Incremental Feature Forest for Real-Time SLAM on Mobile Devices

Yuke Guo[1] and Yuru Pei[2(✉)]

[1] Luoyang Institute of Science and Technology, Luoyang, China
[2] Key Laboratory of Machine Perception (MOE),
Department of Machine Intelligence, Peking University, Beijing, China
Peiyuru@cis.pku.edu.cn

Abstract. Real-time SLAM is a prerequisite for online virtual and augmented reality (VR and AR) applications on mobile devices. Under the observation that the efficient feature matching is crucial for both 3D mappings and camera locations in the feature-based SLAM, we propose a clustering forest-based metric for feature matching. Instead of a predefined cluster number in the k-means-based feature hierarchy, the proposed forest self-learn the underlying feature distribution, where the affinity estimation is based on efficient forest traversals. Considering the spatial consistency, the matching feature pair is assigned a confident score by virtue of contextual leaf assignments to reduce the RANSAC iterations. Furthermore, an incremental forest growth scheme is presented for a robust exploration in new scenes. This framework facilitates fast SLAMs for VR and AR applications on mobile devices.

1 Introduction

The simultaneous localization and mapping (SLAM) play an important role in the VR and AR applications on mobile devices (Fig. 1). The SLAM has undergone rapid developments in recent years with an inception of several SLAM systems, such as PTAM [8], LSD-SLAM [6], and ORB-SLAM [10]. The feature-based SLAM is known to be effective for the 3D global mapping and camera locations, especially invariant to viewpoints and illuminations compared with the direct SLAM methods. A group of image features, including SIFT [9], SURF [1], BRIEF [4], ORB [14], and bag of words [7] have been used in feature-based SLAMs. The ORB feature has obvious advantages over others in fast extractions for the real-time SLAM. However, without the GPU and PC support, the ORB-SLAM has limited processing frame rates on mobile devices [15], which is not enough for online applications.

Considering the time-consuming feature matching for map generations as well as the camera locations in feature-based SLAMs, we investigate an adaptation of the ORB-SLAM by proposing a clustering forest for the fast feature correspondence establishment (see Fig. 2). Compared with the hierarchical vocabulary tree [10], there is no need to predefine the clustering number in the training phase

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11256, pp. 431–438, 2018.
https://doi.org/10.1007/978-3-030-03398-9_37

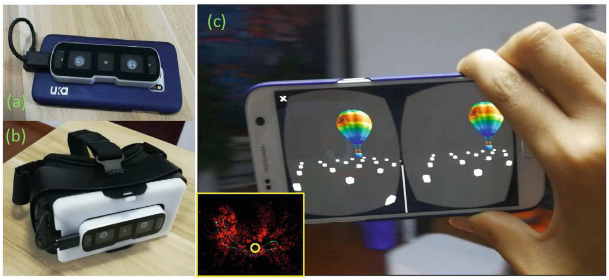

Fig. 1. Real-time feature-based SLAM on mobile devices. (a) A mobile phone mounted with a stereo camera Fingo. (b) A mobile phone on an HMD. (c) One sampled view of the hand-held mobile phone in the exploration of the virtual scene with a colored balloon. The 3D maps (red dots) are shown at the lower left corner along with the viewpoints of keyframes (green pyramids). The corresponding viewpoints are yellow circled in the 3D maps. (Color figure online)

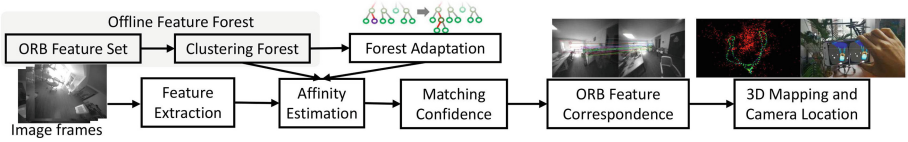

Fig. 2. Flowchart of the proposed forest-based feature matching for the SLAM on mobile devices.

of the feature forest. Moreover, there is just a limited number of binary comparisons in forest traversals for feature affinity estimation. Taking into account the spatial consistency, we propose a confident score for the feature matching by virtue of feature contexts. The matching pairs with similar contextual leaf assignments are assumed to be reliable. Furthermore, we present an incremental adaptation of the forest to accommodate newly-explored keyframes compared with the fixed vocabulary tree. The main point of this paper is to propose a forest-based method for efficient feature matching, and further the fast SLAM on mobile devices.

2 Feature Forest

The clustering forest works in an unsupervised manner without prior labeling, which is known for its self-learning underlying data distributions. The optimal node splitting parameters are learned by maximizing the information gain I as in the density forest [5]. We use the trace operator [12] to avoid the rank deficiency of the covariance matrix $\sigma(F)$ of the high dimensional ORB feature set F. Here we measure the information gain by the Hamming metric.

$$I = -\sum_{k=l,r} \frac{|F_k|}{|F|} \ln tr(\sigma(F_k)), \tag{1}$$

where $|\cdot|$ returns the cardinality of feature set F_k in left and right children nodes. The ORB feature is a 256-dimensional binary vector with each 8-bit byte serving as a feature channel. The binary function $\phi(s, \rho, \tau) = [\|f_{(s)} - \rho\|_h < \tau]$, where $[\cdot]$ is an indicator function. The features bearing channel $f_{(s)}, s \in [1, 32]$ with the Hamming distance to byte ρ lower than threshold τ is assigned to the left child node.

The forest is composed of five independent decision trees learned from randomly-selected feature subsets. The tree growths terminate when the number of instances inside the leaf node is below a predefined threshold γ, and $\gamma = 50$. Each tree has approx. 10 layers. Of course, the binary decision tree in the feature forest is deeper than the vocabulary tree. Fortunately, the forest traversals are extremely fast considering binary tests in branch nodes. Since the parameters of the hierarchical forest model are composed of binary tests in branch nodes, as well as the mean representor f_ℓ and instance number n_ℓ of the leaf nodes, it is easy to load the forest model into the memory of the mobile devices.

2.1 Affinity Estimation

When given the feature forest, it's straightforward to estimate pairwise affinities of ORB features. The ORB feature pair reaching the same leaf node is assumed to be similar with a distance set at 0, and 1 otherwise. The distance matrix $D = \frac{1}{n_T}\sum_{k=1}^{n_T} D_k$ by the forest with n_T trees, where $D_k(f_i, f_j) = 1$ if $\ell(f_i) = \ell(f_j)$. $\ell(f)$ denotes the leaf node of feature f. Given the distance matrix D between ORB feature set F_n of the newly-explored frame and F_o of the already stored keyframes, the feature matching

$$C = \{(f_i^n, f_j^o) | f_i^n \in F_n, f_j^o \in F_o\}, D(f_i^n, f_j^o) = \arg \min_{j' \in [1, |F_o|]} D_{ij'}. \tag{2}$$

The feature pair with the smallest pairwise distance is assumed to be the matching pair.

Note that, the pairwise distance entry is set according to binary functions ϕ stored in branch nodes. The balanced tree depth ν depends on the cardinality of the training data F, and $\nu = \log_2 |F|$. The time cost for the pairwise distance matrix between ORB feature set F_i and F_j is $O((|F_i| + |F_j|) \cdot \nu \cdot n_T)$. In our experiments, $\nu \in [9, 12]$ and $n_T = 5$. The time cost is lower than the common pairwise distance computation of ORB features with a complexity of $O(|F_i| \cdot |F_j|)$.

Similar to the vocabulary tree [10], the feature forest stores the direct and inverse indices between leaf nodes and features on keyframes. There are approx. $|F|/\gamma$ leaf nodes. The leaf index can be denoted by $\log_2(|F|/\gamma)$ bits. On the keyframes of already explored scenes, there is a direct index from the ORB feature to leaf nodes of the feature forest as shown in Fig. 3. On the other hand, the inverse index stores all the ORB features of keyframes that reach the leaf node. For the correspondence estimation between the newly-explored frame F_n and stored keyframes, just the forest traversals of F_n are needed with a complexity of $O(|F_n| \cdot \nu \cdot n_T)$ on byte-based binary comparisons. As we can see, the online distance matrix update cost for the newly-explored frame is extremely lower than

the common pairwise distance computation with a complexity of $O(|F_n| \cdot |F_o|)$. The time cost is also lower than the vocabulary tree with $O(|F_n| \cdot k \cdot \nu)$ of Hamming distance computations for the 256-dimensional features with k clusters for each splitting.

2.2 Matching Confidence

Considering the spatial consistency and perspective geometry, the correspondences of neighboring ORB features of one frame tend to be close in other frames or 3D maps. We no longer treat the matching pairs equally as in traditional features-based SLAMs. Instead, we present a confident score of the matching feature pair (f_i, f_j).

$$\alpha(f_i, f_j) = \frac{1}{Z} \sum_{k=1}^{n_T} \theta_k \left(\mathcal{N}(f_i)\right) \wedge \theta_k(\mathcal{N}(f_j)) \,, \tag{3}$$

where function $\theta_k(\mathcal{N}(f))$ returns leaf indices of surrounding context $\mathcal{N}(f)$ of feature f with respect to the k-th decision tree. The direct index of ORB feature as described in Sect. 2.1 is utilized to get the leaf index set of feature context $\mathcal{N}(f)$. The confident score is computed by the intersection $\wedge$ of the contextual leaf assignments of corresponding features f_i and f_j. Since decision trees in the feature forest are constructed almost independently, we consider all decision trees in the forest to measure the consistency of contextual leaf assignments. Z is a normalization constant. In our experiments, the size of the context patch is set at 1% of the image size. The matching pair is denoted as a triplet $\langle f_i, f_j, \alpha(f_i, f_j)\rangle$.

The feature pairs bearing large confident scores are likely to be correct matchings. The feature matchings are sorted according to the confident scores. The 3D mapping and camera location are prone to use the feature pairs with high confident scores. For instance, the RANSAC process for camera locations prefers the matching pairs with large confident scores. We observe that the weighted RANSAC using the confident scores is likely to terminate after a small number of iterations.

2.3 Online Forest Refinement

The feature forest is trained offline. When the scene exploration goes on, more and more keyframes and ORB features are located and stored. In this work, we present an online forest refinement scheme with incremental tree growths to accommodate the newly-added features on the keyframes, which facilitates the adaptation to the new scene. Similar to [13], we incrementally split the leaf nodes with available online data. There are two criteria to split the candidate leaf node in online forest refinements: (1) The number of newly-added features in the leaf node is larger than a predefined threshold, i.e. γ, the same as the predefined leaf size; (2) The deviation from the mean of the newly-added features $F_{n,\ell}$ to the offline learned leaf node representor f_ℓ is large enough.

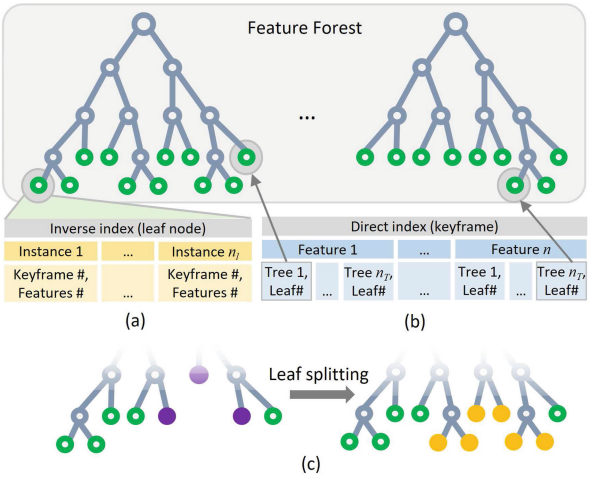

Fig. 3. (a) Inverse and (b) direct index of leaf nodes and keyframes. (c) The online forest refinement with incremental tree growths of leaf splitting. The nodes to be split are purple colored, and the newly-added nodes are orange-colored. (Color figure online)

We measure the deviation between f_ℓ and the representor f'_ℓ of its brother node. When $\|f_\ell - \bar{F}_{n,\ell}\| > \beta\|f_\ell - f'_\ell\|$, the second criterion is met. The constant coefficient β is set at 0.5. The leaf nodes of the feature forest is incrementally split and the tree grows when the above two criteria are met as shown in Fig. 3(c). The optimal splitting parameters are determined by maximizing the information gain as described in Sect. 2. Taking into account the features assigned to the leaf node in the training phase, we employ the weighted covariance matrix to estimate the information gain. The following weights are assigned to newly-added features $F_{n,\ell}$ and offline learned leaf node representor f_ℓ.

$$u_i = \begin{cases} \frac{1}{n_\ell + |F_n|}, \text{for} f_i \in F_{n,\ell} \\ \frac{n_\ell}{n_\ell + |F_n|}, \text{for } f_\ell \end{cases} \tag{4}$$

Different from the unweighted information gain estimation in the training phase (Sect. 2), the trace of the covariance matrix $\sigma(F_k)$ of the child node is defined as

$$tr(\sigma) = \sum_{i=1}^{|F_k|} \frac{u_i^2 \left\| f_i - \bar{F}_k \right\|_h^2}{\sum_{i',j}^{|F_k|} u_{i'} u_j}. \tag{5}$$

The center of the leaf node is computed as a weighted mean, and $\bar{F} = \sum_{i=1}^{|F|} u_i f_i$. Note that, the incremental tree growth changes the tree configurations, and the direct and inverse indices update accordingly. We keep a dynamic leaf node index list. The features in the already explored keyframes can be assigned to the online-split leaf nodes. Considering that the leaf node splitting just handles a limited number of instances, the leaf-splitting-based forest refinement is efficient enough for the online adaptation to new scenes.

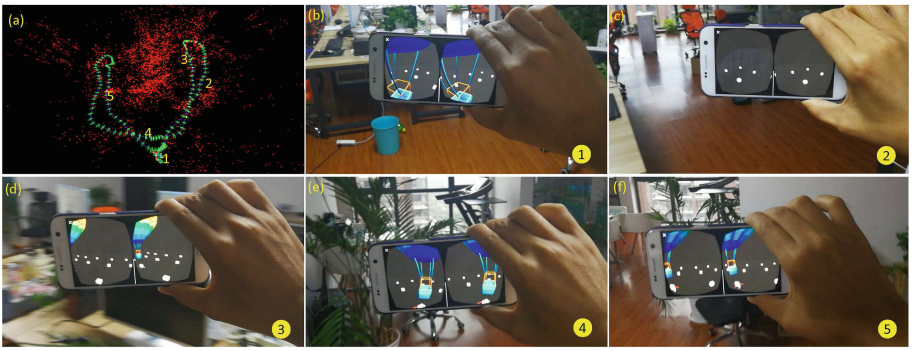

Fig. 4. (a) 3D map (red dots) with the connection of keyframes (blue lines) and viewpoints (green pyramids). (b)–(f) Sampled views with the hand-held mobile phone in the exploration of the virtual scene of a colored balloon with viewpoints (1–5) annotated in the 3D maps. (Color figure online)

3 Experimental Results

We perform experiments on the mobile device to evaluate the proposed method. We use Samsung Galaxy S7 with Snapdragon 820 processor 1.6 GHz and 4 GB RAM. The stereo gray images are captured by uSens Fingo camera as shown in Fig. 1(a, b). The proposed method establishes the feature correspondences in both 3D mapping and tracking processes by the feature forest. The proposed system works real-time and achieves up to 60 FPS without the common GPU and PC support.

Given the feature correspondence, the 3D maps and continuous camera locations are obtained as shown in Figs. 1 and 4. We test one virtual scene with a colored balloon and several white blocks. With the hand-held mobile phone, we can freely explore the virtual environments as shown in the supplemental video. We illustrate the feature matching between keyframes in Fig. 5. The proposed method is robust to obtain the ORB feature matching regardless of the viewpoint and illumination variations.

We report the precision and recall rates of the proposed feature forest (FF) and the incremental feature forest (IFF) with online refinement on public SLAM datasets, including New College [16], Bicocca25b [3], Ford2 [11], and Malaga6L [2] as listed in Table 1. The proposed IFF method achieves an improvement over the comparable bag of word (BoW) [7] and the FF methods.

We also report the precision and recall of the proposed FF and the IFF methods of different types indoor scenes, including the table/chair, the plant, and the poster as shown in Table 2. We observe that the posters with abundant textures have higher precision and recall rates than other types of objects. The IFF approach with online refinement produces an improvement over the original feature forest. We believe the reason is that the adaptation to the new scene enables the accurate affinity estimation and feature matching.

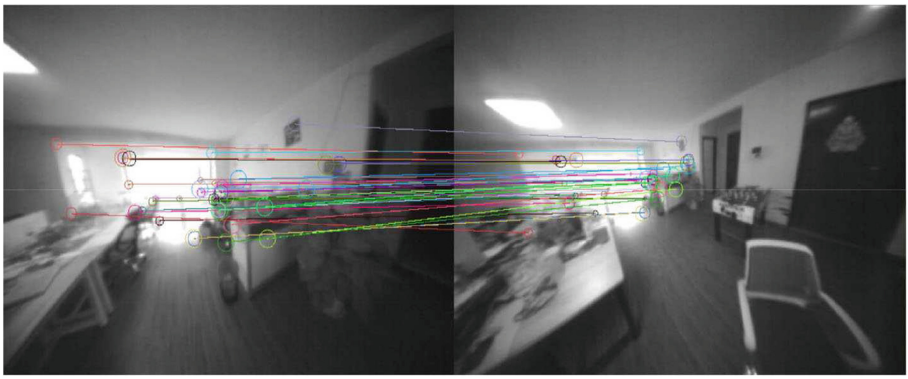

Fig. 5. Feature matching between keyframes. (Color figure online)

Table 1. Precision and recall.

Dataset	Precision (%)	Recall (%)		
		BoW [7]	FF	IFF
New College	100	55.9	63.2	**66.4**
Bicocca25b	100	81.2	81.5	**82.4**
Ford2	100	79.4	80.1	**81.1**
Malaga6L	100	74.7	73.2	**75.1**

Table 2. Precision and recall of indoor objects.

Dataset	Precision (%)		Recall (%)	
	FF	IFF	FF	IFF
Table/Chair	80.1	**82.6**	29.9	**30.5**
Plant	87.8	**90.5**	**40.1**	**40.1**
Poster	93.5	**95.9**	59.7	**60.7**

4 Conclusion

This paper presents a random-forest-based fast feature matching technique for the mobile device mounted SLAM. The proposed method takes advantage of the offline feature forest together with the online incremental forest adaptation for the feature affinity and matching confidences. The matching confident scores reduce the candidate searching space and facilitate the real-time SLAM for VR and AR applications on mobile devices.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_32
2. Blanco, J.L., Moreno, F.A., Gonzalez, J.: A collection of outdoor robotic datasets with centimeter-accuracy ground truth. Auton. Robots **27**(4), 327 (2009)
3. Bonarini, A., Burgard, W., Fontana, G., Matteucci, M., Sorrenti, D.G., Tardos, J.D.: Rawseeds: robotics advancement through web-publishing of sensorial and elaborated extensive data sets. In: Proceedings of IROS, vol. 6 (2006)
4. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_56
5. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Microsoft Research Cambridge, Technical report MSRTR-2011-114 **5**(6), 12 (2011)
6. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part II. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
7. Gálvez-López, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. IEEE Trans. Robot. **28**(5), 1188–1197 (2012)
8. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 225–234. IEEE (2007)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
10. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
11. Pandey, G., McBride, J.R., Eustice, R.M.: Ford campus vision and lidar data set. Int. J. Robot. Res. **30**(13), 1543–1552 (2011)
12. Pei, Y., Kim, T.K., Zha, H.: Unsupervised random forest manifold alignment for lipreading. In: IEEE International Conference on Computer Vision, pp. 129–136 (2013)
13. Ristin, M., Guillaumin, M., Gall, J., Van Gool, L.: Incremental learning of random forests for large-scale image classification. IEEE Trans. Pattern Anal. Mach. Intell. **38**(3), 490–503 (2016)
14. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: IEEE International Conference on Computer Vision, pp. 2564–2571. IEEE (2011)
15. Shridhar, M., Neo, K.Y.: Monocular slam for real-time applications on mobile platforms (2015)
16. Smith, M., Baldwin, I., Churchill, W., Paul, R., Newman, P.: The new college vision and laser data set. Int. J. Robot. Res. **28**(5), 595–599 (2009)