World-scale Mining of Objects and Events from Community Photo Collections

Till Quack[1,2]
[1] kooaba AG
Zurich, Switzerland
quack@kooaba.com

Bastian Leibe[2]
[2] ETH Zurich, BIWI
Zurich, Switzerland
leibe@vision.ee.ethz.ch

Luc Van Gool[2,3]
[3] K.U. Leuven, IBBT
Leuven, Belgium
vangool@esat.kuleuven.be

ABSTRACT

In this paper, we describe an approach for mining images of objects (such as touristic sights) from community photo collections in an unsupervised fashion. Our approach relies on retrieving geotagged photos from those web-sites using a grid of geospatial tiles. The downloaded photos are clustered into potentially interesting entities through a processing pipeline of several modalities, including visual, textual and spatial proximity. The resulting clusters are analyzed and are automatically classified into objects and events. Using mining techniques, we then find text labels for these clusters, which are used to again assign each cluster to a corresponding Wikipedia article in a fully unsupervised manner. A final verification step uses the contents (including images) from the selected Wikipedia article to verify the cluster-article assignment. We demonstrate this approach on several urban areas, densely covering an area of over 700 square kilometers and mining over 200,000 photos, making it probably the largest experiment of its kind to date.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Design, Experimentation,Theory

1. INTRODUCTION

Several recent developments have strongly influenced the state-of-the-art in retrieval from visual databases. First, more powerful (local) visual features [3, 14, 15, 16, 30] have led to significant progress in recognition capabilities, both for specific objects [27] and for object classes [7, 12, 24]. Second, while the state-of-the-art in object class recognition scales to a few thousand images, scaleable indexing methods for retrieval of specific objects have recently allowed scaling up to 1 million images [17, 19]. Third, with the ubiquitous availability of the Internet and the widespread use of digital cameras, large databases of visual data have been created, most notably community photo collections such as Flickr (`http://www.flickr.com`). These collections contain vast amounts of high-quality images, often labeled with keywords or tags. An increasing number of those photos is also annotated with the geographic location the picture was taken at. Annotating photos with their geographic position (often called "geotagging") is either done automatically with a GPS device or by manually placing the photo on a map. However, these textual and geographic annotations are still of far lower quality than their counterparts in "traditional" databases, such as stock photography or news archives.

In this work, we deal with a crucial but often neglected building block towards Internet-scale image retrieval: the automated collection of a high quality image database with correct annotations. More precisely, from the large amount of sparsely labeled content in community photo collections, the task is to mine (clusters of) images containing objects in a fully unsupervised manner. For each mined item, we automatically derive a textual description. The resulting "cleaned" image database for the mined objects and events is of far higher quality than the original data and facilitates a variety of applications. For example, the mined structure can be used for automated annotation of photos uploaded to community collections, for retrieval and browsing of landmark buildings [19], automatic 3D reconstruction of sights [31, 9], or for mobile phone tourist guide applications [18], where users can point the integrated camera to a sight and retrieve information about it.

Our approach is based on photographs which have been tagged with their geographic location. Flickr reports that over 2 million such geotagged photos are currently uploaded each month. This allows us to mine the world in a scalable manner without any prior knowledge on landmarks and their locations. To that end, we partition the world into a grid of square tiles and retrieve for each tile all the corresponding geotagged photos from Flickr. The geographic tiling thus allows us to handle the size of this vast problem and to parallelize computations.

In detail, this paper makes the following contributions. 1) We demonstrate fully automatic, world-scale image mining from community photo collections. To our knowledge, our approach is the first of its kind that can structure, interpret, and annotate such amounts of visual data without user intervention. 2) We cluster the retrieved photos according to several different modalities (including visual content and text labels) and clustering strategies. We show how the intelligent combination of the resulting cluster assignments

CIVR'08, July 7–9, 2008, Niagara Falls, Ontario, Canada.

can capture and discriminate between distinct objects, inside/outside views of landmark buildings, and panoramas, and how it can represent the neighborhood relation between those sights. 3) For each cluster, we additionally calculate a set of cues, such as the number of days covered by its photos, the number of users who took the photos, *etc*. We show how these additional features can be used to train a subsequent classifier, which determines if an image cluster represents an object or an event. 4) We apply Frequent Itemset Mining on the text associated with each cluster in order to assign cluster labels. We propose an algorithm that employs the resulting frequent itemset labels to link clusters to Wikipedia pages providing additional information about the cluster content, and that then in turn takes the Wikipedia entries to verify clusters and filter out false assignments. 5) Closing the loop, we finally demonstrate how the verified clusters can be used to automatically label and geo-locate additional photos, for which no geotags were available.

The paper is structured as follows. The next section discusses related work. Section 3 then introduces our mining approach and describes how we cluster the mined photos. Section 4 details how we classify clusters into objects and events and how we link them to Wikipedia. Finally, Section 5 presents experimental results.

2. RELATED WORK

Since the proposed method covers an entire, multi-modal processing pipeline, it touches on a large variety of previous publications. Mining objects from visual data has been proposed for video for instance in [28, 21]. Those works also built on local features, but focused on the spatial arrangement of quantized features from video data. Working with data from community photo collections has received increasing attention lately [2, 11, 13]. However, most of those approaches are based either on text [11] or only global visual features. The local visual features which are used in this work, however, allow to find very good and extremely accurate matches between the depicted objects even under significant changes in viewpoint, imaging conditions, scale, lighting, clutter, noise, and partial occlusion. A similar approach would not be possible using global measures such as color or texture histograms. Philbin and Zisserman [19] also worked with local features and multiple view geometry on a database of landmark buildings obtained from Flickr. The main goal of that work was to derive a scalable indexing method for local visual features, the database was retrieved and annotated manually. The work most similar to ours is probably [25, 29]. Here, the authors also proposed clustering images from community photo collections using multi-view geometry based matching between images. The goal was to derive canonical views for certain landmarks and to use those as entry points for browsing. Initial image collections were retrieved by querying photo collections with known keywords such as "Rome", "Pantheon", *etc*. As we will demonstrate, our fully unsupervised approach based on geographic tiling is not only more flexible, but also more scalable. (The dataset used in [25] contained 20′000 photos, while ours is one order of magnitude larger). Furthermore, we add several layers of processing which extract semantic information, such as classification into objects and events, and which automatically include other content sources such as Wikipedia for unsupervised labeling of objects. To the best of our knowledge, this work is the first to propose this kind of pipeline, taking as an input only a geographic tiling of the world and resulting in an output of automatically mined landmark objects, together with their semantics in the form of automatically created links to Wikipedia.

Figure 1: Tiles over Paris. The size of a tile is marked in red. Note the overlap of 50% (100m).

3. MINING APPROACH

In Summary, our approach consists of the following steps:

- Gathering the geotagged data from the WWW
- Clustering to group images of the same object/event
- Classification of clusters into objects or events
- Frequent itemset mining to derive cluster labels
- Unsupervised linking to Wikipedia and verification of those links

The following sections describe each of those steps in detail.

3.1 Gathering the data

To gather the raw data, we query community photo collections such as Flickr. First, we divide the earth's surface into square tiles T_k of about 200m side length. A tile center is set every 100m (in longitude and latitude direction), such that the tiles have a high overlap. For each tile, we query the Flickr API with the tile's center coordinates and bounding box to obtain all geotagged photos for that area. Figure 1 shows a section of a map with the tiles used for querying overlaid. In total, we processed about 70′000 tiles for this work, covering several European urban centers, namely Paris, Rome, Venice, Oxford, Zurich, Munich, Tallinn, Prague, and St. Petersburg. Table 1 lists the urban areas we covered and the number of tiles and photos retrieved for each area. In total, we covered an area of about 700 square kilometers. The majority of tiles (about 52′000) were empty. The remaining tiles contained on average 10 and a maximum of 3750 photos.

For each photo we download, we also obtain the associated metadata, namely the textual descriptions (tags, title, description), user-id, and timestamps.

3.2 Photo Clustering

Once the photos for each tile have been downloaded, we process each cell to find clusters of photos with similar content as object candidates. We first create dissimilarity matrices for several modalities by calculating the pairwise distances between photos for each modality. A hierarchical clustering step on the dissimilarity matrices then creates clusters of photos for the same object or event. Below we discuss the features and distances used for each modality.

Name	# tiles	#photos	area (km^2)
Munich	18'228	24'069	184.99
Oxford	2'112	7'431	22.05
Paris	12'532	87'452	127.57
Pisa	723	1'950	7.78
Prague	11'110	28'872	113.22
Rome	14'397	48'750	146.38
St. Petersburg	3'400	2'573	35.18
Tallinn	890	1'350	9.51
Venice	449	7'708	4.92
Zurich	5'663	12'602	58.15
Total	**69'504**	**222'757**	**709.74**

Table 1: Urban areas processed in this paper and the number of tiles and photos per area.

3.2.1 Visual Features and Similarity

To identify pairs of photos which contain the same object, we employ matching based on local, scale invariant features and projective geometry. We first extract the visual features from each photo. For this, we employ SURF [3] features due to their fast extraction times and compact description shown in earlier works. Each image is thus represented as a bag of 64-dimensional SURF feature vectors. For each pair of images in a tile T_k, we find matching features by calculating the nearest neighbor (NN) in Euclidean distance between all feature pairs, followed by a verification with the $2nd$ nearest neighbor criterion from [14]. Note that this linear matching procedure is fast enough, since the problem is separated into the geographic tiles. Using scaleable indexing methods such as [17, 19] could lower the processing times of the system even further.

To find object candidates from the matching features we next calculate homography mappings for each matched image pair $\{i, j\}$ [10]

$$H\mathbf{x}_n^i = \mathbf{x}_n^j \ , \quad n \in 1 \dots 4 \ , \tag{1}$$

where H is the 3×3 homography whose 8 degrees of freedom can be solved with four point correspondences $n \in 1 \dots 4$. To be robust against the aforementioned outliers, we estimate H using RANSAC [8]. The quality of several estimated models is measured by the number of inliers, where an inlier I is defined by a threshold on the residual error. The residual error for the model is determined by the distance of the true points from the points generated by the estimated H. We accept hypotheses with at least 10 inliers I as a match.

Using this kind of homography mapping works well in our case, since we have many photos taken from similar viewpoints. A fundamental matrix could handle larger viewpoint changes, but it is also more costly to compute, since it requires more inliers to find the correct model. Furthermore, mapping planar elements (such as building facades) works very well with homographies. A similar approach has also been successfully applied in [19] for a retrieval engine on a database of landmarks from Oxford handling astonishing viewpoint and scale changes. As mentioned above, the accuracy achieved with these kinds of visual features is far better than with any kind of global features, which are still often used for mining and retrieval in visual databases.

The distance matrix is built from the number of inlying feature matches I_{ij} for each image pair, normalized by the maximum number of inliers found in the whole dataset.

$$d_{ij} = \begin{cases} \frac{I_{ij}}{I_{max}} & \text{if } I_{ij} \geq 10 \\ \infty & \text{if } I_{ij} < 10 \end{cases} \tag{2}$$

In our implementation $I_{max} = 1000$, since we extract at most 1000 SURF features per image (sorted by their discriminance), *i.e.* the distance ranges in $[0.01 \dots 1]$.

3.2.2 Text Features and Similarity

Three sources for text meta-data were considered for each photo downloaded from flickr: tags, title, and description. We combine these three text fields into a single text per photo for further processing stages. The first stage consists of a stoplist. In addition to the common stopwords, this list also contains collection-specific stopwords such as years, months, and terms such as "geotagged", "trip", "vacation", "honeymoon", *etc.* Furthermore, from each photo's geotag we know its location and the corresponding place name, for instance "Rome, Italy". These location-specific place names were added to the stoplist for each photo depending on its geotag. Filtering terms with these custom stoplists turned out to be crucial to obtain good cluster labels in later processing stages.

As with the visual features, we proceed by calculating the pairwise text similarities between the documents (photos). A vector space model with term weighting of the following form is applied:

$$w_{i,j} = L_{i,j} * G_i * N_j \tag{3}$$

Note that in the standard $tf * idf$ ranking [23] $L_{i,j} = tf_{i,j}$, $G_i = \log \frac{D}{d_i}$ and $N_j = 1$, where $tf_{i,j}$ is the frequency of term i in document j, d_i is the number of documents containing term i, and D is the total number of documents. In our system, the weighting elements are as follows

$$L_{i,j} = \frac{\log(tf_{i,j}) + 1}{\sum_j \left(\log(tf_{i,j}) + 1\right)} \tag{4}$$

$$G_i = \log\left(\frac{D - d_i}{d_i}\right) \tag{5}$$

$$N_j = \frac{U_j}{1 + 0.0115 * U_j} \tag{6}$$

where U_j is the number of unique terms in document j. The rationale behind the modifications of the weighting terms over the standard $tf * idf$ are as follows. The logarithm in $L_{i,j}$ adjusts/dampens weights of multiple occurring words per document. G_i is a probabilistic inverse document frequency as proposed in [6], which, unlike *IDF*, assigns negative weights to terms that appear in more than half the documents. Finally, the additional term N_j is a pivoted unique normalization which is used to correct for discrepancies in document lengths [26]. We use the mySQL (`www.mysql.com`) full-text search, which can be configured to use the modified $tf * idf$ ranking, to compute the text disimilarity matrix for the photos belonging to each grid tile.

3.2.3 Additional Features

Besides the visual and text similarities between photos, we also considered several additional cues. We store the user data (*i.e.* which Flickr user took or uploaded a photo) and the timestamps. As we will show below, these cues allow us to classify each cluster candidate into event or object types.

3.2.4 Clustering

For each tile T_k, we apply hierarchical agglomerative clustering [32] to the distance matrix of each modality. This

	Visual	Text
Single-link	0.985	0.989
Complete-link	0.99	0.99
Average-link	0.99	0.99

Table 2: Cut-off distances for clustering

clustering approach was chosen, since it builds on a dissimilarity matrix and is not restricted to metric spaces. It is also rather flexible and very fast, once the full distance matrix is available. Using different linking criteria for cluster merging allows us to create different kinds of clusters. We employed the following well-known linkage methods

$$\text{single-link:} \quad d_{AB} = \min_{i \in A, j \in B} d_{ij} \tag{7}$$

$$\text{complete-link:} \quad d_{AB} = \max_{i \in A, j \in B} d_{ij} \tag{8}$$

$$\text{average-link:} \quad d_{AB} = \frac{1}{n_i n_j} \sum_{i \in A, j \in B} d_{ij} \tag{9}$$

where A and B are the clusters to merge, and i and j index their n_i and n_j elements, respectively.

The motivation behind these measures is to capture different kinds of visual properties that allow us to associate a semantic interpretation with the resulting clusters. Single-link clustering adds images to a cluster as long as they yield a good match to at least one cluster member. This results in elongated clusters that tend to span a certain area. As a result, if visual features are the basis for clustering, it can group panoramas of images that have been taken from the same viewpoint, or series of images around an object. In contrast, complete-link clustering enforces that a new image matches to all cluster members. This strategy will therefore result in very tight clusters that contain similar views of the same object or building. Average-link clustering, finally, takes a compromise between those two extremes and provides clusters that still prefer views of the same object, while allowing more flexibility in viewpoint shifts. In our approach we do not want to restrict ourselves to any single of those alternatives; instead, we pursue them in parallel. Such an approach makes it possible to derive additional information from a comparison of cluster outcomes. For example, we may first identify distinct objects or landmark buildings through complete- or average-link clusters and later find out which of them are located close to each other by their membership in the same single-link cluster. Table 2 summarizes the linkages and cutoff-distances used for each modality.

4. LABELING CLUSTERS

In the preceding sections, images with similar content or annotations were grouped into clusters, which ideally should depict a single entity. In this section, the goal is to look into the contents of the clusters in more detail. First, we classify the clusters into objects (landmarks etc.) and events. In a next step, we derive textual labels for the clusters from the associated metadata. Furthermore, we introduce an approach to formulate text queries from the labels, which are submitted to Wikipedia to assign articles to the clusters. A final verification step uses the images found in the Wikipedia articles to verify this assignment.

4.1 Classification into Objects and Events

To discriminate between objects and events, we rely on the collected metadata for the photos in each cluster. An

Figure 2: Class examples: object, event, none.

"object" is defined as any rigid physical item with a fixed position, including landmark buildings, statues, *etc.* As "events", we consider occasions that took place at a specific time and location, for instance concerts, parties, *etc.* Thus, we include as features the number of unique days the photos in a cluster were taken at (obtained from the photos' timestamps) and the number of different users who "contributed" photos to the cluster divided by the cluster size.

$$f_1 = |D| \tag{10}$$

$$f_2 = \frac{|U|}{|N|} \tag{11}$$

where $|D|$ is the number of days, $|U|$ the number of users, and $|N|$ the number of photos in the cluster. Typically, objects such as landmarks are photographed by many people throughout the year; an event on the other hand usually takes place only at one or two days and is covered by fewer users. Note that we only consider clusters with $N > 4$ here.

We manually labeled a ground truth of about 700 clusters with the class labels "object", "event", and "none". See Figure 2 for an example of each class. We then trained an individual ID3 decision tree [22] for the classes "object" and "event" on half of the labeled data and used the other half for validation. The task in training and testing was to discriminate the target class ("object" or "event") against all other classes. Cross-validated over 10 random data partitions, this simple classifier was able to achieve 88% precision for objects and 94% for events with a standard deviation of 0.07% and 0.04%, respectively.

4.2 Linking to Wikipedia

Having the clusters classified into objects and events, the next processing layer intends to add more descriptive labels. The goal is to not only label the clusters with the most dominant words, but automatically link them to content on the Internet, such as corresponding Wikipedia articles. Such a solution allows auto-annotation of unlabeled images, even down to outlining object-parts using the information from other pictures of the same entity. Potential applications include mobile tourist guides, where tourists use the integrated camera of their mobile phones to take a picture of a landmark building. A recognition service building upon our labeled database could then match the query to the corresponding database entry and return the assigned Wikipedia content to the user's device. Such systems have been proposed before (*e.g.* [18, 20]), but the automatic collection of the database from user-generated content has not been addressed yet.

The proposed approach first finds relevant word combinations from the text associated with each cluster using a frequent itemset mining algorithm. The resulting frequent combinations are then used to query Wikipedia in a second step. An image based matching step finally verifies that the links are indeed correct.

4.2.1 Frequent Labels

Flickr and similar community photo collections provide us with text associated to photos. However, the text is often noisy, and not all images are labeled. Furthermore, if we want to use the text to find out more about the object by querying Internet search engines, we need to create queries from the raw tags. Any combination of words from the text could be the "correct" query. However, finding and trying all possible combinations would mean considering 2^N combinations of words, where N can easily be in the hundreds. We therefore resort to frequent itemset mining to find the most frequent combinations of words. Those can serve as labels for the objects and as query input for the next stage.

We quickly summarize the concepts of itemset mining. Originally, frequent itemset mining algorithms were developed to solve problems in market basket analysis. The task consists of detecting rules in large numbers (millions) of customer transactions, where the rules describe the probability that a customer buys item(s) B, given that he has already item(s) A in his shopping basket. More precisely, as shown in [1], the problem can be formulated as follows.

Let $I = \{i_1 \dots i_p\}$ be a set of p *items*. We call a subset A of I with m items an *m-itemset*. A transaction is an itemset $T \subseteq I$ with a transaction identifier $tid(T)$. A transaction database $D = \{T_1 \dots T_n\}$ is a set of transactions with unique identifiers $tid\{T_i\}$. We say that a transaction T *supports* an itemset A, if $A \subseteq T$. We can now define the support of an itemset $A \in D$ in the transaction database D as follows:

$$supp(A) = \frac{|\{T \in D | A \subseteq T\}|}{|D|} \in [0, 1] \quad (12)$$

An itemset A is called *frequent* in D if $supp(A) \geq s_{min}$, where s_{min} is a threshold for the minimal support. Frequent itemsets are subject to the monotonicity property: all m-subsets of frequent $(m+1)$-sets are also frequent. The APriori algorithm was the first [1] to take advantage of the monotonicity property to find frequent itemsets very quickly.

In our setting, the text associated with each photo (tags, caption, titles, *etc.*) generates a transaction, and the database consists of the set of photos in a cluster. We use an implementation of the *fpgrowth* [4] algorithm to mine the frequent itemsets for each cluster, using a minimal support threshold of 0.15. In order to ensure scalability, only the top 15 itemsets per cluster are kept.

The advantage of using itemset mining over other probabilistic method is its speed and scalability. Tens of thousands of word combinations can be processed in fractions of seconds. Furthermore, mining variants such as maximal or closed frequent itemsets [1], as well as additional statistical tests [4] on the sets, offer further opportunities for optimization. For instance, maximal frequent itemsets (itemsets with no frequent superset) are especially useful for human-readable labels on clusters, since their subsets are not listed as additional labels.

4.2.2 Querying Wikipedia and Link Verification

We use each frequent itemset mined in the previous section to submit a query to an Internet search engine. More specifically, we query Google (`www.google.com`), limiting the search to wikipedia.org. By doing so, the search covers Wikipedia in all available languages, so terms in different languages can be handled automatically. For each result list, the top 8 results are kept. Note that in the worst case, this generates $15 * 8 = 120$ possible URLs per cluster. We keep a score for each page, which counts how often the same page was retrieved using different queries. Next, we open each of the URLs and parse the corresponding Wikipedia page for images. The idea is now to use the Wikipedia content to verify the proposed linking between the cluster and the Wikipedia page. Chances are high that our clusters contain some images taken from similar viewpoints as the ones used in Wikipedia. Thus, we extract features from the Wikipedia images and try to match them to all images in the cluster using the same method as described in Section 3.2.1. If we find a matching image, the proposed link is kept, otherwise it is rejected.

# Images	222'757
Size Metadata	1.1 GB
Size Features	111 GB
# Images assigned to clusters	73'236
# Similarities computed	217'330'144
# Similarities > 0	751'457

Table 3: Dataset statistics

5. RESULTS

In the following, we present results on the whole dataset collected to this date, stemming from the $70'000$ geographic tiles that were inspected by our algorithm. We first give an overview over the dataset, followed by subsections discussing the results of the individual processing layers. Table 3 summarizes the dataset statistics. In total over $220'000$ images were downloaded from Flickr, their visual features amounting to 111 GB, and their metadata (tags, geotags, EXIF data *etc.*) to 1.1GB. Over 200 million pairwise similarities had to be computed, less than 1 million was greater than zero. (Note that without the geographic tiling, we would have had to calculate over 20 billion pairwise similarities). In the end, a little over $73'000$ photos could be assigned to a cluster.

5.1 Clusters

Here, we present results for different types of clustering. We start with a specific example to give an impression of the results we found. Figure 3 shows examples from the area around the Pantheon in Rome. The corresponding tile is among those with the largest number of elements, containing $2'250$ images (several tiles overlap here; we report the numbers for the dominant one). It is well visible how the clustering splits the data into several semantically separate objects and contexts. For example, indoor (a) and frontal outdoor views (b) of the Pantheon are found as separate entities. Both contain a large number of photos: 546 and 481, respectively. Smaller clusters describe more specific elements, such as the view from the Pantheon onto the piazza (e), the obelisk situated behind the Pantheon (c), and even the tomb of Victor Emmanuel II (d) inside the Pantheon. Calculating the mean of the photo locations in each cluster allows us to place the cluster on a map. Clearly, the locations of the different clusters are estimated very close to the true positions of the corresponding entities. The clusters shown in this figure were obtained using single-link clustering. Note how especially for clusters (a),(b), and (c), this allows us to merge a wide variety of views of the same object, since only the closest matching pair has to be connected by a distance smaller than the threshold.

In total 72 clusters were found in this area, with a mean

Figure 3: Clusters found around the Pantheon and the number of photos contained in each. Note the automatic separation into indoor (a), outdoor (b), and panorama views (e), and the discovery of separate objects (c,d). Mean locations of the photos are shown on the map. (e) is estimated at about the same position as (b) and is therefore not drawn on the map.

size of 62 photos. We evaluate clustering accuracy in terms of the cluster precision, *i.e.* the number of correct images divided by the total number of images in the cluster. As "correct", we count every image which contains the object the cluster refers to. If there are special contexts, such as an indoor view for an object, only those (*e.g.* indoor views) are counted as correct. Given that definition, the mean precision of the 10 largest clusters is over 98%. Note that since we deal with an unsupervised mining problem, we cannot give reliable results for recall.

For comparison, we also ran a clustering based purely on text, using all text similarities between the photos in this area. Depending on the parameters, we were only able to get 1-3 clusters with a precision of about 60%. Not only were we not able to discriminate between indoor and outdoor views based on text features, the clusters also contained many outliers which did not contain the relevant object at all. For instance, only 116 of the photos in the area carry tags such as "inside" or "interior", making a discrimination based on text very difficult. In contrast, cluster (a) in Figure 3 contains over 500 photos of the inside of the Pantheon. (The word "Pantheon" appears with 1′245 photos). Also in comparison to [25], we are able to retrieve larger clusters while maintaining high precision.

To examine the results of the different types of visual clustering further, consider another example shown in Figure 4. It depicts the area around the Louvre in Paris. Figure 4(a) shows the estimated mean positions of single-link clusters. In total, the area is covered by 176 clusters; the largest cluster contains 418 elements, the mean size is 17 elements. One of the clusters (marked in yellow) is shown in Figure 4(b). Here, each pin represents the location of one photo. Note how strongly the positions vary. Some examples of the clusters' contents are shown in the column next to the map, again visualizing the mentioned variability in viewpoints. In contrast, Figure 4(c) shows the complete-link clusters for the same area. The more restrictive clustering criterion results in smaller and more compact clusters; the mean size is only 4 elements, and the maximum is 5. 207 complete-link clusters were found for this region; again one cluster is selected and its elements are shown in Figure 4(d). Their locations are more compact, and the contents of the cluster have less variability, as the examples next to the map demonstrate. Also note again the grid overlaid on the maps in (a) and (c), which shows the tiles we used to retrieve photos by their geotags (again, 4 cells make up a tile).

5.2 Objects and events

The classifier described in Section 4.1 allows us not only to detect objects, but in sometimes even events. Applying the ID3-tree to the entire dataset resulted in the following distribution of objects and events: of 6′511 clusters (single-link), 4′315 were classified as objects, 719 as events. Visual inspection on randomly picked clusters showed that the classification precision is very accurate, similar to the results obtained on the validation set in Section 4.1. Figure 5 shows some examples of event clusters. The first cluster contains images from 3 different events in a series taking place on different days ("Oxford Geek nights") and was recognized due to the same location it took place in. The second (a movie premiere in Italy) and third event (an exhibition in a gallery in Paris) were both covered by two photographers. The last line represents the majority of events: an event from a single day, covered by only one photographer. The smaller number of event clusters can be explained by two factors: relying mostly on visual cues, we can only detect events which take

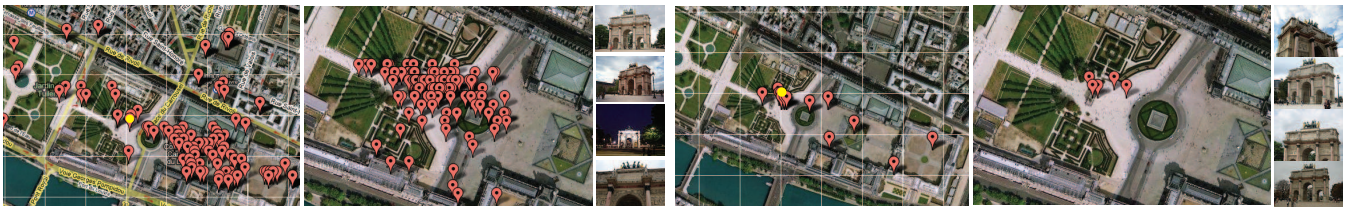

Figure 4: Clusters around the Louvre: (a) shows single-link clusters, the photos of the cluster marked in yellow are located as shown in (b). (c) shows complete-link clusters for the same area, again with the photos of the yellow cluster in (d). (Only clusters with at least 4 elements are shown).

Figure 5: Typical events mined by our methods.

place in a environment where the background matches between photos. Second, it seems that so far, in general fewer people geotag photos of events.

5.3 Linking to Wikipedia

Figure 6 visualizes the individual steps in linking clusters to Wikipedia content. The tags for the cluster (a) are mined to create frequent itemsets (b). Note how the proximity to the Louvre introduces noisy words such as "museum", and how the expression "arc du triomphe" could refer also to the other, larger Arc Du Triomphe in Paris. The frequent itemsets (b) are fed as queries to Google, and the candidate URLs (c) are retrieved. For each URL, the images contained in the page are extracted and matched back to the images in the cluster (a). Figure 6(d) shows the best match from the cluster with the image from the Wikipedia article (e). The final, selected URL is given in (f).

Figure 7 shows some typical results of this process. Each result is represented by a pair of images: the left image was extracted from Wikipedia, the one on the right is its closest match in the cluster (there are typically many more matching images in each cluster.) Below each pair, we provide the URL of the mined Wikipedia article, followed by the cluster statistics. For each cluster, we report the number of photos, the number of users who took them, and the number of different days the photos were taken at. We also report the precision, obtained again by manual inspection as described above. In general the precision is very high, ranging between 93% and 99%. Especially very well known landmarks, such as the Sacre Coeur (1), the Colosseum (4,5), or the Trevi fountain (14) are covered by a large number of photos with very few false positives. Lesser known objects, such as the Radcliffe Camera (15) have fewer images and are thus also more vulnerable to a few false positives. Staying with the Radcliffe Camera (15), note how multiple matching Wikipedia articles have been verified for the object. The same effect can be observed in example (13) or example (14), where articles in multiple languages were retrieved. Some matches are truly amazing, for instance example (5), where a painting matched to a photo of the Colosseum, or (12) and (13) with strong clutter and viewpoint change.

While most examples in Figure 7 refer to rather well known objects, some rare gems were mined, too. A few examples are shown in Figure 8. Example (1) does not only link to the article Sainte Chapelle, but also to an article about stained glass; similarly Mona Lisa (2) is linked to a specific article and a more general one about Leonardo Da Vinci. In example (3), both the context "Forum Romanum" and the specific "Temple of Vesta" could be verified. Examples of smaller, even lesser known entities are shown in (4,5,6), note the maypole on Viktualienmarkt in Munich in (6): one of the articles explains the location, the other the tradition. Destinations with fewer tourists, such as Tallinn and Zurich (7,8) tend to have less photo coverage and also less content on Wikipedia. Nevertheless, some locations could be identified by our mining pipeline (7,8). Finally, example (9) is a lucky shot, where an event could be linked to a person and verified. By coincidence Wikipedia contains an image of an event (Jules Verne Adventures Film Festival, April 2007), which is also covered on Flickr and labeled with the attending actors' name. Clearly, only larger events are covered in Wikipedia, so that the chance of detecting a correct link for any event is rather small. Furthermore, homography based matching between images is well-suited for rigid objects and scenes, but less suited for events. Future work could thus extend the system by classifying event scenes (wedding, concert, *etc.*) based on a bag-of-features approach [5] and rather label it using the textual meta-data than link it to Wikipedia.

In total, 861 unique Wikipedia articles were verified by matching their images to our clusters as described above. The precision of this assignment was about 94%, *i.e.* 94% of the articles referred to a cluster which contained images of the article's correct subject. These articles covered 423 single-link clusters. Querying Wikipedia with the queries given by the frequent itemsets had resulted in over 20′000 URLs for consideration and in more than twice as many images. This demonstrates how effective our method is in mining relevant links out of a vast amount of irrelevant data.

5.4 Auto-annotation

With the database we built in this paper, auto-annotation of unlabeled photos with their geo-location and corresponding Wikipedia article becomes feasible. A user can simply

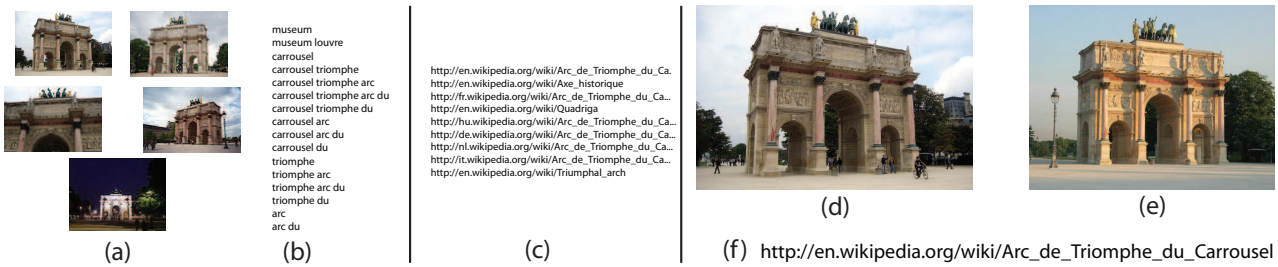

Figure 6: Matching clusters to Wikipedia articles. The text for the photos in a cluster (a) is mined for frequent word combinations (b), which are used to search Wikipedia for candidate URLs (c). Each image (d) of an article is in return matched to the images in the cluster. If a good match (e) can be found, the candidate link is selected (f).

Figure 7: A world tour with Flickr and Wikipedia. The left image in each pair stems from Wikipedia, the right image is the best match in a mined cluster. The Wikipedia links which could be verified this way are reported below the images, together with the cluster statistics. Note the high precision scores and the size of some clusters. (See text for a detailed discussion).

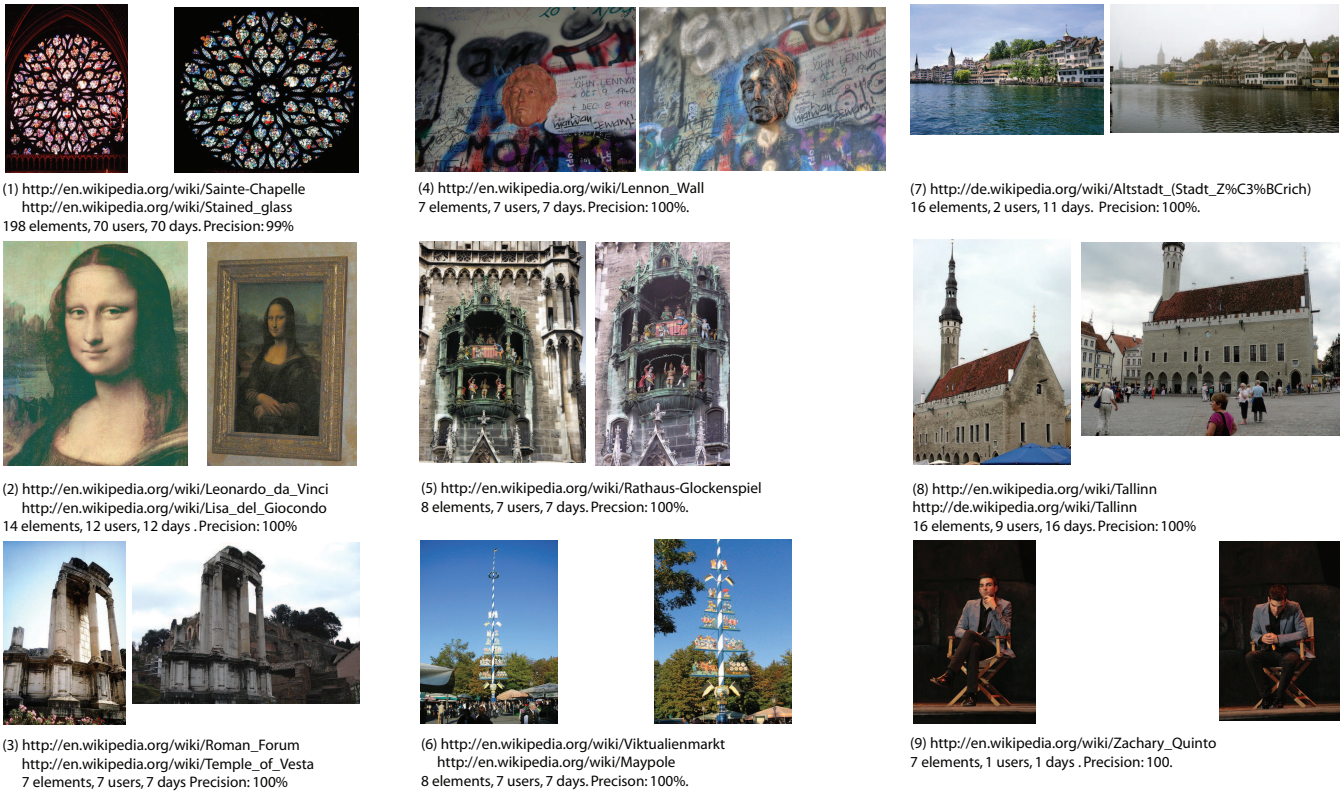

Figure 8: Additional, surprising mining results. See text for a discussion.

select the rough geographic area (*e.g.* by drawing a bounding box around Paris on the map), and the photos will be automatically placed at their exact position and linked to relevant Wikipedia articles. To demonstrate this capability, we downloaded 6 sample query images of sights in Paris from Google, see Figure 9. These are images which are neither present on Flickr, nor on Wikipedia. We load all clusters which we found in the Paris area (full area as given in Table 1) and which could be assigned to a Wikipedia article, as described in the previous steps. These conditions hold for 167 clusters. Now, we simply match the query images to the clusters and record the best-matching image and cluster. This process only takes minutes, and the result is shown in Figure 9. The result location is selected as the mean location of all images in the matching cluster. Note the precision of the placement in the magnified map elements. All images are also linked to the correct Wikipedia article in the spirit of Figures 7 and 8 (the links are not shown due to lack of space). Note how similar the Arc de Triomphe and Arc de Triomphe du Carousel are (first and second image in the left column). Also note how close the two objects Arc de Triomphe du Carousel and the Louvre Pyramid are (second and third map in the left column). Our method is able to handle these uncertainties robustly and to discriminate between similar objects at different locations and different objects at the same location. In contrast, a direct matching of query images to Wikipedia images would not be possible in most cases, since the viewpoint changes might be too large. The number of images in our clusters literallybridges the gap between the unannotated query image and the Wikipedia image via the clusters created from Flickr data. Combining this method with scalable indexing [19] for local features will allow auto-annotation of many holiday snaps within seconds.

6. CONCLUSIONS

We have presented a fully unsupervised mining pipeline for community photo collections. The sole input is a grid of tiles on a world map. The output is a database of mined objects and events, many of them labeled with an automatically created and verified link to Wikipedia. The pipeline chains processing steps of several modalities in a highly effective way. The basis is a pairwise similarity calculation with local visual features and multi-view geometry for each tile. Hierarchical clustering was demonstrated to be a very effective method to extract clusters of the same entities in different contexts (indoor, outdoor, *etc.*). We observed that the clustering step on visual data is far more reliable than on text labels. A simple tree-based classifier on the metadata of photos was introduced to discriminate between object an event clusters. Itemset mining on the text of the clusters created with visual features was proposed to mine frequent word combinations per cluster. Those were used to search Wikipedia for potentially relevant articles. The relevance was verified by matching images from the Wikipedia articles back to the mined clusters. Both the clustering and linking to Wikipedia showed high precision. Finally, in a last step we demonstrated how the database can be used to auto-annotate unlabeled images without geotags.

Besides the effective mining pipeline proposed in the paper, we also carried out one of the largest experiments with local visual features on data from community photo collections by processing over 200′000 photos. The results of this large-scale experiment are very encouraging and open a wealth of novel research opportunities.

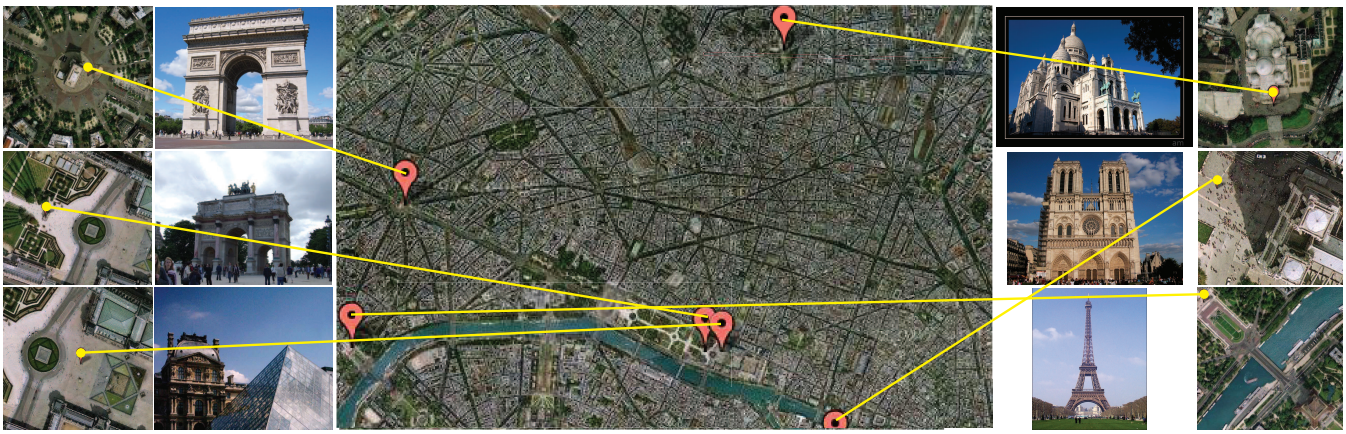

Figure 9: Auto-annotation of novel images using the mined clusters.

Acknowledgements: We acknowledge support by Swiss Project IM2, Swiss Innovation Agency CTI, Hasler Foundation and Schweizerische Volkswirtschaftsstiftung.

7. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD'93*, 1993.

[2] M. Aurnhammer, P. Hanappe, and L. Steels. Integrating collaborative tagging and emergent semantics for image retrieval. In *Collaborative Web Tagging Workshop (WWW'06)*, 2006.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV'06*, 2006.

[4] C. Borgelt. An implementation of the fp-growth algorithm. In *OSDM'05*, 2005.

[5] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification via pLSA. In *ECCV'06*, 2006.

[6] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35, 1997.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*, 2003.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Comm. of the ACM*, 1981.

[9] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz. Multi-view stereo for community photo collections. In *ICCV'07*, 2007.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2004.

[11] A. Jaffe, M. Naaman, T. Tassa, and M. Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *MIR'06*.

[12] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR'05*, 2005.

[13] S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. In *ACM Trans. Multimedia Comput. Commun. Appl.*, 2006.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.

[15] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In *BMVC'02*, 2002.

[16] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1), 2004.

[17] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR'06*, 2006.

[18] L. Paletta, G. Fritz, C. Seifert, P. Luley, and A. Almer. A mobile vision service for multimedia tourist applications in urban environments. In *IEEE Intel. Transp. Syst. Conf.*, 2006.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR'07*, 2007.

[20] T. Quack, H. Bay, and L. Van Gool. Object recognition for the internet of things. In *Internet of Things 2008*, 2008.

[21] T. Quack, V. Ferrari, and L. Van Gool. Video mining with frequent itemset configurations. In *CIVR'06*, 2006.

[22] J. Quinlan. Induction of decision trees. *Mach. Learn.*, 1:81–106, 1986.

[23] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1986.

[24] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV'07*, 2007.

[25] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *ICCV'07*, 2007.

[26] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96*, 1996.

[27] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV'03*, 2003.

[28] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *CVPR'04*, 2004.

[29] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. on Graphics*, 25(3), 2006.

[30] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC'00*, 2000.

[31] M. Vergauwen and L. Van Gool. Web-based 3d reconstruction service. *MVA*, 17(6):411–426, 2006.

[32] A. Webb. *Statistical Pattern Recognition*. Wiley, second edition, 2002.