

**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS
FACULTAD 5
ENTORNOS VIRTUALES**



Título: “Propuesta de algoritmo para simular humo en un entorno virtual”

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: Yadier Escobar Domínguez
Luis Elier Elías López

Tutor: Ing. Yusleidy Guelmes León

Ciudad de la Habana, Junio de 2007

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yadier Escobar Domínguez

Firma del Autor

Luis Elier Elías López

Firma del Autor

Ing. Yusleidy Guelmes León

Firma del Tutor

Datos de Contacto

Tutor(a): Ing. Yusleidy Guelmes León

E-mail: yguelmes@uci.cu

Teléfono: 8372715

Graduada en Ingeniería Informática de la CUJAE y la Universidad de Cienfuegos, en Julio del 2005, desde su graduación se desempeña como profesora en la disciplina Ingeniería y Gestión de Software de la Facultad 5 de la UCI.

Agradecimientos

Nuestro agradecimiento a todas las personas que hicieron posible este trabajo, a nuestra tutora y a nuestros profesores por la formación profesional a lo largo de estos cinco años.

A nuestros padres y amigos, por toda la ayuda que nos han brindado, por su apoyo, preocupación y por la confianza que depositaron en nosotros.

A nuestra Revolución y a Fidel por hacernos partícipes de este proyecto.

Dedicatoria

A nuestros padres quienes nos han apoyado siempre y nos sembraron la semilla de la educación y la importancia de regarla todos los días. Que nos enseñaron el valor de seguir adelante aún en los momentos más difíciles y nos ofrecen su ayuda y su amor incondicionalmente.

A mis amigos que han estado a mi lado siempre y me han tendido la mano en todo momento, que me han ayudado a superarme a lo largo de la carrera y han sabido comprenderme.

Resumen

En este trabajo se propone un algoritmo para la simulación de humo para ser incluida en el simulador de tiro que la empresa SIMPRO se encuentra desarrollando. El objetivo consiste en desarrollar un algoritmo que permita la simulación mediante técnicas físicas y computacionales de un flujo de humo bastante realista que nos de la sensación de estar en un entorno real. Es importante señalar que no existe actualmente un algoritmo equivalente, siendo intención de la investigación utilizar el mismo en el simulador para el entrenamiento de los soldados.

Para ello se hace un estudio de los aspectos relacionados con las propiedades físicas y químicas del humo así como sus principales utilidades. También se hizo un estudio de los diferentes modelos matemáticos-computacionales que permiten simular un flujo de humo, para así determinar un algoritmo capaz de realizar una Simulación realista.

Así, el proyecto pretende servir como herramienta de soporte al simulador para una mejor formación de los soldados. Finalmente se dan algunas recomendaciones para el desarrollo futuro del mismo.

Palabras claves: Entornos virtuales, simulación de humo, algoritmos para la simulación de humo.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTOS TEÓRICOS ACTUALES SOBRE LA SIMULACIÓN DE HUMO.	3
1.1 DEFINICIÓN DEL HUMO.	3
1.1.1 <i>Composición química.</i>	4
1.2 UTILIDADES DEL HUMO.....	4
1.2.1 <i>Pantallas de humo.</i>	4
1.2.2 <i>Granadas de humo.</i>	5
1.2.3 <i>La fumigación.</i>	5
1.2.4 <i>Señales de humo.</i>	6
1.3 CONCEPTOS FUNDAMENTALES.....	6
1.4 IMPORTANCIA Y UTILIZACIÓN DE LA SIMULACIÓN DE HUMO.....	9
1.5 ANTECEDENTES DE LA SIMULACIÓN VISUAL DE HUMO.	10
CAPÍTULO 2 DESCRIPCIÓN DE LOS PRINCIPALES MODELOS Y TÉCNICAS PARA LA SIMULACIÓN DE HUMO.	12
2.1 ANÁLISIS DE LA DETERMINACIÓN DE LOS MÉTODOS ESCOGIDOS.....	12
2.2 MODELO BASADO EN LAS ECUACIONES DE NAVIER-STOKES USANDO HARDWARE GRÁFICO (GPU).	13
2.2.1 <i>Algoritmo 1.</i>	22
2.3 MODELO DE FLUJO DE HUMO BASADO EN LAS ECUACIONES DE REYNOLDS USANDO METABALLS.	23
2.3.1 <i>Algoritmo 2.</i>	35
CAPÍTULO 3 PROPUESTA DE ALGORITMO PARA LA SIMULACIÓN DE HUMO.	37
3.1 ALGORITMO PROPUESTO.....	37
3.1.1 <i>Características principales del método.</i>	37
3.1.2 <i>Ventajas y Desventajas.</i>	38
3.2 COMPARACIÓN ENTRE LOS ALGORITMOS DESCRITOS.....	38
3.3 PSEUDOCÓDIGO PARA EL ALGORITMO 2.....	40
CONCLUSIONES	44
RECOMENDACIONES	45
REFERENCIAS BIBLIOGRÁFICAS.	46
ANEXOS	48
ANEXO 1. LEVANTAMIENTO DE HUMO.....	48
ANEXO 2. LEVANTAMIENTO DE HUMO E INTERACCIÓN CON UN OBJETO.	49
ANEXO 3. HUMO COLOREADO.	50
GLOSARIO DE TÉRMINOS	51

Introducción

Los entornos virtuales son el producto de una tecnología de reciente aparición que se conoce como arquitectura virtual, se trata de escenarios virtuales creados mediante infografía cuyo objetivo es parecerse lo máximo posible a los escenarios reales. Los entornos virtuales significan una práctica no real, un entrenamiento previo a las situaciones reales y en muchos casos, una práctica sustitutiva de los acontecimientos reales. Nos introducen en situaciones en las que hay que tomar decisiones que constituyen aprendizajes transferibles a la vida real.

Con la aparición de los entornos virtuales, los universos representados pueden no existir y provenir de la imaginación de sus creadores. Los conceptos tradicionales de la producción gráfica se ven modificados. Las cámaras se colocan en lugares antes inaccesibles, se producen encuadres imposibles, movimientos de cámara que desafían las leyes físicas, etc. Así, estos suponen una nueva representación digital de la realidad y una construcción de un nuevo mundo posible.

Aprovechando esta tecnología, la empresa SIMPRO (Simuladores Profesionales) se encuentra desarrollado un simulador de tiro donde se intenta situar al usuario en un entorno virtual de manera que se asemeje lo mejor posible a la vida real.

La facultad 5 de la Universidad de las Ciencias Informáticas está estrechamente relacionada con la empresa SIMPRO así como con el simulador de tiro que esta se encuentra desarrollando. En dicho simulador todavía no se ha logrado incluir los efectos de humo que son imprescindibles para darle un entorno más real y que en realidad son básicos para lo tiradores de tiro directo, como son los misiles antitanques y francotiradores donde los mismos guiándose por el humo pueden determinar la dirección del viento y la velocidad, logrando que el disparo sea más efectivo y preciso, puesto que si el disparo se realiza con fuertes vientos estos pueden producir vibraciones en el tubo del lanzador y malograr la puntería y si el disparo es a través de zonas con humo durante el día, el humo puede dificultar la línea de mira del apuntador y oscurecer el blanco.

Analizando detenidamente la situación problémica antes expuesta se ha definido el siguiente **problema científico**, inexistencia de los efectos de humo en el simulador de tiro.

De esta forma y teniendo en cuenta lo anterior planteado el objeto de estudio de este trabajo consiste en la simulación virtual de humo.

El **objetivo general** de este trabajo es determinar un algoritmo que permita representar los efectos del humo y su comportamiento físico en el simulador para perfeccionar el entrenamiento del tirador.

Para dar solución al problema existente se proponen las siguientes tareas a cumplir:

- Análisis y síntesis de la bibliografía relacionada con los modelos y algoritmos existentes sobre simulación de humo.
- Selección de algoritmo y técnicas para simular humo.
- Propuesta de un algoritmo que posibilite la incorporación de efectos de humo en el simulador.

El campo de acción de nuestro trabajo comprende el estudio de modelos y algoritmos para la simulación de humo en el simulador de tiro.

La idea a defender es estudiar los algoritmos de humo, establecer una comparación entre ellos y determinar el óptimo para la simulación de humo.

Capítulo 1 Fundamentos teóricos actuales sobre la simulación de humo.

En este capítulo se dará a conocer los conceptos fundamentales relacionados con la simulación, especialmente los relacionados con la simulación de humo. Se dará a conocer como está la actualidad en Cuba y el mundo con respecto a la simulación de humo, o sea mostrar los trabajos realizados anteriormente con respecto a este tema. Además dar una breve panorámica sobre la simulación para así aclarar muchas dudas y esclarecer situaciones que puedan no ser entendidas.

1.1 Definición del humo.

Para tener un mayor entendimiento sobre la materia relacionada con el humo, a continuación se dará a conocer que es exactamente el humo, su composición química, así como las utilidades del mismo.

El humo es una suspensión en el aire de pequeñas partículas sólidas que resultan de la combustión incompleta de un combustible. Es un subproducto no deseado de la combustión, producido en fogatas, brasas, motores de gasolina y diesel. Cuando una combustión es correcta y completa los únicos subproductos son vapor de agua, dióxido de carbono, ácido carbónico que llevan consigo carbón en polvo muy tenue. (Ver Figura 1)



Figura 1. (Levantamiento de humo)

Es también una sustancia aromática y preservativa para varios comestibles.

La característica física más básica del humo es la distribución del tamaño de sus partículas. La emisión del humo es uno de los elementos básicos para caracterizar un ambiente del fuego. El humo producto de la combustión llameante tiende a tener un contenido grande de carbón (grafítico) elemental.

Las características del humo del interés primario para la comunidad del fuego son extinción, visibilidad, y detección ligeras.

La inhalación de humo es la causa primaria de la muerte en las víctimas de fuegos internos. Las muertes por humo son a causa de una combinación de daño termal, envenenamiento e irritación pulmonar causados por el monóxido de carbono, cianuro de hidrógeno y otros productos de la combustión. La inhalación del humo puede por lo tanto conducir rápidamente a la incapacitación y a la pérdida de sentido [1].

1.1.1 Composición química.

La composición química del humo depende de la naturaleza del combustible ardiente y de las condiciones de la combustión. Los fuegos con alta disponibilidad de oxígeno se queman en temperatura alta y con pequeña cantidad de humo producida; las partículas se componen sobre todo de la ceniza o en diferencias grandes de temperatura, del aerosol condensado del agua. La temperatura alta también conduce a la producción de los óxidos de nitrógeno.

Dependiendo del tamaño de la partícula, el humo puede ser visible o invisible al ojo desnudo [2].

1.2 Utilidades del humo.

A continuación se darán a conocer las principales utilidades del humo tanto en el ámbito militar como el social.

1.2.1 Pantallas de humo.

Una pantalla de humo es un lanzamiento del humo para enmascarar el movimiento o la localización de unidades militares tales como infantería, tanques o naves. Se despliega lo más comúnmente posible en un

frasco, generalmente como granada. La granada lanza una nube muy densa del humo diseñada para llenar los alrededores en un viento ligero. Mientras que las pantallas de humo habrían sido utilizadas originalmente para ocultar el movimiento de la línea de la vista de los enemigos, la tecnología moderna indica que están también disponibles ahora en nuevas formas; permiten ser invisibles al infrarrojo así como el espectro visible de la luz para prevenir la detección por los sensores infrarrojos, y también para prevenir los rayos láser de los buscadores enemigos.

Las pantallas de humo son utilizadas generalmente por la infantería para encubrir su movimiento en áreas de la exposición al fuego enemigo. Pueden también ser utilizadas por los vehículos de lucha armados, tales como tanques, para encubrir un retiro [3].

1.2.2 Granadas de humo.

Consiste en un cilindro de acero con algunos agujeros de emisión en la tapa y en el fondo para permitir el lanzamiento del humo cuando se enciende la granada. Este se llena de fósforo blanco que es separado por la acción explosiva. El fósforo coge fuego en la presencia del aire, y se quema con una llama amarilla brillante, mientras que produce cantidades copiosas de humo blanco (pentóxido de fósforo). Estas son conocidas como granadas incendiarias [4]. (Ver Figura 2).



Figura 2. (Granada de humo utilizada durante un ejercicio militar)

1.2.3 La fumigación.

La fumigación es un método de control de parásitos, se llena totalmente un área de pesticidas gaseosos para sofocar o para envenenar a los parásitos dentro. Se utiliza para el control de parásitos en los edificios

(fumigación estructural), el suelo, el grano, y el producto, y también se utiliza durante el proceso de las mercancías que se importarán o exportarán para prevenir la transferencia de enfermedades [5].

1.2.4 Señales de humo.

Una señal del humo es una forma de comunicación visual usada sobre una distancia, desarrollada en América y en China. Cubriendo un fuego con una manta y rápidamente quitándolo una colina de humo puede ser generada. Con un cierto entrenamiento, los tamaños, las formas, y la sincronización de estas colinas pueden ser controladas. Los soplos se pueden observar a larga distancia y al mismo tiempo vistas por cualquier persona dentro de su gama visual [6].

1.3 Conceptos fundamentales.

El Modelado y la Simulación Computacional de Humo han experimentado un fuerte impulso en los últimos años, debido a su gran utilidad en los distintos campos, desde la industria cinematográfica hasta los entretenidísimos juegos por computadoras. En este trabajo se abordan algunos conceptos que para los que no están familiarizados con el tema serían un poco difícil de entender.

Primeramente se debe comenzar con conceptos relacionados con el título de este trabajo como por ejemplo ¿que es la realidad virtual?, la realidad virtual es una representación de las cosas a través de medios electrónicos, que nos da la sensación de estar en una situación real en la que podemos interactuar con lo que nos rodea.

La realidad virtual puede ser de dos tipos:

Inmersiva:

Los métodos inmersivos de realidad virtual con frecuencia se ligan a un ambiente tridimensional creado por computadora el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano.

No inmersiva:

La realidad virtual no inmersiva utiliza medios como el que actualmente nos ofrece Internet en el cual podemos interactuar en tiempo real con diferentes personas en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales a la computadora. Ofrece un nuevo mundo a través de una ventana de escritorio. Este enfoque no inmersivo tiene varias ventajas sobre el enfoque inmersivo como:

- Bajo costo, fácil y rápida aceptación de los usuarios.
- Los dispositivos inmersivos son de alto costo y generalmente el usuario prefiere manipular el ambiente virtual por medio de dispositivos familiares como son el teclado y el ratón que por medio de cascos pesados o guantes.

Muy estrechamente relacionado con el concepto de realidad virtual está el concepto de entornos virtuales que se trata de escenarios virtuales creados mediante infografía cuyo objetivo es parecerse lo máximo posible a los escenarios reales y en donde el usuario puede interactuar modificando y creando nuevos elementos. Estos entornos virtuales pueden ser compartidos por varios usuarios a través de la red.

Para la simulación de humo existe una técnica muy importante utilizada por todos los que han excavado en este campo, esta técnica es conocida como el confinamiento del vórtice: el movimiento del humo, aire y otros fluidos de viscosidad baja, típicamente contienen flujos rotatorios en una variedad de escalas. Este flujo rotatorio es la vorticidad, para restaurar estos movimientos de escala se utiliza el confinamiento del vórtice. El confinamiento del Vortice trabaja primero calculando la vorticidad y luego de la vorticidad se calcula un campo de vector vorticidad normalizado [7].

Vórtice:

Un vórtice es un flujo turbulento en rotación espiral con trayectorias de corriente cerradas. Como vórtices puede considerarse cualquier tipo de flujo circular o rotatorio que posee vorticidad [8].

Vorticidad:

La vorticidad es un concepto matemático usado en la dinámica de los fluidos que se puede relacionar con la cantidad de circulación o rotación de un fluido. La vorticidad se define como la circulación por unidad de

área en un punto del flujo. El movimiento de un fluido se puede denominar solenoidal si el fluido gira en círculo o en hélice, o de forma general si tiende a rotar en torno a un eje [9]. (Ver Figura 3).



Figura 3. (Humo y Vortices)

Una técnica de gráficos realizada por ordenador para simular interacción orgánica entre diferentes objetos en el espacio es el *Metaball* [10] y fue inventado por Jim Blinn a principios de los años 1980.

Otra técnica también muy utilizada en el campo de la simulación de humo es el *sistema de partículas* ya que es una técnica de los gráficos por computadora en 3D que se utiliza para crear una gran cantidad de objetos que obedezcan reglas de comportamientos bien definidos. Útil no solamente para las multiplicidades que controlan objetos discretos tales como asteroides o pájaros, también como herramienta para crear fenómenos naturales tales como fuego o humo.

La partícula:

Una partícula es un objeto que tiene masa, posición, velocidad y puede estar afectada por fuerzas exteriores, como por ejemplo la aceleración de la gravedad, la resistencia del aire o el viento [11].

Se denomina “BillBoarding” o Billboards animados [12] a una técnica de dibujo mediante la cual, se muestran objetos que geoméricamente deberían ser muy complejos mediante un simple plano y una textura. Dicho plano se ajusta en tiempo real para orientarlo hacia el observador, de modo que no note la falta de tridimensionalidad. Su variante animada puede ser utilizada para simular efectos imposibles de representar en tres dimensiones con los recursos actuales, como explosiones o humo. La animación se simula pregenerando una serie de texturas que, presentadas una tras otra en forma de ciclo, producen el resultado deseado (Ver figura 4).

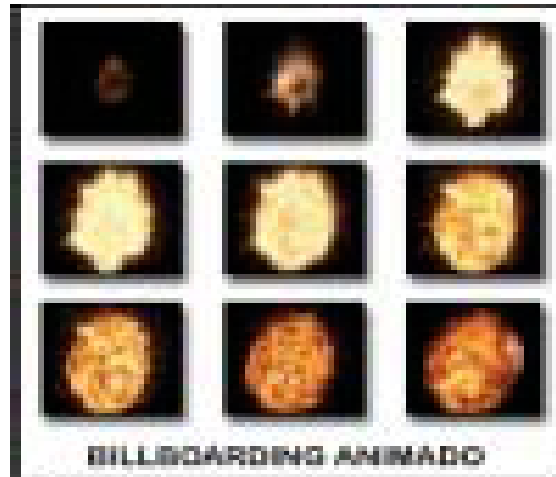


Figura 4. (BillBoarding)

Para lograr una buena simulación de humo es necesario utilizar una gran variedad de ecuaciones matemáticas ya que el humo es un fluido muy difícil de modelar.

1.4 Importancia y utilización de la simulación de humo.

La modelación de los fenómenos naturales tales como humo sigue siendo un problema desafiador en los gráficos por computadora (CG). Esto no es una sorpresa puesto que el movimiento de gases tales como humo es altamente complejo y turbulento. La simulación de humo tiene muchos usos obvios en la industria incluyendo efectos especiales y juegos interactivos. Idealmente, un buen modelo de humo de CG puede producir resultados altamente realistas. Obviamente el modelamiento de humo y de gases es de gran importancia en otros campos de la ingeniería. Más generalmente, el campo computacional de la dinámica de los fluidos (CFD) que se dedica a la simulación de gases y de otros fluidos tales como agua. Recientemente investigadores en los gráficos por computadora comenzaron a excavar en la literatura abundante del CFD para analizar los algoritmos que se puedan adoptar y modificar para los usos de los gráficos por computadora [13].

1.5 Antecedentes de la simulación visual de humo.

La modelación de humo y de otros fenómenos gaseosos ha recibido mucha atención en la comunidad de los gráficos por computadora durante las dos décadas pasadas. Los primeros modelos se centraron en un fenómeno particular y animaron la densidad del humo directamente sin modelar su velocidad. El detalle adicional fue agregado usando las texturas sólidas cuyos parámetros fueron animados en un cierto plazo. Posteriormente, los campos random de velocidad basada en un espectro de *Kolmogoroff* [14] fueron utilizados para modelar el movimiento complejo característico del humo. Crear una simulación dinámica convincente del humo es una tarea consumidora de tiempo.

Una manera más natural de modelar el movimiento del humo es simular las ecuaciones de la dinámica de los fluidos directamente. En el año 1984 *Kajiya y Von Herzen* [15] fueron los primeros en el mundo de los gráficos por computadoras en hacer esto.

Desafortunadamente, la energía de la computadora disponible en ese entonces permitió que se produjeran resultados solamente en rejillas muy gruesas. A excepción de algunos modelos específicos en dos dimensiones no se hizo ningún progreso en esta dirección hasta el trabajo de *Foster y Metaxas* [16]. Para sus simulaciones utilizaron rejillas relativamente gruesas pero produjeron movimientos agradables de humo que remolinaban en tres dimensiones. Debido a que su modelo utiliza un esquema explícito de la integración, sus simulaciones son solamente estables si el paso del tiempo elegido es bastante pequeño. Esto hizo sus simulaciones relativamente lentas, especialmente cuando la velocidad del fluido es grande. Para aliviar este problema *Stam* [17] introdujo un modelo que es incondicionalmente estable y por lo tanto podría funcionar a cualquier velocidad. Esto fue alcanzado usando un esquema de advección *semi-Lagrangian*. A causa de que se utilizó un primer esquema de orden de integración, las simulaciones sufrieron de demasiada disipación numérica y el humo se desvanecía rápidamente.

Los esquemas *semi-Lagrangian* son muy populares en la comunidad atmosférica de las ciencias para modelar los fluidos de escala grande dominados por la advección constante donde se desean los pasos grandes de tiempo.

La clave de la simulación fluida es realizar pasos en tiempo y en cada paso del tiempo, determinar correctamente el campo actual de la velocidad. Se puede hacer esto solucionando un sistema de ecuaciones que describa la evolución del campo de la velocidad en un cierto plazo, bajo variedad de fuerzas. Una vez que se tenga el campo de la velocidad, se pueden hacer cosas interesantes con él, tal como usarlo para mover objetos, densidades del humo, entre otras.

Para describir el movimiento de un fluido se requiere una gran variedad de ecuaciones matemáticas que por cierto hoy en día existen gran cantidad de las mismas, pero podrían considerarse como la raíz de todas ellas a las ecuaciones de *Navier-Stokes*.

Las ecuaciones de *Navier-Stokes* [18] reciben su nombre de *Claude-Louis Navier* y *George Gabriel Stokes*. Se trata de un conjunto de ecuaciones en derivadas parciales no lineales que describen el movimiento de un fluido. Estas ecuaciones gobiernan la atmósfera terrestre, las corrientes oceánicas y el flujo alrededor de vehículos o proyectiles y en general, cualquier fenómeno en todo tipo de fluidos.

Estas ecuaciones se obtienen aplicando los principios de conservación de la mecánica y la termodinámica a un volumen fluido. Haciendo esto se obtiene la llamada formulación integral de las ecuaciones. Para llegar a su formulación diferencial se manipulan aplicando diferentes teoremas matemáticos, llegando así a la llamada formulación diferencial, que generalmente es más útil para la resolución de los problemas que se plantean en la mecánica de fluidos.

Como ya se ha dicho, las ecuaciones de *Navier-Stokes* son un conjunto de ecuaciones en derivadas parciales no lineales. No se dispone de una solución general para este conjunto de ecuaciones y salvo ciertos tipos de flujo y situaciones muy concretas no es posible hallar una solución analítica; por lo que en muchas ocasiones se ha de recurrir al análisis numérico para determinar una solución. A la rama de la mecánica de fluidos que se ocupa de la obtención de estas soluciones mediante el ordenador se la denomina dinámica de fluidos computacional (CFD, de su acrónimo anglosajón *Computational Fluid Dynamics*).

Capítulo 2 Descripción de los principales modelos y técnicas para la simulación de humo.

En este capítulo se hará una breve descripción de los principales métodos que existen para simular humo, incluyendo sus características principales, sus ventajas y desventajas, determinando el algoritmo para cada método. De esta forma al concluir este capítulo quedarán ya creadas las bases para determinar el algoritmo más óptimo para la simulación de humo.

2.1 Análisis de la determinación de los métodos escogidos.

Para la realización de este capítulo se hizo un estudio general de los diferentes métodos que existen para la simulación de humo, que ayudó en gran medida a seleccionar los más relevantes que son los que se describirán en este capítulo. Se estudió el método de Rápida Simulación de la Dinámica de los fluidos que se centra en las ecuaciones de *Navier – Stokes* y describe la utilización de las mismas en el campo de la simulación de humo, pero el resultado que arroja es una simulación en dos dimensiones que no es lo que se persigue en este trabajo. Igual se estudió el material Introducción a los Fluidos Dinámicos en la computación para lograr un mayor entendimiento sobre la materia. Y finalmente se escogieron los más prometedores que son los que se tratarán a continuación, el Modelo basado en las ecuaciones de *Navier - Stokes* usando hardware gráfico (GPU) y el Modelo de Flujo de Humo basado en las ecuaciones de *Reynolds* usando Metaballs.

2.2 Modelo basado en las ecuaciones de Navier-Stokes usando hardware gráfico (GPU).

Basado en las ecuaciones incomprensibles de *Navier-Stokes*, se implementó una solución 3D de fluidos. La solución trabaja en unos simples pasos.

Ahora se presenta una simple visualización para renderizar las animaciones de alta calidad del humo en tiempo real. Finalmente para manejar la interacción entre los fluidos y objetos estacionarios, se presenta un método simple.

Nomenclatura:

- d Densidad del Campo.
- f Fuerza del Campo.
- h Discretizacion Espacial.
- N Campo de la localización del vórtice.
- P Presión del Campo.
- T Temperatura del Campo.
- $T0$ Temperatura Mala.
- t Tiempo.
- u Velocidad del Campo.
- β Coeficiente de Expansión Termal.
- ϵ Factor del control del confinamiento del vórtice.
- η Campo del gradiente de la magnitud del vórtice.
- ω Vorticidad.

∇ Operador del gradiente. $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$

V Viscosidad cinemática.

ρ Densidad.

Con el aumento de la energía de cálculo en ordenadores personales estándares, más y más simulaciones físicas se ponen en juegos de computadora. Recientemente, un salto del quantum en energía de cálculo se ha hecho con la introducción del GPU programable, disponible en los adaptadores modernos de los gráficos. Esto conduce a una más alta calidad en los gráficos de juego, y los reveladores han comenzado a utilizarlo.

Este salto se abrirá para efectos más avanzados en próximos juegos. En los últimos años, muchos desarrolladores de juegos se han dedicado a la simulación de la física rígida del cuerpo. Además se cree que una de las nuevas características que utilizarán los diseñadores de juegos es la simulación de la dinámica de los fluidos. Para animar esto, se describe un método simple para alcanzar el humo animado en 3D en tiempo real.

Trabajo previo:

El movimiento de un fluido no turbulento, viscoso, se modela por las incompresibles ecuaciones de *Navier-Stokes* [18]

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{1}{\rho} \nabla p + \mathbf{f} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

Donde (1) describe el cambio en la velocidad fluida, y (2) hace cumplir la conservación de la masa.

Los métodos numéricos para solucionar estas ecuaciones se han investigado extensivamente durante la década pasada y hoy pueden ser generadas animaciones fluidas muy realistas. En años recientes, se han hecho grandes esfuerzos para llevar los métodos para solucionar estas ecuaciones a un mayor nivel: simulación en tiempo real.

Uno de los primeros métodos para simular las ecuaciones de *Navier-Stokes* en el mundo de los gráficos por computadora, fue el realizado por *Foster y Metaxas*. Los términos de las ecuaciones se solucionan

paso a paso, usando un acercamiento diferencial central de primer orden. La velocidad es hecha con libre divergencia, resolviendo el término de la presión iterativamente.

A causa de que la solución de *Foster y Metaxas* es basada en métodos explícitos, esta solo funciona para pequeños intervalos de tiempo.

Luego *Stam* dio el primer paso hacia la simulación en tiempo real, haciendo la solución incondicionalmente estable.

El término $(\mathbf{u} \cdot \nabla)\mathbf{u}$, es resuelto utilizando un esquema de integración Semi-lagrangian, donde los puntos se trazan al revés para encontrar diferentes velocidades. La difusión descrita por el término $\nu \nabla^2 \mathbf{u}$, es resuelto implícitamente utilizando un método multirejilla, el método de *Demmel JW* [19].

Para quitar divergencia del campo de la velocidad, un campo de la presión es calculado solucionando la ecuación de *Poisson*.

$$\nabla^2 p = -\nabla \cdot \mathbf{u} \quad (3)$$

Y además usando un método multirejilla, el gradiente de la presión es además usado para ajustar el campo de velocidad \mathbf{u} por:

$$\tilde{\mathbf{u}} = \mathbf{u} - \nabla p \quad (4)$$

Produciendo la libre divergencia del campo de velocidad $\tilde{\mathbf{u}}$:

Luego *Fedkiw, Jensen y Stam*, refinaron el mismo método de *Stam*, por el caso especial de simulación de humo.

Se rechazó el término de la difusión puesto que no resultó muy eficiente, debido a la viscosidad baja del aire y la tosquedad de la rejilla de la simulación. Para compensar algo del detalle en reducida escala se utiliza la fuerza del confinamiento del vórtice, descrito por *Steinhoff j y Underhill D*.

$$\mathbf{f}_{vc} = h \varepsilon (\mathbf{N} \times \boldsymbol{\omega}) \quad (5)$$

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} \quad (6)$$

$$\mathbf{N} = \frac{\boldsymbol{\eta}}{|\boldsymbol{\eta}|}, \quad \boldsymbol{\eta} = \nabla |\boldsymbol{\omega}| \quad (7)$$

Esto agrega un comportamiento que remolina, que parece muy natural para el humo, derivado de *Foster* y *Metaxas*, una fuerza termal simple de la flotabilidad, dada por:

$$\mathbf{f}_T = \beta (T - T_0) \cdot (0, 1, 0)^T \quad (8)$$

Permitiendo varios efectos de la temperatura, tales como radiadores que hacen el aire levantarse.

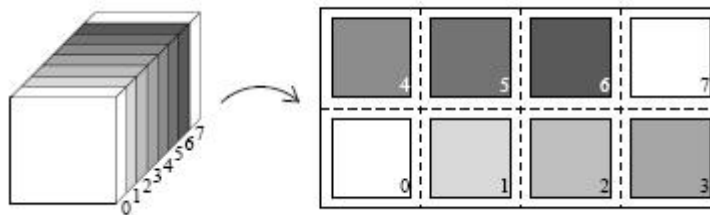


Figura 5. (Textura plana 3D - las rebanadas de un campo 3D se presentan como azulejos en una textura 2D)

Luego *Stam* presentó una solución fácil de implementar basándose en su propio método, el método multirejilla es reemplazado por uno mucho más simple, el método de *Gauss – Seidel*. La solución contiene aproximadamente 100 líneas de código.

Aunque este método es muy eficiente, las puestas en práctica del software todavía no son absolutamente bastante rápidas para las simulaciones en tiempo real en 3D. Para hacer esto, los campos del vector 3D se representan como texturas planas 2D, (Ver Figura 5).

Debido a la inhabilidad de los shaders de fragmento de leer y de escribir la misma memoria, el acercamiento del Gauss-Seidel de Stam se substituye por una variante del método de *Jacobi*, llamada *Gauss-Seidel, Rojo-Negro*.

El método:

Al igual que *Stam* aquí se le dan solución a las ecuaciones de *Navier-Stokes* en dos pasos. En el primer paso el campo de velocidad se actualiza de acuerdo a la ecuación (1). Esto conduce a un no conservador campo de velocidad. En el segundo paso, se quita divergencia, produciendo un campo de velocidad conservador de la masa y manteniendo la velocidad del campo.

Actualizando la velocidad:

La advección de la velocidad se soluciona usando el esquema *semi-Lagrangian*. Este método es estable para grandes pasos de tiempo, que es crucial para esta simulación en tiempo real. Siguiendo el método de Fedkiw, Jensen y Stam, se descuida el término de la difusión.

Dejando fuera el término de la difusión, se adopta la fuerza del confinamiento del vórtice. Esto compensa el detalle en reducida escala en el campo de velocidad que es disipado por la rejilla gruesa y agrega remolinos naturales al humo.

Para interactuar fácilmente con el humo, también se incluye la adición de un campo de fuerza externo, que puede responder a cualquier acción del usuario. Finalmente, se agrega una fuerza termal de la flotabilidad, para poder simular varios efectos relacionados con la temperatura.

Eliminando divergencia:

La divergencia es eliminada en tres muy simples pasos. Primero se calcula la divergencia. Luego se calcula el campo de presión iterativamente. Luego se usa el método simple de *Jacobi*, opuesto al método *Gauss-Seidel* que es implementable en un GPU. Finalmente, la velocidad del campo se ajusta según [16].

Implementación en GPU:

El GPU es un procesador programable disponible en los más recientes adaptadores gráficos. Aunque se diseñó para los procesadores de gráficos, el fin general del procesador en un GPU es atractivo debido a su energía de cómputo.

El GPU trabaja en corrientes de vértices y de fragmentos (píxeles con atributos, tales como profundidad y color) y tiene solamente acceso a lectura de la memoria a través de mapas de textura. Así, para poner a correr una solución de fluido en un GPU, el estado del fluido se debe representar con mapas de textura, y los pasos de la solución se deben poner en ejecución como shaders fragmentados [20].

Representación del campo del vector:

Se adoptó la representación del campo del vector de Harris MJ, Baxter WV, Scheuermann T y Lastra A. Un campo del vector 3D es representado presentando a cada rebanada como un azulejo en una textura 2D, como puede ser visto en la Figura 5. Esto permite que se actualice el campo entero en un solo render [21].

Shaders fragmentados:

Los shaders fragmentados son ejecutados dibujando primitivas regulares, tales como triángulos o cuadriláteros. Cuando estas primitivas son rasterizadas, los shaders fragmentados se ejecutan una vez en cada fragmento resultante, y la salida se escribe al buffer.

La entrada a un shader fragmentado, en el contexto de esta solución, es una coordenada de la textura referenciada a una celda específica en la rejilla de simulación. El valor de cualquier campo en este punto se puede mirar usando una instrucción de lectura de textura en el mapa respectivo de textura del campo. Para mirar valores vecinos, compensaciones se pueden agregar a la coordenada de la textura de entrada para producir coordenadas texturizadas que representando otras celdas en la rejilla. Los vecinos en las direcciones X y Y pueden ser encontrados fácilmente agregando los siguientes vectores:

$(1,0,0)^T$, $(-1,0,0)^T$, $(0,1,0)^T$, $(0,-1,0)^T$, respectivamente, a la coordenada de la textura de la celda actual. Para los vecinos en diversos azulejos, sin embargo, las compensaciones a los azulejos

respectivos deben ser tenidos en cuenta. En oposición a Harris MJ, Baxter WV, Scheuermann T y Lastra A se precalcularon compensaciones relativas entre los azulejos y se pasaron a los shaders fragmentados como coordenadas de entrada. En comparación con Harris MJ, Baxter WV, Scheuermann T y Lastra A esto se salva a una instrucción de textura-lectura de celda de la rejilla por paso de simulación, (aproximadamente de 10^6 instrucciones de lectura por marco con resolución 30x30x30).

Para cada paso de la solución se tiene un shader fragmentado que procesa solamente una celda en la rejilla para cada paso de la solución. La rejilla entera entonces es procesada dibujando primitivas que cubren todas las celdas interiores en la textura plana 3D, correspondiendo a las áreas sombreadas en la figura 1. Las celdas de límite (las áreas blancas en la Figura 5) se procesan por separado con especiales shaders fragmentados, para hacer cumplir las condiciones de límite deseadas del sistema.

Visualización:

Uno de los objetivos de las aplicaciones de simulación de fluidos en tiempo real es la presentación visual del fluido. Un método popular para renderizar humo en juegos es utilizar un sistema de partícula, y renderizar cada partícula usando una textura *BillBoard*, (Ver Figura 6).

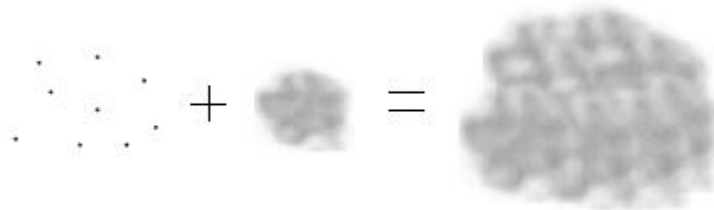


Figura 6. (BillBoarding, las partículas de la izquierda han sido sendereadas utilizando la textura de cuadrilátero mapeada en el centro rendereando el humo con apariencia de nube en la derecha.)

Sin embargo, el método de la partícula no es muy escalable desde los aumentos del tiempo de la simulación lineal con el número de partículas.

En lugar, para animar el humo se utilizó un campo de la densidad. La densidad se debe mover simplemente junto con la velocidad, además se modeló el movimiento de la densidad por:

$$\frac{\partial d}{\partial t} = (\mathbf{u} \cdot \nabla) d \quad (9)$$

Lo que es similar al término advectivo de (1), además se actualiza la posición de la densidad utilizando el esquema de advección Semi-lagrangian, según lo descrito para actualizar la velocidad del fluido.

Similar a todos los campos del vector, se representa este campo de la densidad en una textura plana 3D. Esto permite que se represente fácilmente humo coloreado y transparente, usando el modelo estándar del color RGB y el canal alfa de textura.

Esto corresponde realmente a tener cuatro campos de densidad (d_r , d_g , d_b , d_a), representando al rojo, verde, azul y canales alfa, respectivamente, (Ver Figura 7).



Figura 7 (La textura de la densidad representa 4 campos de densidad de rojo, verde, azul y densidades alfa. Cuando están renderizados, estos cuatro campos de la densidad representan el humo coloreado, transparente (derecha)).

Puesto que las operaciones fragmento son basadas en vector, se puede actualizar los cuatro campos de densidad a la misma velocidad que actualizando una sola capa de densidad.

La representación plana de la textura 3D del campo coloreado de la densidad permite que se renderice fácilmente el campo de densidad simplemente renderizando cada rebanada del campo como un Bill-Board.

Esto implica el dibujar de un cuadrilátero de textura mapeada por rebanada en la textura. Las rebanadas se dibujan en el orden de la distancia al espectador, comenzando por atrás. Cada rebanada se mezcla con el fondo, según el color de la densidad y el valor de alfa, dándole una mirada transparente. Este método es extremadamente rápido, además permite animaciones de alta tarifa.

Para evitar usar un volumen costoso de técnicas de rendereado, se utilizó solamente 16-bit de punto fijo de precisión en la textura de la densidad. De esta manera, la textura de la densidad se puede filtrar automáticamente por el adaptador de gráficos, mejorando calidad de la imagen y acelerando el proceso de actualización.

Por representar humo con un campo de densidad, la velocidad de la animación es independiente de la cantidad de humo en la simulación. Desafortunadamente, esto también significa que la resolución del humo está limitada de la misma manera que la resolución de la rejilla de la simulación.

Interacción con los objetos:

Para ampliar esta simulación de humo, se presenta una manera simple de simular la interacción con objetos estacionarios. Se representan objetos voxelizados en una textura plana 3D. En este mapa de obstáculo se representan las celdas ocupadas por un objeto con el valor 0, y las celdas ocupadas por el fluido con el valor 1.

Usando un shader fragmentado, los valores en cualquier campo se pueden enmascarar fácilmente en lo que respecta a los objetos en el fluido, permitiendo condiciones de límite simples. Así, los objetos estacionarios simples se pueden modelar en nivel de celda.

Resultados:

Ahora se muestran algunos ejemplos de humo simulado con esta solución. Todos los ejemplos han sido corridos en un adaptador de gráficos *ATI Radeon 9600 PRO*. La resolución de la rejilla es de 30 x 60 x 30, y la velocidad es aproximadamente de 30 fps.

Primero se muestra una corriente de humo simple. ([Ver Anexo 1](#)).

En el segundo ejemplo, se agregó un obstáculo esférico en la escena del ejemplo 1, con el objetivo de mostrar la interacción con los objetos estacionarios. ([Ver Anexo 2](#)).

Finalmente, para demostrar las capacidades del campo de la densidad coloreado, el ejemplo 3 muestra marcos de una animación usando dos diversos colores de humo. ([Ver Anexo 3](#)).

Se ha presentado un camino rápido de simular humo en 3D basado en la simulación de la dinámica de los fluidos. Teniendo en cuenta, que esta solución todavía no está optimizada completamente, y se estima una mejora posible en funcionamiento de cerca de un 10-15%.

Además, estos ejemplos han funcionado en un favorable adaptador de los gráficos de ATI Radeon 9600, que ya es considerado obsoleto por estándares actuales del hardware de los gráficos.

Así, aunque esta solución tiene algunos límites, da la impresión que los efectos de fluidos en tiempo real en juegos son posibles de implementar.

Puesto que la solución está basada en las ecuaciones de Navier-Stokes puede ser utilizada para otros propósitos además de animación de humo, por ejemplo la simulación y la animación de explosiones, animación de los efectos misceláneos del viento.

2.2.1 Algoritmo 1.

Inicio

1. Entrar y almacenar valores de variables y constantes.
2. Actualizar el campo de velocidad, para ello se realiza lo siguiente:
 - 2.1. Calcular la vorticidad.
 - 2.2. Calcular la flotabilidad.
3. Eliminar divergencia, para ello se realiza lo siguiente:
 - 3.1. Calcular la divergencia.
 - 3.2. Calcular el campo de presión.
 - 3.3. Ajustar el campo de velocidad.
4. Implementación en el GPU usando Shaders.
5. Visualización usando el Campo de Densidad.

Fin

2.3 Modelo de flujo de humo basado en las ecuaciones de Reynolds usando metaballs.

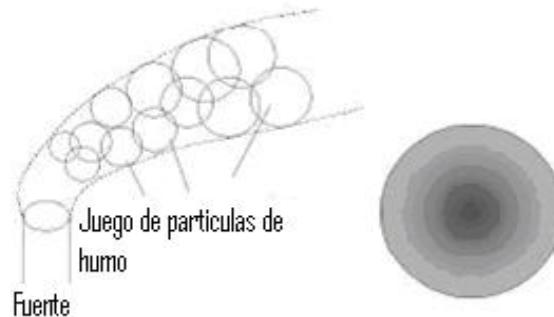
Recientemente, la demostración de fenómenos naturales como el humo se ha hecho un tema de interés a la gráfica realizada por ordenador. La capacidad de simular las formas complejas y el movimiento de las partículas de humo no es sólo importante sino es también un problema difícil de solucionar. En este material se propone un método de mostrar el humo que se arremolina en 3D, incluso la consideración de su paso a través de obstáculos.

Usando metaballs, se puede representar fácilmente la distribución tridimensional de la densidad del humo. Se soluciona la ecuación física del flujo, y se representa el vórtice usando el vector vorticidad. Por lo tanto, se puede hacer un modelo del flujo de humo aún si hay obstáculos en su camino.

Los métodos de mostrar humo son usados en varios campos, como simulación visual, entretenimiento, etc. Por lo tanto, hay una demanda verdadera en varias industrias para imitar el aspecto y movimiento del humo. El problema de modelar esto es el comportamiento complejo del humo en interacciones con el aire circundante y obstáculos. Se puede ver tal movimiento como una parte de tantas escenas diarias.

Se puede manejar el humo como un juego de pequeñas partículas de humo, pero la interacción entre el humo y obstáculos es difícil de evaluar cuando simplemente se consideran las fuerzas que actúan sobre estas partículas de humo. Se puede tratar con el humo como un tipo de fluido, y cuando se trata con ello de esta manera se puede simular más exactamente el flujo de humo alrededor de obstáculos. Por lo tanto, cuando se quiere describir un flujo realista y considerar su paso alrededor de obstáculos, se tiene que tener un poco en cuenta las propiedades del fluido del humo.

Modelación del humo usando metaballs:

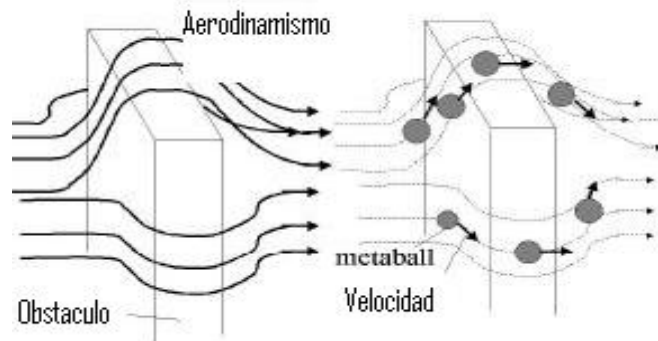


a) Juego de partículas de humo b) Ejemplo de la distribución de la densidad
Figura 8. (El soplo de humo y la distribución de la densidad)

El humo tiene una distribución de la densidad compleja debido a la naturaleza de sus partículas. Para modelar esta distribución usando el sistema de partículas, muchas partículas son necesarias. Para modelar la distribución de la densidad del humo con menos datos, se considera 'un juego' de partículas de humo.

Como se muestra en la Figura 8, se puede tratar con el humo como una combinación de racimos de humo. Este racimo es llamado "un soplo". El soplo es esférico y tiene una distribución de densidad, que satisface la condición siguiente: La densidad del soplo es más alta en el centro y entre más lejos del centro es más baja.

En general, la función Gaussiana puede ser aplicada para representar la distribución de la densidad del soplo de humo, pero esta tiene dominio infinito. Se usa un metaball para definir la distribución de la densidad del soplo. El campo de velocidad es calculado de antemano. Se hace el movimiento de metaballs considerando el cálculo del campo de velocidad. Como se muestra en la Figura 9 (a), el campo de velocidad representa el flujo evitando obstáculos, y entonces el flujo de metaballs también evita obstáculos como se muestra en la Figura 9 (b).



a) Campo de velocidad b) Flujo de humo usando el campo de velocidad

Figura 9. (El campo de velocidad y el flujo de metaballs).

El radio del soplo del humo no es constante, porque las partículas de humo se difunden en el soplo. Para simplificar esto, se asume que el radio del metaball se incrementa proporcionalmente con el paso del tiempo, y entre más grande el radio del metaball, menos denso es el metaball [22].

Cálculo del flujo utilizando la teoría del campo de velocidad:

En esta sección, se sacó un modelo numérico para el flujo de humo. El espacio alrededor de la superficie de la tierra es llamado "la capa de turbulencia divisoria," que tiene componentes de velocidad arbitrarios.

Siendo:

El vector velocidad: $\mathbf{u} = (\bar{u}, \bar{v}, \bar{w})$

El vector de velocidad promedio es: $\bar{\mathbf{u}} = (\bar{u}, \bar{v}, \bar{w})$

El vector de velocidad random es: $\mathbf{u}' = (u', v', w')$

Cada componente de velocidad es la suma del componente promedio y del componente random $\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}'$.

Un fluido es representado como una combinación de un campo de temperaturas y campo de velocidad. En la rotación, la flotabilidad y el movimiento del humo son modelados por las ecuaciones de Navier-Stokes.

Estas ecuaciones son populares, pero en ellas no se consideran los componentes de velocidad random de la turbulencia. Hay mejores ecuaciones para describir las fuerzas que actúan en el flujo del humo. Se usaron las ecuaciones de *Reynolds*, que consideran los componentes de velocidad random del flujo.

La ecuación del flujo:

La velocidad del fluido es afectada por varios factores como la convección, flotabilidad, presión, etcétera. Las ecuaciones Reynolds describen totalmente las fuerzas que actúan dentro del flujo de humo, y consideran los efectos de componentes de velocidad random. Las ecuaciones de Reynolds son:

$$\begin{aligned}
 \frac{\partial \bar{u}}{\partial t} &= -\left(\frac{\partial \bar{u}^2}{\partial x} + \frac{\partial \bar{u}v}{\partial y} + \frac{\partial \bar{u}w}{\partial z}\right) + \nu\left(\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} + \frac{\partial^2 \bar{u}}{\partial z^2}\right) - v_t\left(\frac{\partial \bar{u}^2}{\partial x} + \frac{\partial \bar{u}'v'}{\partial y} + \frac{\partial \bar{u}'w'}{\partial z}\right) - \frac{1}{\rho} \frac{\partial p}{\partial x}, \\
 \frac{\partial \bar{v}}{\partial t} &= -\left(\frac{\partial \bar{v}^2}{\partial y} + \frac{\partial \bar{v}u}{\partial x} + \frac{\partial \bar{v}w}{\partial z}\right) + \nu\left(\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} + \frac{\partial^2 \bar{v}}{\partial z^2}\right) - v_t\left(\frac{\partial \bar{v}^2}{\partial y} + \frac{\partial \bar{v}'u'}{\partial x} + \frac{\partial \bar{v}'w'}{\partial z}\right) - \frac{1}{\rho} \frac{\partial p}{\partial y}, \\
 \frac{\partial \bar{w}}{\partial t} &= -\left(\frac{\partial \bar{w}^2}{\partial z} + \frac{\partial \bar{w}u}{\partial x} + \frac{\partial \bar{w}v}{\partial y}\right) + \nu\left(\frac{\partial^2 \bar{w}}{\partial x^2} + \frac{\partial^2 \bar{w}}{\partial y^2} + \frac{\partial^2 \bar{w}}{\partial z^2}\right) - v_t\left(\frac{\partial \bar{w}^2}{\partial z} + \frac{\partial \bar{w}'u'}{\partial x} + \frac{\partial \bar{w}'v'}{\partial y}\right) - \frac{1}{\rho} \frac{\partial p}{\partial z} \\
 &+ F_{\delta v}, \tag{1}
 \end{aligned}$$

Donde ρ es la densidad, p la presión, ν el coeficiente de la viscosidad, y V_t el coeficiente de la viscosidad del remolino. Para calcular la Ecuación (1), se subdivide el espacio entero que contiene el flujo de humo en voxels. Usando la Ecuación (1), se pueden calcular los componentes de velocidad promedio en cada voxel.

En la parte derecha de la Ecuación (1), los terceros términos son los términos que incorporan los componentes de velocidad random. Estos términos generan el flujo random, que no puede ser representado usando las ecuaciones de Navier-Stokes. Se conoce que la fuerza del flujo random es un gran por ciento del flujo promedio.

Se puede determinar el componente de velocidad random como el valor random cuya fuerza es un gran por ciento de los componentes de velocidad promedios. La Figura 10 es una comparación del campo de velocidad con dos ecuaciones diferentes. Usando las ecuaciones de Navier-Stokes que sólo consideran

un flujo simple, pero los componentes random del flujo pueden ser representados por las ecuaciones Reynolds.

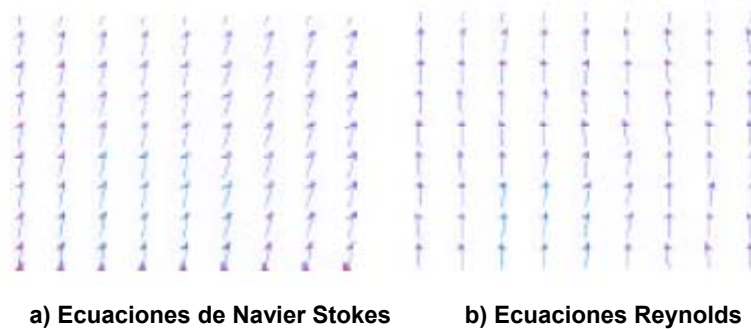


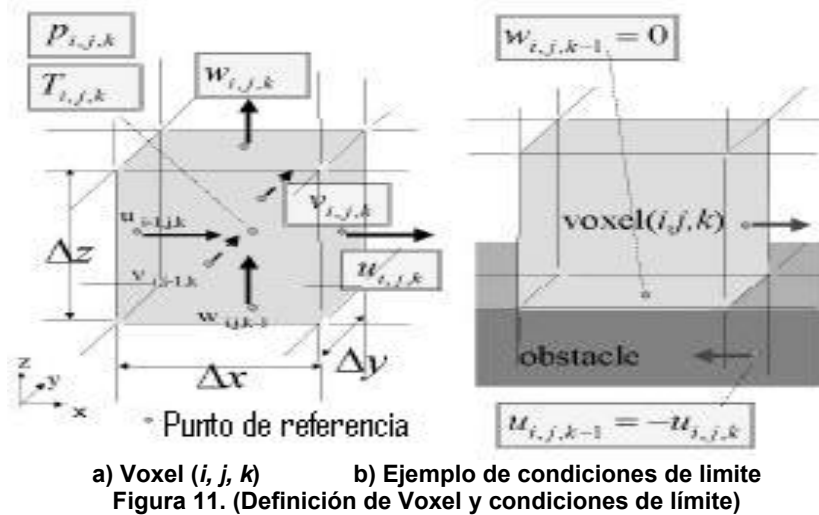
Figura 10. (Comparación del campo de velocidad utilizando dos ecuaciones distintas)

También en la derecha de la ecuación (1), los cuartos términos son los términos de presión. Cuando se calcula la Ecuación (1), se divide el término de presión y los otros términos. Tal división podría minimizar el error.

En la Ecuación (1), hay un término de flotabilidad F_{bv} sólo usado por la tercera ecuación. Para calcularlo, se tiene que considerar el campo de temperaturas. La temperatura T es definida como la suma del componente promedio \bar{T} y el componente random T' . Para calcular el campo de temperaturas, se solucionan las ecuaciones del flujo de calor, que tienen alguna analogía con las ecuaciones Reynolds:

$$\frac{\partial \bar{T}}{\partial t} = -\left(\bar{u} \frac{\partial \bar{T}}{\partial x} + \bar{v} \frac{\partial \bar{T}}{\partial y} + \bar{w} \frac{\partial \bar{T}}{\partial z}\right) + \alpha \left(\frac{\partial^2 \bar{T}}{\partial x^2} + \frac{\partial^2 \bar{T}}{\partial y^2} + \frac{\partial^2 \bar{T}}{\partial z^2}\right) + \alpha' \left(\frac{\partial^2 T'}{\partial x^2} + \frac{\partial^2 T'}{\partial y^2} + \frac{\partial^2 T'}{\partial z^2}\right), \quad (2)$$

donde α es el coeficiente de la difusividad del calor y α' es el coeficiente de remolino y de difusividad del calor. Antes de calcular el término de flotabilidad, se calcula el campo de temperaturas y F_{bv} se calcula luego tal como lo hicieron Foster y Metaxas [16].



Definición de voxel:

Para calcular la Ecuación (1), se subdivide el espacio entero que contiene el flujo de humo en voxels. Se pueden usar otros elementos en vez de voxels, pero la rejilla voxel regular hace fácil entender como calcular los parámetros físicos como velocidad, presión, y temperatura.

Se trata con los sistemas de coordenadas como el espacio tridimensional de *Euclides* en el cual el eje Z representa la dirección vertical y X y Y representan las direcciones horizontales.

El número de subdivisiones es **N_x**, **N_y** y **N_z** en las direcciones **X**, **Y** y **Z** respectivamente.

Dentro de cada voxel, hay algunas variables físicas del humo. El voxel puede ser etiquetado (i, j, k) ($0 \leq i < N_x$, $0 \leq j < N_y$, $0 \leq k < N_z$). El vector velocidad es $(U_{i,j,k}, V_{i,j,k}, W_{i,j,k})$. La temperatura es $T_{i,j,k}$, y la presión es $P_{i,j,k}$.

Para definir estas variables, se ponen algunos "puntos de referencia" en la superficie del voxel, y se calcula el valor de las variables en estos puntos. Como se muestra en la Figura 11 (a), en el centro de cada cara del voxel, se define un perpendicular de componente de velocidad a la cara, y en el centro del voxel se definen variables de temperatura y presión.

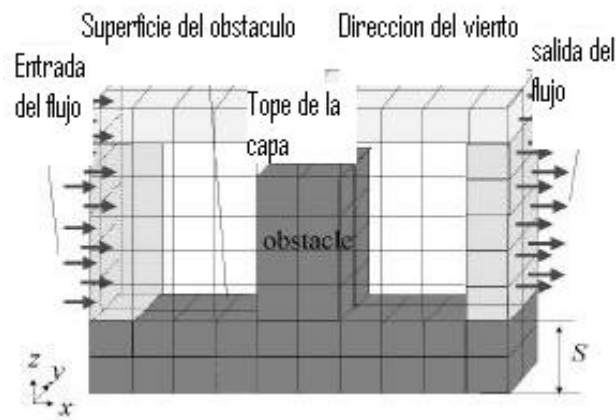


Figura 12. (Condiciones de límite)

Condiciones de límite:

Para calcular la Ecuación (1), se tienen que considerar algunas condiciones llamadas "condiciones de limite." Para las condiciones de límite, se considera la velocidad en la superficie del obstáculo, la velocidad de afluencia, la velocidad de efusión, y la velocidad en la capa superior.

- a) Se considera la velocidad en la superficie del obstáculo. La velocidad en la superficie del obstáculo es definida en cero. Por ejemplo, se considera la situación como se muestra en la Figura 11 (b).

En esta situación, el voxel $(i, j, k-1)$ está en el obstáculo y el voxel (i, j, k) está fuera. Se puede poner el componente de velocidad $W_{i, j, k-1}$ en la dirección vertical que sea cero. En la dirección horizontal, ya que no hay ningún punto de referencia en la superficie, es difícil definir el componente de velocidad en esta dirección. Se puede poner el componente de velocidad $U_{i, j, k-1}$ dentro del obstáculo para que sea $-U_{i, j, k}$ para aproximar la velocidad en la superficie del obstáculo.

- (b) Se tiene que definir el flujo en el límite del espacio analítico. Se llama tal flujo "la afluencia" (ver la Figura 12). Se asume que hay un flujo sólo en la dirección x. Para los voxels en el cual la etiqueta en la dirección x es cero (Voxel $(0, j, k)$, $0 \leq j < N_y$, $0 \leq k < N_z$) hay afluencia.

En una posición cerca de la superficie del obstáculo, la velocidad es cero y en una posición más alta se hace más fuerte y el viento fluye. Para modelar tal distribución de velocidad, se usa uno de los modelos de la distribución de velocidad.

c) Se asume otra vez que hay un flujo en la dirección x. En los voxels cuyas etiquetas en la dirección x son $N_x - 1$ (voxel $(N_x - 1, j, k)$, $0 \leq j < N_y$, $0 \leq k < N_z$), hay flujo en el límite del espacio analítico.

Se llama tal flujo "la efusión" (ver la Figura 12). Ya que el flujo en el espacio subdividido se hace estable, el término diferencial de la velocidad en la dirección horizontal en el voxel $(N_x - 1, j, k)$ es

cero $(u_{N_x-1,j,k} = 2u_{N_x-2,j,k} - u_{N_x-3,j,k})$

Cálculo del término de la presión:

Para calcular los términos de presión, se pueden calcular directamente las variables p y ρ , sin embargo, este cálculo es inestable. En este método, se asume que el fluido es incomprensible y se calculan términos de presión basados en esta asunción. Para minimizar el error, se divide el cálculo del término de presión y el cálculo de los otros términos en las ecuaciones *Reynolds*. El término de presión puede ser calculado solucionando la ecuación de *Poisson* (la preservación de la masa)

$$\nabla \cdot \mathbf{u} = 0. \quad (3)$$

Hay varios métodos de calcular el término de presión, y se aplica uno de ellos, que es simple de realizar. Primero, en cada voxel (i, j, k) , se calcula **divu** y **del**. Entonces se calcula el cambio de la presión Δp en un paso del tiempo. El Δp en el voxel (i, j, k) es expresado por:

$$\begin{aligned} del &= \Delta t \left\{ \left(\frac{\partial}{\partial x} \right)^2 + \left(\frac{\partial}{\partial y} \right)^2 + \left(\frac{\partial}{\partial z} \right)^2 \right\}, \\ divu &= \frac{u_{i,j,k} - u_{i-1,j,k}}{\Delta x} + \frac{v_{i,j,k} - v_{i,j-1,k}}{\Delta y} + \frac{w_{i,j,k} - w_{i,j,k-1}}{\Delta z}, \\ \Delta p &= -\frac{\Omega}{del} divu, \end{aligned} \quad (4)$$

Donde Δx , Δy , y Δz son los tamaños del voxel en las direcciones x , y , y z , respectivamente (ver la Figura 11 (a)). Y Ω es una constante. El usuario puede controlar Ω para acelerar el cálculo. Después de esto, se calculan los componentes de velocidad usando Δp . Se actualiza a

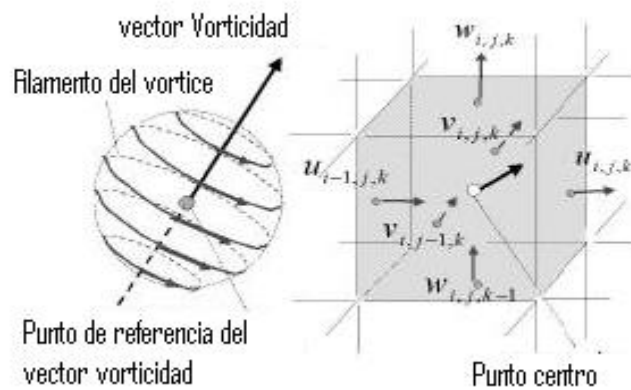
$$u_{i,j,k} \text{ a } u_{i,j,k} + \frac{\Delta p}{\Delta x}, \text{ y } u_{i-1,j,k} \text{ a } u_{i-1,j,k} - \frac{\Delta p}{\Delta x}.$$

Se actualizan los otros componentes de velocidad del mismo modo. Se tiene que calcular la Ecuación (4) recurrentemente. El cálculo se termina cuando el valor de Δp de cada voxel es más pequeño que el valor especificado.

Modelamiento de vórtices usando la vorticidad:

En el flujo turbulento, hay muchos vórtices generados por componentes de velocidad random. *Foster* trató de mostrar el flujo de humo que se arremolina usando sólo el campo de velocidad, pero para calcular los vórtices, se tiene que subdividir el espacio en un número grande de voxels, que hace el cálculo inestable.

Para modelar el vórtice, hay tres parámetros que tienen que ser considerados, el eje rotatorio, la velocidad rotatoria, y la velocidad convectiva del vórtice. Estos tres parámetros dependen de los componentes de velocidad random.



a) Vector vorticidad y filamentos del vórtice b) vector vorticidad en el centro del voxel (i, j, k).

Figura 13. (El vector vorticidad)

Se puede calcular el movimiento de remolino del metaball por la velocidad rotatoria \mathbf{V}_{rot} . La velocidad rotatoria del filamento de vórtice en el centro de metaball \mathbf{P} es calculada como sigue:

Para representar los vórtices simplemente, se usa el vector vorticidad (ver la Figura 13 (a)). El vector vorticidad es un vector que define la dirección del eje rotatorio y la fuerza de la velocidad rotatoria del vórtice. El vector vorticidad esta dado por:

$$\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right). \quad (5)$$

Cada voxel tiene seis puntos de referencia de los componentes de velocidad en su superficie (ver la Figura 13 (b)). Usando estos componentes de velocidad, se puede calcular fácilmente el vector vorticidad en el centro del voxel. Y los vórtices son definidos por el vector vorticidad. Estos vórtices son llamados como filamentos de vórtices. Para modelar la distribución de la velocidad de los filamentos de vórtice se usa el vórtice de Rankinn.

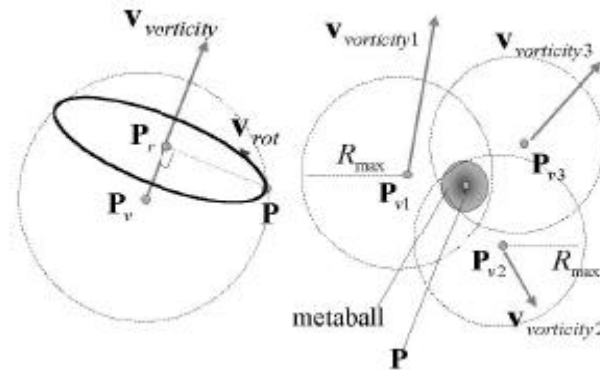
Para calcular la velocidad rotatoria del vórtice, se define un punto para referirse al vector vorticidad, y es nombrado "el punto de referencia del vector vorticidad". Para explicar el vector vorticidad, se define el vector posicional del punto de referencia como \mathbf{P}_v , el vector vorticidad en \mathbf{P}_v como $\mathbf{V}_{vorticidad}$, el paso del tiempo como Δt y la velocidad rotatoria como \mathbf{V}_{rot} . (Ver la Figura 14 (a)).

Ya que se modeló la distribución de la velocidad del vórtice usando el vórtice Rankinn, se puede calcular la velocidad rotatoria \mathbf{V}_{rot} en el punto \mathbf{P} alrededor de \mathbf{P}_v como sigue:

$$\mathbf{P}_r = \mathbf{P}_v + \left\{ \frac{(\mathbf{P} - \mathbf{P}_v) \cdot \mathbf{V}_{vorticidad}}{\|\mathbf{V}_{vorticidad}\|} \right\} \mathbf{V}_{vorticidad},$$

$$\mathbf{V}_{rot} = \frac{\mathbf{P} - \mathbf{P}_r}{\|\mathbf{P} - \mathbf{P}_r\|} \times \mathbf{V}_{vorticidad} \cdot V(\mathbf{P} - \mathbf{P}_v), \quad (6)$$

Donde $V(\mathbf{P} - \mathbf{P}_v)$ es una función para representar la distribución de la velocidad del vórtice Rankinn.

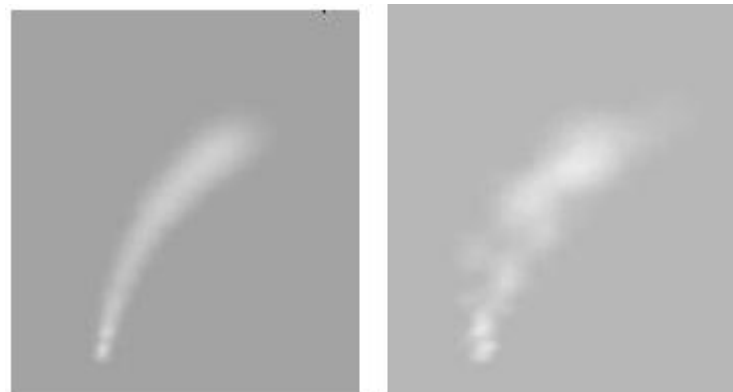


a) Velocidad rotatoria V_{rot} en P b) Puntos de referencia alrededor de P

Figura 14. (El cálculo de la Velocidad rotatoria)

Los puntos de referencia del vector vorticidad son movibles, mientras los otros puntos de referencia de las variables físicas son fijos.

La velocidad del metaball es la suma de la velocidad, que el campo de velocidad representa y la velocidad rotatoria calculada. La figura 15 muestra la comparación del humo con y sin vorticidad.



a) Humo sin vórtices b) Humo con vórtices

Figura 15. (Comparación de la vorticidad)

Ejemplos:

Se tiene variedad de ejemplos de este método. Para renderizar los metaballs, se usó el método de *Dobashi*, que usa el reenderezado de hardware OpenGL. Usando este método, se puede renderizar el humo realista más rápido que Stam.

La Figura 16 muestra el flujo de humo alrededor de la torre de un edificio. Un viento fuerte fluye de la fábrica a la torre. Se puede controlar el color del humo por el coeficiente de atenuación de las partículas de humo. La Figura 17 muestra el humo de una chimenea con una azotea, donde el humo evita la azotea de la chimenea.

Se calcularon estos ejemplos en el Intergráfico TDZ 2000 GX1 (Pentium III de 550MHz). En la Figura 16, el número de voxels es $20 * 20 * 20$. Hay 2,533 metaballs, y el tiempo de cálculo es 52.735 segundos. En la Figura 17 el número de voxels es de $40 * 40 * 20$, el número de metaballs es de 3,212, y el tiempo de cálculo es de 57,841 segundos.

A continuación se mostraran algunos ejemplos donde se aprecia la animación de humo realista empleando este método. (Ver Figura 16)

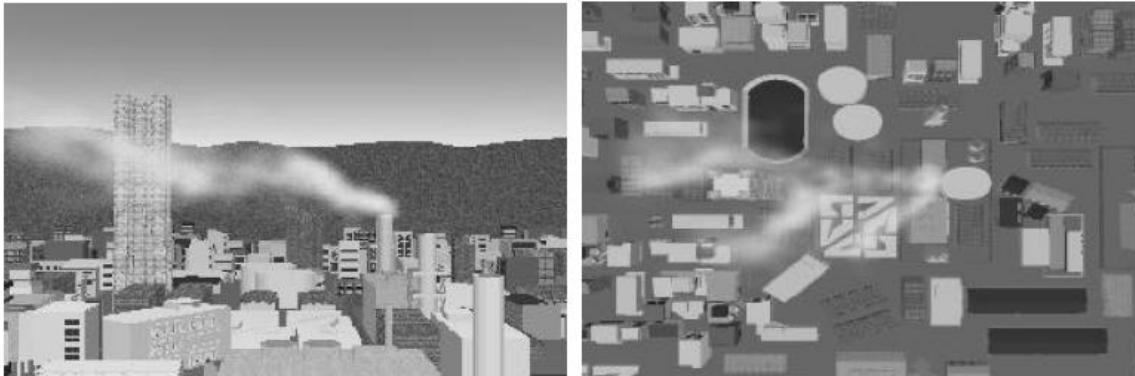


Figura 16. (Vista de humo.)



Figura 17. (Humo saliendo de una chimenea con una azotea en la parte superior)

En este material, se propone un método que demuestra que el humo se arremolina, que tiene una distribución tridimensional de la densidad, considerando su paso alrededor de obstáculos. También la utilización de las ecuaciones de *Reynolds*, que hace que el movimiento del humo ocurra en tiempo real.

2.3.1 Algoritmo 2.

Para calcular el flujo de humo, se usaron las ecuaciones de *Reynolds*, que consideran los componentes de velocidad random. Los cálculos necesarios para un marco de la animación son los siguientes:

Inicio

1. Entrar los valores de variables y constantes.
2. Almacenar valores.
3. Calcular la distribución de la temperatura (es decir el campo de temperaturas). Se necesita esto para el cálculo de la flotabilidad en las ecuaciones de Reynolds.
4. Calcular el campo de velocidad. En este cálculo, se realizó lo siguiente.
 - 4.1 Calcular los términos en las ecuaciones de Reynolds excepto el término de presión.
 - 4.2. Calcular el término de presión en las ecuaciones de Reynolds recurrentemente.

5. Movimiento metaballs según el campo de velocidad calculado.

5.1. Calcular el vector vorticidad en el centro del voxel.

5.2. Para cada P_v , se calcula la velocidad rotatoria V_{rot}

Fin

Capítulo 3 Propuesta de algoritmo para la simulación de humo.

En este capítulo se dará a conocer el algoritmo óptimo para la simulación de humo, teniendo en cuenta sus características principales así como sus ventajas y desventajas. También se hará una comparación entre ambos algoritmos teniendo en cuenta todas sus características y requisitos.

3.1 Algoritmo propuesto.

El algoritmo aquí seleccionado como el óptimo para simular humo es el segundo algoritmo analizado en el capítulo 2, o sea, el llamado *Algoritmo de Flujo de Humo usando Metaballs que toma en cuenta Obstáculos*. A continuación se mostrarán un conjunto de características del método seleccionado.

3.1.1 Características principales del método.

- 1) Se usa *metaball* para simular la distribución de la densidad del humo. Cada metaball tiene una distribución tridimensional de la densidad y se considera que el humo es una combinación de metaballs.
- 2) Se calcula el campo de velocidad, que puede ser expresado por las ecuaciones físicas del flujo. El resultado de las ecuaciones representa un flujo realista alrededor de obstáculos.
- 3) Se generan pequeños vórtices. El vector vorticidad define la fuerza y la dirección de los vórtices. Usando estos vórtices, se puede mostrar el humo que se arremolina.

En el método, se usan metaballs para modelar el humo basado en un modelo físico del fluido, de modo que se pueda representar la distribución tridimensional de la densidad del humo y esto requiere menos datos que el sistema de partículas. Se usan ecuaciones que tienen la velocidad arbitraria en cuenta a fin de representar un flujo más realista. Además, para representar el humo que se arremolina, los vórtices son evaluados usando el cálculo del flujo.

3.1.2 Ventajas y Desventajas.

Ventajas:

1) Se trata con el humo como una combinación de juegos de partículas de humo, llamadas soplos. Se modela el soplo usando un metaball. Usando metaballs, se puede representar una distribución tridimensional de la densidad para el humo.

2) Para representar el movimiento del flujo de humo teniendo obstáculos en cuenta, se usó un método de campo de velocidad. El campo de velocidad es generado solucionando ecuaciones físicas del flujo, las ecuaciones de Reynolds. Estas tienen en cuenta la velocidad random en la turbulencia. El resultado del modelo usando estas ecuaciones es un flujo realista que toma obstáculos en consideración. Para optimizar el cálculo del campo de velocidad, se puede usar localmente las ecuaciones de capa de turbulencia de límite en vez de las ecuaciones de Reynolds.

3) Pequeños vórtices en el flujo son generados del vector vorticidad y filamentos de vórtice. El vector vorticidad representa la fuerza y la dirección de los vórtices, y se puede calcular el vector vorticidad simplemente. Los vórtices en la turbulencia son tratados como filamentos de vórtice.

Desventajas:

Con la utilización de las ecuaciones de Reynolds, debido al grado de dificultad que estas presentan a la hora de implementar el algoritmo sería más costoso y consumidor que si se implementaran otras ecuaciones.

3.2 Comparación entre los algoritmos descritos.

A continuación se realizará una comparación entre los dos algoritmos analizados para tener una mayor seguridad de que el algoritmo escogido es el mejor desde el punto de vista los parámetros que lo caracterizan y que los hacen funcionales, para establecer esta comparación y para hacerla un poco mas

entendible, se nombrará al primer algoritmo, o sea, al algoritmo rechazado con el número 1 y al algoritmo escogido, con el número 2. (Ver Tabla 1.)

Algoritmos	Parámetros					
	Interacción con objetos estacionarios.	Procesador	Soporte de Hardware	Resolución de la rejilla	Velocidad	Ecuaciones Utilizadas
1	si	GPU	adaptador de gráficos ATI Radeon 9600 PRO	30 x 60 x 30	30 fps	Navier – Stokes
2	si	CPU	Intergráfico TDZ 2000 GX1 (Pentium III de 550MHz)	40 x 40 x 20	25 fps	Reynolds

Tabla 1. (Comparación entre algoritmos descritos)

Primeramente se analizará la interacción con los objetos estacionarios, que es lo que hace a este tipo de modelos más eficientes, en cuanto a este parámetro se tiene que ambos algoritmos en su funcionamiento tienen en cuenta la interacción del flujo de humo con los objetos estacionarios. En cuanto a los recursos que estos piden para tener un buen funcionamiento se tiene que el algoritmo 1 fue probado en un GPU (Graphic Process Unit), o sea, Unidad de Procesador de Gráfico, que no es más que una tarjeta de video como comúnmente se le conoce, mientras que el algoritmo 2 es funcional en el CPU enteramente, aunque puede ser implementado también en el GPU. Además 1 ha sido probado en un adaptador de gráficos ATI Radeon 9600 PRO que ya es considerado obsoleto por estándares actuales del hardware de los gráficos, donde la resolución de la rejilla es de 30 x 60 x 30 y la velocidad es de 30 fps, mientras que 2 ha sido probado en el Intergráfico TDZ 2000 GX1 (Pentium III de 550MHz) con resolución de la rejilla de 40 x 40 x 20 y la velocidad es de 25 fps.

En cuanto a la calidad de imagen obtenida se tiene que 2 supera a 1 debido a la utilización de las ecuaciones de *Reynolds* que tienen en cuenta los componentes de velocidad random que permiten un mayor realismo del humo y que permiten que este fluya en tiempo real.

De esta manera se concluye que el algoritmo nombrado con el número 2 es mas eficiente que el 1 por todo lo antes expuesto.

3.3 Pseudocódigo para el Algoritmo 2.

Inicio

Entrar (α , α , β , ρ , v , Vt , Nx , Ny , Nz , T_0 , Ω)

Hacer: (para $i = 0$ hasta $i < Nx$ incrementar i)

Hacer: (para $j = 0$ hasta $j < Ny$ incrementar j)

Hacer: (para $k = 0$ hasta $k < Nz$ incrementar k)

{

Calcular T para el voxel (i, j, k) =

$$\frac{\partial \bar{T}}{\partial t} = -(\bar{u} \frac{\partial \bar{T}}{\partial x} + \bar{v} \frac{\partial \bar{T}}{\partial y} + \bar{w} \frac{\partial \bar{T}}{\partial z}) + \alpha (\frac{\partial^2 \bar{T}}{\partial x^2} + \frac{\partial^2 \bar{T}}{\partial y^2} + \frac{\partial^2 \bar{T}}{\partial z^2}) + \alpha' (\frac{\partial^2 \Gamma'}{\partial x^2} + \frac{\partial^2 \Gamma'}{\partial y^2} + \frac{\partial^2 \Gamma'}{\partial z^2}),$$

}

Hacer: (para $i = 0$ hasta $i < Nx$ incrementar i)

Hacer: (para $j = 0$ hasta $j < Ny$ incrementar j)

Hacer: (para $k = 0$ hasta $k < Nz$ incrementar k)

{

Calcular para el voxel (i, j, k) =

$$\frac{\partial \bar{u}}{\partial t} = -(\frac{\partial \bar{u}^2}{\partial x} + \frac{\partial \bar{u}\bar{v}}{\partial y} + \frac{\partial \bar{u}\bar{w}}{\partial z}) + v (\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} + \frac{\partial^2 \bar{u}}{\partial z^2}) - v_x (\frac{\partial \bar{u}^2}{\partial x} + \frac{\partial \bar{u}'\bar{v}'}{\partial y} + \frac{\partial \bar{u}'\bar{w}'}{\partial z})$$

$$\frac{\partial \bar{v}}{\partial t} = -(\frac{\partial \bar{v}^2}{\partial y} + \frac{\partial \bar{v}\bar{u}}{\partial x} + \frac{\partial \bar{v}\bar{w}}{\partial z}) + v (\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} + \frac{\partial^2 \bar{v}}{\partial z^2}) - v_y (\frac{\partial \bar{v}^2}{\partial y} + \frac{\partial \bar{v}'\bar{u}'}{\partial x} + \frac{\partial \bar{v}'\bar{w}'}{\partial z})$$

Calcular $F_{bv} = \beta (T - T_0)(0,1,0)^T$

$$\frac{\partial \bar{w}}{\partial t} = -\left(\frac{\partial \bar{w}^2}{\partial z} + \frac{\partial \bar{w}u}{\partial y} + \frac{\partial \bar{w}v}{\partial z}\right) + v\left(\frac{\partial^2 \bar{w}}{\partial x^2} + \frac{\partial^2 \bar{w}}{\partial y^2} + \frac{\partial^2 \bar{w}}{\partial z^2}\right) - v_t\left(\frac{\partial \bar{w}^2}{\partial z} + \frac{\partial \bar{w}'u'}{\partial x} + \frac{\partial \bar{w}'v'}{\partial y}\right) + F_{bv}$$

}

Hacer: (para $i = 0$ hasta $i < Nx$ incrementar i)

Hacer: (para $j = 0$ hasta $j < Ny$ incrementar j)

Hacer: (para $k = 0$ hasta $k < Nz$ incrementar k)

{

Hacer para el voxel (i, j, k) :

Mientras (1)

Calcular:

$$del = \Delta t \left\{ \left(\frac{2}{\Delta x}\right)^2 + \left(\frac{2}{\Delta y}\right)^2 + \left(\frac{2}{\Delta z}\right)^2 \right\}$$

$$divu = \frac{u_{i,j,k} - u_{i-1,j,k}}{\Delta x} + \frac{v_{i,j,k} - v_{i,j-1,k}}{\Delta y} + \frac{w_{i,j,k} - w_{i,j,k-1}}{\Delta z}$$

$$\Delta p = -\frac{\Omega}{del} divu$$

Si $(\Delta p < Ve)$

Salir

Sino

$$u_{i,j,k} = u_{i,j,k} + \frac{\Delta p}{\Delta x}$$

$$v_{i,j,k} = v_{i,j,k} + \frac{\Delta p}{\Delta y}$$

$$w_{i,j,k} = w_{i,j,k} + \frac{\Delta p}{\Delta z}$$

$$u_{i-1,j,k} = u_{i-1,j,k} - \frac{\Delta p}{\Delta x}$$

$$v_{i-1,j,k} = v_{i-1,j,k} - \frac{\Delta p}{\Delta y}$$

$$w_{i-1,j,k} = w_{i-1,j,k} - \frac{\Delta p}{\Delta z}$$

}

Hacer: (para i = 0 hasta i < Nx incrementar i)

Hacer: (para j = 0 hasta j < Ny incrementar j)

Hacer: (para k = 0 hasta k < Nz incrementar k)

{

Calcular para el voxel (i, j, k):

Vector **vorticidad** =

$$\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)$$

}

Hacer: (para i = 0 hasta i < Nx incrementar i)

Hacer: (para j = 0 hasta j < Ny incrementar j)

Hacer: (para k = 0 hasta k < Nz incrementar k)

{

Calcular

$$\mathbf{P}_r = \mathbf{P}_v + \left\{ (\mathbf{P} - \mathbf{P}_v) \cdot \frac{\mathbf{v}_{vorticity}}{\|\mathbf{v}_{vorticity}\|} \right\} \mathbf{v}_{vorticity},$$

$$\mathbf{v}_{rot} = \frac{\mathbf{P} - \mathbf{P}_r}{\|\mathbf{P} - \mathbf{P}_r\|} \times \mathbf{v}_{vorticity} \cdot \nu(\mathbf{P} - \mathbf{P}_v),$$

}

Fin

Conclusiones

- Se realizó un estudio de los principales métodos matemáticos computacionales vinculados a la simulación de humo en entornos virtuales.
- A partir del análisis detallado de los métodos, se seleccionaron los algoritmos para la simulación de humo, lo que permitió establecer una comparación entre estos, arrojando la propuesta de un algoritmo óptimo para la simulación virtual de humo.
- Se dieron a conocer las características principales así como las ventajas y desventajas que presenta el algoritmo propuesto en la investigación.

Recomendaciones

A continuación se enumeran una serie de recomendaciones para la ampliación, modificación, mejora y construcción de nuevos algoritmos que posibiliten la simulación de humo en un entorno virtual lo más exacto posible:

1. Realizar un estudio detallado en el campo de la simulación visual, centrándose específicamente en la simulación de humo.
2. Analizar y valorar el método de las ecuaciones de Reynolds que usa metaballs para visualizar el humo, logrando más realismo y obteniendo mejores resultados, aunque trae consigo un mayor tiempo de cómputo.
3. Implementación del algoritmo propuesto para aumentar el realismo en el simulador de tiro de SIMPRO y así aumentar el nivel de preparación de las tropas.
4. Valorar por la empresa SIMPRO la utilización del algoritmo propuesto para la simulación de humo.

Referencias bibliográficas

- [1] WIKIPEDIA. *Smoke*, 2007. [2007]. Disponible en: <http://en.wikipedia.org/wiki/Smoke>
- [2] WIKIPEDIA. *Humo*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Humo>
- [3] WIKIPEDIA. *Fósforo blanco*, 2007. [2007]. Disponible en:
http://es.wikipedia.org/wiki/F%C3%B3sforo_blanco
- [4] WIKIPEDIA. *Hand grenade*, 2007. [2007]. Disponible en: <http://en.wikipedia.org/wiki/Grenade>
- [5] WIKIPEDIA. *Fumigation*, 2007. [2007]. Disponible en: <http://en.wikipedia.org/wiki/Fumigation>
- [6] WIKIPEDIA. *Señal de humo*, 2007. [2007]. Disponible en:
http://es.wikipedia.org/wiki/Se%C3%B1al_de_humo
- [7] ACTIVAMENTE. *Realidad virtual*, 2003. [2007]. Disponible en: <http://www.activamente.com.mx/vrml/>
- [8] WIKIPEDIA. *Vórtice*, 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/V%C3%B3rtice>
- [9] WIKIPEDIA. *Vorticidad*, 2006. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Vorticidad>
- [10] BLINN, J. *Metaball*, 1982. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Metaball>
- [11] REEVES, W. T. *Particle Systems*, 1983. [2007]. Disponible en:
<http://www.siggraph.org/education/materials/HyperGraph/animation/particle.htm>
- [12] MATEY, L. *APLICACIONES AVANZADAS DEL USO DE TEXTURAS PRECOMPUTADAS EN ENTORNOS DE SIMULACIÓN EN TIEMPO REAL*, 2003. [2007]. Disponible en:
www.ici.ubiobio.cl/revista/revista%202/45-58.pdf
- [13] GUSMAN, M. *Uso de tecnologías de hardware grafico en el apoyo al realismo en entornos virtuales arquitectónico*, 2004. [2007]. Disponible en:
http://digeset.ucol.mx/tesis_posgrado/Pdf/MARIO_EZEQUIEL_GUZMAN_GARCIA.pdf
- [14] ZAMBRANO, T. C. *ANÁLISIS DEL FLUJO TURBULENTO SOBRE UN PRISMA DE SECCIÓN TRANSVERSAL CUADRADA MEDIANTE SIMULACIÓN DE VÓRTICES GRANDES*, 2001. [2007].
Disponible en:

<http://www.revele.com.ve/pdf/fiucv/vol17-n2/pag97.pdf>

[15] HERZEN, K. A. V. *Object Modeling and Animation with Smoke*, 1984. [2007]. Disponible en: <http://graphics.cs.uiuc.edu/~linshi/amas.pdf>

[16] FOSTER, M. *Modeling the Motion of a Hot, Turbulent Gas*, 1998. [2007]. Disponible en: http://graphics.ethz.ch/Downloads/Seminar_Arbeiten/1998/pmuller.pdf

[17] STAM. *Stable Fluids*, 1999. [2007]. Disponible en: www.dgp.toronto.edu/people/stam/reality/Research/pdf/ns.pdf

[18] GRATTON, J. *Flujo Viscosos*, 1998. [2007]. Disponible en: www.lfp.uba.ar/Julio_Gratton/fluidos/07.%20Flujosviscosos.pdf

[19] ROBERT. *El Método MULTIGRID*, 2006. [2007]. Disponible en: www.mmc.igeofcu.unam.mx/smc-2006/Archivos/RYS.pdf

[20] HARRIS, M. J. *Fast Fluid Dynamics Simulation on the GPU*, 2004. [2007]. Disponible en: download.nvidia.com/.../SDK/Individual_Samples/DEMOS/OpenGL/src/gpgpu_fluid/docs/GPU_Gems_Fluids_Chapter.pdf

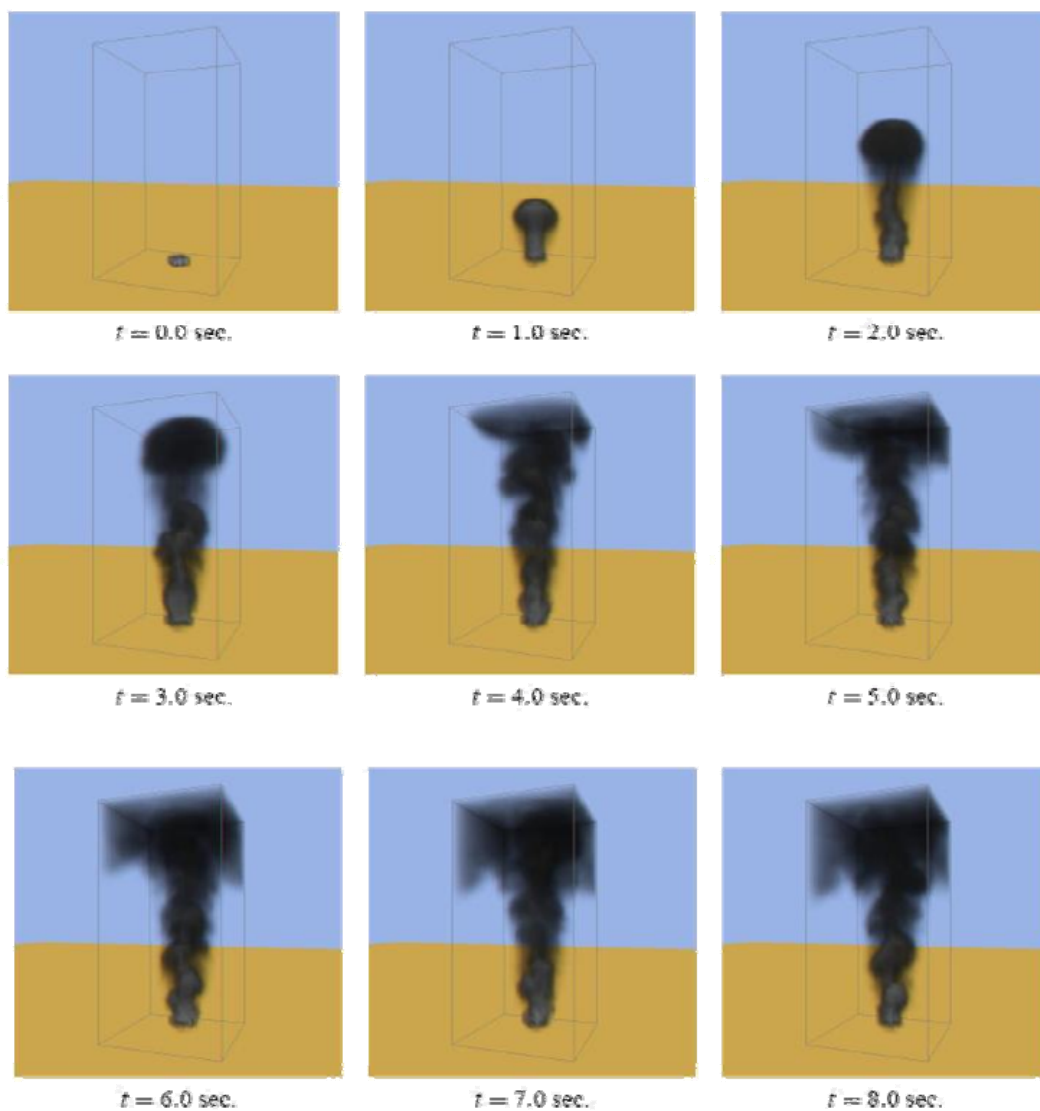
[21] BAXTER. *Physically-Based Simulation on the GPU*, 2004. [2007]. Disponible en: www.gpgpu.org/s2004/slides/harris.SimulationGPU.pp

[22] OSHIDA, S. and T. NISHITA. *Modelling of Smoke Flow Taking Obstacles into Account*, 2001. [2007]. Disponible en: ieeexplore.ieee.org/iel5/7096/19120/00883935.pdf

Anexos

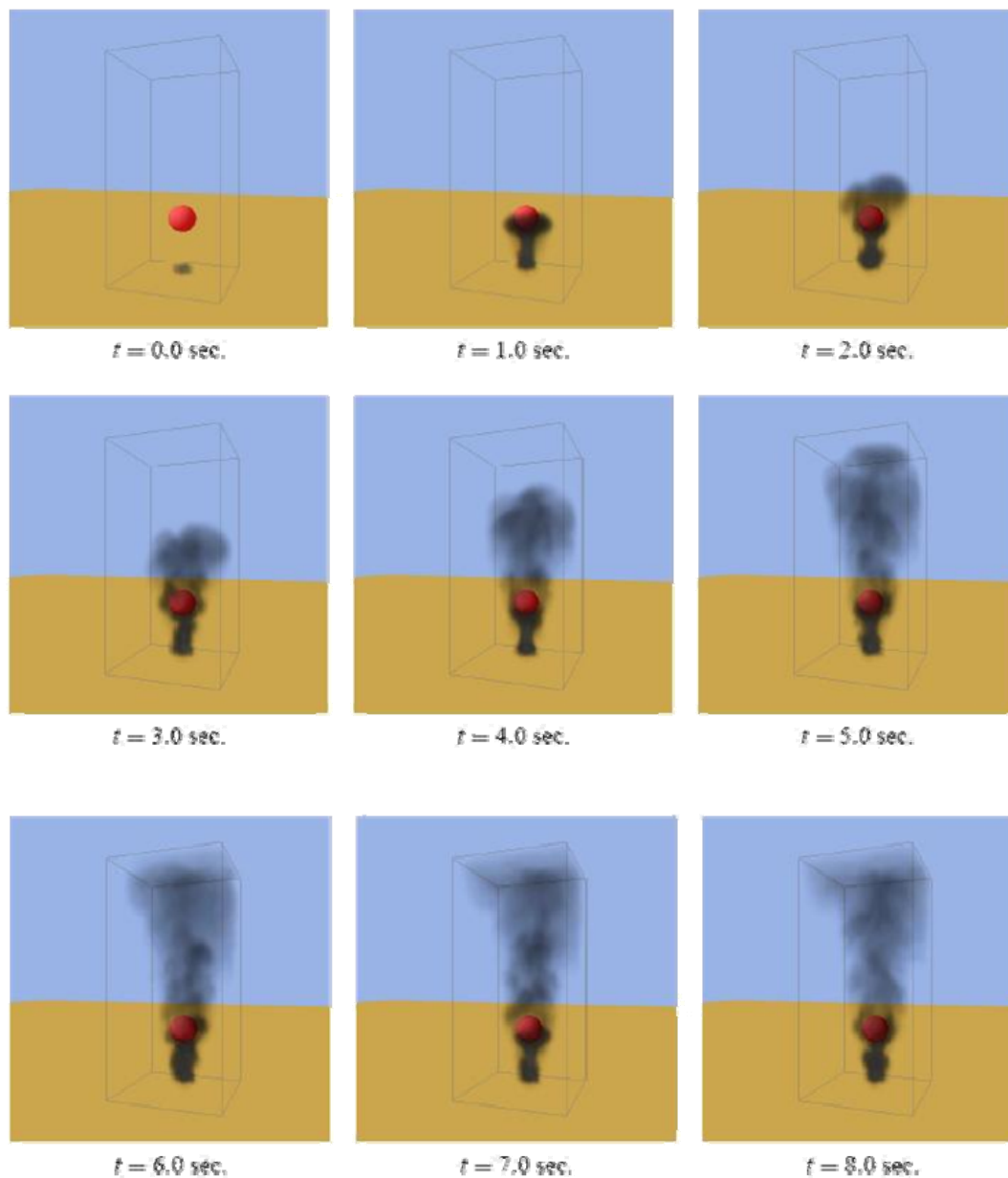
Anexo 1. Levantamiento de humo.

Levantamiento de Humo. El humo tiene remolinos que se ven naturales debido al confinamiento del vórtice, y la flotabilidad termal agrega realismo al movimiento general del humo.



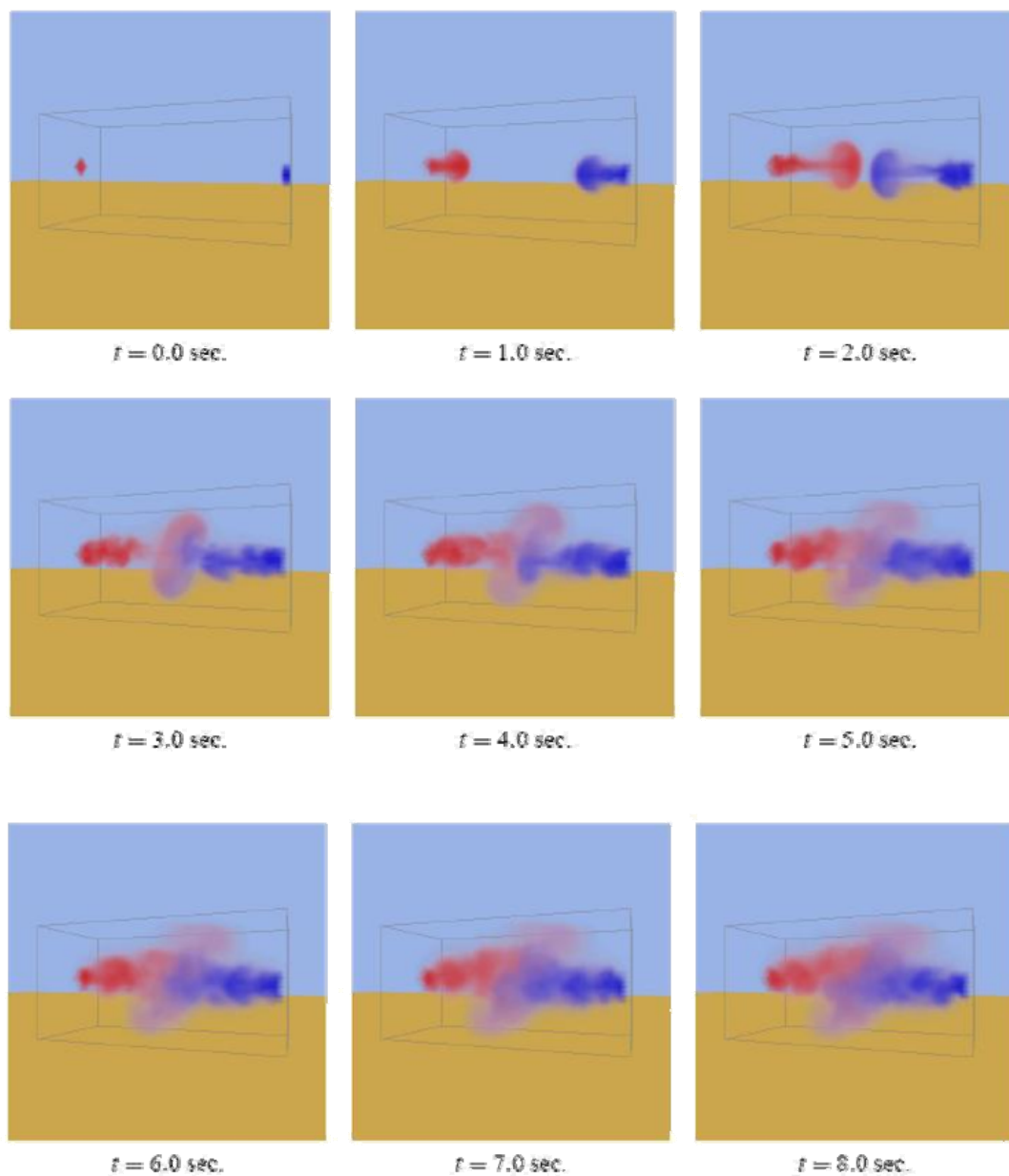
Anexo 2. Levantamiento de humo e interacción con un objeto.

Interacción con un objeto simple. El humo desarrolla una envoltura artificial alrededor del objeto debido a un manejo simple del límite, alcanzando un efecto de interacción. El objeto voxelizado se presenta visualmente, usando un acoplamiento esférico, para mejorar calidad visual.



Anexo 3. Humo coloreado.

Humo multicolor. Dos corrientes de humo con diversos colores se mezclan en un color mezclado.



Glosario de términos

Algoritmo: Un conjunto de reglas bien definidas para la solución de un problema en un número finito de pasos.

Billboards animados: Es una técnica de dibujo mediante la cual, se muestran objetos que geoméricamente deberían ser muy complejos mediante un simple plano y una textura. Dicho plano se ajusta en tiempo real para orientarlo hacia el observador, de modo que no note la falta de tridimensionalidad.

Informática: Es la ciencia que estudia el tratamiento automático y racional de la información.

Iteraciones: número determinado de veces que se realiza un proceso.

Metaball: Es una técnica de gráficos realizada por ordenador para simular interacción orgánica entre diferentes objetos en el espacio.

Realidad Virtual: Simulación de un medio ambiente real o imaginario que se puede experimentar visualmente en tres dimensiones. La realidad virtual puede además proporcionar una experiencia interactiva de percepción táctil, sonora y de movimiento.

Sistema de partículas: Es una técnica de los gráficos por computadora en 3D que se utiliza para crear una gran cantidad de objetos que obedezcan reglas de comportamientos bien definidos. Útil no solamente para las multiplicidades que controlan objetos discretos tales como asteroides o pájaros, también como herramienta para crear fenómenos naturales tales como fuego o humo.

Vórtice: Un vórtice es un flujo turbulento en rotación espiral con trayectorias de corriente cerradas. Como vórtices puede considerarse cualquier tipo de flujo circular o rotatorio que posea vorticidad.

Vorticidad: Es un concepto matemático usado en la dinámica de los fluidos que se puede relacionar con la cantidad de circulación o rotación de un fluido.

Voxel: Es la unidad cúbica que compone un objeto tridimensional. Constituye la unidad mínima procesable de una matriz tridimensional y es por tanto el equivalente del píxel en un objeto 2D.