



**HAL**  
open science

# Incremental hierarchical indexing and visualisation of large image collections

Frédéric Rayar, Sabine Barrat, Fatma Bouali, Gilles Venturini

► **To cite this version:**

Frédéric Rayar, Sabine Barrat, Fatma Bouali, Gilles Venturini. Incremental hierarchical indexing and visualisation of large image collections. 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Apr 2016, Bruges, Belgium. . hal-01315650

**HAL Id: hal-01315650**

**<https://hal.science/hal-01315650>**

Submitted on 2 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Incremental hierarchical indexing and visualisation of large image collections

Frédéric Rayar<sup>1</sup>, Sabine Barrat<sup>1</sup>, Fatma Bouali<sup>1,2</sup> and Gilles Venturini<sup>1</sup>

1- Université François-Rabelais de Tours, LI EA 6300,  
64, avenue Jean Portalis, 37200 Tours, France

2- Université de Lille 2, IUT, Dpt STID  
25-27, rue du Maréchal Foch, 59100 Roubaix, france

**Abstract.** Ever-growing image collections are common in several fields such as health, digital humanities or social networks. Nowadays, there is a lack of visualisation tools to browse such large image collection. In this work, the incremental indexing and the visualisation of large image collections is done jointly. The BIRCH algorithm is improved to incrementally yield a hierarchical indexing structure. A custom web platform is presented to visualise the structure that is built. The proposed method is tested with two large image collections, up to one million images.

## 1 Introduction

The last decade has witnessed the cost reduction of capturing devices such as cameras and scanners, but also storage media. This has resulted in a sharp increase of the amount of captured images. These images can be generated in a private setting, a commercial framework or in the context of digitisation projects. Moreover, the advent of the Internet has emphasised the fact that the number of images that are uploaded online grows exponentially, especially with social networks [1]. Thus, to address these growing image collections, paradigms to visualise them are needed.

Several studies have been proposed to visualise image collection [2]. Among the proposed paradigms, the graph-based approach is a straightforward solution that can be leveraged to build ergonomic and intuitive visualisation interfaces. However, some limitations appear regarding the scalability, in terms of nodes and edges number. First, the ones related to the visualisation itself: *hairball* [3] phenomenon, nodes overlapping or edges drawing. Second the ones related to the interaction: long response time, difficulty to display nodes as images, *etc.*

In this paper, we study the indexing and the visualisation jointly for browsing large image collections. We leverage a *graph* and *clustering* hybrid approach. To do so, the BIRCH [5] algorithm is improved to incrementally yield a hierarchical structure where a proximity graph organise elements at each level. This structure allows a smooth visualisation in a web platform, while highlighting the topology of the image collection.

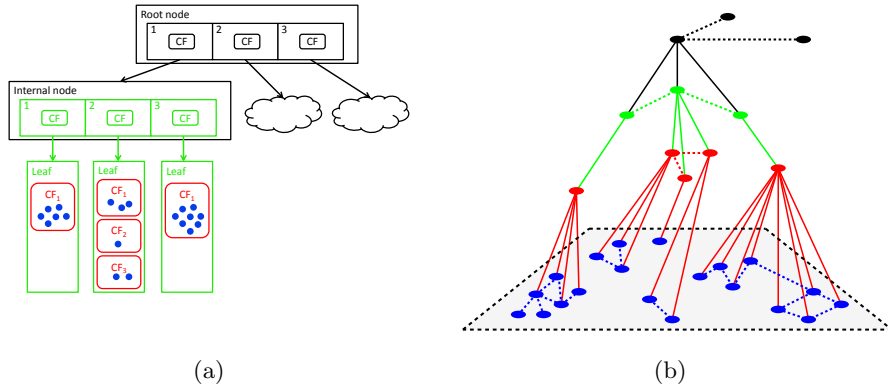


Fig. 1: (a) Illustration of a CF-tree output by the BIRCH algorithm; (b) Representation of the proposed structure: for each node, its entries are organised in a proximity graph; likewise, the data points contained in a leaf cluster are also organised in a proximity graph, allowing to highlight the topology of the data.

## 2 Incremental hierarchical indexing

### 2.1 BIRCH

BIRCH is a clustering algorithm that aims at partitioning large data set that cannot fit entirely in the memory. The key idea is to go through the data set only once and organise the data points in a *Cluster Feature* tree (CF-tree).

A Cluster Feature (CF) is a numeric vector that sums up a set of  $n$  data points  $\{x_1, \dots, x_n\}$ . It is defined by a triple  $CF = (n, LS, SS)$  where  $LS$  and  $SS$  are the linear sum and the square sum, respectively. For a given cluster  $C$  and its CF, one can easily deduce the centroid  $x_0$ , the radius  $R$  (average distance from member points to the centroid), and the diameter  $D$  (average pairwise distance between member points) of  $C$ . Thus, a cluster is well described by its CF.

The CF-tree is a height-balanced tree with three parameters  $B$ ,  $L$  and  $T$ : (i) each internal node contains at most  $B$  entries  $[CF_i, child_i]$ , where  $CF_i$  describe the cluster pointed by  $child_i$ ; (ii) each leaf contains at most  $L$  entries  $[CF_i]$  and (iii) each leaf entry  $CF_i$  must have a radius smaller than the threshold  $T$ .

More details on the BIRCH algorithm can be found in the original article [5]. Figure 1a presents an illustration of a CF-tree.

The choice of the BIRCH algorithm is motivated by several reasons. First, it produces a tree structure which is desirable for a multilevel visualisation. Second, as one can see in the Figure 1a, the internal nodes contain only the CF entries and not the data points. This reduces drastically the memory usage and allows to process large data sets that cannot fit in the memory. Last, the algorithm processes the data points in a *single pass* scheme, thus it could be implemented in an incremental way, allowing to handle growing data sets or data streams.

However, the BIRCH algorithm has a few limitations with regards to our

objective of image collection visualisation. Indeed, it has been designed to study only the leaves content and the tree structure is not leveraged. Moreover, as mentioned previously, the internal nodes contains only CF entries, which are difficult to interpret by the user during a visual exploration of the tree.

## 2.2 Proposed improvements

### 2.2.1 Representatives assignment

In order to allow the user to easily interpret the CF entries, a set of  $k$  relevant images are assigned to each entry as representatives. A bottom-up approach is used: relevant images are pulled up from the leaves to the root of the tree.

At the leaf level, the prototype of each CF entry is computed. For a given CF, the prototype corresponds to the nearest image to the centroid of CF. Then, the  $k$  nearest neighbours of the prototype are defined as the representatives of the CF.

For a given internal node CF entry  $CF$ , the  $k_i$  representatives of each of its child entries  $CF_i$  are pulled up. The values  $k_i$  are automatically computed with regards to the number of images contained in  $entry_i$ . In addition, we have  $\sum_i k_i = k$ . Thus, we assign  $k$  relevant images to  $CF$ , those images being found in the lower levels of  $CF$  child sub-tree.

This representatives assignment has been implemented in a incremental way. Thus, when a new image is inserted in the proposed tree structure, an update operation is performed. The representatives of the entries that belong to the insertion path are updated to take into account the newly inserted image.

### 2.2.2 Proximity graph structuring

In order to structure the tree elements at different levels, *proximity graphs* have been used. Figure 1b illustrates the proposed enhanced tree structure. For each internal node, a proximity graph is computed between its CF entries. Furthermore, given a leaf node CF entry, a proximity graph is computed between all the images that belong to the CF entry. In this work, the relative neighbourhood graph (RNG) has been chosen as the proximity graph. We first define the RNG and then justify this choice.

The relative neighbourhood graph has been introduced in the work of Toussaint [4]. The construction of this graph is based on the notion of *relatively close* neighbours, that defines two vertices as relative neighbours if they are at least as close to each other as they are to any other points. From this definition, we can define  $RNG = (V, E)$  as the graph built from the points of  $D$  where distinct points  $p$  and  $q$  of  $D$  are connected by an edge  $\overline{pq}$  if and only if they are relative neighbours. Thus,

$$E(RNG) = \{\overline{pq} \mid p, q \in D, p \neq q, \\ \delta(p, q) \leq \max(\delta(p, r), \delta(q, r)), \forall r \in D \setminus \{p, q\}\}.$$

where  $\delta : D \times D \rightarrow \mathbb{R}$  is a distance function.

The choice of the RNG is justified as follows: on one hand, the main drawback of the RNG is its construction. The classic and brute-force construction has a complexity of  $O(n^3)$ . This is a major limitation to handle large image collection. However this issue has been addressed in [6], where an incremental paradigm is proposed to build the RNG for large data sets. On the other hand, the RNG is a sparse graph that highlights the topology of the data and embeds local information about vertices neighbourhood. Furthermore, its connectivity property is desirable for most graph drawing algorithms. Moreover, it guarantees that each images can be reachable during a content-based exploration of the image collection.

In [6], a study has shown that the incremental construction paradigm is relevant only when the number of data points is high. Otherwise, it is more interesting to recompute the whole graph in a parallel way. Indeed, the brute-force algorithm is an embarrassingly parallel problem for small data sets.

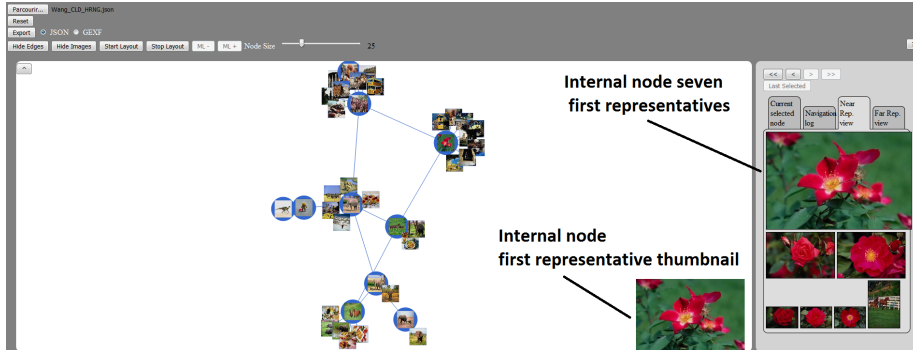
In the CF-tree that is computed by the BIRCH algorithm, the number of children of an internal node is limited by the page size, thus this number is small (less than 100). Hence, when a new image is inserted in our tree structure, the RNG of each internal node that lays in the insertion path is recomputed. Nevertheless, a leaf entry can contain an unlimited number of images. Thus, at this level, the incremental RNG construction algorithm is leveraged.

### 3 Visualisation

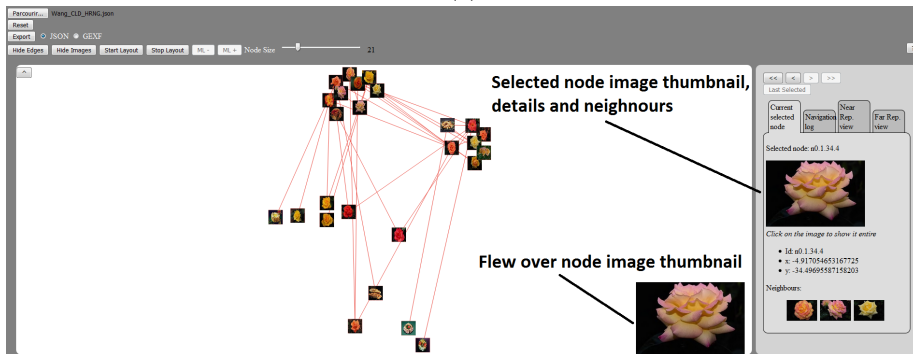
The proposed visual analysis system allows to visualise and interact with graphs of images. Figures 2 show the platform interface and succinctly describe some of the platform interactions. It has been realised using web technologies, namely HTML5, CSS3 and Javascript. This choice is explained by two reasons. First, the platform target is experts from various fields such as health or digital humanities. Nowadays, a majority of users that are not experts in computer science can still manage well web navigation. Thus, such users are familiar with web browsers. We think that presenting the system as a light web platform, would make users more disposed to exploit it. Second, the choice of discarding a server can be justified by two arguments: *(i)* there is no upload of the images to a server, operation that may cost time and *(ii)* as the images are not sent nor stored in an external server, we respect the potential confidentiality or license issues that are related to the images.

### 4 Experiments

The *Wang* [7] image data set has been used to assess the relevance of the indexing structure. *Wang* contains 1,000 images which form 10 classes of 100 images. Machine learning techniques, such as BIRCH, applied on images rely a lot on the images description. The features that are used in this work are visual de-



(a)



(b)

Fig. 2: (a) At an internal level of the tree: internal nodes are represented by the its first representative and organised by the RNG. When the user fly over an internal node, seven of its representatives are displayed. When the user click on an internal node, its subtree is displayed; (b) at a leaf entry: images are also organised with the RNG. Details of the selected node and its relative neighbours are displayed.

scriptors (*e.g.* color, contour), that do not match perfectly the semantic aspect of the images. Thus, we do not obtain the exact classes in the leaves clusters. Nevertheless, the partitioning that is performed is relevant with regards to the visual descriptors. In addition, the graph that is described in the previous section highlights the topology of the studied data set. Thus, in our case, one can observe that similar images may be linked by an edge (see Figure 2b). On the contrary, dissimilar images should not be linked, or at least by a long edge. This last phenomenon may occur because of the connectivity property of the selected proximity graph.

In order to test the scalability of the proposed incremental hierarchical index-

ing, two public larger data sets have been used, namely the MIRFLICKR-25000 and MIRFLICKR-1M image collections. For the latter data set, publicly available visual descriptors <sup>1</sup> have been used. Experiments have been done on an Intel Xeon E5-2620 v2 at 2.10Ghz. Table 1 presents the computation times (in seconds) of the proposed incremental hierarchical indexing method, with  $T = 0.0$ . We have tractable computation time for the larger image collection of one million image (less than 10 hours). Thus, the proposed modifications do not conflict with the scalability of the original BIRCH method. Furthermore, as a hierarchical structure is built, the proposed visualisation is smooth thanks to the reduction of nodes to display.

Data set	$n$	$d$	Computation time
Wang	1,000	192	3
MIRFLICKR-25000	25,000	192	100
MIRFLICKR-1M	1,000,000	80	33.782

Table 1: Datasets description:  $n$  is the number of images and  $d$  is the dimension of the descriptors. The computation times are given in seconds.

## 5 Conclusion

In this work, we have presented a study to index and visualise jointly large image collection. Experiments show that the proposed method is relevant regarding the visualisation and have good performance in terms of computation time. Further work includes improvement on the platform and user evaluation on real world cases.

## References

- [1] Domo, Data Never Sleeps 3.0, <https://www.domo.com/learn/data-never-sleeps-3-0>, 2015.
- [2] W. Plant and G. Schaefer, Visualisation and Browsing of Image Databases, *Multimedia Analysis, Processing and Communications*, 346:3-57, 2011.
- [3] Robert Kosara, Graphs beyond the hairball, <https://eagereyes.org/techniques/graphs-hairball>, 2012.
- [4] G. T. Toussaint, The Relative Neighbourhood Graph of a Finite Planar Set, *Pattern Recognition*, 12:261-268, 1980.
- [5] T. Zhang, R. Ramakrishnan and M. Livny, BIRCH: An Efficient Data Clustering Method for Very Large Databases. In *proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD 1996)*, pages 103-114, 1996.
- [6] F. Rayar, S. Barrat, F. Bouali and G. Venturini, An Approximate Proximity Graph Incremental Construction for Large Image Collections Indexing. In *proceedings of Foundations of Intelligent System 22nd International Symposium (ISMIS 2015)*, 2015.
- [7] Li, Jia and Wang, James Z., Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1075-1088, 2003.

---

<sup>1</sup><http://press.liacs.nl/mirflickr/mirdownload.html>