ROBUST FEATURE MATCHING IN TERRESTRIAL IMAGE SEQUENCES

A. Abbas[1], S. Ghuffar[2] *

[1, 2] Geospatial Research and Education Lab (GREL)
Dept. of Space Science, Institute of Space Technology, Islamabad, Pakistan
(ahsan, sajid.ghuffar)@grel.ist.edu.pk

Commission III, Urban Sensing and Mobility

KEY WORDS: Feature Detection, Feature Matching, SIFT, SURF, RANSAC, 3D Reconstruction

ABSTRACT:

From the last decade, the feature detection, description and matching techniques are most commonly exploited in various photogrammetric and computer vision applications, which includes: 3D reconstruction of scenes, image stitching for panoramic creation, image classification, or object recognition etc. However, in terrestrial imagery of urban scenes contains various issues, which include duplicate and identical structures (i.e. repeated windows and doors) that cause the problem in feature matching phase and ultimately lead to failure of results specially in case of camera pose and scene structure estimation. In this paper, we will address the issue related to ambiguous feature matching in urban environment due to repeating patterns.

1. INTRODUCTION

Many photogrammetric and computer vision applications are relying on more than one image of same scene or object. In order to relate images to one another, the corresponding points of same scene (3D features) are need to be matched across those images. From the last few years, image feature detectors and descriptors are most widely used techniques for such applications which includes 3D scene reconstruction, panoramic mosaicking/stitching, image classification, object recognition and robot localization etc., all are depends upon the presence of stable and representative features in an image space. Thus, the image features detection and extraction are important steps for these applications (Hassaballah et al., 2016).

Nowadays there are number of algorithms available for feature detectors and descriptors, which provide region of interest, edges or corners (Remondino, n.d.) the most common of them are Speeded Up Robust Features (SURF) (Bay et al., 2006), Scale Invariant Feature Transform (SIFT) (Lowe, 2004), Features from Accelerated Segment Test (FAST) (Rosten and Drummond, 2005) or Binary Robust Invariant Scalable Key points (BRISK) (Leutenegger et al., 2011) etc. Ideally the feature matching characteristics reported by (Haralick and Shapiro, 1992) are: *invariant* (independent from geometric and radiometric distortions), *stability* (robust against image noise), *distinctness* (clearly distinguish from background) and uniqueness (distinguishable from other points).

The feature detection and matching can be split into three steps. 1) **Detection**: find the keypoints in each images. 2) **Description**: Ideally, the local appearance around each feature point should be invariant to scale, rotation, noise, change in illuminations and affine transformations. The distinctive feature descriptors are calculated from each region by picking the neighborhood region around the every key point. Normally we end up with a descriptor vector for each keypoint. 3) **Matching**: To identify similar features, descriptors are compared across the images. In successfully matched features we may get the pairs of $(x_i, y_i) \leftrightarrow (x_i, y_i)$. Where (x_i, y_i) is features in first image and (x_i, y_i) is the matched feature in other image.

However in terrestrial imagery of the urban scenes, there are many repeated feature patterns, nearly identical or duplicate structures with similar texture patters, which ultimately cause the problems in feature matching and subsequently lead to applications result failure (e.g. sparse scene 3D reconstruction). Removal of these incorrect matches is a necessary step to perform specially in case of urban scenes, where the accurate recovery of camera pose and scene structure is necessary. Typical feature matching strategies lead to high number of outliers and due to the fact that the ambiguous matches are parallel to the epipolar lines due to inherent scene geometry and camera motion, robust estimators like RANSAC (used to reject incorrect matches) sometimes lead to wrong solution of correspondences and camera poses.

In the current paper, we investigate and discuss the issues related to ambiguous feature matching using SIFT (Vedaldi and Fulkerson, 2008) and SURF (MATLAB based Implementation) algorithms in urban environment due to repeating patterns that ultimately lead to false camera pose estimation for scene reconstruction. We also provide advices and suggestions about the removal of these known issues. The reason of using SIFT and SURF descriptors is due to their good performance and are widely used technique in many applications.

2. RELATED WORK

In urban scene architecture, symmetry and repetition in designs are most commonly used. The buildings contain hierarchy of symmetries and repetitions on frontage: for example windows and doors, which excessively appears along the horizontal direction. Changchang Wu *et al.* (Wu et al., 2010) presented the technique to find the repeated features on architectural frontal plane

*Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-3-3-2018 | © Authors 2018. CC BY 4.0 License.

with precise recovery of boundary selection for finding the repetition. There method works well for horizontal direction repetition and low-count.

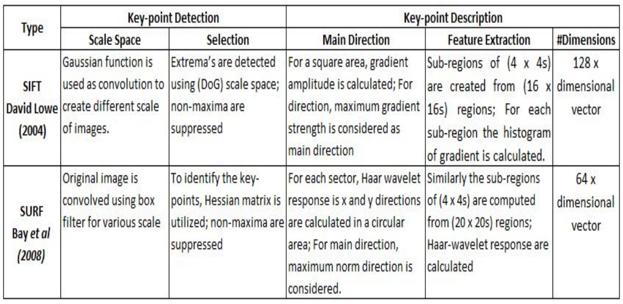
Kyle Wilson et al. (Wilson and Snavely, 2013) also presented the new approach for urban scenes, that contains the repeated features by considering the local visibility graph. There model leads to highly scalable, fast and simple technique for disambiguating the repeated elements without solely relying on geometric reasoning. They used the large datasets drawn from internet photo collections for demonstration of their method and compared it with other geometry based technique of disambiguation.

Richard Roberts *et al.* (Roberts et al., 2011) examined the geometric ambiguities caused by existence of duplicate and repeated structures when different instances are matched on the basis of visual similarity. They proposed the algorithm that recovers the true data association (problem of determining the correspondence either in whole image or feature points) even if there is large number of false pairwise matches exist.

Similarly, the Nianjuan Jiang *et al.* (Jiang et al., 2012) also worked on the repetitive scene structure, which cause the issue in epipolar geometry (EG) due to wrong feature correspondences between image pairs. They proposed the optimization technique called missing correspondences, in which the correct solution was calculated by finding the global minimum of objective function. However, there algorithm contain certain limitations: First, scenes contains complicated occlusion cause the incorrect estimation of visibility. Second, fail in-case of duplicate structures with little background features. Finally, there method may struck at local minimum due to greedy searching and cannot assure to obtain the global minimum, yet its convergence is guaranteed.

3. OVERVIEW OF SIFT AND SURF METHODS

The brief description of both SIFT and SURF operators are illustrated in Fig.1 (Wu et al., 2013).



Type	Key-point Detection		Key-point Description		
	Scale Space	Selection	Main Direction	Feature Extraction	#Dimensions
SIFT David Lowe (2004)	Gaussian function is used as convolution to create different scale of images.	Extrema’s are detected using (DoG) scale space; non-maxima are suppressed	For a square area, gradient amplitude is calculated; For direction, maximum gradient strength is considered as main direction	Sub-regions of (4 x 4s) are created from (16 x 16s) regions; For each sub-region the histogram of gradient is calculated.	128 x dimensional vector
SURF Bay *et al* *(2008)*	Original image is convolved using box filter for various scale	To identify the key-points, Hessian matrix is utilized; non-maxima are suppressed	For each sector, Haar wavelet response is x and y directions are calculated in a circular area; For main direction, maximum norm direction is considered.	Similarly the sub-regions of (4 x 4s) are computed from (20 x 20s) regions; Haar-wavelet response are calculated	64 x dimensional vector

Figure 1. Comparison of SIFT and SURF Operators

3.1 Comparison of SIFT and SURF

In literature, lot of evaluations are done for SIFT and SURF operators related to their performance, time consumption and behavior under different conditions such as change in scale or rotation etc. but choosing the method between them is solely relying on the application.

The feature descriptor used in SIFT and SURF is typically a 128 element vector. The feature descriptor can be reduced to 64 elements, which can lead to faster matching at the cost of lower accuracy. SIFT is invariant to scale change, rotation, affine transformation and rescaling of images, but not good in case of illumination change. Whereas, the SUFT is not fully affine invariant, unstable under extreme rotation and illumination changes (Juan and Gwun, 2009, Hassaballah et al., 2016, Vedaldi and Fulkerson, 2008).

4. EXPERIMENTAL RESULT

The performance evaluation of both SIFT and SURF operators is presented in this section. First, the speed and quantity of keypoints extraction is compared and discussed. Second, the accuracy and speed of key-points matching in two view images. Finally, the multi-view keypoints matching to estimate the camera pose and scene structure (image geometry) which requires a number of correspondence points between input images of same scene. Traditional procedures establish point correspondence are based on the local descriptors. So, images captured by cameras contains repeated structures and occlusions when the view of camera change, which ultimately induces the inaccuracy in scene structure and camera pose estimation.

All the keypoints are attained by using the default parameters presented by their implementations. For testing, two datasets are used containing the duplicate and repeated structure specially the windows which frequently appears along the horizontal axis on building facades. The images was captured using digital camera Nikon D5300. It is 24.2 Million pixel camera with image sensor size of 23.5 x 15.6 mm CMOS and maximum image resolution is 6000 x 4000 pixels.

To detect and extract the keypoints in images, the SIFT implementation by *VLFeat library* (Vedaldi and Fulkerson, 2008) was utilized. Whereas, the SURF implementation in MATLAB environment is used for extrating SURF Features (Bay et al., 2006).

4.1 Keypoints extraction comparison

The extraction time and number of keypoints detected by SIFT and SURF are compared in this section. Table-1 shows the average results obtained after applying both operators to datasets which contain sequence of images. The largest number of keypoints are detected by SIFT operator in both datasets, whereas, the SURF detected relatively less number of keypoints. The variation in number of keypoints is expected due to implementation difference, however one can change the parameters settings to detect the various number of features. For example in SURF implementation, the threshold defined to select strong features using the determinant of the Hessian can be reduced to detect more features. Similarly, in SIFT implementation the thresholds used for detecting peaks in Difference of Gaussian scale space and the threshold to determine the points belonging to an edge can be varied to detect different number of features. But the speed of detecting the keypoints in SURF is more efficient compare to SIFT. In Fig.(2 & 3) below SIFT and SURF keypoints are plot for both dataset.

Dectector	1st Dataset		2nd Dataset	
	Run Time	# Keypoints	Run Time	# Keypoints
SIFT	6 Sec	2200	20 Sec	9950
SURF	2 Sec	1650	8 Sec	7500

Table 1. Comparison of SIFT and SURF keypoints detection and runtime

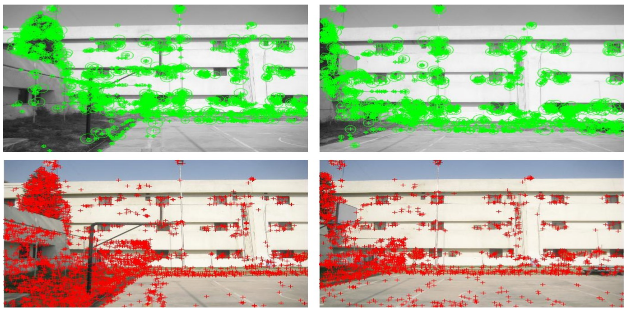


Figure 2. Keypoints detected by SURF (green) and SIFT (red) operators in $1^{st} dataset$

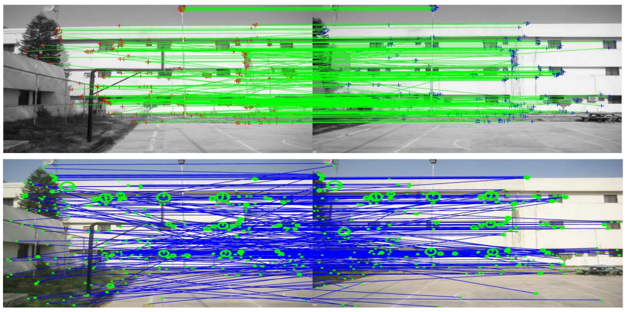


Figure 4. Matched keypoints by SURF (green) and SIFT (blue) operators in $1^{st} dataset$

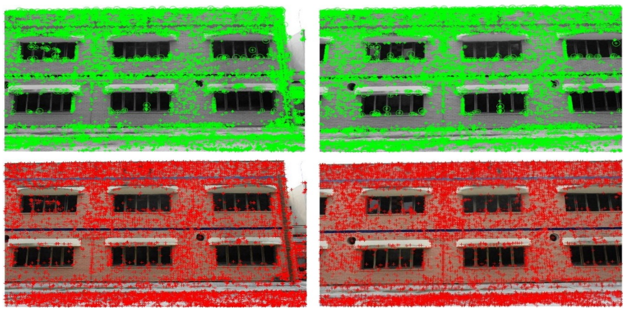


Figure 3. Keypoints detected by SURF (green) and SIFT (red) operators in $2^{nd} dataset$

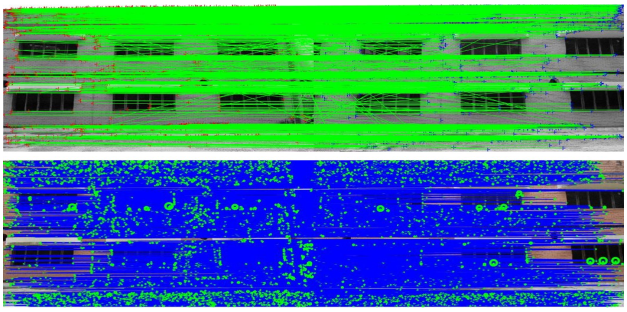


Figure 5. Matched keypoints by SURF (green) and SIFT (blue) operators in $2^{nd} dataset$

4.2 Efficient keypoints matching

In this section, the matching speed and quality of matches between consecutive images pairs are investigated. keypoints are matched between two images at a time using KD-tree data structure (Friedman et al., 1977). This method is effective and efficient in low dimensions, but its efficiency reduces for high dimensional data (Silpa-Anan and Hartley, 2008). Here, the bidirectional search (Jianxiong, n.d.) to effectively match keypoints is used. So, first a KD-tree for all detected keypoints in both image pairs are built and then using KD-tree based nearest neighbor search, two nearest neighbor for each feature point in first image are extracted. All feature matches that do not qualify the ratio threshold (ratio of the distance between the first and the second nearest neighbor) are removed. In the bidirectional search, the nearest neighbor query is performed then for the nearest neighbor extracted in the second image using all the points of the first image. If the the nearest neighbor is the same feature point for which this query was initially performed then the two matched points are stored. The Fig.(4 & 5) show the matched keypoints in both dataset. One can also see there is some wrong correspondences between the matched keypoints in both datasets, to rid of this problem the method called Radom sample consensus (RANSAC) provided by Fischler *et al.* (Fischler and Bolles, 1987) is most commonly exploited.

4.3 Multi-view keypoints matching

Generally, the two view geometry states the epipolar geometry among two images which show the relation of point and line in two images (the corresponding point in one image lies on the epipolar line on other image) which can be estimated using fundamental or essential matrix (Peng et al., 2018). Conventionally, the error in correspondence points can be divided into outlier error and localization error. The localization error is normally caused by noise or due to some quantified effect in image which is in pixels. Whereas, the outlier errors are caused by wrong correspondence points in images which was illustrated in previous Fig.(4 & 5). Usually the outlier causes the large error in results, even a small number of outliers can severely affect the scene geometry and camera pose estimation results.

To accurately estimate the camera pose and scene geometry it is necessary to remove all of them. The robust algorithm RANSAC is use to remove the outlier (incorrect matches), whereas at the same time it finds the inliers (correct matches). However the RANSAC algorithm suffer the decreased in accuracy when the outlier ratio is high (Peng et al., 2018). Therefore, the bidirectional matching step is essential for reducing the number of outliers in the feature matching. Our evaluation has shown that, the RANSAC algorithm typically found the correct inlier set of features following the matching strategy stated above. Occasionally some outliers remained in the matched points which were detected during the *Bundle Adjustment*. Three view geometry using Trifocal tensors can also be used to removed such outliers (Remondino and Ressl, 2006).

The multi-view matching (in three images) is shown in Fig.(6 & 7). The keypoints are matched in three images using the intersection of keypoints between first image pairs $[I_1, I_2]$ and second image pairs $[I_2, I_3]$ and similarly for other image pairs.

$$[I_1, I_2], \text{ intersect}, [I_2, I_3] = [I_1, I_2, I_3]$$

$$[I_2, I_3], \text{ intersect}, [I_3, I_4] = [I_2, I_3, I_4]$$

$$[I_{n-2}, I_{n-1}], \text{ intersect}, [I_{n-1}, I_n] = [I_{n-2}, I_{n-1}, I_n]$$

From the Fig.6 it is clear that, even after applying the RANSAC algorithm, there is still some missed matched keypoints which

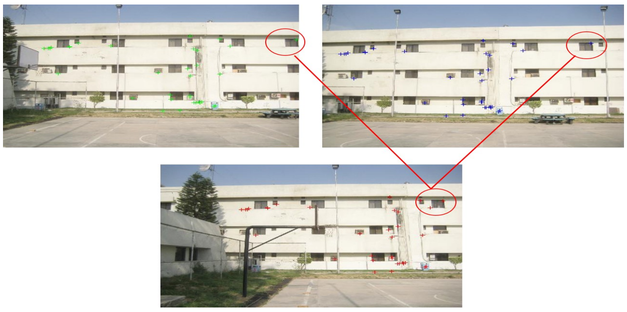


Figure 6. SURF based Multi-view matched keypoints

appears on the epipolar line due to repeated pattern. This issue will be resolve in next step, the 3D scene reconstruction using bundle adjustment.

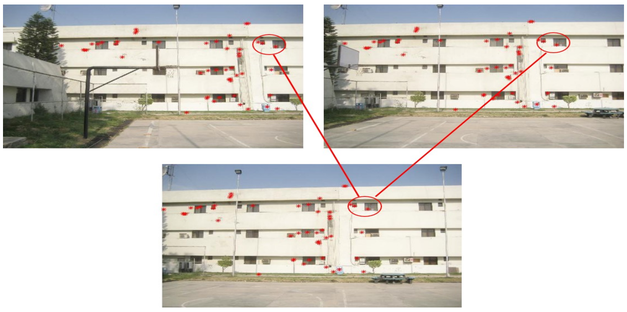


Figure 7. SIFT based Multi-view matched keypoints

5. 3D SCENE RECONSTRUCTION

In previous step, we have matched the keypoints in multi-views which ultimately reduced the point correspondences according to their multiplicity (i.e. how many times the same keypoint appears in images). We maximally found the keypoints correspondence in three images (The Triplets), each triplet contains up to 50 keypoints in case of SIFT operator, whereas less than 50 keypoints in case of SURF operator. The fundamental problem in photogrammetry is to find the 3D location of keypoints in scenes from multiple images captured through different locations (contains both translation and rotation). To find the 3D locations of these multi-view detected keypoints the technique used is called *Triangulation*, if the location and orientation of each image is known which we have calculated during the efficient keypoints matching (4.2) step. We used the SIFT operator results for 3D scene reconstruction due to high keypoints correspondence. The obtained results are shown in Fig.(8).

The red points shows the 3D location of scene points, and blue points are camera positions from where the images was taken. It is clearly shown in figure that, the scene geometry is not clear and also contains some outliers which will be handle in next step. **Bundle Adjustment** (BA) is a non-linear least square optimization technique used to accurately recover the scene structure and refine camera pose parameters by minimizing the re-projection error. However, after applying the bundle adjustment to triangulated keypoints the obtained results are shown in Fig.(9), with optimized scene structure and camera positions.

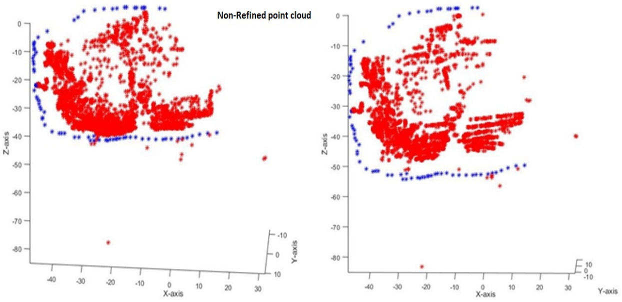


Figure 8. Triangulated point cloud

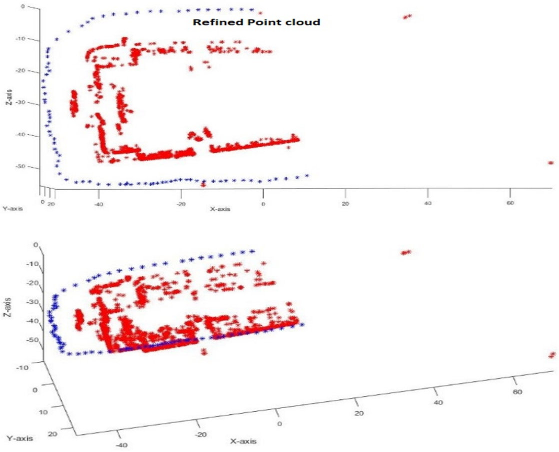


Figure 9. Bundle Adjusted point cloud

The red points are clearly depicting the walls, corners and windows pattern in building facades, whereas the blue points are showing the accurately recovered camera positions. For accuracy assessment, the resultant 3D point cloud were transformed into world coordinate system (UTM) using *Similarity Transformation* technique Fig.(10).

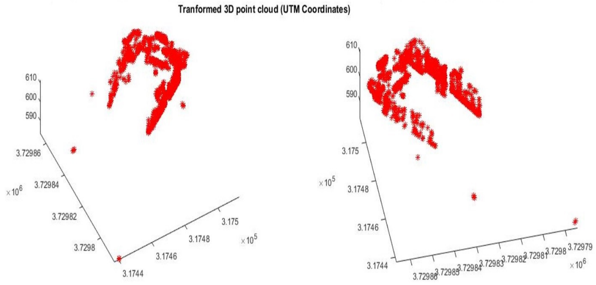


Figure 10. Transformed point cloud

And finally the transformed point cloud were converted into 2D coordinate system so that it can be overlaid on Google Earth Imagery to find its accuracy relative to true position. The Fig.(11) showing the overlaid point cloud on Google Imagery, from the figure it is shown that there is a bit of shift in point cloud due to used of GPS device readings for transformation which contains the error of approximately 10 meters, whereas the overall accuracy of result was very much satisfactory.

Finally, the dense matching technique *Semi-Global Matching* (SMG) was employed to generate the dense structure of the building Fig.(12). However, there are some gaps in it which need to

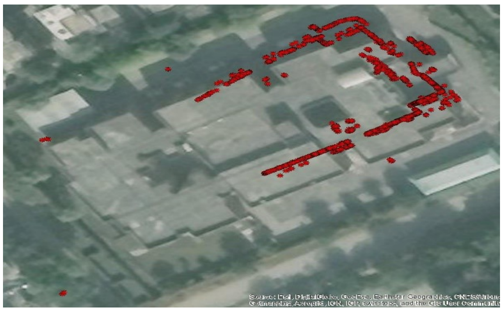


Figure 11. Overlaid point cloud

be fix during matching process. The accurate dense structure is an important requirement for many applications, especially in 3D reconstruction which includes: 3D city/urban modelling, virtual reality, cultural heritage documentation and in 3D animation industry etc. (Remondino and Roditakis, 2003).

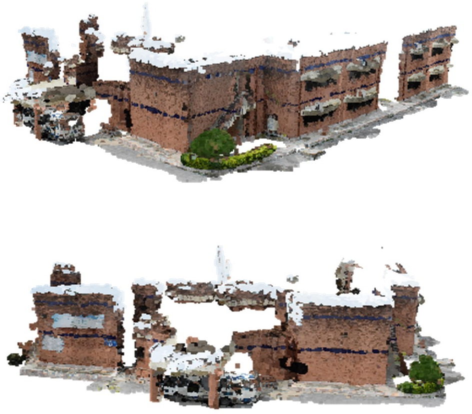


Figure 12. Dense point cloud

6. CONCLUSION

In this paper we illustrated the problems in urban scenes where the duplicate and repetitive structures cause the issue in accurate camera pose estimation and precise recover of scene structure. The experimental results shows that, SURF operator is fast in keypoints detection and matching compared to SIFT operator. However, SIFT detect more keypoints in images and in matching phase. The bidirectional feature matching using Kd tree gives the least number of outliers which are then removed using RANSAC. And finally the sparse and dense scene structure was created.

REFERENCES

Bay, H., Tuytelaars, T. and Van Gool, L., 2006. Surf: Speeded up robust features. In: *European conference on computer vision*, Springer, pp. 404–417.

Fischler, M. A. and Bolles, R. C., 1987. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Readings in computer vision*, Elsevier, pp. 726–740.

Friedman, J. H., Bentley, J. L. and Finkel, R. A., 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* 3(3), pp. 209–226.

Haralick, R. M. and Shapiro, L. G., 1992. *Computer and robot vision*. Addison-wesley.

Hassaballah, M., Abdelmgeid, A. A. and Alshazly, H. A., 2016. Image features detection, description and matching. In: *Image Feature Detectors and Descriptors*, Springer, pp. 11–45.

Jiang, N., Tan, P. and Cheong, L.-F., 2012. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, pp. 1458–1465.

Jianxiong, X., n.d. Sfmedu: A structure from motion system for education, http://mit.edu/jxiao/public/software/sfmedu.

Juan, L. and Gwun, O., 2009. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)* 3(4), pp. 143–152.

Leutenegger, S., Chli, M. and Siegwart, R. Y., 2011. Brisk: Binary robust invariant scalable keypoints. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, pp. 2548–2555.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60(2), pp. 91–110.

Peng, L., Zhang, Y., Zhou, H. and Lu, T., 2018. A robust method for estimating image geometry with local structure constraint. *IEEE Access*.

Remondino, F., n.d. Detectors and descriptors for photogrammetric applications.

Remondino, F. and Ressl, C., 2006. Overview and experiences in automated markerless image orientation. pp. 248–254.

Remondino, F. and Roditakis, A., 2003. 3d reconstruction of human skeleton from single images or monocular video sequences. In: *Joint Pattern Recognition Symposium*, Springer, pp. 100–107.

Roberts, R., Sinha, S. N., Szeliski, R. and Steedly, D., 2011. Structure from motion for scenes with large duplicate structures. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, pp. 3137–3144.

Rosten, E. and Drummond, T., 2005. Fusing points and lines for high performance tracking. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Vol. 2, IEEE, pp. 1508–1515.

Silpa-Anan, C. and Hartley, R., 2008. Optimised kd-trees for fast image descriptor matching. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, pp. 1–8.

Vedaldi, A. and Fulkerson, B., 2008. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`.

Wilson, K. and Snavely, N., 2013. Network principles for sfm: Disambiguating repeated structures with local context. In: *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, pp. 513–520.

Wu, C., Frahm, J.-M. and Pollefeys, M., 2010. Detecting large repetitive structures with salient boundaries. In: *European conference on computer vision*, Springer, pp. 142–155.

Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D. and Gong, S., 2013. A comparative study of sift and its variants. *Measurement science review* 13(3), pp. 122–131.