electronics

MDPI

Article

Hierarchical Clustering-Based Image Retrieval for Indoor Visual Localization

Guanyuan Feng [1,*], Zhengang Jiang [1], Xuezhi Tan [2] and Feihao Cheng [1]

[1] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
[2] Communication Research Center, Harbin Institute of Technology, Harbin 150080, China
* Correspondence: fengguanyuan@126.com

Abstract: Visual localization is employed for indoor navigation and embedded in various applications, such as augmented reality and mixed reality. Image retrieval and geometrical measurement are the primary steps in visual localization, and the key to improving localization efficiency is to reduce the time consumption of the image retrieval. Therefore, a hierarchical clustering-based image-retrieval method is proposed to hierarchically organize an off-line image database, resulting in control of the time consumption of image retrieval within a reasonable range. The image database is hierarchically organized by two stages: scene-level clustering and sub-scene-level clustering. In scene-level clustering, an improved cumulative sum algorithm is proposed to detect change points and then group images by global features. On the basis of scene-level clustering, a feature tracking-based method is introduced to further group images into sub-scene-level clusters. An image retrieval algorithm with a backtracking mechanism is designed and applied for visual localization. In addition, a weighted KNN-based visual localization method is presented, and the estimated query position is solved by the Armijo–Goldstein algorithm. Experimental results indicate that the running time of image retrieval does not linearly increase with the size of image databases, which is beneficial to improving localization efficiency.

Keywords: visual localization; hierarchical clustering; image retrieval; change-point detection

Citation: Feng, G.; Jiang, Z.; Tan, X.; Cheng, F. Hierarchical Clustering-Based Image Retrieval for Indoor Visual Localization. *Electronics* **2022**, *11*, 3609. https://doi.org/10.3390/electronics11213609

Academic Editors: Zhan Li, Zhang Chen and Yiyong Sun

Received: 6 October 2022
Accepted: 31 October 2022
Published: 4 November 2022

1. Introduction

With the development of communication technology, smart mobile terminals, such as smartphones and tablet personal computers, have become indispensable in modern society. Various applications on smart mobile terminals bring convenience to many aspects of people's lives, and one example is navigation applications. The Global Navigation Satellite System (GNSS) allows individuals to acquire position information at any moment in outdoor environments [1–4]. Navigation and positioning services play crucial roles in public traffic, maritime transportation, and aviation flight. As the interior environments of buildings become increasingly complex, demands for indoor position services continue to rise. However, due to the shielding effect of structures, signals of the Global Navigation Satellite System are incapable of penetrating buildings, leading to users not being able to obtain reliable position services by the GNSS. Therefore, a stable and efficient indoor localization method independent of satellite signals has become a research hotspot in late years. Numerous daily activities can benefit from indoor localization technologies, such as shopping in large malls, finding books in libraries, and planning routes in railway stations and airports.

Many signal-based localization technologies have been investigated for querying indoor position information, such as WiFi-based [5,6], Bluetooth-based [7,8], and UWB-based [9–11] methods. All these methods, however, require investments in localization infrastructures. For example, WiFi-based approaches demand that a mobile terminal should receive signals transmitted by more than one access point [12,13]. Generally, more

densely distributed access points contribute to improving the accuracy of localization systems. Similar to WiFi-based methods, high-density base stations must be deployed in indoor environments for Bluetooth-based and UWB-based localization systems. For the implementation of indoor localization, cost investments restrict the development of signal-based systems. By contrast, visual-based localization achieves high accuracy with hardly any infrastructure.

Another category of indoor localization approaches are those which estimate users' positions iteratively by Inertial Measurement Unit (IMU) [14,15]. However, IMU-based systems are prone to cumulative errors accompanied by position iterative estimation, especially for a long trajectory. In addition, since IMU-based methods cannot determine the absolute positions of devices, these methods are generally combined with other localization technologies [16–18]. Different from IMU-based methods, visual localization could achieve either absolute position estimation or relative position estimation [19].

Visual localization aims at estimating the position of a camera (i.e., the query camera) mounted on a smart mobile terminal by image retrieval and geometrical measurement. Specifically, visual localization is usually implemented in known environments in which visual features are collected and stored as database images in the off-line stage [20]. In the on-line stage, a user captures query images by the query camera, and then the images are uploaded to the server. In the process of position estimation, the matched database images are retrieved on the server, and the positions of query images are calculated by localization algorithms. High accuracy and efficiency of image retrieval guarantee the performance of the entire localization system. The typical technology is Content-Based Image Retrieval (CBIR), the core of which is finding the most similar database images as the query image based on visual features. However, the retrieval task is challenging, because the number of database images is large, and there is no feasible strategy to organize the images in the database.

Therefore, in this paper, a hierarchical clustering-based image retrieval algorithm is proposed to organize database images, and an image retrieval strategy is investigated to improve retrieval efficiency. Moreover, a visual localization method is presented based on the Armijo–Goldstein principle. The main contributions of this paper are summarized as follows:

(1) A global feature-based image clustering algorithm is proposed, in which the change-point detection method is adopted to identify which database images are captured in the same indoor scene. By this means, images captured in the same scene are clustered in a group, which achieves scene-level image clustering.
(2) A local feature-based image clustering algorithm is presented, in which feature tracking is employed to further group the images that are in one scene-level cluster, by which means database images are grouped into sub-scene-level clusters.
(3) A hierarchical clustering-based image retrieval algorithm is introduced, and visual localization is achieved based on the Armijo–Goldstein principle.

The remainder of this paper is organized as follows: Related work is reviewed in Section 1. Sections 2 and 4 investigate the image clustering algorithms based on change-point detection and feature tracking, respectively. In addition, image retrieval and visual localization are explored in Section 4. In Section 5, the performances of the image retrieval and visual localization are evaluated. A discussion of the experimental results is presented in Section 6, and conclusions are drawn in the last section.

2. Related Works

Computer vision is a field of artificial intelligence that plays an essential role in widespread applications such as object tracking, object detection, image classification, and image retrieval [21–24]. The use of computer vision for pedestrian visual localization began in 2006, and then a typical framework of visual localization was determined [25,26]. Specifically, geo-tagged images are acquired as database images in the off-line stage, and

then the position of the query image is estimated based on database images by image retrieval and geometrical measurement.

Recently, many studies have focused on visual localization, either in indoor or outdoor environments, with almost the same technical route [27–29]. The users' positions are always estimated by query images in the condition of known or unknown camera-intrinsic parameters. One advantage of indoor visual localization is that the interior scenes of buildings can be reconstructed by mapping equipment [30]. Thus, compared with other positioning approaches, the vision-based methods provide users with more indoor-detail information and a better service experience [31]. Indoor visual localization contains three key technologies, which include: (1) 3D indoor mapping on the off-line stage (including database image acquisition), (2) image retrieval in the database, and (3) position estimation of the query camera. Image retrieval for visual localization can usually be divided into two phases: coarse retrieval and fine retrieval [32]. Specifically, global features on images can be utilized to achieve coarse image retrieval, and local features are appropriate for fine retrieval, resulting in obtaining the matched database image with the query image.

Colors, textures, and shapes are essential information to describe the global features of an image. Various global features, such as color moments, HSV histograms, wavelet transforms, and Gabor wavelet transforms, are extracted from images and used in the CBIR system [33,34]. Generally, more than one feature is selected to form a high-dimensional feature vector in order to overcome the limitations of a single feature. However, high-dimensional feature vectors would bring a heavy burden in measuring the similarity of images. Based on Gabor features, a global feature named Gist was proposed by Oliva et al., designed for scene recognition [35]. Gist features have already been widely used in indoor image retrieval and have achieved some remarkable results, which indicate that Gist features have the potential to address the image retrieval problem in visual localization [36,37].

Compared with global visual feature-based image retrieval technologies, local features are more suitable for fine image retrieval (i.e., finding the most similar database image with the query image). In recent years, research on local feature extraction has attracted much attention. Speeded Up Robust Features (SURF) [38] and Scale-Invariant Feature Transform (SIFT) [39] are the most widely used local features in the fields of object recognition, image stitch, visual tracking, and so on. For a visual localization system, local features are both applied to image retrieval and position estimation. Visual features employed in localization perform well in image similarity measuring and users' position estimating. An efficient alternative feature (i.e., Oriented FAST and Rotated BRIEF, ORB) to SIFT or SURF was proposed by Rublee et al., which has been widely used in visual SLAM and has achieved promising results [40,41]. Based on existing literature and research, ORB features, utilizing fast key points and described by BRIEF descriptors, are also good at content-based image retrieval [42,43].

Most researchers of visual localization focus on position estimation algorithms, such as the authors of [29,44,45], but few of them pay attention to image retrieval in the localization system, much less to off-line image database organization. A typical hierarchical indexing scheme is proposed in [46], but only coarse retrieval is presented, and the best-matched database image cannot be found by this scheme. A well-organized image database contributes to improving the accuracy and efficiency of a localization system. In other words, a scalable image retrieval method is desired to fit different sizes of the database of the indoor localization system. Specifically, the search time of a scalable image system should not increase linearly with the number of database images, so that the response time of retrieval can be limited within a reasonable range. Therefore, aiming at the demand for visual localization, a database image hierarchical clustering method based on change-point detection and feature tracking is investigated in this paper. With the results of image retrieval, visual localization is executed to estimate users' positions.

Visual localization in indoor environments has received widespread attention in recent years due to its extensive applications, such as the mobile museum tourist guide [47] and in-building emergency response [48]. Visual localization is also called image-based

localization, in which images are employed as practical signals for localization [49]. Strictly speaking, visual localization is one of the fingerprinting-based localization methods, since images captured at known locations (i.e., database images) serve as fingerprints for position estimation. By image retrieval, the most similar database images as the query image are selected as fingerprints to calculate the query positions [50]. The majority of recent visual localization works focus on position estimation in the condition of known camera-intrinsic parameters by projective geometry [51–53]. However, the internal parameters of different cameras are not easy to obtain in practical localization scenarios. Therefore, research on image retrieval-based visual localization is crucial and necessary.

Only a limited number of works concentrate on image retrieval-based visual localization without camera calibration parameters. A representative work is the TUM indoor navigation system, in which the nearest neighbor (NN) method is employed for localization (i.e., the position attached to the most similar database image is identified as the query position) [49]. In many fingerprinting-based localization applications, finding the nearest neighbor to the query is regarded as an effective way to acquire the query position [54,55]. However, nearest neighbor-based visual localization may have a significant positioning error in some situations. For example, the query position is far from the fingerprint location when there are common objects visible in both the query image and the fingerprint image, which leads to accuracy degradation due to the improper nearest neighbor being selected to participate in the position calculation. To solve this problem, the K-nearest neighbor (KNN) method is applied in fingerprinting-based localization [56,57]. The KNN method selects the K-nearest fingerprint images and takes the average of their position coordinates as the estimated query position, avoiding the contingency of taking the nearest fingerprint image [58]. It is worth noting that each nearest neighbor in the KNN method has an equal contribution to the position estimation, which is unreasonable, because the average of nearest neighbor positions is hardly in accordance with the query position. More rational thinking is that the nearest neighbor with more similarity to the query is assigned a larger weight for the KNN method. Therefore, a weighted KNN (WKNN) method is presented in this paper to solve the estimated position of the query, taking full consideration of image similarities.

3. Image Clustering Based on Change-Point Detection in Global Features

A typical indoor visual localization system contains two stages: an on-line stage and an off-line stage, as shown in Figure 1. In the off-line stage, images are captured by the database camera mounted on the mapping equipment, and poses of the equipment are recorded simultaneously. In order to construct an indoor 3D map, Microsoft Kinects and laser scanners are also mounted on the mapping equipment [19]. An off-line database should be generated before the implementation of visual localization. It contains the essential elements for localization: database images, poses (including orientations and positions) of the equipment, and indoor 3D maps.

In the on-line stage, a query image is captured by the user and uploaded to the server by wireless networks. The most similar database images (i.e., matched database images) to the query image are retrieved based on the visual features extracted from the images. Then, the position of the matched database images can be employed to estimate the position of the query camera. Accuracy and efficiency of image retrieval are the key to ensuring the good performance of the system. A hierarchical clustering-based image retrieval is proposed in this paper and mainly discussed in the following.

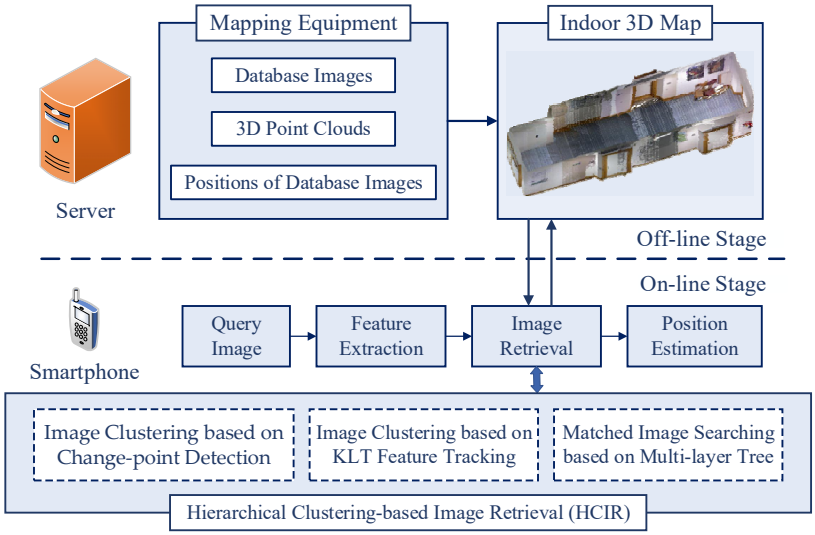

Figure 1. System model of indoor visual localization.

3.1. Feature Extraction and Pre-Processing

Gist is a scene-centered global feature commonly used in scene classification and place recognition. In this paper, Gist features are employed for scene-level image clustering by change-point detection. In the implementation of creating the off-line database (i.e., an indoor 3D map), database images are successively acquired by the mapping equipment in the same indoor scene, and visual features of these images have high correlations. In contrast, when the mapping equipment moves from one indoor scene to another, the correlations of the captured database images are weakened. Based on this characteristic of database images, change-points can be detected in the global features extracted from the images, resulting in the database images captured in the same indoor scene being grouped in a cluster, which achieves scene-level image clustering. The center of each cluster represents the main features of the scene, so that the query image orderly retrieves each cluster by measuring the difference between the query image and the centers of database image clusters.

To reduce computation complexity, visual feature vectors should keep a low dimension, so an image is regarded as a whole from which global features are extracted. For each query image and database image, a three-scale ($S_G = 1, 2, 3$) and six-orientation ($O_G = 0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ$) filters are used in Gist feature extraction, where S_G and O_G present spatial scale levels and cardinal orientations, respectively. During feature extraction, images are processed via convolution operation by multi-channel filters, and then the filtering results are connected to achieve 18 ($3 \times 6 = 18$) dimensional feature vectors. $\boldsymbol{G}_Q$ and $\boldsymbol{G}_D$ denote global features extracted from the query image and database images, respectively. Figure 2 shows examples of database images and the corresponding spectrograms of Gist features.

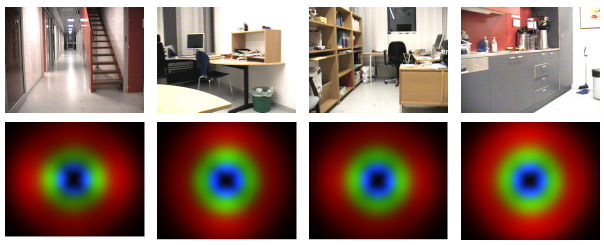

Figure 2. Examples of database images and the corresponding spectrograms of Gist features.

For visual localization, the query image is captured by a hand-held smartphone, and image retrieval is completed on the server side after the image is uploaded to the server. The Gist feature vectors of the query image and database images are separately presented

by $\boldsymbol{G}_Q = \left[g_Q^1, \cdots g_Q^p, \cdots g_Q^{18}\right]$ and $\boldsymbol{G}_D^i = \left[g_D^{(i,\,1)}, \cdots, g_D^{(i,\,q)}, \cdots, g_D^{(i,\,18)}\right]$, where g_Q^p and $g_D^{(i,\,q)}$ denote the p-th and q-th elements in the feature vector $\boldsymbol{G}_Q$ and $\boldsymbol{G}_D$ ($1 \leq p \leq 18$ and $1 \leq q \leq 18$), and i is the index of database images. The database images successively acquired in the same indoor scene have high visual correlations. However, once the mapping equipment switches to another scene, the correlations subsequently decrease. The change-point detection method is employed, aiming to detect the change in the visual correlations between database images, and further, the database images captured in the same scene are grouped into one cluster.

If there are in total n_D database images in the indoor 3D map (i.e., the off-line database), n_D features can be extracted from the database images. Therefore, a Gist feature matrix $\boldsymbol{M}_G$ can be obtained by organizing all feature vectors:

$$\boldsymbol{M}_G = \begin{bmatrix} g_D^{(1,\,1)} & \cdots & g_D^{(1,\,q)} & \cdots & g_D^{(1,\,18)} \\ \vdots & & \vdots & & \vdots \\ g_D^{(i,\,1)} & \cdots & g_D^{(i,\,q)} & \cdots & g_D^{(i,\,18)} \\ \vdots & & \vdots & & \vdots \\ g_D^{(n_D,\,1)} & \cdots & g_D^{(n_D,\,q)} & \cdots & g_D^{(n_D,\,18)} \end{bmatrix} \tag{1}$$

To achieve scene-level clustering, change-point detection acts on each column in $\boldsymbol{M}_G$. Specifically, change points should be separately detected in the column vector $\boldsymbol{M}_G^q = \left[g_D^{(1,\,q)}, \cdots, g_D^{(i,\,q)}, g_D^{(n_D,\,q)}\right]^T$, where $1 \leq q \leq 18$. On account of noise existing in feature extractions, a pre-process should be applied on Gist features. Therefore, the Kalman filter and the Kalman smoother are used in this paper to recover the visual correlations of database images acquired in the same scene. The purpose of feature pre-processing is to avoid false detection caused by noise, namely, detecting a change point without a scene change.

If the state variable and the observed variable of features are separately set as $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, the system equation is:

$$\begin{cases} \boldsymbol{x}_i = \boldsymbol{\Phi}_{i-1}\boldsymbol{x}_{i-1} + \boldsymbol{w}_i \\ \boldsymbol{y}_i = \boldsymbol{H}_i\boldsymbol{x}_i + \boldsymbol{v}_i \end{cases} \tag{2}$$

where $\boldsymbol{\Phi}_{i-1}$ and $\boldsymbol{H}_i$ are gain matrixes. The process noise $\boldsymbol{w}_i$ and measurement noise $\boldsymbol{v}_i$ satisfy:

$$p(\boldsymbol{w}_i) \sim \mathcal{N}(0, \Gamma) \tag{3}$$

$$p(\boldsymbol{v}_i) \sim \mathcal{N}(0, \Sigma) \tag{4}$$

where $\mathcal{N}(\mu, \sigma)$ denotes a normal distribution with the variance σ and the expectation μ. The initial value of $\boldsymbol{x}_i$ is defined as $\boldsymbol{x}_0 = \boldsymbol{\mu}_0 + \boldsymbol{u}$, where $p(\boldsymbol{x}_1)$ satisfies $\mathcal{N}(\mu_0, V_0)$ and $p(\boldsymbol{u})$ satisfies $\mathcal{N}(0, V_0)$ [59].

The discrete Kalman filter estimates the process state by feedback control. The typical Kalman filter can be divided into two parts, namely, the time-update part and the measurement-update part. In the time-update part, to obtain the prior estimate of the next time state, the Kalman filter calculates the state variables of the current time and the estimated covariance of errors by the update equation. In the measurement-update section, new observations are combined with prior estimates to obtain more reasonable posterior estimates by feedback operations.

When using a discrete Kalman filter to process data, the previous error covariance $\boldsymbol{P}_i$ before system updating should be calculated by:

$$\boldsymbol{P}_i = \boldsymbol{\Phi}_{i-1}\boldsymbol{V}_{i-1}\boldsymbol{\Phi}_{i-1}^{\mathrm{T}} + \Gamma \tag{5}$$

where $\boldsymbol{V}_{i-1}$ is the error covariance after system updating. The Kalman gain K_i can be further calculated based on the error covariance $\boldsymbol{P}_i$ by:

$$K_i = \boldsymbol{P}_i \boldsymbol{H}_i^{\mathrm{T}} \left(\boldsymbol{H}_i \boldsymbol{P}_i \boldsymbol{H}_i^{\mathrm{T}} + \Sigma \right)^{-1} \tag{6}$$

According to the un-updated error covariance and Kalman gain, the error covariance can be updated by:

$$\boldsymbol{V}_i = (\boldsymbol{I} - K_i \boldsymbol{H}) \boldsymbol{P}_i \tag{7}$$

Then, the posterior estimates of the state variable $\hat{\boldsymbol{x}}_i'$ (updated value) can be obtained by:

$$\hat{\boldsymbol{x}}_i' = \hat{\boldsymbol{x}}_i + K_i (\boldsymbol{y}_i - \boldsymbol{H}_i \hat{\boldsymbol{x}}_i) \tag{8}$$

where the prior estimate of the state variable can be obtained by the extrapolation formula:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{\Phi}_{i-1} \hat{\boldsymbol{x}}_{i-1}' \tag{9}$$

System iterative updates can be achieved by Equations (7) and (8), which achieves Kalman estimation for all measured values.

After Kalman filtering, Kalman smoothing needs to be performed on the filtering results. According to the Kalman backward smoothing equations, the smoothed estimate $\hat{\boldsymbol{x}}_i''$ of the state variable and the smoothed error covariance $\boldsymbol{V}_{\boldsymbol{i}}'$ can be obtained by:

$$\hat{\boldsymbol{x}}_i'' = \hat{\boldsymbol{x}}_i' + \boldsymbol{J}_i \left(\hat{\boldsymbol{x}}_{i+1}'' + \boldsymbol{\Phi}_i \hat{\boldsymbol{x}}_i' \right) \tag{10}$$

$$\boldsymbol{V}_{\boldsymbol{i}}' = \boldsymbol{V}_i + \boldsymbol{J}_i \left(\boldsymbol{V}_{i+1}' - \boldsymbol{P}_i \right) \boldsymbol{J}_i^{\mathrm{T}} \tag{11}$$

where $\boldsymbol{J}_i$ is defined as:

$$\boldsymbol{J}_i = \boldsymbol{V}_i \boldsymbol{\Phi}_i^{\mathrm{T}} (\boldsymbol{P}_i)^{-1} \tag{12}$$

The optimized vector $\boldsymbol{M}_K^q = \left[g_K^{(1,\,q)}, \cdots, g_K^{(i,\,q)}, \cdots, g_K^{(n_D,\,q)} \right]^{\mathrm{T}}$ can be obtained based on the Kalman smoother by Equations (9) and (10), where $g_K^{(i,\,q)}$ is the feature element after Kalman filtering and smoothing. n_D is the total number of database images, and q is the index of feature vectors. As 18 Gist features are extracted from a database image, the range of q satisfies $1 \leq q \leq 18$.

3.2. Image Clustering Based on CUSUM Change-Point Detection

Change-point detection on Gist features of database images is to group the successive database images between change points into one cluster, thereby realizing scene-level database image clustering. For a random process that occurs in chronological order, change-point detection is detecting whether the distribution or distribution parameters of random elements in the process suddenly change at a certain moment. In this paper, change points are detected on the Gist features extracted from successive database images, thereby finding the database image in which the indoor scene changes. When the change-point detection on all Gist features is completed, the database images between the change points are grouped into one cluster, and these images are deemed to be acquired in the same scene (e.g., office, kitchen, corridor, etc.). The rationality of the algorithm is that when constructing the off-line 3D indoor map, the database camera successively captures the database images in the same scene. Therefore, the obtained database images are successive in the same scene. That is, the Gist features of the database images have a certain correlation. Once the indoor scene changes, it is possible to perceive the occurrence of such a change by detecting the change points of the Gist features.

The CUSUM (Cumulative Sum) algorithm used for change-point detection in this paper is an anomaly detection method commonly used in industrial fields. The CUSUM algorithm is generally applied to all data to detect change points. For data located at a

certain position in the sequence, other data in front of and behind this position are used for change-point detection. However, such a detection method will undoubtedly increase the time overhead, especially as the amount of historical data becomes larger and larger with time, which will eventually cause excessive time overhead. Therefore, a sliding window is introduced to constrain the number of Gist features that need to be processed in change-point detection, as shown in Figure 3.

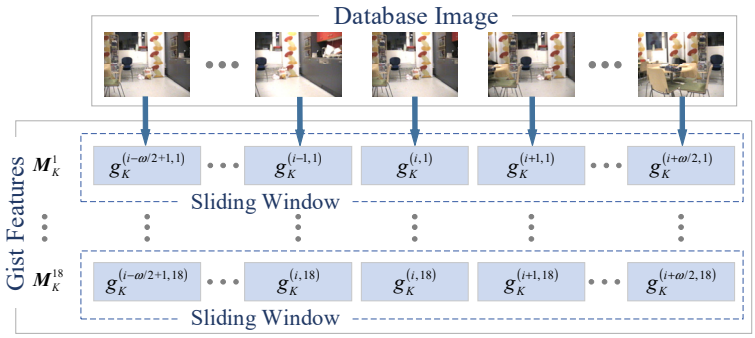

Figure 3. Diagram of sliding windows in CUSUM change-point detection.

As shown in Figure 3, an 18-dimensional Gist feature vector can be extracted with different scales and directions for each database image. After Kalman filtering and smoothing, each Gist feature can be represented as $g_K^{(i,\,q)}$, where the superscript i indicates the index of database images, and the subscript q indicates the element position in a feature vector. For the Gist features extracted from different database images, if they are located in the same column q, the scales and orientations of these features are the same. In order to constrain the data size during change-point detection, a sliding window of the size ω is implemented in CUSUM change-point detection. Specifically, if the currently detected position is i, and the corresponding Gist feature is $g_K^{(i,\,q)}$, only the features in the sliding window are considered. That is to say, only the features between $g_K^{(i-w/2+1,\,q)}$ and $g_K^{(i+w/2,\,q)}$ are detected. In addition, the change-point detection algorithm is only applied to the features that have the same scale and orientation. For different Gist feature sequences, such as $M_K^1, \cdots, M_K^{17}$, and M_K^{18}, they should be separately detected.

In the CUSUM change-point detection, considering the sequence of successively acquired database images is a time series, the acquisition time corresponds to the index of the database image in the sequence. Therefore, the change-point detection in this paper detects the position at which the change point appears in the image sequence. The CUSUM change-point detection algorithm estimates the position of the change point in the sequence by calculating the parameter models of Gist feature sequences. The probability density function is employed to determine the positions of change points in a Gist feature sequence. For a Gist feature sequence $\boldsymbol{M}_K^q = \left[g_K^{(1,\,q)}, \cdots, g_K^{(i,\,q)}, \cdots, g_K^{(n_D,\,q)}\right]^{\mathrm{T}}$, a sub-sequence $\boldsymbol{M}_W^q = \left[g_K^{(i-w/2+1,\,q)}, \cdots, g_K^{(i,\,q)}, \cdots, g_K^{(i+w/2,\,q)}\right]^{\mathrm{T}}$ within the sliding window can be obtained, and change-point detection only acts on feature element $g_K^{(i,\,q)}$. According to the position of $g_K^{(i,\,q)}$, sequence $\boldsymbol{M}_W^q$ can be divided into two sub-sequences: $\boldsymbol{M}_W^A = \left[g_K^{(i-w/2+1,\,q)}, \cdots, g_K^{(i,\,q)}\right]^T$ and $\boldsymbol{M}_W^B = \left[g_K^{(i+1,\,q)}, \cdots, g_K^{(i+w/2,\,q)}\right]^{\mathrm{T}}$.

According to the Neyman–Person lemma, the core of CUSUM change-point detection can be considered a hypothesis-test problem. For this hypothesis test, the null hypothesis H_0 is the case that the feature element is not a change point. In this case, the indoor scene corresponding to the database image does not change. In contrast to the null hypothesis, alternative hypothesis H_1 indicates the scene changes, in which case the feature element does not satisfy the previous parameter model. The purpose of the CUSUM algorithm is to monitor and determine at which point hypothesis H_1 switches to H_0 in the feature sequence.

For sub-sequences $\boldsymbol{M}_W^A$ and $\boldsymbol{M}_W^B$ in the sliding window, it is considered that the feature elements in the sequences are independent variables and subject to the normal distribution, so two parameter models, namely, parameter model A and parameter model B, can be obtained.

The probability density functions of the two parameter models are f_A and f_B which satisfy:

$$f_A\left(g_K^{(i,\,q)}\right) \sim \mathcal{N}(\mu_A, \sigma_A) \tag{13}$$

$$f_B\left(g_K^{(i,\,q)}\right) \sim \mathcal{N}(\mu_B, \sigma_B) \tag{14}$$

where μ_A and μ_B are the expectation and variance of the parameter model A. μ_B and σ_B are the expectation and variance of the parameter model B.

If the hypothesis test is applied to the feature elements in the sliding window, the probability density function under the null hypothesis H_0 is f_A, and the probability density function under the alternative hypothesis H_1 is f_B. Thus, a likelihood ratio function h_i relating to $g_K^{(i,\,q)}$ can be obtained by:

$$h_i\left(g_K^{(i,\,q)}\right) = \ln\left[f_B\left(g_K^{(i,\,q)}\right) / f_A\left(g_K^{(i,\,q)}\right)\right] \tag{15}$$

Based on the likelihood ratio function, a cumulative sum function can be defined as:

$$H_i\left(g_K^{(i,\,q)}\right) = \sum_{j=1}^{i} h_i\left(g_K^{(j,\,q)}\right) = \sum_{j=1}^{i} \ln\left[f_B\left(g_K^{(j,\,q)}\right) / f_A\left(g_K^{(j,\,q)}\right)\right] \tag{16}$$

For the CUSUM algorithm, the position t_{ch} of the change point can be calculated by a given threshold T_{ch}:

$$t_{ch} = \inf\left\{i \geq 1 : \left(H_i - \min_{1 \leq k \leq i} H_k\right) \geq T_{ch}\right\} \tag{17}$$

According to the detection principle shown in (17), a threshold must be set in advance when employing the typical CUSUM algorithm for change-point detection. However, in many cases, it is difficult to determine the threshold for change-point detection due to the complexity and diversity of indoor scenes. Therefore, an improved cumulative sum change-point detection (ICSCD) algorithm is proposed to identify change points without a given threshold.

Since the probability density functions of the two sub-sequences in the sliding window are subject to a normal distribution, the likelihood ratio function can be expanded as:

$$h_i\left(g_K^{(i,\,q)}\right) = \ln\left(f_B\left(g_K^{(i,\,q)}\right) / f_A\left(g_K^{(i,\,q)}\right)\right) = \ln\left(\frac{\left(\exp\left(-\left(g_K^{(i,\,q)} - \mu_B\right)^2 / 2\sigma_B^2\right)\right) / \sqrt{2\pi}\sigma_B}{\left(\exp\left(-\left(g_K^{(i,\,q)} - \mu_A\right)^2 / 2\sigma_A^2\right)\right) / \sqrt{2\pi}\sigma_A}\right) \tag{18}$$

If the variances of parameter model A and parameter model B are considered identical (i.e., $\sigma_A = \sigma_B = \sigma_{AB}$), then the likelihood ratio function h_i can be simplified as:

$$S_L\left(g_K^{(i,\,q)}\right) = \frac{2(\mu_A - \mu_B)g_K^{(i,\,q)} - \left(\mu_A^2 - \mu_B^2\right)}{2\sigma_{AB}^2} \tag{19}$$

The cumulative sum function of $g_K^{(i,\,q)}$ in the proposed ICSCD algorithm is defined as:

$$H_i\left(g_K^{(i,\,q)}\right) = \sum_{j=1}^{i} h_i\left(g_K^{(i,\,q)}\right) = \sum_{j=1}^{i}\left(\frac{2(\mu_A - \mu_B)g_K^{(i,\,q)} - \left(\mu_A^2 - \mu_B^2\right)}{2\sigma_{AB}^2}\right) \tag{20}$$

where expectations μ_A and μ_B corresponding to parameter models A and B can be calculated by:

$$\mu_A = \frac{2}{w} \sum_{k=i-w/2+1}^{i} g_K^{(k,\,q)} \tag{21}$$

$$\mu_B = \frac{2}{w} \sum_{k=i+1}^{i+w/2} g_K^{(k,\,q)} \tag{22}$$

According to Equation (20), the cumulative sum function depends on three variables: the Gist feature element $g_K^{(i,\,q)}$ detected as the change-point, the expectation μ_A of parameter model A, and the expectation μ_B of parameter model B. Therefore, the numerator of Equation (20) can be defined as a change-point detection function to monitor whether the indoor scene changes on the position i:

$$F_C\left(g_K^{(i,\,q)}\right) = 2(\mu_A - \mu_B)g_K^{(i,\,q)} - \left(\mu_A^2 - \mu_B^2\right) \tag{23}$$

Depending on the above analysis, Gist feature sequences can be processed by the change-point detection function. The specific process is as follows: first, for the feature elements between $w/2$ and $(n_D - w/2)$ in the sliding window, the values of the change-point detection function need to be calculated, where w is the size of the window and n_D is the total number of database images. Second, the peaks of the discrete values of function F_C are detected, and the peaks correspond to the change points in feature sequences. More than one Gist feature sequence is extracted from the database images (18 feature sequences extracted in this paper, i.e., M_K^q, $1 \le q \le 18$), and each feature sequence needs to be detected separately. Therefore, it is necessary to propose a strategy to integrate the detected change points from each feature sequence to find the images corresponding to indoor scene changes.

Since 18 Gist feature elements are extracted from each database image, an $18 \times n_D$ matrix containing change-point marks can be obtained for n_D database images. Each mark in the matrix represents whether the Gist feature element on this position is a change point. Specifically, if the feature element in one position is detected as a change point, the value of the mark in this position is defined as 1. Otherwise, the value is defined as 0. As shown in Figure 4, a window with the size l_w (l_w is an odd number, namely $l_w = 2k_w + 1$, and k_w is a positive integer) slides on the matrix, and the values of the marks in the window are added up. The meaning of the accumulated value of the marks is the total number of change points in the window. If the accumulated value of the marks in the window exceeds a given threshold, the center of the window is considered the position of the change point.

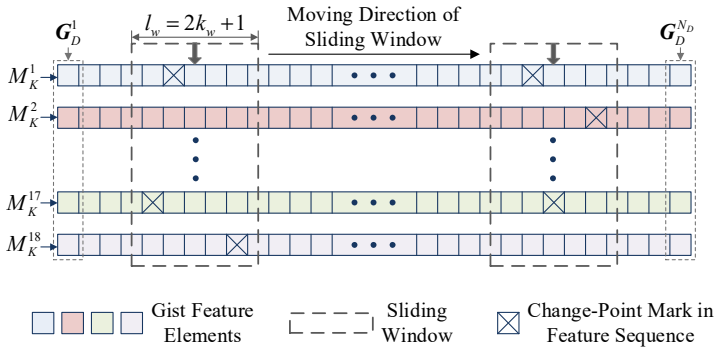

Figure 4. Diagram of change-point detection for database images.

4. Feature Tracking-Based Image Clustering, Hierarchical Retrieval and Visual Localization

Based on the proposed ICSCD algorithm, scene-level database image clustering can be achieved by global visual features of database images. However, in the implementation

of database construction, there are usually too many database images acquired in the same indoor scene, which leads to excessive time overheads of image retrieval in the on-line stage. Therefore, the database images in the same scene-level cluster are further grouped to achieve sub-scene-level clustering. For sub-scene-level clustering, local visual features of images are employed. In the process of local feature tracking, if the change rate of the number of tracked features is below the given threshold, the database image is defined as a breakpoint image. The database images between the two breakpoint images are acquired in the same sub-scene, so the images between the breakpoints are grouped into one sub-cluster.

4.1. Image Clustering Based on KLT Feature Tracking

A database image can be described by the gray-level function $I(x_I,\ y_I,\ t)$, where $(x_I,\ y_I)$ is the position of a pixel on the image, and t presents the time stamp. Since database images are successively acquired, time stamp t is equivalent to the image index. In addition, the interval of image acquisition is small, resulting in high visual correlations between database images, which is the precondition for using feature tracking for image breakpoint detection. When the database camera moves within a sub-scene, an overlap exists between the adjacent database images, and a certain number of local features on the images can be tracked. According to this characteristic of database images, a KLT (Kanade–Lucas–Tomasi) feature tracking-based image-clustering method is proposed in this section.

The features are continuously tracked between the two adjacent database images with index i and $i+1$ by the KLT algorithm. Let Φ_1 denote an image cluster in the scene-level clustering results. In the off-line stage, the local visual features (i.e., ORB features) are extracted from images and stored in the database used in the sub-scene-level image clustering. Let $\boldsymbol{L}_O^i = \left[\boldsymbol{l}_i^1,\ \cdots,\ \boldsymbol{l}_i^j,\ \cdots,\ \boldsymbol{l}_i^{n_O}\right]^{\mathrm{T}}$ present an ORB feature matrix consisting of the vector $\boldsymbol{l}_i^j = \left[l_i^{(j,\,1)},\ \cdots,\ l_i^{(j,\,k)},\ \cdots,\ l_i^{(j,\,32)}\right]$, where i is the index of the database image, j is the index of feature vectors, and k $(1 \le k \le 32)$ is the index of elements in a feature vector. For each ORB feature extracted from an image, there is a feature vector (i.e., $\boldsymbol{l}_i^j$) and a position vector (i.e., $\boldsymbol{p}_i^j = \left[x_I^{(i,\,j)},\ y_I^{(i,\,j)}\right]^{\mathrm{T}}$) corresponding to the ORB feature, where $\boldsymbol{p}_i^j$ is expressed in the image coordinate system.

To track features, a rectangular window on the image needs to be set whose length is $(2w_K+1)$ pixels and whose width is $(2h_K+1)$ pixels. Based on the assumptions of constant brightness, time continuity, and spatial consistency in the rectangular window, it is considered that the matching feature points on the database image satisfy the following relationship:

$$J\left(x_I^{(i,\,j)}+d_x,\ y_I^{(i,\,j)}+d_y,\ i+1\right) = I\left(x_I^{(i,\,j)},\ y_I^{(i,\,j)},\ i\right) \tag{24}$$

where $I\left(x_I^{(i,\,j)},\ y_I^{(i,\,j)},\ i\right)$ is the gray value of the feature point located at $\left[x_I^{(i,\,j)},\ y_I^{(i,\,j)}\right]^{\mathrm{T}}$ on the i-th database image. For the next database image, $J\left(x_I^{(i,\,j)}+d_x,\ y_I^{(i,\,j)}+d_y,\ i+1\right)$ is the gray value of the feature located at $\left[x_I^{(i,\,j)}+d_x,\ y_I^{(i,\,j)}+d_y\right]^{\mathrm{T}}$. d_x and d_y denote the displacement distances in the X and Y directions on the image, respectively. Equation (24) indicates that for the matching feature points on the database image, only the displacement changes occur on the adjacent images, and the magnitude of the gray values does not change. The core of feature tracking is to solve $\boldsymbol{d}_{xy} = [d_x,\ d_y]^{\mathrm{T}}$.

In order to obtain the displacement change, a sum of the squared intensity difference function is defined as:

$$\varepsilon(\boldsymbol{d}_{xy}) = \sum_{x=x_I^{(i,\,j)}-w_K}^{x_I^{(i,\,j)}+w_K} \sum_{y=y_I^{(i,\,j)}-h_K}^{y_I^{(i,\,j)}+h_K} \left(I(x,\,y,\,i) - J\left(x+d_x,\,y+d_y,\,i+1\right)\right)^2 \tag{25}$$

Taking the derivative of the sum of the squared intensity difference function and setting it to zero, the optimal solution $\boldsymbol{d}_{xy}^*$ of the displacement can be obtained by:

$$\frac{\partial \varepsilon(\boldsymbol{d}_{xy})}{\partial \boldsymbol{d}_{xy}} = -2 \sum_{x=x_I^{(i,\,j)}-w_K}^{x_I^{(i,\,j)}+w_K} \sum_{y=y_I^{(i,\,j)}-h_K}^{y_I^{(i,\,j)}+h_K} (I-J)\cdot\left[\frac{\partial J}{\partial x},\,\frac{\partial J}{\partial y}\right] = 0 \tag{26}$$

where $I(x,\,y,\,i)$ and $J\left(x+d_x,\,y+d_y,\,i+1\right)$ are abbreviated to I and J, respectively.

Based on the Taylor formula, the first-order approximation of Equation (26) on $[0,0]^{\mathrm{T}}$ can be obtained by:

$$\frac{\partial \varepsilon(\boldsymbol{d}_{xy})}{\partial \boldsymbol{d}_{xy}} \approx -2 \sum_{x=x_I^{(i,\,j)}-w_K}^{x_I^{(i,\,j)}+w_K} \sum_{y=y_I^{(i,\,j)}-h_K}^{y_I^{(i,\,j)}+h_K} \left(I - J - \left[\frac{\partial J}{\partial x},\,\frac{\partial J}{\partial y}\right]\boldsymbol{d}_{xy}\right)\cdot\left[\frac{\partial J}{\partial x},\,\frac{\partial J}{\partial y}\right] \tag{27}$$

Equation (27) can be further expressed as:

$$\left[\frac{\partial \varepsilon(\boldsymbol{d}_{xy})}{\partial \boldsymbol{d}_{xy}}\right]^{\mathrm{T}} \approx -2 \sum_{x=x_I^{(i,\,j)}-w_K}^{x_I^{(i,\,j)}+w_K} \sum_{y=y_I^{(i,\,j)}-h_K}^{y_I^{(i,\,j)}+h_K} \left(\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\boldsymbol{d}_{xy} - \begin{bmatrix} H_{xy}I_x \\ H_{xy}I_y \end{bmatrix}\right) \tag{28}$$

where I_x, I_y, and H_{xy} are:

$$I_x = \frac{\partial I(x,\,y,\,i)}{\partial x} = \frac{I(x+1,\,y,\,i) - I(x-1,\,y,\,i)}{2} \tag{29}$$

$$I_y = \frac{\partial I(x,\,y,\,i)}{\partial y} = \frac{I(x,\,y+1,\,i) - I(x,\,y-1,\,i)}{2} \tag{30}$$

$$H_{xy} = I(x,\,y,\,i) - J(x,\,y,\,i) \tag{31}$$

Let Equation (28) be equal to zero, and the optimal solution $\boldsymbol{d}_{xy}^*$ can be obtained by:

$$\boldsymbol{d}_{xy}^* = \left(\sum_{x=x_I^{(i,\,j)}-w_K}^{x_I^{(i,\,j)}+w_K} \sum_{y=y_I^{(i,\,j)}-h_K}^{y_I^{(i,\,j)}+h_K} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\right)^{-1} \times \left(\sum_{x=x_I^{(i,\,j)}-w_K}^{x_I^{(i,\,j)}+w_K} \sum_{y=y_I^{(i,\,j)}-h_K}^{y_I^{(i,\,j)}+h_K} \begin{bmatrix} H_{xy}I_x \\ H_{xy}I_y \end{bmatrix}\right) \tag{32}$$

Initialization should be applied to database image set Φ_1, which means that the first database image in the set Φ_1 is defined as a breakpoint. From the second database image in the set Φ_1, the KLT algorithm is employed to track local features between the adjacent images. Let k denote the index of the breakpoint image, and then the number of tracked features can be presented by the vector $\boldsymbol{n}_T = \left[n_T^{k+1},\, n_T^{k+2}, \cdots,\, n_T^{k+k'}, \cdots\right]$, where $n_T^{k+k'}$ is the number of tracked features corresponding to the image with index $k+k'$. To determine the position of the breakpoint image, a threshold is required to monitor the number of

tracked features. For the database image with index $k+k'$, the corresponding breakpoint image-detection threshold $T_{Tr}^{k+k'}$ is defined as:

$$T_{Tr}^{k+k'} = \frac{w_{Tr}}{k'} \sum_{j=k+1}^{k+k'} \left(\frac{n_T^{j-1} - n_T^j}{n_T^j} \right) \tag{33}$$

where w_{Tr} is the scale coefficient, and the change rate of the number of tracked features is $r_{Tr}^{k+k'} = \left(n_T^{k+k'-1} - n_T^{k+k'} \right) / n_T^{k+k'}$.

According to the change rate $r_{Tr}^{k+k'}$ and threshold $T_{Tr}^{k+k'}$, an image can be determined whether or not it is a breakpoint image. Specifically, if $r_{Tr}^{k+k'} \leq T_{Tr}^{k+k'}$, the database image with index $k+k'$ is regarded as a breakpoint image, as shown in Figure 5. Breakpoint detection is applied to each scene-level clustering result, and then all breakpoint images in the database image set can be found. Database images between two breakpoint images and the front breakpoint image are grouped into one cluster, achieving the sub-scene-level image clustering. The feature vector of the breakpoint image of each cluster is the cluster center. That is, the feature vector of the first image in the cluster is the cluster center (as shown in Figure 5). In the i-th scene-level cluster, the center of the j-th sub-scene-level cluster can be denoted by $\boldsymbol{L}_i^j$, which is an ORB feature vector.

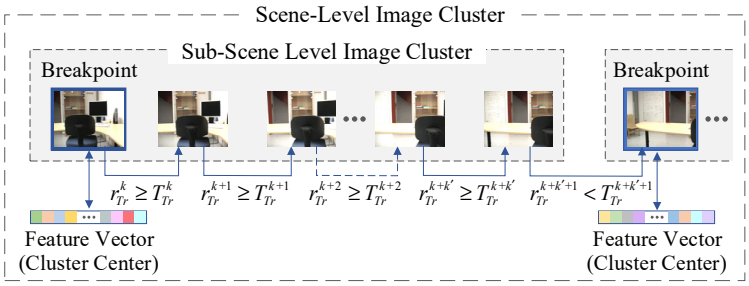

Figure 5. Illustration of sub-scene-level clustering based on feature tracking.

4.2. Hierarchical Image Retrieval and Visual Localization

In the off-line stage, database images are hierarchically grouped, and a search tree with three layers is achieved: (1) the first layer contains centers of scene-level clusters, (2) the second layer consists of centers of sub-scene-level clusters, and (3) the third layer contains database images in sub-scene-level clusters. It should be noted that the center of the scene-level cluster is a global feature vector (i.e., a Gist feature vector), and the center of the sub-scene-level cluster is a local feature vector (i.e., an ORB feature vector). According to the results of hierarchical image clustering, a three-layer search tree can be organized. As shown in Figure 6, there are m clustering results in the first layer of the search tree, and each clustering result in the first layer, such as $\boldsymbol{G}_C^i$, corresponds to more than one second-layer result (i.e., $\boldsymbol{L}_i^1, \cdots, \boldsymbol{L}_i^{n_i}$). In addition, the database images grouped into one sub-cluster are associated with the cluster center, such as $\boldsymbol{L}_i^{n_i}$. Based on the organized multi-layer search tree, a hierarchical clustering-based image retrieval (HCIR) algorithm is proposed for visual localization.

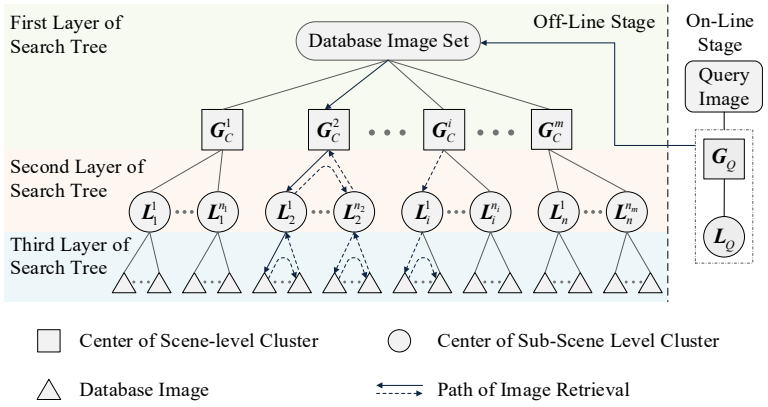

Figure 6. Schematic diagram of search tree in image hierarchical retrieval.

Based on the multi-layer search tree shown in Figure 6, hierarchical image retrieval is applied to find the most similar database image to the query image. In the on-line stage, global and local features are extracted from the query image and uploaded to the server by wireless networks. Then, the similarity between the query image and the centers of scene-level clusters can be defined as:

$$l_S = \left\| \boldsymbol{G}_Q - \boldsymbol{G}_C^i \right\| \tag{34}$$

where $\boldsymbol{G}_Q$ is the Gist feature vector extracted from the query image, and $\boldsymbol{G}_C^i$ $(1 \leq i \leq m)$ is a scene-level cluster center.

By measuring similarities between the global features of the query image and scene-level cluster centers, the scene-level clusters can be ranked. Then, the query image orderly retrieves each scene-level cluster. When the most similar scene-level cluster is found, the sub-scene-level clusters in that cluster should be sorted. Specifically, suppose a query image needs to find its most similar database image in the cluster $\boldsymbol{G}_C^i$. In that case, local features extracted from the query image should be matched with each center of sub-scene-level clusters, i.e., $\boldsymbol{L}_i^1, \cdots, \boldsymbol{L}_i^{n_i}$. The number of matched local features reflects the similarities between the query image and the sub-scene cluster centers. For the cluster $\boldsymbol{G}_C^i$, sub-scene-level clusters are orderly retrieved by the query image. By this means, scene-level clusters and sub-scene-level clusters can be ranked based on visual similarities between the query image and cluster centers. According to the ranked clusters, local feature matching should be orderly executed between the query image and the database images in the third-layer clusters.

Let $n_{mat}^{k_s}$ denote the number of matched features between the query image and the k_s-th database image. Then, the feature matching ratio can be defined as:

$$s_L^{k_S} = n_{mat}^{k_S} / n_Q \tag{35}$$

where $0 \leq s_L^{k_S} \leq 1$ and n_Q is the number of the ORB features extracted from the query image.

The matched features are used in visual localization, and more matched features contribute to improving localization accuracy. In addition, more matched features indicate that the query image is closer to the database image. Therefore, the best-matched database image with the query image is desired to estimate the position of the query camera in visual localization. Since database images are successively captured, when the query image is matched with database images, the trend of feature-matching ratios presents regularity. Specifically, if the query image and the database image are acquired in the same scene, when the query image is orderly matched with the database image, the trend of matching ratios first increases and then decreases. The reason is that when the query image gradually approaches the best-matched database image, the matching ratios will gradually increase until the ratio reaches a maximum value. At this position, the database image is best

matched with the query image. After that, the distance between the query image and the best-matched database image gradually increases, and the matching ratios decrease. Based on the above analysis, if the maximum value of the matching ratios can be found, the best-matched database image can be determined. The method of finding the best-matched database image is named the maximum similarity method in this paper.

To find the best-matched database image, a sliding window should be set as shown in Figure 7. In a sliding window with the size $w_F + 1$, the index of the image at the center is k_S. If the image with the index k_S is determined as the best-matched database image, the matching ratio $s_L^{k_S}$ of the database image should satisfy:

$$\begin{cases} s_L^{k_S} \geq s_L^{k_S-1} \geq \cdots \geq s_L^{k_S-w_F/2} \geq r_{mat} \\ s_L^{k_S} \geq s_L^{k_S+1} \geq \cdots \geq s_L^{k_S+w_F/2} \geq r_{mat} \end{cases} \tag{36}$$

where $r_{mat} \in [0,\ 1]$ is the threshold of the matching ratio. The threshold ensures that the database images and the query image are captured in the same scene.

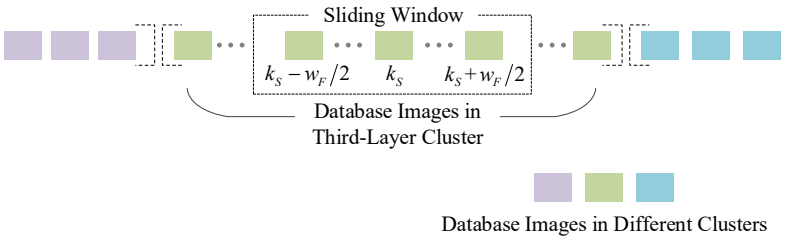

Figure 7. Diagram of maximum similarity method for finding the best-matched database image.

According to the ranking results of the scene-level and sub-scene-level clusters, the query image orderly retrieves each cluster until it finds the best-matched database image. A situation may arise in image retrieval. That is, the query image is matched with all the database images of the most similar scene-level cluster, but the best-matched database image is still not found. Therefore, a backtracking mechanism is introduced in the proposed hierarchical retrieval. In this mechanism, when the best-matched database image with the query image cannot be found after comparing with all the database images in the scene-level cluster, the query image will return to the top of the search tree and continue to retrieve the next cluster according to the ranking results, and so on. In the worst case, all database images are compared with the query image, and the best-matched database image is still not found. Then, the database image with the maximal matching ratio is determined as the best-matched image, but in this case, the distance between the query image and the database image is perhaps far.

By the proposed HCIR algorithm, the best-matched database image with the query image can be found in the database, and the best-matched database image has the following characteristics: (1) the database image is captured in the same scene as the query image; (2) there are a number of matching feature points between the query image and the database image. If the query image is considered to be coincident with the position of the best-matched database image, a preliminary position estimation of the query camera can be achieved. However, this position estimation method is subject to the acquisition density of the database images. In order to improve the localization accuracy, the top-*K* best-matched database images are selected and used to estimate the position of the query image.

4.3. Visual Localization Based on Weighted KNN Method and Armijo–Goldstein Algorithm

In practical localization scenarios, images with higher similarity tend to be closer. Namely, the more similar the query image and the database image are, the smaller the distance between the two images is. With this thinking, a weighted KNN-based visual localization method is proposed, by which the matched database image with a higher

similarity is assigned a larger weight. The similarities between images are evaluated by the number of matched feature points in visual localization.

The top-K best-matched database images with the query image are regarded as the nearest neighbors to estimate the query position, so the localization error function can be defined as:

$$f_e\left(\boldsymbol{p}_Q\right) = \sum_{i=1}^{K}\left(w_i\left\|\boldsymbol{p}_Q - \boldsymbol{p}_D^i\right\|\right) \tag{37}$$

where $\boldsymbol{p}_Q$ is the estimated position of the query image, and $\boldsymbol{p}_D^i$ $(1 \le i \le K)$ is the position of the database image. w_i is the weight that can be calculated by:

$$w_i = \frac{n_{mat}^i}{\sum_{j=1}^{K} n_{mat}^j},\ (i = 1,\ \cdots,\ K) \tag{38}$$

where n_{mat}^i denotes the number of matched feature points between the query image and the database image.

For Equation (37), the Armijo–Goldstein algorithm is used to solve the estimated position of the query image (i.e., the position of the query camera) [60]. The gradient vector of f_e at $\boldsymbol{p}_Q^k = \left[x_Q^k,\ y_Q^k\right]$ is:

$$\boldsymbol{g} = \left[\left.\frac{\partial f_e}{\partial x}\right|_{\left(x_Q^k,\ y_Q^k\right)},\ \left.\frac{\partial f_e}{\partial y}\right|_{\left(x_Q^k,\ y_Q^k\right)}\right] \tag{39}$$

According to the gradient vector, the search direction $\boldsymbol{s}$ can be further determined by $\boldsymbol{s} = -\boldsymbol{g}$. The procedure of visual positioning can be treated as a line search, as shown in Algorithm 1. The count flag s_k, index t_m, maximum number of iterations $k_{\max}$ (=5000), threshold σ_r (=10^{-3}), amplification coefficient γ (=0.4), and step length d (=0.01) are set as inputs.

Algorithm 1: Visual localization based on Armijo–Goldstein algorithm

Input: localization error function f_e, count flag s_k, index t_m, maximum number of iterations $k_{\max}$, threshold σ_r, amplification coefficient γ, and step length d

Output: estimated position $\boldsymbol{p}_Q$ of the query camera

Step 1: set initial values $s_k = 0$ and $t_m = 0$;

Step 2: calculate the norm of the direction vector by $s_n = \|\boldsymbol{s}\|$
if $s_n > \sigma_r$, turn to Step 3,
else turn to Step 5;

Step 3: **if** $f_e\left(\boldsymbol{p}_Q^k + d^{t_m}\boldsymbol{s}\right) < f_e\left(\boldsymbol{p}_Q^k\right) + \gamma d^{t_m}(\boldsymbol{g})^{-1}\boldsymbol{s}$, then $s_k = t_m$ and turn to Step 4,
else $t_m \leftarrow t_m + 1$ and turn to Step 4;

Step 4: update the position of the query camera by $\boldsymbol{p}_Q^{k+1} = \boldsymbol{p}_Q^k + d^{s_k}\boldsymbol{s}$ and $k \leftarrow k+1$,
if $k \ge k_{\max}$, then turn to Step 5,
else turn to Step 1;

Step 5: determine the position of the query camera by $\boldsymbol{p}_Q^k$, i.e., $\boldsymbol{p}_Q = \boldsymbol{p}_Q^k$.

With the visual localization method, the estimated position $\boldsymbol{p}_Q$ of the query camera can be achieved by solving the line search problem, in which the similarity between the query image and the database images is reflected by the weight w_i. Therefore, the estimated position of the query camera is closer to the database images with high similarity.

In summary, visual localization is achieved by two steps: hierarchical clustering-based image retrieval and query camera position estimation. Since the proposed visual

localization method dispenses with camera calibration, it can be widely used in different application scenarios and applied to various smart mobile terminals.

4.4. Performance Analysis on Hierarchical Image Retrieval

The proposed HCIR algorithm aims to decrease the on-line search time by sacrificing the processing time of off-line image clustering. Still, database image clustering is continuously efficacious, which means that once the database image clustering is completed, the results of clustering can be repeatedly applied to on-line image retrieval. The proposed algorithm in this paper achieves multi-layer image clustering. Compared with the single-layer clustering algorithm (i.e., only scene-level image clustering is implemented), an advantage of the proposed algorithm is that the search time does not scale up for the database size. Next, the computation performance of the proposed algorithm and the single-layer clustering-based algorithm will be analyzed in detail. For clustering-based image retrieval, on-line time consumptions contain five parts: (1) the time t_G to extract the global features of the query image, (2) the time t_L to extract the local features of the query image, (3) the time t_{GS} to measure the similarity of global features between the query image and database images, (4) the time t_{LM} to match the local features between the query image and database images, and (5) the time t_{FS} to sort database images according to their similarity to the query image. If there are n_{L_1} scene-level clusters in the first layer, and each cluster contains n_{L_2} sub-scene-level clusters, the average running time T_{SL} of the single-layer clustering-based retrieval algorithm is:

$$T_{SL} = t_G + t_L + n_{L_1} t_{GS} + t_{FS} + k_{L_1} m_{L_1} t_{LM} + \frac{1 + m_{L_1}}{2} t_{LM} \tag{40}$$

where k_{L_1} is the number of scene-level clusters that have been retrieved when the backtracking mechanism is enabled, and m_{L_1} and m_{L_2} are the number of database images in the scene-level cluster and the sub-scene-level cluster, separately.

The average running time T_{ML} of the proposed image retrieval algorithm is:

$$\begin{aligned} T_{ML} &= t_G + t_L + n_{L_1} t_{GS} + 2t_{FS} + k_{L_1} n_{L_1} t_{LM} + k_{L_2} m_{L_2} t_{LM} + n_{L_2} t_{LM} + \frac{1+m_{L_2}}{2} t_{LM} \\ &= t_G + t_L + n_{L_1} t_{GS} + 2t_{FS} + k_{L_1} n_{L_1} t_{LM} + \left(k_{L_2} m_{L_2} + n_{L_2}\right) t_{LM} + \frac{1+m_{L_2}}{2} t_{LM} \end{aligned} \tag{41}$$

where k_{L_2} is the image number of sub-scene-level clusters that have been retrieved when the backtracking mechanism is enabled. In this case, if the query image does not obtain a matched database image after retrieving a complete scene-level and sub-scene-level clustering result, the time consumption of the two processes is $n_{L_1} t_{LM}$ and $n_{L_2} t_{LM}$, respectively. The total number m_{total} of database images satisfies: $m_{total} = n_{L_1} m_{L_1}$ and $m_{L_1} = n_{L_2} m_{L_2}$, where m_{L_1} and m_{L_2} are image numbers of the scene-level cluster and the sub-scene-level cluster, respectively.

For single-layer clustering-based image retrieval, database images are orderly matched with the query, so the average retrieving time of an image cluster is $\left(1 + m_{L_1}\right) t_{LM}/2$. Similarly, for the proposed algorithm, the average retrieval time of a sub-scene-level cluster is $\left(1 + m_{L_2}\right) t_{LM}/2$. According to the principle of multi-layer clustering, the image number m_{L_1} is far more than m_{L_2}. As a result, the query image could find its matched database image in a cluster using less time for the proposed HCIR algorithm. The proposed algorithm has three additional time overheads (i.e., the time $k_{L_2} m_{L_2} t_{LM}$ to retrieve the k_{L_2} results, the time $n_{L_2} t_{LM}$ to match features, and the time t_{FS} to sort database images) compared with the single-layer clustering algorithm. However, in practical applications, feature-sorting time is much shorter than feature-matching time, and in most cases, the value of k_{L_2} is zero. Therefore, the sum of the retrieval time $\left(1 + m_{L_2}\right) t_{LM}$ and the feature-matching time $n_{L_2} t_{LM}$ is still less than the retrieval time $\left(1 + m_{L_1}\right) t_{LM}/2$. If the best-matched database image can be obtained without trigging the backtracking mechanism (i.e., $k_{L_1} = k_{L_2} = 0$),

the difference Δt in time consumption between the single-layer clustering algorithm and the proposed algorithm is:

$$\Delta t = \frac{m_{L_1} - m_{L_2} - 2n_{L_2}}{2} t_{LM} - t_{FS} \tag{42}$$

Compared with the multi-layer clustering-based algorithm, there are more database images contained in the cluster for the single-layer clustering-based algorithms. Moreover, as a multi-layer clustering-based algorithm, the proposed HCIR algorithm has priorities for retrieving the image clusters with high similarities to the query, so that the matched image can be found by searching fewer database images. Therefore, the structure of a multi-layer search tree is beneficial in reducing retrieval time consumption.

5. Experimental Results and Discussion

In this section, the hierarchical clustering-based image retrieval is implemented, and the computation performance of the retrieval algorithm is analyzed. In addition, the position accuracy of the visual localization is evaluated.

5.1. Experimental Results of Database Image-Clustering Algorithm

Two image databases (namely, the KTH image database [61] and the HIT-TUM image database) were used to evaluate the performance of the proposed algorithm. The images in the HIT-TUM database were acquired from the Harbin Institute of Technology and the Technical University of Munich. Each database contains 400 images captured in 10 different indoor scenes, such as an office, a corridor, a restaurant, and so on. All data processing was run on MATLAB 2018A with an Intel Core i7 CPU and 8GB RAM. Randomly selected example images in the databases are shown in Figure 8. It is worth noting that images in the databases are successively captured in indoor scenes, so the visual features extracted from the images captured in the same indoor scene have high correlations.

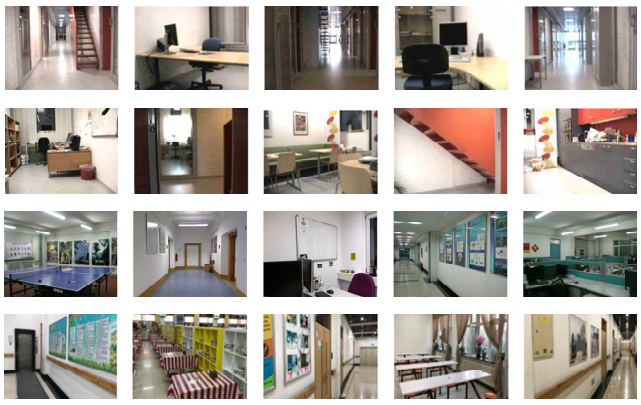

Figure 8. Randomly selected example images of the KTH and HIT-TUM databases.

For scene-level image clustering, database images are grouped by their global features based on the CUSUM change-point detection. The results of the scene-level clustering guide the query image to retrieve the clusters that are similar to the query. Therefore, the performance of database image clustering affects the efficiency of the image retrieval system. For an image retrieval system, the efficiency of the retrieval algorithm is reflected in two aspects: the number of searched database images and the time consumption of the image retrieval. Generally, the fewer database images are searched, the less time retrieval takes.

In the same indoor scene, as the database images are successively captured, the Gist features extracted from these images have high correlations. Taking advantage of the correlations, scene-level clustering of database images can be achieved. However, the noise generated in feature extraction affects the correlation of the features. Therefore, the original Gist features of database images need to be pre-processed (including Kalman filtering and

Kalman smoothing) to restore the correlation of image features. For a database image, Gist features can be extracted according to different scales and directions. In this paper, Gist features are extracted at three scales and six directions, so 18 $(3 \times 6 = 18)$ feature elements are extracted from each image. That is, the Gist feature vector of each image contains 18 feature elements. Figure 9 shows an example of the Gist feature pre-processing result of a database image, including the original Gist feature values, Kalman filtering results, and Kalman smoothing results. It can be found from Figure 9 that the correlation between original feature values is not evident due to the influence of noise. In contrast, by Kalman filtering, noise is effectively suppressed, and the correlations between features are restored. According to Kalman filtering results, more obvious correlations can be obtained by further Kalman smoothing of Gist features. The pre-processing of features recovers the correlations of global features of database images, which is beneficial to scene-level clustering.

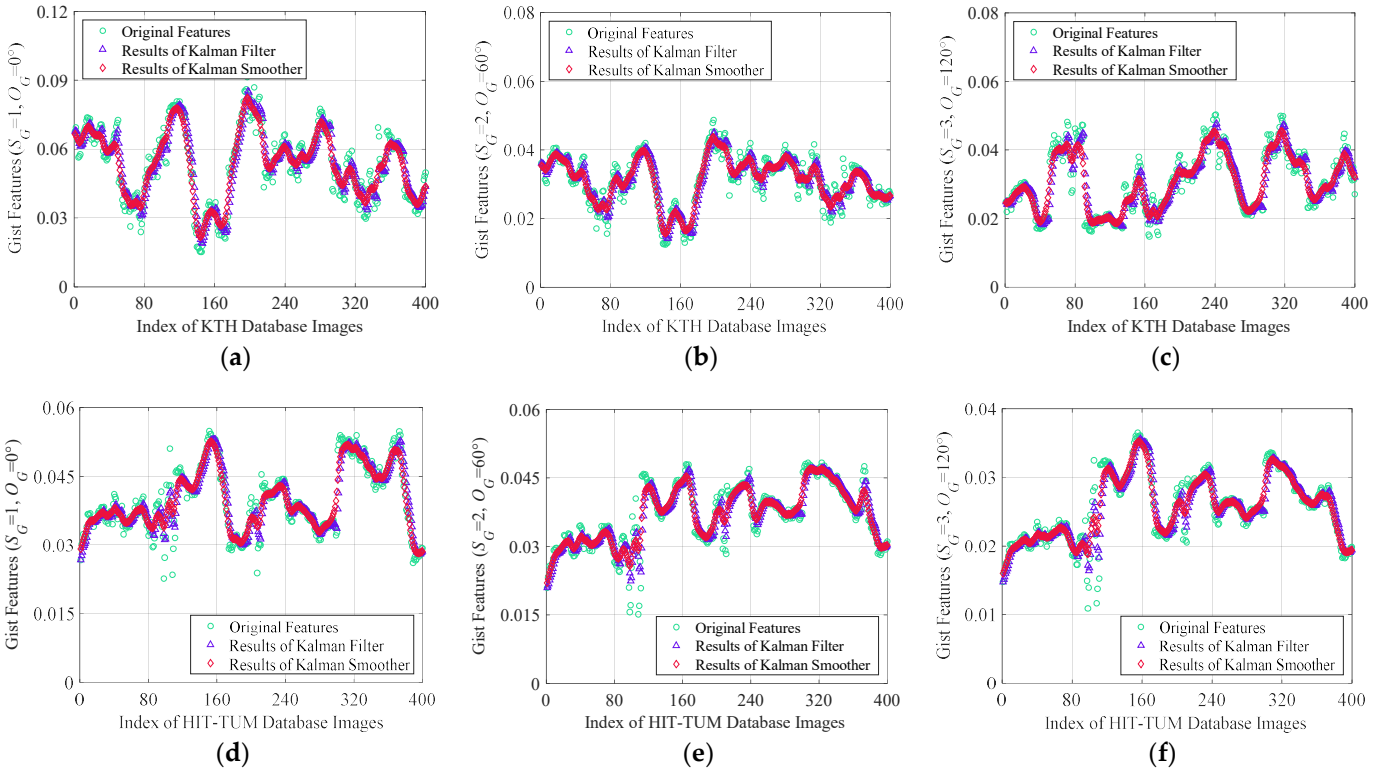

Figure 9. Examples of pre-processing results of Gist features of database images. (**a**) Pre-processing results of Gist features for KTH database images with $S_G = 1$ and $O_G = 0^\circ$; (**b**) Pre-processing results of Gist features for KTH database images with $S_G = 2$ and $O_G = 60^\circ$; (**c**) Pre-processing results of Gist features for KTH database images with $S_G = 3$ and $O_G = 120^\circ$; (**d**) Pre-processing results of Gist features for HIT-TUM database images with $S_G = 1$ and $O_G = 0^\circ$; (**e**) Pre-processing results of Gist features for HIT-TUM database images with $S_G = 2$ and $O_G = 60^\circ$; (**f**) Pre-processing results of Gist features for HIT-TUM database images with $S_G = 3$ and $O_G = 120^\circ$.

Scene-level database image clustering is achieved by detecting the change-points in Gist feature sequences. When image retrieval is executed in the results of scene-level clustering, according to the similarity of the global features, the query image will preferentially search the database image clusters with a higher similarity. Therefore, if all database images in the same scene are grouped into one cluster, the query image captured in this scene can find its matched database images in this cluster. In contrast, if a database image captured in a certain scene is falsely grouped into other clusters, this database image cannot be retrieved when the query searches the right cluster. Depending on the above analysis, the core of the proposed algorithm is that the database images in the same scene are grouped into one cluster as much as possible.

To analyze the performance of the proposed algorithm, scene-level image clustering is executed, and confusion matrices are employed to evaluate clustering accuracy. The confusion matrices of the results of database image clustering are shown in Figure 10. The confusion matrix used to evaluate clustering accuracy in this paper can also be regarded as a clustering error matrix. The row labels of the matrix are the correct cluster labels, and the column labels are the predicted cluster labels. For the matrix in Figure 10a, the values in the third, fourth, and fifth rows of the fourth column are 1, 39, and 6, respectively. This set of values shows that for 46 ($1 + 39 + 6 = 46$) database images that are grouped into one cluster, 39 database images were truly captured in the Office II scene, one image was misclassified into the Corridor II scene category, and six images were misclassified into the Corridor III scene. For a row of the confusion matrix, the sum of all values in that row represents the actual number of images in the cluster. For a column of the confusion matrix, the sum of all values in that column represents the predicted number of images in the cluster. The confusion matrix effectively reflects the performance of the proposed scene-level clustering algorithm. By observing the confusion matrix, it can be known that for image-clustering results, incorrectly grouped images only occur in two adjacent image clusters, and the reason accords with the clustering principle in this paper. Specifically, image clustering acts on the indoor database image sequence, and the change-points are detected based on global features of database images, so that images between two change-points are grouped into one cluster. Therefore, the incorrect image grouping is caused by errors in change-point detection. Obviously, if errors exist in change-point detection, some database images that should belong to a certain cluster are grouped into the former or the latter cluster.

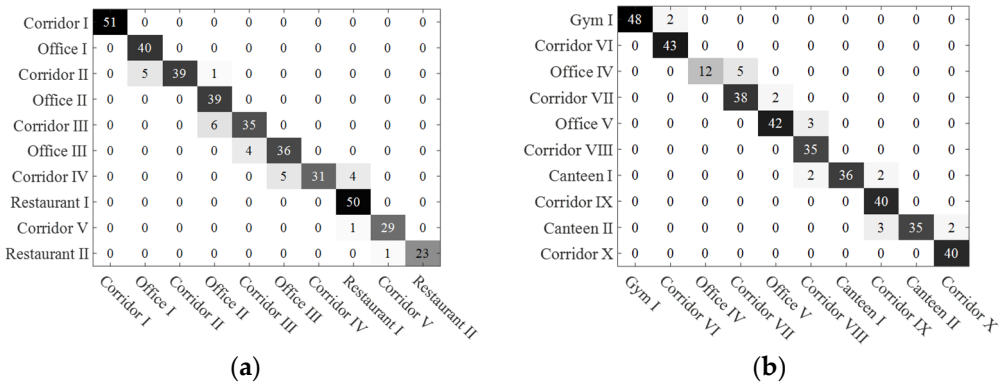

Figure 10. Confusion matrices of database image clustering. (**a**) Confusion matrix of KTH database image clustering; (**b**) confusion matrix of HIT-TUM database image clustering.

In this paper, four criteria (i.e., recall rate, precision rate, accuracy rate, and F1 score) are used to evaluate the performance of the clustering algorithm. The recall rate r_{RR} is the ratio of the number of correctly grouped images to the actual number of images in that cluster. The precision rate r_{PR} refers to the ratio of the number of correctly grouped images to the number of images in the cluster. The accuracy rate r_{AR} refers to the ratio of the number of correctly grouped images to the total number of images. The F1 score s_{F1} is used in statistics to measure the accuracy of a classification model. This score can be calculated by the recall rate and the precision rate:

$$s_{\mathrm{F1}} = \frac{2r_{RR} \cdot r_{PR}}{r_{RR} + r_{PR}} \tag{43}$$

Global features of an image include color features (such as color histogram features and color moment features) and texture features (such as wavelet transform features and Gabor transform features). In simulation experiments, Gist features, color histogram features, color moments, wavelet transform features, and Gabor features are used to perform scene-level clustering on database images, and experimental results are shown in Table 1. For

the experimental results, $\bar{r}_{RR}$, $\bar{r}_{PR}$ and $\bar{s}_{\mathrm{F1}}$ denote the average recall rates, average precision rates, and average F1 scores, respectively. From the results shown in Table 1, the color features (such as color histograms and color moments) perform weakly on scene-level clustering. The reason is that the color difference of indoor scenes is relatively small. Especially in an environment with a white wall as the main background, it is not easy to distinguish the scenes by the color information. Compared with color features of images, texture features of images perform better in terms of clustering performance, especially for Gabor features and Gist features. Because multiple Gabor filters with different scales and directions are used in extracting Gist features, Gist features describe the textures of scenes more comprehensively, thereby achieving more accurate image-clustering results.

Table 1. Performance comparison of scene-level clustering of global features.

Database	Type of Global Feature	Average Recall Rate $\bar{r}_{RR}$	Average Precision Rate $\bar{r}_{PR}$	Average F1 Score $\bar{s}_{\mathrm{F1}}$	Accuracy Rate $\bar{r}_{AR}$
KTH	Color histogram	0.5602	0.5559	0.5546	0.5575
	Color moment	0.5732	0.5697	0.5683	0.5650
	Wavelet transform	0.6946	0.6898	0.6904	0.6875
	Gabor transform	0.8178	0.8071	0.8102	0.8400
	Gist	0.9320	0.9388	0.9322	0.9325
HIT-TUM	Color histogram	0.5384	0.5370	0.5343	0.5475
	Color moment	0.5607	0.5599	0.5578	0.5725
	Wavelet transform	0.6601	0.6611	0.6592	0.6775
	Gabor transform	0.8259	0.8206	0.8221	0.8325
	Gist	0.9324	0.9489	0.9375	0.9225

To reveal the clustering performance of the ICSCD algorithm proposed in this paper, two typical change-point detection algorithms (i.e., the mean shift-based algorithm [36] and the Bayesian estimation-based algorithm [62]) are simulated for grouping database images at the scene level. The experimental results shown in Table 2 indicate that the proposed ICSCD algorithm significantly outperforms the Bayesian estimation-based algorithm in four metrics: the average recall rate, the average precision rate, the average F1 score, and the accuracy rate. The reason is that the Bayesian estimation-based algorithm utilizes local features in change-point detection, but the local features are too sensitive to scene changes and tend to group database images belonging to the same scene into multiple image clusters or group images belonging to the same class into other clusters. This also shows that the local features of the images are more suitable for further classification of the scene-level clustering results, which is why local features are used for the second layer of clustering in this paper. Both the proposed ICSCD algorithm and the mean shift-based algorithm use global features for clustering, but the difference is that the change-point detection function F_C is employed to detect the change points for image clustering in the proposed ICSCD algorithm, whereas the mean shift function is utilized to detect the change points in the mean shift-based algorithm. Since both the influence of the values at the detection position and the influence of the expected values of the parameter models (i.e., the parameter model A and the parameter model B in the hypothesis test) within the sliding window (as shown in Figure 4) are taken account in the change-point detection function F_C, a higher clustering accuracy can be obtained. Specifically, the average recall rate, the average precision rate, the average F1 score, and the accuracy rate of the proposed ICSCD algorithm are greater than 0.92, which is significantly higher than the mean shift-based algorithm.

Table 2. Performance comparison of scene-level image clustering algorithms.

Database	Algorithm	Average Recall Rate $\bar{r}_{RR}$	Average Precision Rate $\bar{r}_{PR}$	Average F1 Score $\bar{s}_{\mathrm{F1}}$	Accuracy Rate $\bar{r}_{AR}$
KTH	Proposed ICSCD algorithm	0.9320	0.9388	0.9322	0.9325
	Mean shift-based algorithm	0.8645	0.9107	0.8491	0.8625
	Bayesian estimation-based algorithm	0.7817	0.8408	0.7759	0.7925
HIT-TUM	Proposed ICSCD algorithm	0.9324	0.9489	0.9375	0.9225
	Mean shift-based algorithm	0.8404	0.8485	0.8333	0.8475
	Bayesian estimation-based algorithm	0.7497	0.7692	0.7387	0.7550

5.2. Experimental Results of Hierarchical Image Retrieval and Visual Localization

In the proposed HCIR algorithm, the best-matched database image is determined by the maximum similarity method. Therefore, the validity of the method needs to be verified by experiments. In this part of the experiments, since the best-matched image is the database image that is most similar to the query image, the database image I_{GS} with the highest matching similarity to the query image is found by the global search, and the index of this database image is k_B^i. In addition, another best-matched database image I_{MS} is determined by the proposed maximum similarity method, and the index of the database image is k_M^i. The average error ε_{index} of the index positions of best-matched database images can be calculated by:

$$\varepsilon_{index} = \frac{1}{n_{TQ}} \sum_{i=1}^{n_{TQ}} k_{BM}^i = \frac{1}{n_{TQ}} \sum_{i=1}^{n_{TQ}} \left(\left| k_B^i - k_M^i \right| \right) \tag{44}$$

where k_{BM}^i is the index error of the best-matched database image in the i-th experiment, and n_{TQ} is the total number of query images for experiments.

Based on the average error ε_{index}, the average distance error between the best-matched database images I_{GS} and I_{MS} can be further defined by $\varepsilon_{dis} = \varepsilon_{index} \cdot d_D$, where d_D is the fixed acquisition distance of database images. The average index error ε_{index} and the average distance error ε_{dis} reflect the performance of retrieving the best-matched database images with the maximum similarity method. For the experimental results, the smaller values of ε_{dis} and ε_{index} indicate that the matched database images are closer to the query image. The results of matched image retrieval are shown in Table 3 under the condition that d_D is set to 10 cm. For the experimental results, the ratio r_S^j $(j = 0,\ 1,\ 2)$ is the percentage of the number of experimental results that satisfy $k_{BM}^i = j$. In addition, the average value $\bar{s}_L$ of the matching similarity and the average value $\bar{n}_{mat}$ of matched feature points are also calculated in experiments.

Table 3. Experimental results of matched database image retrieval.

Database	r_S^0	r_S^1	r_S^2	ε_{index}	ε_{dis}	$\bar{s}_L$	$\bar{n}_{mat}$
KTH	94.75%	4.50%	0.75%	0.06	0.60 cm	0.5453	121.45
HIT-TUM	96.50%	3.00%	0.50%	0.04	0.40 cm	0.6287	134.40

The experimental results shown in Table 3 indicate that for image retrieval experiments with the similarity maximum method separately conducted in the KTH database and the TUM-HIT database, the probability of successfully retrieving a matched database image (i.e., the situation of r_S^0) exceeds 94%. In other cases, although the similarity between the matched image and the query image cannot reach the maximum value, the index errors are less than 2, which indicates that the matched image is close to the query image, and there are enough matched features between the matching image and the query image. Therefore, the matched database images under the situation of r_S^0, r_S^1, and r_S^2 can be used for visual localization. For the two databases, the average error ε_{index} of the index positions is less

than 0.1, and the average distance error is less than 1 cm, showing the effectiveness of the similarity maximum method in determining matched database images. Moreover, the experimental results also show that for different databases, the average matching similarity between the query image and the best-matched database image is greater than 0.5, and there are more than 120 pairs of matched feature points between the query image and the best-matched database image, which provides a fundamental guarantee for visual localization.

In the proposed HCIR algorithm, the scene-level clustering results are sorted based on the global feature similarity, and then the database images in the sub-scene clusters are sorted based on the local feature similarity. After the two-stage sorting, the query image is matched with database images according to the sorting result. From the above process, it is known that when retrieving the scene-level clustering results based on the global feature similarity, the best case is to obtain the matched image in the first clustering result, and the worst situation is obtaining the matched images after all the results are retrieved. Therefore, for the scene-level image retrieval, the success rate of image retrieval within the top-K clusters is proposed in this paper to evaluate the performance of the clustering algorithm in image retrieval. Specifically, after scene-level clustering of database images, more than one image cluster can be obtained. If the matched database image can be retrieved after searching K image clusters, image retrieval is considered to be achieved within K database image clusters. For a total of n_Q query images, if there are n_K query images, and their matched database images are in the K-th cluster, the success rate of the top-K clusters is defined as $\left(100 \cdot n_K / n_Q\right)\%$. The success rate effectively reflects the impact of the scene-level clustering algorithm on the performance of image retrieval. The scene-level clustering algorithms of database images can be divided into two categories: one is based on the method of detecting change points of visual features (such as the proposed HCIR algorithm in this paper, the mean shift-based algorithm, and the Bayesian estimation-based algorithm), and another is clustering a fixed number of database images (such as the C-GIST algorithm [37]). In the C-GIST algorithm, five consecutive database images are grouped into one cluster, and the cluster center is a feature vector of the image that is located at the center position of each cluster. In this paper, two categories of image-clustering algorithms are simulated, respectively, and the success rate of the top-K clusters is calculated. The results are presented in Table 4.

Table 4. Success rates of the top-K clusters for query image.

Database	Algorithm	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
KTH	Proposed HCIR algorithm	66.75%	87.50%	93.50%	99.75%	100%
	Mean shift-based	50.25%	72.75%	84.25%	91.50%	97.75%
	Bayesian estimation-based	45.75%	64.75%	74.25%	82.75%	91.00%
	C-GIST	29.25%	35.25%	39.50%	43.25%	47.25%
HIT-TUM	Proposed HCIR algorithm	60.50%	82.50%	97.00%	99.00%	100%
	Mean shift-based	49.25%	63.25%	83.25%	90.25%	96.25%
	Bayesian estimation-based	43.00%	59.75%	76.50%	87.00%	93.25%
	C-GIST	29.75%	38.25%	49.75%	56.50%	61.75%

The results shown in Table 4 indicate that the proposed HCIR algorithm is beneficial in improving the success rate of the top-K clusters for a query image. For the two databases, the success rates of the top-five clusters achieved by the HCIR algorithm, mean shift-based algorithm, and the Bayesian estimation-based algorithm are more than 90%. At the same time, it is not difficult to find that the HCIR algorithm has more obvious performance advantages. Especially in the KTH database, the success rate of the first cluster is 66.75%, and the success rate of the top-five clusters reaches 99.75%. For the HIT-TUM database and the KTH database, the best-matched database image can be retrieved within the top-five clusters by the HCIR algorithm. In addition, for the sub-scene-level image clustering,

success rates of the first *K*-clusters are also calculated and recorded. Experimental results show that for the HCIR algorithm, the success rate of the first cluster is more than 88%, which indicates that in most cases, the best-matched database image can be found in the first sub-cluster.

To verify the image retrieval efficiency of the proposed HCIR algorithm, image retrieval experiments are performed on the HCIR algorithm and the comparison algorithms. In the experiments, the mean shift-based algorithm, the Bayesian estimation-based algorithm, and the C-GIST algorithm are single-layer clustering algorithms. In addition, another two multi-layer clustering algorithms are considered: the mean shift-KLT algorithm and the Bayesian estimation-KLT algorithm. For the two multi-layer clustering algorithms, database images are firstly grouped by the mean shift-based algorithm or the Bayesian estimation-based algorithm, and then the images are further grouped by the KLT algorithm. According to the average number of retrieved images shown in Table 5, multi-layer clustering algorithms have higher retrieval efficiency, and the number of similar comparisons (i.e., the processes of feature matching) can be limited to 10% of the database size. The reason is that database images are only grouped into scene-level clusters for the single-layer algorithms, and thus the query image needs to match with database images in the scene-level cluster one-by-one. In contrast, database images are further grouped on the basis of scene-level image clusters in multi-layer algorithms. Then, according to visual similarities, image clusters are ranked, and the query image preferentially matches with the database images in the most similar cluster. Therefore, multi-layer algorithms have a better performance at average numbers of retrieved database images.

Table 5. Average numbers of retrieved database images by different clustering algorithms.

Database	C-GIST	Bayesian Estimation	Mean Shift	Bayesian Estimation-KLT	Mean Shift-KLT	Proposed HCIR Algorithm
KTH	122.14	89.11	87.17	79.60	53.69	38.81
HIT-TUM	163.01	78.21	75.32	48.70	46.22	24.47

It can be observed from the experimental results shown in Table 5 that fewer database images are retrieved in the HCIR algorithm compared with the other two multi-layer algorithms. The reason is that the ICSCD algorithm has a better performance at scene-level clustering (as shown in Table 2), so that the cluster center can better express the global features of the images in the cluster.

Table 6 shows the average running time of the image-retrieval system using different clustering algorithms. By comparing the number of retrieved images with the average running time of image retrieval, it can be known that when there are more retrieved database images, the running time consumed by image retrieval is also more. Experimental results shown in Tables 5 and 6 indicate that more database images are retrieved in single-layer clustering algorithms, leading to larger time overheads than in multi-layer clustering algorithms. It is obvious that the HCIR algorithm has advantages in terms of the number of retrieved database images and the running time of image retrieval. The reason is that multi-layer clustering on database images is employed in the HCIR algorithm, and more importantly, the ICSCD algorithm is employed in the proposed retrieval algorithm that achieves a better performance in scene-level database image clustering.

Table 6. Average running time of image retrieval by different clustering algorithms (unit: ms).

Database	C-GIST	Bayesian Estimation	Mean Shift	Bayesian Estimation-KLT	Mean Shift-KLT	Proposed HCIR Algorithm
KTH	131.7622	96.1804	94.6706	91.9858	68.1689	50.5838
HIT-TUM	177.6345	92.8804	89.9179	57.4065	56.0274	37.4582

Table 7 shows the average running time of different stages in image retrieval. In the practical implementation of image retrieval, hundreds of local features are needed to be matched between the query image and the database image, resulting in the most time consumption appearing at this stage.

Table 7. Average running time of different stages in image retrieval (unit: ms).

Extracting Global Features	Extracting Local Features	Measuring Feature Distance	Matching Local Feature	Sorting Feature Vectors
0.7648	3.2247	0.1131	0.9437	0.0169

To reveal the performance difference between the single-layer clustering algorithm and the proposed HCIR algorithm, there are ten indoor scenes used for simulation, and m_{L_1} is separately set as 100, 200, 400, 800, and 1000, and then the running time of image retrieval for different database sizes can be simulated, as shown in Figure 11. As a precondition of simulation, the backtracking mechanism is always not triggered. According to the simulation results, the advantage of the proposed algorithm is that the running time does not linearly increase along with the growth of the database size. Even when the database image size is increased to 10,000, the running time of image retrieval is less than 110 ms. In this case, the running time of image retrieval corresponding to the single-layer clustering-based algorithm almost reaches 500 ms, which means that only two retrievals can be performed per second. In contrast, by the proposed HCIR algorithm, image retrieval can be executed nine times when there are 10,000 images in the database. The reason that the proposed algorithm spends less time coping with image retrieval is that database images are reasonably grouped in the off-line stage. Furthermore, on a deeper level, time for image clustering is sacrificed in the off-line stage to reduce the time consumption of image retrieval in the on-line stage.

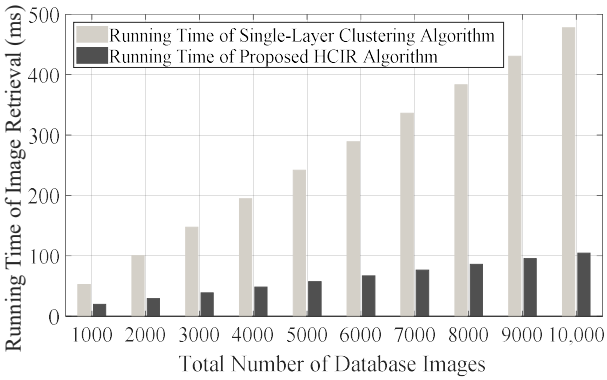

Figure 11. Average running time of the single-layer clustering algorithm and proposed algorithm.

To demonstrate the performance of the proposed WKNN algorithm, two typical image retrieval-based localization methods (i.e., the NN method [49,54,55] and the KNN method [56,57]) are selected and implemented. Each image in the KTH database and the HIT-TUM database is employed as a query image for visual localization. For the proposed WKNN method and the typical KNN method, five nearest neighbors are selected to estimate the query position [57]. In order to reveal the impact of image acquisition intervals (d_{in}) on localization accuracy, database images with different acquisition intervals are set for experiments. Localization errors of query images are calculated, and the cumulative distributions of the errors are shown in Figure 12.

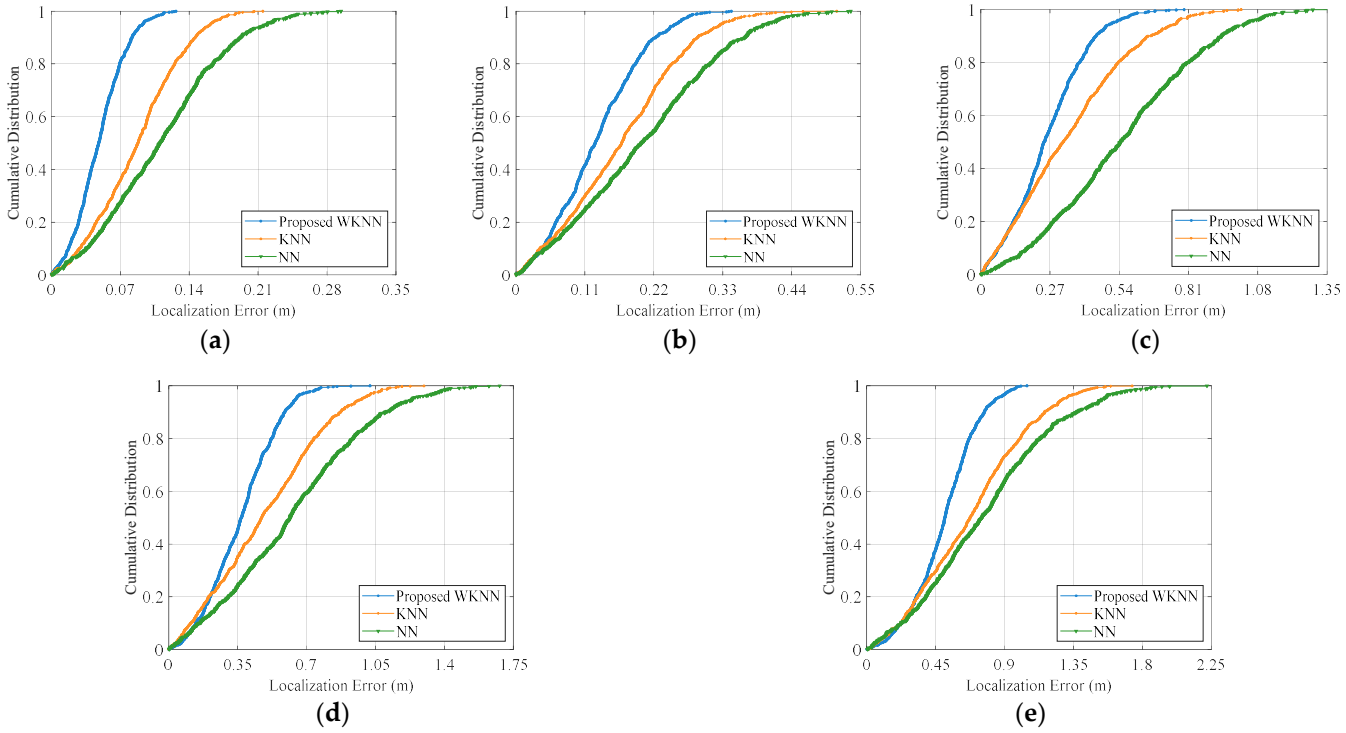

Figure 12. Experimental results of visual localization. (**a**) Cumulative distribution of localization errors with d_{in} = 10 cm; (**b**) cumulative distribution of localization errors with d_{in} = 20 cm; (**c**) cumulative distribution of localization errors with d_{in} = 30 cm; (**d**) cumulative distribution of localization errors with d_{in} = 40 cm; (**e**) cumulative distribution of localization errors with d_{in} = 50 cm.

To quantitatively analyze the performance improvement of the WKNN method, an accuracy improvement rate r_{im} is introduced and defined as:

$$r_{im} = \left(\left|e_p - e_c\right| / e_c\right) \cdot 100\% \tag{45}$$

where e_p and e_c are the average errors by the proposed WKNN method and the comparative method, respectively.

Compared with the NN and KNN methods, the proposed WKNN method achieves a better performance on localization accuracy, as shown in Table 8. In all experimental cases, the improvement of average localization accuracy reaches at least 22% and 34%, respectively, compared with the KNN and NN methods. From the localization results, it can be found that when the database images are more densely captured, the advantage of the proposed method in terms of localization accuracy is more obvious compared with the two other localization methods. The reason is that when the intervals of database images are large, the common visual features between the query image and the database images are few, which weakens the contributions of the weights in the WKNN method.

As illustrated in Table 8, when the database image acquisition intervals are set to be 10 cm, 20 cm, 30 cm, 40 cm, and 50 cm, the average localization errors of the WKNN method are 0.0490 m, 0.1299 m, 0.2604 m, 0.3673 m, and 0.5048 m, respectively. The results indicate that localization accuracy increases along with database image acquisition intervals. Even if the acquisition interval is increased to 50 cm, the sub-meter localization accuracy can be achieved by the proposed method, which satisfies the requirements of most indoor location-based services. But it is worth noting that acquisition intervals that are too small lead to a large off-line image database and result further in a high time overhead of image retrieval. Therefore, when designing a visual indoor localization system, a proper database image acquisition interval should be selected by striking a balance between localization accuracy and efficiency.

Table 8. Localization performance of various localization methods.

Image Acquisition Intervals	Evaluation Criterions	WKNN Method	KNN Method	NN Method
d_{in} = 10 cm	Average Errors (m)	0.0490	0.0874	0.1126
	Maximum Errors (m)	0.1266	0.2149	0.2942
	Improvement Rates (%)	-	43.9022	56.4623
d_{in} = 20 cm	Average Errors (m)	0.1299	0.1689	0.2042
	Maximum Errors (m)	0.3439	0.5118	0.5342
	Improvement Rates (%)	-	23.11014	36.4159
d_{in} = 30 cm	Average Errors (m)	0.2604	0.3446	0.5543
	Maximum Errors (m)	0.7928	1.0151	1.3583
	Improvement Rates (%)	-	24.4261	53.0155
d_{in} = 40 cm	Average Errors (m)	0.3673	0.4883	0.6289
	Maximum Errors (m)	1.0218	1.2965	1.6804
	Improvement Rates (%)	-	24.7678	41.5920
d_{in} = 50 cm	Average Errors (m)	0.5048	0.6504	0.7732
	Maximum Errors (m)	1.0462	1.6518	2.2218
	Improvement Rates (%)	-	22.3886	34.7127

6. Discussion

In the visual localization system, an off-line database generally contains a large number of images for position estimation. For example, over 40,000 database images were captured over a distance of 4.5 km in the TUMindoor localization system, which means that database images were acquired at approximately 10 cm intervals [26]. With the traditional image retrieval strategy, the query image is exhaustively compared with each database image, which is not scalable for a large-scale database. Recently, clustering-based hierarchical image retrieval has been proposed and applied in large-scale image retrieval [63–65]. The main advantage of hierarchical image retrieval is creating an indexing strategy by grouping the images based on the visual cues before retrieval, so that only the relevant clusters are examined in the retrieval process. With this strategy, clustering-based hierarchical image retrieval significantly speeds up the search process at the expense of the time consumption of image clustering beforehand.

However, although the existing works on hierarchical image retrieval achieve high searching efficiency, these works are unsuitable for visual localization. The reason is that geographic factors on image clustering have not been taken into consideration, and database images acquired in the same scene are not necessarily grouped in a cluster. In the visual localization system, the query image is desired to be orderly compared with database images in the relevant scenes according to visual similarity. Specifically, the query image should be compared with similar database images as a priority. In this way, the query image need not be compared with all database images, and the retrieval can be obtained.

Considering the particular requirements of image retrieval in visual localization, a hierarchical clustering-based image retrieval (i.e., HCIR) algorithm is proposed in this paper to organize database images and achieve image retrieval. The main contribution to the HCIR method is that database images are orderly grouped into clusters by visual cues according with geographical distribution characteristics. Since the database images for visual localization are successively captured by the mapping equipment in indoor scenes, visual features in the same scene have high visual correlations. However, once the mapping equipment switches to another scene, the correlations subsequently decrease. Taking advantage of this characteristic of database images, an ICSCD algorithm is presented to group database images into clusters at the scene level. Moreover, an image clustering

algorithm based on KLT feature tracking is proposed to group database images at the sub-scene level. With the ICSCD algorithm and KLT feature tracking-based algorithms, the visual features of different scenes and sub-scenes can be described by the cluster centers. In the process of retrieval, the query image is initially compared with the cluster centers, and then the clusters that have the largest similarity with the query are selected, and the images in these clusters are used to compare the query. By this means, the database images with high similarities to the query are preferentially retrieved, thus reducing the time consumption of image retrieval.

Compared with the existing change-point detection algorithms (i.e., the mean shift-based algorithm [36] and the Bayesian estimation-based algorithm [62]), the proposed ICSCD algorithm achieves a better performance in terms of scene-level clustering on different evaluation criteria, such as the average recall rate, the average precision rate, the average F1 score, and the accuracy rate. The reason is that the typical CUSUM algorithm is improved at targeting change-point detection on database image sequences without threshold selection. In addition, the utilization of appropriate global visual features and the definition of the change-point detection function also contribute to raising cluster accuracy. On the basis of scene-level clustering results, the KLT feature tracking-based clustering algorithm is employed to further group database images, and then a multi-layer search tree can be generated for image retrieval. A distinguishing characteristic of the search tree is that database image clusters attached to the tree are determined by geo-information and visual cues. The query image is initially compared with cluster centers of the scenes with high similarity, which boosts the retrieval efficiency. The experimental results indicate that the image retrieval by our multi-layer search tree is more efficient compared with other single-layer and multi-layer clustering algorithms, such as the mean shift-based [36], C-Gist [37], Bayesian estimation-based [62], Bayesian estimation-KLT, and mean shift-KLT algorithms. Multi-layer clustering algorithms outperform single-layer algorithms because the KLT feature tracking-based algorithm subdivides the scene-level clusters and groups images at the sub-scene level, resulting in reducing the comparison between the query and database images. Due to the higher accuracy of the ICSCD algorithm on scene-level image clustering, the proposed HCIR algorithm outperforms other multi-layer clustering algorithms on retrieval efficiency, as shown in Tables 5 and 6.

Visual localization without camera-intrinsic parameters is essential in indoor navigation and augmented reality. Existing works mainly focus on the NN method [49,54,55] and the KNN method [56,57] to solve the estimated position of the query. However, the effect of similarity between the query image and database images on localization is not taken into consideration either in the NN method or in the KNN method. Therefore, a weighted KNN method is proposed, and the Armijo–Goldstein algorithm is employed to calculate the position of the query camera. The highlight of the weighted KNN method is that a novel localization error function is defined in consideration of visual similarity. Specifically, the matched database image with a higher similarity to the query is assigned a larger weight in the localization error function, resulting in the estimated position being close to similar database images. As discussed earlier, the drawback of the NN method is that the position of the most similar database image is assigned to the query, but the database image may be far from the query. For the KNN method, each neighbor database image has the same weight, which is not in accordance with the actual localization situation. The weighted KNN method overcomes these drawbacks by similarity comparison, and neighbor database images with higher similarity are given a larger weight. Experimental results show that the weighted KNN method achieves a better performance on localization accuracy compared with the NN and KNN methods. For a typical interval of database image acquisition (i.e., the interval is approximately 10 cm [26]), the average localization errors are limited to 5 cm, and this outperforms the NN and KNN methods. The proposed WKNN localization method has the potential to be embedded in visually impaired navigation systems and shopping guide applications.

7. Conclusions

In this paper, a hierarchical clustering-based image retrieval algorithm is presented, in which database images are grouped by improved cumulative sum change-point detection and KLT feature tracking. Taking advantage of hierarchical clusters, database images that are similar to the query image have a priority to match with the query, which effectively reduces the time consumption of image retrieval. For different indoor image databases, the number of similar comparisons can be limited to 10% of the database size. Simulation experiments also indicate that the running time of image retrieval does not linearly increase with the size of image databases. Compared with other single-layer and multi-layer clustering-based image retrieval algorithms, the proposed HCIR algorithm executes less similar comparisons and acquires low time overheads. Under the premise of ensuring image retrieval accuracy, the proposed visual localization system achieves high operational efficiency. With the proposed localization method, the position estimation error is limited to 5 cm when the database image acquisition intervals are set to 10 cm. Future work will focus on improving the accuracy of visual localization, especially reducing the impact of illumination changes on localization.

Author Contributions: Conceptualization, G.F. and X.T.; methodology, G.F.; software, G.F.; validation, G.F., Z.J. and X.T.; formal analysis, G.F.; investigation, G.F.; resources, G.F.; data curation, G.F.; writing—G.F.; writing—review and editing, G.F. and F.C.; visualization, G.F.; supervision, X.T.; project administration, G.F., X.T. and Z.J.; funding acquisition, G.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Jilin Province, China [Grant No. 20210101170JC].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zangenehnejad, F.; Gao, Y. GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. *Sat. Nav.* **2021**, *2*, 24. [CrossRef] [PubMed]
2. Zidan, J.; Adegoke, E.I.; Kampert, E.; Birrell, S.A.; Ford, C.R.; Higgins, M.D. GNSS vulnerabilities and existing solutions: A review of the literature. *IEEE Access* **2020**, *9*, 153960–153976. [CrossRef]
3. Zhu, N.; Marais, J.; Bétaille, D.; Berbineau, M. GNSS position integrity in urban environments: A review of literature. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2762–2778. [CrossRef]
4. Guo, L.; Wang, F.; Sang, J.; Lin, X.; Gong, X.; Zhang, W. Characteristics analysis of raw multi-GNSS measurement from Xiaomi Mi 8 and positioning performance improvement with L5/E5 frequency in an urban environment. *Remote Sens.* **2020**, *12*, 744. [CrossRef]
5. Feng, X.; Nguyen, K.A.; Luo, Z. A survey of deep learning approaches for WiFi-based indoor positioning. *J. Inf. Telecommun.* **2022**, *6*, 163–216. [CrossRef]
6. Mendoza-Silva, G.M.; Costa, A.C.; Torres-Sospedra, J.; Painho, M.; Huerta, J. Environment-aware regression for indoor localization based on WiFi fingerprinting. *IEEE Sens. J.* **2021**, *22*, 4978–4988. [CrossRef]
7. Bencak, P.; Hercog, D.; Lerher, T. Indoor Positioning System Based on Bluetooth Low Energy Technology and a Nature-Inspired Optimization Algorithm. *Electronics* **2022**, *11*, 308. [CrossRef]
8. Lie, M.M.K.; Kusuma, G.P. A fingerprint-based coarse-to-fine algorithm for indoor positioning system using Bluetooth Low Energy. *Neural. Comput. Appl.* **2021**, *33*, 2735–2751. [CrossRef]
9. Guo, H.; Li, M.; Zhang, X.; Gao, X.; Liu, Q. UWB indoor positioning optimization algorithm based on genetic annealing and clustering analysis. *Front. Neurorobot.* **2022**, *16*, 715440. [CrossRef]
10. Wang, J.; Wang, M.; Yang, D.; Liu, F.; Wen, Z. UWB positioning algorithm and accuracy evaluation for different indoor scenes. *Int. J. Image Data Fusion* **2021**, *12*, 203–225. [CrossRef]
11. Zhang, X.; Kuang, Y.; Yang, H.; Lu, H.; Yang, Y. UWB Indoor Localization Algorithm Using Firefly of Multistage Optimization on Particle Filter. *J. Sens.* **2021**, *2021*, 1383767. [CrossRef]
12. Zhang, W.; Hua, X.; Yu, K.; Qiu, W.; Zhang, S.; He, X. A novel WiFi indoor positioning strategy based on weighted squared Euclidean distance and local principal gradient direction. *Sens. Rev.* **2019**, *39*, 99–106. [CrossRef]
13. Huang, B.; Yang, R.; Jia, B.; Li, W.; Mao, G. Accurate WiFi localization by fusing a group of fingerprints via a global fusion profile. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3599–3608. [CrossRef]
14. Lu, C.; Uchiyama, H.; Thomas, D.; Shimada, A.; Taniguchi, R. Indoor positioning system based on chest-mounted IMU. *Sensors* **2019**, *19*, 420. [CrossRef] [PubMed]

15. Zhang, Y.; Guo, J.; Wang, F.; Zhu, R.; Wang, L. An Indoor Localization Method Based on the Combination of Indoor Map Information and Inertial Navigation with Cascade Filter. *J. Sens.* **2021**, *2021*, 7621393. [CrossRef]
16. Sun, M.; Wang, Y.; Joseph, W.; Plets, D. Indoor localization using mind evolutionary algorithm-based geomagnetic positioning and smartphone IMU sensors. *IEEE Sens. J.* **2022**, *22*, 7130–7141. [CrossRef]
17. Guo, G.; Chen, R.; Ye, F.; Liu, Z.; Xu, S.; Huang, L.; Li, Z.; Qian, L. A robust integration platform of Wi-Fi RTT, RSS signal and MEMS-IMU for locating commercial smartphone indoors. *IEEE Internet Things J.* **2022**, *9*, 16322–16331. [CrossRef]
18. Krishnaveni, B.V.; Reddy, K.S.; Reddy, P.R. Indoor tracking by adding IMU and UWB using Unscented Kalman filter. *Wirel. Pers. Commun.* **2022**, *123*, 3575–3596. [CrossRef]
19. Feng, G.; Ma, L.; Tan, X.; Qin, D. Drift-aware monocular localization based on a pre-constructed dense 3D map in indoor environments. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 299. [CrossRef]
20. Piciarelli, C. Visual indoor localization in known environments. *IEEE Signal Process. Lett.* **2016**, *23*, 1330–1334. [CrossRef]
21. Li, X.; Zhao, L.; Ji, W.; Wu, Y.; Wu, F.; Yang, M.H.; Tao, D.; Reid, I. Multi-task structure-aware context modeling for robust keypoint-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 915–927. [CrossRef]
22. Zhang, H.; Hong, X. Recent progresses on object detection: A brief review. *Multimed. Tools Appl.* **2019**, *78*, 27809–27847. [CrossRef]
23. Schmarje, L.; Santarossa, M.; Schröder, S.M.; Koch, R. A survey on semi-, self-and unsupervised learning for image classification. *IEEE Access* **2019**, *9*, 82146–82168. [CrossRef]
24. Wang, X.; Zheng, Z.; He, Y.; Yan, F.; Zeng, Z.; Yang, Y. Progressive local filter pruning for image retrieval acceleration. *arXiv* **2020**, arXiv:2001.08878.
25. Sadeghi, H. Image-Based Localization for Mobile and Vehicular Applications. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2017.
26. Huitl, R.; Schroth, G.; Hilsenbeck, S.; Schweiger, F.; Steinbach, E. TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In Proceedings of the 19th IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012.
27. Ge, R. Real-Time Visual Localization System in Changing and Challenging Environments via Visual Place Recognition. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2022.
28. Humenberger, M.; Cabon, Y.; Pion, N.; Weinzaepfel, P.; Lee, D.; Guérin, N.; Sattler, T.; Csurka, G. Investigating the role of image retrieval for visual localization. *Int. J. Comput. Vis.* **2022**, *130*, 1811–1836. [CrossRef]
29. Yu, L.; Fu, X.; Xu, H.; Fei, S. High-precision camera pose estimation and optimization in a large-scene 3D reconstruction system. *Meas. Sci. Technol.* **2020**, *31*, 085401. [CrossRef]
30. Feng, G.; Ma, L.; Tan, X. Visual map construction using RGB-D sensors for image-based localization in indoor environments. *J. Sens.* **2017**, *2017*, 8037607. [CrossRef]
31. Deretey, E.; Ahmed, M.T.; Marshall, J.A.; Greenspan, M. Visual indoor positioning with a single camera using PnP. In Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation, Banff, AB, Canada, 13–16 October 2015.
32. Wang, J.; Zha, H.; Cipolla, R. Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2006**, *36*, 413–422. [CrossRef]
33. Anandh, A.; Mala, K.; Suganya, S. Content based image retrieval system based on semantic information using color, texture and shape features. In Proceedings of the IEEE International Conference on Computing Technologies and Intelligent Data Engineering, Kovilpatti, India, 7–9 January 2016.
34. Singh, S.R.; Kohli, S. Enhanced CBIR using color moments HSV histogram color auto correlogram and Gabor texture. *Int. J. Comput. Syst.* **2015**, *2*, 161–165.
35. Oliva, A.; Torralba, A. Building the gist of a scene: The role of global image features in recognition. *Prog. Brain Res.* **2006**, *155*, 23–36.
36. Chapoulie, A.; Rives, P.; Filliat, D. Topological segmentation of indoors/outdoors sequences of spherical views. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012.
37. Ma, L.; Xue, H.; Jia, T.; Tan, X. A fast C-GIST based image retrieval method for vision-based indoor localization. In Proceedings of the IEEE Vehicular Technology Conference, Sydney, NSW, Australia, 4–7 June 2017.
38. Bay, H.; Tuytelaars, T.; Van, G.L. SURF: Speeded Up Robust Features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006.
39. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the 7th IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999.
40. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2012.
41. Ni, J.; Wang, X.; Gong, T.; Xie, Y. An improved adaptive ORB-SLAM method for monocular vision robot under dynamic environments. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 3821–3836. [CrossRef]
42. Chhabra, P.; Garg, N.K.; Kumar, M. Content-based image retrieval system using ORB and SIFT features. *Neural. Comput. Appl.* **2020**, *32*, 2725–2733. [CrossRef]
43. Bel, K.N.S.; Sam, I.S. Encrypted image retrieval method using SIFT and ORB in cloud. In Proceedings of the 7th IEEE International Conference on Smart Structures and Systems, Chennai, India, 23–24 July 2020.

44. Sadeghi, H.; Valaee, S.; Shirani, S. 2DTriPnP: A robust two-dimensional method for fine visual localization using Google streetview database. *IEEE Trans. Veh. Technol.* **2017**, *66*, 4678–4690. [CrossRef]
45. Naseer, T.; Burgard, W.; Stachniss, C. Robust visual localization across seasons. *IEEE Trans. Robot.* **2018**, *34*, 289–302. [CrossRef]
46. Boin, J.B.; Bobkov, D.; Steinbach, E.; Girod, B. Efficient panorama database indexing for indoor localization. In Proceedings of the 2019 IEEE International Conference on Content-Based Multimedia Indexing, Dublin, Ireland, 4–6 September 2019.
47. Lanir, J.; Kuflik, T.; Wecker, A.J.; Stock, O.; Zancanaro, M. Examining proactiveness and choice in a location-aware mobile museum guide. *Interact. Comput.* **2011**, *23*, 513–524. [CrossRef]
48. Tseng, P.Y.; Lin, J.J.; Chan, Y.C.; Chen, A.Y. Real-time indoor localization with visual SLAM for in-building emergency response. *Interact. Comput.* **2022**, *140*, 104319. [CrossRef]
49. Zhang, W.; Kosecka, J. Image based localization in urban environments. In Proceedings of the 3rd IEEE International Symposium on 3D Data Processing, Visualization, and Transmission, Chapel Hill, NC, USA, 14–16 June 2006.
50. Vedadi, F.; Valaee, S. Automatic visual fingerprinting for indoor image-based localization applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *50*, 305–317. [CrossRef]
51. Taira, H.; Okutomi, M.; Sattler, T.; Cimpoi, M.; Pollefeys, M.; Sivic, J.; Pajdla, T.; Torii, A. InLoc: Indoor visual localization with dense matching and view synthesis. *IEEE Trans. Pattern Anal.* **2019**, *43*, 1293–1307. [CrossRef]
52. Zhang, Z.; Sattler, T.; Scaramuzza, D. Reference pose generation for long-term visual localization via learned features and view synthesis. *Int. J. Comput. Vis.* **2021**, *129*, 821–844. [CrossRef]
53. Li, N.; Ai, H. EfiLoc: Large-scale visual indoor localization with efficient correlation between sparse features and 3D points. *Vis. Comput.* **2022**, *38*, 2091–2106. [CrossRef]
54. Van, O.D.; Schroth, G.; Huitl, R.; Hilsenbeck, S.; Garcea, A.; Steinbach, E. Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In Proceedings of the IEEE International Conference on Image Processing, Paris, France, 27–30 October 2014.
55. Spera, E.; Furnari, A.; Battiato, S.; Farinella, G.M. Performance comparison of methods based on image retrieval and direct regression for egocentric shopping cart localization. In Proceedings of the IEEE 4th International Forum on Research and Technology for Society and Industry, Palermo, Italy, 10–13 September 2018.
56. Spera, E.; Furnari, A.; Battiato, S.; Farinella, G.M. EgoCart: A benchmark dataset for large-scale indoor image-based localization in retail stores. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *31*, 1253–1267. [CrossRef]
57. He, R.; Wang, Y.; Tao, Q.; Cai, J.; Duan, L. Efficient image retrieval based mobile indoor localization. In Proceedings of the IEEE Visual Communications and Image Processing, Singapore, 13–16 December 2015.
58. Peng, X.; Chen, R.; Yu, K.; Guo, G.; Ye, F.; Xue, W. A new Wi-Fi dynamic selection of nearest neighbor localization algorithm based on RSS characteristic value extraction by hybrid filtering. *Meas. Sci. Technol.* **2021**, *32*, 034003. [CrossRef]
59. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer: New York, NY, USA, 2006; pp. 650–656.
60. Sun, W.Y.; Yuan, Y.X. *Optimization Theory and Methods: Nonlinear Programming*; Springer: New York, NY, USA, 2005; pp. 103–104.
61. Luo, J.; Pronobis, A.; Caputo, B.; Jensfelt, P. Incremental learning for place recognition in dynamic environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007.
62. Ranganathan, A. PLISS: Labeling places using online changepoint detection. *Auton. Robot.* **2012**, *32*, 351–368. [CrossRef]
63. Yan, C.; Bai, X.; Zhou, J.; Liu, Y. Hierarchical hashing for image retrieval. In Proceedings of the CCF Chinese Conference on Computer Vision, Tianjin, China, 11–14 October 2017.
64. Munoz, J.V.; Gonçalves, M.A.; Dias, Z.; Torres, R.D.S. Hierarchical clustering-based graphs for large scale approximate nearest neighbor search. *Pattern Recognit.* **2019**, *96*, 106970. [CrossRef]
65. Xie, H.; Chen, W.; Wang, J. Hierarchical forest based fast online loop closure for low-latency consistent visual-inertial SLAM. *Robot. Auton. Syst.* **2022**, *151*, 104035. [CrossRef]