

# Modelado de Cuerpos 3D Elásticos Para Mundos Virtuales con Retroalimentación Táctil

Juan Manuel IBARRA ZANNATHA<sup>1</sup>, Claudia MARMOLEJO RIVAS<sup>1,2</sup>

<sup>1</sup>Laboratorio de Robótica  
Depto. de Control Automático del CINVESTAV  
Av. IPN No. 2508, Lindavista, 07300 México, DF  
[jibarra@ctrl.cinvestav.mx](mailto:jibarra@ctrl.cinvestav.mx)

<sup>2</sup>Escuela de Informática, U. Autónoma de Sinaloa  
Av. Universidad y Av. Leonismo Internacional s/n  
82017 Mazatlán, Sin.  
[cmarmolejo@ctrl.cinvestav.mx](mailto:cmarmolejo@ctrl.cinvestav.mx)

## Resumen

*Se introduce la estructura de un sistema háptico, sus conceptos fundamentales y la necesidad de contar con modelos geométricos dinámicos (deformables). Se presenta el método de diferencias finitas para resolver la ecuación de movimiento (ley de Hooke) propuesto por Debunne et al. También se presenta un modelo continuo que parte de la energía elástica propuesto por Delinguette et al y, además, el modelo masa resorte con una solución por el método propuesto por Desbrunne et al. Se introduce el método exacto de detección de colisiones de Moore-Wilhelms. Finalmente, se discuten los resultados de simulación obtenidos con el método de Debunne aplicado a un cuadrado unitario usando un sistema desarrollado específicamente para este proyecto y se comentan sus características y sus posibilidades de utilización en el simulador realista*

**Palabras clave:** Mundos virtuales, Respuesta háptica, Modelos físicos, Métodos numéricos, Simulación.

## 1. Introducción

La Simulación por Computadora de sistemas dinámicos es una de las herramientas más poderosas de ayuda al diseño y análisis de esta clase de sistemas. En sus fases iniciales, los resultados de la simulación por computadora se presentaban como listados con la evolución temporal de las variables de interés. La representación gráfica de estos resultados constituyó un gran progreso. Las Ciencias de la Computación ofrecen ahora una variedad amplia de métodos de despliegue de resultados dirigidos prácticamente a todos nuestros sentidos, incluyendo sofisticados mundos virtuales interactivos.

Sin embargo, esto no es suficiente para simular aplicaciones en Robótica caracterizadas por una compleja interacción robot-entorno, en donde se requiere, además, de una retroalimentación táctil importante. Aplicaciones de Robótica Médica, la teleoperación y las aplicaciones didácticas para entrenamiento de operadores son algunos ejemplos

donde se requiere la respuesta háptica del sistema de simulación computarizada.

Incluir la retroalimentación táctil o háptica en los mundos virtuales permite tener una verdadera sensación de inmersión, pues el sentido del tacto proporciona información indispensable para interactuar con el entorno, ya que involucra propiedades físicas importantes (inercia, fricción, compliancia, rugosidad y temperatura).

El término *háptico* evocan el sentido del tacto (o cosas relacionadas con él) que incluye dos categorías: sentido kinestésico, basado en la retroalimentación de fuerza, mediante el cual sentimos el movimiento y las fuerzas en nuestros músculos y articulaciones generadas por el peso de los objetos cogidos, por su inercia o su compliancia mecánica así como por las restricciones de movimiento a que está sometido. Por otro lado, el sentido del tacto permite sentir la rugosidad (texturas) y bordes (formas) de los objetos así como su temperatura [13].

En un sistema realista de simulación, el ciclo háptico consta de un usuario que, a través de un dispositivo mecánico, interactúa bidireccionalmente con un objeto virtual cuya representación gráfica se despliega en pantalla. La forma del objeto virtual está dada por un modelo geométrico (actualizado a 60 Hz) mientras que su comportamiento táctil y de movimiento lo rige un modelo dinámico que se actualiza a 1000 Hz para lograr el realismo buscado.

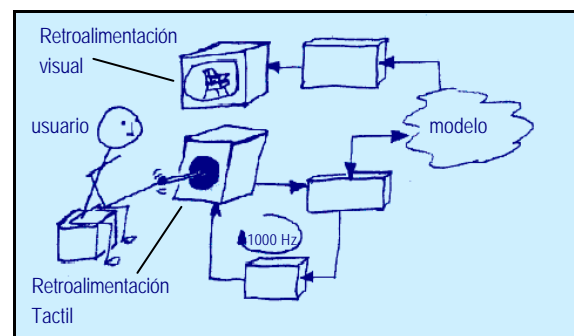


Figura 1. Diagrama de un ciclo háptico.

Así, dentro del ciclo háptico lo primero que hace el modelo físico es detectar el momento en el que los objetos entran en contacto, para evitar que los objetos virtuales intenten ocupar simultáneamente un mismo lugar. Luego, con base en las leyes de la Física, debe calcular la deformación en la región de contacto y generar la posición y velocidad de cada uno de los puntos de los objetos virtuales en contacto, variables necesarias para el despliegue en pantalla de su movimiento y para el control de la interfaz háptica que presenta al usuario las fuerzas de interacción. La inclusión en el modelo físico de los efectos de la gravedad y la fricción añaden realismo a la simulación [12] [4].

El análisis de colisiones debe generar la respuesta apropiada tanto para la retroalimentación visual como para la háptica. Para el despliegue gráfico consiste en detectar la interacción entre objetos para dibujar su deformación y sus desplazamientos correctamente, sin superposición de objetos ni penetración; mientras que, para el despliegue háptico, se deben calcular las fuerzas de reacción [2]. Ambas respuestas suelen tratarse de manera independiente aunque debieran tratarse simultáneamente, pues una buena representación gráfica está en la base de un buen cálculo de fuerzas [13].

Resumiendo, las principales funciones que desempeña un ciclo háptico son las siguientes [13]:

- Desplegar en pantalla un mundo virtual (MV) mediante el modelo geométrico correspondiente.
- Asociar un modelo dinámico a cada objeto presente en el MV, incluyendo al usuario.
- Detectar la interacción entre el modelo virtual de un objeto y el (punto de prueba del) usuario.
- Generar los cambios de movimiento y de forma de los objetos para su despliegue en pantalla.
- Calcular la fuerza de contacto resultante así como la deformación del modelo virtual.
- Generar la fuerza de retorno que se envía al dispositivo háptico y
- Controlar la respuesta háptica.

## 2. Modelado Físico

### 2.1. Modelo Geométrico

La representación gráfica de objetos para su despliegue en pantalla requiere de un modelo geométrico. Este tipo de modelo también es útil en el diseño asistido por computadora, la manufactura o la simulación de dicho objeto. Los modelos geométricos pueden estar basados en la frontera del objeto (b-rep), pueden generarse por geometría constructiva de volúmenes simples, por ocupación del espacio (octrees) o por extrusión del círculo generalizado [5].

En este trabajo se eligió una representación por superficies, en donde un volumen queda representado por una lista de caras, estas se representan por un conjunto de aristas, las que a su vez se definen por dos vértices representados, finalmente, por sus coordenadas 3D. Este modelo permite la generación de un malla de polígonos para representar la superficie del objeto o bien de cubos (o tetraedros) para representar el volumen, sin cambios en la representación como un lista de vértices.

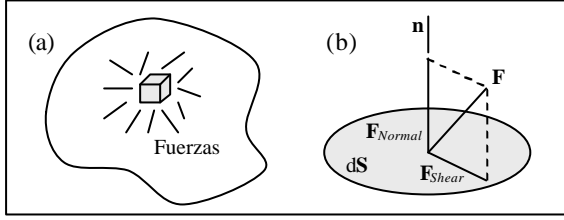
Para realizar su misión, el modelo geométrico usado para despliegue requiere a cada instante el valor de la posición de todos los puntos (vértices del modelo b-rep, nodos de la malla 2D o 3D) de los objetos virtuales, actualizada en permanencia por un modelo dinámico.

### 2.2. Modelo Dinámico

La dinámica de un sistema se representa por medio de ecuaciones diferenciales. Así, el modelo de las propiedades físicas de interés de los objetos dentro del MV se obtendrá mediante el análisis de elemento finito o de diferencias finitas. En este trabajo, las propiedades relevantes son el movimiento y la deformabilidad de los objetos virtuales. El interés es modelar un objeto 3D virtual, de geometría simple, deformable bajo la acción del usuario.

La teoría de la elasticidad trata sobre la mecánica de cuerpos sólidos concebidos como medios continuos. Bajo la acción de las fuerzas aplicadas dichos cuerpos exhiben una cierta deformación (elástica o plástica) que cambia su forma y su volumen. Todas las partes de un cuerpo sin deformar están en equilibrio mecánico, ello significa que la resultante de las fuerzas que actúan sobre cada elemento de masa del cuerpo es cero. Cuando ocurre una deformación, el arreglo de las moléculas cambia y el cuerpo deja de estar en su posición de equilibrio original; surgen entonces fuerzas que tienden a regresar al cuerpo a su posición de equilibrio denominadas tensiones internas [15].

La representación de un cuerpo deformable virtual puede lograrse calculando la fuerza externa que se ejerce sobre el cuerpo, así como la que ejerce la porción deformada sobre los elementos de materia que lo rodean mediante un modelo de medio continuo, o bien aproximando el cuerpo deformable mediante un conjunto de masas enlazadas por resortes. En cualquier caso, basta con encontrar una representación del modelo físico en términos de diferencias finitas y calcular con él la posición y velocidad de cada vértice del objeto. En el presente estudio se han evaluado dos soluciones al modelo continuo [6] [8] y una para el modelo masa resorte [9], los cuales se presentan en las secciones siguientes.



**Figura 2.** a) Fuerzas sobre  $dV$  y b) Descomposición de  $F$

### 3. Modelo de Cuerpo Continuo

Solución de Debutne *et al* por Diferencias Finitas [6]

#### 3.1. Tensor de Restitución (stress tensor)

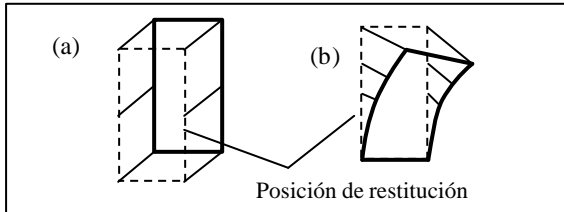
Un elemento de volumen de un objeto cualquiera recibe localmente fuerzas periféricas de todo su entorno, las cuales se pueden describir evaluando la fuerza superficial  $\mathbf{F}$  (denominada de restitución) que actúa sobre la superficie de un elemento dado. Esta fuerza tendrá una componente normal a la superficie, análoga a una presión, y otra ortogonal (figura 2). El tensor de restitución  $\mathbf{s} \in \mathfrak{R}^{3 \times 3}$  define una transformación lineal entre las normales y las fuerzas de restitución asociadas; esta matriz simétrica proporciona la fuerza de restitución aplicada en un elemento de superficie dado, caracterizado por su vector normal  $\mathbf{n}$ , mediante la ecuación:

$$\mathbf{F} = \mathbf{s} \cdot \mathbf{n} \quad (1)$$

A partir de este tensor se deduce la fuerza resultante por unidad de volumen que actúa sobre el elemento de materia como la divergencia de  $\mathbf{s}$ . Entonces, si  $\mathbf{r}$  es la densidad del elemento bajo consideración,  $\mathbf{g}$  la aceleración de la gravedad y  $\mathbf{a}$  la aceleración del elemento se puede escribir la ecuación de movimiento como la siguiente expresión:

$$\mathbf{r}\mathbf{a} = \nabla \cdot \mathbf{s} + \mathbf{r}\mathbf{g} \quad (2)$$

Para calcular el tensor de restitución es necesario conocer el estado actual de la deformación sufrida por el material y de ahí derivar las fuerzas locales. Para lo cual se define el tensor de deformación.



**Figura 3.** a) Traducción y b) Deformación arbitraria.

#### 3.2. Tensor de deformación (strain tensor)

La deformación de un objeto se mide por el campo de desplazamiento  $\mathbf{d}$  o el vector diferencia entre la posición actual y la de reposo o equilibrio. Una traslación creará un campo de desplazamiento

constante, mientras que una deformación crea un campo arbitrario (figura 3). Por definición, el gradiente de un campo vectorial es un tensor:

$$\mathbf{A} = \nabla \mathbf{d} = \begin{bmatrix} \frac{\partial d_x}{\partial x} & \frac{\partial d_y}{\partial x} & \frac{\partial d_z}{\partial x} \\ \frac{\partial d_x}{\partial y} & \frac{\partial d_y}{\partial y} & \frac{\partial d_z}{\partial y} \\ \frac{\partial d_x}{\partial z} & \frac{\partial d_y}{\partial z} & \frac{\partial d_z}{\partial z} \end{bmatrix} \quad (3)$$

cuya parte antisimétrica está asociada con la rotación de los desplazamientos, mientras que la simétrica, denominada tensor de deformación,  $\mathbf{e} \in \mathfrak{R}^{3 \times 3}$ , expresa la tasa intrínseca de deformación de primer orden que actúa sobre el elemento de materia y representa una aproximación lineal de la deformación local:

$$\mathbf{e} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T) = \frac{1}{2} \begin{bmatrix} 2\frac{\partial d_x}{\partial x} & \frac{\partial d_x}{\partial y} + \frac{\partial d_y}{\partial x} & \frac{\partial d_x}{\partial z} + \frac{\partial d_z}{\partial x} \\ \frac{\partial d_x}{\partial y} + \frac{\partial d_y}{\partial x} & 2\frac{\partial d_y}{\partial y} & \frac{\partial d_y}{\partial z} + \frac{\partial d_z}{\partial y} \\ \frac{\partial d_x}{\partial z} + \frac{\partial d_z}{\partial x} & \frac{\partial d_y}{\partial z} + \frac{\partial d_z}{\partial y} & 2\frac{\partial d_z}{\partial z} \end{bmatrix} \quad (4)$$

#### 3.3. Ley de deformación

El modelo del objeto elástico debe definir la deformación que sufre por la aplicación de una fuerza externa, por lo que se requiere de una expresión que relacione al tensor de restitución con el de deformación, tal como lo hace la ley de Hooke:

$$\mathbf{s} = 2\mathbf{m}\mathbf{e} + \mathbf{l} \text{ traza}(\mathbf{e})\mathbf{I}_3 \quad (5)$$

en donde  $\mathbf{I}_3 \in \mathfrak{R}^{3 \times 3}$  es la matriz identidad,  $\mathbf{l}$  y  $\mathbf{m}$  son los coeficientes de Lamé. A partir de esta expresión es posible calcular la divergencia del tensor  $\mathbf{s}$  para el  $k$ -ésimo elemento de volumen:

$$(\nabla \cdot \mathbf{s})_k = \sum_{i=1}^3 \frac{\partial s_{ik}}{\partial x_i} \quad (6)$$

y sustituirla en la ecuación (2), omitiendo, sin pérdida de generalidad, el término gravitacional para obtener:

$$\mathbf{r}\mathbf{a} = \mathbf{m}\Delta \mathbf{d} + (\mathbf{l} + \mathbf{m})\nabla(\nabla \cdot \mathbf{d}) \quad (7)$$

conocida como ecuación de Navier [10], en donde

$$\Delta d_i = \sum_{k=1}^3 \frac{\partial^2 d_i}{\partial x_k^2} \quad \text{y} \quad \nabla(\nabla \cdot \mathbf{d})_k = \frac{\partial}{\partial x_k} \left( \sum_{i=1}^3 \frac{\partial d_i}{\partial x_i} \right) \quad (8)$$

Esta formulación, debida inicialmente a Lamé, involucra las leyes de deformación/restitución en una ecuación diferencial parcial que ofrece una interpretación alternativa a la ley de Hooke. El primer término de este modelo, ecuación (7), es la ecuación de onda, cuya velocidad de propagación vale  $c = (\mathbf{m}\mathbf{r})^{1/2}$ ; mientras que el segundo es la restricción

de conservación de volumen que indica que el gradiente de la expansión de volumen tenderá a restaurar el volumen inicial. Para diferentes valores de  $I$  y  $m$  la ley de Hooke se puede interpretar como una onda de deformación con mayor o menor compresibilidad. La conservación del volumen del objeto se puede considerar como buena para valores de  $I > 100m$

### 3.4. Método de diferencias finitas

Para representar en pantalla el movimiento de un objeto virtual y para generar las fuerzas que producen ese movimiento, se requiere de un modelo discreto del objeto y de la integración de las ecuaciones diferenciales parciales. Por ello se elige aproximar  $\Delta \mathbf{d}$  mediante el método de las diferencias finitas basada en la definición de la derivada, a saber:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (9)$$

Mientras que, la segunda derivada será entonces:

$$\begin{aligned} f''(x) &= \lim_{\Delta x \rightarrow 0} \frac{f'(x + \Delta x) - f'(x)}{\Delta x} \\ &= \lim_{\Delta x \rightarrow 0} \frac{f(x + 2\Delta x) - 2f(x + \Delta x) + f(x)}{\Delta x^2} \end{aligned} \quad (10)$$

En esta expresión se usa la función evaluada en los puntos equidistantes  $x$ ,  $x + \Delta x$  y  $x + 2\Delta x$ , en los cuales vale  $f_{i-1}$ ,  $f_i$  y  $f_{i+1}$  respectivamente. Usando esta notación, la segunda derivada queda como sigue:

$$\begin{aligned} f''(x) &= \lim_{\Delta x \rightarrow 0} \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} \\ &= \lim_{\Delta x \rightarrow 0} \frac{1}{\Delta x} \left( \frac{f_{i+1} - f_i}{\Delta x} - \frac{f_i - f_{i-1}}{\Delta x} \right) \end{aligned} \quad (11)$$

Si los puntos considerados no son equidistantes, la segunda derivada puede expresarse en términos de las diferencias finitas como lo propone Fornberg [11]:

$$f''(x) = \frac{2}{\Delta_i + \Delta_{i+1}} \left( \frac{f_{i+1} - f_i}{\Delta_i} - \frac{f_i - f_{i-1}}{\Delta_{i+1}} \right) \quad (12)$$

Usando esta representación de la segunda derivada por diferencias finitas, se obtiene una aproximación para la ecuación de onda del  $i$ -ésimo nodo 3D, el cual tiene  $j$  vecinos, dada por:

$$\Delta \mathbf{d}_i = \frac{2}{\sum_j \|\mathbf{x}_i - \mathbf{x}_j\|} \sum_j \frac{\mathbf{d}_j - \mathbf{d}_i}{\|\mathbf{x}_i - \mathbf{x}_j\|} \quad (13)$$

Ahora, para obtener la representación equivalente para la restricción de conservación del volumen se parte del laplaciano, observando que sus componentes se pueden escribir como sigue:

$$\Delta d_i = \frac{\partial}{\partial x_i} \sum_{i=1}^3 \left( \frac{\partial d_i}{\partial x_i} \right) - \left( \frac{\partial}{\partial x_2} \left( \frac{\partial d_2}{\partial x_1} - \frac{\partial d_1}{\partial x_2} \right) - \frac{\partial}{\partial x_3} \left( \frac{\partial d_3}{\partial x_1} - \frac{\partial d_1}{\partial x_3} \right) \right) \quad (14)$$

con lo que la ecuación de onda queda así

$$\Delta \mathbf{d} = \nabla(\nabla \cdot \mathbf{d}) - \nabla \times (\nabla \times \mathbf{d}) \quad (15)$$

como función del término buscado  $\nabla(\nabla \cdot \mathbf{d})$ , que es una medida de la expansión de volumen causada sólo por las componentes normales de la deformación. Las componentes radial y rotacional del vector  $\Delta \mathbf{d}$  se obtienen valiéndose de la descomposición de un vector  $\mathbf{v}$  a lo largo de un vector unitario  $\mathbf{u}$  y otro vector normal a  $\mathbf{u}$  (figura 4) mediante

$$\mathbf{v} = (\mathbf{u} \cdot \mathbf{v})\mathbf{u} - \mathbf{u} \times (\mathbf{u} \times \mathbf{v}) \quad (16)$$

Aplicando esta propiedad a  $\Delta \mathbf{d}$  con el vector unitario  $\mathbf{u} = (\mathbf{x}_i - \mathbf{x}_j) / \|\mathbf{x}_i - \mathbf{x}_j\|$  se obtiene una aproximación para la restricción de conservación del volumen del  $i$ -ésimo nodo 3D, el cual tiene  $j$  vecinos, dada por:

$$\nabla(\nabla \cdot \mathbf{d})_i = \frac{2}{\sum_j \|\mathbf{x}_i - \mathbf{x}_j\|} \sum_j \frac{[(\mathbf{d}_j - \mathbf{d}_i) \cdot \mathbf{u}]\mathbf{u}}{\|\mathbf{x}_i - \mathbf{x}_j\|} \quad (17)$$

Con las ecuaciones (13) y (17) ya se puede calcular la fuerza por unidad de volumen, dada por la ecuación (7), que genera el campo vectorial de deformación  $\mathbf{d}$ . Pero, para actualizar las posiciones de todos los nodos en el MV, se necesita integrar dos veces esta fuerza.

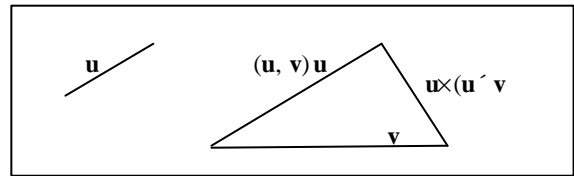


Figura 4. Descomposición del vector  $\mathbf{v}$ .

## 4. Modelo de Cuerpo Continuo

Solución de Delingette *et al* por Integración Explícita [8]

### Elasticidad Lineal

El comportamiento elástico del modelo puede considerarse lineal si los desplazamientos son pequeños; esta aproximación pierde validez conforme los desplazamientos son mayores al 10% del tamaño de una celda de la malla. El modelo lineal es atractivo porque permite calcular cualquier deformación de la malla a partir del conocimiento de un conjunto finito de deformaciones elementales [8].

Lo primero es definir el modelo volumétrico de referencia  $\mathbf{M}_0$  correspondiente a la posición de reposo. Bajo restricciones externas, el modelo  $\mathbf{M}_0$  se deforma. La deformación del modelo, a partir de su posición de referencia se representa usando un vector

de desplazamiento denotado  $\mathbf{d}(x, y, z)$  para el punto  $(x, y, z) \in \mathbf{M}_0$  escribiendo:

$$\mathbf{M}_{def} = \mathbf{M}_0 + \mathbf{d}(x, y, z) \quad (18)$$

El vector de desplazamiento  $\mathbf{d}(x, y, z)$  no caracteriza la deformación del modelo, pues cuando el modelo  $\mathbf{M}$  sufre una translación  $\mathbf{t}$ , el vector de desplazamiento es esa translación:  $\mathbf{d}(x, y, z) = \mathbf{t}$ , pero el modelo no presenta deformación alguna. Una medida de la magnitud de la deformación de un material linealmente elástico  $\mathbf{M}_{def}$  es la energía elástica  $\mathbf{W}_e$  definida, según la ley de Hooke, como:

$$\mathbf{W}_e = \frac{\mathbf{I}}{2} (\text{traza } \mathbf{e})^2 + \mathbf{m} \text{traza } \mathbf{e}^2 \quad (19)$$

donde la matriz simétrica  $\mathbf{e} \in \mathfrak{R}^{3 \times 3}$  es el tensor de deformación mientras que  $\mathbf{I}$  y  $\mathbf{m}$  son los coeficientes de Lamé que caracterizan la dureza del material. Tomando la expresión de  $\mathbf{e}$  dada por la ecuación (4), la ecuación resultante para la energía elástica será:

$$\mathbf{W}_e = \frac{\mathbf{I}}{2} \left( \sum_{i=1}^3 \frac{\partial \mathbf{d}_i}{\partial \mathbf{x}_i} \right)^2 + \mathbf{m} \sum_{i=1}^3 \left( \frac{\partial \mathbf{d}_i}{\partial \mathbf{x}_i} \right)^2 + \frac{\mathbf{m}}{2} \sum_{i,j} \left( \frac{\partial \mathbf{d}_i}{\partial \mathbf{x}_j} + \frac{\partial \mathbf{d}_j}{\partial \mathbf{x}_i} \right)^2$$

en donde  $Q$  significa  $i \neq j$ , con  $i \in [1, 2]$  y  $j \in [2, 3]$ . Para su representación gráfica, Delingette et al construyen el objeto usando una malla de tetraedros, en donde es posible calcular la fuerza  $\mathbf{F}_i$  en el  $i$ -ésimo nodo derivando la energía elástica con respecto a la posición del nodo  $\mathbf{x}_i$ :

$$\mathbf{F}_{\mathbf{x}_i} = \frac{\partial \mathbf{W}_e}{\partial \mathbf{x}_i} = - \sum_{T_j \in L(i)} \frac{\partial \mathbf{W}_e(T_j)}{\partial \mathbf{x}_i}$$

donde  $L(i)$  representa al conjunto de tetraedros adyacentes al vértice  $\mathbf{x}_i$ . Dado que la energía elástica es proporcional al cuadrado del vector de desplazamiento, las fuerzas son funciones lineales de los vectores de desplazamiento para cada nodo. Por tanto, minimizar la energía elástica de un objeto 3D requiere resolver el sistema lineal de la forma:

$$\mathbf{K} \mathbf{d} = \mathbf{F}_{ext} \quad (20)$$

donde  $\mathbf{K}$  es la matriz de rigidez del sistema que representa la topología y la dureza del objeto discreto,  $\mathbf{d}$  representa el desplazamiento de todos los nodos y  $\mathbf{F}_{ext}$  combina las fuerzas externas y las condiciones de frontera. Dentro del tetraedro  $T_i$ , la fuerza  $\mathbf{F}_{Ti(j)}$  aplicada sobre el vértice  $\mathbf{x}_{Ti(j)}$  es:

$$\mathbf{F}_{Ti(j)} = \sum_{k=0}^3 [\mathbf{K}_{jk}^{T_i}] \mathbf{d}_{Ti(k)} \quad (21)$$

Donde, dados los cuatro vértices del tetraedro  $T_i$ :  $\mathbf{x}_{Ti(0)}$ ,  $\mathbf{x}_{Ti(1)}$ ,  $\mathbf{x}_{Ti(2)}$  y  $\mathbf{x}_{Ti(3)}$  los seis tensores  $[\mathbf{K}_{ij}^{T_i}]$  se calculan mediante:

$$[\mathbf{K}_{jk}^{T_i}] = \frac{\mathbf{I}_i \mathbf{M}_k \mathbf{M}_j^T + \mathbf{m}_i \mathbf{M}_j \mathbf{M}_k^T + \mathbf{m}_i (\mathbf{M}_j \mathbf{M}_k) [\mathbf{I}_{3 \times 3}]}{36 V(T_i)}$$

con:

$$\mathbf{M}_0(T_i) = \mathbf{x}_1^0(T_i) \times \mathbf{x}_2^0(T_i) + \mathbf{x}_2^0(T_i) \times \mathbf{x}_3^0(T_i) + \mathbf{x}_3^0(T_i) \times \mathbf{x}_1^0(T_i)$$

Para un vértice dado  $\mathbf{x}_i$ , la fuerza elástica  $\mathbf{F}_i$  es entonces, la suma de todas las contribuciones  $\mathbf{F}_{Ti(j)}$  de todos los tetraedros adyacentes  $T_i$ , de donde la fuerza elástica se expresa como:

$$\mathbf{F}_i = [\mathbf{K}_{ii}] \mathbf{d}_i + \sum_{j \in N(\mathbf{x}_i)} [\mathbf{K}_{ij}] \mathbf{d}_j \quad (22)$$

donde  $[\mathbf{K}_{ii}]$  es la suma de los tensores  $[\mathbf{K}_{ii}^{T_j}]$  asociados a los tetraedros adyacentes al vértice  $\mathbf{x}_i$ ,  $[\mathbf{K}_{ij}]$  es la suma de los tensores  $[\mathbf{K}_{ij}^{T_j}]$  asociados al tetraedro adyacente a la arista  $(i, j)$  y  $N(\mathbf{x}_i)$  es la lista de  $\mathbf{x}_i$  vecinos. Una vez calculada la fuerza  $\mathbf{F}_i$  se resuelve para cada vértice  $\mathbf{x}_i$  la ecuación de movimiento dada por la ley de Newton:

$$m_i \frac{d^2 \mathbf{x}_i}{dt^2} = \mathbf{g}_i \frac{d \mathbf{x}_i}{dt} + \mathbf{F}_i \quad (23)$$

mediante un esquema de integración explícito:

$$\left( \frac{m_i}{\Delta t^2} - \frac{\mathbf{g}_i}{\Delta t} \right) P_i(t + \Delta t) = \mathbf{F}_i + \frac{2m_i}{\Delta t^2} P_i(t) - \left( \frac{m_i}{\Delta t^2} - \frac{\mathbf{g}_i}{\Delta t} \right) P_i(t - \Delta t)$$

ecuación que permite actualizar la posición de cada nodo del modelo.

## 5. Modelo Masa-resorte

Solución de Desbrunne *et al* por integración implícita[9]

Sea un modelo unidimensional en el que cada nodo  $i$  de masa  $m$  y posición  $\mathbf{x}_i$ , está unido a sus vecinos mediante resortes de constante  $k$  (figura 5).



Figura 5. Modelo masa-resorte unidimensional.

Para simular dicho sistema y representarlo gráficamente se puede utilizar el esquema de integración explícito de Euler definido como:

$$Y_{t+\Delta t} = Y_t + \Delta t f(Y_t) \quad (24)$$

La ecuación de movimiento lineal de una partícula o del centro de masa de un cuerpo rígido es:

$$d\mathbf{v} = \frac{\mathbf{F}}{m} dt \quad (25)$$

Si, en vez de usar diferenciales se usan incrementos discretos de tiempo se obtiene una forma aproximada para calcular la velocidad instantánea.

Basándose en la ecuación (24), al inicio de la simulación,  $t_0$ , la condición inicial  $\mathbf{v}_0$  de la partícula define de manera única su velocidad en cada instante:

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \frac{\mathbf{F}_i(n)}{m} \Delta t \quad (26)$$

De manera semejante, puesto que la velocidad es la derivada de la posición respecto al tiempo, al tomar incrementos finitos de tiempo se obtiene la siguiente aproximación, cuya condición inicial es  $\mathbf{x}_i(0)$  [3]:

$$\mathbf{x}_i(n+1) = \mathbf{x}_i(n) + \Delta t \mathbf{v}_i(n+1) \quad (27)$$

Otro esquema, denominado integración implícita de Euler, ha demostrado adaptarse mejor al problema antes enunciado. En lugar de suponer, como lo hace el método anterior, que la derivada es una función de las condiciones iniciales,  $dY/dt = f(Y_0)$ , supóngase que es un promedio ponderado de  $f(Y_0)$ , la derivada al principio del intervalo, y  $f(Y_{t+\Delta t})$  su valor al final del intervalo, entonces [14]:

$$Y_{t+\Delta t} = Y_t + \Delta t \left[ (1-I) f(Y_t) + I f(Y_{t+\Delta t}) \right] \quad (28)$$

donde  $I \in [0, 1]$  es una constante; cuando  $I=0$  se tiene la expresión de la ecuación (24). Cualquier actualización en la que aparezca el término  $f(Y_{t+\Delta t})$  se conoce como fórmula de actualización implícita. Ahora, cuando  $I=1$  se tiene la fórmula de actualización implícita hacia atrás de Euler. Si el estado actual se denota  $f(Y_t) = f(Y_0)$  y el estado final del intervalo se denota  $f(Y_{t+\Delta t}) = f(Y)$ , se tiene que la ecuación (28) queda como :

$$Y = Y_0 + \Delta t \left[ f(Y_0) + I (f(Y) - f(Y_0)) \right] \quad (29)$$

sustituyendo la aproximación en serie de Taylor de primer orden de la derivada alrededor del estado actual  $Y_0$ , dada por:

$$f(Y) \approx f(Y_0) + (Y - Y_0) (\nabla f_{Y=Y_0}) \quad (30)$$

en la ecuación (29) se tiene:

$$f(Y_0) = \Delta Y \left[ \frac{1}{\Delta t} I - I \nabla f_{Y=Y_0} \right] \quad (31)$$

Dados sus valores iniciales, se quiere calcular la posición y la velocidad de cada nodo al instante  $t_0 + \Delta t$ . De acuerdo con el método implícito hacia atrás de Euler se tiene que [1]:

$$\begin{aligned} \Delta \mathbf{x} &= \Delta t (\mathbf{v}_0 + \Delta \mathbf{v}) \\ \Delta \mathbf{v} &= \frac{\Delta t}{m} f(\mathbf{x}_0 + \Delta \mathbf{x}, \mathbf{v}_0 + \Delta \mathbf{v}) \end{aligned} \quad (32)$$

Puesto que el sistema de ecuaciones anterior es no lineal, se aplica la expansión en serie de Taylor a  $f$  y se hace la siguiente aproximación de primer orden:

$$f(\mathbf{x}_0 + \Delta \mathbf{x}, \mathbf{v}_0 + \Delta \mathbf{v}) = f_0 + \frac{\partial f}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial f}{\partial \mathbf{v}} \Delta \mathbf{v} \quad (33)$$

en donde las derivadas parciales se evalúan con las condiciones iniciales  $(\mathbf{x}_0, \mathbf{v}_0)$ . Sustituyendo esta expresión en el sistema de ecuaciones (32) resulta:

$$\begin{aligned} \Delta \mathbf{x} &= \Delta t (\mathbf{v}_0 + \Delta \mathbf{v}) \\ \Delta \mathbf{v} &= \frac{\Delta t}{m} \left( f_0 + \frac{\partial f}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial f}{\partial \mathbf{v}} \Delta \mathbf{v} \right) \end{aligned} \quad (34)$$

Escribiendo la segunda ecuación de este sistema en términos de  $\Delta \mathbf{v}$  se obtiene:

$$\Delta \mathbf{v} = \frac{\Delta t}{m} \left( f_0 + \frac{\partial f}{\partial \mathbf{x}} \Delta t (\mathbf{v}_0 + \Delta \mathbf{v}) + \frac{\partial f}{\partial \mathbf{v}} \Delta \mathbf{v} \right) \quad (35)$$

y reagrupando:

$$\left( I - \frac{\Delta t^2}{m} \frac{\partial f}{\partial \mathbf{x}} - \frac{\Delta t}{m} \frac{\partial f}{\partial \mathbf{v}} \right) \Delta \mathbf{v} = \frac{\Delta t}{m} \left( f_0 + \Delta t \frac{\partial f}{\partial \mathbf{x}} \mathbf{v}_0 \right) \quad (36)$$

entonces, el intervalo hacia atrás de Euler consiste en evaluar  $f_0$ ,  $\partial f / \partial \mathbf{x}$  y  $\partial f / \partial \mathbf{v}$ , resolver el sistema para  $\Delta \mathbf{v}$  y, finalmente, actualizar los valores de  $\mathbf{x}$  y  $\mathbf{v}$ .

## 6. Cálculo de Colisiones

En un MV con objetos en movimiento existe la posibilidad de que se penetren, comportamiento que indeseado en la simulación de mundos reales. Entonces se necesita un algoritmo que, primero, detecte la colisión y, luego, calcule la respuesta correspondiente (deformación visual y fuerza táctil). La colisión es un problema cinético que involucra las posiciones de los objetos, mientras que el cálculo de la respuesta es un problema dinámico [17]. En el caso particular del ciclo háptico, lo que se quiere es detectar el contacto entre el dispositivo háptico y el objeto virtual para calcular las fuerzas de interacción y la deformación producida [2].

En la detección de contacto generalmente se utiliza primero un algoritmo aproximado en donde los objetos se consideran representados por una caja. En el algoritmo AABB (*axis aligned bounding box*) las caras de la caja son paralelas a los ejes del referencial mientras que en el algoritmo OBB (*oriented bonding box*) la caja se alinea con el principal eje de inercia del objeto. Ambos hacen una detección temprana aunque imprecisa del contacto, pero puede refinarse usando un método exacto como el siguiente [13].

Moore y Wilhelms [17], proponen un algoritmo para detectar el contacto en el caso de superficies flexibles cuya representación gráfica se construye usando una malla formada por triángulos. La prueba de contacto entre superficies consiste en verificar la

penetración de cada vértice del primer objeto a través de los planos de cualquier triángulo del segundo objeto y viceversa (penetración de los vértices del segundo en los triángulos del primero) en los extremos de un intervalo de tiempo dado. Inicialmente ambos objetos están separados. Se comparan las posiciones de los vértices al principio y al final de cada intervalo para ver si alguno de ellos atravesó un triángulo durante dicho intervalo; en cuyo caso ha ocurrido una colisión. La complejidad del algoritmo es  $O(n \times m)$  para  $n$  triángulos y  $m$  vértices.

#### Caso I: Superficie fija, vértice en movimiento

El vértice analizado sigue una trayectoria que inicia en  $P$  y termina en  $P'$ , dos puntos 3D; la recta paramétrica que pasa por esos puntos y que determina la posición  $P(t)$  del vértice en el instante  $t \in [0, 1]$  está definida como [16]:

$$P(t) = (1-t)P + tP' = P + (P'-P)t \quad (37)$$

Ahora, para saber si un vértice en movimiento ha intersectado la superficie de un triángulo que no se mueve, se necesita averiguar si un punto de la recta dada por la ecuación (37) es solución de la ecuación paramétrica que define a dicho triángulo. Entonces, igualando la ecuación paramétrica de la recta dada por la ecuación (37) con la ecuación paramétrica del triángulo cuyos vértices son  $P_i, i = 0, 1, 2$ , se obtiene:

$$P + (P'-P)t = P_0 + (P_1 - P_0)u + (P_2 - P_0)v \quad (38)$$

Esta ecuación se resuelve para las variables paramétricas  $t, u$  y  $v$ ; las dos últimas son las variables paramétricas definidas por el triángulo, mientras que  $t$  es la variable temporal que al inicio del intervalo de simulación vale 0 y 1 al final. El miembro izquierdo es la ecuación paramétrica de la trayectoria del vértice y el miembro derecho es la ecuación paramétrica de cualquier punto del plano correspondiente al triángulo analizado. Esta ecuación vectorial representa tres ecuaciones escalares con tres incógnitas y se resuelve invirtiendo una matriz. Si  $0 \leq t \leq 1, u \geq 0, v \geq 0$  y  $u+v \leq 1$ , entonces el vértice ha intersectado al plano en la parte correspondiente al triángulo durante el intervalo de tiempo en cuestión.

#### Caso II: Superficie y vértice en movimiento

Para saber si un vértice en movimiento ha intersectado la superficie de un triángulo que también se mueve, se utiliza la ecuación paramétrica:

$$P + Vt = P_0 + V_0t + ((P_1 - P_0) + (V_1 - V_0)t)u + ((P_2 - P_0) - (V_2 - V_0)t)v \quad (39)$$

donde  $P$  es la posición del vértice que se mueve con velocidad  $V$ ,  $P_i, i = 0, 1, 2$ , define los vértices del

triángulo cuyas velocidades se denotan  $V_i$ , mientras que  $t, u$  y  $v$  son las variables paramétricas definidas anteriormente. De nuevo se cuenta con una ecuación vectorial que puede descomponerse en tres ecuaciones escalares que, al ser expresadas en términos de la variable  $t$ , se obtiene un polinomio de quinto grado que se resuelve de manera aproximada para  $t$  usando una técnica binaria de búsqueda.

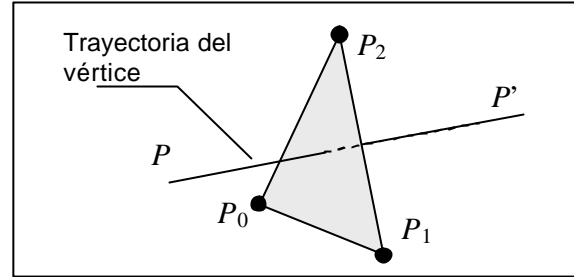


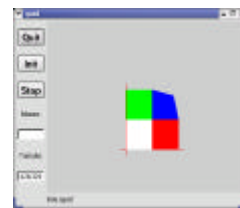
Figura 6. Colisión entre plano (triángulo) y un vértice.

## 7. Resultados Experimentales

Se desarrolló un sistema de simulación basado en el método propuesto por Debunne et al [6] sobre una PC AMD Athlon (tm) XP 1900+ a 1.59GHz con el sistema operativo RedHat 8.0, el lenguaje de programación C++ y con el apoyo de bibliotecas Qt para la creación de la interfaz gráfica del usuario y OpenGL para la construcción de objetos 3D.

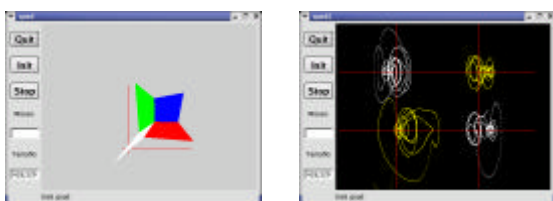
Se desarrollaron las clases: Vector.h y Vector.cpp que contienen la implementación de las operaciones entre vectores (suma, resta, producto punto, producto cruz, etc.); así como quad.h y quad.cpp que incluyen los datos para la construcción del objeto simulado (los ejes coordenados 3D y el cuadro unitario con uno de sus vértices en el origen); painter.h y painter.cpp sirven para colocar los objetos en la escena y darle funcionalidad a los botones de que consta la aplicación; y, finalmente, la clase main.cpp que hace las veces de programa principal.

Se simuló un cuadrado 2D en un mundo virtual 3D, cuya parte geométrica se inspira en el cubo desarrollado por Fraga [7] en el contexto de su curso de graficación. La condición inicial de este objeto es una deformación en su esquina noreste (NE) y  $v_0 = 0$  (figura 7). Se considera una situación de ingravidez,  $g = 0$ , y un objeto con densidad  $\mathbf{r} = 1.0$  y coeficientes de Lamé  $\mathbf{m} = 0.01, \mathbf{I} = 1.0$ .



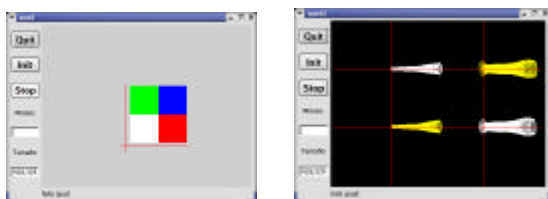
**Figura 7.** Estado inicial

**Caso A.** La deformación inicial es de  $-0.1$  unidades en los ejes  $X$  y  $Y$  (fig alfa). La simulación se dispara con el botón *Init* y se deja correr con  $\Delta t = 0.01$  y  $\text{timer} = 190$  durante 3,710 iteraciones. El estado del objeto al final de la simulación y el retrato de fase de las 4 esquinas (SO, SE, NO, NE) del cuadrado simulado se muestran en la figura 8. La evolución del retrato de fase permite ver que al principio de la simulación sólo se desplaza el vértice NE inicialmente *deformado*, el cual induce el movimiento de sus vecinos los vértices NO y SE y, finalmente, el del vértice SO, momento en que el movimiento de todas las esquinas aumenta rápidamente hasta alcanzar niveles no deseados.



**Fig. 8.** Estado final y retrato de fase de esquinas (caso A)

**Caso B** La deformación inicial es de  $-0.05$  unidades a lo largo de los ejes  $X$  y  $Y$ . La simulación incluye el término  $-0.2 v_0$  para simular la fricción viscosa, usa  $\Delta t = 0.04$  y  $\text{timer} = 190$  durante 9,472 iteraciones. La figura 9 muestra el objeto al final de la simulación y la evolución del retrato de fase de sus vértices, observando que el sistema en todo momento tiende a conservar su forma cuadrada, aunque se desplaza en el plano  $XY$ .



**Fig. 9.** Estado final y retrato de fase de esquinas (caso B)

## 8. Conclusiones

Aunque se han presentado tres métodos para calcular la posición y velocidad de los elementos de un objeto elástico, sólo se ha simulado el método DeBunne, pues para los demás aún no se tiene una representación adecuada, aunque estamos trabajando para obtenerlas. Cuando podamos comparar los resultados de estos dos métodos con los aquí reportados, estaremos en medida de elegir la que tenga el mejor desempeño e implementar con ella objetos elásticos 3D.

En el método simulado [6], los parámetros  $(m, l, r)$  se eligieron arbitrariamente respetando tan sólo las recomendaciones de DeBunne ( $l > 100 m$  y  $m > 0$ ). En un futuro deberán ajustarse tomando en consideración materiales reales para poder evaluar mejor el simulador desarrollado. Al incluir amortiguación y por tratarse de deformaciones más pequeñas que las consideradas en el caso A, el comportamiento del objeto en el caso B es más verosímil y permite simulaciones más largas y estables. El siguiente paso consistirá en refinar la malla asociada al objeto 2D y observar su comportamiento elástico, el cual se espera que sea más apegado a la realidad.

Para la realización del simulador realista se pudo haber utilizado software tipo Matlab, Simnon, etc. muchos de los cuales tienen interfaces para conectarse de manera bidireccional a mundos virtuales pero suelen ser sistemas consumidores de tiempo y recursos lo que los hace muy lentos como para pretender aplicaciones en tiempo real. Mejor se optó por desarrollar un sistema propio que demanda una representación en diferencias finitas de toda ecuación dinámica usada para calcular el comportamiento de movimiento y de deformación de los objetos virtuales. Esto nos lleva a buscar las aproximaciones adecuadas para cada modelo y cada tipo de solución planteada.

## Agradecimientos

Este trabajo es parte del Proyecto HAPTICS realizado conjuntamente entre la UAS y el CINVESTAV con el apoyo del programa PROMEP. También se cuenta con apoyo del Consorcio RIBERO-PARTI del programa iberoamericano CYTED; en particular de los doctores R. Aracil de I DISAM-UPM y L. Basañez del IOC-UPC.

## Referencias

- [1] Baraff David, Witkin Andrew. "Large Steps in Cloth Simulation". Computer Graphics (proc. SIGGRAPH) (1998) 43-54.
- [2] Basdogan, Cagatay, Ho Chih-Hao. "Principles of Haptic Rendering for Virtual Environments". <http://eis.jpl.nasa.gov/~basdogan/Tutorials/haptic-tutorial.htm>
- [3] Bourg D. M. "Physics for game developers". Pekín: O'Reilly (2002) 326 pp.
- [4] Burdea, G. C. "Haptics Issues in virtual Environments". IEEE. 2000.



- [5] Ballard, D. H., Brown, C. M. “*Computer Vision*.” Prentice Hall, Inc. Englewood Cliffs, NJ, USA. 1982
- [6] Debunne G., Desbrun M., Barr A., Cani M. P. “*Interactive multiresolution animation of deformable models*” in Eurographics Workshop on Computer Animation and Simulation '99 (Sept 99) 7 pp.
- [7] De la Fraga. *Curso de Graficación*. Sec. Computación del Depto. de Ing. Eléctrica, CINVESTAV. <http://delta.cs.cinvestav.mx/~fraga/Cursos/graficacion2002/index.html>),
- [8] Delingette H., Cotin S., Ayache N.. “*A Hybrid Elastic Model Allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation*”. Electronic Edition. IEEE Computer Society DL (1999) 70-81.
- [9] Desbrun M., Schröder P., Barr A. “*Interactive Animation of Structured Deformable Object*”. in Graphics Interface '99 (June 99, Kingston, Canada) 8 pp.
- [10] Doghri, I. “*Mechanics of Deformable Solids*”. Springer-Verlag, Berlín (2000) 579 pp.
- [11] Fornberg B. “*Generation of finite difference formulas on arbitrarily spaced grids*”. Math. Comput., 51:699-706, 1988.
- [12] Hannaford, B; Ryu J.H.; Kim, Y.S. ‘*Stable Control Haptics*’ en ‘*Touch in Virtual Environments. Haptics and the design of interactive Systems*’. Ed. McLaughlin, M.L.; Hespanha, J.P.; Sukhatme, G.S. New Jersey: Imcs. p. 47– 70. 2001.
- [13] Ibarra Zannatha, J.M., Marmolejo Rivas C. “*Realistic Computer Simulations Based on Visual and Force Feedback*” Proc. of the CLACA 2002
- [14] Kass, M. “*An introduction to Physically Based Modeling: An Introduction to continuum Dynamics for Computer Graphics*”. Pixar (1997) 12 pp.
- [15] Landau, L.D., Lifshitz, E.M. “*Theory of Elasticity*”. Pergamon Press, Oxford. Tercera Edición en inglés. 1986.
- [16] Lengyel, E. “*Mathematics for 3D game programming & Computer Graphics*”. Ed. Charles Rivers Media (2002)
- [17] Moore, M.; Wilhelm, J. “*Collision Detection and Response for Computer Animation*”. Computer Graphics, [22] 4 (1988)