

# Wide-Baseline Multi-Camera Calibration using Person Re-Identification

Yan Xu

Yu-Jhe Li

Xinshuo Weng

Kris Kitani

Carnegie Mellon University

{yxu2, yujheli, xinshuw, kkitani}@cs.cmu.edu

## Abstract

We address the problem of estimating the 3D pose of a network of cameras for large-environment wide-baseline scenarios, e.g., cameras for construction sites, sports stadiums, and public spaces. This task is challenging since detecting and matching the same 3D keypoint observed from two very different camera views is difficult, making standard structure-from-motion (SfM) pipelines inapplicable. In such circumstances, treating people in the scene as “keypoints” and associating them across different camera views can be an alternative method for obtaining correspondences. Based on this intuition, we propose a method that uses ideas from person re-identification (re-ID) for wide-baseline camera calibration. Our method first employs a re-ID method to associate human bounding boxes across cameras, then converts bounding box correspondences to point correspondences, and finally solves for camera pose using multi-view geometry and bundle adjustment. Since our method does not require specialized calibration targets except for visible people, it applies to situations where frequent calibration updates are required. We perform extensive experiments on datasets captured from scenes of different sizes ( $80m^2$ ,  $350m^2$ ,  $600m^2$ ), camera settings (indoor and outdoor), and human activities (walking, playing basketball, construction). Experiment results show that our method achieves similar performance to standard SfM methods relying on manually labeled point correspondences.

## 1. Introduction

Our task is to solve the 3D camera pose estimation problem for multi-camera networks. We target challenging large-environment wide-baseline scenarios where cameras are static, sparse, and spaced far from each other (e.g., 10 to 20 meters). Conventionally, SfM [55] methods are often used to estimate camera pose. These methods first detect keypoints in the images of different camera views and describe the keypoints using hand-crafted or deep features [33, 10]. They then match the keypoint features [22] across views to obtain 2D-2D point correspondences and

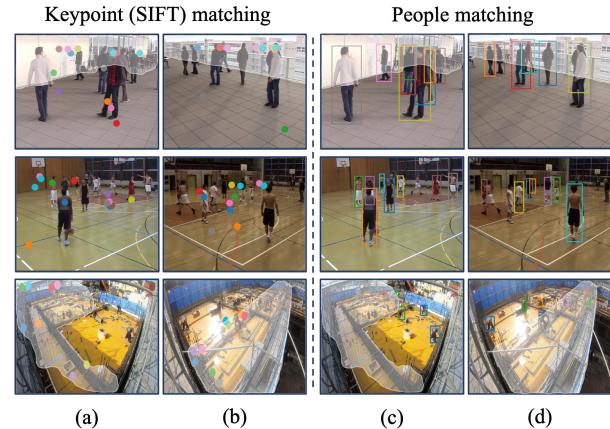


Figure 1: In large-environment, wide-baseline camera networks, backgrounds (top, white area), lighting conditions (middle), and textured areas (bottom, white area) from different camera views can vary massively, causing the failure of keypoint detection and matching ((a), (b)) in standard SfM methods. However, matching the same people across views can still be done correctly using re-ID methods ((c), (d)). (Points/bounding boxes of the same color correspond.)

solve the camera pose using multi-view geometry [19]. In our setting, due to the significant difference between camera poses, images taken from different cameras can have very different backgrounds, lighting conditions, and texture areas, as shown in Figure 1, making detecting and matching the same keypoints across camera views difficult.

Alternatively, we observe that if people are visible in the scene, we can obtain correspondences by detecting and matching people across different cameras. Especially for wide-baseline scenarios, matching the same people across camera views is easier than matching 2D keypoints since many human features can be used. For example, when people are close to the cameras, we can use the appearance information (e.g., height, clothes, length of hair) to match them across camera views (Figure 1). When people are far away from the cameras, we can still successfully match them from their temporal motion information (e.g., speed, smoothness of trajectory). Therefore, to obtain correspon-

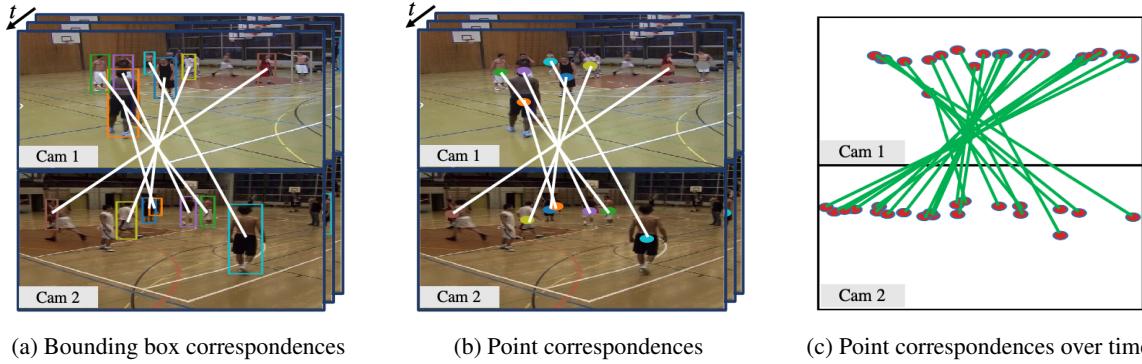


Figure 2: **Method overview.** Given a set of synchronized videos from different views and detected human bounding boxes, our method first associates bounding boxes across camera views using a re-ID network (a), then converts the bounding box correspondences to 2D-2D point correspondences (b). Finally, our method aggregates the point correspondences over time (c) and solves the camera pose with the point correspondences as input using multi-view geometry and bundle adjustment.

dences in wide-baseline scenarios, we can treat people as “keypoints” and associate their bounding boxes across camera views using a re-ID algorithm. Nevertheless, simply obtaining the associated bounding boxes is not enough to establish accurate 2D-2D correspondences necessary for camera pose estimation. As a second step, we can associate the same body parts (*e.g.*, head, foot, the position of body mass) inside the bounding boxes to further obtain point correspondences. In short, we can solve the feature matching problem by breaking it into a two-step process: First, matching people across camera views; Second, converting the bounding box correspondences to point correspondences. Since we assume that people are moving in the scene, which is often true, we can obtain a sufficient number of point correspondences for solving camera poses by aggregating the correspondences from a sequence of video frames.

We summarize our proposed wide-baseline camera pose estimation method in Figure 2. Our method includes three stages: 1) person matching, 2) point correspondence extraction, and 3) geometric camera pose estimation. Given a set of synchronized videos captured by cameras of different views, whose intrinsic and distortion parameters are given from the previous calibration, our method first associates person bounding boxes using a re-ID network pre-trained on open datasets [50, 60]. As a second step, our method converts the bounding box correspondences to point correspondences by extracting and associating the bounding box centers, which approximate the body mass positions. Finally, our method aggregates the point correspondences over time and solves for camera poses using a structure-from-motion pipeline (algebraic estimates of pose pairwise cameras, followed by non-linear optimization via bundle adjustment).

Our method only assumes the existence of visible moving people in the scene without the requirement of any other specialized calibration targets. It is thus suitable for

many situations where consistent camera pose estimation is required, *e.g.*, basketball training where cameras need to be moved for each new game or construction sites where cameras must be moved as the site is constructed. Moreover, our method does not require the re-ID model to output perfect association results since RANSAC [13] in the later stage can filter outliers. We evaluate our method on three datasets collected from scenes of different sizes ( $80m^2$ ,  $350m^2$ ,  $600m^2$ ) and lighting conditions (indoor and outdoor). The human postures in the three datasets also vary significantly, including walking, shooting, running, jumping, crouching, *etc.* We aim to use these datasets of various environment settings and human postures to evaluate the robustness of our method. Experiments show that our method achieves similar performance to a standard SfM pipeline which relies on manually labeled point correspondences.

Our contributions are as follows: 1) We propose to apply person re-ID algorithms to solve camera pose estimation for multi-camera networks. 2) We contribute a two-step process treating people as “keypoints” to obtain correspondences for wide-baseline scenarios. 3) Our method achieves an average accuracy of  $(0.4m, 1.08^\circ)$  across three datasets, comparable with SfM methods using manual annotation. 4) We perform extensive robustness and efficiency analysis for a more comprehensive understanding of our method.

## 2. Related work

Methods for solving the relative camera pose estimation problem can generally be categorized into geometric methods using the SfM pipeline and end-to-end deep pose regressors. We will discuss both categories of methods and briefly review recent deep re-ID works in this section.

**Geometric methods** address the camera pose estimation problem with a two-stage framework by first obtaining 2D-

2D point correspondences and then solving the camera pose using a geometric pipeline. Generally, they first detect the keypoints (Harris [18], FAST [42], etc.) in the images of different views and describe the keypoints with hand-crafted features (SIFT [33], BRIEF [5], ORB [43], etc.). Then, they match the keypoints (BFM [22], FLANN [37], etc.) across images to obtain point correspondences. Recently, many deep learning methods solve the keypoints detection and description simultaneously using neural networks (SuperPoint [10], UR2KiD [58], D2-net [12], LIFT [59], Lfnet [40], Elf [3]). Once obtaining 2D-2D point correspondences, these methods solve camera poses using a multi-view geometry pipeline. They first use the N-point algorithm [19, 28, 39], usually inside a RANSAC [13] loop, to solve for the Essential matrix, which will then be decomposed to the camera rotation and an up-to-scale camera translation [15]. As a final step, they use bundle adjustment [54] to further optimize the 3D poses of all cameras. There are also works [2, 9] using semantic information for solving SfM problems. These geometric methods are quite mature and generally accurate. However, they have difficulty in matching keypoint features across camera views when the distance between cameras is vast, as in our task. In this work, we use person matching to obtain correspondences to solve camera poses for wide-baseline scenarios.

**Deep pose regressor** was first applied in absolute camera pose estimation. PoseNet [25] is the first attempt that treats camera pose estimation as an end-to-end regression problem and solves the problem with a convolutional neural network [17], trained on data labeled using SfM [55]. Since then, many improvements on PoseNet have been proposed, including using different network architectures [57, 38, 34] and new loss designs [24, 27, 4]. Deep pose regressors for relative camera pose estimation [35, 27, 1, 11] have also been applied recently. In General, these methods input a pair of images into a Siamese network architecture [35] to extract deep features, from which they regress the camera pose. Despite the convenience of end-to-end regression, deep regressors still maintain a performance gap compared to geometric methods [46]. Moreover, deep pose regressors require images taken from moving cameras for training. In our task, the cameras are static, meaning that the training images and test images would be almost the same, making the deep pose regressors overfit to one camera pose. Therefore, deep pose regressors are inapplicable in our task.

**Deep re-identification** has been used for people matching in our method. Many existing deep re-ID algorithms, e.g., [31, 47, 48, 23, 8, 6, 7, 53, 52], are developed to address various challenges in re-ID problem, such as background clutter, viewpoint changes, and pose variations. For instance, Yang *et al.* [62] learn a camera-invariant subspace to deal with the style variations caused by different cameras. Liu *et*

*al.* [32] develop a pose-transferable framework based on generative adversarial network (GAN) [16] to yield pose-specific images for tackling pose variations. Several methods addressing background clutter leverage attention mechanisms to emphasize the discriminative parts [30, 51, 49]. In addition to these methods that learn global features, a few methods further utilize part-level information [53] to learn more fine-grained features, adopt human semantic parsing for learning local features [23], or derive part-aligned representations [52] for improving person re-ID. Following these works, we choose the most commonly used model [61] with ResNet-50 [21] as the backbone of our re-ID network.

### 3. Method

To solve the camera pose estimation problem for large-environment, wide-baseline scenarios, we propose to treat people as “keypoints” and use person re-ID for obtaining 2D-2D correspondences. Figure 3 presents the system pipeline of our method, which includes three modules: 1) “keypoint” matching, 2) point correspondence extraction, and 3) the geometric camera pose (GeoPose) solver.

#### 3.1. “Keypoint” Matching Using Person Re-ID

The first module of our method is “keypoint” matching. We adopt the re-ID model in the previous work [61] using ResNet-50 [21] as the backbone. As shown in Figure 3, the learning of the re-ID model is guided by a person id classification loss  $\mathcal{L}_{id}$  and a discriminative triplet loss  $\mathcal{L}_{tri}$ . **Training:** At the training stage, we have an image set  $X = \{x_i\}_{i=1}^N$  and its corresponding label set  $Y = \{y_i\}_{i=1}^N$  with size  $N$ , where  $x_i \in R^{H \times W \times 3}$  and  $y_i \in \mathbb{R}$ . We first employ the classification loss  $\mathcal{L}_{id}$  by computing the negative log-likelihood between the predicted label  $\tilde{y} \in \mathbb{R}^K$  and the ground truth  $\hat{y} \in \mathbb{N}^K$ :

$$\mathcal{L}_{id} = -\mathbb{E}_{(x,y) \sim (X,Y)} \sum_{k=1}^K \hat{y}_k \log(\tilde{y}_k) \quad (1)$$

where  $K$  is the number of identities (classes).

To further enhance the discriminative property, we impose a triplet loss  $\mathcal{L}_{tri}$ , which maximizes the inter-class discrepancy while minimizing intra-class distinctness. Specifically, for each input image  $x$ , we sample a positive image  $x_{pos}$  with the same identity label and a negative image  $x_{neg}$  with different identity labels to form a triplet tuple. The distances between  $x$  and  $x_{pos}/x_{neg}$  can be computed as

$$d_{pos} = \|f_x - f_{x_{pos}}\|_2 \quad (2)$$

$$d_{neg} = \|f_x - f_{x_{neg}}\|_2 \quad (3)$$

where  $f_x$ ,  $f_{x_{pos}}$ , and  $f_{x_{neg}}$  represent the feature vectors of images  $x$ ,  $x_{pos}$ , and  $x_{neg}$ , respectively. We then can have the triplet loss  $\mathcal{L}_{tri}$  defined as

$$\mathcal{L}_{tri} = \mathbb{E}_{(x,y) \sim (X,Y)} \max(0, m + d_{pos} - d_{neg}) \quad (4)$$

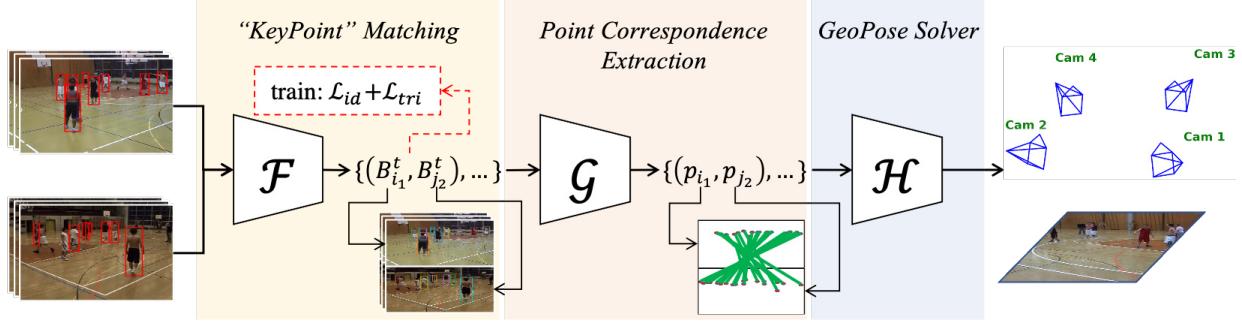


Figure 3: **System pipeline.** Our method includes three stages. First, we use a re-ID network ( $\mathcal{F}$ ) to associate person bounding boxes across camera views. Second, we use a function ( $\mathcal{G}$ ) to convert bounding box correspondences to point correspondences. Finally, we solve the camera pose using a GeoPose solver ( $\mathcal{H}$ ).  $B_{i_1}^t$  is the  $i$ -th bounding box of camera 1 at frame  $t$ ,  $p_{i_1}$  is the  $i$ -th keypoint of camera 1, the definitions of  $B_{j_2}^t$  and  $p_{j_2}$  are similar to  $B_{i_1}^t$  and  $p_{i_1}$ , respectively.

where  $m > 0$  is the margin used to define the difference between the distance of positive image pair  $d_{\text{pos}}$  and the distance of negative image pair  $d_{\text{neg}}$ .

In this work, we allow, to a certain extent, the imperfection of the re-ID model and use RANSAC in the GeoPose solver to reject the mis-associations. We pre-train the re-ID model on open datasets [50, 60]. For the relatively easier datasets, *Terrace* and *Basketball* [14], we directly apply the pre-trained model for inference. We also collected a more challenging dataset, *ConstructSite*, on which we fine-tune the pre-trained re-ID model first before inference.

**Inference:** At the inference stage, we use the re-ID model to associate bounding boxes across cameras. In this work, we assume that person trajectories in each video are provided. To extract re-ID feature of a person bounding box  $b_i$  in frame  $i$ , we first apply the restricted random sampling strategy [29] on the whole tracklet  $\mathcal{T}$  (a sequence of bounding boxes) of this person to obtain a smaller tracklet  $\mathcal{T}_i = \{b_k\}_{k=1}^8$ . We then extract the re-ID feature from each bounding box in  $\mathcal{T}_i$  and do a max-pooling to obtain the feature representation of  $b_i$ . Finally, we apply the Hungarian algorithm [26] to match bounding boxes across cameras.

### 3.2. Point Correspondence Extraction

After obtaining bounding box correspondences, our next step is to find 2D-2D point correspondences from the bounding box correspondences, such that the matched point correspondences are semantically meaningful. The intuitive idea is to associate the same body part (*e.g.* head, foot, the center of body mass) inside the bounding boxes. Our idea leverages the critical observation that the bounding box center can serve as a rough estimate of the body mass position.

We visualize in Figure 4 the bounding box centers and the centers of body mass of various human poses. The visualized person poses include people walking, sporting (running, jumping, shooting a basketball *etc.*), and working (bending, carrying, crouching, *etc.*), captured from differ-

ent camera heights and view angles. As Figure 4 shows, the bounding box center is close to the center of body mass under different camera poses and human postures. We thus use the bounding box center to represent the center of body mass and associate them across camera views as 2D-2D point correspondences. However, such an approximation would not be perfect. There will be an offset between the position of the bounding box center and the center of body mass, meaning that the obtained point correspondences will be noisy. Our method uses RANSAC [13] to reject the correspondences with large position offsets. For other point correspondences, our method treats their position offsets as noise and optimizes over all the point correspondences using bundle adjustment [54] to minimize the impact.

Formally, we define a bounding box as  $[u_{tl}, v_{tl}, u_{br}, v_{br}]$ , in which  $[u_{tl}, v_{tl}]$  and  $[u_{br}, v_{br}]$  represent the top-left and bottom-right corners of the bounding box. We can then obtain the position of the body mass center  $[u, v]$  as :

$$[u, v] = [(u_{tl} + v_{br})/2, (u_{tl} + v_{br})/2] \quad (5)$$

### 3.3. Geometric Camera Pose (GeoPose) Solver

With the 2D-2D point correspondences, our final step is to solve the camera pose with our GeoPose solver. Figure 5 shows the diagram of the GeoPose solver. We first solve the relative pose for each camera pair with a five-step pipeline, then optimize all camera poses using a global bundle adjustment. Note that we assume the camera intrinsic and distortion parameters are provided. Moreover, traditional automatic feature matching for resectioning fails in our challenging setting, where cameras are sparse and spaced far from each other. We thus manually specify the reference camera (camera 0, or  $C_0$ ) to lower the challenge.

Formally, given a set of point correspondences  $(p_i, p_0)$  of camera pair  $(C_i, C_0)$ , we first solve the essential matrix  $E_{i0}$  inside a RANSAC [13] loop. We then decompose  $E_{i0}$  into the relative rotation matrix  $R_{i0}$  and an up-to-scale relative translation  $t_{i0}$ . Next, we triangulate the 3D points  $P_{i0}$



Figure 4: **Body mass positions (green)** and **bounding box centers (red)**. We present samples from three datasets of variant camera poses and human postures and observe that bounding box centers can approximate body mass positions.

using  $(\mathbf{p}_i, \mathbf{p}_0)$  and  $\{\mathbf{R}_{i0}, \mathbf{t}_{i0}\}$ . After that, we use a local bundle adjustment step to jointly optimize the camera pose  $\{\mathbf{R}_{i0}^1, \mathbf{t}_{i0}^1\}$  and 3D points  $\mathbf{P}_{i0}^1$  by minimizing the 2D reprojection error of points from both  $C_i$  and  $C_0$ . We then use the prior 3D knowledge  $\gamma$  to solve the scale ambiguity and obtain  $\{\mathbf{R}_{i0}^2 = \mathbf{R}_{i0}^1, \mathbf{t}_{i0}^2\}$  and  $\mathbf{P}_{i0}^2$ . The above steps can be mathematically represented as:

$$\mathbf{E}_{i0} = \text{essenMatSolver}(\mathbf{p}_i, \mathbf{p}_0) \quad (6)$$

$$\mathbf{R}_{i0}, \mathbf{t}_{i0} = \text{decomposition}(\mathbf{E}_{i0}) \quad (7)$$

$$\mathbf{P}_{i0} = \text{triangulation}(\mathbf{p}_i, \mathbf{p}_0, \mathbf{R}_{i0}, \mathbf{t}_{i0}) \quad (8)$$

$$\mathbf{R}_{i0}^1, \mathbf{t}_{i0}^1, \mathbf{P}_{i0}^1 = \text{localBA}(\mathbf{p}_i, \mathbf{p}_0, \mathbf{R}_{i0}, \mathbf{t}_{i0}, \mathbf{P}_{i0}) \quad (9)$$

$$\mathbf{R}_{i0}^2, \mathbf{t}_{i0}^2, \mathbf{P}_{i0}^2 = \text{ambiguitySolver}(\mathbf{R}_{i0}^1, \mathbf{t}_{i0}^1, \mathbf{P}_{i0}^1, \gamma) \quad (10)$$

After solving the relative poses for each camera pair, we use a global bundle adjustment to further optimize the 3D poses of all cameras. If a 3D point is only visible from one camera pair, we directly use its 3D coordinates as the input of the global bundle adjustment. Otherwise, we take the mean of the coordinates solved from different camera pairs and set the mean to be the coordinate of the 3D point. We call this process “merge 3D points”, from which we obtain  $\hat{\mathbf{P}}$ . Finally, we optimize the camera poses using the global bundle adjustment initialized from  $\{\mathbf{R}_{i0}^2, \mathbf{t}_{i0}^2\}$  and  $\hat{\mathbf{P}}$ :

$$\hat{\mathbf{P}} = \text{merge3DPoints}(\{\mathbf{P}_{i0}^2\}) \quad (11)$$

$$\{\mathbf{R}_{i0}^*, \mathbf{t}_{i0}^*\} = \text{globalBA}(\{\mathbf{R}_{i0}^2, \mathbf{t}_{i0}^2\}, \hat{\mathbf{P}}) \quad (12)$$

in which,  $\{\mathbf{R}_{i0}^*, \mathbf{t}_{i0}^*\}$  is the final estimated camera pose.

We explain here the 3D information we use to solve the scale ambiguity. For *Terrace* dataset, we use the assumption that the height of a person is 1.75m [41]. For the *Basketball* dataset, we use the length of a standard free throw line (3.6m) [20]. For our *ConstructSite* dataset, we use the length of a standard construction steel pipe (1.0m).

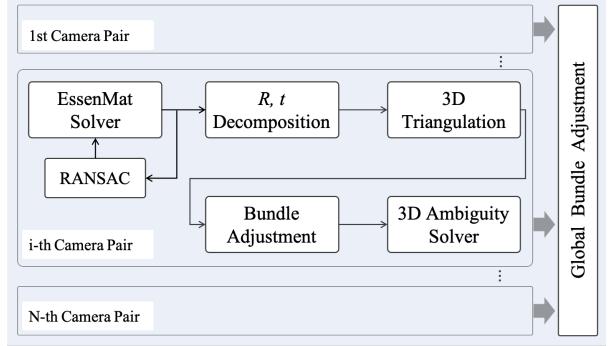


Figure 5: **GeoPose solver**. We first solve the relative camera pose for each camera pair using a five-step process, then optimize all camera poses with a global bundle adjustment.

## 4. Experiment

We report the evaluation results of our method in this section. We first describe datasets in Section 4.1 and evaluation metrics and baselines in Section 4.2. Following in Section 4.3, we present both quantitative and qualitative evaluation results. Next, we analyze the method robustness and efficiency in Section 4.4. Finally, we apply our method in tracking and present the result in Section 4.5.

### 4.1. Datasets

We evaluate our method on three datasets captured from scenes of different sizes, camera settings, and human poses. The camera intrinsic and distortion parameters are provided, and the videos are synchronized for all datasets.

**Terrace** [14] is an outdoor dataset shot on a terrace outside a building. Up to 7 people evolve in front of 4 DV cameras for 6 minutes 14 seconds. The frame rate is 25 fps, and the video resolution is  $720 \times 576$ . The size of the scene is around  $7m \times 11m$ , and the cameras are about  $2m$ -high from the ground plane. People are walking at slow speeds.

**Basketball** [14] is an indoor dataset filmed at a training session of a local basketball team. It was acquired at a basketball court with 4 DV cameras at 25 fps. The cameras are about  $2m$ -high. The videos are 2 minutes and 57 seconds long, with a resolution of  $720 \times 576$ . The size of the scene is about  $17.5m \times 22m$ . Up to 14 people are doing different activities, including running, jumping, shooting a ball, etc.

**ConstructSite** is a new dataset collected by our research collaborators using 4 synchronized GoPro HERO7 Black cameras around an outdoor construction site. The cameras are about  $3m$ -high from the floor. The videos are 2 minutes and 57 seconds long, the resolution is  $1352 \times 760$ , and the frame rate is 60 fps. The scene size is about  $22m \times 28m$ . There are about 20 people in similar suits doing construction works, including standing, crouching, carrying, etc.

Method	Camera Pose Error (CPE, (mm, °)) ↓			Mean ↓
	Terrace	Basketball	ConstructSite	
SIFT [33] + BFM [22]	4599mm, 55.03°	6140mm, 65.45°	10770mm, 45.66°	7170mm, 55.38°
SuperPoint [10] + BFM [22]	358mm, 54.68°	20065mm, 53.97°	2530mm, 50.10°	7651mm, 52.92°
WxBS [36]	1302mm, 54.14°	8.2e <sup>4</sup> mm, 57.01°	7.8e <sup>3</sup> mm, 48.91°	3.0e <sup>4</sup> mm, 53.35°
SuperPoint [10] + SuperGlue [44]	9934mm, 36.96°	1.2e <sup>4</sup> mm, 29.07°	1722mm, 24.08°	7885mm, 30.04°
Oracle(Manual-pts)	390mm, 1.18°	<b>358mm, 0.66°</b>	591mm, 2.08°	446mm, 1.31°
Ours (Manual-bbox)	<b>308mm, 0.52°</b>	490mm, 0.85°	<b>485mm, 1.80°</b>	427mm, <b>1.06°</b>
Ours (ReID-bbox)	308mm, 0.52°	410mm, 0.88°	493mm, 1.85°	<b>404mm, 1.08°</b>

Table 1: **Camera Pose Error (CPE).** We report the camera position and orientation prediction errors. “Manual-pts” denotes manually annotated 2d point correspondences, “Manual-bbox” denotes manually associated bounding box correspondences, “ReID-bbox” denotes associating bounding box correspondences using re-ID network. (The terms are the same for the following tables unless explicitly stated.) For each dataset, we report the mean position and orientation errors of all cameras.

Method	Re-Projection Error (RPE, pixel) ↓			Error Resolution Ratio(ERR, %) ↓		
	Terrace	Basketball	ConstructSite	Terrace	Basketball	ConstructSite
SIFT [33] + BFM [22]	254.40	533.15	177.90	44.17%	92.56%	23.41%
SuperPoint [10] + BFM [22]	53.07	9.50	130.65	9.21%	1.65%	17.19%
WxBS [36]	60.09	80.21	18381.56	10.43%	13.93%	2418.63%
SuperGlue [44] + SuperPoint [10]	36.57	11.03	96.38	6.35%	1.91%	12.68%
Oracle(Manual-pts)	<b>0.45</b>	<b>0.51</b>	<b>13.45</b>	<b>0.08%</b>	<b>0.09%</b>	<b>1.78%</b>
Ours (Manual-bbox)	2.30	0.88	46.26	0.40%	0.15%	6.09%
Ours (ReID-bbox)	2.30	1.01	45.10	0.40%	0.18%	5.93%

Table 2: **Re-projection error (RPE) and error resolution ratio (ERR).** We report the re-projection error, measured by *pixel*, and error resolution ratio, defined as  $\frac{RPE}{\min(H,W)}$ , where  $(H, W)$  is the video resolution. Except for the oracle baseline, our method outperforms all the other baselines that use different hand-crafted/deep features and different matching methods.

## 4.2. Evaluation Metrics and Baselines

**Evaluation Metrics** We use three metrics to measure the performance of our method: (1) Camera pose error (CPE), (2) Re-projection error (RPE), and (3) Error resolution ratio (ERR). CPE includes the location error and the orientation error. The location error is the Euclidean distance between the estimated camera location and the ground truth (GT) camera location. The orientation error is the smallest Euler angle to align the estimated orientation and the GT orientation. RPE reports the mean re-projection error (by pixel) on 15 pairs of 2D-2D point correspondences that we annotate for each dataset. ERR reports the ratio between RPE and the video resolution. ERR (relative) provides a more comprehensive evaluation together with RPE (absolute).

**Baselines** We compare our method with the following

baselines. For the first baseline, we detect keypoints from images, use SIFT [33] feature as the descriptor, and match the keypoints across cameras using Brute-Force Matching [22] (BFM). We then use our GeoPose solver to solve the camera pose. For the second baseline, we use SuperPoint [10] network to detect and describe the keypoints simultaneously. We then apply the BFM matching and the GeoPose solver. The third baseline is WxBS [36] which uses the idea of view synthesis for wide-baseline matching. In our setting, “x” means “geometry” as defined in the literature. The fourth baseline is SuperGlue [44], a method that matches two sets of local features using attention mechanism [56] and graph neural networks [45]. We use SuperPoint [10] as the feature detector following the paper. Our last baseline is an oracle baseline. Specifically, we manually annotate point correspondences and solve camera poses us-



Figure 6: **Correspondence matching.** When the distance between a point and the epipolar line computed from its correspondence is less than a threshold (5 pixels), we consider the correspondence as correct and mark the matching as green. The incorrect matchings are marked with red. Our method performs well under the wide-baseline setting while the baselines fail.

ing our GeoPose solver. We aim to use the oracle baseline to measure the performance gap between our method and standard SfM methods using manual annotations.

#### 4.3. Comparison with Baselines

**Quantitative results** We present the result of CPE in Table 1 and the result of RPE and ERR in Table 2. We have the following observations: (1) All the baselines fail to predict reasonable camera poses on any of the three datasets under our challenging experiment condition. (2) Our method outperforms all the baselines except for the oracle on all three evaluation metrics. (3) Our method achieves comparable performance with the oracle on the relatively easier datasets (*Terrace* and *Basketball*). (4) Our method performs somewhat worse than the oracle on *ConstructSite*, which is a challenging dataset since all the workers wear the same suit, making re-ID mis-associations more likely to happen. Since the oracle baseline is an SfM pipeline using manual annotations, its performance is the best one can achieve. Even though our method does not outperform the oracle on *ConstructSite*, the result is still encouraging, especially considering the challenging wide-baseline setting and the fact that our method does not require manual annotations.

**Qualitative results** We present in Figure 6 the correspondences matching results using our method and the baselines except for the oracle. We use the distance between a point and the epipolar line (not the re-projection distance) computed from its correspondence to measure the correctness of this correspondence. If the distance is smaller

Noise	Camera Pose prediction error (CPE) ↓		
	<i>Terrace</i>	<i>Basketball</i>	<i>ConstructSite</i>
N=0	308mm, 0.52°	410mm, 0.88°	493mm, 1.85°
N=1	339mm, 0.65°	454mm, 1.03°	489mm, 1.86°
N=2	352mm, 1.00°	466mm, 0.95°	477mm, 1.83°
N=5	407mm, 1.50°	532mm, 1.76°	537mm, 1.86°
N=10	433mm, 1.91°	1653mm, 26.20°	601mm, 1.85°
N=20	1692mm, 28.49°	3093mm, 49.20°	867mm, 2.31°
N=50	4675mm, 47.80°	3946mm, 64.70°	2395mm, 23.69°

Table 3: **Noised bounding boxes.** “N=i” means adding uniformly distributed ( $[-i, i]$  pixels) random noise to the top-left and bottom-right corners of the bounding box. We report camera pose errors with different noise strengths.

than a certain threshold, we treat the correspondence as correct. We set the threshold to be 5 pixels and observe that the baselines are not able to correctly match correspondences in wide-baseline scenarios while our method works well.

#### 4.4. Robustness and Efficiency Analysis

**Noised bounding boxes** To evaluate the robustness of our method to the imperfect bounding boxes, we add noises to the person bounding boxes and report camera pose errors on all the three datasets in Table 3. We observe that: (1) Our method demonstrates a certain level of noise robustness on all the datasets. (2) For the larger scene (*ConstructSite*) where people are away from the camera, our method shows

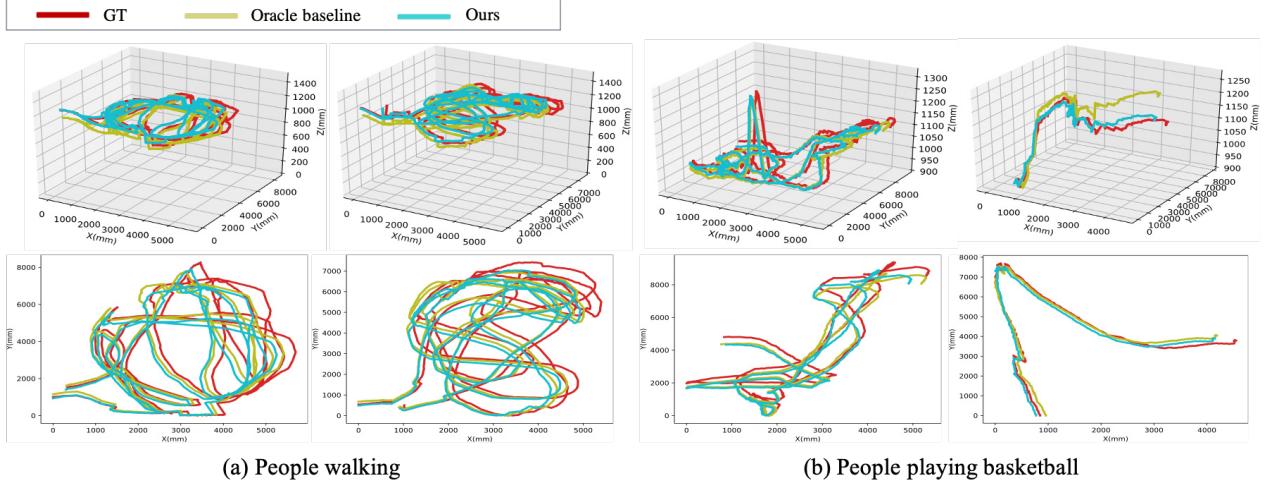


Figure 7: **Tracking results.** We track four people in different motion statuses (walking, playing basketball) and visualize their 3D trajectories and 2D bird’s-eye view trajectory. We compare our method with the ground truth (GT) trajectory and the oracle baseline, which solves camera poses using manually annotated 2D-2D point correspondences. Our method performs similar to the oracle baseline, and the average trajectory difference are  $0.28m$ ,  $0.26m$ ,  $0.59m$ , and  $0.36m$  (from left to right).

better noise robustness. We give our understanding here. Imagine that there is a ray from the camera to the body mass position in the 3D space. When the person is away from the camera, a change in the person position will only cause a small direction (or angle) change on the ray. The same amount of position change when the person is close to the camera will lead to a larger change in the ray angle.

**Number of correspondences** We present in Figure 8 the plots of CPE vs number of person correspondences used for estimating camera poses. We observe that our method converges fast as the number of person correspondences increases, and it reaches a good performance with 30 correspondences for all datasets. Considering that we use RANSAC and compute the Jacobian matrix in BA in our GeoSolver, using more correspondences is more expensive. Our method only requires a small number of correspondences to reach good performance. The cost is low.

#### 4.5. Application on Tracking

In this experiment, we apply our camera pose estimation method in the tracking task for estimating the trajectory of a moving person. We first solve the camera poses using our method. Next, we specify an object person and estimate the 3D body mass positions of the object person over time using the solved camera poses. Lastly, we take the mean of the 3D coordinates solved from all camera pairs as the final estimation of the body mass location. Figure 7 shows the trajectories (body mass location over time) of four people from both 3D view and 2D bird’s eye view. We observe that our method consistently gives good trajectory estimation over time (3 mins for walking people, 40s for sporting people). The difference between the trajectory from our method and the trajectory from the oracle is less than  $0.6m$

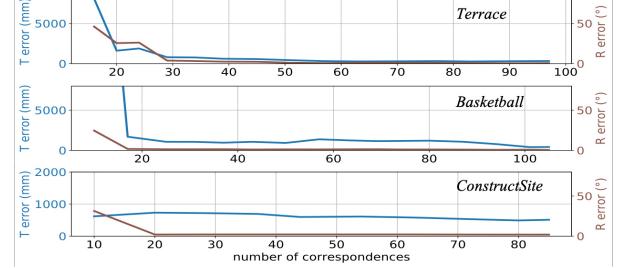


Figure 8: **CPE vs number of correspondences.** We observe that our method converges with the requirement of less than 30 correspondences for all three datasets.

even for intense body motions like playing basketball.

## 5. Conclusion

In this work, we studied the camera pose estimation problem for large-area, wide-baseline camera networks. We contribute a method that treats people as “keypoints” and applies a re-ID network to obtain 2D-2D point correspondence for solving camera poses. We evaluated our method on datasets of diverse camera settings and person postures, and our method achieves comparable performance with SfM methods relying on manual annotations. We also provided extensive robustness, efficiency, and applicability analysis. There are still many aspects that need to be explored, such as how to improve robustness towards the imperfection of the bounding boxes and how to use other objects, e.g., cars, when people are invisible from the scene.

**Acknowledgement:** We thank SHIMIZU CORPORATION for the sponsorship and data creation and Vivek Roy and Zhengyi Luo for their discussion and help on our work.

## References

- [1] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–767, 2018.
- [2] Sid Yingze Bao and Silvio Savarese. Semantic structure from motion. In *CVPR 2011*, pages 2025–2032. IEEE, 2011.
- [3] Assia Benbihi, Matthieu Geist, and Cedric Pradalier. Elf: Embedded localisation of features in pre-trained cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7940–7949, 2019.
- [4] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018.
- [5] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.
- [6] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. 2018.
- [7] Dapeng Chen, Dan Xu, Hongsheng Li, Nicu Sebe, and Xiaogang Wang. Group consistent similarity learning via deep crf for person re-identification. 2018.
- [8] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. 2016.
- [9] Amaury Dame, Victor A Prisacariu, Carl Y Ren, and Ian Reid. Dense reconstruction using 3d object shape priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1288–1295, 2013.
- [10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.
- [11] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera re-localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2871–2880, 2019.
- [12] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019.
- [13] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [14] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):267–282, 2007.
- [15] Georgi Hristov Georgiev and Vencislav Dakov Radulov. A practical method for decomposition of the essential matrix. *Applied mathematical sciences*, 8(176):8755–8770, 2014.
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. 2014.
- [17] Juxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.
- [18] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.
- [19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [20] Zsolt Hartyani. Official basketball rules, 2006.
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [22] Amila Jakubović and Jasmin Velagić. Image feature matching and object detection using brute-force matchers. In *2018 International Symposium ELMAR*, pages 83–86. IEEE, 2018.
- [23] Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmén, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. 2018.
- [24] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017.
- [25] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.
- [26] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [27] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 929–938, 2017.
- [28] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 630–633. IEEE, 2006.
- [29] Shuang Li, Slawomir Bak, Peter Carr, and Xiaogang Wang. Diversity regularized spatiotemporal attention for video-based person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 369–378, 2018.
- [30] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. 2018.

- [31] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, and Yi Yang. Improving person re-identification by attribute and identity learning. *arXiv*, 2017.
- [32] Jinxian Liu, Bingbing Ni, Yichao Yan, Peng Zhou, Shuo Cheng, and Jianguo Hu. Pose transferrable person re-identification. 2018.
- [33] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [34] Iaroslav Melekhov, Juha Ylloinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 879–886, 2017.
- [35] Iaroslav Melekhov, Juha Ylloinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687. Springer, 2017.
- [36] Dmytro Mishkin, Jiri Matas, Michal Perdoch, and Karel Lenc. Wxbs: Wide baseline stereo generalizations. *arXiv preprint arXiv:1504.06603*, 2015.
- [37] Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 2009.
- [38] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530. IEEE, 2017.
- [39] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.
- [40] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *Advances in neural information processing systems*, pages 6234–6244, 2018.
- [41] Max Roser, Cameron Appel, and Hannah Ritchie. Human height. *Our world in data*, 2013.
- [42] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.
- [43] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.
- [44] 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.
- [45] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.
- [46] Yoli Shavit and Ron Ferens. Introduction to camera pose estimation with deep learning. *arXiv preprint arXiv:1907.05272*, 2019.
- [47] Yantao Shen, Hongsheng Li, Tong Xiao, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Deep group-shuffling random walk for person re-identification. 2018.
- [48] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Person re-identification with deep similarity-guided graph neural network. 2018.
- [49] Jianlou Si, Honggang Zhang, Chun-Guang Li, Jason Kuen, Xiangfei Kong, Alex C Kot, and Gang Wang. Dual attention matching network for context-aware feature sequence based person re-identification. 2018.
- [50] Francesco Solera, Simone Calderara, Ergys Ristani, Carlo Tomasi, and Rita Cucchiara. Tracking social groups within and across cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):441–453, 2016.
- [51] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. 2018.
- [52] Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoungh Mu Lee. Part-aligned bilinear representations for person re-identification. 2018.
- [53] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). 2018.
- [54] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.
- [55] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.
- [56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
- [57] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving deeper into convolutional neural networks for camera relocalization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651. IEEE, 2017.
- [58] Tsun-Yi Yang, Duy-Kien Nguyen, Huub Heijnen, and Vassileios Balntas. Ur2kid: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision. *arXiv preprint arXiv:2001.07252*, 2020.
- [59] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.
- [60] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.
- [61] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. In *arXiv preprint*, 2016.
- [62] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. 2018.