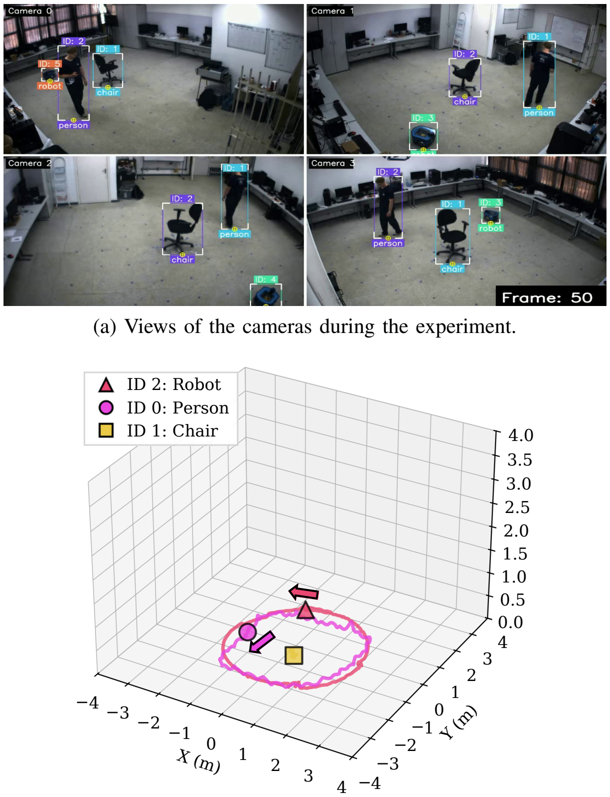
As the second experiment, we demonstrate the effectiveness of the algorithm in handling multiple entities of the same class. To this end, we conducted a test involving two people: one followed a circular path around the origin, while the other followed a lemniscate-shaped trajectory. This setup was designed to evaluate the algorithm's ability to track and triangulate multiple individuals moving along distinct and dynamic paths within the same scene. The 3D footprint reconstruction results are presented in Figure 5.



(a) Views of the cameras during the experiment with two person.

(b) Individuals trajectories during experiment.

Fig. 5. Multiple person experiment.



(a) Views of the cameras during the experiment.

(b) Objects trajectories during experiment.

Fig. 6. Multiple objects experiment.

C. Multi-class results

Finally, the last experiment demonstrate that the algorithm is capable of handling multiple objects from different classes while still producing effective results. In this scenario, a chair was placed at the origin, while a person and a mobile robot moved along a circular path around it (see Figure 6a). This setup highlights the algorithm's ability to simultaneously detect, distinguish, and triangulate heterogeneous entities in a shared environment. The 3D footprint reconstruction results are presented in Figure 6b.

IV. Conclusion

This work presented an extension to existing methods for multi-camera multi-object triangulation and 3D tracking. Our system integrates off-the-shelf 2D object detectors and trackers with a geometry-based cross-view matching algorithm. By leveraging epipolar constraints and a temporal consistency bonus, the system associates detections across multiple viewpoints, which are then fused into a 3D representation using triangulation. The proposed method has a great potential for application in various scenarios, such as surveillance, autonomous navigation, industrial security monitoring, and intelligent spaces (e.g.: museums, smart houses, workspaces and industries).

Despite its effectiveness, the system's performance depends on the quality of the underlying 2D detection and the initial camera calibration. Additionally, the method requires approximately synchronized camera feeds, as the triangulation relies on this premise. Future work will focus on enhancing the matching algorithm to increase its robustness in scenarios with high detection density by incorporating appearance features, generated by deep learning models for re-identification. Furthermore, we also intend to evaluate the proposed method on public benchmark datasets to assess its performance in comparison with existing state-of-the-art methods.

Acknowledgment

The authors would like to thank FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo for the financial support through the project "Cooperação Humano-Robô-Ambiente em um Espaço Inteligente A2AI", T.O. 777/2024, and the scholarships granted to two of the authors.

References

[1] J.-H. Lee and H. Hashimoto, "Intelligent space - concept and contents," *Advanced Robotics*, vol. 16, no. 3, pp. 265–280, 2002.

[2] A. S. Olagoke, H. Ibrahim, and S. S. Teoh, "Literature survey on multi-camera system and its application," *IEEE Access*, vol. 8, pp. 172 892–172 922, 2020.

[3] D. Almonfrey, A. P. do Carmo, F. M. de Queiroz, R. Picoreti, R. F. Vassallo, and E. O. T. Salles, "A flexible human detection service suitable for intelligent spaces based on a multi-camera network," *International Journal of Distributed Sensor Networks*, vol. 14, no. 3, 2018. [Online]. Available: https://doi.org/10.1177/1550147718763550

[4] Y.-J. Lee, M.-W. Park, and I. Brilakis, "Entity matching across stereo cameras for tracking construction workers," in *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC)*, A. A. U. Sattineni, S. A. U. Azhar, and D. G. T. U. Castro, Eds. Auburn, USA: International Association for Automation and Robotics in Construction (IAARC), July 2016, pp. 669–677.

[5] F. Yang, S. Odashima, S. Yamao, H. Fujimoto, S. Masui, and S. Jiang, "A unified multi-view multi-person tracking framework," *Computational Visual Media*, vol. 10, no. 1, pp. 137–160, 2024. [Online]. Available: https://www.sciopen.com/article/10.1007/s41095-023-0334-8

[6] P. Kohl, A. Specker, A. Schumann, and J. Beyerer, "The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[7] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[8] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *ArXiv*, vol. abs/2206.14651, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:250113384

[9] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

[10] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, J. Li, X. Zhang, and J. Sun, "Objects365: A large-scale, high-quality dataset for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8425–8434.

[11] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[12] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[14] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

[15] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.