

***Geometría Computacional y Bases de datos:
Búsquedas por Rangos y Separabilidad geométrica***

Edilma Olinda Gagliardi ¹

LIDIC²

Departamento de Informática

Facultad de Ciencias Físico, Matemáticas y Naturales

Universidad Nacional de San Luis, Argentina

oli@unsl.edu.ar

Fax: 54-2652-430224

Gregorio Hernández Peñalver

Departamento de Matemática Aplicada

Facultad de Informática

Universidad Politécnica de Madrid, España

gregorio@fi.upm.es

Fax: 34-91-3367426

Resumen

Un problema que se presenta a menudo en Bases de datos es el estudio de los rangos y las consultas por rangos, denominado *Búsquedas por Rangos*. Este problema tratado desde una perspectiva geométrica nos permite diseñar y analizar los algoritmos y estructuras de datos utilizadas con herramientas propias de la Geometría Computacional.

En este trabajo presentamos una introducción a la temática, relacionándola específicamente a otra línea de investigación vigente de la Geometría: *Separabilidad geométrica*.

El objetivo de esta propuesta es mostrar un trabajo de investigación, en el que damos el estado del arte del tema expuesto, presentando los aspectos teóricos y prácticos relevantes para las búsquedas por rangos y separabilidad de objetos geométricos. Realizamos una vinculación entre ambas, proponiendo nuevas formas de búsquedas en el espacio, como temas de investigación y desarrollo.

Palabras Claves:

Búsquedas por rangos, Separabilidad geométrica, rangos simpliciales, rangos semiespaciales.

¹ Integrante de RITOS2

² Laboratorio de Investigación y Desarrollo en Inteligencia Computacional. Director: Dr. Raúl Gallard.

• Parcialmente suvencionado por *Proyecto de la UPM de AI2002-1010-2.43 Geometría Computacional*.

1. Introducción

La Geometría Computacional es una disciplina que brinda un marco teórico y formal para el diseño de estructuras y análisis de algoritmos requeridos para dar soluciones a problemas que surgen en las más diversas áreas de la Informática. En ocasiones, descubrir que los datos de un problema verifican propiedades geométricas sirve para poder aplicar técnicas especiales, que permiten describir soluciones óptimas. La Geometría se interesa por demostrar la existencia de la solución de un problema y por encontrar los algoritmos y estructuras de datos eficientes, medidos respecto de su complejidad temporal y espacial respectivamente [AHU74]. Por tanto, esta disciplina forma parte de la teoría del diseño y análisis de algoritmos y estructuras de datos [Abe00], [BKOS97], [BY98], [GO97], [SU00], [Tou85], [Tou92].

Nuestra introducción al estudio de la disciplina se basa en la relación existente con las bases de datos: un problema que se presenta a menudo es el estudio de los rangos y las consultas por rangos, denominado *Búsqueda por Rangos*. Esta temática vista desde una perspectiva geométrica, nos permite diseñar y analizar los algoritmos y estructuras de datos utilizadas con instrumentos propios de la Geometría Computacional [AE98], [Aga97], [AM94], [BF79], [Mat94].

Una línea de investigación vigente y con resultados recientes en Geometría, es la *Separabilidad de objetos geométricos* [Sea02], [Meg79]. Los criterios de separabilidad tienen aplicaciones interesantes, como por ejemplo el análisis de imágenes, clasificación de datos, etc. Siempre que sea necesario discriminar objetos en un espacio de trabajo, por algún atributo del mismo, los criterios de separabilidad juegan un papel importante.

Nuestra propuesta consiste en vincular las Búsquedas por rangos con la Separabilidad geométrica, por medio de cuñas, bandas y combinaciones de ellas.

En este sentido, las regiones se determinan en base a atributos propios de los objetos geométricos y de su ubicación en el espacio. Buscamos generar subregiones del espacio que se distingan por las propias características de sus individuos. Las consultas, básicamente, se reducen a casos particulares de las búsquedas por rangos.

En el presente trabajo, primeramente, introducimos las Búsquedas por rangos, luego las nociones de Separabilidad y la vinculación propuesta entre ambas temáticas. Por último, brindamos nuestras conclusiones y perspectivas de trabajos futuros.

2. Búsquedas por Rangos

La Búsqueda por rangos es uno de los problemas centrales en Geometría Computacional, no sólo por la variedad de aplicaciones que posee sino porque, además, una gran cantidad de problemas geométricos pueden resolverse observándolos como problemas relacionados a las búsquedas por rango.

Un *Espacio de Rangos* es un sistema $\Omega = (U, F)$, donde U es un conjunto de objetos geométricos y F es una familia de subconjuntos de U . Los elementos de F son llamados *Rangos* de Ω . El sistema Ω es llamado *Espacio de Rango Finito* si el conjunto U es finito.

Algunos ejemplos de espacios de rangos generales son:

$$\Omega_1 = (R^d, \{ h / h \text{ es un semiespacio en } R^d \}), \quad \Omega_2 = (R^d, \{ h / h \text{ es una bola en } R^d \})$$

Un problema de búsqueda por rangos puede expresarse del siguiente modo: Dados un espacio de rangos $\Omega = (U, F)$, S un subconjunto de objetos de U y R un rango de F , consultar los objetos geométricos que están en $S \cap R$. En este caso, a R se lo llama *rango de consulta* (query range).

Para resolver este problema de búsqueda por rangos se desea diseñar un algoritmo eficiente que resuelva tal intersección. Si ocurre que se dan los conjuntos S y R y la consulta se realiza una única vez o circunstancialmente, entonces el cálculo de la intersección se soluciona simplemente por medio de un algoritmo que revise para cada objeto de S si pertenece o no a R . Pero el problema real para el cual tiene sentido este estudio, es que el conjunto de objetos geométricos S , se da previamente y posteriormente se realizan repetidas consultas por la intersección, variando el rango R . Es en estos casos donde tiene sentido diseñar una estructura de datos apropiada para almacenar el conjunto S , de forma tal que frente a cada consulta por rango R , ésta pueda ser respondida eficientemente.

La Búsqueda por rangos, esencialmente, consiste en buscar los objetos geométricos que contiene una determinada región (rango) del espacio de objetos geométricos. Entonces, observando la figura o forma de las regiones correspondientes a los rangos de consultas, podemos hacer una clasificación en cuatro tipos de

búsqueda por rangos: *Búsqueda por Rangos Ortogonales (BRO)*, *Búsqueda por Rangos SemiAlgebraicos (BRSA)*, *Búsqueda por Rangos SemiEspaciales (BRSp)* y *Búsqueda por Rangos Simpliciales (BRsx)*.

En cuanto a las consultas, dado un rango de consulta puede interesarnos efectuar alguna de las siguientes clases sobre él: *Consultas de Recuento (range counting)*, *Consultas de Reporte (range reporting)*, *Consultas Booleanas (range emptiness)* y *Consultas Extremales*.

Para unificar los diferentes tipos de consultas, un problema de este tipo puede modelizarse del siguiente modo: Sea el espacio de rangos $\Omega = (U, F)$, sea $S \subseteq U$ y sea $(D, +)$ un semigrupo. Para cada objeto $p \in S$ le asignamos una valoración llamada peso del objeto, mediante una función $w: S \rightarrow D$. Entonces, de acuerdo a la clasificación vista anteriormente tenemos:

Range counting: $\sum_{p \in S \cap R} w(p)$ y $w(p) = 1$, para todo p perteneciente a S . Por ejemplo, $(D, +)$ es $(\mathbb{Z}, +)$.

Range reporting: $\{p / p \in S \cap R\}$, $w(p) = \{p\}$, para todo p perteneciente a S . Por ejemplo, $(D, +)$ es $(2^D, \cup)$.

Range emptiness: 0 si $S \cap R = \emptyset$; 1 en otro caso; $w(p) = 1$, para todo p perteneciente a S . Por ejemplo, $(D, +)$ es $(\{0,1\}, \vee)$.

Extremales: $\min \{w(p) / p \in S \cap R\}$, o, $\max \{w(p) / p \in S \cap R\}$; $w(p)$ = valor de la primera coordenada para todo p perteneciente a S . Por ejemplo, supongamos que $(D, +)$ es (\mathbb{R}, \min) , o análogamente, $(D, +)$ es (\mathbb{R}, \max) .

Asumimos que las operaciones de semigrupo pueden ser ejecutadas en tiempo constante. Como es típico en Geometría Computacional, usamos el modelo de computación Real RAM (*Real Random Access Machine*) donde los datos de entrada contienen números reales arbitrarios y cada operación aritmética con números reales tiene costo unitario. Asumimos también que las raíces de los polinomios de un grado fijo, pueden ser calculados en tiempo constante.

La mayoría de las estructuras de datos para búsquedas o consultas por rango, son construidas en forma recursiva, dividiendo el espacio de objetos geométricos en varias regiones, con propiedades geométricas deseables sobre ellas. Estas estructuras de datos son referidas como esquemas de descomposición jerárquicos. El primer paso para la creación de la estructura de datos para S , es la construcción de una familia de subconjuntos canónicos de S . En la mayoría de los casos, esta familia no es más que una partición de S en subconjuntos disjuntos, tal que la unión de todos ellos es S . El árbol es consultado en forma top down, puede ser una búsqueda en profundidad (*depth first search*) o cualquier otra comenzando desde la raíz. Los análisis de las complejidades tanto en tiempo de construcción de la estructura de datos, en espacio que ocupa y en tiempos de respuestas de las consultas, dependen de la dimensión del espacio, de la cardinalidad de S , de la estrategia de particionamiento de S y del tamaño de la respuesta de la consulta.

Según mencionamos antes, la figura o forma del rango, determina el tipo de búsqueda por rangos. Dado un espacio de rangos $\Omega = (U, F)$, la familia F puede estar formada por los siguientes rangos:

Búsquedas por rangos Ortogonales

Los rangos son d -rectángulos, cada uno de la forma $\prod_{i=1..d} [a_i, b_i]$ con $a_i, b_i \in \mathbb{R}^d$, con lados paralelos a los ejes de coordenadas del espacio subyacente. Ésta es una abstracción del problema de búsqueda por múltiples claves, que a causa de sus numerosas aplicaciones, sobre todo en Bases de Datos, ha sido estudiado por muchos investigadores, a lo largo de las tres últimas décadas, buscando bajar las cotas en el tiempo de respuesta y en el espacio requerido. Por ello, se pueden hallar numerosos informes, monografías, tesis doctorales, libros, artículos, etc. que se refieren al tema [Ben75], [Ben79], [Ben80], [BF79], [Cha90a], [Cha90b], [Gün88], [Mel84], [Sam90a], [Sam90b], [Sam90c].

Búsqueda por rangos SemiAlgebraicos

Los rangos están definidos por conjuntos semialgebraicos [AE98], [AM94], [YDEP89], [YY85], [Y83]. Por ejemplo, el conjunto de todas las bolas cerradas de \mathbb{R}^2 , o el conjunto de todos los paraboloides en \mathbb{R}^3 . Para formalizar, consideremos una familia de subconjuntos de \mathbb{R}^d definida por $\Gamma = \{\Gamma_f(a) / a \in \mathbb{R}^p\}$. Los rangos son $\Gamma_f(a) = \{x \in \mathbb{R}^d / f(x, a) \geq 0\}$ ³, donde f es un polinomio de grado acotado en $(d+p)$ -variables $x_1, x_2, \dots, x_d, a_1, \dots, a_p$, que define el tipo de rango considerado (disco, cilindro, cono, bola, etc.); y a es una p -tupla que describe un

³ Estos son los conjuntos semialgebraicos.

rango específico de un tipo dado (por ejemplo, un disco determinado). Este problema está relacionado a diversos tópicos de investigación como el *problema del vecino más cercano* o el *problema de los k vecinos más cercanos*. Chazelle y Welz [CW89] dieron una solución con espacio lineal para un problema de búsqueda por rango circular en el plano. El tiempo de respuesta es $O(\sqrt{n} \log^2 n)$ y la estructura ocupa $O(n \log n)$ espacio. Y todos los puntos contenidos en el disco de consulta pueden ser reportados en $O(\log n + k)$ tiempo. Una forma de solucionar las búsquedas por rango de Γ , consiste en efectuar una transformación directa a búsquedas por rangos semiespaciales en un espacio de dimensión superior al corriente. Es lo que se denomina *linealización* del problema.

Búsquedas por rangos SemiEspaciales

Los rangos conforman el conjunto de todos los semiespacios cerrados de R^d . Así, en $d=2$ tenemos semiplanos. Las principales ideas se deben a Willard [Wil82], [WL85] sobre búsquedas por rangos con semiplanos, las que se han generalizado para búsquedas por rangos semiespaciales. Se considera una nube de puntos S en el plano de cardinalidad $n = 4^k$, y se construye el árbol en forma recursiva de la siguiente manera:

Para $k=0$, el árbol es un único nodo que almacena las coordenadas de un único punto de S , llamémosle p_1 . Para $k>0$, utilizando el Teorema del Ham-Sandwich [Meg79], [Ede87], encontramos dos rectas l_1 y l_2 que se cortan, tal que en cada uno de los cuadrante Q_i , $1 \leq i \leq 4$, determinado por l_1 y l_2 , hay exactamente $n/4$ puntos. La raíz almacena las ecuaciones de l_1 y l_2 y el valor de n . Para cada cuadrante construimos recursivamente un *árbol de partición* para $S \cap Q_i$ y se coloca como el i -ésimo subárbol hijo de la raíz. En cada cuadrante, repetimos el procedimiento hasta agotar la nube de puntos.

Una consulta de recuento puede responderse como sigue: Sea H un semiplano de consulta. Recorremos el árbol T , comenzando desde la raíz y manteniendo la cuenta global. En cada nodo v se almacena n_v , que es la cantidad de puntos en su subárbol. El algoritmo detecta si la recta h , que determina el semiplano H , interseca el cuadrante Q_v asociado con v ; entonces recursivamente visitamos cada hijo de v . Si $Q_i \cap H = \emptyset$, no hacemos nada. En otro caso agregamos n_v a la cuenta global. Los cuadrantes están determinados por dos rectas, como mencionamos antes, así que h interseca a lo sumo tres, quedando siempre uno excluido totalmente, por lo que el procedimiento de búsqueda trata a lo sumo tres subárboles. El tiempo de consulta de este procedimiento es $O(n^{\log_4 3}) = O(n^{.792})$. Una consulta de reporte agrega un poco más por el hecho de informar los k elementos que satisficieron la consulta: $O(n^{.792} + k)$. Esta estructura ocupa $O(n)$ espacio.

Mejores construcciones se han hallado, incluso para dimensiones superiores. Todos los algoritmos de la literatura se basan en el teorema del Ham-Sandwich, consecuencia del teorema de Borsuk-Ulam, quienes utilizan siempre algún esquema de partición para lograr particionar el espacio respecto de un conjunto de puntos en varias regiones acotadas por hiperplanos. Una propiedad que se destaca es que para cualquier hiperplano de consulta, las regiones intersecadas contienen una cantidad reducida de los puntos del conjunto total. Teniendo un esquema de partición, es posible construir un árbol de partición en forma estándar [EW86], [Wel88], [DE84], [Mat94], [Mat90a], [YDEP89].

Búsquedas por rangos Simpliciales

Las consultas en este contexto son consultas en polígonos; es decir, los rangos son polígonos. Todo polígono puede descomponerse en piezas más pequeñas, realizando una triangulación del mismo. De esta forma, la consulta en el polígono se transforma en consultas sobre triángulos de la triangulación. El resultado de la consulta es la unión de las respuestas de cada triángulo. Este concepto puede extenderse al espacio d -dimensional, donde tenemos símlices. Un símlice es el cierre convexo de $d+1$ puntos independientes [HW87], [Mat90a], [Mat91], [Mat92a], [Mat92c], [Sar98], [Wel88], [Wil82].

Dado que el tamaño y el tiempo de consulta de cualquier estructura de datos tiene al menos complejidad lineal y logarítmica respectivamente, consideremos estos extremos: i) ¿Cuán rápido puede ser contestada una consulta de rango simplicial usando estructura de datos de tamaño lineal? ii) ¿Qué espacio se requiere para una estructura de datos que conteste una consulta en tiempo logarítmico?. La combinación de estos dos extremos nos lleva a una competencia espacio-tiempo. Nosotros dividiremos nuestra exposición en dos partes. Primero consideraremos algoritmos que utilizan espacio lineal (o casi lineal); y luego, los algoritmos cuyo tiempo de respuesta a las consultas es logarítmico.

Algoritmos que utilizan espacio lineal (o casi lineal):

La mayoría de las estructuras de datos de tamaño orden lineal para BRSx, se basan en los *Árboles de Partición* (*Partition Trees*). Dados n puntos en el plano, dividimos el plano en varias regiones, de modo que todas las regiones contengan una cantidad semejante de puntos y que cualquier recta dada corte una o varias de las regiones. La eficiencia del árbol de partición se determina por el esquema de partición utilizado [Wil82].

Supongamos que dado S , un conjunto de puntos en el plano, queremos informar cuáles son los puntos de S que están contenidos en un cierto triángulo, el cual representa un rango de consulta. Una búsqueda en triángulos se reduce a tres búsquedas en semiplanos; por lo tanto, el problema se reduce a tratar consultas por semiplanos.

En principio, generamos una *partición simplicial* de S ; esto es un conjunto $\Pi = \{(S_1, \Delta_1), \dots, (S_r, \Delta_r)\}$. Las siguientes propiedades se mantienen: S_i es una *clase* y Δ_i es un *símplice*, $S_i \subseteq S$, $S_i \subseteq \Delta_i$, $S_i \cap S_j = \emptyset$ para $i \neq j$. El *número de corte* (*crossing number*) de una recta l con respecto a Π es el número de triángulos de Π cortados por l . El *número de corte* de Π es el máximo número de cortes sobre todas las posibles rectas l . La partición simplicial es *fin* si $|S_i| \leq 2n/r$, para cada $1 \leq i \leq r$.

La estructura de datos que usamos es un árbol de partición. La raíz tiene r hijos, siendo cada uno la raíz de un árbol de partición definido recursivamente para cada una de las clases de la partición simplicial fina. La estructura básica es la siguiente: Si S contiene un único punto p , el árbol de partición consiste de una hoja donde p es almacenado explícitamente. En otro caso, la estructura es un árbol T con r hijos. Cada hijo de la raíz corresponde a uno de los triángulos de la partición de S . El triángulo de la partición correspondiente a un hijo v es denotado $t(v)$ o Δ_v , eventualmente. La correspondiente clase S_v se llama *clase canónica* de v , denotada $S(v)$. El hijo v es la raíz de árbol de partición T_v definido recursivamente sobre el conjunto $S(v)$. Con cada hijo v , almacenamos el triángulo $t(v)$ (o Δ_v), con información acerca del subconjunto $S(v)$, tal como su cardinalidad u otro dato de interés.

Dado S , un conjunto de puntos en el plano, $d=2$, para cualquier $\epsilon > 0$, existe una estructura de datos para S , llamada árbol de partición que utiliza $O(n)$ espacio de almacenamiento, tal que los puntos de S contenidos en un triángulo de consulta pueden ser contados en $O(n^{1/2+\epsilon})$ tiempo. Los puntos pueden ser reportados en $O(k)$ tiempo adicional, donde k es el número de puntos informados. La estructura puede ser construida en $O(n^{1+\epsilon})$.

Una consulta por un semiplano en un árbol de partición de una partición simplicial se resuelve en la forma corriente. Para resolver consultas por rango triangular, tenemos que un triángulo de la partición es cortado por la recta borde del rango de consulta. Solamente hay tres rectas, por lo que cada una de estas rectas corta a lo sumo $c\sqrt{r}$ triángulos. Entonces, el rango de consultas cruza a lo sumo $3c\sqrt{r}$ triángulos. Por lo tanto, sólo hemos cambiado de c a $3c$, permaneciendo el tiempo de consulta cercano al visto, aunque que asintóticamente permanece en el mismo orden.

Algoritmos con tiempo de respuesta logarítmico

Podemos mejorar los tiempos de consulta cambiando de $O(\sqrt{n})$ a $O(\log n)$. La idea subyacente es la misma que en los árboles de partición con particiones simpliciales disjuntas, sólo que nos trasladamos al espacio dual. Comencemos con algunas consideraciones. Sea $L = \{l_1, l_2, \dots, l_n\}$ un conjunto de rectas obtenidos después de dualizar una nube de puntos, las cuales serán preprocesadas para consultas por rango triangular.

Definimos un *cutting* como un conjunto de símplices, posiblemente ilimitados, con interiores disjuntos, que cubren el R^d . Sea H un conjunto de n hiperplanos de R^d y r un parámetro, $1 \leq r \leq n$, un $(1/r)$ -*cutting* es para H es un cutting con la propiedad de que ningún símplice de la partición es cortado por más de n/r hiperplanos de H . El *tamaño* del cutting $C(L)$ está definido por la cantidad de símplices que posee.

La noción de cutting es básica para muchos algoritmos geométricos, puesto que un problema relacionado a un conjunto H de hiperplanos, se puede partir en subproblemas definidos por los símplices de un $(1/r)$ -cutting, donde cada símplice trata alrededor de n/r hiperplanos [Cla87], [CSW90], [CSW92], [CW89], [HW87], [BF79] [Mat90a], [Mat91], [Mat92c], [Sar98], [Wel88], [Wil82]. Es importante destacar que se busca, por lo general, cuttings cuyo número de símplices sea el menor posible [AHWW87].

Nuestra estructura de datos almacenará cada triángulo de la partición junto con la información acerca de cuántas rectas pasan por encima (por debajo) de él. Para cada triángulo definimos recursivamente la misma

estructura sobre el conjunto de rectas que lo intersecan. Para ello, oportunamente consideraremos los *niveles de un arreglo*.

El enunciado dual de la búsqueda por rango triangular se plantea del siguiente modo: dado un conjunto L de rectas en el plano y tres puntos q_1 , q_2 y q_3 de consulta rotulados como “superior” o “inferior”, se requiere contar el número de rectas de L contenidas en los lados especificados de los tres puntos de consulta (en el plano dual). Este problema puede ser resuelto con un árbol de cortes de tres niveles

Estos últimos tres tipos de búsqueda son los que han recibido más atención recientemente, puesto que además de sus aplicaciones directas, han brindado soluciones a problemas subyacentes en problemas geométricos de mayor jerarquía. Aunque las complejidades son mayores que las de *BRO* ocurre que ellas han brindado soluciones a problemas, no resueltos por las *BRO*.

3. Separabilidad geométrica

Los trabajos de separabilidad están orientados a dos o más conjuntos disjuntos de objetos geométricos, básicamente para puntos, bajo diversos criterios de separabilidad (bandas, cuñas, sectores, etc.). Los principales resultados se deben a Seara [Sea02], y los resultados de su tesis doctoral fueron publicados previamente [AHMS01] [DHMS01] [HNRS01] [AHMS00] [HMRS99] [HNRS98], por mencionar sólo algunas referencias. Estos criterios de separabilidad tienen aplicaciones interesantes, como por ejemplo el análisis de imágenes, clasificación de datos, etc. Siempre que sea necesario discriminar objetos en un espacio de trabajo, por algún atributo del mismo, los criterios de separabilidad juegan un papel importante.

Se dan dos conjuntos disjuntos P y Q de objetos en el plano, clasificados como dos clases de objetos: *rojos* y *azules*, respectivamente. Eventualmente, los objetos pueden ser: puntos, segmentos, polígonos o círculos. En el caso de los polígonos, n y m representan el número total de segmentos de los polígonos de P y de Q . En otro caso, n y m representan la cardinalidad de P y Q , respectivamente. En cualquiera de los casos, N es el máximo de n y m .

Dada una familia C de curvas en el plano, decimos que los conjuntos P y Q son *C-separables* si existe una curva $C_i \in C$, llamada *separador*, tal que cada componente conectada de $R^2 - C_i$ contiene solamente objetos, o bien de P , o bien de Q .

En este contexto, donde hay dos clases de objetos, los rojos y los azules, se plantean dos tipos de problemas:

- i) *Problemas de Decisión o Existencia*: Dada la nube de puntos $P \cup Q$, de puntos rojos y azules en el plano, se cuestiona la existencia de un separador de los puntos rojos de los azules. Por ejemplo *¿existe una cuña que separe los puntos rojos de los azules?* En caso afirmativo, hallar los vértices de todas las cuñas separadoras.
- ii) *Problemas de Optimización*: Asociado a la separabilidad por cuñas y bandas consisten en hallar la cuña separadora de mínimo (máximo) ángulo y la banda de mínimo (máximo) ancho que separen los puntos rojos de los azules.

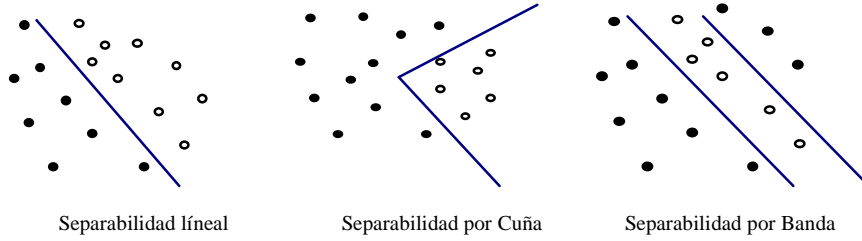
Describimos brevemente a continuación diferentes clases de separabilidad, dando las cotas de complejidad de los algoritmos correspondientes. No presentamos ninguna descripción de su funcionamiento, dado que no es el punto central, sólo nos interesa saber que existen. Para estudiar el tema en detalle, ver [Sea02].

Atendiendo a la familia C , se tienen diferentes tipos de separabilidad:

Si C es la familia de rectas en el plano, tenemos la *separabilidad lineal*. Dos conjuntos P y Q son *linealmente separables* si y sólo si sus cierres convexos no se intersecan.

Dos conjuntos disjuntos P y Q de objetos en el plano son *separables por una cuña* si existe una cuña que contiene solamente todos los objetos de uno de los conjuntos. Aquí se estudia el problema de decidir la separabilidad por cuñas calculando la ubicación de los posibles ápices (vértices) de cuñas; como una extensión al problema, se estudia hallar la cuña de ángulo mínimo (máximo).

Dos conjuntos disjuntos P y Q de objetos en el plano son *separables por banda* si existe una banda, determinada por dos rectas paralelas, que contiene solamente todos los objetos de uno de los conjuntos. Aquí se estudia el problema de decidir la separabilidad por bandas calculando el conjunto de intervalos de pendientes de bandas separadoras; como una extensión al problema, se estudia hallar la banda mínima (máxima).



La separabilidad de conjuntos de objetos geométricos está demostrada para separar dos conjuntos de puntos, segmentos, polígonos o círculos en el plano con cuñas y bandas, pudiendo encontrarse cuñas de máximo (mínimo) ángulo y bandas de máximo (mínimo) ancho. Los algoritmos para resolver los problemas de decisión y optimización propuestos anteriormente se ejecutan en $O(N \log N)$ tiempo; salvo en el caso de la separación lineal, que toma $O(N)$ tiempo.

Los estudios sobre separabilidad en el plano han avanzado sobre separaciones con varias bandas, cuñas y sectores. Así tenemos *separabilidad por dos bandas*, que en caso de existir el par de bandas, se puede estudiar el problema de optimización de hallar las dos bandas de ancho mínimo (máximo). Este problema puede ser resuelto en $O(n^3 \log n)$ tiempo. En el mismo tiempo se puede decidir si las bandas son de mínima (máxima) anchura. En cambio, si se proporciona la pendiente de una de las bandas, el tiempo se reduce a $O(n^2 \log n)$, pudiendo hallarse las dos bandas de mínima (máxima) anchura en el mismo tiempo. Aun mejor si las dos pendientes son conocidas, puesto que se reduce a $O(n)$ tiempo. Como resultado, tenemos que dados dos conjuntos disjuntos P y Q de puntos en el plano, el mínimo número de bandas paralelas (rectas) que los separan y los correspondientes intervalos de pendientes, puede ser determinado en $O(n^2 \log n)$ tiempo.

También, tenemos *separabilidad por dos cuñas*, que en caso de existir el par de cuñas, podemos tener cuñas disjuntas o no; podemos determinarla en $O(n^5 \log n)$ tiempo; pero, en caso de que las cuñas sean disjuntas, podemos decidir la separabilidad en $O(n^3 \log n)$ tiempo.

Hay más variantes, como *separabilidad por sectores*, si todas las cuñas separadoras tienen el mismo ápice. Esta clase de separación es siempre posible. Si damos un punto p , entonces en $O(n \log n)$ tiempo podemos determinar la cantidad de rayos que emanan desde p , separando los conjuntos en sectores monocromáticos. En este caso es interesante minimizar la cantidad de rayos que emanan de un ápice; es decir, es interesante encontrar la región de los centros de estrellas de rayos, tal que se obtenga la cantidad mínima de rayos necesarios para separar los puntos rojos y azules. Como resultado, dados dos conjuntos disjuntos P y Q de puntos en el plano, el conjunto de mínima cardinalidad de rayos que tienen el mismo ápice, separadores de P y Q , puede ser determinado en $O(n^4)$ tiempo. En el caso de que restringamos los ápices a una recta l , el conjunto de mínima cardinalidad de rayos que tienen el mismo ápice sobre l , separadores de P y Q , puede ser determinado en $O(n^2 \log n)$ tiempo. La región de posibles ápices sobre l se puede obtener en la misma cota de tiempo.

En lugar de tener una nube bicolor, podemos considerar diversos colores. Así, dados C_1, C_2, \dots, C_k conjuntos disjuntos en el plano, decimos que el conjunto C_i tiene el color c_i , y denotamos con n_i la cardinalidad de C_i y $n_1 + n_2 + \dots + n_k = n$. Entonces, un conjunto finito S de curvas en el plano es un separador para los conjuntos C_1, C_2, \dots, C_k si cada componente conexa en $R^2 - S$ contienen solamente objetos de C_i . También decimos que cada componente es *monocromática*. Obviamente, cuando $k=2$, son los casos estudiados previamente.

Los siguientes resultados incluyen algoritmos eficientes para hallar separadores de mínima cardinalidad por medio de rectas paralelas y por medio de rayos con un ápice común.

Dados los conjuntos disjuntos C_1, C_2, \dots, C_k coloreados, con n puntos en total, en el plano, la mínima cantidad de rectas paralelas separadoras de los puntos en bandas monocromáticas y los intervalos de pendientes de todas las posibles soluciones, pueden calcularse en $O(n^2 \log n)$ tiempo. En el caso de permitir $k-1$ rectas paralelas únicamente, conseguir los intervalos de pendientes de todas las posibles soluciones, pueden calcularse en $O(kn) + O(k^2 \log k)$ tiempo.

En la separabilidad por sectores, queremos que los k conjuntos coloreados disjuntos estén repartidos en las cuñas monocromáticas definidas por rayos que emanan desde un ápice común. Dados C_1, C_2, \dots, C_k conjuntos disjuntos coloreados, con n puntos en total, un conjunto de mínima cardinalidad de rayos que tienen ápice común, que separen los conjuntos se puede hallar en $O(n^4)$ tiempo. En el caso de tener una recta l , entonces el conjunto de mínima cardinalidad de rayos separadores de los conjuntos C_1, C_2, \dots, C_k con un ápice común sobre l se puede calcular en $O(n^2 \log n)$ tiempo. La región de posibles ápices sobre l se puede obtener con la misma cota de tiempo.

Los principales resultados sobre separabilidad con bandas y cuñas, con diferentes versiones sobre combinaciones y propiedades de ellas, son algoritmos eficientes para decidir la existencia de los separadores y calcular soluciones factibles. Desde la separabilidad lineal a las otras presentadas, hemos visto cómo la complejidad de los algoritmos aumenta por lo menos en un factor $\log n$, tanto para la decisión como el cálculo correspondiente, sin que se conozca aún una cota que los separe en diferentes categorías de complejidad. También es una incógnita la existencia de alguna categoría de problemas de separabilidad ubicada entre dos categorías. Algunos de estos resultados pueden ser extendidos en el plano al uso de otros objetos geométricos que no sean puntos (segmento, círculos, etc.), y al espacio tridimensional en donde sólo existen algunos algoritmos para resolver algunos de los problemas vistos. En este marco, también queda abierto el estudio sobre nuevos criterios de separabilidad; es decir, no sólo usar planos o semiplanos, sino otros objetos como pirámides dobles (bipirámides), cilindros, cónicas, cónicas dobles, etc.

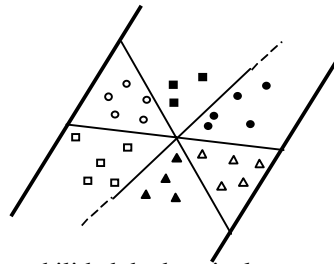
4. Separabilidad en dos niveles

En esta sección describimos ideas generales para dar una visión más amplia de posibles líneas de trabajo e investigación, como elementos de continuidad de este trabajo. Nuestra propuesta consiste en vincular las búsquedas por rangos con la separabilidad de objetos geométricos por medio de cuñas, bandas y variantes. En este sentido, las regiones se obtienen en base a atributos propios de los puntos y de su ubicación en el espacio. El beneficio básicamente está en generar subregiones del espacio para las cuales tiene sentido considerarlas porque se distinguen por las propias características de sus individuos. En cuanto a las consultas podemos pensar en *rangos cuñas* y *rangos bandas*, observando que se corresponden con tipos de consultas de rangos ya presentadas.

Presentamos las siguientes ideas en el plano, puesto que es más clara su comprensión. Pensemos, en principio, que los objetos geométricos a los que nos referimos son puntos. La nube puede tener puntos de diversas categorías; es decir, los puntos no están limitados a dos colores únicamente.

Queremos partir el plano en regiones, mediante algún criterio de separabilidad (rectas, bandas, cuñas, sectores, etc.). Con ello obtenemos un esquema de partición, en donde los objetos de cada clase están agrupados por alguna característica (atributo) propia. Con esto estamos en condiciones de comenzar a armar un árbol, donde el nodo raíz representa la nube de puntos completa, y cada hijo representa una de las regiones obtenidas por el criterio de separabilidad aplicado. La nube completa queda representada en la raíz del árbol de partición y cada banda en un hijo de la raíz, dando origen a un árbol de aridad k en el primer nivel. Luego, por cada hijo analizamos qué posibilidades de separación existen y en base a ello, volvemos a aplicar algunas de las variantes vistas previamente. Repitiendo este tipo de proceso con cada hijo, estamos en condiciones de conformar el segundo nivel del árbol.

Algunas regiones pueden quedar parcialmente acotadas, mientras que otras pueden quedar totalmente acotadas por las rectas que definen un separador de primer nivel y un separador de segundo nivel.



Separabilidad de dos niveles

Si el ejemplo de la figura anterior correspondiese a la banda i , entonces del i -ésimo hijo del nodo raíz, se desprenderían seis hijos uno para cada cual de los sectores obtenidos por la segunda separación.

Estos son casos de separabilidad que podemos utilizar en primera instancia para partir el plano en regiones acotadas o semiacotadas, a fin de obtener una primera descomposición del mismo. Luego, podemos pensar en descomponer las regiones obtenidas. En todos los casos, al partir una región, obtenemos subregiones que conformarán los hijos de cada uno de los nodos del primer nivel del árbol. Éste es un caso de dos niveles de separabilidad, para k conjuntos disjuntos, cada uno con k' conjuntos disjuntos interiores. En cuanto a las consultas, las mismas pueden ser consideradas como los casos vistos. Siempre, habrá que analizar los algoritmos correspondientes y buscar técnicas que permitan resoluciones eficientes.

La idea principal es que si se conoce previamente una clasificación de los puntos, como por ejemplo qué colores hay o qué formas o qué tamaños, entonces nuevamente podemos resolver la descomposición como un problema de separabilidad. Algunos casos serán sencillos de analizar y crear las particiones, estructuras y algoritmos correspondientes. Pero, seguramente, una gran cantidad requerirán de mucho análisis y trabajo futuro. Dadas las cotas de complejidad presentadas en la sección anterior, el lector puede considerar poco óptimo trabajar con separabilidad, pero para nosotros el punto motivo de este ítem es vincular dos temáticas utilizando resultados conocidos en ambas y buscando su aplicación en donde prime la búsqueda por clases y se requiera tener una preclasificación hecha de los datos.

Las ideas presentadas, pretenden llegar a árboles decisión de k niveles. Aunque aquí la problemática esté clasificada como problemas NP-hard.

5. Conclusiones y visión de futuro

El grupo de trabajo en la UNSL, conjuntamente con docentes de la UPM, han dado inicio en el presente año a un proyecto de investigación conjunto, *Proyecto de la UPM de AL2002-1010-2.43 Geometría Computacional*, el que tiene una duración prevista de tres años, y el objetivo principal es la consolidación de la línea de trabajo en Geometría Computacional en la UNSL, aportando nuevos enfoques y técnicas algorítmicas a las líneas de investigación ya establecidas en su Departamento de Informática dentro de LIDIC (Laboratorio de Investigación y Desarrollo en Inteligencia Computacional). Esto constituye una ocasión inmejorable para impulsar investigadores en período de formación. También se abren nuevas perspectivas a quienes ya tienen experiencia y desean reorientar o complementar sus investigaciones. Así el segundo objetivo es la realización de tesis doctorales y de maestría en el campo de estudio. Como actividades relacionadas, se vienen realizando, anualmente, pasantías de docentes, dictado de cursos de postgrado, seminarios, tesis de maestría, trabajos de fin de carrera, etc., con el fin de afianzar y fomentar la disciplina. Como resultados, se espera la obtención de resultados científicos en el campo de las aplicaciones de Geometría Computacional. En particular, nosotros hemos presentado el problema de Búsqueda en rangos dentro de Bases de datos. Este tópico corresponde a uno de los que actualmente se estudian, ya que, paralelamente, están *Sumas de Minkowski* y *Compass Routing*.

En lo referido a *Búsquedas por rangos* y *Separabilidad geométrica*, hemos visto las cotas de complejidad, por lo que el lector puede considerar poco óptimo trabajar con separabilidad, pero para nosotros el punto motivo de esta vinculación es utilizar resultados conocidos en ambas y buscar su aplicación en donde prime la búsqueda por clases, con clasificaciones previas de los datos.

Finalmente, algunos de los resultados expuestos pueden ser extendidos en el plano al uso de otros objetos geométricos que no sean puntos (segmento, círculos, etc.), y al espacio tridimensional en donde sólo existen algunos algoritmos para resolver algunos de los problemas vistos. En este marco, también queda abierto el estudio sobre nuevos criterios de separabilidad; es decir, no sólo usar planos o semiplanos, sino otros objetos como pirámides dobles (bipirámides), cilindros, cónicas, cónicas dobles, etc.

Agradecimientos: Queremos agradecer a los árbitros anónimos que con sus comentarios ayudaron a mejorar esta presentación.

6. Referencias bibliográficas

- [Abe00] Abellanas Oar, M. *Descubriendo la Geometría Algorítmica*, 2000.
<http://www.dma.fi.upm.es/mabellanas/divulgación/GeometriaAlgoritmica.html>
- [AE98] Agarwal, P.; Erickson, J. *Geometric range searching and its relatives*; Advances in Discrete and Comput. Geom. (B. Chazelle, J. Goodman, and R. Pollack, eds.), American Mathematical Society, 1998.

- Society, Providence, 1998.
- [Aga97] Agarwal, P.; *Range searching* CRC Handbook of Computational Geometry (J. Goodman and J. O'Rourke, eds.).
- [AHMS00] Arkin, E.M.; Hurtado, F.; Mitchell, S.B.; Seara, C.; Skiena, S.S.; *Some Separability Problems in the Plane*; 16th European Workshop on Computational Geometry, Eilat, Israel, 2000.
- [AHMS01] Arkin, E.M.; Hurtado, F.; Mitchell, S.B.; Seara, C.; *Some Lower Bounds on Geometric Separability Problems*; 11th Fall Workshop on Computational Geometry, Polytechnic University, Brooklyn, NY, 2001.
- [AHU74] Aho, A.V.; Hopcroft, J. E.; Ullman, J. *The design and analysis of computer algorithms*, Addison-Wesley Series in Computer Science and Information Processing, 1974.
- [AHWW87] Alon, N.; Haussler, D.; Welzl, E.; Wöginger, G.; *partitioning and geometric embedding of ranges spaces of finite Vapnik-Chervonenkis dimension*; Proc. 3 ACM Symposium on Computational Geometry, 331-340, 1987.
- [AM94] Agarwal, P.; Matousek, J.; *On range searching with semialgebraic sets*; Discrete Comput. Geom., 11: 393-418, 1994.
- [BF79] Bentley, J.L.; Friedman, J.H.; *Data Structures for range searching*; ACM Comput. Surv. 11:397-409; 1979.
- [BKOS97] de Berg, M.; Kreveld, Overmars, M.; Schwarzkopf. *Computational Geometry: algorithms and applications*, Springer Verlag, BH 1997
- [BY98] Boissonnat, J.D.; Yvinec, M. *Algorithmic Geometry*, Cambridge University Press, 1998.
- [Cla87] Clarkson, K.L.; New applications of random sampling in computational geometry; Discrete Comput. Geom., 2:195-222, 1987.
- [CSW90] Chazelle, B.; Sahrir, M.; Welzl, E.; *Quasi-optimal upper bounds for simplex searching and new zone theorems*; Proc. 6 ACM Symposium on Computational Geometry, 23-33, 1990.
- [CSW92] Chazelle, B.; Sahrir, M.; Welzl, E.; *Quasi-optimal upper bounds for simplex searching and new zone theorems*; Algorithmica. 8:40-429, 1992.
- [CW89] Chazelle, B.; Welzl, E.; *Quasi-optimal range searching in spaces of finite VC-dimension*. Discrete -Computational Geometry, 4:387-490, 1989.
- [Cha90a] Chazelle, B.; *Lower bounds for orthogonal range searching I*. J.ACM 37:200-212, 1990.
- [Cha90b] Chazelle, B.; *Lower bounds for orthogonal range searching II*. J.ACM 37:439-463, 1990.
- [DE84] Dobkin, D.P.; Edelsbrunner, H.; *Organizing point sets in two and three dimensions*. Tech. Report F130, Technische Universita Graz; 1984.
- [DHMS01] Devillers, O.; Hurtado, F.; Mitchell, S.B.; Seara, C.; *Separating Several Point Sets in the Plane*; 13th Canadian Conference on Computational Geometry, 2001.
- [EW86] Edelsbrunner, H.; Welzl, E. *Halfplanar range searching in linear space and $O(n^{695})$ query time*; Inform. Process. Lett., 23:289-293, 1986.
- [GO97] Goodman, J.; O'Rourke, J. *Handbook of Discrete and Computational Geometry*. CRC Press 1997.
- [Gün88] Günther, O. *Efficient Structures For Geometric Data Management*. Springer Verlag 1988.
- [HMRS99] Hurtado, F.; Mora, M.; Ramos, P.A.; Seara, C.; *Separación de Objetos en el Plano por Doble Cuña y por Θ -Poligonal*; VIII Encuentros de Geometría Computacional, Castelló, España, 1999.
- [HNRS01] Hurtado, F.; Noy, M.; Ramos, P.A.; Seara, C.; *Separating Objects in the Plane with Wedges and Strips*; Discrete Applied mathematics, Vol 109, pp 109-138, 2001.
- [HNRS98] Hurtado, F.; Noy, M.; Ramos, P.A.; Seara, C.; *Separating Objects in the Plane with Wedges and Strips*; 110th Canadian Conference on computational Geometry, Montreal, Canadá, 1998.
- [HW87] Haussler, D.; Welzl, E.; *Epsilon nets and simplex range queries*; Discrete Comput. Geom., 2:127-151; 1987.
- [Mat90a] Matousek, J.; *Construction of E-nets*; Discrete Comput. Geom., 5:427-448, 1990.

- [Mat91] Matousek, J. *Cutting hyperplane arrangements*; Discrete Comput. Geom., 6(5):385-406, 1991.
- [Mat92a] Matousek, J. *Efficient partition tree*; Discrete Comput. Geom., 8:315-334, 1992.
- [Mat92c] Matousek, J. *Range searching with efficient hierarchical cuttings*; Proc. 8 ACM Symposium on Computational Geometry, 276-285, 1992.
- [Mat94] Matousek, J.; *Geometric Range searching*; ACM Comput. Survey, 26:421-461;1994.
- [Meg79] Meggido, N.; *Combinatorial optimization with rational objective functions*. Math. Oper. Res. 4:414-424;1979.
- [Mel84] Melhorn, K. *Multidimensional Searching and Computational Geometry* Springer Verlag 1984.
- [Sam90a] Samet, H. *Applications of Spatial Data Structures* Addison-Wesley, Reading, MA, 1990 .
- [Sam90b] Samet, H. *The Design and Analysis of Spatial Data Structures* Addison-Wesley, Reading, MA, 1990
- [Sam90c] Samet, H.; *Applications of spatial data structures: Computer graphics, Image processing and GIS*; Addison-Wesley, 1990.
- [Sar98] Sarel, Har-Peled; *Constructing planar cuttings in theory and practice*;
- [Sea02] Seara, C; *On geometric separability*; Tesis Doctoral 2002, Univ. Polit cnica de Catalu a.
- [SU00] Sack, J.R.; Urrutia, J.. *Handbook of Computational Geometry*. Elsevier Science B.V. 2000.
- [Tou85] Toussaint, G.T. *Computational Geometry* Edited by North-Holland, Amsterdam, 1985 .
- [Tou92] Toussaint, G.T. *What is computational geometry?* Proc. IEEE, vol. 80, No. 9, pp. September,1992,1347-1363.
- [Wel88] Welzl, E.; *Partition Tree for triangle counting and other range searching problems*; In Proc. 4 ACM Symposium on Computational Geometry, 23-33,1988.
- [Wil82] Willard, D.E.; *Polygon retrieval*; SIAM J. Comput., 11;149-165;1982.
- [WL85] Willard, D.E.; Lueker, G.S: *kadding range restriction capability to dinamic data structures*. J.ACM, 32:597-617;1985.
- [Y83] Yao, F.F.; *A 3-space partition and its applications*. Proc. 15th Annu. ACM Sympos. Theory Comput. 258-263, 1983.
- [YDEP89] Yao, F.F.; Dobkin, D.P.; Edelbrusnner, H; Paterson, M.S.; *Partitioning space for range queries*; SIAM L Compuy. 18:371-384;1989.
- [YY85] Yao, A.C.; Yao, F.F.; *A general approach to D-dimensional geometric queries*; Proc. 17th Annu. ACM Sympos. Theory Comput. 163-168, 1985.