# Feature Tracking for Wide-Baseline Image Retrieval

Ameesh Makadia

Google Research
76 Ninth Ave
New York, NY 10014
makadia@google.com

**Abstract.** We address the problem of large scale image retrieval in a wide-baseline setting, where for any query image all the matching database images will come from very different viewpoints. In such settings traditional bag-of-visual-words approaches are not equipped to handle the significant feature descriptor transformations that occur under large camera motions. In this paper we present a novel approach that includes an offline step of feature matching which allows us to observe how local descriptors transform under large camera motions. These observations are encoded in a graph in the quantized feature space. This graph can be used directly within a soft-assignment feature quantization scheme for image retrieval.

**Key words:** Wide baseline, image retrieval, quantization

## 1 Introduction

In this paper we address the challenge of image retrieval from large databases. While this is a classic problem in Computer Vision, we are interested in the specific scenario of *wide-baseline* image retrieval. In this setting, we assume that for most query images the closest matching (true matches) databases images are of the same scene but from very different viewpoints, and thus have undergone significant transformations relative to the query (see Figure 5 for an example).

We isolate the wide-baseline challenge because it has important practical implications. For example, it is unrealistic in any real-world image retrieval system that the databases will contain many images of all interesting locations, so matching to a few images of a scene is important. Furthermore, the ability to effectively match images from a wide-baseline means one can construct the database accordingly, keeping fewer images of each scene than would otherwise be needed. This would have an impact on both storage costs and retrieval time.

Much of the work for large scale image retrieval has been based on the bag-of-visual-words (BOW) approach [1, 2], which borrows ideas from the text-retrieval community. To summarize briefly, forming an image representation can be broken into three steps: (1) local feature detection and extraction (2) feature quantization, and (3) tf-idf image representation. In our setting, the real challenges lie

within the first two steps. Much effort has been put into identifying local feature detectors and descriptors [3–5] suitable for correspondence problems, but even these will not remain sufficiently invariant (either repeatability of detector or invariance of descriptor) when the camera undergoes very large motions. Also, while feature quantization may mask small descriptor transformations, it is unlikely to deal gracefully with larger transformations.

In this paper we aim to improve wide-baseline retrieval within the BOW framework. To address the primary issue of feature deformations over large camera motions, we perform unsupervised tracking of millions of points through long image sequences in order to observe how corresponding descriptors transform under significant camera motions. As an intermediate representation for millions of feature tracks, we construct a weighted graph embedded in the quantized feature space, where the edge weight between two words is related to the number of tracks having descriptors mapped to both words. We will refer to this graph as the *track-graph*. In a way the track-graph encodes how often we have seen features that are mapped to one word transform into features that are mapped to another word. Importantly, this graph construction provides a purely data-driven way of encoding the observed feature transformations. We avoid the difficult problem of explicitly modeling or parameterizing the space of feature transformations, for example. We utilize the track-graph for the image retrieval application by incorporating it into a soft-assignment scheme similar to that of [6].

Our primary contribution can be summarized as a novel approach for image retrieval that utilizes offline feature tracking to observe feature transformations under large camera motions. To our knowledge this is the first method that successfully incorporates such information within a BOW retrieval approach. Furthermore, we examine properties of the track-graph in detail to understand the added value of the information it encodes. Evaluation of the retrieval system in a wide-baseline setting shows promising performance.

### 1.1 Related work

Retrieval from large image collections is a well-studied problem in Computer Vision. In 2003, Sivic and Zisserman [1] applied a text retrieval approach for object retrieval, and many of the current state-of-the-art methods can be seen as an extension of this BOW approach (a few examples are [2, 7–9]).

At their core, these methods rely on a quantization of the feature space into visual words to make large scale retrieval a tractable problem. To this end a number of papers have explored various ideas related to partitioning and assignment in the feature space. The baseline standard is $k$-means clustering to build a vocabulary, and nearest-neighbor assignment of descriptors to words [1]. In [2], a vocabulary tree (constructed with hierarchical $k$-means) is used for feature quantization and assignment, while [8] considers approximate $k$-means for assignment, and [10] considers a fixed quantization for a vocabulary. More recently, [7] have studied the effects of quantization and introduce a combination of vector quantization and hamming embedding. In [11] kernel density estimation

is applied to capture the uncertainty of assignment from features to words, thus limiting the effects of hard assignment. Philbin et al [6] show the effects of quantization can be remedied in part by a soft-assignment when mapping descriptors to words. Our work is influenced by this last approach as we will incorporate our observations of feature transformations into a similar soft-assignment scheme. While the works above address the issue of feature quantization and assignment, there has been little done to specifically address the challenges of image retrieval under wide-baseline conditions. Efforts in landmark recognition are capable of building rich 3D representations of landmarks or scenes given a sufficient number of images [12–15]. These representations can be utilized for image matching and retrieval. However, such approaches [13] still require the image database to be populated with a large number of images of each landmark. This is in contrast to our wide-baseline setting where we do not expect to have a large number of matching images in the database for any query.

A central premise of our approach is that by tracking or matching features through image or video sequences one can observe how image features transform under large viewpoint changes. Utilizing this information should improve retrieval when query images come from unique viewpoints relative to the database images. Prior related work includes [16], where Implicit Shape Models are constructed for object pose recognition. 3D descriptors are represented by a set of 2D image descriptors that were matched over a set of training images, however manual correspondences (e.g. hand-selected feature matches) are required for initialization. In [17], object level representations are formed by object regions tracked within video shots, which provides viewpoint invariance for retrieval. In [18], feature matching between images in the training set is used to identify a set of "useful" features. This greatly reduces the number of features stored from each image without loss in retrieval precision. Earlier work [19] explores building invariant distance measures with a priori knowledge of patch transformations. In [20] invariance is incorporated into SVM classifiers by introducing artificially transformed copies of support vectors. In [21], feature matching under wide-baseline conditions is treated as a classification problem, where each class corresponds to all views of a point. In the BOW approach of [6], one variation of their soft assignment scheme involves generating multiple feature sets for an image by applying small transformations directly to the image. While our work is related to these last approaches, we note that in both [21] and [6] feature transformations are generated by *simulating* image transformations. This is limited in that one has to model and parameterize the space of patch transformations (e.g. affine transformations), and it is unlikely such an approach can capture the full spectrum of transformations that are observed from actual camera motions. In our work we address this issue by tracking features through image sequences.

## 2 Bag-of-visual-words image retrieval

In this section we give a brief summary of our implementation of the traditional baseline BOW model. We combine a Hessian-Affine interest point detector and

SIFT descriptors [4, 5], and starting with a collection of SIFT features from a large dataset a visual vocabulary is constructed by partitioning the feature space with $k$-means clustering. The ideal number of clusters depends on the dataset, and so in this paper we experiment with vocabularies ranging from 2500 to 500000 in size. We denote a vocabulary as $V = \{v_1, v_2, v_3, \dots\}$, where the visual words $v_i$ are the cluster centers. Feature-to-word mapping assigns a descriptor $x_i$ to the closest visual word $\hat{x_i} = \underset{v}{\text{argmin}} \|x_i - v\|_2$. In practice, this mapping is done with approximate nearest neighbors when the vocabulary size is large. The final tf-idf image representation is simply a weighted histogram over the words appearing in an image. For complete details see [1, 22].

## 3 Constructing the *track-graph*

In this section we describe our process of constructing a graph in the quantized space that encodes feature transformations observed from image sequences. Our intermediate goal here is to develop a method that allows us to characterize how feature descriptors transform under large camera motions. As with any data-driven approach, we must be sure to collect a large number of samples in order to be sure our observations have statistical significance. Our setup consists of 3 cameras on top of a vehicle that acquires images while driving through urban city streets. The three cameras ($c_1$, $c_2$, and $c_3$) are arranged to face the same side of the street, but at different angles. Camera $c_2$ is fronto-parallel to building facades on the side of the street, while $c_1$ and $c_3$ are offset approximately 45° on either side of $c_2$. Due to this arrangement, scene content that is observed by camera $c_1$ at time $t$ is usually observed by $c_2$ and $c_3$ at a later time $t' > t$. Figure 1 shows a sequence of images from 7 consecutive time steps.

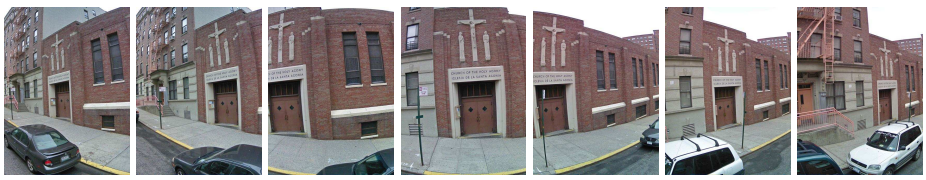


**Fig. 1.** A sequence of seven consecutive frames from our acquisition system that illustrates how the same scene content will be seen from multiple cameras over time separated by a wide baseline. At each of the seven time steps (frames 1116 through 1122), the same scene content is observed from one of the three cameras (1, 2, or 3). With frame-to-frame feature matching we are more likely to generate tracks that observe world points from very different viewpoints angles than if we tried to match features directly between the extreme wide baseline image pairs.

### 3.1 Feature matching and track extraction

Since the sampling rate of the camera is too low to allow true feature tracking between frames, we pursue discrete matching with RANSAC [23] to generate correspondences that are consistent with a Fundamental matrix or Homography transformation. Putative correspondences are obtained with nearest-neighbor matching while discarding ambiguous candidates [5] (however all nearest neighbor-matches are used to generate the final set of inliers after geometry estimation). Since image acquisition is of minor cost, our matching thresholds are fairly strict to protect against too many outliers. In total, we collected five non-overlapping image sequences from the same city that contained a total of $45K$ image frames. Our matcher extracted $3.8M$ feature tracks having an average duration of 5.8 frames (tracks shorter than 3 frames are discarded).[1] Employing discrete matching in place of pure tracking also serves a secondary purpose. It is well known that repeatability of a feature detector is limited under viewpoint changes, and so by generating tracks in this way we make sure to observe feature transformations only for those features where the reliability of the detector is certain.

### 3.2 Graph construction

We define the *track-graph* as a weighted graph $G = (V, E, w)$, where the vertices are the set of visual words in the vocabulary. We would like the weight $w(u, v)$ to reflect the number of feature tracks whose descriptors have mapped to both $u$ and $v$. Let us represent a tracked feature $t$ with its observed SIFT descriptors $t = \{x_1, x_2, x_3, \dots\}$, and let $T$ be the collection of all feature tracks obtained during the offline tracking process ($T = \{t_i\}$). Figure 2 shows the few steps required in constructing the weighted graph. To summarize, the edge weights between vertices $w(u, v)$ count exactly the number of tracks where at least one descriptor mapped to word $u$ and at least one descriptor mapped to word $v$. Note that our construction process ignores self-edges ($w(u, u) = 0, \forall u \in V$).[2]

### 3.3 Properties of the track-graph

The graph construction process described above can be seen as an iterative process, where tracks are incorporated into the track-graph one after another (in arbitrary order). The natural question that arises is how do we determine the stability of the graph as we continue to incorporate more tracks (this can help us determine if we have included enough observations to terminate construction). To study this characteristic we evaluate how the graph changes as we continue

[1] Figures depicting some tracking results can be seen at `http://www.cis.upenn.edu/~makadia/`

[2] We also considered post-processing the graph to account for the frequency in which certain types of features were tracked. One such normalization was, for example, $w'(u, v) = \frac{w(u,v)^2}{\sum_y w(u,y) \sum_y w(v,y)}$. In practice, the few we tried all lowered performance.

**Input**

1. Set of tracked features $T$, and a set of visual words $V$, and a weighted graph over the words $G = G(V, E, w)$ (initially $w(u, v) = 0, \forall u, v \in V$).

**Constructing** $G$

1. For each track $t \in T$:
   (a) $\hat{t} = unique(\{\hat{x_1}, \hat{x_2}, \hat{x_3}, \dots\})$.
   (b) For each pair of words $(u, v)$ in $\hat{t}$:
      i. $w(u, v) = w(u, v) + 1$
      ii. $w(v, u) = w(v, u) + 1$
2. Filter graph by setting all small weights ($w(u, v) < \tau$) to zero. In practice we use $\tau = 5$.

**Fig. 2.** Outline of track-graph construction. Here a track $t$ is represented by the observed feature descriptors $t = \{x_1, x_2, \dots\}$, and the notation $\hat{x_i}$ refers to the visual word assignment for feature $x_i$.

to add more tracks. Given a graph $G_1$, let us define the probability of seeing an edge $(u, v)$ as $P((u, v)) = \frac{w(u,v)}{\sum_{u,v \in V} w(u,v)}$. Note for this task we ignore that $w(u, v)$ and $w(v, u)$ represent the same link. Given two graphs $G_1$ and $G_2$, the KL-divergence of $P_{G_2}$ from $P_{G_1}$ is used to measure their relative difference:

$$D_{KL}(P_{G_2} \| P_{G_1}) = \sum_{(u,v) \in V \times V} P_{G_2}((u, v)) \log \frac{P_{G_2}((u, v))}{P_{G_1}((u, v))} \tag{1}$$

For our purposes here $G_2$ will always represent a graph obtained by integrating more tracks into $G_1$. The relative graph distance is not complete unless we account for the relative "sizes" of the graphs. In other words, the relative change in graphs should be normalized by the number of observations used to construct the graphs. If we define the size of the graph as $W_G = \sum_{u,v \in V} w(u, v)$, we can define the relative change between graphs as the KL-divergence scaled by the relative change in graph size: $D(P_{G_2} \| P_{G_1}) = D_{KL}(P_{G_2} \| P_{G_1}) \frac{W_{G_1}}{W_{G_2}}$. Table 1 shows that the graph changes much less as more and more tracks are incorporated (in this example the vocabulary size is 250000 words). This experiment was performed on graphs before the small edges were filtered out (as per Figure 2), which means the stability is observed even with possibly noisy edges present. The second important consideration is whether or not the constructed track-graph contains information that cannot be obtained from standard distance measures in the feature space. If after constructing a graph from millions of tracked features we find that the nearest neighbors in the graph (according to the edge weights) mimic the neighbors produced with a traditional distance measure, this would indicate that the track-graph will not contribute any orthogonal information. To examine this property, we construct another graph that captures the proximity between visual words in the feature space using a standard distance measure

| $W_{G_1}$ | $12M$ | $20M$ | $29M$ | $36M$ |
|---|---|---|---|---|
| $W_{G_2}$ | $20M$ | $29M$ | $36M$ | 41M |
| $\frac{W_{G_1}}{W_{G_2}}$ | 0.58 | 0.70 | 0.79 | 0.88 |
| $D_{KL}(P_{G_2} \| P_{G_1})$ | 0.36 | 0.20 | 0.13 | 0.07 |
| $D(P_{G_2} \| P_{G_1})$ | **0.21** | **0.14** | **0.10** | **0.06** |

**Table 1.** Each column shows how the track-graph changes as a new collection of tracks is incorporated. See the text for term definitions. The last row indicates that as we incorporate more and more tracks into the graph, the relative change in the graph continues to decrease. Note, in each column $G_2$ represents a graph obtained after incorporating more tracks into $G_1$.

for SIFT features. We call this the $L2$-graph since the edge weight $w(u, v)$ is related to the Euclidean distance between $u$ and $v$. To see how the track-graph and $L2$-graph relate, we compare for each visual word its closest neighbors in the track graph against its closest neighbors in the $L2$-graph. Figure 3 (left) illustrates the average overlap between a visual word's 10 closest neighbors in the two graphs. Even for small vocabulary sizes there is less than 50% overlap between the neighborhoods. A related experiment shown in Figure 3 (middle, right) examines the actual Euclidean distance to a word's $k$-th nearest neighbor in the track and $L2$ graphs, respectively. The differences between the two graphs is an indication that the track graph is capturing feature transformations that may not occur smoothly in the feature space.

A final experiment on the graphs is a simple test of feature assignment. The idea is to see how useful the track-graph weights might be in practice where corresponding features are often initially mapped to different words. Since our graph weights are constructed to reflect this property exactly, in some sense this test can be considered a cross-validation step. We collect feature tracks from an image sequence that was not used during track-graph construction. From these tracks we select 5000 wide-baseline feature pairs. We consider a feature pair $(x_i, x_j)$ to be wide-baseline if $x_i$ was observed in camera $c_1$ and $x_j$ in $c_3$ (or vice-versa). Our measure of correct assignment is if $\hat{x_j}$ is one of the $k$-nearest neighbors of $\hat{x_i}$ in the track/$L2$ graph. Figure 4 shows the results of this experiment for different vocabulary sizes and for $k$ ranging from 0 to 10 ($k = 0$ is just traditional hard assignment, and is thus the same for both the track and $L2$ graphs).

The experiments above depict valuable qualities of the track-graph. First, the graph is relatively stable after incorporating $3.8M$ tracks (Table 1), which gives us confidence we have included enough tracks during construction. Second, the graph encodes some information about how feature descriptors transform that cannot be observed with a traditional distance measure (Figure 3). Finally, initial experiments show the graph may be useful in practical correspondence tasks (Figure 4).

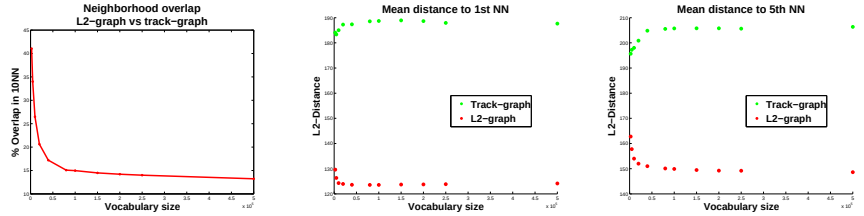


**Fig. 3.** The single plot on the left compares the overlap between a visual word's 10 nearest neighbors in the track-graph and its neighbors in the $L2$-graph. We ignore those words that did not have any tracked neighbors. The plot shows this neighborhood overlap ratio for graphs constructed with 11 different vocabularies (ranging from 2500 to 500000 words). The two plots to the right compare the average distance of a visual word to its $k$-th nearest neighbor ($k = 1$ on the left, $k = 5$ on the right).

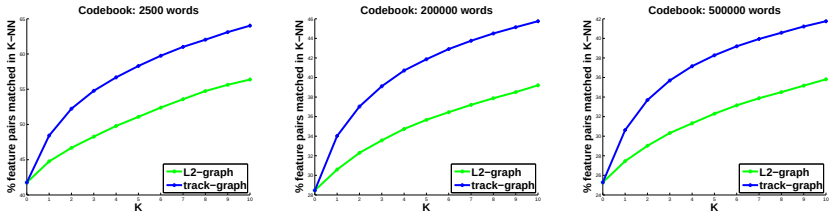


**Fig. 4.** Feature assignment results for the track and $L2$ graphs. Results are shown for graphs constructed with three different vocabulary sizes: 2500 (left), 200000 (middle), and 500000 (right). For each feature pair $(x_i, x_j)$, assignment is considered correct if $\hat{x_j}$ is one of the $k$ nearest neighbors of $\hat{x_i}$ in the graph. In the plots $k$ ranges between 0 and 10.

### 3.4 Image retrieval with the track-graph

To utilize the track-graph in an image-retrieval engine, we develop a natural feature quantization scheme that is motivated by the soft assignment approach of Philbin et al [6]. To summarize [6] briefly, instead of quantizing feature $x$ to its closest word $\hat{x}$, the vote for $x$ in the tf-idf vector is distributed over the $k$-nearest words to $x$. The weights given to each of these words is proportional to $\exp \frac{-d^2}{2\sigma^2}$, where $d = \|x - \hat{x}\|_2$.

We utilize our track-graph in a similar way. Instead of assigning $x$ to its closest word $\hat{x}$ (determined by $L2$ distance), the vote for $x$ will be distributed between $\hat{x}$ and the closest words to $\hat{x}$ in the track graph. For $k$-nearest neighbor assignment, for example, the weight for $x$ will go to $\hat{x}$ and the $(k-1)$-nearest neighbors of $\hat{x}$ in the track-graph (neighbors are determined by sorting the edge weights $w(\hat{x}, v)$ in decreasing order). Here also the tf-idf weights are proportional to $\exp \frac{-d^2}{2\sigma^2}$. The weights for each feature are scaled uniformly so the total tf-idf contribution is 1, and $\sigma$ is set identical to [6]. Note, we are only using the graph weights $w(\hat{x}, v)$ for selecting a word's neighbors, while the tf-idf weights are determined by $L2$ distances. The fundamental difference between our ap-

proach and [6] is that the track-graph provides a unique way of selecting the "closest" words to a feature. The track-graph neighbors will be consistent with the feature transformations observed in our offline tracking process rather than just $L2$ proximity in the feature space. For visual words that have fewer than $k-1$ edges, the neighborhood is supplemented with the original feature's closest ($L2$) words. For example, if a feature's closest word $\hat{x}$ has no track-graph neighbors, its assignment reduces to the soft-assignment of [6]. For the track-graph constructed over 500000 visual words, 36% of the words had no edges. At the other extreme, all words had some neighbors for the graph constructed with the smallest vocabulary of 2500 words. In the next section we evaluate our proposed approach for wide-baseline image retrieval.

## 4 Evaluation

As we wish to evaluate image retrieval specifically in a wide-baseline setting, we prepared a test scenario that reflects the relevant challenges. We begin by collecting an image sequence in the same manner as described earlier. From this sequence, we select 1374 non-overlapping test images, and the remaining images form the database. All test images are chosen from either camera $c_1$ or $c_3$, so that they are not oriented fronto-parallel to the building facades. To create a sufficiently difficult challenge, only the wide-baseline matches are stored in the database (6348 images). Figure 5 shows two test images and their true neighbors in the database. Each test image has on average 4.6 true matches in the database. We supplement the dataset with 247686 images collected from image sequences that have no overlap with the test sequence. In total, we have 1374 test images and 254034 database images.

We note that there is no overlap between the images used for vocabulary and graph construction and the images used to build the evaluation dataset. However, all images come from urban environments using the same image acquisition scheme so the vocabulary and tracked features will still be relevant for the application.


**Fig. 5.** Challenging sample test images and their closest matching database images. The significant camera motion between the queries and their matches makes for a difficult retrieval task.

### 4.1 Evaluation criteria

Most image retrieval systems designed for practical large-scale use perform some form of post-processing or re-ranking of the top results from the initial retrieval (e.g. geometric verification, query expansion, see [8, 24]). In this setting the most important criteria is making sure as many correct results as possible appear in the portion of the list that will be post-processed. In light of this we focus our evaluation on the top returned images. Specifically, we will measure recall (at $n$), which measures what fraction of the true matches appear in the top $n$ results. We will evaluate our track-graph based approach against the soft assignment of [6], as well as the traditional BOW approach. [3]

### 4.2 Results

For our evaluation the two primary parameters are (1) the number of neighbors used in $k$-NN feature-to-word assignment and (2) the cutoff $n$ for which we evaluate mean recall-at-$n$. Figure 7 shows results for $n \in \{10, 20, 50, 100\}$, and $k \in \{3, 4, 5\}$. Of the 11 vocabulary sizes we have experimented with in previous sections (2500, 5000, 10000, 20000, 40000, 80000, 100000, 150000, 200000, 250000, and 500000), we select four of the larger vocabularies (100000, 150000, 250000, and 500000 words) for displaying results here (as expected, all three algorithms performed their best on these larger vocabularies). While both our track-graph approach and the soft assignment of [6] significantly outperform the traditional hard assignment, the results also show the track-graph consistently improving over [6], especially at the largest vocabularies. The improvements are most noticeable at $n = 50$, while performance is closer at $n = 100$. For visual examples, Figure 6 shows three queries where our approach succeeded in retrieving at least one correct result in the top ten while both algorithms we compare against fail to return any correct matches.

Looking at these results in a wider context, the final improvement in the application setting of our method over the approach of [6] may seem modest compared to the possible gains indicated by our earlier isolated experiments (see Figure 4). One explanation for this is that, as mentioned earlier, our feature mapping reverts to [6] for those words where we have no tracking observations. In the case of 500000 words we see that 36% of the words had no track-graph neighbors. Furthermore, the earlier experiments isolate the comparison of tracked neighbors and $L2$ neighbors, whereas the retrieval results in Figure 6 show the results of an entire image retrieval engine, where naturally the differences within a single component will be muted.

Regarding the cost to sparsity using our track-graph approach, we note that for the 500000 word vocabulary, using $3 - NN$ assignment, our approach generates 7% fewer nonzero entries in the tf-idf representation than the comparable [6] (while simple hard assignment produces 66% fewer nonzero entries). Another

[3] We attempted a variation of soft assignment based on simulating image patch transformations [6], but due to the many implementation parameters our best results underperformed the simple BOW baseline, thus those results are not included here.

question is how does our constructed track-graph perform on image retrieval from more general image collections? We emphasize that it is critical that the track-graph encode the types of transformations expected in the retrieval problem (in this paper we focus on wide-baseline camera motions exhibited by street-level imagery). As we discuss in more detail in the following section, extending our automatic tracking and graph construction process to address the types of transformations observed in general collections (e.g. Web datasets) is non-trivial and left to future work[4].

**Fig. 6.** Top images retrieved for three queries using our track-graph method (Vocabulary size 500000, 5-NN assignment). The leftmost image (highlighted in red) is the query. The retrieval images are shown to the right of the query (correct matches highlighted in green). For all three queries, both [6] and traditional hard assignment failed to retrieve any correct matches in the top 10.

## 5 Future work

We have designed a novel data-driven approach to study how image features transform under large camera motions and how such observations can be incorporated into a system targeting wide-baseline retrieval. While our results are promising, we consider this preliminary work and note a number of areas requiring future attention. Most notably is the generalization of our approach. While our current approach encodes wide-baseline (specifically planar) motions in the track-graph, going forward we would like to cover all possible transformations (e.g. descriptor transformations induced by general camera motions, lighting and environment changes, changes in camera modality, etc.). This extension is non-trivial because our approach requires simple data collection and

[4] However, as a validation of our intuition here we do provide an evaluation of our wide-baseline tracks applied out of context to a general Web dataset. The supplemental material at `http://www.cis.upenn.edu/~makadia/` shows performance on the Oxford Buildings dataset [8].

fully unsupervised tracking to generate a large number of observations. However, extending this approach will be challenging because controlling data collection to observe a wide set of transformations, as well as automatically generating sufficient ground-truth correspondences, is not a simple task. Another point for future work is addressing the artifacts of quantization that remain in our development. Our graph construction relies on hard assignment of tracked descriptors to visual words, and similarly during tf-idf construction for identifying the word from which track neighbors are selected. While our decisions here have been motivated by sparsity and computation, we plan for future work to explore a fully probabilistic soft assignment framework for graph construction as well as tf-idf generation. Another aspect of our work worth investigating further is the application of our ideas to different problems. For example, we believe the track-graph may be useful for improving correspondences in two views, and the offline feature tracking can be used to build better visual vocabularies.

**Acknowledgments.** We thank Matthew Burkhart and Alexander Toshev for helpful discussions, and the Google StreetView team for the data.

## References

1. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: International Conference on Computer Vision. (2003) 1470–1477
2. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2006) 2161–2168
3. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC. (2002) 383–393
4. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. Int. J. Comput. Vision **60** (2004) 63–86
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60** (2004) 91–110
6. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2008)
7. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. International Journal of Computer Vision **87** (2010) 316–336
8. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007)
9. Jégou, H., Harzallah, H., Schmid, C.: A contextual dissimilarity measure for accurate and efficient image search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007)
10. Tuytelaars, T., Schmid, C.: Vector quantizing feature space with a regular lattice. In: International Conference on Computer Vision. (2007)
11. Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Smeulders, A.W.: Kernel codebooks for scene categorization. In: ECCV '08: Proceedings of the 10th European Conference on Computer Vision. (2008) 696–709

12. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: International Conference on Computer Vision. (2009)
13. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: European Conference on Computer Vision. (2008) 427–440
14. Zheng, Y.T., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T.S., Neven, H.: Tour the world: building a web-scale landmark recognition engine. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2009)
15. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: SIGGRAPH. (2006) 835–846
16. Arie-Nachimson, M., Basri, R.: Constructing implicit 3D shape models for pose estimation. In: International Conference on Computer Vision. (2009)
17. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. Int. J. Comput. Vision **67** (2006) 189–210
18. Turcot, P., Lowe, D.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: ICCV Workshop on Emergent Issues in Large Amounts of Visual Data, Kyoto, Japan (2009)
19. Simard, P.Y., Cun, Y.A.L., Denker, J.S., Victorri, B.: Transformation invariance in pattern recognition - tangent distance and tangent propagation. In: Lecture Notes in Computer Science, Springer (1998) 239–274
20. Schölkopf, B., Burges, C., Vapnik, V.: Incorporating invariances in support vector learning machines. In: ICANN 96: Proceedings of the 1996 International Conference on Artificial Neural Networks, Springer-Verlag (1996) 47–52
21. Lepetit, V., Pilet, J., Fua, P.: Point matching as a classification problem for fast and robust object pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2004) 244–250
22. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA (1986)
23. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24** (1981) 381–395
24. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: International Conference on Computer Vision. (2007)

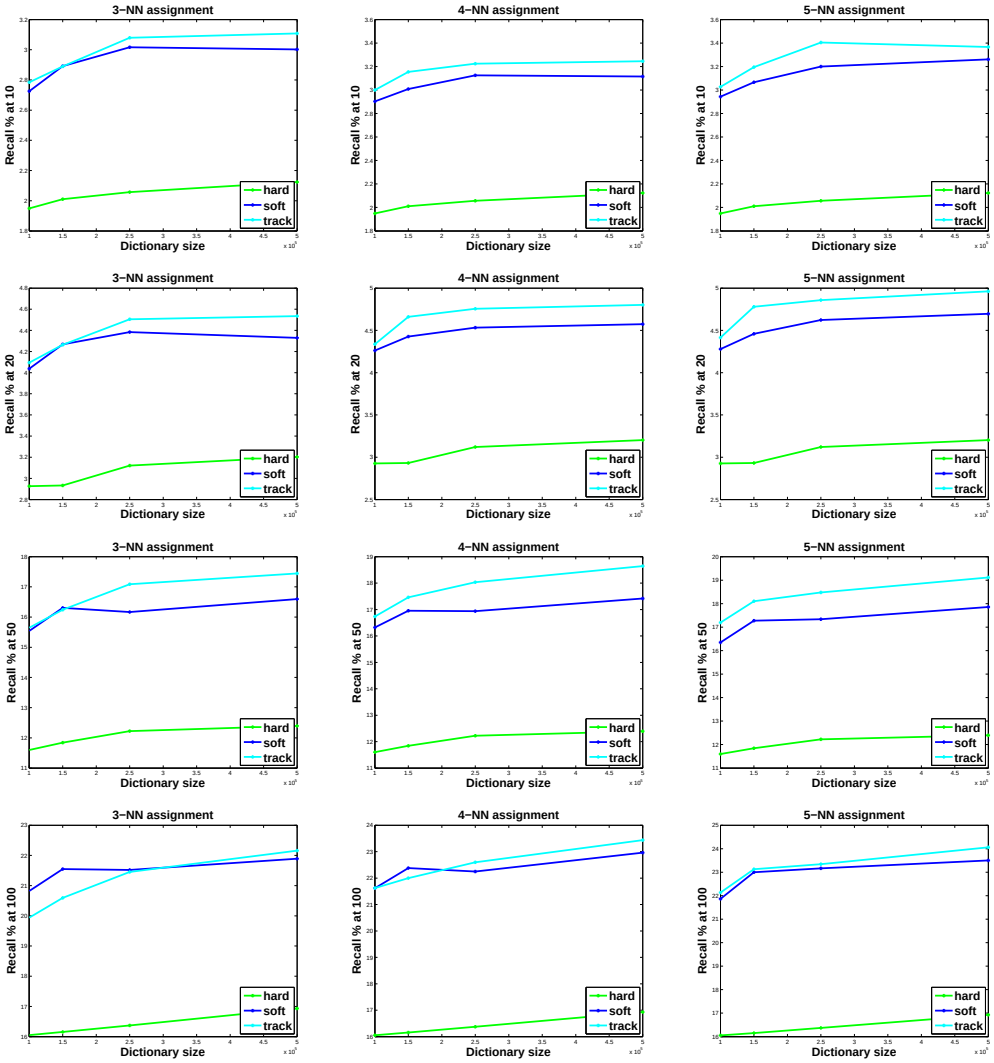


**Fig. 7.** Results of our track-graph approach ('track'), soft assignment [6] ('soft'), and traditional assignment ('hard'). Each plot shows a different combination of $n$ (recall-at-$n$) and $k$ ($k$-NN assignment that is used in both our approach as well as [6]). $n$ is one of 10, 20, 50, or 100. $k$ is one of 3, 4, or 5.