

SIMULACIÓN DE MODELOS DEFORMABLES 3D BASADOS EN PARTÍCULAS

Simulation of 3D Deformable Models based on Particles

DEYBI EXPOSITO¹ y MÁXIMO MERO²

Universidad de Carabobo. Facultad de Ciencia y Tecnología. CEMVICC

¹Departamento de Computación. ²Departamento de Matemáticas

Carabobo. Venezuela

{dexposito, mmero}@uc.edu.ve

Fecha de Recepción: 26/02/2007, **Fecha de Revisión:** 13/07/2007, **Fecha de Aceptación:** 01/11/2007

Resumen

El estudio de la deformación de un objeto bajo ciertas condiciones y/o restricciones es de sumo interés en diversas áreas del conocimiento. Existen múltiples aplicaciones en computación gráfica de modelos matemáticos que son empleados para realizar el estudio de deformaciones. Estas aplicaciones van desde las deformaciones basadas en modelos matemáticos que solo usan restricciones geométricas hasta las deformaciones en las cuales se incorporan propiedades físicas al modelo en estudio. En este trabajo se presenta la simulación de la deformación de objetos volumétricos basados en propiedades de material, empleando el método de mallas libres de los Mínimos Cuadrados (MLS) para resolver las ecuaciones que surgen de la teoría de la elasticidad lineal.

Palabras Claves: Elasticidad Lineal, MLS, Mallas Libres.

Abstract

The study of the deformation of an object under certain conditions and/or restrictions is of large interest in diverse areas of the knowledge. Multiple applications in graphical computation of mathematical models exist that are used to make the study of deformations. These applications go from the deformations based on mathematical models that single use geometric restrictions until the deformations in which physical properties to the model in study are gotten up. In this work the simulation of the deformation of volumetric objects based on material properties appears, using the method of free meshes of Moving Least-Squares (MLS) to solve the equations that arise from the theory of the linear elasticity.

Keywords: Linear Elasticity, MLS, Mesh-free.

1. Introducción

Los modelos deformables son esquemas teóricos que permiten representar las deformaciones que ocurren en un objeto a través del tiempo. En Computación Gráfica y Animación estos modelos se han empleado en la simulación de tela, expresiones faciales, segmentación de imágenes, tejido de órganos humanos, etc.

Los modelos deformables se pueden clasificar según se incorporen propiedades “no físicas” y “físicas”, (Gibson & Mirtinch, 1997). Los primeros, son empleados para crear y modificar curvas complejas, superficies y sólidos. Estos modelos no utilizan propiedades físicas del material, sólo se necesitan ecuaciones implícitas y/o explícitas para la deformación del objeto. Dentro de estos modelos se pueden mencionar los *splines*, parches paramétricos (*parametric patches*) y deformación libre (*Free-form deformation*)

Los modelos basados en propiedades físicas son aquellos que permiten una representación más realista del objeto a ser deformado. En estos modelos las ecuaciones que definen la deformación son representadas mediante ecuaciones diferenciales. La solución de estas ecuaciones se realiza empleando diferentes métodos, entre los cuales cabe mencionar: Masa Muelle (*Mass-spring models*), Elementos Finitos (*Finite Element*), Volumen Finito (*Volumetric Element*), Diferencia Finita (*Finite Difference*), Mallas Libres (*Mesh Free*).

La presente investigación tiene como meta diseñar e implementar una aplicación para la simulación de deformaciones basadas en propiedades físicas en un objeto 3D, utilizando el método de mallas de libres MLS.

Este artículo está organizado como sigue. En la sección 2 se presentan comentarios de trabajos previos y se da una clasificación breve de modelos deformables. El modelo físico-matemático de este trabajo se muestra en la sección 3. En la sección 4 se explicará la dinámica del modelo de partículas y de la superficie del objeto.

En la sección 5 se presentarán los resultados obtenidos. Finalmente, en la sección 6, se mencionarán las conclusiones obtenidas y los posi-

bles trabajos futuros a esta investigación.

2. Trabajos Relacionados

En el área de Computación Gráfica y Animación se han realizado muchas investigaciones sobre los modelos deformables. En el año 1987, Terzopoulos y otros (Terzopoulos *et al.*, 1987), en su publicación titulada “*Elastically Deformable Models*”, utiliza la teoría de la elasticidad para construir las ecuaciones diferenciales que modelan el comportamiento de curvas, superficies y sólidos no rígidos.

Los métodos de Mallas Libres son aquellos donde el objeto es modelado por medio de una nube de partículas sin conectividad previa. Fries (Fries & Hermann-Georg, 2004) expone una diversidad de métodos de mallas libres específicos, entre los cuales menciona: SPH (*Smoothed Particle Hydrodynamics*) (Lucy, 1977), DEM (*Method Element Difuse*), EFG (*Element Free Galerkin*), Mínimos Cuadrados (*Moving Least-Squares*), LBIE (*Local Boundary Integral Equation*), PUM (*Partition of Unity Method*), entre otros.

Desbrun y Gascuel en su trabajo titulado “*Smoothed Particles: A new paradigm for animating highly deformable bodies*” (Desbrun & Gascuel, 1996), presentan una técnica para la simulación de cuerpos con grandes deformaciones dentro de un sistema de partículas. También, ellos hablan sobre simulación de fluidos para la animación de cuerpos no elásticos con un amplio rango de rigidez y viscosidad.

En el trabajo titulado “*Point Based Animation of Elastic, Plastic and Melting Objects*”, realizado por Müller y otros (Müller *et al.*, 2004), se presenta un método para modelar y animar una amplia gama de objetos volumétricos con propiedades de material que van desde los elásticos-rígidos hasta los altamente plásticos. El movimiento de estas partículas es realizado por medio de esquemas de integración del tiempo de forma explícita.

Investigaciones similares en el campo de los modelos deformables se muestran en: “*Mesh Methods: An Overview and Recent Develop-*

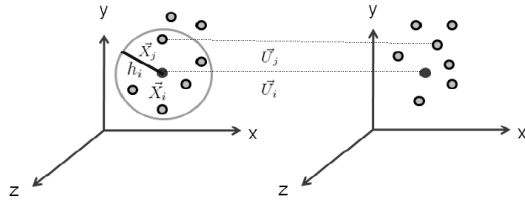


Fig. 1. Partícula i con sus respectivos vecinos. La figura del lado izquierdo muestra el objeto sin deformaciones. La figura del lado derecho representa un objeto con su deformación ($X + U$). Donde X es el conjunto de partículas que representan las coordenadas del material y $X + U$ son las coordenadas de las partículas del objeto deformado.

ment” (Fleming *et al.*, 1996), y “*Physically Based Deformable Models in Computer Graphics*” (Müller *et al.*, 2006).

3. Teoría de Elasticidad Lineal

A continuación se muestra el modelo físico-matemático que surge de la teoría de elasticidad lineal y cuyas leyes son las que gobiernan el modelo deformable en estudio, (Cook *et al.*, 1995).

Considérese un objeto 3D cuyas coordenadas de material son: $\vec{X}_i = [x_i \ y_i \ z_i]^T$. Para describir la deformación de dicho objeto se utiliza un campo vectorial de desplazamiento: $\vec{U}_i = [u_i \ v_i \ w_i]^T$, donde $u_i = u(x_i, y_i, z_i)$, $v_i = v(x_i, y_i, z_i)$ y $w_i = w(x_i, y_i, z_i)$ son funciones de las coordenadas de material. En la Fig. 1 se puede ver que las coordenadas de una partícula originalmente se encuentran en \vec{X}_i y en su posición deformada son $\vec{X}_i + \vec{U}_i$. Se denota por \vec{X}_{ij} el vector diferencia entre los vectores posición de las partículas i y j .

$$\vec{X}_{ij} = \vec{X}_j - \vec{X}_i \quad (1)$$

En el análisis de elasticidad lineal el Jacobiano J de una transformación es fundamental. Éste se expresa para cada partícula i como:

$$J_i = \nabla \vec{U}_i^T + I \quad (2)$$

donde I es la matriz identidad 3×3 . Por comodidad en la escritura, el Jacobiano se escribe de la

siguiente manera:

$$J_i = \begin{bmatrix} J u_i^T \\ J v_i^T \\ J w_i^T \end{bmatrix} \quad (3)$$

Para medir el grado de deformación (*strain*) de una partícula i se empleó el tensor de Green-Saint-Venant el cual está definido por la siguiente matriz:

$$\varepsilon_i = (J_i^T J_i) - I \quad (4)$$

De la teoría de elasticidad lineal se tiene que el grado de deformación ε y la tensión (*stress*) σ están linealmente relacionados, es decir:

$$\sigma_i = C \varepsilon_i \quad (5)$$

donde C es un tensor de rango cuatro relacionado con la ley constitutiva de los materiales y tanto ε como σ son tensores simétricos. Para materiales isotrópicos C tiene sólo dos coeficientes independientes llamados módulo de Young y tasa de Poisson.

Para calcular *stress*, *strain* y fuerzas elásticas es necesario calcular el campo de desplazamiento $\nabla \vec{U}$. Para ello se hace uso del método de mallas libres MLS siguiendo las pautas del trabajo de Müller (Müller *et al.*, 2004). En este caso, las derivadas en una partícula i son estimadas a partir de sus vecinos.

3.1. Aproximación de $\nabla \vec{U}$ empleando el método de MLS

El cálculo de las derivadas parciales $\nabla u_i, \nabla v_i, \nabla w_i$ del vector desplazamiento \vec{U}_i de la partícula i , siguiendo el método de mallas libres MLS, se muestran a continuación.

Haciendo uso de la expansión de Taylor en cada una de las coordenadas de \vec{U}_i , midiendo el error de aproximación promediado con el núcleo polinomial de soporte compacto, se tiene que las derivadas que minimizan este error satisfacen, para ∇u_i , lo siguiente:

$$\sum_j \vec{X}_{ij} \vec{X}_{ij}^T W_{ij} \nabla u_i = \sum_j (u_j - u_i) \vec{X}_{ij} W_{ij} \quad (6)$$

La expresión

$$A_i = \sum_j \left(\vec{X}_{ij} \vec{X}_{ij}^T W_{ij} \right), \quad (7)$$

es conocida como la matriz de momentos.

Mediante estas ecuaciones:

$$\nabla u_i = A_i^{-1} \sum_j \left((u_j - u_i) \vec{X}_{ij} W_{ij} \right) \quad (8)$$

De forma similar se obtiene un resultado para $\nabla v_i, \nabla w_i$. Se deduce entonces que $\nabla \vec{U}_i^T$ es:

$$\nabla \vec{U}_i^T = \begin{bmatrix} \nabla u_i^T \\ \nabla v_i^T \\ \nabla w_i^T \end{bmatrix} \quad (9)$$

Para calcular la matriz A^{-1} se utilizó el algoritmo de Descomposición en Valores Singulares (SVD - *Singular Values Decomposition*) tomado de (Teukolsky *et al.*, 1992).

3.2. Funciones de Peso

El cálculo de la masa, densidad y volumen de una partícula i se hace utilizando una estrategia similar a la usada con el método SPH. La masa de una partícula es fija y se calcula como:

$$m_i = s \bar{r}_i^3 \rho \quad (10)$$

donde s es un factor de escala igual para todas las partículas, ρ es la densidad del material y \bar{r}_i es la distancia promedio de las k partículas más cercanas a la partícula i . La masa m_i esta distribuida alrededor de un núcleo polinomial ó función de peso. En este proyecto se usan los siguientes núcleos polinomiales, obtenidos de (Müller *et al.*, 2004; Fries & Hermann-Georg, 2007).

$$W_1(r, h) = \begin{cases} \frac{315}{64\pi h^9} (h^2 - r^2)^3 & \text{si } r < h \\ 0 & \text{en caso contrario} \end{cases} \quad (11)$$

$$W_2(r, h) = \begin{cases} e^{-\left(\frac{r}{0.4h}\right)^2} & \text{si } \frac{r}{h} \leq 1 \\ 0 & \text{si } \frac{r}{h} > 1 \end{cases} \quad (12)$$

$$W_3(r, h) = \begin{cases} \frac{2}{3} - 4\left(\frac{r}{h}\right)^2 + 4\left(\frac{r}{h}\right)^3 & \text{si } \frac{r}{h} \leq \frac{1}{2} \\ \frac{4}{3} - 4\left(\frac{r}{h}\right) + 4\left(\frac{r}{h}\right)^2 - \frac{4}{3}\left(\frac{r}{h}\right)^3 & \text{si } \frac{1}{2} < \frac{r}{h} \leq 1 \\ 0 & \text{si } \frac{r}{h} > 1 \end{cases} \quad (13)$$

$$W_4(r, h) = \begin{cases} 1 - 6\left(\frac{r}{h}\right)^2 + 8\left(\frac{r}{h}\right)^3 - 3\left(\frac{r}{h}\right)^4 & \text{si } \frac{r}{h} \leq 1 \\ 0 & \text{si } \frac{r}{h} > 1 \end{cases} \quad (14)$$

$$W_5(r, h) = \begin{cases} \frac{2}{3h} \left(1 - \frac{3}{2}\left(\frac{r}{h}\right)^2 + \frac{3}{4}\left(\frac{r}{h}\right)^3\right) & \text{si } \frac{r}{h} \leq 1 \\ \frac{2}{3h} \left(\frac{1}{4}\left(2 - \frac{r}{h}\right)^3\right) & \text{si } 1 < \frac{r}{h} \leq 2 \\ 0 & \text{si } \frac{r}{h} > 2 \end{cases} \quad (15)$$

de estas ecuaciones se define W_{ij} por:

$$W_{ij} = W(\|\vec{X}_{ij}\|_2, h_i), \quad (16)$$

siendo $\|\vec{X}\|_2$ la norma euclídea. Las gráficas de estos núcleos se pueden observar en las Fig. 2, 3 y 4.

El índice j representa la partícula vecina de la partícula i . Las funciones asociadas a estos núcleos relacionan una partícula con sus vecinos, donde r es la distancia entre la partícula i y j , y h es el radio de soporte de estos núcleos.

La esfera centrada en la coordenada de material de la partícula i tiene como radio de soporte: $h_i = 3\bar{r}_i$, donde \bar{r}_i es la distancia promedio entre la partícula i y sus vecinos. La densidad de una partícula i viene dada por $\rho_i = \sum_j m_j W_{ij}$ y el volumen de una partícula i se obtiene de la forma siguiente $vol_i = m_i \div \rho_i$

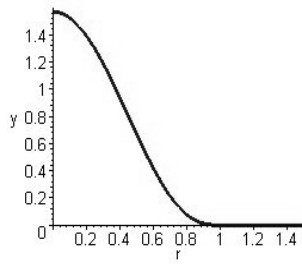


Fig. 2. Función $W_1(r, h)$.

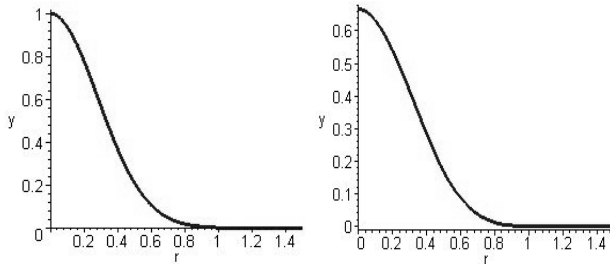


Fig. 3. Función $W_2(r, h)$ (izquierda) y Función $W_3(r, h)$ (derecha).

3.3. Cálculo de las Fuerzas

Las fuerzas internas están formadas por fuerzas elásticas, fuerzas de conservación del volumen y fuerzas de Lennard Jones. Las primeras son descritas de la siguiente manera:

$$F_{e_i} = (2vol_i J_i \sigma_i) A_i^{-1} \sum_j (\vec{X}_{ij} W_{ij}) \quad (17)$$

estas fuerzas conservan momento angular y lineal. Las fuerzas de conservación del volumen se definen como:

$$F_{v_i} = vol_i kv (|J_i| - 1) \varphi_i A_i^{-1} \sum_j (\vec{X}_{ij} W_{ij}) \quad (18)$$

donde kv es una constante de conservación del volumen que penaliza la desviación del Jacobiano y

$$\varphi_i = \begin{bmatrix} (Jv_i \times Jw_i)^T \\ (Jw_i \times Ju_i)^T \\ (Ju_i \times Jv_i)^T \end{bmatrix} \quad (19)$$

Las fuerzas de Lennard Jones (F_{l_i}) (Lennard-Jones, 1931) son fuerzas de atracción

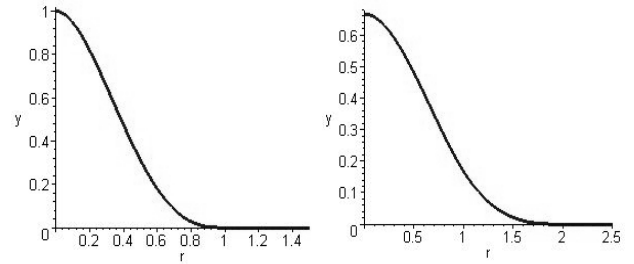


Fig. 4. Función $W_4(r, h)$ (izquierda) y Función $W_5(r, h)$ (derecha).

y repulsión entre partículas. Para cada partícula i esta fuerza es representada por:

$$F_{l_i} = \sum_j L(\alpha, d_{ij}) \vec{p}_{ij} \quad (20)$$

donde $L(\alpha, d_{ij}) = 4\beta \left(12 \frac{\alpha^{12}}{d_{ij}^{13}} - 6 \frac{\alpha^6}{d_{ij}^7} \right)$, β es el mínimo del potencial de energía de Lennard Jones, d_{ij} es la distancia entre la partícula i y j ($d_{ij} = \|\vec{X}_i - \vec{X}_j\|_2$), $\vec{p}_{ij} = \frac{1}{\|\vec{X}_i - \vec{X}_j\|_2} (\vec{X}_i - \vec{X}_j)$ y α es el diámetro de la esfera que representa la partícula i . En la Fig. 5 se puede observar que h_i define el radio de la esfera de la vecindad de la partícula i y α va asociado sólo a la partícula i .

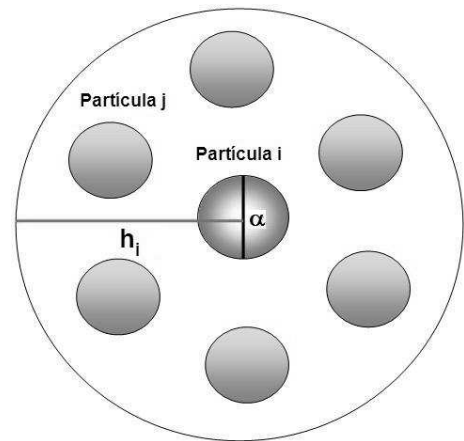


Fig. 5. Diferencia entre h_i y α .

En la Fig. 6 se puede ver la gráfica de la función $L(\alpha, d_{ij})$.

Por lo tanto, las fuerzas internas de una partícula i son representadas por la siguiente expresión:

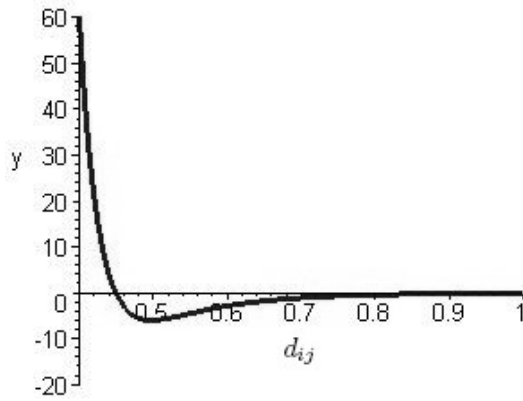


Fig. 6. Función de Lennard Jones $L(\alpha, d_{ij})$.

$$F_{int_i} = (F_{e_i} + F_{v_i})A_i^{-1} \sum_j (\vec{X}_{ij}W_{ij}) + F_{l_i} \quad (21)$$

A las fuerzas internas se le pueden agregar fuerzas externas tales como: gravedad, presión atmosférica entre otras. La suma de las fuerzas internas y externas es la fuerza total en el modelo.

4. Dinámica

La dinámica del modelo de partículas se basa en las leyes de Newton y para ello fue necesario utilizar un esquema de integración explícito de las fuerzas totales. Así, de la ecuación

$$f_i^{t+\Delta t} = m_i a_i^{t+\Delta t} \quad (22)$$

donde $f_i^{t+\Delta t}$ es la fuerza total, $a_i^{t+\Delta t}$ es la aceleración y m_i la masa de la partícula i , se deducen tanto la velocidad

$$v_i^{t+\Delta t} = v_i^t + a_i^{t+\Delta t} \Delta t \quad (23)$$

como la posición de una partícula i

$$x_i^{t+\Delta t} = x_i^t + v_i^{t+\Delta t} \Delta t. \quad (24)$$

4.1. Dinámica de la Superficie del objeto 3D

El esquema utilizado en este trabajo para representar la superficie del objeto en deformación se basa en el utilizado en el trabajo de Pauly

y otros (Pauly *et al.*, 2003). Una vez calculadas las nuevas posiciones, velocidades y desplazamientos de las partículas internas se procede a determinar el desplazamiento de las partículas de la superficie. El desplazamiento de una partícula de la superficie con respecto a las partículas internas vecinas se obtiene de la siguiente manera:

$$u_{surf} = \frac{1}{\sum \omega(r_i)} \sum_i \omega(r_i)(u_i + \nabla u_i^T(x_{surf} - x_i)) \quad (25)$$

donde x_{surf} es la posición de la partícula de la superficie, $\omega(r_i) = \frac{1}{\|x_{surf} - x_i\|_2} = e^{-\frac{r_i^2}{h^2}}$ es una función de peso gaussiano y h es la distancia del radio de la esfera que contiene a la partícula de la superficie y sus partículas internas vecinas.

Una vez calculado el nuevo desplazamiento de la partícula de la superficie se procede a obtener su nueva posición, la cual se determina de la siguiente manera:

$$x_{surf}^{t+\Delta t} = x_{surf}^t + u_{surf} \quad (26)$$

5. Resultados

La aplicación fue corrida en varios sistemas operativos (Windows, Linux y Solaris). Los equipos utilizados para las animaciones se muestran en la tabla 1

Equipo	Características
1	Computador AMD Athlon 64 de 2.20 GHZ con 512 MB de memoria RAM
2	Laptop Toshiba Satellite Mobile Intel Pentium 4, 2.80 GHZ y 752 MB de memoria RAM

Tabla 1. Equipos utilizados.

La construcción de la aplicación permite al usuario realizar cambios en las propiedades físicas (Modulo de Young, tasa de Poisson, Gravedad, etc) con la finalidad de simular la deformación de un objeto 3D. En la Fig. 7 se puede observar la interfaz gráfica, donde el usuario puede cargar un objeto, modificar sus propiedades físicas y simular su deformación.

Para construir esta aplicación se utilizó el lenguaje de programación C++, y las librerías de GTK y OpenGL para el lenguaje seleccionado. Los compiladores usados fueron: Visual C++, G++ y CC.

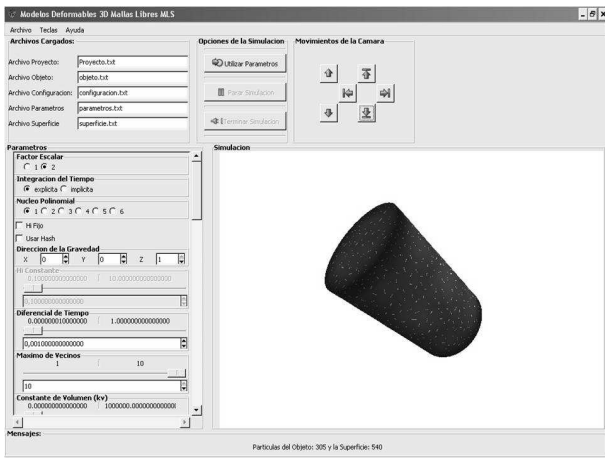


Fig. 7. Interfaz Gráfica de la aplicación.

En la tabla 2 se muestra la información de cada objeto simulado con la aplicación. La primera animación se realizó sobre el objeto 1 que cae dentro de una caja por efecto de la fuerza de gravedad. Las imágenes de la animación se muestran en la Fig. 8.

Objeto	Características
1	Octaedro formado por 299 partículas internas.
2	Elipsoide formado por 568 partículas internas.
3	Cilindro formado por 305 partículas internas y 540 de la superficie y 1076 triángulos.

Tabla 2. Objetos simulados.

La segunda animación se realizó sobre el objeto 2 que cae por fuerzas de gravedad dentro de un cilindro. Las imágenes de la animación se muestran en la Fig. 9.

En la Fig. 10 se muestra la animación del objeto 3 deformándose dentro de otro cilindro. En la tabla 3 se muestran los FPS (Frames por segundo) y TPIS (tiempo promedio por iteración en segundos) por objetos y equipos usados.

En las figuras de las animaciones presentadas se puede notar que los objetos se adaptan al recipiente que los contienen. En las tablas

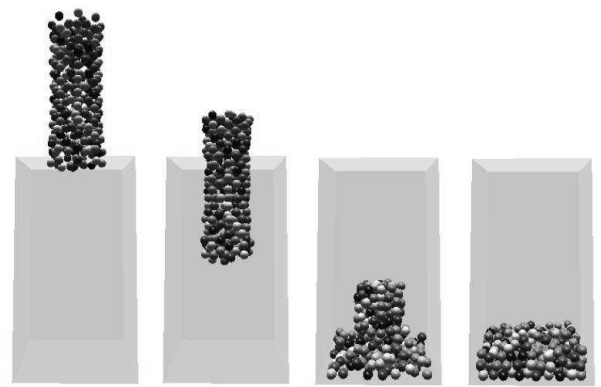


Fig. 8. Animación de un Octaedro con 299 partículas cayendo dentro de una caja.

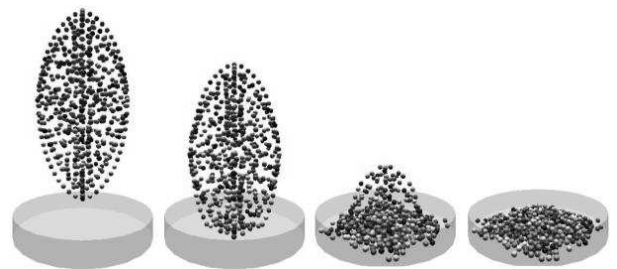


Fig. 9. Animación de un Elipsoide con 568 partículas cayendo dentro de un cilindro.

mostradas se puede notar que cuando aumenta la cantidad de partículas de un objeto los FPS van reduciéndose y el TPIS va aumentando, lo cual hace que no sea posible la ejecución de la animación en tiempo real.

6. Conclusiones y Trabajos Futuros

Los modelos deformables basados en propiedades físicas son aquellos que mejor se ajustan a la realidad. Así, el método basado en partículas MLS es uno de los más apropiados.

Objeto	FPS	TPIS	Equipo
1	7,2944	0,09926	1
1	5,9943	0,1320	2
2	1,2751	0,5123	1
2	1,029	0,6683	2
3	4,2731	0,1872	1
3	3,5186	0,2331	2

Tabla 3. FPS y TPIS de la animaciones realizadas en los distintos equipos en Windows.

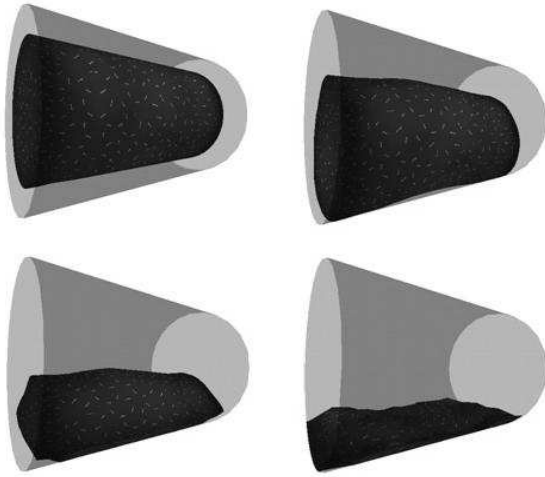


Fig. 10. Animación de un Cilindro con 305 partículas internas, 540 partículas de la superficie y 1076 triángulos dentro de un cilindro.

dos para la simulación de cuerpos 3D altamente elásticos, pero a medida que se aumenta la cantidad de partículas se incrementa la demanda de cálculo y esto indica que la tasa de cuadros por segundo (FPS) disminuye, haciendo que la simulación dinámica no se ejecute en tiempo real.

La construcción de la aplicación gráfica permitió la interacción con el objeto a ser deformado de modo tal que se modificaron, en línea, las diferentes propiedades físicas del mismo.

Este trabajo sirve para que se sigan desarrollando estudios que optimicen tanto la interacción de la interfaz con el usuario así como el tiempo de ejecución del modelo. En este sentido se pueden buscar estrategias que minimicen el tiempo de búsqueda de vecinos de las partículas. Mejorar la técnica que se ajuste mejor a la superficie del objeto volumétrico. Implementar modelos que sean de utilidad en otras áreas de investigación.

7. Bibliografía

Cook, R., D. Malkus & M. Plesha. (1995). *Finite element modeling for stress analysis*. John Wiley & Sons. New York.

Desbrun, M. & M. Gascuel. (1996). Smoothed Particles: A new paradigm for animating highly

deformable bodies. Proceedings of the Eurographics workshop on Computer animation and simulation '96. Poitiers. France. 61–76.

Fleming, M., D. Organ, T. Belytschko, & Y. Krongauz. (1996). Meshless methods: An overview and recent development. *Computer Methods in Applied Mechanics and Engineering*. 139(1):3-47.

Fries, T. & M. Hermann-Georg. (2004). Classification and overview of meshfree methods. Tech. Rep. Informatikbericht-Nr. 2003-03. Institute of Scientific Computing. Technical University Braunschweig. Braunschweig. Germany.

Gibson, S. & B. Mirtinch. (1997). A survey of deformable modeling in computer graphics. *Tech. Report No. TR-97-19*.

Lennard-Jones, J. (1931). Cohesion. Proceedings of the Physical Society. Bristol. England. 43(5): 461-482.

Lucy, L. (1977). Numerical approach to the testing of the fission hypothesis. *Astronomical Journal*. 82(12): 1013–1024.

Müller, M., R. Keiser, A. Nealen, M. Pauly, M. Gross & M. Alexa. (2004). Point based animation of elastic, plastic and melting objects. Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation. Grenoble. France. 141–151.

Müller, M., R. Keiser, E. Boxerman, M. Carlson, & A. Nealen. (2006). Physically based deformable models in computer graphics. *Computer Graphics Forum*. 25(4): 809–836.

Pauly, M, R. Keiser, L. Kobbelt & M. Gross. (2003). Shape modeling with point-sampled geometry. International Conference on Computer Graphics and Interactive Techniques. ACM SIGGRAPH 2003 Papers. San Diego, USA. 641-650.

Terzopoulos, D., J. Platt, A. Barr & K. Fleischer.

(1987). Elastically deformable models. SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques. New York. USA. 205–214.

Teukolsky, S., W. Press, W. Vetterling & B. Flannery. (1992). *Numerical recipes in C: The art of scientific computing*. Cambridge University Press. New York.