

UN MÉTODO PARA LA INTERACCIÓN DE FLUIDO Y OBJETOS FLOTANTES PARA APLICACIONES DE TIEMPO REAL

Marcos G. Lazo^a, Cristian D. García Bauza^a, Gustavo A. Boroni^a, Javier A. Dottori^a y
Alejandro Clause^{a,b}

^aCONICET-Universidad Nacional del Centro, 7000 Tandil, Argentina

^bCNEA, Libertador 8250, Buenos Aires, Argentina

*mlazo@conicet.gov.ar; cgarcia@exa.unicen.edu.ar; gboroni@conicet.gov.ar;
jdottori@exa.unicen.edu.ar; clause@exa.unicen.edu.ar*

Palabras claves: Interacción fluido-objeto, Lattice-Boltzmann, tiempo real, motores físicos, shallow waters.

Resumen. Se presenta un método para resolver la interacción entre fluido y objetos flotantes para aplicaciones de tiempo real. La simulación del fluido se realiza con el método de Lattice-Boltzmann para aguas superficiales el cual ha demostrado su factibilidad para simulaciones en tiempo real aplicables a animación gráfica. Para los objetos flotantes se utiliza un motor de cálculo que simula la física newtoniana, que toma como entrada un modelo de fuerzas y calcula el comportamiento de cuerpos sólidos en base a las leyes de la mecánica clásica. Las fuerzas sobre los objetos se calculan en base a la presión estática en la zona sumergida, y las fuerzas de lift y de arrastre. Para introducir perturbaciones en el fluido generadas por el movimiento de los objetos, se propuso un modelo que estima las fuerzas generadas por la frontera. Se presentan animaciones del movimiento de embarcaciones, que demostraron una performance adecuada para lograr tasas aceptables de cuadros por segundo.

1 INTRODUCCIÓN

Las primeras animaciones gráficas de superficies de fluidos no estaban basadas en leyes físicas, sino en representaciones reducidas como las series de Fourier en dos dimensiones o familias de funciones similares (Schachter, 1980; Masten et al. 1987; Tso et al. 1987). Si bien es posible producir visualizaciones suficientemente realistas, la animación de efectos similares a la interacción con sólidos es muy limitada. Por otro lado, las superficies de líquidos pueden, en principio, ser modeladas mediante las ecuaciones diferenciales parciales de fluidos, pero están muy limitadas por problemas de estabilidad y el costo computacional de una solución numérica. En este sentido, fueron propuestos diferentes métodos que resuelven el conjunto completo de ecuaciones tridimensionales de Navier-Stokes para animar superficies líquidas interactuando con paredes sólidas (Foster and Metaxas, 1996; Stam, 1999; Layton et al. 2002; y Thürey et al. 2007). Takahashi et al. (2002) voxelizó los objetos flotantes para definir las condiciones de contorno para el modelo del fluido. Génevaux et al. (2003) uso una formulación de Euler del fluido acoplado con objetos flotantes representados como sistemas de resortes y puntos de masas. Müller et al. (2004) propuso un método de representación de objetos sólidos deformables mediante elementos finitos calculando el fluido a través de *smoothed-particle-hydrodynamics* (SPH). Irving et al. (2006) presentó una técnica para simular grandes masas de agua interactuando con sólidos usando una combinación de un solver de Navier-Stokes y mapas de alturas. Losasso et al. (2006) desarrolló una representación gráfica para múltiples fluidos interactuando. Wendt et al. (2006) presentó un método para flujos 2D y 3D usando una formulación de volúmenes finitos para capturar las condiciones de borde evitando los problemas de las grillas cartesianas. Mihalef et al. (2007) propuso un método para la simulación de fluidos que permite interacciones en un solo sentido con objetos sólidos y deformables. Robinson-Mosher et al. (2008) presentó un modelo de interacción entre fluidos y cuerpos rígidos delgados. La calidad de estos resultados es muy buena, pero desafortunadamente el costo computacional se opone a su aplicación en escenarios de interactividad en tiempo real.

Kim et al. (2006) implementó en GPU un algoritmo basado en imágenes para calcular las fuerzas de flotabilidad que actúan sobre objetos sólidos arbitrarios. Este enfoque logra buena performance para aplicaciones de tiempo real, sin embargo el movimiento del objeto no afecta al fluido. Harada et al. (2007) presentó un modelo completo de interacción fluido-estructura en tiempo real basado en seguimiento de partículas implementado en GPU. En este caso, los sólidos son representados como fluidos a los que se les impone condiciones de contorno deformables. Yuksel et al. (2007) presentó un método para simular la interacción de sólidos con ondas propagadas en un entorno elástico, capaz de lograr altas tasas de cuadros por segundo. Cords (2008) propuso un modelo de sólidos simulados por conjuntos de partículas que interactúan con una ecuación de superficie de onda bidimensional. En Cords et al. (2009) el modelo fue extendido para permitir la animación de grandes escenarios usando una aproximación de grilla móvil otorgando mayor nivel de detalle en la zona cercana al observador.

Una aproximación simple, que mientras conserva las ecuaciones de transporte del fluido, mantiene el costo computacional bajo control reduciendo la complejidad matemática del problema a dos dimensiones, son las ecuaciones de aguas poco profundas (shallow-waters, SW) (Kass et al. 1990; Chen et al. 1995; O'Brien et al. 1995; Thon et al. 2001; Neyret et al. 2001; Foster et al. 2001; Rinaldi et al. 2012b). En el presente trabajo se representan las ecuaciones de SW mediante un autómata celular de Lattice-Boltzmann (García Bauza et al. 2010), el cual es acoplado con objetos sólidos y estáticos, para producir animaciones interactivas de escenarios náuticos en tiempo real. La Figura 1 muestra tres capturas de la

animación en tiempo real de una lancha a motor que pasa cerca de una línea de boyas. La animación representa los efectos que pueden ser creados con el presente método. La hélice del barco es simulada introduciendo en el mismo una fuerza que lo empuja hacia adelante. Puede verse que las olas creadas por el paso del barco afectan las posiciones de las boyas.



Figura 1. Capturas de una animación en tiempo real de un barco y boyas.

2 MODELO

El modelo de animación propuesto está compuesto por un modelo de aguas superficiales basado en un autómata de Lattice-Boltzmann (LB) y modelos de objetos sólidos representados por mallas triangulares que se mueven mediante un motor físico newtoniano. Ambos modelos son acoplados, es decir, el fluido ejerce fuerzas estáticas y dinámicas sobre los objetos (fuerzas de flotabilidad), y los objetos transfieren momento al fluido (ver [Figura 2](#)).

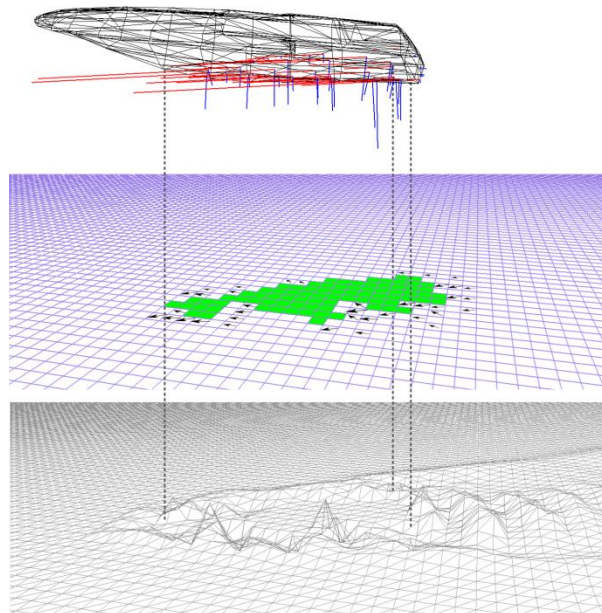


Figura 2. Malla triangular representando el bote y las fuerzas de arrastre y flotabilidad (arriba), la reacción de las fuerzas ejercidas por el objeto en las celdas de fluido (centro), y el movimiento resultante de la superficie del fluido (abajo).

2.1 Modelo de aguas superficiales y LB

Las ecuaciones de SW son un conjunto de ecuaciones diferenciales que describen el flujo de un fluido sobre una superficie con presión. Estas son derivadas de integrar las ecuaciones de Navier-Stokes en el caso de que la escala horizontal sea mucho más grande que la escala

vertical. Bajo esta condición, la conservación de la masa implica que los efectos en el sentido vertical son despreciables respecto de los horizontales bajo una buena aproximación. Las ecuaciones son:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + g \frac{\partial h}{\partial x} = 0$$

donde $u = u(x, t)$ es la velocidad del fluido en la posición x en el instante t , $h = h(x, t)$ es la altura del fluido en dicho punto. Por simplicidad, se supone la superficie del fondo del fluido plana.

Para resolver el problema anterior utilizamos el método de LB con una distribución de equilibrio adecuada para recuperar las ecuaciones de SW (Zhou, 2004).

Entre las aplicaciones de los autómatas de LB se destacan las simulaciones de Higuera and Succi (1989) de flujos alrededor de obstáculos cilíndricos, Foti and Succi (1989) en simulaciones de flujos tridimensionales sobre geometrías complejas, los experimentos en medios permeables de Cancelliere et al. (1990), la simulación de alta resolución de Succi et al. (1988) de gases turbulentos, y la aplicación de Chen et al. (1991) al estudio de la interacción de campos magnéticos y fluidos conductivos como plasmas o metales líquidos.

Otras aplicaciones destacadas son la realizadas por Filippova and Haenel (1997) en la simulación de flujos de partículas gaseosas a través de filtros, la utilización en simulaciones con obstáculos de bordes curvos de Mei et al. (1999), y la aplicación de fenómenos de convección y difusión en embalajes de productos agrícolas de van der Sman (1999).

Muchos autores han producido animaciones de aguas usando el método de LB para las ecuaciones de Navier-Stokes (Thürey, 2007; Thürey and Rüde, 2004; Li et al. 2005; Uoane, 2011). En este trabajo, la superficie del fluido es modelada por medio de un esquema de LB para aguas poco profundas, el cual opera en una grilla espacial regular \vec{x} y un conjunto de nueve vectores de velocidad \vec{v}_α que son usados como una representación interna mesoscópica del fluido, la cual es:

$$\vec{v}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\vec{v}_1 = \begin{pmatrix} v \\ 0 \end{pmatrix} \quad \vec{v}_2 = \begin{pmatrix} v \\ -v \end{pmatrix} \quad \vec{v}_3 = \begin{pmatrix} 0 \\ -v \end{pmatrix} \quad \vec{v}_4 = \begin{pmatrix} -v \\ -v \end{pmatrix}$$

$$\vec{v}_5 = \begin{pmatrix} -v \\ 0 \end{pmatrix} \quad \vec{v}_6 = \begin{pmatrix} -v \\ v \end{pmatrix} \quad \vec{v}_7 = \begin{pmatrix} 0 \\ v \end{pmatrix} \quad \vec{v}_8 = \begin{pmatrix} v \\ v \end{pmatrix}$$

donde $v = \Delta x / \Delta t$ es una unidad de velocidad característica dada por el factor de relación entre las discretizaciones de las celdas y el tiempo. Cada celda de la grilla es completada con pseudo-partículas mesoscópicas, y el estado de las mismas representadas por una función de densidad $f_\alpha = f_\alpha(\vec{x}, t)$ que representa la cantidad de pseudo-partículas en el nodo \vec{x} en el tiempo t moviéndose con velocidad \vec{v}_α . Las variables macroscópicas observables son variables generadas por los momentos de $f_\alpha = f_\alpha(\vec{x}, t)$ respecto de \vec{v}_α , llamadas:

$$h(\vec{x}, t) = h_0 \sum_{\alpha} f_{\alpha}(\vec{x}, t) \quad \text{Altura de la superficie del agua}$$

$$\vec{u}(x, t) = \frac{\sum_{\alpha} \vec{v}_{\alpha} f_{\alpha}(\vec{x}, t)}{h(\vec{x}, t)} \quad \text{Velocidad promedio}$$

donde h_0 es la altura efectiva aportada por cada pseudo-partícula.

El estado de cada celda cambia de acuerdo a un esquema de reglas explícitas, el cual es (Zhou, 2004):

$$f_{\alpha}(\vec{x} + \vec{v}_{\alpha} \Delta t, t + \Delta t) = f_{\alpha}(\vec{x}, t) - \frac{1}{\tau} [f_{\alpha}(\vec{x}, t) - f_{\alpha}^{eq}(\vec{x}, t)] + \frac{\Delta t}{6v^2} \vec{v}_{\alpha} \cdot \vec{F}^f \quad (1)$$

donde \vec{F}^f es la fuerza que actúa sobre el fluido la cual se define en la sección 2.3, τ es un tiempo de relajación que controla la viscosidad, y $f_{\alpha}^{eq}(\vec{x}, t)$ es una función de equilibrio, la cual para las ecuaciones de SW está dada por (Zhou, 2004):

$$f_o^{eq} = h \left(1 - \frac{5gh}{6v^2} - \frac{2u^2}{3v^2} \right)$$

$$f_{\alpha}^{eq} = w_{\alpha} h \left(\frac{gh}{6v^2} + \frac{\vec{v}_{\alpha} \cdot \vec{u}}{3v^2} + \frac{(\vec{v}_{\alpha} \cdot \vec{u})^2}{2v^4} - \frac{u^2}{6v^2} \right), \quad \alpha = 1, \dots, 8$$

donde:

$$w_{\alpha} = \begin{cases} 1, & \alpha = 1, 3, 5, 7 \\ \frac{1}{4}, & \alpha = 2, 4, 6, 8 \end{cases} \quad (2)$$

donde u es el módulo de la velocidad \vec{u} .

2.2 Interacción fluido-a-objeto

El comportamiento de un objeto inmerso en un medio líquido bajo la acción de la gravedad está determinado por fuerzas estáticas y dinámicas (Figura 3). La fuerza hidrostática está dada por la integración de la presión local actuando normal a la superficie del objeto en contacto con el fluido. En el modelo propuesto, la superficie externa de cada objeto es representada por una malla de triángulos y la fuerza de flotabilidad es particionada en componentes individuales que actúan sobre cada triángulo de la superficie, quedando determinada como:

$$\vec{F}_i^b = -\rho g h_i A_i \vec{n}_i \quad (3)$$

donde \vec{n}_i y \vec{F}_i^b son el vector normal unitario del triángulo i -ésimo y la fuerza de flotabilidad actuante en su área mojada A_i , h_i es el vector distancia entre el nivel del líquido y el centroide de la parte sumergida del triángulo correspondiente, ρ es la densidad del líquido y g la aceleración de la gravedad.

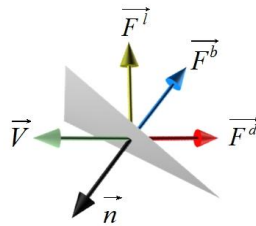


Figura 3. Componentes de fuerza actuando sobre un triángulo de la superficie del objeto.

Las componentes dinámicas son la fuerza de *drag* y *lift* que dependen de la velocidad del objeto relativa al fluido. La fuerza drag es opuesta al movimiento del objeto mientras que la fuerza de lift actúa en dirección normal a la velocidad del objeto. Yuksel et al. (2007) propone el siguiente modelo para calcular dichas fuerzas:

$$\vec{F}_i^d = -\frac{1}{2} \rho C_d A_i^e |\vec{V}| \vec{V} \quad (4)$$

$$\vec{F}_i^l = -\frac{1}{2} \rho C_l A_i^e |\vec{V}| \vec{V} \times \frac{\vec{n}_i \times \vec{V}}{|\vec{n}_i \times \vec{V}|} \quad (5)$$

Donde \vec{V} es la velocidad relativa al líquido, C_d y C_l son los coeficientes de *drag* y *lift*, y A_i^e es el área efectiva del triángulo y está dada por:

$$A_i^e = \left[\frac{\vec{n}_i \cdot \vec{V}}{|\vec{V}|} \alpha + (\alpha - 1) \right] A_i \quad (6)$$

donde α es un parámetro definido por el usuario ($0 \leq \alpha \leq 1$) y se utiliza para ajustar efectos de forma y orientación del objeto (Yuksel et al. 2007). Si toma valores pequeños puede ser utilizado para amortiguar el movimiento del objeto.

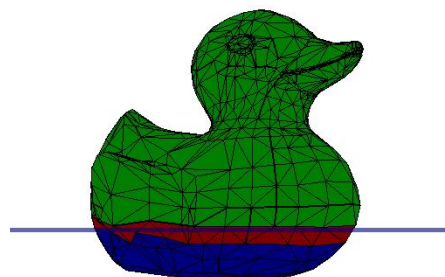


Figura 4. Clasificación de triángulos de acuerdo a la posición respecto del nivel del líquido: totalmente sumergido (azul), totalmente seco (verde) o semi-sumergido (rojo).

Físicamente la fuerza de drag representa el arrastre producido por procesos irreversibles (fricción, turbulencia, etc.); y la fuerza de lift representa la presión dinámica. Los coeficientes C_d y C_l , en este trabajo, por simplicidad se consideran constantes, pero en principio pueden depender de las propiedades del fluido.

Dependiendo de la posición respecto al nivel del líquido, cada triángulo se puede clasificar

en tres categorías: totalmente sumergido, totalmente seco o semi-sumergido (Figura 4). Los triángulos totalmente secos o sumergidos no presentan grandes dificultades en el cálculo de las fuerzas. Sin embargo, el cálculo de las fuerzas que actúan sobre los triángulos semi-sumergidos requiere cierta atención. El cálculo geométrico exacto de la zona húmeda y el centroide correspondiente, dado por la intersección de la malla triangular del objeto y la grilla de la superficie del líquido, consumiría demasiado tiempo de cálculo para aplicaciones de tiempo real. Por lo tanto, se propone una solución aproximada cortando el triángulo horizontalmente a la altura del nivel del líquido en la coordenada proyectada del centroide del triángulo, obtenido por interpolación de los nodos más cercanos.

Como se puede ver en la Figura 5, la superficie sumergida resultante puede ser un triángulo o un cuadrilátero. En el primer caso, el cálculo se realiza sobre el triángulo sumergido como en el caso de los triángulos totalmente sumergidos. Si la región es un cuadrilátero, también se pueden calcular el área y el centroide, pero es más conveniente preservar el triángulo como primitiva por razones de compatibilidad. Por lo tanto, el cuadrilátero se divide en dos triángulos cortando a través de sus diagonales (Figura 5 derecha). Con el fin de minimizar los errores numéricos en la división de triángulos, la malla triangular del objeto debe ser lo más regular posible, con triángulos aproximadamente equiláteros (D'Amato et al. 2012).

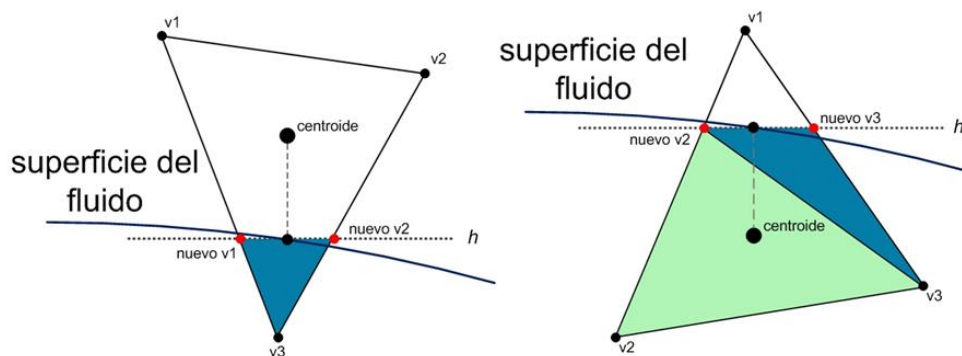


Figura 5. Generación de nuevos triángulos sumergidos a partir de semi-sumergidos.

La Figura 6 muestra las fuerzas que actúan sobre una pelota flotante y un bote en movimiento. Sólo las fuerzas estáticas actúan sobre la pelota ya que está en reposo. A su vez, como el barco está en movimiento, además se aplican las fuerzas de drag y lift. Las fuerzas que actúan sobre los objetos se implementan por medio del motor de física de propósito general *Newton Game Dynamics* el cual representa la mecánica de cuerpos rígidos y proporciona funcionalidad para aplicar fuerzas y velocidades, resolviendo colisiones y fuerzas de reacción teniendo en cuenta las propiedades de los materiales que lo componen. En nuestro caso, cada cuerpo rígido está representado por una malla de colisión que puede ser descrita por una primitiva simple (por ejemplo, una esfera) o una primitiva compleja (por ejemplo, un convex hull). Además, cada cuerpo tiene atributos físicos como la masa, el tensor de inercia, centro de gravedad y propiedades específicas del material que lo compone como los coeficientes de fricción y elasticidad. El motor de física fue utilizado a través de una capa de abstracción (García Bauza et al. 2008), que aísla al usuario de las características específicas del motor en particular, facilitando así el desarrollo de aplicaciones. Igualmente, el modelo de fuerzas aquí presentado puede ser implementado utilizando cualquier otro motor físico similar (por ejemplo, Millington 2010).

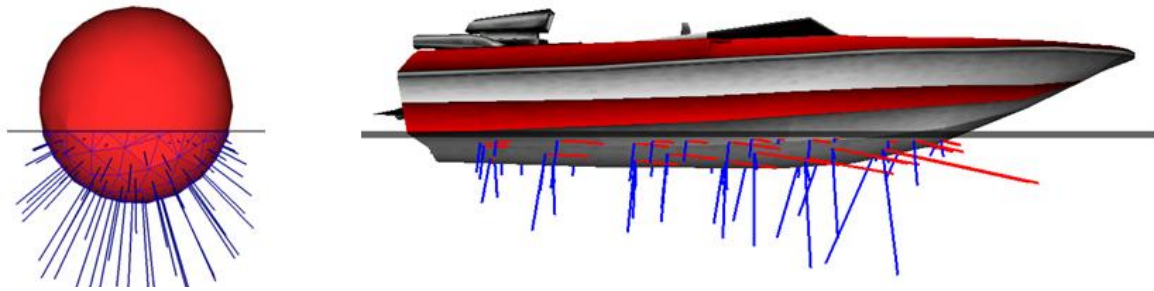


Figura 6. Fuerzas actuando sobre una pelota en reposo y sobre un bote en movimiento (Flechas azules: fuerza estática y *lift*, flechas rojas: fuerzas de *drag*).

2.3 Interacción objeto-a-fluido

El movimiento de un objeto en una superficie líquida genera perturbaciones a través de la transferencia de momento, como las olas detrás de los buques avanzando o las ondas circulares que se expanden hacia afuera desde el punto de contacto de una gota de lluvia en un estanque. En el modelo propuesto, esta situación es simulada por medio de fuerzas que actúan sobre el fluido de LB en los límites de los objetos.

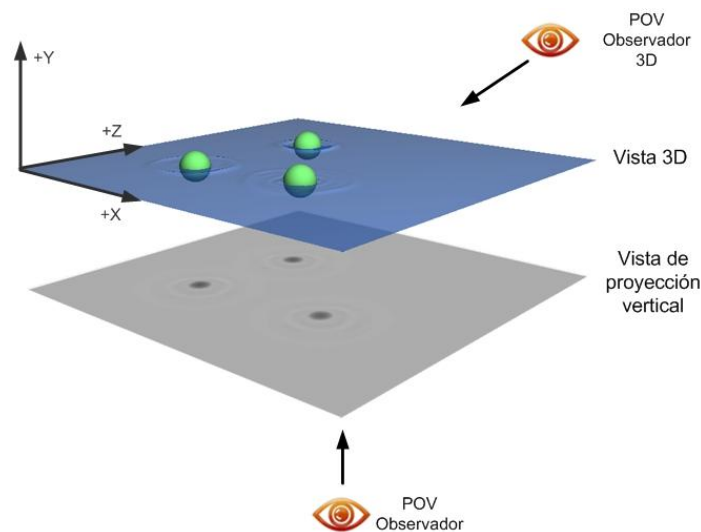


Figura 7. Configuración de las vistas usando el *depth-buffer* de OpenGL.

La sección sumergida de los objetos es calculada en cada paso de tiempo usando la información del *depth-buffer* de la placa gráfica a través de la biblioteca OpenGL (Woo et al. 1999). El método es similar al propuesto por Kim et al. (2006) y Crane et al. (2007). El *depth-buffer* almacena la distancia entre el observador y el punto más cercano de un objeto de interés para el rayo que pasa por cada píxel. De esta manera, posicionando el punto de vista del observador perpendicular al plano horizontal como se muestra en la Figura 7, se puede obtener el mapa de profundidad de la superficie del agua (Figura 8, izquierda) y de los objetos flotantes (Figura 8, centro). Al comparar los dos mapas, se puede obtener una máscara binaria correspondiente a la sección sumergida de los objetos (Figura 8, derecha) la cual se utiliza como condición de contorno para el motor de cálculo del fluido para calcular las fuerzas que actúan sobre el agua. Adicionalmente, se etiqueta cada objeto al fin de poder identificarlos. Determinar la posición del observador es importante, porque si está demasiado cerca de la escena algunos objetos pueden quedar fuera de la vista, mientras que si está demasiado lejos

se reduce la precisión para determinar los bordes.



Figura 8. Mapa de profundidad de la superficie de agua (izq.) y los objetos flotantes (centro), máscara binaria mostrando la frontera mojada (der.).

Una vez que se determina la sección sumergida de cada objeto, se calcula la fuerza ejercida por cada celda sobre la grilla que representa al fluido como:

$$\vec{F}^f = \begin{cases} k_+ (\vec{V} \cdot \vec{n}) \vec{n}, & (\vec{V} \cdot \vec{n}) > 0 \\ k_- (\vec{V} \cdot \vec{n}) \vec{n}, & (\vec{V} \cdot \vec{n}) < 0 \end{cases} \quad (7)$$

donde \vec{V} es la velocidad del objeto en la celda, \vec{n} es una estimación del vector normal al borde, y k_+ y k_- son coeficientes constantes que pueden ser usados para controlar distintos efectos de animación. El efecto físico de k_+ y k_- es desplazar el fluido con mayor o menor intensidad cuando el objeto empuja (k_+) o arrastra (k_-). Se pueden obtener efectos con resultados realistas manteniendo $k_+/k_- \approx 10$. La Ec. (7) provee la expresión usada en la Ec. (1) para introducir momento en el fluido.

El vector \vec{n} de cada celda de la máscara es calculado sumando todos los vectores que tienen origen en el centro de la celda y apuntan a las celdas adyacentes de agua. Por ejemplo, el vector normal de la celda roja en la Figura 9 (izquierda) se estima como:

$$\vec{N} = (-1,0) + (-1,1) + (0,1) + (1,1) = (-1,3) \quad (8)$$

Luego de normalizar:

$$\vec{n} = (-1/\sqrt{10}, 3/\sqrt{10}) \quad (9)$$

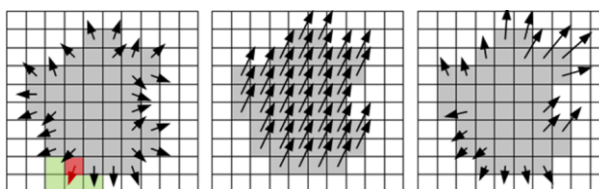


Figura 9. Vectores unitarios \vec{n} representando la normal al perímetro mojado (izquierda), campo de velocidades \vec{V} de las celdas objetos en la superficie plana (centro), campo de fuerzas \vec{g} actuando en el perímetro del fluido (derecha).

3 RESULTADOS

Para verificar la consistencia y estabilidad del método propuesto, se realizaron diferentes casos de prueba involucrando la animación de distintos escenarios con objetos flotantes. Adicionalmente para evaluar la performance y escalabilidad, se realizaron simulaciones con diferente número de objetos en una PC estándar (3.3 GHz Intel i5 con GeForce GTX 560).

3.1 Distribución de peso

La escena de prueba consiste de un bote cargado con un conjunto de objetos pesados. El bote es representado con una malla de 2628 triángulos (500 triángulos/m²). Las dimensiones del bote son 3.23 metros de eslora, 0.43 de puntal y 1.23 de manga. Para simular el peso de los objetos, se aplica una fuerza de 90 kg. sobre el bote en el lugar de contacto de los objetos. La Figura 10 muestra como varía la línea de flotabilidad dependiendo de la posición de los objetos, hundiendo el bote hacia el lado más cargado o balanceándolo si la carga se ubica en el centro. En la misma imagen, se representan las fuerzas de presión ejercidas por el agua sobre la base de la embarcación, mostrando las regiones donde la estructura se encuentra más comprometida.



Figura 10. Distribución de peso. Se muestra la presión que ejerce el fluido sobre el bote.

3.2 Posición del centro de masa

Esta prueba muestra la interacción de un velero de 29 toneladas con un tren de olas de 4 metros de altura. Las dimensiones del barco son de 19 metros de eslora, 2.30 metros de puntal y 4.60 metros de manga. Las olas son simuladas aplicando la siguiente ecuación a los vértices de la malla que representa la superficie del fluido:

$$H(x,t) = A \left| \sin \frac{2\pi}{L}(x+ct) \right|^n \quad (10)$$

donde A es la altura de los picos de las olas, L es la distancia entre picos, x es la coordenada correspondiente espacial, c es el módulo de la velocidad de las olas, n es el exponente de forma y t la coordenada del tiempo. La Figura 11 muestra dos secuencias de capturas que muestran la respuesta del velero a la trayectoria de una ola. En la primer secuencia (arriba) el centro de masa se encuentra localizado 0.5 metros por encima de la base del velero. Cuando la ola pasa, la embarcación se balancea hacia un lado pero recupera la posición vertical después. Por otro lado, en la segunda secuencia (abajo), el centro de masa está localizado 1.65 metros por encima de la base del barco, la cual es una configuración inestable. Se puede observar que la simulación reproduce como el bote vuelca debido a la inclinación producida por la trayectoria de la ola.

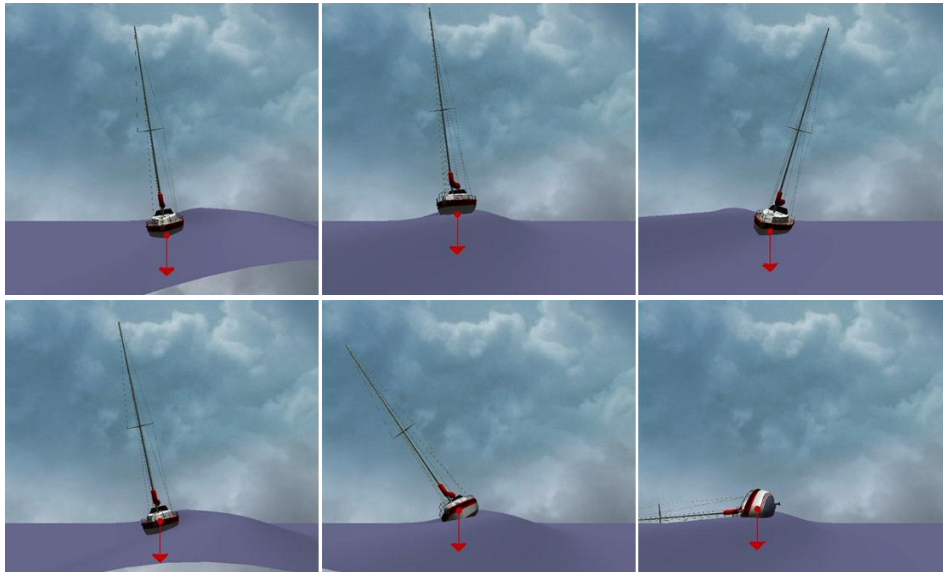


Figura 11. Animación de la trayectoria de una ola debajo de un velero: estable (arriba) e inestable (abajo). Los parámetros de la ola usados en la Ec. (10) son $A=4$ y $n=8$.

3.3 Interacción completa de fluido y sólido

El tercer caso simula el movimiento de una lancha en aguas tranquilas. La Figura 12 muestra distintas capturas de la animación donde se puede observar la formación de olas producidas por el movimiento del barco dejando una estela detrás de su trayectoria. Es interesante observar como el motor físico implementado es capaz de recrear la forma curvada de la estela dejada por los giros de la lancha. La superficie del agua es representada por una grilla de 200×200 celdas. Las dimensiones de la lancha son 12.7 metros de eslora, 2.4 metros de puntal y 3.85 metros de manga; con un peso de 16.5 toneladas y representada por una malla de 460 triángulos.

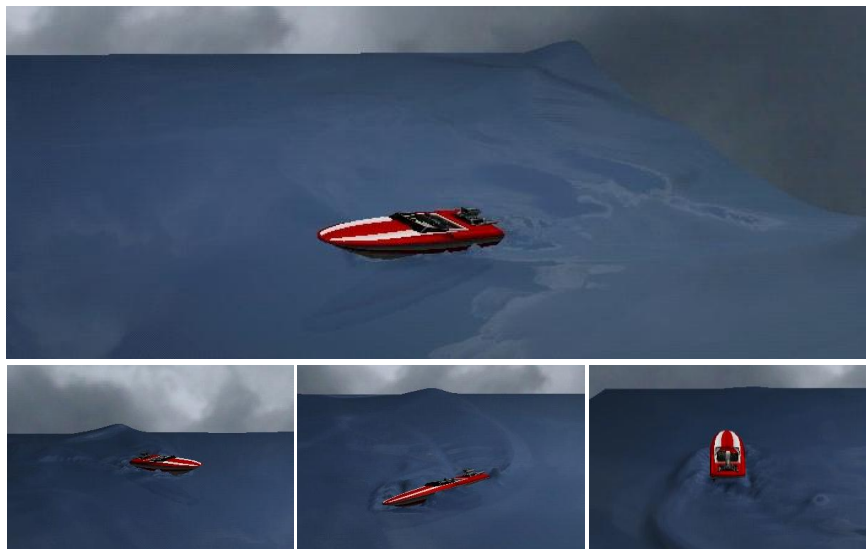


Figura 12. Embarcación desplazándose en un canal.

3.4 Performance y escalabilidad

La siguiente prueba tiene como finalidad mostrar la performance y escalabilidad del

método propuesto cuando se simula una escena con varios objetos. La [Figura 13](#) muestra cuatro grupos de objetos: 10, 100 y 200 esferas, y 50 objetos complejos. Las discretizaciones utilizadas son de 80 triángulos para cada esfera y de 520 triángulos para cada objeto complejo. La grilla utilizada para el fluido es de 175x175 celdas.

El cálculo del fluido se realiza en un hilo de ejecución adicional al utilizado para el resto del algoritmo. Esto permite utilizar grillas de fluido de gran tamaño sin afectar la performance total de la aplicación. Para ello, es importante definir mecanismos de sincronización que eviten inconsistencias al momento de actualizar la representación visual de la grilla y al momento de mapear los bordes de los objetos en la grilla.

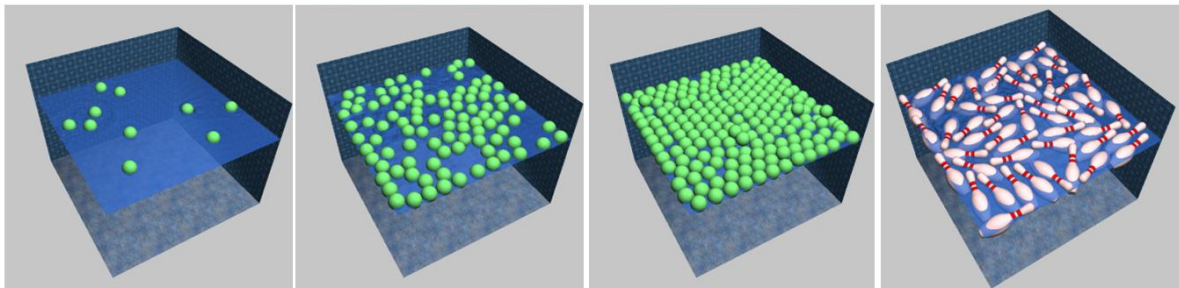


Figura 13. Prueba de performance con 10, 100 y 200 esferas; y con 50 objetos complejos.

La [Tabla 1](#) muestra la tasa de renderizado de cada escena, junto con el tiempo que demanda cada ítem del algoritmo. Los resultados presentes en la [Tabla 1](#) muestran la sensibilidad del método para el cálculo de fuerzas con respecto al número de triángulos en la discretización que se utilice. A medida que se incrementa el número de esferas utilizadas en los tres primeros grupos (80, 8000 y 16000 triángulos respectivamente), el tiempo de cálculo crece de forma lineal. Sin embargo, esto no se cumple para el grupo de objetos complejos (26000 triángulos), ya que el método escala mejor cuando simula gran cantidad de objetos con discretizaciones pequeñas, en lugar de poca cantidad de objetos con discretizaciones densas. Esto se debe a que el método se encuentra optimizado, excluyendo de la simulación objetos que se encuentran fuera del fluido o en estado de reposo.

Cantidad de objetos	10	100	200	50 complejos
Cálculo de fuerzas	1.6	16.8	25.6	96.9
Actualización simulación física	0.06	0.4	8.2	70.8
Cálculo de sección mojada	3.3	7.7	8.1	17.8
Renderizado	1.1	2	2.9	4.1
Tiempo total	6.06	26.9	44.8	189
FPS	165	37	22	5

Tabla 1. Tiempos de proceso para escenas con diferente carga (medido en milisegundos).

4 CONCLUSIONES

Fue presentado un motor físico con capacidad para representar interacción en doble sentido entre estructuras sólidas y superficies de fluidos. Este método es útil para producir animaciones de escenas con objetos flotantes arbitrarios en aguas abiertas, capturando la perturbación de la superficie por los objetos y la reacción del agua sobre los mismos. La simulación del líquido es rápida, escalable y fue testada en diferentes casos produciendo animaciones realistas.

Una de las principales contribuciones de este trabajo es la manera en que se calculan las

fuerzas de interacción utilizando la superficie mojada de los objetos, lo cual evita el costo computacional de calcular el volumen sumergido y los correspondientes centros geométricos. Además, si bien en este trabajo se aplicó el algoritmo de interacción a un fluido simulado a través de LBM, es adaptable a otros algoritmos y técnicas de simulación de superficies de fluidos basados en grilla.

Futuras extensiones del método incluyen aplicaciones de los datos adicionales que provee el motor de Lattice-Boltzmann, como la energía interna o la distribución de entropía para emular efectos visuales superficiales como la espuma o reflejos parpadeantes. También, se pueden explorar diferentes alternativas para el tratamiento de las condiciones de bordes entre objetos y el agua usando el método de frontera inmersa. Adicionalmente, se pueden usar implementaciones en GPU para ejecutar el algoritmo de Lattice-Boltzmann (Rinaldi et al. 2011, 2012a), que reducen ampliamente el tiempo de cálculo.

REFERENCIAS

- Cancelliere A., Chang C., Fotu E., Rothman D. and Succi S., The permeability of a random media: comparison of simulation with theory, *Physics of Fluids A: Fluids Dynamics*, 2(12):2085-2088, 1990.
- Chen J. and Lobo D., Toward interactive-rate simulation of fluids with moving obstacles using Navier-Stokes equations. *Graph. Models Image Process*, 57:107-116, 1995.
- Chen S., Chen H., Martinez D. and Matthaeus W., Lattice Boltzmann model for simulation of magnetohydrodynamics, *Phys. Rev. Lett*, 67(27):3776, 1991.
- Cords H., Moving with the flow: Wave particles in flowing liquids. *16th International Conference in Central European Computer Graphics, Visualization and Computer Vision (WSCG'08)*, 16:145-152, 2008.
- Cords H. and Staadt O., Real-Time open water environments with interacting objects. *Proceedings of Eurographics Workshop on Natural Phenomena (EGWNP'09)*, 35-42, Aire-la-Ville, Switzerland, 2009.
- Crane K., Llamas I. and Tariq S., Real-Time Simulation and Rendering of 3D Fluids, Chapter 30, *GPU Gems 3*, Addison-Wesley Professional, 2007.
- D'Amato J. P. and Lotito P., Mesh optimization with volume preservation using GPU. *Latin American Applied Research*, 41:291-297, 2011.
- Filippova O. and Haenel D., Lattice Boltzmann simulation of gas-particle flows in filters, *Computer and Fluids*, 26(7):697-712, 1997
- Foster N. and Fedkiw R., Practical animation of liquids, *Proc. of ACM SIGGRAPH*, 23-30, 2001.
- Foster N. and Metaxas D., Realistic animation of liquids, *Graphical models and image processing*, 58(5):471-483, 1996.
- Foti E. and Succi S., Three-dimensional flows in complex geometries with the lattice Boltzmann method, *Europhys. Lett*, 10(5):433, 1989.
- García Bauza C., Lazo M. and Vénere M., Incorporación de comportamiento físico en motores gráficos, *Mecánica Computacional* 27:3023-3039, 2008.
- García Bauza C., Boroni G., Vénere M. and Clausse A., Realtime Interactive Animations of Liquid Surfaces with Lattice-Boltzmann Engines, *Australian J. Basic and Applied Sci.*, 4:3730-3740, 2010.
- Génevaux O., Habibi A. and Dischler J., Simulating Fluid-Solid Interaction, *Proc. of Graphics Interface 2003*, Association for Computing Machinery Inc., 31-38, 2003.
- Harada T., Koshizuka S. and Kawaguchi Y., Smoothed particle hydrodynamics in complex shapes, *Proc. of Spring Conference on Computer Graphics*, 26-28, 2007.
- Higuera F. and Succi S., Simulating the flow around a circular cylinder with a lattice

- Boltzmann equation, *Europhys Lett*, 8:517, 1989.
- Irving G., Guendelman E., Lossaso F. and Fedkiw R., Efficient simulation of large bodies of water by coupling two and three dimensional techniques, *Proc. Of SIGGRAPH'06*, 805-811, 2006.
- Kass M. and Miller G., Rapid stable fluid dynamics for computer graphics, *ACM SIGGRAPH Computer Graphics*, 24:49-57, 1990.
- Kim J., Kim S., Ko H. and Terzopoulos D., Fast GPU computation of the mass properties of a general shape and its application to buoyancy simulation, *The Visual Computer*, 22:856-864, 2006.
- Layton A. and Vande Panne M., A numerically efficient and stable algorithm for animating water waves, *The Visual Computer*, 18:41-53, 2002.
- Li W., FanZ., Wei X. and Kaufman A., Flow Simulation with Complex Boundaries, *GPU Gems II*, Ed. Matt Pharr (Nvidia), Addison Wesley, Chapter 47, 2005.
- Lossaso F., Irving G. and Guendelman E., Melting and burning solids into liquids and gases, *IEEE Trans. Vis. Comp. Graph.* 12:343-352, 2006.
- Masten G. and Watterberg P. and Mareda I., Fourier synthesis of ocean scenes, *IEEE Computer Graphics and Application*, 7:16-23, 1987.
- Mei R., Lou L. S. and Shyy W., An accurate curved boundary treatment in the lattice Boltzmann method, *J. Comp. Phys.*, 155(2):307-330, 1999.
- Mihalef V., Kadioglu S., Sussman M., Metaxas D. and Hurmusiadis V., Interaction of two-phase flow with animated models, *Graphical Models*, 70:33-42, 2008.
- Millington I., Game Physics Engine Development, 2nd Ed., *How to Build a Robust Commercial-Grade Physics Engine for your Game*, Ed. Morgan Kaufmann, 2010.
- Müller M., Keiser R., Nealen A., Pauly M., Gross M. and Alexa M., Point based animation of elastic, plastic and melting objects, *Proc. ACM SIG-GRAPH*, Eurographics Symp. Comput. Anim. 141-151, 2004.
- Neyret F. and Praizelin N., Phenomenological simulation of brooks, *Proc. Eurographics Workshop*, 53-64, 2001.
- O'Brien J. and Hodgins J. Dynamic simulation of splashing fluids. *Proc. Computer Animation*, 198-205, Geneva, Switzerland, April 19-21, 1995.
- Rinaldi P., Dari E., Venere M., Clausse A. Lattice-Boltzmann Navier-Stokes simulation on graphic processing units, *Asian J. Applied Science*, 4:762-770, 2011.
- Rinaldi P., Dalponte D., Venere M. and Clausse A., (a) Graph-based cellular automata for simulation of surface flows in large plains, *Asian J. Applied Science* 5:224-231, 2012.
- Rinaldi P., Dari E., Vénere M. and Clausse A., (b) A lattice-Boltzmann solver for 3d fluid simulation on GPU, *Simulation Modelling Practice and Theory* 25:163-171, 2012.
- Robinson-Mosher A., Shinar T. and Gretarsson J., Su J. and Fedkiw R., Two-way Coupling of Fluids to Rigid and Deformable Solids and Shells, *ACM Trans. Graph.*, 27, Article 46, 2008.
- Schachter B., Long crested wave models, *Computer Graphics and Image Processing*, 12:187-201, 1980.
- Stam J., Stable fluids, *SIGGRAPH, Computer Graphics Proceedings*, Rockwood A. (Ed.), Addison Wesley Longman, 121-128, 1999.
- Succi S., Santangelo P. and Benzi R., High resolution lattice gas simulation of two-dimensional lattice gas turbulence, *Phys. Rev. Lett.*, 60:2738-2740, 1988.
- Takashi T., Heihachi U. and Kunimatsu A., The simulation of fluid-rigid body interaction. *In SIGGRAPH, Sketches & Applications*, 226, 2002.
- Thon S. and Ghazanfarpour D., A semi-physical model of running waters, *Comput. Graph. Forum. Proc. Eurographics*, 19:53-59, 2001.

- Thürey N., Physically Based Animation of Free Surface Flows with the Lattice Boltzmann Method, PhD thesis, University of Erlangen-Nuremberg, 2007.
- Thürey N. and Rüde U., Free Surface Lattice-Boltzmann fluid simulations with and without level sets, *Workshop on Vision Modeling and Visualization*, 199-208, 2004.
- Tso P. and Barsky B., Modeling and rendering waves: Wave-tracing using beta-splines and reflective and refractive texture mapping, *ACM Transactions on Graphics*, 6:191-214, 1987.
- Uoane T., Lattice Boltzmann methods for shallow water flow applications, *Master Thesis, University of the Witwatersrand, Johannesburg, South Africa*, 2011.
- van der Sman R., Lattice Boltzmann Schemes for Convection-Diffusion Phenomena; Application to Packages of Agricultural Products, *Ph.D. Thesis, University of Wageningen, Holland*, 1999.
- Wendt J., Baxter W., Oguz I. and Lin M., Finite volume flow simulations on arbitrary domains, *Graphical Models*, 69 (1):19-32, 2007.
- Woo M., Neider J., Davis T. and Shreiner D. The Frame buffer, *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL, Version 1.2 (3rdEdition)*, Chapter 10, 1999.
- Yuksel C., House D. and Keyser J., Wave Particles, *Proceedings of SIGGRAPH 26*, 99, San Diego, California, ACM Press, 2007.
- Zhou J. G., Lattice Boltzmann Methods for Shallow Water Flows. *Springer-Verlag*, 2004.