Recognition of Multiple Characters in a Scene Image Using Arrangement of Local Features

Masakazu Iwamura, Takuya Kobayashi, and Koichi Kise
Graduate School of Engineering, Osaka Prefecture University
Email: {masa, kise}@cs.osakafu-u.ac.jp, kobayashi@m.cs.osakafu-u.ac.jp

***Abstract*—Recognizing characters in a scene helps us obtain useful information. For the purpose, character recognition methods are required to recognize characters of various sizes, various rotation angles and complex layout on complex background. In this paper, we propose a character recognition method using local features having several desirable properties. The novelty of the proposed method is to take into account arrangement of local features so as to recognize multiple characters in an image unlike past methods. The effectiveness and possible improvement of the method are discussed.**

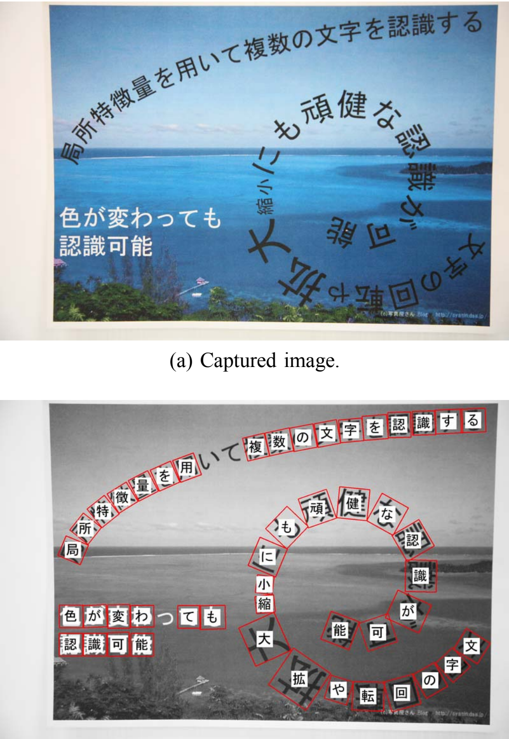
***Keywords*-character recognition in a scene; local features; complex background;**

I. Introduction

Recognizing characters in a scene helps us obtain useful information. Convincing applications include a camera-based translator which enables us to point with a web camera at text in a foreign language and obtain an instantaneous translation, and a voice-navigation service for visually disabled people which enables us to find useful keywords around the user with an omni-directional camera [1].

In order to realize such applications, character recognition methods are required to recognize characters of various sizes, various rotation angles and complex layout on complex background. There are some recognition methods for camera-captured character images. One approach is the orthodox one, that is, segmenting characters from a scene image and recognizing them [2], [3], [4], [5]. In this approach, failure of segmentation means failure of recognition because the recognition process fully depends on the segmentation process. Therefore more robust approach is required. Another approach is exhaustive search for deformed characters without character segmentation [6], [7]. In this approach, many affinely deformed character templates are prepared in advance and then the templates are searched in the query image. While this approach is robust, it takes so much time to cope with recognition of deformed characters. Therefore its appropriate usage would be indexing as Evernote[1] does.

There is another approach going in the middle. That is use of local features such as SIFT and SURF [8], [9], [10], [11]. Local features have been mainly used for object recognition, stereo matching and so on. The advantage of the local features are robustness. Since they are extracted from small regions, they are often less affected by deformations. In addition, they realize robust recognition with loss of some amount of local features. Thus the approach with local features has potential to take advantage of both approaches. However, most existing methods work on a single segmented character. Only method to handle multiple characters in an image employs a simple sliding window strategy to determine each character region [11].



(a) Captured image.

(b) Recognition result.

Figure 1. A recognition result of the proposed method. In (b), character images superimposed on the image are the recognition results and red rectangles are estimated boundaries of the characters. Recall was 94% and precision was 100%.

In this paper we propose a potentially efficient method to determine the character region. The novelty of the method is to take into account arrangement of local features so as to recognize multiple characters in an image unlike past methods. One example of our result is shown in Fig. 1. The effectiveness and possible improvement of the method are discussed with experimental results for Japanese characters.

[1] http://www.evernote.com/

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.283

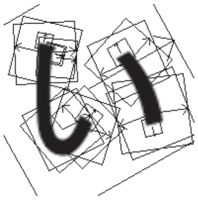


Figure 2. SIFT extracted features from a character image.

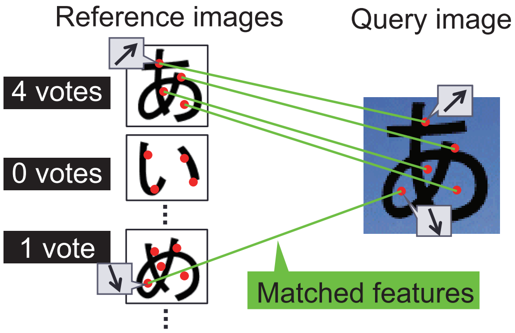


Figure 3. Simple voting technique to determine the category of a character. Red points represents detected local features. Arrows on the rec points represents feature vectors.

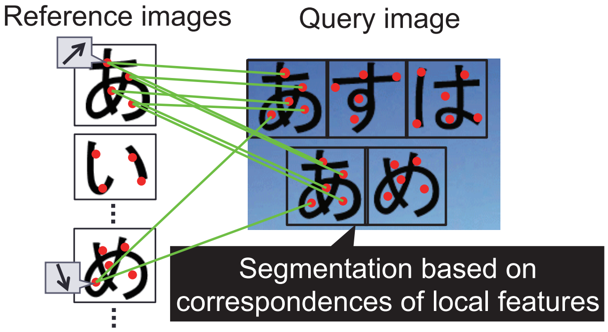


Figure 4. Proposed method to determine the boundaries of characters based on correspondences of local features. Many correspondences are omitted for better looking.

II. Proposed Method

Before presenting the proposed method, let us present a simple method.

A. Simple recognition method with local features

As the local feature, we employ SIFT [12][2]. SIFT features extracted from a character image are shown in Fig. 2. With the local features, most methods employ voting technique to determine the category of a given character image (query image). For explanation purpose, let us introduce a simple voting technique shown in Fig. 3. In advance for recognition, feature vectors are extracted from reference images. For a given query image, feature vectors are extracted in the same manner as the reference images. Then, most similar features of reference images to the ones of the query image are searched. After votes are cast for the most similar categories, the category with highest vote is determined as the category of the query. A feasible voting way is a weighed voting with a weight of 1 / (the number of features in the reference image). We employ this strategy.

While the simple technique successfully works in many cases, there is severe restriction. Since the query image is assumed to contain only one character in a query image, multiple characters in an image cannot be recognized. For the problem, a method using sliding window to determine character regions has been proposed [11].

B. Proposed method to recognize multiple characters

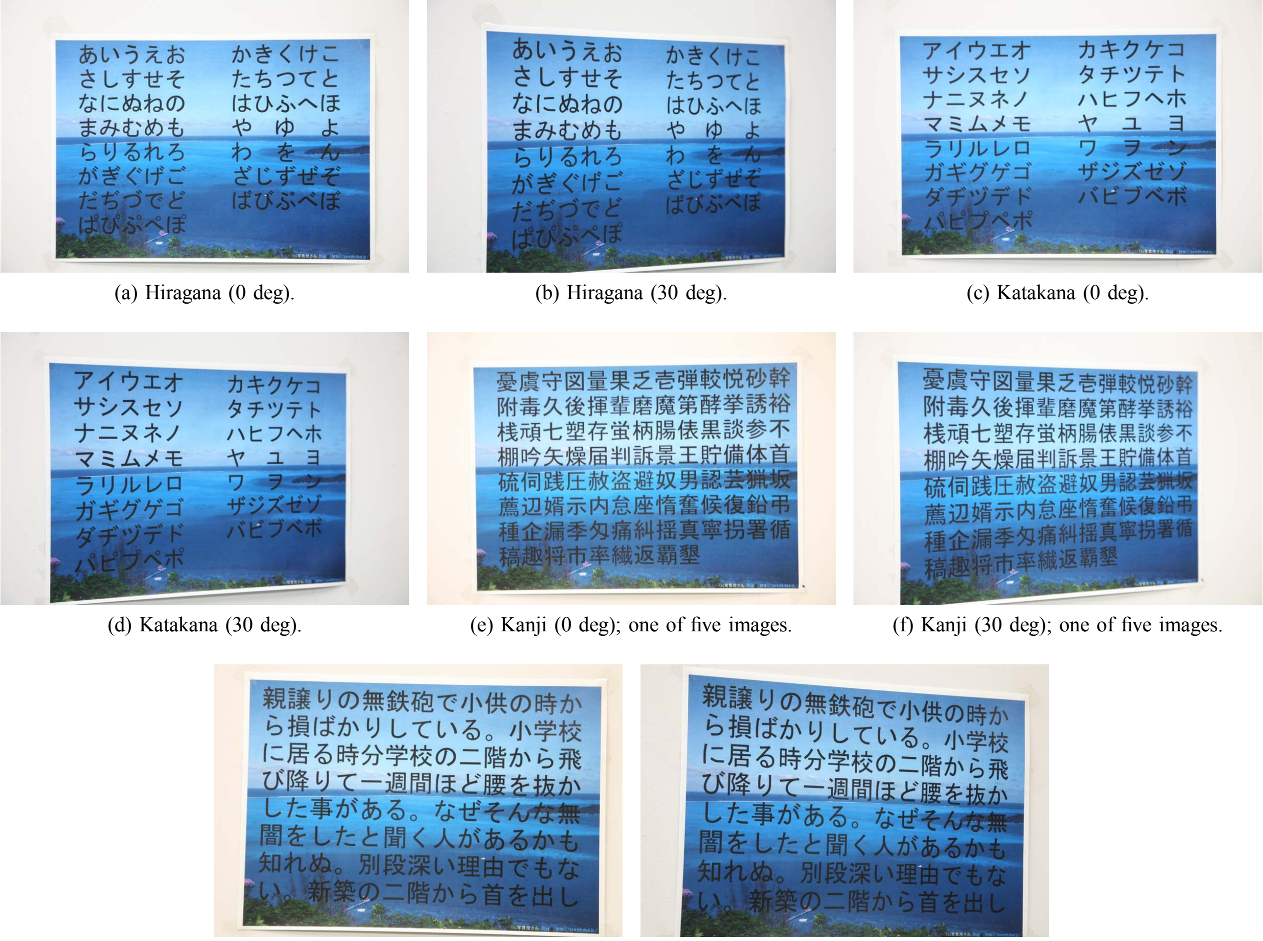
In order to recognize multiple characters contained in a query image, we introduce an idea to utilize arrangement of local features preserved within a character image. While the similar idea is used for an object recognition task [13] using a variant of RANSAC algorithm [14], the assumption that only one object is contained in a query image holds.

The RANSAC algorithm robustly estimates a set of parameters to transform one image to another using all correspondences between features. Thus it cannot handle multiple objects contained in a query image which are described by multiple sets of parameters. Thus, we introduce an idea that restricts regions of features to take into account.

Before presenting our proposed method, we briefly present a recognition strategy with the RANSAC algorithm. Since we handle affine transformation, the number of correspondences used to estimate the parameters is three. In the first step, a set of parameters (hypothesis) for the transformation is estimated using three correspondences. Then, in the second step, e correspondences are used for evaluation of the estimated parameters. $e = 2$ is used in this paper. The evaluation criteria is the number of *hypothetical inliers* which are correspondences satisfying a condition that the distance between a feature in the query image and the corresponding feature in a reference image projected to the query image using the estimated parameters is smaller than a predetermined threshold (10 pixels in this paper). This procedure repeats for many times (100 times in this paper) and the best parameters having lowest criteria is determined. The category of the query and its position (parameters) are determined simultaneously.

The proposed method using the RANSAC algorithm with local restriction is presented. As shown in Fig. 4, characters are segmented based on arrangement of local features. The bounding boxes of reference images are obtained by the following procedure. One reference image is sequentially selected, and only the features of the query images corresponding to those of the reference image are prepared for the following process. One feature of the query image is

[2] Downloaded at http://www.cs.ubc.ca/~lowe/keypoints/.



(a) Hiragana (0 deg).

(b) Hiragana (30 deg).

(c) Katakana (0 deg).

(d) Katakana (30 deg).

(e) Kanji (0 deg); one of five images.

(f) Kanji (30 deg); one of five images.

(g) Mixture (3 deg); one of three images.

(h) Mixture (30 deg); one of three images.

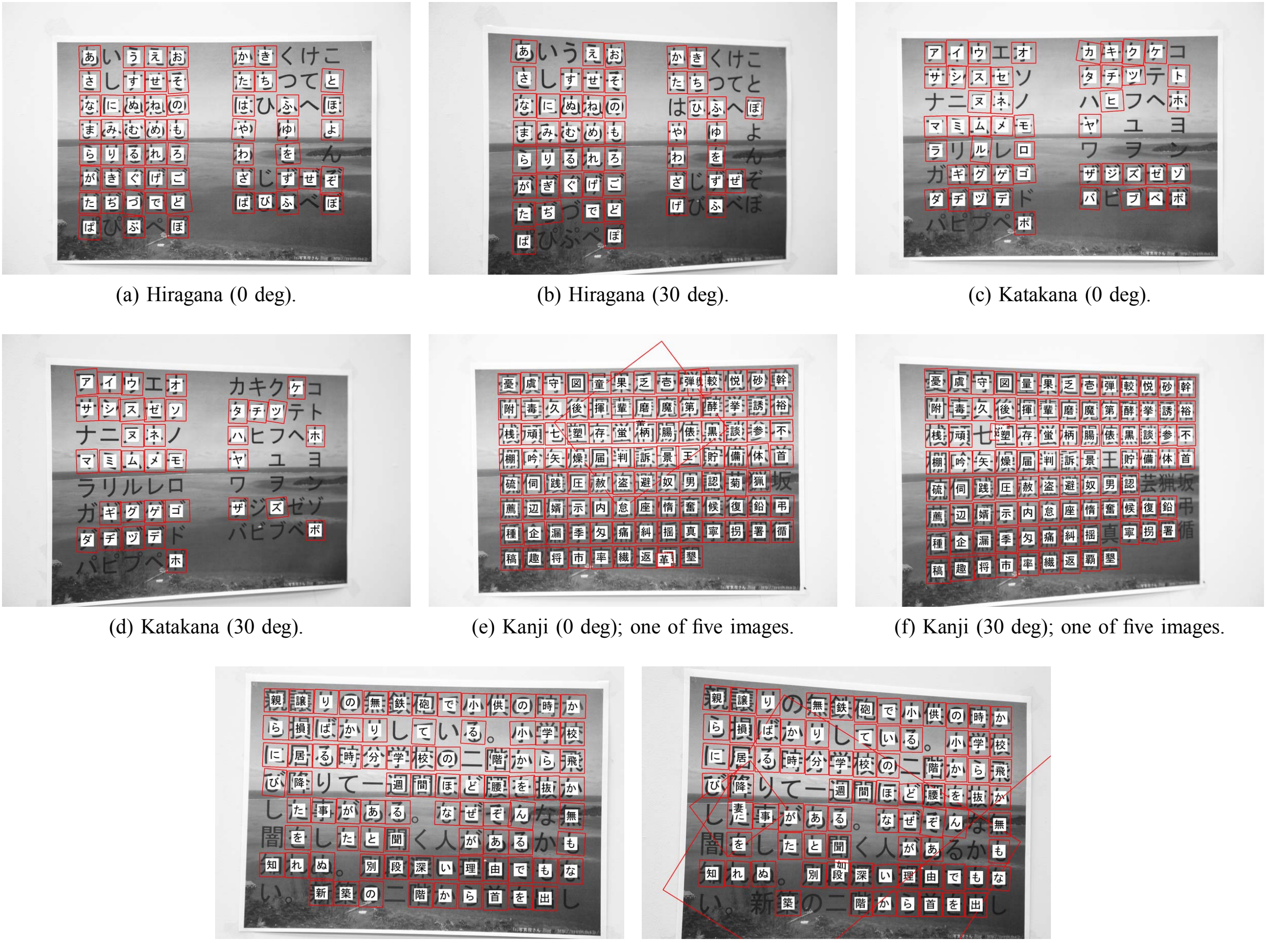
Figure 5. Query images of Hiragana, Katakana, Kanji and Mixture captured at 0 and 30 degrees.

sequentially selected and its nearest $(2 + e)$ features are calculated. Then, the RANSAC algorithm is applied to the $(2 + e)$ features. If the lowest criteria for evaluation is less than the threshold, the bounding box of the corresponding reference image is projected to the query image. This procedure is carried out for all features of the query image and all reference images.

After the procedure above, multiple boundaries in the query image projected with overlap can exist. In such a case, they are deleted except only one boundary having the highest number of weighted votes. The conditions for deletion are as follows: (1) the centers of the bounding box are closer than 20 pixel, and (2) the difference of areas of the boundaries is larger than 10,000 pixels in this paper.

III. Experiment and Discussion

In the experiment, we employed 71 categories of Hiragana, 71 categories of Katakana, 1,945 categories of Kanji (Chinese character) in MS Gothic font. As for reference images, each character image of 60pt was put on a white box of 97×97 pixels and then doubled. Then SIFT features were extracted from the images and their reverse color images to recognize white characters on darker background positioned in the left bottom of Fig. 1(a). In total 353,900 SIFT features were stored in the database. Average number of features in a reference character image is shown in Table I. As for query images, we prepared 10 sheets of query images whose backgrounds were a scene image and foregrounds were characters of 72pt. They consisted of one sheet for Hiragana, one for Katakana, five for Kanji, and three for their mixture borrowed from novels. The sheets were captured at slant angles of 0 and 30 degrees with a digital camera and then SIFT features were extracted from the captured images. Part of the query images are shown in Fig. 5. The dimensions of the images were $4,368 \times 2,912$ pixels. Average number of features in a query image is shown in Table II. For matching features, we used a tree-based approximate nearest neighbor



(a) Hiragana (0 deg).

(b) Hiragana (30 deg).

(c) Katakana (0 deg).

(d) Katakana (30 deg).

(e) Kanji (0 deg); one of five images.

(f) Kanji (30 deg); one of five images.

(g) Mixture (0 deg); one of three images.

(h) Mixture (30 deg); one of three images.

Figure 6. Recognition results of captured images shown in Fig. 5. Character images superimposed on the images represent the recognition results and red rectangles represent estimated boundaries of the characters.

Table I
AVERAGE NUMBER OF FEATURES IN A REFERENCE CHARACTER IMAGE.

Hiragana	Katakana	Kanji
37.2	27.4	88.6

Table II
AVERAGE NUMBER OF FEATURES IN A QUERY IMAGE.

	Hiragana	Katakana	Kanji	Mixture
0 deg.	5055	2397	8149	6831
30 deg.	2882	2278	7429	5238

Table III
RECALL AND PRECISION FOR DIFFERENT TYPES OF CHARACTERS. (%)

		Hiragana	Katakana	Kanji	Mixture
Recall	0 deg.	71.8	66.2	97.2	76.5
	30 deg.	63.4	46.5	92.4	71.7
Precision	0 deg.	89.5	100.0	97.4	99.1
	30 deg.	91.8	94.4	98.3	95.5

Table IV
AVERAGE PROCESSING TIME PER QUERY IMAGE EXCEPT FEATURE EXTRACTION AS WELL AS THAT OF ANN IN THE BRACKET. (SECOND)

	Hiragana	Katakana	Kanji	Mixture
0 deg.	42 (14)	15 (7)	79 (25)	57 (18)
30 deg.	21 (8)	15 (6)	72 (24)	42 (13)

search method called ANN [15]. As for the approximation parameter of ANN, $\epsilon = 2$ was used. A server whose CPU was Opteron 2.8GHz was used for the experiment.

The recognition results corresponding to the query images in Fig. 5 are shown in Fig. 6. The recognition results are summarized in Table III. The figures show that the proposed method has an ability to recognize characters on a complex background. The table shows dependency of recall and precision on query categories. For Kanji, both recall and precision were high. For Hiragana and Katakana, both recall and precision were worse than Kanji. For Mixture, recall was in the middle and precision was better than Kanji. One reason of lower recall in Hiragana and Katakana was that fewer number of features were extracted from these characters than Kanji because of their simpler shapes as shown in Table I. Thus required number of features for RANSAC were not obtained. This can be resolved by employing different kinds of local descriptors and enlarging the query image. One reason of lower precision in Hiragana and Katakana was difficulty of distinguishing characters in similar shapes whose differences are only small points called voiced sound mark (Dakuten in Japanese) and semivoiced sound symbol (Handakuten in Japanese). One cause to reduce both recall and precision was existence of characters sharing similar shapes. This happened mostly in Katakana and Kanji characters. While we did not care, this can be avoided by merging characters in similar shapes when stored in the database. Similar process succeeded in [5].

Processing time of the proposed method except feature extraction is shown in Table IV. Roughly speaking, the processing time is determined based on the number of local features extracted and the number of categories stored in the database. In addition to the processing time in Table IV, feature extraction took about 5 seconds for Hiragana of 0 degree and 10 to 15 seconds for Kanji of 0 degree. In the current implementation, approximately 1/3 of processing time is occupied by searching nearest neighbor by ANN. This can be improved by employing better approximate nearest neighbor search method such as the one we used for camera-based character recognition [5].

IV. Conclusion

In this paper we proposed a recognition method of multiple characters in a scene image with local features in a potentially efficient way unlike past methods. While the proposed method is quite simple, we could achieve robust recognition for query images having a complex background. The proposed method in this paper has a lot of room to be improved. Future work includes (1) employing clustering technique to handle characters having similar parts as in [5], (2) employing different local features and combine them to improve recall, (3) improving computational efficiency including introduction of better approximate nearest neighbor search method such as the one we proposed [5], (4) storing handling multiple fonts in the database to improve performance on real use as in [5].

Acknowledgment

This work was supported in part by JST, CREST, and the Grant-in-Aid for Young Scientists (B) (21700202) from Japan Society for the Promotion of Science (JSPS).

References

[1] M. Iwamura, T. Tsuji, and K. Kise, "Real-life clickable text," *SPIE Newsroom*, Dec. 2010. [Online]. Available: http://spie.org/x43601.xml

[2] G. K. Myers, R. C. Bolles, Q.-T. Luong, J. A. Herson, and H. B. Aradhye, "Rectification and recognition of text in 3-d scenes," *IJDAR*, vol. 7, no. 2-3, pp. 147–158, 2004.

[3] X. Chen, J. Yang, and A. Waibel, "Automatic detection and recognitionof signs from natural scenes," *IEEE Trans. Image Processing*, vol. 13, no. 1, pp. 87–99, Jan. 2004.

[4] L. Li and C. L. Tan, "Recognizing planar symbols with severe perspective deformation," *IEEE TPAMI*, vol. 32, no. 4, pp. 755–762, Apr. 2010.

[5] M. Iwamura, T. Tsuji, and K. Kise, "Memory-based recognition of camera-captured characters," *Proc. DAS2010*, pp. 89–96, Jun. 2010.

[6] Y. Kusachi, A. Suzuki, N. Ito, K. Arakawa, and T. Yasuno, "Scene image indexing and retrieval using candidates of characters," *Trans. IEICE*, vol. J90-D, no. 9, pp. 2562–2572, Sep. 2007, written in Japanese.

[7] Y. Kusachi, A. Suzuki, N. Ito, and K. Arakawa, "Kanji recognition in scene images without detection of text fields — robust against variation of viewpoint, contrast, and background texture—," in *Proc. ICPR2004*, vol. 1, Aug. 2004, pp. 457–460.

[8] T. E. D. Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proc.VISAPP (2)*, Feb. 2009, pp. 273–280.

[9] T. Wu, K. Qi, Q. Zheng, K. Chen, J. Chen, and H. Guan, "An improved descriptor for chinese character recognition," in *Proc. 3rd Int'l Symposium on Intelligent Information Technology Application*, vol. 2, Nov. 2009, pp. 400–403.

[10] S. Uchida and M. Liwicki, "Part-based recognition of handwritten characters," in *Proc. ICFHR2010*, Nov. 2010, pp. 545–550.

[11] Q. Zheng, K. Chen, Y. Zhou, C. Gu, and H. Guan, "Text localization and recognition in complex scenes using local features," in *LNCS (Proc. ACCV2010)*, vol. 6494, Nov. 2011, pp. 121–132.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. CVPR2007*, Jun. 2007.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[15] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching in fixed dimensions," *Journal of the ACM*, vol. 45, no. 6, pp. 891–923, Nov. 1998.