

Validación de la triangulación Delaunay empleando álgebra geométrica conforme

Netz Romero, Ricardo Barrón-Fernández

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

jnromero@correo.uam.mx, rbarron@cic.ipn.mx

Resumen. Cuando la triangulación Delaunay se realiza en forma incremental, la etapa más importante, es la reconstrucción de los triángulos cuando se inserta aleatoriamente un nuevo punto en la red. Para ello existen diferentes técnicas, de la cual utilizaremos la validación del “círculo vacío” descrita por Boris Deloné, nuestro objetivo es utilizar el Álgebra Geométrica Conforme (AGC) para realizar dicha validación. Cambiaremos de ambiente matemático para demostrar las ventajas de las entidades geométricas que nos propone el AGC y emplearlas en un módulo que valide dicha triangulación.

Palabras clave. Álgebra geométrica conforme, círculo vacío, triangulación Delaunay.

Delaunay Triangulation Validation Using Conformal Geometric Algebra

Abstract. When Delaunay triangulation is performed in an incremental fashion, different steps are involved in the process. Within those steps “reconstruction” is the most important stage when a new point is randomly inserted. Although there are several techniques to perform this reconstruction, one of the most relevant is a validation technique called “empty circle”, described by Boris Deloné. In this paper, we focus on the use of the Conformal Geometric Algebra (CGA) to perform such validation. In addition, the proposal includes a mathematical environment change to show the advantages of using CGA’s geometric entities and use them inside a module for validating the triangulation.

Keywords. Conformal geometric algebra, empty circle, Delaunay triangulation.

1. Propuesta del trabajo

La importancia del empleo del AGC en el cómputo geométrico respecto a problemas

relacionados con Triangulaciones Delaunay y Diagramas de Voronoi ha empezado a tomar importancia. Las ventajas de tomar entidades geométricas y poder representarlas como vectores nos provee de fórmulas cortas y con sentido intuitivo geométrico. Además se pueden tomar operadores con diferentes objetos y realizar operaciones fáciles de entender. En el trabajo de Zonnenberg [1] resuelve el problema de la triangulación Delaunay y del diagrama de Voronoi, para ello utiliza conceptos del AGC y los aplica en un software dedicado a algoritmos de cómputo geométrico, llamado CGAL [2]. Leo Dorst *et al.* [3] proponen un ejercicio para construir los diagramas de Voronoi a partir de una triangulación Delaunay utilizando los algoritmos de elevación con su software GA Sandbox y GAViewer [4]. Implementado el algoritmo de elevación mencionado, se resuelve en el trabajo de Linwang Yuan *et al.* [5] quienes desarrollan un software llamado CAUSTA, donde utilizan las propiedades del Álgebra Geométrica para elaborar herramientas útiles para los SIG. De lo anterior, o emplean software existente para resolver los enigmas del AGC o no llegan a detalles de la implementación. La intención de este trabajo es comenzar con el concepto básico de la triangulación Delaunay utilizando una matemática más intuitiva y analizando las opciones para realizar las operaciones con las entidades básicas sin el empleo de una herramienta de software.

Se realiza un ejemplo simple pero poderoso en el sentido que da validez a la triangulación sin llegar a triangular la estructura. Para ello escogemos un algoritmo de construcción sencillo que funciona paso a paso y resulta ideal para

analizar solamente un área local de reconstrucción de triángulos, con base en la relación que hay entre un punto y el círculo. Esta relación es la que queremos llevar a otro dominio para aprovechar las entidades que nos proporciona el AGC, más sin embargo este principio que se plantea es la base para que en un futuro trabajo no se utilicen las intersecciones de líneas o planos como lo describen los trabajos anteriores y proponer una triangulación sin triángulos. Además el triángulo resulta un objeto difícil de implementar y sin ecuaciones que lo describan perfectamente, a diferencia de una esfera que resulta fácil de representar y es idónea en el campo de la geometría.

2. Introducción

La triangulación puede ser contemplada desde diferentes puntos de vista, nos enfocaremos como una forma de subdividir un área empleando triángulos. Si se quiere generalizar la subdivisión para un espacio con d dimensiones arbitrarias, se tomará en cuenta el simplejo que es análogo a las d dimensiones de un triángulo. Para nuestro caso nos enfocaremos a un espacio de dos dimensiones y la triangulación que construiremos será única ya que el método para conectar los puntos arbitrarios en forma de triángulos tiene que cumplir con la condición de Delaunay. Dicha condición establece que ningún punto debe estar contenido en una circunferencia circunscrita en cada triángulo, establecida por el matemático ruso Boris Deloné [6], quien en su forma francesa Delaunay le da el nombre a la triangulación. En la Fig. 1 se aprecia una triangulación Delaunay, que se considera una forma única de realizar el mallado sin que las aristas que unen los vértices se crucen.

Las áreas más comunes para realizar aplicaciones con la triangulación Delaunay fuera del cómputo geométrico son: modelado digital del terreno, métodos de los elementos finitos, robótica, visualización científica, gráficas por computadora, visión por computadora y Sistemas de Información Geográfica por mencionar algunas. Y por citar algunas referencias de SIG con aplicaciones en triangulaciones tenemos [7, 8, 9].

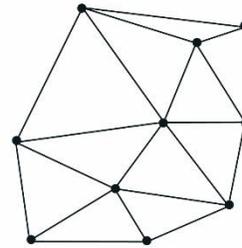


Fig. 1. Una triangulación Delaunay

Antes de que se establecieran los conceptos matemáticos de la triangulación Delaunay, en 1843 William R. Hamilton introduce el álgebra de los cuaterniones, llegando a concretar un espacio de tres dimensiones representado por tres valores. Un año después del trabajo de Hamilton, Hermann Grassmann [10] publica su obra "Teorema de la Extensión" donde expresa su visión matemática para la geometría, pero su trabajo no fue reconocido como debería ser en su tiempo. Una de sus mejores aportaciones fue la definición de un nuevo producto llamado actualmente como producto exterior. Para finales del siglo XIX William K. Clifford [11] observó que las ideas del producto exterior de Grassmann y los cuaterniones de Hamilton se pueden relacionar para obtener el producto escalar y exterior en un nuevo concepto llamado producto geométrico. Sin embargo, no fue hasta los años de 1980 cuando David Hestenes [12] quien reconoció la importancia del álgebra geométrica y a partir de ahí se generaron trabajos de investigación en diversas áreas como la física, robótica, visión por computadora y otros campos de las ciencias computacionales [13]. Con la idea de reformular algunos problemas geométricos el modelo conforme se presenta como una alternativa al espacio euclidiano, sin dejar sus propiedades atrás.

El esquema geométrico que presenta nos da la facilidad de trabajar con puntos e hiperesferas que nos ayudarán a manejar las condiciones Delaunay para cualquier dimensión y en una forma simple y sencilla. Solamente resolveremos la condición Delaunay utilizando el Álgebra Geométrica Conforme, y escribiendo algunas rutinas en un lenguaje de programación demostrando el uso de la matemática alterna.

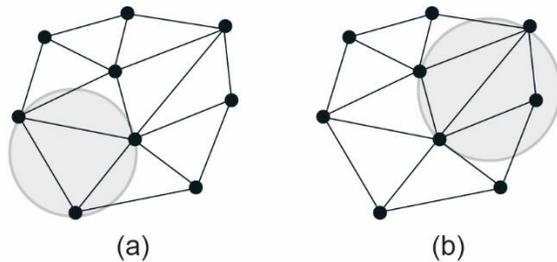


Fig. 2. La condición Delaunay puede mostrarse como:
(a) Triangulación legal (b) Triangulación ilegal

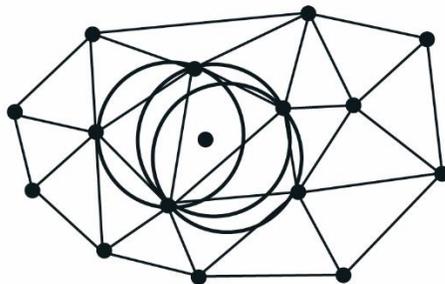


Fig. 3. Algoritmo Bowyer-Watson, se seleccionan los circuncírculos afectados por el punto insertado

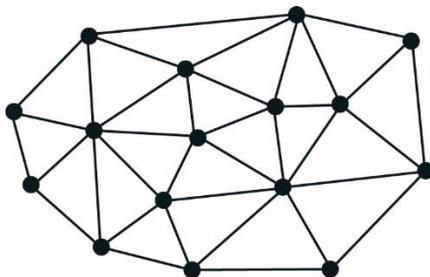


Fig. 4. Algoritmo Bowyer-Watson, reconstrucción de los triángulos eliminados porque su circuncírculo fue afectado por el nuevo punto

3. Condición Delaunay

Entre los algoritmos más comunes para construir la triangulación Delaunay se encuentra el incremental [14, 15], los de barrido [16] y el divide y vencerás [17, 18]. La propiedad principal que hace única a la triangulación Delaunay TD construida a partir de un conjunto finito de puntos, es que ningún punto en el plano este contenido en el circuncírculo de cualquier triángulo de la TD. En

la Fig. 2 se aprecia un ejemplo donde un triángulo cumple con la propiedad del círculo vacío y en la otra donde el circuncírculo de un triángulo contiene un vértice de otro triángulo. En el trabajo de Lawson [19] demuestra que cualquier triangulación puede ser transformada a la TD siempre y cuando se cumpla la propiedad del círculo vacío. Lo anterior se define también como el “método de flip” [20] donde se define la legalidad de un triángulo.

Uno de los algoritmos para realizar la construcción de la triangulación Delaunay que nos parece adecuado para poner en prueba la condición Delaunay será el de Bowyer-Watson [21, 22]. El algoritmo, se agrega secuencial y aleatoriamente de Bowyer-Watson funciona de manera incrementalmente cada punto a la TD. Para reconstruir la triangulación debido al nuevo punto, es necesario encontrar los circuncírculos donde cae dicho punto. Para ello no es necesario barrer todos los circuncírculos de la triangulación, ya que llevará a una complejidad $O(n^2)$, y tendrá un costo computacional muy alto. Por lo que primero es necesario encontrar solamente el triángulo afectado por el punto insertado. El trabajo de Watson [22] menciona una complejidad de $O(n^{1.5})$, mientras que el propuesto por Mucke [23] obtiene una complejidad cercana a $O(n^{1.3})$, para localizar el punto en la estructura triangular. Una vez encontrado el triángulo, éste y sus vecinos son candidatos para ver cuales circuncírculos contienen al punto. Los circuncírculos seleccionados, como se muestran en la Fig. 3, establecen los triángulos que serán eliminados, dejando un hueco en el cual se triangulará con base en el nuevo punto. En la Fig. 4 se observa el resultado de la reconstrucción de la triangulación.

4. Álgebra geométrica conforme

Consideremos el espacio conforme como una mejora del espacio euclidiano. De tal manera que el espacio euclidiano estará embebido en un espacio de mayor dimensión, donde las dimensiones extras tendrán propiedades especiales que nos ayudarán a crear entidades geométricas. El término conforme nos indica que se preservan los ángulos y sus propiedades se nos presentan como una alternativa al espacio

euclidiano para resolver problemas geométricos [24, 25]. Si deseamos trabajar con objetos geométricos en 3D, tendremos que utilizar un espacio euclidiano en \mathbb{R}^3 , por lo que el espacio conforme requiere de cinco dimensiones. Una razón de este cambio es poder formular los problemas más intuitivamente en un espacio con dos dimensiones de más. El AGC utiliza tres vectores euclidianos básicos e_1 , e_2 y e_3 , cuentan con dos adicionales e_+ y e_- , y al elevarlos al cuadrado toman los valores de 1 y -1 respectivamente. Obtenemos la siguiente relación:

$$e_0 = \frac{1}{2}(e_- - e_+), \quad \text{y} \quad e_\infty = \frac{1}{2}(e_- + e_+),$$

donde el primero representa el origen euclidiano y el segundo representa el punto al infinito.

Con las propiedades del producto geométrico y del producto exterior, el AGC nos provee de entidades geométricas básicas que tienen dos presentaciones la IPNS (*Inner Product Null Space*) y la OPNS (*Outer Product Null Space*), la lista de entidades geométricas se observa en la tabla 1 y éstas representaciones son duales.

En la representación OPNS, con el producto exterior \wedge se pueden construir entidades geométricas a partir de la entidad punto P (de las cuales están constituidas), como la esfera, círculo y un par de puntos. Y si incluye el vector que representa el punto al infinito se puede representar el plano o la línea. Como ejemplo, para la esfera S definida en la representación IPNS se puede definir también por cuatro puntos P .

Donde x y n representan entidades en d dimensiones, por ejemplo x se obtiene por la siguiente combinación básica de vectores:

$$x = x_1 e_1 + x_2 e_2 + \dots + x_d e_d.$$

5. Álgebra geométrica conforme con la condición Delaunay

Cabe aclarar que estamos empleando el concepto de esfera para poder generalizar la condición Delaunay en cualquier dimensión. Podemos construir una triangulación Delaunay en dos dimensiones o más y para ello la esfera como elemento del AGC puede cumplir. Sin embargo las explicaciones y ejemplos que trabajaremos serán para una triangulación en dos

Tabla 1. Entidades del AGC

Entidad	Representación IPNS	Representación OPNS
Punto	$P = x + (1/2)x^2 e_\infty + e_0.$	
Esfera	$S = P - (1/2)r^2 e_\infty.$	$S^* = P_1 \wedge P_2 \wedge P_3 \wedge P_4.$
Plano	$\pi = n + d e_\infty.$	$\pi^* = P_1 \wedge P_2 \wedge P_3 \wedge e_\infty.$
Círculo	$Z = S_1 \wedge S_2.$	$Z^* = P_1 \wedge P_2 \wedge P_3.$
Línea	$L = \pi_1 + \pi_2.$	$L^* = P_1 \wedge P_2 \wedge e_\infty.$
Par de puntos	$P_p = S_1 \wedge S_2 \wedge S_3.$	$P_p^* = P_1 \wedge P_2.$

dimensiones (como se ha comentado), sin dejar de usar el término esfera, el cual también se puede reducir a un círculo.

5.1. Punto y la esfera

Necesitamos solamente de dos objetos geométricos que son el punto y la esfera para realizar la validación de la condición Delaunay. El punto P se definen en la forma IPNS como:

$$P = x + \frac{1}{2}x^2 e_\infty + e_0. \quad (1)$$

El punto original $x \in \mathbb{R}^d$, donde x^2 se define como el producto escalar.

$$x^2 = x_1^2 + x_2^2 + \dots + x_d^2. \quad (2)$$

Una esfera S en la forma IPNS, se puede representar por un punto en el centro de la esfera y radio r , como se muestra:

$$S = P - \frac{1}{2}r^2 e_\infty, \quad (3)$$

si el punto P es reemplazado por (1), tenemos:

$$S = x + \frac{1}{2}(x^2 + r^2)e_\infty + e_0. \quad (4)$$

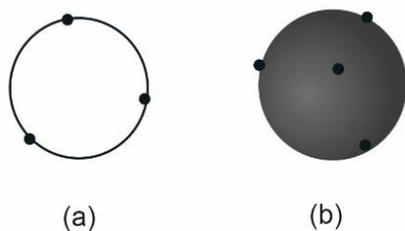


Fig. 5. (a) Construcción de un círculo con 3 puntos (b) Construcción de una esfera con 4 puntos

5.2. Construcción de la esfera

Una vez que se localiza el triángulo afectado por el nuevo punto, es importante construir los semicírculos de los triángulos cercanos al triángulo afectado. Con el objetivo de identificar los círculos que contienen al nuevo punto insertado, ver Fig. 3. Para ello es necesario calcular el centro y radio como característica de cada círculo, donde un círculo se puede construir con un mínimo de tres puntos siempre y cuando no sean colineales, ver Fig. 5a. Un círculo se puede representar en la forma OPNS como:

$$Z^* = P_1 \wedge P_2 \wedge P_3, \quad (5)$$

si la triangulación Delaunay se realizara en tres dimensiones sería necesario trabajar con esferas que se pueden construir a partir de cuatro puntos siempre y cuando no sean coplanares, ver Fig. 5b. La esfera en la forma OPNS se representa como:

$$S^* = P_1 \wedge P_2 \wedge P_3 \wedge P_4. \quad (6)$$

En la matemática del AGC resulta fácil la implementación de un círculo o esfera a partir de tres o cuatro puntos respectivamente, pero para ello necesitamos de un visualizador como: GAViewer [12] y CLUCalc [26]. O si el trabajo es realizar un algoritmo expresado con AGC y poder trabajar las ecuaciones (5) y (6) es necesario utilizar un compilador como el Gallop [27, 28], que trabaja eficientemente este tipo de operaciones. Muchas veces es importante que el programador no conozca los detalles que hay detrás de las matemáticas, por eso las herramientas de

software mencionadas ocultan las operaciones de matrices o sistemas lineales.

Sin embargo, haciendo dichas herramientas computacionales de lado, trabajaremos dentro de las operaciones para construir esferas y las operaciones que puedan existir entre dos entidades como el punto y la esfera. Por lo que realizaremos un módulo que calcule el centro y radio de un círculo utilizando un ajuste por mínimos cuadrados dentro del Álgebra Geométrica. Necesitaremos de tres puntos para obtener las características del círculo, esos tres puntos son: $P_i \in \mathbb{R}^2, i \in \{1,2,3\}$, el punto se representa como (1) y la esfera como (3). El producto escalar provee la medida de distancia y por medio de un ajuste por mínimos cuadrados de:

$$\sum_{i=1}^3 [P_i \cdot S] = \sum_{i=1}^3 \left[p_i \cdot s_i - \frac{1}{2} p_i^2 - \frac{1}{2} (s - r^2) \right]^2, \quad (7)$$

encontrando el mínimo de (7) e igualando a cero, nos proporciona un sistema de tres ecuaciones y tres incógnitas, cuyo sistema es:

$$\begin{pmatrix} \sum_{i=1}^3 p_{i,1} p_{i,1} & \sum_{i=1}^3 p_{i,2} p_{i,1} & -\sum_{i=1}^3 p_{i,1} \\ \sum_{i=1}^3 p_{i,1} p_{i,2} & \sum_{i=1}^3 p_{i,2} p_{i,2} & -\sum_{i=1}^3 p_{i,2} \\ -\sum_{i=1}^3 p_{i,1} & -\sum_{i=1}^3 p_{i,2} & \sum_{i=1}^3 1 \end{pmatrix} \cdot s = \begin{pmatrix} \frac{1}{2} \sum_{i=1}^3 p_i^2 p_{i,1} \\ \frac{1}{2} \sum_{i=1}^3 p_i^2 p_{i,2} \\ \frac{1}{2} \sum_{i=1}^3 p_i^2 \end{pmatrix}. \quad (8)$$

Si deseamos encontrar el centro y radio de una esfera la ecuación (8) será de 4×4 .

A continuación, se muestra un segmento de código en lenguaje de programación java que resuelve el ajuste de un círculo.

```
// Matriz que contiene tres puntos
double pts[][] =
    {{p1.getXplano(), p1.getYplano()},
     {p2.getXplano(), p2.getYplano()},
     {p3.getXplano(), p3.getYplano()}};
// Iteración que calcula las sumatorias
for (int i = 0 ; i < 3 ; i ++ ) {
a[0][0] += pts[i][0] * pts[i][0];
a[0][1] += pts[i][0] * pts[i][1];
a[0][2] += -pts[i][0];
a[1][0] += pts[i][0] * pts[i][1];
a[1][1] += pts[i][1] * pts[i][1];
a[1][2] += -pts[i][1];
a[2][0] += -pts[i][0];
```

```

a[2][1] += -pts[i][1];
a[2][2] += 1;
d = Math.pow(pts[i][0],2) +
    Math.pow(pts[i][1],2);
b[0] += 0.5 * d * pts[i][0];
b[1] += 0.5 * d * pts[i][1];
b[2] += 0.5 * d ;
}
// Método que resuelve un sistema
// de ecuaciones lineales
calcSEL (a ,b ,x ,3);
// Se obtiene el centro del círculo
xcent = x[0];
ycent = x[1];
// Se obtiene el radio del círculo
radio = Math.sqrt(x[0]*x[0]+
    x[1]*x[1] - 2*x[2]);

```

Código 1. Sección encargada de obtener el radio y centro de un círculo a partir de tres puntos

5.3. Determinar si un punto está dentro de una esfera

El concepto importante, es identificar en cuáles semicírculos tiene contenido el nuevo punto que ha sido insertado para generar los nuevos triángulos en la TD. En el AGC usaremos el producto punto entre un punto y una esfera para determinar si está dentro, en o fuera de la esfera. Sea un punto P y sea S una esfera, el producto punto de estos dos cuerpos geométricos es:

$$P \cdot S = \left(\mathbf{p} + \frac{1}{2} \mathbf{p}^2 \mathbf{e}_\infty + \mathbf{e}_0 \right) \cdot \left(\mathbf{s} + \frac{1}{2} (\mathbf{s}^2 + r^2) \mathbf{e}_\infty + \mathbf{e}_0 \right),$$

se reduce a:

$$P \cdot S = -\frac{1}{2} (\mathbf{s} - \mathbf{p})^2 + \frac{1}{2} r^2. \quad (9)$$

El resultado del producto punto de un punto con una esfera multiplicado por -2 , se identifica con la potencia de un punto:

$$-2(P \cdot S) = (\mathbf{s} - \mathbf{p})^2 - r^2. \quad (10)$$

Se identifican las siguientes relaciones:

- Si $P \cdot S > 0$, P está dentro del círculo (Fig. 6a),
- Si $P \cdot S = 0$, P está en el contorno del círculo (Fig. 6b),
- Si $P \cdot S < 0$, P está fuera del círculo (Fig. 6c).

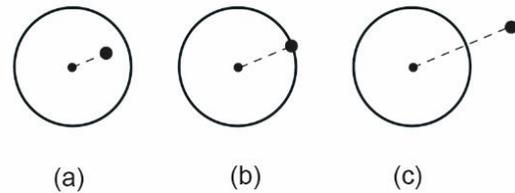


Fig. 6. Ubicación de un punto con respecto a un círculo

Enseguida se muestra un segmento de código en java para determinar si un punto cae dentro de un círculo o no.

```

// Producto punto de P (punto) y S (esfera)
double ps = Math.pow(s.getRadio(), 2) -
    Math.pow(p.getXplano(), 2) +
    Math.pow(p.getYplano(), 2) -
    2*(s.getXcent() * p.getXplano() +
    s.getYcent() * p.getYplano()) +
    Math.pow(s.getXcent(), 2) +
    Math.pow(s.getYcent(), 2 );
// Se evalua la condición del producto punto
if (ps > 0)
    return 1; // P está dentro del círculo
if (ps == 0)
    return 0; // P está en el círculo
if (ps < 0)
    return -1; // P está fuera del círculo

```

Código 2. Sección que evalúa la condición Delaunay

6. Algoritmo

El tipo de triangulación Delaunay en el que estamos enfocados, es el que se implementa en forma incremental insertando punto por punto, para ello es necesario varias estructuras de datos que nos permitirán un mejor eficiencia durante su construcción y la ubicación del punto a insertar.

Lo importante es entender la relación entre la esfera y el simplejo, para nuestro caso será la dualidad entre el círculo y el triángulo. Para el caso de dos dimensiones, se construirá un círculo con tres puntos, los mismos que conforma el triángulo (ver Fig. 7a). Las aristas que comparten los triángulos, se especifican con el segmento de línea delimitado por dos puntos que comparten dos círculos (ver Fig. 7b); por ejemplo en la Fig. 7b el círculo de la derecha es un círculo frontera

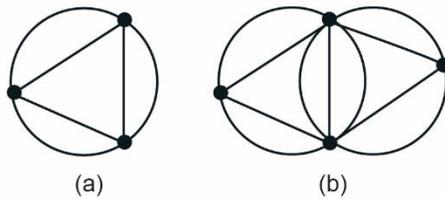


Fig. 7. Relación entre el círculo y triángulo, y con respecto a la arista que comparten

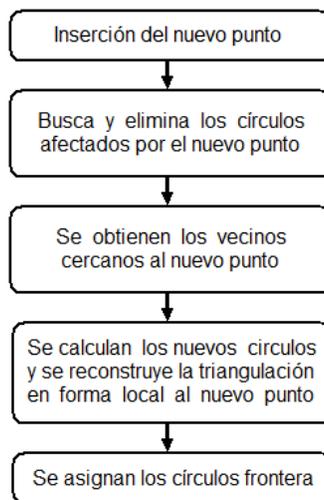


Fig. 8. Algoritmo completo para realizar la triangulación Delaunay con base en círculos

6.1. Propuesta de un algoritmo completo

Para que nuestro algoritmo logre una complejidad $O(n \log n)$ es necesario mejorar el algoritmo caminante, donde Mucke [23] alcanza una complejidad de $O(n^{1.3})$. Para ello, primero se tendrá que utilizar una estructura de datos del tipo de tabla Hash, que se empleará para la búsqueda de los elementos de puntos y esferas, donde su performance es de $O(1)$ siempre que los datos estén distribuidos uniformemente. Para localizar el nuevo punto en la estructura se utilizará una estructura de tipo árbol KD, la cual nos garantiza la complejidad deseada. En la Fig. 8 se muestran los pasos importantes del algoritmo para realizar la triangulación con base en una estructura circular. El AGC nos ayuda a identificar los círculos afectados (segundo paso del algoritmo) y también

nos permite construir los círculos (cuarto paso del algoritmo). Para obtener los vecinos cercanos al punto de inserción se necesita realizar una rutina que extrae los puntos de los círculos afectados (tercer paso del algoritmo) y otra rutina para catalogar los círculos frontera que son necesarios para reconocer las aristas que comparten los triángulos (cuarto paso del algoritmo). El algoritmo se repite cada que se procede a insertar un nuevo punto.

Una de las ideas del presente trabajo yace en el concepto utilizado por el trabajo de Bowyer-Watson, donde usan los circuncírculos para reconstruir la triangulación por el nuevo punto. Nosotros proponemos toda una estructura de círculos, los cuales presentan una dualidad con los triángulos, y de esta forma hacer las operaciones mediante el AGC. Una de las ventajas del álgebra conforme es trabajar con puntos (P) y círculos (Z) como objetos geométricos en toda la estructura y auxiliariamente ir triangulando.

6.2. Triangulación cuando se inserta un punto

A continuación, se realiza un ejemplo práctico del funcionamiento de nuestro algoritmo con el procedimiento incremental para la construcción de la triangulación Delaunay, con el principio de los círculos para un caso de dos dimensiones. Primeramente, se tiene una triangulación con base en las propiedades de círculos, a continuación se agrega un nuevo punto a la estructura inicial y se identifican los círculos afectados por dicho punto, ver Fig. 9a. Los círculos seleccionados se eliminan de la estructura y también las aristas internas, ver Fig. 9b. Posteriormente, se crean los nuevos círculos alrededor del nuevo punto como en la Fig. 9c, para finalmente construir las aristas internas con las propiedades de los círculos, ver Fig. 9d.

De esta forma se realiza la triangulación con la llegada de un nuevo punto.

La construcción de la triangulación Delaunay se propone realizarla a través de círculos y no de triángulos, posteriormente se aplica la relación entre ellos para construir las aristas correspondientes. Si nos comparamos con la metodología más conocida para realizar la triangulación de un nuevo punto, se utiliza el "método de flip" que maximiza el ángulo mínimo de los triángulos afectados para que se cumpla la

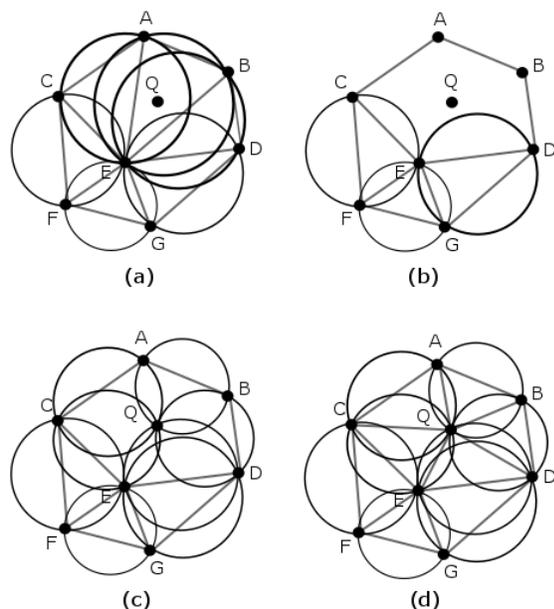


Fig. 9. Metodología para realizar la triangulación cuando cae un nuevo punto

condición Delaunay. Sin embargo, el cálculo de funciones trigonométricas requiere de un alto costo computacional y se continúa con el empleo de triángulos que son difíciles de construir geoméricamente, en especial cuando las dimensiones se incrementan.

6.3. Contribuciones y ventajas

Actualmente la mayoría de los problemas en el cómputo Geométrico se resuelven utilizando la geometría Euclidiana. A pesar de que el álgebra geométrica es una herramienta matemática que ya tiene tiempo de haberse desarrollado, todavía se desconoce y son algunos especialistas en ciencias de la ingeniería y físicos quienes les han dado un mayor uso. Como lo hemos mencionado a lo largo del trabajo, el AGC nos proporciona una manera eficaz de entender un espacio geométrico; por lo que la importancia de esta herramienta será utilizar operaciones intuitivas entre las hipersferas y los puntos. Para nuestro caso emplearemos círculos, los que se utilizarán en el algoritmo para crear las triangulaciones Delaunay. Es por ello que contribuimos con una forma alterna de construcción donde existen operaciones

uniformes entre cuerpos geométricos, a diferencia del uso de los triángulos.

Por simplicidad es eficaz el uso de hipersferas en cualquier dimensión, ya que se pueden representar solamente por su centro y radio. Mientras que el triángulo o simplejo en dos dimensiones está conformado por tres vértices, tres aristas y una superficie, para el tetraedro o simplejo en tres dimensiones será necesario cuatro vértices, seis aristas, cuatro caras y un volumen. Los elementos requeridos para un n -simplejo son de $2^{n+1} - 1$ elementos. También es conveniente usar los conceptos de esferas porque presentan una ventaja en el costo computacional. Saber si un punto está dentro o fuera de un triángulo requiere el uso de determinantes (ecuación 11), y para los círculos utilizamos una operación sencilla (ecuación 10). El problema se complica conforme las dimensiones crecen, ya que encontrar un punto dentro de un tetraedro (ecuación 12) resulta complicado a diferencia de una esfera (ecuación 10). Como se aprecia, encontrar un punto dentro de una hipersfera siempre es el mismo problema, independientemente de la dimensión en la que se trabaje. En la siguiente sección se explica esta ventaja a más detalle.

7. Ventajas de la esfera vs triángulo

Se ha observado que la operación es simple para validar la triangulación Delaunay y muestra una gran ventaja a diferencia de encontrar un punto al interior de un triángulo. Si tenemos un punto $Q = (x, y)$ y un triángulo con vértices en $A = (x_1, y_1)$, $B = (x_2, y_2)$ y $C = (x_3, y_3)$. Para determinar si el punto Q está dentro o fuera del triángulo (ver Fig. 10) podemos emplear el criterio de signo de los siguientes determinantes $D(A, B, Q)$, $D(A, Q, C)$ y $D(Q, B, C)$. Por ejemplo sea el primer determinante:

$$D(A, B, Q) = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x & y & 1 \end{vmatrix}. \quad (11)$$

Para ello se observan varios cálculos, y la situación se complica cuando las dimensiones crecen, si queremos resolver la validación para

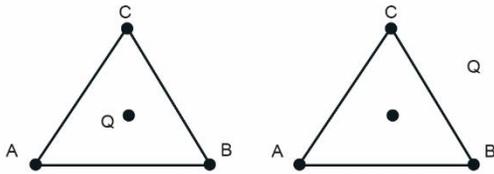


Fig. 10. Punto dentro o fuera de un triángulo

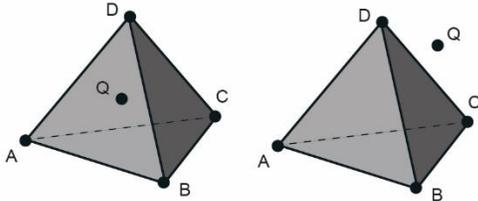


Fig. 11. Punto dentro o fuera de un tetraedro

una triangulación Delaunay en 3D, es necesario saber si el punto está dentro o fuera de un tetraedro (ver Fig. 11). Para ello es necesario usar cuatro determinantes de la forma $D(A, B, C, Q)$, $D(A, B, Q, D)$, $D(A, Q, C, D)$ y $D(Q, B, C, D)$. Sea el primer determinante de la forma:

$$D(A, B, C, Q) = \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x & y & z & 1 \end{vmatrix}, \quad (12)$$

mientras que para cualquier dimensión en el AGC siempre se usará la ecuación 10.

8. Conclusiones y trabajo a futuro

La motivación de este trabajo es resaltar una opción matemática diferente para realizar las validaciones en la triangulación Delaunay, con la ayuda de entidades geométricas que nos proporciona el AGC. Se logra comprender la operación necesaria para conocer si un punto está dentro o fuera de un círculo, ya que el AGC proporciona operaciones sencillas sobre sus entidades punto y esfera. Los cambios que empleamos para usar círculos en vez de triángulos, presenta una ventaja geométrica de construcción en la estructura y además en el costo

computacional, como lo hemos explicado anteriormente.

La idea posterior es construir totalmente la triangulación Delaunay en el ámbito del AGC, con el principio de utilizar la entidad esfera en vez de un triángulo y aprovechar todas las operaciones existentes. Además se beneficia de las características de la esfera, como sus ecuaciones representativas y lo simple que resulta implementarlo solamente con centro y radio. A diferencia del simplejo que es el resultado de varias figuras geométricas y se complica conforme crece en dimensiones.

Agradecimientos

Este trabajo fue realizado gracias al apoyo del Instituto Politécnico Nacional a través del proyecto SIP-20160828.

Referencias

1. **Zonnenberg, C. (2007).** *Conformal Geometric Algebra Package*. Master thesis, Utrecht University.
2. **The Computational Geometry Algorithms Library.** <http://www.cgal.org/>
3. **Dorst, L., Fontijne, D., & Mann, S. (2007).** *Geometric Algebra for Computer Science: an object oriented approach to geometry*. Morgan Kaufman publishers, pp. 415–417.
4. **Geometric Algebra for Computer Science and Object Oriented approach for Geometry.** <http://www.geometricalgebra.net/index.html>
5. **Yuan, L., Yu, Z., Chen, S., Luo, W., Wang, Y., & Lü, G. (2010).** CAUSTA: Clifford Algebra-based Unified Spatio-Temporal Analysis. *Transactions in GIS*, 14, pp. 59–83.
6. **Deloné, B. (1934).** Sur la sphere vide. *Bulletin of the Academy of Sciences of the U.S.S.R.*, 6, pp. 793–800.
7. **Park, D., Cho, H., & Kim, Y. (2001).** A TIN compression method using Delaunay triangulation. *Int. J. Geographical Information Science*, Vol. 15, No. 5, 255–269.
8. **Kang, I., Kim, T., & Li, K. (1997).** A spatial data mining method by Delaunay triangulation. *5th ACM int. workshop on Advances in GIS*, pp. 35–39.
9. **Heller, M. (1990).** Triangulation algorithms for adaptive terrain modeling. *Proc. 4th Internat. Sympos. Spatial Data Handling*, pp. 163–174.

10. **Grassmann, H. (1844).** *Linear Extension Theory (Die Lineale Ausdehnungslehre)*, translated by L. C. Kannenberg in the *Ausdehnungslehre* of 1844 and Other Works, Open Court Publ.
11. **Clifford, W.K. (1878).** Applications of Grassmann's Extensive Algebra. *American Journal of Mathematics*, Vol. 1, No. 4, pp. 350–358.
12. **Hestenes, D. & Sobczyk, G. (1984).** *Clifford Algebra to Geometric Calculus*. Reidel Publ. Co., Dordrecht/Boston.
13. **Bayro, E. (2005).** *Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics*. Springer-Verlag, Berlin Heidelberg.
14. **Guibas, L. J. & Stolfi, J. (1985).** Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, Vol. 4, No. 2, pp. 74–123.
15. **Guibas, L. J., Knuth, D., & Sharir, M. (1992).** Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, Vol. 7, No. 1, pp. 381–413.
16. **Fortune, S. (1987).** A sweep-line algorithm for Voronoi diagrams. *Algorithmica*, 2, pp. 153–174.
17. **Lee, D. T. & Schachter, B. J. (1980).** Two algorithms for constructing a Delaunay triangulation. *Journal of Computer and Information Sciences*, Vol. 9, No. 3, pp. 219–242.
18. **Dwyer, R. A. (1987).** A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica*, 2, pp. 137–151.
19. **Lawson, C.L. (1977).** *Mathematical software III; Software for C1 surface interpolation*. Academic Press, pp. 161–94.
20. **Berg, M., Kreveld, M., Overmars, M., & Schwarzkopf, O. (1997).** *Computational geometry, algorithms and applications*. Berlin: Springer.
21. **Bowyer, A. (1981).** Computing Dirichlet tessellations. *The Computer Journal*, Vol. 24, No. 2, pp. 162–166.
22. **Watson, D. (1981).** Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, Vol. 24, No. 2, pp. 167–172.
23. **Mücke, E.P., Saias, I., & Zhu, B. (1999).** Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations. *Computational Geometry*, 12, pp. 63–83.
24. **Hildenbrand, D. (2013).** *Foundations of Geometric Algebra Computing*. Springer-Verlag, Berlin Heidelberg.
25. **Perwass, C. (2009).** *Geometric Algebra with Applications in Engineering*. Springer-Verlag, Berlin Heidelberg.
26. **Interactive Visualization.** <http://www.clucalc.info>
27. **GAALOP Geometric Algebra Algorithms Optimizer.** <http://www.gaalop.de>.
28. **Schwinn, C., Hildenbrand, D., Stock, F., & Koch, A. (2010).** Gaalop 2.0 - a geometric algebra algorithm compiler. *International Workshop on Computer Graphics, Computer Vision and Mathematics*, 2, pp. 1–8.

Netz Romero obtuvo el título de Ingeniero Físico y el grado de Maestro en Ciencias de la Computación en la Universidad Autónoma Metropolitana, México, en 1999 y 2003 respectivamente. Ocupando varios puestos en empresas privadas y como docente. Actualmente es alumno del programa de Doctorado en Ciencias de la Computación en el CIC del Instituto Politécnico Nacional. Áreas de interés son el cómputo geométrico y el cómputo paralelo.

Ricardo Barrón Fernández obtuvo el título de Matemático en la Facultad de Ciencias de la UNAM en 1985. Es Maestro en Ciencias en Ingeniería de Cómputo y Doctor en Ciencias de la Computación por el Instituto Politécnico Nacional, en 1998 y 2006 respectivamente. Trabaja como profesor e investigador en el laboratorio de Inteligencia Artificial del Centro de Investigaciones en Computación del IPN. Las áreas de interés son las matemáticas computacionales y las aplicaciones de la inteligencia artificial.

*Artículo recibido el 28/06/2016; aceptado el 05/12/2016.
Autor de correspondencia Netz Romero.*