Explore-Exploit Graph Traversal for Image Retrieval

Cheng Chang [*]
Layer6 AI
jason@layer6.ai

Guangwei Yu [*]
Layer6 AI
guang@layer6.ai

Chundi Liu
Layer6 AI
chundi@layer6.ai

Maksims Volkovs
Layer6 AI
maks@layer6.ai

Abstract

We propose a novel graph-based approach for image retrieval. Given a nearest neighbor graph produced by the global descriptor model, we traverse it by alternating between exploit and explore steps. The exploit step maximally utilizes the immediate neighborhood of each vertex, while the explore step traverses vertices that are farther away in the descriptor space. By combining these two steps we can better capture the underlying image manifold, and successfully retrieve relevant images that are visually dissimilar to the query. Our traversal algorithm is conceptually simple, has few tunable parameters and can be implemented with basic data structures. This enables fast real-time inference for previously unseen queries with minimal memory overhead. Despite relative simplicity, we show highly competitive results on multiple public benchmarks, including the largest image retrieval dataset that is currently publicly available. Full code for this work is available here: https://github.com/layer6ai-labs/egt.

1. Introduction

Image retrieval is a fundamental problem in computer vision with numerous applications including content-based image search [31], medical image analysis [16] and 3D scene reconstruction [12]. Given a database of images, the goal is to retrieve all relevant images for a given query image. Relevance is task specific and typically corresponds to images containing same attribute(s) such as person, landmark or scene. At scale, retrieval is typically done in two phases: first phase quickly retrieves an initial set of candidates, and second phase refines this set returning the final result. To support efficient retrieval, first phase commonly encodes images into compact low dimensional descriptor space where retrieval is done via inner product. Numerous approaches have been proposed in this area predominantly based on local invariant features [17, 18, 29] and bag-of-words (BoW) models [26]. With recent advances in deep learning, many of the leading descriptor models now use convolutional neural networks (CNNs) trained end-to-end for retrieval [30, 3, 10, 25].

Second phase is introduced because it is difficult to accurately encode all relevant information into compact descriptors. Natural images are highly complex and retrieval has to be invariant to many factors such as occlusion, lighting, view-angle, background clutter etc. Consequently, while the first phase is designed to be efficient and highly scalable, it often doesn't produce the desired level of accuracy [35, 7]. Research in the second phase have thus focused on reducing false positives and improving recall [7, 15]. A common approach to reduce false positives is to apply spatial verification to retrieved query-candidate pairs [21]. The localized spatial structure of the image is leveraged by extracting multiple features from various regions typically at different resolutions [20]. Spatial verification based on RANSAC [9] is then applied to align points of interest and estimate inlier counts. Filtering images by applying threshold to their inlier counts can significantly reduce false positives, and various versions of this approach are used in leading retrieval frameworks [21, 6].

To improve recall, graph-based methods are typically applied to a k-nearest neighbor (k-NN) graph produced by the first stage [35]. Query expansion (QE) [7] is a popular graph-based approach where query descriptor is iteratively refined with descriptors from retrieved images. QE is straightforward to implement and often leads to significant performance boost. However, iterative neighbor expansion mostly explores narrow regions where image descriptors are very similar [15]. An alternative approach using similarity propagation/diffusion has received significant attention recently due to its strong performance [15, 25, 4]. In diffusion, pairwise image similarities are propagated through the k-NN graph, allowing relevant images beyond the immediate neighborhood of the query to be retrieved thus improving recall [8]. While effective, for large graphs similarity propagation can be prohibitively expensive making real-time retrieval challenging in these models [13]. More efficient alternatives have recently been proposed [14], how-

[*] Authors contributed equally to this work.

ever efficiency is achieved as a trade-off with performance.

In this work we propose a novel image retrieval approach based on the traversal of the nearest neighbor graph. Our approach preserves the connectivity of the k-NN graph and only traverses it by alternating between explore and exploit steps. This leads to highly efficient inference, enabling real-time retrieval on previously unseen images with minimal memory overhead. Empirically, we show that a combination of explore and exploit steps maximally utilizes the immediate neighbors of the query, and effectively explores more distant vertices leading to significant improvements in both precision and recall. In summary, our contributions are as follows:

- We propose a novel image retrieval approach based on the k-NN graph traversal. By alternating between explore and exploit steps we are able to effectively retrieve relevant images that are "far" from the query in the descriptor space.
- We introduce an effective approach to incorporate spatial verification into the k-NN graph through edge reweighting. Incorporating spatial verification reduces topic drift during traversal further improving accuracy.
- The proposed approach naturally generalizes to online inference and we propose a simple yet effective procedure for retrieval with previously unseen images. This procedure is efficient and scales well to large retrieval tasks.
- We conduct extensive empirical evaluation on publicly available benchmarks demonstrating highly competitive performance with new state-of-the-art results on multiple benchmarks.

2. Related Work

In this section we review relevant graph-based image retrieval methods that operate on the k-NN graph produced by the global descriptors. This direction is typically motivated by the hypothesis that descriptor inner product similarity cannot properly capture the highly complex image manifold structure [27]. Early approaches in this area include manifold ranking [35] and query expansion (QE) [7]. In QE, the query descriptor is updated with nearest neighbors and the updated query is re-issued. This approach tends to produce consistent improvements and can be applied to a wide range of image descriptors [2]. Many variants of QE have been proposed including transitive closure [6] and more recently αQE [25]. Spatial verification [21] is often applied in conjunction with QE to reduce topic drift where updated query descriptor deviates significantly from the original [7]. A major drawback of QE is that it can only explore limited regions where image descriptors are very similar [15]. Furthermore, each iteration requires full or partial query reissue which can get prohibitively expensive for large databases [7, 2].

Similarity propagation methods, also known as diffusion, are another popular category of graph-based approaches [8], recently achieving state-of-the-art performance on a number of benchmarks [15, 25, 4]. Extensive study has been conducted on various ways to propagate similarity through the k-NN graph, most of which can be viewed as versions of random walk [8]. Related work hypothesize that relevant objects can be closer in one similarity space while not in another, and explore fusion methods in conjunction with similarity propagation [34, 32, 5]. Despite strong performance, most existing similarity propagation methods are computationally expensive. This makes application to modern large scale image databases difficult, particularly in the online setting where new queries have to be handled in real-time. Spectral methods have been proposed to reduce computational cost [13], but the speedup is achieved at the cost of increased memory overhead and drop in performance.

In this work we propose a novel approach to refine and augment descriptor retrieval by traversing the k-NN graph. Our traversal algorithm enables efficient retrieval, and new queries can be handled with minimal overhead. Moreover, once retrieval is completed, the new query can be fully integrated in the graph and itself be retrieved for other queries with equal efficiency. In the following sections we outline our approach in detail and present empirical results.

3. Proposed Approach

We consider the problem of image retrieval where, given a database of n images $\mathcal{X} := \{x_1, ..., x_n\}$ and a query image u, the goal is to retrieve the top-k most relevant images for u. Images are considered to be relevant if they share a pre-defined criteria, such as containing the same scene, landmark, or person. In many applications n can be extremely large reaching millions or even billions of images. As such, the initial retrieval is typically done using compact descriptors where each image is represented as a vector in a d-dimensional space and similarity is calculated with an inner product. With recent advancements in deep learning, many state-of-the-art descriptor models use convolutional neural networks (CNNs) that are trained end-to-end for retrieval [10, 1, 25]. However given the complexity of natural images, even with powerful CNN models it is difficult to encode all relevant information into compact descriptors. It has been shown that applying additional processing to retrieved images can significantly improve accuracy, and this two-stage approach is adopted by many leading retrieval models [8, 5, 25]. In this work we propose a novel approach based on graph traversal to refine and augment the retrieved set. Specifically, we show that by traversing the k-NN graph formed by the descriptors, alternating between exploration and exploitation steps, we can effectively retrieve relevant images that are "far" away from the query in the descriptor space. We refer to our approach as the Explore-Exploit Graph Traversal (EGT).

k-NN Graph Retrieving the top-k images for every image in $\mathcal{X}$ produces a sparse k-NN graph G_k. Formally, the weighted undirected k-NN graph G_k contains vertices $\{x|x \in \mathcal{X}\}$ and edges described by the adjacency matrix $A_k = (a_{ij}) \in \mathbb{R}^{n \times n}$. The edges are weighted according to the similarity function s_k and the adjacency matrix is defined by:

$$a_{ij} = \begin{cases} s_k(x_i, x_j) & \text{if } x_j \in \text{NN}_k(x_i) \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

where $\text{NN}_k(x)$ is the set of k nearest neighbors of x in the descriptor space; $a_{ij} = 0$ indicates that there is no edge between x_i and x_j. G_k is highly sparse given that typically $k \ll n$, and contains nk edges at most. The sparsity constraint significantly reduces noise [33, 8] making traversal more robust as noisy edges are likely to cause divergence from the query. Since global descriptors trade-off accuracy for efficiency, the immediate neighbors NN_k might not contain all relevant images unless k is very large. To improve recall it is thus necessary to explore regions beyond NN_k, which motivates our approach.

Explore-Exploit Graph Traversal Given G_k as input, our goal is to effectively explore relevant vertices beyond NN_k. However, traversing far from the query can degrade performance due to topic drift [27]. Incorrect vertices chosen early on can lead to highly skewed results as we move farther from the query. A balance of exploration and exploitation is thus required where we simultaneously retrieve the most likely images in the neighborhood of the query and explore farther vertices. Moreover, to avoid topic drift, farther vertices should only be explored when there is sufficient evidence to do so. These ideas form the basis of our approach. We alternate between retrieving images with shortest path to the query and exploring farther vertices. Further improvement is achieved by adopting a robust similarity function s_k.

To control the trade-off between explore and exploit we introduce a threshold t such that only images with edge weights greater than t can be retrieved. Then starting at the query image, we alternate between retrieving all images that pass t (*exploit*) and traversing neighbors of retrieved images (*explore*). During the traversal, if the same not-yet-retrieved image is encountered again via a new edge, we check if the new edge passes the threshold t and retrieve the image if it does. The intuition here is if the edge passes the threshold then the image must be sufficiently similar to an already retrieved image and should also be retrieved. This procedure creates "trusted" paths between the query and far away vertices via edges from already retrieved vertices. Threshold t controls the degree of exploration. Setting $t = 0$ reduces to a greedy breadth first search without exploration, and setting $t = \infty$ leads to Prim's algorithm [23] with aggressive exploration.

Algorithm 1: EGT

```
input : $k$-NN graph $G_k = (\mathcal{X}, A_k, s_k)$,
        query $u$,
        number of images to retrieve $p$,
        edge weight threshold $t$
output: list of retrieved images $Q$
initialize max-heap $H$, list $V$, and list $Q$
add $u$ to $V$
do
    // Explore step
    foreach $v \in V$ do
        foreach $x \in NN_k(v), x \notin Q, x \neq u$ do
            if $x \in H$ and $H[x] < s_k(v, x)$ then
                update weight for $x$: $H[x] \leftarrow s_k(v, x)$
            else if $x \notin H$ then
                push $x$ to $H$ with weight $s_k(v, x)$
            end
        end
    end
    clear $V$
    // Exploit step
    do
        $v \leftarrow pop(H)$
        add $v$ to $V$ and $Q$
    while ($peek(H) > t$ or $|V| = 0$) and $|Q| < p$
while $|Q| < p$ and $|H| > 0$
return $Q$
```

Edge Re-weighting In the original graph G_k returned by the descriptor model, edge weights correspond to inner product between descriptors. However, as previously discussed, these weights are not optimal as global descriptors have limited expressive power. To make traversal more robust, we propose to refine G_k by keeping the edge structure and modifying the scoring function s_k effectively re-weighting each edge. RANSAC [9] and other inlier-based methods are widely used in state-of-the-art retrieval methods as a post-processing step to reduce false positives [21]. We adopt a similar approach here and propose to use the RANSAC inlier count for s_k. Analogous to previous work [6], we found RANSAC to be more robust than descriptor scores, allowing to explore far away

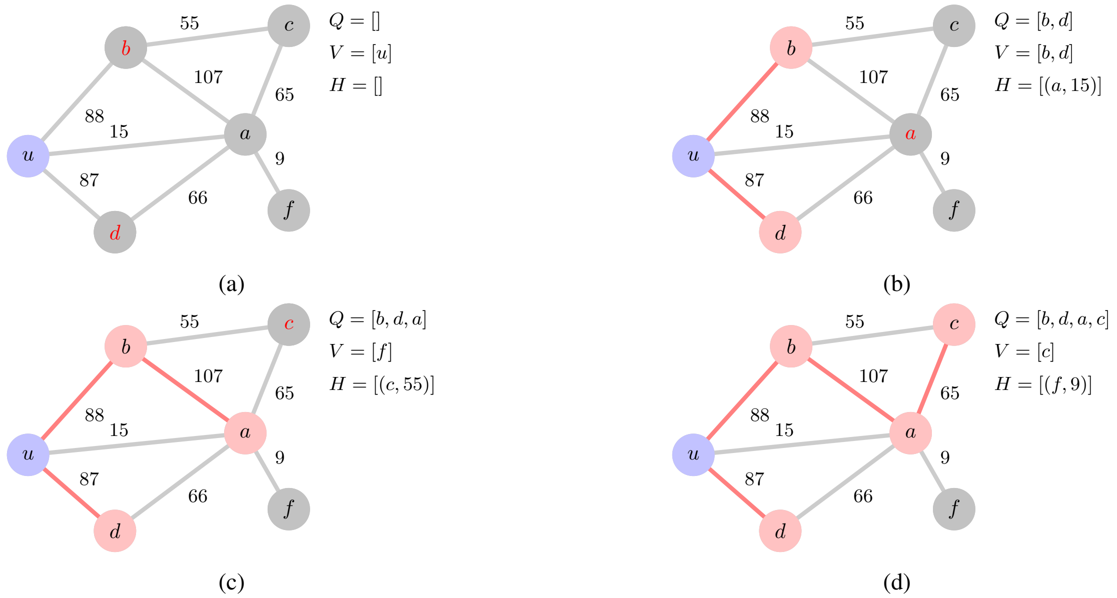


Figure 1: Algorithm 1 example with query image u, database images $\mathcal{X} = \{a, b, c, d, f\}$, $t = 60$, and $p = 4$. States at the beginning of each iteration of the outer loop (line 3) in Algorithm 1 are shown on the right for each figure. Red vertices denote retrieved images and weights on the edges are the inlier counts. Red vertex label indicates that this vertex will be explored at the next iteration. (a) Traversal is initiated by adding query u to V. (b) During the first iteration, vertices $\{a, b, d\} \in \text{NN}_k(u)$ are pushed to H. Both $s_k(u, b) > t$ and $s_k(u, d) > t$, so they are popped from H, added to V, and retrieved to Q. (c) During the second iteration, neighbors of b and d are added to H. At this point, weight a is replaced with the largest visited edge so $H[a] = s_k(b, a) = 107$. Since a's updated weight puts it at the top of the max-heap, it is popped from H next, added to V and retrieved to Q. (d) During the third and final iteration, neighbors of a are added to H. Only c, f are not in Q_u, so f is added to H and c's weight is updated to 65. Finally, c is popped from H and added to Q terminating the algorithm. Note that the order of images in Q directly corresponds to the order in which they were popped from H.

vertices with minimal topic drift [7]. RANSAC calculation is done once offline for all nk edges and the graph remains fixed after that. However, even in the offline case computing RANSAC for all edges in G_k is an expensive operation so we make this step optional. Empirically we show that without RANSAC our approach still achieves leading results among comparable models, while adding RANSAC further improves performance producing new state-of-the-art.

Algorithm 1 formalizes the details of our approach. We use max-heap H to keep track of the vertices to be retrieved, list V to store vertices to be explored and list Q to store already retrieved vertices. The graph traversal is initialized by adding query image u to V. Then at each iteration we alternate between explore and exploit steps. During the explore step we iterate through all images $v \in V$ and add images in their neighborhood $\text{NN}_k(v)$ to the max-heap H. Each image $x \in \text{NN}_k(v)$ is added to H with the weight $s_k(v, x)$, which corresponds to the confidence that x should be retrieved. In cases where x is already in H but with a lower weight, we update its weight to $s_k(v, x)$ so max-heap always stores the *highest* edge weight with which x was visited. Similarly to query expansion, we treat already retrieved images as ground truth and use the highest available similarity to any retrieved image as evidence for x. Finally, once all images in V are explored we clear the list.

During the exploit step, we pop all images from H whose weights pass the threshold t, add them to V to be explored, and retrieve them to Q. The "retrieve" operation always appends images to Q and no further re-ordering is done. This ensures that the visit order is preserved in the final returned list. Conceptually, images retrieved earlier have higher confidence since they are "closer" to the query so preserving the order is desirable here. In cases where no image in H passes the threshold, we pop a single image with the current highest weight so the algorithm is guaranteed to terminate. A detailed example of this procedure is shown in Figure 1.

Online Inference In our approach, G_k is constructed entirely off-line and is not modified during retrieval. For the off-line inference where query image is already in $\mathcal{X}$, retrieval involves a quick graph traversal following Algorithm 1. However, in many applications off-line inference is not sufficient and the retrieval system must be able to handle new images in real-time. In the online inference given a query image $u \notin \mathcal{X}$, we need to retrieve images from $\mathcal{X}$

Method	mAP							
	$\mathcal{R}$Oxford		$\mathcal{R}$Oxford+$\mathcal{R}$1M		$\mathcal{R}$Paris		$\mathcal{R}$Paris+$\mathcal{R}$1M	
	Medium	Hard	Medium	Hard	Medium	Hard	Medium	Hard
Without SV								
R-MAC [11]	60.9	32.4	39.3	12.5	78.9	59.4	54.8	28.0
R-MAC+αQE [25]	64.8	36.8	45.7	19.5	82.7	65.7	61.0	35.0
R-MAC+DFS [15]	69.0	44.7	56.6	28.4	89.5	80.0	83.2	70.4
R-MAC+Hybrid-Spectral-Temporal [14]	67.0	44.2	55.6	27.2	89.3	80.2	82.9	69.2
R-MAC+EGT	73.6	56.3	55.8	35.1	90.6	81.2	79.4	63.7
With SV								
HesAff+rSIFT+HQE [29]+SV	71.3	49.7	52.0	29.8	70.2	45.1	46.8	21.8
DELF [20]+HQE +SV	73.4	50.3	60.6	37.9	84.0	69.3	65.2	35.8
HesAffNet+HardNet++ [19]+HQE+SV	75.2	53.3	-	-	73.1	48.9	-	-
R-MAC+DFS+DELF+ASMK [28]+SV	75.0	48.3	68.7	39.4	90.5	81.2	86.6	74.2
R-MAC+DFS+HesAff+rSIFT+ASMK+SV	80.2	54.8	**74.9**	47.5	92.5	84.0	**87.5**	**76.0**
R-MAC+QE+SV+rEGT	**83.5**	**65.8**	**74.9**	**54.1**	**92.8**	**84.6**	87.1	75.6

Table 1: mAP results on medium and hard versions of the $\mathcal{R}$Oxford and $\mathcal{R}$Paris with and without the 1M distractor set $\mathcal{R}$1M. To make comparison fair we split the table into two parts to separate models with and without spatial verification (SV).

that are relevant to u. This is straightforwardly achieved in our approach. First, using the same descriptor model, we compute k nearest neighbors of u to obtain $\text{NN}_k(u)$. We then add u to G_k by expanding the adjacency matrix with $\text{NN}_k(u)$. Finally, we (optionally) refine the edge weights by computing the RANSAC inlier counts for all images in $\text{NN}_k(u)$. Algorithm 1 can now be applied without modification to obtain relevant images for u. Note that after this process u is fully integrated into G_k and can itself be retrieved for another query.

Complexity We analyze the run-time complexity of our algorithm and compare it with leading graph-based approaches. Following previous work we assume that G_k is already constructed and omit this cost from our analysis. To analyze the complexity, we note that at every iteration at least one vertex is popped from H so that retrieving p images involves at most p iterations. For each vertex popped from H, we traverse its immediate neighbors during the explore step so the total pushes to H are upper-bounded by pk. The outer loop of the Algorithm 1 thus has worst case complexity of $\mathcal{O}(pk\log(pk))$ during online retrieval which is dominated by the max-heap. There is no additional offline cost beyond the computation of G_k. If inlier edge re-weighting is used, the offline complexity increases to $\mathcal{O}(nk)$ since we need to compute RANSAC for the k edges of each vertex. For online inference, k additional RANSAC evaluations are required per query so the complexity remains $\mathcal{O}(pk\log(pk))$.

Recently proposed leading similarity propagation approach by Iscen et al., [15] has online inference complexity of $\mathcal{O}(pk\sqrt{\rho})$ where ρ is the condition number of the diffusion transition matrix. An improvement on this runtime can be achieved by shifting and caching some computation to offline [13]. This leads to offline complexity of $\mathcal{O}(nr(k+r))$ and online complexity of $\mathcal{O}(pr)$ where r is the spectral rank. To achieve good performance the authors suggest to use large r in the range of $5,000$ to $10,000$, which increases runtime complexity for both offline and online procedures as they depend on r^2 and r respectively.

4. Experiments

We present results on three recent and publicly available landmark retrieval benchmarks: revisited Oxford ($\mathcal{R}$Oxford), revisited Paris ($\mathcal{R}$Paris) [24], and Google Landmark Retrieval Challenge dataset [20]. $\mathcal{R}$Oxford and $\mathcal{R}$Paris build on the well-known Oxford [21] and Paris [22] datasets in image retrieval by refining the labels. Notably, many hard examples (severe occlusion, deformation, change in viewpoint etc.) that were not used before are now included in the evaluation [24]. A significantly larger and more challenging distractor set of $1,001,001$ images ($\mathcal{R}$1M) is introduced, replacing the original 100K distractors. Furthermore, the relevant ground truth annotations are subdivided into the easy, medium, and hard subsets. In this paper, we focus on the more difficult medium and hard subsets. The revisited datasets substantially increase the level of difficulty as evidenced by considerably lower re-evaluated model performance [24]. In total, $\mathcal{R}$Oxford contains $4,993$ images and $\mathcal{R}$Paris contains $6,322$ images.

The Google Landmark Retrieval Challenge dataset consists of $1,093,647$ database images and $116,025$ query images. At the time of writing this is the largest publicly available image retrieval dataset. This dataset was at the core of the image retrieval challenge organized by Google [1] to

[1] www.kaggle.com/c/landmark-retrieval-challenge

Rank	Team	mAP@100
1	CVSSP & Visual Atoms	**0.627**
2	Layer 6 AI	0.608
3	SevenSpace	0.598
4	Naver Labs Europe	0.586
5	VPP	0.583
	R-MAC+QE+SV+rEGT	0.619

Table 2: Leaderboard mAP@100 results for the top-5 teams on the Google Landmark Retrieval Challenge. We compare our approach by submitting the predictions to the challenge evaluation server. Over 200 teams participated in this challenge.

benchmark retrieval models at scale in a standardized setting. Over two hundred teams participated in the challenge using a wide array of approaches and we compare our results against the top-5 teams.

Implementation Details Global descriptors for all experiments are obtained using a CNN-based R-MAC descriptor model with ResNet-101 backbone fine-tuned for landmark retrieval [10]. We do not re-train or further fine-tune the model released by the original authors [2]. Standard multi-scale averaging is applied as in [24] to obtain 2048-dimensional descriptors for all images. While re-training can improve performance, our aim is to test the generalization ability of our approach to previously unseen images that can be from an entirely different geographical location. Generalization to previously unseen objects is critically important in image retrieval as most systems have limited labelled training data.

For EGT, the k-NN graph G_k is constructed by computing inner product nearest neighbor retrieval in the 2048-dimensional descriptor space. To validate the effect of inlier edge re-weighting, we also apply RANSAC scoring to G_k, and refer to this variant as rEGT. To compute RANSAC we use the deep local features (DELF) model [20] [3], and follow the default feature extraction pipeline. At most 1000 feature vectors are extracted per image, and the dimension of each vector is reduced to 40 via PCA. Verification is then performed as in [21] to obtain inlier count, which is used to replace inner product as the edge weight. We test our traversal algorithm on both original and re-weighted G_k. We also validate effect of QE and spatial verification (SV) on top of our method. These are implemented in the standard manner outlined by [7] and [29], where verified images from initial results are used to retrieve a new set of results. Through cross validation we set $t = 0.42$ and $t = 50$ for EGT and rEGT respectively and $k = 100$ for all datasets. All experiments are conducted on the 20-core Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz machine with 200GB of RAM.

Method	p	$\mathcal{R}$Oxford+$\mathcal{R}$1M		$\mathcal{R}$Paris+$\mathcal{R}$1M	
		mAP	Time (ms)	mAP	Time (ms)
R-MAC		12.5	194±55	29.8	184 ± 12
+ αQE		19.5	344±23	37.1	463 ± 70
+ DFS	1K	15.3	259±26	42.9	241 ± 12
+ DFS	5K	19.8	279±17	51.8	284 ± 17
+ DFS	10K	21.0	303±25	55.3	305 ± 12
+ DFS	20K	25.9	343±17	58.5	361 ± 15
+ EGT	1K	32.9	198 ± 55	62.4	189 ± 13
+ EGT	5K	**33.1**	205 ± 55	**62.5**	196 ± 13
+ EGT	10K	**33.1**	216 ± 55	**62.5**	210 ± 14
+ EGT	20K	**33.1**	239 ± 56	**62.5**	235 ± 16

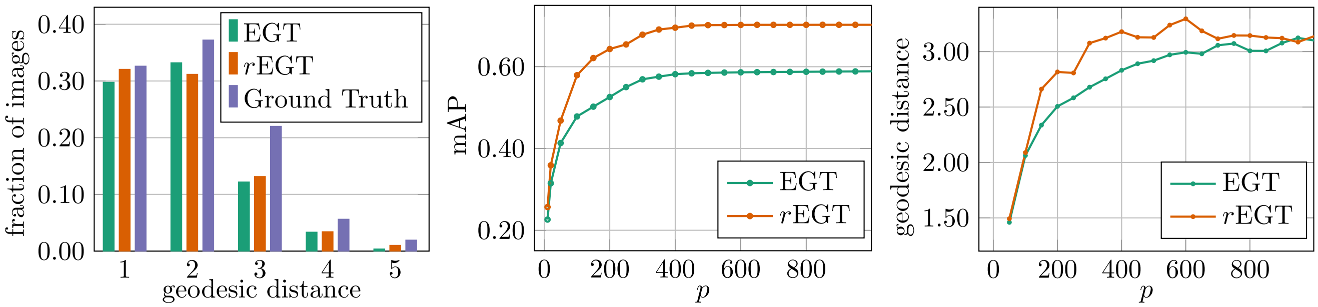
Table 3: Run-time and mAP results on the hard versions of the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets with 1M distractors. Time records online query time in ms and we repeat the experiment 100 times to estimate the standard deviation. All methods use the same global descriptors from R-MAC [11] to do nearest neighbor search, and k is set to 50 to make comparison fair. EGT (without RANSAC) is benchmarked against αQE [25] and DFS [15].

Results Results on $\mathcal{R}$Oxford and $\mathcal{R}$Paris are shown in Table 1. We compare our approach to the state-of-the-art baselines taken from the recent survey by Radenovic et al [24]. The baselines are partitioned into two groups without and with spatial verification shown at the top and bottom half of the table respectively. For both $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets we show results on medium and hard versions with and without the 1M distractor set $\mathcal{R}$1M resulting in eight datasets in total. rEGT with inlier re-weighting achieves new state-of-the-art results on six of the eight datasets and performs comparably to the best model on the other two. Notably the performance is particularly strong on the $\mathcal{R}$Oxford hard category where it outperforms the best baselines by over 20%. We also see that EGT without inlier re-weighting has highly competitive performance beating or performing comparably to the best baselines. The one exception is the $\mathcal{R}$Paris+ $\mathcal{R}$1M dataset where DFS [15] is the best performing model.

The Google Landmark Retrieval challenge results are shown in Table 2. We compare our approach against the top-5 teams by submitting our predictions to the challenge evaluation server. Unlike $\mathcal{R}$Oxford and $\mathcal{R}$Paris which are highly geographically localized, this dataset contains photos of landmarks from all over the world. The photos are from the public, and thus include a lot of variations such as significant occlusion, camera artifacts, viewpoint change and zoom, and lighting changes. From the challenge

[2] code: http://www.europe.naverlabs.com/Research/Computer-Vision/Learning-Visual-Representations/Deep-Image-Retrieval

[3] code: https://github.com/tensorflow/models/tree/master/research/delf



(a) (b) (c)

Figure 2: All figures are for the $\mathcal{R}$Oxford hard dataset. (a) Geodesic distance (shortest path to the query in G_k) distribution of correctly retrieved images from the top-500 list by EGT and rEGT compared to the distribution of all relevant images. (b) mAP@p as p is varied from 1 to 1,000. (c) Averaged geodesic distance for images at different positions in the retrieved list.

Method	$\mathcal{R}$Oxford		$\mathcal{R}$Paris	
	Med.	Hard	Med.	Hard
R-MAC	60.9	32.4	78.9	59.4
R-MAC+QE+SV	68.6	46.2	81.3	64.8
R-MAC+EGT	73.6	56.3	90.6	81.2
R-MAC+QE+SV+EGT	82.0	61.7	91.3	81.2
R-MAC+QE+SV+rEGT	**83.5**	**65.8**	**92.8**	**84.6**

Table 4: Ablation results on medium and hard versions of $\mathcal{R}$Oxford and $\mathcal{R}$Paris.

workshop [4], many of top teams employed model ensembles fine tuned specifically for this dataset to obtain high-quality global descriptors. In contrast, we did not fine tune the R-MAC descriptors and did not ensemble, our submission used the same retrieval pipeline as in the $\mathcal{R}$Oxford and $\mathcal{R}$Paris experiments. From the table we see that our approach is highly competitive placing 2'nd out of over 200 teams. We narrowly miss the top spot with less than a point difference between us and the top team.

Together these results demonstrate that EGT is a simple yet effective retrieval method that can be combined with other models to achieve leading performance. Our approach is particularly strong in challenging settings common in real-world applications, where images are noisy and have significant variations.

Analysis To obtain additional insight into run-time performance in a realistic setting where $p \ll n$, we vary p from 1K to 20K and measure query time. Table 3 shows mAP and average query time for EGT and one of the leading baselines DFS [15]. From the table we see that EGT becomes progressively faster than DFS as p increases. This can be attributed to the simplicity of Algorithm 1 that can be efficiently implemented with common data structures. For small p the overhead of EGT is negligible relative to the base k-NN retrieval with R-MAC. We also see that EGT consistently attains higher mAP accuracy at lower p than DFS. Little improvement in mAP is achieved beyond p = 1K, and with that setting EGT outperforms DFS with much larger p = 20K. This has a direct and practical impact on run-time since p can be set to a small value without sacrificing accuracy in EGT.

We evaluate the importance of each component of the proposed method by conducting an ablation study shown in Table 4. From the table it is seen that both EGT and rEGT improve performance significantly when added to every component combination with or without query expansion. As expected rEGT consistently outperforms EGT across all combinations but at the cost of increased run-time. We also see that EGT performance is further enhanced by combining it with spatially verified QE due to more robust neighbor estimation in G_k.

To test the ability of EGT to retrieve relevant images that are farther away from the query, we analyze the geodesic distance (shortest path to the query) during retrieval. Figure 2a shows geodesic distance distribution for correctly retrieved relevant images together with the ground truth distribution for all relevant images. From the ground truth distribution it is evident that only around 30% of relevant images are in the immediate neighborhood of the query. This further supports the conclusion that descriptor retrieval alone is not sufficient. The explore step enables EGT to traverse farther, and effectively retrieve relevant images with geodesic distance up to 5. Surprisingly, distributions for EGT and rEGT look similar despite significant difference in mAP as shown in Table 4. This indicates that the performance difference between the two methods is primarily due to changes in ranking order rather than recall.

We analyze this further by evaluating performance at different positions in the retrieved list. Figure 2b shows mAP@p as p is varied form 1 to 1,000, and Figure 2c shows average geodesic distance to the query for all retrieved images at each rank position. We see that at the very top

[4] https://landmarksworkshop.github.io/CVPRW2018

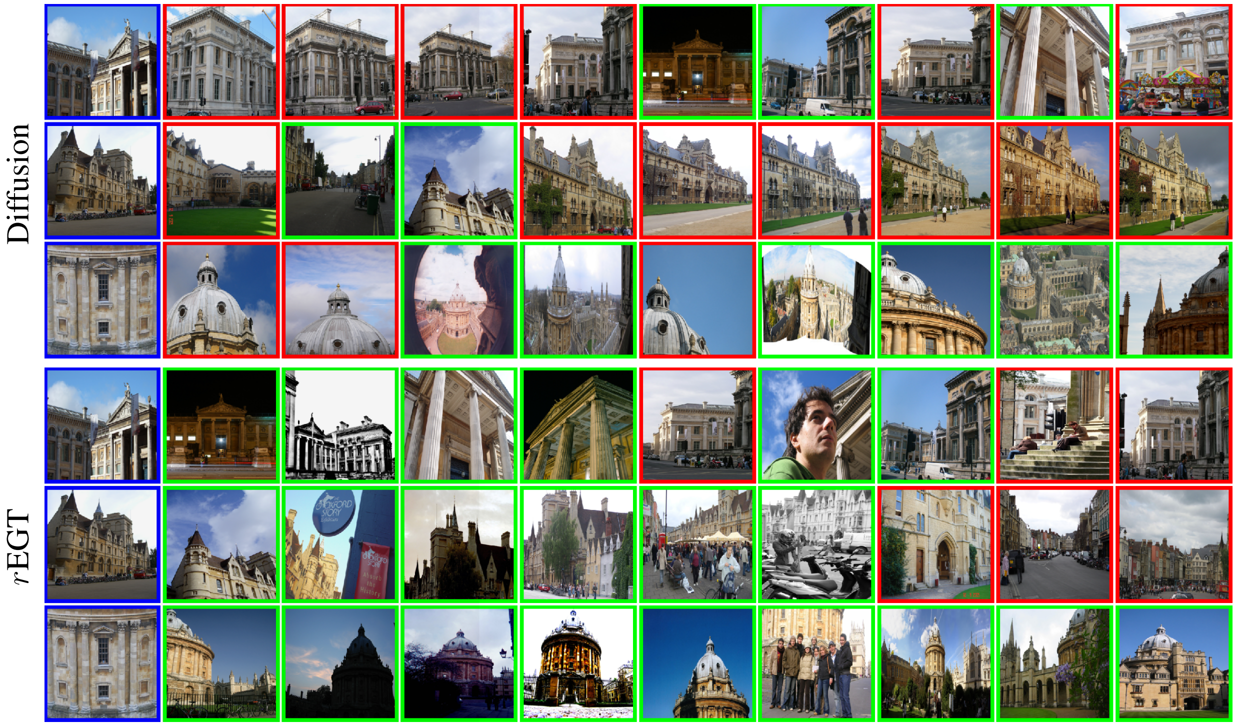


Figure 3: Top retrieved results for three queries from the $\mathcal{R}$Oxford dataset using diffusion [24] and rEGT, both using R-MAC descriptors and SV. Query images are in blue, correctly retrieved images are in green and incorrect ones are in red.

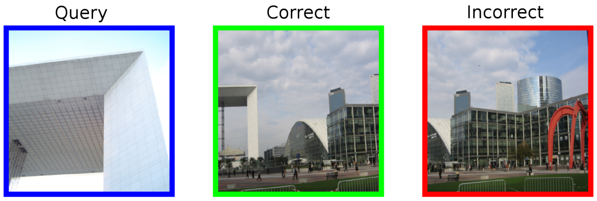


Figure 4: Topic drift example for $\mathcal{R}$Paris+$\mathcal{R}$1M.

of the retrieved list, mAP performance is similar between EGT and rEGT, but rEGT quickly overtakes EGT at higher ranks. Similarly, Figure 2c shows that rEGT starts to explore distant images earlier than EGT. Jointly these results indicate that re-weighting the edges reduces bias towards immediate neighbors, enabling the traversal to start exploring distant neighborhoods sooner.

Qualitative Results Selected qualitative examples of retrieval are shown in Figure 3. We show three queries from $\mathcal{R}$Oxford, along with the top nine retrieved results for diffusion [24] (top) and rEGT (bottom). Here, we see that diffusion tends to retrieve images with similar viewpoints and makes repeated mistakes. In contrast, images retrieved by our approach are much more diverse and include multiple view points and condition (zoom, lighting etc.) variations. The explore step is thus able to successfully capture relevant images that are visually dissimilar to the query.

We noted above that EGT performance on $\mathcal{R}$Paris is not as strong as on $\mathcal{R}$Oxford. After further inspection we noticed that $\mathcal{R}$Paris (particularly the 1M distractor set) contains more cluttered scenes that sometimes lead to topic drift during the explore step in graph traversal. An example of this is shown in Figure 4. Here, given a query image in blue, EGT first retrieves correct image in green. This image contains additional buildings which lead to topic drift, and an incorrect image with those buildings is retrieved next. Cluttered scenes increase the likelihood of topic drift, and large distractor set is likely to contain more images with similar structures. We believe that weaker performance of EGT on $\mathcal{R}$Paris+$\mathcal{R}$1M can be partially attributed to the combination of these factors.

5. Conclusion

In this work, we proposed a new approach for image retrieval based on graph traversal. We alternate between explore and exploit steps to better capture the underlying manifold, and retrieve relevant but visually dissimilar images that global descriptors fail to retrieve. Empirically, we demonstrated that the proposed approach is efficient and outperforms state-of-the-art on multiple recent and large-scale benchmarks. Given the promising results, future work involves exploring other graph-based traversal methods and comparing their efficacy.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 2

[3] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *ICCV*, 2015. 1

[4] S. Bai, X. Bai, Q. Tian, and L. J. Latecki. Regularized diffusion process on bidirectional context for object retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1, 2

[5] S. Bai, Z. Zhou, J. Wang, X. Bai, L. J. Latecki, and Q. Tian. Ensemble diffusion for retrieval. In *CVPR*, 2017. 2

[6] O. Chum et al. Large-scale discovery of spatially related images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 1, 2, 3

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007. 1, 2, 4, 6

[8] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *CVPR*, 2013. 1, 2, 3

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 3

[10] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016. 1, 2, 6

[11] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. End-to-End Learning of Deep Visual Representations for Image Retrieval. *International Journal of Computer Vision*, 2017. 5, 6

[12] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. Reconstructing the world* in six days *(as captured by the Yahoo 100 Million Image Dataset). In *CVPR*, 2015. 1

[13] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum. Fast spectral ranking for similarity search. In *CVPR*, 2018. 1, 2, 5

[14] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum. Hybrid diffusion: Spectral-temporal graph filtering for manifold ranking. *arXiv preprint arXiv:1807.08692*, 2018. 1, 5

[15] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In *CVPR*, 2017. 1, 2, 5, 6, 7

[16] J. Kalpathy-Cramer, A. G. S. de Herrera, D. Demner-Fushman, S. Antani, S. Bedrick, and H. Müller. Evaluating performance of biomedical image retrieval systems-an overview of the medical image retrieval task at ImageCLEF 2004–2013. *Computerized Medical Imaging and Graphics*, 2015. 1

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004. 1

[18] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 2004. 1

[19] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is not enough: Learning affine regions via discriminability. In *ECCV*, 2018. 5

[20] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-Scale Image Retrieval with Attentive Deep Local Features. In *ICCV*, 2017. 1, 5, 6

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*. IEEE, 2007. 1, 2, 3, 5, 6

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 5

[23] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 1957. 3

[24] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: large-scale image retrieval benchmarking. In *CVPR*, 2018. 5, 6, 8

[25] F. Radenovic, G. Tolias, and O. Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1, 2, 5, 6

[26] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1

[27] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000. 2, 3

[28] G. Tolias, Y. Avrithis, and H. Jégou. Image search with selective match kernels: aggregation across single and multiple images. *International Journal of Computer Vision*, 2016. 5

[29] G. Tolias and H. Jégou. Visual query expansion with or without geometry: Refining local descriptors by feature aggregation. *Pattern Recognition*, 2014. 1, 5, 6

[30] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 1

[31] T. Weyand and B. Leibe. Discovering favorite views of popular places with iconoid shift. In *ICCV*, 2011. 1

[32] F. Yang, B. Matei, and L. S. Davis. Re-ranking by multi-feature fusion with diffusion for image retrieval. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015. 2

[33] X. Yang, S. Köknar-Tezel, and L. J. Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR Workshops*, 2009. 3

[34] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *ECCV*. 2012. 2

[35] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS*, 2003. 1, 2