Nuevos Métodos de interpolación de grandes volúmenes de datos provenientes de la aplicación de los métodos de partículas o puntos (libres de malla) para su visualización científica

YAIDEL REYES LÓPEZ (*), HEIKEL YERVILLA HERRERA (*), ALCIDES VIAMONTES ESQUIVEL (*)
y CARLOS A. RECAREY MORFA (**)

RESUMEN En el presente trabajo se desarrolla un nuevo método para interpolar grandes volúmenes de datos esparcidos, dirigido principalmente a los resultados de la aplicación de Métodos libres de Malla, Métodos de Puntos 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 para interpolar los valores en un nuevo punto. Esto agiliza la aplicación de técnicas de visualización científicas 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.

NEW METHODS TO INTERPOLATE LARGE VOLUMES OF DATA FROM POINTS OR PARTICLES (MESH-FREE) METHODS APPLICATION FOR ITS SCIENTIFIC VISUALIZATION

ABSTRACT *In the following paper we developed a new method to interpolate large volumes of scattered data, focused mainly on the results of the Meshfree Methods, Points Methods and the Particles Methods application. Through this one, we use local radial basis functions as interpolating functions. We also use cover-tree 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.*

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.

Keywords: Scattered data, Interpolation, Scientific visualization, Covertree, Radial basis functions, Shepard's method, Points Methods, Particles Methods, Mesh free/less Methods.

1. 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 computa-

(*) Lic. 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 1/2 , 54830, Santa Clara,Villa Clara. Cuba.

(**) Dr. Ing. 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 1/2, 54830, Santa Clara, Villa Clara. Cuba.

ción 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 [10] y [12] 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"[1].

Como consecuencia de las limitantes 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 o donde existan grandes deformaciones del modelo de análisis, algunos métodos que no requieren de la generación de mallas han ido ganando auge. Los Métodos sin Malla, el Método de los Elementos Discretos (MED) –Método de Partículas– y el Método de Elementos Finitos de Puntos (PFEM) son centro en la actualidad de crecientes investigaciones para su aplicación en diversas ramas. En los casos del MED y Método de Partículas permiten realizar un análisis a nivel microestructural de los materiales, obteniéndose como resultado un volumen considerable de datos, que dadas las características que poseen, se consideran esparcidos. La cantidad de información presente en los resultados obtenidos tras la aplicación de estos métodos dificulta el proceso de visualización ya que hace lento el post-procesamiento de los datos, y con esto, la generación de las imágenes. Esta es la causa principal del presente trabajo. De modo similar en los casos de los Métodos sin Malla y PFEM, los datos o variables de respuesta a visualizar están referidos a nubes de puntos, de aquí también que sea necesario el desarrollo de técnicas para interpolar datos con estas características.

2. 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, dándosele 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, 6, 7, 14].

Realizando una definición más precisa del problema: Sean $x_i \in R^s$;i = l..n los puntos esparcidos de manera aleatoria y $f(x_i) \in R$ 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 = l..n con $X = \{x_1, x_2,....,x_n\}$, $X \subset R^s$ y $f(x_i) \in R$ encontrar la función continua $F_a(x)$: $R^s \rightarrow R$ tal que $F_a(x_i)=f(x_i)$.[2]

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.,

$$F_a(x) = \sum_{k=1}^{n} c_k F_K(x), x \in R^s \qquad (1)$$

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

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

donde cada uno de los elementos de la matriz de interpolación A tiene la forma $A_{jk} = F_k(x_j)$,j,k = l..n;$c = [c_1,c_2,....,c_n]^T$; y $y = [f(x_1),f(x_2),....,f(x_n)]^T$.

De aquí que el ***Problema 1*** va a tener una única solución si y solo si la matriz A es no singular. Lo anteriormente descrito es tratado con mayor detalle en [7], incluyendo un análisis adicional para el caso de la multivariabilidad.

3. ANTECEDENTES Y ESTADO DEL ARTE

Por la gran frecuencia con la que se encuentra el problema de la interpolación de datos esparcidos, es posible encontrar varios trabajos relacionados con su solución [4, 15, 16, 17], ya no solo 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*** y a las que pueden encontrarse varias referencias en la bibliografía.

Existen métodos basados en la inversa de la distancia entre los puntos. La idea original fue dada por Shepard [19] y tiene un enfoque global, lo que la hace demasiado ineficiente. Sin embargo varias modificaciones han sido realizadas, tanto para mejorar la eficiencia computacional llevando la idea inicial a un enfoque local, como para incrementar la exactitud de la función interpoladora [5, 9].

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 [20, 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 [17] 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, el MED o Método de Partículas y el PFEM, 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.

[1] "*Scattered data*" en inglés.

[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.

4. 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 [7, 8, 11, 15].

Se define entonces de manera formal las funciones radiales:

Definición *1 (Funciones Radiales):* Una función ϕ: $R^s \rightarrow R$ se dice que es radial si existe una función univariable $\varphi:[0,+\infty]\rightarrow R$ tal que $\varphi(x) = \varphi(\|x\|)$ 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:

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

Existen varios tipos bien conocidos de funciones radiales como son la Gausiana $\phi(r) = e^{-ar}$, *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 sea necesario recalcular todos los coeficientes. Por su parte, la forma Gausiana 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 $e^{-ar_L} < \varepsilon$, 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 [21]. Estas funciones tienen la forma:

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

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 $(r_i, f(x_i))$ influye únicamente en los puntos que se encuentren ubicados a una distancia $L \le r_L$, donde $\phi(r > r_L)$ es cero o despreciable. Esto es equivalente a decir: sea $x \notin X$ entonces $I_x = (x_i; x_i \in X, \|x - x_i\| \le r_L)$ 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 I_x. La obtención de 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.

5. 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. Entre las estructuras más conocidas de almacenamiento de datos espaciales vinculadas con la búsqueda del vecino más cercano está el kd-tree, el vp-tree y el cover tree (árbol de cubrimiento) [2, 3, 18, 22]. En este trabajo es utilizado el cover tree por sus facilidades en su estructura a la hora de buscar los vecinos dentro de un radio dado, además de los buenos resultados obtenidos tras su utilización [13].

El árbol de cubrimiento es una estructura jerárquica utilizada fundamentalmente en consultas de rango y en el problema del vecino más cercano. 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 [3,13]:

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

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

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)) [13] (Algoritmo 1).

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 $d(p,q) \le d(p,Q) + 2^i$ 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).

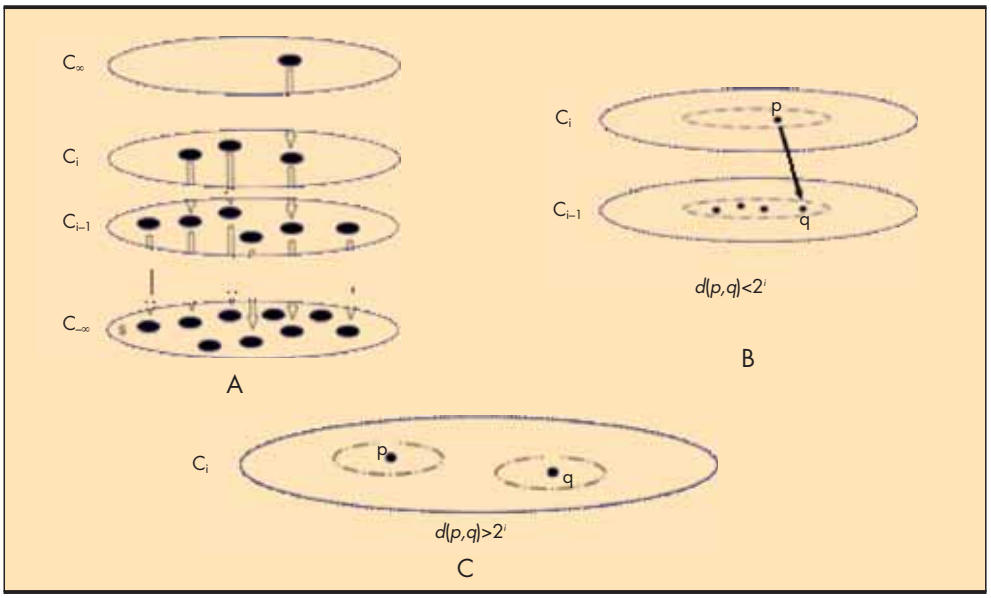

FIGURA 1. Propiedades de los árboles de cubrimiento.
(a) Anidación.
(b) Cubrimiento.
(c) Separación.

```
Insertar (punto p, conjunto de nodos Qi, nivel i);
Q = {Hijos(q): q ∈ Qi}
if d(p,q) > 2^i then
    retornar verdadero;
end
else
    Qi−1 = {q ∈ Qi} : d(p,q) ≤ 2^i};
    encontrado = Insertar (p,Qi−1, i−1);
    if encontrado and d(p,q) > 2^i then
        escoger q ∈ Qi : d(p,q) ≤ 2^i;
        insertar p como hijo de q;
        retornar falso;
    end
    else
        retornar falso;
    end
end
```

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

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

TABLA 3. Algoritmo 3. Procedimiento para crear el árbol de cubrimiento.

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

TABLA 4. Algoritmo 4. Procedimiento para obtener el valor en un nuevo punto.

```
BuscarVMC (punto p, árbol de cubrimiento t);
Q∞ = C∞;
// C∞ es la raíz de t;
for i = ∞ downto − ∞ do
    Q = Hijos (q) : q ∈ Qi;
    Qi−1 = {q ∈ Qi : d(p,q) ≤ d(p,Q) + 2^i};
    retornar argmin_{q ∈Q−inf} d(p,q);
end
```

TABLA 2. Algoritmo 2. Procedimiento para buscar el vecino más cercano en el árbol de cubrimiento.

Una de las principales ramificaciones del problema del vecino más cercano es la búsqueda de los vecinos dentro de un radio r. El cover tree presenta una conveniente notación a la hora de enfrentar dicho problema. Con modificar ligeramente el Algoritmo 2 pueden ser determinados todos los puntos que se encuentran a una distancia máxima r de otro punto cualquiera. Esta propiedad lo hace muy útil y permite su utilización en la interpolación mediante el uso de funciones de base radial de alcance local.

6. 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 3).
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 4).
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 4).

7. CASO DE ESTUDIO: INTERPOLACION USANDO EL MÉTODO DE SHEPARD DE ALCANCE LOCAL

El método de Shepard [19], 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, existe una modificación del mismo que considera únicamente los puntos que se encuentren, como máximo, a una distancia determinada del punto donde se desea interpolar. A continuación se darán detalles de los mismos basado en lo expuesto en [5, 19].

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 sobre x_i. Esta función se obtiene como:

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

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

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.

8. ANÁLISIS DE LOS RESULTADOS

El algoritmo fue empleado para visualizar nubes con distintas cantidades de puntos. La gráfica (Figura 2) 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 2 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, dígase una cantidad similar del promedio de vecinos que se considera debido al radio de influencia. Por tanto, 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, PFEM u otros métodos 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 3). 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

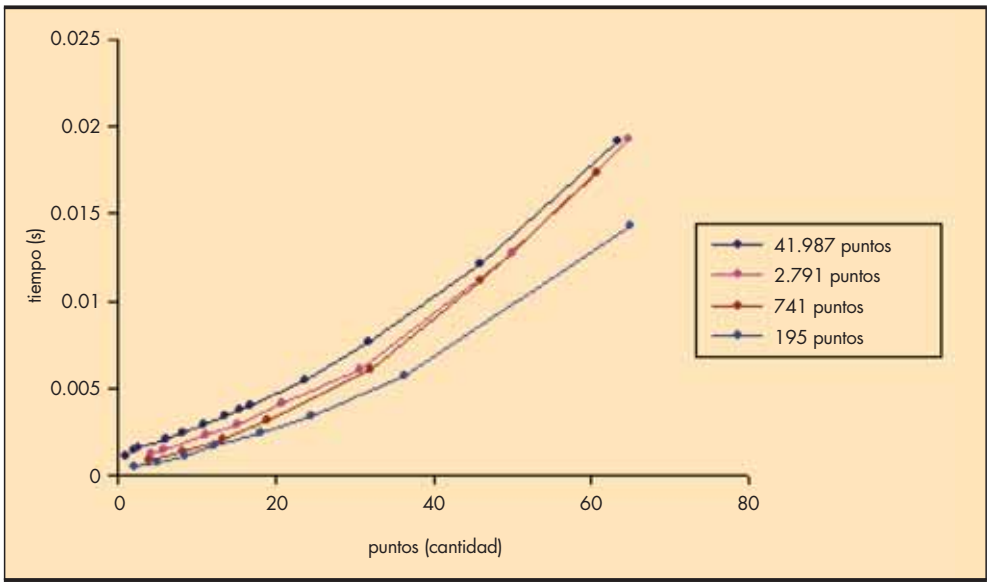

FIGURA 2. Tiempos promedios de interpolación para un punto.

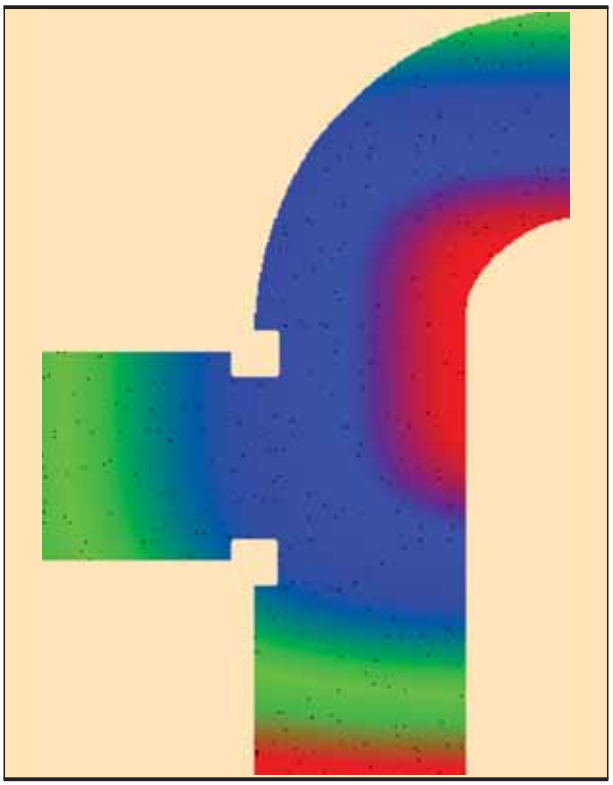

FIGURA 3. Aparición de discontinuidades –puntos negros– en la generación de una imagen debido a la selección de un radio demasiado pequeño.

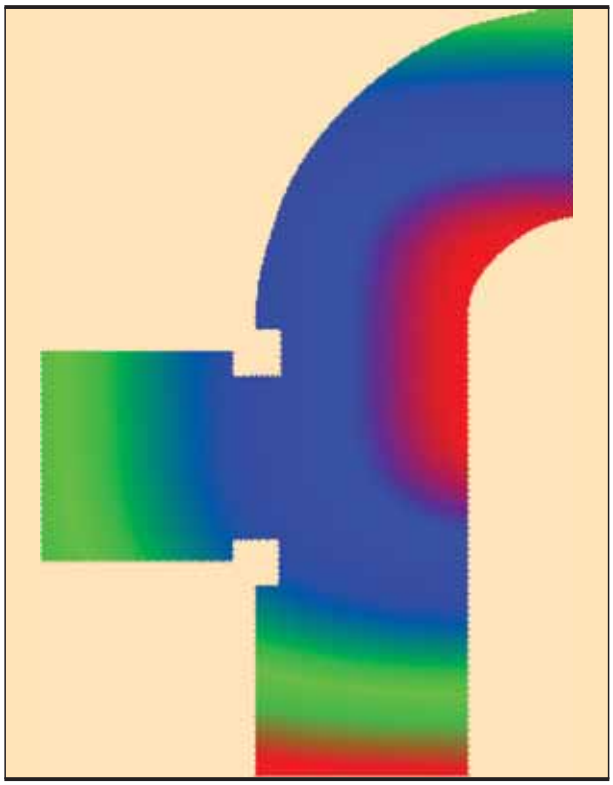

FIGURA 4A. Diferencia entre dos imágenes generadas interpolando a partir de una nube de 41987 puntos con un radio de 0.03u.

significativas. En la Figura 4 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 4 (a) 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 4 (b) 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 4 (c) 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.

9. CONCLUSIONES

En el presente trabajo se crea un nuevo método que utiliza las FBR de alcance local unido a los árboles de cubrimiento. El método es computacionalmente eficiente, y fue comprobado utilizando para ello la función interpolante de Shepard en su modificación a función local. Como una de las ventajas de este método está 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.

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.

10. REFERENCIAS

[1] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. IEEE Transactions on Visualization and Computer Graphics, 09(1):3-15, 2003.

[2] Anna Atramentov and Steven M. LaValle. Effcient nearest neighbor searching for motion planning. In ICRA, pages 632-637, 2002.

[3] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In ICML '06: Proceedings of the 23rd international conference on Machine learning, pages 97-104, New York, NY, USA, 2006. ACM.

[4] Tom Bobach, Martin Bertram, and Georg Umlauf. Comparison of voronoi based scatterd data interpolation schemes. In J.J. Villanueva, editor, Proceedings of the Internationl Conference on Visualization, Imaging and Image Processing, pages 342-349, 2006.

[5] Ken W. Brodlie, Muhammed Rafiq Asim, and Keith Unsworth. Constrained visualization using the shepard inter-

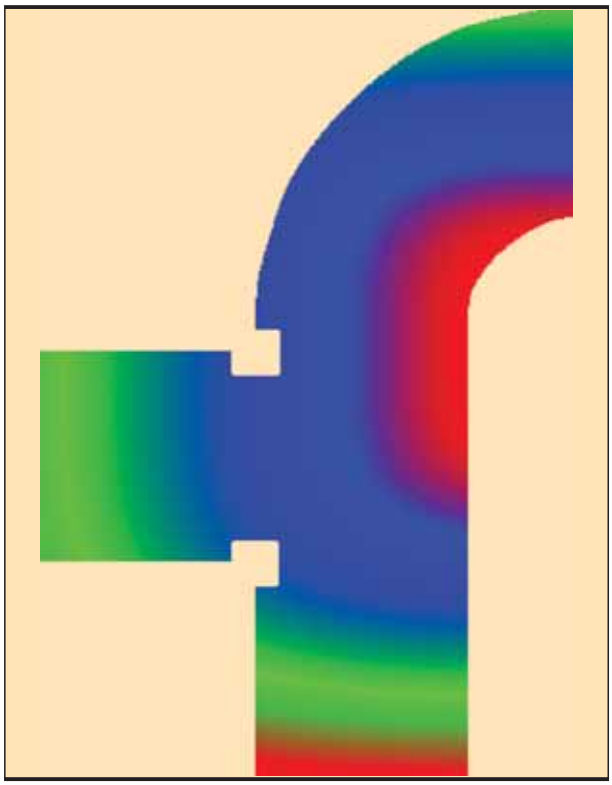

FIGURA 4B. Diferencia entre dos imágenes generadas interpolando a partir de una nube de 41987 puntos con un radio de 0.1u.

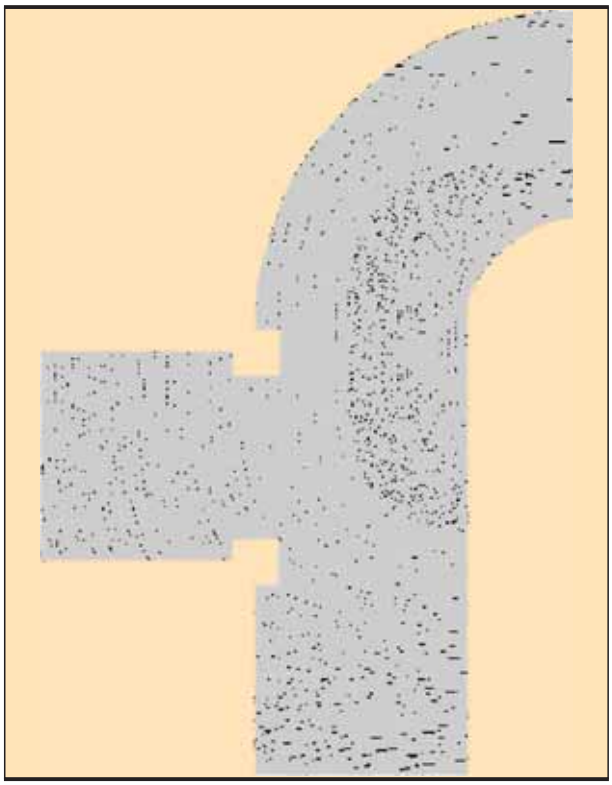

FIGURA 4C. Diferencia entre ambas imágenes; los puntos negros indican los píxeles donde existía diferencia de colores.

polation family. Comput. Graph. Forum, 24(4):809-820, 2005.

[6] Tamal K. Dey and Jian Sun. An adaptive mls surface for reconstruction with guarantees. In SGP '05: Proceedings of the third Eurographics symposium on Geometry processing, page 43, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[7] Gregory E. Fasshauer. Meshfree Approximation Methods with MATLAB, volume 6 of Interdisciplinary Mathematical Sciences. World Scientific, 2007.

[8] Michael S. Floater and Armin Iske. Multistep scattered data interpolation using compactly supported radial basis functions. J. Comput. Appl. Math., 73(1-2):65-78, 1996.

[9] Richard Franke and Greg Nielson. Smooth interpolation of large sets of scattered data. International Journal for Numerical Methods in Engineering, 15:1691-1704, 1980.

[10] Richard S. Gallagher and Solomon Press. Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis. CRC-Press, 1995.

[11] Armin Iske and V. I. Arnold. Multiresolution Methods in Scattered Data Modelling. SpringerVerlag, 2004.

[12] Christopher Johnson and Charles Hansen. Visualization Handbook. Academic Press, Inc., Orlando, FL, USA, 2004.

[13] Thomas Kollar. Fast nearest neighbors. Technical report, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2006.

[14] David Levin. The approximation power of moving least-squares. Math. Comput., 67(224):1517-1531, 1998.

[15] G.R. Liu. Mesh free methods - moving beyond the finite element method, chapter "Point Interpolation Methods". CRC-Press LLC, 2003.

[16] Gregory M. Nielson. Scattered data modeling. IEEE Computer Graphics and Applications, 13(1):60-70, 1993.

[17] Sung W. Park, Lars Linsen, Oliver Kreylos, and John D. Owens. Discrete sibson interpolation. IEEE Transactions on Visualization and Computer Graphics, 12(2):243-253, 2006. Member-Bernd Hamann.

[18] Ryusuke Sagawa, Tomohito Masuda, and Katsushi Ikeuchi. Effective nearest neighbor search for aligning and merging range images. 3dim, 0:79, 2003.

[19] Donald Shepard. A two-dimensional interpolation function for irregularly spaced data. In Proceedings of the 1968 23rd ACM national conference, pages 517-524, New York, NY, USA, 1968. ACM.

[20] R. Sibson. A vector identity for the dirichlet tessellation. Math. Proc. Cambridge Philosophical Soc., 87:151-155, 1980.

[21] Holger 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.

[22] Peter 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, pages 311-321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.