Tạp chí Khoa học Đại học Huế: Kỹ thuật và Công nghệ; pISSN 2588-1175 | eISSN 2615-9732

Tập 129, Số 2A, 2020, Tr. 31–47; DOI: 10.26459/hueuni-jtt.v129i2A.5647

MỘT PHƯƠNG PHÁP CẢI TIẾN CHO BÀI TOÁN TÌM KIẾM ẢNH DỰA TRÊN CÂY R-TREE

Lê Thị Vĩnh Thanh[1], Nguyễn Thị Định[2], Nguyễn Minh Hải[3], Văn Thế Thành[4*]

[1] Trường Đại học Bà Bịa – Vũng Tàu
[2] Khoa Công nghệ thông tin, Trường Đại học Công nghiệp Thực phẩm TP.HCM
[3] Khoa Vật Lý, Trường Đại học Sư phạm TP.HCM
[4] Phòng Quản lý khoa học và Đào tạo sau đại học, Trường Đại học Công nghiệp Thực phẩm TP.HCM

Abstract. Tóm tắt. Trong bài báo này chúng tôi trình bày một cải tiến cho cây phân cụm R-Tree, ký hiệu là RG-Tree (Region Growth Tree), nhằm nâng cao hiệu quả tìm kiếm ảnh tương tự theo nội dung. Trong cải tiến này, chúng tôi lưu trữ các véc-tơ đặc trưng của hình ảnh trên mỗi nút lá của cây RG-Tree theo quy tắc phân hoạch đã được đề xuất. Cây RG-Tree có thể tăng trưởng nhằm lưu trữ các vùng dữ liệu và phân bố trên các nút lá tạo thành các cụm dữ liệu. Việc phân hoạch này được thực hiện phân bố các phần tử càng giống nhau thì càng thuộc về một nhóm các nhánh con trên cây RG-Tree. Trên cơ sở lý thuyết đã đề nghị, một mô hình tìm kiếm ảnh được thiết kế dựa trên cây RG-Tree và được thực nghiệm trên các bộ ảnh ImageCLEF. Cuối cùng chúng tôi tiến hành so sánh hiệu suất tìm kiếm với một số phương pháp gần đây trên cùng bộ dữ liệu.

Từ khoá: RG-Tree, CBIR, Similar Images, Similarity Measure, Image Retrieval

1 Giới thiệu

Để giải quyết bài toán tìm kiếm ảnh theo nội dung (content-based image retrieval - CBIR). Hai vấn đề cần được thực hiện bao gồm (1) tạo véc-tơ đa chiều nhằm mô tả đặc trưng nội dung cấp thấp của hình ảnh, (2) xây dựng mô hình thực hiện tìm kiếm ảnh tương tự dựa trên đặc trưng cấp cao của hình ảnh. Một số kỹ thuật tạo chỉ mục đa chiều cho tập các véc-tơ đặc trưng như kỹ thuật tạo chỉ mục dựa trên đặc trưng cấp thấp của phân vùng dữ liệu ảnh [1, 8, 11], kỹ thuật lập chỉ mục dựa trên đặc trưng không gian và lưu trữ trên các cây KD-Tree [7], Quard- Tree [14], R-Tree [1, 2], v.v. Trong đó, cây KD-Tree là một cấu trúc lưu trữ chỉ mục dựa trên phân vùng không gian, cây Quard- Tree, R-Tree là cấu trúc lưu trữ chỉ mục dựa trên phân vùng dữ liệu. R-Tree là cây đa nhánh cân bằng và dữ liệu được lưu tại các nút lá được dùng để phân vùng dữ liệu thành các khối có thể lồng nhau hoặc chồng lên nhau, được giới thiệu bởi Guttman vào năm 1984 [5]. Các cấu trúc này được ứng dụng hiệu quả trong việc lưu trữ cũng như tìm kiếm dữ liệu hình ảnh [1, 9, 10].

Trong bài báo này, chúng tôi xây dựng mô hình phân cụm dữ liệu cho đặc trưng hình ảnh dựa trên cây RG-Tree (một cải tiến của cây R-Tree). Cây RG-Tree là cây tăng trưởng và là một mô

** Liên hệ: thanhvt@hufi.edu.vn*

Nhận bài: 13–01–2020; Hoàn thành phản biện: 21–01–2020; Ngày nhận đăng: 22–01–2020

hình phân cụm tự động các bộ dữ liệu dựa trên kỹ thuật phân cụm k-Mean, k-NN theo mô hình học bán giám sát.

2 Các công trình liên quan

Trong những năm gần đây, các hệ thống tìm kiếm ảnh được thực hiện bởi nhiều phương pháp khác nhau và mang lại những kết quả tốt. C. P. Singh và cộng sự đã sử dụng phương pháp dò cạnh Sobel và dựa vào độ đo tương tự để thực hiện tìm kiếm ảnh trên cấu trúc cây R-Tree [1]. Nam N.V và Bac L.H đề xuất phương pháp gom cụm thực hiện trên cấu trúc cây R-Tree [2]. N. Amoda và R. K. Kulkarni đã đề xuất một hệ thống tìm kiếm hình ảnh dựa trên vùng RBIR (Region based Image Retrieval) [3]. Shama P.S, và cộng sự đã đề xuất hệ thống tìm kiểm ảnh dựa theo nội dung dựa trên cấu trúc cây R*-Tree để xác định hình ảnh thực vật [4]. Van T.T. và cộng sự đã giới thiệu một phương pháp cải tiến cho hệ tìm kiếm ảnh theo nội dung dựa trên chữ ký nhị phân và cây S-Tree [5]. Các công trình này được thực hiện bởi nhiều phương pháp và thực nghiệm trên các bộ dữ liệu khác nhau và thu được các kết quả khả quan, cụ thể là:

Năm 2011, Chandresh Pratap Singh đã sử dụng một thuật toán dò cạnh Sobel và độ đo ma trận khoảng cách để tìm kiếm ảnh tương tự dựa trên cấu trúc cây R-Tree [1]. Năm 2012, Nam N.V và Bac L.H đã đề xuất một thuật toán phân cụm dựa trên cây R-Tree [2]. Trong các nghiên cứu này kết quả thực nghiệm cho thấy độ chính xác còn phụ thuộc vào ngưỡng đặt ra và phụ thuộc vào cường độ ảnh, chưa tiếp cận dữ liệu về hình ảnh và chưa thực hiện bài toán tìm kiếm ảnh theo nội dung.

Năm 2013, Niket Amoda và Ramesh K Kulkarni đã đề xuất một hệ thống tìm kiếm hình ảnh dựa trên vùng (Region based Image Retrieval- RBIR) sử dụng không gian màu HSV, phép biến đổi Wavelet (Discrete Wavelet Transform DWT) và thuật toán gom cụm K-Means để phân chia hình ảnh thành các vùng. Độ đo Bhattacharyya được sử dụng để tính độ tương tự giữa các vùng [3]. Năm 2015, Shama P.S, và cộng sự đã đề xuất hệ thống tìm kiếm ảnh theo nội dung dựa trên cấu trúc cây R*-Tree và độ đo Euclide để xác định hình ảnh thực vật [4]. Tuy nhiên cả hai công trình này chưa tạo ra được cấu trúc dữ liệu lưu trữ dữ liệu hình ảnh nhằm nâng cao hiệu quả cho bài toán tìm kiếm ảnh theo nội dung.

Năm 2017, Van T.T. và cộng sự đã giới thiệu một phương pháp cải tiến cho hệ tìm kiếm ảnh theo nội dung dựa trên chữ ký nhị phân (binary signature) và cây S-Tree. Trong công trình này, nhóm tác giả đã đề xuất cải tiến một cấu trúc dữ liệu cây đa nhánh và thực nghiệm trên bộ ảnh COREL [5]. Kết quả thực nghiệm cho thấy phương pháp đề xuất giải quyết tốt bài toán tìm kiếm ảnh theo nội dung. Tuy nhiên, nhóm tác giả chưa kết hợp được các đặc trưng của hình ảnh để tăng độ chính xác cho quá trình tìm kiếm.

Năm 2019, Maher Alrahhal và Supreethi K.P đề xuất phương pháp truy vấn ảnh dựa trên kỹ thuật học có giám sát và sử dụng phương pháp lấy mẫu láng giềng cục bộ (Local Neighbor Pattern- LNP) [6]. Ngoài ra, một số mô hình truy vấn ảnh cũng đã được công bố như mô hình truy vấn ảnh dựa trên cây phân cụm tự cân bằng C-Tree trên bộ ảnh ImageCLEF, kết quả thực nghiệm có độ chính xác là 65% và thời gian truy vấn trung bình khoảng 73 milli seconds [13]. Một mô hình truy vấn ảnh dựa trên cây phân cụm phân cấp H-Tree cũng được đề xuất và thực nghiệm trên bộ ảnh này cho kết quả có độ chính xác là 67% [14]. Ngoài ra, nhiều công trình nghiên cứu về tìm kiếm ảnh tương tự dựa trên cấu trúc dữ liệu dạng cây như tạo chỉ mục và cây chữ ký [15]; truy vấn ảnh dựa trên độ đo EMD và cây S-Tree [16]. Mô hình truy vấn ảnh dựa trên cây phân cụm đa nhánh cân bằng [17] v.v. cũng thu được những kết quả khả quan.

Từ kết quả phân tích như trên cho thấy cần phải tạo ra một cấu trúc dữ liệu lưu trữ chỉ mục mô tả cho hình ảnh, đồng thời kết hợp các phương pháp học máy để thực hiện bài toán tra cứu ảnh. Do đó trong bài báo này, chúng tôi tiếp cận theo phương pháp tổ chức và cải tiến cây R-Tree để tạo thành cây RG-Tree nhằm lưu trữ dữ liệu đặc trưng cấp thấp của tập dữ liệu ảnh, đồng thời truy vấn nhanh các hình ảnh tương tự dựa trên kỹ thuật học bán giám sát. Mô hình được thực nghiệm trên bộ ảnh ImageCLEF để minh chứng tính hiệu quả của mô hình truy vấn ảnh đã được đề xuất dựa trên cấu trúc cây RG-Tree.

3 Cây phân cụm dữ liệu không gian đa chiều RG-Tree

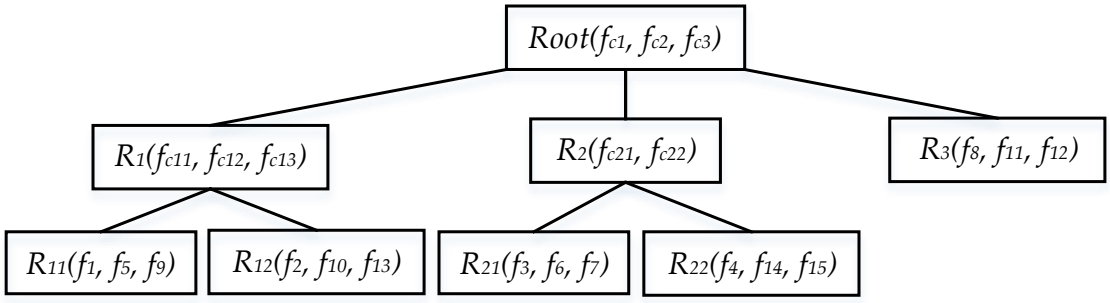
3.1 Cấu trúc cây RG-Tree

Trong cây R-Tree, mỗi nút trong là một vùng không gian hình chữ nhật hoặc đa giác chứa các vùng không gian con bên trong nó và chứa các liên kết đến các nút con. Mỗi nút trên cây có số phần tử tối thiểu là m và số phần tử tối đa là M. Mỗi nút lá là một vùng không gian chứa các liên kết trỏ đến các đối tượng dữ liệu. Mỗi nút lá phân chia dữ liệu thành một cụm trong không gian k-chiều [1]. Việc loại bỏ một phần tử trên cây R-Tree có thể phải tái tạo cây lại từ đầu vì nếu một nút lá có số phần tử bằng m thì khi xóa phần tử đó thì nút lá không tồn tại; phải lấy phần tử còn lại của nút lá đó phân bố lại trên cây. Trong quá trình tạo một nút trong cây nếu nút đó chưa vượt qua số lượng phần tử tối đa thì hai phần tử đó vẫn nằm trong một nút lá. Điều này có nghĩa là hai phần tử khác nhau cũng có thể nằm trên một nút. Và như vậy sai số của quá trình truy vấn có thể xảy ra. Việc tìm kiếm sẽ chọn theo hướng tốt nhất trên cây, tuy nhiên trong trường hợp hướng tốt nhất vẫn không liên quan thì vẫn phải chọn một nút lá. Điều đó dẫn đến kết quả tìm kiếm có thể không liên quan đến ảnh cần tìm kiếm.

RG-Tree là một cải tiến cây R-Tree, là cây đa nhánh gần cân bằng nhằm ứng dụng cho bài toán tìm kiếm ảnh tương tự. Việc gom nhóm dữ liệu được thực hiện trên từng nút của cây RG-Tree dựa vào độ đo tương tự giữa các véc-tơ đặc trưng ảnh và các ngưỡng ε, θ ($0< \varepsilon < \theta<1$) cho

trước nhằm tạo ra một cây đa nhánh để tăng tốc độ tìm kiếm ảnh và có độ chính xác cao. Cây RG-Tree bao gồm một nút gốc (*root*), tập các nút trong (*inNode*) và nút lá (*lvNode*). Mỗi nút trong chứa các liên kết đến các nút con tạo ra đường dẫn từ gốc đến lá. Mỗi nút lá lưu trữ tập các véc-tơ đặc trưng của ảnh. Các phần tử tại mỗi nút trong và nút lá được phân bổ lần lượt theo bán kính θ và ε. Trong cây nút gốc và nút trong chứa véc-tơ tâm và các liên kết đến các nút con; nút lá chứa tập các véc-tơ đặc trưng của ảnh.

Trong phần này, chúng tôi mô tả cấu trúc cây RG-Tree để phân cụm tập véc-tơ đặc trưng ảnh $\{f_1, f_2, \dots, f_n\}$. Cây RG-Tree được mô tả như trong **Hình 1**.



Hình 1. Cây RG-Tree dạng sơ đồ phân cấp

Gọi $E =< f,\ id >$ là một thành phần trong một nút trên cây, với $f = (v_1, .., v_k)$, id lần lượt là vec-tơ đặc trưng ảnh, định danh của ảnh. Cây RG-Tree lưu trữ tập các véc-tơ đặc trưng ảnh $T = \{E_i \mid i = 1, .. N\}$, trong đó N là số lượng ảnh trong bộ dữ liệu. Cây RG-Tree được dùng để phân cụm tập các véc-tơ đặc trưng của ảnh dựa trên khoảng cách *Euclide*.

Gọi f_i, f_j lần lượt là hai véc-tơ đặc trưng của hai ảnh I, J và $\boldsymbol{\varepsilon}$ là một số thực dương khá bé. Độ tương tự được định nghĩa như sau:

Định nghĩa 1. Hai ảnh I, J được gọi là tương tự nhau nếu $\boldsymbol{d(f_I, f_J) < \varepsilon}$. Trong đó $\boldsymbol{d(f_I, f_J)}$ là khoảng cách Euclide giữa hai véc tơ đặc trưng của ảnh I và J.

Cây RG-Tree tạo ra một mô hình phân cụm tập các vec-tơ đặc trưng ảnh nhằm phục vụ cho bài toán tìm kiếm ảnh tương tự. Kết quả quá trình tạo cây là tập các nút trong và nút lá. Gọi $< inE_i, links_i >$ là một bộ mô tả một thành phần của nút trong với lần lượt là phần tử và liên kết đến nút kế cận, $< lvE_i, id >$ là một bộ mô tả các thành phần của một nút lá với lần lượt là phần tử và định danh ảnh. Các nút của cây RG-Tree được định nghĩa như sau:

Định nghĩa 2. Một cây phân cụm RG-Tree là một cây đa nhánh gồm:

a) Một nút gốc $root$ liên kết đến các nhánh kế cận: $root = \{< inE_i, links_i >, i = 1 \dots t\}$;
b) Một tập nút *trong*: $inNode = \{< inE_i, links_i >, i = 1 \dots K\}$, $inE =< f_c, links >$, thỏa $d(f_{ci}, f_{ctb}) \le \theta$, $f_{ctb} = \frac{1}{M}\sum_1^M f_{ci}$.

c) Một tập nút *lá*: $lvNode = \{< lvE_i, id >, i = 1 \dots K\}$, $lvE =< f, id >$ thỏa $d(f_i, f_c) < \varepsilon$, $f_c = \frac{1}{M}\sum_1^M f_i$.

Tại thời điểm ban đầu, cây RG-Tree chỉ gồm một nút gốc chứa các liên kết là rỗng. Sau đó, từng phần tử E_i được thêm vào cây để tạo ra các nhánh tương ứng với các nút lá trong cây dựa trên độ đo Euclide. Quy tắc phân bố các phần tử trên cây được định nghĩa như sau:

Quy tắc phân bố phần tử trong cây:

Phần tử trong cây RG-Tree, bắt đầu thực hiện từ nút gốc và lần lượt thực hiện theo các quy tắc sau:

Quy tắc 1: Chọn hướng đi từ nút hiện hành đến các nút của nhánh kế cận và chọn nhánh có khoảng cách $d(f_i, f_{cj})$ ngắn nhất.

Quy tắc 2: Nếu $d(f_i, f_{cj}) \leq \theta$ thì đi theo nhánh đó và tìm nhánh con phù hợp tiếp theo đến khi gặp nút lá, có 3 trường hợp sau đây:

1) Nếu $d(f_i, f_{cj}) < \varepsilon$ thì đưa phần tử f_i vào nút lá hiện hành (nút lá có tâm f_{cj}).
2) Nếu $\varepsilon \leq d(f_i, f_{cj}) \leq \theta$ thì tạo một nút lá mới ($newLeaf$) chứa f_i và một nút cha mới liên kết đến nút lá có tâm f_{cj} và nút $newLeaf$.
3) Nếu $d(f_i, f_{cj}) > \theta$ thì tạo một nút $newLeaf$ chứa f_i có cùng cha với nút lá hiện hành.

Quy tắc 3: Nếu $d(f_i, f_{cj}) > \theta$ thì tạo một nút $newLeaf$ chứa f_i có cùng cha với nút lá hiện hành.

Vì dữ liệu ảnh được gia tăng nhanh chóng, do đó cây RG-Tree phải có khả năng tăng trưởng để phù hợp cho việc lưu trữ dữ liệu ảnh. Định lý sau đây minh chứng tính tăng trưởng của cây RG-Tree.

Định lý 1. Cây RG-Tree là cây tăng trưởng theo hướng từ gốc tới lá.

Chứng minh: Theo Định nghĩa 3, mỗi khi thêm một phần tử thì phần tử này sẽ chọn được một hướng đi phù hợp và được thêm vào một nút lá hiện tại hoặc tạo một nút lá mới. Do đó, cây RG-Tree là cây tăng trưởng ■

Mỗi phần tử được lần lượt thêm vào cây, do đó cần phải tồn tại một nút để chứa phần tử này. Định lý sau đây chứng minh tính tồn tại và duy nhất của một nút trên cây RG-Tree lưu trữ các phần tử.

Định lý 2. Với mỗi véc-tơ đặc trương *f* cho trước luôn tồn tại một nút trên cây chứa *f*.

*Chứng minh: g*ọi f_i là véc-tơ cần thêm vào một nút trên cây RG-Tree. Theo Định nghĩa 3, thực hiện quy tắc tạo cây thì phần tử f_i này thuộc về một nút lá hiện tại hoặc tạo một nút lá mới phù hợp. Do đó, ta luôn tìm được một nút để lưu trữ vec-tơ f_i ■

Định lý 3. Một véc-tơ đặc trưng f_i được lưu trữ trong một nút lá duy nhất trên cây RG-Tree.

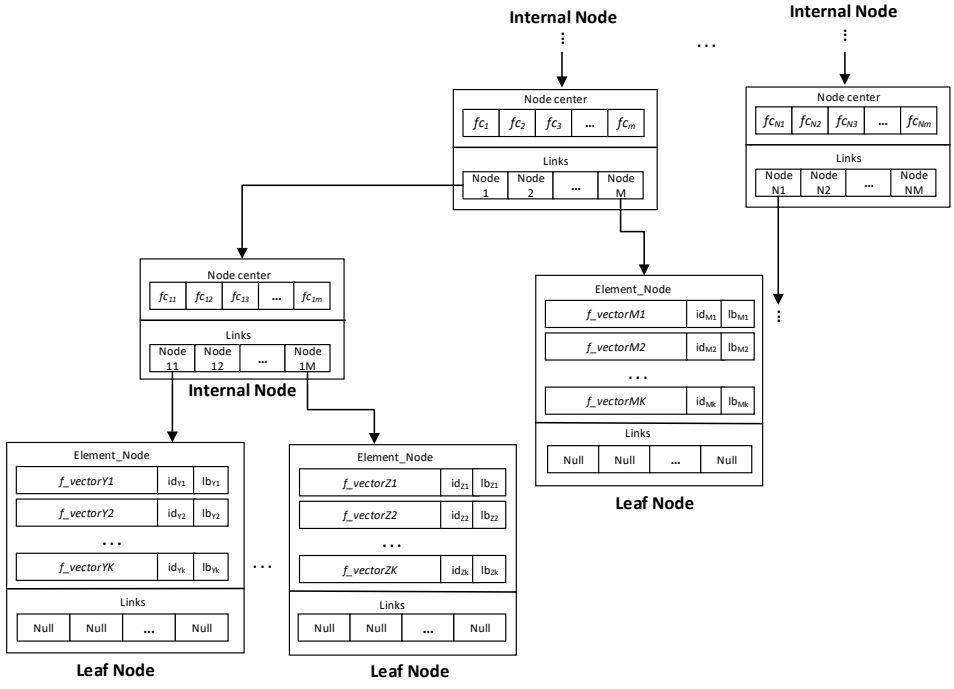
Chứng minh: Giả sử véc-tơ f_i thuộc về hai nút lá lần lượt có hai tâm là f_{c1} và f_{c2}. Theo quy tắc 2 của định nghĩa 3 ta có: $d(f_i, f_{c1}) < \varepsilon$ và $d(f_i, f_{c2}) < \varepsilon$. Mặt khác nếu $d(f_i, f_{ci}) \geq \varepsilon$ thì tạo ra một nhánh mới tức là các nút lá không giao nhau. Do đó điều giả sử ban đầu là vô lý. Vì vậy mỗi phần tử f_i chỉ được lưu trữ trong một nút lá duy nhất ■

Cây RG-Tree tạo ra một phân hoạch đa tầng, do đó khi tìm một cụm có bán kính ε thì độ chính xác đạt cao nhất. Nếu trong phạm vi bán kính θ, cây sẽ phân hoạch vec-tơ đặc trưng về một nhánh, và do đó các phân hoạch sẽ có xu hướng đều đặn vì các phần tử quá khác biệt nhau sẽ không thuộc một nhánh của cây, và như vậy cây RG-Tree có xu hướng tự nhiên trở thành đa nhánh cân bằng.

3.2 Thuật toán xây dựng cây

Hình 2 dưới đây mô tả cấu trúc cây RG-Tree bao gồm một nút gốc, tập nút trong và tập nút lá. Một nút lá chứa các vec-tơ đặc trưng và định danh của các ảnh. Nút trong chứa các phần tử là vec-tơ đặc trưng của tâm nút con. Trong bài báo này chúng tôi sử dụng véc-tơ trung bình của các vector trong nút con.

```
struct Element_Node{
      double [] f;
      int id;
      string label;
}
struct Node{
    Element_Node [] E;
    Node[] links;
    int count;
    Element_Node [] Element_parent;
    Node parent;
}
```



Hình 2. Cấu trúc cây phân cụm chỉ mục RG-Tree

Thuật toán cập nhật tâm cụm

Quá trình cập nhật tâm cụm để tạo ra một đường đi từ nút gốc đến nút lá nhằm cải tiến thời gian tìm kiếm ảnh của người sử dụng. Do đó, việc cập nhật này được thực hiện từ một nút ***Node*** cho đến nút gốc và thực hiện dựa trên Thuật toán $UCRG$ như sau:

Thuật toán 1: *UCRG*

Đầu vào: nút ***Node***;

Đầu ra: Cây phân cụm RG-Tree sau khi cập nhật

Function *UCRG* (*Node*)

Begin

If (***Node***.Element_parent != null) **then**

f_{cN} = avg{ ***Node***.E[i].f | i=1..count};

Node.Element_parent.f = f_{cN};

EndIf

If (***Node***.parent != null) **then**

Node = ***Node***.parent;

$UCRG$ (***Node***);

EndIf

End

Mệnh đề 1. Thuật toán $UCRG$ có độ phức tạp là $O(M\ x\ h)$, với h, M lần lượt là chiều cao và số phần tử tối đa trong một nút của cây RG-Tree.

Chứng minh: Trường hợp xấu nhất, Thuật toán $UCRG$ phải cập nhật tâm cụm từ nút lá đến nút gốc tức là phải duyệt qua chiều cao h. Mỗi lần cập nhật tâm, Thuật toán $UCRG$ duyệt qua tối đa M phần tử của mỗi nút. Do đó, độ phức tạp của Thuật toán $UCRG$ là $O(M\ x\ h)$■

Thuật toán chèn một phần tử vào cây RG-Tree

Khi một phần tử E_i=<f_i,id> được thêm vào cây RG-Tree, việc chèn được thực hiện bắt đầu từ nút gốc, lần lượt duyệt qua tất cả các nút con của nút gốc và tính khoảng cách Euclide của phần tử E_i với từng tâm cụm, chọn cụm có khoảng cách $d_{min} = Min\{d(f_i, f_{cj}), j = 1..M\}$. Nếu $d_{min} > \theta$ tạo một nhánh mới để lưu phần tử E_i ngược lại thì đi theo nhánh đó và lần lượt duyệt hết các nhánh của cây phân cụm RG-Tree cho đến khi tìm được nút lá. Sau đó, khoảng cách phần tử E_i đến tâm nút lá được tính nếu $d_{min} < \varepsilon$ thì chèn vào nút lá đó ngược lại tạo một nút lá mới đồng cấp để lưu E_i.

Thuật toán 2: *IRG*

Đầu vào: phần tử E_t, nút gốc $root$, giá trị ngưỡng ε, θ
Đầu ra: cây RG-Tree sau khi thêm phần tử E_t
Function IRG($E_t, root, \varepsilon, \theta$)
Begin
$\boldsymbol{Node} = root$;
If ($\boldsymbol{Node} = null$) **then**
Initialize $root = \{links_k | \ links_k = null; k = 1..n_k\}$;
Create new $lvnode =< E_t, links >, links = null$;
$root.links_0 = lvnode$;
$UCRG(lvnode)$;
ElseIf
$i = argmin\{euclide(\boldsymbol{Node}.\mathrm{E}[k].f, E.f), k = 1..\mathrm{count})\}$;
$d = Euclide(\boldsymbol{Node}.\mathrm{E}[i].f, E_t.f)$;
If (***Node***.*links=null*) **then**
If $d < \varepsilon$ **then**
Node.count = ***Node***.count + 1;

Node.*E[count+1].f = Eᵢ.f;*
Node.*E[count+1].id = Eᵢ.id;*
$UCRG(\boldsymbol{Node})$;

ElseIf

If $(d \geq \varepsilon \;\&\&\; d \leq \theta)$ **then**
Create new $lvnode =< E_{lv}, links_{lv} >, links_{lv} = null;$

$E_{lv}.f = E_t.f;$

$E_{lv}.id = E_t.id;$

Create new $innode =< E_{in}, links_{in} >, links_{in} = lvnode$
$\boldsymbol{Node}.links_k = innode;$

$UCRG(lvnode);$

ElseIf

Create new $lvnode =< E_{lv}, links_{lv} >, links_{lv} = null;$
$\boldsymbol{Node}.links_{count+1} = innode;$
Node.count = ***Node***.count + 1;

UCRG($lvnode$);

EndIf

EndIf

ElseIf

If $(d \leq \theta)$

$\boldsymbol{IRG}(E_t, \boldsymbol{Node}.links_k, \varepsilon, \theta);$

ElseIf

Create new $lvnode =< E_{lv}, links_{lv} >, links_{lv} = null;$
$\boldsymbol{Node}.links_{count+1} = innode;$
Node.count = *N*.count + 1;

UCRG($lvnode$);

EndIf

EndIf

Return RG-Tree;

End

Mệnh đề 2. Thuật toán IRG có độ phức tạp là $O(M\ x\ h)$, với h, M lần lượt là chiều cao và số phần tử tối đa trong một nút của cây RG-Tree.

Chứng minh: Thuật toán IRG lần lượt thực hiện duyệt từ nút gốc đến nút lá do đó duyệt qua chiều cao h, mỗi lần duyệt qua M phần tử, mỗi lần duyệt Thuật toán IRG thực hiện phép cập

nhật tâm từ nút lá đến nút gốc tức là cập nhật qua chiều cao h và duyệt lại tối đa M phần tử. Do đó, Thuật toán IRG có độ phức tạp là $O(Mxh)$ ■

Thuật toán xóa phần tử trên cây RG-Tree

Các phần tử chỉ nằm ở nút lá của cây RG-Tree, nếu nút lá đó chỉ gồm 1 phần tử thì nút lá đó được xóa và cập nhật lại tâm cho nút cha. Trường hợp nút lá đó có nhiều hơn 1 phần tử thì phần tử bị xóa khỏi nút và cập nhật lại tâm cho nút lá. Thuật toán xóa một phần tử được thực hiện như sau:

Thuật toán 2: DRG
Đầu vào: phần tử E_t, nút gốc $root$, giá trị ngưỡng ε, θ
Đầu ra: cây RG-Tree sau khi xóa phần tử E_t

Function $DRG(E_t, root, \varepsilon, \theta)$
Begin
$\boldsymbol{Node} = root$;
If $\boldsymbol{Node} = null$ **then**
Return *null;*
ElseIf
$i = argmin\{euclide(\boldsymbol{Node}.\mathrm{E}[k].f, E.f), k = 1..\mathrm{count})\}$;
$d = Euclide(\boldsymbol{Node}.\mathrm{E}[i].f, E.f)$;
If (***Node***.*links=null*) **then**
Node.count = ***Node***.count -1;
Node.*E*= ***Node***.*E*/{E_t};
If (***Node***.count > 0) **then**
*UCRG(**Node**);*
ElseIf
*UCRG(**Node**.parent);*
EndIf
ElseIf
If ($d \le \theta$)
$\boldsymbol{DRG}(E_t, \boldsymbol{Node}.links_k, \varepsilon, \theta)$;
EndIf
End If
End If
Return RG-Tree;
End.

Mệnh đề 2. Thuật toán DRG có độ phức tạp là $O(Mxh)$, với h, M lần lượt là chiều cao và số phần tử tối đa của một nút trong cây RG-Tree.

Chứng minh: Thuật toán DRG được chứng minh tương tự thuật toán IRG. Do đó, Thuật toán DRG có độ phức tạp là $O(Mxh)$ ■

4 Tìm kiếm ảnh tương tự trên cây RG-Tree

4.1 Dữ liệu ảnh

Mỗi ảnh sẽ được chia thành nhiều vùng nhau theo phương pháp của Hugo Jair Escalante [25], mỗi vùng được trích xuất một vec-tơ đặc trưng bao gồm đặc trưng vùng: diện tích, chiều rộng và chiều cao; đặc trưng về vị trí; đặc trưng về hình dạng; đặc trưng màu sắc trong không gian RGB và CIE-Lab.

Hình 3. Ảnh gốc và các ảnh phân vùng (20747.jpg)
Hình 3 mô tả một ảnh gốc và 4 ảnh của các vùng thuộc về các lớp ảnh: mammal-other, group-of-persons, plant, plant của ảnh 20747.jpg trong tập ảnh ImageCLEF.

4.2 Thuật toán tra cứu ảnh tương tự trên cây RG-Tree

Từ cây phân cụm dữ liệu RG-Tree đã tạo, chúng tôi đề xuất thuật toán tra cứu ảnh tương tự. Quá trình tìm kiếm ảnh tương tự được thực hiện trên cây RG-Tree và được mô tả như sau:

Thuật toán 4. RGIR

Đầu vào: vec-tơ đặc trưng f_i của ảnh truy vấn I_q, cây RG-Tree, giá trị ngưỡng ε , θ.
Đầu ra: tập ảnh tương tự SI
Function RGIR(f_i , $root, \varepsilon$, θ)
Begin
$\boldsymbol{Node} = root$;
If $\boldsymbol{Node} = null$ **then**
Return *null*;
ElseIf
$i = argmin\{euclide(\boldsymbol{Node}.\mathrm{E}[k].f, E.f), k = 1..\mathrm{count})\}$;
$d = Euclide(\boldsymbol{Node}.\mathrm{E}[i].f, E.f)$;
If (*N.links=null*) **then**
SI= $\boldsymbol{Node}$.*E;*

Return SI;

ElseIf

If $(d \le \theta)$

RGIR$(f_i, \boldsymbol{Node}.links_k, \varepsilon, \theta)$;

EndIf

End If

End If

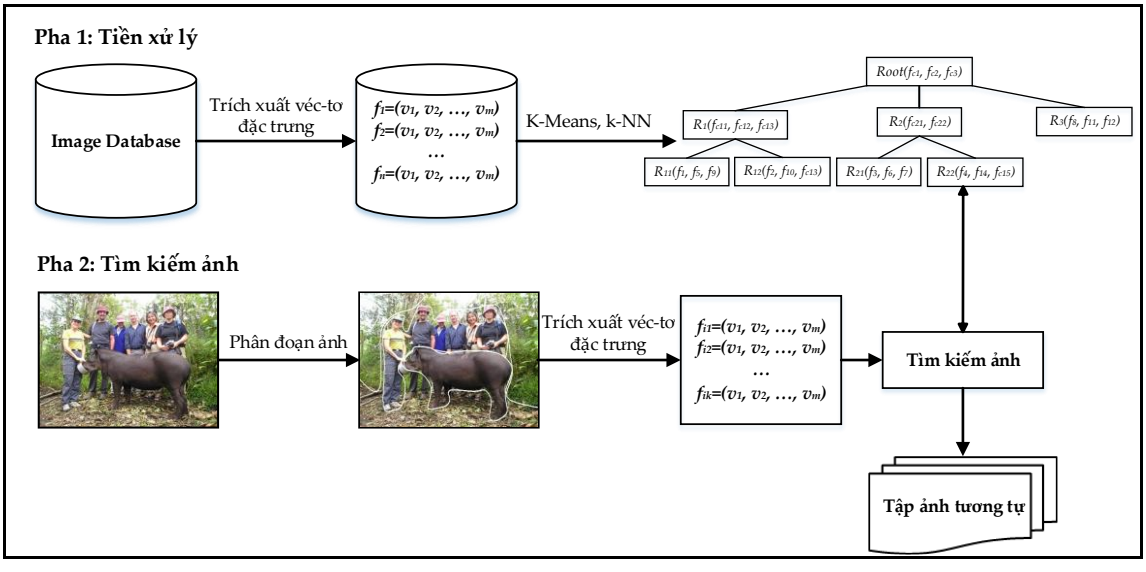
Return SI;

End.

Mệnh đề 4. Thuật toán RGIR có độ phức tạp là $O(hxM)$, với h, M lần lượt là chiều cao và số phần tử tối đa của một nút trong cây RG-Tree.

Chứng minh: Thuật toán RGIR lần lượt duyệt từ gốc đến lá, mỗi lần duyệt một nút sẽ kiểm tra M phần tử của nút đó. Vì vậy, thuật toán RGIR có độ phức tạp là $O(hxM)$ ■

4.3 Mô hình truy vấn ảnh



Hình 4. Mô hình tìm kiếm ảnh theo nội dung dựa trên cây RG-Tree

Quá trình tìm kiếm ảnh được thực hiện gồm hai pha, pha thứ nhất thực hiện gom cụm và lưu trữ trên cây RG-Tree, pha thứ hai thực hiện tìm kiếm các hình ảnh tương tự. Quá trình thực hiện được mô tả như sau:

Pha tiền xử lý: Kết quả của pha tiền xử lý là xây dựng được cây gom cụm RG- Tree dựa trên vector đặc trưng của tập dữ liệu ảnh gồm 2 bước như sau:

Bước 1. Trích xuất tập véc-tơ đặc trưng f_i của tập dữ liệu ảnh.

Bước 2. Dựa trên độ đo tương tự đề xuất và tạo cấu trúc cây gom cụm chỉ mục với mỗi nút lá của cây RG-Tree là tập các véc-tơ f_i mô tả đặc trưng thị giác của hình ảnh.

Pha tìm kiếm: Việc tìm kiếm ảnh tương tự được thực hiện với đầu vào là một hình ảnh truy vấn và đầu ra là tập ảnh tương tự dựa trên cây gom cụm chỉ mục RG-Tree. Quá trình tìm kiếm ảnh tương tự được thực hiện theo các bước như sau:

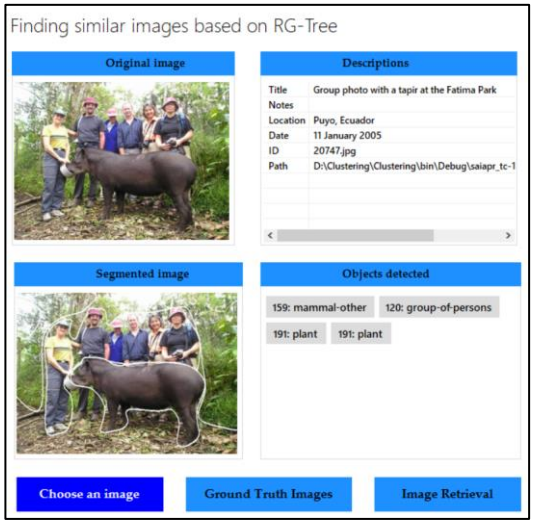
Bước 1. Trích xuất véc-tơ đặc trưng của ảnh cần truy vấn.

Bước 2. Thực hiện truy vấn ảnh tương tự dựa trên cây RG-Tree.

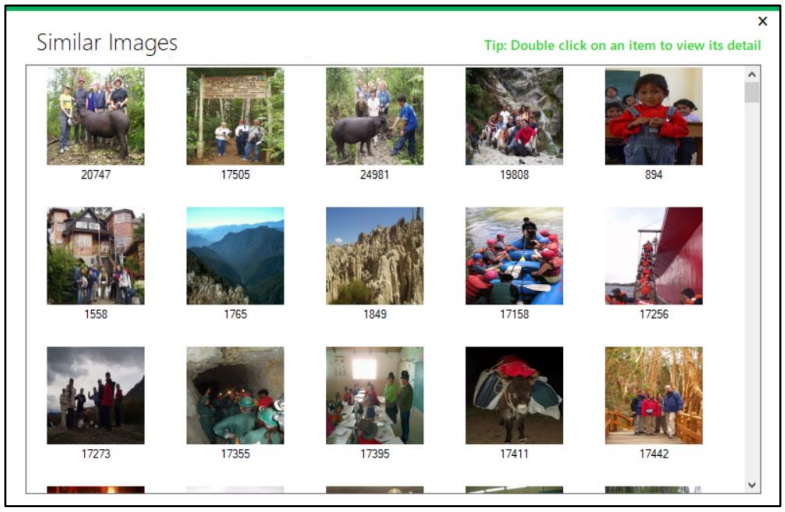
Bước 3. Tra cứu tập ảnh tương tự dựa trên tập chỉ mục đã được truy vấn.

5 Kết quả thực nghiệm

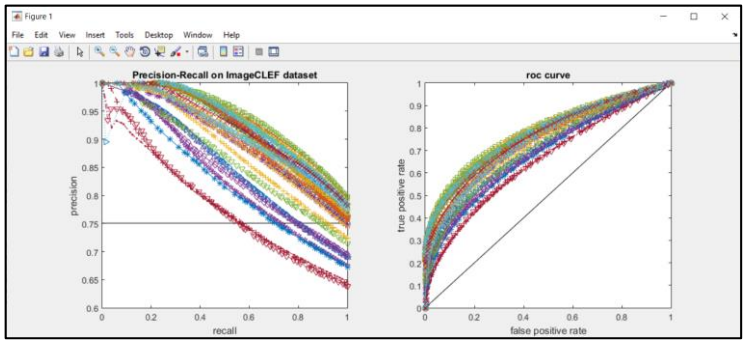
Pha tiền xử lý được thự hiện trên máy PC CPU 2.3GHz 8-core 9th-generation Intel Core i9, 16GB 2666MHz memory, 1TB flash storage. Pha tìm kiếm được thực nghiệm trên máy PC CPU Intel Core i7-6500U CPU @ 2.50GHz, 8.0GB RAM, hệ điều hành Windows 10 Pro 64 bit. Kết quả thực nghiệm được đánh giá trên bộ dữ liệu imageCLEF chứa 20,000 ảnh, được chia thành 276 lớp và lưu trữ trong 41 thư mục (từ thư mục 0 đến thư mục 40); bộ dữ liệu của kích thước 1.64 GB. Để đánh giá hiệu quả của phương pháp tìm kiếm ảnh, phần thực nghiệm được đánh giá các giá trị gồm: độ chính xác (precision), độ phủ (recall) và độ đo dung hòa F-measure.



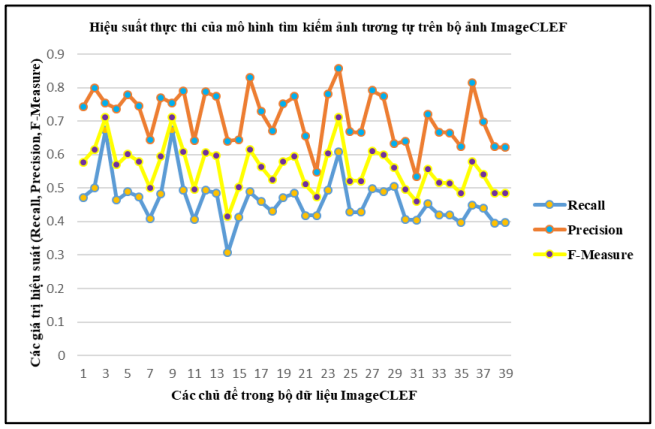
Hình 5. Giao diện truy vấn ảnh dựa trên cây RG-Tree



Hình 6. Một kết quả truy vấn dựa trên cây RG-Tree



Hình 7. Biểu đồ Precision-Recall và ROC của hệ truy vấn trên cây RG-Tree



Hình 8. Giá trị trung bình của Precision, Recall, F-measure của tập dữ liệu ImageCLEF

Bảng 1. Hiệu suất truy vấn ảnh của phương pháp đề xuất trên tập ảnh ImageCLEF

Tập ảnh	Số ảnh	Precision	Recall	F-Measure
00-10	3790	0.7050	0.4555	0.5496
11-20	2198	0.7455	0.4809	0.5812
21-30	1742	0.7262	0.5142	0.5904
31-40	2573	0.6800	0.4355	0.5281
	10303	**0.7110**	**0.4659**	**0.5579**

Bảng 2. So sánh độ chính xác giữa các phương pháp trên bộ dữ liệu ImageCLEF

Phương pháp	Mean Average Precision (MAP)
Hakan Cevikalp, 2017 [18]	0.4678
C.A. Hernández-Gracidas, 2013 [19]	0.5826
Nguyễn Minh Hải, 2019 [14]	0.6753
Nguyễn Thị Uyên Nhi, 2019 [13]	0.6510
Phương pháp đề xuất của chúng tôi	**0.7110**

6 Kết luận

Trong bài báo này, chúng tôi đã xây dựng một cấu trúc cây RG-Tree áp dụng cho bài toán tìm kiếm ảnh tương tự theo nội dung. Đây là một cấu trúc cải tiến của cây R-Tree nhằm nâng cao hiệu quả cho kỹ thuật lưu trữ dữ liệu trên cây và áp dụng được cho bài toán có tập dữ liệu tăng trưởng. Trong bài báo này, cây RG-Tree được cải tiến từ cây R-Tree và thực nghiệm trên bộ ảnh ImageCLEF có độ chính xác là 71.10%, độ phủ 46.59%, độ đo dung hòa 55,79%. Kết quả thực nghiệm đã được so sánh với các công trình khác trên cùng một tập dữ liệu ảnh, đồng thời so sánh với các phương pháp dựa trên cấu trúc lưu trữ cây RG-Tree. Phương pháp đề xuất của chúng tôi đã làm tăng đáng kể hiệu suất truy vấn ảnh theo nội dung. Hướng phát triển tiếp theo, chúng tôi sẽ cân bằng cây RG-Tree và áp dụng phương pháp phân hoạch ngữ nghĩa hình ảnh trên mỗi nút lá của cây để từ đó thực hiện truy vấn ảnh trên cơ sở tiếp cận theo ngữ nghĩa.

Lời cảm ơn

Chúng tôi trân trọng cảm ơn Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế, nhóm nghiên cứu SBIR-HCM, Trường Đại học Sư phạm TP.HCM và Trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh đã hỗ trợ về chuyên môn và cơ sở vật chất để nhóm tác giả hoàn thành nghiên cứu này. Nghiên cứu này do Trường Đại học Công nghiệp thực phẩm Thành phố Hồ Chí Minh bảo trợ và cấp kinh phí theo Hợp đồng số 147/HĐ-DCT.

Tài liệu tham khảo

1. Chandresh Pratap Singh, " R-Tree implementation of image databases ". Signal & Image Processing : An International Journal (SIPIJ) Vol.2, No.4, December 2011.
2. Nam N.V, Bac L.H, " Simple Spatial Clustering Algorithm Based on R-Tree". C. Sombattheera et al. (Eds.): MIWAI 2012, LNCS 7694, pp. 236–245, Berlin Heidelberg, 2012.
3. N. Amoda and R. K. Kulkarni, "Efficient image retrieval using region based image retrieval," in Signal Image Processing : An International Journal (SIPIJ), vol. 4, no. 3, pp. 17–22, 2013.
4. Shama P.S, Badrinath K, Anand Tilugul, " An Efficient Indexing Approach for Content based Image Retrieval". International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 15, May 2015.
5. Van T. T. , Le M. T., "Một số cải tiến cho hệ truy vấn ảnh dựa trên cây S-Tree". Proceeding of Publishing House for Science and Technology, 2017.
6. Maher Alrahhal ; K.P. Supreethi, "Content-Based Image Retrieval using Local Patterns and Supervised Machine Learning Techniques". Amity International Conference on Artificial Intelligence (AICAI), 2019.
7. A Guttman, "R-Tree: a dymanic index structure for spatial searching". In proc. ACM SIGMOD, 1984.
8. Gurchetan Singh, "Introductory guide to Information Retrieval using kNN and KDTree", November 28, 2017.
9. Jayashree Das and Minakshi Gogoi, "Indexing of Voluminous Data Using K-D Tree with Reference to CBIR", International Journal of Computer Sciences and Engineering, Volume-4, Special Issue-7, Dec 2016.
10. Hasan Al-Jabbouli, "Data clustering using the Bees Algorithm and the Kd-Tree structure", Intelligent Systems Research Laboratory, Manufacturing Engineering Centre, Cardiff University, United Kingdom, 2009.
11. Vignesh Ramanathan, Shaunak Mishra and Pabitra Mitra, "Quadtree Decomposition based Extended Vector Space Model for Image Retrieval", Indian Institute of Technology, Kharagpur - 721302, 978-1-4244-9497-2/10/ IEEE, 2010.
12. A. Alzu'bi, A. Amira, N. Ramzan, "Semantic Content-based Image Retrieval: A Comprehensive Study", J. Vis. Commun. Image R, 2015.
13. Nguyễn Thị Uyên Nhi, Văn Thế Thành, Lê Mạnh Thạnh, A Self-Balanced Clustering Tree apply for Semantic-Based Image Retrieval, Fundamental and Applied IT Reseach (FAIR), Hue University, NXB Khoa học Tự nhiên và Công nghệ, ISBN: tr.xx-xx, 2019.
14. Nguyễn Minh Hải, Lê Thị Vĩnh Thanh, Văn Thế Thành, Trần Văn Lăng, Tra cứu ảnh theo ngữ nghĩa dựa trên cây phân cụm phân cấp, Kỷ yếu Hội thảo Quốc gia về Nghiên cứu cơ bản và ứng dụng CNTT (FAIR), ĐH Huế, Nhà xuất bản Khoa học Tự nhiên và Công nghệ, ISBN: xx, tr.xx-xx, 2019.

15. Thanh Manh Le, Thanh The Van - Image retrieval system based on emd similarity measure and S-Tree, ICITES-2012, Springer Verlag, LNEE 234, 139-146, 2013.
16. N.V.T. Thanh The Van, Thanh Manh Le, "The Method Proposal of Image Retrieval Based on K-Means Algorithm", Advances in Intelligent Systems and Computing, vol. 746, no. 2, pp. 481–490, 2018.
17. Nguyễn Phương Hạc, Văn Thế Thành, "Một phương pháp tra cứu dữ liệu ảnh dựa trên cây phân cụm đa nhánh cân bằng", Tạp chí Khoa học Công nghệ và Thực phẩm 18 (1), 140-153, 2019.
18. H. Cevikalp, M. Elmas, S. Ozkan (2017), "Large-scale image retrieval using transductive support vector machines", Computer Vision and Image Understanding, vol. no. pp.1-11.
19. C.A. Hernández-Gracidas, Sucar, L.E. & Montes-y-Gómez (2013), "Improving image retrieval by using spatial relations", Multimed Tools Application, vol. 62, no. 2, pp. 479–505.

Abstract. In this paper, we present an improvement of R-Tree, denoted RG-Tree (Region Growth Tree), to improve the efficiency of finding content-based similar images. In this improvement, the feature vectors of the images are stored on each leaf node of the RG-Tree by the proposed partition rules. RG-Tree can grow to store and distribute data areas on the leaf nodes to create data clusters. The more elements are similar, the more elements belong to one of the branches on the RG-Tree. We designed an image retrieval model on RG-Tree and tested it on ImageCLEF image sets and compared it's retrieval performance to several recent methods on the same data set.

Keywords: RG-Tree, CBIR, Similar Images, Similarity Measure, Image Retrieval.