Fast, Compact and Highly Scalable Visual Place Recognition through Sequence-based Matching of Overloaded Representations

Sourav Garg and Michael Milford

Abstract— **Visual place recognition algorithms trade off three key characteristics: their storage footprint, their computational requirements, and their resultant performance, often expressed in terms of recall rate. Significant prior work has investigated highly compact place representations, sub-linear computational scaling and sub-linear storage scaling techniques, but have always involved a significant compromise in one or more of these regards, and have only been demonstrated on relatively small datasets. In this paper we present a novel place recognition system which enables for the first time the combination of ultra-compact place representations, near sub-linear storage scaling and extremely lightweight compute requirements. Our approach exploits the inherently sequential nature of much spatial data in the robotics domain and inverts the typical target criteria, through intentionally coarse scalar quantization-based hashing that leads to more collisions but is resolved by sequence-based matching. For the first time, we show how effective place recognition rates can be achieved on a new very large 10 million place dataset, requiring only 8 bytes of storage per place and 37K unitary operations to achieve over 50% recall for matching a sequence of 100 frames, where a conventional state-of-the-art approach both consumes 1300 times more compute and fails catastrophically. We present analysis investigating the effectiveness of our hashing overload approach under varying sizes of quantized vector length, comparison of near miss matches with the actual match selections and characterise the effect of variance re-scaling of data on quantization. Resource link:** https://github.com/oravus/CoarseHash

I. INTRODUCTION

Visual Place Recognition (VPR) is a key capability for a mobile robot, enabling it to localize itself within a known environment. The topic has been extensively researched for decades [1] with researchers exploring different aspects of the problem, such as dealing with appearance [2], [3] and viewpoint variations [4], [5], and large-scale localization [6], [7], [8] and navigation [9]. FAB-MAP [6] was one of the earliest VPR methods to demonstrate large-scale mapping, also incorporated into visual SLAM systems like LSD-SLAM [10]. Large-scale retrieval has also been a topic of significant interest in the computer vision community, leading to highly-scalable retrieval techniques like BoVW [11] and VLAD [12].

Such VPR and retrieval solutions are typically characterized by their ability to compactly represent places for low overall storage and linear growth in terms of time and memory requirements during deployment. These algorithms typically trade off storage footprint with computational requirements or vice versa to achieve high performance. A vast literature
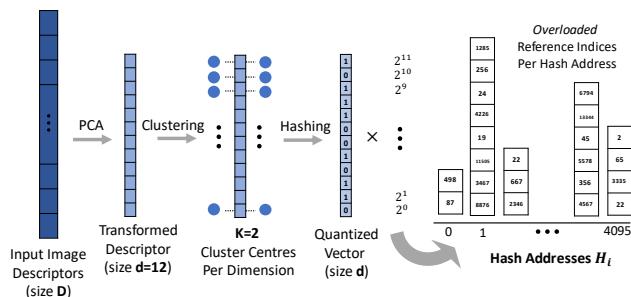

Fig. 1. *Coarse Hash*: Our hashing approach based on coarse scalar quantization generates short binary vectors (hash addresses) with long lists of inverted indices of reference data. When querying the hash space, collisions due to overloaded lists are resolved using sequence-based matching.

exists for developing compact place representations [13], [14] and demonstrating sub-linear scaling in both computation time [15], [16] and recently storage [17], [8] requirements.

In this paper, we present a novel visual place recognition solution that 1) generates and uses highly compact place representations leading to very small storage footprint, 2) benefits from extremely fast retrieval, 3) achieves near sub-linear growth in storage, and 4) has the capability to perform real-time localization in a map containing 10 million places with very lightweight computational requirements (37 kflops for 1 Hz performance) - the largest *sequential-nature* VPR benchmark to date. This unprecedented performance is achieved through massively overloading existing quantization and hashing techniques (ignoring the typical criteria used to optimize such techniques, see Figure 1), and then leveraging the sequential nature of robotic or autonomous vehicle data streams to disambiguate the resulting highly noisy single-frame matching performance.

Existing quantization methods have mainly focused on large vocabularies and longer resultant vectors to reduce the quantization error [18], [19]. This leads to better search accuracy and reduces the computation time requirements due to shorter lists of candidate matches which can then be re-ranked by superior techniques to find the best match [18], [20]. This approach is highly suitable for retrieval systems where the searched database and the query data are un-ordered. However, for data of a spatio-temporal nature, for example, in VPR and localization, a different approach to quantization can be used to leverage the sequential nature of the information. In particular, we invert the target criteria for quantization techniques and deliberately aim for coarse

quantization leading to much shorter vectors and longer lists of candidate matches. The typical disadvantage of long lists, a drastically increased probability of selecting a wrong match, is mitigated by the additional source of information in the form of temporal sequences, as also demonstrated conceptually in the original SeqSLAM [2]. Where SeqSLAM and follow on work leveraged sequences to disambiguate very challenging perceptual data, here we take the complementary approach of deliberately degrading matching performance to gain significant advantages in compute and storage requirements. This approach enables successful place recognition performance with extremely compact representations.

II. LITERATURE REVIEW

From large-scale place recognition and mapping solutions [21], [22] based on traditional feature representations [23], [24] to creating compact representations [13], [14], [25] and sequential-matching pipelines [16] based on deep-learnt image descriptors [26], [27], the visual place recognition literature has grown rapidly in recent years.

For large-scale image retrieval and place recognition, apart from retrieval techniques based on data structures like trees [28], [29], [30], [31] and graphs [32], [33], [34], [35], [36], quantization and hashing based methods have also been well explored. Vector quantization [37], Product Quantization [18], Scalar Quantization [38], [39], and their improved versions [19], [40], [41] have been demonstrated to be highly scalable in terms of computation time. Similarly, hashing [15], [42], [16], [43], [44] and efficient indexing [18], [20], [45], [46] techniques have been shown to be capable of retrieving accurate matches in nearly constant time. However, an overall low storage footprint is not always guaranteed with these methods. Furthermore, the aim for most of these methods is to either reduce the quantization error or to retrieve a short list of candidate matches for effectively finding the best match. With additional sequential information, some of the conditions of these existing systems can be relaxed to allow longer lists of candidate matches, which can then be filtered by well-established sequence-search techniques for VPR [2], [47], [48].

Within the context of VPR, some of the existing works that leverage sequential information include the use of sequential cyclic patterns [17] and binary tree encoding approach to directly infer the matched index [8]. Although achieving sub-linear growth in storage, these methods have their limitations due to certain assumptions: the former expects particular frequency patterns to occur within the encoded traverses and the latter requires the entire encoded traverse to maintain the order of adjacency. In our proposed system, we do not make any such assumptions about the data, instead we rely on unsupervised data transformation [38] that suits the following scalar quantization-based encoding [39], [40].

III. PROPOSED APPROACH

The existing quantization and hashing techniques for large-scale retrieval mainly focus on unordered datasets [18],

[19], [38]. In the context of visual place recognition and localization, the underlying datasets are usually sequentially ordered [2], [6], [49]. Our proposed VPR pipeline demonstrates the effective use of hashing and inverted-index lists for sequential data. In particular, we show that using coarse scalar-quantization based hashing [38], one can allow a large number of collisions which can then be resolved by sequence-based matching [2]. This enables storage-efficient encoding of the image descriptors with much shorter codes and correspondingly long inverted lists of candidate matches. This is in contrast to the existing state-of-the-art large-scale retrieval methods where relatively longer codes and short lists are often the preferred choices to achieve high performance [18], [45]. In the following, we first describe the quantization process for the reference and the query datasets and then present the sequence-based matching approach.

A. Reference Data to Hash Addresses

The reference data for VPR, available beforehand in online operations, is quantized and hashed to an integer address space where each hash address is linked to multiple reference image indices. This is obtained as described below:

1) Image to Descriptor: The raw image data of size N_x is first converted into D-dimensional global image representations using a state-of-the-art image description technique [26], [50], [51]. We used NetVLAD [26] for this purpose, however, any other method can be used as a drop-in replacement.

2) PCA Transformation: The $N_x \times D$ feature matrix obtained above is then transformed into a decorrelated orthogonal space using PCA. We use Incremental PCA [52], [53], [54] for this purpose to keep the transformation computationally tractable. We only retain the first d components of the transformed feature matrix.

3) Hashing: The transformed feature matrix $N_x \times d$, being decorrelated and mutually-orthogonal, can be independently quantized along each of its dimensions [39], [38]. This scalar quantization per dimension can be obtained using K-means clustering, K being the number of quantization bins. While we keep K fixed for every dimension, we note that some existing approaches use different values of K depending on the variance along that dimension [38], [43], [40].

$$q_{x_{ij}} = \arg\min_{k \in [1,K]} |x_{ij} - c_{kj}| \quad \forall j \in [1, d] \tag{1}$$

where x_i and q_{x_i} represent the transformed reference image descriptor and its quantization index vector respectively and c_k represents the kth cluster center along the jth dimension.

The quantization index vectors are converted into integer hash addresses as below:

$$h_{x_i} = \sum_{j=1}^{d} K^{(d-j)} q_{x_{ij}} \tag{2}$$

At each of these hash addresses, multiple image indices are stored as a list. The maximum number of addresses possible are K^d, however, in practice, a number of these addresses comprise empty lists due to collisions at other hash addresses.

B. Query Searching

Query images are first converted into global representations using the same description method as for the reference data. The transformation matrix obtained from PCA training of the reference data is used for transforming the query descriptors to d-dimensional vectors. A hash address for a given query vector is obtained using Equation 1 and 2. Therefore, a list of matched reference indices for a given query image can be obtained with the computational complexity of $\mathcal{O}(1)$. The list of matched indices represents the candidate matches for the sequence-based filtering described in the subsequent section. For single frame-based matching, we only store the best match corresponding to a hash address instead of a list of reference indices. This is obtained on the basis of minimum quantization error:

$$i_{single} = \arg\min_{i \in [1, N_x]} \sum_{j=1}^{d} \min_k |x_{ij} - c_{kj}| \qquad (3)$$

which can be computed along with the hashing of reference data before the query phase begins. As no further computation is needed during the query search, search complexity of $\mathcal{O}(1)$ is retained. We highlight the effect of the above selection procedure on single frame-based matching performance in Section V.

Queries Landing at Unoccupied Hash Addresses: Due to perceptual aliasing, images (and their corresponding descriptors) obtained from revisited places are not guaranteed to exactly match to their ground truth in the reference database. Therefore, a quantization index vector of a query image can lead to a hash address which is not associated to any reference image index (or a list of indices). Such queries are assigned the nearest occupied hash address (numerically closest) from the sorted list of all the occupied addresses. This search is completed in $\mathcal{O}(\log_2 H_o)$ time where H_o is the number of occupied addresses. Alternatively, it would also be possible to pre-compute and store the nearest neighbours in the hash address space during the training, representing a different operating location in the trade-off between storage footprint and query time.

C. Sequence-based Matching

Inspired by SeqSLAM [2], we use a similar strategy of disambiguating place matches using sequential information. A combined set of reference candidates obtained from the lists of potential matches of a query sequence are probed to find the best match. For this purpose, the distance between a quantized query and reference vector is defined as below:

$$\delta(x_i, y_t) = \sum_{j=1}^{d} \left| c_{jq_{x_{ij}}} - c_{jq_{y_{tj}}} \right| \qquad (4)$$

where $q_{x_{ij}}$ and $q_{y_{tj}}$ refer to the cluster centers assigned to the jth dimension of the reference and query vectors x_i and y_t respectively. q_{x_i} for any reference image is directly obtained from the base K representation of its hash address h_{x_i} (see Equation 2). The distance calculation in Equation 4

is similar to the Symmetric Distance Computation (SDC) defined in [18].

For a given L-length sequence of quantized query vectors centered at y_t, the lists of matching reference indices are obtained from their respective hash addresses h_{y_t}. The set of unique reference indices, r, obtained from these lists is then used for sequence searching and the best match is obtained as below:

$$i_{seq} = \arg\min_{i \in [1, N_r]} \sum_{l=-L/2}^{L/2-1} \delta(r_{il}, y_{tl}) \qquad (5)$$

where r_i represents a shortlisted candidate from a set of size N_r. Here, we assume a constant velocity between consecutive samples of reference and query data.

IV. EXPERIMENTAL SETUP

A. Datasets

We used two types of datasets in our experiments: a newly collected large-scale localization dataset - FAS100K and a commonly used benchmark dataset for large-scale image retrieval - Deep1B [20]. The datasets chosen are a result of the sparsity of very large scale spatial navigation datasets: we describe pre-processing, benchmarking and analysis in depth to show that our treatment of the data is valid and appropriate.

a) FAS100K: This dataset is comprised of two traverses of 238 and 130 kms respectively where the latter is a partial repeat of the former. The data was collected using stereo cameras in Australia under sunny day conditions. It covers a variety of road and environment types including urban and rural areas. The raw image data from one of the cameras streaming at 5 Hz constitutes 63650 and 34497 image frames for the two traverses respectively. We sub-sample these image sets with GPS information such that consecutive image frames are 5 meters apart. The sub-sampled data of size 47781 and 26638 respectively form the reference and query traverses for our experiments. The images from both the traverses are converted into 4096-dimensional global descriptors using the NetVLAD [26] representation.

b) Deep1B: This is a recently introduced 1 billion image descriptor dataset [20] comprising 96-dimensional PCA-transformed descriptors. The original images or their hyperlinks are not publicly available, however, the dataset is one of the largest of its kind and is typically used for benchmarking large-scale nearest neighbor retrieval [20], [55], [56], [57]. Unlike localization datasets [49], [2], [6], these 1 billion descriptors are unordered and are temporally unrelated. In order to use this dataset in conjunction with FAS100K for our localization experiments, we perform pre-processing to create three new datasets of varying size: 20K, 1M, and 10M, comprising approximately $20,000$; $1,000,000$; and $10,000,000$ descriptors in both reference and query sets.

c) 20K, 1M, and 10M: These three new 'localization' datasets use FAS100K and Deep1B in parts (see Table I). For the reference traverse, 20K uses the first 10K samples from both Deep1B and FAS100K reference data (out of

47781). 1M and 10M use the first 1 million and 10 million samples from Deep1B respectively and the entire reference data from FAS100K, leading to $1,047,781$ and $10,047,781$ reference descriptors. For the query counterparts of these reference datasets, the Deep1B part of the data in each of the three cases is re-used but with two different noise models of varying noise intensity, described later in this section. These Deep1B query datasets are then appended with FAS100K query data (out of 26638): only the first 10k for the 20K dataset, and the entire query traverse for 1M and 10M datasets. Before concatenation, the source datasets are pre-processed as described in the following section.

B. Data Pre-Processing and Concatenation

We perform the following pre-processing before concatenating the Deep1B and FAS100K datasets to obtain either of 20K, 1M, and 10M datasets:

a) Homogenization: As also indicated earlier, the raw adjacent samples in the Deep1B descriptor data are unrelated, unlike typical localization datasets. In order to emulate local temporal perceptual similarity within the Deep1B dataset, we perform a sliding window average of the raw descriptors using a window size w to make the data somewhat locally similar. w is set to 40 in our experiments which is equivalent to 200 meters in the FAS100K spatial dataset. The homogenization enables the modified dataset to behave like a localization dataset in terms of matching performance in a region. Furthermore, this makes the Deep1B dataset more appropriately challenging: in its raw form, sequences of its unordered data are highly distinctive and can be easily matched.

b) Aligning Descriptor Dimensions and Variance: For the Deep1B dataset, neither the original images corresponding to its descriptors nor the PCA transformation matrix are publicly available. Hence, for any of the combined datasets (20K, 1M, and 10M), we concatenate the Deep1B and FAS100K datasets following a two step procedure. 1) PCA training is done independently on the reference data of Deep1B and FAS100K to match the descriptor length to $D = 96$ and to obtain mutually decorrelated descriptor components ordered by their variance. 2) As the distribution of variance across the principal components for both the datasets is different, we re-scale the standard deviation of transformed Deep1B dataset to match it to FAS100K data before finally concatenating them. For the query data counterparts, in case of FAS100K, the query data is transformed using the PCA parameters of FAS100K training data, and in case of Deep1B, the noisy version (explained below) of the PCA-transformed and variance-equalized Deep1B data is used. Figure 2 shows how the standard deviation for PCA-transformed Deep1B descriptors is matched to the PCA-transformed FAS100K descriptors.

c) Noise Model for Query Data: We add noise to the modified Deep1B data to obtain a corresponding query dataset. The noise is added from a random normal distribution with mean μ_n and variance σ_n^2. The mean and variance are calculated from the FAS100K dataset which is first
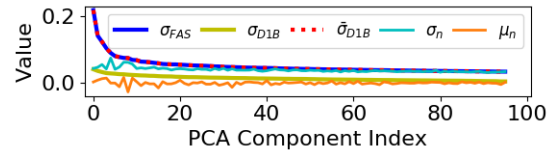

Fig. 2. Standard deviation of PCA-transformed Deep1B descriptors σ_{D1B} is modified to $\bar{\sigma}_{D1B}$ to match with PCA-transformed FAS100K descriptors σ_{FAS} before concatenation to form the 20K dataset.

transformed using PCA to match the dimension size D of the Deep1B dataset, that is, 96:

$$\Delta_f = x_i^f - y_i^f, \ \ \mu_n = \sum_i^{N_f} \frac{\Delta_f}{N_f}, \ \ \sigma_n^2 = \sum_i^{N_f} \frac{(\Delta_f - \mu_n)^2}{N_f} \tag{6}$$

where x_i^f and y_i^f are corresponding descriptor pairs from the FAS100K dataset and N_f is 26638. x_i^f, y_i^f, μ_n, and σ_n are all 96-dimensional vectors. Figure 2 shows the mean and standard deviation of noise compared against the standard deviation of merged datasets. We use two variants of the query data in our experiments: QM1 with distribution parameters μ_n and σ_n and QM2 with parameters μ_n and $2\sigma_n$. This is done to study the impact of noise on the performance. Note that the noise is only added to the Deep1B (not the FAS100K which already comes with natural image variation) parts of the full query datasets.

TABLE I
NEW LOCALIZATION DATASETS

Dataset Size	Reference Data		Query Data	
	Deep1B	FAS100K	Deep1B	FAS100K
20K	10,000	10,000	10,000	10,000
1M	1,000,000	47781	1,000,00	26638
10M	10,000,000	47781	10,000,00	26638

C. Baseline for Comparative Study

We compare our proposed system with a baseline system which can be considered as a modified version of SeqSLAM [2]. While both the systems are benchmarked on the same datasets: 10K, 1M, and 10M, both approaches are constrained to the same minimal storage footprint: in the case of the baseline, by limiting the number of descriptor dimensions. We perform PCA on the input dataset, similar to the process described in Section III, and only retain the initial principal component(s) for the baseline system - this enables a fair comparison under similar storage constraints as the first principal component divides the whole data with maximum variance.

A second constraint on the baseline system is imposed during the sequence matching process for equating the computation time of sequence searching. As in SeqSLAM, the baseline system linearly searches for matching candidates. However, we limit the total number of candidates to match with based on the average number of candidates shortlisted by our proposed approach N_r.

D. Parameters Settings

In our experiments conducted on the three datasets: 10K, 1M, and 10M, we use the following parameter settings: 1) The input descriptor dimensions for all the datasets is fixed to $D = 96$; 2) the PCA-transformed descriptor dimension d is set to 12, 20, and 24 for the three datasets respectively for our proposed approach, while for the baseline system d is set to 1 to match the storage footprint; 3) Sequence length L used for different experiments is 1, 50, and 100 where 1 implies single image based retrieval; 4) The number of cluster centers per dimension K is fixed to 2 for our proposed system for all the experiments. In order to keep the running time of experiments tractable, we only query a single frame or a sequence of frames every zth index. z is set to 10, 100, and 10000 for 20K, 1M, and 10M dataset respectively.

E. Evaluation

For benchmarking the two methods, we use recall rate which is often used for evaluating place recognition [51] and image retrieval systems [18], [20]. Recall rate is defined as the ratio of correctly matched queries to the total number of queries within a given localization radius. The localization radius is varied from 0 to 20 frames which is equivalent to a maximum of 100 meters for the 5 meter frame separation in the FAS100K section of the datasets and half-window size $w/2$ for the Deep1B chunk of the datasets.

V. RESULTS

a) Proposed vs Baseline: Figure 3 shows the performance comparison between our proposed system and the baseline. It can be observed that the baseline system performs very poorly under the same storage constraints as compared to our system, despite using different query noise models and varying sequence lengths. It can further be observed that more noise leads to a faster reduction in performance when scaling up to very large datasets. Similarly, the reduction in the performance of the baseline system scales up with the size of the datasets; even the use of a longer sequence length does not recover much performance. From the FAS100K part of the query data (267 queries) of the 20K dataset, the baseline was only able to retrieve 1 successful match within a localization radius of 100 meters whereas the proposed system correctly recalled 98 matches.

b) Longer Candidate Lists and Sequence Matching: Figure 4(a) shows the variation in performance when a relatively shorter vector length d is chosen for 1M dataset. A short quantization vector leads to longer overloaded lists of matches N_r which can then be effectively filtered by the sequence matching process. It can be observed that as d decreases, N_r increases and leads to high performance. However, the baseline system, despite the availability of more candidate matches, does not perform well.

c) Single Frame Matching - Best Match Selection: For single frame matching, the selection of the best match out of the list of matches is based on the minimum quantization error as calculated using Equation 3. In Figure 4(b), we compare the percentage of correctly selected matches with the percentage
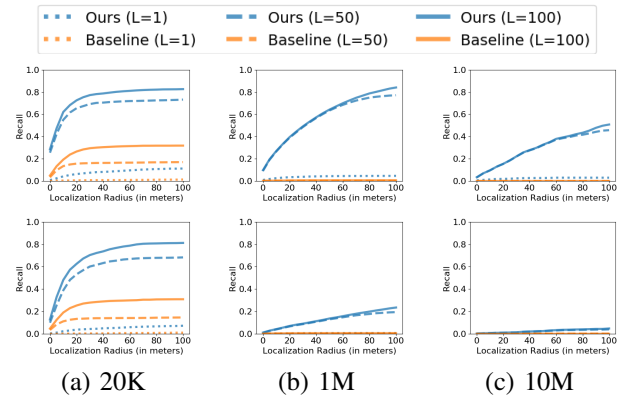

Fig. 3. Performance comparisons using two different noise models for the query data: QM1 (top) and QM2 (bottom). Line style represents different sequence lengths.

of correct matches that existed *somewhere* in the list of candidates, which can be regarded as a maximal upper bound on performance. This result indicates that the proposed system is generally able to achieve a higher recall when considering the correct matches within the list of candidates. The recall rate could be further improved beyond our match selection technique by using a re-ranking based on full descriptor matching [18] or geometric verification [51]. Furthermore, with the increasing size of the dataset, the length of the quantized vector is also chosen to be proportionally longer. A longer quantization vector generally leads to a sparser distribution of reference indices with shorter lists of candidate matches. Therefore, the scope for further improving the single frame performance on very large datasets diminishes with the size of the dataset.

d) Storage Growth and Overall Footprint: Figure 4(c) shows a comparison between the storage growth of different stored components of the proposed system, namely P1: Reference Indices to Hash Address Map, P2: Cluster Centers, P3: PCA Transformation Matrix, and P4: Mean Reference Descriptor. The absolute storage is represented as $\log_n$ of the raw values for visual clarity. It can be observed that P4 has a constant storage as it is proportional to the length of reference descriptors D; P2 and P3 grow sublinearly with the size of the database as they only depend on the choice of number of cluster centers K and length of quantization vector d; P1 takes up the bulk of storage space as it is directly proportional to the the size of the reference database N_x. The overall storage used for the three datasets: 20K, 1M, and 10M for ours and the baseline system was 0.2, 8.4 and 80.4 MB respectively.

VI. DISCUSSION

A. Computation Time Analysis

Our proposed system is extremely fast as compared to the baseline system. The total number of unitary operations (additions and multiplications) required during query searching for both the systems are compared below:

1) PCA transformation: The subtraction of mean vector and matrix multiplication for PCA transformation require D
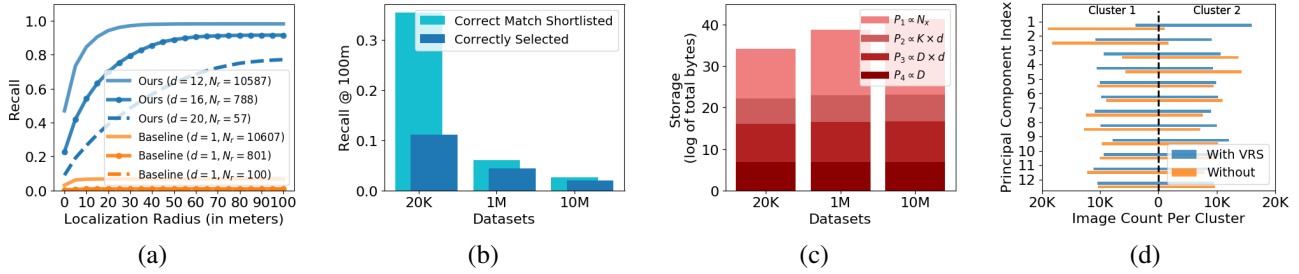
Fig. 4. (a) Performance variation for the 1M dataset with respect to vector lengths d. (b) Single Frame Matching: Performance comparison between the selected best match and the list of candidate matches using QM1. (c) Storage growth with respect to dataset size. (d) Cluster distribution of reference data (20K) across principal components of its binary vector both with and without variance re-scaling of the Deep1B dataset for unbiased dataset concatenation.

and $d(2D - 1)$ operations respectively. d for the baseline system is chosen to be smaller than that for the proposed system to match the overall storage footprint. As $d \ll D \ll N_x$, the computational advantage for baseline is minimal.

2) Quantization and Hashing: Quantizing a given query vector requires $d(2K - 1)$ operations for assigning the cluster centers as defined in Equation 1. Further, $2d - 1$ operations are required to obtain a hash address from the quantized vector as defined in Equation 2. These two steps are only required for the proposed system. However, these computations do not depend on the size of the database N_x and are extremely fast in practice.

3) Lists of Reference Candidates: For the proposed system, this is achieved by searching for the hash address key in the dictionary with values as the lists of reference indices. Hence, the computational complexity is $\mathcal{O}(1)$. For the baseline system, a query descriptor is compared against all the reference descriptors using Euclidean distance and requires $3dN_x$ operations. A list of candidates is then obtained by retaining N_r candidates with lowest Euclidean distance.

4) Sequence Matching: For a practical VPR scenario, sequence matching can be performed online requiring incremental computations for newly observed query images. Therefore, for the baseline and the proposed system, only $3N_r - 1$ computations are required if cumulative sequence scores are stored for corresponding pairs of reference and query image indices. Unlike the baseline method, an exhaustive matching between query and reference data is not performed for the proposed system as it obtains a list of matching candidates from its hash address. Hence, for a paired sequence defined in Equation 5 and the distance function defined in Equation 4, the proposed system requires $L'N_r(d/p + d - 1)$ operations for sequence matching where L' is the number of new pairs, p is the number precision used which is 64-bit for all the experiments, and d/p represents the bitwise xor operation to find the distance in address space (Equation 4). As $L' < L \ll N_x$ and $N_r \ll N_x$, the computation time for this step for the proposed system is significantly smaller than the candidates' retrieval time for the baseline system described in the previous step which is $\mathcal{O}(N_x)$.

B. Possible Throughput

The compute time for our method primarily depends on N_r which depends on the size of both the database (N_x) and the

hash address space. The latter is given by K^d. With $K = 2$, we fix d so that it is just above $\log_2 N_x$ which comes out to be 15, 20, and 24 respectively for the 20K, 1M, and 10M datasets. In an ideal scenario with a uniform distribution of reference indices, there would be no more than 1 reference index per hash address. However, in practice, we found the average number of reference indices (N_r) for a sequence of 50 frames to be 74, 57, and 32 for the three datasets, indicating a sub-linear growth in N_r with respect to N_x. For the 10M dataset with $d = 24$, $L' = L = 50$, and $N_r = 32$, sequence matching requires 37400 unitary operations. On a hardware platform like a Jetson TX2, capable of 1.3 TFLOPs [58], the proposed method could potentially localize within the 10M reference database at a rate of 35 **MHz**.

C. Unbiased Dataset Concatenation

Figure 4(d) shows the distribution of reference indices across each of the dimensions (principal components on vertical axis) of the quantized vector for the 20K dataset. As we use two cluster centers per dimension, the left and the right side of the vertical dashed line represent clusters 1 and 2 respectively. Color indicates whether the variance for Deep1B dataset was re-scaled (blue) or not (orange), as described in Section IV-B. It can be observed that re-scaling of the variance leads to a uniform distribution of reference data across most of the dimensions. This indicates that concatenation of the Deep1B and FAS100K datasets does not favor any particular section of the 20K dataset. However, clustering is as expected imbalanced when no re-scaling is performed (orange) particularly for the first few components which leads to a disproportionate distribution of reference indices to hash addresses, and consequently poor performance.

VII. CONCLUSION

In this paper, we have demonstrated a highly-scalable VPR pipeline that uses coarse scalar-quantization based hashing, leading to long lists of inverted reference indices due to shorter quantization vectors. The collisions in the hash space due to overloaded lists are then resolved by sequence-based matching. Our proposed system exhibits: low overall storage footprint, extremely fast retrieval, and near sub-linear storage growth with increasing size of the reference database, demonstrated on a new 10 million place dataset of *sequential nature*.

REFERENCES

[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[2] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1643–1649.

[3] T. Naseer, W. Burgard, and C. Stachniss, "Robust visual localization across seasons," *IEEE Transactions on Robotics*, 2018.

[4] S. Garg, N. Suenderhauf, and M. Milford, "Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics," in *Proceedings of Robotics: Science and Systems XIV*, 2018.

[5] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1687–1694, 2018.

[6] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[7] A.-D. Doan, Y. Latif, T.-J. Chin, Y. Liu, T.-T. Do, and I. Reid, "Scalable place recognition under appearance change for autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9319–9328.

[8] H. Le, T. Hoang, and M. J. Milford, "Btel: A binary tree encoding approach for visual localization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4354–4361, 2019.

[9] M. Chancán and M. Milford, "Mvp: Unified motion and visual self-supervised learning for large-scale robotic navigation," *arXiv preprint arXiv:2003.00667*, 2020.

[10] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of International Conference on Computer Vision (ICCV)*. IEEE, 2003, p. 1470.

[12] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[13] S. Lowry and H. Andreasson, "Lightweight, viewpoint-invariant visual place recognition in changing environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 957–964, 2018.

[14] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Fusion and binarization of cnn features for robust topological localization across seasons," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4656–4663.

[15] A. Gionis, P. Indyk, R. Motwani, *et al.*, "Similarity search in high dimensions via hashing," in *Vldb*, vol. 99, no. 6, 1999, pp. 518–529.

[16] O. Vysotska and C. Stachniss, "Relocalization under substantial appearance changes using hashing," in *Proceedings of the IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles, Vancouver, BC, Canada*, vol. 24, 2017.

[17] L. Yu, A. Jacobson, and M. Milford, "Rhythmic representations: Learning periodic patterns for scalable place recognition at a sublinear storage cost," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 811–818, 2018.

[18] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[19] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 4, pp. 744–755, 2013.

[20] A. Babenko and V. Lempitsky, "Efficient indexing of billion-scale datasets of deep descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2055–2063.

[21] R. Paul and P. Newman, "Fab-map 3d: Topological mapping with spatial and visual appearance," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2649–2656.

[22] N. Kejriwal, S. Kumar, and T. Shibata, "High performance loop closure detection using bag of word pairs," *Robotics and Autonomous Systems*, vol. 77, pp. 55–65, 2016.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[25] M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, and M. Milford, "A hybrid compact neural architecture for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 993–1000, 2020.

[26] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[27] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," in *Australasian Conference on Robotics and Automation*, vol. 2, 2014, p. 4.

[28] H. Lejsek, B. T. Jónsson, and L. Amsaleg, "Nv-tree: nearest neighbors at the billion scale," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, 2011, pp. 1–8.

[29] T. Liu, A. W. Moore, K. Yang, and A. G. Gray, "An investigation of practical approximate nearest neighbor algorithms," in *Advances in neural information processing systems*, 2005, pp. 825–832.

[30] Y. Hou, H. Zhang, and S. Zhou, "Tree-based indexing for real-time convnet landmark-based visual place recognition," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, p. 1729881416686951, 2017.

[31] D. Schlegel and G. Grisetti, "Hbst: A hamming distance embedding binary search tree for feature-based visual place recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3741–3748, 2018.

[32] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 577–586.

[33] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li, "Scalable k-nn graph construction for visual descriptors," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1106–1113.

[34] B. Harwood and T. Drummond, "Fanng: Fast approximate nearest neighbour graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5713–5722.

[35] M. G. Gollub, R. Dubé, H. Sommer, I. Gilitschenski, and R. Siegwart, "A partitioned approach for efficient graph–based place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst./Workshop Planning, Perception Navigat. Intell. Veh.*, 2017.

[36] E. Garcia-Fidalgo and A. Ortiz, "Hierarchical place recognition for topological mapping," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1061–1074, 2017.

[37] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[38] J. Brandt, "Transform coding for fast approximate nearest neighbor search in high dimensions," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1815–1822.

[39] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European conference on computer vision*. Springer, 2008, pp. 304–317.

[40] H. Sandhawalia and H. Jégou, "Searching with expectations," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 1242–1245.

[41] Y. Kalantidis and Y. Avrithis, "Locally optimized product quantization for approximate nearest neighbor search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2321–2328.

[42] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*. IEEE, 2006, pp. 459–468.

[43] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2009, pp. 1753–1760.

[44] L. Han and L. Fang, "Mild: Multi-index hashing for appearance based loop closure detection," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 139–144.

[45] A. Babenko and V. Lempitsky, "The inverted multi-index," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 6, pp. 1247–1260, 2014.

[46] T. Cieslewski and D. Scaramuzza, "Efficient decentralized visual place recognition using a distributed inverted index," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 640–647, 2017.

[47] O. Vysotska and C. Stachniss, "Lazy data association for image sequences matching under substantial appearance changes," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 213–220, 2016.

[48] E. Pepperell, P. Corke, and M. Milford, "Routed roads: Probabilistic vision-based place recognition for changing conditions, split streets and varied viewpoints," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1057–1179, 2016.

[49] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset." *IJ Robotics Res.*, vol. 36, no. 1, pp. 3–15, 2017.

[50] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1808–1817.

[51] S. Garg, N. Sünderhauf, and M. Milford, "Semantic-geometric visual place recognition: A new perspective for reconciling opposing views," *The International Journal of Robotics Research*, 2019.

[52] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International journal of computer vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[53] A. Levey and M. Lindenbaum, "Sequential karhunen-loeve basis extraction and its application to images," *IEEE Transactions on Image processing*, vol. 9, no. 8, pp. 1371–1374, 2000.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[55] M. Douze, A. Sablayrolles, and H. Jégou, "Link and code: Fast indexing with graphs and compact regression codes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3646–3654.

[56] C.-Y. Chiu, A. Prayoonwong, and Y.-C. Liao, "Learning to index for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[57] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *arXiv preprint arXiv:1702.08734*, 2017.

[58] N. Corporation. (2019). [Online]. Available: https://developer.nvidia.com/embedded/develop/hardware