

Utilización de árboles de cubrimiento para interpolar usando funciones de base radial enfocado a la visualización científica de grandes volúmenes de datos

Yaidel Reyes López, Heikel Yervilla Herrera, Alcides Viamontes Esquivel,
Carlos A. Recarey Morfa
Centro Investigación de Métodos Computacionales y Numéricos en la Ingeniería
(CIMCNI Aula UCLV-CIMNE)
Universidad Central de Las Villas (UCLV)
Carretera a Camajuaní km 5 $\frac{1}{2}$
54830, Santa Clara, Villa Clara, Cuba
Tel./Fax: 53-42-224746
e-mail: yaidelr@uclv.edu.cu; yervilla@uclv.edu.cu; alve@uclv.edu.cu; recarey@uclv.edu.cu

Resumen

En el presente trabajo se desarrolla un método para interpolar grandes volúmenes de datos esparcidos, dirigido principalmente a los resultados de la aplicación de Métodos libres de Malla, Métodos de Punto y de Partículas. En el mismo se hace uso de las funciones de base radial con alcance local como funciones interpoladoras. Se utilizan los árboles de cubrimiento como la estructura de datos que permite acelerar la localización de los datos que influyen para interpolar los valores en un nuevo punto, lo cual agiliza la aplicación de técnicas de visualización científica para la generación de imágenes a partir de grandes volúmenes de datos provenientes de la aplicación de Métodos libres de Malla, Métodos de Puntos, y de Partículas, en la resolución de diversos modelos de la física-matemática. Como ejemplo se muestran los resultados obtenidos tras la utilización de dicho método, empleando la función interpoladora de Shepard de alcance local.

Palabras clave: *datos esparcidos, interpolación, visualización científica, árbol de cubrimiento, funciones de base radial, Método de Shepard, Métodos de Puntos, Métodos de Partículas, Métodos libres de Malla.*

USING COVER-TREES TO INTERPOLATE WITH RADIAL BASIS FUNCTION FOCUSED ON SCIENTIFIC VISUALIZATION OF LARGE VOLUMES OF DATA

Summary

In the following paper it is developed a new method to interpolate large volumes of scattered data, focused mainly on the results of the Mesh-free Methods, Points Methods and the Particles Methods application. Through this one, local radial basis functions are used as interpolating functions. Also, cover trees are used as the data structure that allows to accelerate the localization of the data that influences to interpolate the values at a new point, speeding up the application of scientific visualization techniques to generate images from large data volumes from the application of Mesh-free Methods, Points and Particles Methods, in the resolution of diverse models of physics-mathematics. As an example, the results obtained after applying this method using the local interpolation functions of Shepard are shown.

Keywords: *scattered data, interpolation, scientific visualization, cover-tree, radial basis functions, Shepard's Method, Points Methods, Particles Methods, Mesh free/less Methods.*

INTRODUCCIÓN

El término **visualización científica** se refiere al proceso que envuelve la utilización de la computadora con el fin de obtener imágenes a partir de datos. En el caso de la mecánica computacional se refiere a la visualización científica de grandes volúmenes de datos escalares, vectoriales y de otra índole provenientes de los resultados de simulaciones y modelaciones de problemas de la física-matemática. En esta disciplina es necesaria la utilización de técnicas de computación gráfica y análisis numérico. La visualización científica constituye una herramienta extremadamente útil para los investigadores ya que facilita un estudio de los datos basándose en la capacidad visual de los seres humanos. La obtención de las imágenes se realiza a partir de la aplicación de técnicas de visualización, que son seleccionadas de acuerdo al tipo de datos a representar, con el objetivo de lograr una mejor percepción. En [13,17] se hace una amplia referencia a las técnicas existentes, así como a su selección de acuerdo al tipo de datos.

Los datos a visualizar son magnitudes de determinadas propiedades -físicas, mecánicas, térmicas, etc.- obtenidas ya sea por mediciones realizadas en el mundo real, o a partir de procesos numéricos; en ambos casos y aunque generalmente representan medios continuos, lo que se tiene es un conjunto finito de valores pertenecientes a puntos discretos del dominio de estudio. Estos puntos pueden ser tomados de forma regular o irregular/aleatoria. Cuando los puntos son tomados de manera aleatoria entonces se les da el nombre de “datos esparcidos”^{*}.

Como consecuencia de las limitaciones presentadas en la aplicación de métodos tradicionales -dígase Métodos de los Elementos Finitos, Diferencias Finitas o Elementos de Contorno- para tratar problemas con altos niveles de discontinuidad, problemas en campos no ingenieriles como la biomecánica, entre otros, algunos métodos que no requieren de la generación de mallas han ido ganando auge. Los Métodos sin Malla (MSM) y el Método de Partículas - Método de los Elementos Discretos (MED) - son centro en la actualidad de crecientes investigaciones y es fácil encontrar artículos muy recientes vinculados con su desarrollo o aplicación en diversas ramas [14,15,21,27,28,34].

Las simulaciones que se realizan tienden a ser cada vez más realistas, lo cual se traduce, específicamente en el caso de los Métodos sin Malla y Métodos de Partículas, en un incremento significativo de los puntos y las partículas empleadas respectivamente. En los casos del MED y del Método de Partículas permiten realizar un análisis a nivel microestructural de los materiales, obteniéndose como resultado un volumen considerable de datos lo cual dificulta el post-procesamiento y estudio de los resultados. Una vez realizado el proceso de análisis o simulación, los resultados son presentados como un conjunto de valores numéricos asociados a localizaciones discretas esparcidas por todo el dominio de estudio y que no tienen asociación estructural dada la característica que tienen estos métodos de carecer del uso de mallas durante el proceso de cálculo. Un ejemplo de esto son los resultados mostrados en [8] donde se aplica el método de puntos finitos para mecánica de fluidos utilizando conjuntos de más de 500.000 partículas. Otro ejemplo que se puede mencionar es [22] donde se reportan empaquetamientos en la fase inicial del DEM de más de un millón de partículas.

Un proceso de generación de mallas con esa cantidad de puntos -suponiendo que se considere el centro geométrico de las partículas- sería bastante costoso.

Uno de los aspectos importantes a considerar en el ámbito de la visualización científica es que las imágenes son dibujadas sobre una matriz de dimensiones finitas compuestas de píxeles que representan un único color. Cuando a partir de los puntos esparcidos se realiza la triangulación correspondiente para obtener la malla, normalmente a cada triángulo corresponde un conjunto determinado de píxeles en la imagen resultante. Sin embargo, cuando se cuenta con una cantidad relativamente grande de puntos, al realizar la triangulación va-

^{*} *scattered data* en ingles

rios triángulos corresponden a un mismo píxel. Ejemplificando: si se genera una imagen con resolución de 500×500 píxeles a partir de un conjunto de datos conformado por dos millones de puntos, entonces se van a tener 250 mil píxeles, que son insuficientes para representar ni siquiera, uno a uno los puntos del conjunto de datos. Un ejemplo real de esta situación es descrito en el apartado “Análisis de los resultados” de este trabajo. En estos casos, la utilización de mallas para visualización no es la solución más óptima. Este problema ha sido y continúa siendo tratado y se pueden encontrar varios trabajos relacionados [25,29,31] aunque estos están enfocados fundamentalmente a la visualización de superficies. Este es un aspecto interesante a considerar en el caso de otros métodos relacionados como el PFEM (*Particle Finite Element Method*) [24] y otros métodos que utilizan las mallas durante la simulación en aquellos casos en los que la malla empleada alcanza el orden de los millones de elementos.

Todos los aspectos planteados en esta sección hacen que se considere útil, si no necesaria, la utilización de un método para realizar interpolaciones a partir de nubes de puntos fundamentalmente cuando se consideran volúmenes significativos de información.

INTERPOLACIÓN DE DATOS ESPARCIDOS

En varias áreas científicas es frecuente el siguiente problema: se tiene un conjunto de posiciones/puntos independientes en los cuales se conoce el valor de cierta magnitud, y se desea encontrar una función que permita deducir el valor de la magnitud en nuevos puntos y de esta forma estudiar el comportamiento del proceso. Esto es: se quiere encontrar una función F_a que ajuste “adecuadamente” los datos que se tienen. Existen básicamente dos enfoques que abordan este problema. En el primero de ellos F_a mantiene exactamente los valores iniciales en los puntos conocidos, dándole el nombre de función interpolante. El segundo enfoque se basa en obtener F_a de forma tal que la distancia de los puntos conocidos a ella sea mínima, y en este caso se le da el nombre de función aproximante. En este trabajo se va a abordar únicamente el enfoque de la función interpoladora. En el caso de las funciones aproximantes pueden encontrarse aspectos esenciales relacionados con ellas en [1,7,9,19].

Realizando una definición más precisa del problema: sean $x_i \in \mathbb{R}^s; i = 1..n$ los puntos esparcidos de manera aleatoria y $f(x_i) \in \mathbb{R}$ el valor asociado a cada uno de ellos. Entonces la definición formal del problema es la siguiente:

Problema 1 (*Interpolación de datos esparcidos*): Dada la dupla $(x_i, f(x_i)), i = 1..n$ con $\mathbb{X} = \{x_1, x_2, \dots, x_n\}, \mathbb{X} \subset \mathbb{R}^s$ y $f(x_i) \in \mathbb{R}$ encontrar la función continua $F_a(x) : \mathbb{R}^s \rightarrow \mathbb{R}$ tal que $F_a(x_i) = f(x_i)$. **

Asumir que F_a es una combinación lineal de un conjunto de funciones básicas es una manera muy común y conveniente de resolver este problema, *i.e.*,

$$F_a(x) = \sum_{k=1}^n c_k F_k(x), \quad x \in \mathbb{R}^s. \quad (1)$$

Esto, dada la condición $F_a(x_i) = f(x_i)$, se convierte en un sistema de ecuaciones lineales de la forma

$$Ac = y, \quad (2)$$

** Aunque en este artículo se hace alusión únicamente a funciones de tipo escalar, los elementos que se formulan pueden ser fácilmente generalizados a funciones de tipo vectorial.

donde cada uno de los elementos de la matriz de interpolación A tiene la forma $A_{jk} = F_k(x_j)$, $j, k = 1..n$; $c = [c_1, c_2, \dots, c_n]^T$; $yy = [f(x_1), f(x_2), \dots, f(x_n)]^T$. De aquí que el **Problema 1** va a tener una única solución si y sólo si la matriz A es no singular. Lo anteriormente descrito es tratado con mayor detalle en [9], incluyendo un análisis adicional para el caso de la multivariabilidad.

ANTECEDENTES Y ESTADO DEL ARTE

Por la gran frecuencia con la que se encuentra el problema de la interpolación de datos esparcidos, es posible hallar varios trabajos relacionados con su solución [4,20,23,26], ya no sólo enfocados al problema de la visualización, sino en un ámbito más general. De aquí que existan un gran número de técnicas que son bien conocidas en la actualidad y que dan solución al **Problema 1** de las cuales se pueden encontrar varias referencias en la bibliografía.

Existen métodos basados en la inversa de la distancia entre los puntos. La idea original fue dada por Shepard [32] y tiene un enfoque global, lo que la hace demasiado ineficiente. Sin embargo varias modificaciones han sido realizadas, tanto para mejorar la eficiencia computacional para lo cual la idea inicial se ha llevado a un enfoque local, como para incrementar la exactitud de la función interpoladora [5,11].

Por otra parte existen métodos que se construyen a partir de la vecindad natural entre los datos. Estos se basan de alguna manera en la construcción del diagrama de Voronoi correspondiente a los puntos que conforman los datos iniciales [33,4]. La principal dificultad radica en la lentitud de estos métodos ante la presencia de grandes volúmenes de datos, especialmente por la necesidad de obtener el diagrama de Voronoi. Sin embargo en [26] se obtienen buenos resultados en este sentido.

Una familia de técnicas que ha venido ganando en importancia dado el auge que han alcanzado los Métodos sin Mallas y el MED o Método de Partículas, es la interpolación mediante el uso de funciones de base radial, ya que no requieren de una estructuración de los datos, como en el caso de las interpolaciones basadas en el diagrama de Voronoi.

INTERPOLACIÓN USANDO FUNCIONES DE BASE RADIAL(FBR)

La técnica de interpolación de datos esparcidos usando FBR^{***} consiste en la obtención de la combinación lineal de un conjunto finito de funciones de la forma $\phi(\|\cdot\|)$, donde $\|\cdot\|$ generalmente hace alusión a la distancia Euclideana, y que son radialmente simétricas. Una de sus ventajas radica en la independencia que poseen de la dimensionalidad del problema, ya que utilizan mayormente la norma Euclideana, lo que las hace fácilmente extensibles a cualquier dimensión. Elementos relacionados con la teoría de las FBR pueden encontrarse en [9,10,16,20].

Se define entonces de manera formal las funciones radiales:

Definición 1 (Funciones Radiales): Una función $\phi : \mathbb{R}^s \rightarrow \mathbb{R}$ se dice que es radial si existe una función univariable $\varphi : [0, +\infty) \rightarrow \mathbb{R}$ tal que $\phi(x) = \varphi(\|x\|)$ con $\|\cdot\|$ una norma cualquiera -generalmente la norma Euclideana-.

Si se aplica esta definición y se sustituye en (1) se obtiene la formulación de la solución al **Problema 1** usando FBR:

$$F_a(x) = \sum_{k=1}^n c_k \phi(\|x - x_k\|), \quad x \in \mathbb{R}^s. \quad (3)$$

^{***} RBF(Radial Basis Function) en inglés.

Existen varios tipos bien conocidos de funciones radiales como son la Gaussiana $\phi(r) = e^{-\alpha r}$, *thin-plate splines* $\phi(r) = r^2 \log(r)$ y las multicuádricas $\phi(r) = \sqrt{r^2 + c^2}$ para ejemplificar. Si se analizan las dos últimas funciones puestas como ejemplo, a medida que el radio aumenta, aumenta también el valor que ellas devuelven, por tanto, son funciones de influencia global en el conjunto de datos esparcidos. Este comportamiento se traduce en que la matriz $A(2)$ sea una matriz muy densa, lo cual implica que ante la adición de un nuevo punto es necesario recalcular todos los coeficientes. Por su parte, la forma Gaussiana tiende a cero cuando se incrementa el radio, haciendo que la influencia en un punto x de los elementos $(x_i, f(x_i))$ sea cero si $\|x - x_i\| < r_L$, donde r_L es tal que $e^{-\alpha r_L} < \varepsilon$, para ε suficientemente pequeño. Esta característica de “localidad” da como resultado una matriz A de banda y dispersa, por lo que adicionar nuevos puntos solo afectaría un pequeño conjunto de coeficientes. Por estas razones, el enfoque local es computacionalmente menos costoso, ya que solo es necesario considerar los puntos pertenecientes a una vecindad y no a la totalidad de ellos.

Algunos ejemplos muy interesantes de funciones radiales con influencia local son las presentadas por Wendland [35]. Estas funciones tienen la forma

$$\phi(r) = \begin{cases} p(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases}$$

donde $p(r)$ es uno de los polinomios dados por Wendland, y son fácilmente escalables.

Analizando de modo general las funciones radiales de alcance local: sea $\phi(r)$ la función radial de alcance local que se va a utilizar para construir la función de interpolación de base radial, entonces $(x_i, f(x_i))$ influye únicamente en los puntos que se encuentren ubicados a una distancia $L \leq r_L$, donde $\phi(r > r_L)$ es cero o despreciable. Esto es equivalente a decir: sea $x \notin \mathbb{X}$ entonces $\mathbb{I}_x = (x_i; x_i \in \mathbb{X}, \|x - x_i\| \leq r_L)$ es el conjunto de puntos que ejercen determinada influencia sobre x . Localmente, el problema sería resuelto hallando la solución a (1), pero ahora únicamente considerando los puntos pertenecientes a \mathbb{I}_x . La obtención de \mathbb{I}_x es un proceso computacionalmente costoso valorando que se trata de un volumen considerable de datos esparcidos sin ninguna relación previa, más allá de su ubicación geométrica. La utilización de estructuras de datos que garanticen un almacenamiento adecuado y que faciliten las consultas de vecindad se hace necesaria para agilizar computacionalmente los cálculos necesarios. En este trabajo se hace uso de los árboles de cubrimiento con ese objetivo.

ÁRBOLES DE CUBRIMIENTO

El problema de la búsqueda del vecino más cercano es hoy de mucho interés y particular relevancia en una gran diversidad de disciplinas científicas. Este es un problema de una gran complejidad computacional, $\Omega(n)$, lo que unido a que en la mayoría de los casos el universo de búsqueda es extenso -en este caso lo es- lo convierte en un proceso lento y computacionalmente costoso.

Con el objetivo de mejorar los tiempos de búsqueda son utilizadas estructuras de datos. Existe un gran número de estructuras que permiten el almacenamiento de los datos espaciales vinculadas con la búsqueda del vecino más cercano. Entre las más conocidas podemos mencionar el *kd-tree*, el *vp-tree*, el *R tree* y algunas variantes importantes de este, el *cuadtree* para 2D y el *octree* que es su variante para tres dimensiones, entre varias otras posibilidades [2,6,12,30,36]. Estas estructuras presentan tiempos que crecen logarítmicamente de acuerdo a la cantidad total de elementos para encontrar el vecino más cercano, lo cual teóricamente resulta en comportamientos muy similares a la hora de resolver el problema. No existe en la actualidad criterio exacto para darle a una de ellas la primacía en cuanto

a la eficiencia con que lo hacen. La selección y aplicación de estas está sujeta al problema específico en el que van a ser empleadas. Otro criterio interesante que puede ser de peso, es el de los requerimientos de memoria que las puede hacer descartables en problemas donde se requiere almacenar cantidades importantes de datos.

Existen varias derivaciones del problema del vecino más cercano entre las que se pueden mencionar la aproximación del vecino más cercano y la búsqueda de los vecinos en un rango. Esta última fundamentalmente se realiza en rangos ortogonales donde estructuras como el *kd-tree*, el *octree/cuadtrees*, el *R tree* y el *range tree* presentan buenos resultados por la forma que tienen de particionar el espacio -realizan particionamientos utilizando hiperplanos o espacios ortogonales-. Especialmente en este caso el *range tree* presenta ventajas en los tiempos de respuesta, sin embargo requiere de una utilización mayor de memoria. También es importante señalar que un crecimiento en la dimensión en la que se trabajan los datos -2D a 3D- implican un crecimiento en la complejidad de las consultas de rango, pues los test de interioridad en los nodos requieren de una mayor cantidad de comprobaciones dado el incremento de los elementos necesarios para definir el espacio que estos ocupan. Estos incrementos en los tiempos ya han sido estudiados y en [6] se muestra qué sucede en estos casos para el *range tree* y el *kd-tree*.

Uno de los problemas de búsqueda en un rango es cuando se quieren encontrar los vecinos en un radio determinado. En este caso se trata de un rango circular que, independientemente de la dimensionalidad que se trate, está definido únicamente por el centro y el radio. Este problema puede ser igualmente abordado con las estructuras mencionadas anteriormente, pero su naturaleza no se encuentra representada realmente en ellas debido a la subdivisiones ortogonales que estas realizan del espacio; además del innecesario incremento de complejidad a medida que aumenta la dimensión en la que se aborda el problema.

El *cover tree* (árbol de cubrimiento) [3,18] es una estructura de datos relativamente nueva. Se basa en la descomposición jerárquica del espacio subdividiéndolo en elementos circulares que se van anidando de acuerdo a determinadas características que se enuncian más adelante. Ha sido empleado obteniendo buenos resultados en la búsqueda del vecino más cercano resultando de su aplicación tiempos similares a los obtenidos tras la aplicación de las estructuras mencionadas anteriormente. La forma circular de sus nodos y el mecanismo de anidación y cubrimiento que emplea, lo hace un fuerte candidato para la realización de consultas de rango circular ya que, además de tener una correspondencia entre la geometría de los nodos y la del rango de consulta, la complejidad de las búsquedas no se incrementa con la dimensión en la que se encuentra el problema.

El árbol de cubrimiento es un árbol nivelado donde cada nivel es “cubierto” por el nivel inferior. Cada nivel es indexado con un número entero el cual decrece a medida que se desciende por el árbol. Independientemente de la dimensión el espacio utilizado es $O(n)$. Cada punto en el árbol puede estar asociado con múltiples nodos, pero se requiere que cada punto aparezca a lo sumo una vez en cada nivel.

El árbol de cubrimiento tiene que cumplir las siguientes propiedades [3,18]:

1. $C_i \subset C_{i-1}$. Esto implica que cuando el punto p aparezca por vez primera entonces cada nivel inferior del árbol contiene al nodo asociado con p (Anidación). (Figura 1a)
2. Para todo $p \in C_{i-1}$ existe $q \in C_i$ tal que $d(p, q) \leq 2^i$ y el nodo en el nivel i asociado con q es el padre del nodo en el nivel $i-1$ asociado con p . (Cubrimiento). (Figura 1b)
3. Para todo punto $p, q \in C_i$, tal que $p \neq q$, $d(p, q) > 2^i$. (Separación). (Figura 1c)

Donde C_i es el conjunto de puntos de S que pertenecen al nivel i y $d(p, q)$ la distancia Euclídeana entre p y q .

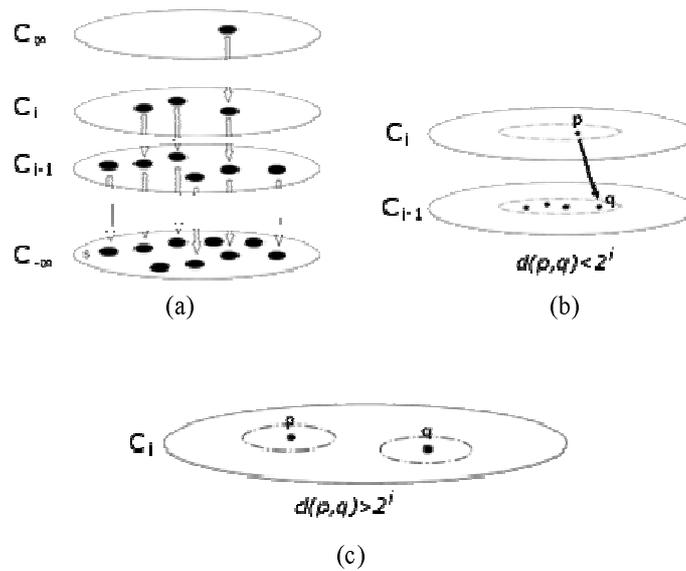


Figura 1. Propiedades de los árboles de cubrimiento. a) Anidación, b) cubrimiento, c) separación

El algoritmo de creación del árbol es sencillo en lo que se refiere a una representación implícita ya que se tendrían infinitos niveles donde C_∞ contiene el punto de S asociado a la raíz del árbol y $C_{-\infty}$ contiene todos los puntos de S .

Para insertar un nuevo nodo se recorre el árbol a partir de la raíz hasta encontrar la posición que cumpla con las propiedades del árbol de cubrimiento (anidación, cubrimiento y separación). La complejidad de este procedimiento es $O(\log(n))$ [18] (Algoritmo 1).

Algoritmo 1: Procedimiento para insertar un punto cualquiera en el árbol de cubrimiento.

```

Insertar( punto  $p$ , conjunto de nodos  $Q_i$ , nivel  $i$ );
 $Q = \{ \text{Hijos}(q): q \in Q_i \}$ ;
if  $d(p, q) > 2^i$  then
    retornar verdadero;
end
else
     $Q_{i-1} = \{q \in Q_i : d(p, q) \leq 2^i\}$ ;
    encontrado = Insertar ( $p, Q_{i-1}, i - 1$ ) ;
    if encontrado and  $d(p, q) < 2^i$  then
        escoger  $q \in Q_i : d(p, q) \leq 2^i$ ;
        insertar  $p$  como hijo de  $q$ ;
        retornar falso;
    end
    else
        retornar falso;
    end
end

```

El algoritmo de búsqueda del vecino más cercano utilizando el árbol de cubrimiento es también sencillo. La búsqueda comienza en la raíz del árbol y utilizando el principio de ramas y cotas se van obteniendo los posibles nodos a elegir. Un nodo es elegible si cumple que $d(p, q) \leq d(p, Q) + 2^i$ siendo p el punto base de la consulta, q un posible vecino o nodo a elegir y Q el conjunto de todos los posibles vecinos. Se define $d(p, Q)$ como la menor de las distancias de p a todos los puntos de Q . (Algoritmo 2).

Las modificaciones necesarias al Algoritmo 2 para realizar la búsqueda de los vecinos en un radio r son mínimas. Estas se muestran en el Algoritmo 3.

Algoritmo 2: Procedimiento para buscar el vecino más cercano con el árbol de cubrimiento.

```

BuscarVMC(punto  $p$ , árbol de cubrimiento  $t$ );
 $Q_\infty = C_\infty$ ;
// $C_\infty$  es la raíz de  $t$ ;
for  $i = \infty$  downto  $-\infty$  do
     $Q = \text{Hijos}(q) : q \in Q_i$ ;
     $Q_{i-1} = \{q \in Q_i : d(p, q) \leq d(p, Q) + 2^i\}$ ;
end
retornar  $\text{argmin}_{q \in Q_{-\infty}} d(p, q)$ ;

```

Algoritmo 3: Procedimiento para buscar los vecinos en un radio con el árbol de cubrimiento.

```

VecinosEnRadio(punto  $p$ , radio  $r$ , árbol de cubrimiento  $t$ );
 $Q_\infty = C_\infty$ ;
// $C_\infty$  es la raíz de  $t$ ;
for  $i = \infty$  downto  $-\infty$  do
     $Q = \text{Hijos}(q) : q \in Q_i$ ;
     $Q_{i-1} = \{q \in Q_i : d(p, q) \leq r + 2^i\}$ ;
end
retornar  $\{q \in Q_{-\infty} : d(p, q) \leq r\}$ ;

```

PASOS Y ALGORITMOS DEL MÉTODO

Ya descritos los elementos para la implementación de este método, a continuación se describen los pasos necesarios para su aplicación:

1. Creación del árbol de cubrimiento. La creación del árbol de cubrimiento se realiza mediante la sucesiva inserción de cada uno de los puntos pertenecientes a los datos en un árbol inicialmente vacío. La construcción del árbol se realiza al inicio del proceso y no es necesario repetirla ya que independientemente del punto en que se desee interpolar el árbol de cubrimiento es el mismo. (Algoritmo 4)
2. Obtención de los puntos localizados en el radio de influencia. Una vez creado el árbol de cubrimiento, se utilizan las facilidades que este brinda para obtener los vecinos que se encuentran a una distancia máxima determinada de un punto. (Algoritmo 5)
3. Evaluación de la función de interpolación. Se utilizan los puntos encontrados para evaluar la función de interpolación seleccionada y obtener un valor aproximado de la magnitud que se estudia en el nuevo punto. (Algoritmo 5)

La metodología aquí descrita puede ser utilizada independientemente de la finalidad con que se desee emplear el método de interpolación y los nuevos puntos a interpolar serán elegidos de acuerdo a la técnica a aplicar. Las nuevas localizaciones sobre las que se desea interpolar pudieran ser, por ejemplo, tanto los vértices de una malla que se pudiera tener como soporte de visualización o malla de superficie de algún cuerpo tridimensional, o los vértices de las celdas utilizados por las técnicas *marching squares* o *marching cubes* [17] empleados para obtener isolíneas o isosuperficies respectivamente.

Algoritmo 4: Procedimiento para crear el árbol de cubrimiento.

```

CrearArbol(conjunto de puntos P);
A = CrearArbolVacio();
foreach p in P do
    Insertar(p, A, ∞);
end
retornar A;

```

Algoritmo 5: Procedimiento para obtener el valor en un nuevo punto.

```

Interpolar(punto p, árbol de cubrimiento A);
V = VecinosEnRadio(p, A, radio);
v = FunciónInterpoladora(p, V);
retornar v;

```

CASO DE ESTUDIO: INTERPOLACIÓN USANDO EL MÉTODO DE SHEPARD AL ALCANCE LOCAL

El método de Shepard [32], como se mencionó anteriormente, realiza la interpolación utilizando la inversa de las distancias entre los puntos. Originalmente, este es un método de alcance global, ya que utiliza todos los puntos comprendidos en los datos, sin embargo, en [5,32] se explica una modificación del mismo que considera únicamente los puntos que se encuentren, como máximo, a una distancia determinada del punto donde se desea interpolar, lo cual le da a este método un enfoque local. Dicho enfoque es el empleado en este caso de estudio y a continuación se dan detalles de su formulación basado en lo expuesto en [5,32].

En el caso de la interpolación mediante el Método de Shepard la función interpolante tiene la forma:

$$F(x) = \sum_{i=1}^n w_i(x) f(x_i),$$

donde $w_i(x)$ es la función que determina el peso -grado de influencia- que tiene x_i sobre x . Esta función se obtiene como:

$$w_i = \frac{\psi_i(x)}{\sum_{j=1}^n \psi_j(x)}$$

con $\psi_i(x) = \frac{1}{d_i^2(x)}$, $d_i = \|x - x_i\|$.

La modificación necesaria para convertir esta función en una de alcance local es sencilla. Para ello basta con modificar la función de peso, haciendo 0 el peso de aquellos puntos que se encuentran más allá de una distancia r_w . Esto es:

$$\psi_i(x) = \frac{1}{d_i^2(x)} \left(1 - \frac{d_i^2(x)}{r_w} \right)^2,$$

esta modificación convierte el problema en la búsqueda de los vecinos que se encuentran dentro de un radio r_w del punto de interpolación.

Para determinar los vecinos que se encuentran a una distancia máxima r_w del punto donde se desea interpolar se realizó la construcción del árbol de cubrimiento, como se describió anteriormente.

ANÁLISIS DE LOS RESULTADOS

El algoritmo fue empleado para visualizar nubes con distintas cantidades de puntos. En la Figura 2 se pueden apreciar cuatro de los conjuntos de datos visualizados. En estas figuras se observa la interpolación realizada y los puntos negros representan los datos del conjunto de entrada.



Figura 2. Aplicación de la técnica a varios conjuntos de datos. Los puntos en negro reflejan la posición de los puntos que se tienen como entrada. Las cantidades de puntos de cada conjunto son a) 195, b) 2791, c) 41987 y d) 181304

El primer aspecto fundamental a señalar es el resultado obtenido en la Figura 2a), donde es notable la formación de pequeñas circunferencias que tienen como centro los puntos iniciales y no se obtiene una transición de color lo suficientemente suave como para interpretar correctamente el significado de los datos. La formación de estas circunferencias

es debido a que el valor de la derivada de la función interpolante utilizada en los puntos conocidos es cero. Este problema se aborda en [5] y se muestra una solución al mismo, sin embargo, esta traería el incremento del costo computacional del método. Dicha modificación es innecesaria ya que este trabajo se enfoca a grandes cantidades de puntos y en estos casos a los efectos visuales las transiciones de valores ocurren lo suficientemente lento como para no notar las circunferencias, lo cual se puede mostrar en la Figura 2b) y c).

El otro punto importante a resaltar es como los puntos negros lógicamente van ocupando todo el espacio de la imagen -cada punto negro ocupa solamente un píxel de la imagen y corresponde a la posición de un punto del conjunto original que se tiene como entrada- a medida que se van incrementando los datos, hasta llegar al caso extremo de la Figura 2d) donde todos los puntos de la imagen son negros. En cada una de las imágenes de la Figura 2 existen 65290 píxeles empleados para representar el dominio de estudio -sin incluir los píxeles blancos que rodean el cuerpo-, en el caso específico de d) se están representando 181204 puntos, casi a 3 puntos por píxel si se considera que están distribuidos uniformemente. Por esta razón la imagen queda completamente cubierta con el color negro.

La gráfica de la Figura 3 muestra el comportamiento de los tiempos de interpolación de acuerdo a la cantidad de puntos empleados. En el eje de las y se tiene el tiempo promedio necesario para interpolar el valor en un punto del dominio, y por las x se tiene la cantidad promedio de puntos que influyeron en una interpolación.

Si se analiza la gráfica de la Figura 3 es perceptible que a pesar del incremento notable de la cantidad de puntos utilizados en cada uno de los casos, no crece de modo acelerado la distancia entre las funciones, y por tanto, tampoco crece en gran medida el tiempo necesario para interpolar un punto, esto, bajo condiciones similares, dígame una cantidad similar del promedio de vecinos que se considera debido al radio de influencia. En este sentido puede observarse la gráfica de la Figura 4 la cual fue elaborada a partir de los datos de la Figura 3.

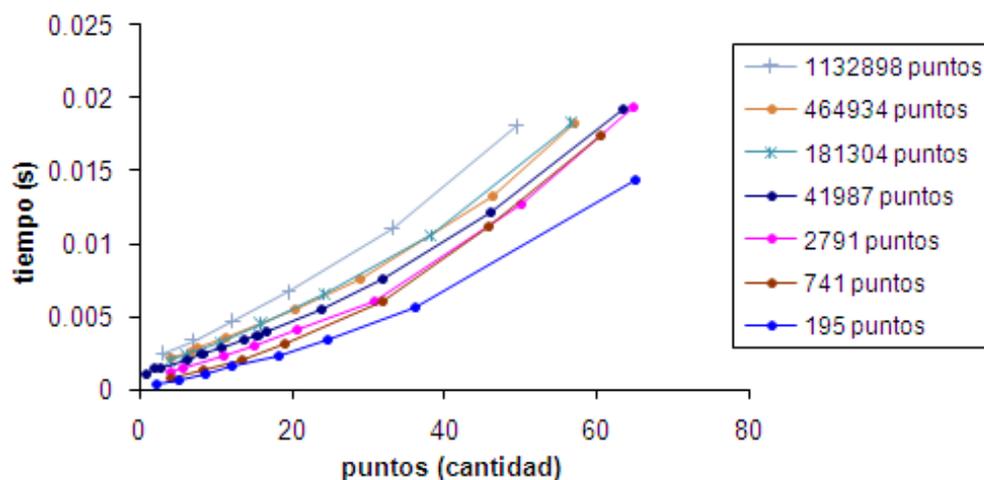


Figura 3. Tiempos promedios de interpolación para un punto

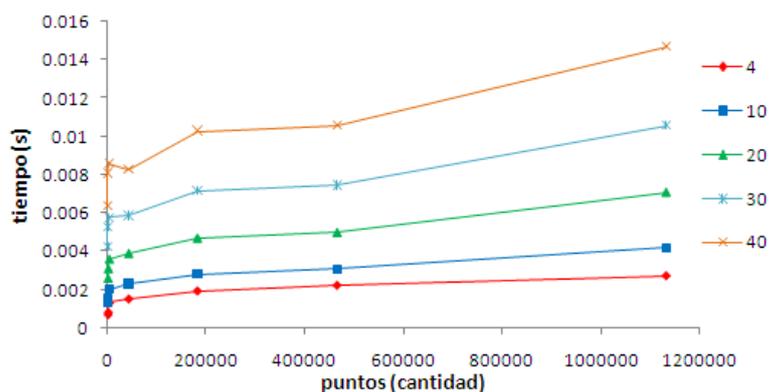


Figura 4. Tiempos de interpolación para un punto considerando para cada conjunto de datos que la cantidad de puntos que se encuentran dentro del radio de influencia son 4, 10, 20, 30 y 40, como se muestra en la gráfica



Figura 5. Aparición de discontinuidades -puntos negros- en la generación de una imagen debido a la selección de un radio demasiado pequeño

En ella cada línea refleja el comportamiento del tiempo necesario para interpolar en un punto a partir de los distintos conjuntos de datos si se consideran la misma cantidad de puntos dentro del radio de influencia. Como puede apreciarse, a pesar de la diferencia en las cantidades de puntos existentes entre los conjuntos de datos, el crecimiento del tiempo necesario para interpolar puede considerarse como lento. Por estas razones, se puede considerar el uso de árboles de cubrimiento como una estructura adecuada para acelerar el problema de la búsqueda de los vecinos en un radio determinado, enfocado a la interpolación de datos esparcidos cuando se utilizan FBR de alcance local. En consecuencia, la utilización de este algoritmo ante resultados compuestos por grandes volúmenes de datos provenientes del MED o Método de Partículas, Métodos sin Malla u otros como el PFEM es factible, ya que aligera el costo computacional y el tratamiento de estas grandes cantidades de puntos. Este método agiliza el post procesamiento de resultados donde las nubes de puntos obtenidas son muy densas y los valores presentan cambios significativos en vecindades no lejanas, haciendo de la interpolación local la mejor opción para la eficiencia y la exactitud de la imagen resultante.

El radio de influencia debe ser seleccionado de manera correcta de acuerdo con la densidad de la nube de puntos. La selección de un radio demasiado pequeño puede traer como resultado la aparición de discontinuidades no deseadas en la imagen final (Figura 5). Por otra parte, mientras mayor sea el radio seleccionado mayor será la cantidad de vecinos que influyen sobre un punto, y por tanto mayores los tiempos de interpolación, sin embargo, en nubes de puntos suficientemente densas, la diferencia entre las imágenes generadas para un radio mayor o menor que otro no son generalmente significativas.

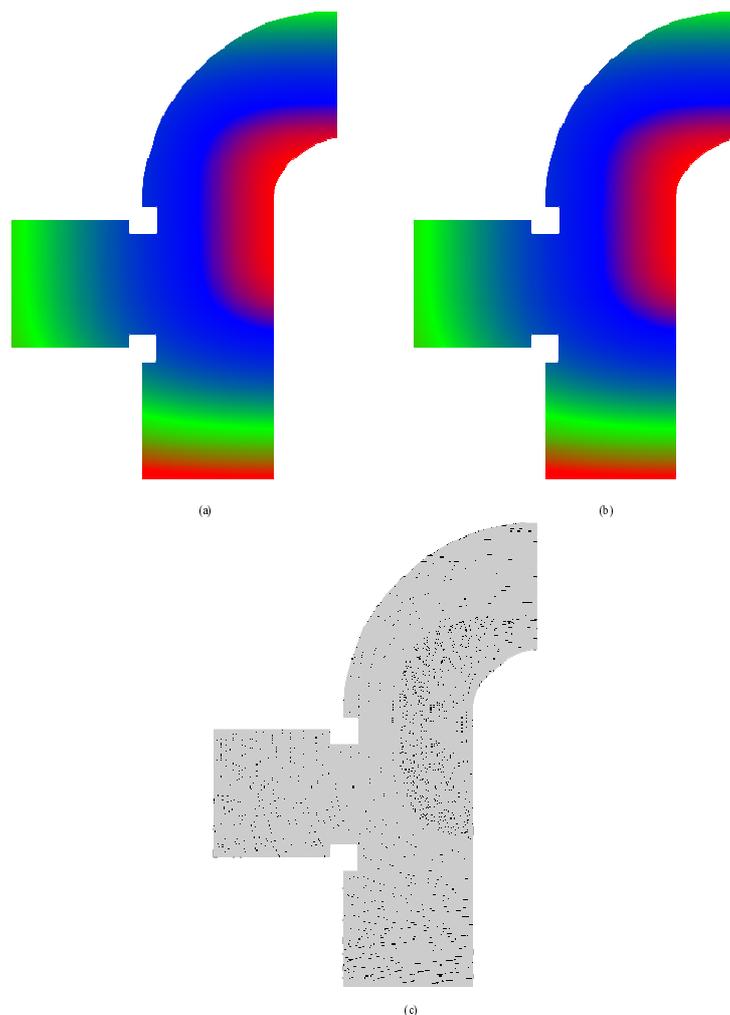


Figura 6. Diferencia entre dos imágenes generadas interpolando a partir de una nube de 41987 puntos. a) Con un radio de 0.03u, b) Con un radio de 0.1u, c) Diferencia entre ambas imágenes; los puntos negros indican los píxeles donde existía diferencia de colores

En la Figura 6 se puede apreciar la diferencia entre dos imágenes generadas para la misma nube de puntos, en este caso 41987 puntos. El radio considerado para la Figura 6a) fue de 0.03u, lo cual trajo consigo que el promedio de los puntos que se consideraron y el tiempo promedio para cada interpolación fueran 6.21 y 0.0021s respectivamente. Para la Figura 6b) el radio fue de 0.1u, y el promedio de los puntos que se consideraron y el tiempo promedio para cada interpolación fueran 63.45 y 0.0192s respectivamente. Puede notarse en la Figura 6c) que las diferencias entre ambas imágenes -puntos negros- no son considerables,

además de ser prácticamente imperceptibles a simple vista. De cualquier forma, hasta el momento, esta es una decisión del usuario, dependiendo de los resultados que se deseen resaltar y de la suavidad en los cambios de valores en puntos correspondientes a vecindades cercanas. Si se trata de eficiencia computacional, un radio pequeño sería lo ideal, pero el comportamiento de los valores en la nube de puntos es muy importante también para tener resultados más exactos.

Para obtener las visualizaciones presentadas en esta sección se realizó la interpolación directa en cada uno de los puntos del dominio que se hicieron corresponder a los píxeles de la imagen. Si bien esta es una técnica poco eficiente, el objetivo fue tener una mayor cantidad de muestreos para cada uno de los juegos de datos empleados y de esta manera poder valorar con más fundamento los tiempos de respuesta. Además, de esta forma es posible obtener imágenes que aproximan de una manera más exacta el comportamiento real de la función interpolante que se está utilizando y permiten, por tanto, estudiar de una mejor manera su comportamiento. Estos dos aspectos mencionados son objetivo fundamental de este trabajo pues constituyen la esencia para validar el método planteado.

La combinación del método aquí expuesto con técnicas conocidas que permitan disminuir la cantidad de muestreos realizados para la obtención de una imagen, como pudiera ser una malla de soporte para el proceso de visualización, seguramente daría buenos resultados. Sin embargo, introduciría nuevas problemáticas que deben ser tratadas con cuidado como son la exactitud de la imagen resultante y selección de niveles de detalle.

CONCLUSIONES

En el presente trabajo se desarrolla un método que utiliza las FBR de alcance local unido a los árboles de cubrimiento. El método muestra un comportamiento deseable en la medida que la cantidad de puntos de entrada aumenta, lo cual fue comprobado utilizando la función interpolante de Shepard en su modificación a función local.

Una de las ventajas de este método está en la independencia de la dimensión en la que se trabaja, ya que tanto las FBR como los árboles de cubrimiento dependen únicamente de la distancia entre dos puntos, sin importar la dimensión en que estos se encuentren, de aquí que la complejidad computacional en 2 o 3 dimensiones es la misma. Esto no sucede así en los métodos que emplean la vecindad natural de los puntos ya que en ellos el costo computacional se incrementa con la dimensión en la que se encuentre el problema abordado. También se destacan las deficiencias de la función en el caso que se trate de nubes de puntos poco densas dada la característica de que su primera derivada evaluada en los puntos de entrada es cero.

Además se realizó un estudio sobre la influencia en el tiempo de interpolación de la cantidad de puntos dentro del radio indicado, considerando la cantidad total de puntos en la nube. Se hizo alusión a la necesidad de la correcta selección de un radio para obtener una imagen adecuada, brindando algunos criterios relacionados con la selección del radio indicado.

El trabajo deja abierta nuevas investigaciones sobre la selección adecuada de un radio de influencia que dependa de la densidad de la nube de puntos y del comportamiento de los valores. También se deja abierta a investigaciones futuras la combinación de este método con otros que permitan disminuir la cantidad de muestreos para obtener la imagen resultante, como pudiera ser introducir una malla de soporte para la visualización.

REFERENCIAS

- 1 M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C.T. Silva, "Computing and rendering point set surfaces", *IEEE Transactions on Visualization and Computer Graphics*, Vol. **9**, N° 1, pp. 3-15, (2003).
- 2 A. Atramentov, S. M. LaValle, "Efficient nearest neighbor searching for motion planning", In *ICRA*, pp. 632-637, (2002).
- 3 A. Beygelzimer, S. Kakade, J. Langford, "Cover trees for nearest neighbor", In ACM, *ICML '06: Proceedings of the 23rd International Conference on Machine learning*, pp. 97-104, New York, NY, USA, (2006).
- 4 T. Bobach, M. Bertram, G. Umlauf, "Comparison of voronoi based scattered data interpolation schemes", In J.J. Villanueva Editor, *Proceedings of the International Conference on Visualization, Imaging and Image Processing*, pp. 342-349, (2006).
- 5 K.W. Brodlie, M. Rafiq Asim, K. Unsworth, "Constrained visualization using the shepard interpolation family", *Comput. Graph. Forum*, Vol. **24**, N° 4, pp. 809-820, (2005).
- 6 M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, "*Computational Geometry: Algorithms and Applications*", Springer-Verlag, Second Edition, (2000).
- 7 T. K. Dey, Jian Sun, "An adaptive mls surface for reconstruction with guarantees", In *SGP '05: Proceedings of the third Eurographics Symposium on Geometry Processing*, Eurographics Association. p. 43, Aire-la-Ville, Switzerland, Switzerland, (2005).
- 8 S. Idelsohn, E. Ortega, E. Oñate, "A finite point method for adaptive three-dimensional compressible flow calculations", *International Journal for Numerical Methods in Fluids*, Vol. **60**, N° 9, pp. 937-971 ISSN: 0271-2091SFX - UPC, (2009).
- 9 G.E. Fasshauer, "*Meshfree Approximation Methods with MATLAB*", Vol. **6** of *Interdisciplinary Mathematical Sciences*. World Scientific, (2007).
- 10 M.S. Floater, A. Iske, "Multistep scattered data interpolation using compactly supported radial basis functions", *J. Comput. Appl. Math.*, Vol. **73**, N° 1-2, pp. 65-78, (1996).
- 11 R. Franke, G. Nielson, "Smooth interpolation of large sets of scattered data", *International Journal for Numerical Methods in Engineering*, Vol. **15**, pp. 1691-1704, (1980).
- 12 V. Gaede, O. Günther, N. Volker Gaede, N. Oliver Gunther, "Multidimensional access methods" *ACM Computing Surveys*, Vol. **30**, pp. 170- 231, (1997).
- 13 R.S. Gallagher, "*Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis*", Solomon Press, CRC-Press, (1995).
- 14 H. Gómez, I. Colominas, F. Navarrina, M. Casteleiro, "Un nuevo enfoque para el tratamiento de los términos difusivos de la ecuación de convección-difusión en el Método de Galerkin discontinuo", *Rev. Internac. Mét. Num. Cál. Dis. Ing.*, Vol. **23**, pp. 343-362, (2007).
- 15 R. Gutiérrez, U. Lugrís, J. Cuadrado, L.E. Romera, "Modelización de sistemas multicuerpo flexibles para análisis dinámico con no linealidad geométrica", *Rev. Internac. Mét. Num. Cál. Dis. Ing.*, Vol. **23**, pp. 159-176, (2007).
- 16 A. Iske, V.I. Arnold, "*Multiresolution Methods in Scattered Data Modelling*", Springer Verlag, (2004).
- 17 C. Johnson, C. Hansen, "*Visualization Handbook*", Academic Press, Inc., Orlando, FL, USA, (2004).
- 18 T. Kollar, "Fast nearest neighbors", Technical report, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, (2006).

- 19 D. Levin, "The approximation power of moving least-squares", *Math. Comput.*, Vol. **67**, N° 224, pp. 1517-1531, (1998).
- 20 G.R. Liu, "*Mesh free methods - moving beyond the finite element method*", Chapter "Point Interpolation Methods", CRC-Press LLC, (2003).
- 21 O.L. Manzoli, B. Shing, "Elementos finitos híbridos com descontinuidades fortes incorporadas", *Rev. Internac. Mét. Num. Cál. Dis. Ing.*, Vol. **22**, N° 4, pp. 393-408, (2006).
- 22 I. Pérez Morales, R. Roselló Valera, Y. Pérez Brito, H. Díaz-Guzmán Casañas, Á. Fuentes Suárez, R. Grau Ávalos, L. Argüelles Cortés, C.A. Recarey Morfa, "Procedure for packing generic particles for the discrete element method" *International Journal for Numerical Methods in Engineering*, Pendiente a aceptación.
- 23 G.M. Nielson, "Scattered data modeling", *IEEE Computer Graphics and Applications*, Vol. **13**, N° 1, pp. 60-70, (1993).
- 24 E. Oñate, S.R. Idelsohn, F. Del Pin, R. Aubry, "The particle finite element method. An overview", *International Journal of Computational Methods*, Vol. **1**, N° 2, pp. 267-307, (2004).
- 25 R. Pajarola, M. Sainz, P. Guidotti, "Object-space point blending and splatting", **IEEE Transactions on Visualization and Computer Graphics**, Vol. **10**, N°5, pp. 598-608, (2004).
- 26 S.W. Park, L. Linsen, O. Kreylos, J.D. Owens, "Discrete sibson interpolation", *IEEE Transactions on Visualization and Computer Graphics*, Member-Bernd Hamann, Vol. **12**, N° 2, pp. 243-253, (2006).
- 27 O. Pelliccioni, M. Cerrolaza, M. Herrera, "Análisis tridimensional de la interacción fluido-estructura en una válvula cardíaca mecánica doble hoja utilizando la ecuación generalizada de iattice boltzmann", *Rev. Internac. Mét. Num. Cál. Dis. Ing.*, Vol. **22**, N° 4, pp. 377-392, (2006).
- 28 F. Perazzo, S. Oller, J. Miquel, E. Oñate, "Avances en el método de puntos finitos para la mecánica de sólidos", *Rev. Internac. Mét. Num. Cál. Dis. Ing.*, Vol. **22**, N° 4, pp. 153-168, (2006).
- 29 S. Rusinkiewicz, M. Levoy, "QSplat: A multiresolution point rendering system for large meshes", In Proceedings of *ACM SIGGRAPH 2000*, pp. 343-352, Julio (2000).
- 30 R. Sagawa, T. Masuda, K. Ikeuchi, "Effective nearest neighbor search for aligning and merging range images", *Int. Conf 3-D Digital Imaging and Modeling*, Oct. (2003).
- 31 M. Sainz, R. Pajarola, "Point-based rendering techniques", *Computers & Graphics*, Vol. **28**, N° 6, pp. 869-879, (2004).
- 32 D. Shepard, "A two-dimensional interpolation function for irregularlyspaced data", In *Proceedings of the 1968 23rd ACM national conference*, pp. 517-524, New York, NY, USA, (1968).
- 33 R. Sibson, "A vector identity for the dirichlet tessellation", *Math. Proc. Cambridge Philosophical Soc.*, Vol. **87**, pp. 151-155, (1980).
- 34 G. Simonetti, A. Cardona, "Métodos sin malla para resolver la ecuación de conducción del calor", *Rev. Internac. Mét. Num. Cál. Dis. Ing.*, Vol. **16**, N° , pp. 33-48, (2000).
- 35 H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree", Technical report, Instituts für Numerische und Angewandte Mathematik, Universität Göttingen, Göttingen, Germany, (1995).
- 36 P.N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces", In *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics., pp. 311-321, Philadelphia, PA, USA, (1993).