



MỘT PHƯƠNG PHÁP PHÂN CỤM DỰA TRÊN CÂY KD-TREE CHO BÀI TOÁN TÌM KIẾM ẢNH

Nguyễn Thị Định¹, Lê Thị Vĩnh Thanh², Nguyễn Văn Thịnh¹, Văn Thế Thành^{3*}

¹ Khoa Công nghệ Thông tin, Trường ĐH Công nghiệp Thực phẩm TP.HCM

² Viện Công nghệ Thông tin – Điện - Điện tử, Trường Đại học Bà Rịa – Vũng Tàu

³ Phòng Quản lý khoa học và Đào tạo sau đại học, Trường ĐH Công nghiệp Thực phẩm TP.HCM

Tóm tắt. Trong bài báo này, chúng tôi trình bày một mô hình tìm kiếm ảnh dựa trên phân cụm dữ liệu bằng cây BKD-Tree, một cải tiến cải tiến của cây KD-Tree, gồm: (1) lưu trữ các đối tượng đa chiều tại nút lá để tạo ra một sự phân cụm trên cơ sở phương pháp học bán giám sát; (2) tạo ra một cấu trúc cây cân bằng nhằm tăng hiệu suất cho bài toán tìm kiếm ảnh. Dựa trên cơ sở lý thuyết đề nghị, mô hình truy vấn ảnh trên cây BKD-Tree được đề xuất và thực nghiệm trên bộ ảnh ImageCLEF (gồm 20.000 ảnh). Kết quả thực nghiệm của chúng tôi được so sánh với một số công trình gần đây trên cùng bộ dữ liệu để minh chứng tính hiệu quả của phương pháp đã được đề xuất. Theo kết quả thực nghiệm cho thấy phương pháp của chúng tôi là hiệu quả và có thể áp dụng được cho các hệ thống tìm kiếm ảnh tương tự theo nội dung.

Từ khóa: BKD-Tree, độ đo tương tự, phân cụm, ảnh tương tự, tìm kiếm ảnh

1 Giới thiệu

Trong những thập niên gần đây, cùng với sự phát triển nhanh chóng của kho dữ liệu ảnh, các kỹ thuật tìm kiếm cũng được quan tâm nghiên cứu và tập trung theo 3 hướng chính: tìm theo từ khóa TBIR (Text-based Image Retrieval), tìm theo nội dung CBIR (Content-based Image Retrieval) hay tìm theo ngữ nghĩa SBIR (Semantic-based Image Retrieval) [8], [9]. Trong tìm kiếm ảnh, vấn đề gom cụm dữ liệu theo các chủ đề là một yêu cầu quan trọng. Ngày nay, nhiều phương pháp gom cụm dữ liệu được thực hiện bằng nhiều thuật toán khác nhau, trong đó kỹ thuật gom cụm sử dụng cây KD-Tree cho kết quả khá tốt.

Cây KD-Tree là một cấu trúc dữ liệu được sử dụng để đánh chỉ mục đa chiều, là một cấu trúc dữ liệu phân vùng không gian tổ chức thành những điểm trong không gian. Cây KD-Tree thuộc dạng cây nhị phân tìm kiếm mà mỗi nút là một véc-tơ k -chiều. Mỗi nút không phải là nút lá thì chia không gian dữ liệu thành hai phần trên mặt phẳng k -chiều. Dựa trên cây KD-Tree nguyên thủy này, chúng tôi xây dựng cây BKD-Tree là cây nhị phân cân bằng để ứng dụng cho bài toán tìm kiếm ảnh và thực nghiệm trên bộ ảnh ImageCLEF. Cây BKD-Tree được dùng để lưu trữ các véc-tơ đặc trưng thị giác của hình ảnh đã phân đoạn. Việc phân lớp dữ liệu được thực

* Liên hệ: nguyenthidinh.hcm@gmail.com

hiện trên từng nút của cây BKD-Tree để tạo ra một cây cân bằng nhằm hỗ trợ cho quá trình tìm kiếm nhanh và tăng độ chính xác.

2 Các công trình liên quan

Năm 2002, Y. He và cộng sự đã khảo sát sử dụng cây KD-Tree trong nâng cao hiệu quả tìm kiếm ảnh. Nhóm tác giả thực nghiệm trên bộ dữ liệu 10.115 ảnh. Kết quả thu được là thời gian truy vấn ảnh nhanh gấp ba lần so với cách tìm kiếm tuyến tính [1].

Năm 2007, S. J. Redmond kết hợp thuật toán k-Means và thuật toán Katsavounidis, thực nghiệm trên 36 bộ dữ liệu tổng hợp và 2 bộ dữ liệu của UCI (UC Irvine Machine Learning Repository). Thuật toán này đã cải thiện quá trình phân cụm, là cơ sở cho việc nghiên cứu mở rộng cây BKD-Tree [15].

Năm 2009, H. Al-Jabbouli đã đề xuất thuật toán gom cụm dựa vào cấu trúc cây KD-Tree [13]. Thuật toán này khắc phục những nhược điểm của thuật toán K-means, đồng thời kết hợp thuật toán K-means và C-means để tìm ra phương pháp gom cụm tối ưu gọi là thuật toán Bees. Kết quả thực nghiệm trên nhiều bộ dữ liệu cho thấy thuật toán Bees hiệu quả hơn thuật toán k-Means. Năm 2011, K. Velmurugan đã đề xuất thuật toán tìm kiếm ảnh tương tự theo nội dung dựa trên cây KD-Tree bằng cách kết hợp đặc trưng SURF (Speed up robust feature) với đặc trưng màu sắc để nâng cao độ chính xác cho hệ tìm kiếm ảnh [2].

Năm 2011, H. Zouaki và B. Abdelkhalak dựa trên cấu trúc cây KD-Tree đã đề xuất mô hình truy vấn ảnh bằng cách lập chỉ mục nhằm giảm kích thước của đối tượng, giảm thời gian tính toán, sử dụng khoảng cách EMD. Kết quả thực nghiệm cho độ chính xác khá cao [6]. Năm 2016, J. Das và cộng sự đã đề xuất một phương pháp lập chỉ mục và xây dựng hệ thống truy vấn ảnh dựa trên cây KD-Tree k -chiều bằng cách trích xuất đặc trưng màu sắc của đối tượng ảnh. Phương pháp này đã giảm đáng kể thời gian truy vấn ảnh trên cây KD-Tree. Kết quả thực nghiệm cho thấy cây KD-Tree có thời gian tìm kiếm nhanh hơn cây tứ phân QuadTree và kết quả tìm kiếm trên cây KD-Tree có độ chính xác cao hơn [7].

Năm 2013 M. Otair đã đề xuất một phương pháp gom cụm dữ liệu dựa trên cây KD-Tree với thuật toán k-NN. So sánh với các phương pháp trước đó thì thời gian tìm kiếm giảm đáng kể [3]. Năm 2014 Logamani. K và Punitha. S. C đã đưa ra mô hình phân cụm dữ liệu dựa trên cây KD-Tree. Nhóm tác giả thực hiện gom cụm bằng phương pháp k-Means đồng thời xây dựng cây KD-Tree là một cây tăng trưởng nên việc xây dựng cây mất nhiều thời gian nhưng thích hợp cho bài toán có dữ liệu gia tăng [4].

Năm 2015, Y. H. Sharath Kumar giới thiệu mô hình lập chỉ mục các đối tượng. Trong nghiên cứu này, cơ sở dữ liệu đầu vào rất lớn nên thời gian truy xuất lớn. Một giải pháp cho tăng tốc quá trình truy xuất là thiết kế mô hình lập chỉ mục. Cách lập chỉ mục của cây KD-Tree cho hệ

thống truy xuất dữ liệu dựa trên đặc trưng SIFT (Scale Invariant Feature Transform), biểu đồ phân lớp HOG (Histogram of Gradients), biểu đồ hướng cạnh EOH (Edge orientation histograms) và hình dạng SC (Shape context) [5].

Năm 2017, H. Cevikalp đã đề xuất phương pháp truy vấn ảnh bằng cách sử dụng cây nhị phân phân cấp kết hợp với máy vectơ hỗ trợ chuyển đổi (TSVM). Hệ thống thực nghiệm đánh giá trên bộ dữ liệu ImageCLEF. Hiệu suất truy vấn thu được với độ chính xác là 46.78% [18]. Cùng thời điểm đó, M. Jiu đã đề xuất một phương pháp mới để truy vấn ảnh bằng cách kết hợp mạng nhiều lớp học sâu đồng thời kết hợp kỹ thuật máy hỗ trợ vec-tơ (SVM) để phân lớp hình ảnh. Phương pháp này được thực nghiệm trên bộ dữ liệu ImageCLEF với hiệu suất truy vấn độ chính xác là 59.70% [19].

3 Cấu trúc cây BKD-Tree

3.1 Mô tả cây BKD-Tree, một cải tiến của KD-Tree

Cây KD-Tree là một cây nhị phân mà dữ liệu tại mỗi nút là một điểm k chiều trong không gian, bao gồm: một nút gốc, các nút trong và nút lá. Nút gốc là nút chỉ có liên kết tối đa đến hai nút con và không có nút cha. Nút trong là nút có một liên kết đến nút cha, đồng thời liên kết tối đa đến hai nút con; nút lá là nút chỉ có một liên kết đến nút cha và không có liên kết đến nút con. Một nút không phải là nút lá trên cây chia không gian thành hai phần, các điểm ở bên trái của không gian này biểu thị bằng cây con trái và các điểm ở bên phải không gian này biểu thị bằng cây con phải. Cây KD-Tree lưu dữ liệu tại tất cả các nút, do đó khi áp dụng cho bài toán có tập dữ liệu tăng trưởng sẽ làm cây tăng trưởng theo chiều sâu và cây mất cân bằng. Vì vậy, chúng tôi xây dựng cây BKD-Tree cân bằng là một cải tiến của cây KD-Tree, để thời gian tìm kiếm trên các nút lá là gần như nhau và dữ liệu chỉ lưu trữ tại nút lá; đồng thời xây dựng mô hình tìm kiếm ảnh tương tự dựa trên kỹ thuật xây dựng cây BKD-Tree cải tiến đã đề xuất. Cấu trúc dữ liệu này tạo ra một mô hình lưu trữ tập dữ liệu ảnh, phân cụm tự động các phần tử trên nút lá và tìm kiếm phần tử trên cây.

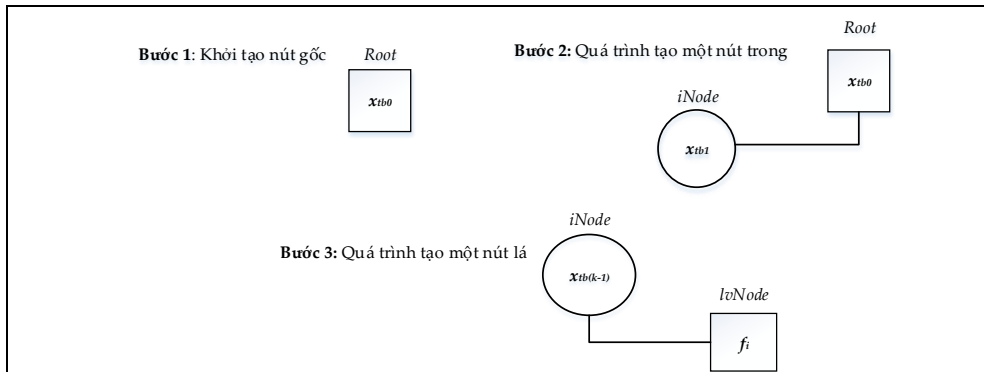
Cây BKD-Tree được xây dựng gồm một nút gốc (*root*), các nút trong (*iNode*) và các nút lá (*lvNode*). Nút lá là nơi lưu dữ liệu đồng thời đóng vai trò để phân cụm dữ liệu. Gọi $\{f_1, f_2, \dots, f_n\}$ là tập véc-tơ đặc trưng của bộ dữ liệu ảnh ban đầu. Trong đó $f_j = (x_{j0}, x_{j1}, \dots, x_{j(k-1)})$; $V_{tb} = (x_{tb0}, x_{tb1}, \dots, x_{tb(k-1)})$ là véc-tơ trung bình mà mỗi thành phần x_{tbi} là giá trị trung bình của thành phần x_{ji} thuộc tập véc-tơ f_j , cấu trúc nút gốc, nút trong và nút lá được định nghĩa như sau:

Định nghĩa: Cây BKD-Tree là một cây nhị phân cân bằng gồm:

- i. Nút gốc là nút không có nút cha và có tối đa hai nút con *left, right*. Cấu trúc của nút gốc là $root = \langle x, left, right \rangle$. Trong đó $x = x_{tb0}$; *left, right* lần lượt là liên kết đến cây con trái, phải.

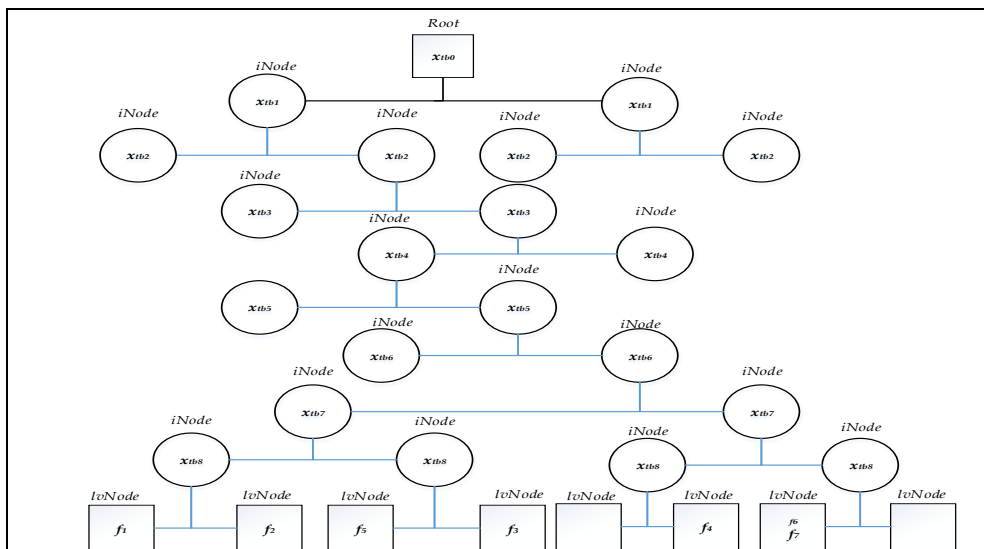
- ii. Nút trong $iNode = \langle x, ileft, iright \rangle$, với $x = x_{tbl}$ là một giá trị tại mức thứ l .
- iii. Nút lá $lvNode = \langle f_i, id_i \rangle$ là nút không có nút con, trong đó f_i, id_i lần lượt là véc-tơ đặc trưng và định danh của ảnh thứ i .

Cây BKD-Tree dùng để lưu trữ và truy vấn tập ảnh tương tự trên bộ ảnh ImageCLEF được xây dựng chiều cao m thì số nút lá trên cây là 2^m . Bộ ảnh ImageCLEF được chia thành 276 phân lớp khác nhau. Chiều cao của cây cần thiết để lưu trữ tập dữ liệu là $\log_2 276 \approx 8.1$. Do đó chúng tôi chọn 9 thuộc tính của véc-tơ đặc trưng ảnh để tiến hành xây dựng cây BKD-Tree. Minh họa các bước xây dựng cây BKD-Tree từ tập các véc-tơ f_j . Trong đó $f_1 = (x_{10}, x_{11}, \dots, x_{1(k-1)})$ và véc-tơ trung bình $V_{tb} = (x_{tb0}, x_{tb1}, \dots, x_{tb(k-1)})$.



Hình 1. Minh họa các bước tạo cây BKD-Tree

Tương tự, thêm các véc-tơ f_1, f_2, \dots, f_8 với các giá trị cụ thể, ta có cây BKD-Tree như sau:



Hình 2. Một ví dụ về cây BKD-Tree

3.2 Xây dựng cây BKD-Tree

Trên cơ sở Định nghĩa các nút trên cây, quá trình tạo cây BKD-Tree được mô tả như sau:

Bước 1. Tại thời điểm ban đầu cây BKD-Tree chỉ có một nút gốc $root = \emptyset$, mức $l_0 = 0$;

Bước 2. Tính véc-tơ trung bình $V_{tb} = (x_{tb0}, x_{tb1} \dots, x_{tb(k-1)})$, gán giá trị nút gốc là $root.x = x_{tb0}$

Bước 3. Gán giá trị tại mỗi nút trong mức l_i là x_{tbi} ;

Bước 4. Với mỗi véc-tơ $f_j = (x_{j0}, x_{j1} \dots, x_{j(k-1)})$ lần lượt so sánh giá trị x_{jl} ($l = 1 \dots (k-1)$) với giá trị x_{tbi} ($i = 0 \dots (k-1)$). Nếu $x_{jl} \geq x_{tbi}$ thì véc-tơ f_j thuộc cây con bên phải; ngược lại f_j thuộc cây con bên trái. Quá trình này lặp lại cho đến khi tìm được nút lá để lưu trữ véc-tơ f_j .

Cây BKD-Tree có chiều cao $h = m$, tại mỗi nút trong luôn tồn tại nút con trái và nút con phải, ở mức $(m-1)$ mỗi nút trong lưu trữ hai nút lá. Bên cạnh đó, chiều cao của cây con trái và chiều cao cây con phải $h_l = h_r = m$. Vậy BKD-Tree được xây dựng là một cây nhị phân cân bằng, các phần tử trên cùng một nút lá thì phù hợp về độ đo tương tự, nghĩa là các phần tử thuộc một nút lá có độ tương tự nhiều hơn so với các phần tử khác nút lá. Mỗi véc-tơ đặc trưng f_i của ảnh I có tính chất sau:

Định lý. Tồn tại duy nhất một nút lá trên cây BKD-Tree để lưu trữ véc-tơ đặc trưng f_i .

Chứng minh:

Tính tồn tại: Vì BKD-Tree là cây nhị phân cân bằng, dữ liệu chỉ lưu trữ tại nút lá. Do đó, tại mỗi nút trong của cây BKD-Tree luôn tồn tại một nhánh con trái hoặc một nhánh con phải để véc-tơ f_i tìm đến vị trí của nút lá. Do đó, luôn tồn tại nút lá để lưu trữ véc-tơ f_i .

Tính duy nhất: Duyệt từ nút gốc của cây, tại mỗi nút trong thuộc cây BKD-Tree, ta chỉ chọn được duy nhất một hướng (trái hoặc phải) để tìm vị trí lưu trữ véc-tơ đặc trưng f_i . Do đó, ta chỉ chọn được một nút lá để lưu trữ véc-tơ f_i , nghĩa là véc-tơ f_i chỉ thuộc về một nút lá duy nhất.

3.3 Thuật toán tạo cây BKD-Tree

Input: Tập phần tử f_v cần thêm vào cây BKD-Tree;

Output: Cây BKD-Tree;

Function IEBKT ($f_v, BKD - Tree, m$)

Begin

$BKD - Tree = \emptyset$;

$V_{tb} = (x_{tb0}, x_{tb1} \dots, x_{tbl})$; $x_{tbi} = avg \{x_{ji} : i = 0 \dots (m-1); j = 0 \dots (m-1)\}$;

$root.x = x_{tb0}; l_0 = 0;$

If $(f_v \cdot x_{vi} < V_{tb} \cdot x_{tbi})$ **then** $BKD-Tree = BKD-Tree \cup \text{IEBKT}(f_v, BKD-Tree.left, m+1)$

Else $BKD-Tree = BKD-Tree \cup \text{IEBKT}(f_v, BKD-Tree.right, m+1)$

EndIf

Return $BKD-Tree;$

End.

Mệnh đề 1. Độ phức tạp của thuật toán IEBKT là $O(n)$. Với n số phần tử trên cây BKD-Tree.

Chứng minh: Gọi m, n lần lượt là chiều cao và số phần tử cần chèn vào cây BKD-Tree. Mỗi phần tử f_v cần chèn vào cây, thuật toán IEBKT duyệt qua m mức của cây nhằm tìm được nút lá phù hợp để lưu trữ phần tử. Do đó, số phép toán so sánh của thuật toán là $m \cdot n$. Trong trường hợp này, chiều cao của cây BKD-Tree giới hạn là $m = k$ (với k là một số cho trước). Vậy độ phức tạp của Thuật toán IEBKT là $O(n)$ ■

4 Truy vấn ảnh dựa trên cây BKD-Tree

4.1 Mô hình truy vấn ảnh dựa trên cây BKD-Tree

Các vùng ảnh phân đoạn của ảnh 2092.JPG được minh họa như **Hình 3**.



Hình 3. Ảnh gốc và các vùng của ảnh đã phân đoạn

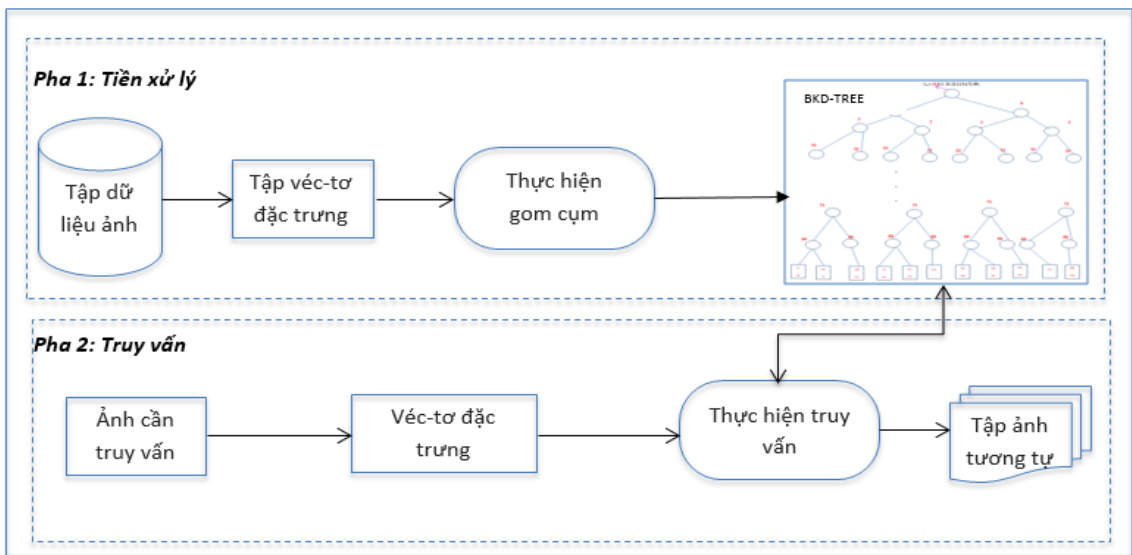
Trên cơ sở xây dựng thuật toán tạo cây BKD-Tree (IEBKT), thuật toán truy vấn ảnh trên cây được thực hiện bằng cách: Mỗi vùng ảnh được trích xuất một véc-tơ đặc trưng, quá trình truy vấn để tìm ra tập ảnh tương tự của hình ảnh được thực hiện dựa vào véc-tơ đặc trưng từng vùng của ảnh cần truy vấn. Hệ thống thực hiện tìm kiếm ảnh tương tự bằng cách so sánh véc-tơ đặc trưng $f = (x_0, x_1, \dots, x_{(m-1)})$ của mỗi vùng ảnh truy vấn với các thành phần tương ứng của véc-tơ trung bình $V_{tb} = (x_{tb0}, x_{tb1}, \dots, x_{tb(m-1)})$ trên cây BKD-Tree theo hướng từ nút gốc đến nút lá và theo quy tắc xây dựng cây BKD-Tree. Kết quả của quá trình này là tập hợp các nút lá chứa véc-tơ đặc trưng của từng vùng thuộc ảnh cần truy vấn.

Thuật toán IRBKD (truy vấn ảnh trên cây BKD-Tree)**Input:** Tập véc-to f_i của ảnh I_Q , cây BKD-Tree**Output:** Tập nút lá L_i chứa phần tử f_i **Function IRBKD** ($f_i, BKD\text{-Tree}, m$)**Begin** $Result = \emptyset ;$ **Foreach** ($f_i \in f$) **do****If** ($f_i \in leaf$) **then** $Result = Result \cup L_i;$ **ElseIf****If** ($f_i \cdot x_{im} < V_{tb} \cdot x_{tbm}$) **then** **IRBKD** ($f_i, BKD\text{-Tree}.left, m+1$);**ElseIf** **IRBKD** ($f_i, BKD\text{-Tree}.right, m+1$);**EndIf****End Foreach****Return** $Result;$ **End.**

Mệnh đề 2. Độ phức tạp của thuật toán IRBKD là $O(\log(n))$. Với n là số phần được lưu trữ trên cây BKD-Tree.

Chứng minh: Thuật toán IRBKD lần lượt duyệt qua m mức của cây từ nút gốc đến nút lá thì độ phức tạp là $O(m)$. Tại mỗi mức trên cây thực hiện so sánh giá trị tại nút *trong* của cây với thành phần thứ i của véc-tơ f_j . Hơn nữa BKD-Tree là cây nhị phân tìm kiếm cân bằng, chiều cao là $m = k$ (với k là một số cho trước). Do đó, với cây BKD-Tree có n phần tử thì độ phức tạp khi tìm kiếm một phần tử trên cây là $O(\log(n))$. ■

Mô hình tìm kiếm ảnh tương tự dựa trên cây BKD-Tree được đề xuất như sau:



Hình 4. Mô hình truy vấn ảnh theo nội dung dựa trên cây BKD-Tree

Pha tiền xử lý:

Bước 1. Trích xuất tập véc-tơ đặc trưng f_v của bộ dữ liệu ảnh, mỗi véc-tơ đặc trưng có n thành phần, mỗi thành phần là một giá trị mô tả một đặc trưng của hình ảnh.

Bước 2. Gom cụm các véc-tơ đặc trưng f_v trên cây BKD-Tree. Nghĩa là, mỗi véc-tơ đặc trưng $f_v = (x_{v0}, x_{v1}, \dots, x_{v(k-1)})$, thành phần x_{vi} được so sánh với giá trị nút *trong* tại mức i . Nếu $x_{vi} \geq x_{tbi}$ thì f_v thuộc cây con bên phải; ngược lại f_v thuộc cây con bên trái. Do đó, véc-tơ f_v được chọn hướng ngay nút *lá* tại mỗi mức trên cây.

Pha truy vấn:

Bước 1. Với mỗi hình ảnh truy vấn, trích xuất các véc-tơ đặc trưng f_{vi} . Trong đó f_{vi} là véc-tơ có n thành phần, mỗi thành phần là một giá trị mô tả đặc trưng của ảnh phân đoạn.

Bước 2. Sử dụng véc-tơ f_{vi} truy vấn ảnh tương tự trên cây BKD-Tree. Kết quả của quá trình truy vấn này là một tập Q_I chứa các véc-tơ tương tự với véc-tơ f_{vi} .

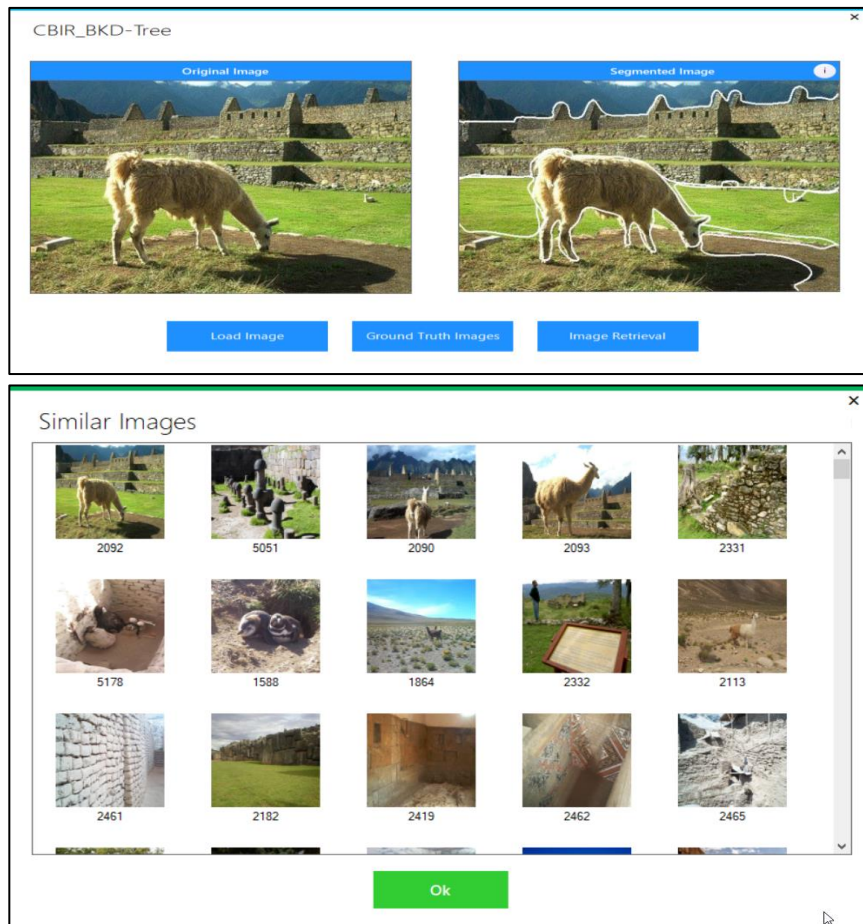
Bước 3. Truy hồi tập ảnh tương tự dựa trên tập Q_I tại bước 2. Sau đó sắp xếp tập các ảnh tương tự với ảnh cần truy vấn.

4.2 Tổ chức thực nghiệm

Để đánh giá phương pháp tiếp cận của bài toán trên cơ sở các thuật toán đã đề xuất, chúng tôi xây dựng thực nghiệm trên bộ dữ liệu ImageCLEF được lưu trữ trong 41 thư. Mỗi ảnh được

phân đoạn thành các vùng, có một véc-tơ đặc trưng và thuộc một phân lớp. Bộ dữ liệu có 99.535 vùng, phân thành 276 lớp.

Thực nghiệm truy vấn ảnh CBIR-BKD-Tree được xây dựng trên nền tảng dotNET Framework 4.5, ngôn ngữ lập trình C#. Các đồ thị được xây dựng trên Matlab 2015. Cấu hình máy tính thực nghiệm: Intel(R) Core™ i5-5200U, CPU 2.2GHz, RAM 8GB và hệ điều hành Windows 10 Professional.



Hình 5. Hệ truy vấn ảnh CBIR_BKD-Tree

Để đánh giá hiệu suất của phương pháp, các giá trị thực nghiệm gồm: độ chính xác (precision), độ phủ (recall) và độ đo dung hòa F-measure. Công thức tính các giá trị này như sau:

$$\text{precision} = \frac{|\text{relevant images} \cap \text{retrieved images}|}{|\text{retrieved images}|}$$

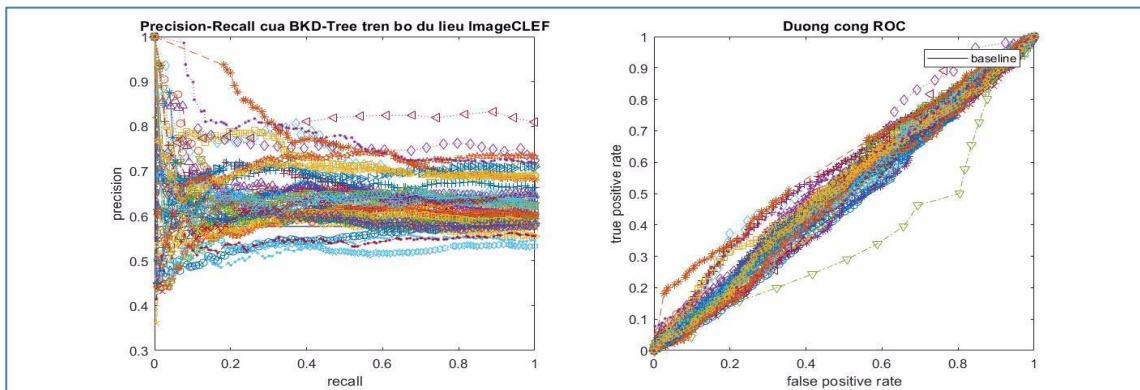
$$\text{recall} = \frac{|\text{relevant images} \cap \text{retrieved images}|}{|\text{relevant images}|}$$

$$F - measure = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

Trong đó, relevant images là tập ảnh tương tự với ảnh truy vấn, retrieved images là tập ảnh tìm kiếm được. Các giá trị độ chính xác, độ phủ và độ dung hòa được tính theo tỷ lệ % và được quy đổi thành giá trị trên đoạn [0, 1]. Kết quả thực nghiệm thể hiện trong **Bảng 1**.

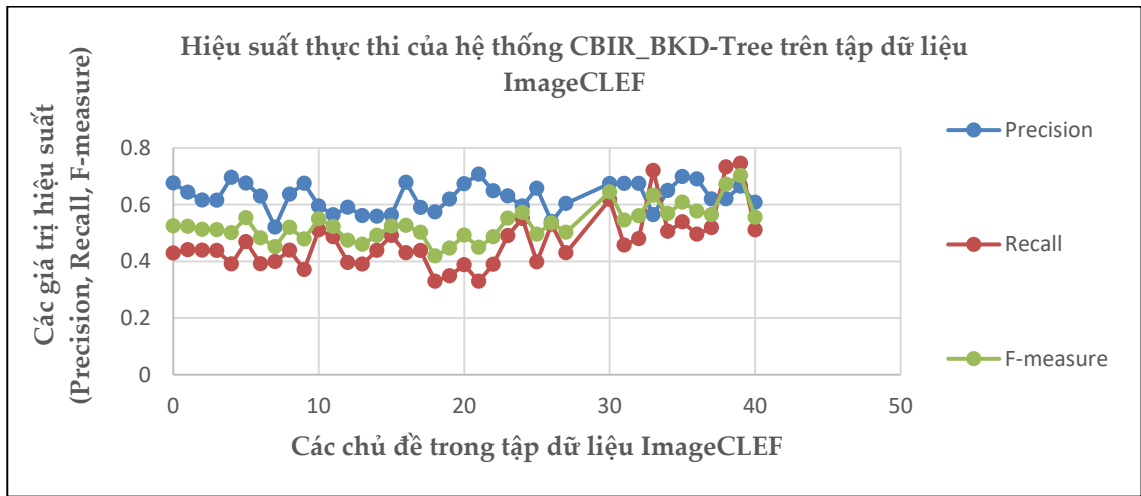
Bảng 1. Hiệu suất truy vấn ảnh theo phương pháp đề xuất trên bộ ImageCLEF

Chủ đề	Số ảnh	Trung bình độ chính xác	Trung bình độ phủ	Trung bình độ dung hòa	Thời gian truy vấn trung bình (ms)
00-10	1832	0.598713	0.446594	0.503931	30.73533
11-20	1796	0.586611	0.434149	0.498994	29.60023
21-30	1957	0.615845	0.488197	0.544642	24.51016
31-40	2194	0.698048	0.520518	0.599148	27.50432
AVG	7779	0.624804	0.472365	0.536815	28.08751



Hình 6. Đồ thị Precision, Recall, đường cong ROC của hệ CBIR_BKD-Tree trên bộ ImageCLEF

Đồ thị Precision-Recall bộ dữ liệu ImageCLEF cho thấy diện tích nằm dưới đường cong Precision-Recall chưa cao, do tính chính xác của hệ truy vấn nằm tập trung ở vùng 0.4 đến 0.7. Đồ thị đường cong ROC biểu diễn các giá trị true positive và false positive chủ yếu nằm tập trung trên đường baseline.



Hình 7. Trung bình Precision, Recall, F-measure hệ CBIR_BKD-Tree trên tập dữ liệu ImageCLEF

Các đồ thị trong Hình 6 và Hình 7 mô tả giá trị trung bình của độ chính xác, độ phủ, độ dung hoà. Từ các đồ thị này cho thấy, tính chính xác của truy vấn nằm ở mức trung bình, cần cải thiện thêm để nâng cao hiệu suất truy vấn. Độ phủ của truy vấn khá thấp nên độ dung hoà của truy vấn chưa cao. Tuy nhiên tốc độ truy vấn khá nhanh, cho thấy hệ thống truy vấn ảnh CBIR_BKD-Tree được đánh giá là khá tốt về thời gian truy vấn. Giá trị trung bình độ chính xác MAP của hệ truy vấn CBIR_BKD-Tree được so sánh với các phương pháp khác trên cùng bộ dữ liệu ImageCLEF, thể hiện trong Bảng 2.

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

Phương pháp	Giá trị trung bình độ chính xác (MAP)
H. Cevikalp, 2017 [20]	0.4678
M. Jiu, 2017 [21]	0.5970
CBIR_BKD-Tree	0.6248

Từ kết quả so sánh ở Bảng 2, cho thấy hệ truy vấn hình ảnh CBIR-BKD-Tree có độ chính xác khá tốt so với các nghiên cứu gần đây trong cùng lĩnh vực trên thực nghiệm cùng bộ dữ liệu.

5 Kết luận

Trong bài báo này, chúng tôi đã xây dựng một cấu trúc cây BKD-Tree áp dụng cho bài toán tìm kiếm ảnh tương tự. Đây là một cải tiến của cây KD-Tree nguyên thủy nhằm nâng cao kỹ thuật lưu trữ dữ liệu trên cây, đồng thời cải thiện thời gian tìm kiếm đến các nút lá gần bằng nhau.

Chúng tôi đã đề xuất mô hình truy vấn ảnh dựa trên cây BKD-Tree và thực nghiệm trên bộ ảnh ImageCLEF có độ chính xác là 62.48%, độ phủ 47.23%, độ dung hòa 53.68% và thời gian truy vấn trung bình là 28.08 (ms). 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 KD-Tree. Thực nghiệm cho thấy tính đúng đắn và hiệu quả của phương pháp đề xuất. Hướng phát triển tiếp theo của chúng tôi là tạo một cây chỉ mục với mỗi nút liên kết tới một phần tử của bảng tra cứu, với bảng tra cứu này được xây dựng một cách độc lập với cây BKD-Tree để từ đó tăng tính hiệu quả trong việc phân lớp trên cây BKD-Tree.

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. Y. He, G.Lu and S. Teng, "An Investigation of Using K-d Tree to Improve Image Retrieval Efficiency". Digital Image Computing Techniques and Application, 21--22 January 2002, Melbourne, Australia.
2. B.S. Banerjee M., Pal S.K, "Content-Based Image Retrieval using SURF and Colour Moments", Global Journal of Computer Science and Technology, Volume XI Issue X Version. [\[1\]](#)
3. J. Mohammed Otair, "Approximate K-Nearest Neighbour Based Spatial Clustering Using K-D Tree". International Journal of Database Management Systems (IJDMMS) Vol.5, No.1, February 2011.
4. L. K. Punitha. S. C, "Density Based Clustering using Enhanced KD Tree", International Journal of Computer Science Engineering and Technology (IJCSET), Vol 4, Issue 11, 314-318, November 2014.
5. Y. H. S. Kumar, "KD-Tree Approach in Sketch Based Image Retrieval", International Conference on Mining Intelligence and Knowledge Exploration, pp 247-258, 2015 [\[2\]](#)
6. J. H. Zouaki, B.Abdelkhalak, " Indexing and content-based image retrieval", 2011 International Conference on Multimedia Computing and Systems, 10.1109/ICMCS.2011.5945587, 12 July, 2011.
7. J. Das and M. 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.
8. Thanh Manh Le, Thanh The Van, "Image retrieval system based on emd similarity measure and S-Tree", ICITES-2012, Springer Verlag, LNEE 234 (2013) 139-146,
9. 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.
10. Nguyễn Thị Uyên Nhi, Văn Thế Thành, Lê Mạnh Thành, "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, 2019.
11. 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.

12. Văn Thế Thành, Lê Mạnh Thành, "Truy vấn ảnh tri thức dựa trên chữ ký nhị phân", Jos.hueuni.edu.vn, Tập. 97; Số. 9; Năm 2014.
13. 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.
14. Shadi Abudalfa, Mohammad Mikki, "A Dynamic Linkage Clustering using KD-Tree", the International Arab Journal of Information Technology, Vol. 10, No. 3, May 2013.
15. Stephen J. Redmond, Conor Heneghan. "A method for initialising the K-means clustering algorithm using kd-trees " ScienceDirect, Pattern Recognition Letters 28 (2007) 965–973.
16. Zhi-chun huang, patrick p. K. Chan, wing w. Y. Ng, daniel s. Yeung. "Content-Based Image Retrieval Using Color Moment And Gabor Texture Feature". Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, 11-14 July 2010.
17. S. Mangijao Singh, K. Hemachandran, "Content-Based Image Retrieval using Color Moment and Gabor Texture Feature ", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012.
18. Cevikalp H., Elmas M., Ozkan S., "Large-scale image retrieval using transductive support vec-to machines", Computer Vision and Image Understanding, vol. no. pp.1-11, 2017.
19. Jiu M., Sahbi H., "Nonlinear Deep Kernel Learning for Image Annotation", IEEE Transactions on Image Processing, vol. 26, no.4, pp.1820-1832, 2017.
20. Thinh N.V., Thanh T.V., Thanh M.L., "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.

Abstract. The paper proposes the data clustering model based on the BKD-Tree, an improvement of KD-Tree for the image retrieval. This model includes: (1) storing multi-dimensional data objects at the leaf nodes of the tree to create data clusters based on semi-supervised learning method; (2) create a balanced tree structure to increase the efficiency of image search. We use BKD-Tree to make experiment on ImageCLEF image set (including 20,000 images). Our experimental results are compared with several recent works on the same data set to demonstrate the effectiveness of the proposed method. This shows that our method is effective and can be applied to similar image retrieval systems by content.

Keywords: BKD-Tree, clustering, similar image, similar measure, image retrieval.