A NOVEL FILTERING APPROACH FOR ROBUST AND FAST KEYPOINT MATCHING IN MOBILE ENVIRONMENT

Jonghoon Seo, Seungho Chae,Yoonsik Yang, Heeseung Choi, Tack-Don Han

Author Affiliation(s)

ABSTRACT

The abstract should appear at the top of the left-hand column of text, about 0.5 inch (12 mm) below the title area and no more than 3.125 inches (80 mm) in length. Leave a 0.5 inch (12 mm) space between the end of the abstract and the beginning of the main text. The abstract should contain about 100 to 150 words, and should be identical to the abstract text submitted electronically along with the paper cover sheet. All manuscripts must be in English, printed in black ink.

Index Terms— One, two, three, four, five

1. INTRODUCTION

Image matching is a fundamental problem in a variety of computer vision applications, including simultaneous localization and mapping[1, 2], object recognition[3], panorama stitching[4, 5], augmented reality[6, 7], and visual odometry[8, 9]. To enhance the image matching quality in various environments, many related techniques have been proposed, such as keypoint-based local matching, histogram-based global matching[10, 11], color-based matching[12, 13], and template-based matching[14], etc. Among them, keypoint detection and matching has created great interest since it can provide relatively high matching quality against severe occlusion and do not require segmentation for regions of interest. Also, recent work has concentrated on making invariant to image transformation with low computing power[15, 16].

The overall flowchart of keypoint matching and recognition is shown in Fig. 1. These procedure can be divided into two main phases: offline (training) and online (testing) procedure. Offline learning is prerequisite to online matching process. In offline learning phase, a set of reference images to be recognized is analyzed and stored as as types of descriptors in a database. In online learning phase, a newly captured image is analyzed and compared with the reference images in the database to find a nearest reference image. In each phase, common procedures for matching are keypoint detection, description, and matching. To analyze training images, at first, keypoints are detected from the images. Then, from those keypoints, local textures are analyzed and described. In this procedure, to provide robustness against rotation, scale, perspective transform, descriptors are constructed. Then, to be used in online phase, efficient matching structures, as databases, are constructed, such as partitioning trees[17, 18, 19], hashing[20, 21, 22]. In the online matching phase, the database is used to find the most similar corresponding keypoints pair with a given query image. To find the most similar keypoint pairs, with given a query image, keypoints are detected, detected keypoints are described about local texture, and compared with the preconstructed database.

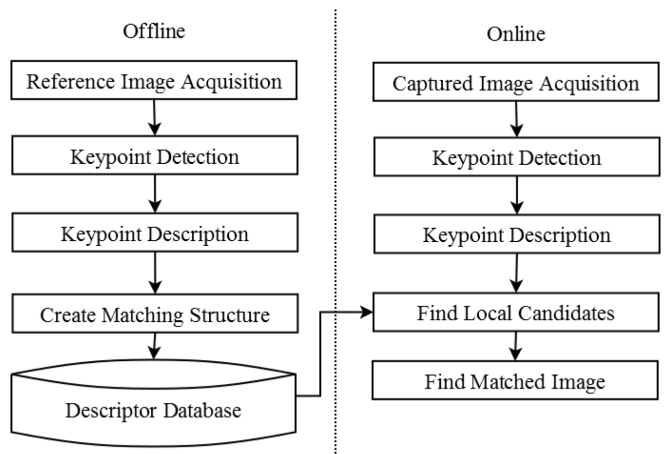


Fig. 1. Overall process of conventional keypoint-based matching

Conventional keypoint matching methods stores almost every keypoints which are detected by keypoint detection process. Keypoint detection processes are designed to extract repeatable keypoints and robust against arbitrary image transformation. Then, detected keypoints are independent to the follow matching procedures, and do not reflect quality of descriptors. Therefore, as seen Fig. some keypoints are not distinguishable, and they tend to cause inter-keypoint confusion and miss matching. `bad keypoint` 이미지 추가 Also, those detected keypoints are stored in database and are compared with keypoints in query images in every frame while matching. Then, it decreases the speed of matching. To overcome these problem, in offline learning procedure, detected keypoints are evaluated with respect to proposed matching quality criteria and filtered by the goodness score.

Thanks to KIST and NFS for funding.

With this filtering method, only a small subset of keypoints is stored in the database. Accordingly, it provides more improved matching performance with faster matching speed.

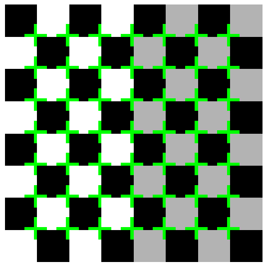


Fig. 2. Example of high repeatable but poor distinguishable keypoints. Conventional keypoint matching systems do not consider the discriminability of keypoints, so these keypoints usually stored and negatively affected matching.

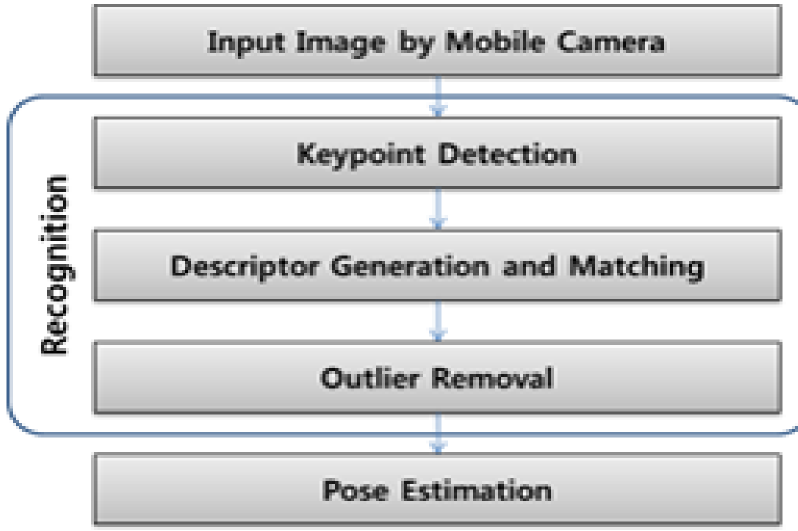


Fig. 3. Feature based MarkerlessAR Process

2. RELATED WORKS

To provide 'Planar Model based Marker-less AR', the feature matching based algorithm shown in Figure 3 is performed. First, in the offline learning process, the keypoints are detected in the reference images. In the keypoints detected as above, the descriptors that are able to describe the keypoints invariably against the distortion are generated by way of computation in various ways. To facilitate the quick reference in the online phase using the descriptors generated, the matching data structure needs to be trained. After that, in the online phase, the keypoints are detected and described in the input query images using the algorithm in the offline process. The descriptors calculated as above search the nearest neighbors in the matching data structure that was already trained and compose correspondence. Only good pairs are filtered from the correspondence composed in compliance with the specific standard and then robust pose estimation is performed to calculate camera extrinsic parameters using the final correspondence residue. After that, 3-D objects rendering is performed using the calculated extrinsic parameters.

However, since this feature matching based markerless AR algorithm requires a large amount of computation, the natural motion is hard to achieve in a smart space environment such as mobile phone. To overcome this limitation, various lightweight algorithms are proposed. As shown in Figure 4, the speed in the feature description and matching phase that greatly affects the overall computational performance has significantly improved.

First, in the study of keypoint descriptor', the established vector value-based description methods such as SIFT[24] or SURF[25] provided high recognition rate, but complex computation was performed to generate the robust descriptors against the distortion such as orientation and scale, etc. In recent years, a wide range of binary value-based descriptors, such as BRIEF[26], ORB[27], BRISK[28], FREAK[29], are under development. These binary descriptors compare the brightness values with focus on the keypoints as shown in Figure **??** by using a wide range of form patterns and express the results in binary codes. Since descriptors are computed only by a simple comparison computation, its computational speed is significantly faster than vector-based descriptors. Moreover, since the orientation and scale are normalized on the basis of generation patterns, they show significantly rigid performance against a wide range of distortion. In particular, studies using the binary descriptors that can be processed by simple comparison computation rather than complex vector-value descriptor based computation are increasing in the environment of limited performance such as a smart space.

Having designed an efficient matching data structure, the following methods perform nearest neighbor matching quickly. The established brute force matching compares all the keypoints of query images with all the keypoints of ref-

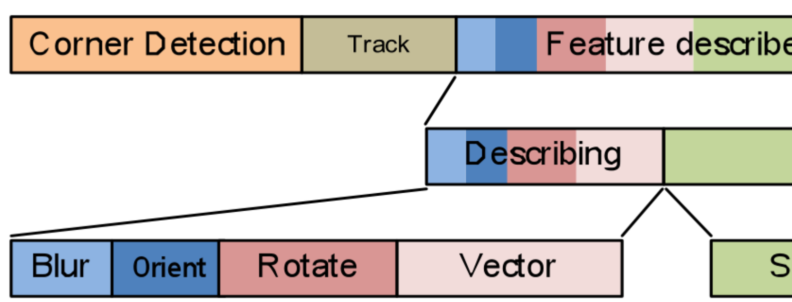
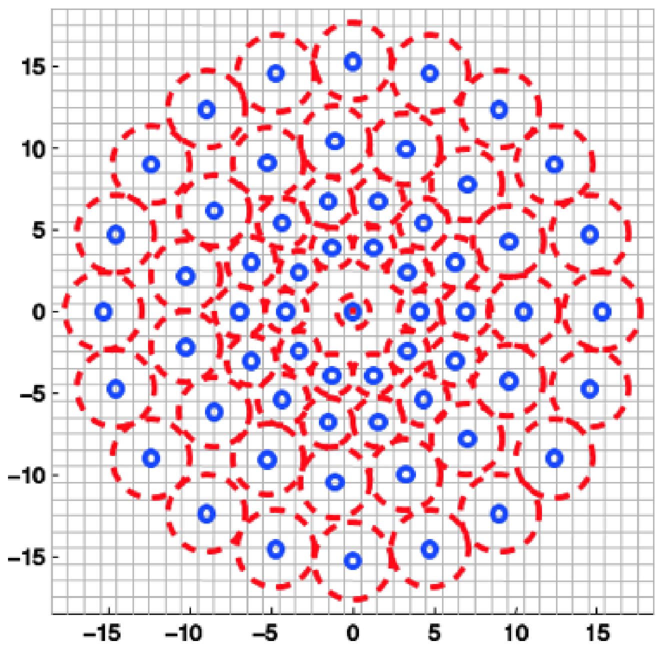
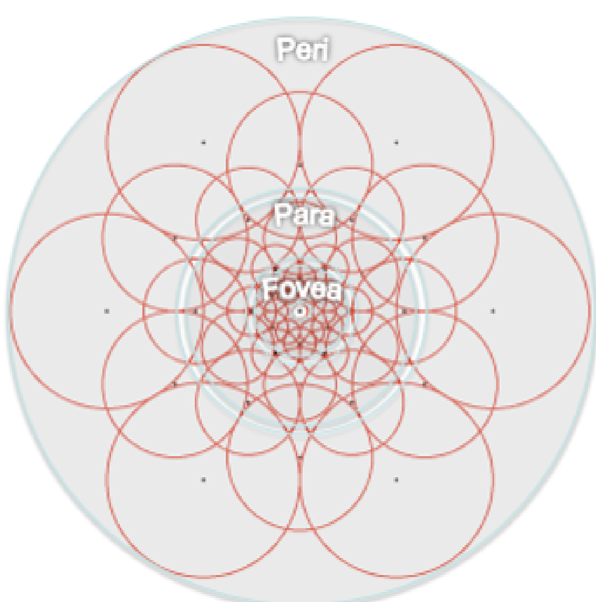


Fig. 4. Relative timings of the SIFT tracker [23]

erence images so is the slowest, but has the advantage of being able to detect the most accurate nearest neighbors. kD tree based approximation method is proposed in [18]. This method shows good performance in the vector-value descriptor method such as SIFT or SURF with relatively low dimension of features but the improvement of its performance is unlikely to be achieved in the latest binary descriptor with high dimension. A method enabling it to accelerate matching



(a) BRISK[28] Sampling Pattern



(b) FREAK[29] Sampling Pattern

Fig. 5. Sampling Patterns of Binary Descriptors

using Hashing based structure in LSH[21] was proposed[27]. As for this method, it is critical to compose appropriate hash function set to distribute the keypoints points evenly in the offline training phase. In addition, 'Random Forest[30]' or 'Random Fern[31]' composed a matching structure by applying a binary description method to the tree structure or list structure. To increase the recognition speed and obtain more accurate approximation values, these matching methods generate efficient matching structure by way of the application of supplementary computation using the descriptors computed in the offline training phase. However, since the supplementary structure that uses this method is complex and large, its the matching structure is too heavy to use in a mobile environment. Moreover, since the properties of the detected keypoints are not taken into consideration, the set for which it is difficult to classify the detected keypoints may lead to performance degradation. To solve this problem, this dissertation proposes a keypoints filtering based matching methods.

3. KEYPOINT DESCRIPTOR FILTERING ALGORITHM

Feature matching based augmented reality system generates the feature database, the subjects of comparison, by way of the offline training of the reference images before online process. In particular, as shown in the foregoing study of matching data structure, a method that composes a matching structure by applying a various computations to offline process to increase the online recognition speed and recognition rate is proposed. Such established kaypoints matching methods used simply all of the keypoints detected in feature detection module for training. However, since the feature detection algorithm is performed independently from the description algorithm, the descriptor is not able to ensue the matching performance for the detected features.

Thus, in this dissertation, the keypoints of the subjects of training in the offline training process were assessed and only the keypoints providing rigid real-time recognition performance were selected. As a result, a method to increase the recognition speed by saving these features only while maintaining the quality of recognition was proposed.

3.1. Definition of Good Keypoins

The proposed method filters only good keypoints by analyzing the detected keypoints and measuring the degree of the effectiveness of them on recognition, To this effect, good keypoints were defined.

The conditions of good keypoints for recognition are as follows:

First, good keypoints need to be stably detected as good points in an environment where targeted images change in various ways. In fact, a wide range of form transformation, such as the rotation, size, perspective, noise and lighting of the targeted images, are applied to the camera images used in the actual matching process. The good keypoints for recognition are detected stably in the converted images, generating descriptors.

The detection of stable keypoints can be measured by *Repeatability* condition. Repeatability is calculated by the ratio between the total number of converted images and the number of cases where the converted keypoints are existent in the converted images.

$$p_{repeatability}(p_i) = \frac{n_i^{overlap}}{N} \tag{1}$$

where $n_i^{overlap}$ is calculated by the frequency of the existence of converted keypoint(p_i) in the set of keypoints($T(p_i) \in K_t'$) of converted images $T_t(I)$; N is the total number of converted images ; and all keypoints have single value.

Second, Good keypoints need to be well-matched with identical keypoints even though targeted images change in various ways(*Similarity* condition). With regard to a certain keypoint(p_i) of reference images, genuine distribution' and imposter distribution' for the corresponding keypoint can be measured by calculating the matching between the descriptors of all the sets of keypoints(p_i) in images($T_t(I)$) converted in various ways during the training process. At this time, to reduce the failure in matching the corresponding keypoints and the descriptors in the converted images, the genuine distribution needs to have small value, being far enough away from match distance threshold. To this effect, it was measured using the mean of genuine distribution. As shown in Equation (2), the keypoints with the decreasing the genuine distribution are better, so the evaluation function was calculated by normalizing the mean of the genuine distribution and subtracting its value from 1.

$$p_{similarity}(p_i) = 1 - \frac{\mu_{gen,i} - \min_i \mu_{gen,i}}{\max_i \mu_{gen,i} - \min_i \mu_{gen,i}} \tag{2}$$

Third, the trained keypoints and other keypoints shall not be matched(*Separability*), which is associated with the imposter distribution of each keypoint. Of the keypoints extracted from the images converted in various images, the distribution of the matching with other keypoints rather than the converted keypoints themselves are referred to as imposter distribution. Thus for a specific keypoint to show the low success rate of matching with other keypoints rather than themselves, in is necessary that the genuine distribution and imposter distribution are well classified. To this effect, in this dissertation, *Fisher's Discriminant Ratio[32]* was used. It measures the distance between two classes by the mean and distribution of sample in 1-dimensional, two class problems. Since the second Similarity condition ensures the genuine distribution is small enough, the nonexistence of the matching with the keypoints in the imposter distribution is ensured if the importer distribution is far enough away compared with the genuine distribution. Separability value also requires the normalization process as shown in equation (4).

$$FDR(p_i) = \frac{(\mu_{gen,i} - \mu_{imp,i})^2}{\sigma_{gen,i}^2 + \sigma_{imp,i}^2} \tag{3}$$

$$p_{separability}(p_i) = \frac{FDR(p_i) - \min_i FDR(p_i)}{\max_i s_i} \tag{4}$$

The score functions of each keypoint can be defined using 3 criteria calculated as above. The 3 conditions are dependent, so can be defined as shown in Equation (5).

$$gf(p_i) = p_{repeatability}(p_i) p_{similarity}(p_i) p_{separability}(p_i) \tag{5}$$

4. REFERENCES

[1] H.J. Chang, C. S G Lee, Yung-Hsiang Lu, and Y.C. Hu, "P-SLAM: Simultaneous localization and mapping with environmental-structure prediction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 281–293, Apr. 2007.

[2] AJ. Davison, ID. Reid, N.D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 2161–2168.

[4] M. Brown and D.G. Lowe, "Recognising panoramas," in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, Oct. 2003, pp. 1218–1225 vol.2.

[5] Daniel Wagner, Alessandro Mulloni, Tobias Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," in *2010 IEEE Virtual Reality Conference (VR)*, Mar. 2010, pp. 211–218.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007. ISMAR 2007*, Nov. 2007, pp. 225–234.

[7] Daniel Wagner, D. Schmalstieg, and H. Bischof, "Multiple target detection and tracking with guaranteed framerates on mobile phones," in *8th IEEE International Symposium on Mixed and Augmented Reality, 2009. ISMAR 2009*, Oct. 2009, pp. 57–64.

[8] Yang Cheng, M.W. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging," *IEEE*

Robotics Automation Magazine, vol. 13, no. 2, pp. 54–62, June 2006.

[9] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, June 2004, vol. 1, pp. I–652–I–659 Vol.1.

[10] Xuesong Le and Ruben Gonzalez, “Improving histogram-based image registration in video sequences through warping,” in *Advances in Multimedia Information Processing – PCM 2013*, Benoit Huet, Chong-Wah Ngo, Jinhui Tang, Zhi-Hua Zhou, Alexander G. Hauptmann, and Shuicheng Yan, Eds., number 8294 in Lecture Notes in Computer Science, pp. 269–280. Springer International Publishing, Jan. 2013.

[11] H. Goncalves, J.A. Goncalves, and L. Corte-Real, “HAIRIS: A method for automatic image registration through histogram-based image segmentation,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 776–789, Mar. 2011.

[12] Babu M. Mehtre, Mohan S. Kankanhalli, A. Desai Narasimhalu, and Guo Chang Man, “Color matching for image retrieval,” *Pattern Recognition Letters*, vol. 16, no. 3, pp. 325–331, Mar. 1995.

[13] Mohan S. Kankanhalli, Babu M. Mehtre, and Ran Kang Wu, “Cluster-based color matching for image retrieval,” *Pattern Recognition*, vol. 29, no. 4, pp. 701–708, Apr. 1996.

[14] S. Korman, D. Reichman, G. Tsur, and S. Avidan, “FasT-match: Fast affine template matching,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 2331–2338.

[15] Gerardo Carrera, Jesus Savage, and Walterio Mayol-Cuevas, “Robust feature descriptors for efficient vision-based tracking,” in *Progress in Pattern Recognition, Image Analysis and Applications*, Luis Rueda, Domingo Mery, and Josef Kittler, Eds., number 4756 in Lecture Notes in Computer Science, pp. 251–260. Springer Berlin Heidelberg, Jan. 2007.

[16] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[17] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu, “An optimal algorithm for approximate nearest neighbor searching fixed dimensions,” *J. ACM*, vol. 45, no. 6, pp. 891–923, Nov. 1998.

[18] J.S. Beis and D.G. Lowe, “Shape indexing using approximate nearest-neighbour search in high-dimensional spaces,” in *, 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings*, June 1997, pp. 1000–1006.

[19] M. Muja and D.G. Lowe, “Fast matching of binary features,” in *2012 Ninth Conference on Computer and Robot Vision (CRV)*, May 2012, pp. 404–410.

[20] Ruslan Salakhutdinov and Geoffrey Hinton, “Semantic hashing,” *International journal of approximate reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[21] Aristides Gionis, Piotr Indyk, and Rajeev Motwani, “Similarity search in high dimensions via hashing,” in *Proceedings of the 25th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1999, VLDB ’99, pp. 518–529, Morgan Kaufmann Publishers Inc.

[22] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li, “Multi-probe LSH: Efficient indexing for high-dimensional similarity search,” in *Proceedings of the 33rd International Conference on Very Large Data Bases*, Vienna, Austria, 2007, VLDB ’07, pp. 950–961, VLDB Endowment.

[23] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg, “Pose tracking from natural features on mobile phones,” in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, Washington, DC, USA, 2008, ISMAR ’08, pp. 125–134, IEEE Computer Society.

[24] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[25] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[26] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, “BRIEF: Binary robust independent elementary features,” in *Computer Vision – ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios, Eds., number 6314 in Lecture Notes in Computer Science, pp. 778–792. Springer Berlin Heidelberg, Jan. 2010.

[27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 2564–2571.

[28] S. Leutenegger, M. Chli, and R.Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[29] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 510–517.

[30] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1465–1479, Sept. 2006.

[31] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 448–461, Mar. 2010.

[32] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, Sept. 1936.