International Journal of
Geo-Information

Article

Research on Feature Extraction Method of Indoor Visual Positioning Image Based on Area Division of Foreground and Background

Ping Zheng [1], Danyang Qin [1,*], Bing Han [1], Lin Ma [2] and Teklu Merhawit Berhane [3]

[1] Department of Electronic and Communication Engineering, Heilongjiang University, Harbin 150080, China; 2201700@s.hlju.edu.cn (P.Z.); 2201701@s.hlju.edu.cn (B.H.)
[2] Department of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150080, China; malin@hit.edu.cn
[3] Department of Computer and Information Sciences, Dire-Dawa Institute of Technology, Dire Dawa 3000, Ethiopia; mberhane63@yahoo.com
* Correspondence: qindanyang@hlju.edu.cn

Abstract: In the process of indoor visual positioning and navigation, difficult points often exist in corridors, stairwells, and other scenes that contain large areas of white walls, strong consistent background, and sparse feature points. Aiming at the problem of positioning and navigation in the real physical world where the walls with sparse feature points are difficult to be filled with pictures, this paper designs a feature extraction method, ARAC (Adaptive Region Adjustment based on Consistency) using Free and Open-Source Software and tools. It divides the image into foreground and background and extracts their features respectively, to achieve not only retain positioning information but also focus more energy on the foreground area which is favourable for navigation. In the test phase, under the combined conditions of illumination, scale and affine changes, the feature matching maps by the feature extraction algorithm proposed in this paper are compared with those by SIFT and SURF. Experiments show that the number of correctly matched feature pairs obtained by ARAC is better than SIFT and SURF, and whose time of feature extraction and matching is comparable to SURF, which verifies the accuracy and efficiency of the ARAC feature extraction method.

Keywords: visual positioning; indoor navigation; feature extraction; feature matching; Free and Open-Source Software

check for updates

Citation: Zheng, P.; Qin, D.; Han, B.; Ma, L.; Berhane, T.M. Research on Feature Extraction Method of Indoor Visual Positioning Image Based on Area Division of Foreground and Background. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 402. https://doi.org/10.3390/ijgi10060402

Academic Editors: Paolo Dabove, Bianca Federici and Wolfgang Kainz

Received: 26 March 2021
Accepted: 8 June 2021
Published: 11 June 2021

1. Introduction

Indoor positioning cannot use the GNSS (Global Navigation Satellite System) due to the obstruction of the GPS (Global Positioning System) signal by the building [1]. The field of indoor positioning and navigation is developing rapidly [2]. Many experts and scholars at home and abroad have conducted a lot of research on indoor positioning methods [3] and technologies [4–6]. The positioning methods such as RFID (Radio Frequency IDentification) and Bluetooth will reduce the positioning accuracy due to various problems such as building materials, multipath fading, and noise interference, so their applicability is limited. With the continuous advancement of AI (Artificial Intelligence), machine vision positioning has developed rapidly. Compared with other indoor positioning methods, visual positioning has unique advantages because it does not require additional equipment. The feature extraction [7] and matching [8] of images in the process of visual positioning are important research contents, which are also the core technical issues in the field of computer vision. They are of high application value in scene recognition, image stitching, positioning navigation, intelligent vision diagnosis, robot vision, security monitoring, automatic driving and other engineering fields [9]. Image feature extraction is a key step in image recognition. The effect of feature extraction directly determines the effect of image recognition, thereby affecting subsequent positioning and navigation. How to extract image features with highly representative characteristics from original images is a research hotspot in intelligent image processing, and it is a prerequisite for feature matching, scene recognition, and indoor positioning and navigation.

Common feature point extraction methods include SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features), and ORB (Oriented FAST and Rotated BRIEF) methods [10]. The SIFT proposed by Lowe in 1999 is one of the most common local feature representation methods. This algorithm is being used to detect and describe local interest points each of which is accompanied by a corresponding size factor on objects. It is invariant to geometric characteristics and noise, and it is also stable to viewpoint changes [11]. However, its feature vector is 128-dimensional, so it has a large amount of calculation and a slower calculation speed [12,13], which is not suitable for navigation and other situations with high real-time requirements. Researchers have proposed many improvement methods on its basis. The SURF algorithm [14] proposed by Bay et al. in 2006 reduces the dimensions of the SIFT descriptive features, making it superior to the SIFT algorithm in performance and speed. The algorithm comparison experiment in literature [15] shows that the SURF algorithm is the most robust local feature algorithm. Based on this advantage, the SURF algorithm is widely used in image matching and related fields. The FAST (Features from Accelerated Segment Test) method has a faster calculation speed and has been applied in the ORB (Oriented FAST and Rotated BRIEF) description method. The traditional ORB method uses the BRIEF (Binary Robust Independent Elementary Feature) descriptor with directional information to calculate the grayscale of random point pairs in the neighbourhood of feature points. However, the feature descriptor is more sensitive to noise, so the matching effect is not ideal [16,17].

Whether it is SIFT, SURF or their improved algorithms, they all extract many features indiscriminately without screening. If it is in a wide-area outdoor or indoor condition where there are particularly many feature points, the features extracted by SIFT, SURF, and their improved methods are very good. However, in the process of positioning and navigation, we often encounter some corridors, stairwells, hallways, partitions in shopping malls and supermarkets, libraries, computer rooms and other places with large areas of white walls. The grayscale of the background of such scenes changes smoothly and the feature points are relatively sparse. The number of feature points provided for feature matching is limited, which increases the difficulty of positioning and navigation. In this type of scene, when SIFT and SURF are used for feature extraction, the large number of extracted feature points will increase the false matching rate and greatly extend the time of feature matching due to problems such as different illumination and affine transformation for the lack of screening. What is more, in the existing research, especially in the process of indoor positioning and navigation, a large number of markers, images and logos are often posted on the walls of the corridor by researchers themselves to improve the accuracy. Some objects are also placed on the table to increase the feature points to be extracted and matched to assist our positioning [18], but this is not the real situation. In the actual physical world, a large number of consistent backgrounds are difficult to be filled with logos and pictures, which leads to the studies of image feature extraction and matching in such sparse feature point scenes to ensure the accuracy of indoor positioning and navigation.

Therefore, it is especially aiming at scenes with more white walls, strong background consistency, and sparse feature points to serve indoor navigation. This paper proposes a feature extraction method ARAC (Adaptive Region Adjustment based on Consistency) to support high-precision positioning and navigation in response to the changes in the image due to different conditions such as shooting position, angle and illumination. This method is based on the free and cross-platform text editor, Visual Studio Code, and the cross-platform computer vision library, Open CV (Open source Computer Vision library) issued under the BSD license (Berkeley Software Distribution) and uses python for programming to achieve feature extraction in a specific environment. Firstly, the Grab Cut method is used to segment the foreground and background of the indoor scene images. It is divided into foreground areas with abundant scene information and distinct features and background areas with gentle changes. Then it extracts features of the foreground and background areas respectively, so as not to lose the position information in the background of the image, but also focus more energy on the foreground areas with salient features. It achieves the purpose of improving the speed and accuracy of feature extraction to adapt to real-time navigation. In addition, faster speeds of extraction and matching can be achieved if foreground and background are processed in parallel.

The roadmap of the research is shown in Figure 1.

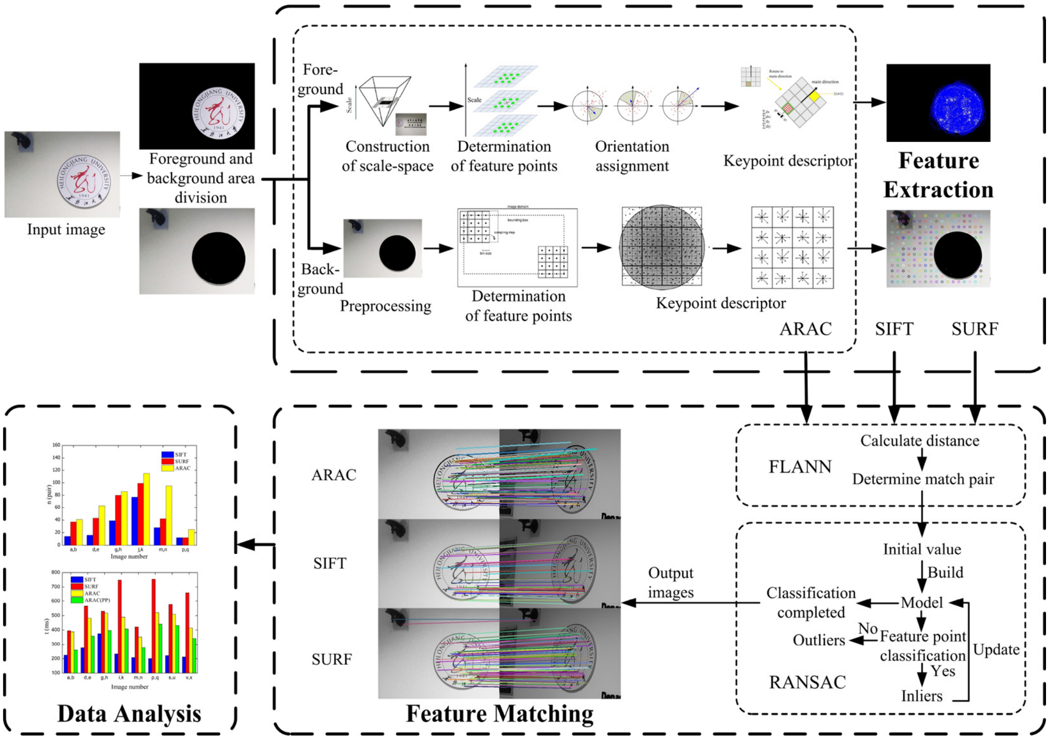


Figure 1. Roadmap of the research. In the data analysis, blue, red, yellow, and green represent SIFT, SURF, ARAC, and ARAC under parallel processing, respectively.

2. Basic Concepts

2.1. Features of the Image

The features of an image are the essential characteristics that can be distinguished from other types of images. The geometric characteristics of these features can be obtained from the extracted data through measurement or processing. Some features are natural features that can be felt intuitively, such as brightness, edges, textures and colours; some are obtained through transformation or calculations, such as matrix, histograms, scale-invariant and principal components [19].

Image recognition is actually a classification process. To identify the category to which an image belongs, we need to distinguish it from other images of different categories. This requires that the selected features not only describe images well but more importantly, they must be able to distinguish images of different categories. We hope to select those image features with small differences between images of the same type and large differences between images of different categories, which we call the most discriminative features.

2.2. Features Matching

Image matching refers to the method of finding similar images in two or more images through a certain algorithm. In the research of digital image processing, image feature extraction and matching have always been key issues, which play vital roles in image registration, target detection, pattern recognition, computer vision and other fields.

2.2.1. FLANN

Feature matching records the feature points of the target image and the image to be matched, and constructs a descriptor according to the feature point set, compares and filters this feature descriptor, and finally obtains a mapping set of matching points. We can also measure the matching degree of two pictures according to the size of this set.

Image feature point matching is commonly used by BF Matcher (Brute Force Matcher) and FLANN matching [20]. The difference between the two is that BF Matcher tries all possible matches to find the best match. FLANN matching is an approximation method. It is faster because what it found is the nearest neighbouring match. The matching accuracy can be improved by adjusting the parameters.

In 2009, Muja and Lowe proposed the FLANN algorithm, which is a collection of nearest neighbour search algorithms for large data sets and high-level features, and the algorithm is not affected by local sensitive hashing. FLANN is mainly implemented based on a k-d tree or k-means tree. The effective search type and retrieval parameters are given by the distribution characteristics of the known data set and the required space resource consumption. The feature space required by this algorithm is usually a vector space where n is a real number. The key point of this algorithm is to find the nearest neighbour point in the nearest neighbourhood according to the Euclidean distance. The mathematical definition of the Euclidean distance is shown as Equation (1):

$$D(x,y) = ||X,Y|| = \sqrt{\sum_{i=1}^{d}(X_i - Y_i)^2} \tag{1}$$

If the obtained value of D is smaller, it means that the distance between the existing feature points is very “close”, that is, the higher similarity of the feature point pairs in the image.

We use FLANN based on KNN (K Nearest Neighbour search) method to search and filter the search results [15]. The KNN search method is adopted when K = 2, that is, for each feature point in the search image, the KNN two-nearest neighbour search method is used in the training image to find its nearest neighbour and second nearest neighbour. We need to compare the distance between the feature point and two neighbouring points, denoted by D_1 and D_2 respectively. Only when the distance between the feature point and the nearest neighbour point is much smaller than the distance between the feature point and the next neighbour point, we consider that the feature point and the nearest neighbour point are correct matching pairs. Otherwise, when the two distances are relatively close, we can consider that the feature point and the two neighbouring points are not correctly matching pairs, which should be eliminated. Alternatively, we could think that the feature point can be correctly matched with the two neighbouring points (for example, there are several identical objects in the scene), but we should also eliminate them to reduce the subsequent impact of this matching relationship. In the finally realized system, when the ratio of the distance between the feature point and the nearest neighbour to the distance between the feature point and the next neighbour is less than 0.75, the corresponding matching pair is retained, otherwise, it is eliminated.

2.2.2. RANSAC

In the FLANN pre-matching process, the accuracy of the obtained matching points is not ideal, and the accuracy of target recognition will be affected to a certain extent, so the wrong matching points need to be deleted. We delete false matching points based on RANSAC. This algorithm can handle matching pairs with an error rate of more than 50%. It is one of the more robust feature matching screening algorithms and will greatly reduce false matching points.

RANSAC is the abbreviation of Random Sample Consensus. It is a set of sample data sets containing abnormal data. The mathematical model parameters of the data are calculated and the effective sample data is obtained. Fischler and Bolles proposed this algorithm in 1981, and it is widely used in computer vision.

The basic assumption of the RANSAC algorithm is that the sample contains inliers that can be described by the model and outliers that deviates far from the normal range and cannot adapt to the mathematical model. It means that the data set contains noise. The mismatched feature pairs in this paper are outliers.

RANSAC is an iterative process. In each iteration of the original RANSAC, the characteristic points are randomly sampled as initial values, and a model is fitted based on the inliers. This model is adapted to the assumed inliers, and all unknown parameters can be calculated from the assumed inliers. The currently obtained model is used to test all other data. If a certain point is suitable for the estimated model and the error is less than the set threshold, it is considered to be an inlier. If enough points are classified as hypothetical inliers, then the estimated model is reasonable enough. Then all the assumed inliers could be used to update the model estimated based on the inliers of the initial hypothesis. We need to iterate continuously, and finally, evaluate the model by estimating the error rate of the inliers.

2.3. Open Source Computer Vision Library

In the field of GIS (Geographical information system) research, Free and Open-Source Software and tools have been widely used. In the research of indoor positioning and navigation, more and more researchers are developing and exploring based on Open Source Software and tools such as OpenCV. It is a cross-platform computer vision and machine learning software library released under the BSD license. This manuscript uses the free tool OpenCV for programming to improve the replicability, reusability and accessibility, which is conducive to other researchers for subsequent verification and development.

3. Division of Foreground and Background Area and Feature Extraction

3.1. Feature Extraction Model Construction

When using visual methods for indoor navigation, the quality of the feature points extracted from the images and the effect of feature matching are closely related to the results of indoor navigation. The more correct matching pairs, the higher the probability of determining the same situation, the better effect of positioning and navigation. In addition, for indoor images, due to different shooting positions, angles, and lighting conditions, the corresponding scale and brightness changes will affect the detection results. Therefore, in the process of feature extraction, we must also consider the interference caused by factors such as image angle, spatial location, and illumination. Designing an effective feature extraction method will help the subsequent feature matching and navigation in the indoor positioning process.

Aiming at the problems in the engineering context mentioned in the introduction, this paper proposes a feature extraction method ARAC, which can adaptively divide the foreground and background regions. The feature extraction model is shown in Figure 2.

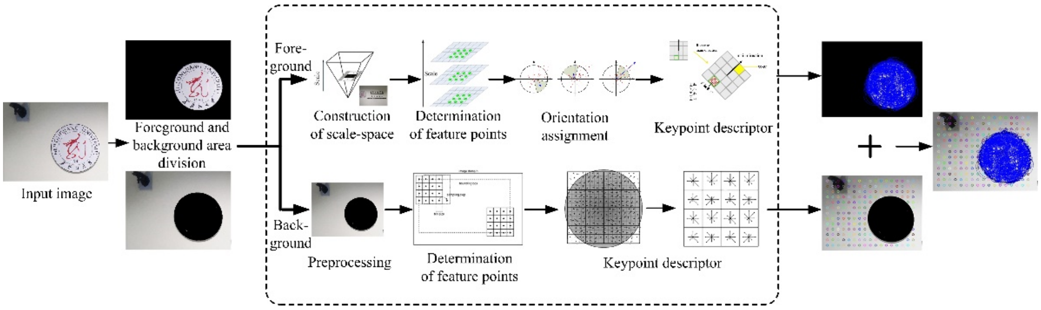

Input image
Foreground and background area division
Fore-ground
Construction of scale-space
Determination of feature points
Orientation assignment
Keypoint descriptor
Back-ground
Preprocessing
Determination of feature points
Keypoint descriptor
+

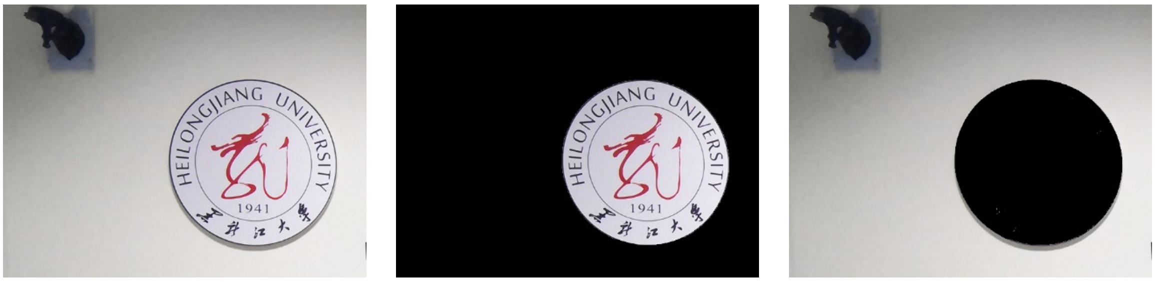
Figure 2. Feature extraction model. The blue circles in the foreground area represent the feature points extracted in this area, and the feature points extracted in the background area are represented by uniformly distributed circles of various colours.

1. This feature extraction model takes the corridor on the seventh floor of the experimental building of Heilongjiang University as an example. The first step is to read the image to be feature extracted in Visual Studio Code;
2. Grab Cut in OpenCV is used to adaptively divide the input image into foreground and background. Then different areas are processed separately;
3. In the foreground area, the method proposed in this paper firstly builds a scale-space, and secondly compares the response value of the candidate feature points with the Hessian matrix threshold to determine the feature points. Then it calculates the Haar response value to determine the main direction, and finally generates the 64-dimensional feature descriptors of the foreground area;
4. In the background area, ARAC firstly preprocesses the image to limit the size of the image and divides the background area into grids of equal size. It extracts features in each grid, and finally generates 128-dimensional feature descriptors of the background area;
5. This completes the feature extraction of the Heilongjiang University badge in the foreground area of the corridor and the wall and lamp in the background.

3.2. Adaptive Region Division

For the input indoor image, the foreground and the background area are divided adaptively. This manuscript uses the Grab Cut method to process the image. This method is an image segmentation algorithm based on graph cut, which uses a bounding box specified by the user as the location of the segmentation target to achieve segmentation of the foreground object and the background image.

This method first loads the image to be processed, creates a rectangular mask with the same size as the foreground image and fills it with zeros. The area outside the rectangle is automatically recognized as the background. According to the rectangular area defined by the user, the data in the background can be used to distinguish the background and foreground areas in the rectangular frame. Then the GMM (Gaussian Mixture Model) is used to model the background and foreground, and the undefined pixels will be marked as possible foreground and background. Each pixel in the image is considered to be connected with surrounding pixels through a virtual edge, and each edge has a probability of belonging to the background or foreground, based on the colour similarity between itself and the surrounding pixels. Each pixel is connected to a foreground or background node. After the nodes are connected, if the edges between the nodes belong to different terminals, that is, one node belongs to the foreground and one node belongs to the background, the edges between them will be cut off, which will separate the parts of the image. This method is used for the foreground and background segment of the indoor images. The results of regional segmentation are to retain the foreground or background, and the rest is filled with black which are shown in Figure 3.



(**a**) (**b**) (**c**)

Figure 3. Foreground and background image segmentation. The part of the badge of Heilongjiang University is the foreground and the rest is the background: (**a**) origin image; (**b**) foreground; (**c**) background.

3.3. Feature Extraction in Foreground Area

In feature extraction of foreground region, Hessian matrix and integral calculation template are introduced, and the process of Gaussian filtering in SIFT is replaced by several addition and subtraction operations. It greatly improves the efficiency of feature description to reduce the dimension of the feature descriptor instead of the 128-dimension of SIFT. In addition, the complexity of the segmented image is reduced, which leads to lower computational complexity. Moreover, the foreground feature extraction method only works in the local foreground area, so it need not traverse the whole image like SIFT and SURF, which greatly improves the operation speed.

3.3.1. Introduction of Calculation Template

To speed up feature extraction and detection, this paper introduces a determinant approximation image of the Hessian matrix. The Hessian matrix was proposed by the German mathematician Ludwin Otto Hessian in the 19th century. It is a square matrix composed of the second-order partial derivatives of a multivariate function, which describes the local curvature of the function. Its determinant is shown in Equation (2), and the corresponding Hessian matrix can be calculated for each pixel.

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \tag{2}$$

Through the determinant of the Hessian matrix shown in Equation (3), whether it is an extreme point could be judged.

$$\det(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 \tag{3}$$

When the discriminant of the Hessian matrix obtains a local maximum value, it is determined that the current point is brighter or darker than other points in the surrounding neighbourhood, to locate the position of the key point.

Before constructing the Hessian matrix, Gaussian filtering is required to keep scale-invariant. In discrete space, because the Gaussian kernel can construct response images at different scales, this paper uses the second-order standard Gaussian kernel function to convolve the image to obtain the four elements of the Hessian matrix. At scale σ, at point $f(x,y)$, the corresponding Hessian matrix is shown in Equation (4)

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{4}$$

Among them, according to the LoG (Laplacian of Gaussian), the derivative of a function is equal to the convolution of the function and the derivative of the Gaussian function, that is, $L_{xx}(x,\sigma)$ is the convolution result of the second-order partial derivative of the Gaussian function $g(x,y,\sigma)$ at the point $f(x,y)$, as shown in Equations (5) and (6). The calculation method of $L_{xy}(x,\sigma)$ and $L_{yy}(x,\sigma)$ is the same.

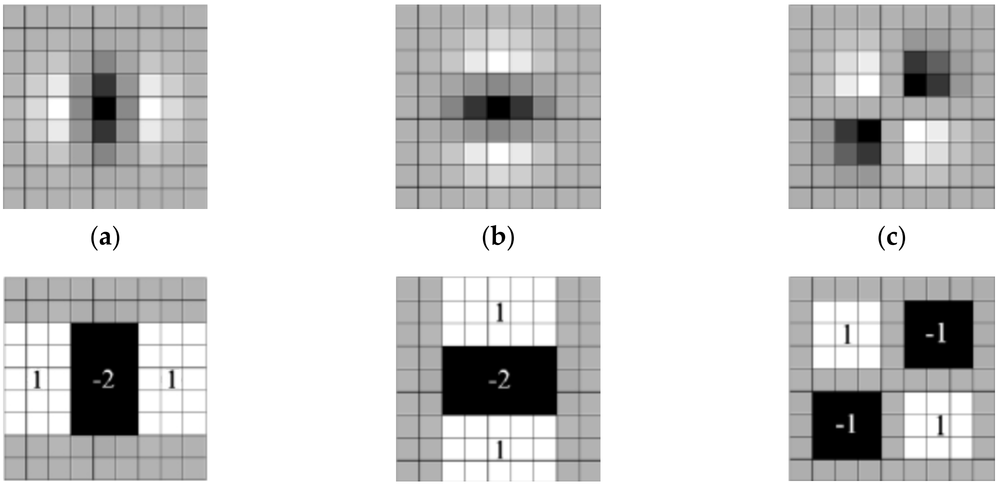
$$L_{xx}(x,\sigma) = G(x,y,\sigma) \otimes f(x,y) \tag{5}$$

$$G(x,y,\sigma) = \frac{\partial^2 g(x,y,\sigma)}{\partial x^2} \tag{6}$$

To speed up the calculation of the foreground area, this paper operates on the integral image to achieve acceleration. The box filter is used instead of the Gaussian filter to approximate the Gaussian second-order derivative template. It only takes a few addition and subtraction operations to calculate the Hessian matrix of each pixel, and the amount of calculation is independent of the template size, so the scale pyramid of ARAC can be quickly constructed. The value of each pixel in the integral image, calculated according to Equation (7), is the sum of all elements in the upper left corner of the corresponding position on the original image.

$$I_{\sum}(x,y) = \sum_{i=0}^{i\leq x}\sum_{j=0}^{j\leq y} I(x,y) \tag{7}$$

A template can replace the two steps of Gaussian filtering and finding the second derivative which is shown in Figure 4. D_{xx}, D_{yy} and D_{xy} are approximations to L_{xx}, L_{yy} and L_{xy} respectively. The numbers in Figure 4 indicate the weights of the corresponding colour areas. The weight of the grey is zero. The box filter and image convolution results are denoted as D_{xx}, D_{yy} and D_{xy} respectively.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 4. Gaussian filter template and its substituted template: (**a**) L_{xx}; (**b**) L_{yy}; (**c**) L_{xy}; (**d**) D_{xx}; (**e**) D_{yy}; (**f**) D_{xy}.

The template is widely used because of its fast calculation speed on the integral image. However, the template is an approximation of the original Hessian matrix. The derivation can prove that the following formula is closer to the true value. When we use Gaussian second-order partial derivatives with σ = 1.2, the template size is 9 × 9, which is the smallest scale-space value for image filtering and spot detection. As shown in Equation (8), the determinant of the Hessian matrix can be simplified as follows:

$$\begin{aligned} Det(H) &= L_{xx}L_{yy} - L_{xy}L_{xy} \\ &= D_{xx}\tfrac{L_{xx}}{D_{xx}}D_{yy}\tfrac{L_{yy}}{D_{yy}} - D_{xy}\tfrac{L_{xy}}{D_{xy}}D_{xy}\tfrac{L_{xy}}{D_{xy}} \\ &= D_{xx}D_{yy}\left(\tfrac{L_{xx}}{D_{xx}}\tfrac{L_{yy}}{D_{yy}}\right) - D_{xy}D_{xy}\left(\tfrac{L_{xy}}{D_{xy}}\tfrac{L_{xy}}{D_{xy}}\right) \\ &= A\left(\tfrac{L_{xx}}{D_{xx}}\tfrac{L_{yy}}{D_{yy}}\right) - B\left(\tfrac{L_{xy}}{D_{xy}}\tfrac{L_{xy}}{D_{xy}}\right) \\ &= \left(A - B\left(\tfrac{L_{xy}}{D_{xy}}\tfrac{L_{xy}}{D_{xy}}\right)\left(\tfrac{D_{xx}}{L_{xx}}\tfrac{D_{yy}}{L_{yy}}\right)\right)\left(\tfrac{L_{xx}}{D_{xx}}\tfrac{L_{yy}}{D_{yy}}\right) \\ &= (A - BY)C \end{aligned} \tag{8}$$

At the same time, according to Equation (9), the Hessian matrix determinant is further modified to calculate the weight of each expression.

$$Y - \frac{|L_{xy}(1.2)|_F |D_{xx}(9)|_F}{|L_{xx}(1.2)|_F |D_{xy}(9)|_F} = 0.912 \cong 0.9 \tag{9}$$

The constant C does not affect the comparison of extreme points, so the final simplified formula is shown in Equation (10):

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{10}$$

The Hessian matrix response value of each pixel is calculated by Equation (10) as the response image at the scale σ.

From this, we can get a determinant that approximates the Hessian matrix. In addition, to balance the error caused by using the box filter approximation, the response value should be normalized according to the filter size to ensure that the F-norm of any size filter is uniform. The corresponding relative weight w of the filter is 0.9. Theoretically, the size of the template corresponding to different σ is not the same, and the value of w is different. For the sake of simplicity, it is considered to be the same constant.

3.3.2. Construction of Scale-Space

The purpose of constructing the scale space is to find the extreme points in the spatial domain and the scale domain as preliminary feature points. The traditional method of constructing scale space is to construct a Gaussian pyramid with the original image as the bottom layer, and then perform Gaussian blur and downsampling on the image as the next layer of the image. Next, it continues to iterate until meeting the condition. In the Gaussian pyramid, the size of the original image is constantly changing, and the size of the Gaussian template remains unchanged. The establishment of each layer can only be processed after the construction of the previous layer is completed. The dependence is very strong, which causes the speed to be very slow.

The method used in this paper to construct the scale pyramid is the opposite. The original image size remains unchanged, but the template size is changed, that is, the original image is filtered by the template box with the size changing to construct the scale space. Parallel operations are used to process each layer of the pyramid simultaneously. The response image of the Hessian matrix generated by the convolution of the gradually increasing box size filter template and the integral image is used to suppress the 3D non-maximum value on the response image to obtain various spots of different scales. The down-sampling process in traditional scale pyramid construction is omitted, thereby improving the processing speed. The pyramid image constructed by ARAC is shown in Figure 5.

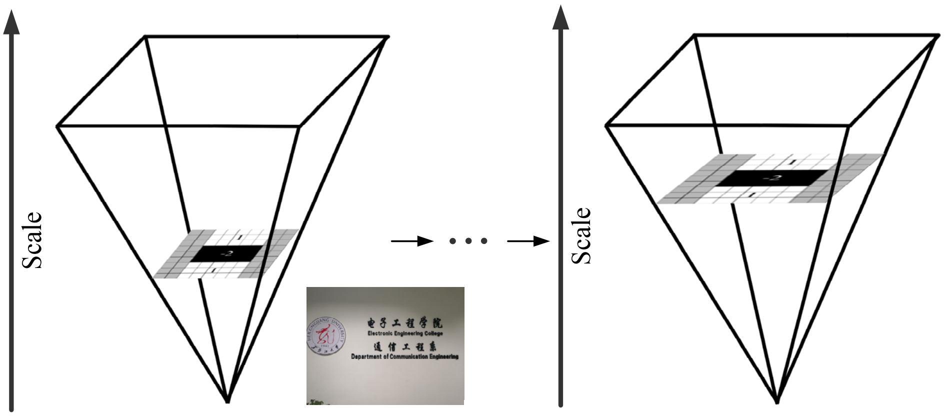

Scale
Scale
电子工程学院
Electronic Engineering College
通信工程系
Department of Communication Engineering

Figure 5. Pyramid image. The original image size remains unchanged, but the template size is changed.

ARAC uses the box filter of 9 × 9 as the initial filter, the response image obtained is taken as the bottom image. Then the size of the filter gradually increases which can be calculated by Equation (11), and the original image continues to be filtered.

$$\text{Filter\_Size} = 3 \times \left(2^{\text{octave}} \times \text{interval} + 1\right) \tag{11}$$

Both octave and interval start from one in Equation (11), that is, when on the 0th interval of the 0th octave, in the Equation (11), octave = 1 and interval = 1.

The reasons for choosing this method to define the filter size are as follows. The pyramid image consists of a number of fixed layers. Since the integral image is discretized, the smallest scale change between the two layers is determined by the Gaussian second-order derivative filter. It is determined by the length l_0 of the positive and negative spot response, whose size is 1/3 of the box filter template. For the box filter of 9 × 9, l_0 is 3. The response length of the next layer should be increased by at least 2 pixels to ensure one pixel on each side, that is l_0 = 5 so that the size of the template is 15 × 15. Analogously, we can get a sequence of templates with gradually increasing sizes. Their sizes are 9 × 9, 15 × 15, 21 × 21 and 27 × 27 respectively, and the length of the black and white areas increases by an even number of pixels to ensure the existence of a central pixel. According to this template sequence, a scale-space is constructed.

3.3.3. Determination of Foreground Feature Points

In the selection process of feature points, to locate points of interest in images of different sizes, a 3 × 3 × 3 filter is taken as an example. The response values of each pixel on each layer of the image are compared with the values in spatial and the scale neighbourhood (excluding the first and last layer). There are 8 neighbourhood pixels on the same layer and 2 × 9 = 18 pixels in the vector scale space, with a total of 26-pixel values for comparison. The maximum and minimum candidate feature points are selected. If the response value of the feature point is less than the threshold of the Hessian determinant, it is discarded. Increasing the threshold can reduce the number of detected feature points, and eventually, only a few of the strongest points will be detected. If the feature value of the pixel marked "x" in the figure is greater than the surrounding pixels, it can be determined that the point is the feature point of the area as Figure 6 shows.

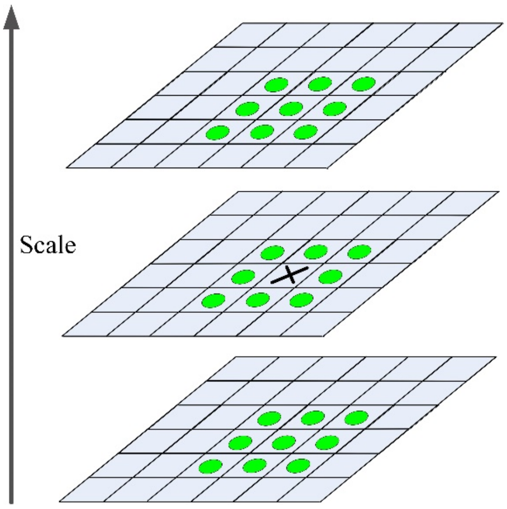


Figure 6. Determination of feature points. The pixel marked "x" in the middle layer is the feature point of the area whose value is greater than green points.

To ensure rotation invariance, it is necessary to assign main directions to the feature points. Taking the feature point as the centre, the Harr wavelet responses of all pixels are counted in a circle with a radius of 6S. The results of image convolution as shown in

Figure 7 are adopted as the horizontal and vertical Harr wavelet response of each pixel point. The size of the Harr wavelet is 4S, as shown in Equation (12). The scale value S where the feature point is located is calculated according to the size of the current template.

$$\sigma_{approx} = Current\_Filter\_Size \times \frac{Base\_Filter\_Scale}{Base\_Filter\_Size} = Current\_Filter\_Size \times \frac{1.2}{9} \quad (12)$$

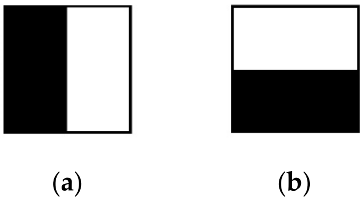


Figure 7. Haar wavelet template. The black is 1 and the white is –1: (**a**) horizontal response; (**b**) vertical response.

After calculating the response values of the image in the horizontal and vertical directions of the Haar wavelet, the two values are Gaussian weighted with a factor of 2S, and the weighted values represent the directional components in the horizontal and vertical directions, respectively. The Harr feature value reflects the change of the image grayscale, and then this main direction is to describe those areas where the grayscale changes particularly sharply.

The main direction can be obtained by taking the feature point as the centre and sliding the sector with an opening angle of 60° as shown in Figure 8. Then the accumulation of the horizontal and vertical responses of the Harr wavelet value in the window are calculated according to Equations (13) and (14).

$$m_w = \sum_w \mathrm{dx} + \sum_w \mathrm{dy} \quad (13)$$

$$\theta_w = \arctan\left(\sum_w \mathrm{dx} / \sum_w \mathrm{dy}\right) \quad (14)$$

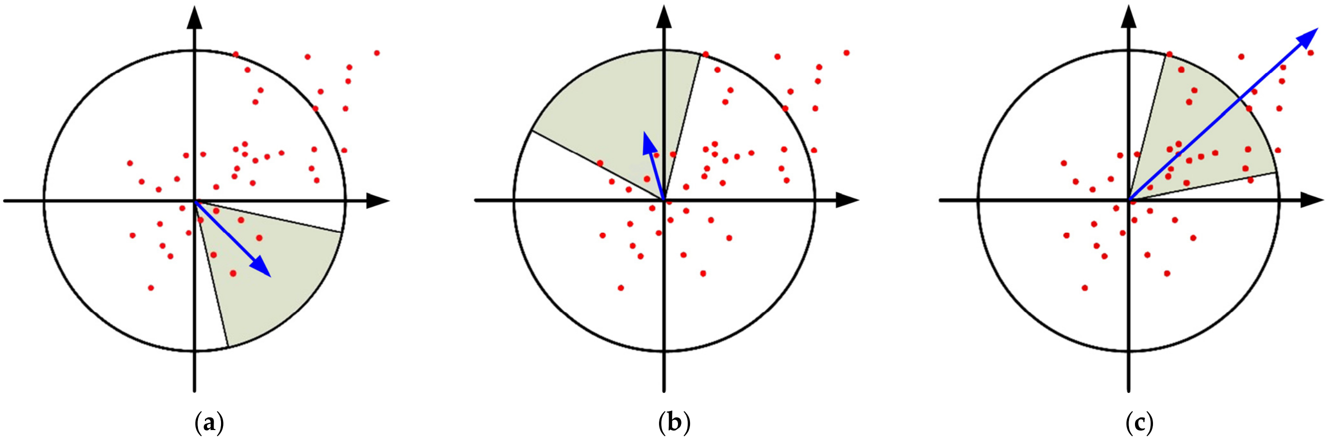


Figure 8. Determination of the main direction of the ARAC feature point. Take the feature point as the centre and slide the sector with an opening angle of 60°: (**a**) general vector length; (**b**) minimum vector length; (**c**) the main direction whose victor is the largest.

3.3.4. Foreground Feature Point Descriptor

When generating feature descriptors, ARAC selects a square frame with a side length of 20S around the feature point and coincides the direction of the square with the main direction determined in the previous step as shown in Figure 9. The square frame is divided into 4 × 4 sub-regions, each with a size of 5S × 5S pixels. Then we count the horizontal and vertical Haar wavelet features of 25 pixels in each region. Each Haar wavelet feature contains 4 values which are the sum of the horizontal direction Σdx, the absolute value of the horizontal direction Σ|dx|, the sum of the vertical direction Σdy and the sum of the absolute value of the vertical direction Σ|dy|, where the horizontal and vertical directions are relative to the main direction. These four values are taken as the feature vector of each sub-block region, that is, a common 4 × 4 × 4 = 64-dimensional vector is used as the feature descriptor of the foreground region of ARAC. Compared with traditional SIFT, the feature dimension is generally reduced, which greatly improves the speed of scene recognition. Finally, the feature vector is normalized according to Equation (15) to prevent the influence of illumination and contrast ratio.

$$v_{subregion} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy|\right] \tag{15}$$

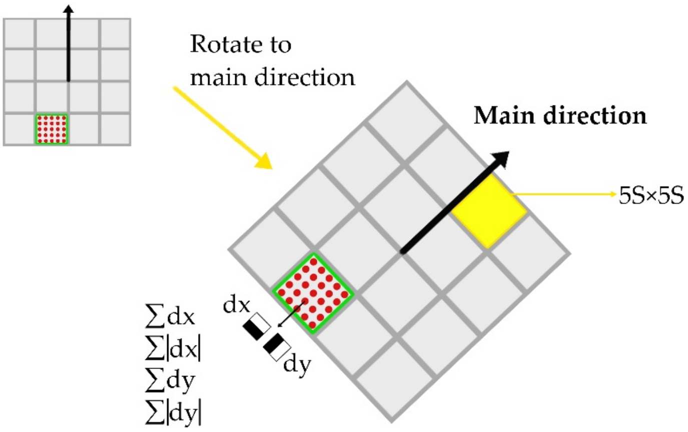


Figure 9. ARAC foreground feature descriptor generation. The square is rotated to coincide with the main direction.

The ARAC algorithm is used to extract features of the foreground area. It can be seen from Figure 10 that the ARAC feature descriptor can intensively and effectively describe the characters and patterns in the school badge of Heilongjiang University, which is fundamental to the determination of the scene and provides effective evidence for indoor positioning and navigation.

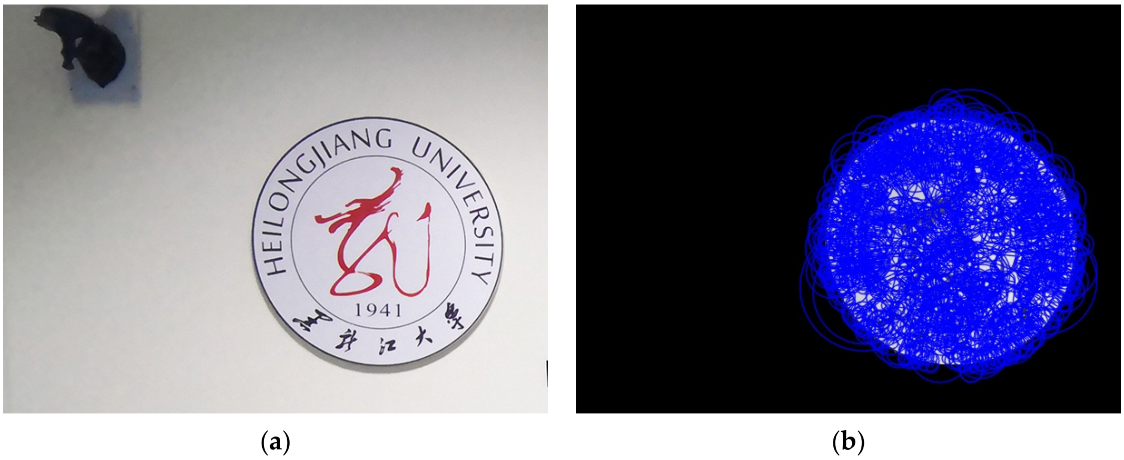


Figure 10. Foreground feature extraction based on ARAC. The feature points extracted from the foreground area are represented by blue circles: (**a**) origin image; (**b**) ARAC foreground feature.

3.4. Feature Extraction in Background Area

In the processing of the background area with strong consistency, the image is preprocessed to reduce the amount of calculation for subsequent feature extraction. When detecting feature points in the background area, ARAC does not construct a scale space like SIFT and SURF but adopts a uniform sampling method, which greatly improves the processing speed. If the background can be processed in parallel with the foreground, the running time of the method proposed will be even less.

3.4.1. Image Background Preprocessing

The background feature extraction process of the ARAC adopts a fixed-step and fixed-grid sliding window to scan the background of the entire image, and an ARAC feature is extracted in each grid. Therefore, the higher the image resolution, the more local feature points are extracted. For an image containing large areas of smooth background, there are often a large number of similar feature points in the extracted features. These feature points not only are very helpful for classification and positioning but will increase the amount of calculation for subsequent processing. To solve such problems, this paper proposes an image preprocessing method, which can greatly reduce the resolution of the image while preserving the details of the image.

For the original image $I \in R^{a \times b}$, assuming that the conversion matrix is $R_m \in R^{c \times d}$, the preprocessed image is $I_n \in R^{c \times d}$, where a and b are the size of the original image, c and d are the size of the converted image, according to Equations (16)–(18), the relationship between I, R and I_n can be expressed as follows with a mathematical model:

$$I_n = R_m I_\gamma R_m^{\mathrm{T}}, \tag{16}$$

$$\gamma = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \cdots \right), \tag{17}$$

$$R_m = \begin{pmatrix} R_{11} & \cdots & R_{1d} \\ \vdots & \ddots & \vdots \\ R_{c1} & \cdots & R_{cd} \end{pmatrix} \tag{18}$$

In Equation (16), $c = \gamma \times a$ and $d = \gamma \times b$, I_γ is the way of sampling the original image pixels. R_m is a block matrix, and each block is a matrix of $(\gamma^{-1}) \times 1$. In this paper, the area is divided into 2 × 2 non-overlapping grids, and the elements in R_m are shown in Equation (19).

$$R_{11} = R_{12} = \cdots = R_{cd} = \frac{1}{\sqrt{2}} \times \left[\begin{array}{cc} 1 & 1 \end{array} \right] \tag{19}$$

The preprocessed image is shown in Figure 11. If the original image is 600 × 500, using Equations (16) and (18), the image can be expressed compactly as 300 × 225. Most of the details of the image after preprocessing are not lost as shown in Figure 11.

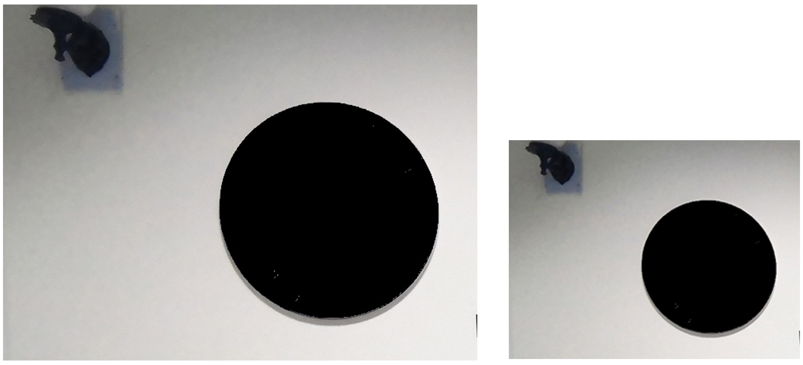

(a) (b)

Figure 11. Image preprocessing: (**a**) origin background image; (**b**) preprocessed background image.

If you want to further reduce the image resolution, you can do the same operation on I_n again to reduce the size of the image. Through the above-mentioned preprocessing operation, when extracting the features of the background area, this paper limits the resolution of all images uniformly within the range 300 × 300.

3.4.2. Determination of Background Feature Points

In the process of feature extraction of the background area of the indoor image, the background area of the image is divided into grids of equal size, and the features of the image are extracted in each grid. This process can be regarded as a uniform sampling of the image. It can extract the local features of the whole image completely. Even in areas where the texture and colour change smoothly, the local features of the image can also be expressed.

The generation of the ARAC background area descriptor does not acquire feature points in the Gaussian scale space, but directly samples the image uniformly in a sampling window of a specified size, and treats each pixel in the image as a key point. Then it obtains the sampling point coordinates and its feature descriptors in the image by sliding window.

A certain point is taken in the image as an example which is shown in Figure 12. A custom-sized patch is adopted to slide on the image with a certain step, and then the ARAC features of each patch block are calculated from left to right and top to bottom. This window is the sample area of the descriptor. The window size is 4bins × 4bins, and the bin size can be specified by yourself. The bin mentioned here corresponds to the sub-region in the ARAC feature extraction. The bounding box in the figure is the range of ARAC feature points.

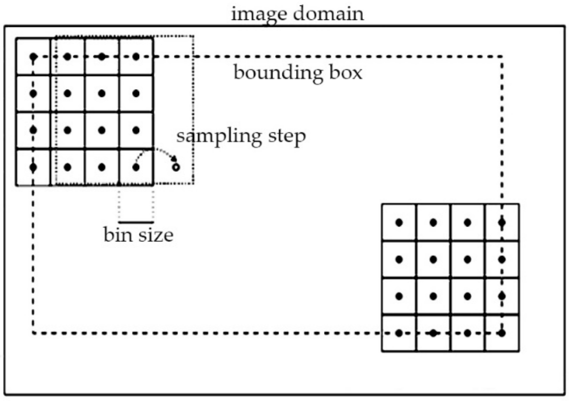


Figure 12. Acquisition of background sampling points. Use a patch of custom size to slide on the image in a certain step.

3.4.3. Background Feature Point Descriptor

The algorithm divides the rectangular area representing the target into small blocks of the same size, calculates the characteristics of each small block, and then samples the ARAC features of each small block at the centre position. The black dot at the centre is the selected pixel, and the outermost frame represents the selected image block as shown in Figure 13.

It calculates the gradient of each pixel, counts the gradient histogram of the pixel in each bin in 8 directions, and takes the peak of the histogram as the main direction of the block. The 8-bit histograms of 4 × 4 small blocks in each sampling window are connected to form a 128-dimensional ARAC feature descriptor. The effect of feature extraction on the background with ARAC is shown in Figure 14.

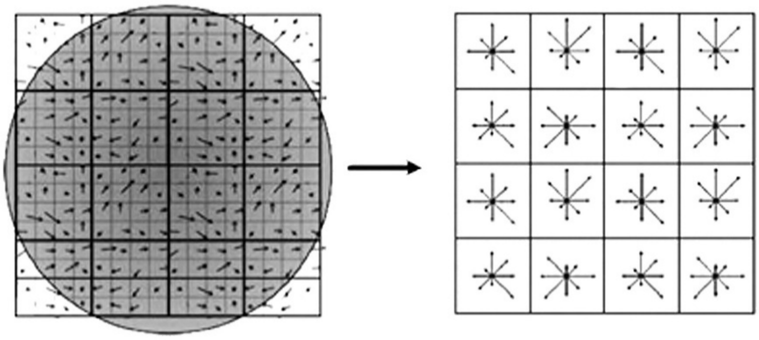


Figure 13. Background feature descriptor. Each small square represents a pixel, the arrow on the small square represents the gradient information of each point, the direction of the arrow is the direction of the gradient, and the length of the arrow represents the magnitude of the gradient.

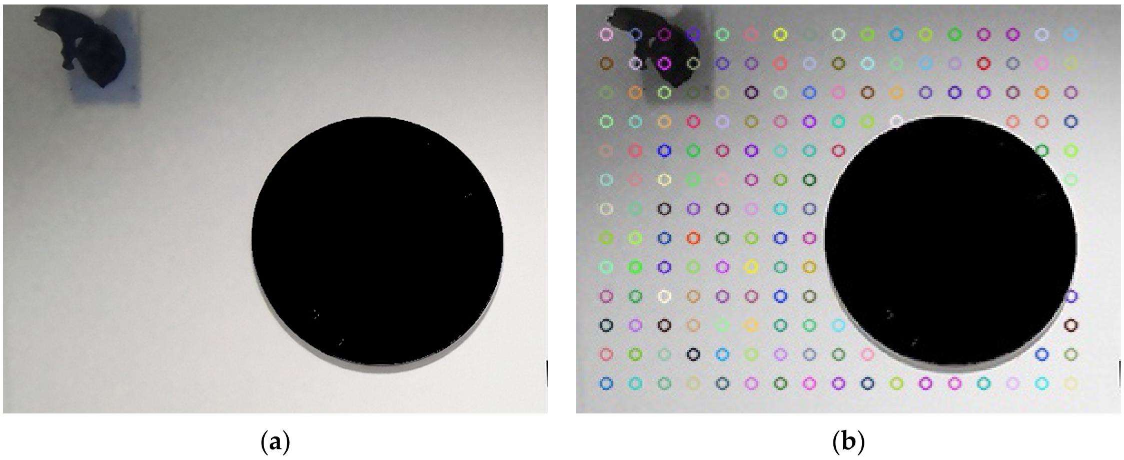


Figure 14. ARAC background feature extraction. The uniformly distributed circles of various colours in the background area represent the feature points extracted in this area: (**a**) origin background image; (**b**) background feature extracted by ARAC.

4. Match Test and Performance Analysis

4.1. Match Condition Construction

When students have classes, take large exams, listen to lectures, and participate in conferences, indoor navigation with high real-time performance is required. In the actual navigation process, we often encounter scenes with strong background consistency such as corridors and stairwells, or indoor environments with many rooms such as teaching buildings, hospitals, science parks, and libraries. Such scenes have fewer feature points and higher similarity, so feature extraction and matching are more difficult. Therefore, we shot in the Huiwen Building of Heilongjiang University whose construction meets the standards of large-scale examinations. The appearance of the teaching building and several indoor panoramic pictures are shown in Figure 15. Nine floors of the building are the main teaching areas, with 80 classrooms on each floor. It also contains four large lecture theatres, where meetings and lectures are often held.

To verify the effectiveness and accuracy of the ARAC proposed in this paper, as well as the robustness under different conditions of illumination and shooting angles, SIFT, SURF and ARAC algorithms are used to extract the features of the image. Under the combined conditions of illumination, affine and scale variations, FLANN is adopted to match the extracted features, and RANSAC is used to filter the matching pairs. Then the images matching results are compared and the data are analysed

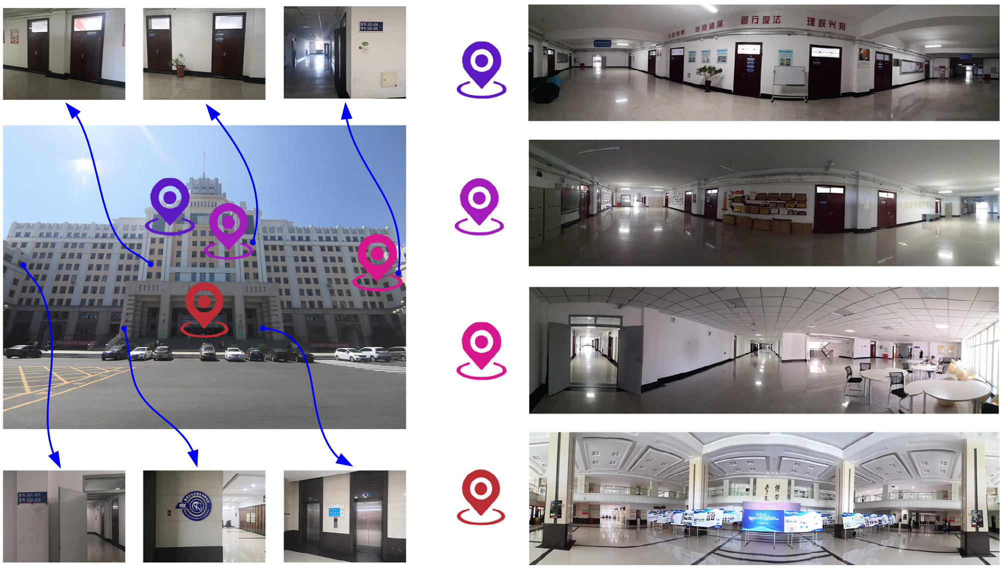


Figure 15. The actual scene and appearance of the Huiwen Building. The blue points represent the positions of some experimental images on the exterior of the building. Some areas of the teaching building are displayed in four panoramic pictures on the right according to the corresponding colours.

In the experiment, the PC is configured as Inter(R) Core(TM) i5-5200U CPU @ 2.2 GHz, and the memory is 4 G. The integrated development environment is Visual Studio Code, and the development language is python3.7. Based on the Open-Source tool, OpenCV, the image features are extracted and matched. The specific matching flowchart is shown in Figure 16 below.

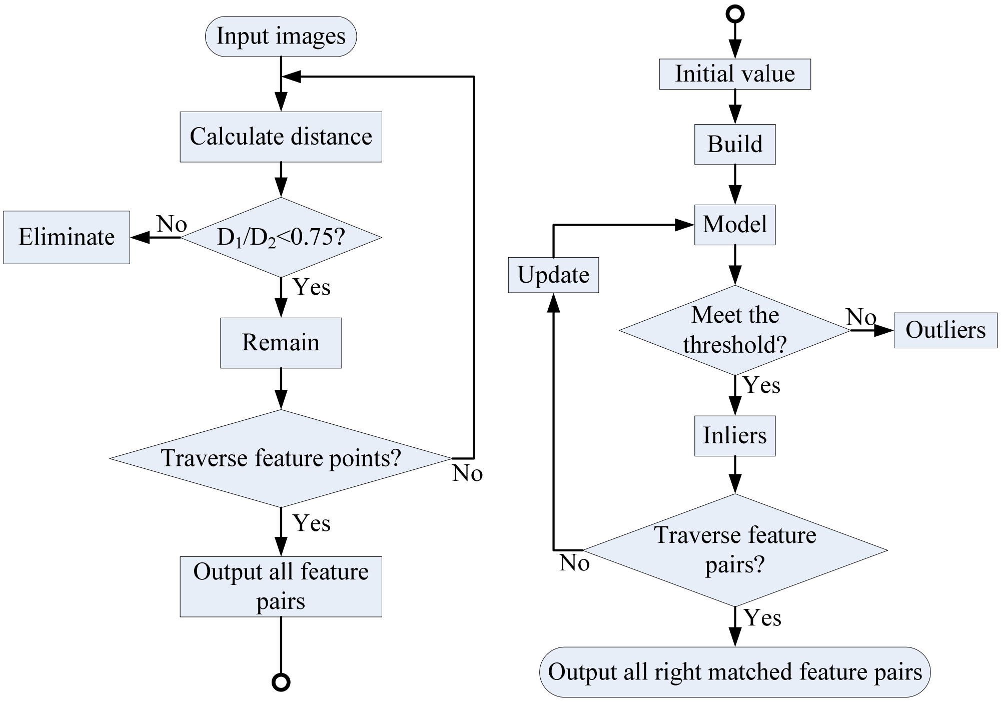


Figure 16. Matching flowchart.

4.2. Feature Extraction Performance Analysis

4.2.1. Test results of Each Feature Point Detection Method

To compare the actual detection effects of several feature extraction methods, eight indoor images are used as experimental objects. It includes corridors with logos and signposts, as well as classrooms and other real scenes with fewer background feature points and higher similarity. We use SIFT, SURF and the ARAC feature extraction method proposed in this paper to extract feature of the images which are shown in Figure 17a.

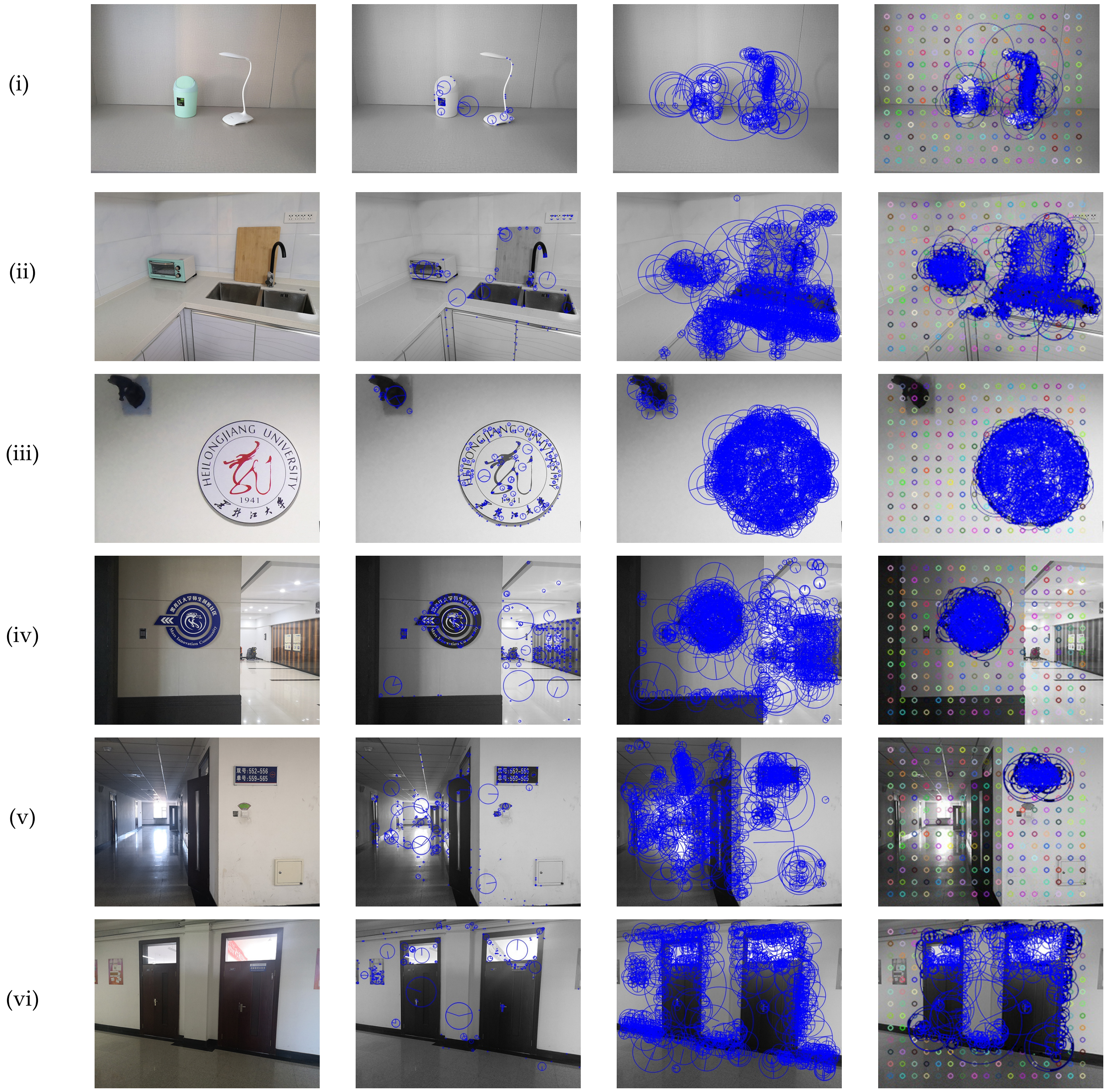


Figure 17. *Cont.*

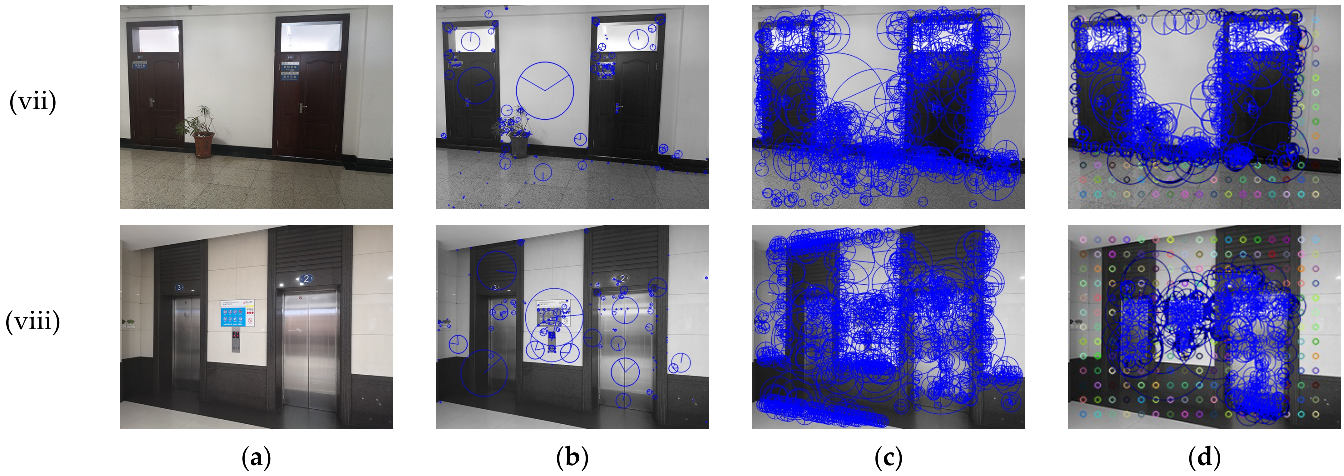


Figure 17. Feature extraction effect comparison. The first group to the eighth group are arranged in order from top to bottom. The feature extraction method for each column is the same: (**a**) origin images; (**b**) features extracted by SIFT; (**c**) features extracted by SURF; (**d**) features extracted by ARAC.

From the detection results in Figure 17, it can be found that there are differences in the ability of various feature detection methods to extract features. The results of SIFT feature detection show that different feature points have different sizes of feature points in the original image because of the different scale spaces detected. This method extracts feature points from the letters and Chinese characters on the school badge of Heilongjiang University, the corners of the oven, the cutting board, and even the shadow of the trash can, but the number of feature points extracted by this method is relatively small. The feature detection method of SURF detects a large number of feature points, and the selected neighbourhood range is larger. As shown in the fifth group of Figure 16, it not only extracts a large number of feature points in the foreground area but also extracts a lot of features in the distant light and shadow. From the comparison of the feature extraction results, it can be seen that the feature points extracted by SIFT and SURF are concentrated in the foreground area with obvious feature changes. The ARAC feature detection method in this paper divides the foreground and background areas, and then extract features with their own methods individually. It not only detects the overall and local features of the oven, school badge and signs in the foreground area but also extracts the features of the white walls, cabinets and desktops with gently changing in the background area, achieving the effect of not losing positioning information.

4.2.2. Feature Extraction Performance Analysis

To compare the feature extraction methods in more detail, each method is used for detecting 100 times to get the average time of images as shown in Table 1.

Table 1. Feature extraction time of three spot detection methods.

Group	Detection Method	Number of Feature Points	Time (ms)
(i)	SIFT	40	199.13
	SURF	232	197.42
	ARAC	165	190.86
	ARAC(PP)	165	174.31
(ii)	SIFT	166	141.82
	SURF	1256	283.19
	ARAC	897	241.24
	ARAC(PP)	897	218.73

Table 1. *Cont.*

Group	Detection Method	Number of Feature Points	Time (ms)
(iii)	SIFT	342	200.76
	SURF	1581	264.94
	ARAC	1381	258.36
	ARAC(PP)	1381	229.09
(iv)	SIFT	479	216.43
	SURF	1116	240.48
	ARAC	658	239.17
	ARAC(PP)	658	174.71.
(v)	SIFT	337	195.45
	SURF	890	234.63
	ARAC	380	226.53
	ARAC(PP)	380	163.54
(vi)	SIFT	213	168.99
	SURF	997	247.19
	ARAC	633	256.07
	ARAC(PP)	633	192.58
(vii)	SIFT	249	181.26
	SURF	1075	244.54
	ARAC	738	276.22
	ARAC(PP)	738	209.97
(viii)	SIFT	193	138.67
	SURF	1257	256.58
	ARAC	718	258.19
	ARAC(PP)	718	197.23

The results of Table 1 show that the number of features detected by SURF is four to eight times that of SIFT, and the detection time is also longer than SIFT. Nevertheless, provided that the number of features extracted is the same, SURF needs less time. Through calculation, it can be found that the feature extraction efficiency of SURF is about three times that of SIFT. The extraction time of ARAC is longer than that of SIFT but it is similar to SURF because it extracts a large number of features from both foreground and background. If ARAC could be processed in parallel, it will take less time as shown in Table 1 named ARAC(PP). Data in Table 1 suggested that it takes longer to extract features of the images of the groups (vi) to (viii) with ARAC than SURF because the foreground areas of these three groups of images are relatively large. Moreover, when the image is segmented, a part of the background area is mistakenly divided into the foreground, which results in a slightly longer extraction time. But no matter which group of images, if parallel processing is adopted for feature extraction of foreground and background areas, the time consumed is always less than that of SURF.

4.3. Feature Matching Effect Comparison

The previous section compares the speed and number of features extracted by ARAC and other feature detection methods in this manuscript, but only the extraction of feature points does not make much sense. The repeatability and discrimination of features are the references for the quality of feature extraction methods. Feature detection is related to the repeatability of features, and feature descriptors are related to the distinguishability of features. The features applied to scene recognition need to be able to cope with various changes. The repetitiveness and discrimination of features can be reflected in the matching of images. Eight groups of indoor images under different shooting conditions are taken as the matching objects which are shown in Figure 18. Among them, the first two groups are regular scenes with desktops and cabinets as the background. The 3rd and 4th groups are scenes with school badges and logos in the corridor. The 5th to 8th groups are all scenes inside the teaching building. The foreground of the 5th group is a signpost, and the 6th to 8th groups contain very similar classroom doors and elevator doors.

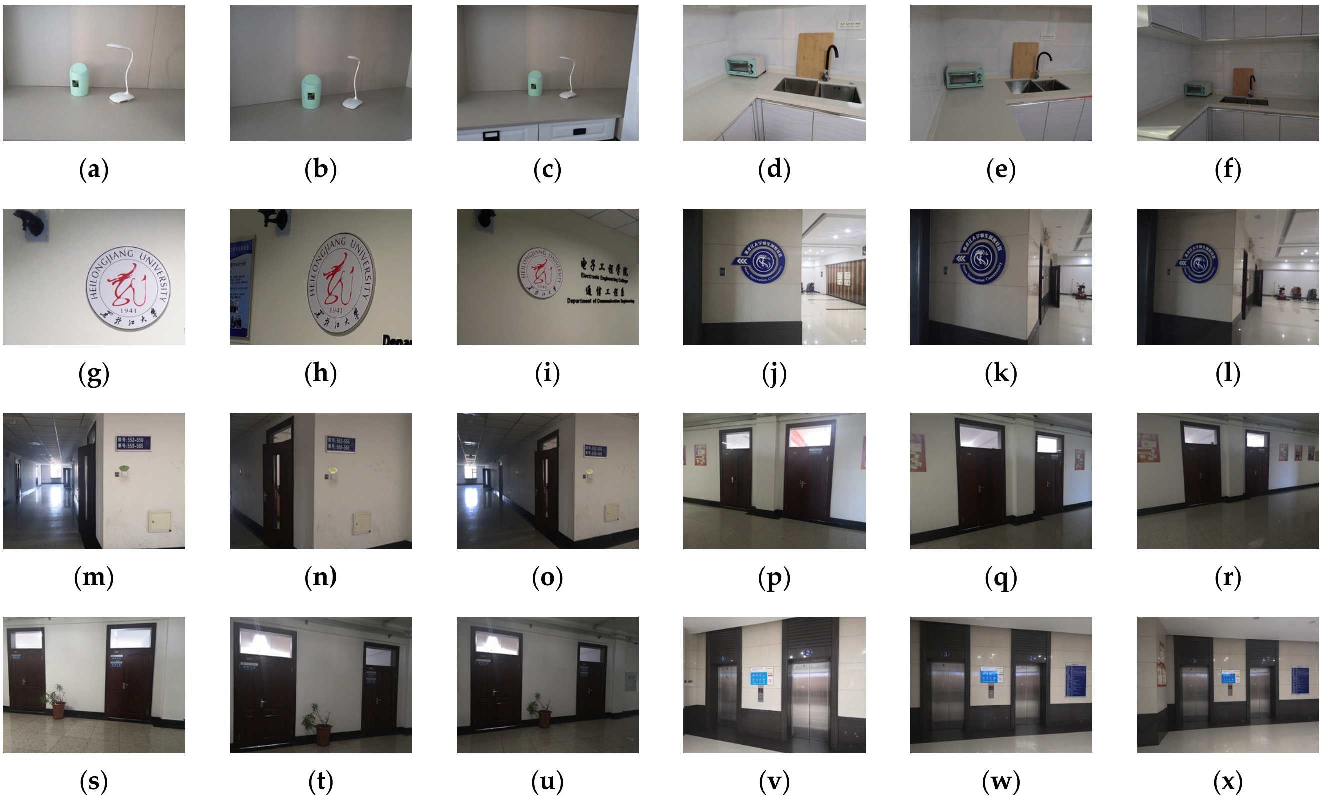

(a) (b) (c) (d) (e) (f)
(g) (h) (i) (j) (k) (l)
(m) (n) (o) (p) (q) (r)
(s) (t) (u) (v) (w) (x)

Figure 18. Original image of indoor scene to be matched: (**a**,**d**,**g**,**j**,**m**,**p**,**s**,**v**) reference images; (**b**,**e**,**h**,**k**,**n**,**q**,**t**,**w**) images of illumination and affine changes; (**c**,**f**,**i**,**l**,**o**,**r**,**u**,**x**) images of illumination, scale, and affine changes.

The pixel size of each image in the object to be matched shown in Figure 18 is 600 × 450. If the first image in each group of scenes is considered as a reference, there are illumination and affine variations in the second image compared with the first one. Similarly, the third image in this scene adds scale changes to the conditions of the second one. In a corridor environment with limited space, there will be a certain scale change, which can cause differences in the number of objects contained in the two images to be matched and the range of pixels occupied. Figure 18i contains more characters than Figure 18g,h. Similarly, Figure 18r has more posters on the wall than Figure 18p,q. Although the indoor corridor environment space has limitations, changes such as the increase in the number of objects due to scale changes will affect the efficiency of feature extraction. All images are extracted by three methods. Then we match the features of the first image and the other two in the same scene in Figure 18 by FLANN. After screening by the RANSAC method, the average time of their 100 matches, the number of feature points of the two images and the number of correct matching pairs are counted to verify the effect of the experiment.

4.3.1. Matching Effect under Affine and Illumination Conditions

The eight groups of pictures are matched under the conditions of illumination changes and affine transformation as shown in Figure 18. The data are recorded in the Table 2.

Table 2. Matching results under the condition of illumination change and affine transformation.

Matching Pictures	Feature Extraction Method	Number of Feature Points (Left)	Number of Feature Points (Right)	Number of Correct Matching Pairs	Dimension of Feature	Extraction Time (ms)	Matching Time (ms)	Total Time (ms)
a, b	SIFT	40	29	14	128	208.76	16.01	224.77
	SURF	232	218	37	64	378.19	16.67	394.86
	ARAC	465	423	41	64 + 128	371.50	16.44	387.94
	ARAC(PP)	465	423	41	64 + 128	253.49	8.00	261.47
d, e	SIFT	166	91	16	128	261.03	16.24	277.27
	SURF	1256	650	43	64	510.73	55.64	566.37
	ARAC	897	600	63	64 + 128	437.86	44.64	482.51
	ARAC(PP)	897	600	63	64 + 12	322.60	34.98	357.58
g, h	SIFT	342	360	39	128	342.76	31.98	374.74
	SURF	1581	1619	80	64	459.43	70.46	529.89
	ARAC	1381	1315	86	64 + 128	446.99	69.73	516.72
	ARAC(PP)	1381	1315	86	64 + 128	344.30	52.23	396.53
j, k	SIFT	479	379	77	128	177.47	46.20	233.67
	SURF	1116	1065	99	64	682.24	65.63	747.87
	ARAC	658	705	115	64 + 128	471.74	51.32	523.06
	ARAC(PP)	658	705	115	64 + 128	368.20	38.83	407.03
m, n	SIFT	337	147	28	128	180.27	27.58	207.85
	SURF	890	410	42	64	376.11	44.77	420.88
	ARAC	380	345	95	64 + 128	370.59	22.18	392.77
	ARAC(PP)	380	345	95	64 + 128	266.62	11.79	278.41
p, q	SIFT	213	129	12	128	159.24	42.64	201.88
	SURF	997	718	12	64	619.95	134.65	754.60
	ARAC	633	517	25	64 + 128	511.58	47.99	559.57
	ARAC(PP)	633	517	25	64 + 128	403.91	36.49	440.40
s, t	SIFT	249	185	15	128	172.63	49.57	222.20
	SURF	1075	719	42	64	499.61	78.15	577.76
	ARAC	738	827	57	64 + 128	499.04	56.27	555.31
	ARAC(PP)	738	827	57	64 + 128	386.00	45.47	431.47
v, w	SIFT	193	220	27	128	190.16	22.00	212.16
	SURF	1257	883	38	64	552.19	106.86	659.05
	ARAC	718	492	92	64 + 128	416.60	43.31	459.91
	ARAC(PP)	718	492	92	64 + 128	310.67	30.98	341.65

It can be seen from Table 2 that in the case of a small angle of affine transformation and illumination changes, the number of feature points extracted by SURF in the group (ii) to group (viii) is the largest, followed by ARAC and SIFT. It can be observed that except for the two objects in the foreground, the trash can and the lamp in Figure 18a,b, the large area is the background area with gentle grayscale changes. In this region, SIFT and SURF have lost feature extraction and matching capabilities. Because the ARAC background feature extraction method extracts a large number of feature points in this type of area when using ARAC to extract images of Figure 19a,b, the number of feature points obtained is more than SURF.

The feature pairs matched are restricted by the strict conditions of RANSAC, the matching effect of SIFT is the worst and ARAC is the best. After experimenting with SIFT on the groups (v) and (viii), only 28 and 27 correct matches were obtained respectively. The SURF method obtained 42 and 38 correct matching pairs, while the ARAC method obtained the largest number of correct matching pairs, 95 and 92 respectively.

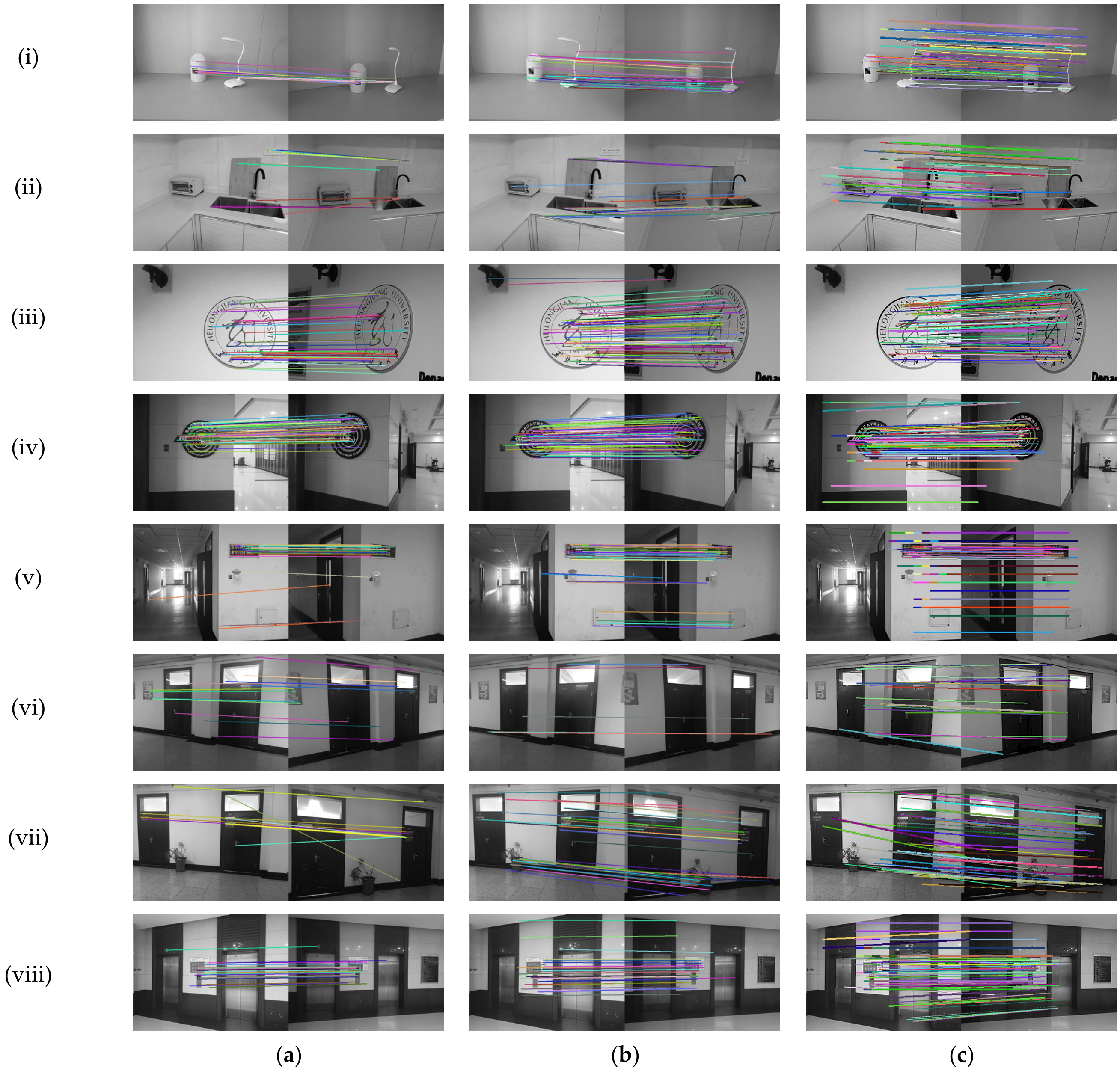


Figure 19. The effect of each feature matching method under the condition of illumination change and affine transformation: (**a**) SIFT; (**b**) SURF; (**c**) ARAC.

In terms of feature extraction and matching time, SURF is the longest and SIFT is the shortest but combined with the number of feature points, it can be found that the feature extraction efficiency of SURF is about three times that of SIFT. The number of feature points extracted by SIFT is the least, so the matching time is the shortest. The ARAC proposed in this paper is longer than SIFT in feature extraction and matching time, but shorter than SURF. If the feature extraction of foreground and background is processed in parallel, the total time becomes shorter. Considering the number of correct matching pairs, feature extraction and matching time, this result proves that the method of feature extraction divided into foreground and background in this paper has good accuracy and effectiveness.

4.3.2. Matching Effect under Large-Angle Affine and Illumination

The eight groups of images are matched under the conditions of large-angle affine and illumination as shown in Figure 20. The data are recorded in Table 3.

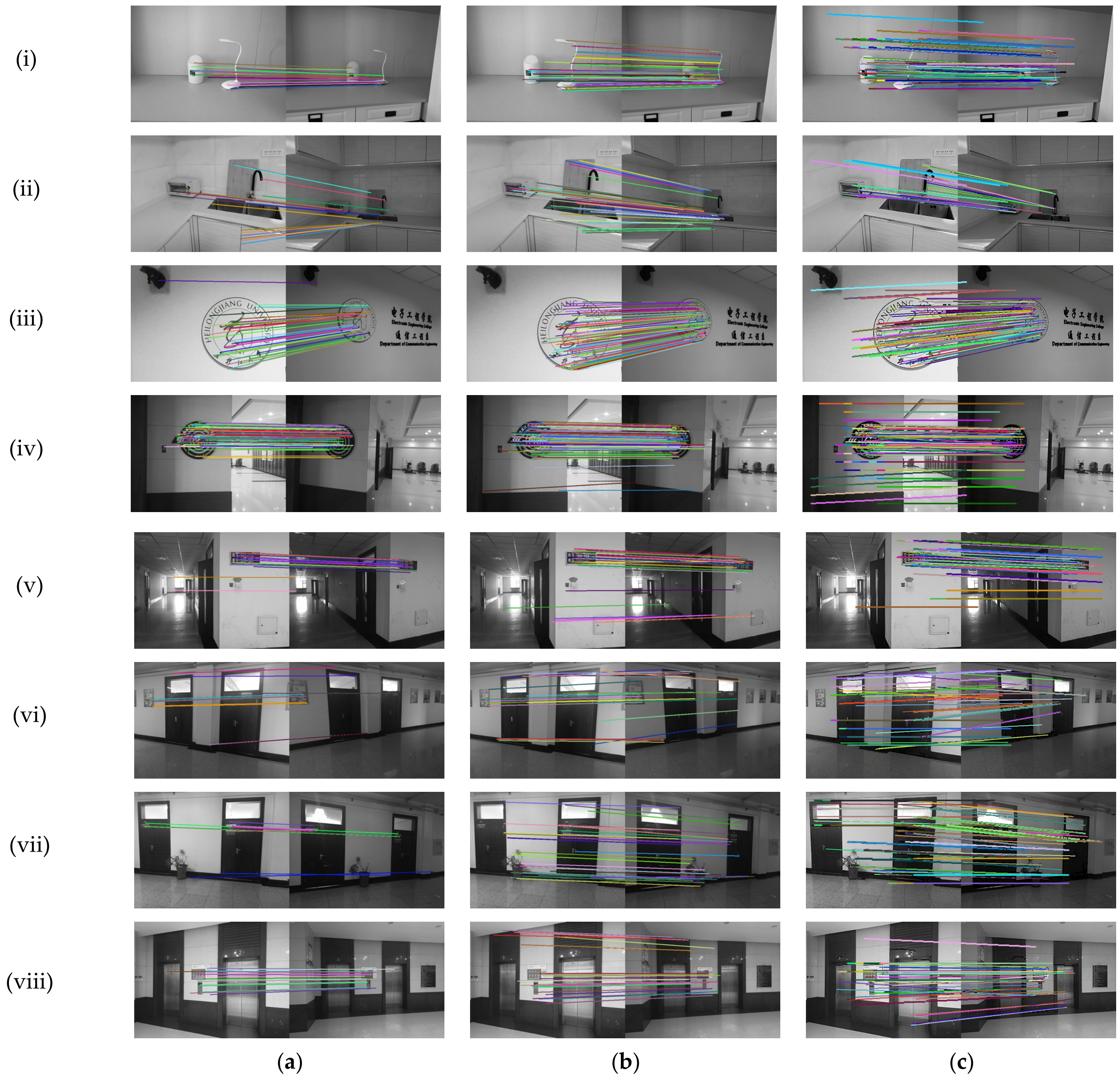


Figure 20. The effect of each feature matching method under the condition of illumination change and large-angle affine transformation: (**a**) SIFT; (**b**) SURF; (**c**) ARAC.

Table 3. Matching results under the condition of illumination change and large-angle affine transformation.

Matching Pictures	Feature Extraction Method	Number of Feature Points (Left)	Number of Feature Points (Right)	Number of Correct Matching Pairs	Dimension of Feature	Extraction Time (ms)	Matching Time (ms)	Total Time (ms)
a, c	SIFT	40	55	21	128	204.48	13.25	217.73
	SURF	232	444	57	64	380.90	23.99	404.89
	ARAC	465	389	91	64 + 128	370.96	19.97	390.93
	ARAC(PP)	465	389	91	64 + 128	255.85	12.00	267.85

Table 3. *Cont.*

Matching Pictures	Feature Extraction Method	Number of Feature Points (Left)	Number of Feature Points (Right)	Number of Correct Matching Pairs	Dimension of Feature	Extraction Time (ms)	Matching Time (ms)	Total Time (ms)
d, f	SIFT	166	114	17	128	302.13	16.99	319.12
	SURF	1256	395	64	64	520.91	46.98	567.89
	ARAC	897	261	65	64 + 128	401.43	40.48	441.91
	ARAC(PP)	897	261	65	64 + 128	288.00	25.98	313.98
g, i	SIFT	342	450	48	128	400.15	31.65	431.80
	SURF	1581	945	122	64	411.69	57.47	469.16
	ARAC	1381	625	151	64 + 128	385.70	59.97	442.67
	ARAC(PP)	1381	625	151	64 + 128	344.30	50.99	395.29
j, l	SIFT	479	421	97	128	185.34	36.31	221.65
	SURF	1116	761	124	64	576.36	55.30	631.66
	ARAC	658	476	159	64 + 128	388.47	42.36	430.83
	ARAC(PP)	658	476	159	64 + 128	288.48	29.15	317.63
m, o	SIFT	337	218	21	128	187.75	32.12	219.87
	SURF	890	620	39	64	457.04	49.57	506.61
	ARAC	380	308	71	64 + 128	372.93	22.78	395.71
	ARAC(PP)	380	308	71	64 + 128	264.63	11.79	276.42
p, r	SIFT	213	143	11	128	187.33	31.96	219.29
	SURF	997	665	28	64	615.67	132.29	747.96
	ARAC	633	389	39	64 + 128	386.01	42.97	428.98
	ARAC(PP)	633	389	39	64 + 128	291.82	30.65	322.47
s, u	SIFT	249	165	14	128	166.37	29.78	196.15
	SURF	1075	607	39	64	482.56	95.36	577.92
	ARAC	738	704	74	64 + 128	466.31	58.08	524.39
	ARAC(PP)	738	704	74	64 + 128	357.96	45.14	403.10
v, x	SIFT	193	244	24	128	176.11	23.44	199.55
	SURF	1257	733	41	64	545.57	113.28	658.85
	ARAC	718	494	63	64 + 128	460.16	55.93	516.09
	ARAC(PP)	718	494	63	64 + 128	354.21	43.60	397.81

According to Figure 20, it can be seen that the positions of the correct matching pairs using SIFT and SURF are concentrated on the foreground objects changed significantly, such as the characters on the school badge and the signpost, the corners and edges of the oven, the lamp, and the trash can. The correct matching pairs after using ARAC appear not only on the objects with distinct features but also on the walls, cabinets and desktops as shown in the group (iv) in Figure 20.

From the results in Table 3, it is obvious that under such a large angle change, SURF extracts the most feature points, followed by ARAC, and SIFT does the least. In the experiments of groups (i), (v) and (vii), SIFT only got 21, 21 and 14 correct matching pairs, while ARAC got the most correct matching pairs, 91, 71 and 74, respectively. The ability of feature extraction and matching time of ARAC are comparable to SURF and even slightly better than SURF. Similarly, the feature extraction and matching time of ARAC using parallel processing are shorter.

The number of correct matching pairs and the total time of extraction and matching obtained by the three methods for feature matching in the two cases are compared as shown in Figures 21 and 22.

According to Figure 21, the ARAC algorithm identified by the yellow bar has the largest number of correct matching pairs in any case, which is better than SIFT and SURF. It verifies the accuracy of the method proposed in this paper. It is found from Figure 21, the extraction and matching time of SIFT is the shortest, but whose ability to extract and match features in such scenes is particularly poor shown in Figure 20. Even if its time of extraction is short, it cannot meet positioning and navigation demand. According to the yellow bar and the red bar, it can be observed that the feature extraction and matching speed of ARAC is slightly better than SURF. And the ARAC with parallel processing shown in the green column with the legend of ARAC(PP) takes less time. Combined that the number of correct matching pairs of ARAC is the largest, it verifies the efficiency of ARAC and meets the requirements of real-time indoor visual positioning and navigation.

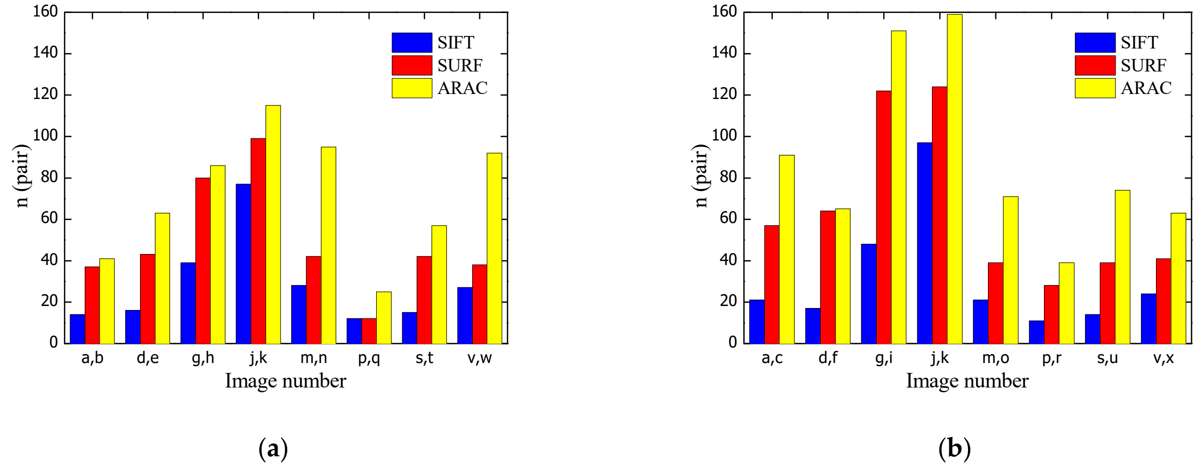


Figure 21. Comparison of the number of correct matching pairs denoted by "n" on the vertical axis: (**a**) data under the condition of illumination change and affine transformation; (**b**) data under the condition of illumination change and large-angle affine transformation.

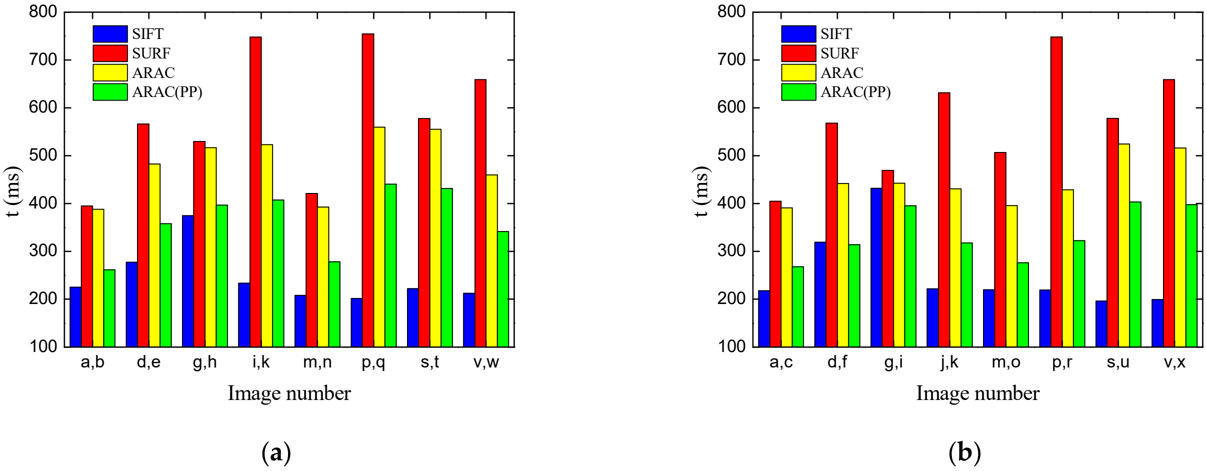


Figure 22. Comparison of total extraction and matching time denoted by "t" on the vertical axis. (**a**) data under the condition of illumination change and affine transformation; (**b**) data under the condition of illumination change and large-angle affine transformation.

5. Discussion

The scenes used in this manuscript are corridors, stairwells, libraries, computer rooms, etc. The foreground and background scenes are very different, with clear outlines and obvious colour divisions. Therefore, this paper uses Grab Cut to segment the area when distinguishing the foreground and the background. This method can find the edge of the foreground and the background in the candidate area according to the determined background area. I found that in a dazzling array of shopping malls, supermarkets and other areas with rich colours, the blur of the foreground and the background is not much different, and too much colour information is not conducive to the discrimination of the foreground and the background. At this time, the effect and advantages of feature extraction in this paper are not obvious. Being able to better distinguish the foreground and the background will provide more effective information for indoor positioning and scene recognition. How to distinguish the foreground and the background more accurately is one of my follow-up research directions.

6. Conclusions

This paper focuses on the feature extraction of corridors, stairwells, libraries, computer rooms and other scenes in indoor navigation, which contain large areas of white walls and cabinets, with strong background consistency and sparse feature points. They are the difficulties of indoor visual positioning and navigation. In the current research, methods such as SIFT and SURF are suitable for indoor and wide-area outdoor scenes with rich feature points, and it is not realistic for researchers, themselves, to post pictures and logos on the wall to increase the number of feature points to assist positioning. Therefore, this paper designs a feature extraction method ARAC. It mainly carried out the following work:

Firstly, an ARAC feature extraction model is built. Secondly, the Grab Cut in OpenCV is used to divide the foreground and background areas of the indoor image in Visual Studio Code. Then, this paper designed the ARAC feature extraction method to extract the features of the foreground and background areas of the image individually. Next, the image under the combined conditions of illumination, scale and affine transformation is matched with SIFT, SURF algorithms in OpenCV. At last, the matching results of ARAC are compared with those of the SIFT and SURF algorithms.

According to the matching data, SIFT algorithm is the fastest in the face of indoor environments where various interference conditions change randomly, but the number of matching pairs is too small, which does not meet the requirements of indoor positioning and navigation. Although the number of feature points extracted by SURF is the largest, the basis for judging whether the feature extraction and matching method are good or not is the number of correct matching pairs. The feature extraction method ARAC designed in this paper has the largest number of correct matching pairs under the combined conditions of illumination, scale and affine transformation, and the time of feature extraction and matching is comparable to that of SURF. Moreover, the time extracted by ARAC with parallel processing is even less. Therefore, the ARAC algorithm in this paper is more adaptable. It not only maintains a certain degree of stability for illumination and viewing angle changes, but also is more robust and real-time. It meets the needs of fast, efficient and stable indoor navigation in this paper.

Author Contributions: Methodology, software, writing—original draft preparation and writing—review and editing, Ping Zheng; formal analysis, Ping Zheng and Danyang Qin; resources and project administration, Danyang Qin; supervision, Danyang Qin, Lin Ma, Bing Han and Teklu Merhawit Berhane. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (61771186), Postdoctoral Research Foundation of Heilongjiang Province (LBH-Q15121) and Outstanding Youth Project of Provincial Natural Science Foundation of China in 2020 (YQ2020F012).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: We thank Heilongjiang University for providing support for the experimental scenes. We also thank the reviewers for the thorough review and greatly appreciate the comments and suggestions, which significantly contributed to improving the quality of the publication.

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

References

1. Ang, J.L.F.; Lee, W.K.; Ooi, B.Y. GreyZone: A Novel Method for Measuring and Comparing Various Indoor Positioning Systems. In Proceedings of the 2019 International Conference on Green and Human Information Technology (ICGHIT), Kuala Lumpur, Malaysia, 16–18 January 2019; pp. 30–35.
2. Zafari, F.; Gkelias, A.; Leung, K.K. A Survey of Indoor Localization Systems and Technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599. [CrossRef]
3. Gezici, S.; Zhi, T.; Giannakis, G.B.; Kobayashi, H.; Molisch, A.F.; Poor, H.V.; Sahinoglu, Z. Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks. *IEEE Signal Process. Mag.* **2005**, *22*, 70–84. [CrossRef]
4. Yujin, C.; Ruizhi, C.; Mengyun, L.; Aoran, X.; Dewen, W.; Shuheng, Z. Indoor Visual Positioning Aided by CNN-Based Image Retrieval: Training-Free, 3D Modeling-Free. *Sensors* **2018**, *18*, 2692.
5. Mautz, R.; Tilch, S. Survey of optical indoor positioning systems. In Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation, Guimaraes, Portugal, 21–23 September 2011; pp. 1–7.
6. Jinhong, X.; Zhi, L.; Yang, Y.; Dan, L.; Xu, H. Comparison and analysis of indoor wireless positioning techniques. In Proceedings of the 2011 International Conference on Computer Science and Service System (CSSS), Nanjing, China, 27–29 June 2011; pp. 293–296.
7. Yang, A.; Yang, X.; Wu, W.; Liu, H.; Zhuansun, Y. Research on Feature Extraction of Tumor Image Based on Convolutional Neural Network. *IEEE Access* **2019**, *7*, 24204–24213. [CrossRef]
8. Jiang, J.; Ma, Q.; Lu, T.; Wang, Z.; Ma, J. Feature Matching Based on Top K Rank Similarity. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2316–2320.
9. Mentzer, N.; Payá-Vayá, G.; Blume, H. Analyzing the Performance-Hardware Trade-off of an ASIP-based SIFT Feature Extraction. *J. Signal Process. Syst.* **2016**, *85*, 83–99. [CrossRef]
10. Ebrahim, K.; Siva, P.; Shehata, M. Image matching using SIFT, SURF, BRIEF and ORB. Performance comparison for distorted images. In Proceedings of the 2015 Newfoundland Electrical and Computer Engineering Conference, St. johns, NL, Canada, 5–6 November 2015.
11. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
12. Yu, H.; Fu, Q.; Yang, Z.; Tan, L.; Sun, W.; Sun, M. Robust Robot Pose Estimation for Challenging Scenes With an RGB-D Camera. *IEEE Sens. J.* **2019**, *19*, 2217–2229. [CrossRef]
13. Shan, Y.; Li, S. Descriptor Matching for a Discrete Spherical Image With a Convolutional Neural Network. *IEEE Access* **2018**, *6*, 20748–20755. [CrossRef]
14. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In Proceedings of the Computer Vision—ECCV 2006, Berlin/Heidelberg, Germany, 7–13 May 2006; pp. 404–417.
15. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630. [CrossRef] [PubMed]
16. Luo, J.; Gwun, O. A comparison of SIFT, PCA-SIFT and SURF. *Int. J. Image Process. (IJIP)* **2009**, *3*, 143–152.
17. Srinivasa, K.G.; Shree Devi, B.N. GPU Based N-Gram String Matching Algorithm with Score Table Approach for String Searching in Many Documents. *J. Inst. Eng. (India) Ser. B* **2017**, *98*, 467–476. [CrossRef]
18. Wang, R.; Wan, W.; Di, K.; Chen, R.; Feng, X. A High-Accuracy Indoor-Positioning Method with Automated RGB-D Image Database Construction. *Remote Sens.* **2019**, *11*, 2572. [CrossRef]
19. Li, Z.; Cai, X.; Liu, Y.; Zhu, B. A Novel Gaussian–Bernoulli Based Convolutional Deep Belief Networks for Image Feature Extraction. *Neural Process. Lett.* **2019**, *49*, 305–319. [CrossRef]
20. Suju, D.A.; Jose, H. FLANN: Fast approximate nearest neighbour search algorithm for elucidating human-wildlife conflicts in forest areas. In Proceedings of the 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 16–18 March 2017; pp. 1–6.