

Avances en Propuestas Heurísticas para Consultas a R-Tree

Edilma O. Gagliardi, Juan E. Gomez

Departamento de Informática, Universidad Nacional de San Luis
San Luis, Argentina

y

Gilberto Gutiérrez Retamal

Departamento de Auditoría e Informática, Universidad del Bío-Bío
Chillán, Chile

RESUMEN

En la actualidad las bases de datos espaciales han tomado gran interés dado la cantidad de aplicaciones en las que son utilizadas, esto ha generado una mayor necesidad en investigar métodos que mejoren el desempeño en la resolución de consultas de tipo espacial.

Los trabajos de investigación en Bases de Datos espaciales se han centralizado en el modelado y la resolución de consultas, basándose en la geometría asociada a los objetos almacenados en una base de datos[1]. Nuestro trabajo de investigación se abocará a poder resolver eficientemente consultas a índices R-tree's, considerando atributos derivados de la geometría de los objetos espaciales como lo son por ejemplo el *área* en el caso de los polígonos o la *longitud* en las poligonales.

Nuestro interés radica en este tipo de índices R-Tree's debido a que es un método de acceso estándar para las bases de datos espaciales, elegido por la mayoría de los Sistemas de Administración de Bases de Datos.

Palabras claves: índices espaciales, consultas espaciales, procesamiento de consultas espaciales, acceso multidimensional, bases de datos espaciales.

1. INTRODUCCIÓN

Hay aplicaciones del mundo real en donde la necesidad de poder almacenar y recuperar datos de tipo espacial es muy importante, ejemplo de esto son los Sistemas de Información Geográfica (SIG), Diseño Asistido por Computador(CAD), diseño VLSI, visión por computador y robótica, entre otros, todo esto ha provocando una revisión respecto a como resolver consultas espaciales[1,2].

El costo de procesar una consulta espacial puede llegar a ser muy alto debido a que los datos espaciales son más complejos que los alfanuméricos. Hay varios métodos para procesar datos espaciales[3][4][5], que consideran consultas sólo de tipo espacial. Pero, dado que las bases de datos espaciales almacenan datos espaciales y no espaciales, las consultas podrían mezclar subconsultas espaciales con subconsultas no espaciales[6][7]. Y este hecho deja de ser considerado en los métodos que exclusivamente se dedican a considerar los aspectos espaciales únicamente.

Por ello, parte de este trabajo considera el procesamiento de consultas cuyo predicado establece restricciones sobre atributos derivados de la geometría de objetos espaciales. Por ejemplo, una consulta podría ser: *seleccionar todas las ciudades que tengan una superficie mayor que 5000 Km²*. En el ejemplo, el atributo *superficie* es un atributo derivado de la geometría de los objetos almacenados en la base de datos y el índice debería ser capaz de considerarlo al momento de la resolución de la consulta.

Presentaremos a continuación aspectos importantes relacionados a nuestro trabajo de investigación.

2. R-TREE

R-Tree[8] es un método de acceso multidimensional, que se ha adoptado como el método de acceso estándar para las bases de datos espaciales y es el elegido por la mayoría de los Sistemas de Administración de Bases de Datos. También es el más estudiado con respecto a tópicos tales como procesamiento y optimización de consultas, modelos de costo, paralelismo, control de concurrencia y recuperación, entre otros. Además, gran parte de los métodos de acceso espacio-temporales propuestos hoy en día usan como base a R-Tree.

En un R-Tree no se almacenan los objetos espaciales en forma directa sino que se almacena su MBR (Minimum Bounding Rectangle), es decir el menor rectángulo que contiene al objeto en cuestión. Cada nodo en el R-Tree corresponde al MBR que contiene a sus hijos. Los nodos hoja del R-Tree contienen punteros a los objetos en la base de datos en vez de punteros a otros nodos. Cada nodo se almacena en una página de disco. Los nodos hoja del R-Tree contienen entradas de la forma $\langle I; oid \rangle$ donde I es el menor rectángulo n -dimensional que contiene al objeto apuntado por oid ; es decir, es de la forma $I = (I_0, I_1, \dots, I_n)$. Aquí n es el número de dimensiones e I es un intervalo cerrado $[a, b]$ que describe los límites del objeto en la dimensión i . En caso de que un objeto espacial se extienda más allá de los límites del espacio definido, entonces I_i puede tener uno o ambos puntos extremos igual a infinito. Los nodos que no son de tipo hoja, contienen entradas de la forma $\langle I; pchild \rangle$ donde $pchild$ es un puntero a un hijo del nodo e I contiene a todos los MBR's del nodo apuntado por $pchild$. En un R-Tree, cada nodo, con la posible excepción del nodo raíz, contiene entre m y M entradas donde $m \leq M/2$ y M es el número máximo de entradas por nodo; el nodo raíz tiene al menos dos hijos a menos que sea una hoja; y todas las hojas están al mismo nivel.

En el procedimiento de búsqueda se desciende por el árbol a partir de la raíz, siguiendo por los hijos cuyo MBR se intersecta con el área de consulta y así en forma recursiva, hasta llegar a las hojas. Los MBR's que encierran los diferentes nodos pueden superponerse; además, un MBR puede estar incluido, en el sentido geométrico, en varios nodos, pero está asociado sólo con uno de ellos. Esto implica que una búsqueda puede seguir más de un camino, incluyendo caminos innecesarios.

Para insertar un objeto se desciende recursivamente por el árbol a partir de la raíz; siguiendo por los hijos cuyo MBR crecerá menos, producto de la inserción de un nuevo objeto hasta llegar a un nodo hoja. El objeto se inserta en la hoja si hay espacio, en caso contrario el nodo se divide usando alguna de las técnicas de división conocidas[8]. Posteriores variantes del R-Tree difieren principalmente en la forma en que se insertan los objetos.

Al eliminar un objeto si el nodo que lo contenía tiene insuficientes entradas, éstas se eliminan y se reinsertan. Los cambios de MBR's, producto de la eliminación, se propagan hacia arriba.

3. HEURISTICAS

Existen varios métodos que podemos utilizar para procesar consultas espaciales[3][4][5]. La mayoría de tales métodos consideran dos etapas: *filtrado* y *refinamiento*. En la etapa de filtrado se utiliza el índice espacial R-Tree para seleccionar objetos candidatos de la repuesta. Luego, en la etapa de refinamiento, se considera la geometría del objeto para decidir si definitivamente cumple con las restricciones de la consulta.

Dado un conjunto $S1$ de objetos espaciales de tipo polígono y otro conjunto $S2$ que contiene poligonales; dos consultas $Q1$ (encontrar todas las ciudades con un área mayor o igual a 5000 Km^2) y $Q2$ (obtener todas las poligonales con una longitud menor o igual 300 cm); ambas consultas son espaciales pero el predicado de estas considera atributos derivados de la geometría de los objetos de $S1$ y $S2$, en un caso el atributo derivado es el área y en el otro es la longitud; y finalmente $R1$ y $R2$,

los índices R-Tree para $S1$ y $S2$ respectivamente, es decir tenemos una estructura R-Tree para cada conjunto.

Bajo el supuesto de que no existe un índice adicional, como ejemplo un B-Tree, sobre los atributos derivados, para responder las consultas $Q1$ y $Q2$ es necesario recorrer uno por uno los objetos de $S1$ y $S2$, luego verificar el predicado, lo cual implica acceder todos los bloques utilizados para almacenar los objetos, lo cual obviamente es costoso.

Como posible heurística a seguir, una alternativa consistiría en apoyarse en el índice $R1$ y utilizar las aproximaciones MBR de los objetos espaciales para responder a la consulta $Q1$ [9]. En ese caso, para el procesamiento de las consultas espaciales se procede de manera similar a lo descrito en [3], donde el método considera dos etapas. La primera etapa consiste en seleccionar el conjunto de objetos que puedan satisfacer $Q1$. Estos objetos se seleccionan utilizando las correspondientes aproximaciones MBR's. El conjunto obtenido en esta etapa es un súper conjunto del conjunto de objetos espaciales que conforman la repuesta definitiva de $Q1$. En la segunda etapa se considera específicamente la geometría de los objetos obtenidos en la primera etapa y se verifica si el objeto satisface el predicado de la consulta realizada $Q1$. El mismo procedimiento es aplicable para realizar la consulta $Q2$ utilizando el índice $R2$.

Lo que pretendemos en nuestro trabajo de investigación es tratar con conjuntos de objetos geométricos. Hemos seleccionado polígonos y poligonales.

Para el caso de los polígonos, utilizamos el área de los MBRs como una aproximación del área de los polígonos. Como se muestra en la Figura 1 se cumple la propiedad $AREA_{MBR} \geq AREA_{Poligono}$, esta propiedad nos permitirá resolver consultas que involucren como atributo derivado el *área*.

En el caso de las poligonales, utilizamos las diagonales de los MBRs como una aproximación de la longitud de las poligonales, se cumple que $LONGITUD_{Diagonal\ MBR} \leq LONGITUD_{Poligonal}$ tal cual lo muestra la Figura 2, dicha propiedad nos permitirá resolver consultas que involucren como atributo derivado la *longitud*.

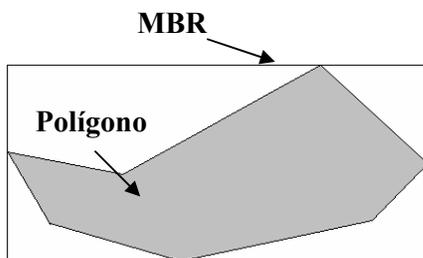


Figura 1: Relación entre el área de un Polígono y su MBR .

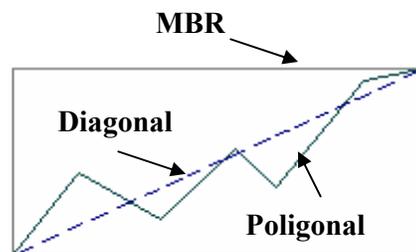


Figura 2: Relación entre la longitud de una poligonal y la longitud de la diagonal principal del MBR.

En nuestra propuesta de investigación utilizamos los mbr's como una aproximación de la geometría real de los objetos, lo cual creemos es una buena medida para obtener un conjunto de objetos a partir del tipo de consultas que hemos planteado anteriormente, los resultados de estas consultas se aproximan muy bien al conjunto real de objetos que se desea obtener. Lo que pretendemos alcanzar con lo expuesto anteriormente es mejorar la etapa de filtrado en el procesamiento de consultas del R-Tree, esto hará que al algoritmo de búsqueda me devuelva una menor cantidad de mbr's, luego estos en la etapa de refinamiento tienen una mayor probabilidad de coincidir con el valor real de los objetos. Entonces decimos que se logra mejorar las dos etapas del procesamiento de consultas en el R-Tree.

4. TRABAJO FUTURO

Como el R-Tree es una estructura base de otras estructuras, el poder plantear modificaciones que resulten en una mejora de su desempeño, o en una extensión de su desempeño básico, nos permitirá poder aplicar estas ventajas a otras estructuras.

Nosotros estamos trabajando y experimentando principalmente en el algoritmo de búsqueda, se pretende realizar una implementación en algún lenguaje de programación y observar en la evaluación experimental diferentes comparaciones estadísticas, las cuales nos conducirán a determinar cuánto se aproxima a la realidad el método planteado.

En particular, nuestro trabajo de investigación se dedica al estudio y análisis del diseño de estructuras de datos y algoritmos que permitan resolver los distintos tipos de consulta espaciales tomando como índice estándar R-Tree. Estos trabajos están enmarcados dentro de la Línea de Investigación Geometría Computacional y Bases de Datos Espacio-Temporales, perteneciente al Proyecto Tecnologías Avanzadas de Bases de Datos 22/F314, Departamento de Informática, Universidad Nacional de San Luis; en el Proyecto AL07-PAC-027 Geometría Computacional, subvencionado por a Universidad Politécnica de Madrid; y en el marco de la Red Iberoamericana de Tecnologías del Software (RITOS2), subvencionado por CYTED. Por todo ello, se ha establecido un grupo de interés en el tema conformado por docentes investigadores y alumnos avanzados de la Universidad Nacional de San Luis.

5. REFERENCIAS

- [1] Ralf Gutting. *An Introduction to spatial database system*. In VLDB Journal, volume 3, pages 357-399, 1994.
- [2] King-Ip Lin, H.V. Jagadish, and Christos Faloutsos. *The TV-Tree: An index structure for high-dimensional data*. VLDB Journal: Very Large Data Bases, 3(4):517-542, 1994.
- [3] Thomas Brinkhoff, Hans-Peter Kriegel, Ralf Schneider and Bernhard Seeger. *Multi-step processing of spatial joins*. In ACM SIGMOD Conference on Management of Data, pages 197-208, Minesota, USA, 1994.
- [4] Thomas Brinkhoff, Hans-Peter Kriegel and Bernhard Seeger. *Efficient processing of spatial joins using r-trees*. In ACM SIGMOD Conference on Management of Data, pages 237-246, Washington, DC, USA, 1993. ACM.
- [5] Yun-Wu Huang, Ning Jing, and Elke Rundensteiner. *Spatial joins using r-tree: Breadth-first traversal with global optimizations*. IN 23rd Conference on Very Large Data Bases, pages 396-405, Athens, Greece, 1997.
- [6] Ho-Hyun Park, Chan-Gun Lee, Yong-Ju Lee, and Chin-Wan Chung. *Early separation of filter and refinement steps in spatial query optimization*. In Database System for Advanced Applications, pages 161-168, 1999.
- [7] Ho-Hyun Park, Yong-Ju Lee, and Chin-Wan Chung. *Spatial query optimization utilizing early separated filter and refinement strategy*. Information Systems, 25(1):1-22, 2000.
- [8] Antonin Guttman. *R-Tree: A Dinamic Index Structure for Spatial Searching*. In ACM SIGMOD Conference on Management of Data, pages 47-57, Boston, 1984.
- [9] Gilberto Gutiérrez. *Procesamiento de Consultas Espaciales con Restricciones sobre Atributos Derivados de la Geometría de Objetos Espaciales*. Bío-Bío Chile.