

Bài báo khoa học

Phương pháp tối ưu hóa trong lưu trữ và truy vấn dữ liệu không gian

Dương Thị Thúy Nga^{1*}, Nguyễn Văn Kiên¹, Nguyễn Thị Tường Vi¹, Phạm Minh Khan¹, Từ Thanh Trí¹, Đặng Đức Trung¹

¹ Trường Đại học Tài nguyên và Môi trường TP HCM; dtnga_cntt@hcmunre.edu.vn; nvkien@hcmunre.edu.vn; nttvi@hcmunre.edu.vn; pmkhan@hcmunre.edu.vn; tttri@hcmunre.edu.vn; ddtrung@hcmunre.edu.vn

*Tác giả liên hệ: dtnga_cntt@hcmunre.edu.vn; Tel: +84-938589589

Ban Biên tập nhận bài: 5/2/2022; Ngày phản biện xong: 16/3/2022; Ngày đăng bài: 25/4/2022

Tóm tắt: Cơ sở dữ liệu không gian dần trở nên phổ biến và được sử dụng rộng rãi trong nhiều lĩnh vực nhờ sự phát triển của công nghệ trong việc thu thập dữ liệu. Lượng cơ sở dữ liệu không gian phong phú được thu thập từ các vệ tinh, cảm biến và các thiết bị khác có thể giúp cải thiện kết quả của các tác vụ phân tích nhưng cũng tạo ra thử thách trong việc xử lý dữ liệu. Một số công trình nghiên cứu đã đề ra các giải pháp xử lý đối với cơ sở dữ liệu không gian có kích thước lớn như tạo chỉ mục, thu giảm số chiều dữ liệu. Trong bài báo này, chúng tôi sẽ trình bày hai phương pháp để tối ưu việc lưu trữ và truy vấn dữ liệu không gian. Thứ nhất, chúng tôi đề xuất một phương pháp nén hình ảnh dạng raster để thu giảm kích thước nhưng không mất thông tin của dữ liệu. Ngoài ra, chúng tôi cũng đưa ra một số kỹ thuật để tối ưu thời gian thực thi câu truy vấn. Kết quả cho thấy phương pháp đề xuất của chúng tôi giúp cải thiện thời gian thực thi truy vấn đồng thời thu giảm được không gian lưu trữ dữ liệu.

Từ khóa: Dữ liệu không gian; Nén ảnh raster; Truy vấn không gian; Tối ưu truy vấn.

1. Giới thiệu

Cơ sở dữ liệu không gian là một kiểu cơ sở dữ liệu hỗ trợ lưu trữ và xử lý những dữ liệu liên quan đến các đối tượng trong không gian hình học như điểm (*Point*), đường thẳng (*Line*), đa giác (*Polygon*) hoặc những cấu trúc dữ liệu thông tin địa lý. Ngày nay, cơ sở dữ liệu không gian dần phổ biến và có vai trò ngày càng trọng trong đời sống với các ứng dụng trong nhiều lĩnh vực khác nhau như dự đoán các vấn đề về môi trường, quy hoạch, thiết kế đô thị. Khác với các kiểu cơ sở dữ liệu truyền thống chủ yếu chỉ lưu trữ dữ liệu ký tự và số, sự phức tạp của dữ liệu không gian đòi hỏi phải chú trọng sự tối ưu hóa trong việc lưu trữ, xử lý. Nhiều công trình nghiên cứu đã đề xuất các phương pháp để xử lý cơ sở dữ liệu không gian như tạo chỉ mục để hỗ trợ việc truy vấn [1], thu giảm chiều dữ liệu [2], xử lý song song [3] nhằm mục đích cải thiện hiệu suất xử lý. Phần còn lại của bài báo được tổ chức như sau: Phần 2, cung cấp một số cơ sở lý thuyết liên quan đến cơ sở dữ liệu không gian. Phần 3, trình bày phương pháp đề xuất để tối ưu việc lưu trữ và truy vấn dữ liệu không gian. Phần 4, mô tả về dữ liệu và kết quả thực nghiệm. Phần 5, kết luận.

2. Cơ sở lý thuyết

2.1. Một số khái niệm

Dữ liệu không gian (*spatial data*) là dạng dữ liệu liên quan đến các thông tin địa lý của các thực thể trong thế giới thực hoặc sự liên hệ giữa vị trí giữa các thực thể. Dựa theo đặc điểm, dữ liệu không gian được chia thành 2 loại: dữ liệu Raster và dữ liệu Vector.

Dữ liệu Raster được biểu diễn dưới dạng một ma trận gồm nhiều hàng và cột, mỗi ô trong ma trận lưu trữ các giá trị liên quan đến thông tin được biểu diễn [4], ví dụ như ảnh vệ tinh.

Dữ liệu Vector gồm các điểm tọa độ được liên kết với nhau trong không gian 2 hoặc 3 chiều tạo nên các đường thẳng, đa giác, mặt phẳng [4]. Ví dụ các đối tượng con đường, dòng sông, ngôi làng có thể biểu diễn dưới dạng nhiều đường thẳng hoặc nhiều đa giác trong các hệ thống thông tin địa lý.

Truy vấn không gian (*spatial query*) là kiểu truy vấn nhằm tìm kiếm dữ liệu trong một cơ sở dữ liệu không gian theo một số điều kiện cụ thể do người dùng cung cấp [5].

Chỉ mục dữ liệu không gian (*indexing spatial data*) là một cơ chế cho phép thu giảm không gian tìm kiếm trong quá trình truy vấn dữ liệu [6].

Nén ảnh (*image compression*) là quá trình nén dữ liệu trên các ảnh số (*digital image*) để giảm bớt sự dư thừa về dữ liệu nhằm mục đích giảm kích thước không gian lưu trữ và thời gian xử lý. Hai kỹ thuật nén thường được áp dụng đối với ảnh Raster là: Lossless [7] và Lossy [7].

Lossless là kỹ thuật nén gồm các giải thuật thu giảm kích thước dữ liệu nhưng vẫn đảm bảo thông tin. Dữ liệu sau khi nén bằng các kỹ thuật Lossless có thể khôi phục lại như dữ liệu gốc ban đầu. Một số giải thuật nén Lossless như: Huffman Coding [8], Arithmetic Coding [9], LZW (*Lempel–Ziv–Welch*) [10].

Lossy là kỹ thuật nén gồm các giải thuật thu giảm kích thước dữ liệu nhưng bị mất mát thông tin và dữ liệu sau khi nén không thể khôi phục lại như dữ liệu gốc ban đầu. Kỹ thuật nén Lossy được sử dụng trong những trường hợp không quan tâm nhiều đến chi tiết thông tin dữ liệu mà chỉ cần sự chính xác tương đối. Phương pháp nén Lossy thường được sử dụng nhất là biến đổi cosin rời rạc [11] (*Discrete Cosine Transform – DCT*).

2.2. Tối ưu xử lý dữ liệu không gian

Tối ưu xử lý dữ liệu không gian gồm các kỹ thuật dùng để tìm kiếm giải pháp tối ưu khi xử lý các dữ liệu có cấu trúc phức tạp hoặc kích thước lớn, các kỹ thuật tối ưu liên quan đến việc tối thiểu hóa hoặc tối đa hóa các hàm mục tiêu thỏa mãn các điều kiện cụ thể.

Trong phần này, chúng tôi giới thiệu hai kỹ thuật thường được sử dụng trong việc tối ưu hóa xử lý dữ liệu không gian là lập chỉ mục và xử lý song song.

Chỉ mục dữ liệu không gian được sử dụng để tăng tốc độ truy vấn lấy dữ liệu bằng cách áp dụng một số cấu trúc dữ liệu. Cấu trúc dữ liệu thường được sử dụng nhất là R-Tree [12] và một số biến thể của nó như R*-Tree [13], R+-Tree [14]. Ý tưởng của R-Tree là tổ chức dữ liệu theo cấu trúc dạng cây, những đối tượng gần nhau được gộp lại và đưa vào các hình chữ nhật biên tối thiểu (*MBR – Minimum Bounding Rectangle*) nhằm tối thiểu khoảng cách và sự chồng chéo giữa các đối tượng. Ngoài R-Tree, nhiều cấu trúc dữ liệu khác cũng được sử dụng để lập chỉ mục cho dữ liệu không gian như QuadTree [15] phân chia một cách đệ quy không gian dữ liệu thành các góc phần tư hoặc cấu trúc Grid index [16] dựa trên định nghĩa một vùng gồm các ô liên tục được gán một giá trị định dạng để sử dụng cho việc lập chỉ mục.

Một cơ chế khác hỗ trợ tối ưu xử lý dữ liệu không gian là thực hiện song song việc xử lý dữ liệu dựa vào các mô hình lập trình. MapReduce [3] là một mô hình cho phép thực hiện hiệu quả việc xử lý song song dữ liệu và được sử dụng rộng rãi hiện nay. MapReduce sử dụng hai thủ tục là map và reduce thực hiện phân chia dữ liệu để xử lý và gộp các dữ liệu lại sau khi đã xử lý xong.

3. Phương pháp thực hiện

Trong bài báo này, chúng tôi sẽ trình bày hai giải pháp nhằm mục đích tối ưu việc xử lý và lưu trữ dữ liệu không gian. Thứ nhất, chúng tôi sử dụng phương pháp nén hình ảnh dạng

Raster để thu giảm chiều dữ liệu trước khi thực hiện truy vấn. Thứ hai, chúng tôi thực hiện thay thế cấu trúc lồng trong câu truy vấn bằng cách sử dụng phép kết và câu lệnh Case When.

3.1. Nén và truy vấn dữ liệu ảnh viễn thám dạng Raster

Kỹ thuật nén dữ liệu nhằm giảm không gian lưu trữ nhưng vẫn đảm bảo độ chính xác khi truy xuất dữ liệu. Trong bài báo này, chúng tôi thực hiện nén dữ liệu ảnh viễn thám dạng Raster. Khi lưu trữ các ảnh viễn thám có kích thước nhỏ thì không gian bộ nhớ lưu trữ có thể không cần quan tâm. Tuy nhiên đối với một ảnh có kích thước lớn, bộ nhớ cần để lưu trữ sẽ là rất lớn (khoảng 6.250.000 dòng cho một ảnh có kích thước 2500×2500 điểm ảnh). Do đó, nếu không thực hiện nén dữ liệu, việc đưa dữ liệu Raster vào hệ quản trị cơ sở dữ liệu sẽ mất rất nhiều thời gian và bộ nhớ để lưu trữ cũng sẽ rất lớn.

Việc đầu tiên trước khi thực hiện nén ảnh là cần phải xác định một kích thước nén X x Y. Dựa vào kích thước nén, thực hiện quét toàn bộ ma trận con của ảnh có kích thước nén X x Y (Hình 1) và thực hiện biến đổi tất cả các giá trị của ma trận con để thu giảm kích thước bằng các kỹ thuật nén Lossless hoặc Lossly (Hình 2).

	1	2	3
1	97	38	146
2	125	104	211
3	80	91	190
4	78	72	140

Hình 1. Ảnh kích thước 3×4.

	1	2	3
1	1		2
2			
3			
4	3		4

Hình 2. Ảnh sau khi nén với kích thước nén 2×3.

Hình 1 và hình 2 được áp dụng kỹ thuật nén ảnh dạng Raster với kích thước 2×3. Các điểm ảnh sẽ được duyệt từ trái sang phải và từ trên xuống dưới, những điểm ảnh nằm trong ma trận 2 cột 3 dòng sẽ được lưu trong cùng 1 Raster (Điểm ảnh màu vàng của hình 1). Những điểm ảnh nằm ở vùng biên thì số dòng và số cột thường nhỏ hơn 2 cột 3 dòng tuy nhiên cũng sẽ được lưu thành 1 Raster riêng (Điểm ảnh màu cam, màu xanh lá và màu trắng). Hình 2 cho thấy những điểm ảnh thuộc cùng một màu sẽ được lưu trong cùng 1 Raster tương ứng với rid (rid là định danh của Raster) là 1, 2, 3, 4.

Dựa vào cách nén ảnh trên có thể suy ra công thức tính gần đúng số Raster dùng để lưu trữ tất cả các điểm ảnh của ảnh vệ tinh khi áp dụng nén như sau:

$$\text{Số lượng Raster dùng để lưu trữ} \approx \frac{M \times N}{X \times Y}$$

Trong đó M×N: Kích thước của ảnh vệ tinh; X×Y: Kích thước nén của ma trận X×Y điểm ảnh trên 1 Raster.

Công thức chỉ tính gần đúng số lượng Raster vì có những Raster sau khi nén sẽ không lưu đủ ma trận X x Y giá trị điểm ảnh do nằm ở vùng biên của ảnh.

Trong bài báo này, chúng tôi sử dụng thư viện của hệ quản trị cơ sở dữ liệu PostgreSQL để nén dữ liệu ảnh dạng Raster, thực hiện lưu trữ và kiểm tra hiệu suất xử lý trên dữ liệu sau khi nén.

3.2. Tối ưu hóa câu truy vấn

Truy vấn lồng là một câu truy vấn hoàn chỉnh bắt đầu bằng mệnh đề SELECT và được đặt lồng vào một câu truy vấn khác. Truy vấn lồng có thể xuất hiện trong mệnh đề SELECT, WHERE và một phần của mệnh đề FROM. Đối với một câu truy vấn có kết quả trả về là một số lượng lớn các dòng, không nên sử dụng truy vấn lồng trong mệnh đề SELECT hoặc WHERE vì truy vấn lồng đó sẽ phải duyệt qua hết từng dòng. Do đó thời gian thực thi sẽ rất

lâu và hiệu suất sẽ bị ảnh hưởng, đặc biệt trong trường hợp câu truy vấn lồng có tham chiếu đến các thuộc tính của câu truy vấn bên ngoài.

Để tối ưu câu truy vấn trong quá trình xử lý dữ liệu không gian, chúng tôi sẽ thực hiện viết lại câu truy vấn và thay thế các truy vấn lồng bằng cách sử dụng phép kết INNER-JOIN kết hợp với câu lệnh CASE WHEN nhằm mục đích cải thiện tốc độ truy vấn. Việc điều chỉnh và chạy thực nghiệm các câu truy vấn được thực hiện trên hệ quản trị cơ sở dữ liệu PostgreSQL.

4. Kết quả thực nghiệm

Nghiên cứu tiến hành thực nghiệm trên cơ sở dữ liệu không gian về thông tin địa lý của khu vực Hồ Chí Minh và tỉnh khác. Kết quả cho thấy, phương pháp đề xuất của chúng tôi không chỉ giúp thu giảm không gian lưu trữ dữ liệu mà còn giảm thời gian thực thi truy vấn nhưng vẫn đảm bảo độ chính xác so với khi thực thi trên dữ liệu gốc ban đầu.

4.1. Thực nghiệm 1

Trong mục này chúng tôi sẽ trình bày kết quả thực nghiệm khi thực hiện truy vấn khi nén dữ liệu với 3 ảnh vệ tinh có kích thước lần lượt 2500×2900 điểm ảnh, 3800×3000 điểm ảnh, 4500×3900 điểm ảnh có 1 kênh nén với kích thước lần lượt là 10×10, 30×30, 50×50, 100×100, 200×200. Câu truy vấn thực hiện lấy ra tất cả các giá trị Raster của huyện Cầu Kè tỉnh Trà Vinh với các kích thước nén khác nhau từ ảnh vệ tinh.

Bảng 1. Bảng so sánh thời gian truy vấn, bộ nhớ lưu trữ và thời gian thêm ảnh của 3 ảnh vệ tinh với các kích thước khác nhau.

		Kích thước nén				
		10×10	30×30	50×50	100×100	200×200
Ảnh 2500x2900	Bộ nhớ lưu trữ (KB)	14670	8349	2355	213	22
	Thời gian import ảnh (s)	2106	299	37	31	25
	Thời gian truy vấn (ms)	68	88	115	278	1117
Ảnh 3800x3000	Bộ nhớ lưu trữ (KB)	29696	16376	4356	443	29
	Thời gian import ảnh (s)	9929	1297	74	52	128
	Thời gian truy vấn (ms)	72	77	105	290	1266
Ảnh 4500x3900	Bộ nhớ lưu trữ (KB)	48580	28715	76387	501	41
	Thời gian import ảnh (s)	15809	46128	85	79	260
	Thời gian truy vấn (ms)	65	77	97	267	1854

Dựa vào bảng 1, có thể thấy nếu nén Raster thì bộ nhớ lưu trữ và thời gian thêm dữ liệu ảnh vệ tinh được cải thiện nhiều nhất, tốc độ truy vấn có tăng lên tuy nhiên không đáng kể. Trong số các kích thước nén, 100×100 là kích thước vừa đạt được sự thu giảm không gian lưu trữ nhưng vẫn đảm bảo tốc độ truy vấn.

4.2. Thực nghiệm 2

Trong thực nghiệm thứ hai, chúng tôi sẽ so sánh tốc độ câu truy vấn có cấu trúc lồng và câu truy vấn sử dụng Inner Join kết hợp Case When để thay thế truy vấn lồng. Câu truy vấn được thực hiện là lấy danh sách các quận của TP. HCM có tổng chiều dài đường ray xe lửa lớn hơn 1000 m.

Bảng 2. Câu truy vấn lấy dữ liệu sử dụng truy vấn lồng và Inner Join kết hợp Case When.

Sử dụng truy vấn lồng	Sử dụng Inner Join kết hợp Case When
SELECT a.ten, (SELECT COUNT (*) AS cnt	SELECT a.ten, COUNT (r.gid) AS cnt,
FROM vnm_roads as r	COUNT (CASE WHEN ST_Length(r.geom) > 0.1
WHERE ST_Intersects(a.geom, r.geom)) As cnt,	THEN 1 ELSE NULL END)
(SELECT COUNT (*) AS cnt	As cnt_gt_1000
FROM vnm_roads as r	FROM quan_huyen as a
WHERE ST_Intersects(a.geom, r.geom)	INNER JOIN vnm_roads as r
AND ST_Length(r.geom) > 0.1) As cnt_gt_1000	ON (ST_Intersects(a.geom, r.geom))
FROM quan_huyen as a	GROUPBY a.ten
WHERE EXISTS (SELECT r.gid	ORDERBY a.ten;
FROM vnm_roads as r	
WHERE ST_Intersects(a.geom, r.geom))	
ORDERBY a.ten;	

Bảng 3. Kết quả sử dụng truy vấn lồng và Case when kết hợp Inner-Joins.

Lần thực hiện	Sử dụng truy vấn lồng (ms)	Inner Join kết hợp Case When (ms)
Lần 1	100	68
Lần 2	96	73
Lần 3	98	68
Trung bình	99	70

Từ kết quả bảng 3, truy vấn có sử dụng phép kết Inner Join kết hợp câu lệnh Case When có thời gian truy vấn nhanh hơn truy vấn lồng, thời gian trung bình giảm từ 99 ms xuống còn 70 ms. Qua đó có thể thấy được sự hiệu quả của việc thay thế truy vấn lồng bằng phép kết Inner Join kết hợp câu lệnh Case.

5. Kết luận

Bài báo đã giới thiệu hai phương pháp giúp cải thiện hiệu suất trong việc xử lý dữ liệu không gian. Phương pháp nén dữ liệu ảnh dạng raster giúp giảm không gian lưu trữ nhưng vẫn đảm bảo tính chính xác của dữ liệu. Việc sử dụng Inner Join kết hợp Case When thay thế truy vấn lồng cho thấy sự cải thiện về tốc độ truy vấn. Ngoài ra hai phương pháp này còn có thể kết hợp với nhau và kết hợp với các phương pháp khác như lập chỉ mục, xử lý song song để nâng cao hiệu suất trong việc xử lý dữ liệu không gian.

Đóng góp của tác giả: Xây dựng ý tưởng nghiên cứu: L.H.T.; Lựa chọn phương pháp nghiên cứu: L.H.T., D.N.D.P.; Xử lý số liệu: D.N.D.P., P.T.H.; Chạy mô hình: L.H.T., D.N.D.P., P.T.H.; Phân tích và đánh giá kết quả: L.H.T., N.K.L.; Viết bản thảo bài báo: L.H.T., D.N.D.P., P.T.H., N.K.L., N.T.H.; Chính sửa bài báo: L.H.T., D.N.D.P., P.T.H., N.T.H., N.K.L.

Lời cảm ơn: Để hoàn thành nghiên cứu này, chúng tôi trân trọng gửi lời cảm ơn đến Sở Khoa học và Công nghệ tỉnh Gia Lai đã cấp kinh phí thông qua đề tài: “Xác định tập đoàn giống cây trồng nông nghiệp chủ lực của tỉnh Gia Lai”, Mã số: KHGL-09-18.

Lời cam đoan: Tập thể tác giả cam đoan bài báo này là công trình nghiên cứu của tập thể tác giả, chưa được công bố ở đâu, không được sao chép từ những nghiên cứu trước đây; không có sự tranh chấp lợi ích trong nhóm tác giả.

Tài liệu tham khảo

1. Suhaibah, A.; Uznir, U.; François, A.; Darka, M.; Alias, R. Review of Spatial Indexing Techniques for Large Urban Data Management. International Symposium & Exhibition on Geoinformation (ISG), Kuala Lumpur, Malaysia, 2013.
2. Michael, W.; An, L.K.N.; Tahar, K. Data Reduction in Very Large Spatio-Temporal Datasets. 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, Larissa, Greece, 2010.
3. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Communications ACM* **2008**, 51(1), 107–113.
4. Guo, D.; Onstein, E. State-of-the-art geospatial information processing in NoSQL databases. *Int. J. Geo-Inf.* **2020**, 9(5), 1–20.
5. Geetha, S.; Velavan, S.A. Optimization of location based queries using spatial indexing. *ICTACT J. Soft Comput.* **2014**, 4(3), 738–742.
6. Jitkajornwanich, K.; Pant, N.; Fouladgar, M.; Elmasri, R. A survey on spatial, temporal, and spatio-temporal database research and an original example of relevant applications research and an original example of relevant applications. *J. Inf. Telecommun.* **2020**, 4(4), 524–559.
7. Yogesh, C.; Mishra, S. Review Paper of Image Compression. *Int. J. Emerging. Applied Sci.* **2015**, 2(2), 56–58.
8. Huffman, D. A method for the construction of minimum redundancy codes. *Proceedings of the Institute of Radio Engineers* **1952**, 40(9), 1098–1101.
9. Witten, I.H.; Neal, R.M.; Cleary, J.G. Arithmetic coding for data compression. *Commun. ACM* **1987**, 30(6), 520–540.
10. Kinsner, W.; Greenfield, R.H. The Lempel–Ziv–Welch (LZW) data compression algorithm for packet radio. WESCANEX '91, Regina, SK, Canada, 1991.
11. Ahmed, N.; Natarajan, T.; Rao, K.R. Discrete Cosine Transform. *IEEE Trans. Comput.* **1974**, C-23(1), 90–93.
12. Guttman, A. R-Trees: A Dynamic Index Structure for. *ACM SIGMOD Record* **1984**, 14(2), 47–57.
13. Beckmann, N.; Kriegel, H.P.; Schneider, R.; Seeger, B. The R*-Tree: An efficient and robust access method for points and rectangles. *ACM SIGMOD Record* **1990**, 19(2), 322–331.
14. Sellis, T.; Roussopoulos, N.; Faloutsos, C. The R+-Tree: A dynamic index for multi-dimensional objects. In *Proceedings of the 13th International Conference on Very Large Data Bases*, Brighton, 1987.
15. Finkel, R.; Bentley, J.L. Quad Trees: A data structure for retrieval on composite keys. *Acta Inf.* **1974**, 4(1), 1–9.
16. Sahr, K.; White, D.; Kimerling, A.J. Geodesic Discrete Global Grid Systems. *Cartography Geographic Inf. Sci.* **2003**, 30(2), 121–134.

Optimization methods in storage and querying spatial data

Duong Thi Thuy Nga^{1*}, Nguyen Van Kien¹, Nguyen Thi Tuong Vi¹, Pham Minh Khan¹,
Tu Thanh Tri¹, Dang Duc Trung¹

¹ Ho Chi Minh City University of Natural Resources and Environment;
dttnga_cntt@hcmunre.edu.vn; nvkien@hcmunre.edu.vn; nttvi@hcmunre.edu.vn;
pmkhan@hcmunre.edu.vn; tttri@hcmunre.edu.vn; ddtrung@hcmunre.edu.vn

Abstract: Due to recent technological developments that have facilitated the collection of large amounts of data, spatial data has become popular and has been widely used in many areas. The rich spatial datasets, which come from satellites, sensors, and other devices, can help improve results in data analysis tasks. However, it is also a challenge to address the complexity of data processing. There has been some research work that proposed methods for handling large spatial databases, such as indexing spatial data, reducing the dimension of spatial data. In this paper, we present two effective methods for optimizing storing and querying spatial data. In our method, we devised an image compression method that helps in reducing the size of raster images but retains meaningful properties of the original data. Furthermore, we implemented some query techniques to decrease the query execution time for spatial data. Experimental results show that our proposed methods not only improve the time of data query but also reduce the storage space.

Keywords: Spatial data; Raster image compression; Spatial query; Query optimization.