

# Geometría Computacional

## Desarrollo Práctica Análisis numérico

Sebastián Arcila Valenzuela (*sarcilav@eafit.edu.co*),  
Sergio Botero Uribe (*sbotero2@eafit.edu.co*),  
Cristian Camilo Isaza Arias (*cisazaar@eafit.edu.co*),  
Eliel David Lobo Vesga (*eloboves@eafit.edu.co*),  
Hernán Darío Metaute Sarmiento (*hmetaute@eafit.edu.co*).

31 de agosto de 2009

### 1. Introducción

Desde hace ya algún tiempo, se ha notado que la geometría computacional cobra importancia en varias áreas, como la computación gráfica, la robótica y el diseño asistido por computador (CAD). Esto se debe a que la geometría en general, trabaja sobre una de las cualidades más importantes de la materia: su forma. Cuando queremos modelar objetos pertenecientes a la realidad, debemos dar cuenta de la forma que tienen y de ciertas propiedades que delimitan forma interactúa con el entorno. Desde aquí que sea esta materia la que nos permita trabajar propiedades tan importantes como el área en 2 dimensiones, el volumen en 3 y las relaciones de los objetos en el espacio (intersecciones, distancias, etc.). Sabido esto, es de extrañarse que en muy pocas universidades se imparta un curso que aborde los métodos computacionales que se pueden aplicar cuando se trata de resolver problemas relacionados con la geometría computacional. Es por esto que en el presente proyecto se hace un esfuerzo por acomodar de alguna manera los métodos numéricos computacionales al área de nuestro interés para así construir un conocimiento aplicado por medio de la evaluación de estos métodos, la aplicación al problema planteado y las comparaciones en rendimiento algorítmico de las soluciones propuestas al problema planteado.

### 2. Descripción del problema

Cuando se trabaja con representaciones de la realidad en el computador, a menudo se tiene que hacer una simplificación de esta; la modelación permite obtener información relevante del objeto que se estudia, dejando de lado las cualidades secundarias que no atañen a cierto problema que se desea atacar y obteniendo tanta información como sea posible de las características que se

usan en el modelo como delimitantes. En los diferentes campos de acción que tienen las ciencias relacionadas con (o que se ayudan de) la computación, a menudo se hace necesario representar objetos de la realidad para permitir que el computador agilice cálculos que, de otra forma, sería tedioso trabajar. Una primera aproximación a cualquier forma geométrica en 2D puede ser el contorno o silueta de la figura modelada. Desde esta silueta se puede comenzar un proceso que encaminado a hacer más compleja la figura para que dé cuenta de más propiedades, y que estas, a la vez, sean más fieles al comportamiento de las características que tienen los objetos en el mundo real. El problema viene entonces cuando se trata de pasar de la simplificación que se hace de una figura a la forma que originalmente tenía. En este traspaso se trabaja con unas unidades mínimas (puntos) que representan, en algunos casos, la medición de un aparato externo al computador, y que es necesario *traducir* de nuevo a una imagen única y completa para reconstruir el objeto (en este caso su contorno) y poder trabajar con él. En general, la primera técnica algorítmica que se aplica cuando se trabaja con nubes de puntos en 2D, se conoce como Convex Hull. El convex hull es a la geometría computacional, lo que el ordenamiento es para muchos problemas algorítmicos; esto es, nos permite hacer un trabajo sobre una entrada de puntos sin ninguna estructura para poder hacer luego operaciones más interesantes sobre los datos. El convex hull de un conjunto de puntos, se define como el polígono convexo (ángulos interiores menores de 180°) mínimo que contiene todos los puntos de dicho conjunto. De aquí que, al tener esta primera aproximación se pueda trabajar con una delimitación del área de trabajo que tiene propiedades importantes para el cálculo de áreas, por ejemplo. Sin embargo, el convex hull termina siendo sólo una primera aproximación a los datos, puesto que, por la definición misma del método, deja ciertos puntos sin incluir y nos da un *contorno brusco* de los puntos que se trabajan. Por otro lado, hay casi tantas técnicas para el convex hull como de algoritmos de ordenamiento; todos ellos basándose en algún criterio para clasificar los puntos. Lamentablemente, ninguno de ellos utiliza métodos numéricos. Para lograr superficies suaves y curvas más complejas, son necesarias técnicas más complejas que esta primera aproximación. Cuando necesitamos llegar a un contorno suave, estamos en el dominio de las curvas conocidas como splines, las que tienen la propiedad de poder representar tales curvas con cierta facilidad para el programador, lo que hace que varios paquetes gráficos traigan ya implementadas una forma particular de estos splines, llamados Curvas de Bézier. Nuestro problema se resume, entonces, en la reconstrucción de contornos a partir de una nube de puntos con base en métodos numéricos y splines que pasen por los puntos que lo representan. Esto desde un contorno *brusco* obtenido a partir de un convex hull.

### 3. Definición de la practica

El resultado del proyecto será tangible, se tratará de un proyecto de software libre, desarrollado inicialmente bajo la motivación de la materia de análisis

numérico, que buscará abarcar inicialmente ciertos algoritmos particulares que involucren geometría computacional y métodos numéricos para ver luego hasta dónde se puede llevar esta fusión de técnicas. Así se desarrollarán en paralelo dos algoritmos para atacar este problema, a partir de la información de trabajos ya realizados en el área y en temas afines; se analizará su desempeño y las diferencias con otros algoritmos ideados para solucionar este tipo de problemas. Se presentarán informes de lo que se conoce como *perfil* de los algoritmos desarrollados; esto es, su desempeño bajo condiciones de prueba.

## 4. Alcance inicial del proyecto

La metodología escogida para el trabajo es la división del grupo en dos equipos de trabajo. Así se trabajará en paralelo para llegar a dos soluciones (algoritmos y su implementación en programas) para este problema de contornos en 2 dimensiones. Se realizarán los informes planteados en el cronograma de trabajo en las fechas destinadas para ello. Hay que hacer claridad en que nuestra materia de estudio NO comprende cómo las nubes de puntos llegan al computador a partir de una modelación del mundo real. Teniendo esto entendido, se trabajará sobre datos que representan la nube ingresados al computador por entrada estándar, un archivo u otro método que facilite la entrada de estos datos *en bruto* al computador. El formato de estos puntos puede ser una lista de las coordenadas x,y de dichos puntos para que los algoritmos desarrollados trabajen sobre ellos en la reconstrucción de contornos.

## 5. Equipos de trabajo:

**Equipo 1** Eliel David Lobo, Hernan Dario Metaute, Sebastián Arcila (Líder)  
**Equipo 2** Cristian Camilo Isaza Sergio Botero Sebastián Arcila (Líder)  
**Asesor** Francisco Correa Zabala

## 6. Entregables al final del proyecto

- Seudocódigo de los algoritmos.
- Implementaciones en C++ de los algoritmos.
- Paper planteado por nuestro asesor.

## 7. Extras planteados

Estos son los posibles aditamentos que pueden abordarse como una adición a la práctica. Se trata de trabajo posterior que no está contemplado en el alcance

- Un analizador gráfico en OpenGL que muestre como se comporta el algoritmo respecto a la nube de puntos a medida que itera.

- Una aproximación a la idea algorítmica para hallar el contorno en nubes de puntos en 3D. Desde aquí se planea plantear.
- El pseudocódigo de esta aproximación a 3D
- Implementación en C++

Esperamos orientación de nuestro asesor.

## 8. Métodos asociados

[2]

## 9. Eficiencia de los métodos planteados

### Referencias

- [1] Skiena, Steven S y Revilla, Miguel A. *Programming Challenges : The Programming Contest Training Manual* : Geometry, Computational Geometry (págs 291-337). 2003. New York : Springer 2003. 359p. TEXTS IN COMPUTER SCIENCE. ISBN 0387001638.
- [2] Burden, Richard L y Faires, J. Douglas. *Análisis numérico* 7 ed. Mexico: Thomson Learning, 2002. 839p. ISBN 9706861343.
- [3] Cormen, Thomas H. *Introduction to algorithms*. 2 ed. MIT Press y McGraw-Hill. ISBN 0-262-53196-8.
- [4] M de Berg, M. van Kreveld, M. Overmars, y O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 2 ed, 2000.
- [5] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, New York, 2 ed, 2000.
- [6] Preparata, Franco y Shamos, Ian. *Computational Geometry: An introduction*. 1985, New York : Springer-Verlag. TEXTS AND MONOGRAPHS IN COMPUTER SCIENCE. ISBN 0-387-96131-3.
- [7] Edelsbrunner, Herbert. *Algorithms in combinatorial Geometry*. 1987. Berlin: Springer-Verlag .MONOGRAPHS ON THEORETICAL COMPUTER SCIENCE. ISBN 3-540-19772.
- [8] Melhorn, Kurt. *Data structures and algorithms 3: multi-dimensional searching and computational geometry*. 1984.
- [9] De Berg Mark, Otfried Cheong, Van Krevld Marc, y Overmars Mark. *Computational Geometry* 3ra edición revisada. 2008 Springer-Verlag. ISBN 3-540-77973-6, 1st edition (1987): ISBN 3-540-61270-X

- [10] Convex Hull Algorithms. Recurso en línea. <http://www.cse.unsw.edu.au/~lambert/java/3d/hull.html>. Consultado el 16 de agosto de 2009.
- [11] 3D Convex Hull. Recurso en línea. <http://student.ulb.ac.be/~claugero/3dch/index.html>. Consultado el 18 de agosto de 2009.
- [12] Selim G. Akl y Lyons Kelly A. *Parallel Computational Geometry*. 1993 Prentice-Hall. ISBN 0-13-652017-0.
- [13] Computational Geometry Community Bibliography O'rourke. Recurso en línea <ftp://ftp.cs.usask.ca/pub/geometry>. Consultado el 25 de agosto de 2009.
- [14] Computational geometry: theory and applications. Elsevier. (Journal) Recurso en línea. [http://www.elsevier.com/wps/find/journaldescription.cws\\_home/505629/description#description](http://www.elsevier.com/wps/find/journaldescription.cws_home/505629/description#description). Consultado el 25 de Agosto de 2009.
- [15] Computational geometry and applications. World Scientific. (Journal) Recurso en línea. <http://www.worldscinet.com/ijcga/>. Consultado el 25 de agosto de 2009.