Size Does Matter: Improving Object Recognition and 3D Reconstruction with Cross-Media Analysis of Image Clusters

Stephan Gammeter[1], Till Quack[1], David Tingdahl[2], and Luc van Gool[1,2]

[1] BIWI, ETH Zürich
{gammeter,tquack,vangool}@vision.ee.ethz.ch
http://www.vision.ee.ethz.ch
[2] VISICS, K.U. Leuven
david.tingdahl@esat.kuleuven.be
http://www.esat.kuleuven.be/psi/visics

Abstract. Most of the recent work on image-based object recognition and 3D reconstruction has focused on improving the underlying algorithms. In this paper we present a method to automatically improve the quality of the reference database, which, as we will show, also affects recognition and reconstruction performances significantly. Starting out from a reference database of clustered images we expand small clusters. This is done by exploiting cross-media information, which allows for crawling of additional images. For large clusters redundant information is removed by scene analysis. We show how these techniques make object recognition and 3D reconstruction both more efficient and more precise - we observed up to 14.8% improvement for the recognition task. Furthermore, the methods are completely data-driven and fully automatic.

Keywords: Image retrieval, image mining, 3D reconstruction.

1 Introduction

Recognition, reconstruction and analysis of 3D scenes are topics with broad coverage in the Computer Vision literature. However, in recent years the enormous amount of photos shared on the Internet has added a few new twists to these research problems. On the one hand there is the obvious challenge of scale, on the other hand there is the benefit that photos shared online usually come with meta-data in form of (geo-) tags, collateral text, user-information, *etc*. Besides the interesting research that can be done with this data, they also open doors for real-world deployments of computer vision algorithms for consumer applications, as recent examples from 3D scene browsing [1], or face recognition [2] have shown.

Consequently, a number of works have started to exploit these cross-media data in several ways [1, 3–13]. Quack *et al*. [10] have used a combination of GPS tags, textual and visual features to identify labeled objects and events in data

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 734–747, 2010.

from community photo collections such as Flickr[1]. Crandall *et al.* [6] have done similar experiments, but at even larger scale (up to 10s of millions of photos) and analyzing temporal movement patterns of photographers in addition to GPS, textual and visual features. Very recently, with works such as [7], the community has started to exploit these cross-media data collections from the Web in order to build applications for auto-annotation.

Also in the 3D reconstruction field there has been a long-lasting interest in reconstruction the whole world in 3D, and not astonishingly, community photo collections nowadays serve as a data source for this purpose as well [1, 5, 12]. In spite of the different target applications, all these works have one theme in common: the underlying data structures are clusters of photos depicting the same object or scene, accompanied by some cross-modal data, such as (geo-)tags *etc.* In this work we are particularly interested in clusters of consumer photos showing "places". Places include any geographic location, which is of interest to people, such as landmark buildings, museums, mountain peaks, *etc.* Similar to most works cited above, in a first step we also cluster images in order to identify relevant places. While attention has recently been directed towards harvesting larger and larger collections of data, in this paper we want to take a step back and look at the collected image clusters in more detail. The objective is to investigate if and how basic knowledge about the 3D scene in combination with analysis of cross-media data is helpful towards improving the quality of the database of places as well as the performance of applications building on top of the database. More precisely, we show how

- cross-media retrieval helps identifying missing information for small clusters.
- scene analysis helps removing redundant data in large clusters.
- those measures affect performance of object recognition and 3D reconstruction applications relying on the database of image clusters.

In other words, if we take the analogy of a web search engine for hypertext documents, we focus on the crawling and indexing part of the system. While in the hypertext retrieval community this topic is well documented, in the Computer Vision field most work has focussed on the retrieval side of things [14–16]. For instance, Chum *et al.* [14] could show how to improve retrieval precision using query expansion, using an algorithm which operates mainly at retrieval time.

With our improvements on the crawling and indexing stages of the pipeline, we can indirectly achieve significant improvements in an object recognition setting. We focus on the object recognition task, since there are clearly defined evaluation metrics available. In addition our contributions are valuable for unsupervised 3D reconstruction as well, however, the improvement in this application is in general less easily quantified, but easily visualized. Most importantly, for both scenarios, every proposed improvement happens offline and all the processes we show in this paper are fully automated.

The paper is structured as follows: Section 2 describes our basic methods for image cluster mining and object recognition. The core of our methods for

[1] `www.flickr.com`

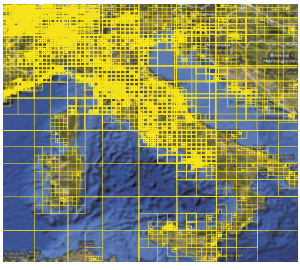


Fig. 1. Geographic quadtree used for image crawling. The example shows the area around Italy. Note how the density of tiles adapts to the number of photos available, e.g. densely covering populated areas and with large tiles on the ocean.

automated cross-media cluster analysis and optimization follows in Section 3. Experiments and analysis of the effects of optimization on retrieval tasks follow in Section 4. Section 5 concludes the paper.

2 Mining and Recognition of Objects

As discussed in the introduction, harvesting photos from online services for landmark mining, recognition or 3D reconstruction has been addressed in a number of recent works. We build on some of those ideas in order to construct our own image mining pipeline. We also introduce the object recognition methods, which we apply on top of the mined data.

2.1 Object Mining

Several ways have been proposed to collect data from online photo collections in order to solve computer vision tasks. They either start out by querying with certain keywords such as "Rome", "Venice" [1, 12, 17, 18], or with collecting geo-tagged photos [6, 10]. For bootstrapping our system we chose the latter strategy.

In order to harvest photos from Flickr based on their geo-tags, we overlay several geographic quad-trees over the world and retrieve the number of photos in each tile. Each of the trees is initialized by a country's geographic bounding box coordinates. Recursively this initial area is then subdivided as follows. We retrieve the number of photos in the current area from the Flickr API. When the number of photos is higher than a threshold (250 in our implementation), we split the area into 4 tiles of equal size and repeat the process for each tile. The recursion stops when the threshold for the number of photos is reached. In addition, the dimension of the tile in meters also serves as a second stopping criterion: the process returns when the tile's extent is less than $200m$ (on the smaller side). The outcome of this is shown in Fig. 1. Photos are then downloaded for all child leaves, and the photo clustering is also distributed based on the

child leaves of the geographic quadtree. For clustering photos, we then proceed as proposed in [10] in three steps

1. Match photos pair-wise using local image features (we use SURF [19]).
2. Build a set of image similarity matrices. We create one matrix per geographic leaf tile. The similarity is the number of inlying matches after RANSAC filtering of feature matches for each similar image pair.
3. Cluster the photos using single-link hierarchical agglomerative clustering.

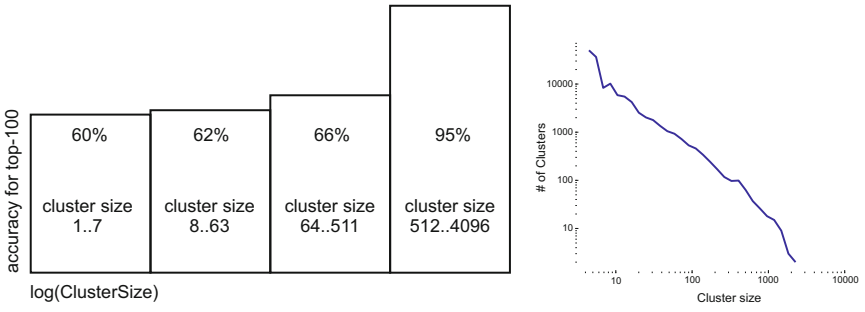
For each cluster we keep its photos including their meta data (tags, titles, user information *etc.*) for further processing. Very similar to [10] we observed that the image clusters usually represent one common object, but covered with photos from various viewpoints and under various lighting conditions *etc.* Thus, we think of each cluster representing one particular object and consider the images of a cluster to form an exemplar based object model.

Qualitatively, we think our crawling method ends-up with very similar data like [10], but is significantly more efficient ([10] scans the world in evenly distributed tiles of equal size, in effect querying a lot of empty cells unnecessarily.) We believe our crawling approach is also beneficial over [6], since we can split the clustering problem into smaller parts, and the tree based approach is directly "pulled" towards densely populated areas already while collecting the data. In contrast, [6] is one huge clustering problem. Finally we crawled a significantly larger dataset than [7] with our quadtree method (17 million images w.r.t. 4 million), to be able to compare our results in terms of object recognition with theirs as a baseline, for the remainder of this paper we conduct all our analysis on the same data (the dataset is available from the authors web-site).

2.2 Object Recognition

Given a query image depicting a landmark, the goal is now to identify and label this object based on the information aggregated in our reference database of image clusters. This task is very similar to the one recently posed by Gammeter *et al.* [7]. (In contrast to image/object retrieval [20–22], where the expected outcome is a ranked list of similar images or images showing the same object as the query, sorted by similarity).

Much like the work of [7], at the lowest level, our object recognition system builds on "standard" visual word based image retrieval. Local image features [19] are clustered into a visual vocabulary of 1 million visual words using approximate k-Means (AKM) [21]. An initial *top-n list* of the n most similar images in the database in terms of set intersection is efficiently computed using an inverted file structure. We then use RANSAC to estimate a homography between the query image and every image in the top-n list. Candidate images for which the RANSAC estimate yields less inliers than a threshold (13 in our implementation) are discarded. We then simply let the image with the highest number of inliers to identify the object in the query image. This is in contrast to [7], where the images in the filtered *top-n list* are used to vote for "their" object.



(a) Objects represented by small clusters yield lower recognition rates.

(b) Many clusters are small

Fig. 2.

3 Cross-Media Cluster Analysis

The main object of study for the remainder of this paper are the image clusters mined from the Internet as described in the preceding section. Given our target applications — object recognition or 3D reconstruction — we can now analyze and improve the image clusters in several aspects. The first and most obvious aspect is cluster size. Intuitively, objects which are represented by a smaller cluster should be more difficult to recognize, since they may lack images taken from an important viewpoint. Fig. 2(a) shows a (histogram) plot of the cluster size versus recognition rate. It confirms that recognition tends to be more successful for larger image clusters. (Detailed results for recognition are given in Section 4 of this paper.) Further, as illustrated in Figure 2(b), it seems that the cluster size distribution follows a power law: $p(ClusterSize) \propto \frac{1}{ClusterSize^{\alpha}}$ with a maximum likelihood estimate of $\alpha_{MLE} = 1.41$. Such distributions are extremely heavy tailed, and thus imply several characteristics. For instance, one should note that it is unreasonable to consider an average cluster size, since the expectation value diverges for $\alpha \leq 2$. Further, from the power law distribution also follows that the majority of clusters is small, but due to the heavy tail quite a few clusters are disproportionally large. It stands to reason that these extremely large clusters carry a large amount of redundant information. Thus, in the following, we investigate the effect of expanding small clusters with additional (non geo-tagged) images, and propose strategies for reducing redundant information contained in very large image clusters.

3.1 Expansion of Small Clusters

Even though an increasing number of digital images shared online contain geo-tags, owning a GPS-equipped camera is still not standard today. Consequently, a significant fraction of clusters mined using an approach relying on geo-tags, consists only of a handful of images (Fig. 2(b)). In fact, in our dataset 81% of all clusters contain 10 images or less. For some places this is simply because they are not

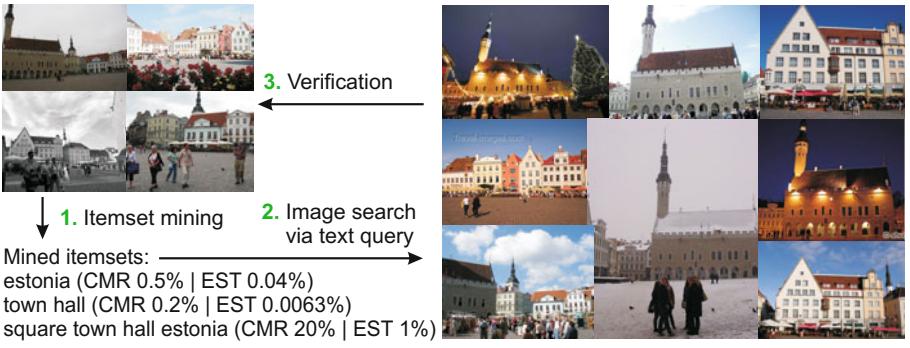


Fig. 3. Cross-media expansion of image clusters: 1) starting out from clusters of images (clustered by their visual similarity with the help of geo-tags for efficiency), we use itemset mining to generate text queries from frequent tags. 2) in order to retrieve additional images thus expanding the image cluster with additional information, 3) and finish with a verifying matching based on visual similarity. We also show the Cluster Match Rate (CMR) for each itemset query (see Section 3.2.)

popular enough. Note that with keyword based mining we would not have been able to find such rare objects in the first place — a list of terms that extensive that it covers such locations is simply not available. But even for much-visited locations many images can lack GPS tags, if the location is *e.g.* inside a building. In order to enrich such small clusters, we propose to use a cross-media crawling method. First, text queries are generated using the tags associated with an existing image cluster. To that end, we follow the approach taken by [10], where text queries are automatically created from the meta-data of the photos in each cluster. They then use these queries for crawling Wikipedia articles intended to serve as descriptions for image clusters. In order to generate the text queries automatically, the authors propose to use itemset mining [23] to form frequent combinations of tags for each cluster. We follow the same approach, but query the WWW for images instead for Wikipedia articles. For the remainder of this paper we call these automatically generated text queries *itemset queries*. The itemset queries are used to query common *Google* for additional photos. The retrieved images are then matched against the images inside the cluster, again by estimating a Homography using RANSAC and SURF [19] features. Matching images are added to the cluster. Match vs. no match is determined based on an inlier threshold of 15 feature correspondences. This procedure is illustrated in Fig. 3.

3.2 Efficient Itemset Query Selection

It turns out, that for a surprisingly large amount of clusters additional images can be retrieved (96% of clusters in our test dataset have been expanded by at least one image). Furthermore, one should note that as shown in Fig. 2(a) this procedure is more likely to be successful for larger clusters than for smaller ones. Obviously, when applying such an automatic query generation approach for a large amount of data with many clusters, the number of text queries can

reach a level, where efficiency considerations become crucial. (Each cluster can generate dozens or even up to hundreds of different itemset queries). Unlike other resources like bandwidth or storage, the amount of HTTP requests that can be made to a public service like Google is often limited. Furthermore, results from search engines are returned aggregated to pages. Each page usually contains only about 20 images and requires one additional HTTP request to retrieve it. While the prices of resources like computation power (Moore's Law), bandwidth (Nielsen's Law) and storage (Kryder's Law) drop exponentially over time, this most likely does not imply the same exponential increase in the number of queries that can be made to search engines. (They are already confronted with a rapidly growing user base.) So, unless one has the resources to crawl the entire Internet in order to avoid public search engines, it is of great interest to minimize the number of queries required. However, if an itemset query is not very specific (e.g. "town hall", compare Fig. 3), it might lead to the retrieval of a large number of images, which do not have anything in common with the object in the cluster, and consequently won't match to its images. In other words, to be efficient, we have to find a way to automatically select itemset queries which have a higher probability of returning relevant images.

As a basic measure for how successful an itemset query is in retrieving additional images of the object, we define first the *cluster matching rate (CMR)*.

$$CMR = \frac{\#\text{ Matching images}}{\#\text{ Retrieved images}} \tag{1}$$

This is a straightforward choice, which records for a given itemset query the fraction of retrieved images that match to the images in the database cluster. While CMR is useful to determine the quality of an itemset query once all images have already been retrieved and matched, an efficient approach should discard itemset queries with low CMR well before that. This could entail estimating the CMR, which in turn would require in the order of $(1/\text{CMR} - 1)$ images. Thus, the lower the CMR of an itemset query, the more images we would have to download before we can reject it. By comparing the improvement in recognition quality on the test set when considering all queries vs. the improvement when only accepting queries with a CMR above a given threshold, we find that the largest improvement comes from queries with a CMR between 0.01 and 0.1. This is shown in Fig. 7. In other words, we might have to download at least 100 images before we can safely reject any itemset query. We can, however, exploit an observation made by [10, 24]. The authors used text queries in order to retrieve Wikipedia articles intended as descriptions for the image clusters. The trick they came up with, is to verify the retrieval result by matching images from the articles to the source cluster. They found that itemset queries yielding articles containing images matching the cluster have a higher probability of yielding matching images from other sources as well. This could be a crude indicator to a-priori assess an itemset query's CMR. In order to test this hypothesis, we downloaded and indexed a dump of all English Wikipedia articles and their images. Then, as illustrated in Fig. 4, for any given cluster, we query both the text index with the itemset queries and the image index with the cluster's

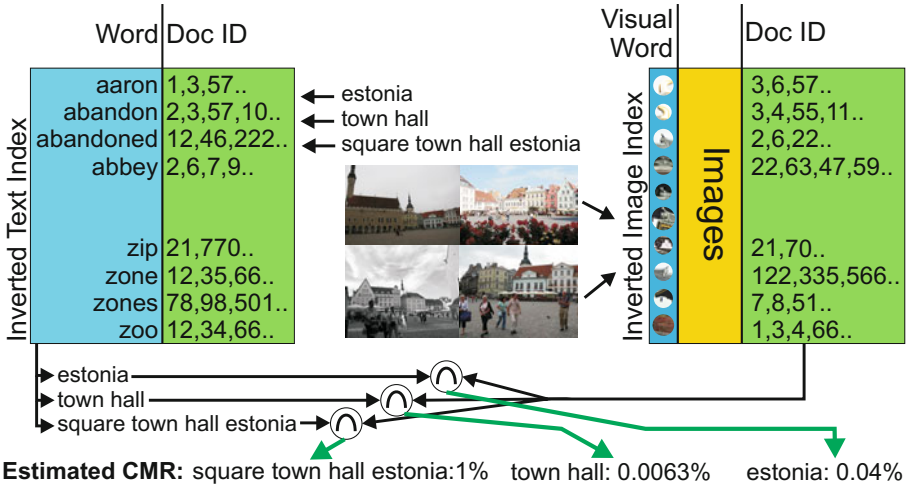


Fig. 4. A priori estimation of CMR using a local copy of Wikipedia. An inverted text index and an image index are queried simultaneously. The result sets are intersected in order to determine if the text could have yielded any useful images

images. The text index returns ids of Wikipedia articles with words matching the itemset query, while the image index returns ids of Wikipedia articles with images matching the cluster images. The two result sets are then intersected. The ratio of the number in the intersected set and the number of elements in the set returned by the text index can be taken as a crude estimate of the CMR. This estimation is shown as EST value in Fig. 3. With this measure at hand, we are able to discard a significant amount of irrelevant itemset queries early on.

3.3 Reduction of Large Clusters

While small objects that are only modeled by few images in their respective image clusters are more difficult to recognize, having too much data is not a blessing either. Unusually large amounts of photos are often collected at popular tourist destinations such as Notre Dame de Paris, or the Eiffel Tower. Many of these photos contain redundant information, which in an image retrieval scenario, unnecessarily increase the size of the inverted index. Furthermore, since our method from Section 3.1 allows for augmenting almost any cluster by an arbitrary amount of images, we desire to find a method that purges the redundant information, while leaving complementary information untouched. Note that it is a-priori also unclear what "a good" number of images would be for an arbitrary cluster, since it strongly depends on the 3D scene structure of the given object. This is illustrated in Fig. 5. The object on the top left is a free standing structure which can be photographed from arbitrary viewpoints, so an image cluster which serves as a model for this object has to contain many images. In contrast, the example on the bottom left is the extreme case of a painting in a museum, which can be seen from a small number of viewpoints only, so fewer reference

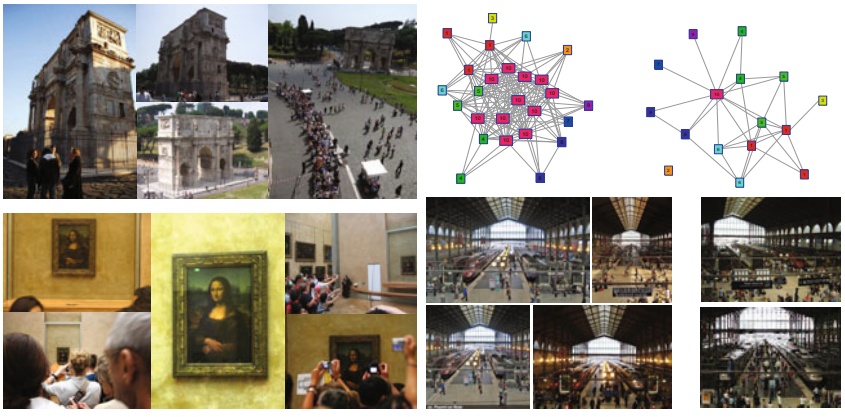


Fig. 5. Left column: example of a free-standing 3D object which can be photographed from many viewpoints (top) vs. one which is visible nearly from only a single viewpoint (and even only 2-dimensional in this particular case). Right column: example of cluster reduction. the full single-linked matching graph is shown on the top left. A complete-linked section which is removed on the top right. Bottom left: images from the removed complete-link segment. Bottom right: images which stay in the cluster.

images are necessary to "describe" the object. In fact, while 3D scene structure makes it impossible to generalize to a "good" cluster size, it is at the same time key to attack the problem of extraordinary cluster size. It turns out, that with some simple 3D scene analysis we can compact the clusters in both an effective and scalable manner. Remember, that the image clusters were created using single-link clustering (Section 2). We now decompose these single-link clusters into several overlapping complete-link clusters. Note the definition of single-link and complete-link criteria in hierarchical agglomerative clustering [25]

$$\text{single-link:} \quad d_{AB} = \min_{i \in A, j \in B} d_{ij} \qquad \text{complete-link:} \quad d_{AB} = \max_{i \in A, j \in B} d_{ij}$$

where for clusters A and B the indices i, j run over the images in the clusters and d_{ij} is the image distance measure that is proportional to the inverse of the number of inliers. Complete-link requires that all image pairs in a segment are fully connected to each other. In our setting this is the case if all image pairs match to each other, which means that they are all taken from a very similar viewpoint. This procedure is illustrated in Fig. 5. Then, for every complete-link cluster with more than 3 nodes, we find the node with the minimum edge-weight-sum (*i.e.* the image most similar to all its neighbors) and remove all other nodes. In essence it is an idea similar to the scene graph in [26], but can here be derived with standard tools using the already calculated distance matrix.

When we remove these highly similar images from the index we automatically remove highly redundant information, while guaranteeing that we keep relevant data. As demonstrated in the experiments in section 4.1 this procedure reduces the index size for retrieval tasks significantly, without affecting precision.

4 Experiments and Results

For all our experiments we used the dataset of [7], which can be obtained from the authors website. The dataset consists of roughly 1 Million images from Flickr that were clustered into 63′232 objects and a test set of 676 images which are associated with 170 of the 63′232 objects. The goal is to correctly identify which object in the database is shown in the images of the test set. The percentage of images correctly associated with their object serves as an evaluation metric. We first report evaluation results on overall recognition performance including the overall effects of cluster expansion and cluster reduction. Finally, we demonstrate that our additionally mined images can be vital in 3D reconstruction.

4.1 Object Recognition

We compare our object recognition system to the one of [7]. On their benchmark dataset we achieve similar baseline performance, as shown in Fig. 6. Adhering to the original evaluation protocol of [7] we consider the percentage of test images for which the correct object is returned in its top-n candidate list vs. the toplist size n. This is an upper limit for the recognition rate after geometric verification. We then applied our cluster expansion and reduction methods to the image clusters in the benchmark dataset. For each of the 170 clusters in the testset we generated itemset queries in order to retrieve additional images for cluster expansion according to the methods described in Section 3.1. We carried out experiments with 3 major image search engines and found that using *Google* yielded the best resultsFor every itemset query we retrieve the first 420 images returned by *Google* to expand our object models. Fig. 6 clearly shows that expanding clusters substantially improves recognition. However, since we only expand clusters that are relevant to the test dataset, we created an unfair situation: the expanded clusters now have a higher probability of randomly occurring in a top-n list. We thus plot the chance level in Fig. 6 (dashed lines) for each expanded index. The comparison highlights that the observed improvement is not simply an artifact of an increased chance level.

A summary of the achieved improvements over the baseline is given in Table 1. The first two columns show cluster retrieval results with bag of visual words lookup for finding the correct cluster in the top n ranked results. The third column shows results for identifying the correct object/cluster on the first rank, using geometric verification. To that end the top ranked 1000 results after bag of words lookup were verified by estimating a Homography mapping between query and retrieved images using RANSAC. For this last task, we achieve 14.8% improvement over [7], when using our cluster expansion method. We also applied the reduction strategy from Section 3.3 to the baseline index, as well as to the expanded index. In both cases we find that our strategy for "purging" unnecessary images does not significantly influence recognition quality as demonstrated in Fig. 6. However, it reduces the inverted index file size significantly, as shown in Table 2. One can also observe that the relative reduction in size is much larger for the expanded index. This is due to the fact, that retrieving additional images via itemset queries more often leads to duplicates or

Table 1. Absolute number of testsets with a correct cluster within the top-100 and top-1000 list. The last column is the absolute number if test images correctly labeled after geometric verification of the top-1000 list.

Description	top-100	top-1000	top-1 Geo.Ver.
Baseline	63.4%	78.6%	73.52%
Expanded	73.0%	86.1%	78.1%

Table 2. Index size comparison for indices built from the original clusters vs. reduced clusters

Description	Original	Reduced
Baseline	1.5GB	1.3GB (−13%)
Expanded	2.1GB	1.5GB (−29%)

near duplicate images compared to images retrieved using GPS queries during the initial crawling of clusters.

4.2 Efficient Itemset Query Selection

In total 2030 itemset queries were generated for the testset of 170 image clusters. As mentioned in Section 3.2, on the fly estimation of CMR based on retrieved images is not beneficial, since at least 100 images have to be retrieved before an itemset query can be safely discarded. However we can use the *estimated CMR* (*c.f.* Section 3.2) as an indicator if an itemset query is useful or not. This is demonstrated in Fig. 7. We found that if we do not discard itemset queries with an *estimated CMR* above 0.01% we retain about 75% of the original improvement. For the test dataset only 40% of all queries fulfill this requirement, however as visible in Fig. 7 these queries alone are responsible for the 75% improvement in recognition quality.

4.3 3D Reconstruction

So far we focussed on demonstrating the outcome of the proposed cluster expansion and reduction methods on an object recognition task. As mentioned earlier, this is due to the easy quantification of the evaluation. However, the same methods can also be beneficial in a 3D reconstruction scenario. The outcome of image based 3D reconstruction is highly dependent on the images used as input. In essence, a large number of high-resolution images taken from a wide variety of viewpoints is desired.

That a simple keyword search or geographic query yields enough images for a decent reconstruction of an arbitrary object is far from a given. Such a strategy in fact only works for a fraction of all landmarks. Even for popular sites, manual keyword search is not trivial, because it is not feasible to efficiently come up with so many appropriate keywords. For less famous landmarks, the situation is

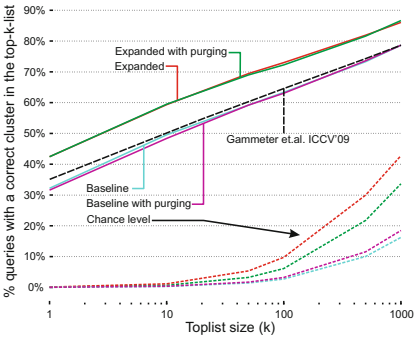


Fig. 6. Expanding clusters with additional images from Google significantly improves top-n score. Reduction of clusters (with "purging" of complete-link segments) does not affect performance, neither for the baseline nor for previously expanded clusters. Improvements can not be attributed to chance, as the dashed lines show.

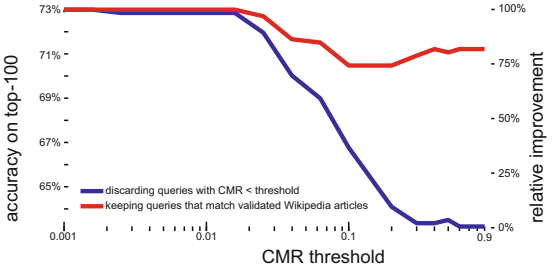


Fig. 7. Considering top-100 accuracy, we compare the overall improvement to the baseline obtained when considering only queries with a CMR above a certain threshold (blue line). The red line shows what happens if we discard itemset queries if and only if their CMR is below a certain threshold *and* their *estimated CMR* is below 0.01%.

even more dire. Even a keyword search with the precise description of the object may not yield enough useful images nor would GPS-based retrieval.

In such cases every single image matters, and a couple of additional images of high quality may dramatically change the outcome of the reconstruction. In this section, we briefly demonstrate with two examples that our cluster expansion method yields additional images crucial for 3D reconstruction. Using the publicly available ARC3D [27] reconstruction tool, we compare the 3D reconstruction of the originally mined image clusters of [7] to the reconstruction based on our expanded clusters. From a set of uncalibrated images, ARC3D generates dense, textured depth maps for each image. Input images are uploaded and processing is performed remotely on a cluster, so that results can be obtained within short time. As demonstrated in Fig. 8, additionally mined images clearly help in reconstructing more complete 3D models.

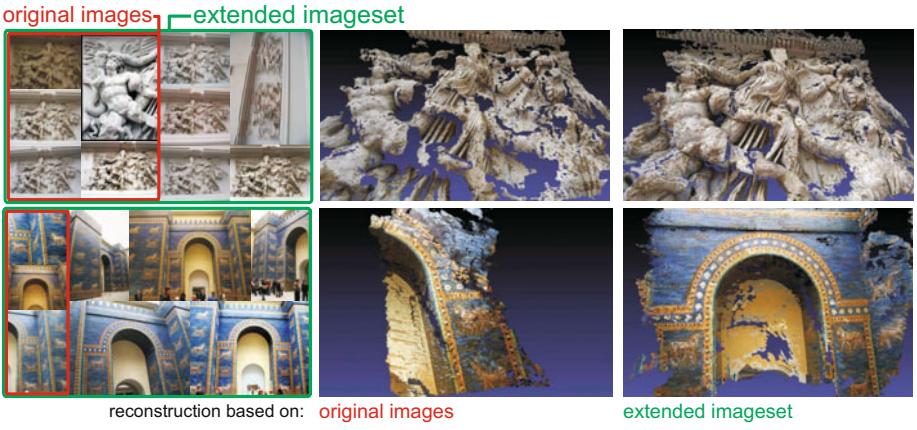


Fig. 8. Unsupervised 3D reconstruction using ARC3D. The first row shows the initial clusters (red box) and the additional mined images (green box). The second row shows the 3D reconstruction only using the initial image set. Reconstruction based on the extended set is shown in the third row. As can be seen, our additionally mined images clearly make the reconstructed 3D models more complete

5 Conclusion

We have shown a fully automated cross-media method to improve the quality of reference databases for object recognition. Small image clusters were enriched with additional information by automatically generating text-queries from image meta-data. Redundant information was purged from large clusters by a simple graph based approach. The combination results in better performance and higher efficiency (in index size) for object recognition tasks on recent benchmark data for object instance recognition. We have also shown that it is possible to exploit the wisdom of crowds to a-priori determine if a potential text query may be useful for retrieving additional images. Finally, while this paper focussed on object recognition, the cluster expansion method would be also valuable for unsupervised 3D reconstruction.

Acknowledgments. We would like to thank the *Swiss National Science Foundation Project IM2* and *Google* for their support of this research.

References

1. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: Exploring photo collections in 3rd. ACM Trans. on Graphics 25 (2006)
2. Stone, Z., Zickler, T., Darrell, T.: Autotagging facebook: Social network context improves photo annotation. In: IEEE Workshop on Internet Vision CVPR 2008 (2008)
3. Zheng, Y., 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)
4. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections (2009)

5. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: ICCV (2009)
6. Crandall, D., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In: WWW '09: Proceedings of the 18th International Conference on World Wide Web (2009)
7. Gammeter, S., Bossard, L., Quack, T., Van Gool, L.: I know what you did last summer: object level auto-annotation of holiday snaps. In: ICCV (2009)
8. Hays, J., Efros, A.A.: Im2gps: estimating geographic information from a single image. In: CVPR (2008)
9. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A.A., Hertzmann, A.: Image sequence geolocation with human travel priors. In: ICCV (2009)
10. Quack, T., Leibe, B., Van Gool, L.: World-scale mining of objects and events from community photo collections. In: CIVR (2008)
11. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. In: ICCV (2007)
12. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: ICCV (2007)
13. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. PAMI 30, 1958–1970 (2008)
14. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: CVPR (2007)
15. Chum, O., Matas, J.: Web scale image clustering. Technical report, Czech Technical University Prague (2008)
16. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
17. 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)
18. Philbin, J., Zisserman, A.: Object mining using a matching graph on very large image collections. In: ICVGIP (2008)
19. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
20. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
21. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
22. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: ICCV (2003)
23. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: SIGMOD (1993)
24. Gool, L.V., Breitenstein, M.D., Gammeter, S., Grabner, H., Quack, T.: Mining from large image sets. In: CIVR (2009)
25. Webb, A.: Statistical Pattern Recognition, 2nd edn. Wiley, Chichester (2002)
26. Snavely, N., Seitz, S., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR (2008)
27. Vergauwen, M., Van Gool, L.: Web-based 3rd reconstruction service. MVA 17, 411–426 (2006)