Recuperación Eficiente de Información Multimedia

Luis Britos, María E. Di Gennaro, Jacqueline Fernández, Veronica Gil-Costa, Fernando Kasián, Verónica Ludueña, Marcela Printista, Nora Reyes, Patricia Roggero
LIDIC, Departamento de Informática, Fac. de Ciencias Físico Matemáticas y Naturales
Universidad Nacional de San Luis
{*lebritos,mdigena jmfer, gvcosta, fkasian,vlud, mprinti,nreyes, proggero*}@unsl.edu.ar

Edgar Chávez
Escuela de Ciencias Físico–Matemáticas
Universidad Michoacana de San Nicolás de Hidalgo
elchavez@umich.mx

Resumen

En general, es tan difícil para los usuarios que intentan recuperar información multimedia poder especificar claramente sus intereses a través de una consulta bien definida, como para los diseñadores del sistema decidir qué características de los objetos multimedia pueden resultar relevantes. La forma en que los datos multimedia se representan, cómo se almacenan y el costo de transferirlos, entre distintos niveles de la jerarquía de memoria o sobre una red, afectan directamente las respuestas del sistema. Dada una consulta, el objetivo clave de un sistema de recuperación de información es obtener aquello que podría ser útil o relevante para el usuario, en general haciendo uso de un índice especialmente diseñado para responder a las consultas de manera eficiente.

Así, nuestra línea de investigación tiene como principal objetivo desarrollar herramientas eficientes para la recuperación de información multimedia. Se investigan nuevas técnicas que soporten la interacción con el usuario, nuevas estructuras de datos (índices) capaces de manipular eficientemente datos multimedia y que permitan manejar grandes volúmenes de este tipo de datos.

Palabras Claves: *Recuperación de Información, Bases de Datos Multimedia, Indexación, Paralelismo.*

Contexto

Esta línea de investigación se encuentra enmarcada dentro del Proyecto Consolidado 30310 de la Universidad Nacional de San Luis y en el Programa de Incentivos (código 22/F034): "Nuevas Tecnologías para el Tratamiento Integral de Datos Multimedia", dentro de la línea "Recuperación de Datos e Información Multimedia", desarrollada en el ámbito del Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la UNSL.

En este marco se pretende avanzar en la integración de las investigaciones sobre adquisición, preprocesamiento y análisis de datos no estructurados y su aplicación en dominios no convencionales. Se espera, como principal aporte de esta propuesta, incorporar información no estructurada en los procesos de toma de decisiones y resolución de problemas que quedan sin considerar en los enfoques clásicos.

Dentro de este contexto nuestra línea se dedica, principalmente, al diseño de índices eficientes que sirvan de apoyo a sistemas de recuperación de información orientados a datos no estructurados, en particular datos multimedia. Se espera así contribuir a estos sistemas obteniendo índices más eficientes para memorias jerárquicas, con I/O eficiente y capaces de manejar grandes volúmenes de datos. Se propone analizar las estructuras de datos existentes, proponer optimizaciones o diseñar nuevas estructuras, para manipular y recuperar algunos de los tipos de datos no estructurados que aparecen en entornos multimedia, considerando en algunos casos la paralelización de los mismos con el objetivo de hacer aún más eficiente la recuperación.

1. Introducción y Motivación

Los sistemas de computación hacen uso intensivo de información estructurada, es decir datos elementales o estructuras, generadas con un formato específico. Una característica principal en estos casos, es que la estructura o formato de esta información puede ser fácilmente interpretada y directamente utilizada por un programa de computadora. Pero el hecho de restringirse al uso de este tipo de información conduce, muchas veces, a representar una visión parcial del problema y dejar fuera información

que podría ser importante para la resolución efectiva del mismo. En este contexto gran parte de la información que se requiere para la toma de decisiones y la resolución de problemas de índole general proviene de información no estructurada.

En general, para responder eficientemente consultas para recuperación de información sobre bases de datos multimedia se utilizan diferentes métodos de acceso o índices [13, 5, 11], principalmente por el volumen de datos con el que se trabaja.

Un enfoque prometedor para implementar sistemas de recuperación usando búsqueda por similitud es una búsqueda basada en contenidos, la cual usa el dato multimedia mismo. Para calcular la similitud entre dos objetos multimedia, se debe definir una función de distancia. Dicha función mide la similitud, o más bien la disimilitud, entre dos objetos. En muchos casos para modelar la similitud de objetos multimedia se transforman los objetos en puntos de un espacio vectorial, el cual es un tipo particular de espacio métrico. Cada objeto es representado por un vector de características o descriptor, generalmente de alta dimensionalidad. Sobre espacios vectoriales se han definido numerosas funciones de distancia (distancia Euclidiana). El tipo de aplicación, las características a explotar o la dimensionalidad son aspectos fundamentales a considerar para definir la mejor función de distancia a utilizar. Por lo tanto, es necesario resolver un problema de optimización.

El concepto de búsqueda por similitud se puede definir a partir del concepto de espacios métricos, que da un marco formal independiente del dominio de aplicación. Un espacio métrico está compuesto por un *universo* $\mathcal{U}$ de objetos y una función de distancia $d : \mathcal{U} \times \mathcal{U} \longrightarrow R^+$, que satisface las propiedades que la hacen una métrica. Las consultas por similitud, sobre una *base de datos* $\mathcal{S} \subseteq \mathcal{U}$, son usualmente de dos tipos: *Búsqueda por rango:* (recuperar los elementos de $\mathcal{S}$ a distancia r de un elemento q dado) y *Búsqueda de los k vecinos más cercanos* (dado q recuperar los k objetos más cercanos a q en $\mathcal{S}$).

En el caso de los espacios métricos, la función de similitud (distancia) mide el mínimo esfuerzo (costo) necesario para transformar un objeto en otro. Dependiendo de los tipos de datos multimedia reales la función de similitud puede ser muy compleja y puede no satisfacer las propiedades de una métrica.

Si la base de datos $\mathcal{S}$ posee n objetos, las consultas pueden ser respondidas llevando a cabo n evaluaciones de distancia. Sin embargo, en la mayoría de las aplicaciones las distancias son costosas de computar (comparación de huellas digitales), por lo que la búsqueda secuencial no sirve para problemas de tamaño medio o grande, que son los tamaños más habituales de las bases de datos multimedia. Así debemos preprocesar la base de datos, construyendo un índice, para que las consultas puedan ser respondidas con la menor cantidad de cálculos de distancia. Además, es probable que la base de datos, el índice o ambos no puedan almacenarse en memoria principal con lo cual se debe considerar minimizar el número de operaciones de E/S realizadas en cada operación, tener siempre presente la existencia de la jerarquía de memorias y tratar de lograr mayor eficiencia a través de paralelizar las operaciones a realizar.

En suma, esta propuesta se enfoca en mejorar las herramientas de recuperación desarrollando nuevas técnicas que soporten la interacción con el usuario, diseñando estructuras de datos (índices), capaces de manipular eficientemente grandes volúmenes de datos multimedia y facilitando la realización de operaciones sobre los mismos de modo de acercarse a la madurez de las bases de datos tradicionales.

2. Líneas de Investigación

Se pretende investigar sobre distintos aspectos de los sistemas de recuperación de información multimedia: diseñar nuevos índices, definir representaciones que reflejen características de interés de los objetos y manejar distintas operaciones sobre estos tipos de bases de datos, considerando trabajar eficientemente sobre grandes volúmenes de datos.

Diseño de Índices

Un catálogo importante de índices para espacios métricos aparece en [11, 5, 13]. La mayoría usan la desigualdad triangular para evitar el análisis secuencial de la base de datos. La distancia entre la consulta q y los objetos de la base de datos puede ser estimada calculando de antemano algunas distancias a objetos distinguidos llamados *pivotes* y sin calcular las distancias reales desde q a los objetos de la base de datos durante una búsqueda. Otra técnica común es indexar a través de una partición del espacio en regiones denominadas *particiones compactas*.

Existen dos posibles situaciones por el tipo de base de datos con la que se va a trabajar, que determinan una característica importante que debe tener el índice que la manipulará: los objetos de la base de datos se conocen de antemano y por lo tanto el índice se creará de una sola vez y se realizarán consultas sobre él (índices estáticos). O no se conocen los objetos de la base de datos de antemano y por lo tanto

el índice se debe ir creando a medida que arriban los elementos y preferentemente de manera incremental (índices dinámicos). Las estructuras estáticas se benefician desde el conocimiento de la base de datos seleccionando los mejores puntos de referencia para una estructura de datos determinada, lo cual no es posible en las estructuras de datos dinámicas donde tanto los objetos como las consultas arriban al azar.

Índices Estáticos

En este caso, al conocer de antemano los elementos a indexar, es posible elegir con más información cómo hacerlo de manera tal que las búsquedas sean eficientes. Sin embargo, hay ejemplos como el del *Árbol de Aproximación Espacial*, SAT [8], que por ser una estructura estática debería ser más eficiente que la versión dinámica, el DSAT [9], y no lo es. En estos casos ha sido posible investigar alguno de los motivos por los que la versión dinámica, usando menos información, proporciona búsquedas mas rápidas. En nuestras investigaciones hemos detectado que una condición clave para mejorar la performance de SAT es modificar la estrategia de selección de vecinos, es por ello que se está trabajando en diferentes heurísticas, como la de utilizar un orden de inserción arbitrario de los vecinos o hasta elegirlos de manera totalmente contraria a lo que la versión original lo hacía y se están consiguiendo en este caso resultados preliminares muy interesantes [4].

En algunos sistemas de recuperación de información que trabajan con datos masivos, con vistas a mejorar el compromiso con el usuario entre completitud de las respuestas a una consulta por similitud y tiempo de respuesta del sistema, se puede adoptar un enfoque aproximado. En los enfoques aproximados se mejora el tiempo de respuesta a estas consultas gracias a bajar los tiempos requeridos, pero a costa de obtener una respuesta no exacta a la consulta. Este enfoque es valioso cuando el método, a pesar de no obtener la respuesta exacta a la consulta devuelve los resultados más similares a la misma. Por lo tanto, se está investigando un nuevo índice que permita obtener la respuesta aproximada a una consulta por similitud, logrando que sea de alta calidad (que obtenga buenos valores en las métricas de *Precision* y *Recall*) y minimizando cantidad de cálculos de distancia realizados y número de operaciones de E/S.

Índices Dinámicos

Aquí el interés está en mejorar el desempeño de índices dinámicos jerárquicos (árboles), que es el caso de algunos de los índices para espacios métricos. Estos índices dinámicos, en general, se construyen incrementalmente vía inserciones. De tal manera, la raíz del árbol es el primer objeto que llega, y esto se repite recursivamente en cada nivel del árbol.

En esta línea se ha propuesto una técnica donde el "buffering" logra un buen compromiso entre una estructura estática, construída con toda la información necesaria y una dinámica con conocimiento local de los datos. Entonces, en lugar de elegir al primer elemento como la raíz, se demora la selección hasta que hayan arribado suficientes elementos para estar en condiciones de realizar dicha selección, y de esta manera se toma una decisión en base a más información. Dado que las consultas arriban a un ritmo desconocido, para mantener el dinamismo es necesario contar con un índice que responda a las consultas con mejor desempeño que la técnica de fuerza bruta. La idea ha sido, entonces, dar una estructura propia al "buffer" de manera que fuera capaz de responder consultas. Es por ello que el índice del "buffer" debería ser rápido y eficiente. Esta técnica provee un marco adecuado para diseñar estructuras de datos dinámicas estables. Por lo tanto, tener un "buffer" en todos los niveles de una estructura jerárquica debería ser útil cuando se diseñan estrategias de ruteo para guiar las búsquedas, lo cual resulta un área promisoria de investigación [6].

En muchos casos los volúmenes de información con los que se debe trabajar (millones de imágenes en la Web), hacen necesario que los índices sean almacenados en memoria secundaria. En este caso, para hacerlos eficientes, no sólo se debe considerar que durante las búsquedas se realice el menor número de cálculos de distancia sino también, dado el costo de las operaciones sobre disco, se efectúe la menor cantidad posible de operaciones de E/S. Por ello, en esta línea nos hemos dedicado a diseñar índices especialmente adaptados para trabajar en memoria secundaria, logrando un buen desempeño de los mismos, principalmente en las búsquedas.

Hemos diseñado e implementado las siguientes estructuras *DSACL*-tree* y el *DSACL+-tree* [2], las cuáles son optimizaciones para memoria secundaria de la estructura propuesta en [1] y demostraron ser competitivas frente a otras de las estructuras conocidas tales como el *M-tree* y *DSA*-tree* y *DSA+-tree* [9]. Además, existen nuevas propuestas en evaluación que prometen ser aún más adecuadas para memoria secundaria. Por otro lado, nos proponemos optimizarlas todavía más gracias a la aplicación de técnicas de computación de alto desempeño, apli-

cando y comparando distintas estrategias de paralelización con el fin de determinar la más adecuada.

Diversificación de Resultados

La técnica de diversificación de resultados provee una manera de hacer frente a las preguntas ambiguas por medio de la reordenación de un conjunto de documentos recuperados como resultado de una consulta. Los enfoques actuales suelen ser ambiciosos y costosos, requieren $O(n^2)$ comparaciones de documentos con el fin de diversificar un ranking de n documentos. Una alternativa de menor costo y que permite mantener una buena calidad de los resultados es utilizar un enfoque que aplica las propiedades de espacios métricos, en el cual se reduce la sobrecarga que se produce por las comparaciones de documentos requeridas como resultado de la diversificación. Para este fin, la diversificacion de resultados se modela como una búsqueda por similitud en un espacio métrico, aprovechando las propiedades de este espacio para identificar de manera eficiente los documentos novedosos. En particular, se explota la propiedad desigualdad triangular para reducir drásticamente el número de comparaciones de documento requeridos. En este contexto se estudian técnicas de indexación existentes que permitan mejorar la eficiencia del proceso de diversificación y mantenengan una buena calidad de los resultados.

El trabajo presentado en [12] aplica el enfoque de espacios métricos por primera vez en el contexto de búsqueda de imágenes para mejorar la eficiencia del proceso de diversificación. En [7] se presenta el primer intento de aprovechar las propiedades de espacios métricos para la diversificación de documentos de texto. En este trabajo se manifiesta que el número de cálculos necesarios para determinar la novedad de un documento se puede reducir utilizando un algoritmo basado en pivotes, que selecciona documentos pivote disímiles (distantes en el espacio métrico).

Consultas sobre Bases de Datos Multimedia

Aunque las operaciones más comunes sobre bases de datos multimedia son las consultas por similitud (búsquedas por rango o de k-vecinos más cercanos), existen otras operaciones de interés entre las cuales se encuentran las distintas variantes del *join* por similitud. Para estas operaciones se consideran dos bases de datos A y B, ambas subconjuntos del mismo universo del espacio métrico $\mathcal{U}$. El resultado de cualquier operación de join por similitud entre A y B obtiene el conjunto de pares formados por un objeto de A y otro de B, tales que entre ellos se satisface el predicado de similitud considerado. Las variantes más conocidas del join por similitud son: el join por rango, el join de k-vecinos más cercanos y el join de vecino más cercano; aunque existen otras.

Formalmente, dadas $A, B \subseteq \mathcal{U}$, se define el *join por similitud* entre A y B ($A \underset{\Phi}{\bowtie} B$) como el conjunto de todos los pares (x, y), donde $x \in A$ e $y \in B$; es decir, $(x, y) \in A \times B$, tal que entre x e y se satisface el criterio de similitud considerado Φ. De acuerdo al criterio de similitud el join puede llamarse:

- *Join por Rango*: $A \bowtie_r B = \{(x, y) : x \in A,\ y \in B \wedge d(x, y) \leq r\}$.
- *Join de k-Vecinos Más Cercanos*: $A \bowtie_k B$ es el conjunto de k-pares, donde $\forall (x, y) \in A \bowtie_k B$, $x \in A$, $y \in B$ y $\forall (u, v) \in ((A \times B) \setminus (A \bowtie_k B))$, $u \in A$, $v \in B$, entonces $d(x, y) \leq d(u, v)$. En caso de empate elegimos cualquier conjunto de k-pares que satisfaga la condición.

En el caso particular en que $A = B$ el join por similitud se denomina *auto-join*. Existen dos situaciones distintas sobre las que se puede trabajar, para resolver el join por similitud: que ambas bases de datos se encuentren indexadas, cada una por separado; o que ambas bases de datos se indexen conjuntamente, con un índice diseñado para el join.

Como calcular cualquiera de las variantes del join por similitud de manera exacta sobre conjuntos de datos masivos es muy costoso [10], vale la pena obtener más rápidamente una respuesta aproximada al join, siempre y cuando se pueda dar una respuesta rápida y de buena calidad. Para ello, estamos investigando un nuevo índice, diseñado para memoria secundaria y que permita obtener una buena respuesta realizando pocos cálculos de distancia y la menor cantidad de operaciones de E/S posibles.

PostgreSQL es el primer sistema de base de datos que permite realizar consultas por similitud sobre algunos atributos, particularmente indexación para búsquedas de k-vecinos más cercanos (KNN-GiST indexes). Estos índices pueden ser usados sobre texto, comparación de ubicación geoespacial, etc.. Sin embargo, los índices K-NN GiST proveen plantillas para índices con estructura de árbol balanceado (B-tree, R-tree), aunque el balance no siempre es bueno para los índices que se utilizan en búsquedas por similitud [3]. Además este tipo de consultas no está disponibles para todo tipo de datos métricos. Así, es importante proveer un manejador de bases de datos capaz de administrar bases de datos métri-

cas que manejen todos los posibles datos métricos y todas las operaciones de interés sobre ellos.

3. Resultados

Se ha comprobado experimentalmente que las estrategias de "buffering" mejoran el desempeño en un índice dinámico [6]. Se seleccionó el Árbol de Aproximación Espacial Dinámico (DSA-tree) [9] y se obtuvo una mejora sistemática en los costos de las consultas usando un "buffer" en el primer nivel del árbol. En particular, se verificó que esta estructura es mejor que su versión estática [9], por dejar como "vecinos" de un nodo objetos alejados, permitiendo así avanzar en la exploración espacial a "pasos más grandes". Entonces, se pretende analizar el efecto de elegir como vecinos objetos cercanos y lejanos. Si clasificáramos los objetos por distancia a la raíz, usando la información de su histograma de distancias, se podría elegir con esa misma densidad a los vecinos, para mejorar su desempeño y que esto pueda aplicarse a otros índices jerárquicos.

En este mismo sentido, se implementaron dos versiones: DSACL*-tree y DSACL+-tree, que trabajan con grandes volúmenes de datos, por haber sido diseñadas para memoria secundaria y que mostraron ser competitivas contra otras estructuras diseñadas para tal fin [2]. Se espera lograr para estos índices una implementación paralela eficiente.

4. Formación de Recursos

Considerando la importancia de la formación, para contribuir al desarrollo de sistemas de recuperación de información multimedia, se están capacitando los siguientes investigadores:

Tesis de Doctorado en Cs. de la Computación: uno de los integrantes se encuentra definiendo su plan de doctorado sobre temas de diseño y optimización de índices para búsquedas por similitud, para aplicaciones de minería de datos multimedia.

Tesis de Maestría en Cs. de la Computación: una sobre índices dinámicos eficientes sobre datos masivos (con una beca de posgrado de la UNSL), una sobre índices dinámicos para búsqueda aproximada por similitud sobre datos masivos, una sobre índices para join aproximado por similitud sobre datos masivos y otra un sistema para administrar bases de datos métricas.

Referencias

[1] M. Barroso, N. Reyes, and R. Paredes. Enlarging nodes to improve dynamic spatial approximation trees. In *Proc. of the 3rd SISAP*, pages 41–48. ACM Press, 2010.

[2] L. Britos, A. M. Printista, and N. Reyes. Dynamic spatial approximation trees with clusters for secondary memory. In *XVI CACIC Selected Papers*, 2011.

[3] E. Chávez, V. Ludueña, and N. Reyes. Revisiting the VP-forest: Unbalance to improve the performance. In *Proc. de las JCC08*, 26, 2008.

[4] E. Chávez, V. Ludueña, N. Reyes, and P. Roggero. Reaching near neighbors with far and random proxies. In *CCE, 8th Int. Conf. on*, pages 1–8, oct. 2011.

[5] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM*, 33(3):273–321, sep 2001.

[6] E. Chávez, N. Reyes, and P. Roggero. Delayed insertion strategies in dynamic metric indexes. In *SCCC*, pages 34–42, 2009.

[7] V. Gil-Costa, R. Santos, C. Macdonald, and I. Ounis. Sparse spatial selection for novelty-based search result diversification. In *SPIRE*, pages 344–355, 2011.

[8] G. Navarro. Searching in metric spaces by spatial approximation. *VLDBJ*, 11(1):28–46, 2002.

[9] G. Navarro and N. Reyes. Dynamic spatial approximation trees. *Journal of Experimental Algorithmics*, 12:1–68, 2008.

[10] R. Paredes and N. Reyes. Solving similarity joins and range queries in metric spaces with the list of twin clusters. *JDA*, 7:18–35, 2009.

[11] H. Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. 2005.

[12] R. van Leuken, L. Garcia, X. Olivares, and R. van Zwol. Visual diversification of image search results. In *Proc. of the 18th*, WWW '09, pages 341–350. ACM, 2009.

[13] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Springer-Verlag., 2005.