

# Procesamiento de consultas espaciales con restricciones sobre atributos derivados de la geometría de objetos espaciales \*

Gilberto Gutiérrez R.  
Avenida La Castilla S/N - Chillán / Chile  
Universidad del Bío-Bío  
(56) 42-203415  
ggutierr@ubiobio.cl

## Resumen

El procesamiento de consultas espaciales ha recibido bastante atención durante el último tiempo. Sin embargo, los trabajos se han centrado sobre consultas espaciales cuyos predicados involucran, principalmente, la intersección de regiones. En este trabajo se estudian consultas sobre un conjunto de objetos espaciales en las cuales el predicado establece restricciones sobre atributos derivados de la geometría de los objetos. Se estudió procesar estas consultas mediante el método propuesto en [GG98, BKSS94] el cual usa, en la etapa de *filtrado*, la aproximación MBR (Minimum Bounding Rectangle) de los objetos espaciales mantenidos en la estructura de índice multidimensional R-tree. Se analizaron dos atributos derivados - área y longitud- y se obtuvieron las estimaciones producidas por la etapa de *filtrado* para consultas cuyas respuestas consideraban entre un 10% y un 90% del conjunto de objetos. Resultados preliminares permiten inferir que el conjunto de objetos estimado en la etapa de filtrado es muy cercano al conjunto real. Estos resultados muestran que es conveniente obtener la respuesta de una consulta sobre atributos derivados utilizando las aproximaciones (MBR) de los objetos espaciales, y por lo tanto, en ausencia de un índice para el atributo derivado (árbol B), el uso de un R-tree en la etapa de *filtrado* para estimar el conjunto de objetos es una buena alternativa.

**Palabras claves:** Procesamiento de consultas espaciales, acceso multidimensional, bases de datos espaciales, consultas espaciales

## 1 Introducción

En el último tiempo se ha producido una necesidad creciente por el almacenamiento y recuperación de *datos espaciales*. Aplicaciones relacionadas con la superficie de la tierra, específicamente los Sistemas de Información Geográfica (SIG) componen un ámbito de aplicaciones las cuales hacen un uso intensivo de los datos espaciales. También se usan en forma importante en las áreas de Diseño Asistido por Computador (CAD), diseño VLSI, visión por computador y robótica [LJF94, Gut94]. El área de medicina también constituye un dominio interesante de aplicación de estos tipos de dato. Por ejemplo, a partir de una imagen del cerebro se pueden recuperar rápidamente casos pasados con síntomas similares y ser utilizados para apoyar un diagnóstico. Aplicaciones multimedias, también hacen uso de estos tipos de datos para representar y ubicar objetos que coinciden exacta o aproximadamente con un objeto dado en una consulta. Los objetos pueden corresponder a elementos en tres, dos o una dimensión, series de tiempo, música digitalizada, video clips, etc. [BYRN99].

El procesamiento de consultas espaciales constituye una de las áreas de investigación en bases de datos espaciales que ha recibido bastante atención durante el último tiempo [PLC00]. El costo de procesar una consulta espacial puede llegar a ser muy alto ya que los datos espaciales son mas complejos que los datos alfanuméricos. Se han propuesto varios métodos para procesar consultas espaciales [BKS93, BKSS94,

---

\*Este trabajo ha sido desarrollado como parte del proyecto 002714-3 / 2000 financiado por la Dirección de Investigación de la Universidad del Bío Bío.

HJR97]. La mayoría de tales métodos consideran dos etapas a saber: *filtrado* y *refinamiento*. En la etapa de filtrado se utiliza el índice espacial para seleccionar objetos candidatos a integrar la respuesta. Luego, en la etapa de refinamiento, se utiliza la geometría del objeto para decidir si definitivamente cumple con las restricciones de la consulta. Los métodos anteriores consideran consultas con predicados solo de tipo espacial. Sin embargo, debido a que las bases de datos espaciales almacenan datos espaciales y no espaciales, las consultas pueden mezclar subconsultas espaciales con subconsultas no espaciales [PLLC99, PLC00]. Este trabajo contempla el procesamiento de consultas cuyo predicado establece restricciones sobre atributos derivados de la geometría de objetos espaciales. Por ejemplo *seleccionar todas las comunas que tengan una superficie mayor que 5000 km<sup>2</sup>*. En el ejemplo anterior el atributo *superficie* es un atributo derivado de la geometría de los objetos almacenados en la base de datos.

La organización de este artículo es la siguiente: En el punto 2 se hace una breve revisión de los métodos utilizados para el procesamiento de consultas espaciales, en el punto 3 se presenta el método propuesto. En el punto 4 se analizan algunos resultados preliminares. Finalmente, en el punto 5 se entregan las conclusiones del trabajo.

## 2 Métodos de acceso multidimensional R-tree

Un R-tree es una extensión de un B-tree para objetos multidimensionales (puntos y regiones) [Gut84, SRF97]. Cada nodo corresponde a una página o bloque de disco. Los nodos hojas de un R-tree contienen entradas de la forma  $(I, oid)$  donde *oid* es el identificador del objeto espacial en la base de datos e *I* es un rectángulo *n*-dimensional y que corresponde al mínimo rectángulo que encierra al objeto espacial (Minimum Bounding Rectangle - MBR), es decir,  $I = (I_0, I_1, \dots, I_n)$ . Aquí *n* es el número de dimensiones e *I<sub>i</sub>* es un intervalo cerrado  $[a, b]$  que describe los límites del objeto en la dimensión *i*. En caso de que un objeto espacial se extiende más allá de los límites del espacio definido, entonces *I<sub>i</sub>* puede tener uno o ambos puntos extremos igual a infinito. Los nodos internos (nodos no-hojas) contienen entradas de la forma  $(I, pchild)$  donde *pchild* es la dirección del correspondiente nodo hijo en el R-tree e *I* cubre todos los rectángulos definidos en las entradas del nodo hijo.

Sea *M* el número máximo de entradas que se pueden almacenar en un nodo y sea  $m \leq \frac{M}{2}$  un parámetro especificando el número mínimo de entradas en un nodo. Un R-tree satisface las siguientes propiedades [Gut84]:

1. Cada nodo contiene entre *m* y *M* entradas a menos que corresponda a la raíz.
2. Para cada entrada  $(I, oid)$  en un nodo hoja, *I* es el mínimo rectángulo que contiene (espacialmente) al objeto.
3. Cada nodo interno tiene entre *m* y *M* hijos, a menos que sea la raíz.
4. Para cada entrada de la forma  $(I, pchild)$  de un nodo interno, *I* es el rectángulo más pequeño que espacialmente cubre los rectángulos definidos en el nodo hijo.
5. El nodo raíz tiene al menos dos hijos, a menos que sea una hoja.
6. Todas las hojas se encuentran al mismo nivel

La figura 1 y 2 muestran un conjunto de rectángulos y su correspondiente R-tree.

La altura de un R-tree que almacena *N* claves es a lo más  $\lceil \log_m N \rceil - 1$ , ya que el número de hijos es al menos *m*. El número máximo de nodos es  $\lceil \frac{N}{m} \rceil + \lceil \frac{N}{m^2} \rceil + \dots + 1$  [Gut84]. La utilización del almacenamiento (peor caso) para todos los nodos, excepto la raíz, es  $\frac{m}{M}$  [Gut84]. Los nodos tienden a mantener más que *m* entradas lo que permitirá que la altura del árbol disminuya y mejore la utilización del almacenamiento.

El algoritmo de búsqueda en un R-tree es similar al utilizado por un B-tree, es decir, la búsqueda se inicia en la raíz del árbol y se desciende por él hasta alcanzar un nodo hoja. Sin embargo, puede suceder que en un subárbol sea necesario visitar más de un nodo -todos aquellos cuyos rectángulos se intersectan con el de búsqueda. Por ejemplo el rectángulo *Q* de la figura 1 se intersecta con los rectángulos *R3* y *R4* lo que determina que el algoritmo debe recorrer ambos subárboles.

La inserción en un R-tree es similar a la inserción en un B-tree en el sentido que el elemento se inserta en las hojas produciendo eventualmente una división del nodo hoja y propaga un elemento hacia el padre el que a su vez también se puede dividir y propagar un elemento a su padre. El proceso continúa recursivamente hasta llegar a la raíz la cual también se puede dividir generando una nueva raíz.

En [SRF87] se propone una variante de R-tree llamada R+-tree. Esta variante resuelve el problema de sobreposición de rectángulos de tal manera que si el MBR de un objeto intersecta mas de un rectángulo en los nodos intermedios será almacenado en varias páginas.

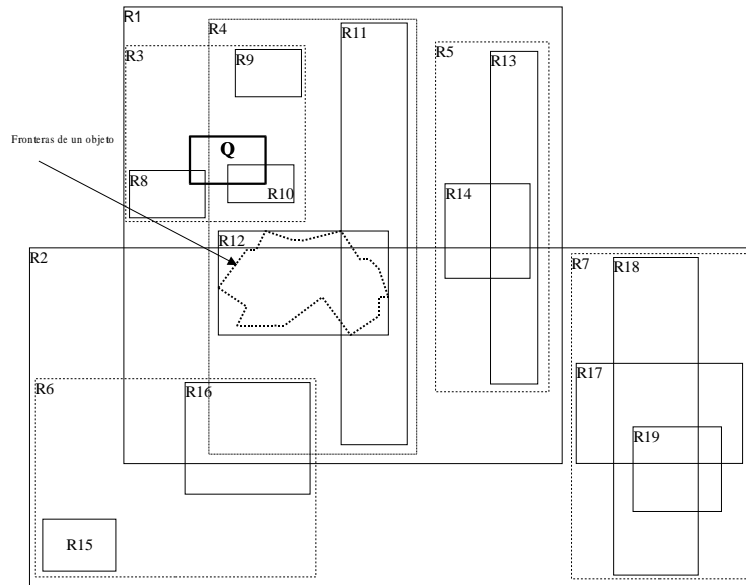


Figura 1: Rectángulos organizados en un R-tree

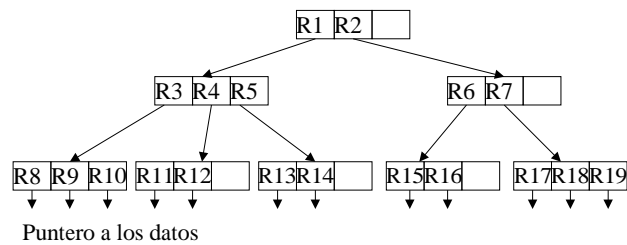


Figura 2: R-tree de los rectángulos de la 1

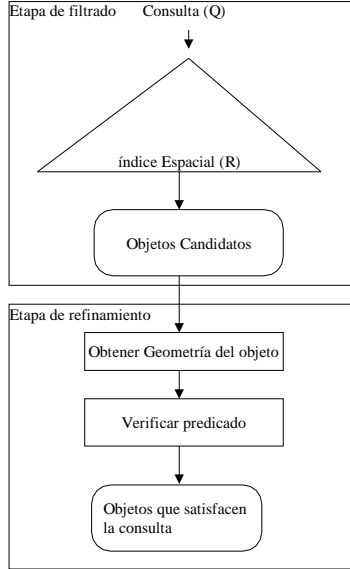


Figura 3: Etapas contempladas en el procesamiento de consultas

Otra variante fue propuesta por [BKSS90] la cual introduce una política de inserción llamada "reinserción forzada" la cual consiste en no dividir un nodo en forma inmediata cuando éste se llena. En lugar de ello propone eliminar  $p$  entradas del nodo y reinsertarlas en el árbol. Además los algoritmos minimizan la sobreposición de regiones, los perímetros de MBR y maximizan la utilización del almacenamiento.

### 3 Procesamiento de consultas espaciales

El procesamiento de consultas espaciales constituye una de las áreas de investigación en bases de datos espaciales que ha recibido bastante atención durante el último tiempo [PLC00]. El costo de procesar una consulta espacial puede llegar a ser muy alto ya que los datos espaciales son mas complejos que los datos alfanuméricos. Se han propuesto varios métodos para procesar consultas espaciales [BKS93, BKSS94, HJR97]. La mayoría de tales métodos consideran dos etapas a saber: *filtrado* y *refinamiento* (figura 3). En la etapa de filtrado se utiliza el índice espacial (R-tree) para seleccionar objetos candidatos de la respuesta. Luego, en la etapa de refinamiento, se utiliza la geometría del objeto para decidir si definitivamente cumple con las restricciones de la consulta. Por ejemplo al procesar la consulta  $Q$  de la figura 1, la etapa de filtrado genera el conjunto  $\{R8, R10\}$ . Luego, en la etapa de refinamiento, se recupera la geometría de los objetos cuyos MBR son  $R8$  y  $R10$  respectivamente y se verifica con el predicado espacial dado en la consulta  $Q$ . De esta forma el conjunto de objetos generado por la etapa de filtrado es siempre un superconjunto del conjunto de objetos que conforman la respuesta de la consulta.

### 4 Método propuesto

Sea  $C$  un conjunto de objetos espaciales del mismo tipo (polígonos o polilíneas) y  $Q$  (*Encontrar todas las comunas con una área mayor o igual a 5000 km<sup>2</sup>*) una consulta espacial cuyo predicado considera atributos derivados de la geometría de los objetos de  $C$ ; y  $R$  el índice (R-tree) para  $C$ .

Bajo el supuesto de que no existe un índice (árbol B) sobre el atributo derivado, para responder  $Q$  es necesario recorrer uno a uno los objetos de  $C$  y verificar el predicado, lo cual implica acceder todos los bloques utilizados para almacenar los objetos.

Una forma alternativa consiste en apoyarse en el índice  $R$  y utilizar las aproximaciones (MBR) de los objetos espaciales para responder  $Q$ . Similar al esquema descrito en [BKSS94], para el procesamiento

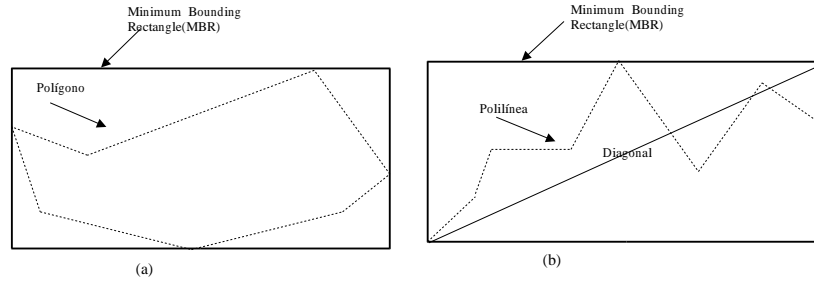


Figura 4: (a) Relación entre en área de un polígono y su MBR. (b) Relación entre la longitud de una polilínea y la longitud de la diagonal principal de su MBR.

Conjunto	Area Mínima	Area Máxima	Area Promedio	# Polígonos
1	671	43000800	876490	135
2	1515	2929060610	1316880	7973
3	745	1979829760	609559	7060
4	1026	37653400	826945	1561
5	129	1835451390	10056073	4388

Figura 5: Descripción de los conjuntos de objetos poligonales

de consultas espaciales, el método considera dos etapas (ver figura 3). La primera etapa consiste en seleccionar el conjunto de objetos que pueden satisfacer  $Q$ . Estos objetos se seleccionan usando sus correspondientes aproximaciones (MBR's). El conjunto obtenido en esta etapa es un superconjunto del conjunto de objetos espaciales que conforma la respuesta definitiva de  $Q$ . En la segunda etapa se accesa la geometría de los objetos obtenidos en la etapa anterior y se verifica si el objeto satisface el predicado de  $Q$ .

Con el propósito de verificar la calidad de la etapa de filtrado se estudiaron dos atributos derivados: el área (para el caso de polígonos) y la longitud (para el caso de polilíneas). Al observar el área de un polígono ( $A$ ) y el área de su MBR ( $A'$ ) podemos establecer la siguiente relación:  $A \leq A'$ . (figura 4-a). De manera similar la longitud de la diagonal principal de un MBR ( $P'$ ) y la longitud de una polilínea ( $P$ ) cumplen la siguiente relación:  $P' \leq P$  (figura 4-b).

Para los atributos derivados- área y longitud- se analizaron, en cada caso, 5 conjuntos obtenidos del proyecto SEQUOIA [SEQ00]. Algunos parámetros de los conjuntos se encuentran en la figuras 5 (polígonos) y 6 (polilíneas). Se realizaron consultas las cuales consideraron predicados sobre los atributos derivados que implicaban devolver 90, 80, 70, 60, 50, 40, 30, 20 y 10 por ciento de objetos sobre el total contenido en el conjunto. Para cada uno de estos porcentajes se obtuvo el porcentaje estimado en la etapa de filtrado (figuras 7 y 8).

De las figuras (7 y 8) se puede inferir que la etapa de filtrado estima bastante bien el conjunto de objetos de la respuesta. Por ejemplo al considerar el atributo derivado longitud y consultas sobre este atributo y de selectividad entre un 10 y 20%, la etapa de filtrado sobrestima en sólo un 1% el conjunto

Conjunto	Longitud Mínima	Longitud Máxima	Longitud Promedio	# Polilíneas
1	0	28476	1570	50001
2	5	20079	1629	30301
3	10	34133	1670	35351
4	10	32651	2258	28651
5	10	24189	2029	38001

Figura 6: Descripción de los conjuntos de objetos polilíneas

	Porcentaje de objetos seleccionados por la consulta								
Conjunto	90	80	70	60	50	40	30	20	10
1	97	95	91	87	85	80	74	61	38
2	97	92	85	77	68	59	47	33	20
3	97	95	90	80	70	58	47	35	20
4	98	91	84	76	68	58	46	34	17
5	96	91	87	78	69	58	48	35	18

Figura 7: Porcentaje de objetos que satisfacen la consulta versus porcentaje de objetos estimado por la etapa de refinamiento

	Porcentaje de objetos seleccionados por la consulta								
Conjunto	90	80	70	60	50	40	30	20	10
1	93	84	75	64	53	42	32	21	10
2	93	85	76	66	54	43	32	21	11
3	92	84	75	64	53	42	31	21	11
4	92	84	75	64	54	42	32	21	10
5	93	84	75	65	54	42	31	21	11

Figura 8: Porcentaje de objetos que satisfacen la consulta versus porcentaje estimado

respuesta (figura 8).

Los resultados presentados en las figuras 7 y 8 promueven la idea de utilizar las aproximaciones (MBR) de los objetos para estimar el conjunto respuesta de una consulta con atributos derivados (área y longitud). De esta forma aparece como conveniente utilizar el índice  $R$  (R-tree) del conjunto de objetos en la etapa de filtrado. En la figura 9 se muestra un algoritmo para la etapa de filtrado. Este algoritmo está especializado para filtrar predicados sobre el atributo derivado área.

```

/* Dado un R-tree con raíz  $T$  obtiene un conjunto de objetos que sobreestima */
/* el conjunto de la respuesta de una consulta espacial que contempla el atributo */
/* área */
ConjuntoOid Filtrar(MBR T, float area) {
/* F conjunto de objetos espaciales cuyo MBR tiene una área  $\geq$  area
 $F = \emptyset$ 
if ( $T$  no es una hoja)
  for(cada entrada  $E \in T$ )
    if( $E.I.Area() \geq area$ )  $F = F \cup Filtrar(E.pchild, area)$ 
else /*  $T$  es una hoja */
  for(cada entrada  $E \in T$ )
    if( $E.I.Area() \geq area$ )  $F = F \cup \{E.Oid\}$ 
}
return F
}

```

Figura 9: Algoritmo para la etapa de filtrado para consultas que contemplan el atributo derivado área

## 5 Conclusiones

En este artículo se estudiaron alternativas para procesar consultas sobre conjuntos de objetos espaciales en las cuales el predicado establece restricciones sobre atributos derivados de la geometría de los obje-

tos y para los cuales no existe un índice. Específicamente se estudiaron consultas sobre los atributos derivados área de un polígono y longitud de polilíneas. Se utilizó la aproximación MBR en la etapa de filtrado. Se experimentó considerando 5 conjuntos de datos espaciales para cada atributo. Los resultados permiten inferir que es conveniente procesar consultas espaciales sobre atributos derivados utilizando la aproximación MBR en la etapa de filtrado y por lo tanto, en ausencia de índice para el atributo derivado, el uso de un R-tree para estimar el conjunto de objetos aparece como una alternativa promisoría.

Como trabajo futuro sería interesante analizar consultas que consideren, en su predicado, restricciones espaciales, no espaciales y restricciones sobre atributos derivados. También es de interés diseñar algoritmos para la reunión espacial (Spatial Join) en la cual se consideren estos atributos. Finalmente es necesario disponer de un modelo de costo para estimar el número de operaciones de I/O para el procesamiento de consultas sobre atributos derivados.

## Referencias

- [BKS93] 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.
- [BKSS90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The  $r^*$ -tree: An efficient and robust access method for points and rectangles. In *ACM SIGMOD Conference on Management of Data*. ACM, 1990.
- [BKSS94] 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, Minnesota, USA, 1994.
- [BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley / ACM Press, 1st edition, 1999.
- [GG98] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.
- [Gut84] Antonin Guttman. A dynamic index structure for spatial searching. In *ACM SIGMOD Conference on Management of Data*, pages 47–57, Boston, 1984. ACM.
- [Gut94] Ralf Guting. An introduction to spatial database systems. In *VLDB Journal*, volume 3, pages 357–399, 1994.
- [HJR97] Yun-Wu Huang, Ning Jing, and Elke Rundensteiner. Spatial joins using r-trees: Breadth-first traversal with global optimizations. In *23rd Conference on Very Large Data Bases*, pages 396–405, Athens, Greece, 1997.
- [LJF94] 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.
- [PLC00] 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.
- [PLLC99] 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 Systems for Advanced Applications*, pages 161–168, 1999.
- [SEQ00] SEQUOIA. Iglobal change research project. In <http://s2k-ftp.cs.berkeley.edu>. Accesado el 29 de enero de 2002, 2000.
- [SRF87] Timos Sellis, Nick Roussopoulos, and Christos Faloutsos. The  $r^+$ -tree: A dynamic index for multi-dimensional objects. In *13th Conference on Very Large Data Bases*, pages 507–518, Brighton, England, 1987.
- [SRF97] Timos Sellis, Nick Roussopoulos, and Christos Faloutsos. Multidimensional access methods. trees have grown everywhere. In *23rd Conference on Very Large Data Bases*, pages 13–14, Athens, Greece, 1997.