Efficient Decentralized Visual Place Recognition From Full-Image Descriptors

Titus Cieslewski and Davide Scaramuzza

***Abstract*— Visual multi-robot simultaneous localization and mapping (SLAM) is an effective way to provide state estimation to a group of robots that operate in an unstructured and GPS-denied environment. This is a problem that can be solved in a centralized way, but in some instances it can be desirable to solve it in a decentralized way. Decentralized visual place recognition, then, becomes a key component of a decentralized visual SLAM system. Achieving it by having all robots send queries to all other robots would use vast amounts of bandwidth, and diverse approaches have been explored by the robotics community to reduce that bandwidth. In previous work, we have proposed a decentralized version of bag-of-words place recognition, which, by using a distributed inverted index, is able to reduce bandwidth requirements by a factor of n, the robot count. In this short paper, we instead propose a decentralized visual place recognition method that is based on full-image descriptors. The method consists in clustering the full-image descriptor space into several clusters and assigning each cluster to one robot. As a result, place recognition can be achieved by sending each place query to only one robot. We evaluate the performance of our new method versus a centralized implementation using the Oxford Robotcar and KITTI datasets and explore an inherent trade-off between performance and load balancing.**

I. Introduction

Many robotic applications can benefit from parallel deployment of multiple robots. In a search and rescue mission for example, the search area can be subdivided, so that each robot has less space to cover, resulting in quicker task completion. In order for the robots to efficiently collaborate, they need to know where they are with respect to each other and to the environment. In unstructured, GPS-denied environments a popular method for this, which generalizes well to all kinds of different robots, is visual simultaneous localization and mapping (SLAM). Visual SLAM takes as input camera images and produces as output an estimate of the robot's trajectory as well as typically some crude representation of the environment (map).

Visual SLAM as deployed on a single robot has recently reached maturity [1]. We identify three components of a state-of-the-art visual SLAM system:

1) *Visual Odometry* is a real-time component that converts sensor measurements into a pose estimate considering only data from the most recent past. It is in the nature of visual odometry algorithms to exhibit drift [2]: metric accuracy decreases over time and if the robot returns to a place it has visited before, the current pose estimate will most likely be inconsistent with the previous pose estimate at that place.
2) To mitigate such drift, SLAM systems have a *Place Recognition* module which uses visual cues to recognize previously visited places in spite of the inconsistent pose estimate.
3) Once previously visited places have been recognised, an *Optimization* module incorporates them to make a consistent map. The optimization module can also be used without place recognition, to reduce linearization errors of the visual odometry.

The authors are with the Robotics and Perception Group, Dep. of Informatics, University of Zurich, and Dep. of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland—http://rpg.ifi.uzh.ch.

This research was funded by the DARPA FLA Program, the National Center of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation, the SNSF-ERC Starting Grant and the Armasuisse project 043-12.

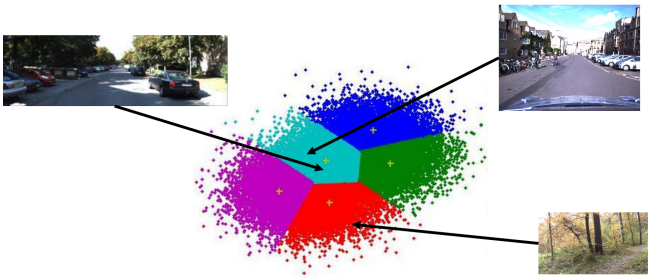


Fig. 1. We propose an efficient, full-image descriptor-based method for decentralized visual place recognition. The method consists in clustering the descriptor space of NetVLAD [9] into several clusters and assigning each cluster to one robot. As a result, place recognition can be achieved by sending each place query to only one robot. Image credit: Yi Cao / Mathworks, [10], [11]

How can these components be extended to multiple robots? It is important to note that in multi-robot SLAM both place recognition and optimization need to consider data from all robots to build a meaningful and consistent global map. Hence, a centralized system, where all the data is sent to a central instance that runs place recognition and optimization is a popular choice for such a system [3], [4], [5], [6], [7], [8].

In certain situations, however, it can be interesting to opt for a decentralized system. A centralized system, for example, has a computational bottleneck at the central station and thus limited scalability. A well-designed decentralized system could defy this bottleneck. Centralized systems furthermore typically require permanent or regular connection to a dedicated central machine and thus preclude for example applications where a group of lightweight robots goes deep into the field. Finally, there exist militaristic and privacy

arguments for using a decentralized system [12], [13].

Decentralized systems, then, have the challenge of implementing place recognition and optimization in a decentralized way that does not require too much communication: technically, it would be possible for every robot to share its data with every other robot, but that would typically require a lot of data to be transmitted. In our work, we focus on decentralized place recognition. For recent work in decentralized optimization instead, we refer the reader to [12], [14], [15].

In this short paper we propose an efficient, full-image descriptor-based method for decentralized visual place recognition. The core of the method consists in clustering the descriptor space of NetVLAD [9] into several clusters and assigning each cluster to one robot. As a result, place recognition can be achieved by sending each place query to only one robot.

II. Related work

Decentralized visual place recognition by sending queries to every other robot scales poorly in terms of the robot count n. The complexity of every query is $\mathcal{O}(n)$. Of course, the overall bandwidth can me mitigated by adopting one of many existing approaches to map compression: among others, visual maps can be compressed by pruning unnecessary map features [16], [17], reducing the dimensionality of feature descriptors [18], or using an overall non-canonical visual place representation, such as a frequency-domain place representation [19] or one that relies on object extraction [13]. These approaches, however, ultimately do not reduce the complexity in robot count.

In previous work [20], we have proposed a Bag-Of-Words based [21], [22], [23] decentralized place recognition algorithm that reduces the query complexity by one order. The algorithm has been inspired by distributed hash tables [24], [25], and some previous work in image retrieval [26], [27]. The method we proposed there, however, suffers from two drawbacks: firstly, the method is somewhat complex and uses an assumption whose full implications we do not yet fully understand. Secondly, the method requires for every query to have a message sent to every other robot (of size $\mathcal{O}(\frac{1}{n})$), thus still causing a lot of traffic.

In contrast, the method proposed here is much less complex, and, for every query, a message only needs to be sent to a single robot (plus to another robot for geometric verification, if place recognition succeeds). The key lies in substituting the bag-of-words approach with a full-image descriptor approach. This allows us to cluster the image descriptor space with k-means, and assign each cluster to a robot: any query from a cluster will be sent only to that robot. We show that this results in a competitive place recognition algorithm with minimal bandwidth requirements. We furthermore show how a problem of poor load balancing arises in practical deployment, and how it can be mitigated by sacrificing some recall. This work expands on self-published preliminary work that can be found at [28].

III. Methodology

A. Bag-of-Words method

In [20], we have shown how the data exchange incurred in decentralized visual place recognition can be reduced by a factor of up to n, the robot count. This can be achieved by casting the place recognition problem to a key-value lookup problem, which can be efficiently distributed using deterministic key-to-peer assignment, as is for example common in distributed hash tables [24], [29]. In [20], we have thus cast the bag-of-words (BoW) place recognition method [21], [22] used in [23], [30]. In broad strokes, this is how the resulting method works:

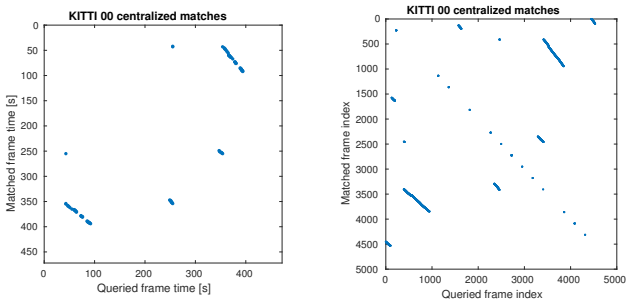
1) Before deployment, deterministically assign words of the visual vocabulary to the different robots.
2) When querying place recognition of an image frame, calculate the BoW vector and split it up into partial BoW vectors such that one partial BoW vector can be sent to each robot r, containing the coefficients of the words assigend to r.
3) The robots receive and process each their own partial query, returning the identity of the single frame which best matches the query frame according to the partial BoW vector. They also store the query, making it available as a result for subsequent queries.
4) Gather all partial results and determine which frame is most consistently returned as result.
5) Send a full query to the robot that has observed that frame for geometric verification.

The last step involving geometric verification serves the purpose of rejecting false positives of the method and can at the same time be used to establish relative pose between the query and matched image frames. We have shown that this methods results in a bandwidth reduction of up to n (depending on the network infrastructure), while reducing recall by $10 - 20\%$ depending on the robot count. A lot of the recall reduction is due to steps 3) and 4) of the method, which are based on a simplifying assumption that we do not yet fully understand. See Sec. IV C. and Fig. 4 of [20] for a detailed discussion.

B. Full image descriptor method

Here, we instead propose to use a full-image descriptor place recognition method as a basis. In particular, we use the recent, deep-learning based NetVLAD method [9] which has been shown to perform excellently even under severe appearance and viewpoint changes. Indeed, as can be seen in the centralized evaluation of this method (Fig. 2), its recall qualitatively looks better than the one of the BoW method we used in [20]. NetVLAD uses a deep neural network to calculate a low-dimensional feature vector $\vec{v} \in \mathbb{R}^d$ from an input image. Place matches can then be found by looking for the nearest vectors of other images according to the ℓ_2 distance.

This method can now efficiently be decentralized in the following way:



(a) Centralized recall in [20] (b) Centralized recall with [9]

Fig. 2. On the right, the confusion matrix for a centralized evaluation of the KITTI 00 dataset with 20 subtrajectories, using NetVLAD [9]. The threshold is manually selected and no geometric verification is performed. NetVLAD exhibits a visibly larger recall than the bag-of-words method we used in [20], on the left. The dots on the diagonal indicate place matches on the boundaries between sub-trajectories. Matches within the same trajectory are excluded.

1) Before deployment, cluster the feature vector space and assign each cluster center to a robot.
2) When querying place recognition of an image frame, calculate the feature vector and send it as query to only the robot assigned to the corresponding cluster.
3) That robot processes the query, stores it for future reference, and replies with the best matching frame identifier.
4) Send a full query to the robot that has observed that frame for geometric verification.

Similarly to our previous method, the last step rejects false positives and provides a method to find the relative pose between the query and matched image frames. Evidently, this method is less complex than the one proposed in [20]: before geometric verification, data is sent to only one robot, and no assumptions on the fidelity of partial responses are made. As for the geometric verification query, it is with the new method possible to skip it if the NetVLAD distance of step 3) exceeds a certain threshold, resulting in further bandwidth reduction at the cost of potentially reduced recall. We use k-means clustering as clustering method for step 1).

It is essential to note that every query that is sent to a robot is stored to the place recognition database of that robot for retrieval in future relevant queries. In [20], we have referred to this as *add-querying*. Since we assume that the robots constantly send place recognition queries, at any given time all places seen up to that time will be stored in the place recognition database of the appropriate robot. This is even robust to message delay: two queries of the same place sent by different robots will both arrive at some point at the robot responsible for that place, independently of the message delay, so at least one of the two robots will be notified of the place match.

C. Mitigating poor load balancing

Clustering the image feature space should ideally be done on a very general dataset, in order to account for deployment in many different environments. Deployment, on the other hand, can often occur in very specific environments, which

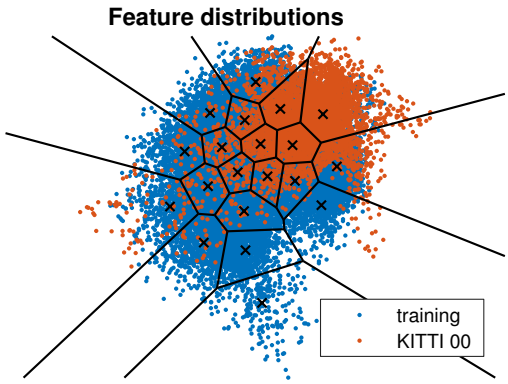


Fig. 3. The bad load balancing during evaluation can be explained by the difference in distribution of the image features. This is the distribution of the first two dimensions for the training and testing data. Training data should be more general / cover more environments than the deployment data, which could come from only a very specific kind of environment. The superimposed k-means clustering is not an actual clustering used in our method, but a 2D k-means clustering that serves as illustration. As we can see, clustering the training data can result in an uneven distribution of features among clusters at deployment time. This leads to poor load balancing.

only constitute a subset of the general, trained descriptor space, see Fig. 3. This poses an interesting problem: because the features at deployment time only come from a subspace of the trained feature space, there will be some clusters that will contain only very little features at deployment time, while other clusters will contain disproportionally many features from deployment time. In practice, this should translate in poor load balancing, since the robots assigned to the clusters with many features will have to handle much more queries than the other robots, which reduces scalability of the approach.

To mitigate this poor load balancing we propose to instead train several clusters per robot, and randomly assign the clusters to robots. This should distribute the robot's responsibilities in the feature space more evenly, and ensure that all robots are assigned features, even in narrow subspaces. However, this comes at the cost of increasing the amount of cluster boundaries, and, with that, the possibility that a matching pair of features is not in the same cluster. This would result in those features not being possible to match, and thus reduce recall.

IV. Experiments

Unlike in [20] we do not actully implement the method on multiple processes. It is evident form the method that it needs n times less data exchange than if all queries were sent to all robots. To evaluate the place recognition performance the method would have if deployed on a group of robots, we simply exclude all images that are not in the same cluster as the query from the pool of possible responses to a query. Furthermore, we evaluate our method without geometric verification, again unlike in [20]. Evaluation with geometric verification is on one hand closer to practical

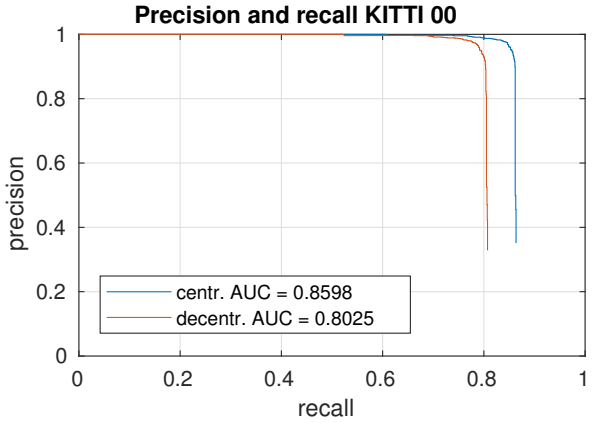


Fig. 4. We evaluate the place recognition performance using the area-under-curve measure (AUC) of the precision-recall curve, since we do not apply geometric verification. As we can see, NetVLAD exhibits excellent precision for the most part. We furthermore see how the clustering of the decentralized method results in reduced recall. This instance of decentralized place recognition has been run with 20 robots.

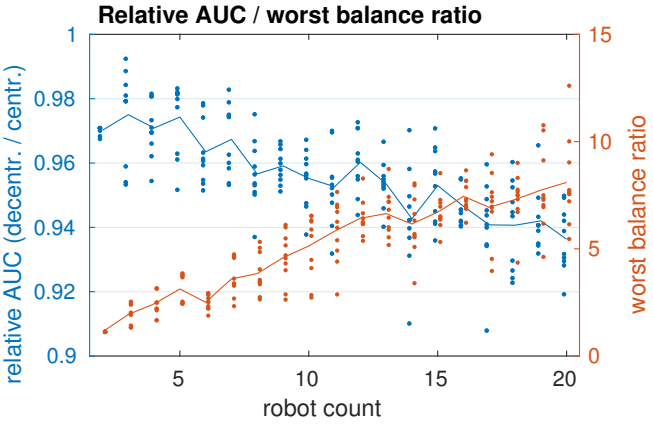


Fig. 6. Relative AUC (decentralized to centralized) and worst balance ratio of our method for different robot counts. The worst balance ratio is the ratio of the busy-ness of the most queried robot compared to what it would be if the feature-to-cluster assignments were perfectly balanced. Results are averaged over 10 runs and dots indicate the results of the individual runs.

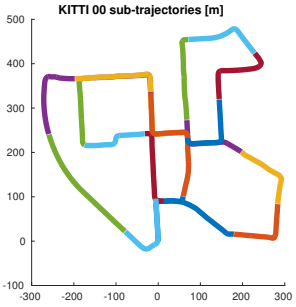


Fig. 5. The subtrajectories resulting from splitting KITTI 00 into 20 parts.

deployment, but on the other hand contaminates the results with the performance of the used geometric verification implementation. To provide a fair evaluation, we evaluate precision and recall for all possible feature vector distance thresholds and consider the area under that curve (AUC) as metric for place recognition performance, see Fig. 4. We use then NetVLAD feature vector dimension $d = 128$ (tunable thanks to a final layer that does principal component analysis). The clustering is trained on image data from the Oxford RobotCar dataset [11] and the method is evaluated on KITTI 00 [10] by splitting the sequence into n sub-sequences, one per robot, see Fig. 5.

We first evaluate the method in general, and show how it performs as we increase the number of robots. At the same time, we show how load balancing behaves as we increase the amount of robots. Then, we evaluate how using several clusters per robot can improve load balancing at the cost of performance.

V. Results

Fig. 6 shows relative AUC (decentralized to centralized) of the method when applied to groups of $n \in [2, 20]$ robots. Recall suffers if the true match of a query is not in the same cluster as the query. It would seem that the performance of the decentralized NetVLAD method is only marginally better than the performance of the decentralized BoW method (see Fig. 7 in [20]). Consider however that as qualitatively seen in Fig. 2, NetVLAD already has a higher recall than BoW in the first place. Furthermore, the method uses far less bandwidth for its distributed query than the BoW method. Recall from Table II in [20] that its distributed query size is 16 kilobytes plus overhead from sending the query to n robots. This method, when using single precision, only needs $d \times 4$ bytes per query, so 512 bytes with $d = 128$, plus overhead from only sending to one robot.

In Fig. 6 we furthermore report the *worst balance ratio*, a measure for how much more queries the busiest robot receives compared to how much it would receive if the queries were perfectly balanced. The experiments confirm the bad balancing discussed in Section III-C. As we can see, the busiest robot handles up to half of all queries!

Fig. 7 shows how both load balancing improves and performance depreciates as we increase the amount of clusters in the system.

VI. Conclusion

In this short paper, we have proposed an improvement over our previous work on decentralized place recognition [20]. The new method relies on recent, machine-learned full image descriptors and k-means clustering. We have explored how a problem of bad load balancing can arise when training and deployment feature distributions differ, and have shown how this problem can be mitigated by sacrificing some performance. Our method enables decentralized visual place recognition by sending only a lightweight query to a single other robot in the robot team. If a place is matched, a second query can be sent to the robot who observed the matching place for geometric verification.

Acknowledgement

We would like to thank Antonio Loquercio for the code reviews and helpful feedback.

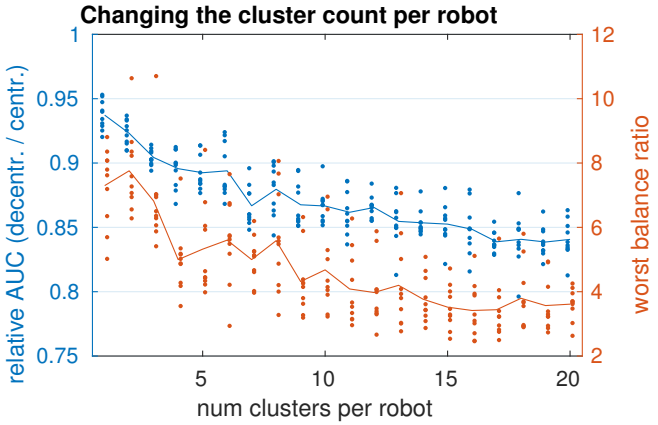


Fig. 7. Relative AUC (decentralized to centralized) and worst balance ratio as we increase the amount of clusters per robot, with 20 robots. Results are averaged over 10 runs and dots indicate the results of the individual runs.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, 2016.

[2] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, 2012.

[3] D. Zou and P. Tan, "Coslam: Collaborative visual slam in dynamic environments," *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[4] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013.

[5] L. Riazuelo, J. Civera, and J. Montiel, "C2tam: A cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems*, 2014.

[6] J. G. Morrison, D. Gálvez-López, and G. Sibley, "Moarslam: Multiple operator augmented rslam," in *Distributed Autonomous Robotic Systems*, 2016.

[7] M. Gadd and P. Newman, "Checkout my map: Version control for fleetwide visual localisation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.

[8] P. Schmuck and M. Chli, "Multi-uav collaborative monocular slam," in *IEEE Int. Conf. on Robotics and Automation*, 2017.

[9] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The Int. Journal of Robotics Research*, 2013.

[11] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000km: The oxford robotcar dataset," *The Int. Journal of Robotics Research*, 2016.

[12] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed trajectory estimation with privacy and communication constraints: a two-stage distributed gauss-seidel approach," in *IEEE Int. Conf. on Robotics and Automation*, 2016.

[13] ——, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *arXiv preprint arXiv:1702.03435*, 2017.

[14] A. Cunningham, V. Indelman, and F. Dellaert, "Ddf-sam 2.0: Consistent distributed smoothing and mapping," in *IEEE Int. Conf. on Robotics and Automation*, 2013.

[15] L. Paull, G. Huang, M. Seto, and J. J. Leonard, "Communication-constrained multi-auv cooperative slam," in *IEEE Int. Conf. on Robotics and Automation*, 2015.

[16] M. Volkov, G. Rosman, D. Feldman, J. W. Fisher, and D. Rus, "Coresets for visual summarization with applications to loop closure," in *IEEE Int. Conf. on Robotics and Automation*, 2015.

[17] M. Dymczyk, S. Lynen, M. Bosse, and R. Siegwart, "Keep it brief: Scalable creation of compressed localization maps," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

[18] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *Robotics: Science and Systems*, 2015.

[19] A. Jacobson, W. Scheirer, and M. Milford, "Deja vu: Scalable place recognition using mutually supportive feature frequencies," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017.

[20] T. Cieslewski and D. Scaramuzza, "Efficient decentralized visual place recognition using a distributed inverted index," *IEEE Robotics and Automation Letters*, 2017.

[21] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Int. Conf. on Computer Vision*, 2003.

[22] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[23] D. Galvez-Lopez and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Transactions on Robotics*, 2012.

[24] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, 2001.

[25] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, *A scalable content-addressable network*. ACM, 2001.

[26] R. Ji, L.-Y. Duan, J. Chen, L. Xie, H. Yao, and W. Gao, "Learning to distribute vocabulary indexing for scalable visual search," *IEEE Transactions on Multimedia*, 2013.

[27] X. Lin, Y. Shen, L. Cai, and R. Ji, "The distributed system for inverted multi-index visual retrieval," *Neurocomputing*, 2016.

[28] T. Cieslewski and D. Scaramuzza, "Efficient decentralized visual place recognition from full-image descriptors," *arXiv preprint arXiv:1705.10739*, 2017.

[29] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *IFIP/ACM Int. Conf. on Distributed Systems Platforms and Open Distributed Processing*, 2001.

[30] R. Mur-Artal, J. Montiel, and J. D. Tardós, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, 2015.