# Image Clustering Based on a Shared Nearest Neighbors Approach for Tagged Collections

Pierre-Alain Moëllic
CEA LIST
route du Panorama
92265 Fontenay-Aux-Roses
FRANCE
pierre-alain.moellic@cea.fr

Jean-Emmanuel Haugeard
ETIS-CNRS
avenue du Ponceau
95014 Cergy
FRANCE
jean-emmanuel.haugeard@ensea.fr

Guillaume Pittel
CEA LIST
route du Panorama
92265 Fontenay-Aux-Roses
FRANCE
guillaume.pittel@cea.fr

## ABSTRACT
Browsing and finding pictures in large-scale and heterogeneous collections is an important issue, most particularly for online photo sharing applications. Since such services know a huge growing of their database, the tag-based indexing strategy and the results displayed in a traditional "in a single file" representation are not efficient to browse and query image collections. Naturally, data clustering appeared as a good solution by presenting a summarized view of an image set instead of an exhaustive but useless list of its element. We present a new method for image clustering based on a shared nearest neighbors approach that could be processed on both content-based features and textual descriptions (tags). We describe, discuss and evaluate the SNN method for image clustering and present some experimental results using the Flickr collections showing that our approach provides useful representations of an image set.

## General Terms
Algorithms, Measurement, Experimentation

## Keywords
Unsupervised data categorization, information retrieval and browsing, multimedia indexing.

## 1. INTRODUCTION
For large-scale image collection like Flickr[1] (two billions of images at the end of 2007) or other online photo sharing applications[2], efficient and user-friendly representation of image sets is an important challenge. These sets are usually the results of a query over the entire database or a part of a user collection. The indexing process of most of the available photo sharing applications is based on the keywords – tags – that people add to their images. Actually, only a small part of the Flickr collection is tagged but, when seriously done, the tags provide rich semantic information that well described a photo. Nevertheless, exploring the Flickr collection with a single or few concepts generally leads to face up with a very large amount of images. For instance, the query "Eiffel Tower" provides about 150,000 images (considering tags and free text legends). It is rather impossible to correctly show such an image set with the traditional "in a single file" representation (in our example, more than 6,000 pages with 24 pictures each) even with the proposed ranking criteria (Flickr proposes the "interestingness" and the date of publication). To handle this problem, Flickr rapidly proposed a tag-based clustering application. The clustering process does not take into account the visual content of the photos but considers the other tags that have been associated to the query tag, enhancing the three most representative tags to build the cluster. In our example, Flickr proposes two clusters relating to "Eiffel Tower": (1) Paris, France, Eiffel and (2) Las Vegas, Nevada, Casino. The number of clusters may appear relatively poor but an efficient geographic differentiation is automatically inferred by this way.

Indexing images with tags provided by users has a lot of well known advantages and drawbacks that have been widely described and commented. Among the issues, the variability of a concept ("Louvre", "Le Louvre", "Louvre Musuem" …) and the large amount of highly specific, personal and noisy tags are probably the most important ones. Linguistic processing techniques do not overcome yet all these problems that make a tag-based clustering of the images a hard task. Thus, the use of visual content appears as a complementary solution that could hardly be avoided despite additional processing times or database management issues. We also highlight the (bad) habit of many users consisting in associate tags on-the-fly to a set of images with the risk of tagging pictures with irrelevant keywords.

### 1.1 Related works
Among the available photo sharing applications, we focus this paper on Flickr that offers a very large scale collection of images and seems to be (to our knowledge) the most active platform handling with retrieval problems (more particularly and recently with the geo-referenced tags[3]).

For a lot of tags indexed in the Flickr database, Flickr proposes clusters (see Figure 1.) based on statistics computed over the tags and not taking into account the visual content of the images. For tags associated to a large amount of images, clusters provide a efficient overview of the principal tags frequently linked to the pictures. Each cluster is characterized by the three most frequent tags and when users "see more in a cluster" the application proposes a maximum of 1200 images (regularly updated) that seem to have at least two of the three characteristic tags of its cluster (according to our experiments). It is not rare to find only

[1] http://www.flickr.com

[2] Photobucket, Smugmug, Snapfish, …

[3] http://tagmaps.research.yahoo.com/worldexplorer.php

one cluster or a very small cluster set that could be explained by a strict filtering of the tags considered in the clustering process.


**Figure 1. The two clusters proposed by Flickr for the tag "Eiffel Tower".**

Data clustering techniques have been widely studied in many active research fields. [2], [6], [9] and [1] agreed to distinguish four principal approaches: hierarchical clustering, partitioning clustering, density-based clustering and grid-based partitioning. The first one is based on the representation of data in a tree-like structure built with the similarity between each pair of data. Then, cutting the tree to different levels enables to get requested number and size of clusters. The second kind of approach tries to create a partition of the data by maximizing the inter-cluster distance and minimizing the intra-cluster distance. K-Means and BSAS (Basic Sequential Algorithmic Scheme) are the most popular algorithms of this kind. The density-based approaches gather data in respect with density criteria. Thus, clusters are high density regions separated by low density ones. Note that, contrary to the previous techniques, the cluster set is usually not a partition of the data since low density regions are rarely considered in the final cluster set. Grid-based partitioning techniques quantified the data space to represent the data in a grid. Then, the clustering is processed using the connected cells of the grid and not directly the data. Different sizes and numbers of clusters can be achieved by changing the granularity of the grid.

We highlight an important difference that deals with the data coverage of these techniques, from partitioning approaches which consider all the data, i.e. potential noise included, to density-based algorithms that aim at cutting the high density area from the low density ones that are usually ignored. Actually, the strategies are clearly different and should answer different goals. In our case, dealing with large-scale image collections usually heterogeneous and noisy, we want to extract representative patterns of an image set but do not aim at showing an exhaustive representation of the diversity of the results.

## 1.2 Contributions

We propose a clustering approach based on the shared nearest neighbors algorithm (SNN) applied on both textual data (tags) and visual features. The SNN is a density-based unsupervised categorization approach that uses a shared nearest neighbors graph (SNNG) built in respect with the similarity between the data. The SNN can easily be applied on purely text information like tags or low-level visual features. To our knowledge, this approach has never been tried for image clustering. We evaluate the SNN algorithm compared to other popular clustering methods and show that this approach enables to build representative clusters. We experiment it on image sets coming from the Flickr collection and build clusters using the tags and visual descriptors (based on a bag-of-features approach). We discuss a *user-friendly* way to use these sets for image collections.

This paper is organized as follows: Section 2 contains a description and an evaluation of the clustering algorithm based on the SNN; Section 3 describes the SNN approach applied on the tags and visual content features with tests on Flickr datasets; before concluding and proposing some future works, we discuss the best ways to use our cluster sets in Section 4.

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


# 2. SHARED NEAREST NEIGHBORS APPROACH

## 2.1 Basic idea of the SNN

The Shared Nearest Neighbor is a density-based algorithm firstly developed by Jarvis and al. [10] and improved for linguistic purposes in [4], [5] and [15]. Another description of this approach can be found in [7] and [18]. As a density-based algorithm, the SNN approach clusters objects according to the detection of high density part of a similarity graph, a shared nearest neighbors graph (SNNG). The basic idea of the algorithm is to consider that the more two objects share the same neighbors, the more these objects should appear in a cluster.

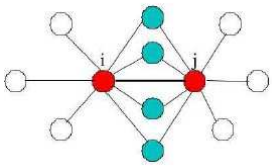


**Figure 2: Object i and j are *linked* together with four shared neighbors.**

The algorithm computes weighted links between objects depending on the number of shared neighbors and labels each object using three classes: core, noise and aggregated. We explain in details the different stages of the clustering process.

## 2.2 Clustering process

The first step of the clustering process is the building of a similarity matrix that provides the similarity between each couple of objects. To optimize the processing time, the size of the matrix is reduced by keeping the K-nearest objects for each object. Then, the SNNG is computed as follows: each object is a node of the graph and each node is linked to another by an edge if it belongs to the K most similar list of the second object. A weight is computing for the edge linking each couple of objects depending on the number of shared neighbors. The initial version of the algorithm [10] adds the rank of the neighbors to penalize shared neighbors that are too far in the neighborhood:

$$weight(i,j) = \sum_{v \in (L_i, L_j)} (k - r_{v,i} + 1) + (k - r_{v,j} + 1) \quad \text{(Eq. 1)}$$

Where $L_i$ and $L_j$ are the lists of the *k*-nearest neighbors of the objects *i* and *j*; *k* is the size of the lists; $r_{v,i}$ and $r_{v,j}$ are, respectively, the position of the object *v* in $L_i$ and $L_j$.

Here, the point is that we do not take into account the similarity between the neighbors and the objects *i* and *j*. Yet, both the rank and the similarity values are representative of the importance of a nearest neighbor. Indeed, two objects can have the same rank in neighborhood lists but with different similarities implying different weights. Thus, we add the similarity (*sim*) in Equation 1. We will evaluate in section 2.4 the two options.

$$weight(i,j) = \sum_{v \in (L_i, L_j)} (k - sim_{v,i}.r_{v,i} + 1) + (k - sim_{v,j}.r_{v,j} + 1)$$

(Eq.2)

Where $sim_{v,i}$ is the similarity between the object *i* and its neighbor *v*.

A threshold is applied to select the strongest links of the graph and the *connectivity* is defined as the number of the strong links of an object. According to the *connectivity*, two classes of objects are built: the *core* objects (object with highest density) and the *noise* objects (low density). Core objects are the germs that initialize the final clustering process: objects that are not labeled as core or noise objects are assigned to the nearest cluster and called aggregated objects. The nearest cluster is defined as the one which has the maximum number of strong links between the object and its core. Too small clusters are finally eliminated.

## 2.3 Some advantages and drawbacks of the SNN

Contrary to other clustering techniques, the SNN approach does not need to initially fix the number of cluster that is not realistic in many clustering tasks. The SNN approach is not dependant of initialization conditions contrary to other approaches like the well known K-Means that usually have to be applied several times to find a result close to the optimal solution.

As explained in [4] the SNN enables to deal with high dimensional data and to provide clusters of different size and shapes. For image sets coming from a large-scale collection like Flickr, it is important to be able to provide clusters with different sizes and complex shapes since the data could easily be heterogeneous even for a precise complex. Most of the usual content-based descriptors are high dimensional histograms requiring robust clustering techniques. These points have to be linked to the difference between density-based approach and partitioning techniques that we have already mentioned in the related work section. Since the SNN is not focused on a partition of the data but on high density and noise areas detection it provides clusters with complex shapes.

The SNN algorithm has another advantage since each cluster is built with two kinds of objects: the core objects and the aggregated objects. Indeed, each cluster is easily characterized by its high density core, facilitating the representation of the cluster set. For instance, a user interface could only show the highest density object(s) of each core and thus propose a summarized but efficient visualization of the clusters even for large-scale dataset. Then, the user can select one of these representative images to navigate into a cluster and browse both core and aggregated objects.

The main drawback is that the algorithm is highly parametrizable. Principal parameters are focused on the strong link definition, the classification of the core and noise objects and the definition of minimal cluster size. We fix some of the parameters to values representing a good compromise, for example 30% of the objects having lowest values are labeled as noise objects. We automatically fix the minimum cluster size in respect with the size of the database. Thus, we reduce the degrees of freedom of the system to two parameters that we call *Focus* and *Coverage*. The *Focus* parameter influences the size of the clusters; the more the *Focus* value is low the more the clusters are precise (focused). The *Coverage* parameter is a float value from 0 to 1. A value of 0.2 means that 20% of the most pertinent objects are used to create the clusters; the *Coverage* parameter will influence the number of clusters. These two parameters are well understandable so that it is easy to change the shape of the clustering result using the *Focus* and *Coverage*.

## 2.4 Evaluation

We evaluate the performance of the SNN algorithm on subset of the Corel database [19] composed of 1,000 images classified in ten well separated categories: (1) African people and villages, (2) beach, (3) buildings, (4) buses, (5) dinosaurs, (6) elephants, (7) flowers, (8) horses, (9) mountains and glaciers, (10) food. Figure 3 shows a representative image of each category. Even if the images of the Corel database do not represent the difficulties that could be found within internet corpuses, the interest of such a database is to enable a technological evaluation with well-defined categories.

**Figure 3. Sample images of the ten classes of the Corel1000 database**

Different metrics and protocols are regularly proposed for clustering evaluation from traditional recall and precision, purity (entropy-based) and mutual information metrics. Here, we evaluate the SNN using simple recall and precision metrics that we compute on each cluster and on the global cluster set. We define an intra-cluster precision (*iP(c)*) and an intra-cluster recall (*iR(c)*) for each cluster *c*, a global precision (*P*) and a global recall (*R*). Note that the *iP* measure is also a good evaluation of the *purity* of a cluster.

$$iP(c) = \frac{A(c)}{\#(c)} \quad iR(c) = \frac{A(c)}{GT(Class_c)}$$

$$P = \frac{\sum_c A(c)}{\sum_c Card(c)} \quad R = \frac{\sum_c A(c)}{\sum_c GT(Class_c)} \qquad \text{(Eq.3)}$$

Where, $Class_c$ is the principal class represented in the cluster *c*; *GT(k)* is the number of images of the class *k* (i.e. 100 for Corel1000); *A(c)* is the number of images of cluster *c* correctly

classified (i.e. belonging to $Class_c$) and *Card(c)* is the number of images in cluster *c*. *iP* and *iR* are respectively the mean intra-precision and mean intra-recall over the clusters.

We use only one visual descriptor (bags-of-features with SIFT descriptors, described in section 3) and compare the SNN against two algorithms: K-Means and BSAS algorithm. The BSAS approach is a classical partitioning technique, relatively close to the K-Means but with simpler association criteria that make it faster [17]. For the K-Means and the BSAS methods, we perform ten randomized initialization and use different *K* values. We finally keep the best performance. We also provide the best result for the SNN algorithm after testing different (Focus, Coverage) combinations.

The results presented in Table 1 show that the SNN outperforms the K-Means and BSAS. Note that we also indicate the ratio (*r*) between the number of clusters and the number of classes (10 classes for Corel1000).

**Table 1: Results of the SNN approach compared to K-Means and BSAS algorithms on the Corel1000 database. F is the F-measure and r the ratio of the numbers of clusters**

| | iP | iR | F | P | R | r |
|---|---|---|---|---|---|---|
| K-Means | 0.647 | 0.162 | 0.258 | 0.622 | 0.622 | 4.0 |
| BSAS | 0.605 | 0.125 | 0.202 | 0.577 | 0.577 | 4.0 |
| SNN | **0.8251** | **0.372** | **0.5128** | **0.8397** | **0.658** | **1.8** |

Table 2 presents the gain of precision when adding the similarity in the weights formula (see section 2.2).

**Table 2: Difference of precision according to the weight computing (Eq. 1 and 2.)**

| | iP | P |
|---|---|---|
| Simple weight | 0.8167 | 0.808 |
| With similarity | **0.8251** | **0.8497** |

Note that the ratio for the SNN result is naturally better since the algorithm does not produce a partition but keep the most relevant data (according to density criteria) for the clustering and thus avoid over-clustering. K-Means and BSAS clusterized the 1000 images and the SNN kept 816 images.

# 3. SNN CLUSTERING FOR TAGGED IMAGES COLLECTION

## 3.1 Test databases

### *3.1.1 Building the databases using Flickr*

We use the Flickr collection to build three test databases: a famous location (the Eiffel Tower), a personality (Roger Federer) and a general event (Presidential). Each database has been built using a target tag ("Eiffel Tower", "Federer" and "Presidential") and images and XML descriptions (containing the tags and other information for each picture) have been gathered using the Flickr API[4]. Due to classical database restriction, it is not directly possible to collect a whole data set when the target tag represents too many data. Usually, additional tags enable to collect inaccessible pictures and then gather the whole set. In our case, that could bias our results since additional tags should influence the creation of clusters. Thus, we only used the target tag as main and unique query for the Flickr API and applied the different available ranking criteria (interestingness, date of publication, etc.) and tries to pick pictures from as many different users as possible. We collected the images and tags at the end of 2007.

Here, we provide some statistics about the experimental sets:

- For the Eiffel Tower, we picked 24454 images, representing an amount of 8377 different tags. At the end of 2007, the set represented more than one third of the global set (about 55,000 of tagged images).
- For Federer, we gathered 3599 images of the famous tennis player. The set represents 1134 different tags. That represented almost the whole available collection.
- For Presidential, 8077 images were gathered representing an amount of 5181 different tags. The set represented about two-third of the whole available set on Flickr.

For each target tag, Flickr proposes the following clusters:

Eiffel Tower:

(1) **paris**, **france**, **eiffel**, tower, toureiffel, night, europe, travel, architecture, seine

(2) **lasvegas**, **nevada**, **casino**, hotel, vegas

Federer

(1) **tennis**, **roger**, **rogerfederer**, wimbledon, atp, usopen, nadal, open, grandslam, agassi

Presidential

(1) **president**, **election**, **library**, politics, clinton, campaign, arkansas, littlerock, french, candidate

(2) **france**, **sarkozy**, **paris**, elections, royal, sarko, presidentielles, bayrou

(3) **obama**, **barack**, **democrat**, barackobama, senator, speech

(4) **usa**, **washington**, **dc**, America

### *3.1.2 Formalism and notations*

For each experimental set, we define the following entities:

- $S = \{I_i\}_{i \in 1...N}$ set of $N$ images gathered with the target concept
- $W = \{w_j\}_{j \in 1...M}$ set of the $M$ tags represented in $S$

Each image $I_i$ is described by two kinds of descriptors:

- $\{w_k^i\}_{i \in 1...M^i}$ a list of $M^i$ tags from $W$
- $v_i$ the visual features of the image. In our case, $v_i$ is a histogram (bag-of-features descriptors)

We applied the SNN algorithm on both descriptors and we note $C_{tag} = \{c_i^{tag}\}_{i \in 1...K}$ and $C_{visual} = \{c_i^{visual}\}_{i \in 1...K}$ the cluster sets.

[4] http://www.flickr.com/services/api

## 3.2 Tag-based clustering

### *3.2.1 Tags filtering*

We have already seen the main issues of using user tags in online sharing photo applications (variability of the concept representation, high subjectivity, noise). These drawbacks make inevitable the use of filtering processes. First, we do not use the Flickr *raw tags* but the *machine tag* (lower case without blank) and we eliminate all the tags that are not representative enough, i.e. that have a document frequency below a fixed threshold. Among the tags that have been kept, we naturally find the target tag with a document frequency of 1. We eliminate it and all the tags that are too close to it (i.e. tags that have a too important string intersection). We also eliminate tags mainly composed of figures (date, zip code, type of camera, etc.), the tags "photo", "photography" and some camera trademarks ("nikon", "canon", …). For the Presidential image set, only 266 over the 5181 tags (5.1 %) were kept for the clustering process using a document frequency ratio of 0.01. With the same threshold, we kept 123 among the 1134 tags for Federer (10.8 %) and only 73 over the 8377 tags for the Eiffel Tower showing how this set is noisy and composed of various forms for the same concept. We discuss how we want to improve this filtering and tags selection in the future work section.

### *3.2.2 SNN application*

To apply the SNN algorithm to the tags, we first have to define a similarity measure between two tags. We use the Pointwise Mutual Information (PMI) that measures the mutual dependence of two words ($w_i$, $w_j$) in $W$:

$$PMI(w_i, w_j) = \log\left(\frac{P(w_i, w_j)}{P(w_i).P(w_j)}\right) \quad \text{(Eq. 4)}$$

Where $P(w_i,w_j)$ is the probability to find $w_i$ and $w_j$ in the same picture and $P(w_i)$ the probability to find the tag $w_i$ in a picture. These probabilities are evaluated to simple appearance frequencies of tag and pair of tags over $S$. We build our similarity matrix and the SNNG according to the pointwise mutual information.

For each cluster, tags are ranked in respect with the document frequency (*df*).

### *3.2.3 Populating the clusters with images*

Dealing with picture sets, we need to populate the tag-based clusters with representative pictures. For each picture $I$ from a set $S$ we computed a weight $\alpha(I, c_i^{tag})$ as follows:

$$\alpha(I, c_i^{tag}) = \frac{card(\{w_k^I\} \cap c_i^{tag})}{card(c_i^{tag})} + df \quad \text{(Eq. 5)}$$

The first term is the (normalized) number of tags from $I$ that are in $c_i^{tag}$ normalized by the number of tags of $c_i^{tag}$. The second term (*df*) is the mean document frequency of these tags (here a document is an image). Then, a picture is assigned to the cluster $c_i^{tag}$ with the maximum $\alpha(I, c_i^{tag})$ value. At the end of the populating process, in each cluster, the pictures are ranked in respect with the $\alpha$ value. To represent a cluster to a user, added to the list of core tags, we pick a maximum of ten images with the highest $\alpha$ and taken by different photographers (a user id being available in the XML information file provided by the Flickr API).

### *3.2.4 Experimental results*

We present below the tag-based clusters for our three sets with the representative pictures. We highlight the five most important tags (in respect with the *df*). See Table 3 for the size of each cluster. For evident setting restrictions, we only propose five representative pictures.

Eiffel Tower, six clusters:

(1): **paris notredame arcdetriomphe sacrecoeur lesinvalides** rugbyworldcup montmarte latinquarter

(2): **damedefer capitale fer monumentdeparis acier** gold lumiere illumination parisnuit

(3): **europe vacation city trip church** sculpture cathedral

(4): **lasvegas nevada casino vegas hotel**

(5): **bw blackandwhite sunset sky**

(6): **architecture lights french**

Federer, five clusters:

(1): **atp rogerscup toronto mastersseries menstennis** richard gasquet richardgasquet canadianopen rogerfederer

(2): **open us nalbandian garros final** tenis paris roland ingiro agassi rolandgarros lubijic roddick master grandslam nadal france

(3) **wimbledon london champion safin city** life south Williams

(4) **tennis sampras petesampras clashoftimes malaysia** shahalam malawatistadium

(5) **masters cup shanghai**

Presidential, five clusters:

(1) **elections france paris sarkozy photojournalism** french presidentielles protest royal bastille sarko politique bayrou

(2) **campaign obama newhampshire presidentialprimary obama08** michelleobama oprahwinfrey oprah barrackobama barrack

(3) **usa speech vote john edwards** johnedwards political forum american health care democrats action


(4) **primary boston massachusetts mitt romney** governor mittromney carolina


(5) **clinton candidates hillary mccain hillaryclinton**


**Figure 4. Tag-based clusters for the three sets with five representative images**

The clusters are as good and even more focused as the ones proposed by Flickr even if the comparison must be done with caution since our data sets are samples from the whole collection. We generally find the tags presented in the Flickr clusters, like the same Las Vegas cluster for the Eiffel Tower set. We extract several clusters for the Federer set that quite well separate famous tennis tournaments. For the Eiffel Tower set, we keep the Paris and Las Vegas distinction but propose more clusters for the famous Parisian monument. Some clusters gather other monuments of Paris and are composed of pictures that are not always showing the Eiffel Tower. That is mainly explained by the presence of overviews taken from the top of the Eiffel Tower (and, indeed, showing other monuments) and by the habit of users that associate tags for a whole set of images, as we already mentioned it. The Eiffel Tower is a critical example with a user proposing about 1000 images all tagged with the same set of tags whatever the pictures depict. Note also the presence of a tag linked to the Rugby World Cup held in France in autumn 2007. The presence of this tag is mainly explained by the fact that we downloaded data from Flickr during this period (November 2007). More generally, it could be interesting to study the temporal evolution of the clusters; the most spectacular evolution would probably be seen for the Presidential set because of the expectable amount of pictures dealing with the US primary and presidential campaign in 2008.

## 3.3 Content-based clustering

### *3.3.1 The bag-of-features descriptor*

The transposition of the "bag of words", a very popular method in linguistic processing, to the "bag of visual word" or "bag of features" for images bring to the image processing community new and powerful methods to build robust description of images. [3], [11], [16] proposed very interesting analysis and uses of this method for content-based image retrieval, objects or scenes classification purposes.

The basic idea of the bag of visual words is to produce a visual vocabulary built after a clustering process applied on a set of patches extracted from images and described with classical features such as SIFT descriptors [3]. The extraction of the patches can be done using a random selection of the pixels or using interest points detectors (Harris Laplace, Gaussian based detectors, …). Usually, the amount of patches is important and the clustering process can rapidly become a real issue. K-Means (or some "enhanced" derivates) is generally applied several times with different initializations to reach or tend to reach an optimal partition. After this quantification step, we have at our disposal a codebook composed of "visual words" enabling to represent an image with the histogram of the occurrence of each "visual word" of this vocabulary.

Two different solutions can be considered to build our codebook. First, we can compute a specific vocabulary for each concept. A second solution is to build a general visual vocabulary that could be used for every set of images. We think that the second solution is the most realistic with respect to processing times since the construction of codebook using a very large amount of high dimensional data with classical K-Means is a real computing issue. In both solutions, we have to select carefully the images that will be used for the codebook creation and try to pick images that well represents the diversity of the image collection. Thus, we use about 10,000 images from Flickr gathered using popular type of pictures: portrait, family reunion (birthday, wedding), natural landscapes (mountain, sea, and beach), monument view, urban

landscapes, vegetation and animals. For each image we extract a maximum of 1000 Harris keypoints with the rotation and scale invariant SIFT descriptors [13] and produce a 5,000 size codebook with K-Means. To overcome the initialization dependency of the K-Means algorithm we computed ten K-Means with random initializations and kept the best result (using the optimal intra-cluster distance). Hörster and al. [8] propose a different method based on the merging on multiple K-Means results computed on different subsets of a large-scale collection. In our case, using comfortable calculation capacities and a parallel implementation of the K-Means, based on [12], we could afford to apply the K-Means on the whole dataset.

Note that building the codebook with the presented SNN approach is not the purpose of this article but will be proposed in a future and parallel work. Actually, the problem of the creation of the codebook is a part of a wider recent effort of research about bag-of-features methods gathering issues such as the optimal size of the codebook or the optimal matching process between the visual vocabulary words and the patches of an image.

### *3.3.2 Applying the SNN method*

To compute the clusters, we first define a similarity matrix providing the similarity between each pair of images. As in text mining for the traditional bags-of-words, we compute this similarity using the Cosine distance:

$$d(I_i, I_j) = \frac{v_i . v_j}{\|v_i\| . \|v_j\|} \quad \text{(Eq. 6)}$$

Where $v_i$ and $v_j$ are the 5000-bins histogram of the image $I_i$ and $I_j$. We experiment some tests with different Focus and Coverage values (close to the ones find for the Corel1000 evaluation) and use the best parameters for our three experimental testbeds.

### *3.3.3 Experimental results*

Naturally, the content-based clustering produces more clusters than the tag-based clustering. The first reason is that we consider the whole set and do not filter images as for the tags. Secondly, the visual descriptors, the 5000-bins histograms, characterized high focused patterns (different spatial or scale representations of an object or a scene) and then produce more high density areas in the SNNG. Table 3 presents the number of tag-based and content-based clusters for our three datasets. As a future work we will try and combine other descriptors to reduce the number of clusters. However, the number of clusters is not a real problem since each cluster is dense enough to be represented by few images, contrary to the tag-based clusters that are more heterogeneous. Then, it is possible to present the clusters to a user by displaying a representative picture of each cluster, for example in a simple mosaic style. We detail some core images and the first aggregated images for clusters of the three experimental sets in figure 5.

**Table 3: Number of clusters (NC) and number of images (NI) for the tag-based clustering and the content-based clustering**

| | Eiffel Tower | | Federer | | Presidential | |
|---|---|---|---|---|---|---|
| | NC | NI | NC | NI | NC | NI |
| Tag | 6 | 2547 | 5 | 1357 | 5 | 2439 |
| Visual | 103 | 6812 | 31 | 3066 | 42 | 6879 |

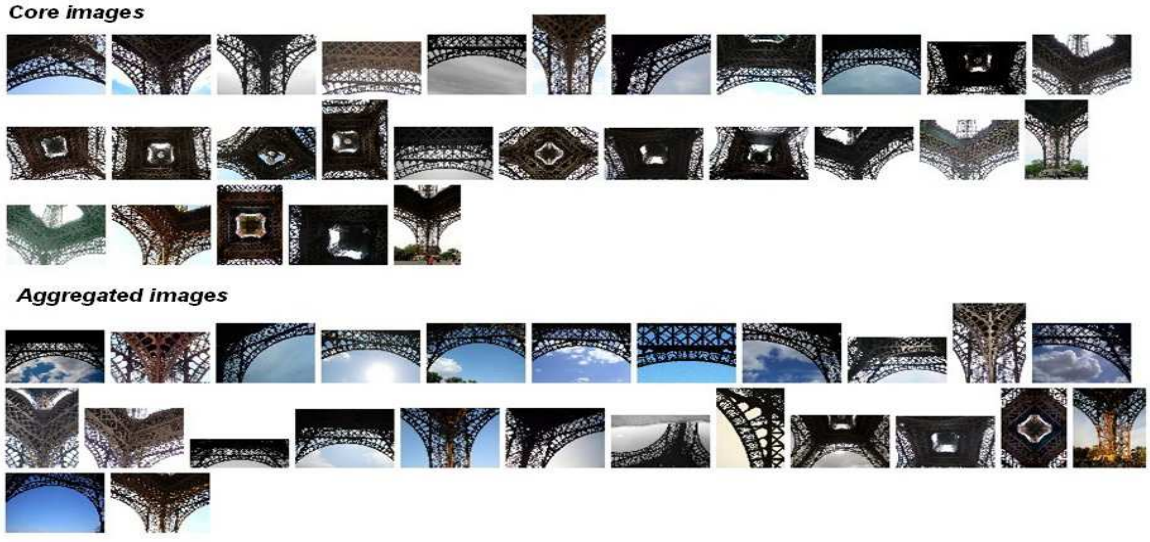


**Figure 5. Top to down: the core images and the first aggregated images of a cluster from the Eiffel Tower set; the core images of a cluster from the Presidential set; the core images and some aggregated images for two clusters of the Federer set**

## 4. USING THE CLUSTER SETS

In this section we discuss different ways to use the clusters. Indeed, the clustering process applied on textual information and content-based features produce clusters of unlike sizes and shapes with strong difference of granularity and then must be cautiously combined and used to provide an understandable result to a user.

### 4.1 Keep the cluster sets separate

After applying the SNN on both tags and visual descriptors we produce two sets of clusters. Then, an image could be an element of a tag-based cluster $c^{tag}$ and a content-based cluster $c^{visual}$. For each picture of the set, we assign a cluster couple thanks to an assignation function *F*:

$$F : I \rightarrow \left(c_I^{tag}, c_I^{tag}\right)$$

Note that $c^{tag}$ or $c^{visual}$ could be null since the SNN does not produce a partition.

By displaying the two cluster sets with their representative form, it is easy to browse a cluster and switch from a cluster set to the other. Let's consider a user selecting the tag-based clusters and browse a cluster of his interest. Then, because some images of this cluster are also linked to clusters of the other set thanks to the function *F*, the user can switch to a content-based cluster and so on. Actually, this solution can be seen as a way, for a user, to travel along a conceptual or a visual representation of an image set and cross the semantic gap when necessary. Keeping separate the two cluster sets answers two complementary and not antagonist ways of querying and browsing image collection: (1) users who do not have any specific visual representation of their query and *need* semantic or conceptual propositions (tags) to help focusing or enlarging their research; (2) users that have possible "mental" visualizations of their query and look for a set of visual representations of the results. The complementarities of conceptual information available with keywords or free texts and content-based features enable the users to improve both speed and precision of the retrieval process by switching from one querying and browsing approach to the other.

## 4.2 A hierarchical approach

Because tag-based clusters are focused on high-level information, they are usually a preferred entry point for browsing an image set. For concepts gathering huge image sets, the tag-based clusters may have a too important size. This approach proposes to start with the tag-based clusters and apply the content-based clustering on each cluster to extract representative visual patterns. We illustrate that point with the Figure 6 showing the hierarchical approach for the Federer set.

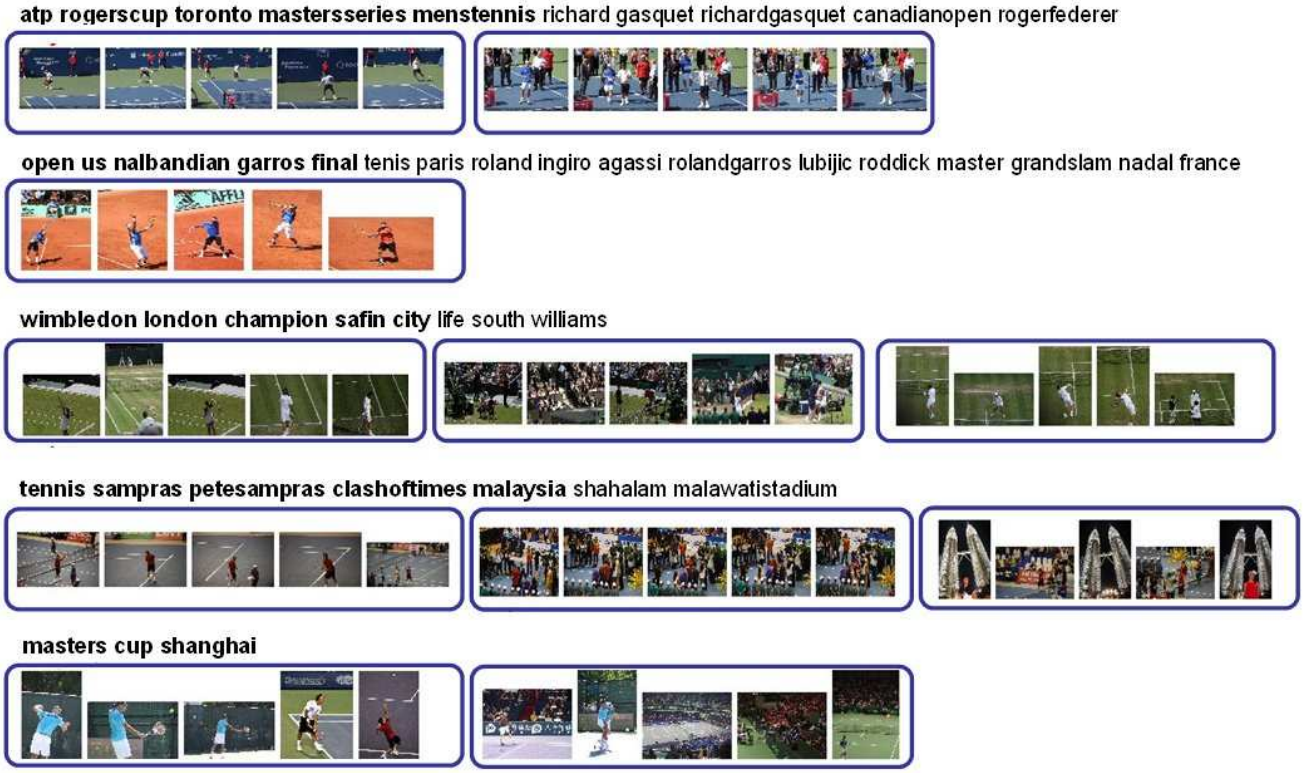


**Figure 6. Hierarchical approach for the Federer set. Each tag-based cluster is clustered on its turn according content-based features.**

## 4.3 Combining clusters

Dealing with images and textual information, it is also possible to apply the clustering on a unique feature space, for instance, by combining the similarity matrix. As a result of this early fusion, we should provide a unique cluster set of images that would take into account conceptual and visual information. We experiment this approach on the Eiffel Tower set. To combine the matrix, we first have to transform the similarity matrix we built for the tag-based clustering since we processed tags and not the images that were added later in a populating process of the clusters. We use the weight $\alpha$ detailed in the section 2 and define the similarity between two images $I_i$ and $I_j$ as follows:

- $sim(I_i, I_j) = 1$ if $I_i$ or $I_j$ does not belong to any cluster or if they belong to different clusters
- $sim(I_i, I_j) = |\ \alpha(I_i) - \alpha(I_j)\ |$ if they belong to the same cluster

Then, we simply add this similarity matrix with the one built with the content-based descriptor and the cosine distance and apply the SNN algorithm.

We find five clusters gathering almost 90% of the dataset. Actually, these clusters do not bring any benefit compare to the previous cluster sets we presented and lack of homogeneity and relevance. This is explained by the fact that combining multi-modal information requires finding a common and relevant feature space (for instance with information theory based criteria

as explained in [14]) to efficiently exploit the richness of each modality which is not the case by simply added our two similarity matrix. However, combining the two cluster sets in unique clustering raises another important problem linked to the way a cluster is easily understood and thus exploited by a user. When combining the two clusterings we lost the particularities of the visual and the textual information and produce clusters that are less defined and hardly understandable by a user. Nevertheless, the more we'll summarize an image set and extract representative and understandable patterns the more we will handle large-scale collections, thus a multi-modal fusion strategy for tagged images clustering is an effort we will focus on in future work.

## 5. CONCLUSION AND FUTURE WORKS

This work studies an image clustering process based on a shared nearest neighbors approach enabling to handle clusters of different sizes and shapes. We apply the SNN approach on descriptors of different nature, using textual information and visual features based on bags-of-SIFT descriptors. We evaluate the SNN approach for the content-based clustering against two classical data clustering techniques and show that our method performs well producing robust and representative clusters. We experiment our approach for tagged collection using three image sets coming from the Flickr collection and build cluster sets based on the tags and the content-based descriptors. Then, we discuss how to efficiently use these clusters with three approaches combining the clusters. These first results are very promising and make us continue this effort for such challenging and popular collections. In parallel to the multi-modal fusion strategy combining textual and visual descriptors, our future works will focus on (1) the realization of a technical and a user-based evaluations of our approach based on more Flickr datasets to measure the purity of the clusters and the benefit for users when browsing such tagged collections; (2) an improvement of the tags filtering, using linguistic processes and resources to merge the different tags of the same concept and then enhance the homogeneity of the tag-based clusters; (3) an improvement of the content-based clustering by adding complementary features like color-based descriptors.

## 6. ACKNOWLEDGMENTS

This work is sponsored by the French research project InfoM@gic.

## 7. REFERENCES

[1] Batistakis, Y. Halkidi, M. Vazirgiannis, M. On clustering validation techniques. Journal of Intelligent Information Systems, 107-120, 2001

[2] Berkhin, P. Survey of clustering data mining techniques, Technical report, Accrue Software, San Jose, CA, 2002

[3] Csurka, G., Dance C.R., Fan, L., Willamowski, J., Bray C., Visual categorization with bags of keypoints. In Proc. of ECCV Workshop on Statistical Learning in Computer Vision, pages 1-22, 2004

[4] Ertoz, L., Steinback, M., Kumar, V. Finding clusters of different size, shapes and densities in noisy, high dimensional data, SIAM International Conference on Data Mining (SDM '03). 2003

[5] Ertoz, L., Steinback, M., Kumar, V. Finding topics in documents, a shared nearest neighbors approach. Clustering and Information Retrieval, Kluwer Academic Publishers. 2002

[6] Hegland, M. Data mining, challenges, models, methods and algorithms

[7] Hofman, I., Jarvis, R., Robust and efficient cluster analysis using a shared nearest neighbors approach, In Proc. 14th International Conference on Pattern Recognition, Washington D.C., 1998

[8] Hörster E., Lienhart R., Slaney M. Image Retrieval on Large-Scale Image Databases. In Proc ACM International Conference on Image and Video Retrieval (CIVR '07). 17-24, Amsterdam, Netherlands, 2007.

[9] Jain, A.K., Murty, M.N., Flynn, P.J. Data clustering : a review, ACM Computing Surveys, Vol.31, 265-322, 1999

[10] Jarvis, R.A., Patrick, E.A., Clustering using a similarity measure based on shared nearest neighbors, IEEE Transaction on computers, C-22(11), 1025-1034, 1973

[11] Lazebnik S., Schmid C., Ponce J., Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, IEEE Conference on Computer Vision and Pattern Recognition, CVPR 06 - Volume 2, 2169-2178, 2006

[12] Liao, W.K., A parallel K-Means data clustering: www.ece.northwestern.edu/~wkliao/Kmeans/index.html

[13] Lowe, D. Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

[14] Magalhaes, J., Rüger, S. Information-theoretic semantic multimedia indexing, ACM International Conference on Image and Video Retrieval (CIVR '07). Amsterdam, 2007

[15] Mathieu, B., Besancon, R., Fluhr, C. Multilingual document clusters discovery, Recherche d'Information Assistée par Ordinateur, RIAO'2004, 1-10. Avignon, France, 2004

[16] Sivic, J. Efficient visual search of images and videos, PhD thesis (2006), University of Oxford

[17] Theodoridis, S., Koutroumbas, S. Pattern Recognition. Elsevier Academic Press, second edition, 2003.

[18] Wang, H.B., Yu, Y.Q., Zhou, D.R., Meng B. Fuzzy nearest neighbor clustering of high-dimensional data, International Conference on Machine Learning and Cybernetics, 2003 , Vol.4, 2569-2572. Nov. 2003

[19] Wang, J.Z., Li J., Wiederhold, G., SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 23, No.9, 947-963, 2001