

## ANIMACIÓN DE TORNADOS EN TIEMPO REAL MEDIANTE MOTORES FÍSICOS

**M. Lazo, C. García Bauza y A. Clause**

*CNEA-CONICET-CICPBA y Universidad Nacional del Centro, 7000, Tandil, Argentina  
marcos.g.lazo@gmail.com, cristiangb@gmail.com, clause@exa.unicen.edu.ar*

**Palabras clave:** Animación, Motores Físicos, Motores Gráficos, Tornado.

**Resumen.** Se presenta un modelo de motor físico para animación de tornados basado en un campo de fuerzas sobre partículas arrastradas por un vórtice vertical con su interacción con el suelo generando un flujo secundario de atracción hacia el eje de rotación. El motor físico alimenta un motor gráfico para animación de entornos 3D. La solución propuesta permite obtener tasas de renderizado aceptables utilizando un gran número de partículas u objetos interactuando con el tornado, logrando animaciones realistas a partir de efectos físicos. El modelo permite el control de la intensidad y la forma del tornado a través de parámetros constantes definidos en las expresiones del sistema de fuerzas.

## 1 INTRODUCCIÓN

Las aplicaciones de la visualización animada son clásicas en el campo del entretenimiento, pero también son importantes en las áreas de defensa, medicina, educación, construcción, entre otras. La mayoría de las animaciones desarrolladas hasta el presente son producto de efectos especiales que emulan la realidad, pero con algoritmos que no contienen base física. Recientemente se han comenzado a desarrollar algoritmos de animación basados en leyes físicas. La inclusión de comportamiento físico en aplicaciones de escenarios virtuales permite al usuario desempeñarse en un ambiente más similar al mundo real, el cual ya conoce, facilitándole el aprendizaje de las reglas necesarias para dominarlo. Un ejemplo claro de la relevancia de estos desarrollos es la evolución en calidad y velocidad en los videojuegos y simuladores de entrenamiento. El volumen de la investigación, la transferencia de tecnología, y de participación en el mercado en estas aplicaciones ha venido creciendo exponencialmente durante la última década, y todo indica que este crecimiento seguirá por mucho tiempo.

En el último tiempo, conjuntamente con la simulación de comportamiento de objetos sólidos, los fenómenos climáticos y ambientales han cobrado gran importancia, particularmente para ser incluidos en aplicaciones de tiempo real que incluyen escenarios con condiciones meteorológicas adversas; por ejemplo, simuladores de maniobras y rescate (D'Amato 2004a; Ki 2009). Situaciones como las que se presentan en este tipo de sistemas, permiten a los operarios entrenar, evitando poner vidas en riesgo y disminuyendo los costos de puesta en funcionamiento de los equipos necesarios.

Los fenómenos naturales como lluvia, nieve, tornados y tormentas de arena, junto a fenómenos físicos como fuego, humo, y explosiones son algunos de los ejemplos que están comenzando a ser incluidos en aplicaciones de tiempo real. Como es de esperar, el realismo de la simulación y la performance de la animación compiten entre sí. Por esta razón, los modelos físicos realistas en general son aplicados en técnicas offline debido al alto costo computacional que demandan; un ejemplo típico son los efectos especiales utilizados en cinematografía (Nguyen 2003; Fedkiw 2003). En cambio, en aplicaciones interactivas como juegos suelen utilizarse heurísticas o métodos aproximados para obtener tasas de renderizado altas a costa de perder realismo (Shen 2008; Beaudoin 2001). Esto demuestra la importancia de la investigación en esta área, desarrollando técnicas innovativas para simular estos fenómenos conservando las propiedades físicas de estos a costos computacionales aceptables para aplicaciones de tiempo real.

A lo largo de los años, se han propuesto varias soluciones para representar condiciones climáticas en tiempo real. Para representar tornados, Bouali (2003) propuso un enfoque heurístico para modelar la generación de vórtices, donde un modelo 2D no lineal representa una analogía dinámica de un torbellino que son conectados a través de vínculos apropiados para dar forma al tornado. Ding (2004) propuso un enfoque para simular tornados generando el flujo de aire a través de ecuaciones de Navier-Stokes e introduciendo partículas para definir la forma y volumen del tornado. Sin embargo, en este último caso, dada la gran cantidad de partículas introducidas, la tasa de renderizado obtenida es muy baja. Más tarde, Liu y otros (2007a) presentaron un modelo de dos fluidos (el vapor denso es tratado como fluido principal y las partículas girando dentro del tornado, como fluido secundario) simulando el flujo de aire del tornado y la interacción con los objetos que lo rodean, implementado en GPU para lograr simulaciones en tiempo real. Este mismo enfoque fue adaptado por Liu y otros para simular tormentas de arena (Liu 2007b).

Un tornado es un fenómeno natural muy complejo e impresionante, que es un objeto de gran interés en el campo de la animación gráfica. La película de Hollywood "Twister" fue un

hito importante en este tema, donde se aplicaron un conjunto de técnicas gráficas logrando efectos especiales como el arrastre de cuerpos, destrucción de construcciones, etc. Estos efectos de animación se lograban mediante resolución de ecuaciones de mecánica de fluidos y mecánica clásica, pero con tiempos de cálculo extremadamente largos. La animación en tiempo real de tornados para aplicaciones en juegos hasta ahora sólo podía hacerse con reproducciones de simulaciones previamente calculadas (que no son interactivas) o con emulaciones gráficas sin base física (que tienen poco realismo). El desarrollo de modelos de simulación con base física y rendering suficientemente rápido como para implementar animaciones de tornados interactivas, es un tema abierto que actualmente atrae el interés de los investigadores de computación gráfica.

En el presente trabajo se presenta un modelo de tornado basado en sistemas de fuerzas de atracción, rotación y repulsión para describir un vórtice vertical y el flujo secundario asociado con la interacción con el suelo, aplicado a entornos 3D utilizados en computación gráfica y simuladores de entrenamiento de tiempo real. La simulación tiene en cuenta la interacción de los vórtices con los objetos del ambiente tridimensional que se está visualizando, afectando a cuerpos sólidos y partículas cercanas.

## 2 MODELO DE FUERZAS

Un motor físico es esencialmente un resolvidor rápido de las ecuaciones de movimiento de un conjunto de partículas. Para construir un modelo de animación en base a estos motores es necesario modelar las fuerzas actuantes en cada partícula. En particular, para una simulación completa de los efectos de un tornado sobre los cuerpos que encuentra a su paso, habría que resolver las ecuaciones de fluidos en la capa atmosférica. Este problema matemático es muy complejo y no es viable para una animación en tiempo real. La alternativa que se propone en este trabajo es postular un modelo ideal aproximado del campo de velocidades del aire alrededor de un vórtice vertical y su flujo secundario al interactuar con el suelo. Este campo de velocidades a su vez ejerce una fuerza de arrastre sobre cada partícula localizada en el entorno del vórtice. Por simplicidad, se modeló directamente este sistema de fuerzas.

Supongamos un dominio tridimensional que contiene un vórtice vertical. La fuerza  $\overline{F}_{(P)}$  que se aplica a una partícula localizada en el entorno del vórtice esta dada por:

$$\overline{F}_{(P)} = \overline{fr}_{(P)} + \overline{fa}_{(P)} + \overline{ft}_{(P)} \quad (1)$$

Donde,  $\overline{fr}_{(P)}$  es la fuerza radial que controla la atracción o repulsión de la partícula hacia el eje del vórtice,  $\overline{fa}_{(P)}$  es la fuerza axial que modela el ascenso y descenso de la partícula en el flujo de aire y  $\overline{ft}_{(P)}$  es la fuerza azimutal producida por la rotación.

### 2.1 Movimiento del eje

La distancia de cada partícula  $P_j$  localizada en  $x_j, y_j, z_j$  al eje está dada por (Figura 1):

$$R_j = \sqrt{(x_j - e_x)^2 + (y_j - e_y)^2} \quad (2)$$

donde  $e$  es la posición del eje, que depende en general del tiempo y la altura  $z$ . El movimiento del eje  $e$  se modela discretizándolo en nodos (Figura 1), cada uno de los cuales

seguirá una ecuación de movimiento que simula estiramientos y ondulaciones.

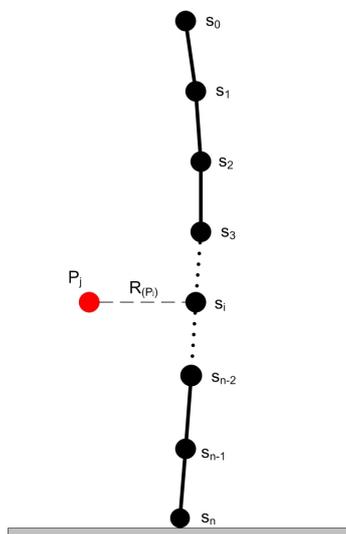


Figura 1: Representación del eje del tornado.

Cada nodo- $i$  se mueve según la siguiente fuerza:

$$\vec{F}_i = \frac{(\vec{x}_0 - \vec{x}_i)}{H - y_i} f_0 - \beta \frac{\partial \vec{x}_i}{\partial t} \quad (3)$$

Donde,  $H$  es la altura del tornado,  $\vec{x}_0$  es la posición del nodo superior del tornado, y  $\vec{x}_i$  es la posición del nodo- $i$ . Las constantes  $f_0$  y  $\beta$  permiten controlar efectos de flexión, ondulación y estiramiento.

## 2.2 Fuerza radial

La fuerza radial simula el arrastre debido al flujo secundario resultante de la interacción del vórtice principal con el suelo, generando un vórtice circular secundario centrado en el eje del tornado. El efecto visual es que las partículas cercanas al suelo son atraídas hacia el eje y las partículas alejadas del suelo son repelidas. Se propone un campo de fuerza radial  $f_r(P_i)$  de la forma:

$$f_r(P_i) = Fr \frac{(1 - y/H_o)^{Kr}}{1 + r/r_o} \quad (4)$$

Donde  $r$  es la distancia al eje,  $Fr$  y  $H_o$  son parámetros de control de la intensidad y la altura de inversión, y  $Kr$  y  $r_o$  son parámetros de forma. La Figura 2 muestra el perfil de la fuerza para diferentes valores de  $Kr$ .

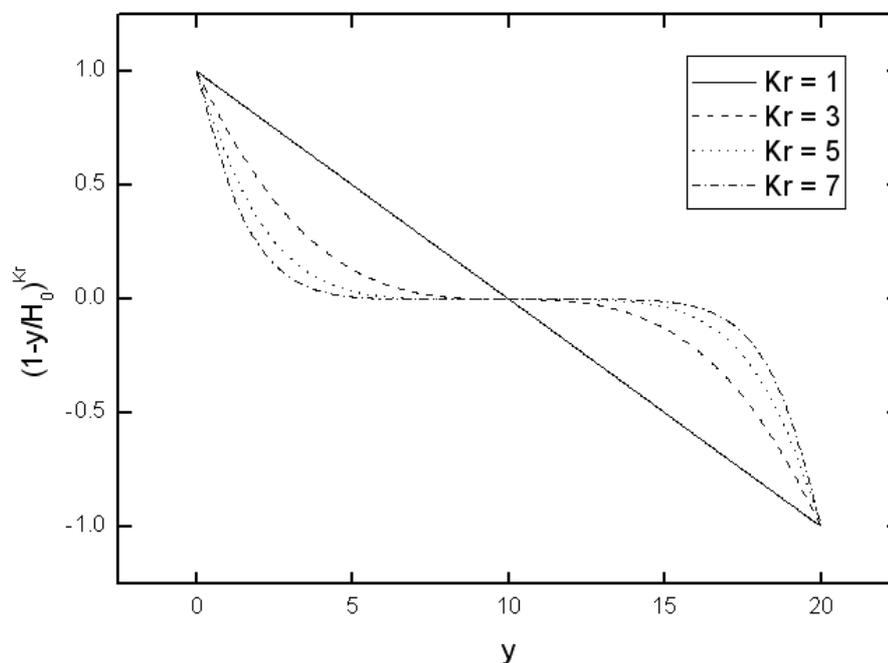


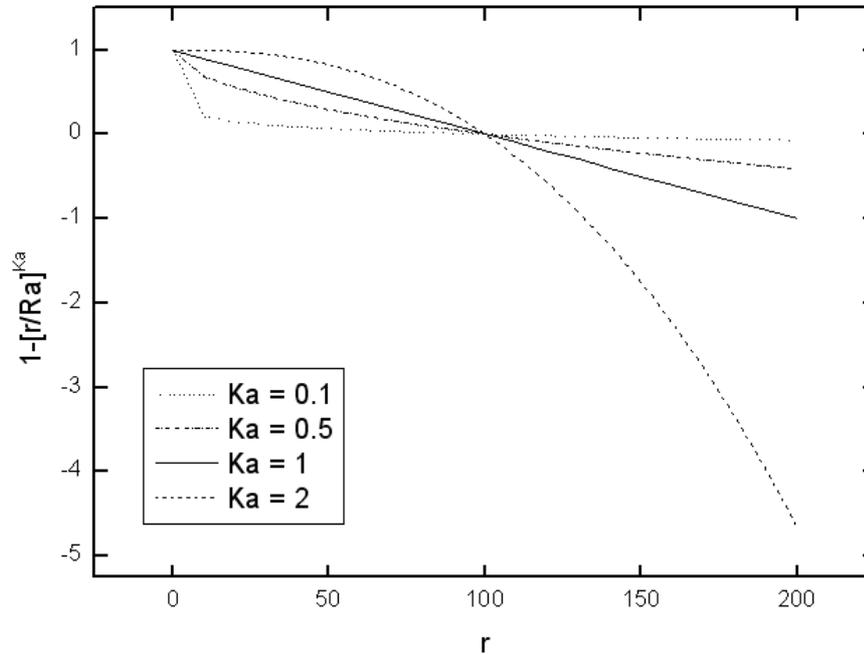
Figura 2: Perfil de fuerza radial en función de la altura ( $H_o = 10$ ,  $r_o = 1000$ ).

### 2.3 Fuerza axial

El vórtice secundario produce también una fuerza en la dirección del eje del tornado, que arrastra las partículas cercanas al eje hacia arriba y a las lejanas hacia abajo. Se propone la siguiente función para esta fuerza:

$$fa(P_i) = Fa \left[ c - \exp\left(-\frac{y}{H_o}\right) \right] \left[ 1 - \left(\frac{r}{Ra}\right)^{Ka} \right] \exp\left(-\frac{r}{r_1}\right) \quad (5)$$

donde  $Ra$  es el radio de inversión,  $Fa$ ,  $Ka$  y  $r_1$  son parámetros de intensidad y forma, y  $c$  es un parámetro mayor que 1 que controla a fuerza al nivel del suelo ( $y=0$ ). La Figura 3 muestra la distribución de la función para diferentes valores de  $Ka$ .

Figura 3: Perfil de fuerza axial. ( $Ra = 100$ )

## 2.4 Fuerza tangencial

El vórtice principal arrastra las partículas haciéndolas rotar alrededor de su eje. Siguiendo la conservación de momento angular, se propone la siguiente función:

$$f_t(P_i) = -\frac{Mo}{1+r/r_2} \exp(yKz) \quad (6)$$

donde  $Mo$ ,  $r_2$  y  $Kz$  son parámetros de intensidad y de forma.

## 3 IMPLEMENTACIÓN

### 3.1 Aplicación del modelo de fuerzas

Para aplicar el modelo de fuerzas, se utilizó el motor físico de propósito general NEWTON GAME DYNAMICS ([www.newtondynamics.com](http://www.newtondynamics.com)) que representa objetos como cuerpos rígidos, con funcionalidad para aplicar fuerzas o velocidades a estos, resolviendo las colisiones y calculando fuerzas resultantes de acuerdo a las propiedades de los materiales que tengan asignados dichos cuerpos. El motor físico se utilizó a través de una capa de abstracción (García et al. 2008) que permite abstraer al usuario de las particularidades de un conjunto de motores físicos soportados, facilitándole el desarrollo de aplicaciones que incluyan comportamiento físico. Sin embargo, el modelo de fuerzas aquí presentado puede ser

implementado utilizando cualquier motor físico de partículas (Seugling et al. 2006).

### 3.2 Representación visual del tornado

La representación visual del tornado y los objetos que interactúan con él se generaron mediante el motor gráfico Impromptu (D'Amato et al. 2004b). Como es común en los casos donde es necesario renderizar masivamente geometrías iguales, se utilizó la técnica de *billboarding* para representar las partículas de polvo que dan forma al tornado (Whatley 2005; Risser 2007). Estas partículas son modeladas utilizando planos cuadrados perpendiculares al vector de vista. Se aplican materiales con diferentes texturas y colores para evitar la homogeneidad (Figura 4). Además, los planos son escalados para lograr diferentes tamaños.



Figura 4: Tipo de textura utilizada para simular el efecto de polvo.

El efecto visual se obtiene aplicando a las partículas de polvo solamente la fuerza tangencial. Las partículas son distribuidas en un conjunto de anillos centrados en el eje del tornado. A su vez, cada partícula gira sobre su centro emulando el efecto de ascenso o descenso. El radio de cada anillo sigue la siguiente ecuación:

$$r(a_i) = c_1 \exp(Kz [H_0 - H(a_i)]) \quad (7)$$

Donde  $H_0$  representa la altura total del tornado,  $H(a_i)$  representa la altura del anillo  $i$ -ésimo.  $c_1$  y  $kz$  son parámetros de forma. El número de partículas de cada anillo es proporcional al radio del anillo. El tamaño  $\tau$  de las partículas se calcula como:

$$\tau = \varphi + \kappa \quad (8)$$

donde  $\kappa$  es un parámetro aleatorio que toma valores del segmento  $[0,5]$ , y  $\varphi$  es:

$$\varphi = [2\pi r(a_i)] / \eta(a_i) \quad (9)$$

donde  $\eta(a_i)$  es el número de partículas ubicadas sobre el anillo  $i$ .

La Figura 5 muestra una vista esquemática de la representación del tornado. Con líneas blancas se resaltan los anillos que componen el tornado. Los recuadros rojos muestran como son ubicadas las partículas sobre un anillo.

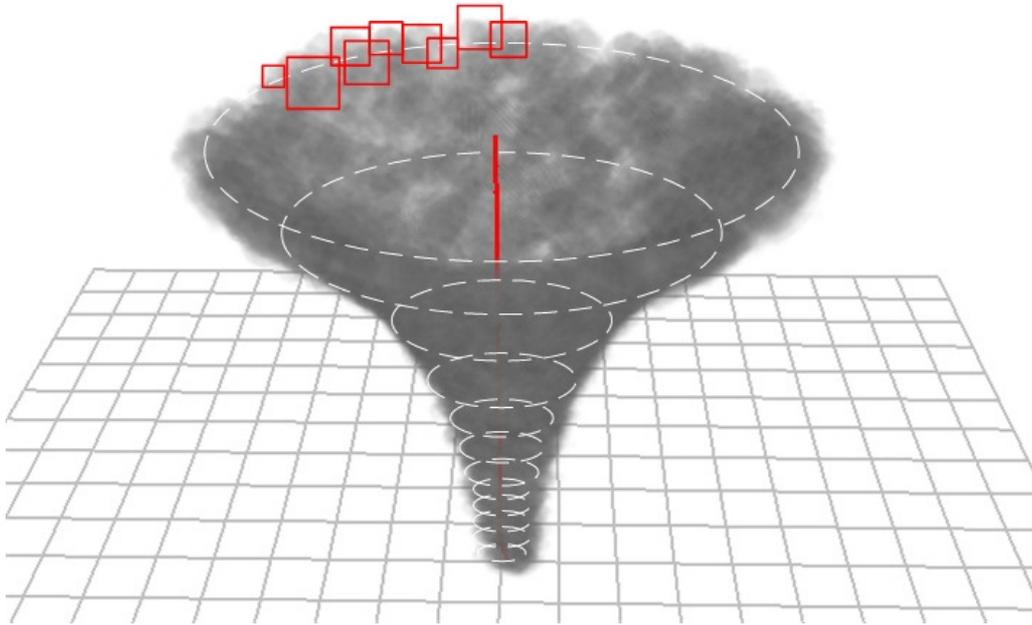


Figura 5: Vista esquemática de la representación del tornado

#### 4 RESULTADOS

La Figura 6a recrea un escenario abierto de un tornado de 200 metros de altura con 2000 partículas. La escena mantiene un conjunto de 300 modelos tridimensionales con un total de 350.000 polígonos aproximadamente. Como no es necesario calcular exactamente cada colisión, cada modelo es representado simplificado por una caja de volumen mínima (*Bounded Box*). Fueron asignados distintos valores de masa a cada objeto para obtener una variación de interacciones con el flujo de aire. Se pueden recrear distintas situaciones cambiando las condiciones iniciales, como el tipo de objetos, ubicación de los mismos, etc.

Adicionalmente a esta animación se implementaron vistas esquemáticas de la escena generada que permiten visualizar aspectos asociados al movimiento del tornado y los objetos que interactúan con él. Para permitir el seguimiento del movimiento y curvatura del tornado a medida que se traslada se visualiza el eje central con un conjunto de segmentos rojos (Figura 6b). Asimismo, se permite visualizar la historia de los objetos dentro del flujo de aire, representándola con líneas de trayectorias azules (Figura 6c).

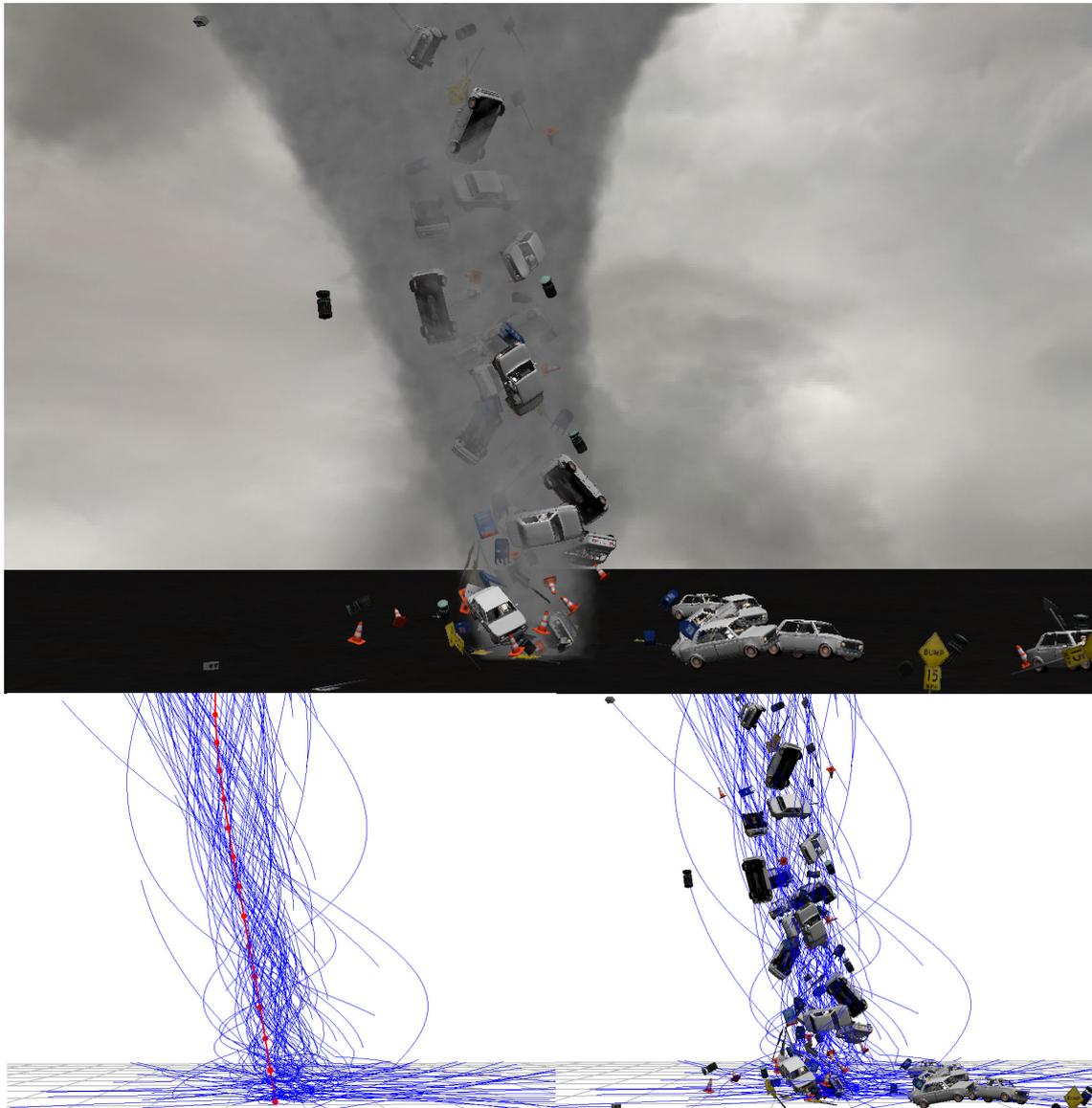


Figura 6: Vistas animadas y esquemáticas de una escena.

La Figura 7 muestra otra configuración del modelo de fuerzas. Se puede observar mayor turbulencia a alturas cercanas al suelo que hace girar rápidamente los objetos que se encuentran dentro del radio de acción del tornado. A medida que los objetos ascienden y se alejan del centro pierden velocidad y son despedidos, en cambio, los objetos que se mantienen cerca del centro ascienden a gran velocidad.



Figura 7: Vista esquemática y animada de una escena. ( $Fr=850$ ;  $H_0=200$ ;  $r_0=5$ ;  $Kr=1$ ;  $Fa=350$ ;  $c=1.25$ ;  $Ra=60$ ;  $Ka=3$ ;  $r_1=55$ ;  $Mo=100$ ;  $Kz=0.01$ ;  $r_2=2$ ).

La secuencia de capturas de la Figura 8 muestra el estado de la escena al cabo de 5 segundos. Un tornado cuya fuerza axial predomina sobre las demás fuerzas, arrastra y eleva a gran altura objetos que se encuentran a distancias cercanas de la base del tornado.

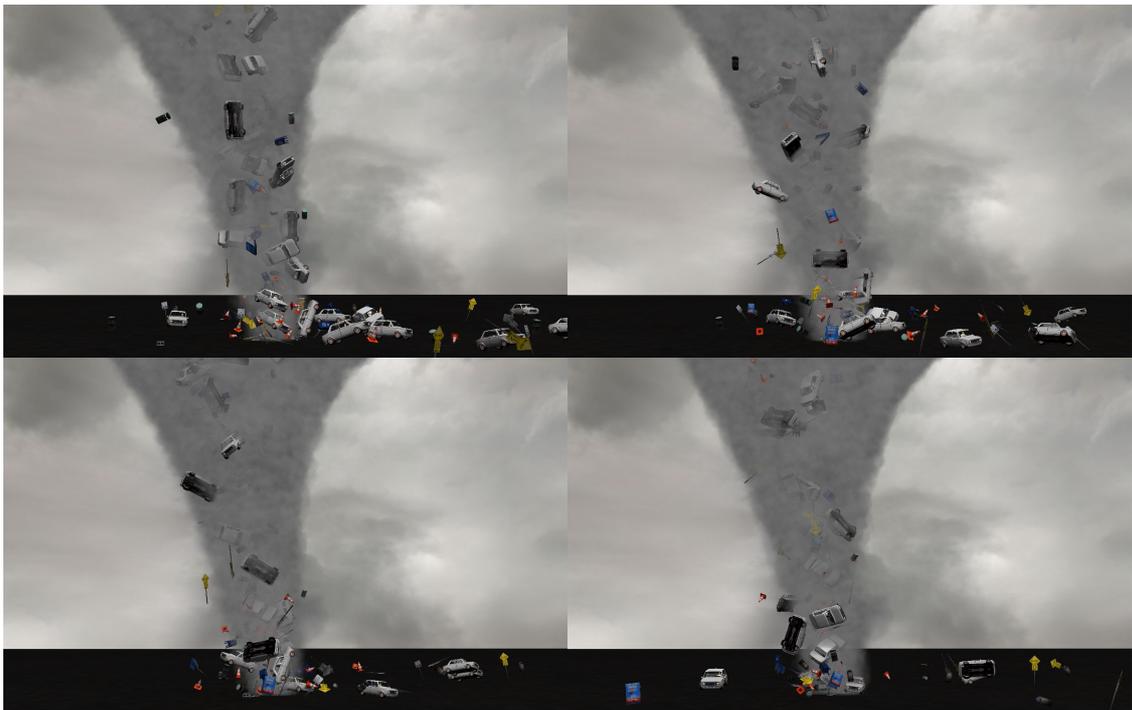


Figura 8: Tornado con mayor fuerza axial. ( $Fr=850$ ;  $H_0=200$ ;  $r_0=20$ ;  $Kr=1$ ;  $Fa=650$ ;  $c=1.25$ ;  $Ra=80$ ;  $Ka=1$ ;  $r_1=55$ ;  $Mo=150$ ;  $Kz=0.015$ ;  $r_2=2$ ).

La secuencia de capturas de la Figura 9 muestra una vista esquemática de un tornado con mayor poder de atracción y mayor fuerza tangencial que el tornado mostrado en la Figura 8. Los objetos son atraídos desde distancias mayores girando alrededor del eje del tornado a gran velocidad.

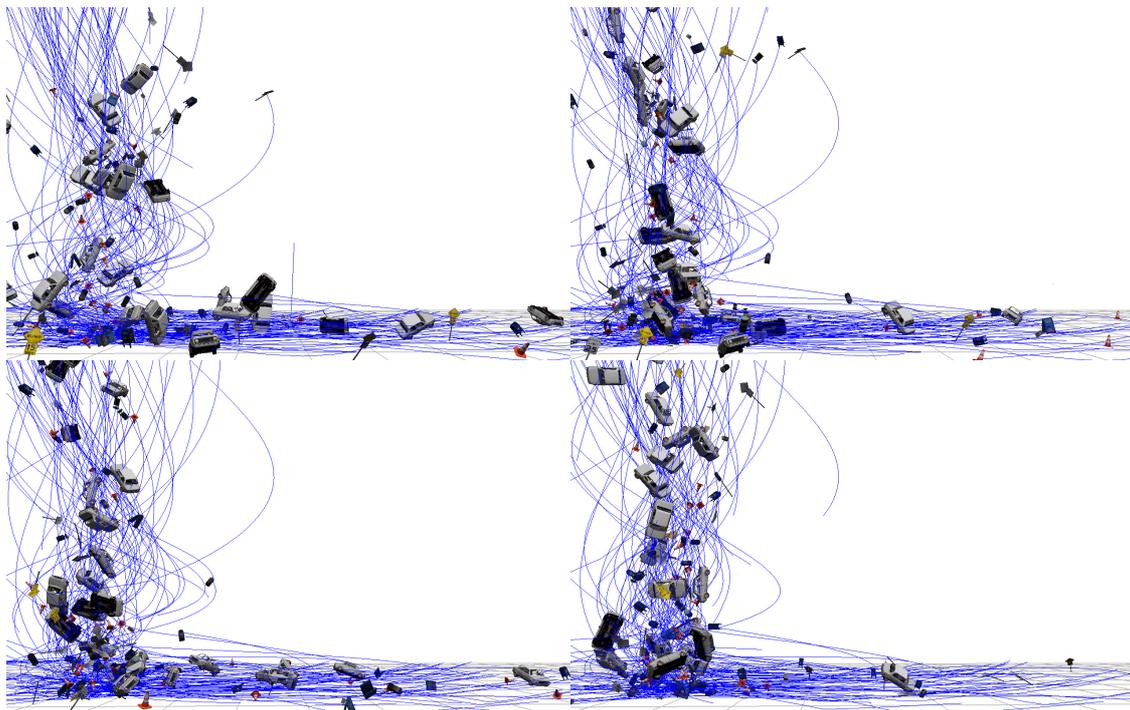


Figura 9. Tornado con mayor fuerza de arrastre por rotación. ( $Fr=1250$ ;  $H_0=200$ ;  $r_0=25$ ;  $Kr=1$ ;  $Fa=350$ ;  $c=1.25$ ;  $Ra=80$ ;  $Ka=3$ ;  $r_1=55$ ;  $Mo=350$ ;  $Kz=0.01$ ;  $r_2=1.5$ ).

## 5 CONCLUSIONES

En este trabajo se presentó un modelo de tornado basado en sistemas de fuerzas de atracción-repulsión y rotación hacia y alrededor del eje. La herramienta fue aplicada para producir animaciones en tiempo real de tornados interactuando con un conjunto arbitrario de objetos. El usuario puede ajustar los parámetros del modelo para definir la forma y poder del tornado. El modelo resultante es fácil de implementar utilizando cualquier motor físico de propósito general que permita aplicar fuerzas a los objetos.

## 6 REFERENCIAS

- Beaudoin P., Paquet S., Poulin P., *Realistic and controllable fire simulation*, Canadian Information Processing Society, 2001.
- Bouali S., *Twisting Rolls. An Heuristic Model and 3D Numerical Simulations of Vortex Patterns*, 2003.
- D'Amato J., García Bauza C., Vénere M., *Simulación del Entorno de una Embarcación Pesquera*, Actas de las 33 Jornadas Argentinas de Informática e Investigación Operativa, 2004.
- D'Amato J., García Bauza C., *Simulación de Escenarios Tridimensionales Dinámicos*. Tesis de grado de Ingeniería de Sistemas, Universidad Nacional del Centro. 2004.
- Ding X., *Physically Based Simulation of Tornadoes*, Master Thesis, Waterloo University, 2004.
- Fedkiw R., *Simulating Natural Phenomena for Computer Graphics*, Springer Verlag, 2003.
- García Bauza C., Lazo M., Vénere M., *Incorporación de comportamiento físico en motores gráficos*. Mecánica Computacional (ISSN 1666-6070), 27, 3023-3039, 2008.

- Ki J., *Interactive training simulator for aerial working platform in a virtual environment*, Computer Applications in Engineering Education, Wiley Periodicals, 2009.
- Liu S., Wang Z., Gong Z., Peng Q. *Real time simulation of a tornado*, Springer Berlin / Heidelberg, 2007.
- Liu S., Wang Z., Gong Z., Huang L., Peng Q. *Physically based animation of sandstorm*, Computer Animation and Virtual Worlds Vol. 18, 2007.
- Nguyen D., Enright D., Fedkiw R., *Simulation and Animation of Fire and Other Natural Phenomena in the Visual Effects Industry*, Western States Section, Combustion Institute, Fall Meeting, UCLA, 2003.
- Risser E., *True Impostors, Chapter 21, GPU Gems 3*, Addison-Wesley Professional, 2007.
- Seugling A., Rolin M., *Evaluation of Physics Engines and Implementation of a Physics Module in a 3d-Authoring Tool*. Masters Thesis, Department of Computing Science, Umea University, 2006.
- Shen H., Yin Y., Li Y. *Real-time dynamic simulation of 3D cloud for marine search and rescue simulator*, Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry, 2008.
- Whatley D., *Toward Photorealism in Virtual Botany, Chapter 1, GPU Gems 2*, Addison-Wesley Professional, 2005.
- Zienkiewicz, O.C., and Taylor, R.L., *The finite element method*, volume II. McGraw Hill, 1991.