

RESEARCH ARTICLE

# Image mosaicking using SURF features of line segments

Zhanlong Yang<sup>1,2\*</sup>, Dinggang Shen<sup>2,3</sup>, Pew-Thian Yap<sup>2</sup>

**1** School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, Shaanxi, China, **2** Department of Radiology and Biomedical Research Imaging Center (BRIC), University of North Carolina, Chapel Hill, NC, United States of America, **3** Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea

\* [zhanlongy@gmail.com](mailto:zhanlongy@gmail.com)



## Abstract

In this paper, we present a novel image mosaicking method that is based on Speeded-Up Robust Features (SURF) of line segments, aiming to achieve robustness to incident scaling, rotation, change in illumination, and significant affine distortion between images in a panoramic series. Our method involves 1) using a SURF detection operator to locate feature points; 2) rough matching using SURF features of directed line segments constructed via the feature points; and 3) eliminating incorrectly matched pairs using RANSAC (RANDOM SAMple Consensus). Experimental results confirm that our method results in high-quality panoramic mosaics that are superior to state-of-the-art methods.

## OPEN ACCESS

**Citation:** Yang Z, Shen D, Yap P-T (2017) Image mosaicking using SURF features of line segments. PLoS ONE 12(3): e0173627. <https://doi.org/10.1371/journal.pone.0173627>

**Editor:** Kewei Chen, Banner Alzheimer's Institute, UNITED STATES

**Received:** October 10, 2016

**Accepted:** February 22, 2017

**Published:** March 15, 2017

**Copyright:** © 2017 Yang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All relevant data are available within the manuscript and its Supporting Information file.

**Funding:** This work was partially supported by the National Natural Science Foundation of China (No. 61540047), the Northwestern Polytechnical University Foundation for Fundamental Research (NPU-FFRJC201226), and NIH grants (NS093842, EB006733, EB008374, and EB009634). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## 1 Introduction

The automatic construction of large, high-resolution image mosaics is an active area of research in the fields of photogrammetry, computer vision, image processing, and computer graphics [1]. It is considered as important as other image processing tasks such as image fusion [2], image denoising [3], image segmentation [4] and depth estimation [5]. Image mosaicking finds applications in a wide variety of areas. A typical application is the construction of large aerial and satellite images from collections of smaller photographs [1, 6]. More applications include scene stabilization and change detection [7], video compression [8], video indexing [9] and so on [1]. Some widely used commercial software packages for image mosaicking are available, such as AutoStitch [10], Microsoft ICE [11], and Panorama Maker [12].

The key problem in image mosaicking is to combine two or more images by stitching them seamlessly together into a new one that distorts the original images as little as possible [13]. Image mosaicking techniques can be mainly divided into two categories: grayscale-based methods and feature-based methods. Grayscale-based methods are easy to implement, but they are relatively sensitive to grayscale changes especially under variable lighting. Feature-based methods extract features from image pixel values. Because these features are partially invariant to lighting changes, matching ambiguity can be better resolved during image matching. Matching robustness can be further improved by using feature points that can be detected reliably. Many methods have been shown to be effective for the extraction of image feature

**Competing interests:** The authors have declared that no competing interests exist.

points, for example, Harris method [14], Susan method [15], and Shi-Tomasi method [16]. Feature-based image mosaicking methods afford two main advantages: (1) the computation complexity of image matching will be significantly reduced since the number of feature points is far smaller than the number of pixels; (2) the feature points are very robust to unbalanced lighting and noise, resulting in better image mosaicking results.

A wide variety of feature detectors and descriptors have been proposed in the literature (e.g. [17–21]). Detailed comparisons and evaluations of these detectors and descriptors on benchmark datasets were performed in [22, 23]. Among various methods, SIFT [18] has been shown to give the best performance [22]. Recent efforts (e.g. SURF [24], BRISK [25], FREAK [26], NESTED [27], and Ozuysal’s method [28]) have been focused on improving SIFT-based matching accuracy and reducing computation time. Arguably SURF [24] is among the best methods. Fei Lei et al. proposed a fast method for image mosaicking based on a simple application of SURF [29]. Jun Zhu et al. proposed an image mosaicking method that uses the Harris detector and SIFT features of line segments [30]. For performance and efficiency, this method uses Harris corner detection operator to detect key points. Then features of line segments are used to match feature points owing to their effective representation of local image information, such as textures and gradients. However, the Harris corner detector is very sensitive to changes in image scale; so it does not provide a good basis for matching images of different sizes. Motivated by this observation, we propose an image mosaicking method that is based on SURF features [24] of line segments. First, the method uses the SURF detection operator to locate feature points and then constructs a directed graph of the extracted points. Second, it describes directed line segments with SURF features and matches them to obtain rough matching of points. Finally, it adjusts matching points and eliminates incorrectly matched pairs through the RANSAC algorithm [31]. The framework of our method is summarized in Fig 1.

## 2 SURF

SURF, like the SIFT operator, is a robust feature detection method that is invariant to image scaling, rotation, illumination changes, and even substantial affine distortion. Both of these descriptors encode the distribution of pixel intensities in the neighborhoods of the detected points. SURF is computationally more efficient than SIFT owing to the use of integral images [32] and the box filters [33] that approximate second order partial derivatives of Gaussian convolutions. Similarly to many other approaches, SURF consists of two consecutive parts, including feature point detection and feature point description.

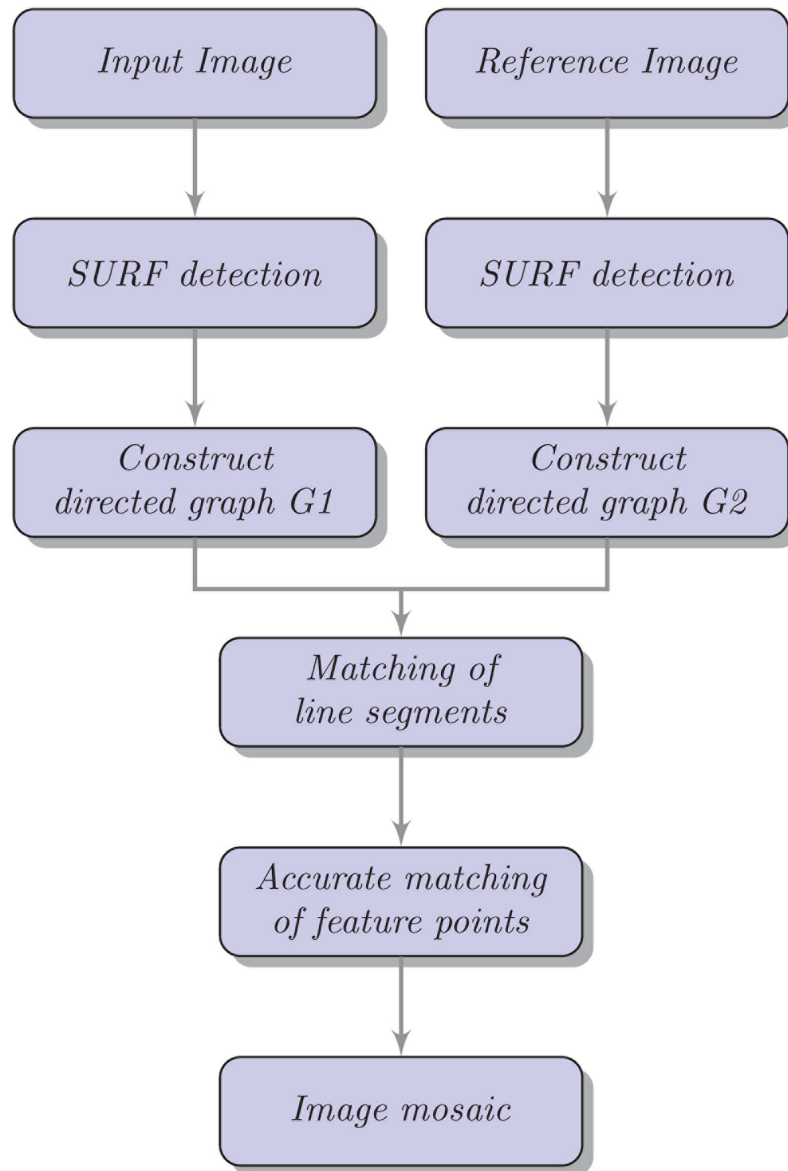
### 2.1 SURF feature-point detector

Similarly to the SIFT method, the detection of features in SURF relies on a scale-space representation combined with first and second order differential operators. The key feature of the SURF method is that these operations are approximated using box filters computed via integral images. So, the procedure of SURF feature detection involves first computing an integral image, establishing an image scale space with box filters, and finally locating feature points in the scale space.

The SIFT detector is based on the determinant of the Hessian matrix, which is defined at point  $\mathbf{x} = (x, y)$  and scale  $\sigma$  as

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \tag{1}$$

where  $L_{xx}(\mathbf{x}, \sigma)$  is the convolution of the Gaussian second order derivative  $\frac{\partial^2}{\partial x^2} g(\sigma)$  with the

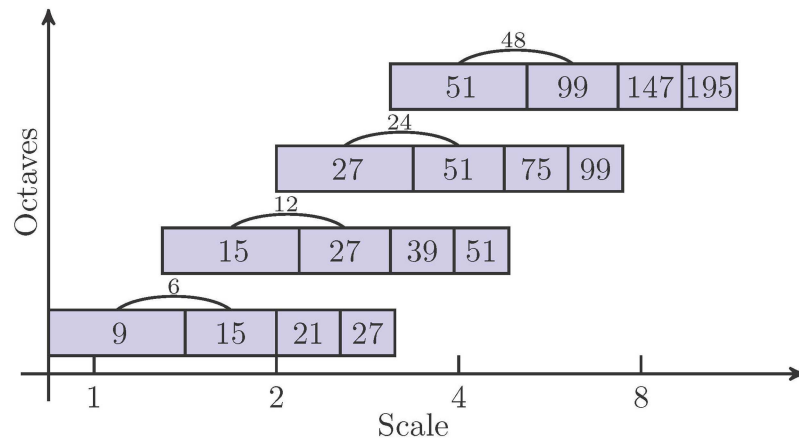


**Fig 1. An overview of our method.**

<https://doi.org/10.1371/journal.pone.0173627.g001>

image  $I$  at point  $\mathbf{x}$ , and similarly for  $L_{xy}(\mathbf{x}, \sigma)$  and  $L_{yy}(\mathbf{x}, \sigma)$ . As mentioned before, in order to reduce computation, SURF approximates  $L_{xx}$ ,  $L_{xy}$ ,  $L_{yy}$  with the box filtering using sum of the Haar wavelet responses, resulting respectively in  $D_{xx}$ ,  $D_{xy}$ ,  $D_{yy}$  and

$$H_{\text{approx}}(\mathbf{x}, \sigma) = \begin{bmatrix} D_{xx}(\mathbf{x}, \sigma) & D_{xy}(\mathbf{x}, \sigma) \\ D_{xy}(\mathbf{x}, \sigma) & D_{yy}(\mathbf{x}, \sigma) \end{bmatrix}. \quad (2)$$



**Fig 2. Filter sizes for four different octaves and intervals (marked by arcs).**

<https://doi.org/10.1371/journal.pone.0173627.g002>

This can be performed very efficiently using an integral image  $I_{\Sigma}$ , which given an input image  $I$  is calculated as

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j). \tag{3}$$

The determinant of the approximated Gaussians is

$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - (0.9D_{xy})^2. \tag{4}$$

Thus, the interest points, including their scales and locations, are detected in approximate Gaussian scale space. The size of the box filter is varied with octaves and intervals [34]:

$$\text{Filter Size} = 3 \times (2^{\text{octave}} \times \text{interval} + 1). \tag{5}$$

The filter sizes for various octaves and intervals are illustrated in Fig 2. Only pixels with greater responses than their surrounding pixels are classified as interest points. The maximal responses are then interpolated in scale and space to locate interest points with sub-pixel accuracy.

## 2.2 SURF descriptor

The goal of a descriptor is to provide a unique and robust description of the intensity distribution within the neighborhood of the point of interest. In order to achieve rotational invariance, the orientation of the point of interest needs to be determined. Orientation is calculated in a circular area of radius  $6s$  centered at the interest point, where  $s$  is the scale at which the interest point is detected. In this area, Haar wavelet responses in  $x$  and  $y$  directions are calculated and weighted with a Gaussian centered at the point of interest. By computing the sum of the horizontal and vertical responses within a sliding orientation window of size  $\pi/3$  and traversing the entire circle every 5 degrees, 72 orientations can be obtained. The two summed responses then yield a local orientation vector. The longest of such vector over all windows defines the main orientation.

Once position, scale and orientation are determined, a feature descriptor is computed. The first step consists of constructing a square region centered around the feature point and

oriented along the orientation determined previously. The region is divided uniformly into smaller  $4 \times 4$  sub-regions. For each sub-region, Haar wavelet responses are computed at  $5 \times 5$  regularly-spaced sample points. The  $x$  and  $y$  wavelet responses, denoted by  $dx$  and  $dy$  respectively, are computed at these sample points weighting with a Gaussian centered at the interest point and summed up over each sub-region to form a first set of entries to the feature vector. In order to obtain information on the polarity of the intensity changes, the sums of the absolute values of the responses,  $|dx|$  and  $|dy|$ , are also extracted. Therefore each sub-region is associated with a four-dimensional vector

$$v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|). \tag{6}$$

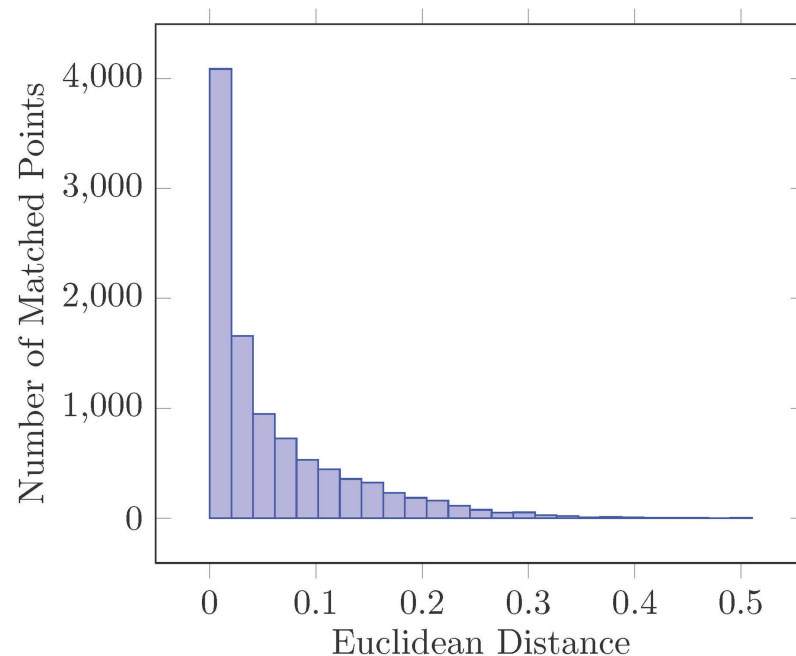
Combining the vectors,  $v$ 's, from all sub-region yields a single 64-dimensional descriptor, which is normalized to unit-norm for contrast invariance.

### 3 Matching of directed line segments

#### 3.1 Rough matching

The best candidate match for each keypoint is found by identifying its nearest neighbor in the set of keypoints generated from a reference image. The nearest neighbor is defined as the keypoint with the minimal Euclidean distance determined based on the invariant descriptor vector described above.

However, many features from an image do not have any matching counterparts in the reference image because they arise from background clutter or cannot be detected in the reference image. Therefore, we use a global threshold on the distance to discard keypoints without good matches. Fig 3 shows the Euclidean distance of 10000 keypoints with correct matches for real image data. This figure was generated by matching images with different scales, rotation angles, changes in illumination, and affine distortions. As shown in Fig 3, most of the matched



**Fig 3. Euclidean distances of 10000 matched keypoints.**

<https://doi.org/10.1371/journal.pone.0173627.g003>

pairs have small Euclidean distances ranging from 0 to 0.15. We set the global threshold to 0.1 in our experiments, eliminating more than 90% of the false matches while discarding less than 5% of the correct matches.

### 3.2 Line segment features

Features of line segments are effective representation of local image information, such as textures and gradients. Given two images  $I$  and  $I'$  to be matched, the feature points are detected for each image using SURF to construct two directed graphs,  $G = (V, E)$  and  $G' = (V', E')$ , where  $V = \{a_1, a_2, \dots, a_n\}$  and  $V' = \{b_1, b_2, \dots, b_m\}$  are key points extracted from  $I$  and  $I'$ , and  $E = \{(a_i, a_j), i \neq j\}$  and  $E' = \{(b_i, b_j), i \neq j\}$  are the edge sets of directed graphs  $G$  and  $G'$ , respectively. Features are generated for each line segment between two key points. For each edge of graph  $G$ ,  $e_{ij} \in E$ , with starting point  $a_i$  and end point  $a_j$ , we equidistantly sample three points  $\{p_1, p_2, p_3\}$ , with  $p_k = p_i + ((k-1)/2)(p_j - p_i)$ ,  $k = 1, 2, 3$ .  $p_i$  is the coordinates of point  $a_i$ . The SURF features are extracted for each of these points, giving a feature matrix  $S = [s_1, s_2, s_3]$ . Each  $s_k$  is a 64-dimensional vector. For each line segment, we have a 192-dimensional feature vector.

### 3.3 Nearest neighbor matching

We use the nearest-neighbor matching criterion proposed in [30] for rough matching of line segments. Assuming image  $I$  has  $n_1$  directed line segments,  $L = [l_1, l_2, \dots, l_{n_1}]$ , and image  $I'$  has  $n_2$  directed line segments,  $L' = [l'_1, l'_2, \dots, l'_{n_2}]$ , the nearest-neighbor pairs can be encoded using an adjacency matrix  $K \in R^{n_1 \times n_2}$ :

$$K(i, j) = \begin{cases} 1 & l'_j \text{ is the nearest neighbor of } l_i \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The distance between a pair of line segments  $l_i$  and  $l'_j$ , with feature matrices  $S_i$  and  $S'_j$  respectively, is defined using the  $F$ -norm of the feature matrices:  $d(l_i, l'_j) = \|S_i - S'_j\|_F$ .

The matching is further refined as follows. With the sets of key points in two given images,  $V = \{a_1, a_2, \dots, a_n\}$  and  $V' = \{b_1, b_2, \dots, b_m\}$ , we use the statistical voting method reported in [30] to obtain the matching frequency of each point. A matrix  $G \in R^{n \times m}$  is initiated as a null matrix. If based on  $K$  two straight lines match each other, we vote for the starting point pairs and the ending point pairs of the two lines once. This is carried out by incrementing the corresponding element in  $G$  by 1. A larger element in matrix  $G$  indicates higher probability of matching of two points. The procedure for computation of matrix  $G$  is detailed in Algorithm 1.

**Algorithm 1** Computation of  $G$

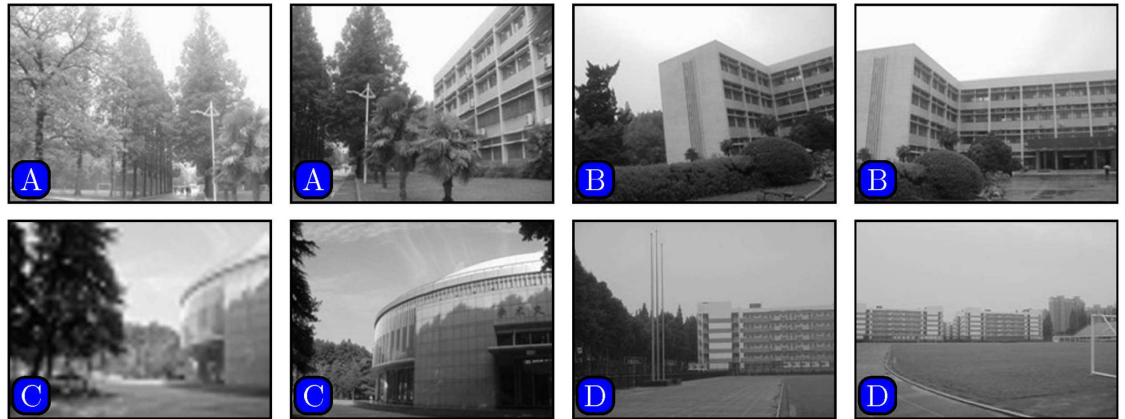
```

Input: Matrix  $K$ 
Output: Matrix  $G$ 
1: procedure COMPUTEMATRIX ( $K, G$ )
2:   Initialize  $G \in R^{m \times n}$  as a null matrix
3:   for  $i = 1, 2, \dots, n_1, j = 1, 2, \dots, n_2$  do
4:     if  $K(i, j) = 1$  then
5:       Find directed line segment  $l_i[a_i \rightarrow a_m], l'_j[b_p \rightarrow b_q]$ 
6:        $G(l, p) = G(l, p) + 1, G(m, q) = G(m, q) + 1$ 
7:     end if
8:   end for
9:   Output matrix  $G$ 
10: end procedure

```

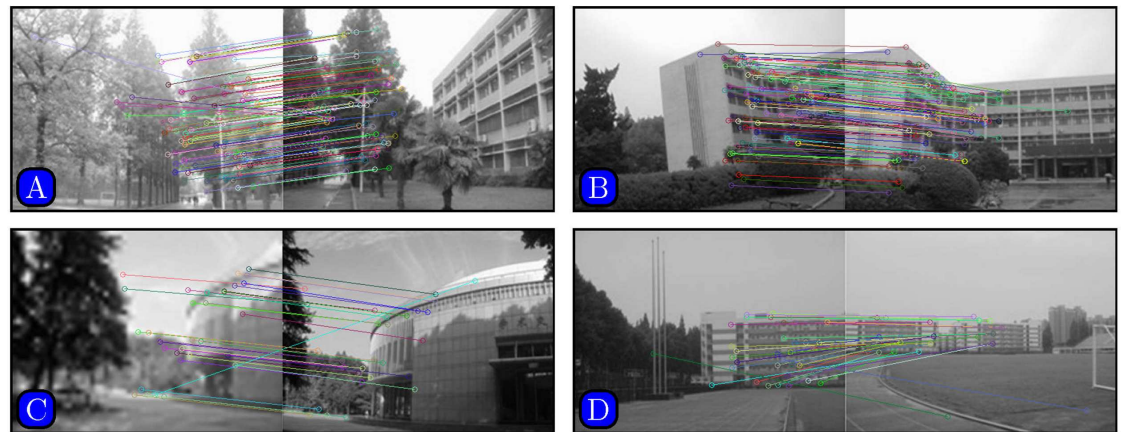
To avoid matching to too many points to one point, the criteria to select matching points are as follows:





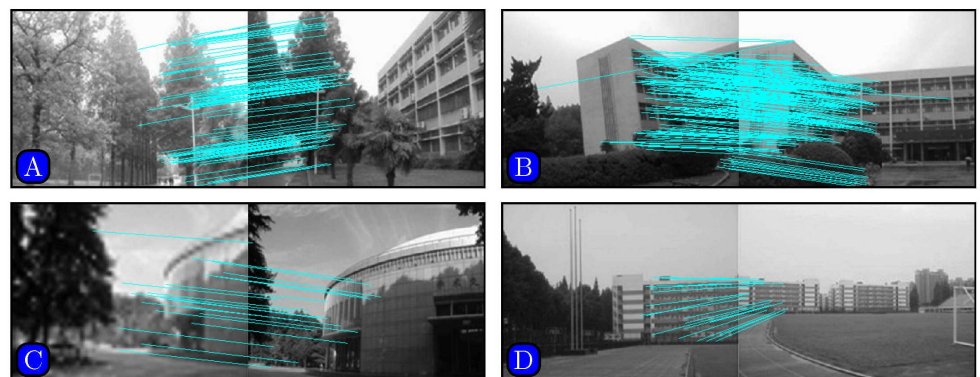
**Fig 4. Image pairs with photometric or geometric variations.** (A) lighting, (B) rotation, (C) blur, (D) scaling. Reprinted from [30] under a CC BY license, with permission from [Computational and Mathematical Methods in Medicine], original copyright [2014].

<https://doi.org/10.1371/journal.pone.0173627.g004>



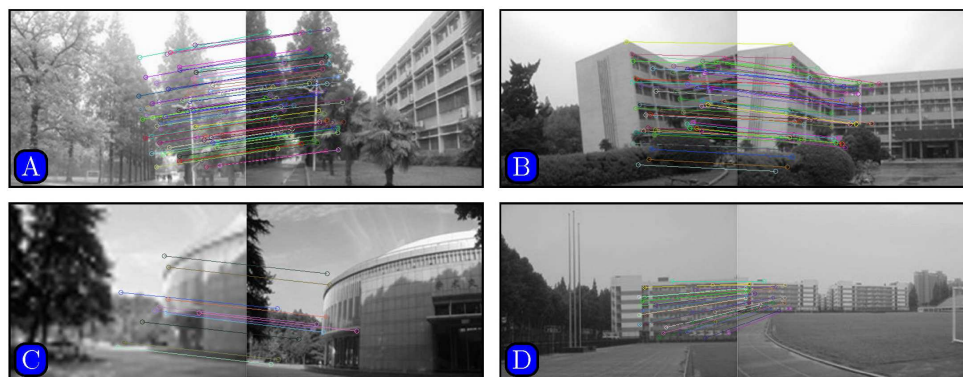
**Fig 5. Matching results based on SURF method.**

<https://doi.org/10.1371/journal.pone.0173627.g005>



**Fig 6. Matching results based on SIFT method.**

<https://doi.org/10.1371/journal.pone.0173627.g006>

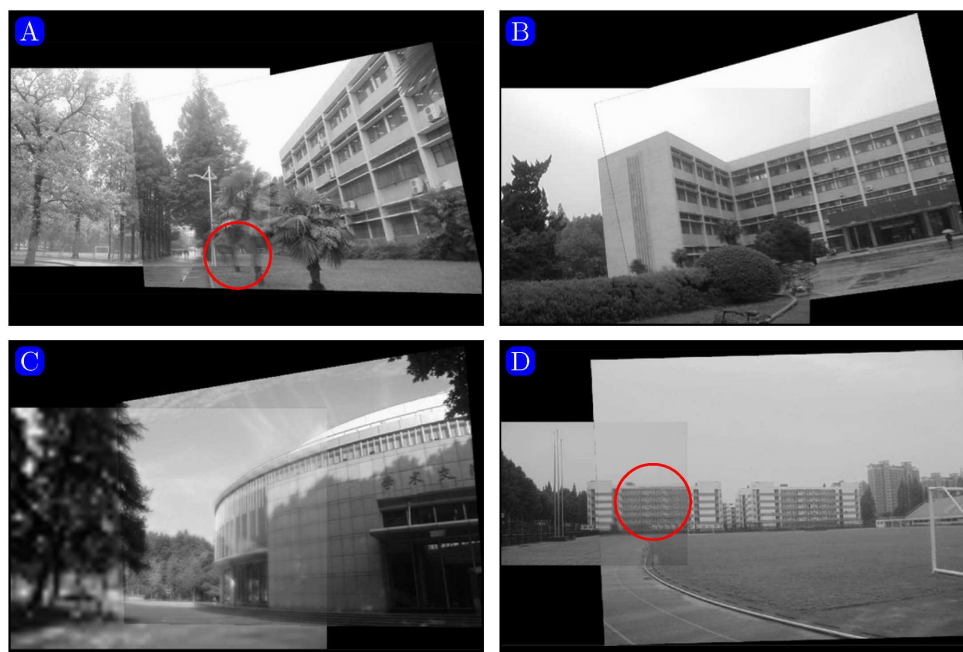


**Fig 7. Matching results based on proposed method.**

<https://doi.org/10.1371/journal.pone.0173627.g007>

- Discard pairs with  $G(i, j) \leq \sigma$ , where  $\sigma = 0.5 \max_{i, j} G(i, j)$ .
- Select pairs giving maximal values in all rows and columns as matched pairs.
- If the maximal element in row  $i$  and the maximal element in column  $j$  are not the same, select the larger one. For example, assuming  $G(i, p)$  is the maximal element in row  $i$  and  $G(q, j)$  is the maximal element in column  $j$ , if  $G(i, p) > G(q, j)$ , then  $a_i$  and  $b_p$  match each other.

Incorrectly matched pairs are further removed by using RANSAC (RANDOM Sample Consensus) [31] and then a homography matrix  $M$  is estimated for image alignment.



**Fig 8. Mosaicking results given by the method proposed in [30].** Reprinted from [30] under a CC BY license, with permission from [Computational and Mathematical Methods in Medicine], original copyright [2014].

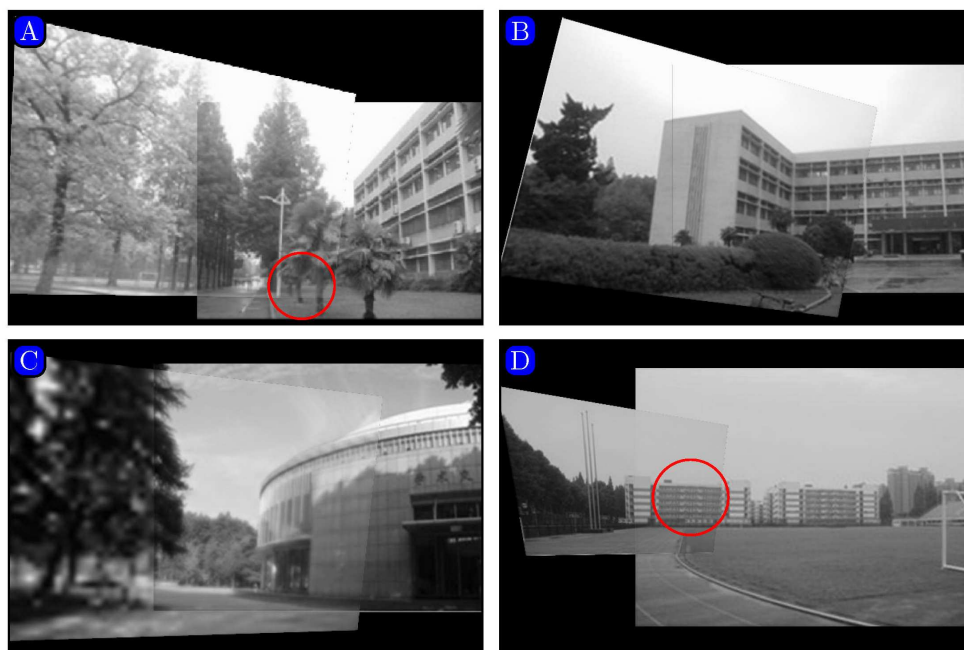
<https://doi.org/10.1371/journal.pone.0173627.g008>



## 4 Experimental results

In this section, the experimental results of the proposed method are presented. Evaluation was performed with gray level images with different rotation angles, scales, illumination, and affine distortions are used. Representative results are shown here.

In order to compare our proposed method with a recent state-of-the-art method presented in [30], images downloaded from the website [35] were used. Representative image pairs are shown in Fig 4. The lighting conditions in the two images are largely different in Fig 4(A). The left image has longer exposure time than the right one. The two images in Fig 4(B) were taken by ordinary camera in different orientations. The two images have different resolutions



**Fig 9. Mosaicking results given by our method.**

<https://doi.org/10.1371/journal.pone.0173627.g009>

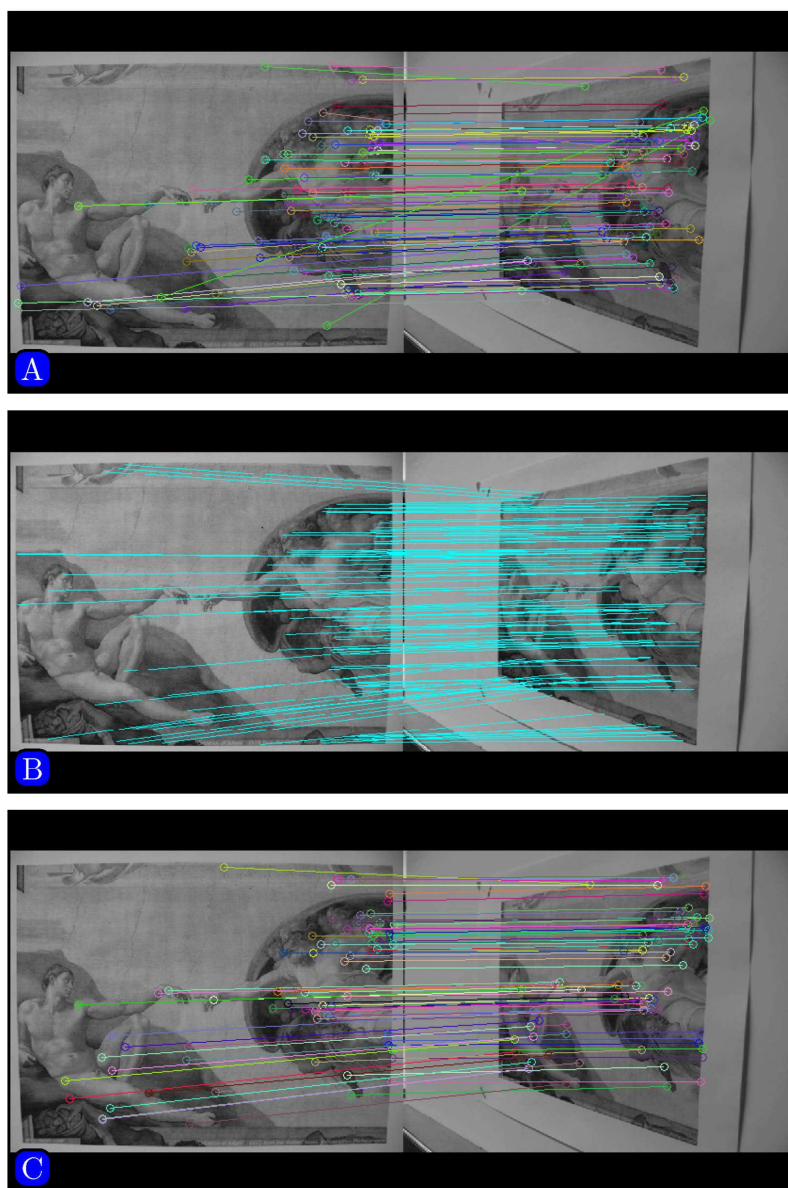


**Fig 10. Images with affine distortion.**

<https://doi.org/10.1371/journal.pone.0173627.g010>

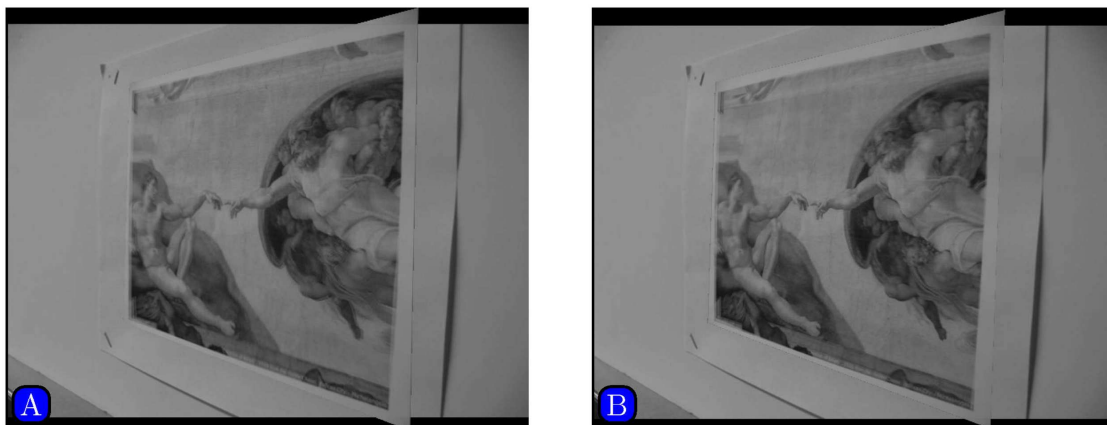
in Fig 4(C). The left one is a blurred low-resolution image and the right one has higher resolution. In Fig 4(D), the left image is taken with the lens of the camera zoomed relative to the right one. Therefore, the buildings in the left image appear larger than the ones in the right.

Results of matching by different methods are shown in Figs 5–7. Fig 5 indicates that SURF cannot even stitch the images correctly due to incorrectly matched points. Figs 6 and 7 demonstrate that both SIFT and our method obtain good results. However, Fig 6(B) indicates that SIFT still results in wrongly matched points. Our method incorporates robust statistical voting and rough matching strategies that could eliminate incorrectly matched pairs.



**Fig 11. Matching by different methods.** (A) SURF, (B) SIFT, (C) Our method.

<https://doi.org/10.1371/journal.pone.0173627.g011>



**Fig 12. Mosaicking results by different methods.** (A) SIFT, (B) Our method.

<https://doi.org/10.1371/journal.pone.0173627.g012>

Figs 8 and 9 show the panoramic images stitched by our method and the algorithm presented in [30]. As shown in Fig 8(A) and 8(D) (in regions marked with red circles), the comparison method results in ghosting due to inaccurate matching.

As shown in Figs 8(C) and 9(C), we can see Fig 9(C) is not clear as the Fig 8(C). The reason is that the quality of the original image downloaded from the website is not good.



**Fig 13. Test image pairs taken from textured and structured scenes under photometric or geometric transformations.** (A) Bikes (blur), (B) tree (blur), (C) Leuven (lighting), (D) bark (scaling and rotation), (E) wall brick (viewpoint), (F) boat (rotation), (G) graffiti (viewpoint), (H) UBC (JPEG).

<https://doi.org/10.1371/journal.pone.0173627.g013>

**Table 1. Performance comparison with state-of-the-art methods.**

Image	#CM/#TM			1-Precision		
	SURF	SIFT	Proposed	SURF	SIFT	Proposed
A	122/156	234/353	<b>133/156</b>	0.22	0.34	<b>0.15</b>
B	67/95	140/592	<b>81/96</b>	0.29	0.73	<b>0.16</b>
C	63/88	97/192	<b>74/88</b>	0.28	0.49	<b>0.16</b>
D	*	76/156	<b>28/56</b>	*	0.51	<b>0.50</b>
E	60/133	176/445	<b>73/133</b>	0.55	0.60	<b>0.45</b>
F	29/62	158/186	<b>53/62</b>	0.53	0.15	<b>0.15</b>
G	8/23	9/57	<b>13/23</b>	0.65	0.84	<b>0.43</b>
H	368/422	484/723	<b>388/422</b>	0.13	0.33	<b>0.08</b>

CM: correct matches; TM: total matches; \*: matching failed.

<https://doi.org/10.1371/journal.pone.0173627.t001>

Fig 10 shows an image pair with significant affine distortion. Results of matching by different methods are shown in Fig 11(A)–11(C). Fig 12 shows the panoramic images stitched by SIFT and our method. We can see that the panoramic image stitched by our method is cleaner than the one given by SIFT.

To evaluate the proposed method quantitatively, we used some representative test image pairs from website [36], taken for the textured and structured scenes, as shown in Fig 13. The following metric is used:

$$1 - \text{precision} = \frac{\# \text{ false matches}}{\# \text{ true matches} + \# \text{ false matches}} \tag{8}$$

Note that a correct match is a match where two keypoints correspond to the same physical location, and a false match is one where two keypoints come from different physical locations.

Table 1 presents the comparison of the matching results, including the number of correct matches over the number of total matches and 1-precision. The results in the table indicate that our proposed algorithm is superior in terms of 1-precision.

## 5 Conclusion

In this paper, we have introduced a novel image mosaicking method based on SURF features of line segments. This method firstly uses SURF detection operator to detect feature points. Secondly, it constructs directed line segments, describes them with SURF feature, and matches those directed segments to acquire rough point matching. Finally, the RANSAC (RANDOM SAMPLE CONSENSUS) algorithm is used to eliminate incorrect pairs for robust image mosaicking. Experimental results demonstrate that the proposed algorithm is robust to scaling, rotation, lighting, resolution and a substantial range of affine distortion.

Recently, Ji et.al [37] proposed a novel compact bag-of-patterns (CBoP) descriptor with an application to low bit rate mobile landmark search. The CBoP descriptor offers a compact yet discriminative visual representation, which significantly improves search efficiency. In the future, we will try these new methods [37–39] proposed in the fields of mobile visual location recognition and mobile visual search to further improve the performance of our algorithm.

## Supporting information

**S1 File. Euclidean distances of 10000 matched keypoints.**  
(TXT)



## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 61540047), the Northwestern Polytechnical University Foundation for Fundamental Research (NPU-FFRJC201226), and NIH grants (NS093842, EB006733, EB008374, and EB009634).

## Author Contributions

**Conceptualization:** ZY DS PY.

**Data curation:** ZY.

**Formal analysis:** ZY.

**Funding acquisition:** ZY.

**Investigation:** ZY.

**Methodology:** ZY.

**Project administration:** ZY DS PY.

**Resources:** ZY.

**Software:** ZY.

**Supervision:** ZY DS PY.

**Validation:** ZY.

**Visualization:** ZY.

**Writing – original draft:** ZY.

**Writing – review & editing:** ZY DS PY.

## References

1. Shum HY, Szeliski R. Systems and Experiment Paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*. 2000; 36(2):101–130. <https://doi.org/10.1023/A:1008195814169>
2. Li S, Kang X, Fang L, Hu J, Yin H. Pixel-level image fusion: A survey of the state of the art. *Information Fusion*. 2017; 33:100–112. <https://doi.org/10.1016/j.inffus.2016.05.004>
3. Chen G, Zhang P, Wu Y, Shen D, Yap P. Denoising magnetic resonance images using collaborative non-local means. *Neurocomputing*. 2016; 177:215–227. <https://doi.org/10.1016/j.neucom.2015.11.031> PMID: 26949289
4. Mesejo P, Ibanez O, Cordon O, Cagnoni S. A survey on image segmentation using metaheuristic-based deformable models: state of the art and critical analysis. *Applied Soft Computing*. 2016; 44: 1–29. <https://doi.org/10.1016/j.asoc.2016.03.004>
5. Ji R, Cao L, Wang Y. Joint Depth and Semantic Inference from a Single Image via Elastic Conditional Random Field. *Pattern Recognition*. 2016; 59:268–281. <https://doi.org/10.1016/j.patcog.2016.03.016>
6. Francis HM, Edward MM. *Photogrammetry*. Harper & Row, New York, 3rd edition; 1980.
7. Saur, G, Kruger, W, Schumann, A. Extended image differencing for change detection in UAV video mosaics. In *Proceedings of SPIE Conference Video Surveillance and Transportation Imaging Applications*. 2014; 9026.
8. Chiu Y, Chung K, Lin C. An improved universal subsampling strategy for compressing mosaic videos with arbitrary RGB color filter arrays in H.264/AVC. *Journal of Visual Communication and Image Representation*. 2014; 25(7):1791–1799.
9. Sooknanan K, Kokaram A, Corrigan D, Baugh G, Harte N, Wilson J. Indexing and selection of well-lit details in underwater video mosaics using vignetting estimation. *OCEANS 2012-Yeosu*. Yeosu, South Korea, 2012; 1–7.



10. <http://matthewalunbrown.com/autostitch/autostitch.html>.
11. <http://research.microsoft.com/en-us/um/redmond/projects/ice/>.
12. <http://www.arcsoft.com/support/panorama-maker/>.
13. Yang Z, Guo B. Image mosaic based on SIFT. In Proceedings of Intelligent Information Hiding and Multimedia Signal Processing. 2008; 1422–1425.
14. Zhu J, Ren M, Yang Z, Zhao W. Fast matching algorithm based on corner detection. Journal of Nanjing University of Science and Technology. 2011; 35(6):755–758.
15. Chae K, Dong W, Jeong C. SUSAN window based cost calculation for fast stereo matching. International Conference on Computational Intelligence and Security. 2005; 3802:947–952.
16. Shi J, Tomasi C. Good features to track. In Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1994; 593–600.
17. Lindeberg T. Feature Detection with Automatic Scale Selection. International Journal of Computer Vision. 1998; 30(2):79–116. <https://doi.org/10.1023/A:1008045108935>
18. Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 2004; 60(2):91–110. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
19. Mikolajczyk K, Schmid C. An Affine Invariant Interest Point Detector. European Conference on Computer Vision. Springer Berlin Heidelberg. 2002; 2350:128–142.
20. Matas J, Chum O, Urban M, et al. Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing. 2004; 22(10):761–767. <https://doi.org/10.1016/j.imavis.2004.02.006>
21. Tuytelaars T, Gool LV. Wide baseline stereo based on local, affinely invariant regions. In Proceedings of BMVC. 2000; 412–425.
22. Mikolajczyk K, Schmid C. A Performance Evaluation of Local Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005; 27(10):1615–1630. <https://doi.org/10.1109/TPAMI.2005.188> PMID: 16237996
23. Mikolajczyk K, Tuytelaars T, Schmid C, Zisserman A, et al. A Comparison of Affine Region Detectors. International Journal of Computer Vision. 2005; 65(1-2):43–72. <https://doi.org/10.1007/s11263-005-3848-x>
24. Bay H, Ess A, Tuytelaars T, Gool LV. Speeded-up robust features (SURF). Computer Vision and Image Understanding. 2008; 110(3):346–359. <https://doi.org/10.1016/j.cviu.2007.09.014>
25. Leutenegger S, Chli M, Siegwart RY. BRISK: binary Robust invariant scalable keypoints. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 11). 2011; 58(11):2548–2555.
26. Alahi A, Ortiz R, Vandergheynst P. FREAK: Fast retina keypoint. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012). 2012; 510–517.
27. Byrne J, Shi J. Nested shape descriptors. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 13). 2013; 1201–1208.
28. Ozuysal M, Calonder M, Lepetit V, Fua P. Fast keypoint recognition using random ferns. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010; 32(3):448–461. <https://doi.org/10.1109/TPAMI.2009.23> PMID: 20075471
29. Lei F, Wang W. A Fast Method for Image Mosaic Based on SURF. In Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications (ICIEA). 2014; 79–82.
30. Zhu J, Ren M. Image Mosaic Method Based on SIFT Features of Line Segment. Computational and Mathematical Methods in Medicine. 2014; 115(1):46–58.
31. Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communication of the ACM. 1981; 24(6):381–395.
32. Viola, P, Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2001; 1:511–518.
33. Patrice YS, Léon B, Patrick H, Yann L. Boxlets: a fast convolution algorithm for signal processing and neural networks. In Proceedings of the 1999 Conference on Advances in Neural Information Processing Systems. 1999; 11:571–577.
34. Christopher E. Notes on the OpenSURF Library. 2009. <http://www.cs.bris.ac.uk/publications/papers/2000970.pdf>.
35. <http://www.hindawi.com/journals/cmmm/2014/926312/>.
36. <http://www.robots.ox.ac.uk/vgg/research/affine/>.

37. Ji R, Duan L, Chen J, Huang T, Gao W. Mining Compact Bag-of-Patterns for Low Bit Rate Mobile Visual Search. *IEEE Transactions on Image Processing*. 2014; 23(7):3099–3113. <https://doi.org/10.1109/TIP.2014.2324291> PMID: 24835227
38. Zhang Y, Guan T, Duan L, Wei B, Gao J, Mao T. Inertial sensors supported visual descriptors encoding and geometric verification for mobile visual location recognition applications. *Signal Processing*. 2015; 112:17–26. <https://doi.org/10.1016/j.sigpro.2014.08.029>
39. Guan T, Wang Y, Duan L, Ji R. On-Device Mobile Landmark Recognition Using Binarized Descriptor with Multifeature Fusion. *ACM Transactions on Intelligent Systems and Technology*. 2015; 7(1):Article 12. <https://doi.org/10.1145/2795234>