

Feature Matching with Improved SIRB using RANSAC

Prof. K.K. Sharma

Received: 12 December 2020 Accepted: 4 January 2021 Published: 15 January 2021

Abstract

In this paper, we suggest to improve the SIRB (SIFT (Scale-Invariant Feature Transform) and ORB (Oriented FAST and Rotated BRIEF)) algorithm by incorporating RANSAC to enhance the matching performance. We use multi-scale space to extract the features which are impervious to scale, rotation, and affine variations. Then the SIFT algorithm generates feature points and passes the interest points to the ORB algorithm. The ORB algorithm generates an ORB descriptor where Hamming distance matches the feature points. We propose to use RANSAC (Random Sample Consensus) to cut down on both the inliers in the form of noise and outliers drastically, to cut down on the computational time taken by the algorithm. This postprocessing step removes redundant key points and noises. This computationally effective and accurate algorithm can also be used in handheld devices where their limited GPU acceleration is not able to compensate for the computationally expensive algorithms like SIFT and SURF. Experimental results advocate that the proposed algorithm achieves good matching, improves efficiency, and makes the feature point matching more accurate with scale in-variance taken into consideration.

Index terms—

1 Introduction

eature extraction and the matching continue to be the cornerstone of computer vision technologies. Essentially in feature detection, we abstract the image information into numbers and make a local decision at every point to see if there exists an image feature at that point. These local decisions taken at multiple spots are what we call interest points or key points. Ideally, techniques used for key point detection and matching should be impervious to different changes like -rotation, scale variance, illumination changes, noise and perspective changes.

Another desirable characteristic of these algorithms is that the key points generated should be unique to a greater degree to match a single feature with high rates of success.

The process of feature detection and matching can be broken down into some essential steps:

1. Detecting Interest points -It is done based on the brightness of the image or on the boundary extraction method. 2. Description of interest points -We generate a description vector for each feature point that

Author ? ? ? ? ¥: e-mails: meetpalod18@gmail.com, manasj514@gmail.com, amberjain.98@gmail.com, viroangrawat@gmail.com, kkssgs@gmail.com feature point that is invariant under changes in illumination, translation, scale, and in-plane rotation. 3. Matching the interest points across images -We match similar features across the images and map them, establishing a connection between two similar images. Now, these general steps can be implemented using distinct Algorithms. To this extent, numerous algorithms have been proposed. The SIFT algorithm has been historically the most popular algorithm due to its application in several fields using visual features like object recognition, image stitching, visual mapping etc. The overhead of computational burden that accompanies SIFT led to an intensive search for its replacement.

Real-time systems and low-power devices like cell phones cannot bear that computational burden. SIFT generally performs better than SURF in computational cost and is the most accurate featurdetector-descriptor for scale, rotation and affine variations.

44 SURF detects more features than the SIFT algorithm and the features are detected in a scattered form
45 generally all over the image. SURF has the highest computational cost for feature matching. Using SURF on
46 low-end devices was not possible, and so a faster and more efficient algorithm was required.

47 ORB detects the highest number of features that are more concentrated on corners. In all practical fields
48 of comparison ORB performs better than SURF and SIFT or any other algorithm for that matter. It is
49 computationally the fastest and most efficient.

50 SIRB is the latest addition to this arsenal of keypoint detection and matching algorithms. The SIRB algorithm
51 conflates the accuracy of the SIFT algorithm and combines it with the computationally efficient algorithm of
52 ORB, thus making it fast as well as accurate while also being unaffected by scale, rotation and affine variations.
53 We have proposed to take this one step further and improve on the existing SIRB(SIFT+ORB) algorithm using
54 RANSAC(Random sample consensus). The idea behind using RANSAC is to reduce the number of interest
55 points and thus eliminating the non matching points. This can potentially reduce more than 50% of the points
56 and because the number of points to be matched have been reduced we can considerably reduce the matching
57 time.

58 (Scale-Invariant Feature Transform) and ORB (Oriented FAST and Rotated BRIEF)) algorithm by incorpo-
59 rating RANSAC to enhance the matching performance. We use multi-scale space to extract the features which are
60 impervious to scale, rotation, and affine variations. Then the SIFT algorithm generates feature points and passes
61 the interest points to the ORB algorithm. The ORB algorithm generates an ORB descriptor where Hamming
62 distance matches the feature points. We propose to use RANSAC (Random Sample Consensus) to cut down on
63 both the inliers in the form of noise and outliers drastically, to cut down on the computational time taken by the
64 algorithm. This post-processing step removes redundant key points and noises. This computationally effective
65 and accurate algorithm can also be used in handheld devices where their limited GPU acceleration is not able to
66 compensate for the computationally expensive algorithms like SIFT and SURF. Experimental results advocate
67 that the proposed algorithm achieves good matching, improves efficiency, and makes the feature point matching
68 more accurate with scale in-variance taken into consideration.

69 The RANSAC algorithm removes outliers or the data which does not fit the model. Its application in feature
70 matching comes from the fact that it can handle more than 50% of the data as outliers, if applicable, and that has
71 a drastic effect on time needed to generate a result. It is worth mentioning that RANSAC is a nondeterministic
72 algorithm and that the results produced are reasonable only with a certain probability. This probability increases
73 with the number of iterations. In RANSAC, we broadly perform these two steps iteratively:

74 1. Select a Minimal Sample Set(MSS) randomly from the input dataset and generate a model based on only
75 these elements. 2. We now test which elements of the entire dataset are consistent with the model generated
76 using the MSS. The set of elements which fit the model constitute a consensus set.

77 This process terminates when the probability of finding a better consensus set drops below a certain threshold.

78 The proposed method will reduce the number of points to be matched and offer a better alternative to the
79 existing SIRB algorithm.

80 A feature or a keypoint is the distinctive piece of information, which is used for image matching, image
81 stitching, and image registration. In an image, features are the specific structures like edges, corners, or objects.
82 We detect corner points from an image which are also called interest points. An interest point in an image has
83 a well defined position. These features extracted through the SIFT algorithm are invariant to rotation, scaling,
84 and partially to illumination.

85 Keypoints are important because no matter how the image changes, we find the same key points in the modified
86 image when comparing with the original image.

87 To detect keypoints, SIFT starts by generating the scale space for the image. Image at different scales is
88 blurred by convolving a Gaussian kernel.

89 Scale-space is divided into octaves. In the original paper, Lowe [1] suggested that 4 octaves and 5 blur levels
90 at each octave are ideal for the algorithm.

91 Each octave's image size is half of the previous octave and each image within an octave is increasingly blurred
92 by a factor of k . The image is progressively blurred within an octave.

93 Mathematically, blurring is convolving the Gaussian operator and the image.

94 Here, G is the Gaussian kernel, I is an image, x, y are the location coordinates and defines the amount of blur.
95 A bigger value of σ implies more blur. Now with the blurred images, we create another set of images, known as
96 Difference of Gaussians (DoG). This DoG images are used for finding keypoints in the image. It is obtained as
97 the difference of two progressively blurred images in an octave. This process is done for all octaves of the image
98 in the scale space, and the desired scale space is obtained. It is represented in below image:

99 2 Local Maxima/Minima Detection

100 For computing local maxima or minima in the Difference of Gaussian, each point is compared to its eight neighbors
101 in the image, 9 in the scale above and 9 in the scale below. Pixel is considered as a keypoint if its pixel intensity
102 is more than all of the 26 neighboring points. Also, it is computationally efficient, because points are eliminated
103 in the first few inspections.

3 III. Key Point Localization Removing Keypoints with Low Contrast

After finding a possible keypoint by comparing it to neighboring pixels, the next task is to find datapoint location, scale, and ratio of principal curvatures. This process allows us to reject data points with low contrast which is sensitive to noise or localized along an edge. After it we fit a 3D quadratic function to the sample points to resolve the interpolated location of maximum and minimum. This approach uses Taylor expansion (2nd order) of the scale-space function, $D(x, y, z)$. $L(x, y, z) = G(x, y, z) * I(x, y, z)$ $D(x) = D + \nabla D^T \nabla x + \frac{1}{2} \nabla x^T \nabla^2 D \nabla x$ (2)

Where D and the derivatives of D are calculated at the keypoint and $x=(x, y, z)$ T is the o set from this point. The correct location of keypoint extrema is x , and is evaluated by calculating the derivative of function $D(x)$ concerning x and placing it to zero.

The function value at extrema, (i.e. at x) $D(x)$ is useful for removing keypoints with low contrast. The value $D(x)$ can be calculated by substituting value of x in $D(x)$.

Values less than 0.3 for the function value $D(x)$ are rejected. A final trial for removing key points/feature points on edges are performed because these are unnecessary points and will create redundant key points in the matching process. A poorly defined extrema located at the ridge in the DoG, which depicts the edge in the image has broader principle curvature across the ridge and a low one along with it which is not present in a blob or (corner) having broad principle curvature along with both directions. Principle of curvature is evaluated by Hessian Matrix H .

The eigenvalues corresponding to the Hessian matrix are tantamount to principle curvature. The ratio of eigenvalues λ_1 and λ_2 (which is proportional to principle curvature), are calculated with the help of Hessian matrix (H). This ratio is compared to certain threshold ratio r and high ratio points are repudiated. Where $r = \lambda_1 / \lambda_2$.

Apply homogeneity to equations. The value of the term $(r+1)^2 / r$ is minimum when both eigenvalues are equal, and it gradually increases with the value of r . Hence to check whether the ratio is below certain value, we used

We use the value of $r = 10$ in this paper. Feature points above this threshold are repudiated.

4 IV.

5 Orientation Assignment

Assigning orientation to keypoints ensures that it can be represented relative to its orientation and thus rotation invariance is achieved. To assign orientation, Histogram of Oriented Gradient (HOG) is used.

The scale of the keypoint is used to select the Gaussian smoothed image. Then, a 16x16 square window is considered around the detected keypoint.

For each pixel, $L(x, y)$, at this scale, the gradient magnitude, $m(x, y)$, and edge orientation, (x, y) , is precomputed using pixel differences:

The edges below a threshold gradient magnitude are considered as weak edges and discarded. From the surviving edges, an orientation histogram is created. It has 36 bins covering the 360degree range of orientations. Each pixel orientation added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a that is 1.5 times that of the scale of the keypoint.

Peaks in HOG correspond to the dominant orientation of the keypoint. If other peaks are detected within 80 percent of the dominant peak, then multiple keypoints are generated at the same location and scale with different orientation. These multiple orientation keypoint notably enhance the stability of matching.

V.

6 rBRIEF: Rotation-Aware Briefs

After detecting all the key points, descriptors are generated for each keypoint. Feature descriptors encode unique information into a series of numbers and differentiate one feature from another. In this paper, we use the rBRIEF algorithm, a rotation-invariant version of the BRIEF algorithm. The BRIEF descriptor was proposed by M. Calonder [9]. BRIEF is an acronym for Binary robust independent elementary features, and it is robust to photometric and geometric image transformation.

The defined neighborhood, which is a square of some pixel width and height, around the keypoint is known as a patch. Brief depends on intensity difference tests to represent an image patch as a binary vector. Image patches could be effectively classified based on a relatively small number of pairwise intensity comparisons. $x = ?$ $?^2 D^2 ?^1 ?^2 \nabla D^2 \nabla x$ (3) $D(x) = D + \nabla D^T \nabla x + \frac{1}{2} \nabla x^T \nabla^2 D \nabla x$ (4) $x \times x \times x \times H = D \times x \times D \times y \times D \times y \times D \times y \times y$ (5) $T \text{r}(H) = D \times x \times D \times y \times y = ?^1 + ?^2$ (6) $\text{Det}(H) = D \times x \times D \times y \times y \times (D \times x \times x)^2 = ?^1 \times ?^2$ (7) $T \text{r}(H)^2 \text{Det}(H) = (?^1 + ?^2)^2 \times ?^2 = (r^2 + ?^2)^2 \times r^2 = (r + 1)^2 \times r$ (8) $T \text{r}(H)^2 \text{Det}(H) = (?^1 + ?^2)^2 \times ?^2 \times ?^1 \times ?^2 < (r + 1)^2 \times r$ (9) $?(x, y) = \tan^{-1} ? L(x,y+1) ? L(x,y-1) ? L(x+1,y) ? L(x-1,y)$ (10)

x act like a numerical "fingerprint" that can be used to Fig. 1: Histogram of Gradient(HOG) [14] Each keypoint is described by a feature vector. Brief convert image patches into binary feature vectors, represent an object. The binary feature vector contains only 1's and 0's. Because BRIEF deals at the pixel level, it is very

noise-sensitive. Hence it starts by smoothing the image with a Gaussian kernel. This reduces the sensitivity and also increases the stability and repeatability of the descriptors by pre-smoothing the patch.

To generate a 256-bit vector, it defines a randomly generated set of 256 pairs of pixels. The first pair in the random pair is taken from a Gaussian distribution drawn around the keypoint with a standard deviation of $0.04 * S^2$ (where S is the dimension of the patch). The second pair in the random pair is drawn from a Gaussian distribution drawn around the first pixel(x) with a standard deviation of $0.01 * S^2$. rBRIEF achieves rotation invariance by moving these pixel locations by an angle equal to the orientation angle of the keypoint.

Then, it performs a binary test comparing the intensity of the first pixel in pair with the second pixel. If the intensity of the first pixel is less than the intensity of the second pixel, we append value 1 in the feature vector, else we append 0. When this binary test is performed on all the 256-pixel pairs in an image patch, we get a 256-bit binary vector, which is the desired descriptor.

7 VI.

RANSAC [8] algorithm is an extensively adapted mechanism in image processing for cleaning outliers from huge datasets. This algorithm helps in outlier removal, image noise filtration and optimal descriptors selection. RANSAC [6][7] does statistical estimations, which helps in evaluation for the likelihood of achieving accurate predictions.

RANSAC uses a minimal amount of data set to gain noise-repudiated data points, unlike other algorithms which initiate with huge datasets and then remove outliers.

After some formulations, N is chosen by the formula: $N = \frac{\log(1 - p)}{\log(1 - (1 - V)^M)}$ (13) Year 2021

8 F

Feature Matching with Improved SIRB using RANSAC) which are 128-512 bits string, so that together they can

9 Working of RANSAC:

? First, a minimum number of points are arbitrarily selected to define model parameters. From all the points, the points which fit according to the tolerance (ϵ) are determined. ? Then we calculate the fraction of the number of inliers to total points present in the set. ? If the fraction is greater than a threshold (τ), which is predefined, we reevaluate parameters of the model using inliers. ? Else, we reiterate from step 1 to step 4, at most N times.

In this procedure, the value of N is taken large enough so that the minimum of one of the sets of random samples does not include any outlier. P represents probability which is set to 0.99, U represents chances of a selected data point as an inlier, V represents chances of selected data point as an outlier and N times the minimum number of points denoted M are required, so VII.

10 Match Feature Points Via Hamming Distance

The binary feature descriptor generated makes the matching process computationally more efficient because these binary vectors are stored in the form of bits. Performing operations like XOR on these binary vectors are much faster because of the ability to quickly compare descriptor pairs using few processor-level instructions. Algorithms like SIFT/SURF use Euclidean distance to compare binary feature vectors. We found that using hamming distance [11] for matching the key points is more efficient when comparing binary feature vectors. Therefore, we have used Hamming distance for our research.

The hamming distance is the value of the number of positions where both binary vectors differ. It is denoted by $d(a, b)$, where a and b are two binary vectors. It can also be calculated by performing XOR operation on given two binary vectors.

In our methodology, we aim to improve performance of SIRB by using RANSAC. RANSAC has been proven highly proficient in removing mismatched inconsistent sets of points. Since SIRB incorporates SIFT and ORB, therefore RANSAC has an intrinsic role in boosting the overall performance of SIRB. Due to the inaccuracy of SIFT, some points generated by it are mapped incorrectly, but RANSAC helps in filtering out these points, making SIFT more efficient. ORB [5] is an essential feature descriptor in low-end devices. So, it becomes crucial to eliminate shortcomings of ORB [12] and hence RANSAC is used as a post-quality improvement step to remove outliers and redundant key-points. RANSAC, thus makes SIRB more efficacious than some pre-existing methods, consequently making it an overall vigorous image recognition set-up. We proposed a new algorithm, SIRB+RANSAC. It inherits the innate accuracy of the SIFT algorithm and the fast superiority of ORB. Its performance is further enhanced by using the RANSAC algorithm, which eliminates mismatches to a great extent. We start by describing the SIFT keypoint detection algorithm. We then introduced the rBrief method for binary descriptor generation. We used Hamming Distance to match the binary descriptors generated. Finally, we used the RANSAC algorithm to remove the mismatched keypoint pairs and eliminate the outliers.

The results of the experiment show that the new enhanced SIRB+RANSAC algorithm effectively resolves the problem that the ORB algorithm performs poorly in terms of scale invariance, and it is sensitive to image illumination. Based on the results of the experiment, we conclude that:

221 ? The average matching accuracy of our algorithm is 89.85% as compared to the 86.95% accuracy of
222 the traditional SIRB algorithm. The SIFT algorithm still presents an accuracy of 94.20%. ? The average
223 computational time of the new improved SIRB+RANSAC algorithm is 87.69ms, which is about 42 times faster
224 than the SIFT algorithm which takes an average of 3723ms. On the other hand, the SIRB algorithm takes
225 74.33ms.

226 The proposed algorithm will be exceptionally useful on low-end hand-held devices with low GPU power because
227 it involves relatively less computational expenses than the prevalent algorithms. It also has applications in
228 traditional systems where SIFT or SURF is currently being used.

229 11 IX.

230 12 Future Work

231 It is evident from experimental results that the average matching time for SIRB is significantly less when compared
232 with the SIFT algorithm. However, the matching accuracy has also dropped by 4.61%. We have released the
233 source code of our algorithm so that it can be used by other researchers. There is the scope for improvement in
234 our proposed algorithm, and these are the fields where future research can be directed:

235 ? Instead of using Gaussian filters for convolution, filters like average filter and median filters can be used to
236 reduce the run time of the algorithm while maintaining accuracy. ? Use improved RANSAC, modified RANSAC
237 or optimal RANSAC to remove parts of the error feature point, thus increasing the proportion of correct matching
features. ? Remove the repeatable and unstable key points generated after normalizing the scale space. ¹

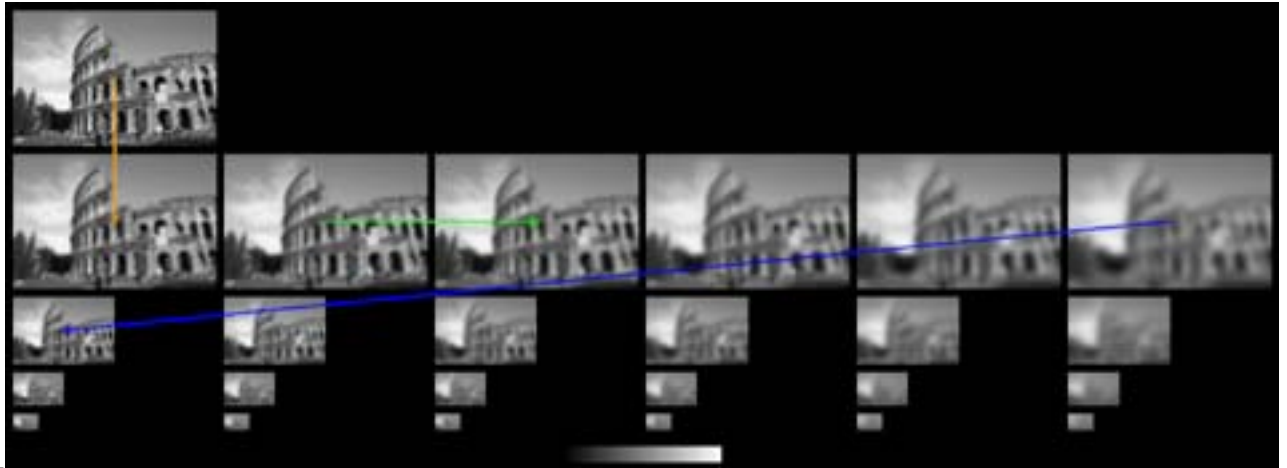
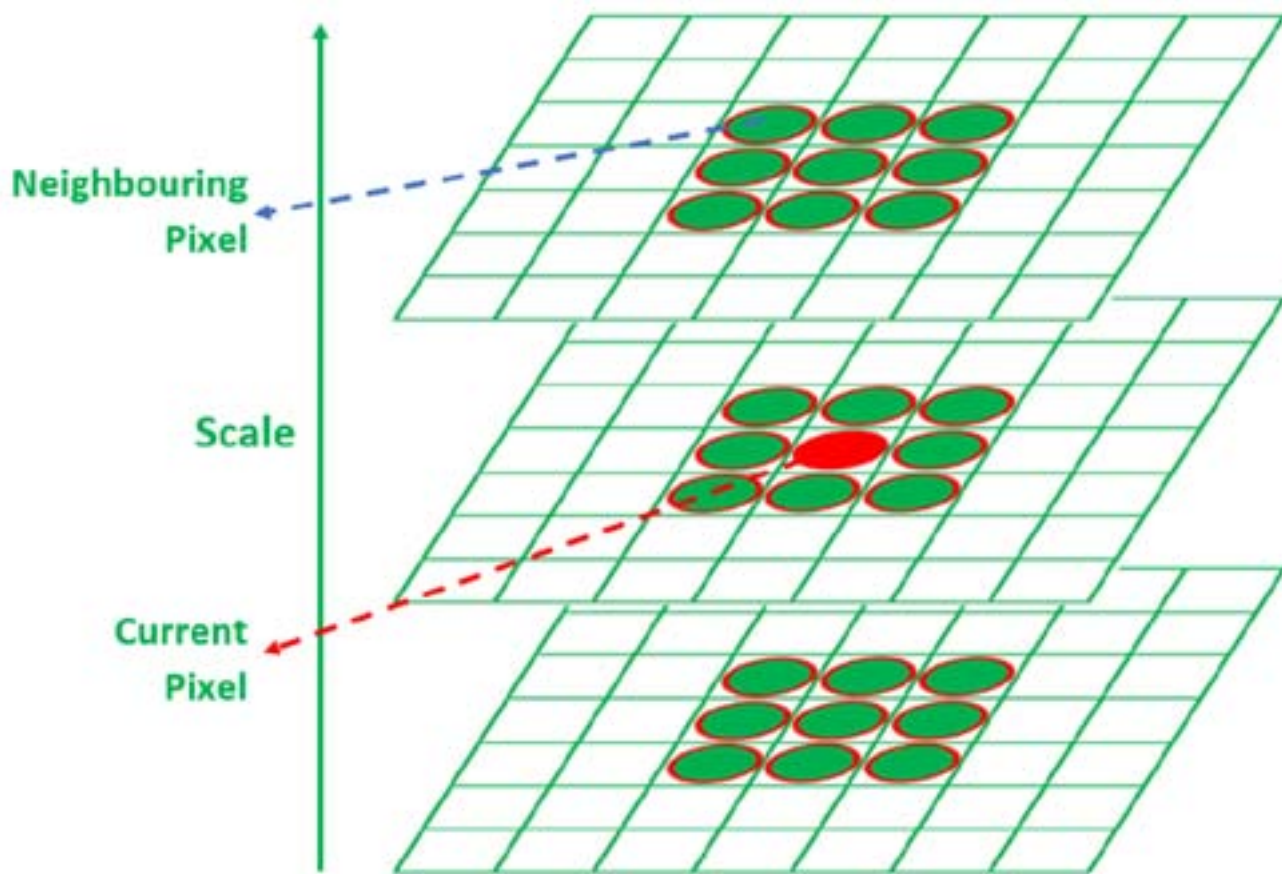


Figure 1: Fig. 1 :

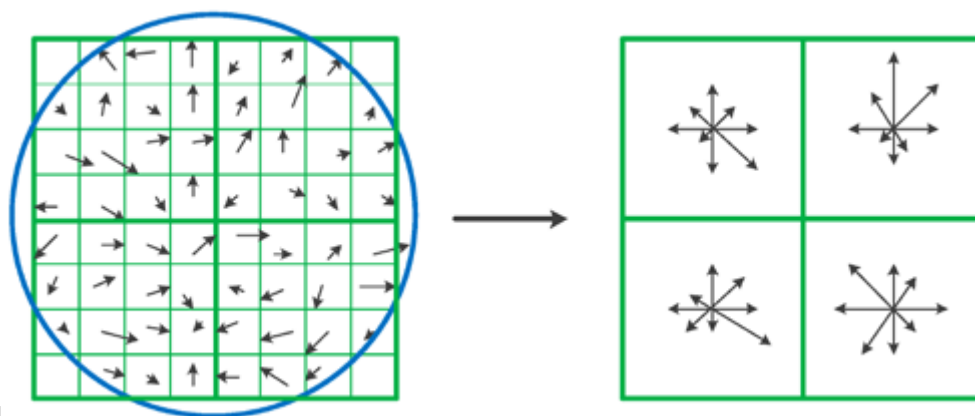
238

¹()Year 2021 F © 2021 Global JournalsFeature Matching with Improved SIRB using RANSAC



2

Figure 2: Fig. 2 :



4

Figure 3: Fig. 4 :

-
- 239 [Su et al. ()] ‘A Binary SIFT Matching Method Combined with the Color and Exposure Information’. M Su ,
240 Y Ma , X Zhang , S Li , Y Zhang . IC-NISC.2017.00062. *2017 International Conference on Network and*
241 *Information Systems for Computers (ICNISC)*, 2017.
- 242 [Yang et al. ()] ‘An Improved ORB Algorithm of Extracting Features Based on Local Region Adaptive Thresh-
243 old’. K Yang , D Yin , J Zhang , H Xiao , K Luo . *2019 6th International Conference on Systems and*
244 *Informatics (ICSAI)*, 2019, ICSAI48974. 2019.9010209.
- 245 [Calonder et al. ()] ‘BRIEF: Computing a Local Binary Descriptor Very Fast’. M Calonder , V Lepetit , M
246 Ozuysal , T Trzcinski , C Strecha , P Fua . TPAMI.2011.222. *IEEE Transactions on Pattern Analysis and*
247 *Machine Intelligence*, 2012.
- 248 [Lowe ()] ‘Distinctive Image Features from Scale-Invariant Keypoints’. D G Lowe .
249 10.1023/B:VISI.0000029664.99615.94. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
250 *International Journal of Computer Vision* 2004. 60 p. .
- 251 [He et al. ()] Wei He , Takayoshi Yamashita , Hongtao Lu , Shihong Lao , Tracking . ICCV.2009.5459360. *2009*
252 *IEEE 12th International Conference on Computer Vision*, 2009.
- 253 [Qin et al. ()] ‘Image feature points matching via improved ORB’. Y Qin , H Xu , H Chen . PIC.2014.6972325.
254 *2014 IEEE International Conference on Progress in Informatics and Computing*, 2014.
- 255 [Tian and Shi ()] ‘Image Mosaic using ORB descriptor and improved blending algorithm’. F Tian , P Shi . *2014*
256 *7th International Congress on Image and Signal Processing*, 2014. 2014.7003867.
- 257 [Leng and Yang ()] ‘Improvement of ORB algorithm’. Xue Leng , Jinhua Yang . 10.2991/meita-15.2015.169.
258 <https://doi.org/10.2991/meita-15.2015.169> *Proceedings of the 2015 International Conference on*
259 *Materials Engineering And Information Technology Applications*, (the 2015 International Conference on
260 *Materials Engineering And Information Technology Applications*) 2015.
- 261 [Lowe ()] ‘Object recognition from local scaleinvariant features’. D G Lowe . ICCV.1999.790410. *Proceedings of*
262 *the Seventh IEEE International Conference on Computer Vision*, (the Seventh IEEE International Conference
263 *on Computer Vision*) 1999.
- 264 [Chum and Matas ()] ‘Optimal Randomized RANSAC’. O Chum , J Matas . TPAMI.2007.70787. *IEEE Trans-*
265 *actions on Pattern Analysis and Machine Intelligence*, 2008.
- 266 [Rublee et al. ()] ‘ORB: An efficient alternative to SIFT or SURF’. E Rublee , V Rabaud , K Konolige , G
267 Bradski . *2011 International Conference on Computer Vision*, 2011. 2011.6126544.
- 268 [Yang and Li ()] ‘Outliers Elimination Based Ransac for Fundamental Matrix Estimation’. S Yang , B Li .
269 ICVRV.2013.63. *2013 International Conference on Virtual Reality and Visualization*, 2013.
- 270 [Zhao et al. ()] ‘Research on image matching based on improved RANSAC-SIFT algorithm’. M Zhao , H Chen
271 , T Song , S Deng . ICOCN.2017.8121270. *2017 16th International Conference on Optical Communications*
272 *and Networks (ICOCN)*, 2017.
- 273 [Zhang et al. ()] ‘RST Resilient Watermarking Scheme Based on DWT-SVD and Scale-Invariant Feature
274 Transform’. Y Zhang , C Wang , X Zhou . 10.3390/a10020041. <https://doi.org/10.3390/a10020041>
275 *Algorithms* 2017. 10 (2) p. 41.
- 276 [Shi et al. ()] ‘SIFT Feature Point Matching Based on Improved RANSAC Algorithm’. G Shi , X Xu , Y Dai .
277 IHMSC.2013.119. *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*,
278 2013.