applied sciences

Article

Line Segment Matching Fusing Local Gradient Order and Non-Local Structure Information

Weibo Cai [1,2,†], Jintao Cheng [1,2,†], Juncan Deng [1,2], Yubin Zhou [1,2], Hua Xiao [1,2], Jian Zhang [1,2,*] and Kaiqing Luo [1,2,*]

[1] Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials, School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou 510006, China; caiweibo@m.scnu.edu.cn (W.C.); chengjintao@unity-drive.com (J.C.); dengjuncan@m.scnu.edu.cn (J.D.); zhouyubin@m.scnu.edu.cn (Y.Z.); xiaoh@scnu.edu.cn (H.X.)

[2] Guangdong Provincial Engineering Research Center for Optoelectronic Instrument, School of Electronics and Information Engineering, South China Normal University, Guangzhou 510006, China

* Correspondence: jianzhang@m.scnu.edu.cn (J.Z.); kqluo@scnu.edu.cn (K.L.)

† These authors contributed equally to this work.

Abstract: Line segment matching is essential for industrial applications such as scene reconstruction, pattern recognition, and VSLAM. To achieve good performance under the scene with illumination changes, we propose a line segment matching method fusing local gradient order and non-local structure information. This method begins with intensity histogram multiple averaging being utilized for adaptive partitioning. After that, the line support region is divided into several sub-regions, and the whole image is divided into a few intervals. Then the sub-regions are encoded by local gradient order, and the intervals are encoded by non-local structure information of the relationship between the sampled points and the anchor points. Finally, two histograms of the encoded vectors are, respectively, normalized and cascaded. The proposed method was tested on the public datasets and compared with previous methods, which are the line-junction-line (LJL), the mean-standard deviation line descriptor (MSLD) and the line-point invariant (LPI). Experiments show that our approach has better performance than the representative methods in various scenes. Therefore, a tentative conclusion can be drawn that this method is robust and suitable for various illumination changes scenes.

Keywords: line segment matching; line feature; gradient order; non-local structure information

check for updates

Citation: Cai, W.; Cheng, J.; Deng, J.; Zhou, Y.; Xiao, H.; Zhang, J.; Luo, K. Line Segment Matching Fusing Local Gradient Order and Non-Local Structure Information. *Appl. Sci.* **2022**, *12*, 127. https://doi.org/10.3390/app12010127

Academic Editor: Hoon Yoo

Received: 3 November 2021
Accepted: 13 December 2021
Published: 23 December 2021

1. Introduction

Feature matching is important for many applications [1–5]. A typical matching method usually includes three steps. To be more specific, extract salient and stable features from the image first, then construct descriptors with the appearance of geometric features of the encoding neighborhood, and finally evaluate the correspondence between features by measuring the similarity between descriptors to achieve feature matching. At present, researches on point feature are most common. Compared with point features, there are more line features in industrial environments and indoor scenes. Moreover, line features contain more scene and object structure information, which can better reflect environmental information. Based on the information from line features, the structural details can be described more comprehensively, and the line features of the image can be supplemented [6]. Therefore, it is essential to further explore the characteristics of line features.

More and more researchers have made extensive research on the effective and reliable correspondence between lines in recent years. The latest matching algorithms fall into two categories.

The first type of matching algorithm uses individual lines to match line pairs. They use the local appearance, geometric features, etc. For instance, color is a typical appearance feature. The color histogram is used for generating a set of a line segment correspondence [7]. Gradient and intensity also are common local appearances. The mean-standard deviation line descriptor (MSLD) is constructed by counting the gradient vectors of each sub-region in the four directions of the pixel neighborhood. The method enables the length of descriptors fixed and improves the robustness of descriptors [8]. However, MSLD did not deal with the image of scale changes, resulting in its failure in scale changes image pairs. To overcome scale changes and segment fragmentation, the line band descriptor (LBD) uses the line segments extracted in the image pyramid [9]. LBD is similar to MSLD. The gradient mean and standard deviation of the four directions of the line band are calculated. Based on LBD, an optical flow method is introduced to reduce candidate matching line pairs [10]. On one hand, this method improves real-time performance. On the other hand, it reduces reliability under illumination change scenes. Recently, more and more attention has been paid to the illumination robustness of the line segment matching methods. There are two main methods: the methods based on intensity order [11,12] and the purely geometric ones [13]. The former introduces the local feature descriptor based on intensity order by constructing several concentric ring structures. The intensity order-based method has been proved to be effective. The latter is regularized using geometric constraints. This method improves the real-time performance and illumination robustness at the cost of reducing the matching line pairs. Another type of method that matches lines in individual incorporate point matching into line matching. Among them line-point invariant (LPI) is widely used to encode the information between the line and the neighboring point [14,15]. LPI is robust to the mismatched line, and it is extended to line matching across wide-baseline views [16]. However, those approaches fail when there is a lack of points in the scenes, such as low-textured scenes.

Another type of line segment matching method is matching in the group, which determines the corresponding line pairs by using the topological relationship and radiometric information [17–21]. Li et al. [17] propose a new dual-line matching method, which introduce the ray-point-ray (RPR) structure to describe line segment groups. To improve the matching accuracy of low-texture images under uncontrolled illumination, Lopez et al. [18] study a two-view line matching algorithm by combining geometric characteristics of lines. Hyunwoo Kim et al. [19] use the intersection context of a common plane line pair to match the line. Kim and Lee [20] determine the corresponding line pairs by using geometric attributes. The line-junction-line (LJL) gradient descriptor of the local region with the junction point as the center can be constructed by using the intersecting point and local texture information. Li et al. [21] implement line matching under the multi-layer Gaussian image pyramid based on the LJL gradient descriptor. For those unmatched line segments, Li et al. utilize the local homology estimated by its neighboring matched LJLs to match them. Compared with matching methods based on individual line segments, the group-based method can obtain a better correspondence relationship. Based on the studies of LJL, Chen et al. propose a method to match hierarchical line segments in huge viewpoint change cases [22]. However, the calculation process is still complex and requires a large number of computing resources.

Inspired by MSLD, LBD and previous studies [10–12], we design an alternative method, matching the individual lines by using the descriptor. Most of the previous approaches are focused on the regular line support region (LSR) and local appearance. Those methods are unable to describe the order of the line neighborhood pixel and the interactions between long-range pixel and a local neighborhood. Therefore, the line segment matching method is proposed. The approach is fusing the local and non-local structure information by exploring local gradient order, local intensity sequence information, global intensity sequence information, and non-local structure information. There are our main contributions:

(1) We use the line support region intensity histogram to perform adaptive intensity partitioning. The sub-regions are determined by intensity order, which increases the distance between descriptors of different line segments and will not affect the real-time performance;

(2) We use the local gradient order to describe the line segment. The local gradient order changes very little when the illumination changes and rotates. This means that the local gradient order of the same line segment has a high similarity in the images of different scenes;

(3) We fuse local gradient order information and non-local structural information of the line segment. Non-local structural information is not easily affected by image transformation. In addition, the sampling center information neglected in the local sampling process is supplemented. We fuse that information in an attempt to improve the matching performance in various scenes.

These improvements ensure certain real-time performance and excellent matching performance of the line segments matching method.

2. Approach Overview

Figure 1 is the flowchart of the approach. Feature lines in the image pyramid are extracted by the EDLines [23]. For the same line in different octave images, a vector is used to represent them. Based on this, the line support regions are determined. Using the intensity histogram of the line support region, the sub-regions are obtained by adaptive partitioning. The pixels in each sub-region are grouped and sampled in a way that the corresponding index of the sampling points could be obtained through their local gradient order. Then, corresponding index position of the histogram is voted, and the histograms of different partitions and groups are cascaded together to get the local gradient order histogram. As for anchor points, they are calculated by utilizing the global image. Thereafter, the histogram of non-local structural information is obtained by encoding the structural information between the sampled points and the anchor points. Next, these two histograms are normalized separately and then cascaded to obtain the final line descriptor. The nearest neighbor distance ratio (NNDR) algorithm is implemented to get the candidate matching line pairs. Finally, the adjacency matrix [9,24] is constructed by making use of the geometric properties and descriptor of the line segment. Then the greedy algorithm is used [25] and final match results will be available.

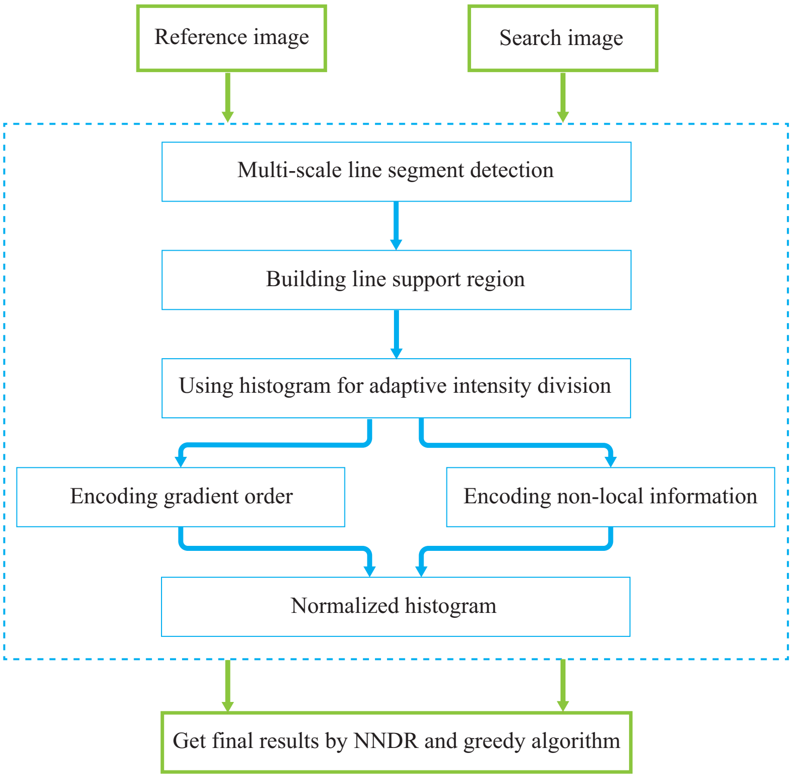

Reference image
Search image
Multi-scale line segment detection
Building line support region
Using histogram for adaptive intensity division
Encoding gradient order
Encoding non-local information
Normalized histogram
Get final results by NNDR and greedy algorithm

Figure 1. The proposed approach flowchart.

3. Methodologies

3.1. Line Support Region

For the line segments extracted from octave images by EDLines, building a line support region for them is necessary. Similar to MSLD [8] and LBD [9], the line support region is designed as a rectangle. The average gradient direction defines the local coordinate system (defined as d_0) and the counter-clockwise orthogonal direction (defined as d_L) of the pixels on the line segment. The length of the LSR is defined as L. The midpoint is selected as the origin of the coordinate system. In addition, the width of the LSR is defined as h (subsequent experiments will determine this parameter). The gradient in the line support region is converted to the gradient in the local region. The line support region generation demonstration is shown in Figure 2.

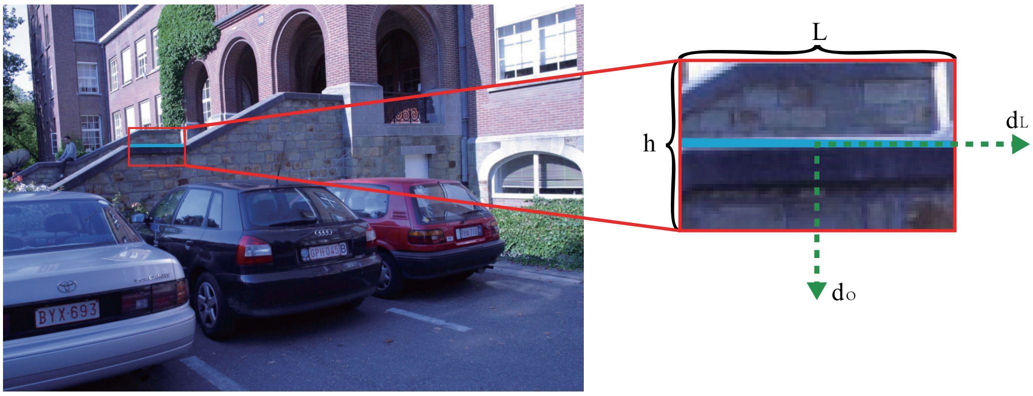


Figure 2. Line support region generation demonstration.

3.2. Adaptive Intensity Partition

Many popular and advanced line segment description methods adopt a geometric division strategy to divide the supported region into several fixed and regular sub-regions, such as the division based on line band—LBD [9]. However, this strategy underutilized intensity information. Xing et al. [11,12] used intensity order to partition and obtained certain robustness. However, their method suffers from two major problems:

(1) Using sequential partitioning requires a lot of sorting operations;

(2) When the intensity value is excessively concentrated in a specific value, it will lead to uneven partition and significant partition change when the illumination changes.

The present study adopts the intensity histogram for adaptive intensity partition to address these two problems.

Ideally, if one wants to divide N pixels into B parts, each part will have N/B pixels. However, in natural images, pixel intensity is often stacked at a specific value, making it challenging to achieve uniform partitions. Therefore, multiple homogenization is adopted to get relatively uniform partitions adaptively.

Assume that there are N pixels in the line support region, which is expressed as $\Omega = \{x_1, x_2, x_3, \cdots, x_N\}$. Their intensity is expressed as $I = \{I_{x_1}, I_{x_2}, I_{x_2}, \cdots, I_{x_N}\}$. The pixel intensity in the support region is traversed to get its intensity histogram as

$$H_I = \{H_0, H_1, H_2, \cdots, H_{255}\} \tag{1}$$

where $H_0, H_1, H_2, \cdots, H_{255}$ represent the value is the number of occurrences of each intensity, and H_I meet $H_0 + H_1 + H_2 + \cdots + H_{255} = N$. If there are B sub-regions divided by the LSR, it is necessary to select $B-1$ intensity values as thresholds, and set these $B-1$ intensity thresholds as $T_k, k \in [1, B-1]$. First, for the first threshold T_1, the cumulative pixel number of intensity histograms H_0 to H_{T_1} is close to $\frac{N}{B}$. For the second threshold T_2, the cumulative pixel number of intensity histograms H_{T_1+1} to H_{T_2} is close to $\frac{(N-\sum_0^{T_1} H_I)}{(B-1)}$. For the kth threshold T_k, the cumulative pixel number of intensity histograms $H_{T_{k-1}+1}$ to

H_{T_k} is close to $\frac{(N-\Sigma_0^{T_k-1} H_I)}{(B-k+1)}$. The above process is iterated for $(B-1)$ times, and $(B-1)$ intensity thresholds are finally acquired. The calculation formula of the above process is represented as

$$\frac{\sum_{T_{k-1}+1}^{T_k} H_I}{\left\lfloor \frac{\left(N-\Sigma_0^{T_{k-1}} H_I\right)}{B-k+1} \right\rfloor} \approx 1 \tag{2}$$

where, in order for $k = 0$ to satisfy the formula, $T_0 = 0$ is set. According to the $(B-1)$ adaptive intensity thresholds, divide the pixels in the support region into B sub-regions. Then, we can define a mapping $\eta(x, T)$ to map all pixels x in LSR to the corresponding sub-region.

$$\eta(x, T) = \begin{cases} 1, & I(x) \leq T_1 \\ k, & T_{k-1} < I(x) \leq T_k, k = 2, 3, \cdots, B-1 \\ B, & I(x) > T_{B-1} \end{cases} \tag{3}$$

where the integer obtained by $\eta(x, T)$ is the index of the sub-region, Bin_k is represented as the kth sub-region. Figure 3 shows a schematic representation of this sub-region partition based on adaptive strength partitions, where each sub-region is colored differently. It can be seen that compared with the computational complexity of sorting the supporting region, the histogram method has lower computational complexity. The method of multiple equalizations makes the partition less affected by the illumination change.

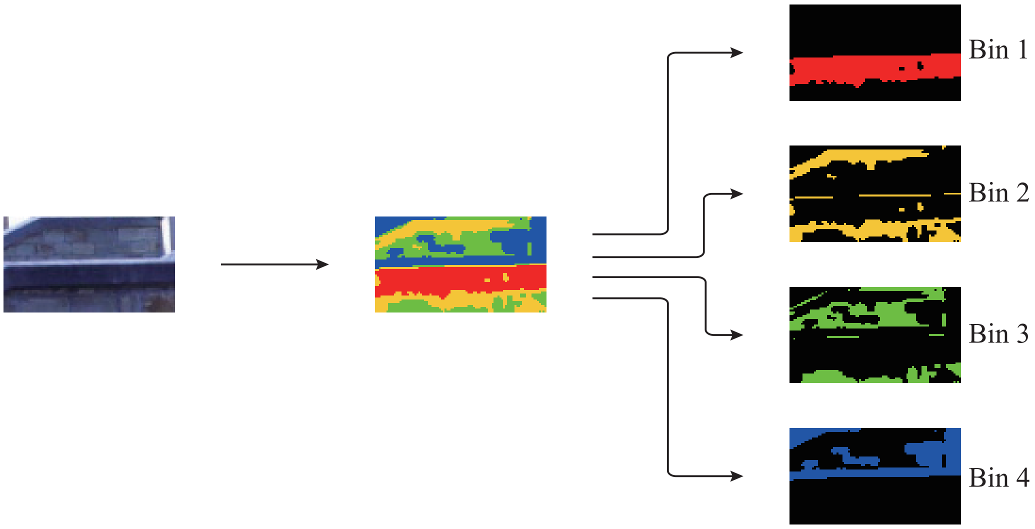


Figure 3. Demonstration of sub-region generation.

3.3. Local Gradient Order Encoding

Xing et al. [11] introduced the LIOP [26] descriptor into line feature matching and obtained certain illumination robustness, proving that the method based on intensity sorting is more robust to illumination change than the one based on direction estimation [8,9]. However, they only focus on the local intensity information of the LSR, ignoring the gradient information which can better describe the line segment. Song et al. [27] proposed a texture classification descriptor by sorting the intensity difference of the center and its neighboring pixels. Inspired by them, we combined the adaptive intensity partition to carry out local gradient order encoding for the sub-region of the line support region.

Before constructing the local gradient order descriptor, an index table is defined. Let Π^P be the set of permutations of P integers, which would have $P!$ elements in total. Next, number the elements in the collection Π^P in non-descending order to ensure that each arrangement has a unique corresponding index. Table 1 is an example.

Table 1. Index table.

Arrangement	Index
1, 2, 3	0
1, 3, 2	1
2, 1, 3	2
2, 3, 1	3
3, 1, 2	4
3, 2, 1	5

Before sampling, gradient information should be extracted. First, two Sobel operators are employed to process u direction and v direction of the image, respectively, to obtain u direction and v direction gradients g_u and g_v. To guarantee the invariance of the pixel gradient rotation, the gradients are projected under the original coordinate system of the image to the line support region coordinate system,

$$\mathrm{g}_{d_L} = g_u \times \cos_{d_L} + g_v \times \sin_{d_L} \tag{4}$$

$$\mathrm{g}_{d_O} = g_u \times \cos_{d_O} + g_v \times \sin_{d_O} \tag{5}$$

where g_{d_O} and g_{d_L} represent the gradient projected to the d_O direction and d_L direction under the local coordinate system of the support region, $\cos_{d_L}$ represents the cos value in the d_L direction, and the same for the other ones. Thereafter, the sum of g_{d_O} and g_{d_L} is calculated to obtain the gradient g under the local coordinate system of the line support region.

$$\mathrm{g} = g_{d_O} + g_{d_L} \tag{6}$$

In the latter, unless otherwise specified, all references to gradients refer to gradients in the local coordinate system of the line support region.

Considering a pixel $x_i(x_i \in \mathrm{Bin}_k)$ in the sub-region Bin_k, when constructing the feature descriptor of this pixel, the gradient order information of its neighborhood sampling points is needed. To achieve this goal, a neighborhood circle centered at this pixel with radius R is formed and P sampling pixels are selected within the circle, denoted as

$$\mathrm{G}(x_i) = \left(g_{x_i,0}, g_{x_i,1}, g_{x_i,2}, \cdots g_{x_i,P-1}\right) \tag{7}$$

where $\mathrm{G}(x_i)$ stores the gradient of the sampling points of pixel x_i, and $g_{x_i,p}$ is defined as the gradient of the pth sampling point of the pixel x_i. To reduce the dimensionality, divide these P sampling points into M groups with Q points within each group. (Q is limited to 3).

The starting point of sampling is defined as the position where the gradient of adjacent sampling points is maximum. This leaves the descriptor invariant to rotation When rotating the sampling sequence cyclically the point with the largest gradient is located at the first position. Then, the group corresponding to this sampling point will be the starting sampling group, thus forming a rotation-invariant sampling sequence.

$$\mathrm{G}_m(x_i) = \begin{cases} \left(g_{x_i,0}, g_{x_i,M}, g_{x_i,2M}, \cdots, g_{x_i,(Q-1)M}\right), & m = 1 \\ \left(g_{x_i,1}, g_{x_i,M+1}, g_{x_i,2M+1}, \cdots, g_{x_i,(Q-1)M+1}\right), & m = 2 \\ \vdots & \vdots \\ \left(g_{x_i,M-1}, g_{x_i,2M-1}, g_{x_i,3M-1}, \cdots, g_{x_i,P-1}\right), & m = M \end{cases} \tag{8}$$

where $\mathrm{G}_m(x_i)(m \in [1, M])$ represents the gradient value of the mth set of adjacent sampling pixels of the ith pixel in the sub-region Bin_k. The local gradient of each group of sampling points is sorted by order, and the resultant sequence is used as an index vector. The vector is converted to a unique integer by the index table. The corresponding histogram is obtained by this integer.

$$\mathrm{H}_m(x_i) = \left(0,\cdots,0, \underset{\mathrm{Ind}(\gamma(G_m(x_i)))}{1} ,0,\cdots,0 \right) \tag{9}$$

where $\gamma(\cdot)$ is defined to sort a sequence in non-descending order, $Ind(\cdot)$ is a mapping function. $Ind(\cdot)$ maps each index to the corresponding number. Figure 4 illustrates this process by taking $M = 3, Q = 3$.

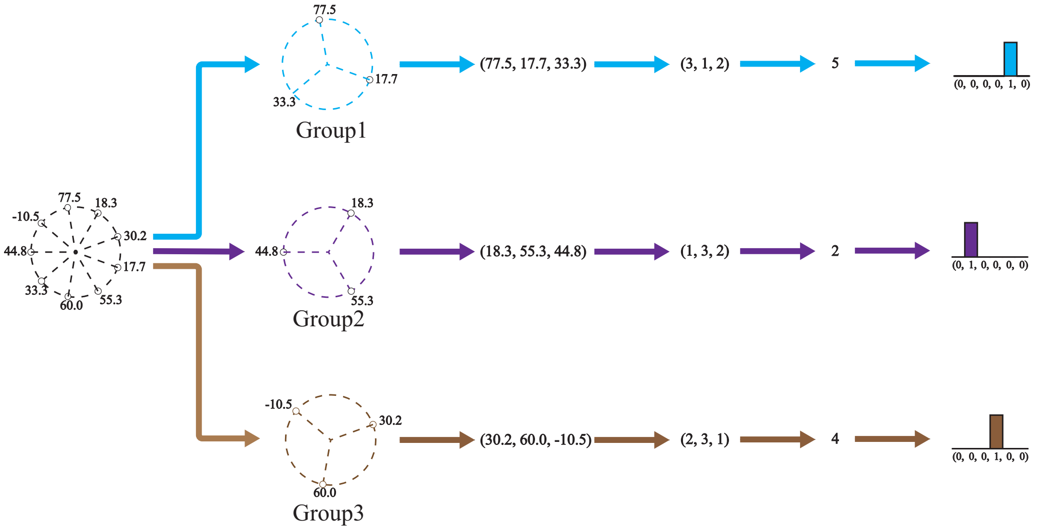


Figure 4. $H_m(x_i)$ Histogram construction demonstration.

After that, we repeated this procedure for all the pixels in Bin_k, and added up the histograms of each pixel to obtain:

$$\mathrm{des}_{k,m} = \sum w(x_i) H_m(x_i), x_i \in \mathrm{Bin}_k \tag{10}$$

where the dimension of the des_k is $Q! \times M$. Most dimensions are reduced compared to $P!$. The above process is carried out for all B sub-region to obtain the histograms of B sub-regions, and then the B histograms are cascaded together to obtain the final local gradient encoding descriptor of the line segment:

$$\mathrm{D}_{local} = (\mathrm{des}_1, \mathrm{des}_2, \mathrm{des}_3, \cdots, \mathrm{des}_B) \tag{11}$$

where D_{local} represents the local gradient order descriptor of the line segment, and its dimension is $Q! \times M \times B$. Figure 5 illustrates the calculation of a local gradient order descriptor.

The local gradient order encoding descriptor has the following characteristics: Firstly, the adaptive intensity partition is used to divide the line support region. Second, the local gradient order is used for encoding instead of the intensity order [11,28,29], which makes the descriptor more robust to illumination changes. Third, the encoding sequence is determined by the maximum value of the gradient of the sampling points to make the descriptors more discriminative.

3.4. Non-Local Structural Information Encoding

Mehta et al. [30], Fathi et al. [31], and Liu et al. [32] pointed out that non-uniformity is useful for describing some texture structures and Song et al. [29] used non-local structural information for encoding and obtained some resistance to noise. Most of the common methods focus only on the local information. For better robustness, the non-local structural information is encoded in this work. The non-local structure information encodes the inter-relationship between sampled pixels and pixels outside the line support region. For better adaptivity, several anchor points based on global intensity information are computed.

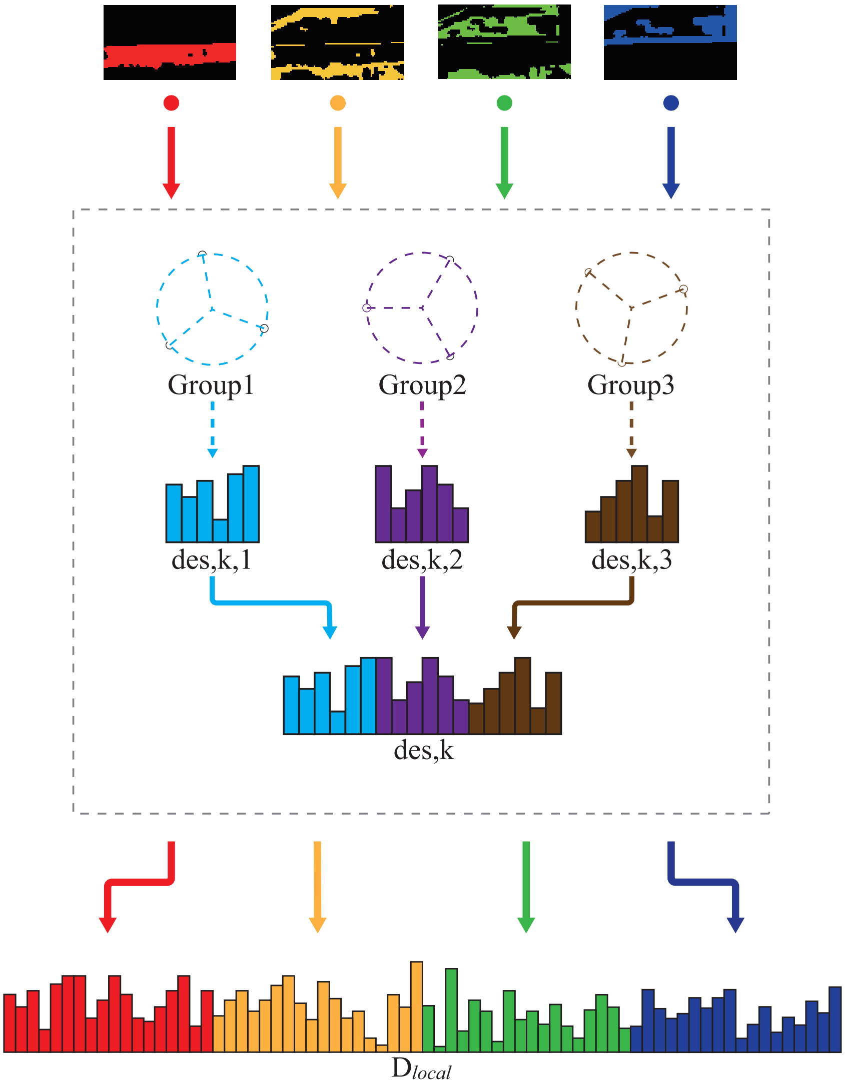


Figure 5. Demonstration of local gradient order descriptor construction process.

By encoding the intensity relationship between locally sampled points and non-local anchor points, the structural variation of line segments concerning the whole image is obtained. In this case, the structural information obtained based on the global image has better robustness compared to local information [27,29,33]. The local gradient order descriptor ignores the center of sample points, nevertheless, this central point information can be supplemented by encoding non-local information.

Song [29] used the method of sorting intensity to calculate anchor points. However, sorting intensity is computationally expensive. Therefore, the image intensity histogram proposed above is taken to calculate the anchor points.

Suppose that there are W pixels in the image, and the histogram is defined as $H_I = \{H_0, H_1, H_2, \cdots, H_{255}\}$. According to Equation (2), the intensity histogram of the image is divided into V intervals, and there are $V-1$ intensity thresholds defined as $T_k, k \in [1, V-1]$. Thereafter, calculate an anchor for each interval as follows:

$$\Delta_{Av} = \frac{1}{\lfloor W/V \rfloor} \sum_{I=T_{v-1}}^{I=T_v} H_I \times I \tag{12}$$

where I is intensity, $\Delta_{Av}(v\epsilon[1,V])$ denotes the intensity of the vth anchor. In order for $v=1$, $T_0=0$ is set. Figure 6 illustrates this process by taking $v=4, P=9$. The anchor points calculated by the intensity histogram are rotation-invariant. When the intensity of the image changes monotonically, the anchor points will also change.

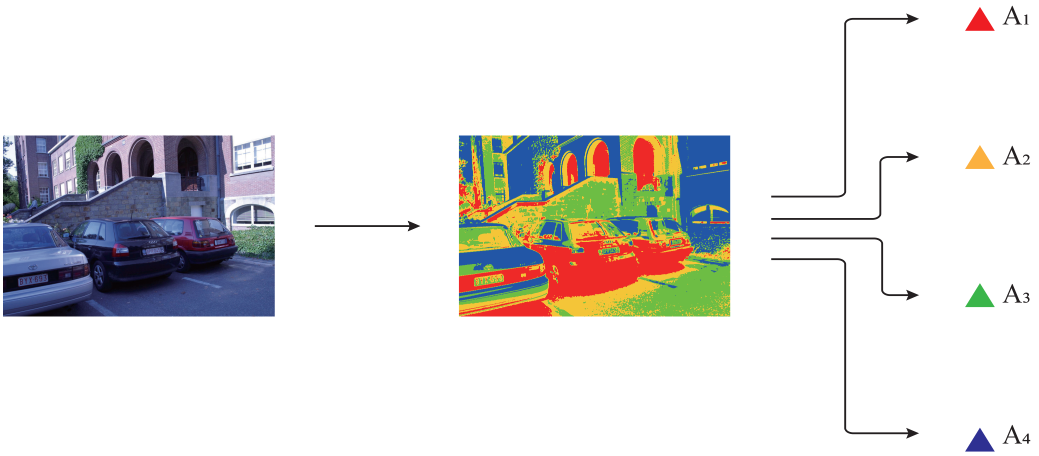


Figure 6. Demonstration of anchors generation.

Then, according to the relationship between the sampling point and the anchor point, the uniformity measurement U is obtained:

$$\begin{aligned} U(x_i) = &\left| s\left(I_{x_{i_p}} - \Delta_{Av}\right) - s\left(I_{x_{i_1}} - \Delta_{Av}\right) \right| \\ &+ \sum_{p=1}^{P-1} \left| s\left(I_{x_{i_p}} - \Delta_{Av}\right) - s\left(I_{x_{i_{p-1}}} - \Delta_{Av}\right) \right| \end{aligned} \tag{13}$$

where $U(x_i)$ is defined as the uniformity of a pixel x_i in the LSR. $I_{x_{i_p}}$ is the intensity value of the sampling point centered on x_i. $s(\cdot)$ is defined as

$$\mathrm{s}(a) = \begin{cases} 1, a \geq 0 \\ 0, a < 0 \end{cases} \tag{14}$$

In combination with uniformity measurement, encode the pixels in the line support region:

$$\mathrm{Index}_{i,v} = \begin{cases} \sum_{p=0}^{P-1} s\left(I_{x_{i_p}} - \Delta_{Av}\right), & U(x_i) \leq 2 \\ P+1, & U(x_i) = 4 \\ P+2, & U(x_i) = 6 \end{cases} \tag{15}$$

where $Index_{i,v}$ represents the index of the non-local structural information histogram of the pixel x_i. Then, the histogram corresponding to the vth anchor point is obtainable:

$$\mathrm{H}_v = \sum_{x_i \in \Omega} \left(0,\cdots,0, \underset{Index_{i,v}}{1} ,0,\cdots,0 \right) \tag{16}$$

The dimension of H_v is $P+3$, and then cascade the V histograms to obtain:

$$\mathrm{D}_{non-local} = (H_1, H_2, \cdots, H_V) \tag{17}$$

Figure 7 illustrates the calculation of a non-local descriptor.

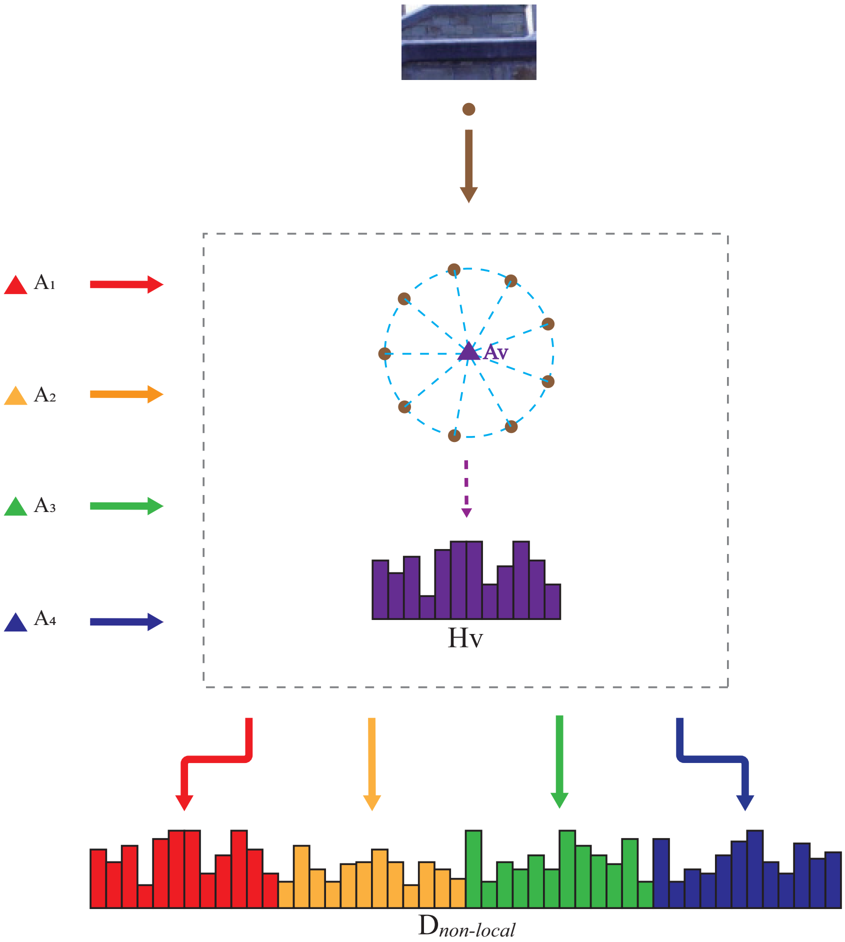


Figure 7. Demonstration of non-local descriptor construction process.

3.5. Normalized Histogram

D_{local} and $D_{non\text{-}local}$ encode the local gradient order information and the non-local structure information of the LSR, respectively, which are complementary to each other and could be cascaded together. Considering different sizes of them, D_{local} and $D_{non\text{-}local}$ are first normalized, respectively, to reduce non-linear interference, and then cascaded together to obtain the descriptor, which is described as

$$\mathrm{D} = (Normalizer(D_{local}), Normalizer(D_{non-local})) \tag{18}$$

In summary, the final dimensionality of the descriptor D is $Q! \times M \times B + (P+3) \times V$. Because $Q = P/M$, Q is limited to a small value, which also avoids the dimensionality explosion caused by stratification operations.

3.6. Generating Candidate Line Pairs and Obtain Final Result

Firstly, it is necessary to calculate the similarity of the line descriptors. The similarity between the two descriptors is determined by calculating the minimum Euclidean distance of the feature descriptor vector. It is worth noting that the line segments in different octaves should be considered when calculating the minimum distance because the same line segments in different octaves are extracted through the image pyramid during the line segment extraction process.

After that, the NNDR is adopted, which refers to the minimum distance divided by the maximum distance. If the ratio is less than a threshold, then the two line segments are regarded as a set of a candidate matching line. To guarantee the accuracy of matching, the threshold is set to a rough value. In this experiment, it is set to 0.7, which is an empirical value.

After screening, there will be some mismatches in the alternate matching line segments. To eliminate these mismatches, the cross-ratio, projection ratio, relative angle of the line segment are taken use of, and the minimum distance of the descriptor obtained in the last section is utilized to establish the link matrix [9,24]. Then solve the problem through a greedy algorithm [25] to obtain the final matching result.

4. Experiment

4.1. Experimental Datasets

Eight pairs of the typical image are selected to test the performance of the proposed approach as shown in Figure 8. All these images are taken from public datasets on the Internet and are often used in previous line matching studies [8,9,34,35]. They also include typical scenes: scale changes, rotation changes, viewpoint changes, occlusion, low textures, and illumination changes.

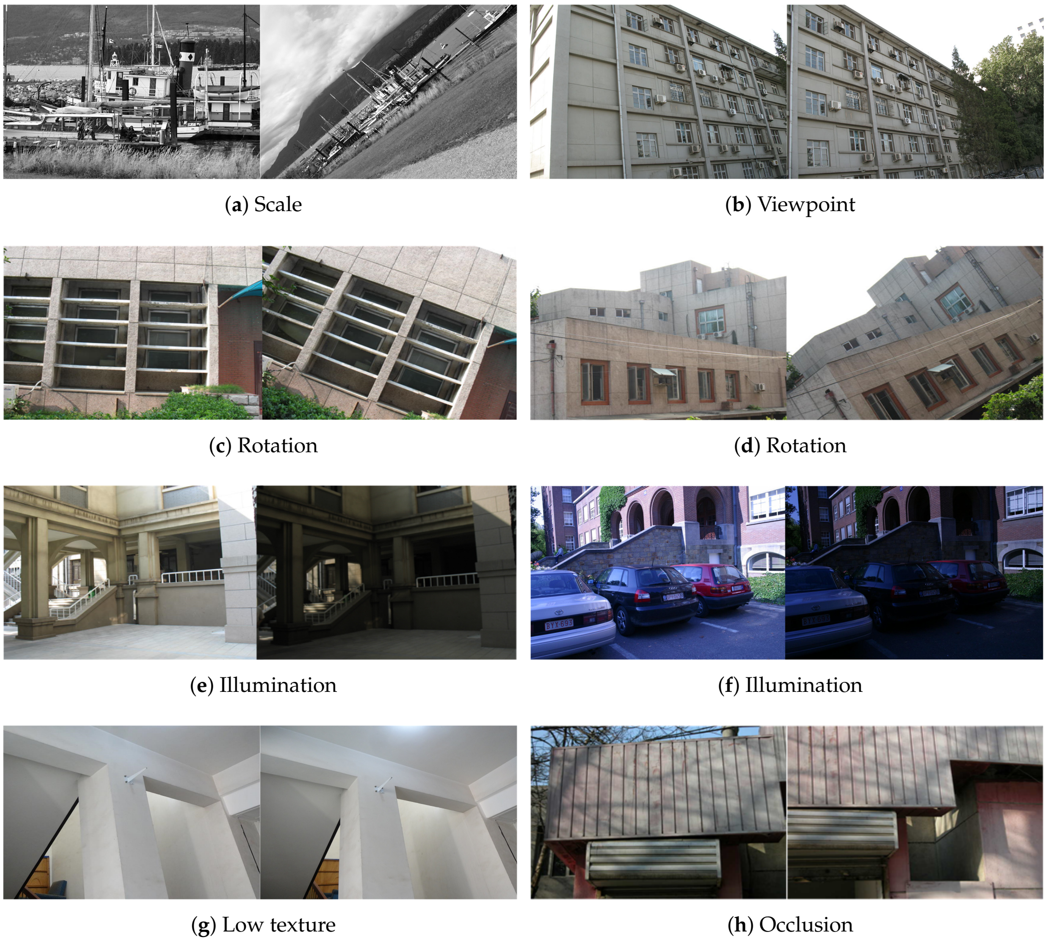

(**a**) Scale
(**b**) Viewpoint
(**c**) Rotation
(**d**) Rotation
(**e**) Illumination
(**f**) Illumination
(**g**) Low texture
(**h**) Occlusion

Figure 8. The datasets used in this experiment where (**a**) is the scale changes image pair, (**b**) is the Viewpoint change image pair, (**c**,**d**) are the rotation changes image pairs, (**e**,**f**) are the lighting change image pairs, (**g**) is the low-texture scene image pair, (**h**) is the occlusion image pair.

4.2. Evaluation of Parameters

In this section, the selection of parameters used in the proposed method through experiments will be discussed. The parameters including the height of the LSR, the number of sub-regions, the sampling radius, the number of groups, and the number of non-local anchors. The number of correctly matched is used to roughly measure the matching performance. The following experiment was performed on an Intel i5-8500 processor with 8 GB of RAM.

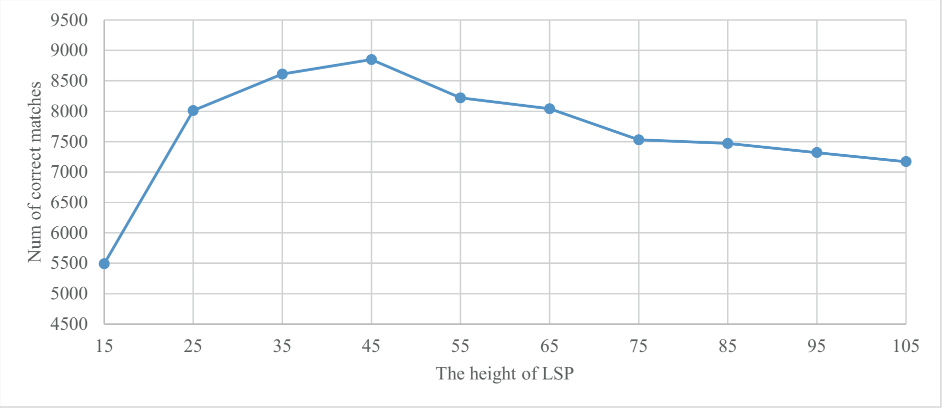
Figure 9a shows the influence of different line support region heights on matching performance. As h increases, the matching performance first increases and then decreases, reaching the maximum at $h = 45$.

Figure 9b is the influence of the number of sub-regions on matching performance. As B increases, the matching performance first increases and then decreases, and when $B = 4$, it reaches the maximum.

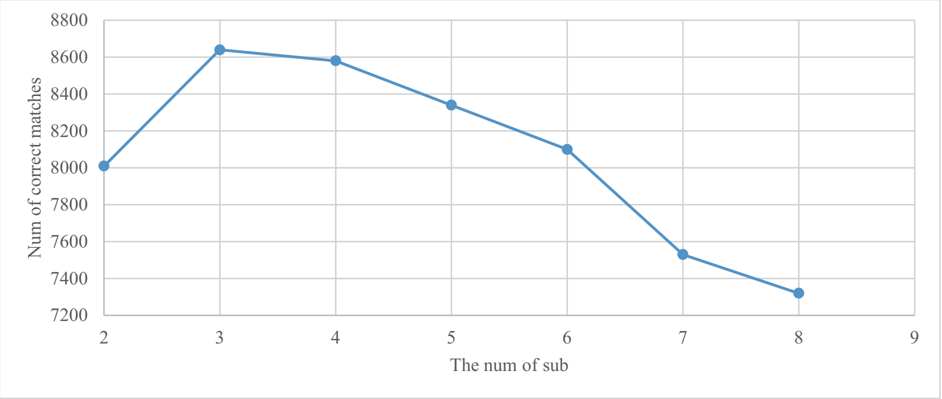
Figure 9c shows the influence of sampling radius on matching performance. When the sampling radius is too large, the performance decreases sharply, and it reaches the maximum when $R = 5$.

Figure 9d is the influence of the number of groups on matching performance. When the number of groups is greater than 3, the matching performance tends to be unchanged, so we choose $M = 3$.

Figure 9e shows the influence of anchor numbers on matching performance. When the anchor number is too large or too small, the matching performance is poor. When $V = 4$, it reaches the maximum.

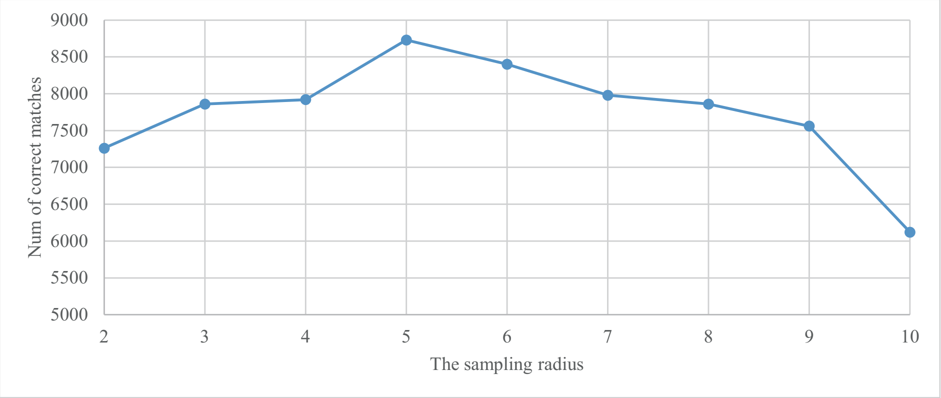


(**a**) The height of LSP

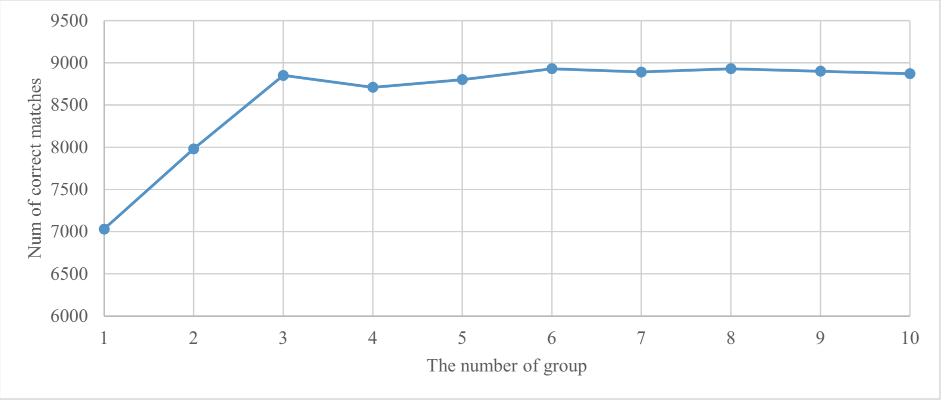


(**b**) The number of sub-regions

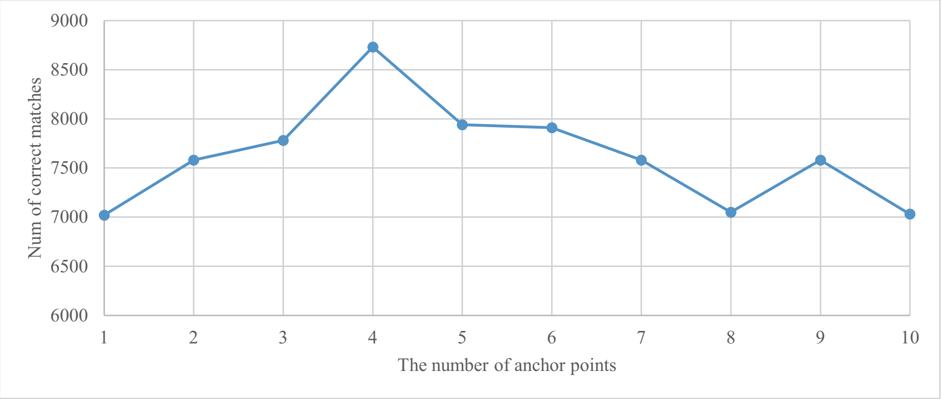
Figure 9. *Cont.*



(**c**) The length of sample radiuses



(**d**) The number of groups



(**e**) The number of anchors

Figure 9. Parameter selection experiment. (**a**–**e**), respectively, show the influence of the setting of h, B, R, M, and V on our experiment.

Informed by the experiment above experiment result, we determined the parameters used in the proposed method, which were summarized in Table 2.

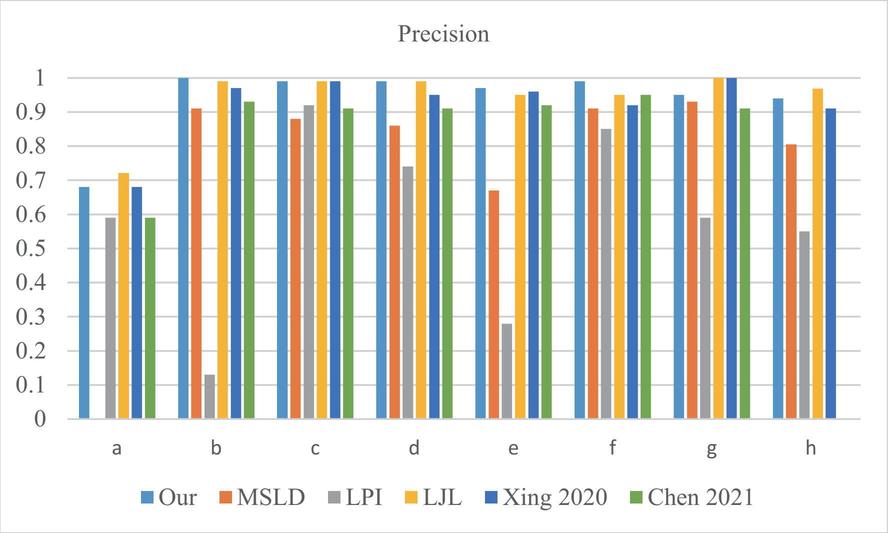
4.3. Comparative Experiments

In this section, the image pairs in Figure 10 are compared with MSLD [8], LPI [15], and LJL [20], whose codes are obtained on the Internet [36]. MSLD and LPI are both classical line segment matching methods. The framework based on MSLD is still being improved [9,11]. LPI is improved based on LP [14] proposed by the original paper author [15], and it was extended in 2016 [16]. Therefore, these two methods still have the value of research and are widely used in the comparative experiments of the latest papers. In addition, we compare the LJL [20]. It is another type of method that matches line segments in the group. The comparative experiments compare three different types of line segment matching methods. All the line segments used are extracted through EDLines [23].

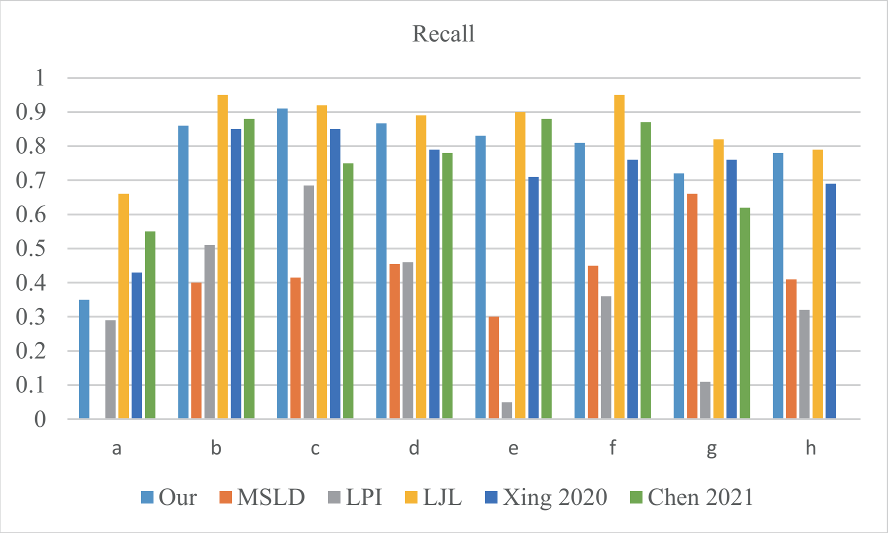
Table 2. Parameters summary.

Parameter	Description	Value
h	the length of LSR	45
B	the number of sub-regions	4
R	the sampling radius	5
M	the number of groups	3
V	the number of anchor points	4

To evaluate line segment matching performance, three commonly used metrics are adopted: precision, recall, and F1-Measure.

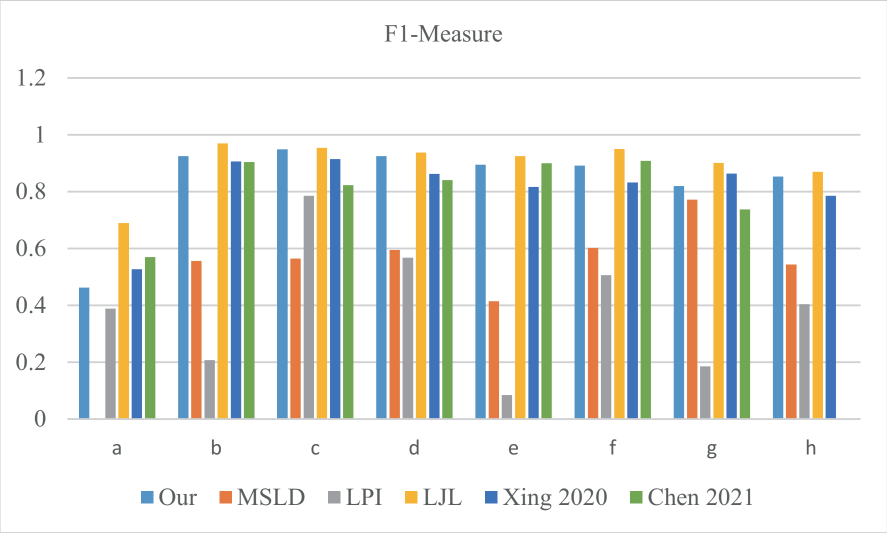


(**a**) precision



(**b**) recall

Figure 10. *Cont.*



(**c**) F1-Measure

Figure 10. (**a**–**c**), respectively, show the comparison of our method, MSLD, LPI, LJL, reference [11] (2020) and reference [22] (2021) in precision, recall, and F1-Measure. It is worth noting that results on the occlusion image pairs is missing from reference [22].

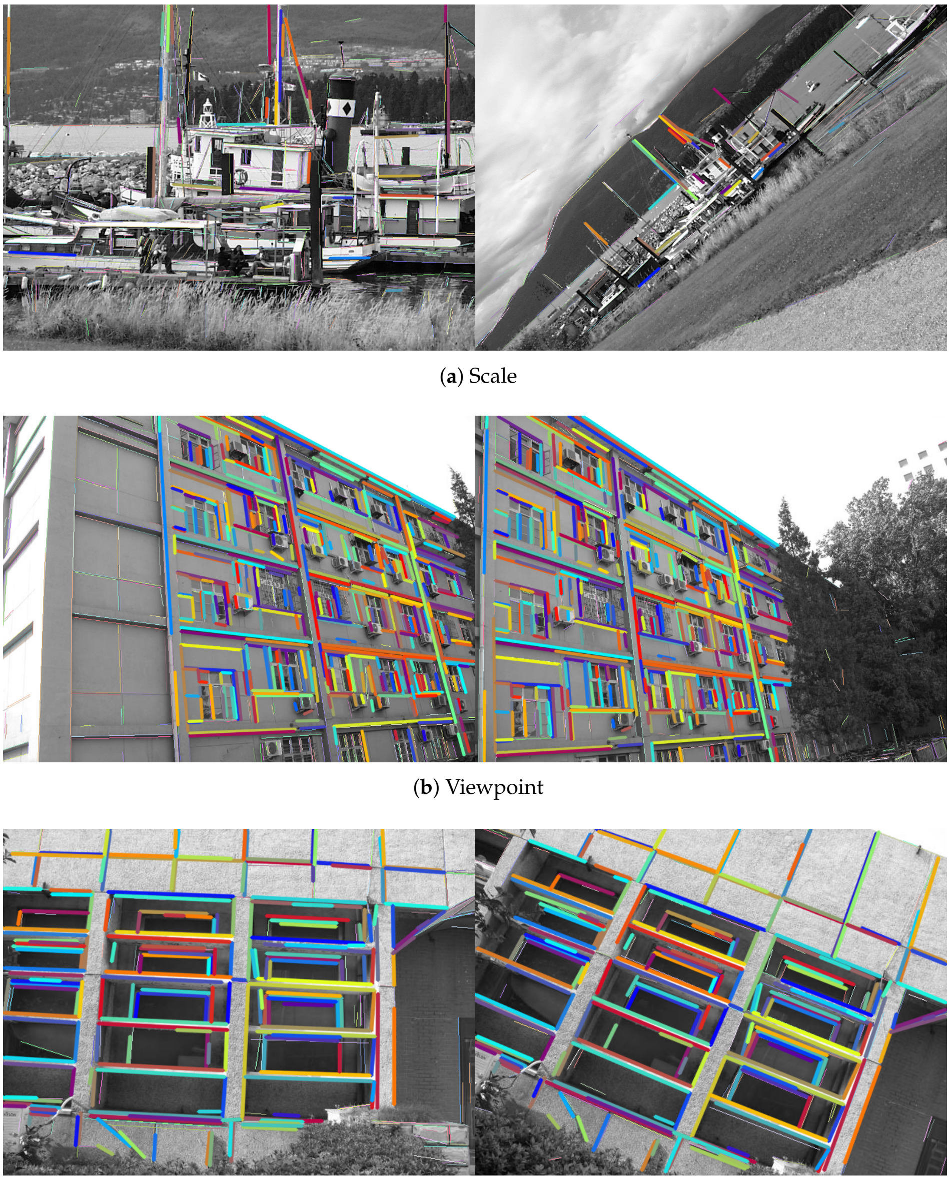
According to Figure 10c, the matching performance of the approach is close to LJL in most scenes. In the scene of large-scale change, the approach is lower than LJL. By analyzing Figure 10a,b, the precision of the proposed method is as same as LJL. However, the recall of the approach is lower than LJL under these scenes, which limits the comprehensive matching performance of our approach. Both LJL and the proposed approach use the same method that matches lines in the image pyramids. However, LJL matches line segments in groups and the proposed method matches line segments in individuals. It is difficult to keep the appearance of the line segments unchanged during large-scale change so that LJL obtains more matched line segments.

Figure 10 also shows our comparison with the other two methods. The proposed method outperforms MSLD and LPI in precision, recall, and F1-measure in eight pairs of images.

Figure 11c,d are image pairs with rotation. The matching precision of the proposed method is close to 1, while the recall is much higher than that of MSLD and LPI. This is mainly because our computation is based on a rotation-invariant local coordinate system. Secondly, our encoding order is based on the maximum gradient direction, which is also rotationally invariant. Finally, the adaptive sub-regions obtained using the intensity histogram are only intensity-dependent and are independent of image rotation.

In the illumination changes image pairs (Figure 11e,f), we still obtained an accuracy close to 1 and a high recall, with better performance than either of the other two methods. The main reason is that the gradient order does not change greatly when the illumination changes. MSLD based on direction estimation. Although illumination changes, its mean value will also change in different directions, as a result, more line segments cannot be correctly matched. LPI relies on line-point invariants. However, the endpoints of line segments extracted from image pairs will change when the illumination changes. This makes the invariants between line and point are unreliable. This unreliability is also evident in the image pairs of viewpoint changes and occlusion.

LJL, MSLD, and LPI are among the best and most classic open source codes. They are widely used in comparative experiments in the recent studies. However, in order to prove that our method is competitive with the latest paper, we select two latest line segment matching methods for comparison. References [11,22], they were published in 2020 and 2021. Unfortunately we do not have access to their source code, so the data we will use later are from their papers. Due to different environments, we did not compare real-time performance with these two methods.

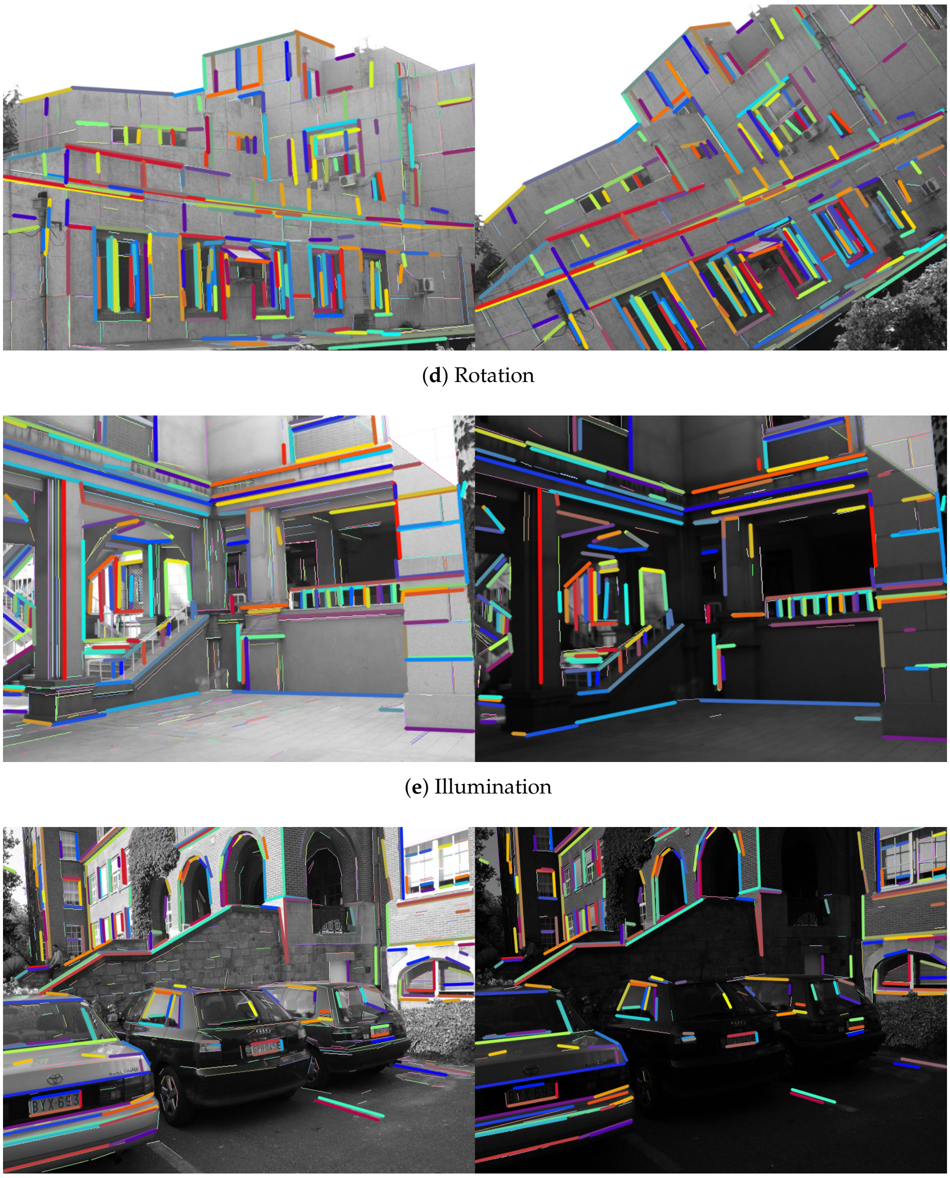


(**a**) Scale

(**b**) Viewpoint

(**c**) Rotation

Figure 11. *Cont.*

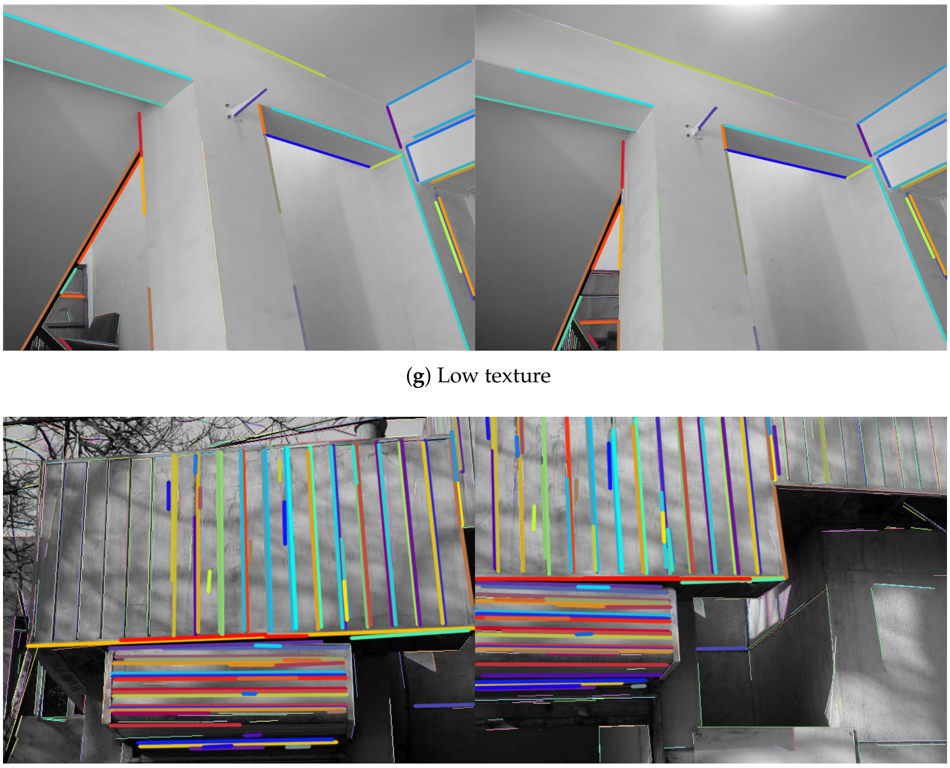


(**d**) Rotation

(**e**) Illumination

(**f**) Illumination

Figure 11. *Cont*.



(**g**) Low texture

(**h**) Occlusion

Figure 11. Lines of the same color in both figures represent matched lines. Those thin lines are extracted by EDLines but not successfully matched.

It can be seen from Figure 10 that the precision of the proposed method is higher than the other two latest methods in most image pairs. In the recall, our method is slightly lower than the other two methods in some scenes. However, in these image pairs (a, b, e, and f), our precision is higher than the other two methods. This is because we tend to set parameters for precision rather than quantity. Finally, according to F1-Measure comprehensive comparison, it can also be seen that the matching performance of our method is better than the other two latest methods in general.

4.4. Real-Time Performance

Table 3 shows the average time in milliseconds consumed per matched line pair. The average time consumed by the proposed method varies considerably over different image pairs. Among the four methods compared, LJL has the worst real-time performance. The main reason is that matching line segments in a group involves a huge number of the corresponding relationship between line and line, line and junction. Although this improves its matching performance, the real-time performance is drastically reduced.

Compared with the other two methods matching line segments in individual, our real-time performance is close to MSLD and better than LPI.

In the two image pairs in Figure 11d,f, the proposed method consumes the least average matching time per line segment. The encoding order of the proposed method is the maximum gradient direction, which is low computational cost and invariant when rotation and illumination change. Furthermore, the non-local structure information is the interactions between the line support region and the image interval, which is also relatively constant. Therefore, the proposed method is efficient for rotation and illumination scenes.

Table 3. Real-time performance comparison.

Image Pair	Our Method	MSLD	LPI	LJL
a	**22.6**	NULL	41.2	280.7
b	**2.4**	**2.4**	35.3	217.7
c	3.0	**2.5**	6.7	78.3
d	**1.7**	2.2	6.6	128.4
e	4.4	**2.8**	7.1	179.5
f	**2.2**	2.3	9.5	170.6
g	4.2	**1.2**	9.3	52.1
h	2.9	**2.5**	4.9	200.3

In the low texture image pairs (Figure 11g), the proposed method does also not perform well. The main reason is that the local appearance of line segments is to similar. It causes the gradient order of line segments to become unrecognizable, and more mismatched candidate line pairs cannot be found. This decreases the efficiency of the proposed approach. The real-time performance is close to MSLD and better than LJL and LPI in other scenes.

5. Conclusions

The present paper proposes a line segment matching method fusing intensity histogram adaptive partitioning, local gradient order information, and non-local structure information, in an attempt to match line features in various cases. The experiment shows that the designed method succeeded in improving the effectiveness of line segments matching in various scenes. The proposed method achieved higher scores in precision, recall, and F1-Measure than MSLD and LPI, especially in the cases of rotation and illumination changes. Our matching performance is slightly lower than that of LJL, but LJL's time cost is significantly higher than our method. In addition, compared with the latest methods (reference [11] (2020) and reference [22] (2021)), our method also has better matching performance in most scenes. Therefore, the proposed method not only ensures certain real-time performance, but also ensures excellent matching performance.

Author Contributions: Conceptualization, W.C. and K.L.; methodology, W.C. and J.C.; software, W.C. and J.C.; validation, W.C., J.C. and J.D.; formal analysis, J.C.; investigation, W.C.; resources, K.L.; data curation, J.D.; writing—original draft preparation, W.C. and J.C.; writing—review and editing, K.L., J.Z. and H.X.; visualization, Y.Z.; supervision, K.L.; project administration, K.L.; funding acquisition, K.L. and H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Science Foundation of China (NSFC)—Guangdong Big Data Science Center Project under Grant U1911401, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2015A030401086, and in part by the South China Normal University National Undergraduate Innovation and Entrepreneurship Training Program under Grant 202010574050 and 202110574048.

Data Availability Statement: The datasets used in this article and the code for MSLD, LPI, and LJL are available from this website: http://kailigo.github.io/projects/LineMatchingBenchmark (accessed on 11 December 2021).

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

References

1. Sharma, K.; Goyal, A. Classification based survey of image registration methods. In Proceedings of the International Conference on Computing, Communications and Networking Technologies, Tiruchengode, India, 4–6 July 2013; pp. 1–7. [CrossRef]
2. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287. [CrossRef] [PubMed]
3. Gomez-Ojeda, R.; Moreno, F.; Zuñiga-Noël, D.; Scaramuzza, D.; Gonzalez-Jimenez, J. PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments. *IEEE Trans. Robot.* **2019**, *35*, 734–746. [CrossRef]
4. Wei, H.; Tang, F.; Xu, Z.; Zhang, C.; Wu, Y. A Point-Line VIO System With Novel Feature Hybrids and With Novel Line Predicting-Matching. *IEEE Robot. Autom. Lett.* **2021**, *6*, 8681–8688. [CrossRef]

5. Li, D.; Liu, S.; Xiang, W.; Tan, Q.; Yuan, K.; Zhang, Z.; Hu, Y. A SLAM System Based on RGBD Image and Point-Line Feature. *IEEE Access* **2021**, *9*, 9012–9025. [CrossRef]
6. Schmid, C.; Zisserman, A. Automatic line matching across views. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997; pp. 666–671. [CrossRef]
7. Bay, H.; Ferraris, V. Wide-baseline stereo maching with line segments. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 329–336. [CrossRef]
8. Wang, Z.; Wu, F.; Hu, Z. MSLD: A robust descriptor for line matching. *Pattern Recogn.* **2009**, *42*, 941–953. [CrossRef]
9. Zhang, L.; Koch, R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *J. Vis. Commun. Image Represent.* **2013**, *24*, 794–805. [CrossRef]
10. Huang, L.; Chang, Q. Line segment matching of space target image sequence based on optical flow prediction. In Proceedings of the IEEE International Conference on Progress in Informatics and Computing PIC, Nanjing, China, 18–20 December 2015. [CrossRef]
11. Xing, J.; Wei, Z. A Line Matching Method Based on Multiple Intensity Ordering with Uniformly Spaced Sampling. *Sensors* **2020**, *20*, 1639. [CrossRef]
12. Xing, J.; Wei, Z.; Zhang, G. A robust line matching method based on local appearance descriptor and neighboring geometric attributes. In Proceedings of the SPIE Society of Photo-Optical Instrumentation Engineers, Beijing, China, 9–11 May 2016; Volume 10157. [CrossRef]
13. Gomez-Ojeda, R.; Gonzalez-Jimenez, J. Geometric-based Line Segment Tracking for HDR Stereo Sequences. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 69–74. [CrossRef]
14. Fan, B.; Wu, F.; Hu, Z. Line matching leveraged by point correspondences. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 390–397. [CrossRef]
15. Fan, B.; Wu, F.; Hu, Z. Robust line matching through line–point invariants. *Pattern Recogn.* **2011**, *45*, 794–805. [CrossRef]
16. Jia, Q.; Gao, X.; Fan, X. Novel Coplanar Line-Points Invariants for Robust Line Matching Across Views. *Lect. Notes Comput. Sci.* **2016**, *9911*, 599–611. [CrossRef]
17. Li, K.; Yao, J.; Lu, X. Robust Line Matching Based on Ray-Point-Ray Structure Descriptor. *Lect. Notes Comput. Sci.* **2015**, *9008*, 554–569. [CrossRef]
18. López, J.; Santos, R. Two-view line matching algorithm based on context and appearance in low-textured images. *Pattern Recognit.* **2015**, *48*, 2164–2184. [CrossRef]
19. Kim, H.; Lee, S. A novel line matching method based on intersection context. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 1014–1021. [CrossRef]
20. Kim, H.; Lee, S. Simultaneous line matching and epipolar geometry estimation based on the intersection context of coplanar line pairs. *Pattern Recogn. Lett.* **2012**, *33*, 1349–1363. [CrossRef]
21. Li, K.; Yao, J.; Lu, X. Hierarchical line matching based on Line–Junction–Line structure descriptor and local homography estimation. *Neurocomputing* **2016**, *184*, 207–220. [CrossRef]
22. Chen, M.; Yan, S.; Qin, R. Hierarchical line segment matching for wide-baseline images via exploiting viewpoint robust local structure and geometric constraints. *ISPRS J. Photogramm. Remote Sens.* **2021**, *181*, 48–66. [CrossRef]
23. Akinlar, C.; Topal, C. EDLines: A real-time line segment detector with a false detection control. *Pattern Recogn. Lett.* **2011**, *32*, 1633–1642. [CrossRef]
24. Wang, L.; Chen, B. Geometry consistency aware confidence evaluation for feature matching. *Image Vision Comput.* **2020**, *103*, 103984. [CrossRef]
25. Leordeanu, M.; Hebert, M. A spectral technique for correspondence problems using pairwise constraints. In Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, 17–20 October 2005; Volume II, pp. 1482–1489. [CrossRef]
26. Wang, Z.; Fan, B.; Wu, F. Local Intensity Order Pattern for feature description. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 603–610. [CrossRef]
27. Song, T.; Xin, L.; Gao, C.; Zhang, G.; Zhang, T. Grayscale-Inversion and Rotation Invariant Texture Description Using Sorted Local Gradient Pattern. *IEEE Signal Proc. Lett.* **2018**, *25*, 625–629. [CrossRef]
28. Wang, Z.; Fan, B.; Wang, G.; Wu, F. Exploring Local and Overall Ordinal Information for Robust Feature Description. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2198–2211. [CrossRef]
29. Song, T.; Feng, J.; Luo, L. Robust Texture Description Using Local Grouped Order Pattern and Non-Local Binary Pattern. *IEEE Trans. Circ. Syst. Vid.* **2021**, *31*, 189–202. [CrossRef]
30. Mehta, R.; Egiazarian, K. Dominant Rotated Local Binary Patterns (DRLBP) for texture classification. *Pattern Recogn. Lett.* **2016**, *71*, 16–22. [CrossRef]
31. Fathi, A.; Naghsh-Nilchi, A.R. Noise tolerant local binary pattern operator for efficient texture analysis. *Pattern Recogn. Lett.* **2012**, *33*, 1093–1100. [CrossRef]
32. Liu, L.; Fieguth, P.; Pietikainen, M.; Lao, S. Median robust extended local binary pattern for texture classification. *IEEE Trans. Image Process.* **2016**, *25*, 1368–1381. [CrossRef] [PubMed]
33. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]

34. Wei, D.; Zhang, Y.; Li, C. Robust line segment matching via reweighted random walks on the homography graph. *Pattern Recogn.* **2020**, *111*, 107693. [CrossRef]
35. Wang, J.; Zhu, Q.; Liu, S.; Wang, W. Robust line feature matching based on pair-wise geometric constraints and matching redundancy. *ISPRS J. Photogramm. Remote Sens.* **2020**, *172*, 41–58. [CrossRef]
36. Li, K.; Yao, J.; Lu, M. Line segment matching: A benchmark. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Placid, NY, USA, 7–10 March 2016. Available online: http://kailigo.github.io/projects/LineMatchingBenchmark (accessed on 30 May 2021).