Atlantis Studies in Uncertainty Modelling, volume 3
Joint Proceedings of the 19th World Congress of the International Fuzzy Systems Association (IFSA), the 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and the 11th International Summer School on Aggregation Operators (AGOP)

Classification Method of Image Feature Matching Using Naive Bayes Classifier

Jin Yong Kim[1], **Eun Kyeong Kim**[1], **Baekcheon Kim**[1], **Daekeon Ha**[1] and *__Sungshin Kim__[2]
[1]Department of Electrical and Electronics Engineering, Pusan National University, Busan, Korea
{skes1234,kimeunkyeong,1001000,gkeorjs136}@pusan.ac.kr
[2]Department of Electrical Engineering, Pusan National University, Busan, Korea
sskim@pusan.ac.kr

Abstract

Since matching accuracy determines the performance of the overall algorithm, studies using image features require sophisticated classification technique for matching. However, there is a critical problem that the factors used to classify true or false matches are extremely limited. To solve this problem, we defined a new factor through geometric and statistical analysis of matched features. And then we performed the naive Bayes classifier with three factors to classify true or false matches. To verify the proposed method, we compared it with the traditional method using benchmark datasets (Heinly dataset, VGG dataset) where homography is provided as ground truth. As a result of the comparison experiments, the proposed method derived higher precision, recall, and F1 scores than the traditional method.

Keywords: Feature Detection, Feature Matching, Heinly Dataset, Homography, Naive Bayes Classifier, ORB, PROSAC, VGG Dataset

1 Introduction

Image feature has been used in various computer vision or machine vision technologies including object classification, object tracking, pose estimation of robot and camera, Augmented Reality (AR), Visual Simultaneous Localization and Mapping (VSLAM) and so on. Studies using image feature mainly perform feature matching with the features detected in two images, and utilize geometric relationships between the matched features or the number of matched features, etc. Therefore, the matching accuracy determines the performance of the entire algorithm. Accordingly, many studies have been conducted to improve the matching accuracy. However, it is not easy to classify exactly whether the matching is true or false. Because the factors that determine the matching are limited to the distance between descriptors with image features. To solve this problem, we defined a new factor and used it in conjunction with two factors that are mainly used in traditional method. And then, we studied on classification method of matching using the Naive Bayes Classifier (NBC) [1].

The rest of paper is organized as follows. Section 2 of this paper covers related works, and Section 3 explains details on the proposed method. And Section 4 deals with the performance evaluation of the proposed method. Finally, Section 5 presents the conclusions.

2 Related Works

The flowchart of the traditional feature matching method is divided into three steps as shown in Figure 1.

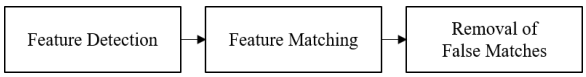


Figure 1: The flowchart of the traditional feature matching method. Traditional feature matching method consists of three steps: feature detection, feature matching, and removal of false matches.

The first step is the feature detection step, and various methods have been studied in feature detection such as Scale Invariant Feature Transform (SIFT), Speed-Up Robust Features (SURF), and Oriented FAST and Rotated BRIEF (ORB) [2]. Because each feature detection method has different advantages and disadvantages, it is recommended to select a suitable method for characteristics and purposes of the dataset to be applied. The second step is the feature matching step. The factor used to match two features is a distance

between descriptors. L1-norm, L2-norm, and Hamming distance are used for it, and matching method such as Nearest Neighbor (NN), k-Nearest Neighbor (k-NN) are mainly used. The third step is to remove false matches that occurred in the second step. Most commonly used algorithms to remove false matches include Least Median of Squares (LMEDS) [3], Random Sample Consensus (RANSAC) [4], and Progressive Sample Consensus (PROSAC) [5].

It is essential to select feature detection method by considering the purpose (scale and rotation invariant, real-time, etc.) in feature matching method. However, it is more important to accurately classify the true or false matches given that the matching accuracy determines the performance of the entire algorithm. Some of the most well-known classification methods include Logistic Regression (LR), NBC, Support Vector Machine (SVM), NN, and decision tree. Among these algorithms, NBC is an easy and quick way to make judgment by simplifying the association between the characteristics of the data to be classified. Also, it is required a assumption that related data are independent of each other to simplify the characteristics of the data, and when data is properly preprocessed, it has the advantage of being more competitive than other methods. So, we used NBC for matching classification.

3 Proposed Method

The flowchart of the proposed method is divided into 6 steps as shown in Figure 2.

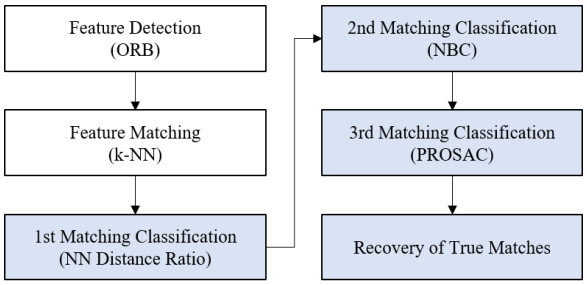


Figure 2: The flowchart of the proposed method. The proposed method handles three classification steps and one recovery step to maximize the performance of ORB-based feature matching.

In the feature detection step of the proposed method, we detected features using ORB method. ORB is not only able to detect many features quickly, but also the computational cost of matching is low. And, since ORB can detect many features compared to SIFT or SURF, the statistical and geometric information of the matched features can be useful. It is a crucial reason why we used ORB.

The second step is the feature matching step. We used the k-NN to consider not only the 1st NN but also the 2nd NN.

In the third step, we classified the true or false matches using NN distance ratio, which means the ratio of 1st NN distance to 2nd NN distance, obtained using the k-NN. The more features are detected, the more similar features exist. In environments with many similar features, false matches are more likely to occur. Therefore, we classified matches whose NN distance ratio is larger than a certain threshold as false match. As shown in Figure 3, if the threshold is too low, many true matches are classified as false matches. Conversely, if the threshold is too high, false matches caused by similar features cannot be properly classified. Therefore, we selected threshold within the range that do not break down the statistical characteristics of the features while preserving the true matches as much as possible.

In the fourth step, we considered three factors to be used for NBC: 1st NN distance, NN distance ratio, and locality score. 1st NN distance and NN distance ratio are factors that have been commonly used to remove false matches. And locality score is a our newly defined factor for NBC's performance improvement. These three factors and how they were applied to NBC are described in Subsection 3.1, 3.2, and 3.3.

In the fifth step, we derived homography using PROSAC method for matches classified as true matches in the fourth step, and classifies false matches that were not used in homography derivation.

In the sixth step, we recovered matches classified as false matches, even though they are actually true matches. For that, homography obtained in the fifth step is used, and the details are described in Section 3.4.

3.1 Three Factors

For the binary classification of matches, we used three factors that can reflect how accurate the matching is. The 1st factor is the shortest distance between descriptors, i.e. 1st NN distance. This factor is a representative of similarity between features, but it is vulnerable to noise and has the disadvantage of being unable to cope with the occurrence of many similar features. Therefore, we considered the factor that can cope with situations in which many similar features exist. So, we selected NN distance ratio as the 2nd factor. The 2nd factor represents the uniqueness of feature, which can cope with the presence of many similar features. And we defined the 3rd factor according to the following Assumption 1.

Assumption 1. *If matching is true, the two matched*

(a) Threshold = 0.3 (b) Threshold = 0.6 (c) Threshold = 0.9

Figure 3: Results of feature matching according to threshold of nearest neighbor distance ratio. nearest neighbor distance ratio is inversely proportional to the uniqueness of the descriptor. Therefore, many matches using ORB descriptor with low uniqueness have high nearest neighbor distance ratio.

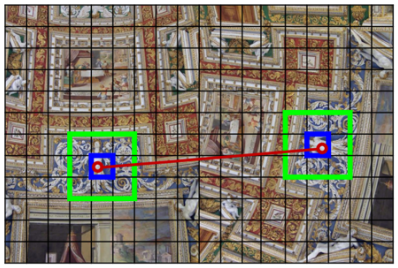
features share the surrounding region.

According to the Assumption 1, if a match is true, two matched features share the surrounding areas. So it is likely that many matches will occur between the surrounding areas. Conversely, if a match is false, the surrounding areas of two features are different, and it is unlikely that matches will occur between the surrounding areas. Therefore, we defined the 3rd factor called 'locality score' by modeling Assumption 1 mathematically. Details of the 3rd factor are covered in Subsection 3.2.

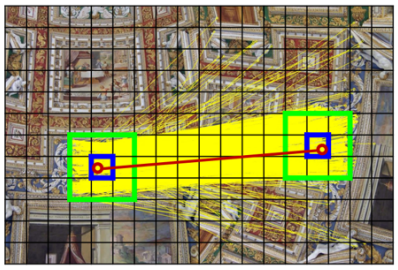
3.2 Locality Score

We referred to some of the ideas from the paper 'GMS: Grid-based Motion Statistics for Fast, Ultra-robust Feature Correspondence [6]' to model Assumption 1. First, we used grid framework to define the surrounding region of the matched features. With grid framework, the effort to calculate the surrounding region for each matched feature can be reduced. Process for obtaining the 3rd factor using the grid framework is as follows.

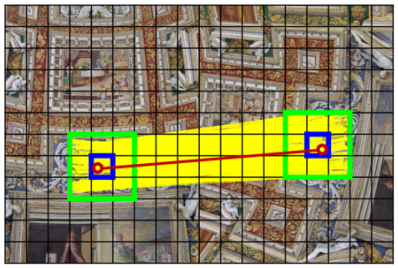
1. Apply grid framework to both images.
2. Search for a grid corresponding to the location of the matched features.
3. In both images, set 9 grids including one grid corresponding to the location of a matched feature, and 8 grids around it as a region of interest (ROI).
4. Search for matches that occurred within the reference image's ROI.
5. Search for matches that occurred only within the ROI in the two images.
6. Calculates the ratio of the number of matches derived from process 5 to the number of matches derived from process 4.



(a) Explore grid and ROI of matched features



(b) Explore all matching that occurred in the ROI of the reference image



(c) Explore matching that occurred only in ROI of two images

Figure 4: Process of obtaining locality score using grid framework. Note that assuming that true match shares the surrounding areas, the number of features matched in surrounding areas can be used as a criterion to classify true or false matching.

Figure 4 is a visual representation of the above process. We defined the ratio obtained through the above process as the 3rd factor.

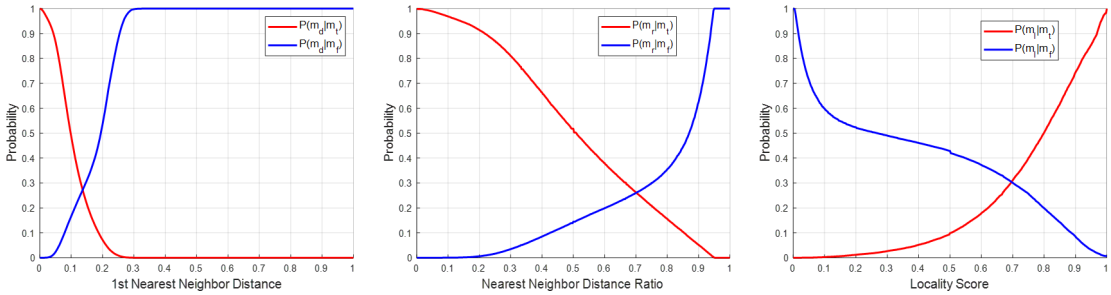


Figure 5: The likelihood of three factors. Likelihoods of the three factors derived from statistical analysis suggests that there is a limit to classifying true or false matches with simple thresholding method.

3.3 Naive Bayes Classifier

First of all, the notation used for NBC is shown in Table 1.

We used three factors to NBC. To use these three factors in NBC, we needed to know the likelihood of each factor in the case of true or false match respectively. So, we analyzed statistically and stochasticized each factor using benchmark datasets where homography is provided as ground truth, and the process is as follows.

1. Classify true or false matches using homography.
2. Derive values of three factors and perform min-max normalization (0-1) for all matching.
3. Calculate the histogram by splitting from zero to one value at a specific interval.
4. Calculate the cumulative histogram considering the property of factors in the case of true or false match respectively.
5. Calculate likelihood by dividing the cumulative histogram by the total number of data.

Figure 5 shows the likelihood of each factor derived by the above process. As shown in Figure 5, while there are certain differences in likelihood graphs in the case of true or false match, there are some overlapped ranges. That is, matching classification using only one factor has limitation. Thus, we used NBC to perform matching classification using three factors rather than one. For that, we set up Assumption 2 as below.

Assumption 2. *Three factors are independent of each other.*

Under the condition that Assumption 2 is established, we modeled NBC using three factors, as shown in Equations 1 and 2.

Notations	
m_t, m_f	Matching is {true, false}, respectively
m_d	1st nearest neighbor distance = d
m_r	Nearest neighbor distance ratio = r
m_l	Locality score = l
$\{f_t(d), f_f(d)\}$	$\{p(m_d\|m_t), p(m_d\|m_f)\}$, respectively
$\{g_t(r), g_f(r)\}$	$\{p(m_r\|m_t), p(m_r\|m_f)\}$, respectively
$\{h_t(l), h_f(l)\}$	$\{p(m_l\|m_t), p(m_l\|m_f)\}$, respectively

Table 1: Notation of events and likelihood functions.

$$P(m_t|m_d,m_r,m_l) = \frac{P(m_d|m_t)P(m_r|m_t)P(m_l|m_t)P(m_t)}{P(m_d)P(m_r)P(m_l)} \quad (1)$$

$$P(m_f|m_d,m_r,m_l) = \frac{P(m_d|m_f)P(m_r|m_f)P(m_l|m_f)P(m_f)}{P(m_d)P(m_r)P(m_l)} \quad (2)$$

The likelihood function for each factor used in Equations 1 and 2 was derived using Gaussian fitting or polynomial fitting method. And then we assumed that the prior probabilities $P(m_t)$ and $P(m_f)$ are the same under the principle of insufficiency reason. The large-small relationship between $P(m_t|m_d,m_r,m_l)$ and $P(m_f|m_d,m_r,m_l)$ determines the true or false matches, and two equations can be simplified as Equations 3 and 4 for comparison of the large-small relationship.

$$P(m_t|m_d,m_r,m_l) \approx f_t(d)g_t(r)h_t(l) \quad (3)$$

$$P(m_f|m_d,m_r,m_l) \approx f_f(d)g_f(r)h_f(l) \quad (4)$$

(a) Before performing the Naive Bayes Classifier

(b) After performing the Naive Bayes Classifier

Figure 6: Result before and after performing naive Bayes classifier. For complex image that is difficult to obtain unique descriptors, it can be confirmed that NBC has effectively classified true or false matches.

We computed Equations 3 and 4 for all matches, then perform matching classification by comparing small-to-large relationships. One of the classification results using NBC is shown in Figure 6.

3.4 Recovery of True Matches

In the recovery of true matches step, we found and recovered matches that were classified as false even though they were actually true. We used homography obtained using PROSAC method in the fifth step to recover lost true matches. Their process is as follows.

1. Project features of the reference image onto the target image using homography for false matches.
2. Calculate L2-norm between projected and matched features.
3. Classify matches with L2-norm below specified thresholds as true matches

One of the recovery results is shown in Figure 7. As you can see, many true matches that were classified as false matches are recovered.

4 Performance Evaluation

We evaluated and compared the performance of traditional and proposed method using precision, recall, and F1 score as performance metrics. We used parts of the benchmark datasets (Heinly dataset [7] and VGG dataset [8]), where homography is given as a ground truth. Information of the benchmark datasets is shown in Table 2.

(a) Before performing the Recover True Matching step

(b) After performing the Recover True Matching step

Figure 7: Result before and after performing the recovery true matches step. This step can improve stability for subsequent algorithms (such as pose estimation, object classification, etc.) by recovering true matches that were classified false from the previous steps.

Dataset	Heinly	VGG
Number of Data	35	40
Ground Truth	Homography	
Description	Rotation, Zoom, Light	Blur, Light, Viewpoint, Zoom + Rotation, JPEG Compression

Table 2: Details on benchmark datasets. Both datasets present homography as ground truth.

As shown in Figure 8, Heinly dataset and VGG dataset consist of complex images having a relationship of changes in Rotation, Zoom, Light, Blur, Zoom + Rotation, and JPEG Compression of reference images. Using these datasets, we derived precision, recall, and F1 score of the traditional and proposed method using homography given as ground truth.

The results of performance evaluation are shown in Figure 9, and the detailed figures are summarized in Table 3. For most datasets, the proposed method derived higher precision, recall, and F1 score than the traditional method.

Figure 8: Parts of Heinly dataset and VGG dataset used for performance evaluation. Both datasets contain many challenges such as zoom, blur, light, viewpoint, etc., and are used to evaluate the robustness for each situation.

5 Conclusion

We studied matching classification using NBC to improve the performance of feature matching method in the field of computer vision and machine vision. To evaluate the performance of the proposed method, we used Heinly dataset and VGG dataset providing homography as a ground truth. Experimental results show that the proposed method derived excellent precision, recall, and F1 score, and improved the matching accuracy over traditional method significantly. In the future works, we would like to further study matching classification method that can be used as other feature detection methods as well as ORB. Also, we will study how to give robustness to perspective variations, shading, rotation, etc.

Acknowledgement

This work was supported by BK21PLUS, Creative Human Resource Education and Research Programs for ICT Convergence in the 4th Industrial Revolution and was supported by the Technology Innovation Program(10073147, Development of Robot Manipulation Technology by Using Artificial Intelligence) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea) and was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE)(P0008473, HRD Program for Industrial Innovation).

References

[1] I. Rish, et al., An empirical study of the naive bayes classifier, in: IJCAI 2001 workshop on empirical methods in artificial intelligence, Vol. 3, 2001, pp. 41–46.

[2] S. A. K. Tareen, Z. Saleem, A comparative analysis of sift, surf, kaze, akaze, orb, and brisk, in: 2018 International conference on computing, mathematics and engineering technologies (iCoMET), IEEE, 2018, pp. 1–10.

[3] P. J. Rousseeuw, Least median of squares regression, Journal of the American statistical association 79 (388) (1984) 871–880.

[4] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[5] O. Chum, J. Matas, Matching with prosac-progressive sample consensus, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1, IEEE, 2005, pp. 220–226.

[6] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, M.-M. Cheng, Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4181–4190.

[7] J. Heinly, E. Dunn, J.-M. Frahm, Comparative evaluation of binary features, in: European Conference on Computer Vision, Springer, 2012, pp. 759–773.

[8] K. Mikolajczyk, Affine covariant features. visual geometry group, university of oxford (2004).

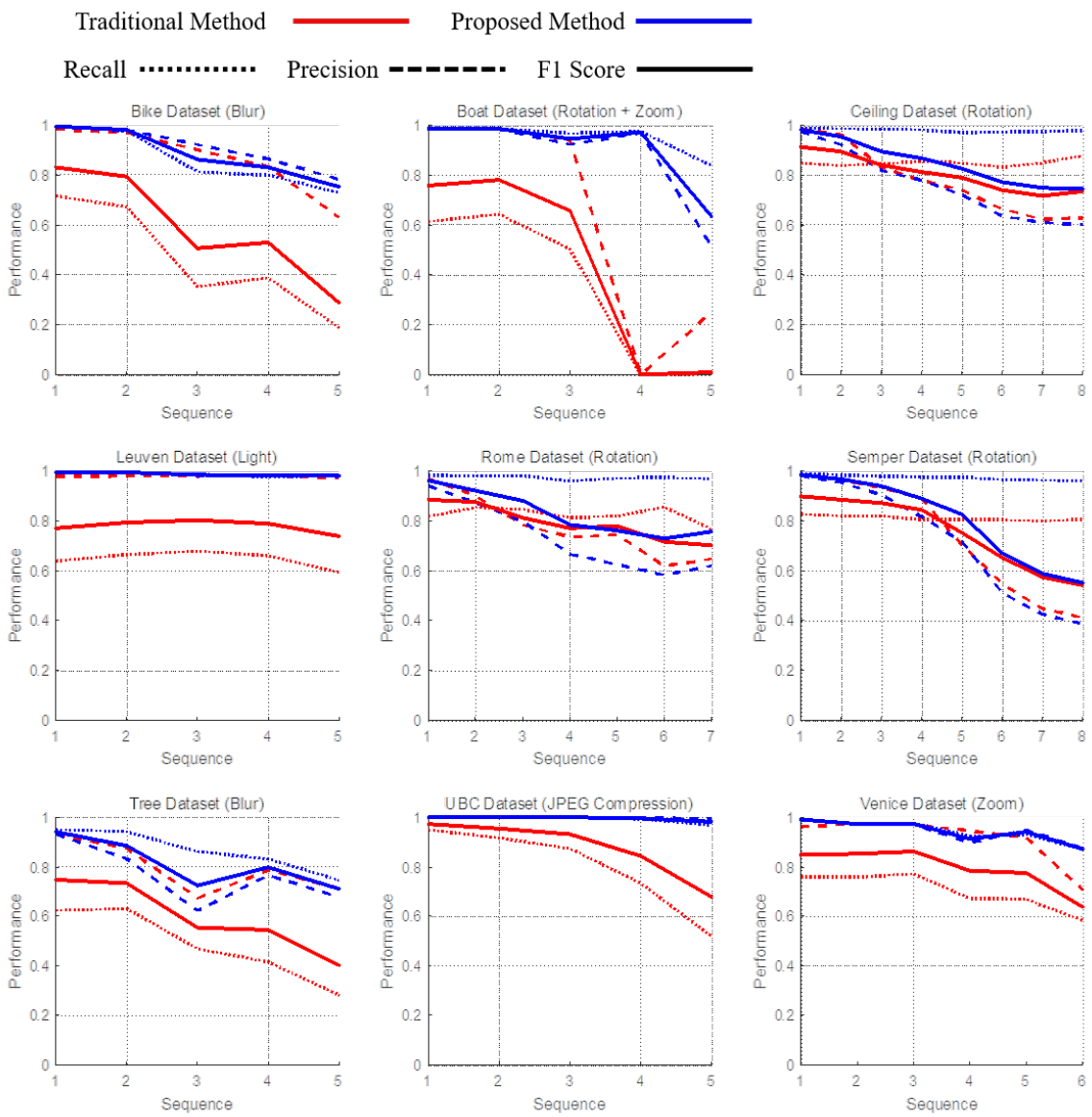


Figure 9: Performance evaluation results of the proposed and traditional method using benchmark datasets. In the graph, the x-axis represents the image index, and the y-axis represents the performance. And red line represents the traditional method and blue line represents the proposed method.

Image Pair Index	Precision		Recall		F1 Score	
	Traditional Method	Proposed Method	Traditional Method	Proposed Method	traditional Method	Proposed Method
Bike Dataset (Blur)						
#1	0.9840	**0.9930**	0.7167	**0.9942**	0.8293	**0.9936**
#2	0.9706	**0.9824**	0.6733	**0.9778**	0.7951	**0.9801**
#3	0.9014	**0.9225**	0.3522	**0.8137**	0.5065	**0.8647**
#4	0.8353	**0.8651**	0.3884	**0.7998**	0.5302	**0.8312**
#5	0.6308	**0.7833**	0.1881	**0.7294**	0.2898	**0.7553**
Boat Dataset (Rotation + Zoom)						
#1	0.9825	**0.9869**	0.6148	**0.9890**	0.7563	**0.9880**
#2	0.9871	**0.9853**	0.6433	**0.9869**	0.7790	**0.9861**
#3	0.9405	**0.9233**	0.5033	**0.9681**	0.6557	**0.9452**
#4	0.0000	**0.9688**	0.0000	**0.9751**	0.0000	**0.9720**
#5	0.2500	**0.5139**	0.0051	**0.8384**	0.0099	**0.6372**
Ceiling Dataset (Rotation)						
#1	0.9856	**0.9739**	0.8492	**0.9892**	0.9123	**0.9815**
#2	0.9622	**0.9223**	0.8385	**0.9868**	0.8961	**0.9535**
#3	0.8327	**0.8192**	0.8458	**0.9842**	0.8392	**0.8942**
#4	0.7786	**0.7783**	0.8552	**0.9822**	0.8151	**0.8685**
#5	0.7393	**0.7216**	0.8493	**0.9713**	0.7905	**0.8280**
#6	0.6646	**0.6366**	0.8322	**0.9734**	0.7390	**0.7698**
#7	0.6231	**0.6095**	0.8495	**0.9758**	0.7189	**0.7503**
#8	0.6290	**0.6024**	0.8792	**0.9801**	0.7333	**0.7462**
Leuven Dataset (Light)						
#1	0.9786	**0.9939**	0.6397	**0.9951**	0.7737	**0.9945**
#2	0.9818	**0.9922**	0.6656	**0.9952**	0.7934	**0.9937**
#3	0.9812	**0.9855**	0.6794	**0.9893**	0.8029	**0.9874**
#4	0.9852	**0.9839**	0.6609	**0.9784**	0.7911	**0.9812**
#5	0.9724	**0.9810**	0.5947	**0.9842**	0.7380	**0.9826**
Rome Dataset (Rotation)						
#1	0.9703	**0.9423**	0.8189	**0.9839**	0.8882	**0.9626**
#2	0.9016	**0.8740**	0.8553	**0.9814**	0.8778	**0.9246**
#3	0.7852	**0.8005**	0.8465	**0.9823**	0.8147	**0.8821**
#4	0.7365	**0.6678**	0.8144	**0.9607**	0.7735	**0.7879**
#5	0.7460	**0.6267**	0.8199	**0.9731**	0.7812	**0.7624**
#6	0.6194	**0.5845**	0.8568	**0.9743**	0.7190	**0.7307**
#7	0.6490	**0.6211**	0.7683	**0.9715**	0.7036	**0.7578**
Semper Dataset (Rotation)						
#1	0.9829	**0.9812**	0.8280	**0.9922**	0.8988	**0.9867**
#2	0.9677	**0.9554**	0.8206	**0.9848**	0.8881	**0.9699**
#3	0.9334	**0.9056**	0.8204	**0.9806**	0.8732	**0.9416**
#4	0.8899	**0.8176**	0.8058	**0.9764**	0.8457	**0.8899**
#5	0.7043	**0.7180**	0.8070	**0.9761**	0.7522	**0.8274**
#6	0.5489	**0.5119**	0.8049	**0.9672**	0.6527	**0.6695**
#7	0.4492	**0.4262**	0.8025	**0.9641**	0.5760	**0.5911**
#8	0.4122	**0.3877**	0.8061	**0.9611**	0.5455	**0.5526**
Tree Dataset (Blur)						
#1	0.9316	**0.9329**	0.6240	**0.9506**	0.7474	**0.9417**
#2	0.8756	**0.8325**	0.6318	**0.9412**	0.7340	**0.8835**
#3	0.6741	**0.6228**	0.4699	**0.8624**	0.5537	**0.7233**
#4	0.7869	**0.7668**	0.4171	**0.8324**	0.5452	**0.7983**
#5	0.7130	**0.6789**	0.2831	**0.7463**	0.4053	**0.7110**
UBC Dataset (JPEG Compression)						
#1	1.0000	**1.0000**	0.9493	**0.9996**	0.9740	**0.9998**
#2	1.0000	**1.0000**	0.9166	**0.9995**	0.9565	**0.9998**
#3	0.9997	**0.9998**	0.8740	**0.9978**	0.9326	**0.9988**
#4	0.9980	**0.9997**	0.7345	**0.9945**	0.8462	**0.9971**
#5	0.9803	**0.9936**	0.5199	**0.9664**	0.6795	**0.9798**

Table 3: Performance evaluation results of the proposed and traditional method using benchmark datasets. Recall, precision, and F1 score were used as performance metrics. And the proposed method was shown to perform better than traditional method.