Green Buildings and Materials
Volume 1 Issue 1 No. 2
Published Online July 10 2022

An Efficient way for indoor localization with content-based image retrieval algorithm

Yuan Xiao, Shengwu Xiong, Li Li
School of Computer Science & Technology, Wuhan University of Technology, Wuhan, China

Abstract: With the increasing demand of navigation technology, indoor localization technology has been getting more and more important. Traditional approaches often based on WiFi localization with high cost and cannot be promoted massively. In this paper, we propose a method for the indoor wireless localization based on image retrieval, with high accuracy rate of localization but lower cost. We use Scale-Invariant Feature Transform (SIFT) operator for image feature extraction specifically, and choose the Fast Library for Approximate Nearest Neighbors (FLANN) algorithm to search the target image according to features of the sample image, so as to correct the user's real-time position. The experiment result shows that the position accuracy of the proposed algorithm is up to 85.7%, proving that it's an efficient way for indoor localization.

KeyWords: Content-based Image Retrieval, Indoor Localization, Scale-Invariant Feature Transform, Fast Library for Approximate Nearest Neighbors

1. Introduction

At present, there are many wireless localization methods, and each has its own scope of application. Global Position System (GPS) performs well outdoors, while it can't cover the indoor environment because of weak signal strength and poor availability due to construction blocking. The cellular network is very effective in the sight distance, once in the non-line sight, the accuracy of measured signal and environmental impacts generates poor results [1,2]. The WiFi method can make full use of the existing hardware, but it needs the support of an access point (AP) [3]. Moreover, its calculation complexity is unbearable when the number of AP grows. Other indoor localization methods, such as Infrared, Bluetooth, ZigBee, demand for indoor position hardware which incur high costs, can't be used for wide area applications. There are non-wireless localization methods, including computer vision and inertial localization. The former is still in the early stage of research, while the latter's time error leads to serious localization loss [4].

Google Maps for mobile has joined the indoor navigation in some areas, this scheme mainly relies on the GPS, WiFi, mobile phone base stations to realize the indoor localization, but the accuracy wasn't fairly satisfactory [7]. Google later released a mobile phone application named "Google Maps floor plan marker", calling for the user to improve indoor navigation accuracy through certain steps. Nevertheless, it's difficult to get a universal application. Figure 1 shows the location accuracy and application difficulty comparison among different indoor localization. We can find that CBIR can achieve an acceptable accuracy with easy application difficulty.

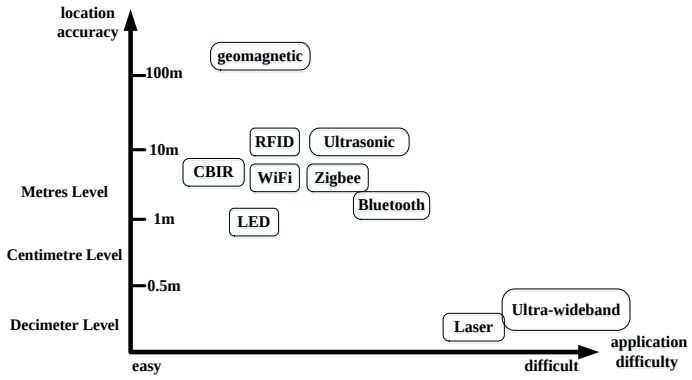


Figure 1. Location accuracy and application difficulty comparison among different indoor localization

With the mass popularity of digital cameras and high-pixel camera phones, image processing can provide a good solution for indoor localization. After more than 40 years development, the image retrieval technology gradually formed two aspects: One is the Text-based Image Retrieval (TBIR) [8] which is the earliest retrieval method. It applies mature search text to image, and uses the human description information to carry on the retrieval. The other one is Content-based Image Retrieval (CBIR) [9][10] which uses the image features for retrieval. First, it uses the computer to extract the feature information of images, such as color, edge, shape and other feature, and then matches them with the image feature database to find out the desired one.

TBIR needs to add keywords and descriptions manually for each image, which is a huge workload. Besides, people's descriptions are subjective, the added information may inadequately describe image content, which limits its application prospect. CBIR, on the other hand, is mainly based on image characteristics, and has reduced human intervention. It is more suitable for the actual retrieval needs. Thus, this paper proposes a new indoor position way based on CBIR that can get a rough location with a better position efficiency.

In the remaining of this paper, we briefly introduce the framework of the indoor position model based on CBIR in section two. A feature descriptor called SIFT and how do we use it to extract and describe the image feature is shown in section three. In section four, the construction process of kd-tree and how does it apply to FLANN to search image feature are described in detail. The experiment process is introduced and the result of the image retrieval is analyzed in section five. In the end, a brief conclusion of this paper and the future work to improve the accuracy of indoor localization is given in section six.

2. Indoor positioning model

CBIR uses the underlying image characteristics to achieve image retrieval, which is more efficient than the traditional text retrieval. In this paper, there is a large number of image data collected in the experiment scene, and the image contains rich feature information, so we also make use of the image characteristics to achieve indoor positioning based on CBIR.

The process of standard CBIR is shown in Fig.2. The feature information of the image sets is stored in the image feature database after feature extraction. The image retrieval process is as follows: we can get the feature vector of the sample image by using the same feature extraction algorithm as the one for image sets feature extraction, then we can get the output image with the highest similarity after image matching.

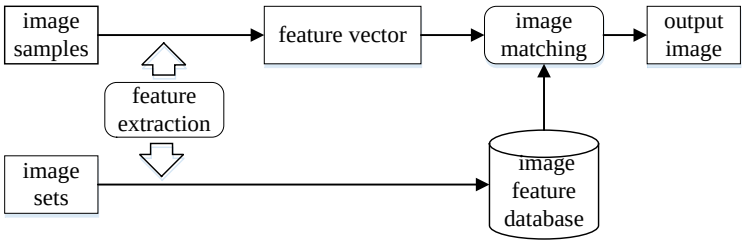


Figure 2. The process of standard CBIR

We proposed an efficient way for indoor localization based on CBIR. The framework of the model is shown in Figure 3. As the figure shows, every image in the image database contains its spatial location information. The image which is collected by the camera in real-time matches with the image database to get the target image, and then the user's current position can be adjusted according to the position information of target image.

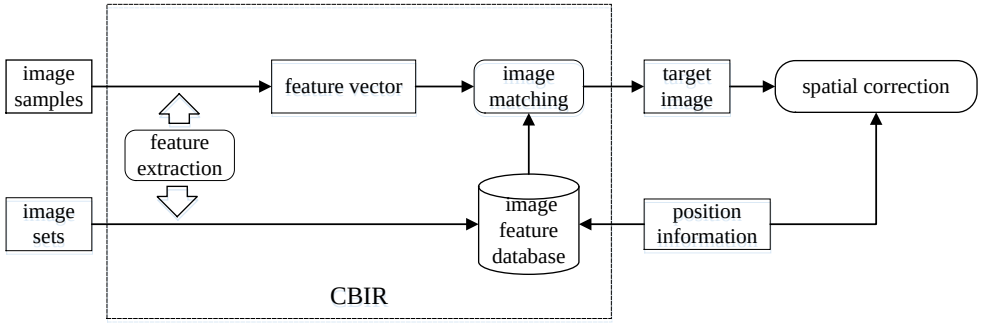


Figure 3. Image matching and image location information

The key to CBIR is how to carry out the feature extraction and feature matching. For the paper, the Scale-Invariant Feature Transform (SIFT) [11] operator was used to extract the image features, and the Fast Library for Approximate Nearest Neighbors (FLANN) [12] algorithm was used to deal with feature matching.

3. Feature extraction

Image feature point is an important indicator of image feature, and it's important to extract these points accurately for image matching. There are two main methods for image feature point extraction: SIFT, SURF [13]. Table 1 shows the differences between SIFT and SURF. We can see that the SIFT operator has high stability for angle transformation, affine transformation and noise, while the SURF has much better adaptability for illumination variation.

Table 1. Differences between SIFT and SURF

	Advantages	Disadvantages
SIFT	High stability for angle transformation, affine transformation and noise	● Week stability for non-rigid transformation ● Large calculation amount
SURF	● High calculate speed ● High adaptability for illumination variation	● Week stability for non-rigid transformation ● Week adaptability for noise and grayscale change

In this paper, we choose the SIFT operator for two reasons: (1) most of the experimental data was from indoors, there is little change in the illumination condition, (2) it is necessary to retain more image information for improving the accuracy of image match.

SIFT extracts features according to the image points and its related scale, rotation and other description information, and it can deal with the problem of different image scale very well. The implementation process of SIFT operator can be divided into the following two steps.

3.1 Determine the main direction of image feature points

The SIFT operator can deal with the rotated images by recording the main direction of each feature point to ensure the rotation invariance. The direction of the feature points is determined by using the distance and direction relation between the feature points and the pixel points in its adjacent range, which is to calculate the gradient information.

The formula for the calculation of the gradient value of pixel point (x,y) is as the following formula shows:

$$m(x,y) = \sqrt{\left(L(x+1,y) - L(x-1,y)\right)^2 + \left(L(x,y+1) - L(x,y-1)\right)^2} \quad (1)$$

L represents the spatial scale value.

The formula for the calculation of the gradient direction is as follows:

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right) \quad (2)$$

After calculating the gradient values of many pixel points adjacent to the feature point, it needs to simplify the expression of feature points in the direction. Using 10 ° as the interval, a total of 36 intervals were used to calculate the statistical value of gradient direction, the main direction of the feature points determined by the interval value of the largest statistical value. Figure 4 shows the main direction determining process (the figure contains only eight directions for simplification).

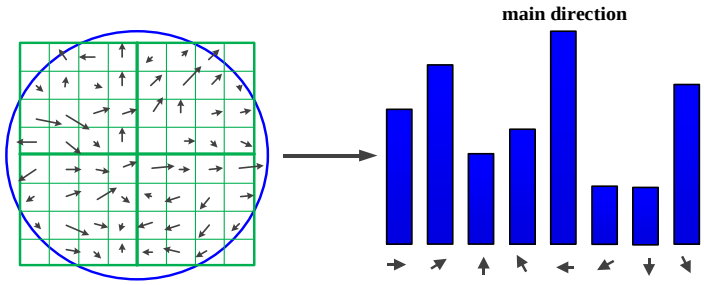


Figure 4. Determining the main direction [14]

3.2 Generate feature descriptor

To ensure the rotation invariance of the feature points, we need to rotate the coordinate axis, so that the X axis and the feature points keep the same direction. Then, select a 16*16 sampling area in feature point neighborhood, which is composed of 16 small areas with a size of 4*4. Calculated for each pixel in the small region, with respect to the feature points of the gradient value and direction, it's necessary to describe the feature points. Due to excessive direction information, we divided the statistics 45° intervals, and then get the histogram distribution of each small region in eight directions, forming 16 small regions. Take the 8*8 neighborhood for an example, the generation process of the feature descriptor is shown in Fig.5:

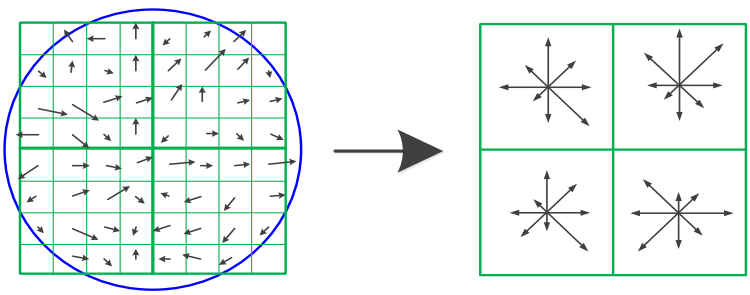


Figure 5. The generation process of the feature descriptor [14]

As the figure shows, the center of the left figure is the feature point. Each small square represents a pixel point and the direction of the arrow represents the pixel's direction relative to the feature point. The length of the arrow represents the gradient value. The circle area is a Gaussian weighted range. The small square in the right represents the direction of statistics in eight directions. So the SIFT descriptor is the total of 16 * 8, which is a 128-dimension vector, and it can adapt to the light effects after normalized.

4. Image feature matching

When using the sample image to match with the image database, it is necessary to calculate the high dimensional feature point vector, which is to find the target feature points that are similar to the sample image feature points in the feature point database.

We use FLANN algorithm to search the target image by matching the sample image feature with the feature of the image set. FLANN uses the kd-Tree to store the feature vectors, and uses the nearest neighbor search algorithm to search features.

In a high dimensional space, kd-Tree [15] is suited for fast nearest neighbor search because of the ability to divide the data collection in k-dimensional space, and then search the neighbors in the division space and build a balanced binary tree. It helps one to avoid looking into a deep branch and reducing the searching efficiency. The construction process of kd-Tree is shown in Table 2.

Table 2. The construction process of kd-Tree

a)First, find a direction k which is more dispersed in the k-dimensional space from the data collection. Then use the median value mValue of all the data in this direction as the dividing boundary, so that the two sub data sets are more distributed and uniform. Create a tree node, used to store mValue, expressed as <k, mValue>. b)The two sub sets of (a) are continue to be divided according to (a) until the stop condition is reached. So the leaf node is composed of the data at the end of each division.

In step (a), select the maximum variance of k-dimensional data as the division dimension. The maximum variance shows that the data distribution in this dimension is more dispersed. It can guarantee the data division in this dimension is easier, and can avoid excessive accumulation of data. In addition, using the median value of the dividing dimension to divide the set, it can ensure that the number of data points in two sub sets are equal, and can guarantee the balance of kd-tree.

After constructing the kd-tree, the next step is to find the nearest neighbor of the feature points. The search process is shown in 0

Table 3. The search process of kd-Tree

a) Search starts from the root node of the kd-Tree, and then compares the search data S with each node in turn: If S(k) < mValue, search node in the left subtree; If S(k) > mValue, search node in the right subtree; Search the leaf node to stop, and calculate the distance Dmin between S and the leaf node, it's the current minimum distance. b) Backtrack the search to find whether there are nearest neighbors closer than the current node. If the shortest distance is less than the current in other branches of the parent node, then calculate the distance between search data and the node in the right subtree: If less than Dmin, then enter the branch to find, repeat the search process of step (a) and update the minimum distance Dmin; If more than Dmin, then the branch doesn't contain a closer neighbor than the current; so from the bottom up, find the nearest node until the query node can be found. So the nearest node of the search node can be found through the backtracking search method.

5. Experiment

We collected the image data of the two floors within a building, and recorded the position of each image in the retrieval image database. The resolution of each image was 1000*750, 239 images in total. We extracted the prominent feature parts in the scene from the image database to establish a feature library. Sample images shooting with iPhone5, and most of them shot at the front or side of the prominent character position. The image resolution was 2592*1936 and there were 77 in total. In order to test the matching results of different data, we also collected some images, which with no obvious features and composed of complex shooting scene.

In the image matching process, the resolution of the sample images is too high. In order to reduce the matching time, it's compressed to 40% of the original. This speed up the process of matching with the feature library.

We choose HDF5 [16] database for data storage. The storage is a binary file, and is hierarchical in format. Based on the data index, it overall organized into a binary tree, so that the speed of data processing can be improved. In addition, it

is suitable for the large-scale experimental data where every image has hundreds of feature points. In this paper, every feature point descriptor has 128 dimensions.

OpenCV used the matching function *knnMatch (queryDescriptors, matches, k)* where *queryDescriptors* represente the descriptor to be searched, *matches* represents the feature descriptor vector to be searched, and *k* represents the number of neighbors. In the experiment, k was set to 2 for it only needs two neighbors. When removing the wrong matching feature points, a threshold was used to limit the distance between the nearest neighbor and the next neighbor. If the distance is within the scope of the threshold value, it adopts the matching point and deletes the point otherwise. The threshold value was between 0.5 and 0.7. We designed the experiment process as Fig.6 shows.

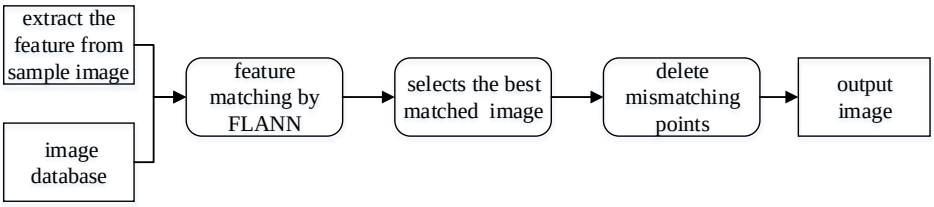


Figure 6. Experiment process

According to the above experimental process, we first extracted the features of the sample image, and then matched them with the features library to get the target image. The experimental results showed that 66 sample images were correctly matched. The matching success rate reached 85.7% and the average matching time was 12.25s.

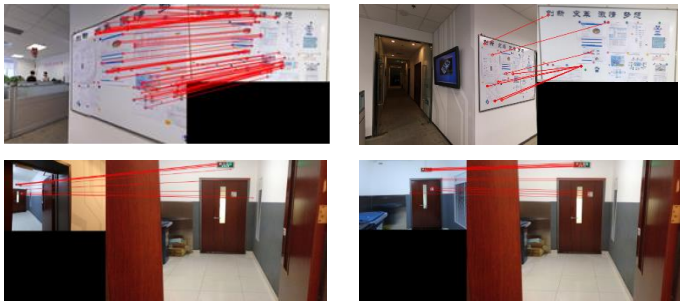
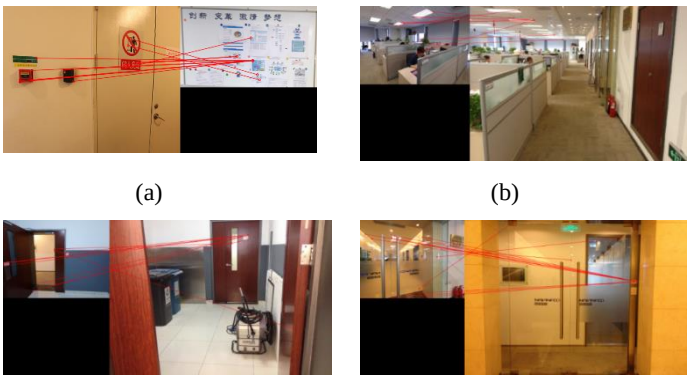


Figure 7. Successful matched images

Analyzing the successful matched images, we found that the proposed algorithm can successfully match the rotating image and the image in different scales, as shown in Figure 7.



(a) (b)

(c) (d)

Figure 8. Matching failed image

However, there were many image that can't match the target image correctly as shown in 0. Analyzing the failed images, we got the following reasons:

- The image to be matched cannot find the corresponding image in the database. As shown in Figure 8 (a), the switch on the wall is not related to the image in the database.
- The image to be matched does not have a unique feature, the infrastructure with similar features in the scene are too similar, cause error matching results, as shown in Figure 8 (b).
- Features in the image library are not unique, and they appear in multiple places in the scene. For example, the two doors of the elevator, although unique, they are not the only doors in the image, as shown in Fig.8 (c), (d).

Due to the image library and sample images containing some interference images, there were image matching failures and some images matching error. In order to improve the matching accuracy, the following aspects need to be taken into account:

- The feature in the image library should be unique; approximate features cause wrong matching.

- The image library should contain all the features in the scene as clear as possible.
- The image shooting angle should highlight the image feature. It's best to sample the features in the front or the side; in order to prevent the features that are not obvious or difficult to match.

6. Conclusion

In this paper, the image retrieval technique is combined with spatial location information, and the feature library is constructed for the collected images to reduce the possibility of mismatches. HDF5, which is suitable for large-scale data and enables efficient data reading, is used for data storage. Then, feature matching is conducted via the FLANN algorithm. Mismatches are removed from the matching results. In the experiments, we achieved a matching success rate of 85.7% and an average matching time of 12.25s. We believe our work forms the foundation for the use of VR for indoor localization. However, due to the limits on experimental conditions, the time efficiency of the proposed method is poor and needs to be improved further.

References

[1] Tsui B Y. Fundamentals of Global Positioning System Receivers. A Software Approach[J]. Wiley, 2005, 51(6-7):93~99

[2] Hada Y, Takase K. Multiple mobile robot navigation using the indoor global positioning system (iGPS)[C]// Ieee/rsj International Conference on Intelligent Robots and Systems, 2001. Proceedings. 2001:1005-1010 vol.2

[3] Liu H, Darabi H, Banerjee P, et al. Survey of Wireless Indoor Positioning Techniques and Systems[J]. IEEE Transactions on Systems Man & Cybernetics Part C, 2007, 37(6):1067~1080.

[4] Liu X Y, Aeron S, Aggarwal V, et al. Adaptive Sampling of RF fingerprints for Fine-grained Indoor Localization[J]. Hungarian Journal of Industry & Chemistry, 2015, 58(3):1~1.

[5] Chen Y, Lymberopoulos D, Liu J, et al. FM-based indoor localization[C]// International Conference on Mobile Systems, Applications, and Services. ACM, 2012:169~182.

[6] Fang Y, Yong K C, Zhang S, et al. Case Study of BIM and Cloud–Enabled Real-Time RFID Indoor Localization for Construction Management Applications[J]. Journal of Construction Engineering & Management, 2016, 142(7).

[7] Verma P, Bhatia J S. Design and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring[J]. International Journal of Computer Science Engineering & Applica, 2013, 3(3):33~40.

[8] Li W, Duan L, Xu D, et al. Text-based image retrieval using progressive multi-instance learning[C]// International Conference on Computer Vision. IEEE Computer Society, 2011:2049~2055.

[9] Smeulders A W M, Worring M, Santini S, et al. Content-Based Image Retrieval at the End of the Early Years[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2000, 22(12):1349~1380.

[10] Kokare M, Chatterji B N, Biswas P K. A Survey on Current Content based Image Retrieval Methods[J]. Iete Journal of Research, 2015, 48(3-4):261~271.

[11] Wu X, Tang Y, Bu W. Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform[J]. IEEE Transactions on Information Forensics & Security, 2014, 9(3):526~536.

[12] Indyk P. Approximate nearest neighbors: towards removing the curse of dimensionality[J]. Theory of Computing, 2015, 604~13.

[13] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF)[J]. Computer vision and image understanding, 2008, 110(3): 346~359.

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

[15] Zhou K, Hou Q, Wang R, et al. Real-time KD-tree construction on graphics hardware[J]. Acm Transactions on Graphics, 2008, 27(5):126.

[16] Folk M, Heber G, Koziol Q, et al. An overview of the HDF5 technology suite and its applications[C]// Edbt/icdt Workshop on Array Databases, Uppsala, Sweden, March. 2011:36~47.

[17] Han J, Kamber M. Data Mining Concept and Techniques[M]. 2006. P421~424