An Evaluation of Two Automatic Landmark Building Discovery Algorithms for City Reconstruction

Tobias Weyand, Jan Hosang, and Bastian Leibe

UMIC Research Centre, RWTH Aachen University
{weyand,hosang,leibe}@umic.rwth-aachen.de
http://www.mmp.rwth-aachen.de

Abstract. An important part of large-scale city reconstruction systems is an image clustering algorithm that divides a set of images into groups that should cover only one building each. Those groups then serve as input for structure from motion systems. A variety of approaches for this mining step have been proposed recently, but there is a lack of comparative evaluations and realistic benchmarks. In this work, we want to fill this gap by comparing two state-of-the-art landmark mining algorithms: spectral clustering and min-hash. Furthermore, we introduce a new large-scale dataset for the evaluation of landmark mining algorithms consisting of 500k images from the inner city of Paris. We evaluate both algorithms on the well-known Oxford dataset and our Paris dataset and give a detailed comparison of the clustering quality and computation time of the algorithms.

1 Introduction

Recently, significant advances in large-scale city reconstruction have been made. Structure from motion (SfM) is used as a basic tool for reconstructing environments as point clouds [1–3], dense 3D representations [4, 5], or for photo browsing applications [6]. A prerequisite for SfM is a high-quality set of photos of the object to be reconstructed. A simple and cheap approach for obtaining such image sets is to collect them from community photo sharing sites However, this typically results in unordered photos of several different buildings with a significant fraction of unrelated photos. Therefore, there is a need for efficient image mining algorithms that group photos on a building or view level and remove photos that do not show buildings. Such photo clustering approaches are also a prerequisite for other interesting applications such as photo auto-annotation [7, 8], landmark recognition [2] or automatic landmark detection [9, 10].

Despite their importance, there is not yet a suitable benchmark for evaluating and comparing large-scale landmark mining algorithms. In this paper we take a first step in this direction by performing an evaluation of two state-of-the-art approaches: The first [11] is a top-down method that builds the complete pairwise matching graph of the image collection and segments it using *spectral clustering*. The second [12, 13] is a fast and approximate bottom-up approach that finds *cluster seeds* using the *(geometric) min-hash* image hashing algorithm. The seeds are then grown to clusters using query expansion [14].

In their original publications, both clustering approaches were evaluated on the Oxford buildings dataset which was originally created for evaluating image retrieval [15].

K.N. Kutulakos (Ed.): ECCV 2010 Workshops, Part II, LNCS 6554, pp. 310–323, 2012.

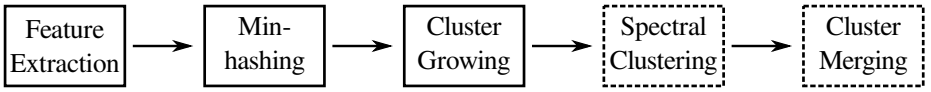
The dataset was constructed by collecting images of touristic sites by querying Flickr with the site labels. This results in a clear segmentation into groups that show a particular building, making the clustering task very simple. We use this dataset in our evaluation for consistency, but show that due to its structure the results are not very meaningful.

An important question not fully answered in the original publications [11, 13] is how the performance of these approaches translates to a more unconstrained setting, *i.e.* unstructured photos of an entire city. In this paper, we investigate this question by applying spectral clustering and geometric min-hash to a dataset of 500k geotagged images from the inner city of Paris. We furthermore present a ground truth for the evaluation of landmark mining systems on this dataset[1]. We closely examine the performance tradeoff between the two methods and propose a combination of them that can help eliminate the shortcomings of each approach. The tradeoff between computation time and clustering recall can then be adjusted using a single parameter.

Related Work. In the following, we describe the most closely related approaches from the literature in more detail. Agarwal *et al.* [1] present a large-scale SfM system with a highly distributed clustering pipeline. Effectively, the major landmarks of Rome are discovered and reconstructed from 150,000 images in 21 hours (using 495 compute nodes) of which 13 hours are spent in the image matching stage. For the clustering, a full $tf \cdot idf$ matching is performed and the top 10 matches for each image are verified using epipolar geometry. The resulting clusters are then merged and extended using query expansion to produce the largest possible connected components. Opposed to this, Strecha *et al.* [16] propose to reconstruct cities at a building level and to then join the partial reconstructions into a city-scale model using meta data. A prerequisite for this is a clustering on the building level. Gammeter *et al.* [7] build a system for automatic tagging of landmarks in touristic photos. Retrieval is performed by overlaying a square grid of 200×200 m cells over entire cities. By performing matching only within these cells, scalable and distributed preprocessing is possible. Meta information such as tags are used as a cue to cluster photos and to distinguish between photos of events and photos of landmark buildings. An object-driven pruning of the inverted index is performed in order to speed up the retrieval process. Finally, the discovered clusters are associated with Wikipedia articles, which serves as an additional verification. Zhang *et al.* [17] build a web-scale image-based landmark search engine by compiling a list of landmarks from geotagged photos and online travel guides and then collecting images of these landmarks from community photo collections and image search engines. In settings where meta information is not available, approaches based only on image information are necessary. Philbin *et al.* [11] present an exhaustive method for landmark detection. A full pairwise matching graph is constructed and segmented using spectral clustering. The approach is discussed in detail in Section 2. Because this approach requires a complete pairwise matching including spatial verification of the image collection it does not scale well to larger datasets. Chum *et al.* [13] present a faster but approximate approach using a randomized hashing scheme that allows for constant-time discovery of near duplicates in web-scale databases. The authors propose to use the hash collisions as "cluster seeds" from which to start a graph discovery using query expansion [14].

[1] Both the dataset and the ground truth are available from `http://www.mmp.rwth-aachen.de/data/paris-dataset`.

(a) Spectral clustering.



(b) Min-hash pipeline.

Fig. 1. The two different landmark discovery pipelines. The dashed lines denote the spectral clustering add-on that we propose for improving clustering precision.

2 Spectral Clustering on the Matching Graph

This section outlines the steps for clustering an image collection using spectral clustering, as proposed in [11]. Fig. 1a gives an overview of the pipeline.

Image Representation. A local feature representation of the images is built by first extracting scale-invariant Hessian affine regions [18] and SIFT descriptors [19]. The collected SIFT features are quantized into a codebook of 1M entries using k-means. To make this step computationally feasible, an approximate nearest neighbor (NN) search based on randomized kd-trees [15] is employed. The visual vocabulary is constructed on a smaller subset of the data with low-precision but a fast NN search. A higher precision NN search is used when matching features against the visual vocabulary.

Efficient Image Matching. The final image representation comprises the bag of visual words, feature positions, and the affine regions from the detector. Image retrieval is conducted in a similar fashion as text retrieval: query matches are retrieved using an inverted file structure that maps every visual word to all images it occurs in. The results are ranked using the cosine distance of their *"term frequency · inverse document frequency"* ($tf \cdot idf$) vectors. This results in a shortlist of k candidate matches, which are then geometrically verified by fitting a homography using SCRAMSAC [20]. The images for which the estimated homographies have sufficient support, are re-ranked above all other images [15] by adding 1.0 to the $tf \cdot idf$ score. This yields the final ranking score. The matching graph is constructed efficiently by querying the inverted file and inserting an edge for each match that exceeds a certain matching score threshold.

Spectral Clustering. In general, the connected components of the resulting matching graph correspond to a rough under-segmentation of the landmarks. Following [11] we first over-segment the connected components to get basic image clusters which can then be merged again with sufficient spatial verification. To this end, we use the spectral clustering algorithm of [21]. For each connected component, the optimal number of clusters is found by optimizing the Newman Q measure [22].

Re-merging Clusters. Philbin *et al.* [11] employ a heuristic to determine which spectral clusters show the same building and should therefore be merged: Each cluster is represented by its member image with the highest valence in the matching graph. The

image boundaries are then projected using the homographies of the edges along the shortest paths between the representative images. This yields the length of the shortest paths and the size of the overlapping area between representative images. Thresholds for the path length and overlap influence precision and recall as we will show in Sec. 5.2.

3 Min-hash Cluster Discovery

Min-hash [12, 13, 23] is a technique from text retrieval [24] used for efficiently discovering pairs of similar images in large image collections. The special property of min-hash is that the probability of an image pair being discovered increases with its similarity. This makes min-hash suitable as a near-duplicate image detector [25]. Chum *et al*. [12] demonstrate that the discovered image pairs can also serve as *seeds* for image clustering. Clusters are discovered in a growing step starting with the cluster seeds. This way, instead of building a complete matching graph, the min-hash approach reduces computational time by only exploring certain connected components. An overview of the pipeline is given in Fig. 1b. The stages in dashed boxes are an extension that we propose later in this section.

Hashing Images. A min-hash is a pseudo-random number generated from the visual words of an image. Let V be a visual word codebook. Given a random permutation of the numbers $\{1, \ldots, |V|\}$, the min-hash of an image is the first of the image's visual words occurring in the permutation. Typically, about 500-1000 random permutations are pre-generated and used for computing a set of min-hashes for each image. The probability of two images having the same min-hash equals the set overlap of their visual words [25]. To decrease the number of random collisions, several min-hashes are summarized into s-tuples called *sketches* ($s = 3, \ldots, 5$). An image pair is said to cause a *collision* if all min-hashes in a sketch are identical.

Detecting Collisions. To efficiently find min-hash collisions, hash-tables are created, storing for each min-hash the list of images with this hash. Then, sketch collisions are the intersections of the s sets of colliding images. This hashing procedure enables constant-time collision detection [12]. The price to pay for this efficiency is a very low recall, particularly for less similar, but still relevant, image matches.

Geometric Min-Hash. In geometric min-hash [13], sketches are created from features in a spatial neighborhood. This is done by selecting the first min-hash in a sketch randomly and then restricting the search for the remaining min-hashes to the affine region around the first feature. With this extension, a sketch collisions means that the colliding images not only have the colliding visual words in common, but also that the corresponding features come from the same image region. Because of this more distinctive definition of a sketch, Chum *et al*. report [13] an increase in precision and recall over standard min-hash even with the sketch size reduced to $s = 2$ and the number of min-hashes per image reduced to $k = 60$. Therefore, we only use geometric min-hash in our evaluation.

Cluster Growing. Given a set of cluster seeds, clusters are *grown* by recursively applying query expansion [14]. For each cluster seed discovered by min-hash, codebook-based image retrieval is performed (Sec. 2). For each match above a ranking score

Table 1. Statistics of the datasets, their corresponding ground truths and matching graphs

(a) Statistics of Oxford and Paris datasets. GT denotes ground truth.

	Oxford	Paris
# Images	5,063	501,356
# Features	16,334,970	1,564,381,034
# GT Images	568	94,303
# GT Clusters	11	79

(b) Statistics of the complete matching graphs.

	Oxford	Paris
# Nodes	5052	501,356
# Edges	11,957	11,356,090
avg. valence	4.7	45.3
max valence	83	4,100

threshold, a recursive retrieval is performed until no new images are found. This process can be thought of as finding connected components in the matching graph.

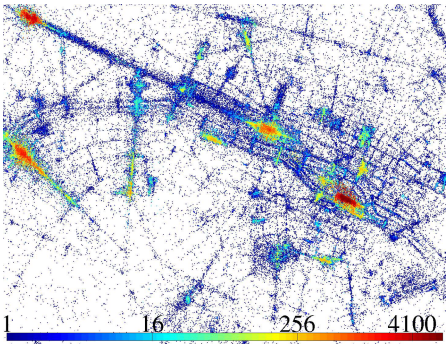
Extension: Spectral clustering. The cluster growing process aims at maximizing recall by growing single-link components in the matching graph. Multiple landmarks can thus potentially end up in the same cluster (see Fig. 2b). We thus propose to segment the grown components with spectral clustering and subsequent merging (Sec. 2).

4 Experimental Setup

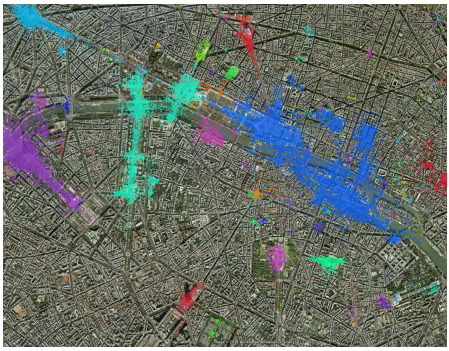
Datasets. We use two different datasets in our evaluation (Table 1a). The well-known Oxford Buildings dataset consists of selected photos depicting eleven distinct landmark buildings in Oxford that were retrieved from Flickr using keyword searches. With 5,063 images and well-separated objects, it is however quite limited. We use the dataset for the sake of comparison but show that the results are not very expressive. Following the approach of Philbin *et al*. [11], we build a clustering ground truth from the provided image retrieval ground truth by combining the sets of "good" and "ok" relevant images for each query.

Due to the lack of a large-scale landmark mining database with a "natural" distribution of tourist photos, we built a larger corpus of photos downloaded from Flickr. We deliberately neither queried particular landmarks nor filtered the query results, so the resulting dataset is closer to real-world conditions. We downloaded all geotagged photos from a bounding box around the inner city of Paris from Flickr and Panoramio and rescaled them to 1024×768 pixels. The Paris corpus therefore contains noise like heavily post-processed images, images of parties, pets, etc. that do not depict landmarks as well as many duplicates and near-duplicates, which we filtered out in order not to bias our evaluation. To establish a ground truth we first over-segmented the complete matching graph using spectral clustering on the connected components (Sec. 2). Inspection showed that the resulting clusters had a high purity with only a negligibly low number of outliers. We then manually joined clusters which showed the same buildings from the same view. The ground truth consists of 79 clustering covering 94k images (Tab. 1a).

Evaluation Measures. We adopt the measures *precision* and *recall* from classification evaluations. Let G denote the ground truth and C a clustering. Then N_C and N_G denote



(a) Density of the matching graph. Color encodes node valence (log scale).



(b) Connected components of the matching graph larger than 20.

Fig. 2. Distribution of downloaded photos in Paris

the total number of images covered by C and G, respectively. To measure how well an algorithm groups similar photos, we use the well-known concept of purity:

$$P = \frac{1}{N_C} \sum_{c \in C} \max_{g \in G} \{|c \cap g|\} \tag{1}$$

Note that this formulation allows more than one cluster in C to be "assigned" to the same ground truth cluster. This measure is insensitive with respect to over-segmentation and missing borderline cases.

We define recall similar to [12]: For each ground-truth cluster g, we find its best representative c in the clustering C and sum up the fraction of member images actually represented by c.

$$R = \frac{1}{N_G} \sum_{g \in G} \max_{c \in C} \{|c \cap g|\} \tag{2}$$

The *Mean Cluster Recall* allows multiple ground-truth clusters to be assigned to the same cluster c, so assigning all photos to the same cluster would optimize recall. Thus, recall is insensitive with respect to under-segmentation and including borderline cases.

5 Results

We now evaluate spectral clustering and min-hash on the Oxford Buildings and Paris datasets. In particular, we show what level of performance spectral clustering can achieve and how the performance of min-hash compares to this. Finally, we give a detailed analysis of the computation time of both algorithms.

5.1 Matching Graph

The first step of the spectral clustering pipeline is to build a matching graph (Section 2). Table 1b shows statistics of the graphs for the two datasets. Interestingly, the average valence of the Paris dataset is an order of magnitude higher than the average valence

Table 2. Statistics of the connected components. The first column gives the number of connected components with a particular size, and the second column gives the total number of images in these components. Connected components of size 1 are images for which no match was found.

(a) Oxford

	CCs	images
total	3,297	5,052
= 1	2,917	2,917
≥ 2	380	2,135
≥ 20	11	929
≥ 100	2	518
≥ 500	0	0

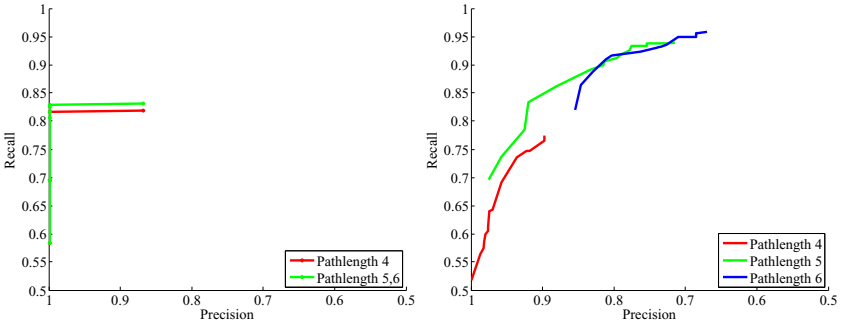
(b) Paris

	CCs	images
total	303,522	501,356
= 1	277,490	277,490
≥ 2	26,032	223,866
≥ 20	397	150,367
≥ 100	63	138,122
≥ 500	19	129,961

of the Oxford dataset. This is due to the extreme density of tourist photos at the most popular public places. The photo with the highest valence (4,100) is a frontal shot of the facade of Notre Dame. Fig. 2a shows the distribution of valences in the matching graph of Paris and Figure 2b shows the connected components. The largest connected component (blue, 58,652 images) spans an area ranging from Notre Dame to the Louvre. This shows that connected components can give a good initial grouping [11], but further segmentation is required for a building-level clustering. In contrast, the largest connected component on the Oxford dataset is All Souls College (406 images). Table 2 gives statistics of the connected component sizes.

5.2 Spectral Clustering

For each connected component we perform a spectral clustering as described in Section 2. This results in 3,881 clusters for the Paris dataset and 410 clusters for the Oxford dataset. Since spectral clustering results in an over-segmentation (images of the same building are split up into several clusters), a subsequent homography-based merging step is performed [11]. We found that this step requires some tuning to produce the desired results, since the error in the estimated homographies increases when accumulating the transformations along long paths. Limiting the path length is an effective way to restrict this effect. Furthermore, it is necessary to define a lower bound on the overlap between the two cluster centers. A too low value results in different views being merged. A too high value limits the permitted degree of viewpoint change too much. Fig. 3 shows the effect of both parameters on precision and recall. On the Oxford dataset, when increasing the overlap threshold, we see an increase in recall while maintaining 100% precision, which means that only correct join operations are performed. From a certain point on, only wrong merges are performed, resulting in a decrease of precision without any change in recall. Due to the simplicity of the Oxford task we cannot draw any conclusions regarding the merging parameters. On the Paris dataset, the tradeoff is more clearly visible: A larger path length leads to a loss in precision, since clusters are incorrectly joined. Too short paths cause us to miss cluster pairs that should be joined, resulting in low recall. The best tradeoff is a path length of 5. Similarly, a too high



(a) Oxford (b) Paris

Fig. 3. Cluster merging performance when varying the overlap parameter from 0% to 100%. The colored lines show results for different path length settings.

overlap threshold will cut off paths between valid matches, while a too low threshold will allow paths between barely overlapping images.

The merging step can still perform false merges in some problematic cases typical for internet photo collections: For example, time-stamps in photos or embedded signatures of the photographer can create false-positive edges in the matching graph that serve as "tunnels" between normally unconnected landmarks. Here, further heuristics would be necessary to discard such matches, which is not done here.

5.3 Geometric Min-Hash

Using spectral clustering as a baseline, we now evaluate the performance of min-hash. Min-hash generates suitable *cluster seeds i.e.* entry points into the image collection [12]. The reason is that many similar images are made at popular places, and thus the probability for a seed is high at landmark buildings. To show this, we compare the clusters discovered starting from min-hash seeds to clusters discovered starting from randomly drawn images. We then apply the spectral clustering and merging steps [11] to break down the connected components to building-level clusters and evaluate the resulting clustering.

Seed Generation. The parameters of the seed generation procedure are the sketch size s and the number of sketches k. Fig. 4 shows the influence of these parameters (dashed lines). The more sketches are used, the more collisions occur and the more seeds are generated. For larger sketch sizes the algorithm becomes more selective and returns only very similar images, which significantly decreases the number of seeds.

Duplicate Removal. Since by design, the probability for min-hash collisions is proportional to the similarity (visual word set overlap) of the colliding images, duplicate images are returned first and introduce arbitrary seeds which have a lower probability of belonging to a landmark cluster. Therefore, it is necessary to filter the duplicates from the seeds. Chum *et al.* [12] manually removed duplicates for their experiments. We chose to perform duplicate detection using the $tf \cdot idf$ distance of a seed image pair. By visual inspection of min-hash seeds, we determined a $tf \cdot idf$ threshold of 0.3. Seeds with a higher $tf \cdot idf$ score are considered duplicates and are removed. Figure 4 (solid

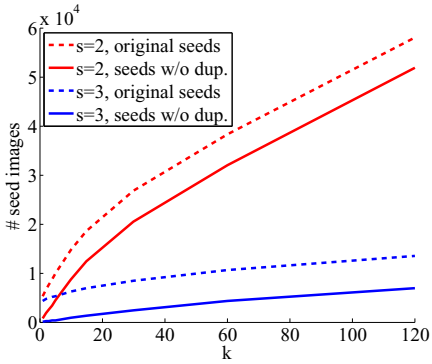


Fig. 4. Number of cluster seed images for the Paris dataset for different sketch counts k and sketch sizes s.

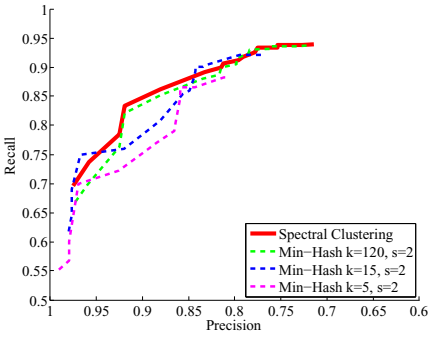


Fig. 5. Min-hash and spectral clustering recall and precision for varying overlap threshold. The path length is set to 5.

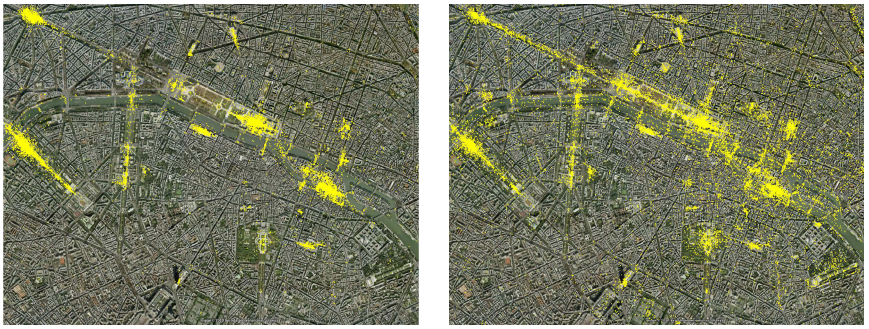


Fig. 6. Distribution of min-hash seeds (drawn as yellow dots) for k=60 and s=2 (left) and the distribution when drawing the same number images (31,946) randomly (right).

lines) shows the effect on the number of seeds for different settings of s. For $s = 3$, most of the returned seeds are duplicates, and the number of seeds detected increases only very slowly. We thus use a setting of $s = 2$ for our following experiments.

Min-Hash vs. Random. Fig. 6 (left) shows the distribution of the min-hash seeds for $k = 60$ and $s = 2$ (31,946 images), and Fig. 6 (right) shows the distribution when the same number of images is selected randomly. The random images are much more scattered over the city while the images selected by min-hash concentrate around the landmarks, which is the desired behaviour for a seed selection algorithm.

For a quantitative comparison, we use the following procedure: For each setting of s, we consider the number of seed images N_s that min-hash produces and randomly draw N_s images from the dataset. This is done 10 times, for each value of s. In our evaluation, we give the average results for the 10 sets of images. We only perform this comparison on the Paris dataset.

Cluster Growing. Starting from the seeds, we grow clusters by query expansion (Sec. 3). Each resulting cluster corresponds to a connected component of the matching graph. Table 3 shows the results of the cluster growing process. The number of discovered clusters

Table 3. Results of the cluster growing process starting from min-hash seeds and random images. On the Paris dataset, the ground truth consists of the 79 largest landmark clusters. CC and GT denote connected component and ground truth respectively. The sketch size is $s = 2$.

(a) Oxford

k	# seeds	# clusters	# images covered	GT covered
1	8	3	4	38.98%
2	12	4	40	52.56%
3	16	6	43	66.31%
5	36	8	54	66.31%
10	55	13	71	75.84%
30	167	26	108	80.07%
120	422	63	367	84.48%

(b) Paris

		min-hash			random (avg. of 10)		
k	# seeds	# CCs	avg. CC size	GT covered	# CCs	avg. CC size	GT covered
1	784	58	2,158.2	81.87%	590.0	222.6	87.11%
2	1,883	102	1,268.9	90.39%	1,407.5	97.6	94.53%
3	2,570	141	941.1	93.38%	1,887.9	74.5	96.70%
5	4,437	220	620.5	94.53%	3,236.1	44.9	98.63%
10	8,753	360	389.3	97.56%	6,292.2	24.5	99.60%
30	20,453	915	161.6	99.56%	14,463.7	11.8	99.98%
120	51,855	3,022	52.8	100.00%	35,607.5	5.7	100.00%

is roughly proportional to the number of seeds. However, we find less new images when increasing the number of sketches k, because the largest clusters have the highest probability of being found [12]. On the Paris dataset, the 79 largest clusters in the dataset (which make up the ground truth) are almost fully discovered already for $k = 10$.

Comparing min-hash to a random selection of seed images shows that roughly ten times the number of connected components are found, but their average size is roughly ten times smaller. This shows that randomly selected images more likely belong to small connected components than images selected using min-hash.

Spectral Clustering of Discovered Connected Components. Even very low settings of k produce impressive recall but the clustering lacks precision, because the clusters discovered using query expansion become too large and thus cover multiple landmarks. To break up the clusters to a building level, we apply spectral clustering and homography-based cluster merging (Sec. 2) on top of the min-hash pipeline [12]. Table 4 shows the effect of this additional step on the results. Since the results vary only slightly for different min-hash sketch counts, we only give mean and standard deviation values computed using settings of k from the range $[1, 120]$. The additional steps strongly improve the clustering precision while only slightly decreasing recall. Fig. 5 shows a comparison of the precision-recall curves of the two approaches when varying the minimum overlap parameter of the merging step. The recall of min-hash increases with a growing number of sketches and almost reaches the recall of spectral clustering at 120 sketches. Note that the precision of min-hash does not change much when varying the number of sketches.

Table 4. Summary of precision/recall of min-hash and subsequent cluster growing before and after spectral clustering on the Paris dataset. Tolerance values are given in standard deviation.

	Precision	Recall
w/o spectral clustering	50.1% ± 0.04	97.0% ± 0.07
w/ spectral clustering	85.7% ± 7.4	83.9% ± 9.9

To summarize, the extended min-hash pipeline achieves performance comparable to the spectral clustering pipeline for high values of k. However, the largest landmark clusters are already discovered for low settings of k. Reducing the number of sketches k trades off recall for computational speed. In the following section, we will investigate this tradeoff more closely.

5.4 Runtime Analysis

Pairwise Matching. The first step of the spectral clustering pipeline the pairwise image matching (Sec. 2). The runtime of this step consists of the inverted-file matching and the RANSAC verification of the top k matches. Performing inverted file lookup for such a database size has an effective runtime that is quadratic in the number of images[2]:

$$T_{if} = (N \cdot (N-1))/2 \cdot c_m \quad . \tag{3}$$

Here, N is the number of database images and c_m is a constant for the matching time of one image pair. In our measurements, $c_m \approx 5.75 \cdot 10^{-6}$ seconds. The time complexity of the RANSAC verification is linear in the number of images and depends on the number l of matches that we verify for a query.

$$T_v = N \cdot l \cdot c_v \tag{4}$$

Here, c_v denotes the time required for the spatial verification of one image pair. It can approximately be considered a constant. We measured the verification time to be on average $c_v = 0.0005$ seconds. The number of top l matches trades off missing potential matches for computational time. Following [15], we choose $k = 800$ for the Oxford dataset. For the Paris dataset we use $l = 15,000$ following considerations of worst-case match counts. Table 5a gives the number of operations and an approximate computation time for both databases using our implementation.

Cluster Growing. Cluster growing (Sec. 3) is performed using query expansion, *i.e.* by querying the inverted index with the seed image and using each verified result as another query. This process is iterated until no new results are found and the whole connected component is explored. Thus, the number N_c of queries for exploring a connected component corresponds to its size. The runtime for this can be written as:

$$T_g(c) = N_c \cdot (N \cdot c_m + l \cdot c_v) \quad . \tag{5}$$

[2] Assuming N images, f matching features per image and a codebook with C entries, the expected number of inverted file entries processed per query is (without stop word removal) $\frac{N \cdot f^2}{C}$.

Table 5. (a) Number of matching operations and runtime estimates of the pairwise matching for both datasets. (b) Approximate runtime of cluster growing for different min-hash settings. For all results, a sketch size of 2 was used.

(a)

	Oxford		Paris	
	Ops	t	Ops	t
p/w matching	$1.2 \cdot 10^7$	73 s	$1.3 \cdot 10^{11}$	201 h
verification	$4.0 \cdot 10^6$	34 m	$7.5 \cdot 10^9$	44 d
$\sum$		35 m		52 d

(b)

	Oxford		Paris	
k	# imgs	t	# images	t
5	54	23 s	102,865	12.4 d
10	71	30 s	108,982	13.1 d
15	80	34 s	113,328	13.6 d
30	108	46 s	120,760	14.5 d
60	246	106 s	126,814	15.2 d
120	367	157 s	134,884	16.2 d

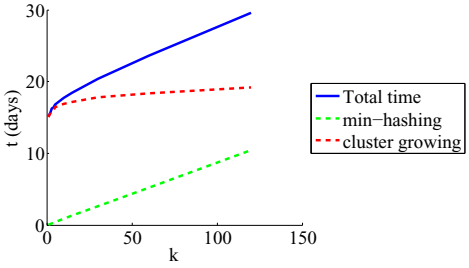


Fig. 7. Computation time of the min-hash pipeline for varying k

Table 5b gives a comparison of the computation times for different sketch counts. Since most of the large clusters are already discovered using very few sketches, the sketch count does not affect the computation time of the cluster growing step much.

Min-Hashing. An appealing property of min-hash is that its time complexity is (in practice) linear in the number of images [12]. The computation of the hashes itself takes up the major part of the processing time. Insertion into the hash table and finding collisions is comparably fast. Computation time increases linearly in both the number of sketches s and the sketch size k. In our implementation, the computation of a geometric min-hash took on average 0.015 seconds per image and sketch for $s = 2$, and 0.016 seconds per image and sketch for $s = 3$. Hashing and finding collisions took 0.0008 seconds per image and sketch. So, for $k = 5$ and $s = 2$, the total time for computing min-hash seed candidates for the Paris dataset is 10.4 hours, whereas for $k = 120$ computation would take 10.4 days. Additionally, a spatial verification of the candidate seeds is performed, but the computation time for this is negligible in comparison.

Spectral Clustering. Spectral clustering involves three basic steps. First, it is necessary to compute a singular value decomposition on a modified matching graph ($N \times N$) into k singular values, where k is the number of clusters we want to obtain for every connected component. Then we need to find k out of N vectors of dimension k which are orthogonal to each other to initialize k-means clustering in order to obtain stable

results. Finally, we need one run of k-means with k centroids, N points, and k dimensions. This procedure has to be repeated for different k, to find the appropriate number of clusters for each connected component.

Experiments showed that runtime of one spectral clustering run is approximately linear in the number of photos, but quartic in the number of clusters k. For large connected components we also need larger values for k, in order to discover cluster on a building level. Therefore, runtime is dominated by the few largest connected components: Clustering the largest four connected components takes about 2 weeks, whereas clustering all other connected components (smaller than 5,000 images) only takes 2 hours in total. Since all methods explore the four largest connected components, we can approximate the runtime for spectral clustering with 2 weeks in each case.

Summary. We now summarize the computation times of both approaches for the Paris dataset. We will not cover feature extraction time, because this step is necessary for both approaches. The total computation time of the spectral clustering pipeline includes pairwise matching (47 days), spectral clustering (14 days) and cluster merging (12 hours). The total computation time of the spectral clustering pipeline on the Paris dataset is thus 61.5 days. (Computation was performed on a cluster of PCs.)

The computation time of min-hash is influenced by the sketch count k. This parameter directly affects the time for computing the min-hashes and it indirectly affects the cluster growing time through the number of discovered clusters. Fig. 7 gives an overview of the computation time of the min-hash pipeline for different settings of k. For a choice of $k = 5$, the total runtime is 16 days, and for $k = 120$, the runtime is 30 days. Depending on the parameter settings, min-hash is thus two to four times faster.

6 Conclusion and Outlook

We evaluated two approaches for landmark mining in large-scale image collections. We presented a new dataset and ground truth for the evaluation of such approaches. Our results show that spectral clustering is capable of clustering the pairwise image matching graph into building-level clusters, however at high computational cost.

Min-hash focuses the cluster growing step on promising entry points, and thus trades off speed for recall. However, it is necessary to implement duplicate removal to suppress low-quality seeds. We also showed that using the connected components directly as clusters, as proposed by Chum *et al.* [14], results in low precision, which can be improved by spectral clustering the connected components. The resulting approach was shown to be a good tradeoff between computation time and clustering quality, but is relatively complex. In particular, it seems overkill to first over-segment the image collection and to then again join clusters of the same building.

An ideal approach would find seed images with a high probability if they are good representatives for their neighbors. The growing step should avoid under-segmentation, so that it becomes unnecessary to run a costly re-segmentation process. That is, the seed growing step should stop as soon as a single building is covered. Achieving these goals will require deeper investigation, which is ongoing research.

Acknowledgements. This project has been funded by the cluster of excellence UMIC (DFG EXC 89). We thank O. Chum and J. Philbin for their help and for interesting discussions.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building Rome in a Day. In: ICCV 2009 (2009)
2. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.-M.: Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 427–440. Springer, Heidelberg (2008)
3. Snavely, N., Seitz, S., Szeliski, R.: Modeling the World from Internet Photo Collections. IJCV 80, 189–210 (2008)
4. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.: Multi-View Stereo for Community Photo Collections. In: ICCV 2007 (2007)
5. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: CVPR 2009, IEEE (2009)
6. Snavely, N., Seitz, S., Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. In: SIGGRAPH 2006 (2006)
7. Gammeter, S., Quack, T., Van Gool, L.: I Know What You Did Last Summer: Object-Level Auto-Annotation of Holiday Snaps. In: ICCV 2009 (2009)
8. Simon, I., Seitz, S.M.: Scene Segmentation Using the Wisdom of Crowds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 541–553. Springer, Heidelberg (2008)
9. Simon, I., Snavely, N., Seitz, S.: Scene Summarization for Online Image Collections. In: ICCV 2007 (2007)
10. Quack, T., Leibe, B., Van Gool, L.: World-Scale Mining of Objects and Events from Community Photo Collections. In: CIVR 2008 (2008)
11. Philbin, J., Zisserman, A.: Object Mining using a Matching Graph on Very Large Image Collections. In: ICCVGIP 2008 (2008)
12. Chum, O., Matas, J.: Large-scale discovery of spatially related images. In: PAMI (2010)
13. Chum, O., Perdoch, M., Matas, J.: Geometric min-Hashing: Finding a (Thick) Needle in a Haystack. In: ICCV 2007 (2007)
14. Chum, O., Philbin, J., Sivic, J., Zisserman, A.: Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In: ICCV 2007 (2007)
15. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. In: CVPR 2007 (2007)
16. Strecha, C., Pylvanainen, T., Fua, P.: Dynamic and Scalable Large Scale Image Reconstruction. In: CVPR 2010 (2010)
17. 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: CVPR 2009 (2009)
18. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV 60, 63–86 (2004)
19. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. IJCV 60 (2004)
20. Sattler, T., Leibe, B., Kobbelt, L.: SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter. In: ICCV 2009 (2009)
21. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS 2001. MIT Press (2001)
22. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69 (2004)
23. Chum, O., Philbin, J., Zisserman, A.: Near Duplicate Image Detection: min-Hash and tf-idf Weighting. In: BMVC 2008 (2008)
24. Broder, A.: On the resemblance and containment of documents. In: SEQS 1997 (1997)
25. Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: CIVR 2007 (2007)