Object Mining using a Matching Graph on Very Large Image Collections

James Philbin Andrew Zisserman
Visual Geometry Group, Department of Engineering Science, University of Oxford

Abstract

Automatic organization of large, unordered image collections is an extremely challenging problem with many potential applications. Often, what is required is that images taken in the same place, of the same thing, or of the same person be conceptually grouped together.

This work focuses on grouping images containing the same object, despite significant changes in scale, viewpoint and partial occlusions, in very large (1M+) image collections automatically gathered from Flickr. The scale of the data and the extreme variation in imaging conditions makes the problem very challenging.

We describe a scalable method that first computes a matching graph over all the images. Image groups can then be mined from this graph using standard clustering techniques. The novelty we bring is that both the matching graph and the clustering methods are able to use the spatial consistency between the images arising from the common object (if there is one).

We demonstrate our methods on a publicly available dataset of 5K images of Oxford, a 37K image dataset containing images of the Statue of Liberty, and a much larger 1M image dataset of Rome. This is, to our knowledge, the largest dataset to which image-based data mining has been applied.

1 Introduction

Large scale image collections present serious challenges to our current ability to organize, retrieve and utilize the contained image data. In image collections, and especially in collections of tourist photographs collected from sites such as Flickr, certain scenes and objects tend to be photographed much more frequently than others. Automatically grouping these frequently occurring objects is an important first step towards a higher level understanding of these datasets.

Our objective is an association based on *objects*, not on images – we want to associate a set of images of the same object, even if a particular pair of images is quite dissimilar. The objects may vary significantly in scale, viewpoint, illumination or even be partly occluded. The extreme variation in imaging conditions, and the size of the image collections considered (upto 1M images) presents serious challenges to the current state of the art in image-based data mining.

The ability to group objects in such large collections has many potential applications: the frequently occurring objects in a large collection can quickly be perused to summarize the collection; the clusters can provide an access mechanism to the collection; image-based particular object retrieval could use such methods as a filter to reduce data requirements and so reduce search complexity at query time; and techniques such as automatic 3D reconstruction which take as an input multiple views of the same object can then be applied to these very large image collections [23, 29] and can discover canonical views [25].

The central idea we propose is to efficiently generate a sparse matching graph over the entire image corpus: each image is a node in the graph, and the graph edges represent the spatial consistency between sub-parts of the pairs of images linked by the edge – if the images contain a common object then the edge strength will reflect this. We can generate this graph using efficient text-based query mechanisms [17, 18, 27] coupled with accurate spatial verification, using each image in turn as a query.

Given this graph, standard clustering methods can be applied to find the images associated by containing the same object. However, a second major novelty here is that we develop a method which looks for further geometric consistency within the clustering by examining how spatially verified query regions propagate over the graph.

In the statistical text community, many methods and algorithms have been developed for efficient clustering and data mining on *very* large datasets (e.g. the web). Algorithms such as non-negative matrix factorization (NMF), the a priori algorithm, and graph clustering methods are frequently used. Given the success of these algorithms, several works have applied them to the visual domain – as we do here, replacing text words with visual words [7, 27]. Latent topic models from the statistical text literature such as probabilistic Latent Semantic Analysis (pLSA) [10] (a variation on NMF) and Latent Dirichlet Allocation (LDA) [3] have been applied in the visual domain [8, 22, 26], though the aim in these cases was to discover visual *categories*, such as cars or bikes in the image collection, rather than clus-

ter particular objects. Also, these methods simply represent images as a bag-of-visual-words, rather than also requiring geometric consistency between the matches, as we do in our work. Matching graphs have been built previously for image collections though not based on common objects, but on object [9] or image [32] categories and without using spatial consistency between images.

Until recently the two most convincing recent examples for datamining employing some spatial consistency were [20, 28] where the methods were applied in video to cluster particular objects (such as people or scenes). However, recently three papers [4, 11, 21] have appeared with differing approaches to the large scale problem in Flickr image collections.

Chum and Matas [4] explore random sampling for clusters on a 100K corpus using the min-hash method of [5]. This is a very efficient first step, and avoids the more costly building of a complete matching graph employed here – the authors report a runtime of 33mins on a 100K image dataset[1]. However, as the number of visual words in common between images decreases, the chance of discovering a cluster "seed" in [4] decreases, so that potential clusters mined in the complete graph can be missed.

Quack *et al.* [21] mine a large Flickr corpus of 200K photos, but as a first step use geotagging information to decimate the corpus into sets no larger than 4K. The set is then partitioned into clusters using a combination of spatial consistency (as here) and textual similarity.

Li *et al.* [11] mine a 45K Statue of Liberty Flickr photo collection (the corpus differs from the one used here). Their approach is to first cluster the images using the GIST descriptor. Again, this decimates the problem, and spatially consistent clustering can then proceed efficiently within a cluster. As in [4] this first step avoids the expense of building a complete matching graph, but because *images* are matched, rather than objects, the risk is that images with more extreme changes in viewpoint will not be assigned to the same cluster, and will not be associated in subsequent cluster merging.

2 Datasets

Our method is demonstrated on three datasets of varying sizes, all collected automatically from Flickr by searching for images with particular text tags. However, despite restricting the search to these tags, many images are retrieved which bear no relation to the tag initially searched for, as the manual annotation on Flickr tends to be extremely noisy.

2.1 Oxford buildings dataset (5K images)

For groundtruth evaluation, we use the *Oxford Buildings* dataset available from [1]. This consists of 5,062 high resolution (1024×768) images automatically retrieved from Flickr, together with groundtruth occurrences for 11 different landmarks in Oxford.

[1] From email discussions with Ondřej Chum

Dataset	# images	# descriptors
Oxford	5,062	16,334,770
Statue of Liberty	37,034	44,385,173
Rome	1,021,986	1,702,818,841

Table 1. Statistics for each image collection.

2.2 Statue of Liberty dataset (37K images)

This is a larger dataset of 37,034 images downloaded from Flickr containing a tag for the "Statue of Liberty". Although all of these images were tagged with the Statue of Liberty, the annotations are extremely noisy and the dataset contains a large number of other, unrelated scenes. The images were provided by the authors of [25].

2.3 Rome dataset (1M images)

This is a much larger dataset of 1,021,986 images collected from Flickr tagged with "Rome". The dataset contains a large number of tourist and other sites generally taken in Rome, including sites such as the Sistine Chapel and the Colosseum. Again, the images were provided by the authors of [25].

3 Building the Matching Graph

To cluster objects in a corpus of millions of images, one could imagine an algorithm that represented each image by a tf-idf weighted bag-of-visual-words vector (as is done in [27] for example) and measured similarity by a distance on these vectors (such as L_2 distance or χ^2). Then any number of clustering algorithms could be applied (depending on the distance, complexity, desired goal, etc). For example, if only very near-duplicate images are required very tight clusters could be obtained using k-means. However, an image is more than a bag-of-visual-words, the spatial layout of the regions is a stronger indicator of similarity.

This section describes our method for efficiently building a "matching graph" which links *spatially verified* images in the dataset, i.e. an edge is established where there is similarity of both appearance and *spatial layout*.

3.1 Image Representation

Each image is represented as a "bag of visual words" coupled with the position and shape of each underlying feature.

For each image in the dataset, we find affine-invariant Hessian regions [13]. In high resolution images of size 1024×768 approximately 3,300 regions are detected. For each of these affine regions, we compute a 128-D SIFT descriptor [12]. The number of descriptors generated for each dataset is shown in table 1.

For the Oxford and Statue of Liberty datasets, all the descriptors are used for clustering and generating the visual vocabulary. For the Rome dataset, a subsample of 50M descriptors is used for clustering. Clustering is performed using the approximate K-means algorithm of [18, 24]. For all datasets we use a vocabulary size of 1M visual words. Each descriptor in our datasets is then associated with the nearest cluster centre (visual word) using the approximate nearest neighbour method.

Search Engine Our search engine uses the vector-space model [2] common in information retrieval. The query and each document (image) in the corpus is represented as a sparse vector of term (visual word) occurrences and search proceeds by calculating the similarity between the query vector and each document vector, using an L_2 distance. The document vectors are weighted using the simple tf-idf weighting scheme used in text retrieval. This downplays the contribution from commonly occurring and therefore uninformative words.

For computational speed, the word occurrences are stored in an inverted index which maps individual visual words (i.e. from 1 to K, where K is the size of the vocabulary) to a list of the documents in which they occur. Only words which occur in the query need to be considered and generally this is a small percentage of the total (words not in common do not contribute to the distance). In the worst case, the computational complexity of querying the index is linear in the corpus size, but in practice it is close to linear in the number of documents which match a given query, which can provide a substantial saving. Note also that this method is trivially scalable as the corpus can be distributed amongst many computing nodes where each node can query in parallel and merge the results.

With large corpora of images, memory usage becomes a major concern. To help ameliorate this problem, the inverted index is stored in a space-efficient binary packed structure. "Rice compression" is applied to the inverted indices to further reduce memory usage at the cost of a slight increase in search times. The inverted indices for the Rome dataset take 2.8GB in memory, just enough to run on 32-bit nodes. Querying for one image on the Rome dataset takes approximately 1s for a typical query on a 3GHz machine.

3.2 Spatial Re-ranking

The output from performing a query on the inverted file data structure previously described is a ranked list of images for a significant proportion of the corpus. Until now the image features have only been used as a bag-of-words and any spatial configurations of the features have been ignored. The spatial verification procedure estimates an affine homography between the query image and each target image based on how well its feature locations are predicted by the estimated transformation. The top images are then re-ranked using the number of inliers found between the query and target images under the discovered homography. For all three datasets, we re-ranked using the top 400 images retrieved from the inverted index. This is the slowest part of retrieval as the individual region locations have to be retrieved from disk. Querying the Rome dataset can now take several seconds depending on whether the operating system has cached the relevant data.

3.3 Matching Graph

The matching graph is built in the following way. Our graph is initially empty. For each image indexed in our dataset, we query with the whole image over the entire corpus. The top 400 results from the inverted index are spatially verified as described in the previous section. Documents retrieved with more than 20 verified inliers to the query image contribute a new edge to the graph linking the query image to that document. This is repeated for each document in the corpus. The weights on the edges are given by $\frac{N_{I_m}}{(N_q+N_r)/2}$, where N_{I_m} is the number of spatially verified inliers and N_q, N_r are the number of visual words in the query and result respectively. This normalizes for the effect of varying document lengths in the corpus.

3.4 Timings

The graph generated is generally very sparse – for example, the matching graph for the 5K Oxford set contains 24,561 edges (a thousand times less than if every image matched to every other). Searching for every image in the 37K Statue of Liberty dataset takes around 2 hours on a single 3GHz machine. In practice the complexity is $O(Nm)$, where N is the size of the corpus (the number of queries) and m the number of images which match a given query.

4 Graph Clustering for Object Mining

Given the matching graph, we now describe two main methods to mine this graph for different types of information. The first simply takes connected components over the matching graph. Although crude, this often gives a good initial segmentation of the data on which more complex algorithms can be run. Our second method is applied *within* each connected component and comprises a hybrid clustering approach which uses spectral clustering to over-segment the data and a spatially driven merging step to join different views of the same object. A quantitative evaluation of the methods is given in section 5.

4.1 Connected Components

One of the simplest operations for splitting the data is to find the connected components on the matching graph. This greatly reduces the complexity of any subsequent clustering



(a) 56 images | (b) 71 images | (c) 26 images | (d) 25 images | (e) 56 images

Figure 1. Examples of the connected components automatically found on the 5K Oxford dataset. Some components are already extremely accurate in isolating individual buildings/landmarks (see (a)-(c)). (d) & (e) show examples of components linking disjoint objects via connecting views. The number of images in each component is shown beneath the label. Note the significant variation of scale and viewpoint within each component.

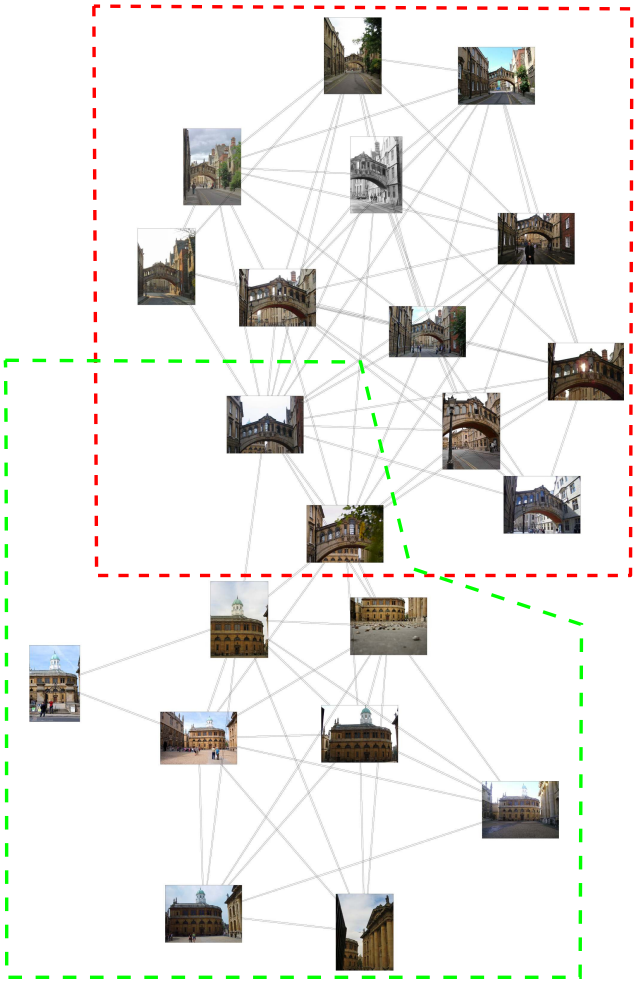


Figure 2. A portion of the full Oxford matching graph for a single connected component. The images in the red and green regions contain the "Bridge of Sighs" and the "Ashmolean Theatre" respectively. Note the two connecting images which contain both objects.

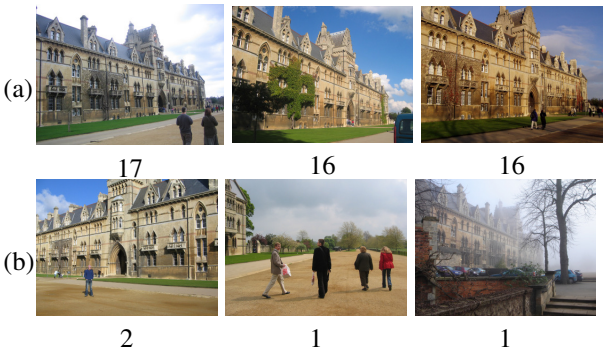


Figure 3. Obtaining canonical views. The three highest (a) and lowest (b) images ranked by degree for a connected component of Christ Church College. The degree is listed beneath each image.

step, as now much smaller groupings of images need to be considered. An example of the sub-graph automatically discovered for a connected component is shown in figure 2.

Even though the method is crude, it can be surprisingly effective at pulling out commonly occurring objects in photo datasets. It naturally achieves a "transitive association" over the views of an object: views A and C may have no matches, even though they are of the same object – this lack of matches may arise from detector drop out, SIFT descriptor instability, partial occlusion etc, and was the subject of the "Total Recall" method of [6]. However, provided A links to B, and B links to C, then A and C will be associated. Thus images can be associated transitively without requiring the additional overhead of [6].

This transitive advantage is also a problem though, in that it joins too much together – a "connecting image" (one that contains multiple disjoint objects) pulls all images of these objects into a single connected component. Figure 1 shows some examples of connected components found on the Oxford dataset. Figure 1(d),(e) show examples of connected components joining disjoint objects via connecting images.

By themselves, the connected components do not imply any ordering on the images they contain. For most practical applications, however, the order is important and here we try to address this problem. Our aim is to return common or canonical views [25] of an object before more obtuse or unusual views. To achieve this, we sort the images within each component in non-increasing order of their *degree* in the matching graph. The degree of an image in the graph is the number of edges entering and leaving the node. Images of particular common views should therefore be ranked higher than those of rarer views (see figure 3).

4.2 Hybrid Method

As previously described, when taking connected components, some graph components are already very pure containing only the one object while others might contain more than one disjoint object due to a connecting view. In this section we aim to detect impure components and rectify the clusters by further splitting of the data, while leaving the already pure clusters as they are. To achieve this, we introduce a novel hybrid graph clustering algorithm which incorporates domain specific knowledge to produce improved visual clusters.

At its core, the method uses a spectral clustering method which produces clean clusters but tends to over-segment the data (see figure 4). A merging step is then applied which examines the clusters produced and recursively joins clusters which seem to be different views of the same object. A measure of the "clusterability" of the component is also used to jointly determine the size of the spectral clustering to be performed and to prevent us from splitting pure components.

Spectral clustering. To perform the spectral clustering we follow the procedure of [16]. Given the desired number of clusters, K, this involves finding the K largest algebraic eigenvectors of the sparse matrix of weighted edges. k-means is then run (with $k = K$) over each normalized row of the eigenvector matrix. This gives a cluster assignment for each image in the component.

The eigenvectors are found using the Implicitly Restarted Lanczos Method (IRLM) [30]. With sparse graphs (as in our case), the IRLM method is found to be extremely fast, taking time close to linear in the number of non-zero edges in the graph.

Inferring K. A great disadvantage of the spectral clustering method is the need to specify the number of cluster centres, K. Obviously in our case, it's impossible to know *a priori* how many clusters should be used for general sets of images. To ameliorate this deficiency we use the selection method of [31]. This involves performing multiple clusterings for different values of K and choosing the optimum value of K to be the one that maximizes the Newman Q measure [15], defined for a clustering, P_k of the data as:

$$Q(P_k) = \sum_{c=1}^{k} \left(\frac{A(V_c, V_c)}{A(V, V)} - \left(\frac{A(V_c, V)}{A(V, V)} \right)^2 \right)$$

where $A(V_i, V_j)$ is the sum of all edge weights between the nodes V_i and V_j, V_c is the set of nodes assigned to cluster c and V is the set of all nodes.

For a large variety of both simulated and real-world graphs, it was shown [14, 15] that large values of Q correlated with better graph clusterings. We also find this to be the case. Additionally, we find that when the optimum value for k has been discovered, if $Q(P_k)$ (which ranges from 0 to 1) is less than some threshold, the component has a low "clusterability" and is probably already a pure cluster. Therefore, in this work, if $Q(P_k) < t$ then we assume the component doesn't require further segmentation. We have found $t = 0.5$ gives reasonable results.

Re-joining different views of the same object. The output from the spectral clustering algorithm previously described tends to be a set of quite pure clusters often containing the object taken from the canonical view. This can lead to problems when an object is frequently taken from a number of different views (see figure 4). Our aim is now to rejoin these two smaller clusters.

To do this, we first find the shortest path in the graph between the two images with highest degree in each cluster. This is efficiently achieved using a breadth-first-search. Once found, we then propagate a "query region" from one of the high degree images to the other using the homographies found during the initial matching. If this projected query region overlaps the other image more than a certain amount after being propagated along the path, then we join the clusters and otherwise leave them separate. A query region could be a generated from the image boundary rectangle (as used here) or a convex hull of the inlier matches from other images. Figure 4 shows an example for two views of All Soul's Oxford using the image boundary rectangle slightly reduced in size. We have found this technique to be effective for joining canonical object views.

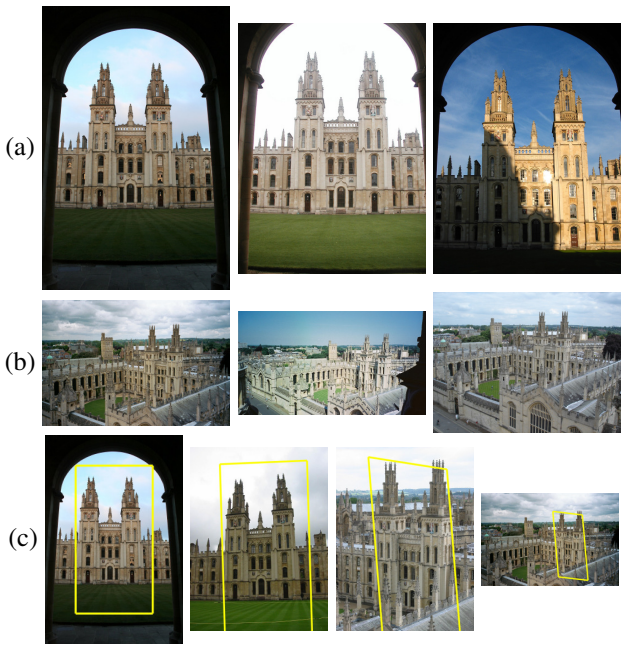


Figure 4. Joining separate clusters. (a) & (b) show examples of two correct but separate clusters found for the facade of All Soul's college, ordered by degree. By examining the shortest path, shown in (c) from the two highest degree images from both clusters and propagating the query regions, our hybrid method is able to rejoin these two clusters.

In [19] we give an alternative method for clustering on particular objects based on generative modelling including spatial layout.

5 Results and Discussion

5.1 Clustering accuracy

We evaluate the performance of our method on the publicly available Oxford Dataset [1], using the groundtruth provided for 11 different landmarks. This groundtruth labels all the images in the dataset in which each of the 11 landmark buildings appear.

To compute the performance for a groundtruth object, we find the cluster which gives the maximum average precision for that landmark. We then average these scores for each landmark to give a mean average precision (mAP) score. Results for three methods (randomly ordered connected components, degree ordered connected components and our hybrid clustering method) are shown in table 2.

Groundtruth object	(a)	(b)	(c)
all_souls	0.196	0.147	0.937
ashmolean	0.647	0.627	0.627
balliol	0.333	0.333	0.333
bodleian	0.248	0.578	0.612
christ_church	0.595	0.676	0.676
cornmarket	0.449	0.651	0.651
hertford	0.557	0.705	0.705
keble	0.773	0.937	0.937
magdalen	0.204	0.204	0.204
pitt_rivers	1.000	1.000	1.000
radcliffe_camera	0.688	0.922	0.973
Average	0.517	0.616	**0.696**

Table 2. mAP results for the 5K Oxford groundtruth and dataset. (a) is the random connected components method. (b) is the degree ordered connected components method. (c) is our hybrid clustering method.

Here we can clearly see the hybrid approach outperforming the connected components for some queries, while not impacting the performance for the others.

5.2 Dataset Examples

No groundtruth is available for the Statue of Liberty and Rome datasets, so here we show some automatically discovered object clusters. Figure 5 shows samples from the three largest clusters found for the Statue of Liberty dataset. Figure 6 shows three large clusters for the 1M image Rome dataset. Both these figures shown the extreme changes in imaging conditions which our method is able to handle.

6 Conclusion and Discussion

We have introduced a new graph-based representation for an image corpus based on the commonality of objects within the images, and have demonstrated that the graph can be constructed in a scalable manner over very large datasets. We have also demonstrated an application of the graph in automatically finding frequently occurring objects. However, we expect the graph to support many other applications, e.g. returning canonical images for a set of focal lengths of the object (close-up, medium-shot, long-shot) and viewpoints; page-rank type weightings for retrieval; and pre-processing for 3D reconstruction tasks such as Phototourism [29].

In comparison to recent work [4, 11] our approach differs in that we first find *all* potential matches (the matching graph), and then associate image sets within this graph. This is in contrast, to [4, 11], where initial clusters are obtained first at lower cost, but at the risk of missing matches. There is clearly an interesting comparison to be made on the measures of speed vs what is missed, between the method presented here and the methods of [4, 11].

Acknowledgements. We are grateful for discussions with Michael Berry, Michael Isard and Josef Sivic. Financial support was provided by EPSRC, Microsoft and the Royal Academy of Engineering.

References

[1] http://www.robots.ox.ac.uk/∼vgg/data/oxbuildings/.
[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, ISBN: 020139829, 1999.
[3] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, Jan 2003.
[4] O. Chum and J. Matas. Web scale image clustering: Large scale discovery of spatially related images. Technical Report CTU-CMP-2008-15, Czech Technical University in Prague, 2008.
[5] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In *Proc. CIVR*, 2007.
[6] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007.
[7] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.
[8] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, Jun 2005.
[9] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *Proc. CVPR*, 2006.
[10] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 43:177–196, 2001.
[11] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *Proc. ECCV*, 2008.
[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
[13] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 1(60):63–86, 2004.
[14] M. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E, 69*, 2004.
[15] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E, 69*, 2004.
[16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.
[17] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, 2006.
[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.
[19] J. Philbin, J. Sivic, and A. Zisserman. Geometric LDA: A generative model for particular object discovery. In *Proceedings of the British Machine Vision Conference*, 2008.
[20] T. Quack, V. Ferrari, and L. Van Gool. Video mining with frequent itemset configurations. In *Proc. CIVR*, 2006.
[21] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *Proc. CIVR*, 2008.
[22] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *Proc. ICCV*, pages 883–890, 2005.
[23] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. ECCV*, volume 1, pages 414–431. Springer-Verlag, 2002.
[24] C. Silpa-Anan and R. Hartley. Localization using an image-map. In *Australasian Conf. on Robotics and Automation*, 2004.
[25] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *Proc. ICCV*, 2007.
[26] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proc. ICCV*, 2005.
[27] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.
[28] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *Proc. CVPR*, Jun 2004.
[29] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *Proc. ACM SIGGRAPH*, pages 835–846, 2006.
[30] D. Sorensen. Implicitly restarted arnoldi/lanczos methods for large scale eigenvalue calculations. Technical report, 1996.
[31] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SDM*, 2005.
[32] A. Yavlinksy, D. Heesch, and S. Rueger. A large-scale system for searching and browsing images from the world wide web. In *CIVR*, 2005.

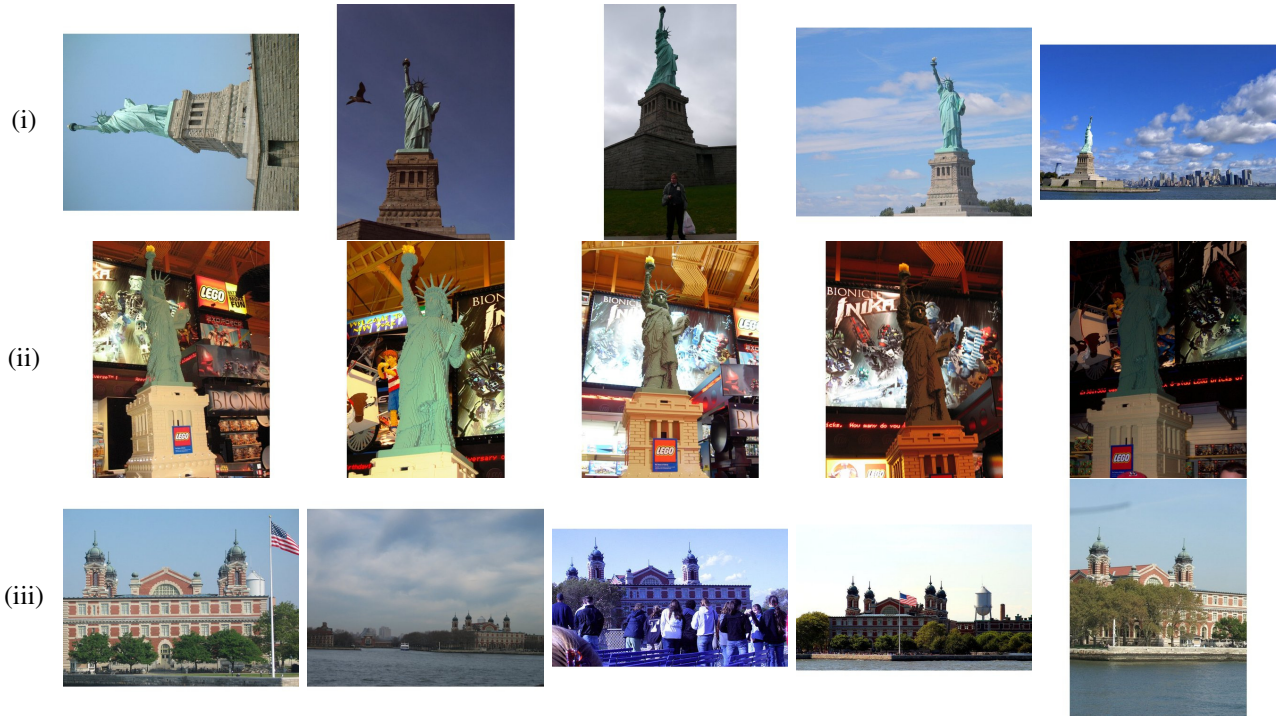


Figure 5. Random samples of the three largest clusters automatically found from the Statue of Liberty dataset with our hybrid clustering approach. Note the extreme variety of imaging conditions (changes in scale, viewpoint, lighting and occlusion) (i) – the Statue of Liberty (11170 images). (ii) – A lego Statue of Liberty (59 images). (iii) – A Staten Island building (52 images).

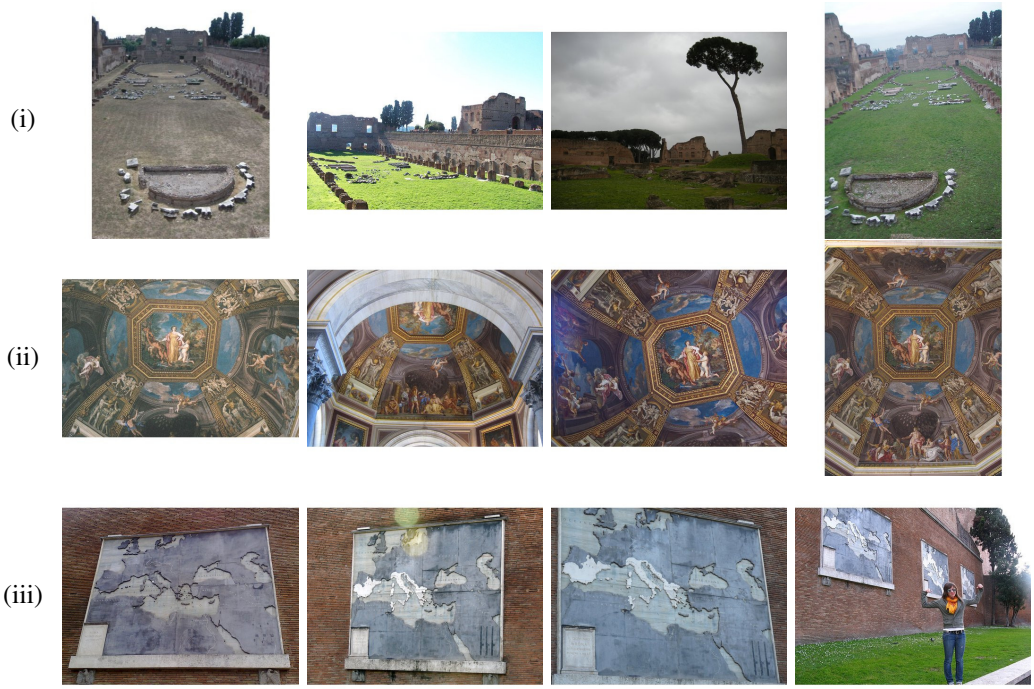


Figure 6. Three clusters discovered from the 1M+ image Rome dataset.