

Simulación computacional de fluidos tridimensionales

David Vargas¹, Carolina Valerio² y Esteban Esquivel³

CI-1441 Paradigmas computacionales

Escuela de Ciencias de la Computación e Informática

Facultad de Ingeniería

Universidad de Costa Rica

¹ dav793@gmail.com, ² caro.cvs@gmail.com, ³ esprimos26@gmail.com

Julio de 2015

Resumen

La simulación computacional de fluidos en espacios físicos es de interés en varias áreas pertenecientes o relacionadas a la computación. En el área de prevención de desastres podría ayudar a mitigar o predecir los efectos de inundaciones y otros fenómenos naturales. En CGI (*computer-generated imagery*) permite la visualización de océanos, ríos y otros cuerpos de agua con impresionante detalle para su uso en producciones audiovisuales. En el área de videojuegos puede introducir interesantes mecanismos de interacción con el usuario, que mejoren su experiencia de juego. En este último dominio particularmente interesa la simulación de fluidos en tiempo real, sacrificando precisión con el fin de lograr un buen desempeño en computadores personales modernos. Nos enfocamos en la implementación de un modelo simplificado de simulación de fluidos en tiempo real, con énfasis de uso en videojuegos.

1. Introducción

La simulación de fenómenos físicos ha dado auge a una enorme gama de videojuegos de distintos géneros, en los cuales la interacción del usuario con un mundo análogo al nuestro aporta a la calidad y capacidad de entretenimiento del producto. Un caso especial es la computación precisa de fluidos, un problema cuya complejidad es elevada cuando se requiere resolver en tiempo real, y siendo este el caso para muchos videojuegos. En general, los computadores personales existentes no cuentan con suficiente capacidad de procesamiento para computar fluidos en tiempo real con alta fidelidad, y por esto el problema suele ser simplificado en la práctica mediante el uso de heurísticas.

La computación de fluidos también es de interés para la simulación de desastres naturales (inundaciones,

tsunamis...) como pueden transcurrir en espacios físicos reales, con el objetivo de mitigar o al menos anticipar sus efectos.

Implementamos una solución al problema mediante el paradigma computacional de sistemas multiagente. Para disminuir la complejidad del problema lo simplificamos mediante el uso de una heurística, la cual consiste en la discretización del entorno a una cuadrícula de celdas cúbicas que pueden contener líquido, y de ser así lo distribuyen hacia sus celdas adyacentes.

2. Marco teórico

Consideramos la propuesta de Chentanez y Muller por asemejarse en conceptualización e implementación a la nuestra. Abordaron el problema de simulación computacional de fluidos tridimensionales a gran escala mediante la construcción de un modelo que utiliza una cuadrícula híbrida de celdas cúbicas regulares sobre una capa de *celdas altas*. En su modelo utilizan las ecuaciones diferenciales de Euler, que describen el comportamiento de un fluido compresible no viscoso. También hacen uso del GPU para el procesamiento de la simulación.

Varias celdas pueden ser unificadas en lo que se conoce como una celda alta, y se procesan los fluidos en las celdas pertenecientes a una celda alta en menos tiempo de lo que tomaría procesar cada celda individualmente. Así, se concentra la capacidad de procesamiento en la superficie del cuerpo líquido, donde ocurre la mayor interacción entre celdas.

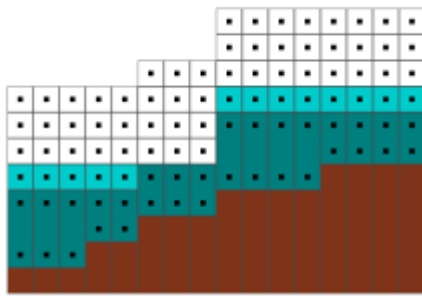


Figura 2.1. Representación de una sección bidimensional de la cuadrícula, con celdas regulares y celdas altas.

Al utilizar las ecuaciones de Euler para modelar el comportamiento de los fluidos, esta solución tiene un nivel de complejidad mayor. El resultado es una simulación muchas veces más realista que la nuestra. Además, el mecanismo de celdas altas y el goce de la capacidad de procesamiento con la que cuentan los GPUs modernos contribuyen fuertemente con el desempeño del modelo.

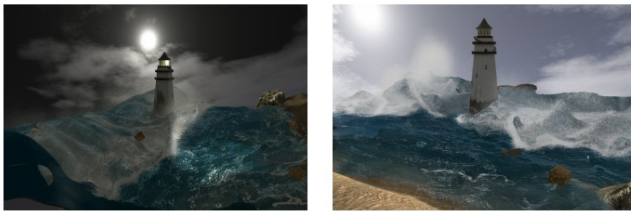


Figura 2.2. Capturas de la simulación en tiempo real

3. El problema

Pretendemos simular fluidos computacionalmente en tiempo real, mediante la elaboración de un modelo computacional de fluidos, que desempeñe simulaciones de distribución de líquido a gran escala en un espacio físico con aproximada precisión. Aceptamos el modelo si desempeña una distribución de líquido que parezca al menos aceptablemente familiar de acuerdo con nuestra percepción de la realidad, observando la influencia de fenómenos como:

- Las fuerzas que actúan sobre el líquido.
- El desplazamiento natural del líquido sobre superficies sólidas.

4. Objetivos

La solución debe observar los siguientes supuestos, y de ser así el modelo resuelve el problema adecuadamente de acuerdo a nuestro criterio:

- La cohesión molecular de un líquido le permite tener movilidad, mientras evita que se disperse como un gas. Esperamos observar el líquido fluyendo y asentándose, pero que no se evapore o disipe.
- Un líquido es influenciado por fuerzas de atracción gravitacional y presión hidráulica que lo intentan desplazar hacia abajo, o de no ser posible, hacia los lados.
- Un sólido es (generalmente) más denso que un líquido. Esperamos que el líquido nunca atraviese o se filtre por el terreno sólido, sino que sea desplazado por éste.

5. La propuesta de solución

Paradigma: Sistema multiagente.

Representación del conocimiento y la información:

Entorno: Representación del espacio en forma de cuadrícula de tres dimensiones (x, y, z). Está compuesto por secciones cúbicas o *celdas* de tamaño regular.

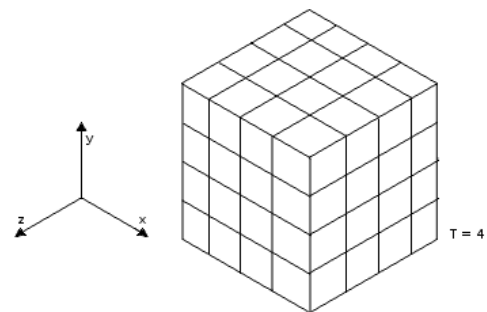


Figura 5.1. Representación del entorno de simulación

Celda: Agente que representa una sección cúbica del espacio, y refleja la concentración de líquido en esa sección en cualquier momento determinado.

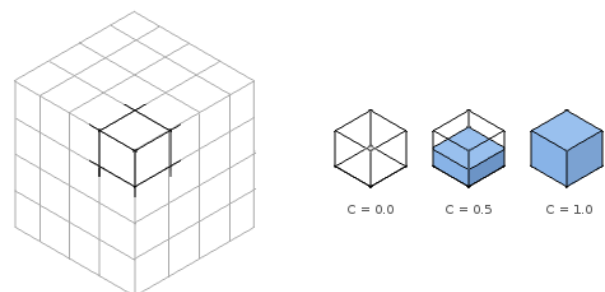


Figura 5.2. Representación de una celda del entorno, con diferentes niveles de concentración de líquido

- Posee un índice (x, y, z) único, que determina su posición en el entorno.
- Posee un nivel de concentración de líquido C , un número real entre 0.0 y 1.0 .
- Puede ser sólida (perteneciente al terreno), en cuyo caso nunca fluye líquido hacia ella.
- Es la unidad mínima de discretización del espacio.



Figura 5.5. Ejemplo de un mapa de alturas

Celda abajo: Una celda B está *abajo* de una celda A si $B.y = A.y - 1$

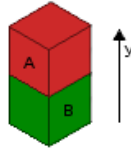


Figura 5.3. La celda B está abajo de la celda A

Celdas contiguas: Una celda A es *contigua* a una celda B si

$$B.y = A.y \text{ AND } ((B.x = A.x \text{ AND } B.z = A.z \pm 1) \text{ OR } (B.z = A.z \text{ AND } B.x = A.x \pm 1))$$

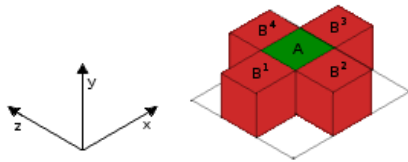


Figura 5.4. Las celdas B^i son contiguas a la celda A

Celdas vecinas: Una celda A es *vecina* de una celda B si A está abajo de B ó A es contigua a B .

Terreno: Conjunto de celdas sólidas hacia las que nunca se distribuye el líquido.

Mapa de alturas: Archivo de imagen utilizado para generar el terreno de la simulación.

- La imagen está en escala de grises.
- Cada pixel de la imagen representa un índice (x, y, z) en el entorno.
- La posición de cada pixel en la imagen determina sus subíndices x, z .
- El color de cada pixel en la imagen determina su subíndice y . Entre más cercano sea al blanco máximo, más alto es su subíndice y .

Mecanismos y procedimientos:

La generación del terreno: Previo a la simulación se genera el terreno utilizando un mapa de alturas seleccionado por el usuario. Se utiliza la información de cada pixel en el mapa para construir columnas de celdas sólidas que constituyen el terreno.

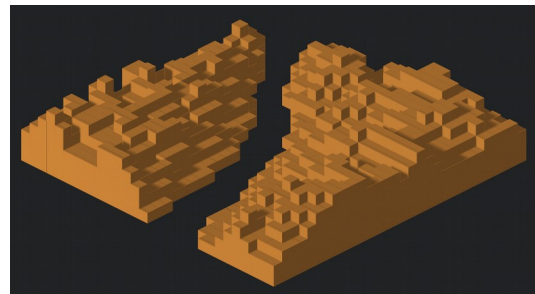


Figura 5.6. Captura de un terreno generado a partir del mapa de alturas de la figura 5.5

La simulación: Durante cada ciclo o *tick* de la simulación se procesan los *flujos verticales* y los *flujos horizontales* de las celdas cuya concentración de líquido es mayor que 0 . El procesamiento de flujos de una celda consiste en la redistribución de concentración de líquido entre la celda en cuestión y un subconjunto de sus celdas vecinas. Esto puede suceder de dos formas:

1. Procesamiento de flujos verticales de A a B (B abajo de A):

Se intenta redistribuir todo el líquido posible de A hacia B hasta que no quede más líquido en A o hasta que B haya alcanzado la concentración máxima.

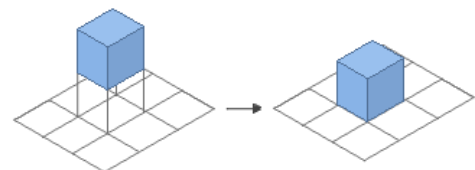


Figura 5.7. Distribución de líquido hacia abajo

2. Procesamiento de flujos horizontales de A a B (B contiguo a A):

Asumiendo que B es el conjunto de las celdas contiguas a A , se modifica la concentración de líquido de A y B al promedio entre las concentraciones de A y de las celdas en B .

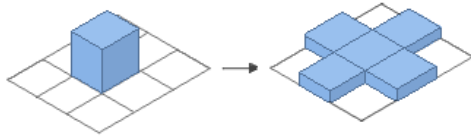


Figura 5.8. Distribución de líquido hacia celdas contiguas

El *renderer*: Construimos un visualizador del entorno que posterior al procesamiento de flujos en cada tick de la simulación, actualiza las celdas activas para reflejar la concentración de líquido que posee cada una. También construye las celdas sólidas durante la etapa de generación del terreno.

6. Desarrollo, prueba y validación

Se utilizó Unity 5 para la elaboración de todo el proyecto. Unity es un motor de videojuegos multiplataforma, utilizado en el desarrollo de videojuegos para PC, consolas, dispositivos móviles y web.

Unity no es un software de simulación, por eso toda la lógica involucrada en la generación del terreno, la simulación y la visualización del entorno fue codificada directamente por nosotros. Aparte de Unity no utilizamos más recursos, librerías o herramientas.

Se eligió Unity para la elaboración del proyecto por la familiaridad con que contábamos de antemano con la plataforma, que nos facilitó la tarea de construcción de un *renderer* o visualizador de la simulación. Sin embargo, de haber conocido NetLogo antes es probable que lo utilizáramos en vez de Unity, por ser una plataforma más amigable para la construcción de simulaciones.

Construimos una aplicación independiente, ejecutable en plataformas Windows y Mac sobre arquitecturas de 64 y 32 bits. Incorporamos una interfaz gráfica simple para permitir al usuario interactuar con el modelo, alterando la velocidad de la simulación, el ángulo y acercamiento del campo visible, la forma en la que se introduce líquido en el entorno, y para desplegar información general del estado de la simulación. Previo a la simulación se despliega un menú

mediante el cual permitimos al usuario escoger uno de tres terrenos, con el propósito de ofrecer escenarios distintos en los cuales ejecutar la simulación.

7. Experimentación y análisis

En general se obtuvieron resultados satisfactorios. El modelo cumple con los objetivos propuestos, desempeñando la distribución de fluidos en tiempo real y de forma análoga a la realidad, con la condición de aceptar secciones cúbicas como unidad mínima de granularidad del entorno. Adicionalmente, los distintos terrenos ofrecidos permiten visualizar el comportamiento del modelo bajo circunstancias diferentes.

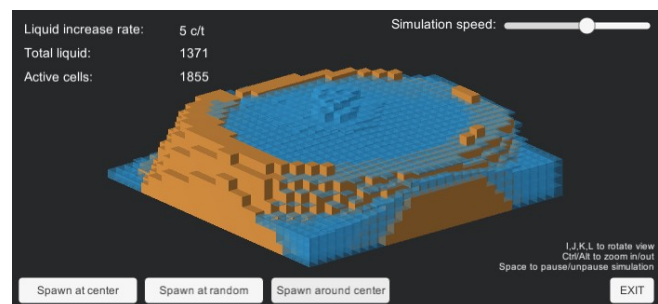


Figura 7.1. Captura de simulación, con líquido desbordándose fuera del contenedor formado por el terreno

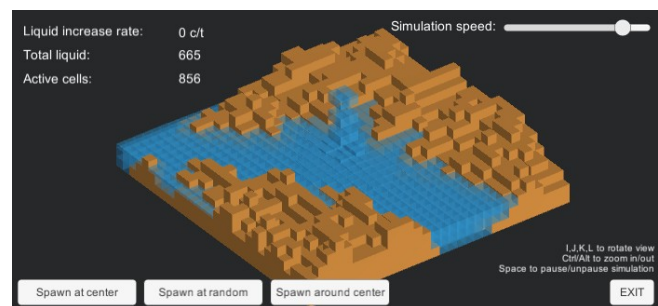


Figura 7.2. Captura de simulación, con líquido llenando el cauce formado por el terreno

8. Problemas abiertos y problemas futuros

Originalmente se planteó el modelo de forma que en cada tick de la simulación se recorran ordenadamente todas las celdas del entorno, procesando los flujos de cada una. Esto llevó a tomar decisiones de diseño que favorecían este mecanismo de iteración ordenada, pero que introdujeron problemas imprevistos. En la práctica resultó en el flujo desproporcionado de líquido, el cual tendía a fluir en algunas direcciones más que en otras. Esto no es deseable pues violenta el supuesto de distribución de líquido de

acuerdo con las fuerzas gravitacionales naturales. Además, el mecanismo hace difícil la paralelización de la simulación. Para resolverlo se rediseñó parte del modelo y el sistema de rendering, de forma que ahora se lleva registro de las celdas con concentración de líquido mayor que 0, llamadas celdas activas. En cada tick de la simulación se consideran únicamente las celdas activas, y así no se recorren todas las celdas en el entorno. Esta modificación en la progresión de la simulación también nos permitió simplificar el algoritmo de distribución horizontal de líquido. La solución resolvió el problema de uniformidad de los flujos, y adicionalmente observamos una mejora sustancial en el desempeño de la aplicación. Por último, esta solución resulta conveniente en caso de una futura paralelización de la simulación, pues solo hace falta verificar que la distribución de líquido no exceda la concentración máxima de ninguna celda. Esto suponiendo que flujos desde varias celdas hacia una misma celda se serializan.

Inicialmente se planeó también que el terreno fuera generado a partir de mapas geográficos reales haciendo uso del *Google Elevation API*, una herramienta que permite extraer información de relieves de mapas de *Google Maps*. La herramienta resultó ser demasiado imprecisa para su uso en este proyecto. Para resolver el problema se implementó un algoritmo para la generación de terrenos a partir de mapas de alturas.

9. Bibliografía

[1] N. Chentanez y M. Muller. Real-time eulerian water simulation using a restricted tall cell grid. ACM SIGGRAPH 2011.
<http://matthias-mueller-fischer.ch/publications/tallCells.pdf>