Learning Bipartite Graph Matching for Robust Visual Localization

Hailin Yu[1,2] Weicai Ye[1] Youji Feng[2] Hujun Bao[1] Guofeng Zhang[1]*

[1]State Key Lab of CAD&CG, Zhejiang University[†]
{hailinyu, yeweicai}@zju.edu.cn, {bao, zhangguofeng}@cad.zju.edu.cn

[2]SenseTime Research
fengyouji@sensetime.com

Abstract

2D-3D matching is an essential step for visual localization, where the accuracy of the camera pose is mainly determined by the quality of 2D-3D correspondences. The matching is typically achieved by the nearest neighbor search of local features. Many existing works have shown impressive results on both the efficiency and accuracy. Recently emerged learning-based features further improve the robustness compared to the traditional hand-crafted ones. However, it is still hard to establish enough correct matches in challenging scenes with illumination changes or repetitive patterns due to the intrinsic local properties of local features. In this work, we propose a novel method to deal with 2D-3D matching in a very robust way. We first establish as many potential correct matches as possible using the local similarity. Then we construct a bipartite graph and use a deep neural network, referred to as Bipartite Graph Network (BGNet), to extract the global geometric information. The network predicts the likelihood of being an inlier for each edge and outputs the globally optimal one-to-one correspondences with a Hungarian pooling layer. The experiments show that the proposed method can find more correct matches and improves localization on both the robustness and accuracy. The results on multiple visual localization datasets are obviously better than the existing state-of-the-arts, which demonstrates the effectiveness of the proposed method.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality; Computing methodologies—Artificial intelligence—Computer vision—Computer vision problems

1 Introduction

Visual localization aims to estimate the 6-Degree-of-Freedom (6DoF) camera pose from a single image, which is a fundamental technique for many applications, such as augmented reality, autonomous driving, and mobile robotics.

A typical pipeline of visual localization firstly performs feature matching between 2D local features in a query image and 3D points in an offline reconstructed 3D model, and then estimates the 6DoF camera pose from the established 2D-3D correspondences by solving a standard Perspective-n-Points (PnP) problem [16,19]. As there are possible erroneous correspondences, the RANSAC [12] algorithm is often used to filter outliers. The accuracy of the estimated pose is closely related to the quantity of the retained correct matches. In general, the greater it is, the more accurate the pose would be. Thus many previous works [21,36,40,41] design delicate matching strategies to find more potential correspondences. However, due to the limited invariance and discriminative power of local features, a large number of erroneous matches may exist in challenge conditions such as day-night change, seasonal variation, and the inlier ratio would be too low for RANSAC to find the true matches. Accurate localization in these situations is still an open problem.

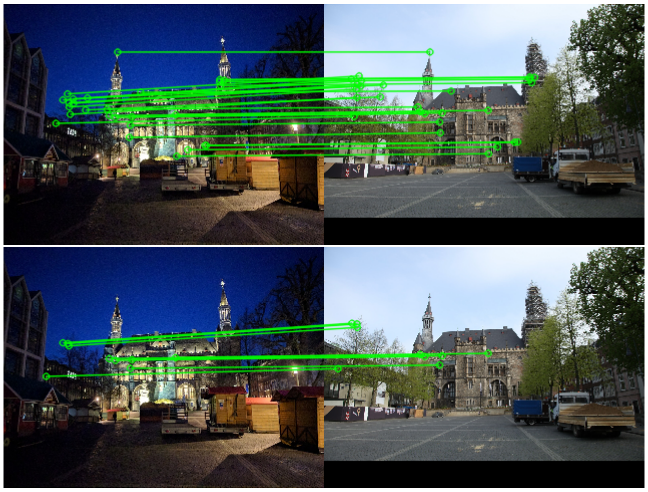

Figure 1: Comparison of the correct matches obtained by BGNet (the top row) and mutual check (the bottom row). On each row, the left is the query image, and the right is the retrieved nearest database image. All the correctly matched 3D points are projected onto this database image and the projections beyond the border are not displayed.

One way to improve the accuracy and robustness of visual localization is to enhance the discriminative power of local features. The most commonly used local feature in traditional localization is SIFT [22]. While performing well in normal circumstances, it is less effective in scenes with illumination or large viewpoint changes. ORB [30] is another popular feature in 3D vision tasks [27]. The simple binary representation makes it extremely efficient and well suited for mobile devices, but also degrades the performance. In the last few years, as convolutional neural networks (CNN) have emerged to be the most powerful tool for feature representation in many vision tasks such as detection and classification, a large number of CNN-based local features have also been proposed. They use deep networks to extract descriptors solely [23,24,42] or extract keypoints and descriptors simultaneously [10,11]. These features show impressive robustness against illumination variations and viewpoint changes and perform better than hand-crated ones on image matching. Some of them have already been used to improve localization [10,11]. However, there are many similar local patterns in real environments, e.g. the facades of modern buildings. Ambiguous matches would still be frequently encountered even using the powerful learning-based features. This problem will be more serious when the matching is performed on a large-scale scene. The problem can be alleviated by embedding some contextual information from the whole image into the features [23,24], but this will obscure the raw

*Corresponding author: Guofeng Zhang

[†]The authors from State Key Lab of CAD&CG, Zhejiang University are also affiliated with ZJU-SenseTime Joint Lab of 3D Vision. This work was partially supported by NSF of China (Nos. 61672457 and 61822310), and the Fundamental Research Funds for the Central Universities (No. 2019XZZX004-09).

representation of local details. Ratio test [22] and mutual check are commonly used to eliminate part of the ambiguities, with the price that the recall of true matches is dramatically reduced. Therefore, it is desirable to develop a more effective method to deal with these ambiguous matches.

End-to-End localization methods [6, 14, 44] proposed in recent years use deep neural networks to directly regress the 6DoF pose from a given query image. They do not suffer from the problems of local feature matching and are very robust in scenes with illumination changes or textureless regions. These methods behave more like image retrieval, and thus the accuracy is unsatisfactory, especially for various AR applications. Another type of learning-based methods [3, 4, 39] firstly regress the scene coordinates of the query image and then compute the camera pose using the PnP algorithm. These methods are more accurate than direct pose regression but are hard to converge in large-scale scenarios. Besides, the generalization ability of the learning-based methods remains a knotty problem. The requirement of a large amount of training data limits their scalability and applications.

In this work, we follow the traditional pipeline to perform visual localization. To improve the robustness and accuracy, we propose a neural network to deal with ambiguous 2D-3D matches. It leverages global geometric information to pick out the globally optimal matches from a set of candidates and can find more true inliers than the commonly used strategy like mutual check, as shown in Figure 1. Specifically, we first establish a set of many-to-many 2D-3D correspondences using the KNN search to reduce the false rejection and maintain the high recall. Then a bipartite graph is constructed from these correspondences. The vertices of the bipartite graph are represented by both the 2D points and the 3D points and the edges are represented by the 2D-3D correspondences. The network takes this graph as the input and predicts a weight for each edge. The weight indicates the likelihood that the corresponding edge is an inlier. The prediction of the weight completely depends on geometric information, i.e. the coordinates of the 2D points and the 3D points, so we refer to the weight as geometric prior. Since the true matches must be one-to-one, we introduce a Hungarian pooling layer on top of the network to find out the set of the globally optimal one-to-one matches with the largest sum of weights. These optimal matches are used for the final pose estimation. Being different from the end-to-end learning methods, the proposed method focuses on the learning of matching. It is flexible to combine other strategies in the traditional localization pipeline.

In this paper, we show that using a scene retrieval strategy along with the proposed network achieves state-of-the-art localization performances. Our major contributions are as follows:

- We propose a bipartite graph network, referred to as BGNet, to deal with 2D-3D matching. It is able to predict the geometric prior of each 2D-3D match and output the globally optimal match set.
- We propose a hierarchical visual localization method with a new scene retrieval strategy, which further improves the robustness of pose estimation.
- We show that the proposed method outperforms the existing state-of-the-art methods through extensive experiments.

2 More Related Works

In this section, we briefly review some of the methods related to this article.

Local Feature Matching. Local feature matching typically takes two steps: 1) establishing initial matches through Nearest Neighbor (NN) search, 2) selecting true correspondences. The most commonly used criteria include ratio test [22], distance threshold, and mutual check. If a geometric model, e.g. a fundamental matrix or an absolute pose, fits for the true matches, robust model estimation approaches, such as RANSAC [12] and PROSAC [8], can be used to find out the inliers. Some works use non-parametric methods to perform the selection. [2] leverages local consistency and uses neighborhood consensus to gather true matches. [26, 46] use a deep neural network to classify initial matches as inliers or outliers. They have shown impressive results on 2D-2D matching.

Graph Matching. Graph matching aims to establish correspondences between two graphs. It is usually formulated as a quadratic assignment problem which is NP-hard. For a bipartite graph, if candidate matches and the corresponding weights are given, some methods [9, 18] are able to find the global optimal solution with the maximum sum of weights. Recent works show that graph matching can also be solved by learning methods. [7] is the first work to leverage learning to accelerate the matching process. [45] uses deep architecture to improve the performance. SuperGlue [33] combines the feature learning and graph matching model into a single deep architecture to get more consistent results.

Visual Localization. The traditional visual localization methods can be roughly divided into two classes: the direct methods [21, 35, 36, 40] and the indirect methods [27, 31, 32, 41]. The former performs direct matching between local features in a query image and the 3D points in an SfM model. As the feature set in the database is typically very large, many previous works aim to improve the efficiency. [35] uses a large vocabulary to quantize the local features and prefers to match the features with lower cost in priority. [20] compresses the 3D model and ranks the 3D points according to the chances they might be seen. Direct matching may produce many false correspondences, some methods [21, 34, 36] use the constrain of co-visibility to obtain positive matches. The indirect method adopts a hierarchical paradigm to cope with large scale scenarios. Image retrieval is firstly performed to find similar images in the database and then the query features are matched with the points visible in the retrieved images. These methods are more robust if a superior image retrieval method [1] is provided, and are easy to integrate some location priors. Apart from the low-level local features, some approaches [17, 28, 43] also use the high-level semantic information to improve the robustness against seasonal changes or extreme illumination changes.

The learning-based methods can be classified as either directly regressing the camera pose [6, 14] or regressing scene coordinates [3, 4, 39]. PoseNet [14] is the first work that proposes to train a convolutional neural network to estimate the camera pose. Based on this pioneering work, many improvements on the accuracy are achieved through the novel design of network architectures or loss functions [6, 44]. SCoRe Forest [39] uses the regression forest to infer the coordinate of each pixel in the RGB-D image and then uses RANSAC+PnP to solve the camera pose. DSAC [3] proposes a differentiable RANSAC to train this process in an end-to-end way. [4] further improves the accuracy through a fully convolutional neural network for densely regressing scene coordinates.

3 Bipartite Graph Nearul Network

Given a set of initial 2D-3D correspondences which are many-to-many, a bipartite graph can be constructed with the 2D points and the 3D points being the vertices, the correspondences being the edges. Taking the graph as input, BGNet predicts the likelihood of being an inlier for each edge and outputs the optimal one-to-one matches. The details are described below.

3.1 Problem Fomulation

The bipartite graph is denoted as $\boldsymbol{G} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{E})$. $\boldsymbol{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M\}$ is the 2D point set, in which $\boldsymbol{u}_i = (x_i, y_i)$ is the 2D coordinate for the i-th point, where $1 \leq i \leq M$. $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_N\}$ is the 3D point set, and $\boldsymbol{v}_j = (X_j, Y_j, Z_j)$ is the 3D coordinate for the

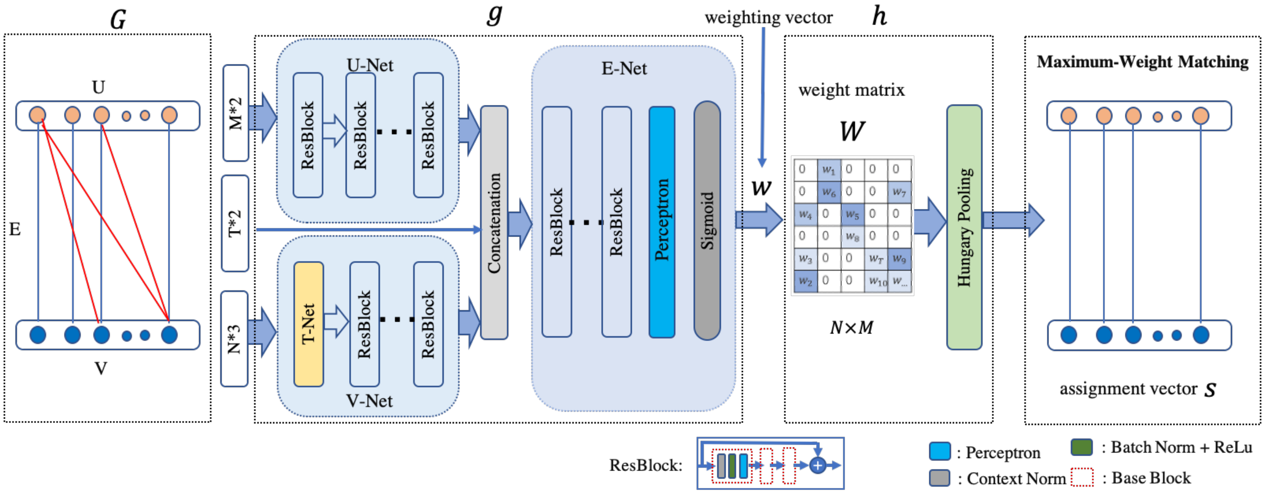

Figure 2: **BGNet Architecture.** The proposed network takes a bipartite graph as input, and outputs the maximum-weight matching and the corresponding probability of being an inlier for each selected correspondence. The input graph is composed of the 2D point set($\boldsymbol{U}$), the 3D point set($\boldsymbol{V}$), and the 2D-3D correspondence set($\boldsymbol{E}$) in which the inliers are displayed in blue and the outliers are displayed in red. Three subnetworks (U-Net, V-Net, and E-Net) are used to extract the geometric features for $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{E}$ respectively.

j-th point, where $1 \leq j \leq N$. $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_T\}$ is the edge set, where each edge $\boldsymbol{e}_k = (i, j)$ with $1 \leq k \leq T$ represents the correspondence between the i-th 2D point and the j-th 3D point. The bipartite graph network aims to predict a weight w_k for each edge $\boldsymbol{e}_k$ and then find the maximum-weight matching from $\boldsymbol{G}$. The weight w_k represents the likelihood that $\boldsymbol{e}_k$ is an inlier. The outputs of the network are formally written as a weighting vector $\boldsymbol{w} = (w_1, w_2, ..., w_T)$ with $w_k \in [0,1]$ and an assignment vector $\boldsymbol{s} = (s_1, s_2, ..., s_T)$ with $s_k \in \{0,1\}$ which indicates whether the k-th edge is contained in the maximum-weight matching.

The workflow of the network can be divided into two steps. The first step is to predict weighing vector $\boldsymbol{w}$,

$$\boldsymbol{w} = g(\boldsymbol{G}; \boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{\theta}$ is the learning parameters. The second step is to find out the maximum-weight matching result from $\boldsymbol{G}$ and $\boldsymbol{w}$, denoted as

$$\boldsymbol{s} = h(\boldsymbol{G}, \boldsymbol{w}), \tag{2}$$

where the function h is non-parametric. The final output $(\boldsymbol{w}, \boldsymbol{s}) = f_\theta(\boldsymbol{G})$ is obtained by combining the above two parts. The architecture of g and h is elaborated in the next section.

3.2 Network Architecture

This section describes the proposed deep architecture f_θ. The overall architecture is illustrated in Figure 2. The input is a bipartite graph $\boldsymbol{G}$ constructed from the 2D-3D correspondences established by feature matching.

Since the three parts, i.e. the two point sets $\boldsymbol{U}$ and $\boldsymbol{V}$, and the edge set $\boldsymbol{E}$, of the bipartite graph are all unordered sets, we use Perceptron [13] as the basic layers to extract the geometric features and Context Normalization [26] to aggregate the global information.

As shown in Figure 2, $g(\boldsymbol{G}; \boldsymbol{\theta})$ contains three sub-networks, respectively represented by U-Net, V-Net, and E-Net, respectively. The input to U-Net is the 2D point set $\boldsymbol{U}$, which can be represented as a $M \times 2$ matrix. U-Net embeds the M 2D point coordinates into M d_1-dimensional vectors, so the output of U-Net is a $M \times d_1$ matrix $\boldsymbol{X_u}$. Similarly, the input to V-Net is the 3D point set $\boldsymbol{V}$, which can be represented as a $N \times 3$ matrix. V-Net embeds the N 3D point coordinates into N d_2-dimensional vectors, and the output of V-Net is a $N \times d_2$ matrix $\boldsymbol{X_v}$. In order to make the 3D point set network robust to rotation, we add a T-net [29] on the bottom of the 3D point network. According to the edge set $\boldsymbol{E}$, we concatenate the corresponding 2D and 3D feature vectors to form the input to E-Net $\boldsymbol{X_e}$, which is a $T \times (d_1 + d2)$ matrix. For a specific edge $\boldsymbol{e_k} = (i, j)$, the feature vector input to E-Net can be obtained by:

$$\boldsymbol{X}_{\boldsymbol{e}}^k = \left[\boldsymbol{X}_{\boldsymbol{u}}^i || \boldsymbol{X}_{\boldsymbol{v}}^j\right], \tag{3}$$

where $\boldsymbol{X}_*^r$ denotes the r-th row of matrix $\boldsymbol{X}_*$ and $[\cdot||\cdot]$ denotes concatenation. The output of E-Net is a T-dimensional vector and a sigmoid layer on the top of E-Net is used to ensure that each value of the output vector $\boldsymbol{w}$ is in the range of $[0,1]$.

Hungarian Pooling. If we train $g(\boldsymbol{G}; \boldsymbol{\theta})$ directly, the network parameters will be very difficult to learn because the geometric consistency may conflict with the supervision. The conflict is shown in Figure 3. Two correspondences are close in the 3D space and the image spaces are nearly geometric consistent, i.e. they may both have small reprojection errors with the same camera pose. The network is prone to generate similar weights for them according to the extracted geometric features. However, only one of them is inlier. This discrepancy that the two correspondences have similar geometric features but with different labels makes the network hard to converge. Moreover, the output from $g(\boldsymbol{G}; \boldsymbol{\theta})$ is only the likelihoods to be inlier and still contain a large number of outliers.

To solve this problem, we introduce the Hungarian algorithm [18] into the network for end-to-end training. Hungarian algorithm can find the global optimal one-to-one matches. Because only one of the two correspondences is selected, the discrepancy between the geometric consistency and the supervision can be eliminated.

Based on the weight vector $\boldsymbol{w}$ predicted by $g(\boldsymbol{G}; \boldsymbol{\theta})$ and the bipartite graph $\boldsymbol{G}$, a weight matrix $\boldsymbol{W}$ is constructed as:

$$\boldsymbol{W}[e_k(0), e_k(1)] \leftarrow w_k, \tag{4}$$

where the unfilled elements of $\boldsymbol{W}$ is set to 0. Then the Hungarian algorithm is applied on this weight matrix $\boldsymbol{W}$ to get the maximum-weight matching $\boldsymbol{M}$. The assignment vector $\boldsymbol{s}$ is obtained by

$$s_k = \begin{cases} 1 & \boldsymbol{e_k} \in \boldsymbol{M}; \\ 0 & otherwise. \end{cases} \tag{5}$$

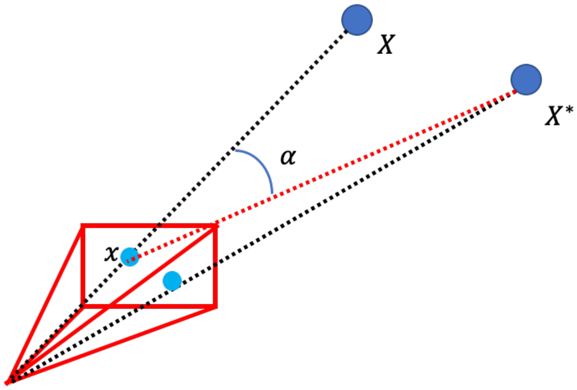

Figure 3: $\boldsymbol{X}$ and $\boldsymbol{X}^*$ are two 3D points, and $\boldsymbol{x}$ is the projection of $\boldsymbol{X}$ on the image plane. The correspondence $(\boldsymbol{X}^*, \boldsymbol{x})$ is an outlier established by feature matching. $(\boldsymbol{X}, \boldsymbol{x})$ is the correct match. When the angle α is small, two correspondences $(\boldsymbol{X}^*, \boldsymbol{x})$ and $(\boldsymbol{X}, \boldsymbol{x})$ should have similar weights from a geometric view, but from a learning view, the weight of correspondence $(\boldsymbol{X}^*, \boldsymbol{x})$ should be much smaller than correspondences $(\boldsymbol{X}, \boldsymbol{x})$ due to their respective label. This produces a conflict that makes the network hard to train.

Since the output edges come from a subset of the input edges, the layer introducing the Hungarian algorithm can be regarded as a special sampling layer, which we referred to as the Hungarian pooling. The back-propagation used in the end-to-end training is formulated as:

$$\frac{\partial h(\boldsymbol{G}, \boldsymbol{w})}{\partial w_k} = \begin{cases} 1 & \boldsymbol{e_k} \in \boldsymbol{M}; \\ 0 & otherwise. \end{cases} \tag{6}$$

3.3 Learning from SfM Model

In this section, we describe how to learn the parameters of BGNet. There are three parts: 1) training data generation, 2) data augmentation, and 3) the loss function.

Training data generation. We use sparse SfM models to automatically generate the training data. Since our method aims to learn the global 2D-3D geometry consistency, all types of 2D-3D correspondences can be used theoretically. Dense models can also be used if available.

We construct a bipartite graph $\boldsymbol{G}^* = (\boldsymbol{U}^*, \boldsymbol{V}^*, \boldsymbol{E}^*)$ for each image in a given SfM model. For a specific image $\boldsymbol{I}$, we take all the 2D keypoints to form the 2D point set $\boldsymbol{U}^*$. Then, all the 3D points observed by the images that are co-visible with $\boldsymbol{I}$ are used to form the 3D point set $\boldsymbol{V}^*$. All the 2D points that have been triangulated in the image $\boldsymbol{I}$ and their corresponding 3D points are used to construct the edge set $\boldsymbol{E}^*$. Note that an edge is represented by a tuple of the index of 2D point and 3D point. We mark all edges in the set $\boldsymbol{E}^*$ as inliers. Outliers are added randomly during the data augmentation phase. To eliminate the influence of the camera intrinsics, all 2D point coordinates are projected onto the normalized plane.

Data augmentation. Data augmentation is performed online. For an initial bipartite graph $\boldsymbol{G}^*$, we randomly select a certain proportion of edges from $\boldsymbol{E}^*$ and use the edges along with the corresponding 2D and 3D points to construct a new bipartite graph $\bar{\boldsymbol{G}} = (\bar{\boldsymbol{U}}, \bar{\boldsymbol{V}}, \bar{\boldsymbol{E}})$. Then, a certain proportion of 2D points and 3D points from $\boldsymbol{U}^* - \bar{\boldsymbol{U}}$ and $\boldsymbol{V}^* - \bar{\boldsymbol{V}}$ are randomly selected to add in $\bar{\boldsymbol{U}}$ and $\bar{\boldsymbol{V}}$ respectively. For each 2D point in $\bar{\boldsymbol{U}}$, we randomly choose several 3D point from $\bar{\boldsymbol{V}}$ to generate edges and add these edges in $\bar{\boldsymbol{E}}$. These newly added edges are marked as outliers. This strategy can generate different $\bar{\boldsymbol{G}}$ from $\boldsymbol{G}^*$ at training time. To make the network robust to rotation, we randomly apply a rotation to the 3D point set $\bar{\boldsymbol{V}}$ before $\bar{\boldsymbol{G}}$ is fed into BGNet.

Loss Function. The matching problem is essentially an classification problem, so we use the most commonly used cross-entropy loss function for training:

$$\boldsymbol{L}_\theta = \frac{1}{T}\sum_{k=1}^{T}(t_k log(w_k)s_k + (1-t_k)log(1-w_k)s_k). \tag{7}$$

where t_k is true label.

4 Hierarchical Visual Localization

In this section, we introduce a hierarchical visual localization method, as is shown in Figure 4. For a query image, its global feature and local features are extracted. The global feature is used for coarse localization, while the local features are used to establish 2D-3D correspondences. BGNet is applied after the local feature matching to find the global optimal one-to-one matches. The whole localization process is divided into four modules namely scene retrieval, 2D-3D local feature matching, finding maximum-weight matching, and prior-guided pose estimation. The following describes these modules in detail.

Scene retrieval. We define a set of the 3D points observed in one image in an SfM model as a meta scene. A set of the meta scenes $\boldsymbol{S} = \{S_1, S_2, ..., S_N\}$, where N is the number of images, can be obtained from a whole SfM model. We use the global descriptor of the query image to retrieve the top R images $\boldsymbol{I} = \{I_1, I_2, ..., I_R\}$ from the database. The corresponding meta scenes are denoted as $\hat{\boldsymbol{S}} = \{\hat{S}_1, \hat{S}_2, ..., \hat{S}_R\}$. Instead of directly using the meta scenes for feature matching, we further perform an expansion. We denote $\beta = |S_i \cap S_j|$, which is the number of co-visible 3D points of two scenes. $\beta > 0$ means that the meta scenes S_i and S_j are co-visible. For each retrieved meta scene $\hat{S}_i$ in $\hat{\boldsymbol{S}}$, we expand it according to co-visibility. This is achieved by finding all meta scenes from $\boldsymbol{S}$ that are co-visible with $\hat{S}_i$ and then select the top m ones with the most co-visible points. Then, all the selected m scenes are merged as an expanded sub-scene $\bar{S}_k$. The expansion is performed for the retrieved meta scenes in descending order of the retrieval scores. If the next retrieved meta scene has already appeared in the previous sub-scenes, we simply skip it. Finally, we get a set of sub-scenes $\bar{\boldsymbol{S}} = \{\bar{S}_1, \bar{S}_2, ..., \bar{S}_K\}$, which will be used for local feature matching.

Local feature matching For the retrieved sub-scenes $\bar{\boldsymbol{S}} = \{\bar{S}_1, \bar{S}_2, ..., \bar{S}_K\}$, 2D-3D feature matching is performed sequentially in the order of retrieval. Since the number of 3D points contained in each sub-scene is very small compared to the entire map, either approximate nearest neighbor search or brute force search can be used. At this stage, we try to find all possible correct matches and avoid falsely rejecting the hard matches due to illumination changes or repeated patterns. Thus, for each 2D point, multiple 3D points are kept as candidate matches only if the descriptor distances are below a given threshold.

The maximum-weight matching A bipartite graph $\boldsymbol{G} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{E})$ is constructed from the candidate 2D-3D matches and then fed into BGNet to obtain the assignment vector $\boldsymbol{s}$ and the weighting vector $\boldsymbol{w}$. All edges with s_k equal to 1 are selected as the maximum-weight matching $\boldsymbol{M}$. Only the maximum-weight matches and the corresponding weights from the vector $\boldsymbol{w}$ are used for the pose estimation.

Prior-guided pose estimation. A PnP solver inside a RANSAC loop can be applied to the 2D-3D correspondences to compute the camera pose. In the RANSAC loop, the probability of sampling 2D-3D correspondences is decided by the predicted likelihood. This allows us to sample possible inliers with larger chances.

Our pipeline is partially inspired by the current state-of-the-art method [31]. It clusters the retrieved images to form several sub-maps by using the co-visibility. The fine localization is performed

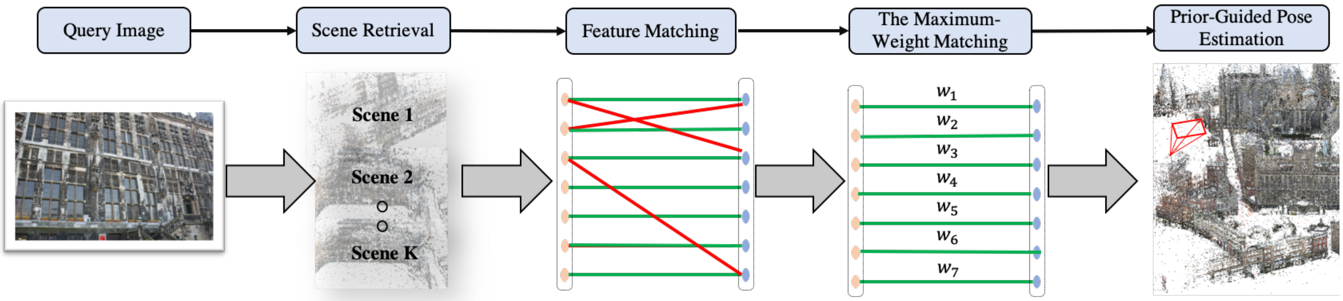

Figure 4: **Visual Localization Pipeline.** BGNet is embedded in a hierarchical visual localization pipeline. For each 2D local feature extracted from a query image, multiple 3D points from the retrieved scene may be matched. Then a bipartite graph constructed from 2D-3D correspondences is fed into the proposed network, which outputs the globally optimal matches and the corresponding likelihoods to be inlier. Finally, Prior-Guided RANSAC is used to solve the camera pose from the globally optimal matches.

against each sub-map in descending order of the map size. The localization is terminated if a satisfactory result, e.g. the number of inliers is larger than a threshold, is obtained. We use the same early-stop strategy, but the construction of the sub-map is different. First, the sub-map in [31] contains the retrieved images only, we make an expansion in the original model to effectively find more relevant 3D points. Second, the clustering in [31] uses an unlimited transitive co-visibility. This may gather many images that are far away from each other and of which most of the scenes are indeed different. Many outliers may be produced when performing feature matching with the sub-map. Instead, we limit the size of the sub-map and prefer the non-transitive co-visibility to get a cleaner sub-map.

5 Experiment Results

In this section, we show the effectiveness of the proposed BGNet through extensive experiments and demonstrate the state-of-the-art localization performance on multiple public datasets. We first compare the proposed localization method with the existing traditional ones on two challenge datasets, i.e. Aachen-day-night and RobotCar [37], and nextly compare with the learning-based methods on a relatively small dataset, namely the Cambridge Landmarks [14]. Then we perform an ablation study to analyze the factors which may impact the results. Lastly, we demonstrate an AR application using the proposed method.

5.1 Comparison to Structure-based Methods

Datasets. We conduct experiments on two challenging large datasets introduced by [37] to compare with state-of-the-art traditional methods and verify the effectiveness of the proposed BGNet. The images of each dataset consist of two parts, the reference images that are used to construct the sparse SfM model and the query images that are used for evaluation. All query images are annotated with the ground truth 6DoF poses. Both datasets provide sparse SfM models. Aachen Day-Night [38] contains 4,328 reference images (all captured in the day time) and 922 query images (824 captured in the day time and 98 in the night time). All the query images are collected using mobile phones, which are very suitable for augmented reality scenes. The light changes during the day and night bring great challenges to visual localization. The RobotCar dataset [25] consists of several video sequences collected in different seasons, including 26,121 reference images and 11,934 query images. The reference images from one season are used to construct the SfM model. Compared to Aachen Day-Night, the query images are even more challenging because both seasonal changes and illumination changes are included.

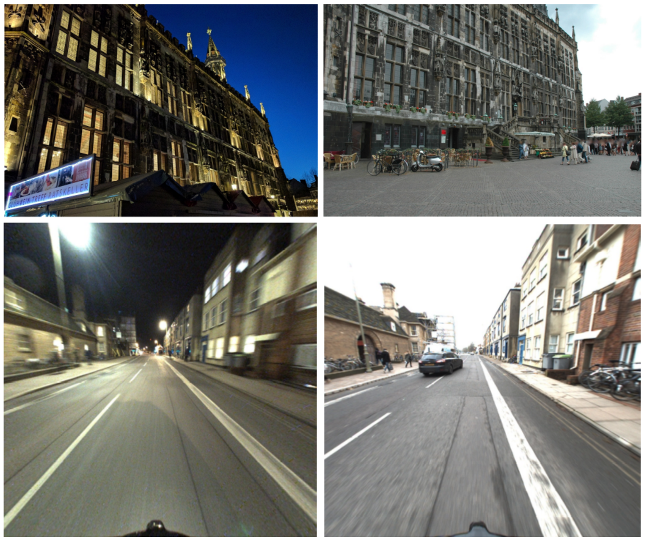

Figure 5: The figure shows the challenges of the two datasets. The left shows the query images and the right shows the reference images. In addition to changes in illumination in the two datasets, the Aachen Day-Night dataset (the first row) also contains large viewpoint changes and repeated patterns, and the RobotCar dataset (the second row) exists the serious motion blur.

Implimentation Details. We use NetVLAD as the global feature and SuperPoint as the local feature in our pipeline. For each query image, we retrieve the top 100 reference images. The maximum number of images in each sub-scene is limited to 100 at the scene retrieval stage. For each 2D point in the query image, we keep up to 8 3D points as the candidate matches at the stage of feature matching, and the threshold of the cosine similarity of candidate matches is 0.68. The SfM models are re-triangulated with SuperPoint features, and the network is trained on the new SfM models. We train 320 epochs for Aachen Day-Night and 64 epochs for RobotCar.

Compared Methods. We compare our method with four state-of-the-art methods, namely the Active search (AS) [36], City Scale Localization (CSL) [40], and the recently proposed hierarchical localization method NV+SP [31] and SuperGlue [33]. NV+SP uses the same features as our method. The local feature is SuperPoint and the global feature is NetVLAD. For a fair comparison, we re-

Table 1: **Pose Recall.** We report the pose recall [%] at different accuracy levels of positions (m) and orientations (deg) on the Aachen Day-Night and RobotCar Seasons datasets. **Bold** numbers denote the best result. A dash (-) indicates that the result was not reported by the corresponding method.

	Aachen		RobotCar			
	day	night	dusk	sun	night	night-rain
distance [m]	.25/.50/5.0	.5/1.0/5.0	.25/.50/5.0	.25/.50/5.0	.25/.50/5.0	.25/.50/5.0
orientation [deg]	2/5/10	2/5/10	2/5/10	2/5/10	2/5/10	2/5/10
Active Search [36]	**85.3** / 92.2 / **97.9**	27.6 / 38.8 / 56.1	52.0 / 83.0 / 95.9	29.6 / 57.4 / 84.1	1.6 / 3.9 / 10.5	2.0 / 10.9 / 18.0
CSL [40]	52.3 / 80.0 / 94.3	24.5 / 33.7 / 49.0	56.6 / 82.7 / 95.9	28.0 / 47.0 / 70.4	0.2 / 0.9 / 5.3	0.9 / 4.3 / 9.1
NV+SP [31]	79.9 / 90.2 / 96.6	37.8 / 60.2 / 79.6	55.8 / **83.5** / 95.9	52.2 / 74.8 / 92.6	6.4 / 14.8 / 40.6	5.7 / 26.8 / 48.6
SuperGlue [33]	-	45.9 / **70.4** / **88.8**	-	-	-	-
SR+NN+RT(ours)	81.7 / 88.7 / 96.2	38.8 / 57.1 / 80.6	56.1 / **83.5** / 96.7	53.7 / 77.4 / 94.3	8.0 / 17.8 / 49.1	7.7 / 28.2 / 53.2
SR+NN+MC(ours)	83.9 / 92.0 / 97.6	43.9 / 62.2 / 84.7	56.3 / 83.2 / **97.5**	54.6 / **77.6** / **96.7**	7.1 / 18.7 / 46.6	7.3 / 25.7 / 46.1
SR+BGNet(ours)	84.5 / **92.4** / 96.2	**46.9** /63.3 / 84.7	**56.9** / 83.2 / 95.9	**55.7** / 77.0 / 94.1	**8.4** / **21.7** / **50.9**	**10.0** / **35.0** / **59.5**

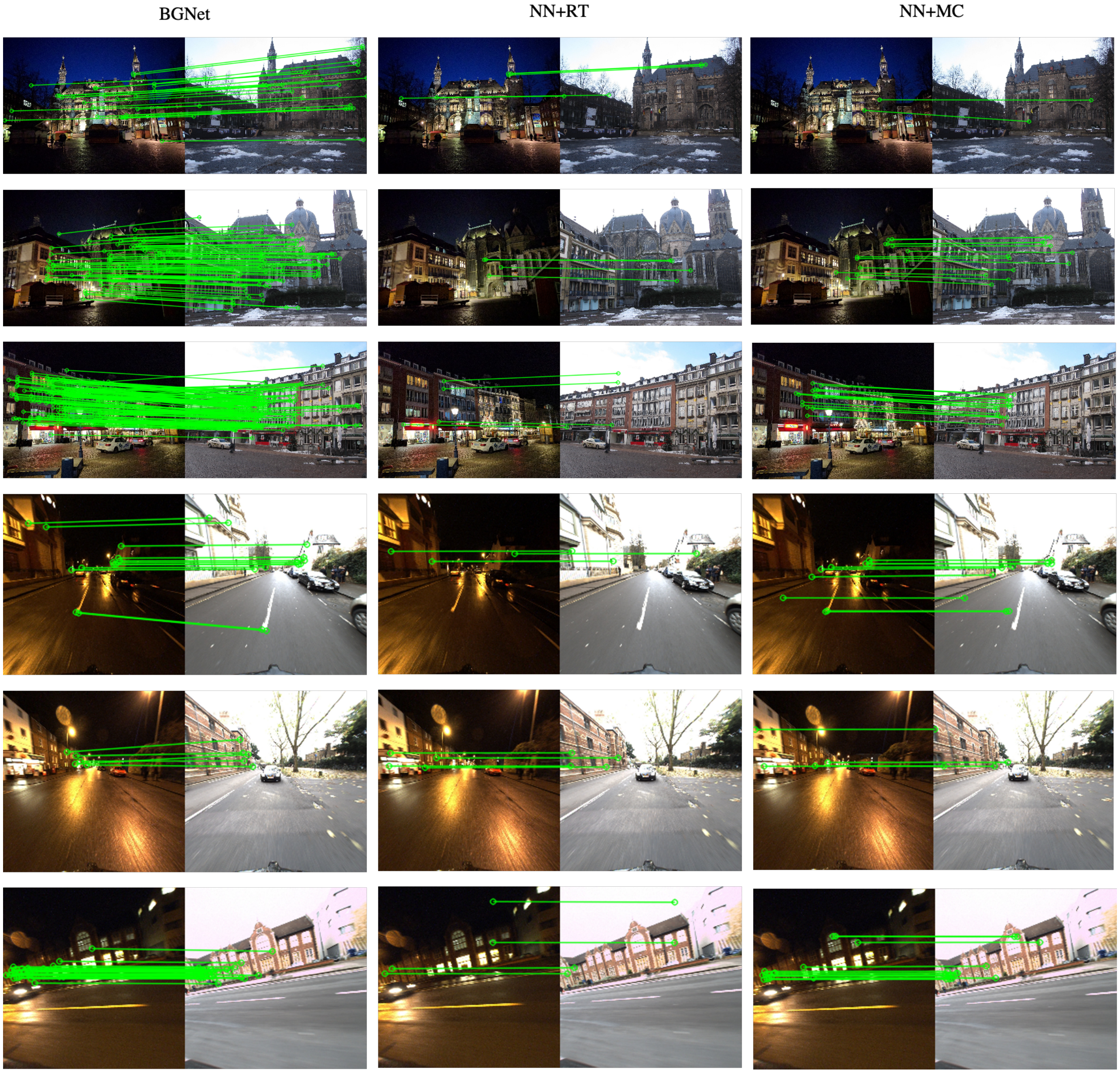

Figure 6: We draw the inliers on the retrieved image and compare BGNet with the commonly used ratio test and mutual check. Our method can find more inliers, making the solved camera pose more accurate.

implement NV+SP with the same configuration as ours. We name our proposed scene retrieval method as SR and feature matching as BGNet. In order to show the effect of BGNet, two commonly used methods for ambiguity eliminating, namely the mutual check (MC) and the ratio test (RT), are combined with our scene retrieval to form two baseline methods.

Results. We report the pose recall at different accuracy levels of positions and orientations in Table 1. These metrics follow the benchmarks [37].

As can be seen from Table 1, the pose recall with the proposed SR+RT is consistent higher than NV+SP [31] on RobotCar Dataset. This shows that the scene retrieval method provides a better local map than co-visibility clustering.

The method that uses the proposed BGNet consistently gets higher recall under high precision conditions, especially at night time which has dramatic illumination changes. In the Aachen Night scene, the pose recall under $(0.5m, 2^\circ)$ is 3 percentage higher than NN+MC. In the night-rain scene on the RobotCar dataset, the pose recall under $(5m, 10^\circ)$ is 13.4 percentage higher than NN+MC. This shows that more matches are found through BGNet in the complex environment. Qualitative results can be seen in Figure 6. We project the 3D points in inliers onto the first retrieved image. As can be seen, our method can get more inliers, and the solved camera pose is more accurate. However, in the daytime scene, like Aachen day and RobotCar dusk and sun, the pose recalls using BGNet under low precision is slightly worse than NN+MC. This may be because the daytime scene is relatively simple, and enough matches can be obtained using NN+MC.

Active Search is slightly better than our method at 0.25m and 5m on the day subset of Aachen Day-Nigth dataset and lower than our method on Aachen Night and RobotCar.

SuperGlue only reported the result on the night subset of Aachen Day-Night. The proposed method SR+BGNet is slightly better than SuperGlue at the error threshold of 0.5m but performs worse at 1m and 5m. SuperGlue performs better under low precision, in part because SuperGlue performs feature matching multiple times when establishing 2D-3D correspondences.

5.2 Comparison to Learning-based Methods

Datasets We use the Cambridge Landmarks dataset [14] for evaluation, which is commonly used for end-to-end learning methods. The Cambridge dataset contains 6 medium-scale outdoor scenes. Each scene includes the training images and the test images which are collected on different paths. All images have the ground-truth camera poses obtained by SfM.

Implimentation Details. To verify the generalization ability of the proposed method, we use the network trained on the Aachen Day-Night dataset and employ ORB to perform 2D-3D feature matching. NetVLAD is still selected as the global feature extractor. We retrieve the top 80 database images for each query image since the scales of the scenes are smaller than Aachen and RobotCar. The maximum number of images in a sub-scene is set as 80. For each 2D point in the query image, up to 4 3D points are kept as candidate matches, and the Hamming distance threshold is 50.

Compared Methods. We compare our proposed pipeline to five learning-based methods. PoseNet [14] and the spatial LSTM [44] directly regress the camera pose. DSAC [3] is the method that regresses scene coordinates and then solves pose using RANSAC+PnP. DSAC++ [4] and NG-DSAC [5] are improved methods based on DSAC. All these methods learn the parameters in an end-to-end manner by minimizing the error between the predicted pose and the ground truth. Different from these methods, our method disassembles the whole process and only focuses on learning to find the optimal matching, the remaining steps still use traditional geometric methods. This would bring better generalization ability. In this evaluation, we use ratio test combining with scene retrieval as a baseline.

Results We list the experimental results in Table 2 and Figure 7. The results of PoseNet, spatial LSTM, DSAC and DSAC++ come from the paper [4]. The NG-DSAC results are obtained by using the model released by the author. SR+RT consistently outperforms the end-to-end learning-based methods by a large margin in all scenes. Street is a relatively large scene, many learning-based methods cannot converge on this challenging scene. PoseNet works on this dataset but has a medium error of more than 20m. The medium error of our method on Street dataset is below 25cm, which is several orders more accurate than PoseNet. On several smaller scenes, like Kings College, Old Hospital, and Shop Facade, SR+RT can achieve an accuracy of 3.7cm on average.

Even if ST+RT has reached a very high accuracy, SR+BGNet still outperforms SR+RT on the CreatCourt, Kings College, St M. Church, and Street. The accuracy is further improved by 5.9cm on the Street scene. This shows that using global geometric information to reject outliers brings significant benefits, especially in the challenging conditions where local similarity is not reliable. On the Shop Facade and Old Hospital scenes, which are small and simple, BGNet performs slightly better than ratio test, because using local similarity can already achieve very good results.

As can be seen from Table 2, the methods that regress scene coordinates, such as DSAC, DSAC++, and NG-DSAC, perform better than the methods that directly regress camera pose, such as PoseNet and Spatial LSTM. Our method is more accurate than the methods of regressing scene coordinates. These results indicate that for the complicated task of localization, using the learning in the specific modules may be more effective than learning the whole process.

Generalization. BGNet used here is trained on the Aachen Day-Night dataset, while the evaluation is performed on the different datasets and a different type of feature. The improvement brought by using BGNet shows that the network not only has good generalization ability for the different scenes but also for different features.

Qualitative results. We show qualitative results in Figure 8. We compare SR+BGNet with the best performing NG-DSAC among end-to-end learning-based methods. As can be seen in Figure 8, our method has higher accuracy than NG-DSAC.

5.3 Detailed Studies

Different Local Features. We replace SuperPoint by ORB to conduct experiments on the Aachen Day-Night dataset and keep other factors unchanged. We compare the accuracy improvements brought by using BGNet on different features. The experimental results are shown in Table 3. SuperPoint+BGNet (SP+BGNet) increases the recall by 3.3% and 5.4% on average compared to SuperPoint+ratio test (SP+RT) in the day and night scenes respectively, while ORB+BGNet increase the recall by 1.0% and 2.1%. Overall, the improvement of ORB+BGNet is slightly smaller than SP+BGNet.

End-to-End Training. We analyze the impact of the Hungarian pooling. We set the method that does not use the Hungarian algorithm and the method that uses the Hungarian algorithm as the post-processing as the two baselines, named BGNet (none) and BGNet (hpost), respectively. We refer to the end-to-end training method as BGNet (hpool). We conduct the experiment on Aachen Day-Night. The experimental results listed in Table 4 show that using Hungarian pooling for end-to-end training can effectively improve performance. It has higher recall than either without this pooling or using the Hungurain algorithm as a post-processing step.

Table 2: **Median Localization Error.** We report results for the Cambridge Landmarks dataset. A dash (-) indicates that a method failed completely. We mark best results **bold**.

	Pose Regresion		Scene Corrdinate			Ours	
	PoseNet	Spatial LSTM	DSAC	DSAC++	NG-DSAC	SR+RT(ours)	SR+BGNet(ours)
Great Court	700cm, 3.7°	-	280cm, 1.5°	40.3cm, 0.20°	34.8cm, 0.18°	16.0cm, 0.08°	**13.2cm, 0.06°**
Kings College	99cm, 1.1°	99cm, 1.0°	30cm, 0.5°	17.7cm, 0.30°	12.2cm,0.23°	5.1cm, 0.10°	**4.9cm, 0.08°**
Old Hospital	217m, 2.9°	151cm, 4.3°	33cm, 0.6°	19.6cm, 0.30°	21.2cm, 0.45°	14.6cm, 0.24°	**12.6cm, 0.20°**
Shop Facade	105cm, 4.0°	118cm, 7.4°	9cm, 0.4°	5.7cm, 0.30°	5.4cm, 0.29°	**2.9cm**, 0.12°	**2.9cm, 0.11°**
St M. Church	149cm, 3.4°	152m, 6.7°	55cm, 1.6°	12.5cm, 0.40°	9.9cm, 0.31°	3.7cm, 0.13°	**3.3cm, 0.11°**
Street	2070cm, 25.7°	-	-	-	-	24.2cm, 0.71°	**18.3cm, 0.53°**

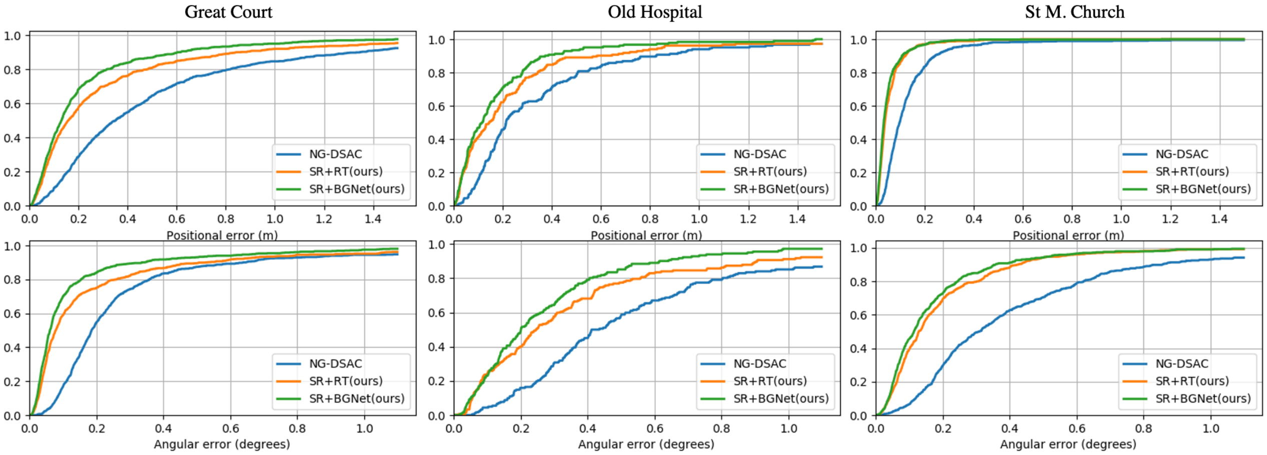

Figure 7: **Cumulative distribution of orientation and position errors** for three scenes in the Cambridge Landmarks dataset. Our methods consistently outperform NG-DSAC and the accuracy can be further improved through BGNet.

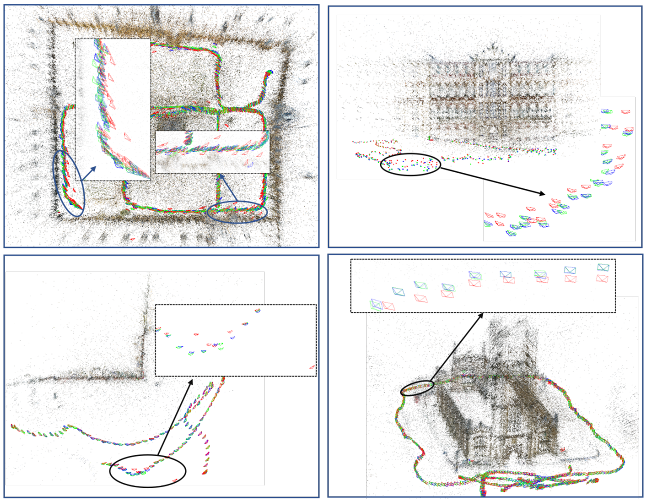

Figure 8: **Estimated Camera Pose.** We plot the estimated camera pose in the respective sparse 3D point cloud. We compare the proposed method SR+BGNet(blue) with NG-DSAC(red). The ground truth is shown in green.

K values. We evaluate the influence of different K at the feature matching stage. We conduct the experiment on the night queries of the Aachen Day-Night dataset. The experiment results are shown in Table 5. As the value of K increases, the performance of KNN+Distance drops rapidly. This shows that the traditional RANSAC fails when there are too many outliers, even if the number of inliers increases. In contrast, the accuracy of the KNN+Distance+BGNet is getting higher with the increase of K. It starts to decline after reaching a certain value. This shows that BGNet can find the correct matches from the matching that contains a large number of outliers through geometric prior, but as the K value increases, this geometric prior will also be disturbed by outliers.

Table 3: **Different Local Features.** The results on the Aachen Day-Night dataset with different local features. Numbers in green denote the improved recall [%] with regard to the given precision, and numbers in red denote the decreased recall [%].

Methods	Day	Night
SP+RT	81.7 / 88.7 / 96.2	38.8 / 57.1 / 80.6
SP+BGNet	84.5(+2.8) / 92.4(+3.7) / 96.2	46.9(+6.1) / 63.3(+6.2) / 84.7(+3.9)
ORB+RT	68.1 / 75.5 / 83.3	12.2 / 15.3 / 22.4
ORB+BGNet	69.8(+1.1) / 76.6(+1.1) / 83.1(-0.2)	14.3(+2.1) / 17.3(+2.0) / 24.5(+2.1)

Table 4: The results of End-to-End learning of BGNet on Aachen Day-Nigh. We report pose recall [%] of different variants at different accuracy of positions and orientations.

	day	night
BGNet(none)	83.9 % / 92.1% / 95.6%	43.9% / 60.2% / 83.7%
BGNet(hpost)	83.4% / 91.0% / 96.2%	42.9% / 59.2%/ 82.7%
BGNet(hpool)	84.5% / 92.4% / 96.2%	46.9% / 63.3% / 84.7%

Timing. We measure the run-time of the main components of the proposed method on the machine with an Intel Core i7-8700 CPU and a GeForce GTX 1080 GPU. We resize the larger dimension

Table 5: The pose recall with different K on the night query images of the Aachen Day-Night dataset.

K	KNN+Distance	KNN+Distance+BGNet
1	40.8% / 61.2% / 80.6%	42.9% / 60.2% / 85.7%
2	40.8% / 59.2% / 82.7%	43.9% / 61.2% / 84.7%
4	40.8% / 59.2% / 81.6%	43.9% / 61.2% / 85.7%
6	37.8% / 56.1% / 75.5%	45.9% / 62.2% / 84.7%
8	34.7% / 53.1% / 71.4%	46.9% / 63.3% / 84.7%
10	33.7% / 43.9% / 70.4%	44.9% / 61.2% / 83.7%

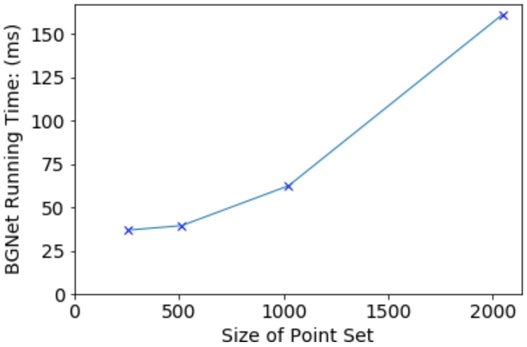

Figure 9: **Run-time of BGNet.** The sizes of 2D point set and 3D point set are equal and the size of edge set is twice as much as the size of 2D point set. BGNet takes 39.39ms and 62.4ms when the point set size is 512 and 1024, respectively.

of the query images to 640 for SuperPoint and 360 for NetVLAD. For a given 3D model with 8K reference images, Scene Retrieval takes 22.5ms, where NetVLAD takes 18.5ms. KNN Matching takes 25.5 ms for matching between 2,000 2D features and 20,000 3D points. The final pose estimation takes 30.2ms averagely. The details about the run-time of BGNet can be seen in Figure 9.

5.4 Application for Augmented Reality

Global localization with 6DoF pose estimation is crucial for augmented reality applications in a large-scale scene. Although SLAM technique [15, 27] can be used to estimate 6DoF poses, it easily accumulates error in a large-scale scene. So it is better to leverage global relocalization techniques like our method to correct the pose to eliminate tracking drift. To demonstrate the accuracy and robustness of our localization result, we capture a video and insert some virtual objects into each video frame by firstly aligning them in the 3D map reconstructed by SfM, and render them according to the localization results.

The AR effect is shown in Figure 10, which also shows the recovered camera trajectory. It can be seen that the recovered trajectory by our method is already quite smooth even the pose of each frame is estimated independently. Almost the poses of all frames are faithfully recovered, which demonstrates the effectiveness of the proposed visual localization method. Actually, for AR application, we do not need to estimate the pose of each frame independently by global relocalization, since SLAM technique can be used to smoothly recover the poses of each online frame, and global relocalization can be used to align the 3D coordinate of SLAM into the world coordinate and eliminate the tracking drift.

6 Conclusion

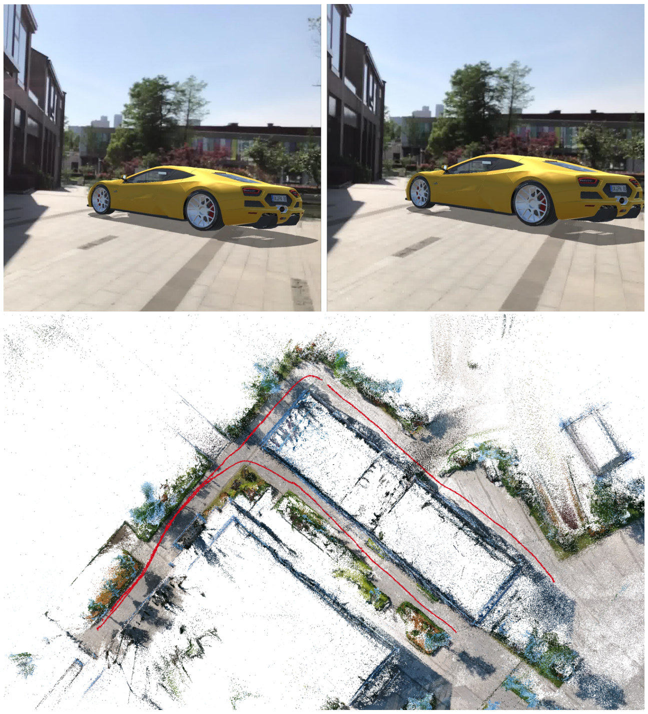

Figure 10: Visual localization for AR application. Top: two selected augmented frames. Bottom: the recovered camera trajectory superimposed in the offline 3D map recovered by SfM.

This paper proposes an effective Bipartite Graph Network for visual localization. BGNet extracts geometric features for a set of 2D-3D correspondences and is able to learn the global geometric consistency to predict the possibility of being a true match for each correspondence. BGNet also integrates the Hungarian algorithm into the network as a special pooling layer to find maximum-weight matching in an end-to-end manner. This approach enables the localization to obtain more correct matches and hence improves the robustness and accuracy of localization. We further combine BGNet with a novel scene retrieval strategy. The experiments show that the proposed method outperforms the existing state-of-the-art methods.

Acknowledgments

The authors would like to thank all the reviewers for their valuable comments, and Mingxuan Jiang and Chongshan Shen for their help in producing the results of AR application.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307, 2016.

[2] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4181–4190, 2017.

[3] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC-differentiable RANSAC for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6684–6692, 2017.

[4] E. Brachmann and C. Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4654–4662, 2018.

[5] E. Brachmann and C. Rother. Neural-guided RANSAC: Learning where to sample model hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4322–4331, 2019.

[6] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pp. 2616–2625, 2018.

[7] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1048–1058, 2009.

[8] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 220–226. IEEE, 2005.

[9] D. F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

[10] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 224–236, 2018.

[11] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A trainable CNN for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

[12] M. A. Fischler and R. 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.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[14] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2938–2946, 2015.

[15] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234. IEEE, 2007.

[16] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2969–2976. IEEE, 2011.

[17] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *European Conference on Computer Vision*, pp. 748–761. Springer, 2010.

[18] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[19] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(N) solution to the PnP problem. *International journal of computer vision*, 81(2):155, 2009.

[20] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision*, pp. 791–804. Springer, 2010.

[21] L. Liu, H. Li, and Y. Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2372–2381, 2017.

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

[23] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan. ContextDesc: Local descriptor augmentation with cross-modality context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2527–2536, 2019.

[24] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *Proceedings of the European Conference on Computer Vision*, pp. 168–183, 2018.

[25] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[26] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2666–2674, 2018.

[27] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[28] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *2017 IEEE International Conference on Robotics and Automation*, pp. 2614–2620. IEEE, 2017.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pp. 2564–2571. IEEE, 2011.

[31] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

[32] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. *arXiv preprint arXiv:1809.01019*, 2018.

[33] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. *arXiv preprint arXiv:1911.11763*, 2019.

[34] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2102–2110, 2015.

[35] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pp. 667–674. IEEE, 2011.

[36] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *European Conference on Computer Vision*, pp. 752–765. Springer, 2012.

[37] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8601–8610, 2018.

[38] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, vol. 1, p. 4, 2012.

[39] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937, 2013.

[40] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1455–1461, 2016.

[41] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7199–7209, 2018.

[42] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11016–11025, 2019.

[43] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. Semantic match consistency for long-term visual localization. In *Proceedings of the European Conference on Computer Vision*, pp. 383–399, 2018.

[44] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 627–637, 2017.

[45] A. Zanfir and C. Sminchisescu. Deep learning of graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2684–2693, 2018.

[46] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5845–5854, 2019.