HAL
open science

Construction incrémentale d'une structure hiérarchique pour l'exploration visuelle et interactive de larges collections d'images

Frédéric Rayar, Sabine Barrat, Fatma Bouali, Gilles Venturini

To cite this version:

Frédéric Rayar, Sabine Barrat, Fatma Bouali, Gilles Venturini. Construction incrémentale d'une structure hiérarchique pour l'exploration visuelle et interactive de larges collections d'images. 16e Conférence Internationale Francophone sur l'Extraction et Gestion des Connaissances, Jan 2016, Reims, France. hal-01315642

HAL Id: hal-01315642

https://hal.science/hal-01315642v1

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

Construction incrémentale d'une structure hiérarchique pour l'exploration visuelle et interactive de larges collections d'images

Frédéric Rayar*, Sabine Barrat*,
Fatma Bouali*,** Gilles Venturini *

*Université François-Rabelais de Tours, Laboratoire d'Informatique
64 avenue Jean Portalis, 37200 Tours, France,
frederic.rayar@univ-tours.fr, sabine.barrat@univ-tours.fr, gilles.venturini@univ-tours.fr
**Université de Lille2, IUT, Dpt STID
25-27 Rue du Maréchal Foch, 59100 Roubaix, France,
fatma.bouali@univ-lille2.fr

Résumé. Dans cet article, nous étudions de manière conjointe la construction et l'exploration visuelle d'une structure de classification pour de très grande base d'images. Pour garantir que la structure construite vérifiera les contraintes de taille nécessaires à sa visualisation dans une interface Web tout en reflétant les propriétés topologiques des données (*clusters*), nous combinons la classification hiérarchique de BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) avec la construction de graphes de voisinage : un graphe de voisinage est créé et mis à jour de manière incrémentale pour représenter les fils de chaque nœud de l'arbre. De plus, un ensemble d'images représentatives est remonté à chaque nœud interne pour guider l'utilisateur lors de l'exploration visuelle de l'arbre. L'ensemble des algorithmes utilisés sont incrémentaux pour gérer l'insertion de nouvelles images dans la collection. Nous présentons les premiers résultats sur des dizaines de milliers d'images qui peuvent être ainsi structurées en une minute de temps de calcul. L'exploration dans l'interface est fluide grâce aux propriétés de la structure construite.

1 Introduction

Dans ces travaux, nous nous intéressons à la structuration et la visualisation de larges collections d'images. En effet, la dernière décennie a vu la réduction du coût des appareils photos, des webcams et des scanners, mais aussi celui des supports de stockage. De fait, la quantité d'images capturées par tout un chacun a explosé, qu'elles soient générées dans le cadre privé, commercial ou dans celui de projets de numérisation (humanités numériques). De plus, l'avènement d'Internet a accentué le fait que le nombre d'images mises en ligne croît de manière exponentielle, notamment avec les sites d'instituts ou encore les réseaux sociaux (Domo, 2015). Ainsi, les collections d'images devenant conséquentes, il est apparu nécessaire de proposer des paradigmes pour les visualiser.

Un certain nombre de travaux ont été proposés pour visualiser des collections d'images. Un état de l'art plus complet peut être trouvé dans (Plant et Schaefer, 2011). Parmi les paradigmes proposés, l'approche à base de graphes est une solution simple sur laquelle des interfaces de visualisation ergonomiques et intuitives peuvent être construites. Néanmoins, on se confronte à certaines limitations liées à la *scalabilité* en terme de nombre de nœuds et d'arêtes. Tout d'abord, celles concernant la visualisation elle-même : limitation du nombre d'images, le phénomène de boule de poils (*hairball*, (Kosara, 2012)), superposition des nœuds, tracé des arêtes ; puis celles qui concernent les interactions : temps de réponse trop long, difficulté d'afficher les images liées aux nœuds, *etc*.

Dans cet article, l'étude de la construction et de la visualisation d'une structure de classification d'une très grande base d'images est menée de manière conjointe. Nous proposons d'utiliser une approche hybride mêlant *clustering* et *graphe*. Pour ce faire, nous présentons une modification de l'algorithme BIRCH (Zhang et al., 1996) afin de construire de manière incrémentale une structure hiérarchique contenant les images. Chaque niveau de la hiérarchie est organisé à l'aide de graphes de voisinage. Cette structure permet une visualisation fluide dans une interface web tout en reflétant les propriétés topologiques des données. La structure est conçue de manière incrémentale pour gérer l'insertion de nouvelles images dans la collection.

2 Algorithme proposé

2.1 BIRCH

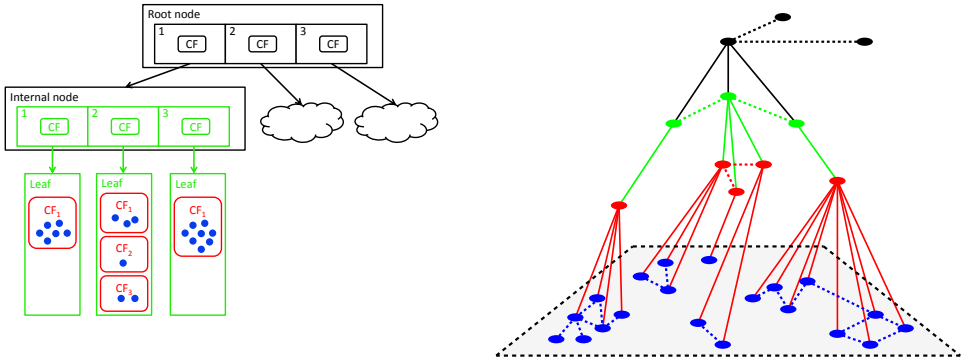
BIRCH est un algorithme de clustering qui a pour objectif premier de partitionner de très grands jeux de données, qui ne peuvent être stockés entièrement dans la mémoire. L'idée est de parcourir l'ensemble des données une fois et de les organiser dans un arbre de *Cluster Feature* (arbre CF). Un Cluster Feature (CF) est un vecteur numérique qui résume un ensemble de N données $\{x_1, ..., x_N\}$. Il est défini par un triplet $CF = (N, LS, SS)$ où LS et SS sont respectivement la somme linéaire et la somme quadratique. Pour un cluster C donné et son CF, on calcule facilement le centroid x_0, le rayon R (distance moyenne entre les données et le centroid) et le diamètre D (distance moyenne entre deux données) de C. On a donc une description du cluster C à l'aide de son CF.

L'arbre CF est un arbre équilibré qui dépend de 3 paramètres B, L et T :

- chaque nœud interne contient au plus B éléments $[CF_i, child_i]$, où CF_i décrit le cluster pointé par $child_i$;
- chaque feuille contient au plus L élements $[CF_i]$;
- chaque élément CF_i d'une feuille doit avoir un rayon inférieur à T.

Plus de détails sur l'algorithme BIRCH sont disponibles dans l'article original de Zhang et al. (1996). La Figure 1a présente une illustration d'un arbre CF.

Le choix de cet algorithme se justifie tout d'abord par la structure d'arbre qui est générée, structure souhaitable dans le cadre d'une visualisation multi-niveaux. De plus, comme on le voit sur la Figure 1a, ce sont les CF qui apparaissent dans les nœuds internes de l'arbre et non les données elles-mêmes, ce qui constitue une économie importante en mémoire, permettant ainsi de traiter de grands jeux de données. Enfin, l'algorithme BIRCH permet de traiter ces grands volumes de données en ne parcourant qu'une seule fois ces données, et peut être implémenté de manière incrémentale.



(a) (b)

FIG. 1: (a) Illustration d'un arbre CF généré par l'algorithme BIRCH ; (b) Représentation de la structure proposée : pour chaque nœud, on organise ses CF à l'aide d'un graphe de voisinage ; de même, les données contenues dans un cluster d'une feuille sont elles-aussi organisées par un graphe, ce qui permet de faire ressortir la topologie des données.

Néanmoins, il possède quelques limitations quant à notre objectif de visualisation de collections d'images. En effet, l'algorithme a été pensé pour ne tirer parti que des feuilles de l'arbre et la structure d'arbre n'est pas exploitée. De plus, comme mentionné ci-dessus, les nœuds internes contiennent des CF, peu interprétables par l'utilisateur durant l'exploration visuelle de l'arbre.

2.2 Algorithme proposé

Afin d'adapter l'algorithme BIRCH à notre cas d'utilisation, nous avons apporté plusieurs modifications. Parmi les plus importantes, on peut citer :

- *L'assignation à chaque nœud interne d'un ensemble de k images représentatives des images contenues dans son sous-arbre*. L'objectif ici est de permettre à l'utilisateur d'interpréter un nœud interne et de guider son exploration visuelle de l'arbre. Une approche ascendante (*bottom-up*) a été utilisée : on remonte des images représentatives depuis les feuilles jusqu'à la racine. Ainsi, lors de l'insertion d'une nouvelle image, une opération de mise à jour des représentants est effectuée. Au niveau d'une feuille, on détermine son prototype, *i.e.* l'image la plus proche du centroïde de l'ensemble des images contenues dans la feuille. Puis on désigne les k plus proches voisins du prototype comme étant les représentants de la feuille. Au niveau d'un nœud interne NI, on remonte les k_i représentants de chaque fils $fils_i$ de NI. k_i est déterminé proportionnellement au nombre d'images de $fils_i$. De plus, on a $\sum_i k_i = k$. On assigne ainsi k images pertinentes à NI, images que l'on retrouvera aux niveaux inférieurs de l'arbre.
- *La structuration des fils d'un nœud interne et les images d'une feuille par un graphe de voisinage.* Nous avons retenu le graphe des voisins relatifs (RNG) (Toussaint, 1980) pour cette opération. Ce choix de graphe est justifié par le fait qu'il s'agit d'un graphe connecté, propriété utile pour l'affichage de graphe. La complexité de la construction du

RNG est $O(n^3)$. De fait, ce graphe ne semble pas adapté pour traiter de grands volumes de données. Néanmoins, il existe une approche de construction incrémentale du RNG pour structurer de grands volumes de données. Une étude préalable a montré que l'intérêt de ce paradigme de construction incrémentale est pertinent uniquement lorsque le nombre de données est suffisamment élevé. Dans le cas contraire, il est plus intéressant de recalculer l'ensemble du graphe de manière parallèle : en effet, il est extrêmement facile de paralléliser la construction du RNG pour un petit volume de données.
Dans l'algorithme BIRCH, le nombre de fils d'un nœud interne étant limité par la taille d'une page mémoire, on n'a que peu de fils (< 100). Aussi, lors de l'opération de l'insertion d'une nouvelle image dans l'arbre, on recalcule entièrement le RNG pour chaque nœud interne concerné. Cependant, un cluster d'une feuille peut contenir un nombre non limité d'images. Aussi, nous utilisons à ce niveau la construction incrémentale du graphe à l'aide d'opérations de mises à jour locales. La Figure 1b illustre la structure d'arbre enrichie des graphes.

3 Plateforme de visualisation

La plateforme de visualisation interactive de grandes bases d'images a été réalisée à l'aide de technologies web : HTML5, CSS3 et Javascript. Ce choix se justifie par deux raisons :

1. de nos jours, une majorité des utilisateurs non spécialisés en informatique reste néanmoins habituée à utiliser des navigateurs web. Ainsi, en évitant l'installation d'un logiciel dédié, ce type d'utilisateur sera peut-être plus enclin à utiliser la plateforme ;
2. le choix de ne pas utiliser de serveur se justifie quant à lui par deux arguments : *(i)* on évite ainsi l'envoi des images sur un serveur externe, qui peut être long en présence de larges collections d'images, *(ii)* les images n'étant ni transmises, ni stockées sur un serveur externe, on respecte ainsi l'aspect confidentiel que peuvent avoir les collections d'images dans certains domaines (*e.g.* médical, surveillance).

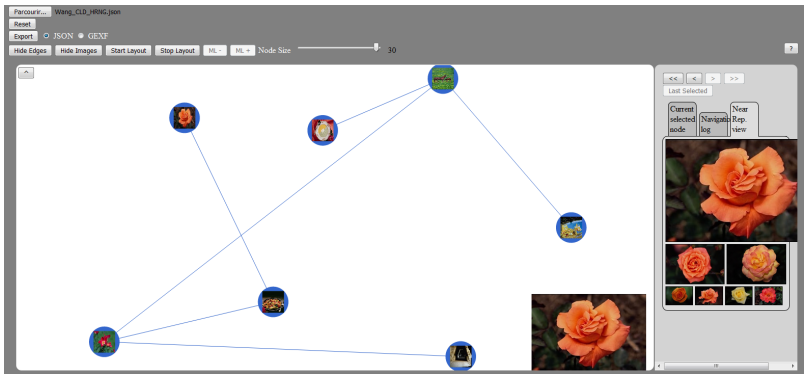
Le plateforme permet de visualiser et d'interagir avec des graphes plats ou hiérarchiques tels que celui que nous proposons dans cet article. Parmi les vues et les interactions disponibles, on peut citer :

- au niveau des nœuds internes : ceux-ci sont positionnés grâce au RNG. Lors du survol d'un nœud interne, un aperçu de l'image représentative de son sous-arbre est affiché dans l'interface du graphe. De plus, un ensemble de k images représentatives de son sous-arbre est affiché à droite de l'interface ($k = 7$ dans le prototype actuel). Ainsi, l'utilisateur possède une information générale qui lui permet de continuer son exploration au sein de l'arbre ;
- au niveau des feuilles : le RNG des images contenues dans la feuille est dessiné, et l'utilisateur a la possibilité d'afficher les images associées ou non : l'aperçu reste affiché lors du survol d'une image. Sur la sélection d'une image, un focus est réalisé sur cette dernière et ses adjacences sont mises en avant. De plus, ses images voisines sont affichées à droite de l'interface : l'utilisateur peut ainsi naviguer de proche en proche à l'aide de ce menu ;
- les interactions classiques de type zoom et déplacement sont disponibles sur la plateforme à tout niveau de l'arbre ;

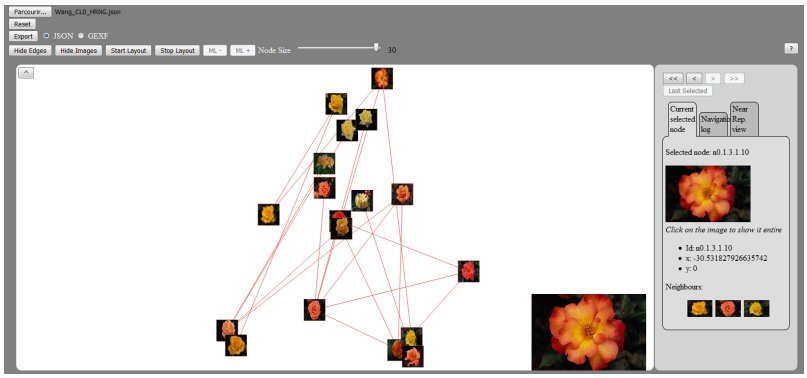
– enfin, des fonctionnalités de gestion d'historique de l'exploration sont présentes et permettent à l'utilisateur de revenir dans son historique de navigation s'il le désire.

Deux collections d'images ont été utilisées dans nos expériences préliminaires : *Wang* (Li et Wang, 2003), collection de 1000 images et une collection de 43000 images en libre accès du National Gallery of Art [1]. Des descripteurs de couleur ont été extraits de ces images. La génération de la structure proposée a respectivement pris $7,5s$ et $73,6s$ sur un Intel Core i5 CPU M460 (quadcore) à 2.53Ghz et 8Go de RAM.

Des aperçus de l'interface de visualisation de la collection *Wang* sont présentés Figure 2.



(a)



(b)

FIG. 2: (a) À un niveau intermédiaire de l'arbre : les nœuds internes sont représentés par une image représentative et organisés à l'aide de leur RNG. 7 images représentatives d'un nœud interne sont affichées lors de son survol ; (b) au niveau d'un cluster d'une feuille : les images contenues sont elles aussi organisées à l'aide de leur RNG ; sur la sélection d'une image, ses informations sont affichées ainsi que ses voisins dans le graphe.

1. `https://images.nga.gov`

4 Conclusion

Dans cet article, nous avons proposé une modification de l'algorithme BIRCH en vue de construire et d'explorer visuellement une structure de classification d'une très grande base d'images. Les améliorations apportées permettent de structurer de manière hiérarchique une collection d'images tout en reflétant les propriétés topologiques de cette collection. Une plateforme de visualisation interactive a été réalisée afin d'exploiter la structure d'arbre de graphes. Les expériences préliminaires ont montré des retours positifs des utilisateurs vis-à-vis de notre plateforme d'exploration. Les futurs travaux seront surtout dédiés à l'amélioration de la plateforme. En effet, de nombreuses perspectives s'ouvrent pour enrichir cette dernière. Tout d'abord en terme de visualisation, en permettant à l'utilisateur de se repérer au sein de l'arbre, en affichant l'arbre ou un fil d'ariane. Puis en terme d'interaction, avec des outils tels que des *lenses* ou d'édition de graphes. De même, nous pouvons imaginer un système de recommandation pour suggérer à l'utilisateur quelle(s) partie(s) de l'arbre visiter. Enfin, l'évaluation de notre plateforme via une étude utilisateur sera réalisée.

Références

Domo (2015). Data never sleeps 3.0. https ://www.domo.com/learn/data-never-sleeps-3-0.

Kosara, R. (2012). Graphs beyond the hairball. https ://eagereyes.org/techniques/graphs-hairball.

Li, J. et J. Z. Wang (2003). Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Anal. Mach. Intell. 25*(9), 1075–1088.

Plant, W. et G. Schaefer (2011). Visualisation and browsing of image databases. In *Multimedia Analysis, Processing and Communications*, Volume 346 of *Studies in Computational Intelligence*, pp. 3–57.

Toussaint, G. T. (1980). The relative neighbourhood graph of a finite planar set. *Pattern Recognition 12*, 261–268.

Zhang, T., R. Ramakrishnan, et M. Livny (1996). Birch : An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, New York, NY, USA, pp. 103–114. ACM.

Summary

In this paper, we jointly study the construction and visualisation of a classification structure of large image collection. Our main goal is to assure the fact that the built structure will be viewable in a web interface while highlighting the topology of the data. To do so, we combine the BIRCH algorithm tree (Balanced Iterative Reducing and Clustering using Hierarchies) with the construction of proximity graphs: such a graph is built and incrementally updated to represent the children of each node in the tree. Furthermore, images representatives are pulled up to internal nodes to help the user during the tree exploration. All the structures have been built to allow dynamic updates regarding the insertion of new images in the collection. The first experiments allow to structure thousands of images in a minute and provides a fluid and intuitive navigation.