

# Módulo de Deformación para Objetos Deformables basado en el Algoritmo ChainMail

## *Deform module for deformable objects based on chainmail algorithm*

Yausel Ruiz Marine, Yerandi Marcheco Diaz

Universidad de las Ciencias Informáticas

[ymarine@uci.cu](mailto:ymarine@uci.cu)

### **Resumen**

En los últimos años los sistemas de realidad virtual se han convertido en una parte importante en el entrenamiento de las más diversas actividades humanas. Sin embargo, las limitaciones del hardware representan un obstáculo para el desarrollo de aplicaciones complejas, como es el caso de la simulación quirúrgica. El objetivo de este trabajo es brindar una solución al problema de la simulación de objetos deformables, prestando especial atención a la simulación de órganos.

Se ha desarrollado un estudio para conocer el estado del arte a nivel mundial de las principales técnicas usadas en la deformación de objetos, tanto geométricas; como son Splines, Curvas Bezier, la Deformación de Libre Forma y el Algoritmo 3D ChainMail. Así como métodos basados en física, entre los que se encuentran, el Método de Elementos Finitos, Método de Elementos de Frontera y Sistemas Masa-Resorte.

En este documento se propone una solución con el desarrollo de un módulo de software basado en la técnica 3D ChainMail, que será acoplado a la Scene Toolkit (STK).

**Palabras clave:** Algoritmo, 3D ChainMail, simulación.

### **Abstract**

*In the last years the systems of virtual reality have become an important part in the training of the diverse human activities. Nevertheless, the limitations of hardware represent an obstacle for the development of complex applications, as it is the case of the surgical simulation. The objective of this work is to offer a solution to the problem of the simulation of deformable objects, paying special attention to the organs simulation.*

*A study has been developed to know the state of the art at world level used in the deformation of objects, Studying geometric techniques as Splines, Bezier Curves, the Free Form Deformation and the 3D ChainMail Algorithm. As well as methods based on physics, like Finite Elements Method, Boundary Elements Method and Mass-Springs Systems. In this document a solution with the development of a module of software based on the technique sets out 3D ChainMail, that will be connected to Scene Toolkit (STK).*

**Key words:** Algorithm, 3D ChainMail, Simulation.

### **Introducción**

El desarrollo de los simuladores es cada vez más acelerado, así como su alto poder de representación de la realidad; ellos tienen un especial uso en la medicina, principalmente en la cirugía. El entrenamiento habitual de los cirujanos en las técnicas quirúrgicas se realiza utilizando cadáveres, ya sea de personas o de animales; lo que trae consigo una serie de dificultades, incluyendo el alto costo que representa [2], estas se han ido eliminando con el desarrollo de los Simuladores Quirúrgicos. De ahí la importancia de los mismos y de su capacidad de representar la realidad. Lograr representar objetos deformables en un entorno virtual ante la

influencia de diversos agentes es un paso primario para el futuro desarrollo de simuladores de cirugía que permitan representar deformaciones, cortes, cauterizaciones y otras acciones sobre los órganos.

En la Universidad de las Ciencias Informáticas, se desarrollan aplicaciones en el área de realidad virtual como juegos y simuladores dentro de ellos el Simulador Quirúrgico. Existe ya una biblioteca de clases desarrollada, llamada *SceneToolKit* (STK) [1], con implementaciones que facilitan el trabajo en la producción de simuladores. Se dispone de un módulo para manipulación de objetos deformables, pero se requiere mayor velocidad en aras de representar escenas cada vez más complejas.

En la actualidad se hace un poco difícil la representación real de las deformaciones de los órganos debido a las limitaciones de hardware existentes, o a la falta de algoritmos eficientes. Pero gracias a las grandes ventajas que aportan los simuladores quirúrgicos al entrenamiento de cirujanos son varios los grupos de investigaciones que le dedican un gran esfuerzo a la producción de estos. Existen algunos países que ya han tenido resultados positivos en estas investigaciones, tal es el caso de España con el InsightMIST [3] y Francia con el Hepatic Surgery Simulator desarrollado por S. Cotin [4]. Muchos son también los centros que se dedican a la investigación y desarrollo de los simuladores como es el Laparoscopic Cholecystectomy desarrollado por Michael Downes en la Universidad de California [5], y el Laparoscopic Simulator desarrollado por Srinivasan en el Laboratory for Human and Machine Haptics de la Universidad de Missouri, Columbia.

Uno de los primeros modelos para la deformación de objetos fue introducido por Dimitri Terzopoulos [8]. Este primer modelo, es basado en la Ley de Hooke para objetos perfectamente elásticos, y es tomado como base para el desarrollo de las demás técnicas de deformación que se han desarrollado posteriormente.

Estas técnicas se pueden dividir en dos grupos: en técnicas basadas en métodos geométricos y técnicas basadas en métodos físicos.

Las técnicas basadas en métodos geométricos son aquellas que emplean modelos de interpolación paramétrica (modelos polinomiales, splines...) para la estimación de la deformación, los cuales dependen sólo de relaciones no físicas. Esta técnica es muy efectiva en simuladores quirúrgicos por su rapidez en el cálculo de la deformación. Entre estas se pueden encontrar Splines y Ajuste de Curvas, Deformación de Libre Forma y 3D ChainMail.

Entre las técnicas basadas en métodos físicos se pueden encontrar los Sistemas Masa-Resorte, Método de Elementos Finitos (MEF) y Método de los Elementos de Frontera (BEM) que se sustenta en la física de materiales y la mecánica continua. Aunque muchas personas son del criterio que estas técnicas aportan un mayor realismo a las aplicaciones que la usan, hoy en día son varios los Simuladores Quirúrgicos que eliminan la física de los modelos de órganos mediante la limitación de la interacción a una mera detección de choques entre objetos rígidos [9] o bien incorporan únicamente características físico elásticas [10] debido al alto costo computacional que implica el uso de las técnicas basadas en métodos físicos.

## **2. Métodos de deformación geométrica**

### **2.1 Deformación de Libre Forma.**

La Deformación de Forma Libre (*Free-Form Deformation FFD*) es un método general para la deformación de objetos mediante el ajuste de puntos de control. Este método basa su funcionamiento en alojar un objeto en una región del espacio o volumen deformable, de tal forma que cada punto del objeto presente una parametrización única en el espacio 3-D, que define su posición en la región. La región de deformación está descrita por un paralelepípedo al cual se le asocia un sistema de coordenadas locales. La región de deformación de coordenadas es mallada de manera que se establecen planos perpendiculares formando una malla o rejilla de puntos de control en el espacio 3-D. Esta rejilla es posteriormente alterada al asignársele propiedades de deformación elástica, causando un ajuste de la posición de los puntos del objeto, basado en su parametrización inicial (Fig. 1). Tal

ajuste en la posición de los puntos del objeto, está definido por una función de deformación que se encarga de establecer la correspondencia de los puntos del objeto inicial y de los puntos del objeto deformado. La función de deformación está definida por el producto tensorial de funciones polinomiales de Bernstein [11].

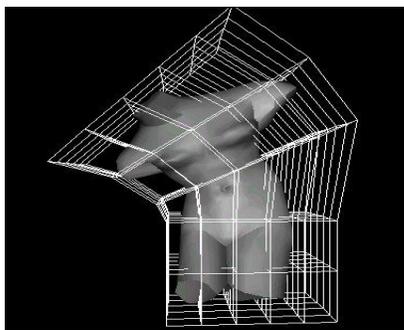


Fig. 1 Deformación de un cuerpo mediante FFD

Los primeros trabajos en deformación libre fueron introducidos por Sederberg y Parry [12] en 1986 y estaban basados en la deformación de objetos contenidos en una estructura espacial *lattice*, regularmente dividida y definida por una matriz tridimensional de puntos de control [15]. Posteriormente Griessmair y Purgathofer [13] presentaron una nueva técnica que estaba basada en estructuras tri-variadas B-Spline (B-Spline FFD). A continuación la técnica de FFD es extendida por Coquillart [14], ampliando las estructuras utilizadas a formas no paralelepípedas.

La carencia de propiedades internas que impiden crear deformaciones realistas y las limitaciones que presenta a la hora de deformar estructuras complejas son alguna de las desventajas que presenta este método. Pero gracias a su facilidad de uso y el poder de crear muchos tipos de deformaciones con poca interacción por parte del usuario y por su rapidez en las deformaciones, este método ha sido usado en varios trabajos como en el de simulación de cavidades cardiacas, y en sistemas de simulación quirúrgica como el de *Laparoscópica* implementado por C. Basdogan en 1998 [16].

## 2.2 3D ChainMail.

En 1997 Sara F. F. Gibson [17] presentó un algoritmo de deformación llamado 3D ChainMail, inicialmente era para objetos deformables de escenarios quirúrgicos, aunque es un algoritmo basado en geometría, este algoritmo puede modelar una amplia variedad de materiales [18], incluyendo rígidos, deformables, elásticos y sustancias plásticas. Además el método puede modelar materiales anisótropos tales como músculos. La idea básica de este es que cada elemento del volumen es vinculado con sus seis vecinos más cercanos. Cuando un elemento de la estructura es presionado, los enlaces cercanos absorben el movimiento cumpliendo sus restricciones de movimiento. Si un vínculo entre dos elementos es estirado o comprimido a su límite, los desplazamientos son transferidos a sus enlaces cercanos. De este modo los desplazamientos pequeños de un área seleccionada en un sistema relativamente suave resulta solamente en deformaciones locales del sistema, al igual que en una cadena los elementos vecinos solo responden si las restricciones de distancia son violadas (Fig. 2). Cambiando estas restricciones se pueden modelar tanto cuerpos rígidos como deformables.

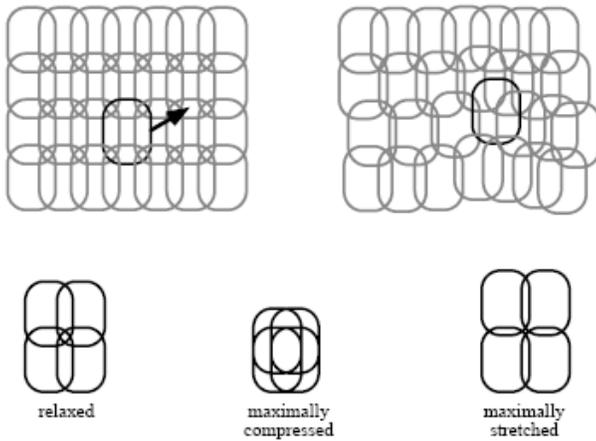


Fig. 2 Representación 2d del algoritmo ChainMail al ser movido un elemento.

Este algoritmo cuenta con tres ventajas principales, lo primero es que cada elemento es considerado a lo máximo una vez en cada deformación, en segundo lugar cada elemento es comparado con solo un vecino, para determinar como el debe moverse y por último la deformación se propaga de adentro hacia fuera y esta debe terminar lo antes posible. Esto hace que permita trabajar con gran cantidad de datos y seguir comportándose de forma interactiva. Este algoritmo soporta cambios topológicos en caso de una modelación dinámica, por lo que los vínculos de los elementos con sus vecinos pueden ser fácilmente desconectados al ser necesario, esto es muy utilizado en el proceso de corte de tejidos en la simulación quirúrgica.

No obstante este algoritmo también posee desventajas, por ejemplo fue diseñado para funcionar en mallas rectilíneas solamente y con datos homogéneos, pero desde entonces se han escrito otras variantes que arreglan esta situación, primeramente en 1998 M. A. Schill [19] introdujo el algoritmo permitiendo el procesamiento con datos heterogéneos, pero con el problema de que no asegura que cada elemento pueda ser verificado como máximo una vez, por lo que la ventaja de la velocidad se pierde un poco, M. A. Schill resuelve esto con listas ordenadas en el proceso de selección de los vecinos a mover.

En el 2002 Y. Li y k. Brodlie [20] escribieron una nueva variante del algoritmo, donde solucionaban el problema de la malla rectilínea en este momento los elementos pueden tener cualquier número de vecinos y no se supone acerca de la topología de los elementos vecinos. Para manejar las mallas no uniformes se cambian las restricciones absolutas por relativas para los elementos vecinos, cuando se dice malla no uniforme se quiere decir cualquier malla topológicamente equivalente a una uniforme que tiene conexiones con elementos de su interior (Derecho, Izquierdo, Arriba, Abajo, Delante y Detrás).

Recientemente en el año 2005 C. Dräger [16] publica una nueva variante del algoritmo ChainMail donde reutiliza algunas de las mejoras de estos trabajos además de utilizar la administración de la memoria explotando el hecho de que las deformaciones se propagan localmente y hacia fuera, por lo que se divide el volumen en mallas de macro cubos que son cargados según la demanda de los mismos minimizando el uso de memoria, esto repercute en velocidad para el algoritmo.

Este algoritmo puede ser comparado con el método basado en física Masa-Resorte y permite la realización de corte en tejidos presentando ventajas, además de representar sus deformaciones de manera interactiva.

### 2.3 Splines y Ajustes de Curvas.

Un Spline es comúnmente referida como una curva compuesta de secciones polinomiales satisfaciendo ciertas condiciones de continuidad entre ellas; y donde B-Splines son curvas de aproximación (no interpolan los puntos de control) que tienen la ventaja de permitir una manipulación local de la curva y requerir menos cálculos para la determinación de sus coeficientes.

Estos métodos se basan en controlar las deformaciones a objetos volumétricos mediante curvas (Splines, B-Splines, NURBS, Bezier o interpolación polinomial), de manera general estas curvas están formadas por una serie de puntos llamados puntos de

control. La forma de los objetos puede ser ajustada moviendo los puntos de control o variando la cantidad de ellos (Fig. 3). La deformación estaría definida entonces por funciones paramétricas (3D Spline), las que se determinan por la ubicación de los puntos de control.

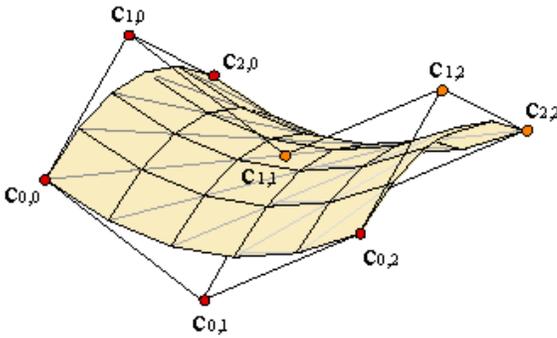


Fig. 3 Curva B-Spline.

Este método es muy ventajoso desde el punto de vista del bajo costo que implica su representación en el rendimiento de la PC, pero se ve muy limitado a la hora de realizar representaciones muy precisas de determinados objetos, debido a que la colocación de los puntos de un objeto determinado es muy difícil de lograr. La utilización de este método se ha visto mayormente en la representación de objetos como órganos, no así en el uso de deformaciones [23].

### 3. Métodos de deformación basados en física

#### 3.1 Sistemas Masa-Resorte

Un sistema Masa-Resorte consiste en un conjunto de  $n$  puntos con masa, cada uno vinculado a sus vecinos por resortes sin masas y largo mayor que cero [24] (Fig. 4), dominados por la 2da Ley de Newton  $f = m \times a$ , donde  $m$  es la masa de cada punto y  $f$  es la sumatoria de las fuerzas incidentes sobre dicho punto, las fuerzas están divididas en dos categorías; internas, dada por la tensión de los resortes regida por la Ley de Hooke, que es el resultado de la rigidez de la vinculación de todos los resortes de este punto a sus vecinos, y externas, que pueden diferir en su naturaleza dependiendo de qué tipo de simulación se quiera modelar, frecuentemente son la gravedad y el amortiguamiento viscoso.

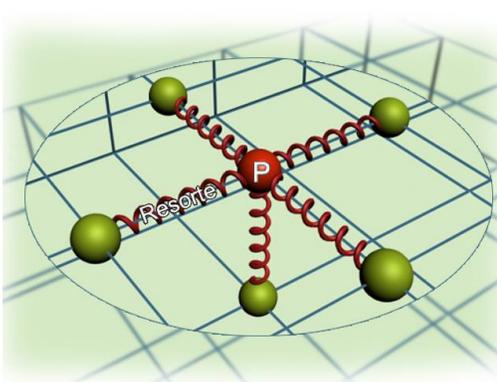


Fig. 4 Sistema Masa-Resorte Relajado

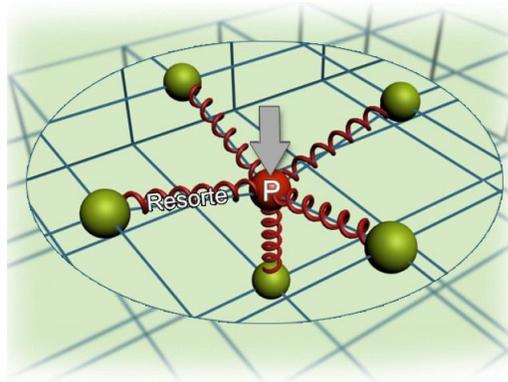


Fig. 5 Sistema Masa-Resorte Aplicando Fuerza

Si una deformación ocurre esta se propaga a través del objeto por sus tensores según las fuerzas aplicadas y las características de estos ( Fig. 5), este método es de fácil implementación y provee al sistema con un cómputo de sus deformaciones rápidas, permitiendo la simulación de un rango de objetos en los que se incluyen los tejidos visco-elásticos en el caso de la cirugía.

El uso de sistemas Masa-Resorte ha sido difundido en trabajos como la animación facial antes mencionada, en la animación de telas [25], en la modelación de deformaciones de músculos, en el año 1998 Nedel y Thalman [26], y recientemente en el 2007 para la representación de deformación de órganos en el simulador KHEIPROS por Orosco y Bretau[1].

### 3.2 Método de Elementos Finitos.

El Método de los Elementos Finitos (MEF) es un método de aproximación de problemas continuos, donde el continuo se divide en un número finito de elementos, cuyo comportamiento se especifica mediante un número finito de parámetros asociados a ciertos puntos característicos denominados nodos. Estos nodos son los puntos de unión de cada elemento con sus adyacentes, la solución del sistema completo sigue las reglas de los problemas discretos. El sistema completo se forma por ensamblaje de los elementos, las incógnitas del problema dejan de ser funciones matemáticas y pasan a ser el valor de estas funciones en los nodos, el comportamiento en el interior de cada elemento queda definido a partir del comportamiento de los nodos mediante las adecuadas funciones de interpolación ó funciones de forma. El MEF, por tanto, se basa en transformar un cuerpo de naturaleza continua en un modelo discreto aproximado, esta transformación se denomina discretización del modelo (Fig. 6). El conocimiento de lo que sucede en el interior de este modelo del cuerpo aproximado, se obtiene mediante la interpolación de los valores conocidos en los nodos.

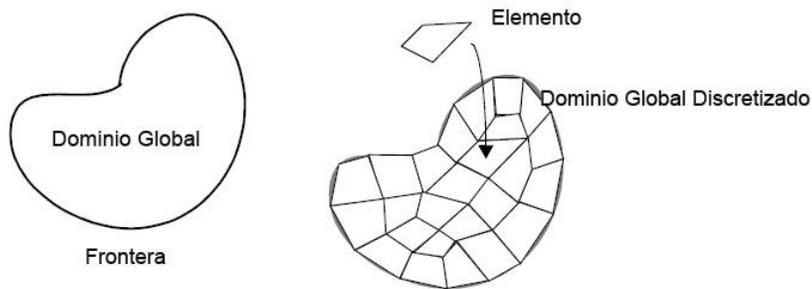


Fig. 6 Discretización de un dominio en elementos.

Aunque es una eficiente manera para calcular la representación de diferentes procesos físicos con un elevado realismo, dentro de ellos las deformaciones de objetos en las aplicaciones virtuales; dado su fuerte cálculo numérico, la resolución de sus ecuaciones resulta muy costosa computacionalmente por lo que limita su uso en aplicaciones que necesitan interactividad en tiempo real.

Muchos trabajos han encontrado solución con el Método de Elementos Finitos desde hace aproximadamente dos décadas atrás. Terzopoulos [28] lo usó para modelar deformaciones físicas como la animación facial basado en la teoría de la Elasticidad, Bro-Nielsen y Cotin [29] usan el método MEF linealizado en la simulación quirúrgica, otros como Nienhuys y Van der Stappen usan un algoritmo interactivo para el método MEF linealizado para simulación de deformaciones proceso de corte. En el 2004 Irving y Fedkiw [31] desarrollan un método basado en MEF que permite deformaciones estables.

### 3.3 Método de los Elementos de Frontera.

El Método de los Elementos de Frontera (*Boundary Element Method, BEM*) es una importante técnica matemática que da solución a diferentes problemas científicos. Se puede decir que el BEM es alternativa del MEF, aunque nada más se puede aplicar este método a cuerpos donde su interior está compuesto por un material homogéneo.

En el Método de los Elementos de Frontera los cálculos se realizan en la superficie del cuerpo elástico en lugar de su volumen. El método logra un aumento sustancial de la velocidad debido a que el problema tridimensional original, es reducido a dos dimensiones. (Fig. 7).

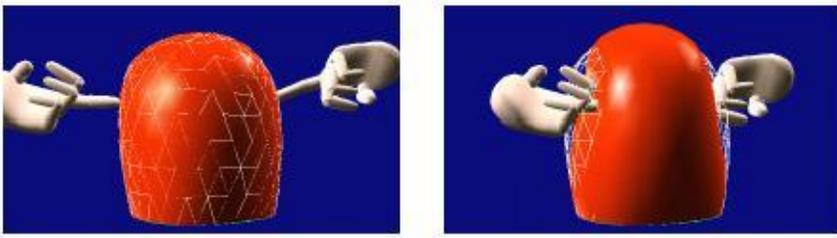


Fig. 7 El Método de los Elementos de Fronteras en la deformación de un objeto.

Una de las primeras aplicaciones exitosas del BEM, a problemas no lineales independientes del tiempo en mecánica de sólidos con la formulación para plasticidad fue debido a Swedlow y Cruse [32]. Esto fue seguido por una implementación numérica por Riccardella [33]. Otras aplicaciones del BEM incluyen la elástica y muy recientemente la mecánica de fractura inelástica [36].

El Método de Elementos de Frontera ha demostrado su eficacia para resolver algunos tipos de problemas. Por ejemplo, permite la simulación de problemas que tienen variaciones muy pronunciadas de esfuerzos, como es el caso de la fractura o agrietamiento de un sólido, así como simulación de fluidos, y de objetos deformables. Permite además el modelado de problemas complejos, en especial aquellos que requieren la resolución en regiones abiertas o una definición precisa de los contornos.

#### 4. Método de deformación, 3d chainmail.

##### 4.1 Generación de la Rejilla Regular.

La herramienta STK permite la representación de mallas triangulares, lo que posibilita acceder a los vértices de estas. A partir de la información que brinda la herramienta se puede generar una Rejilla Regular, donde los nodos que conforman dicha rejilla contendrán información como: los vértices de la malla, la dirección de sus nodos vecinos y otros datos necesarios. Tomando la información de los vértices de la malla se calcula los extremos máximo y mínimo del objeto representado, recorriendo la lista de vértices y escogiendo los valores máximos y mínimos de cada eje de coordenada; con este resultado se obtiene la longitud de las aristas  $a_1, b_1, c_1$  del paralelepípedo envolvente.

Para la creación de los nodos que conforman la rejilla se calcula el Área promedio de los triángulos que forman la malla, esta Área se iguala al Área de un cubo, despejando el lado se obtiene la longitud de la arista de los nodos. Como sigue:

$$A_{\text{triángulo}} = \frac{b \cdot h}{2} \quad (1)$$

Donde  $b$ ,  $h$  son la base y la altura del triángulo respectivamente.

Sumando las Áreas de los triángulos y dividiéndola entre la cantidad total de triángulos se obtiene el Área promedio. Como sigue:

$$A_{\text{promtriángulo}} = \frac{\sum (A_1 + A_2 + \dots + A_n)}{n} \quad (2)$$

Donde  $A$  es el área de un triángulo y  $n$  es la cantidad total de triángulos.

El área del cubo se obtiene de la siguiente manera:

$$A_{\text{cubo}} = 6a^2 \quad (3)$$

Cada cubo que forma parte de la rejilla regular debe de contener la menor cantidad posible de los vértices que forman la rejilla, para ello se iguala el área del cubo  $A_{\text{cubo}}$  al área promedio de los triángulos  $A_{\text{promtriángulo}}$  con el objetivo de conocer el valor de la longitud que tendrá la arista  $a$  de los cubos que formarán la rejilla regular, como sigue:

$$A_{\text{promtriángulo}} = 6a^2 \quad (4)$$

Despejando para obtener el valor de la arista  $a$  se obtiene:

$$a = \sqrt{\frac{A_{\text{promtriángulo}}}{6}} \quad (5)$$

Para Calcular la cantidad de cubos que se generaran en cada eje coordenada se hace la división entera de las longitudes de los ejes de coordenadas  $a_1, b_1, c_1$  pertenecientes al paralelepípedo entre la longitud de la arista  $a$  de los cubos.

Con estos datos se generará una rejilla regular tri-dimensional que será la representación espacial del objeto ChainMail en cubos de tamaños iguales.

Cada cubo tendrá una posición dentro de la matriz tridimensional que forma la rejilla regular, es decir (Fila, Columna, Profundidad). Por ejemplo el cubo cuya posición es (3, 3, 3) sería el que se muestra en la Fig. 8.

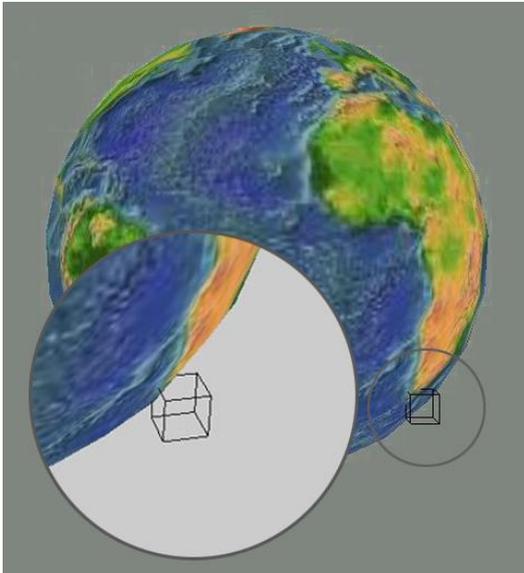


Fig. 8 Posición del Cubo (3, 3, 3) en la Rejilla Regular generada.

Cada cubo contendrá la lista de vértices que pertenecen a su dominio espacial, así como sus extremos máximo, mínimo y su respectivo centro, además de conocer a sus seis vecinos: Derecho, Izquierdo, Superior, Inferior, Delantero y Trasero.

#### 4.2 Algoritmo 3D ChainMail

Con la rejilla regular formada cada nodo de ella deberá cumplir con un rango horizontal  $MinDx$  y  $MaxDx$  para sus vecinos izquierdo y derecho, un rango vertical  $MinDy$  y  $MaxDy$  para sus vecinos superior e inferior y dentro de un rango de profundidad  $MinDz$  y  $MaxDz$  para sus vecinos anterior y posterior respectivamente.

Existen 6 listas de nodos candidatos a movimiento para tratar la deformación, cuando un cubo es atraído o empujado sus vecinos son automáticamente agregados a sus respectivas listas de candidato, se verifican sus restricciones para cada eje coordenada, con respecto a su vecino referencia, que sería quien le transfirió el movimiento. Por ejemplo si un nodo es comprimido el elemento y su vieja posición son adicionados en una lista de elementos movidos, y sus vecinos cercanos son adicionados a sus respectivas listas de candidatos.

La lista de los elementos candidatos es procesada una a la vez en el siguiente orden: derecho, izquierdo, superior, inferior, anterior y posterior. La lista de candidatos de los vecinos derechos es procesada de la siguiente manera: comenzando con el primer elemento en la lista, sus restricciones de deformación son chequeadas entre el elemento de la lista y el elemento que lo adicionó (siempre es el vecino izquierdo). Si las restricciones son violadas el elemento es movido una mínima distancia hasta que cumpla con las restricciones. La nueva posición del cubo se calcula de la siguiente manera:

$$(x - xneigh) < MinDx, \quad x = xneigh + MinDx \quad (6)$$

$$(x - xneigh) > MaxDx, \quad x = xneigh + MaxDx \quad (7)$$

Donde  $x$  es el valor de las coordenadas del nodo comprimido,  $xneigh$  es el valor de la coordenada de su vecino izquierdo.

Si el elemento es movido, sus vecinos son adicionados a sus respectivas listas de candidatos. Cada candidato en la lista es procesado hasta que no quede ninguno.

La lista de los vecinos izquierdo es procesada de forma similar excepto que el elemento izquierdo es introducido por su vecino derecho, entonces el movimiento del nodo izquierdo causa que sus vecinos izquierdo, frente, trasero, superior e inferior sean adicionados a la lista de candidatos. Ahora la nueva posición del cubo se calcularía usando (6), (7); pero ahora  $xneigh$  sería el valor de la coordenada del vecino derecho.

Las demás listas de vecinos son procesadas de forma similar para cada caso; solo que con los vecinos superior e inferior la nueva posición del cubo se calcularía de la siguiente manera:

$$(y - yneigh) < MinDy, \quad y = yneigh + MinDy \quad (8)$$

$$(y - yneigh) > MaxDy, \quad y = yneigh + MaxDy \quad (9)$$

Donde  $y$  es el valor de las coordenadas del nodo comprimido y  $yneigh$  es el valor de la coordenada de su vecino superior o inferior, depende del vecino que se esté tratando.

Para el caso de la lista de vecinos anterior y posterior la nueva posición de calcularía de la siguiente forma:

$$(z - zneigh) < MinDz, \quad z = zneighs + MinDz \quad (10)$$

$$(z - zneigh) > MaxDz, \quad z = zneigh + MaxDz \quad (11)$$

De esta forma cada nodo que es chequeado cambia su posición y propaga la deformación para cada uno de sus seis vecinos.

De manera general pudiera escribirse el algoritmo como sigue:

Para cada triángulo calcular el área usando (1).

Calcular el área promedio de los triángulos usando (2).

Determinar los valores de los extremos máximos y mínimos de la malla.

Calcular longitud de la arista de los cubos de la rejilla usando (3), (4), (5).

Generar la rejilla regular.

En cada ciclo de render:

Para cada nodo de la rejilla:

Determinar si fueron violadas sus restricciones de deformación.

Adicionar nodo a la lista de vecinos para ser procesados.

Calcular la nueva posición usando (6), (7), (8), (9), (10), (11).

La implementación se realizó siguiendo el paradigma de programación orientada a objeto en lenguaje C++ con Microsoft Visual C++ .NET 2003 para garantizar su compatibilidad con el STK.

Las nomenclaturas y estándares de codificación son las mismas que las del STK, la escritura del código fuente será en C++ estándar para su compresión por Windows así como Linux.

## Resultados

Finalmente se realizó una comparación en igualdades de condiciones entre el módulo de deformación existente, que está basado en el método de Masa Resorte y el módulo de deformación obtenido; arrojando como resultado que el módulo obtenido supera en velocidad de render al existente, como se muestra a continuación.

Tabla 1. Resultados.

Técnica	Cantidad de Polígonos	Prestaciones de la PC.	Velocidad de Render
3D ChainMail	8170	Intel PIV, 3.0GHz,1.0 GB RAM	44.0 fps
		Intel PIV, 3.0GHz, 1.0 GB RAM, NVIDIA Quadro FX 500/FX 600.	75.5 fps
Sistema Masa-Resorte	8170	Intel PIV, 3.0GHz,1.0 GB RAM	28.5 fps
		Intel PIV, 3.0GHz, 1.0 GB RAM, NVIDIA Quadro FX 500/FX 600.	45.0 fps

### Conclusiones

Como parte del cumplimiento de los objetivos planteados se realizó un estudio de las diferentes técnicas y algoritmos de deformación de cuerpos, con profundidad en el Algoritmo ChainMail del que surgió la solución y al que se le incluyo un nuevo parámetro de dureza para permitir la deformación regional del cuerpo deformable, lo que hace novedoso el trabajo.

Finalmente se obtuvo un módulo de deformación funcional, que se ajusta a los objetivos planteados y acoplado a la STK, y que puede ser usado en simuladores quirúrgicos que utilicen dicha librería.

### Recomendaciones

Se recomienda el estudio y posible incorporación de propiedades físicas al módulo obtenido en pos de lograr un mayor realismo en las deformaciones.

### Referencias Bibliográficas

- [1] Ramirez R. Bretau O. "Deformación de Objetos para Sistema de Realidad Virtual", UCI, 2007.
- [2] Dawson S.L., Kaufman J.A., "The Imperative for Medical Simulation", Proc. IEEE, 86 (3), pp. 479-483, 1998.
- [3] Álvarez Clemente "Simulador quirúrgico con sentido del tacto [http://www.elpais.com/articulo/futuro/Simulador/quirurgico/sentido/tacto/elpepusocfut/20070711elpepifut\\_5/Tes](http://www.elpais.com/articulo/futuro/Simulador/quirurgico/sentido/tacto/elpepusocfut/20070711elpepifut_5/Tes)
- [4] S.Cotin, H.Delingete, N. Ayache, *Real Time Volumetric Deformable Models for Surgery Simulation*, Visualization in Biomedical Computing (Proc. VBC '96), K.H. Höhne, R. Kikinis (eds.), Lecture Notes in Computer Science, vol. 1131, Springer-Verlag, 1996
- [5] Downes, M.: Hsu, A. A virtual environmet for training laparoscopic cholecystectomy. University of California, Berkeley. CS294-5 Virtual Reality, Spring Semester, 1997.

- [8] Terzopoulos D, Platt J, Barr A, and Fleischer K. "Elastically deformable models." July 1987. Proceedings of SIGGRAPH'87 (Anaheim, California).
- [9] Geiger, B.; Kikinis, R. *Simulation of endoscopy*. In Ayache, N. (ed), Proceedings of CVRMed'95, pp.277-281. Springer-Verlag, Berlin.
- [10] Cover, S.A.; *Interactively deformable models for surgery simulation*, IEEE Computer Graphics & Applications, pp. 68-75, Nov. 1993.
- [11] Bernd Hamann, Donhua Wu, and Robert J. Moorhead II. On particle path generation based on quadrilinear interpolation and Bernstein-Bézier polynomials. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):210–217, 1995.
- [12] Sederberg, T.W., Parry, S.R. *Free-form deformation of solid geometric models*. Siggraph'86. Computer Graphics 20 (4), 151-196, 1986.
- [13] Griessmair J., Purgathofer W. *Deformation of solids with trivariate B-spline*. Proceedings of Eurographics'89, 137-148, 1986.
- [14] Coquillart, S. Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. Computer Graphics 24 (4), 187-196, 1990.
- [15] Bechmann, D. *Space Deformation Models Survey*. Computers & Graphics 18 (4), 571-586, 1994.
- [16] Dräger Ch. "A ChainMail Algorithm for Direct Volumen Deformation in Virtual Endoscopic Simulation". Vienna University of Technology. Mayo 2005.
- [17] Sarah F. F. Gibson, 3D ChainMail: a fast algorithm for deforming volumetric objects. In Michael Cohen and David Zeltzer, editors, 1997 Symposium on Interactive 3D Graphics, pages 149-154. ACM SIGGRAPH, April 1997. ISBN 0-89791-884-3.
- [18] Web-Based Surgical Simulators and Medical Education Tools.<http://synaptic.mvc.mcc.ac.uk/simulators.html>
- [19] M. A. Schill, S. F. F. Gibson, H.-J. Bender, R. Manner (1998). Biomechanical Simulation of the Vitreous Humor in the Eye Using Enhanced ChainMail Algorithm. Proceedings of Medical Image Computation and Computer Assisted. Interventions (MICCAI) 1998, S. 679{687.
- [20] K. Brodlie Y. Li (2003). Soft Object Modeling with Generalized ChainMail Extending the Boundaries of Web-based Graphics. Proceedings of Computer. Graphics Forum, 22(4):717{727.
- [21] Gibson S. F. F., Mirtich B. "A Survey of Deformable Modeling in Computer Graphics" A Mitsubishi Electric Research Lab, Noviembre 1997.
- [23] Vassilev T, Spanlang B.(2001) "A Mass-Spring Model for Real Time Deformable Solids".
- [24] Villard,1 Julien; Borouchaki, Houman(2005). Adaptive Meshing For Cloth Animation.
- [25] Nedel, L. P., And Thalmann (1998), D., Real Time Muscle Deformations using Mass-spring Systems, Proceedings of the Computer Graphics International.
- [26] Eduardo Frías Valero. El Método de los Elementos Finitos,Departamento de Ingenieria Electrica UPC, 2004.
- [27]D. Terzopoulos, K. Waters (1990). Physically-Based Facial Modeling, Analysis, and Animation. The Journal of Visualization and Computer Animation, 1:73,80.
- [28] Bro-Nielsen M. Cotin S. Real-time volumetric deformable models for surgery Simulations using finite elements and condensation.Computer Graphics Forum, 1996.
- [29] S.Cotin, H.Delingete, N. Ayache, *Real Time Volumetric Deformable Models for Surgery Simulation*, Visualization in Biomedical Computing (Proc. VBC '96), K.H. Höhne, R. Kikinis (eds.), Lecture Notes in Computer Science, vol. 1131, Springer-Verlag, 1996

- [31] Swedlow, J. L. and Cruse, T. A. Formulation of the boundary integral equation for three-dimensional elastoplastic flow, *International Journal of Solids and Structures*, 7, 1673-1681 (1971).
- [32] Riccardella, P. An Implementation of the Boundary Integral Technique for plane problems of Elasticity and Elastoplasticity, PhD Thesis, Carnegie Mellon University, Pitsburg, PA (1973).
- [33] Banerjee, P. K. and MUSTOE, G. C. W. The boundary element method for two-dimensional problems of elastoplasticity. *Recent Advances in Boundary Element Methods*, C. Brebbia (ed.), Pentech Press, Plymouth, Devon, UK, 283-300 (1978).
- [34] Mukherjee, S. A. *Boundary Element Methods in Creep and Fracture*. Applied Science Publishers LTD, Barking Essex, England (1982).