# From a Bird's Eye View to See: Joint Camera and Subject Registration without the Camera Calibration
## *Supplementary Material*

Zekun Qian[1], Ruize Han[2,3†], Wei Feng[1], Song Wang[4]
[1]College of Intelligence and Computing, Tianjin University
[2]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[3]City University of Hong Kong [4]University of South Carolina
{clarkqian, han_ruize, wfeng}@tju.edu.cn, songwang@cec.sc.edu

## Details of the Subject Matching Algorithm

As presented in the '**Subject Registration**' in Section 3.4. We consider two constraints for accurate matching. The first one is cycle consistency, which means the connection of the same subject from all views should form a loop. The second one is uniqueness, which means one subject should not be connected to more than one subject in another view. To clearly explain the two constraints and our solutions. We present an example to illustrate.

After applying the binarization operation with thresholds, we can get a mask matrix to show which pairs may be the same person, as illustrated in Figure 1(a). Analyzing the matrix, we can know that person A in view 1 matches both persons B and C in view 2 and person D in view 3. Similarly, person B in view 2 may match person D and E in view 3. The black arrows in Figure 1(b) visually represent the matching relationships.

Considering the matching results, the first problem here is the lack of cycle consistency. we can see that A and B are connected, as well as B and E are connected. If these two connections are correct, the cycle consistency requires that A and E should also be connected as the same person. But we can see from the mask matrix that A and E have no connection between them.

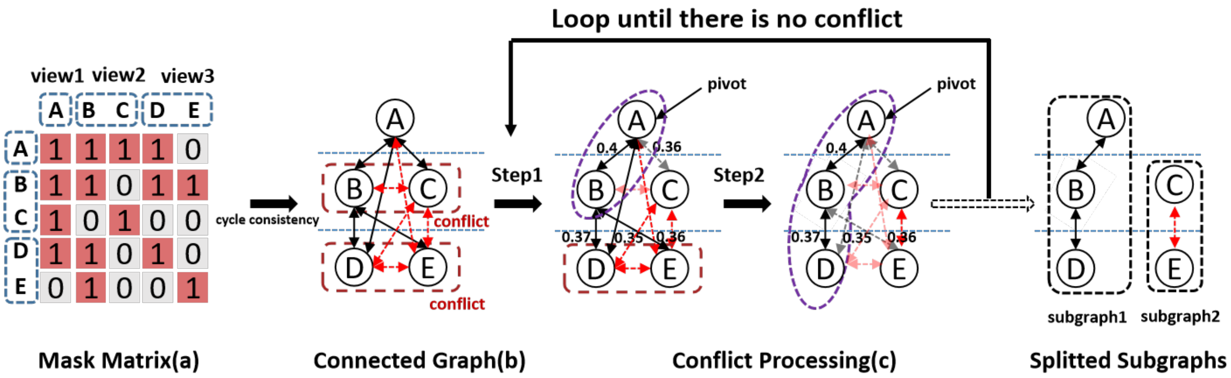
To solve the problem, we use a data structure called union-find to aggregate the transitive relation in the mask matrix. For every aggregated union from the union-find, we create an augmented graph with hidden edges as the red dashed arrow shown in Figure 1(b). Now, there is an implicit connection between A and E by the transitive path: A to B and B to E, where the weight of each edge is the confidence score from the similarity matrix.

**Algorithm 1:** Uniqueness conflict solving:

```
Input: $S_{\rm ori}$: A set of node indices with some uniqueness conflicts,
       $M_{\rm pred}$: A similarity score matrix between all persons,
       Mask: A mask matrix to denote the connection between different persons.
Output: $L_{\rm res}$: A list of sets of node indices without uniqueness conflict.
$L_{\rm res}$ = [] //used to record the result of divided subgraphs.
$L_{view}$ = DivideGraphByView($S_{\rm ori}$) //Dividing the nodes into different views.
while Length($L_{\rm view}$) $> 0$ do
    n = Length($L_{\rm view}$)
    pivot = $L_{\rm view}$[0][0]
    tmp_set = new set()
    tmp_set.add(pivot)
    $L_{\rm view}$[0].pop(pivot) // Removing the pivot node from the original graph.
    for $v = 1 : n - 1$ do
        node = GetMaxScoreOfView($L_{\rm view}[v]$, tmp_set, $M_{pred}$, Mask) //Used to get the selected node in this view, if no node meets the condition will return -1.
        if node $\ ! = -1$ then
            tmp_set.add(node)//Adding the selected node to subgraph.
            $L_{\rm view}$[v].pop(node)//Removing the selected node from the original graph.
    $L_{\rm res}$.append(tmp_set)//Saving the subgraph.
    RemoveEmptyView($L_{\rm view}$)//Removing the layer(view) with no node remaining.
return $L_{\rm res}$
```

We divide the nodes of the graph into different layers (representing different views) as separated by blue dashed lines in Figure 1(b). We can find some conflicts of uniqueness between B and C (both connected to A), D and E

[†] Corresponding author.



|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 1 | 1 | 1 | 1 | 0 |
| B | 1 | 1 | 0 | 1 | 1 |
| C | 1 | 0 | 1 | 0 | 0 |
| D | 1 | 1 | 0 | 1 | 0 |
| E | 0 | 1 | 0 | 0 | 1 |

Figure 1. An example of solving the cycle consistency and uniqueness.

(both connected to B) in the graph, as highlighted within red dashed rectangles. We consider cutting the graph into reasonable subgraphs without uniqueness conflicts. We define the problem as a hierarchical maximum spanning subgraph problem, the layer-by-layer (view-by-view) spanning constraint that a subject is connected at most to one node in each view to avoid the uniqueness conflict. Figure 1(c) shows the complete flow of our solution of an example.

Specifically, first, we select A from view 1 as a pivot and search the max confidence edge to view 2, the edge A-B with the highest 0.4 score is selected. Then nodes A and B are divided into the sub-graph as indicated by the purple dashed area in the figure, and the uniqueness conflict has been resolved in view 2. After that, we detach the sub-graph from the original graph in view 1 and view 2 by cutting off the connections A-C and B-C, represented as transparent dashed arrows in the figure. Second, we search the maximum spanning node between the detached sub-graph and nodes in view 3. There are three candidates B-D with a similarity score of 0.37, A-D with 0.35, B-E with 0.36, and the max one is B-D with 0.37. So, we merge node D into the sub-graph and cut off all the conflicted edges to solve the uniqueness conflict in view 3. Here, a maximum spanning subgraph A-B-D is divided from the original graph. Third, we repeat the flow as the above two steps layer by layer: choosing a pivot in the remaining nodes and dividing the maximum spanning graph in sequence. The flow won't stop until there is no uniqueness conflict. The pseudo code of the above algorithm is shown in Algorithm 1. When there is no conflict, the remaining nodes will be divided into different subgraphs depending on their connection relations.

Overall, we consider both the implicit connection relations for cycle consistency constraint and the hierarchical maximum spanning for uniqueness constraint.

## Dataset Statistics

The dataset statistics for CSRD-II, CSRD-V, and CSRD-R are shown in Table 1.

Table 1. Dataset statistics.

|  | # Images | # Annotations | # Views | # Sub./Frm. | # Scenarios |
|---|---|---|---|---|---|
| CSRD-II | 3K | 51K | 2 | 5-25 | 1 |
| CSRD-V | 5K | 97 K | 5 | 5-25 | 1 |
| CSRD-R | 15 K | 170 K | 2-4 | 7-12 | 5 |

## Details of Comparison Methods

We first compare our method with other methods for the *camera registration* task.

• *DMHA*: DMHA [2] achieves the task of camera registration by using the real BEV image. Besides the FPV images, we additionally provide the corresponding BEV image generated by our data engine to DMHA. To evaluate the results, we use the ground-truth position of camera wearers and predicted camera wearers in the generated BEV to calculate the distance and angle errors.

• *SIFT + KNN* and *other deep-learning-based methods*: We also compare with some key point matching based methods, including both the traditional method like SIFT[6] and the latest CNN based matching methods [5, 7–9]. The input of both methods is a pair of FPV images and then we can get some key point matching pairs. After that, we use the classical camera pose estimation method with the matched key points to generate the essential matrix and convert it to the relative camera location and (yaw-axis) direction. Note that, the error of SIFT is relatively large, some camera position estimation is out of the scene border, in this case, we crop the position of estimation to the outer boundary of that axis. But the same problem does not occur in the deep-learning-based methods for their relatively higher precisions.

For the second task of *subject registration*, we first compared with a single-view human depth estimation method namely Monoloco++ [1]. Also, we include several works[3,

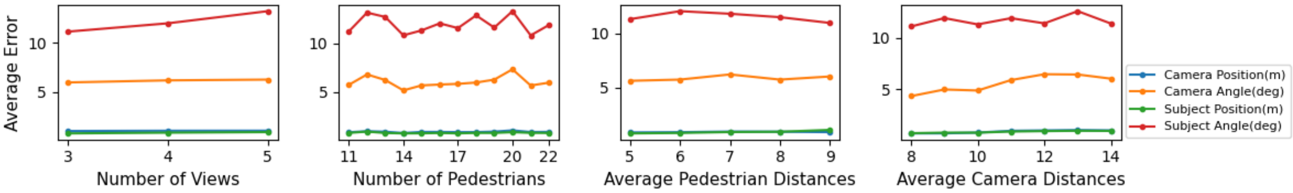


Figure 2. Results for sensitive analysis.

4] for multi-view detection, which both require the camera calibration to project all views into a shared plane to create the occupancy map.

• *Monoloco++* [1]: Monoloco++ is a network trained on KITTI and nuScenes datasets, which is used to predict the 3D-localization and face orientation of each person in the view. We concatenate it with our proposed geometric transformation and subject fusion methods for evaluation.

• *MVDet* and *MVDetr* [3, 4]: These two methods need camera calibrations for generating the results of subject registration. So we calculate the camera calibrations by using the 3D localization of feet (with height = 0) and the 2D position of the bottom of the bounding box of each person predicted in our methods. With the calibration, these two methods generate multi-view human detection predictions (without human identifications) in the BEV. Then we evaluate the results by using the Hungarian matching algorithm to match the identification of all the predicted points with the ground-truth ones through the minimum spatial distance.

## Sensitivity Analysis

Here, we provide the sensitivity analysis of our method to the number of views/pedestrians and the locations of pedestrians and cameras in the figure above. As shown in Figure 2, we can see that the angle prediction results are more sensitive, but the overall fluctuation of the angle prediction basically stays within 2 degrees, while the position prediction results are quite stable within a very small range.

## Time Complexity Analysis

As it is shown in Table 2, we compute the time efficiency of the proposed method. Specifically, we counted the average speed (fps) of different modules and the overall speed. We can first see that the overall speed is *over the real-time efficiency*. Moreover, the feature extraction operations in VTM and Association modules take up the main time cost, which can be parallel implemented for multi-view input for further acceleration.

Table 2. Time efficiency of different components in our method.

| Module | VTM | Association | SAM | Registration | Overall |
|---|---|---|---|---|---|
| FPS | 126.14 | 51.21 | 1490.36 | 3423.51 | 35.19 |

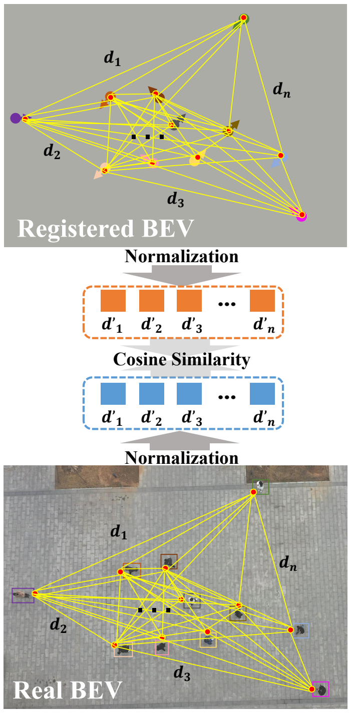


Figure 3. An illustration of the proposed geometric similarity-based localization metric. We use $d_i$ to represent the distance between a pair of subjects in BEV and $d'_i$ to represent the normalized distance.

## Evaluation Metric for Real-world Dataset

*Geometric similarity-based localization metric:* Evaluating results on real datasets can be challenging for the absence scale between the real BEV and the registered BEV. Here, we propose a geometric similarity-based localization metric, allowing cross-domain performance evaluation between the real BEV and the registered BEV. To achieve this, we first calculate the normalized distances among all subjects in the real and the registered BEVs separately. Then, we flatten the normalized distances as vectors by aligned IDs between the real BEV and the registered BEV. We calculate the cosine similarity between these two vectors and use it to measure the result of cross-domain registration, as shown in Figure 3.

## More Visualization Results

We show more visualization results in different situations. As shown in the following Figures 4-7, we can see that our method can accomplish the task very well, even in some difficult and special cases.

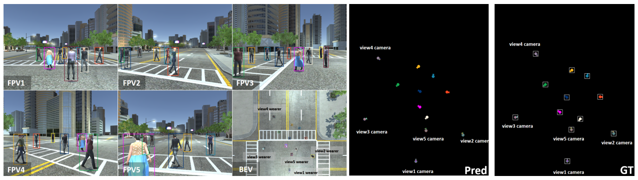


Figure 4. In this case, the camera wearer of view3 does not appear in any FPV. Our camera registration method can still predict it accurately.

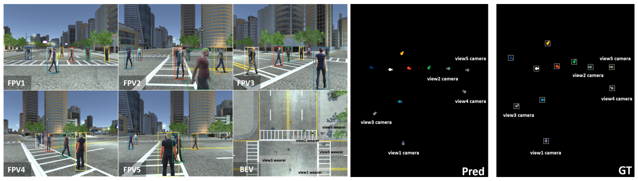


Figure 5. In this case, six people are standing in a row with serious occlusion. Our method makes full use of information from complementary views to finish the task of registration.

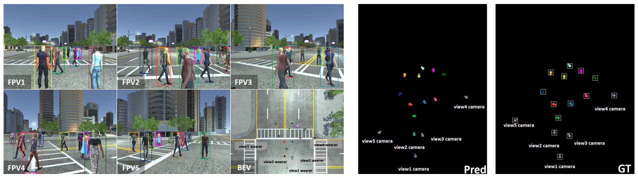


Figure 6. This case is a dense crowd scene, and the camera is located very close to the crowd, and some of the camera wearers are part of the crowd.

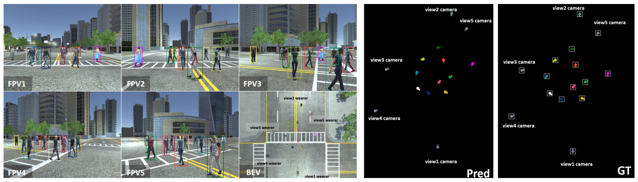


Figure 7. In this case, camera wearer2 is standing opposite to camera wearer1 and camera wearer4 is standing opposite to camera wearer5, which is the most difficult case of the camera registration task.

## References

[1] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Perceiving humans: from monocular 3D localization to social distancing. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7401–7418, 2021. 2, 3

[2] Ruize Han, Yiyang Gan, Jiacheng Li, Feifan Wang, Wei Feng, and Song Wang. Connecting the Complementary-View Videos: Joint Camera Identification and Subject Association. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2416–2425, 2022. 2

[3] Yunzhong Hou and Liang Zheng. Multiview detection with shadow transformer (and view-coherent data augmentation). In *Proceedings of the ACM International Conference on Multimedia*, pages 1673–1682, 2021. 2, 3

[4] Yunzhong Hou, Liang Zheng, and Stephen Gould. Multiview detection with feature perspective transformation. In *Proceedings of the European Conference on Computer Vision*, pages 1–18, 2020. 3

[5] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation verification for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5374–5384, 2022. 2

[6] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[7] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. 2

[8] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021.

[9] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. R2former: Unified retrieval and reranking transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19370–19380, 2023. 2