

# Minimizing the Number of Keypoint Matching Queries for Object Retrieval

Johannes Niedermayer  
niedermayer@dbis.fhnw.ch  
Peer Kröger  
kroeger@dbis.fhnw.ch

Institute for Informatics  
LMU Munich  
Munich, Germany

While the development of the SIFT-Descriptor made effective object retrieval on a large scale feasible, its initial use of nearest neighbor queries lead to slow runtimes even on relatively small data sets. These slow runtimes were first compensated by rough quantization using the Bag of Visual Words (BoVW) technique. In recent years, the focus turned back more and more to approximate  $k$ NN queries (e.g. [3]) due to their possible gain in matching accuracy [2]:  $k$ NN queries provide an accurate ranking of the match candidates and a measure of proximity between database features and query vectors. However, with the vast amount of features that have to be matched during recognition (up to a few thousand), even very fast  $k$ NN indexing techniques that can provide approximate query results in under ten milliseconds (e.g. [3]), would yield recognition runtimes of several seconds. We argue that the use of  $k$ NN queries for object recognition in large-scale systems cannot be achieved by developing efficient indexing techniques alone. The problem of efficiency has to be approached from additional research directions as well, such as the *number* of  $k$ NN queries posed on the system. In this paper, we evaluate an alternative recognition pipeline that ranks query features by assessing their matchability. In order to reduce the number of  $k$ NN queries, only the most promising features in this ranking are matched against the database. However, despite gaining efficiency, the enforced reduction of  $k$ NN queries causes a reduction of feature matches, decreasing the quality of the query result. While recall can be increased by increasing  $k$ , to increase Mean Average Precision (MAP) we expand matches on the image level, aiming at bypassing the indexing step in order to reduce runtimes. This approach stands in contrast to research in the area of BoVW-based retrieval: Research involving the BoVW pipeline assumes that the matching step is relatively cheap, especially if approximate cluster assignment techniques are used. In contrast, this paper aims at maximizing MAP for a small number of processed features, assuming that feature matching is expensive. To summarize, let  $n$  denote the upper bound on the number of matching queries, constraining the number of  $k$ NN queries. We aim at modifying the standard interest-point based image recognition pipeline such that a given performance measure (in our case MAP) is maximized for a fixed (and low)  $n$ . Therefore the problem setting is similar to BoVW-based approaches, however in such a context it is usually assumed that  $n = n_{max}$ . In this paper we address the opposite case where  $n \ll n_{max}$ .

The retrieval pipeline considered in this publication is shown in Figure 1. During feature extraction (Step 1), a set of keypoints and descriptors is extracted from the query. Then (Step 2), features are ranked based on their matchability. Feature ranking is based on the idea that some features in an image contain more information than other features. We aim at ordering the extracted features by a given quality measure and query only the features with the highest chance of providing good match hypotheses. Feature matching (Step 3) aims at finding match hypotheses for the features with highest expected matchability. For each of the first  $n$  features in the ranking, a  $k$ NN query is posed on the database, where  $k$  can be seen as a way to tweak *recall* at a given number of query features, as the number of images returned by the query is at most  $n \cdot k$ . To increase the efficiency of feature matching, we rely on fast state-of-the-art (approximate) indexing techniques optimized for high-dimensional data, for example Locally Optimized Product Quantization [3]. Then (Step 4), match expansion is performed on the resulting correspondences. In our scenario where we want to pose a small number of  $k$ NN queries on the system, we face the problem that even if we find some correspondences between the query and a database image, their number will be relatively low, increasing the probability that a good match is outranked by an image containing common random matches only. Expansion exploits the keypoint information of the seed matches that provide scale, rotation, and possibly affine information. These properties can be used to identify spatially close keypoints such as for example in [4]: Given that a match hypothesis is correct, not only the corresponding feature pair should match, but also its spatial neighborhood. The similarity of a match's neighborhood is evaluated using the

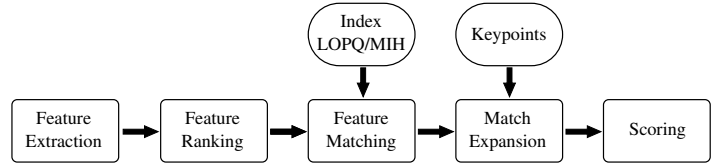


Figure 1: Recognition pipeline.

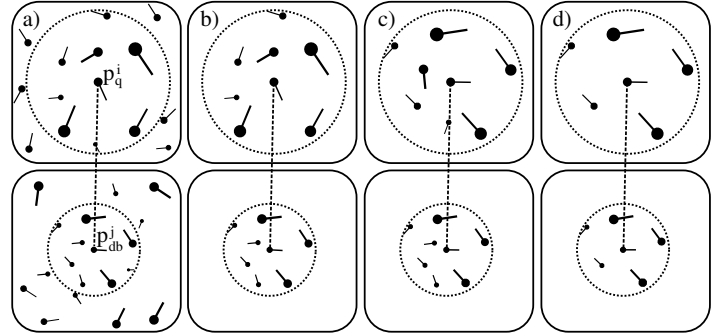


Figure 2: Generation of additional match hypotheses.

procedure shown in Figure 2. The figure shows an initial seed match, i.e. a  $k$ NN of a query feature and keypoints surrounding the seed match. The size of each keypoint is represented by the icon diameter, and the gradient direction is represented by a line anchored in the icon's center. The top row of this figure visualizes the features of the query image  $I_q$ , while the bottom row visualizes the image features of a tentative match image  $I_{db}$ . Starting point is an initial correspondence pair  $(p_q^i, p_{db}^j)$  established by  $k$ NN-search in feature space, see Figure 2 a). In a first step, features in a given spatial range are retrieved in the image  $I_q$  for  $p_q^i$  and in Image  $I_{db}$  for  $p_{db}^j$ , see Figure 2 b); the spatial range is visualized by a dotted circle. Spatially close keypoints with a significantly different scale than their reference feature are discarded (see the small features in the figure) similar to [1], resulting in two sets of features. These remaining features are rotation-normalized using the reference keypoint's gradient orientation information, rotating the set of keypoints and their corresponding gradient orientations, see Figure 2 c). Then the two lists of keypoints are traversed in parallel. If two features correspond sufficiently well in their appearance, orientation and position, the corresponding features are accepted as a matching pair (see Figure 2 d)). Based on the expanded list of matches, a score is computed for every database image (Step 5); we adapt the technique from [2], weighting scores based on their distance to the query feature and the number of features in the image. To increase the memory efficiency of the feature database, we compress the feature vectors used during match expansion using Product Quantization.

In our experimental evaluation we evaluate the effect of the number  $n$  of features queried, the effect of  $k$  in relation to the number of keypoints queried, and the pipeline's behaviour on different feature descriptors including real-valued and binary features.

- [1] Ondrej Chum, Michal Perdoch, and Jiri Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *Proc. CVPR*, pages 17–24, 2009.
- [2] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Exploiting descriptor distances for precise image search. Technical Report 7656, INRIA, 2011.
- [3] Yannis Kalantidis and Yannis Avrithis. Locally optimized product quantization for approximate nearest neighbor search. In *Proc. CVPR*, 2014.
- [4] Cordelia Schmid and Roger Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. CVPR*, pages 872–877, 1996.