제 110 회 박사학위논문
지도교수 홍 현 기

카메라 위치 파악을 위한 복셀 표현의 covisibility 기반 참조 이미지 클러스터링

Clustering reference images based on the covisibility in voxel representation for camera localization

중앙대학교 대학원

융합공학과 디지털이미징 전공

이 상 윤

2022년 2월

제 110 회 박사학위논문
지도교수 홍 현 기

카메라 위치 파악을 위한 복셀 표현의 covisibility 기반 참조 이미지 클러스터링

Clustering reference images based on the covisibility in voxel representation for camera localization

중앙대학교 대학원

융합공학과 디지털이미징 전공

이 상 윤

2022년 2월

카메라 위치 파악을 위한 복셀 표현의 covisibility 기반 참조 이미지 클러스터링

Clustering reference images based on the covisibility in voxel representation for camera localization

이 논문을 박사학위논문으로 제출함

2022년 2월

중앙대학교 대학원

융합공학과 디지털이미징 전공

이 상 윤

이상윤의 박사학위논문으로 인정함

심 사 위 원 장 김 태 용 (인)

심 사 위 원 홍 병 우 (인)

심 사 위 원 박 경 주 (인)

심 사 위 원 권 준 석 (인)

심 사 위 원 홍 현 기 (인)

중앙대학교 대학원

2022년 2월

Contents

[Picture table of contents]

[Table of contents]

1. Introduction

This chapter briefly describes the background, purpose, and content of the study, and then explains the overall composition of the paper.

1.1 Background

Camera video has become an essential element in various fields of industry, starting with individuals. Cameras are not only major sensors for computer vision, but also sensors for mobile robots such as drones and self-driving cars.

As shown in Figure 1, This camera sensor is important in augmented reality, autonomous driving, and robotics applications. In these applications, it is necessary to estimate the pose (6-degree of freedom) of the camera [1-5].

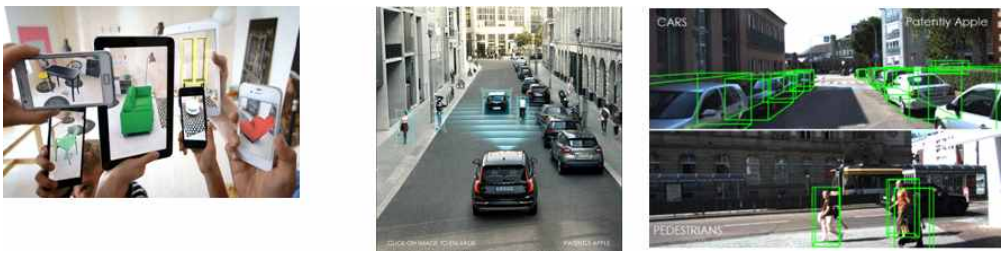

Fig 1. ikea mobile AR catalog(left image), Patently Apple(middle, right image).

This visual localization has been demonstrated by simultaneous localization and mapping (SLAM) technology, which means simultaneous position estimation and mapping of robots, and structure from motion (SfM) to perform camera tracking and coordinate system calibration using projection and geometric conditions.[1,2,6,7]. As the scope of application of these visual localization measurements expands, reliable operation is required both indoors and outdoors regardless of lighting, seasonal changes, or weather.

In particular, indirect SLAM optimizes geometric errors by estimating

3D geometry from a series of keypoint matches. Direct SLAM systems, such as direct sparse odometry, do not require preprocessing steps such as establishing a correspondence and optimize photometry directly defined in the image for reverse depth estimation[8].

However, direct approaches are heavily influenced by strong geometric noise from rolling shutters or inaccurate unique corrections, for example. This is because points with sufficiently high image tilt sizes of keyframes are tracked in subsequent frames using discrete searches along the epipolar line to minimize photometry errors.

The main approaches of traditional structure-based methods rely primarily on estimating the correspondence between the 2D keypoints of the query and the 3D points of the sparse model using a local descriptor.

In general, scale-invariant feature transform (SIFT)[9] is used as a feature descriptor, which is greatly affected by challenging changes in scenes such as viewpoints, lighting, and weather conditions.

In addition, feature descriptors are randomly extracted binary vectors from Gaussian distributions centered on key points of pixels, and there are binary robust independent elements (BRIEF) that are very sensitive to in-plane rotation.[10] And Oriented FAST and Rotated BRIEF (ORB) descriptors to compensate for the sensitivity of this rotation are scale invariant.[11]

Convolutional neural networks (CNNs) have recently been used to identify visual localization and estimate camera pose based on deep learning.[3,4,12-17]. PoseNet[3] estimates the location of a camera with only one image, and for this to this, it is the first application using CNN as end-to-end learning. That is, unlike the handcrafted method, it is possible to directly estimate the camera pose with only the image without

setting the inter-frame feature correspondence or storing the keyframe. That is, a visual feature vector is generated with one monocular image, and the last two connected layers constitute translation and orientation information. It shows fast learning speed with transfer learning and has considerably fast inference time, but has greater camera pose errors than conventional structure-based methods.[5,6,18]

There is VlocNet++[4], which is based on multi-task learning of 6-DoF camera pose, semantic segmentation, and odometry estimation by applying such an absolute pose regression approach. Because geometric and structural information encoded by the odometry model is shared with the localization head, VlocNet++ can be trained in a dataset consisting of successive pairs of monocular images.

The above method is designed to aggregate motion-specific time information based on region activation and fuse semantic features into localization streams, but images of objects [12, 13]. The main advantage of this method is the ability to identify parts of objects in complex scenes to deal with occlusion problems. The goal of this study is mainly robot manipulation of certain objects such as chairs and boxes and human-robot interactions. Deep neural networks were also used to track 6-DoF of rigid bodies in large occlusion environments.[16,19]. However, this method requires RGB-D sensing data, so its actual use is limited to indoor scenes.

In addition, approaches such as Scene point regression, structure-based with image regression, and Relative focus estimation are being researched to improve performance.

Camera pose estimation performance is influenced by the limitations of feature tracking techniques, although recent structure-based methods are

more accurate than learning-based models. For example, the SIFT-based SfM method requires a short view interval between the two images in order for feature tracking to operate successfully.[9] The overlapping part of the scene varies depending on the base line length, and if it is too far, the overlapping part disappears, so feature tracking fails. Deep learning-based models can learn scene features, so they estimate camera poses directly from a single image without tracking features. Therefore, many end-to-end deep learning methods have almost real-time inference time and attractive simple pipelines. In contrast, the SfM method requires large-scale prior information about the scene to estimate the camera pose from the input image. This means that the SfM method requires higher computational and memory costs than the deep learning model. However, localization errors obtained from deep learning models are generally greater than structure-based methods.

To improve the localization performance of deep learning-based methods, coarse-fine localization has been introduced that uses the entire learned image global (for image retrieval) and local descriptor (for 2D-3D matching)[14,15]. Here, the global search process is used to obtain the k-nearest image for a given image for localization at the map level (location hypothesis). In the local search, the camera pose is subsequently estimated more accurately based on the 2D-3D match within the candidates space of k using feature descriptors. Previously, handcrafted features such as SIFT were matched, but computational costs are high because they generate a large number of features. To address this, deep learning-based Hierarchical Feature Net maximizes computational sharing by detecting points of interest and collectively calculating global and local descriptors [15].

This deep learning architecture is reconstructed into a sparse 3D model using SfM techniques, which detects and matches points of interest using SuperPoint[20]. However, studies on hierarchical models as described above do not suggest how to construct training datasets for 2D-3D matching, and the limitations of SfM techniques in these models remain unresolved.

Many deep learning methods have been proposed to establish pixel-level correspondence between images.[20,21-23]. The approach adopted by the SuperPoint architecture is to reduce the dimension of the image to be processed with the VGG style [37] encoder. The regression work is then divided into two decoder heads for pixel-level point of interest detection and point of interest description [20]. SuperPoint requires point pre-training in a composite dataset consisting of simple geometric shapes without ambiguity in the locations of the point of interest. In addition, homographic adaptation is applied to warp the input image so that the point detector can recognize the scene at different viewpoints and scales. because changes in homography do not sufficiently reflect possible visual changes in scenes with multiple complex surfaces, Christopher et al. A convolutional space converter was included that mimics the corresponding contrast loss function of the fully convolutional architecture and the patch normalization of SIFT[22]. Similarly, with R2D2, detection and descriptor parameters are shared, and repeatability and reliability are used for both tasks, respectively [21]. Therefore, to train a deep learning model to identify detected points of interest based on a descriptor, a large number of image pairs and corresponding point sets must be included in the training dataset, and effects such as lighting and viewpoint changes must be considered.

Candidate images similar to query images can be extracted using this coarse-to-fine approach, and matching time can be significantly shortened with this information. However, even these candidate descriptors may include a plurality of images at a different time point from the query image. That is, a plurality of outliers may be included in the matching correspondence, which hampers accurate pose estimation.

1.2 Research Purpose and content

As mentioned above, visual-based camera position estimation is generally a technology for estimating the current position and direction of a camera using only two-dimensional images acquired from the camera without the help of additional equipment such as GPS, gyroscope, and RFID. To this end, several existing studies are largely classified based on structure or image. To identify this visual location, the former performs a direct match of local descriptors between 2D keypoints of query images and 3D models configured using structural from motion (SFM) or something like that.[23,24,25,26,27]. This method can estimate an accurate pose, but is computationally intensive due to thorough matching and large scale DB information. In addition, as the size of the model increases, the matching becomes ambiguous as perceptual aliasing occurs. It is estimated by directly regressing the camera pose from a single image, but the performance is poor in terms of accuracy.[28] The image-based method can literally search for an image, discretize the database, and estimate the approximate pose, which is not accurate enough for many applications.[29,30] However, since it relies on global image-wide information, it is much more powerful than direct local matching.

Local features learned in recent years have been developed to replace hand-made descriptors. In particular, deep convolutional neural networks (dCNNs) have been successfully demonstrated in other computer vision tasks such as image segmentation [31,32], object detection [33,34], and image classification [35,36]. Among these deep learning methods using CNN, HF-Net, a hierarchical localization approach, shows good

performance. This coarse-to-fine approach finds candidate images similar to query images without having to match descriptors for all 3D models through a global descriptor. These candidate information can significantly shorten the matching time. However, even these candidate descriptors may include a plurality of images at a different view point from the query image. In the end, a number of outliers may still be included in the matching correspondence relationship.

This paper proposes a method to reduce the outlier gap between the global descriptor and the local descriptor. Recently, hierarchical localization methods have been widely used in recent years to extract global descriptors from NetVLAD[38], a large-scale, up-to-date network for image retrieval proposed by Relja Arandjelovic and others. There are many key point extraction methods to select candidates using the global descriptor and generate local descriptor, and SuperPoint was used in this paper. Although the time and accuracy of 2D-3D matching for camera pose estimation are increased with a hierarchical approach, there is a problem of setting the number of image DB candidates in the global descriptor. And the outlier of the matching result according to the number may sufficiently affect the pose estimation performance.

To solve this problem, this paper focused on increasing matching performance by clustering candidate images extracted from Glover matching between those seen from similar camera points of view using clusters based on the importance of reference images.

To this end, a voxel-based scene representation was constructed using 3D information of reference images. Using this voxel scene information, a new co-visibility algorithm can be presented, and a cluster based on importance can be created through co-visibility. After the scenes of

similar images are clustered, matching with query images becomes easier and the number of outliers decreases, making the final camera pose estimation more accurate, showing better results than the performance of the pipeline of previous studies.

1.3 Thesis Organization

Following the introduction to Chapter 1, Chapter 2 summarizes various existing approaches to estimate the posture of the 6-degree camera and explains deep learning-based methods to estimate the global and local descriptors that are the basis for database configuration for camera localization. Chapter 3 describes voxel-based scene representations for clustering based on the importance of reference images and algorithms for clustering through them, and describes deep learning network methods used for hierarchical localization used to test the performance of this algorithm implementation. Finally, an overall system configuration diagram of a camera pose method using clustering will be described. Chapter 4 explains the data sets and performance evaluation indicators used in the experiment, and compares and analyzes the performance of the proposed method. Chapter 5 explains the conclusions on the results of the proposed method and the direction of future research.

2. Related Work

This chapter summarizes several existing approaches for estimating the pose of a 6-degree camera and describes deep learning-based methods for estimating global and local descriptors that are the basis for database configuration for camera localization proposed in this paper.

2.1 SFM(Structure From Motion)

Structure from Motion (SFM) is an algorithm that structures the relationship between images and cameras after backtracking the location and direction of the camera of the captured image using motion information of the two-dimensional image. In order to find out the movement of the camera from a plurality of images, it is necessary to know a correspondence relationship between images. The correspondence of an image mainly uses feature points that are easy to extract from an image, such as a corner, and the RANdom Sample Consensus (RANSAC) algorithm [39] is widely used to increase the matching rate.

2.1.1 Camera Projection Relation(2D-3D)

In order to structure the relationship between cameras, you first need to know about the projection geometry of the camera caps. A camera is defined as a morphism relationship between a 2D image and a 3D space. For camera geometry analysis, an ideal virtual camera mathematically modeled, that is, a pinhole camera, is generally used. An ideal pinhole camera is determined by one point C (center of the camera) and one image plane D in space as follows, and the focal point f of the camera is defined as the distance between C and D.

$$Cam = (C, D),\ f = d(C, D) \tag{1}$$

Pinhole cameras can be interpreted linearly using a homogeneous ordinate system within the projection space. In Fig. 2, a point $P_w = (X, Y, Z)$ in space can be expressed as P=[X Y Z 1] which is one

four-dimensional vector within the homogeneous coordinate system, and a point on the two-dimensional plane can be expressed as p=[x y 1]. Using the homogeneous coordinate system, Equation (2) can be expressed as a determinant using the 3X4 matrix as follows, which represents a linear projection relationship of camera geometry.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \propto HP_w = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \\ 1 \end{bmatrix} \quad (2)$$

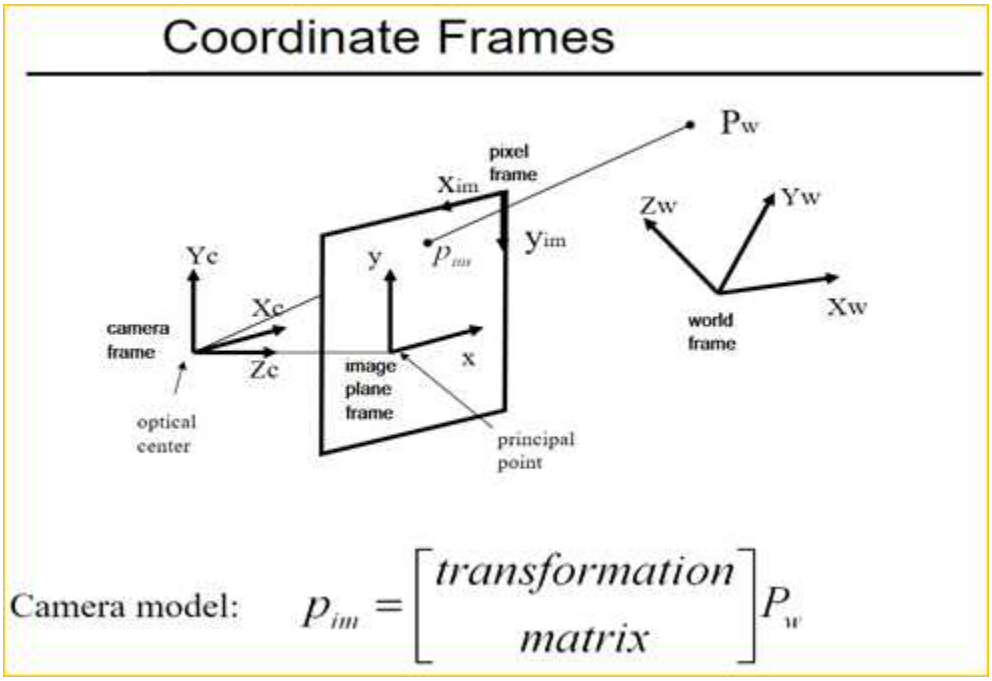

Fig 2. Camera coordinate system and camera model.

Here, the matrix H is a projection matrix indicating a projection

transformation between an image and a 3D space. Equation 2 is a relational expression for a case where the main point of the camera coincides with (0,0) as the origin of the image. However, in general, the origin of the image coordinate system and the main point of the camera do not match, and when the coordinates of the main point of the camera are $(p_x, p_y)^T$, the following projection relationship exists.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \propto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3)$$

In this relationship, if the intrinsic parameter K of the camera is defined as a 3X3 matrix as follows, Equation (3) can be summarized as Equation (4) below.

$$x = K[I|0]X_{cam},\ K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

In Equation (4), I is the identity matrix, and X_{cam} means the coordinate value of the 4x1 homogeneous coordinate system within the camera coordinate system.

The intrinsic parameter should also include skewness in columns 1 and 2, but the torsion of elements entering the camera today is ignored.

Now, if H′ is the matrix that converts a point on the world coordinate system into a point (x,y) on the image plane, the relational expression is shown in Equation (5) below.

$$s\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H' \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5)$$

In this case, H′ is a 3x4 matrix and may be decomposed and

expressed as follows.

$$s\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = KT_{pers}(1)[R|t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

In Equation (6), [R|t] converts the world coordinate system into a camera coordinate system, Tfers (1) into a projection matrix that projects 3D coordinates on the camera coordinate system into a normalized image plane, and K changes the normal image coordinates into pixel coordinates. Tfers 1 refers to projection transformation into a plane with d=1 and Zc=1.

This equation is as follows.

$$s\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K[R|t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (7)$$

In this way, image-based 3D coordinates can be estimated by the projection relationship.

2.1.2 Epipolar geometry

Since the depth of the 2D-3D correspondences defined by the projection model relationship described in Section 2.1.1 above is unknown when the 2D point is historically projected into the 3D space, a single image alone cannot estimate the 3D coordinates from two different points. This is called epipolar geometry.

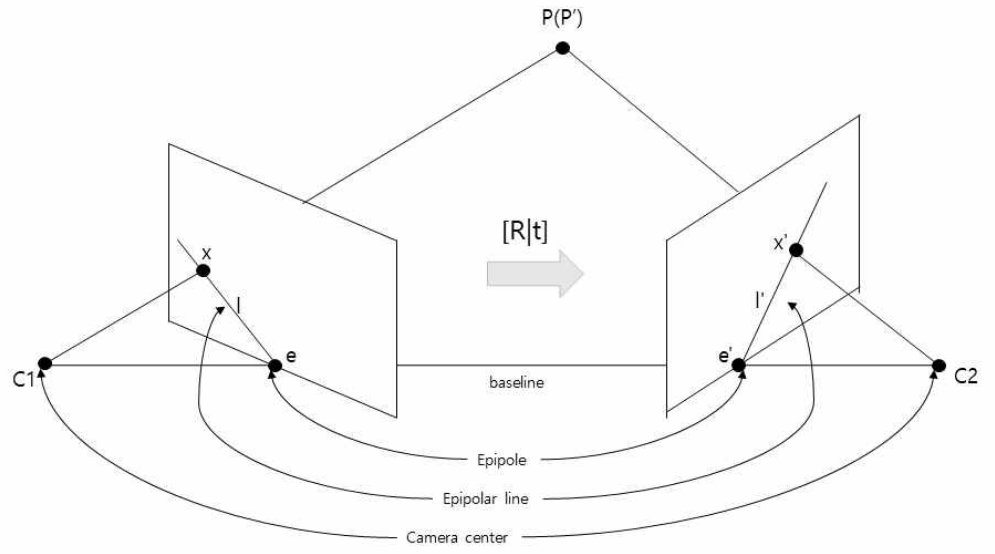

Fig 3. Epipolar geometry between the two images.

As shown in Figure 3, when a point P in a three-dimensional space is projected onto the image x of camera C1 and projected onto the image p′ of camera C2, an epipolar plane is defined by the three points P, x, x′, and the point where each image plane meets the line connecting the origin of the two cameras is called an epipole. And the straight lines l, l′ connecting this plane and epipole are called epipolar lines.

From the image coordinate x of C1, the corresponding image coordinate p′ of C2 cannot be determined alone, but l′, which is a straight line through p′, can be determined only.

In this case, the transformation relationship for calculating the corresponding epipolar line between the two images may be expressed as x′=Hp·x by homography Hp. In each image, the epipolar straight line can be calculated as $l = e \times x$ and $l' = e' \times x'$, and the following relationship is established between the epipolar straight line and the point projected in

two dimensions.

$$l' = e' \times x' = [e']_x H_P x = Fx \quad (8)$$

$[e']_x H_P = F$; This equation describes the conversion relationship between the two images, and F is called a fundamental matrix. This means a point-to-straight transformation between two images, and by this relationship, the following correspondence relationship is established between the corresponding points x↔x′ between the two images.

$$x'^T Fx = 0 \quad (9)$$

2.1.3 Triangulation

Triangulation is given a geometric relationship between the two image planes, and given matching pairs x, x′ on the two image planes, the original 3D spatial coordinates P can be determined therefrom.

Given Equation (9) and the camera matrix H, H′ for the two images, a relationship between x = HP and x = H′P occurs. Here, the external appearances of x and x′ on both sides of the relational expression between the camera matrices H and H′ and the three-dimensional coordinate X are calculated as follows.

$$x \times (HX) = 0,\ x' \times (P'X) = 0 \quad (10)$$

By summarizing Equation 10, a linear determinant for the camera matrix and the 3D coordinate P of each image may be obtained through three relational expressions for each of H and H′.

$p^{iT}$and p'^{iT} mean the I-th row in H and H′, respectively.

$$AP = 0,\ A = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix} \tag{11}$$

In Equation (11), the 3D coordinate P can be estimated using Least Square estimation such as Single Vector Composition (SVD).

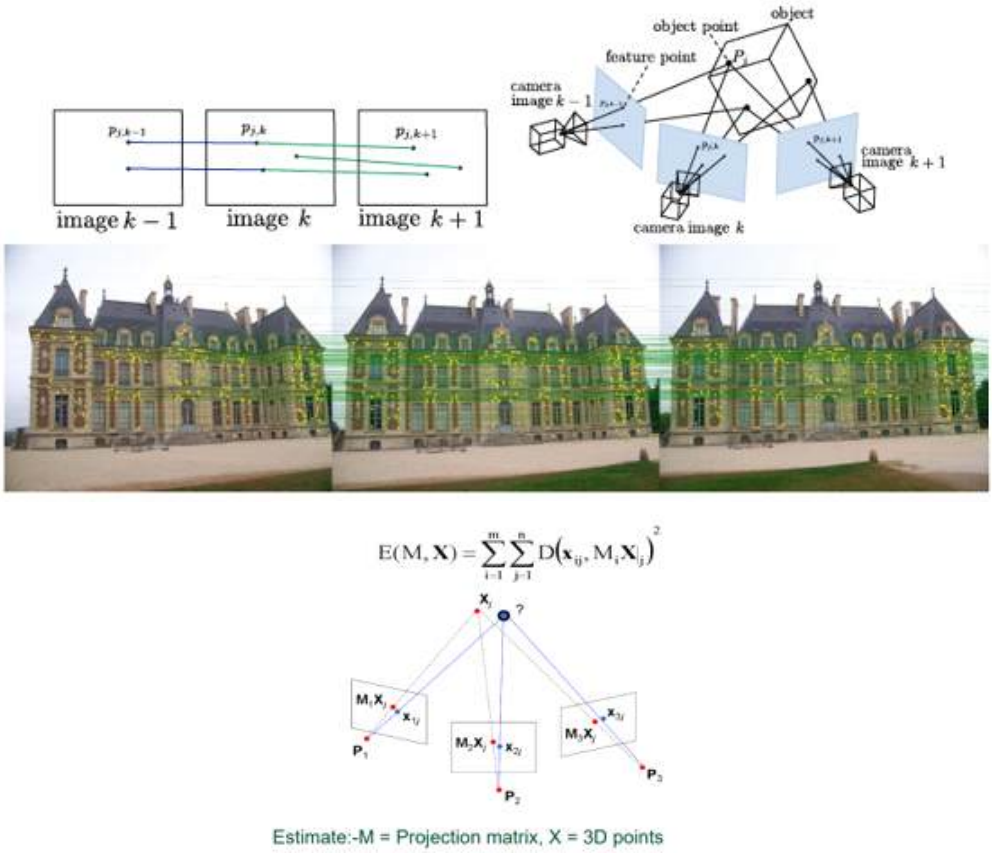

Fig 4. Structure From Motion Pipeline.

Figure 4 summarizes the structure from motion (SFM) process. Based on epipolar geometry, camera pose is estimated in consideration of matching correspondence between 2D-2D in many images, 3D points are

measured using triangulation, and camera pose and 3D points are optimized as Levenberg-marquardt (LM) algorithm [40].

2.2 6-DoF Visual Localization

6-degree of freedom visual localization is a method of estimating the current pose of a camera with respect to a viewpoint only with visual information of a camera image.

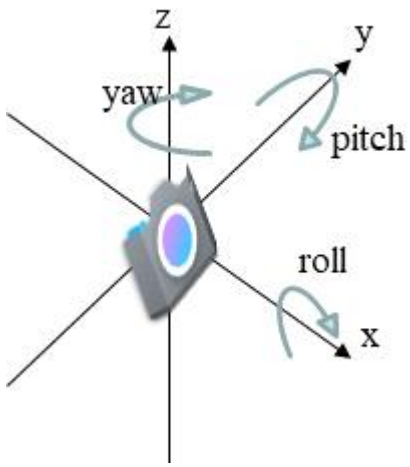

Fig 5. 6-DoF camera pose.

The camera pose of 6-degree of freedom is shown in Fig. 5 with three degrees of freedom in the rotational direction with respect to pitch, yaw, and roll and three degrees of freedom representing the position change in X, Y, and Z axes.

That is, the purpose of visual localization is to estimate the absolute camera position with these rotation and translation parameters. As explained in Chapter 1, the classical method estimates camera pose by minimizing errors with RANSAC using a 3D point cloud made of SFM through matching with query images after extracting local features from the learning data set to DB. In addition to this old structure-based

method, the approach and characteristics of the method are described in Table 1 below[41], such as a recent Scene position regression method using deep learning or an end-to-end input and a pose regression method.

Approach	3D map	Pros	Cons
Structure-based	yes	Perform very well in most scenarios	Challenging in large environments in terms of processing time and memory consumption
Structure-based with image retrieval	yes	Improve speed and robustness for large-scale settings	Quality heavily relies on image retrieval
Scene point regression	yes/ no	Very accurate position in small-scale settings	To be improved in large environments
Absolute pose regression	no	Fast pose approximation, can be trained for certain challenges	Low accuracy
Pose interpolation	no	Fast and lightweight	Quality relies heavily on image retrieval and only provides a rough pose
Relative pose estimation	no	Fast and lightweight	Quality relies heavily on image retrieval and, e.g., local feature matches or a DNN used for relative pose estimation

Table 1: Comparison of different methods

In addition, [41] provides pipelines for various visual localization methods as pictures so that you can see them at a glance. Figure 6 below

is a reference image for it [41].

To localize the query image, all visual localization approaches require a set of pose-tagged reference images to create a map or other representation of the environment.

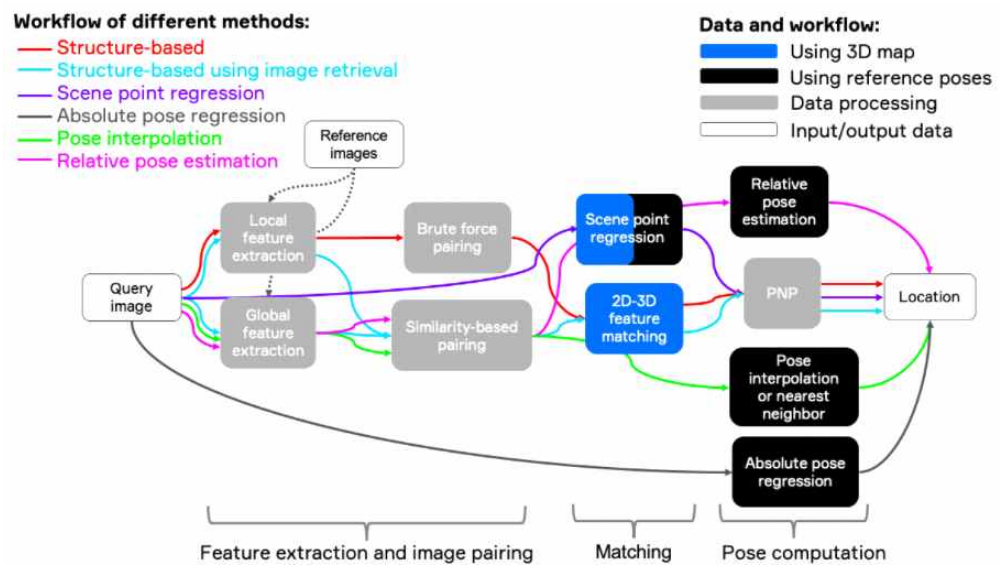

Fig 6. Overview of different methods of visual localization.

Referring to Fig. 6, the structure-based method generates a 3D map by 3D reconstruction using a correspondence relationship by local feature matching, and localizes a query image within that map. Image search can be used in structure-based methods, which can be used to reduce search space by pairing only similar images instead of all possibilities. Alternatively, these similar images can be used for pose interpolation or opponent pose estimation. The scene point regression method can directly determine the correspondence between the 2D pixel position and the 3D point using a deep neural network (DNN), and also uses 3D reconstruction

in the learning process. Finally, the absolute pose regression method estimates end-to-end poses using DNN.

Looking at the research progress so far, the Geometric method is still better than the end-to-end method. For the best performance, it is recommended to change several modules from geometric to deep learning solutions. This is because, for example, local/global feature extractors are more robust to various variations than Handcrafted methods. In this paper, we also propose camera pose estimation using this approach.

2.3 Hierarchical Localization

Section 2.2 confirmed that the hybrid hierarchical approach using deep learning is the most effective performance studied to date. Since this paper also uses it, this section describes hierarchical localization methods.

Recall the techniques used for hierarchical approaches.

Two methods are used, an image search method and a structure-based method. First, the image search method may quickly extract a result for the query image, but provides an approximate camera pose estimate. Structural-based methods show excellent performance in places where the 3D-DB (DataBase) model is small, but require a 3D model that linearly increases with the size of this scene. For this reason, searching through the shared descriptor space (i.e., 2D-3D matching) slows down and increases the probability of errors as the scene grows. In other words, there is a high possibility that the search space will increase and ambiguous matches will occur. In a recent study [15], researchers have proposed a hierarchical paradigm to form a synergy effect between the two approaches.

2.3.1 Prior retrieval

In a large-scale scene space, the shared descriptor space is quite large, so it is computationally intensive, and 2D-3D matching errors for camera pose estimation increase. In order to solve this problem, a prior-image search that can reduce the matching space will be used. The approximate

search at the map level is performed by matching the query image with the database images using global descriptors. Given a query image, K-closest images called prior frames are extracted using an image search method and the candidate location of the map is indicated.

As a result, it is considerably efficient because the candidate database images have been reduced much more than searching for the entire points of the SfM model.

In addition, it is necessary to cluster K-candidate images into similar categories in order to increase accuracy in local feature matching. To this end, images corresponding to co-visibility are connected from SfM and classified into one cluster. This is like finding a connected component called 'place' in a visibility graph that connects database images to a 3D point in a model. That is, a small sub-scene ('place') is identified by mapping to the connected components.

2.3.2 Local feature matching

Candidate images mapped to each place and clustered For each cluster, a 6-DoF pose is estimated by continuously matching the 2D keypoints detected in the query image with the 3D points included in each cluster to solve the problem of Person-n-Point (PnP) [41][51][52] performing geometric consistency checks within the RANSAC[39][53][54] scheme. If the most effective pose is estimated for each cluster, it is adopted and the algorithm is terminated.

2.3.3 Hierarchical localization pipeline

Section 2.3 describes Coarse-to-Fine localization by dividing it into parts by approach and method. Now, by synthesizing this, one pipeline configuration is shown in Figure 7.

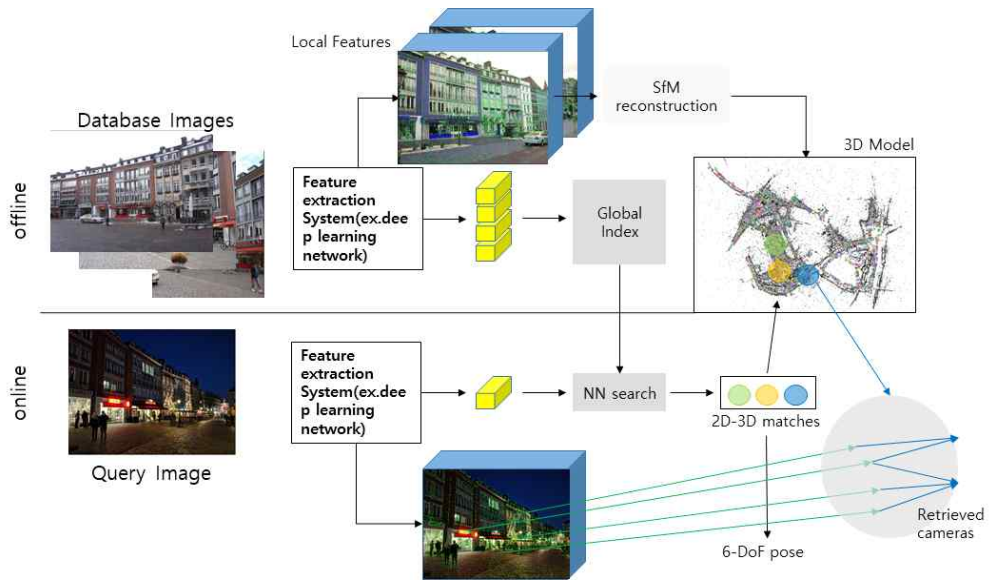

Fig 7. Pipeline of visual hierarchical localization.

3. Proposed Method

This chapter describes the proposed method. We describe the expression of the voxel-based scene for cohesiveness clustering and the algorithm for clustering based on it, and describe the deep learning network method used to evaluate it. Finally, the configured pipeline is presented and explained.

3.1 Introduction

In Section 2, we looked at hierarchical localization methods using deep learning. This method creates global and local descriptors of all DB images offline with a coarse-to-fine approach. When the query image enters the input, the global and local descriptors are extracted in the same way, first searching for images globally to cluster candidate images, and significantly reducing the range of matches in the region. Then, the local key points of the query image and the key points of the candidate images are matched to create a correspondence with the 3D points of the 3D model generated in SfM. Geometric consistency is found with PnP[41] with the 2D-3D correspondence pair configured in this way, and the camera pose is estimated.

This approach has shown the best performance in recent visual localization studies, and in this paper, we focused on enhancing the performance of this approach a little bit.

In recent studies on place recognition, NetVLAD[38], inspired by VLAD (Vector of Local Aggregated Descriptor) [42], one of the methodologies of image presentation used in image retrieval boasting state-of-the-art, is used to extract global descriptors from hierarchical localization. Even with this latest technology method, it is still composed of K candidates, so the method of finding K that is experimentally appropriate acts as a disadvantage. Even though K were found, only images at a similar view point to the query image are not selected and extracted exactly. In this paper, we focused on how to solve this problem.

Figure 8 below is the result of The Aachen Day-Night dataset, a large scale dataset used in this paper. Figure 8 below shows 50 candidate images for a query image and a query image using NetVLAD.

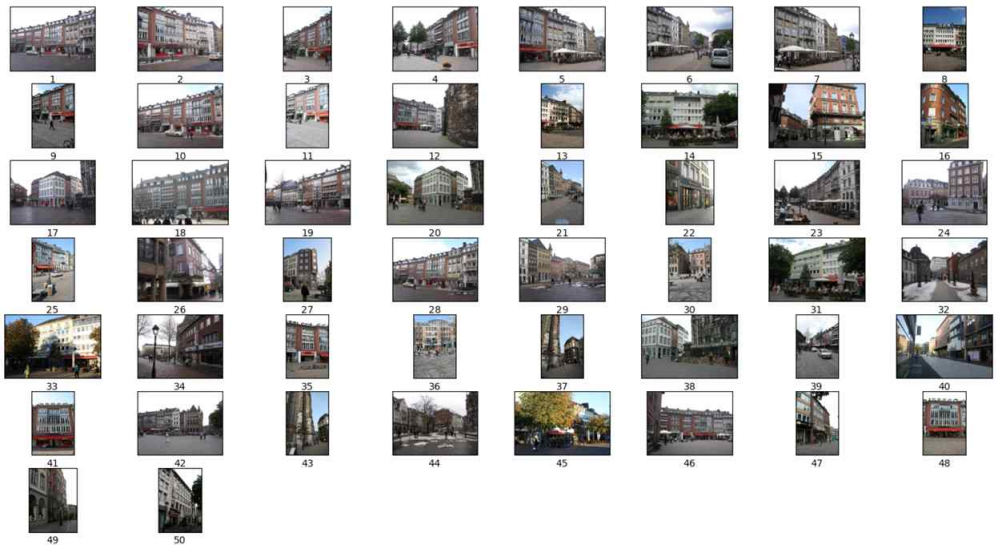

Fig 8. 50 global descriptor candidate images.

As shown in Figure 8, there are many different parts of the viewpoint (position) between 50 images. Compare this in more detail in Figure 9.

Looking at Figure 9, since it is a large-scale space, images from various viewpoints are searched even if reduced to 50 candidates. That is, even if the local descriptors for these 50 are matched with the query image, there is room for errors to occur. In order to solve this problem, this paper adopts the most appropriate pose estimation result by performing cohesiveness clustering to form clusters at similar view and matching each with a query image based on this.

Fig 9. 50 global descriptor candidate images.
(Top left : Query Image, Others : Candidate Images)

As suggested in [15], in the existing hierarchical localization method, concatenated 3D points observed at the same view are grouped into connected components. However, if you connect all the images in which 3D points are observed, even images at a point out of the query image can be grouped into one. The next section solves this part and explains the expression of 3D space for flexible clustering based on voxels, and then proposes an algorithm for clustering methods that can be clustered with images within a similar view point to a query image.

3.2 Voxel-based scene representation

The SfM 3D model previously provided by Dataset (based on Aachen dataset in this section) is provided by the dataset author. Popular SfM tools include COLMAP[44,45], Bundler[46], and VisualSFM[47]. Aachen data uses RootSIFT to build Structure from motion with COLMAP.

Referring to Fig. 10[15], it can be seen that the 2D projection points for the 3D model thus generated are gathered in specific parts. This can be an element of error when localizing. In addition, it is also a disadvantage in cohesiveness clustering presented in this paper. The reason for this is that using voxel-based scene expressions, points that can be viewed simultaneously are clustered based on points in the voxel, and as will be described in detail in the next section, it is advantageous that the points are evenly spread in the video. Key points detected by Super point [20] in Figure 10 are evenly spread throughout the image. For this reason, this paper reconstructs the SfM 3D model using Superpoint. A method of reconstructing the 3D model using the COLMAP tool is presented below.

1. 2D-2D matching is performed based on the key point detected by superpoint.
2. Matching items are further filtered within the COLMAP tool using two view geometry.
3. 3D points are triangulated using the ground truth camera pose of the previously provided dataset.

With this newly configured 3D model, a voxel-based scene dataset is built. In order to voxelize the 3D space by section, the scene is expressed as regular tessellation of cubes in Euclidean space. Each cube or voxel

divides the 3D points into six square planes of the same ratio to divide them evenly. It is also uniquely indexed with an identification number based on a global location.

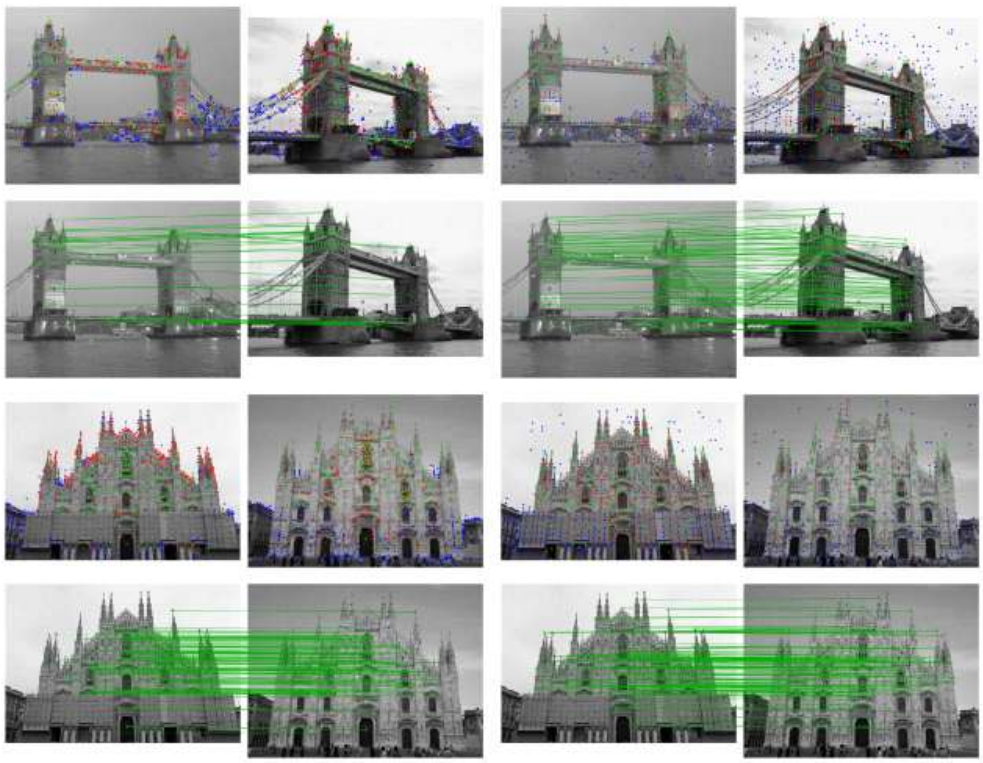

Fig 10. Qualitative results on the SfM dataset for
SIFT (left-two columns), SuperPoint (right-two columns)[15]

Referring to Figures 11 and 12, there are many points in the 3D model generated by the SIFT provided in advance that are very far from the cloud distribution of 3D points. These points are factors of error as an outlier. The 3D points reconstructed with the superpoint in Fig. 12 can be seen that severely distant points have been removed, and the voxel representation proposed in this paper has refined once more, confirming

that a considerable number of outliers, a noisy 3D point, have been removed. Then, the number and size of voxels are determined according to the scene volume and the number of 3D points.

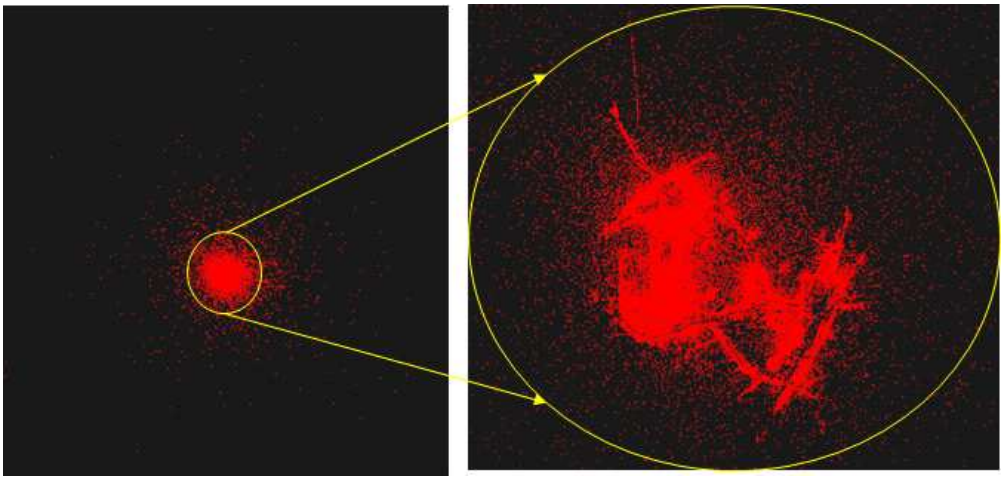

Fig 11. Aachen dataset - SfM 3D Points Configuration (top view) - Previously provided SfM model (by RootSIFT, COLMAP)

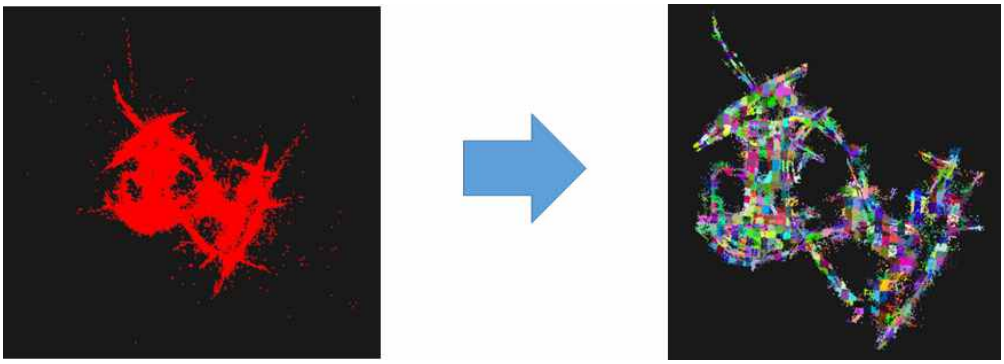

Fig 12. Aachen dataset - SfM 3D Points Configuration (top view) (by Superpoint, COLMAP), (left: before voxel division, right: after voxel division)

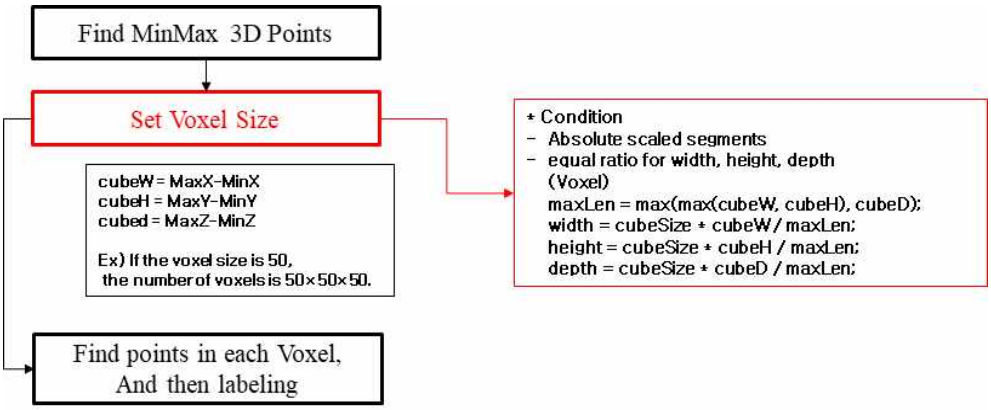

Fig 13. Determining Voxel Size & Labeling 3D Points in Voxels

Figure 13 shows the process of expressing 3D points in voxels. First, find the minimum and maximum values of the x, y, and z values of the 3D points, then make segments on an absolute scale according to the conditions presented in the figure according to the voxel size determined, and calculate width, height, and depth at an equivalent ratio. Find points entering each voxel and if they are below a specific threshold, view them as outliers and remove them. Then, label the remaining things and index them.

3.3 Covisibility Clustering

In this chapter, we propose a new algorithm that can cluster images from similar viewpoints, that is, images that can be seen at the same view, using voxel-based scene representation data.

3.3.1 covisible histogram

A histogram is a graph showing the frequency of variables by interval from some data information. Voxel information described in the previous section is used to understand the relationship between similar viewpoints between K reference images extracted from the global descriptor. In one voxel, one 3D points of one of the zones on the 3D space are gathered, and each voxel can explain the location of a specific space for each voxel. A histogram is a way to easily see where the 3D points of K candidate images are located.

First, voxels to which each of the 3D points of DB candidates belongs are listed as a series of lists. Then, the information in the list contains index information of voxel. Now, to obtain the histogram of the list, we set the bin size of the histogram that can contain information on each voxel by the number of voxel divisions obtained in Figure 12. In this histogram space, each voxel information in the list is stored and the index corresponding to the same place is counted. An example of the obtained histogram is shown in Figure 14.

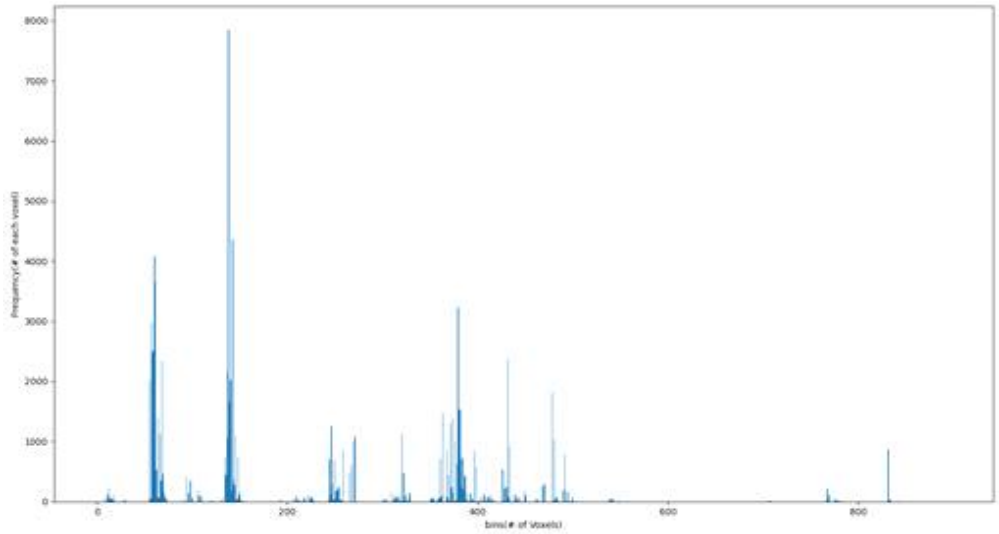

Fig 14. Voxel Hitogram

Analyzing Figure 14, it can be seen that a Gaussian distribution is formed around some voxels. This can be interpreted that DB candidate images are not only images with voxels in the part of the query image, but are composed of several images, and form a cluster around some voxels. In other words, it is an indicator of where DB candidates are seen a lot. The part where the histogram appears less frequently means that there are few parts that are actually seen in common in the image, so it is less important. Through this, it is possible to sort the frequencies of the histogram in descending order and organize the data in the order of strong importance.

However, if the images in which the voxels appear are clustered with only a histogram of T rankings, it can be a factor in performance degradation because the query image and voxels from a distance can be mixed, as shown in Figure 15 below.

This histogram is a simple form of non-parametric density estimation, but this alone is unreasonable to cluster. An appropriate clustering method of candidate DB images using histogram information will be described in the next section.

Fig. 15. DB images cluster in which voxels for T=20 ranking appear (top left: query image, others: conjugate images)

3.3.2 meanshift clustering

The histogram method has the point that discontinuity appears at the boundary of bin, the histogram varies depending on the starting position of bin, and it is difficult to use in high-dimensional data as a memory

problem. And as explained in the previous section, it is difficult to approach according to the purpose of this paper. The histogram in 3.3.1 can interpret the clustering parts, but since it is difficult to apply them thereafter, this paper proposes a clustering method by applying the KDE (Kernel Density Estimation) method using this information.

KDE is a method of estimating the probability density function by applying the kernel function to individual data points, summing all the return values, and dividing them by the number of individual data. Representatively, there is a Gaussian distribution function. Equation (12) is the KDE formula.

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \qquad (12)$$

h is the bandwidth parameter of the kernel function and serves as smoothing. That is, when the h value decreases, the kernel becomes sharp, and when the value increases, the kernel becomes a gentle shape. x is random variable, K is a kernel function, and n is the number of data.

Since each candidate image has a predetermined Field of View, a similar view exists in a proximity space. In the previous section, we checked the histogram distribution of voxels where 3D points exist, analyzed the exposure (importance) ranking of voxels, and confirmed that the wrong position away from the query image could sufficiently enter the ranking. To this end, it will be described to cluster images from similar viewpoints using a mean shift method using KDE in consideration of spatial proximity.

The operation process of Meanshift will be described with reference to Figure 16.[48]

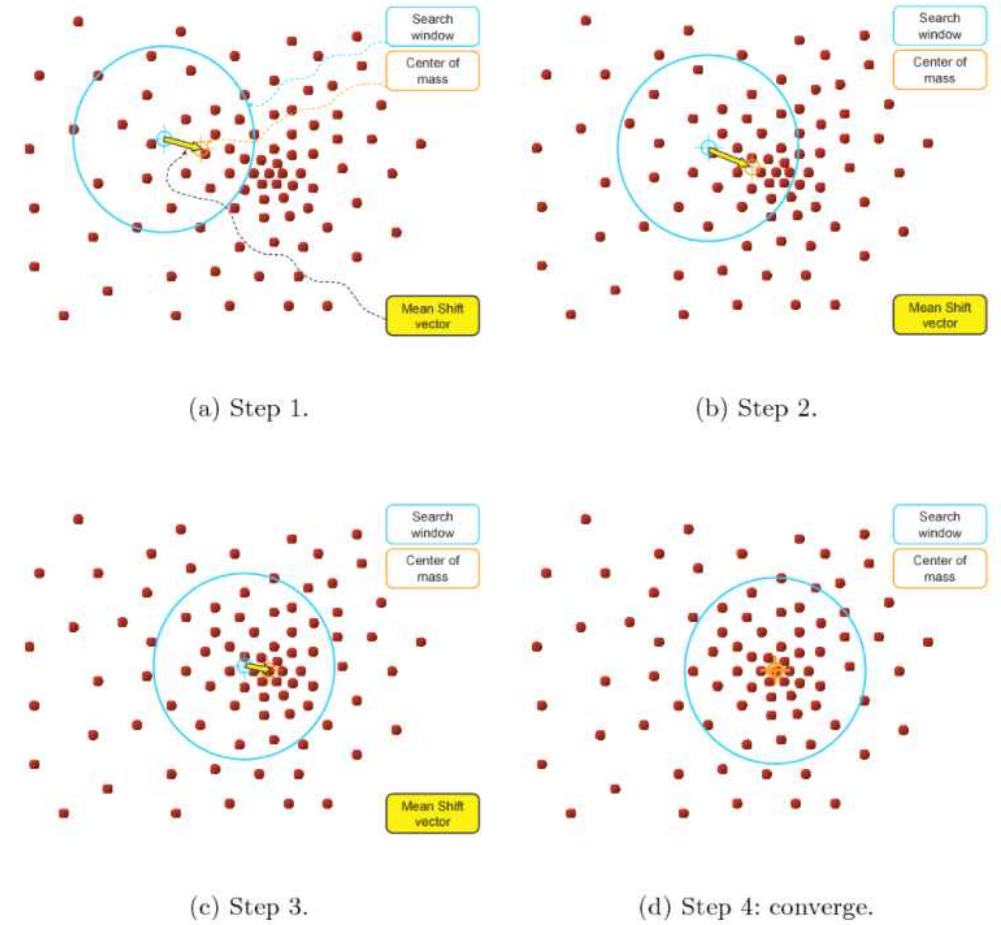

Fig. 16. The steps in mean shift algorithm using mass as an example.

One individual point moves within the radius of the sky blue circle to a place where the probability of data distribution is as high as possible. At this time, in order to find the "highest density" to move, the distance value from the surrounding data points is entered as the kernel function value, and the returned value is moved from the initial position to update the location.

Step 1: Calculate the data distribution diagram including surrounding data within a specific radius of individual data.

Step 2: Move the position of the data in the direction of the highest density among the calculated data distributions. In this case, the KDE function is used to find the most dense place.

Step 3: From the updated location, calculate the data distribution including surrounding data again within the radius like Step 1, and update the location to the most dense place using the KDE function like Step 2.

Step 4: Repeat Step 3 continuously and stop repeating when the location of the data is no longer updated (when it does not move).

Step 5: Apply the above process to all individual data one by one.

Assuming that T=20, the application of the above method in this paper is as follows. The histogram frequencies up to the top 20 rankings in Figure 14 are extracted to obtain voxels corresponding to them. Data is constructed by weighting the center coordinates of voxels obtained for meanshift clustering using spatial proximity in proportion to the number of each bin. Figure 17 shows the central coordinates by rank, starting from blue to first, and the position of voxels from red to 20th.

Figure 18 is a figure showing the position on the cluster by X after meanslift. Based on this result, Figure 19 shows the comparison of the cohesiveness clustering images with the cohesiveness clustering results of the previous [15] study. Since there are many cluster results, detailed results will be analyzed in the experimental results of Chapter 4, where only the best clustered images are compared to previous studies.

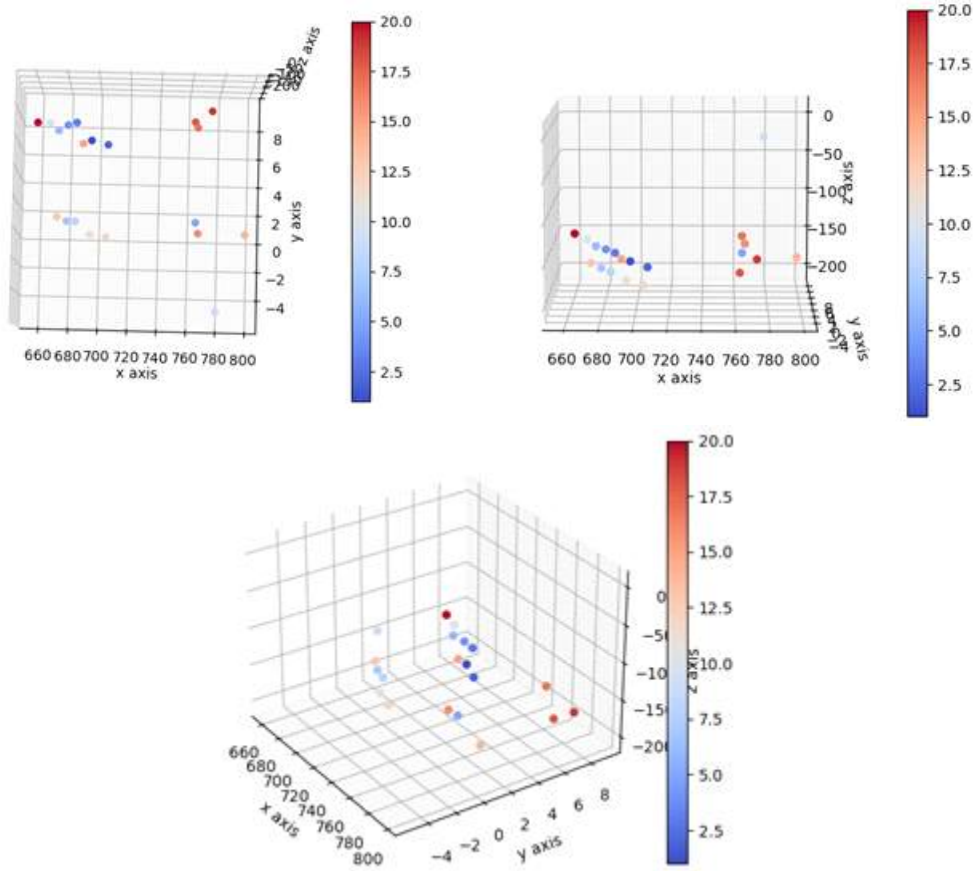

Fig. 17. In the voxel histogram, the center voxel coordinates up to the 20th rank.

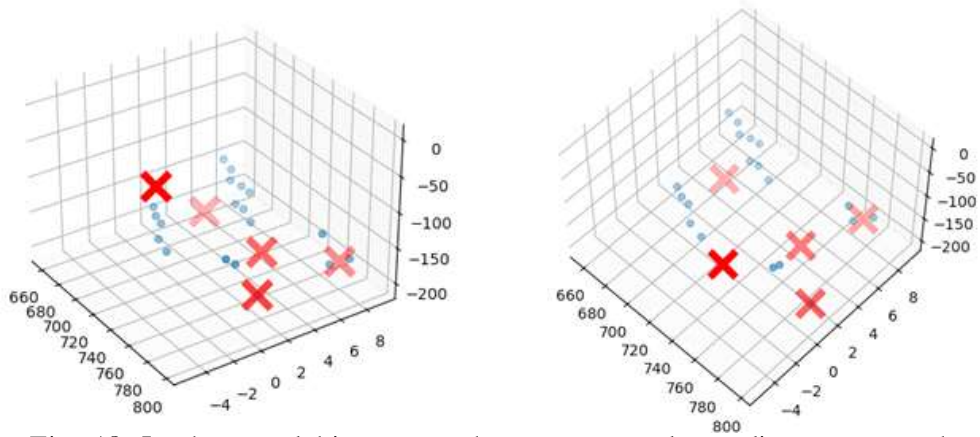

Fig. 18. In the voxel histogram, the center voxel coordinates up to the

20th rank and After meanshift, mark the position on the cluster with X.

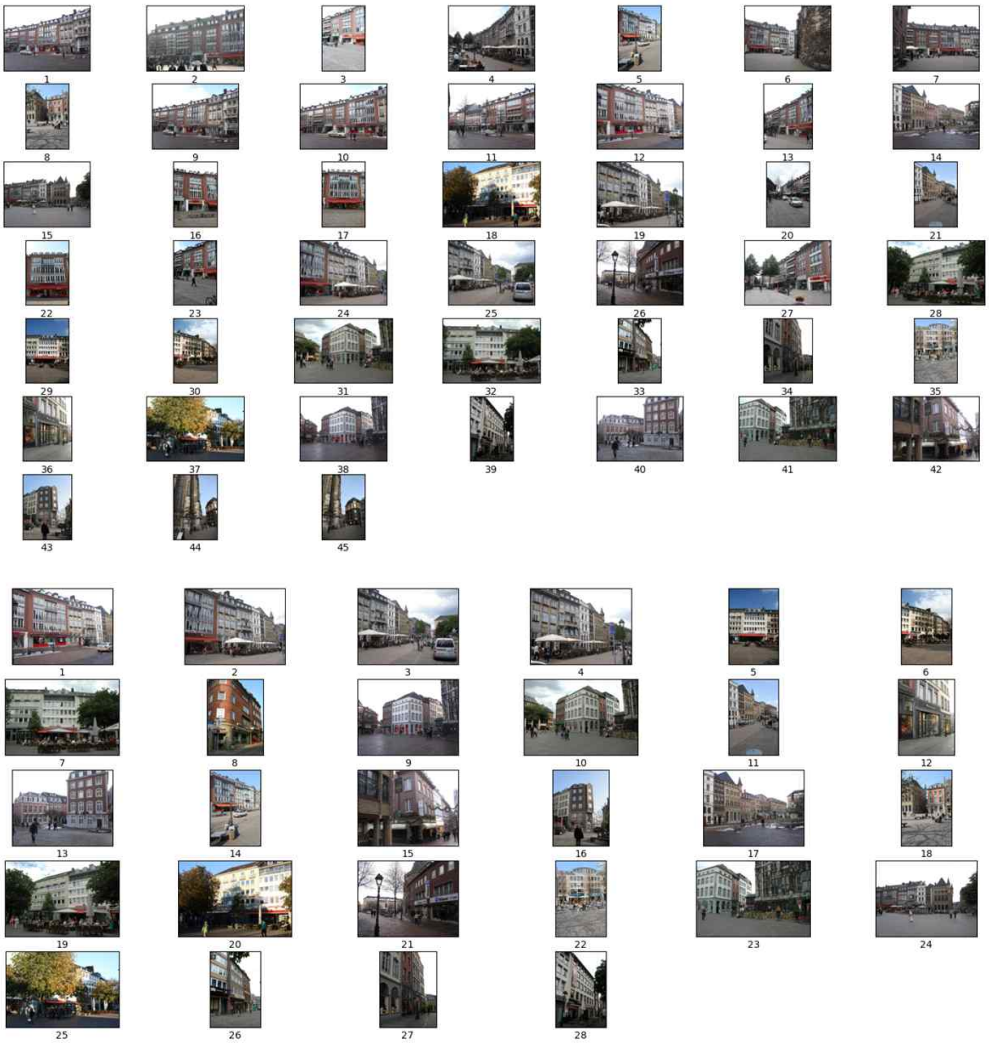

Fig. 19. covisibility cluster (Top : cluster of [15], Bottom : proposed cluster).

3.3.3 graph clustering

Meanshift clustering is greatly affected by the bandwidth parameter of the kernel function. The importance of the histogram distribution of voxels is that clustering using spatial proximity is also bound to be affected by this parameter. In other words, appropriate parameters must be found. It also affects the processing speed because it is performed by rotating all data within the bandwidth radius. The second proposed method makes a graph configuration for the covisibility relationship between voxels in which the covisibility of the scene can be considered before estimating the camera pose. By combining the voxel histogram with this voxel graph, a cluster is created by checking the connection elements of the voxel graph corresponding to the ranking. As a result, voxel graph clustering has a relatively shorter cluster creation time than meishift and is free to parameters such as bandwidth.To estimate the camera pose, candidate images extracted from the global descriptor collect local descriptors by checking whether there are voxels included in each cluster. After that, it is estimated by matching with local descriptors for the query image to create a 2D-3D correspondence and checking PnP geometric geometry consistency.

Below is a description of voxel graph clustering step by step.

step1. To calculate the availability for the entire voxel, DB images included in each voxel are stored. -> It is a task to check whether the corresponding voxel is in the selected DB image.

Step2. In the images included in each voxel, the number of voxels in which the corresponding voxels are simultaneously visible based on the

selected voxel is stored. For example, based on voxel No. 1, the number of other voxels visible simultaneously is obtained from DB images in which this voxel is visible. In other words, it is to calculate the covisibility.

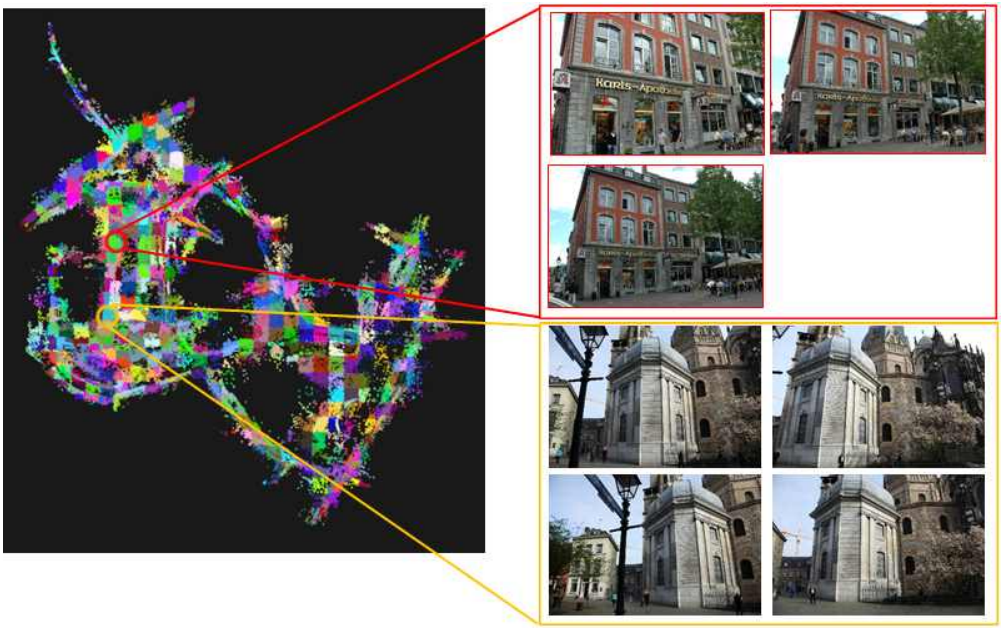

Fig. 20. step1: DB images included in each voxel.

step3. Based on the graph of each voxel, an average ratio table of covisibility for each level is generated, and the number obtained in step 2 is converted into a ratio. Since the number of coverage for each level is different for each voxel, if calculated in a ratio, each average ratio can be set and connected to threshold when searching for connected components.

Step4. Using the Breadth First Search (BFS) method and the information generated in step3, connected components voxels with the covisibility of each voxel above average are obtained. This means a cluster of reliable voxels that can be viewed simultaneously in the DB

image.

	Voxel 1	Voxel 2	Voxel 3	…	Voxel 896
Voxel 1	100	50	35		0
Voxel 2	50	90	31		0
Voxel 3	35	31	91		0
…					
Voxel 896	0	0	0		88

Table 2. step2 : The number of voxels in which corresponding voxels are simultaneously visible based on each voxel.

	level 1	level 2	level 3	…	level 20	level 21	level 22	…	level 30
Voxel 1	0.32	0.043	0.032		0.0	X	X	X	X
Voxel 2	0.68	0.069	0.048		0.0	0.0	X	X	X
Voxel 3	1.0	1.0	0.75		0.0	0.0	X	X	X
…									
Voxel 500	0.67	0.187	0.082		0.0	0.0	0.0	0.0	0.0
…									
Voxel 896	0.45	0.21	0.092		0.0	0.0	0.0	X	X
	Avg: 0.42	Avg: 0.16	Avg: 0.07		Avg: 0.0	Avg: 0.0	Avg: 0.0	Avg: 0.0	Avg: 0.0

Fig. 21. step2 : The number of voxels in which corresponding voxels

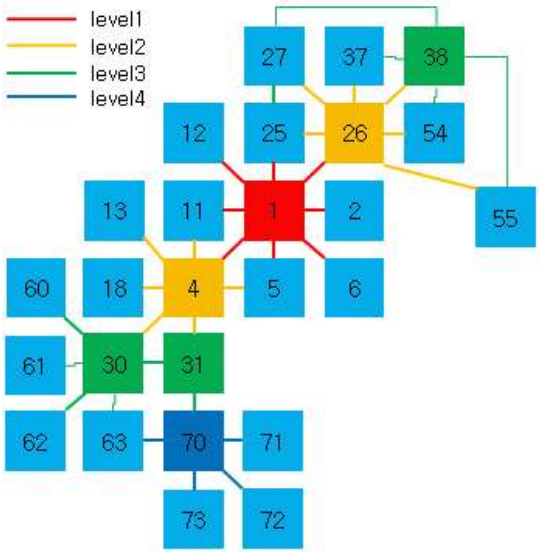

Fig. 22. step4: Connected components voxels with the covisibility of each voxel above averages

Figure 22 shows the results of connection only for edges above the average value of mobility for each level based on voxel 1. That is, that is the result of covisibility clustering for one voxel.

3.4 Camera pose estimation using the proposed method

This section describes the overall system configuration for estimating a camera pose using the proposed cluster method. As in the study of [15], in this paper, camera poses are estimated using hierarchical localization methods. As shown in Figure 7, the overall configuration is the same, and there is a difference in clustering work that reduces the search range for local matching by extracting global descriptors and detecting candidate DB images. In this paper, NetVLAD [38] is used as the global descriptor

extraction method, and SuperPoint [20] is used as the local descriptor extraction method.

Before explaining the overall system configuration, explain and move on to NetVLAD and SuperPoint utilized in the system.

3.4.1 NetVLAD : CNN architecture for weakly supervised place recognition[38]

There are three main points of this paper, first introducing NetVLAD network using CNN for place recognition, and secondly, we developed a learning procedure based on weak-supervised ranking loss to learn CNN architecture proposed using data extracted from Goolgle Street View Machine. Finally, it is suggested that they performed better in benchmark than existing methods.

It approaches and solves place recognition as an image retrieval problem, and defines a set of known images as {Ii} and a function that converts qery images into q and vectors of the way the image is expressed.

The image retrieval is to find the image Ii most similar to the image q in the set of images known (Ii} given the query image q. In this case, f and Euclidean distance d are used to compare images q and Ii. F(q) and f(Ii) are vectors of the same size, and the distance between the two can be expressed as d(q, Ii) using d. The similar degree of image q and image Ii is expressed as the distance between vectors. If the vector distance is short, the two images are similar, and if it is far, it can be said that they are not similar.

To accurately compare images, it is important to make a function f well, and NetVLAD learns CNN architecture and uses it as a function f, and expresses CNN's parameter as θ of function f.

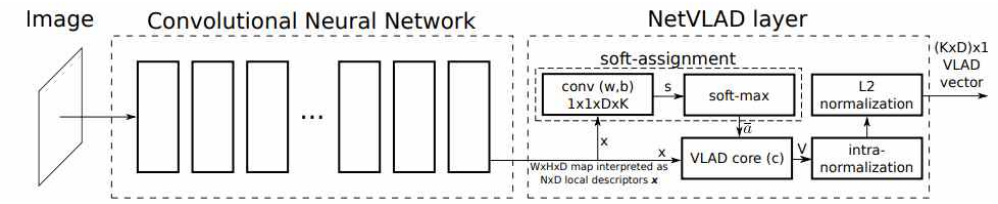

Fig. 23. CNN architecture with the NetVLAD layer.

CNN layer applies transfer learning, extracts only convolution layer excluding pooling layer, and uses it as a sense descriptor extractor.
The NetVLAD layer serves to pool the local image descriptor {xi} with N D sizes output through the convolution layer. Given this {xi} and K cluster centers ("visual words") {ck}, the formula for VLAD image presentation V with a K x D size is as follows.

$$V(j,k) = \sum_{i=1}^{n} a_k(x_i)(x_i(j) - c_k(j)) \qquad (13)$$

{xi} shows that all elements of this set have D elements, respectively, corresponding to the convolution layer output x on CNN in Figure 20. Visual word, a cluster centre, represents an image by calculating the difference between the center points of the cluster, which corresponds to $(x_i(j) - c_k(j))$ is 1 when xi is the closest cluster center of all {ck} and 0 otherwise. In other words, it is 1 only when xi and ck correspond to the same cluster, and 0 if they belong to different clusters. Eventually, V(k) is

{xi} Among them, the difference between the xi corresponding to the k-th cluster and the cluster center of the cluster is added.

In Learning from Time Machine data, to learn in an end-to-end manner, Google Street View Time Machine collects and solves data about the same places photographed at different times and seasons, and develops and solves the weakly supplied triplet tracking loss function.
Google Street View Time Machine is a training dataset that has GPS information at the moment of taking a picture.

The Euclidean distance $d_\theta(q,I)$ between the query image q and the close image Ii* should be less than the distance from other images in Ii. This is expressed by the formula as follows.

$$d_\theta(q,I_i^*) < d_\theta(q,I_i) \tag{14}$$

To satisfy the above formula, the paper proposes running loss between training triplets {q, Ii*, Ii} as a loss function. For the loss function, the learning data may be configured as a tuple. $(q,\{p_i^q\},\{n_j^q\})$ Here, q is a query image, $\{p_i^q\}$ is a positive image, and $\{n_i^q\}$ is a definitive negative image. And then, the following equation is defined.

$$p_i^{q*} = \arg\min_{p_i^q} d_\theta(q,p_i^q) \tag{15}$$

Equation 16 below is expressed,

$$d_\theta(q,p_i^{q*}) < d_\theta(q,n_j^q),\ \forall j \tag{16}$$

Based on this, weakly supervised tracking loss L_θ is defined as follows.

$$L_\theta = \sum_j l(\min_i d_\theta^2(q,p_i^q) + m - d_\theta^2(q,n_j^q) \quad (17)$$

The function l represents l(x)=max(x,0), and m represents a constant parameter and margin. Finally, this loss learns in a Stochastic Gradient Descent (SGD) method.

3.4.2 SuperPoint: Self-supervised interest point detection and description[20]

The method in this paper presents a self-supervised framework to learn point detectors and descriptors of interest. The VGG network is used as a backbone, and the location of the keypoint and descriptor information are calculated jointly.

Since Superpoint aims to match good keypoints in multiple image pairs, it is important to extract only well-matched keypoints. In order to learn this, in the learning process, an image pair matched well with ground truth data is required. Therefore, as a pretext task, the network learns the function of finding key points in a simple environment, and proceeds in the corresponding way (a) in Figure 21. Virtual rendering was used to learn the network based on image data and vertex location data of the rendered model, and virtual image data adds a variety of lights and noise to enhance robustness. To transfer the learned results to the real world dataset (transfer), the MS-COCO dataset [49] is re-learned. Here, there is no label for the key point. To make this label, we use a technique called homographic adaptation(figure 22).

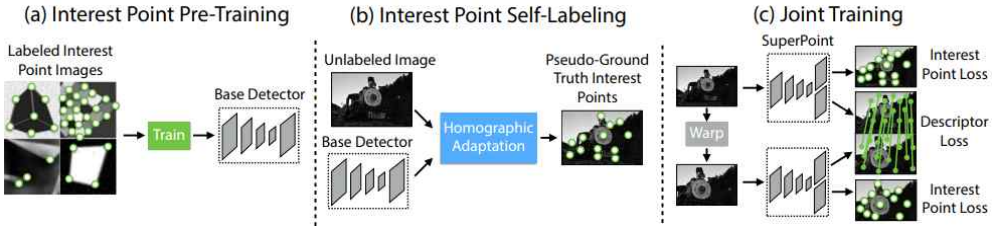

Fig. 24. Self-Supervised Training overview[20].

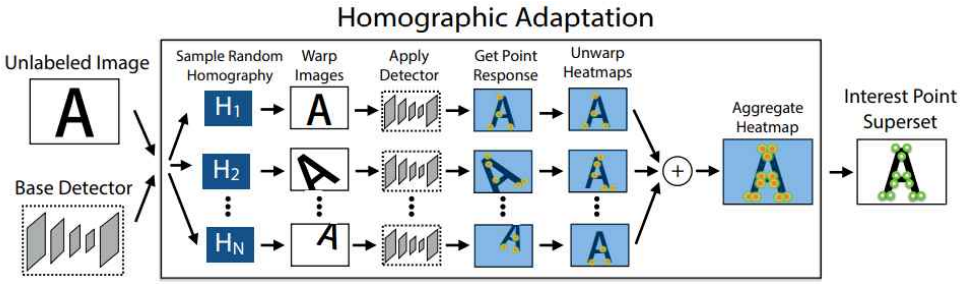

Fig. 25. homographic adaptation[20].

If the camera motion between the two images can be known, keypoint positions on one image can be moved over the other. In other words, it becomes a self-labeling technology. This self technique can avoid false keypoint detection and improve the repeatability of keypoint detection (i.e., ability to detect keypoints at the same location even from different angles).

Homeography is applied to the original image without a label to generate images viewed from various angles, and key points are extracted by applying MagicPoint (Figure 24(a) to each image. When these key points are transferred back to the original image by homography transformation, a robust set of key points are created.

Now describe the loss function for learning.

$$L(X,X',D,D';Y,Y',S) = L_p(X,Y) + L_p(X',Y') + \lambda L_d(D,D',S) \quad (18)$$

Lp and Ld are point of interest detectors and descriptor detectors, respectively.

Lp assumes that there is only one key point in the patch of size 8x8. That is, in the 8x8 patch, there are 64 candidate pixels that may be key points. Here, 65 distributions are created by adding one candidate named 'dustbin', which acts as a garbage value rather than a key point.

$$L_p(X,Y) = \frac{1}{H_c W_c} \sum_{h=1,w=1}^{H_c W_c} l_p(x_{hw};y_{hw}) \quad (19)$$

$$l_p(x_{hw};y) = -\log\left(\frac{\exp(x_{hwy})}{\sum_{k=1}^{65}\exp(x_{hwk})}\right) \quad (20)$$

$$L_d(D,D',S) = \frac{1}{(H_c W_c)^2} \sum_{h=1,w=1}^{H_c W_c} \sum_{h'=1,w'=1}^{H_c W_c} l_d(d_{hw}, d'_{h'w'}; s_{hwh'w'}) \quad (21)$$

$$l_d(d,d';s) = \lambda_d * s * \max(0, m_p - d^T d') + (1-s) * \max(0, d^T d' - m_n) \quad (22)$$

That is, in the 8x8 patch, there are 64 candidate pixels that may be key points. Here, 65 distributions are created by adding one candidate named 'dustbin', which acts as a garbage value rather than a key point.

λ_d is a parameter that balances mp and mn with a weighting term.

And equation (21) uses hinge loss with positive margin mp and negative margin mn.

3.4.3 Overall System Configuration Chart

This section explains the network structure for evaluating the proposed method and examines where the method of this paper was used.

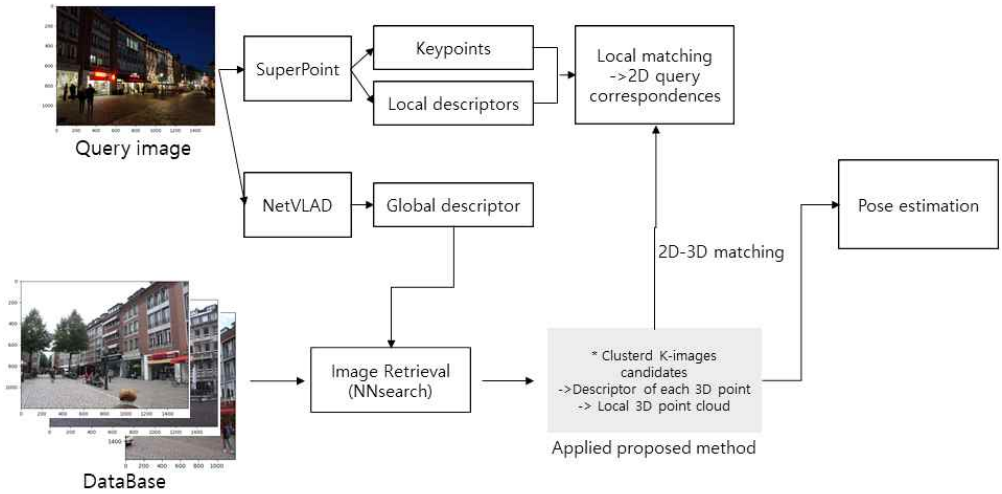

Fig. 26. Hierarchical localization system using proposed method.

In Figure 26, the gray area is the section to which the proposed method is applied. In that part, the clusters obtained in Section 3.3 collect descriptor information for each cluster. And matching the 2D local descriptor of the query image, the pose of the camera from the 2D-3D correspondence relationship is estimated by checking the PnP geometry consistency within the RANSAC scheme.

4. Experiments and Results

In this chapter, descriptions of datasets and experimental results are described and performance is analyzed.

4.1 Experimental Environment

The computer experiment environment is as follows.

CPU : Intel i9 2.9GHz, 32GB RAM(64bit OS-Ubuntu 18.04)

All query images are resized to gray scale and the maximum resolution size is 1024, and input.

For all datasets, the dimensions of the NetVLAD global descriptor were reduced to 1024 dimensions using PCA. In addition, weights learned with the Pitts-30k dataset were used, and the number of search videos K was set as 50.

Superpoint is applied with a non-maximum-suppression (NMS) of a keypoint radius 3 detected in the query image, of which about 2K is maintained.

4.1.1 Aachen Dataset[43]

Aachen Day-Night dataset describes an old city center in Aachen, Germany. It contains 4,328 images of the old city's weekly database and 824 and 98 queries, respectively, under day and night conditions.

4.1.2 Cambridge Landmarks Dataset[3]

The Cambridge Landmarks dataset is a mid-scale urban outdoor dataset. It includes four scenes, ranging in their spatial extent from 35x25m (Shop Façade) to 140x40m (King's College). King's College scene from Cambridge Landmarks, a large scale outdoor visual relocalisation

dataset taken around Cambridge University.

4.2 Experimental results and analysis

In this chapter, the results of Voxel-based scene presentation and performance evaluation by voxel size are performed, and the results of voxel histogram are compared and analyzed.

4.2.1 Voxel-based scene representation

An experiment is conducted on the ratio of clusters to the top T rankings of the histogram according to the size of Voxel, and the performance is compared.

Referring to Figure 27, the reason why voxel expression is necessary will be explained.

Figure 27 shows the false matching between the query and the DB image, which can cause an error in pose estimation because it is determined that the match has been made. And in Figure 27, the viewpoint between the query image and the DB image is completely different. That is, since the positions are different, if this is divided into voxel regions, matching for these parts can be filtered.

Figure 28 is a voxel segmentation area in which a distant outlier is removed from the original 3D model for a performance comparison experiment according to the sizes of two types of voxels. It is compared and analyzed by dividing it into the number of large voxels and the number of small voxels.

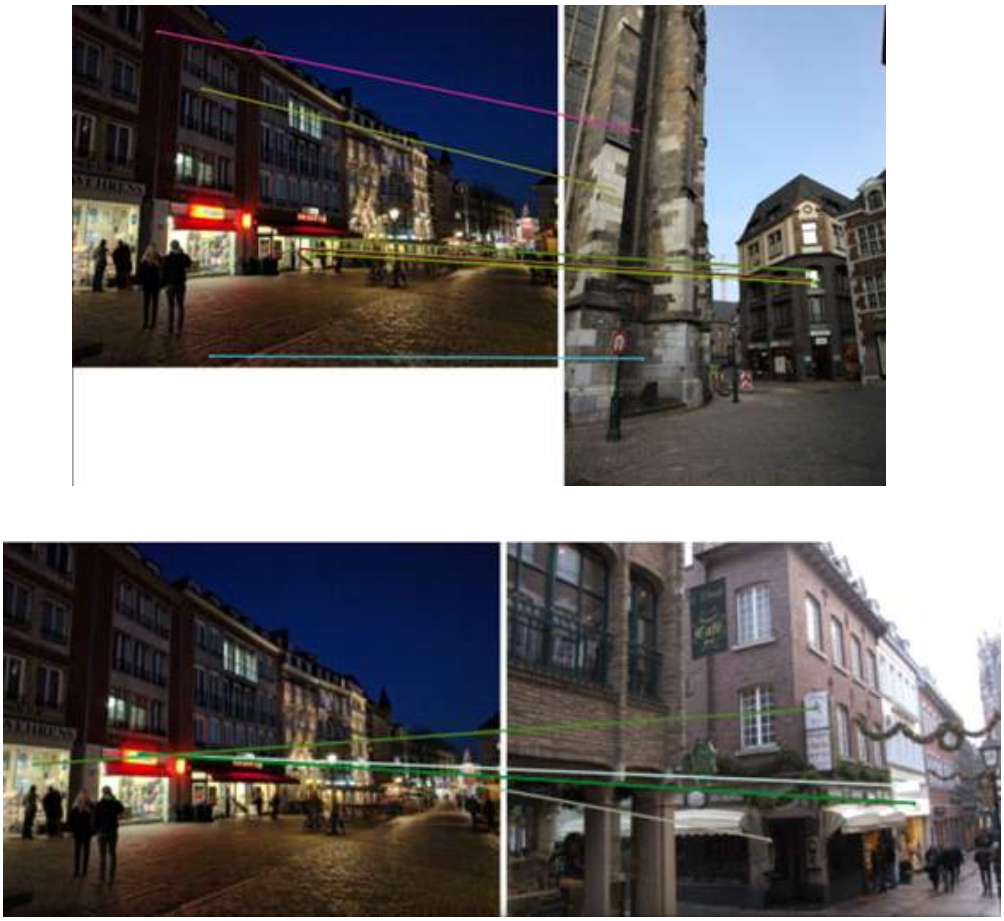

Fig. 27. False matching.

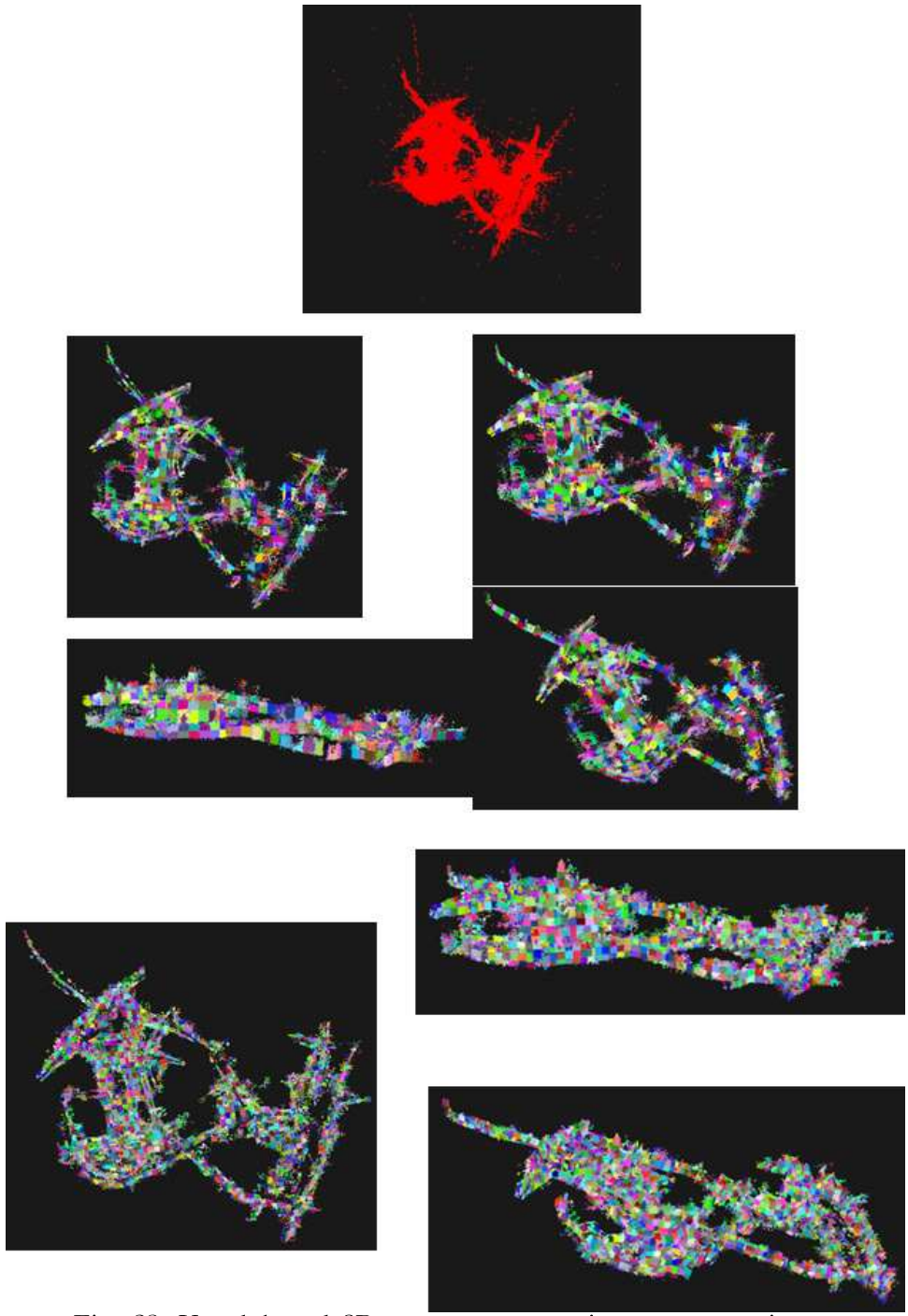

Fig. 28. Voxel-based 3D space segmentation representation (Top:Original 3D points, Middle:898 voxels, Bottom: 3239 voxels).

4.2.2 Voxel-Hitogram analysis

In this section, we analyze the histogram shown simultaneously in the candidate DB image, and analyze whether it is true that the performance increases by subtracting the reference image corresponding to the outlier presented in this paper.

To prove this, the camera pose is estimated by grouping the DBs included in the histogram by T rankings, and the number of re-photographing errors and inliers is measured and verified. Table 1 shows the measurement results.

	# of inlier	reprojection error
T=5 (GT:25)	158	0.341
T=5 (Included:26)	157	0.354
T=10 (GT:27)	156	0.345
T=10 (Included:38)	136	0.351

Table 3. Verification result for pose estimation based on the number of histogram T rankings

In Table 3, GT is the result of verification measurement by estimating the number of similar view to the actual query image, and Included is the

number of images of voxels included in the T ranking, and the result of estimating the pose and measuring it.

Based on Figure 29, the number of images passing through when there are voxels belonging to T=1, 5, 10, and 20 respectively is the number of images at the time when 14, 27, 32, and 42 out of 50 can be viewed simultaneously in order. Looking at the histogram graph, there are several peak values, so the more the number of rankings is selected, the more images are co-observed. When T=1, as shown in Figure 30, the passed image is all similar to the query image. However, in the five T=s, there are more images at a similar view point than that, so only the top voxels should not be selected as views visible at the same time and clustered later. That is, if appropriate clustering is performed on the remaining T except for voxels with little frequency, clusters at similar views may be extracted.

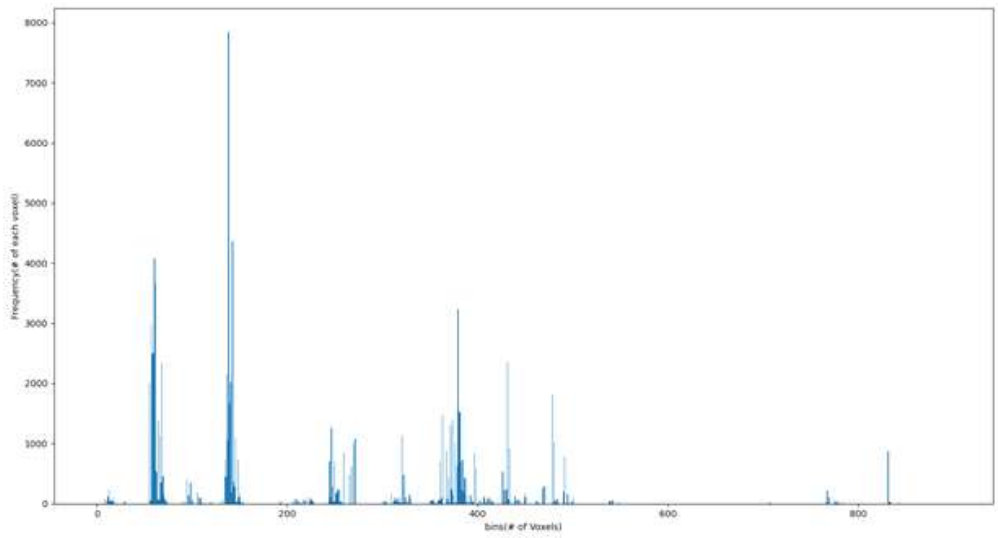

Fig 29. Voxel Hitogram

Fig. 30. T=1 - 14 passed images

4.2.3 clustering analysis

Figure 31 shows the position of voxels in the coordinate system in the rankings of T=20, 100.And the clustered center was marked with 'X'. It is expressed that blue starts as the first priority, and the ranking increases as it goes to red.

Images of first-ranked voxels do not always share similar points of view the most. Figure 32 shows the situation. Since the 2,3,4,6, and 7th-ranked cohesiveness is higher when grouped into clusters, voxels containing the most images in the cluster are not the first-ranked class, but second-ranked clusters.

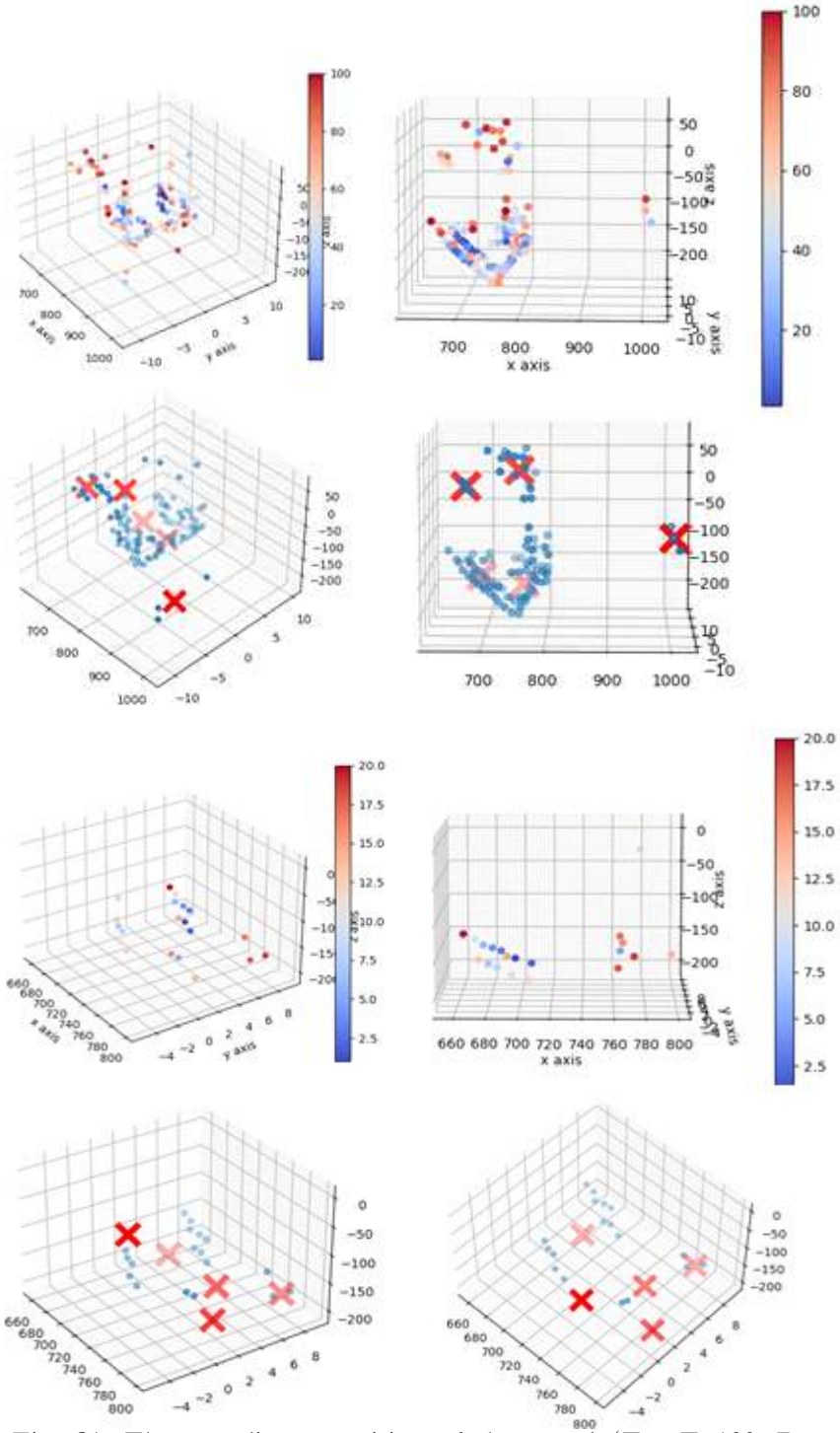

Fig. 31. The coordinate position of the voxel (Top:T=100, Bottom:T=20)

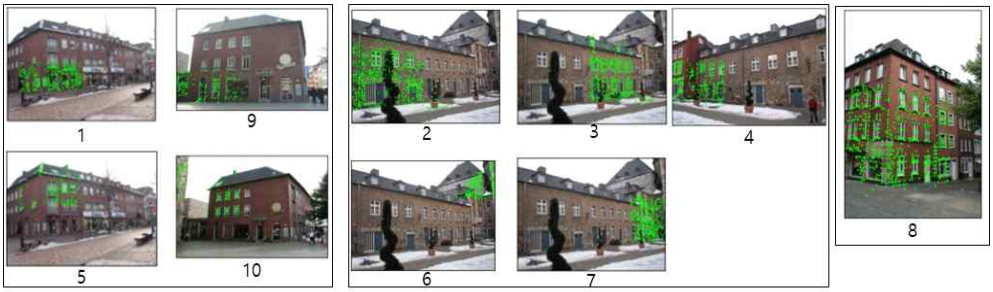

Fig. 32. Images in which the corresponding voxel appears up to the T=10 ranking (number represents the ranking)

Figure 33 shows the image cluster results for global descriptor candidates using mean shift clustering and graph clustering.

In the voxel histogram, candidate images that are well grouped based on the top 20 voxels are shown.

Method	**Day (0.25/0.5/5.0 m) (2/5/10 °)**	**Night (0.25/0.5/5.0 m) (2/5/10 °)**
Hloc-covisibility best set[15]	86.4 / 94.3 / 97.8	69.4 / 83.7 / 95.9
Hloc-nonCovis All descriptor[15]	86.3 / 93.6 / 97.8	71.4 / 83.7 / 95.9
T=20 meanshift(0.15)	86.3 / 94.3 / 97.7	69.4 / 83.7 / 94.9
T=50 meanshift(0.15)	86.0 / 93.9 / **98.1**	72.4 / **85.7** / 94.9
T=100 meanshift(0.15)	86.0 / 93.7 / 97.9	**74.5** / 84.7 / 95.9
graphCluster(T=20, intersection>0.5)	86.8 / 94.2 / 97.9	71.4 / 84.7 / 95.9
graphCluster(T=50, intersection>0.5)	**86.9** / 94.3 / 97.9	71.4 / 84.7 / 95.9
graphCluster(T=100, intersection>0.5)	86.8 / **94.4** / 97.9	71.4 / 84.7 / 95.9
graphCluster(T=20, Allpass)	86.8 / 94.2 / 97.8	72.4 / 83.7 / 95.9

graphCluster(T=50, Allpass)	86.8 / 94.2 / 97.9	73.5 / 84.7 / 94.9
graphCluster(T=100, Allpass)	86.7 / 94.2 / 97.9	73.5 / 84.7 / 94.9

Table 4. Pose estimation results according to voxel reference images in clusters

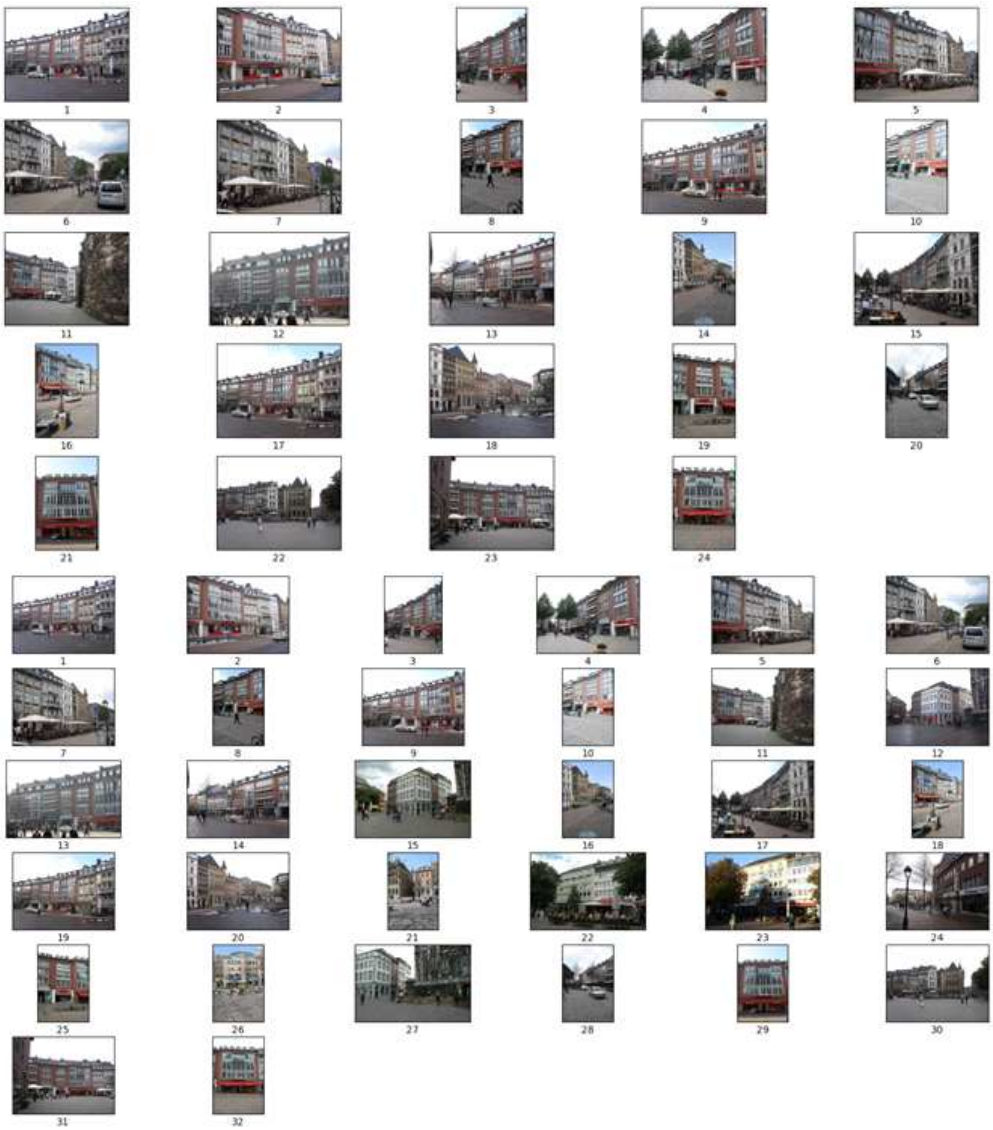

Fig. 33. covisibility cluster (Top : proposed meanshitf cluster, Bottom : proposed graph cluster).

Compared to the existing coviability method, it was confirmed that both proposed methods obtained image clusters at a similar time to the query image, and Table 4 shows the results of comparative experiments according to parameter control of parameters.

The estimation results by mean shift and graph clustering are better compared to calculations that do not consider covisibility and covisibility in previous studies [15]. In addition, the comparison between meanshift and graph cluster showed better performance on Day and meanshift on Night, resulting in differences due to the number of clusters of global candidate images passed through each voxel cluster. There are more clusters in the graph, which is difficult to match descriptors in the night time domain, so the clustered cluster should generate image candidates for a slightly wider view. That's why the performance was better. The table below shows the average number of clusters for passed image candidates.

Method	mean clusters cnt	std clusters
T=20,meanshift(0.15)	6.13	1.45
T=50,meanshift(0.15)	5.61	1.5
T=100,meanshift(0.15)	4	1.4
graphCluster(T=20, intersection=0.5)	4.3	1.8
graphCluster(T=50, intersection=0.5)	7.3	2.74
graphCluster(T=100, intersection=0.5)	10.4	3.11
graphCluster(T=20, intersection=0.0)	1.77	0.78
graphCluster(T=50, intersection=0.0)	2.42	1.19
graphCluster(T=100, intersection=0.0)	2.9	1.32
graphCluster(T=20, Allpass)	20	0
graphCluster(T=50, Allpass)	50	0
graphCluster(T=100, Allpass)	100	0

Table 5. The number of clusters of global candidate images passed through the Voxel cluster.

4.2.4 camera pose estimation results

First, Table 6 is the result of grouping and matching co-observed images into one descriptor based on voxels belonging to the top T in the histogram.

Method	Day (0.25/0.5/5.0 m) (2/5/10 °)	Night (0.25/0.5/5.0 m) (2/5/10 °)
Hloc-HFnet covisibility	86.4 / 94.3 / 97.8	69.4 / 83.7 / 95.9
Hloc-nonCovis All descriptor	86.3 / 93.6 / 97.8	71.4 / 83.7 / 95.9
Hloc-Voxel Histo(T=1)	74.8 / 80.9 / 83.7	66.3 / 76.5 / 86.7
Hloc-Voxel Histo(T=5)	83.7 / 91.1 / 94.8	71.4 / 82.7 / 92.9
Hloc-Voxel Histo(T=10)	85.3 / 92.4 / 96.4	72.4 / 84.7 / 94.9
Hloc-Voxel Histo(T=20)	86.2 / 93.4 / 97.3	70.4 / 83.7 / 94.9

Table 6. Pose estimation results according to voxel reference images in the top T histogram.

In Table 6, the top two methods were the results of PnP estimation with the largest number of inliers when the 3D points presented by HFnet[15] were shared as clusters, and the second row was the result of PnP estimation by collecting and matching all local descriptors for K=50 candidates. The remaining rows are the result of PnP estimation by simply collecting descriptors of images in which voxels belonging to the top T appear before performing the mean shift. There are some parts that show some good performance in night query, but overall, the performance is lower than that of the upper two rows. The reason is that when T is

1, many images at a similar view point to the query image come in, but the number is small, and images at different images can be selected. Conversely, if there are many views at different points in the top T, the matching error may increase accordingly. Therefore, a proposed method considering spatial proximity was devised. Table 7 below shows the estimation results.

The results of clustering differ according to the bandwidth value of the mean shift. Meanshift is performed by estimating bandwidth using KNN techniques, for example, if the total number of data is 100 and the estimation parameter is 0.3, knn is performed on 30 subjects. Thereafter, the bandwidth is determined based on the average pair-wise distance between data within the same cluster. In this paper, it was experimentally confirmed that 0.15 showed the best performance. It is shown in Table 6 based on T=100. (If it's too small, you can't get the bandwidth.)

Method	Day (0.25/0.5/5.0 m) (2/5/10 °)	Night (0.25/0.5/5.0 m) (2/5/10 °)
Hloc-covisibility best set	86.4 / 94.3 / 97.8	69.4 / 83.7 / 95.9
Hloc-nonCovis All descriptor	86.3 / 93.6 / 97.8	71.4 / 83.7 / 95.9
Hloc-Voxel Histo(T=1)	74.8 / 80.9 / 83.7	66.3 / 76.5 / 86.7
Hloc-Voxel Histo(T=5)	83.7 / 91.1 / 94.8	71.4 / 82.7 / 92.9
Hloc-Voxel Histo(T=10)	85.3 / 92.4 / 96.4	72.4 / 84.7 / 94.9
Hloc-Voxel	86.2 / 93.4 / 97.3	70.4 / 83.7 / 94.9

Histo(T=20)		

Table 7. Voxels belonging to the top T histogram and pose estimation results according to the mean shift cluster.

Method	Day (0.25/0.5/5.0 m) (2/5/10 °)	Night (0.25/0.5/5.0 m) (2/5/10 °)
Hloc-covisibility best set	86.4 / 94.3 / 97.8	69.4 / 83.7 / 95.9
Hloc-nonCovis All descriptor	86.3 / 93.6 / 97.8	71.4 / 83.7 / 95.9
p=0.25	86.3 / 93.9 / 97.7	72.4 / 84.7 / 95.9
p=0.20	86.3 / 93.9 / 97.9	72.4 / 84.7 / 95.9
p=0.15	86.3 / 93.9 / 98.1	72.4 / 84.7 / 95.9

Table 8. Voxels belonging to the top 100 histogram T=100 pose estimation results according to the mean shift cluster (parameter adjustment)

Table 9 shows comparisons with benchmark [58].

The method of [51] in Table 5 is the same as the system structure used in this paper, but there is a slight difference in results and whether it is the difference in hyper-parameter setting. For fair comparison, the results of Hierarchical Localization - superpoint+superglue performed under the same conditions in our code showed better performance on day by adding the proposed method as 89.0/95.5/98.7, 85.7/92.9/100.0 (day, night).

Method	Day (0.25/0.5/5.0 m) (2/5/10 °)	Night (0.25/0.5/5.0 m) (2/5/10 °)
Hierarchical Localization - SuperPoint +	89.6 / 95.4 / 98.8	86.7 / 93.9 / 100.0

SuperGlue[51]		
SuperGlue + Patch2Pix (HLoc) [55]	89.2 / 95.5 / 98.5	87.8 / 94.9 / 100.0
KAPTURE-R2D2-FUSION [56]	89.4 / 96.4 / 99.2	84.7 / 92.9 / 98.0
ours -Hloc SuperPoint + SuperGlue(T=100, p=0.20)	89.3 / 96.0 / 98.7	85.7 / 92.9 / 100.0
ours - Hloc SuperPoint (T=100, p=0.15)	86.3 / 93.9 / 98.1	72.4 / 84.7 / 95.9
NV+SIFT [15]	82.8 / 88.1 / 93.1	30.6 / 43.9 / 58.2
NV+SP [15]	79.7 / 88.0 / 93.7	40.8 / 56.1 / 74.5
HF-Net [15]	75.7 / 84.3 / 90.9	40.8 / 55.1 / 72.4
Active Search [57]	57.3 / 83.7 / 96.6	19.4 / 30.6 / 43.9

Table 9. Results of comparison with benchmarks.

An experiment was also conducted in a Cambridge dataset with a medium scale space slightly smaller than the scale of a large space, such as Aachen dataset. Table 10 shows the comparison results. Up to the top 8 rows are the results of the method of this paper according to the T rank, and the last 4 rows are the results of the method using HF-net's covisibility.

It is contradictory to the results of HFnet's visibility and the proposed method, and Figure 34 shows that all images are at a similar view. Even if the position of the voxel is far away, it can be seen that there is no difference in performance because almost all of the top 50 candidate DB images come in even if clustering is done well because it enters the camera view point.

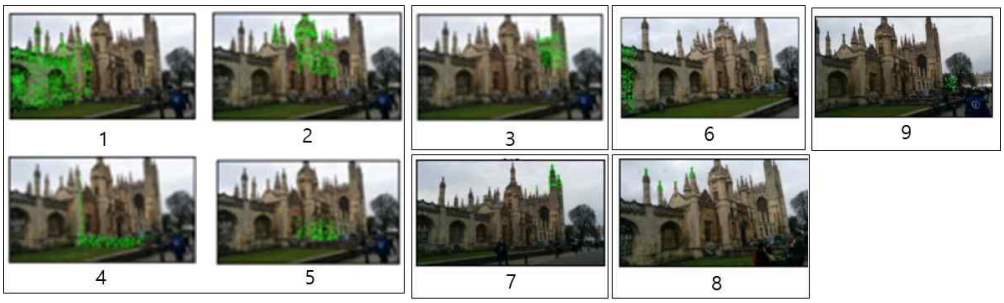

Fig. 34. Images in which the corresponding voxel appears up to the T=10 ranking (number represents the ranking)

Method	Day (0.25/0.5/5.0 m) (2/5/10 °)	Median translation (m) and ratation (°)
king's college - SP(T=100)	74.34 / 90.67 / 100	0.118 / 0.212
OldHospital - SP(T=100)	67.78 / 89.01 / 100	0.146 / 0.289
ShopFacade - SP(T=100)	95.15 / 98.06 / 100	0.042 / 0.222
StMarys Church - SP(T=100)	97.79 / 99.62 / 99.81	0.073 / 0.218
king's college - SP(T=10)	74.34/ 92.43 / 100	0.118 / 0.208
OldHospital - SP(T=10)	68.13 / 89.56 / 100	0.144 / 0.301
ShopFacade - SP(T=10)	96.12 / 98.06 / 100	0.040 / 0.195
StMarys Church - SP(T=10)	96.23 / 99.62 / 99.81	0.076 / 0.225
king's college - SP(HF Covis)	74.76 / 92.13 / 100	0.116 / 0.208
OldHospital - SP(HF Covis)	69.23 / 89.56 / 100	0.150 / 0.289
ShopFacade - SP(HF Covis)	96.12 / 98.06 / 100	0.042 / 0.221
StMarys Church - SP(HF Covis)	96.23 / 99.62 / 99.81	0.073 / 0.219

Table 10. Voxels belonging to the top T histogram and pose estimation results according to the mean shift cluster.

Table 11 analyzes the improvement performance of the proposed method by applying various combinations of global and local technicians in the hierarchical approach process.

DIR[59] as a global descriptors and R2-D2[56] as a local descriptors were further considered. In other words, it is an experiment in which the proposed method was applied in a total of six cases, including two global engineers and three regional engineers.

The results of the global descriptors were better because images at view point to the query image were extracted as candidates on NetVlad. and for all descriptors, it can be seen that the results of the proposed method are not affected by the descriptors because they are relatively better than the existing results, which proves that the proposed method is effective in improving performance.

Table 12 was compared in the same manner as Table 11 using the reconstructed voxel scene representation for graph clustering. The places with the best results for each case are marked in red.

In order to evaluate the accuracy of the camera pose estimation result, the re-projection error was calculated and shown in Table 13.

Methods				Test	
Global matching	Co-visibility		Local matching	Day (0.25/0.5/5.0m) (2/5/10 °)	Night (0.25/0.5/5.0m) (2/5/10 °)
NetVlad	Hloc-covisibility		R2-D2	**87.**6 / 94.5 / 97.9	75.5 / 89.8 / 94.9
			Superpoint	86.4 / 94.3 / 97.8	69.4 / 83.7 / 95.9
			Superpoint + Superglue	89.1 / **95.9** / 98.5	83.7 / **93.9** / 100.0
	Hloc-nonCovis		R2-D2	87.0 / 93.9 / 98.7	78.6 / 90.8 /99.0
			Superpoint	86.3 / 93.6 / 97.8	71.4 / 83.7 / 95.9
			Superpoint + Superglue	89.0 / 95.5 /**98.7**	85.7 / 92.9 / 100.0
	Proposed method	T=50, q=0.15	R2-D2	87.1 / **94.7** / **98.8**	**79.6** / **92.9** / **99.0**
			Superpoint	**86.7** / **94.4** / 97.7	**70.4** / **85.7** / 94.9
			Superpoint + Superglue	89.0 / 95.5 / 98.5	**86.7** / 92.9 / 99.0
		T=100, q=0.15	R2-D2	87.4 / 94.3 / **98.9**	**78.6** / **91.8** / **100.0**
			Superpoint	86.3 / 93.9 / **98.1**	**72.4** / **84.7** / **95.9**
			Superpoint + Superglue	**89.2** / 95.6 / 98.5	**86.7** / 92.9 / 100.0
		T=50, q=0.10	R2-D2	87.3 / **94.4** / **98.8**	77.6 / **92.9** / **99.0**
			Superpoint + Superglue	86.3 / 93.4 / **98.2**	68.4 / **84.7** / 94.9
			SP+SG	89.0 / 95.5 / 98.5	**86.7** / 91.8 / 99.0
		T=100, q=0.10	R2-D2	87.1 / **94.3** / **98.9**	**80.6** / **91.8** / **100.0**
			Superpoint	86.3 / 94.2 / **98.1**	**71.4** / **85.7** / **95.9**
			Superpoint + Superglue	**89.3** / 95.5 / 98.5	83.7 / 91.8 / 100.0
DIR[59]	Hloc-covisibility		R2-D2	76.2 / 83.4 / 89.4	56.1 / 68.4 / **78.6**
			Superpoint	75.0 / 82.0 / **86.9**	53.1 / 60.2 / 73.5
			Superpoint + Superglue	**81.1** / **88.3** / 93.7	72.4 / **85.7** / 89.8
	Hloc-nonCovis		R2-D2	76.9 / 83.5 / 89.0	55.1 / 66.3 / 74.5
			Superpoint	72.6 / 80.7 / 85.3	50.0 / 59.2 / 69.4
			Superpoint + Superglue	81.1 / 88.3 / 93.7	72.4 / 85.7 / 89.8
	Proposed method	T=50, q=0.15	R2-D2	**77.5** / **84.0** / 89.3	**59.2** / **68.4** / 77.6
			Superpoint	74.0 / 81.7 / 86.2	52.0 / **61.2** / 73.5
			Superpoint + Superglue	80.7 / 88.2 / 93.6	70.4 / 83.7 / 89.8
		T=100, q=0.15	R2-D2	77.2 / **83.7** / 89.2	**60.2** / **70.4** / 77.6
			Superpoint	73.5 / 81.1 / 85.7	52.0 / 60.2 / **74.5**
			Superpoint + Superglue	80.5 / 88.2 / 93.7	72.4 / 84.7 / 89.8
		T=50, q=0.10	R2-D2	77.1 / 83.9 / **89.4**	58.2 / 67.3 / 76.5
			Superpoint	**75.4** / **82.2** / 86.7	**54.1** / **62.2** / **74.5**
			Superpoint + Superglue	80.3 / 88.2 / 93.7	72.4 / 83.7 / 89.8
		T=100, q=0.10	R2-D2	77.2 / **83.6** / 89.1	58.2 / **69.4** / 75.5
			Superpoint	73.7 / 81.3 / 86.4	**54.1** / **61.2** / **75.5**
			Superpoint + Superglue	80.5 / 88.2 / 93.6	**73.5** / 84.7 / 89.8

Table 11 Ablation experiment results for various global and local descriptors.

Methods				Test	
Global matching	Co-visibility		Local matching	Day (0.25/0.5/5.0m) (2/5/10 °)	Night (0.25/0.5/5.0m) (2/5/10 °)
NetVlad	Hloc-covisibility		R2-D2	**87.6** / 94.5 / 97.9	75.5 / 89.8 / 94.9
			Superpoint	86.4 / **94.3** / 97.8	69.4 / 83.7 / 95.9
			Superpoint + Superglue	**89.1** / **95.9** / 98.5	83.7 / **93.9** / **100.0**
	Hloc-nonCovis		R2-D2	87.0 / 93.9 / 98.7	**78.6** / 90.8 / **99.0**
			Superpoint	86.3 / 93.6 / 97.8	71.4 / 83.7 / 95.9
			Superpoint + Superglue	89.0 / 95.5 /**98.7**	**85.7** / 92.9 / **100.0**
	Proposed method (mean shift clustering)	T=50, q=0.15	R2-D2	87.4 / **94.7** / **98.8**	76.5 / 89.8 / 99.0
			Superpoint	86.0 / 93.9 / **98.1**	**72.4** / **85.7** / 94.9
			Superpoint + Superglue	**89.1** / 95.6 / 98.5	**85.7** / 92.9 / 99.0
		T=100, q=0.15	R2-D2	87.4 / 94.4 / 98.5	**78.6** / 89.8 / **100.0**
			Superpoint	86.0 / 93.7 / **97.9**	**74.5** / **84.7** / 95.9
			Superpoint + Superglue	**89.1** / **96.0** / **98.7**	84.7 / **93.9**/ **100.0**
		T=50, q=0.10	R2-D2	87.3 / **94.9** / **98.7**	76.5 / **92.9** / **99.0**
			Superpoint	**86.9** / 93.9 / **98.1**	69.4 / 83.7 / 94.9
			Superpoint + Superglue	**89.1** / 95.6 / 98.5	**85.7** / 92.9 / 99.0
		T=100, q=0.10	R2-D2	87.3 / 94.4 / **98.8**	77.6 / 90.8 / **100.0**
			Superpoint	85.9 / 93.7 / **98.1**	**73.5** / **85.7** / 95.9
			Superpoint + Superglue	**89.1** / **96.0** / **98.7**	84.7 / **93.9** / **100.0**
DIR	Hloc-covisibility		R2-D2	76.2 / 83.4 / 89.4	56.1 /68.4 / **78.6**
			Superpoint	**75.0** / **82.0** / **86.9**	**53.1** / 60.2 / 73.5
			Superpoint + Superglue	**81.1** / 88.3 / 93.7	72.4 / **85.7** / 89.8
	Hloc-nonCovis		R2-D2	76.9 / 83.5 / 89.0	55.1 / 66.3 / 74.5
			Superpoint	72.6 / 80.7 / 85.3	50.0 / 59.2 / 69.4
			Superpoint + Superglue	81.1 / 88.3 / 93.7	72.4 / 85.7 / 89.8
	Proposed method (mean shift clustering)	T=50, q=0.15	R2-D2	**77.2** / **83.6** / 89.1	**60.2** / **71.4** / 76.5
			Superpoint	73.4 / 81.2 / 85.8	**53.1** / **61.2** / **74.5**
			Superpoint + Superglue	80.3 / 88.2 / 93.3	71.4 / 84.7 / 89.8
		T=100, q=0.15	R2-D2	**77.4** / **84.0** / 89.3	**59.2** / **69.4** / 77.6
			Superpoint	74.2 / 81.8 / 86.4	50.0 / **61.2** / 71.4
			Superpoint + Superglue	80.9 / **88.5** / 93.6	72.4 / 84.7 / 89.8
		T=50, q=0.10	R2-D2	**77.4** / **84.2** / 89.0	**60.2** / 68.4 / 76.5
			Superpoint	74.0 / 81.9 / 86.5	**53.1** / 60.2 / 73.5
			Superpoint + Superglue	80.3 / 88.2 / 93.3	71.4 / 84.7 / 89.8
		T=100, q=0.10	R2-D2	**77.1** / **83.7** / 89.1	**59.2** / **70.4** / 77.6
			Superpoint	73.8 / 81.6 / 86.0	**53.1** / **62.2** / **74.5**
			Superpoint + Superglue	80.9 / **88.5** / 93.6	72.4 / 84.7 / 89.8

Methods				Test	
Global matching	Co-visibility		Local matching	Day (0.25/0.5/5.0m) (2/5/10 °)	Night (0.25/0.5/5.0m) (2/5/10 °)
NetVlad	Hloc-covisibility		R2-D2	**87.6** / **94.5** / 97.9	75.5 / 89.8 / 94.9
			Superpoint	86.4 / 94.3 / 97.8	69.4 / 83.7 / 95.9
			Superpoint + Superglue	**89.1** / **95.9** / 98.5	83.7 / **93.9** / 100.0
	Hloc-nonCovis		R2-D2	87.0 / 93.9 / 98.7	**78.6** / 90.8 / **99.0**
			Superpoint	86.3 / 93.6 / 97.8	71.4 / 83.7 / 95.9
			Superpoint + Superglue	89.0 / 95.5 /**98.7**	**85.7** / 92.9 / **100.0**
	Proposed method (graph clustering)	T=50, intersection =0.5	R2-D2	86.9 / 94.3 / **98.7**	**78.6** / **91.8** / **99.0**
			Superpoint	**86.9** / 94.3 / **97.9**	**71.4/ 84.7** / 95.9
			Superpoint + Superglue	**89.1** / 95.8 / 98.5	84.7 / 92.9 / 100.0
		T=100, intersection =0.5	R2-D2	86.8 / 94.3 / **98.7**	77.6 / **91.8** / **99.0**
			Superpoint	86.8 / **94.4** / **97.9**	**71.4** / **84.7** / 95.9
			Superpoint + Superglue	**89.1** / 95.8 / 98.5	**85.7** / 92.9 / 100.0
DIR	Hloc-covisibility		R2-D2	76.2 / 83.4 / **89.4**	56.1 / **68.4** / **78.6**
			Superpoint	**75.0** / **82.0** / **86.9**	**53.1** / 60.2 / 73.5
			Superpoint + Superglue	81.1 / 88.3 / **93.7**	72.4 / **85.7** / 89.8
	Hloc-nonCovis		R2-D2	**76.9** / 83.5 / 89.0	55.1 / 66.3 / 74.5
			Superpoint	72.6 / 80.7 / 85.3	50.0 / 59.2 / 69.4
			Superpoint + Superglue	81.1 / 88.3 / **93.7**	72.4 / **85.7** / 89.8
	Proposed method (graph clustering)	T=50, intersection =0.5	R2-D2	**77.1** / **84.0** / 89.2	**58.2** / **68.4** / 77.6
			Superpoint	73.9 / 81.6 / 86.5	51.0 / 60.2 / **74.5**
			Superpoint + Superglue	81.1 / **88.6** / 93.6	72.4 / 83.7 / 89.8
		T=100, intersection =0.5	R2-D2	**76.9** / **83.7** / 89.1	**58.2** / 67.3 / 77.6
			Superpoint	74.0 / 81.9 / 86.7	51.0 / 60.2 / **74.5**
			Superpoint + Superglue	80.9 / **88.5** / 93.6	72.4 / 83.7 / 89.8

Table 12. Results using the reconstructed voxel scene representation for graph clustering (Top : meanshift clustering , Bottom : graph clustering)

Methods				Test
Global matching	Co-visibility		Local matching	Reprojection error
NetVlad	Hloc-covisibility		R2-D2	0.138
			Superpoint	0.252
			Superpoint + Superglue	0.116
	Hloc-nonCovis		R2-D2	**0.133**
			Superpoint	0.262
			Superpoint + Superglue	0.115
	Proposed method (mean shift clustering)	T=50, q=0.15	R2-D2	0.175
			Superpoint	**0.248**
			Superpoint + Superglue	**0.115**
DIR	Hloc-covisibility		R2-D2	0.304
			Superpoint	**0.502**
			Superpoint + Superglue	0.249
	Hloc-nonCovis		R2-D2	0.302
			Superpoint	0.507
			Superpoint + Superglue	0.220
	Proposed method (mean shift clustering)	T=100, q=0.10	R2-D2	**0.302**
			Superpoint	0.513
			Superpoint + Superglue	**0.237**

Methods				Test
Global matching	Co-visibility		Local matching	Reprojection error
NetVlad	Hloc-covisibility		R2-D2	0.138
			Superpoint	0.252
			Superpoint + Superglue	0.116
	Hloc-nonCovis		R2-D2	**0.133**
			Superpoint	0.262
			Superpoint + Superglue	0.115
	Proposed method (graph clustering)	T=50, intersection=0.5	R2-D2	0.173
			Superpoint	**0.254**
			Superpoint + Superglue	**0.115**
DIR	Hloc-covisibility		R2-D2	0.304
			Superpoint	**0.502**
			Superpoint + Superglue	0.249
	Hloc-nonCovis		R2-D2	**0.302**
			Superpoint	0.507
			Superpoint + Superglue	0.220
	Proposed method (graph clustering)	T=50, intersection=0.5	R2-D2	0.304
			Superpoint	0.508
			Superpoint + Superglue	**0.246**

Table 13. Error in re-projection of camera pose estimation results (Top : meanshift clustering , Bottom : graph clustering)

5. Conclusion

This chapter discusses the conclusions of the study and future research directions.

5. Conclusion

In this paper, clustering considers the importance of reference images as spatial accessibility.

We discussed improving the performance of this camera pose performance. Chapter 2 introduces the prior knowledge necessary to do it and existing research methods, as it estimates the camera pose, the ultimate purpose of the proposed method.

The importance of the reference image in the method proposed in this paper is used for cohesiveness clustering also mentioned in HF-net[15]. The histogram serves as a simple measure to extract this importance. For this experiment, we described a voxel-based representation of 3D space that can help flexible clustering. By using the indexing information of voxels, it was possible to know which point the reference images were looking at and observed, and by listing them in the order of high covisibility, we checked the location of prominent voxels and examined the clustering of local descriptors for matching later. In the experiment, the higher the importance, the more images from a similar view point to the voxels shown in the query video, but sometimes more views of the location of the voxels that are far from that point but shared.

The conclusion from this analysis confirmed that removing the image of the reference point of view that becomes the outlier when creating the local descriptor increases the matching performance, which soon increases the camera estimation performance as well. In addition, since the importance of information on the voxel histogram alone could not take into account the proximity of space, mean shift clustering and graph

clustering was performed in consideration of the location of the voxel. In the experiment, it was confirmed that the clustered images were better divided into similar categories than the results of conventional cohesiveness clustering, and the performance of the final camera pose estimation was improved.

However, the results of this experiment showed improved performance on a large scale Aachen dataset, but on a medium scale Cambridge dataset, there was no difference in performance because image views of voxels that seemed to be shared together almost came into the top 50 candidate. In the future, there are still studies that can remove reference data that may become outliers even on a small scale in advance. And the limitation of this study is that it only allows the wrong matching to be avoided, but it is impossible to change the matching back to the right matching.

By further analyzing and researching the nature and importance of the image, it remains a future task to improve performance by not only removing the outlier but also converting it into an inlier.

References

[1] Klein, G., & Murray, D. Parallel tracking and mapping for small ar workspaces. In Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13-16 November 2007; pp. 1-10.

[2] Wu, Ch. Towards linear-time incremental structure from motion. In Proceedings of the International Conference on 3D Vision, Seattle, Washington, 29 June - 1 July 2013; pp. 127 - 134.

[3] Kenall, A., & Grimes, M., & Cipolla, R. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7 - 13 December 2015; pp. 2938 - 2946.

[4] Radwan, N., & Valada, A., & Burgard, W. VLocNet++: deep multitask learning for semantic visual localization and odometry. IEEE Robotics and Automation Letters, 2018, 3(4), 4407-4414.

[5] Sattler, T., & Zhou, Q., & Pollefeys, M., & Leal-Taixe´, L. Understanding the limitations of CNN-based absolute camera pose regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA., 16-20 June 2019, pp.3302-3312.

[6] Sattler, T., & Leibe, B., & Kobbelt, L. Efficient & effective prioritized matching for large-scale image-based localization. IEEE Trans. Pattern Anal. Mach. Intell., 2016, 39(9), 1744-1756.

[7] Davison, A. J., & Reid, I. D., & Molton, N. D., & Stasse, O. MonoSLAM: real-time single camera slam. IEEE Trans. Pattern Anal. Mach. Intell., 2007, 29(6), 1052-1067.

[8] Engel, J., & Cremers, D. Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell., 2018, 40(9), 611-625.

[9] Lowe, D. G. Distinctive image features from scale invariant keypoints. International journal of computer vision, 2004, 60(2), 91-110

[10] Calonder, M., & Lepetit, V., & Strecha, C., & Fua, P. BRIEF: binary robust independent elementary features. In Proceedings of the 11th European Conference on Computer Vision, pp 778-792, Heraklion, Crete, Greece, September 5-11, 2010.

[11] Rublee, E., & Rabaut, V., & Konolige, K., & Bradski, G. ORB: an efficient alternative to SIFT or SURF. In Proceedings of the IEEE Conference on Computer Vision, pp. 2564-2571, Barcelona, Spain, November 6-13, 2011.

[12] Xiang, Y., & Schmidt, T., & Narayanan, V., & Fox, D. PoseCNN: a convolutional neural network for 6d object pose estimation in cluttered scenes. In Proceedings of the Robotics: Science and Systems XIV, Pittsburgh, Pennsylvania, 26-30 June 2018; pp. 1 - 10.

[13] Crivellaro, A, & Rad, M., & Verdie, Y., & Yi, K. M., & Fua, P., & Lepetit, V. Robust 3d object tracking from monocular images using stable parts. IEEE Trans. Pattern Anal. Mach. Intell., 2018, 40(6), 1465-1479.

[14] Sarlin, P., & Debraine, F., & Dymczyk, M., & Siegwart, R., & Cadena, C. Leveraging deep visual descriptors for hierarchical efficient localization. In Proceedings of the 2nd Conference on Robot Learning, Zürich, Switzerland, 29-31 October 2018; pp. 456-465.

[15] Sarlin, P., & Cadena, C., & Siegwart, R., & Dymczyk, M. From coarse to fine: robust hierarchical localization at large scale. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, California, 16-20 June 2019; pp. 12716-12725.

[16] Garon, M., & Lalonde, J. Deep 6-dof tracking. IEEE Trans. on Visualization and Computer Graphics, 2017, 23(11), 2410-2418.

[17] Brahmbhatt, S., & Gu, J., & Kim, K., & Hays, J., & Kautz, J. Geometry-aware learning of maps for camera localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT., 18-23 June 2018; pp. 2616-2625.

[18] Shavit, Y., & Ferens, R. Introduction to camera pose estimation with deep learning. arXiv:1907.05272, 2019.

[19] Su, J., & Cheng, S., & Chang, C., & Chen, J. Model-based 3D pose estimation of a single rgb image using a deep viewpoint classification neural network. Appl. Sci., 2019, 9(12), 2478.

[20] DeTone, D., & Malisiewicz, T., & Rabinovich, A. SuperPoint: Self-supervised interest point detection and description. In Proceedings of the IEEE/CVF Conference on Computer Vision

and Pattern Recognition Workshops, Salt Lake City, UT., 18-23 June 2018; pp. 224 - 236.

[21] Dusmanu, M., & Rocco, I., & Pajdla, T., & Pollefeys, M., & Sivic, J.; Torii, A.; Sattler, T. 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, Long Beach, CA., 16-20 June 2019, pp.8092-8101.

[22] Choy, C. B., & Gwak, J. Y., & Savarese, S., & Chandraker, M. Universal correspondence network. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 4-9 December 2016, pp.2414 - 2422.

[23] Mahajan, D., & Girshick, R., & Ramanathan, V., & He, K., & Paluri, M., & Li, Y., & Bharambe, A. and van der Maaten, L., 2018. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 181-196).

[24] Tan, M. & Le, Q.V., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946.

[25] Szegedy, C., & Liu, W., & Jia, Y., & Sermanet, P., & Reed, S., & Anguelov, D., & Erhan, D., & Vanhoucke, V. & Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[26] Westlake, N., & Cai, H. & Hall, P., 2016, October. Detecting people in artwork with CNNs. In European Conference on Computer Vision (pp. 825-841). Springer, Cham.

[27] Zhu, Y., & Sapra, K., & Reda, F.A., & Shih, K.J., & Newsam, S., Tao, A. & Catanzaro, B., 2019. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8856-8865).

[28] Badrinarayanan, V., & Kendall, A. & Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), pp.2481-2495

[29] Torsten S., & Will M., & Carl T., Akihiko T., & Lars H., & Erik S., & Daniel Safari., & Masatoshi O., & Marc P., & Josef S., Fredrik K., & Tomas P. Benchmarking 6DOF outdoor visual localization in changing conditions. In CVPR, 2018.

[30] Hajime T., & Masatoshi O., & Torsten S., & Mircea C., & Marc P., & Josef S., & Tomas P., & Akihiko T. InLoc: Indoor visual localization with dense matching and view synthesis. In CVPR, 2018.

[31] Mahajan, D., & Girshick, R., & Ramanathan, V., & He, K., & Paluri, M., & Li, Y., & Bharambe, A. & van der Maaten, L., 2018. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 181-196).

[32] Tan, M. & Le, Q.V., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946.

[33] Szegedy, C.,& Liu, W., & Jia, Y., & Sermanet, P., & Reed, S., &

Anguelov, D., & Erhan, D., & Vanhoucke, V. & Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[34] Westlake, N., & Cai, H. & Hall, P., 2016, October. Detecting people in artwork with CNNs. In European Conference on Computer Vision (pp. 825-841). Springer, Cham.

[35] Zhu, Y., & Sapra, K., & Reda, F.A., & Shih, K.J., & Newsam, S., & Tao, A. & Catanzaro, B., 2019. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8856-8865).

[36] Badrinarayanan, V., & Kendall, A. & Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), pp.2481-2495

[37] Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA., 7-9 May 2015.

[38] Arandjelovic, R., et al. "NetVLAD: CNN architecture for weakly supervised place recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[39] M. A. Fischler, & R. C. Bolles : Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Communications of the ACM, vol. 24,

no. 6, pp. 381-395, (1981).

[40] Ranganathan, A. "The levenberg-marquardt algorithm." Tutoral on LM algorithm 11.1 (2004): 101-110.

[41] https://europe.naverlabs.com/blog/methods-for-visual-localization/

[42] Jégou, H., & Douze, M., & Schmid C., & Pérez, P. "Aggregating local descriptors into a compact image representation," in CVPR, 2010

[43] Torsten S., & Tobias W., & Bastian L., & Leif K.. Image retrieval for image-based localization revisited. In CVPR, 2008

[44] Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4104-4113).

[45] Schönberger, J. L., & Zheng, E., Frahm, J. M., & Pollefeys, M. (2016, October). Pixelwise view selection for unstructured multi-view stereo. In European Conference on Computer Vision (pp. 501-518). Springer, Cham.

[46] Snavely, N., & Seitz, S.M. & Szeliski, R., 2006, July. Photo tourism: exploring photo collections in 3D. In ACM transactions on graphics (TOG) (Vol. 25, No. 3, pp. 835-846). ACM.

[47] Wu, C., 2011. VisualSFM: A visual structure from motionsystem. http://www. cs. washington. edu/homes/ccwu/vsfm

[48] Huang, M., & Men, L., & Lai, C. (2013). Accelerating mean shift segmentation algorithm on hybrid CPU/GPU platforms. In Modern Accelerator Technologies for Geographic Information Science (pp. 157-166). Springer, Boston, MA.

[49] Lin, T. Y., & Maire, M., Belongie, S., & Hays, J., & Perona, P., & Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[50] Sarlin, P. E., & DeTone, D., & Malisiewicz, T., & Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4938-4947).

[51] Kneip, L., & Scaramuzza, D., & Siegwart, R. (2011, June). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In CVPR 2011 (pp. 2969-2976). IEEE.

[52] Kukelova, Z., & Bujnak, M., & Pajdla, T. (2013). Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2816-2823).

[53] Chum, O., & Matas, J. (2008). Optimal randomized RANSAC. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(8), 1472-1482.

[54] Lebeda, K., & Matas, J., & Chum, O. (2012, September). Fixing the locally optimized ransac - full experimental evaluation. In British machine vision conference (Vol. 2). Citeseer.

[55] Zhou, Q., & Sattler, T., & Leal-Taixe, L. (2021). Patch2pix: Epipolar-guided pixel-level correspondences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition

(pp. 4669-4678).

[56] Humenberger, M., & Cabon, Y., & Guerin, N., & Morat, J., & Revaud, J., & Rerole, P., ... & Csurka, G. (2020). Robust image retrieval-based visual localization using kapture. arXiv preprint arXiv:2007.13867.

[57] Sattler, T., & Leibe, B., & Kobbelt, L. (2016). Efficient & effective prioritized matching for large-scale image-based localization. IEEE transactions on pattern analysis and machine intelligence, 39(9), 1744-1756.

[58] https://www.visuallocalization.net/benchmark/

[59] Revaud, J., & Almazán, J., & Rezende, R. S., & Souza, C. R. D. (2019). Learning with average precision: Training image retrieval with a listwise loss. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5107-5116).

국문초록

카메라 위치 파악을 위한 복셀 표현의 covisibility 기반 참조 이미지 클러스터링

이 상 윤

융합공학과 디지털 이미징 전공

중앙대학교 대학원

컴퓨터 비전은 모바일 로봇, 자율주행 자동차, 드론, 증강현실 애플리케이션 등 다양한 분야에서 널리 활용되고 있다. 입력 이미지에서 카메라의 자세를 추정하고 위치를 식별하기 위한 시각적 위치 파악(localization)은 중요한 컴퓨터 비전 주제 중 하나다. 카메라 위치를 식별하기 위해 움직임의 구조(Structure from Motion) 방법이 개발되었다. 최근 컨볼루션 뉴럴 네트워크(Convolutional Neural Networks)가 성능 측면에서 시각적 위치 파악 방법으로 주목 받고 있다.

본 논문에서는 대규모 장면의 복셀 표현을 기반으로 하는 계층적 위치 파악 방법을 소개한다. 논문의 대략적인 접근 방식은 장면에 대한 전역 설명자(descriptor)를 사용하여 질의 영상과 유사한 후보(참조 뷰(reference view))를 추출하여 검색 범위를 줄이고 로컬 일치로 카메라 자세를 추정한다. 그러나, 참조 뷰는 이미지 특징을 기반으로 하는 전역 설명자를 사용하여 추출되기 때문에, 질의 영상과 유사하지만 정확한 위치에서 상당히 멀리 떨어진 참조 뷰가 선택 될 수 있다.

본 논문에서는 희소 또는 조밀한 3차원 포인트 클라우드 및 참조 뷰와 같은 3차원 장면 표현이 가능하다고 가정한다. 참조 자세는 일반적으로 희박하거나 조밀한 3차원 재구성에 의해 얻어지기 때문에 이 요구 사항은 실제로

쉽게 충족된다. 또한 3차원 포인트의 복셀 위치가 저장되어 어떤 복셀에 있는지를 나타낸다.

각 복셀에서 전역 설명을 기반으로 추출된 참조 뷰의 3차원 포인트 수를 검사한다. 이 히스토그램은 3차원 포인트의 분포를 보여준다. 즉, 어떤 장면 영역이 더 자주 추출되는지를 의미한다. 상대적으로 중요한 복셀은 복셀 기반의 히스토그램을 사용하여 빈도가 높은 순서대로 선택한다. 중요한 복셀 위치에 평균 이동 방법(meanshift)과 그래프 클러스터링을 적용하여 복셀 클러스터를 구성할 수 있다. 즉, 참조 영상은 동시에 보여 지는 가시성(covisibility)을 기반으로 클러스터링 된다. 각 클러스터에서 질의 영상과 참조 뷰 간에 로컬 매칭이 수행된다. 여기서 참조 뷰의 SuperPoint 설명 정보가 사용되며 2차원-3차원 대응이 설정된다. PnP(Perspective-n-Point) 방법을 사용하여 클러스터로 카메라 자세를 추정한 후 가장 많은 inlier 2차원-3차원 대응 집합을 가진 결과가 최종 카메라 자세로 결정된다.

제안된 방법의 성능을 검증하기 위해 대규모 Aachen 데이터 셋으로 실험을 진행하였다. 그러나 중간 규모의 Cambridge 데이터 셋에서 대부분의 뷰는 공통된 관점을 공유한다. 즉, 대부분의 영역이 동시에 표시된다. 따라서 Cambridge 데이터 셋에 대한 실험 결과에서는 성능에 큰 변화가 없었다.

Abstract

Clustering reference images based on the covisibility in voxel representation for camera localization

Sangyun Lee
Major in Digital Imaging
Department of Integrative Engineering
The Graduate School, Chung-Ang Universyty

Computer vision has been widely used in various fields such as mobile robots, self-driving cars and drones, and augmented reality applications. From input images, visual localization to estimate the pose of the camera and identify the location is one of the important computer vision topics. To identify the camera location, structure from motion methods have been developed. Recently, convolutional neural networks (CNN) have been attracting attention as a visual localization method in viewpoint of performance.

In this dissertation, we introduce a hierarchical localization method based on the voxel representation of the large-scale scene. Our coarse-to-fine approach reduces the search range by extracting candidates (reference views) similar to a query image with the global descriptors for the scene and estimates the camera pose with local matching. However, since the reference views are extracted using the global description based on the image features, the reference views similar to the query image but

considerably far away the accurate location may be selected.

In this dissertation, it is assumed that a 3D scene representation such as a sparse or dense 3D point cloud and the reference views is available. This requirement is easily met in practice since the reference poses are usually obtained by sparse or dense 3D reconstruction. In addition, the voxel position of 3D points is stored, representing which voxel they are in.

In each voxel, the number of 3D points in the reference views extracted based on the global description are examined. This histogram shows the distribution of 3D points, meaning which scene areas are extracted more frequently. The relatively important voxels are chosen with the voxel-based histogram in the order of high frequency. By applying mean shift method and graph clustering method to the important voxel locations, some clusters of voxels can be constructed. That means reference images are clustered based on the covisibility. In each cluster, local matching is performed between the query image and the reference views. Here the superpoint's description information of the reference views is used and 2D-3D correspondence is established. After PnP method is employed to estimate the camera pose with the clusters, the result with the most inlier 2D-3D correspondence set is determined as the final camera pose.

The experiment was conducted with Aachen dataset on a large scale to verify the performance of the proposed method. However, in the Cambridge dataset on a middle scale, most views share common viewpoints. That means most areas are visible at the same time. Therefore, in the experimental results on Cambridge dataset, there was no significant change in performance.

감사의 글

연구실에 들어온 것이 엊그제 같은데 벌써 논문 심사를 마치고 졸업을 앞두게 되었습니다. 시간은 빠르게 흘러갔지만 결코 짧지 않은 시간이었고, 그 시간동안 여러 가지 일들을 하면서 마무리 짓지 못했던 것, 해보고 싶었던 것을 못했던 일들로 다소 아쉬움이 남습니다. 그럼에도 불구하고 많은 도움을 주셨던 분들이 계셨기에, 지금의 제가 있는 것 같습니다.

2014년부터 연구실과의 인연이 시작 되었는데, 대략 8년이라는 시간동안 부족한 제자를 변함없이 열과 성을 다해서 가르쳐주시고 키워주신 지도 교수인 홍현기 교수님께 정말 한없는 감사를 드립니다. 연구뿐만 아니라 인생 전반에 대한 조언도 항상 해주시고, 친자식처럼 대해 주셔서 이런 감사의 인사로는 너무나도 부족한 것 같습니다. 그리고 제 2의 지도 교수님의 역할을 해주시고, 아낌없는 조언과 열정을 다해서 연구에 많은 도움을 주신 임창경 교수님께도 깊은 감사를 드립니다.

논문 심사를 맡아 많은 조언과 도움을 주신 김태용 교수님, 홍병우 교수님, 권준석 교수님, 박경주 교수님께도 깊은 감사를 드립니다.

석사시절부터 지금까지 인연을 계속 이어오고 있는 연구실 동료들, 민이, 은아, 익수, 종철이형, 대윤이, 서영이, 재혁이 항상 응원 해줘서 고맙습니다. 특히 연구실과 자취방을 넘나들며 24시간을 같이, 희노애락을 모두 겪으며 서로 의지가 되어준 동기 김익수에게 한 번 더 감사의 말을 전합니다. 이후, 박사과정동안 부족한 제 밑에서 고생하고 있는 후배들에게, 많은 것을 전해주지 못해 미안하고, 잘 따라 와줘서 감사합니다. 먼저 졸업하여 각자의 위치에서 고군분투하고 있는 선배님들, 특히 인생의 멘토 역할을 해주시는 태협이 형에게 감사를 드립니다. 힘들 때 마다 푸념을 들어주고 항상 응원해준 친구들에게도 감사를 드립니다.

마지막으로 지금까지 아들 뒷바라지 하시느라 한평생을 바치시고 고생하신

부모님, 끝까지 믿고 응원해 주셔서 정말 감사합니다. 앞으로 더욱 노력하여 부끄럽지 않은 훌륭한 아들 되겠습니다. 이 모든 과정 동안 수많은 고민과 한계 그리고 좌절을 겪었고, 때로는 환희도 있었습니다. 이 시간을 견디고 즐길 수 있었던 건 주변에서 항상 응원해주신 여러분들 덕이었습니다. 앞으로도 힘든 일들이 있겠지만, 포기하지 않고 긍정적으로 살아가는 훌륭한 사람이 되도록 노력하겠습니다. 모두 감사합니다.

2022년 2월