


A modular authoring tool for atmospheric cloud modelling

Darío Rodríguez-Hernández¹ and Ignacio García-Fernández² 

¹ETSE - Universitat de València, Spain

²CoMMLab - Departament d'Informàtica. Universitat de València, Spain



Figure 1: *Varios ejemplos de nubosidad generados con la herramienta propuesta. De izquierda a derecha, y de arriba a abajo, un cúmulo, un cúmulo de estilo dibujos animados, un paisaje de cúmulos del mismo estilo, dos paisajes formados con cúmulos de aspecto realista y un paisaje de cirros.*

Abstract

Modelling of atmospheric effects and, in particular, of clouds is a recurrent topic in computer graphics. Achieving realistic results poses a great challenge, and manual modelling of such complex structures is unfeasible in production. The use of procedural techniques, either through simulation or heuristic methods, is, thus, frequent. In this work we propose a model that allows atmospheric cloud modelling with small computational effort and a good degree of control by the modeller, speeding up the artistic process. The model is designed hierarchically, can be extended easily due to its modular structure and it allows an intuitive control through a graphical user interface. The tests show a variety of effects that can be generated, and the influence of the different parameters on the render time of the final scene.

CCS Concepts

• **Computing methodologies** → Shape modeling; Volumetric models;

1. Introducción

Los fenómenos atmosféricos son un componente importante en la generación por computador de entornos exteriores; son un elemento que puede dotar de realismo a una escena y además tienen gran influencia en la iluminación, pudiendo condicionar mucho el aspecto de la imagen final. En el caso de las nubes, su modelado supone un gran reto debido a la naturaleza y variedad de sus for-

mas y a la complejidad del proceso físico subyacente [Gos21]. La utilización de simulaciones implica un alto coste computacional. Además, el control del resultado por medio de un simulador es difícil y requiere formación especializada. El uso de metodologías procedimentales, no basadas directamente en simulaciones físicas, permiten un mejor control sobre el modelado de las nubes con un menor coste computacional. En este trabajo presentamos un siste-

ma procedimental de modelado de nubes volumétricas, fácilmente parametrizable, que permite generar tanto nubes individuales como coberturas extensas de nubes.

La contribución principal de este sistema es la capacidad de controlar la forma de la nube y su diseño jerárquico y modular que permiten extender fácilmente su funcionalidad. Además, al estar diseñado como un generador de densidades sobre un dominio volumétrico, su implementación es simple y altamente paralelizable y puede emplearse de forma directa con un motor de render estándar.

2. Trabajo relacionado

En el estado del arte existen varias aproximaciones a la hora de modelar nubes. Por un lado, las metodologías basadas en física realizan simulaciones a partir de las leyes de la dinámica de fluidos y la termodinámica que consiguen muy buenos resultados a cambio de un gran coste en cómputo. Algunos de estos métodos utilizan las ecuaciones de Navier-Stokes, como el trabajo pionero de Kajiyama et al. [KVH84] y posteriores modificaciones de Miyazaki et al. [MDN02], incluyendo variables como la temperatura, enfriamiento adiabático y velocidad. Entre estos trabajos encontramos propuestas basadas tanto en el enfoque Euleriano [HBSL03] como el Lagrangiano [GN17, BDY15]. Otras metodologías basadas en física emplean el potencial de Lennard-Jones [EES*16] o los diagramas de termodinámica SkewT/LogP muy utilizados en meteorología [DG17].

Debido al coste computacional y a la complejidad de los modelos basados en física, se han propuesto estrategias heurísticas, que crean nubes procedimentales empleando métodos alternativos, ya sea por medio de fórmulas simplificadas o de otras aproximaciones, consiguiendo resultados rápidos en cómputo. Algunos de los primeros métodos aplicaban texturas 2D y 3D a planos, elipsoides y otras geometrías [Gar85, EP90, ES00, TB02a]. A menudo se hace uso de texturas procedimentales [Sak93, HKA06, Sch15, WCGG18] que, por medio de la combinación de diferentes capas, permiten conseguir resultados de calidad. Otra aproximación más directa es el modelado de la superficie de las nubes. Para ello se han usado isosuperficies y el algoritmo *marching cubes* [TB02b], superposición de elipsoides y funciones de ruido [SSEH03, BN04], a partir de superficies B-Spline [Set97], haciendo uso de figuras 2D y 3D y transformarlas en nubes con dicha forma [WBC08, SBR10, YW11, MMPZ12] o con la generación de metabolas a partir de información meteorológica [HHS07].

Este último trabajo nos introduce en el uso de estrategias basadas en datos, que emplean datos reales para conseguir los resultados deseados. Algunos autores usan imágenes tomadas desde tierra para generar entornos [ALM*14, YLH*14], o imágenes de satélite para paisajes a gran escala [DNYO99, KK09, DYN10, DSY10, WYMS12, YLHY13]. Iwasaki et al. [IND11] presentan un método basado en simulaciones físicamente realistas que emplean para generar una base de datos con la que crean animaciones. Algunos de los métodos que aparecen en la literatura no pueden categorizarse en una única metodología, sino que combinan varias estrategias consiguiendo un equilibrio entre rigor físico, control y coste computacional. Así, encontramos modelos híbridos entre métodos basados en física y estrategias heurísticas

cas [AFGC07, DYN06, LHW*07] o entre métodos heurísticos y basados en datos [IND11].

Nuestro método sigue una metodología procedimental heurística basada en la aplicación de ruidos de forma similar a la propuesta de [HKA06]. En nuestro caso, esta estrategia se combina con la aplicación de varios niveles de ruido de Perlin [Per02] y de Worley [Wor96] siguiendo la propuesta de [Sch15], pero ampliándola y añadiendo una estructura modular.

3. Descripción del modelo

El modelo propuesto se basa en la generación de una distribución volumétrica de densidad que se empleará en el proceso de generación de las imágenes (*render*). Para la generación de una nube se parte de un dominio $\Omega \subset \mathbb{R}^3$, que supondremos que es un prisma rectangular alineado con los ejes. Nuestra definición de nube será una función $\rho : \Omega \rightarrow [0, 1]$ que, dado $\mathbf{x} \in \Omega$, nos determinará un valor de densidad $\rho(\mathbf{x})$. El objetivo será, por tanto, construir funciones de densidad que, una vez renderizada la información volumétrica, resulten en formaciones nubosas.

Para ello emplearemos un esquema modular, en el que diferentes módulos generarán una serie de funciones de densidad parciales que se combinarán para obtener la función de densidad de la nube. El sistema propuesto cuenta con dos módulos principales: el módulo de forma y el módulo de detalle. El módulo de forma base, descrito en la Sección 4, genera una función de densidad, ρ_b , que define la forma global de la nube y permite escoger entre varios tipos de nubosidad. A continuación, el módulo de detalle, descrito en la Sección 5, genera una nueva función de densidad, ρ_d , que permite introducir detalle en las nubes, consiguiendo resultados más realistas y con mayor variabilidad. En la Figura 2 estos dos grupos corresponden a los dos grandes bloques grises.

Una vez calculada la función de densidad con detalle, ρ_d , se lleva a cabo un proceso de limpieza, que elimina restos de nube alejados de la densidad base. Por último, la densidad obtenida se binariza antes de enviarla al proceso de render. Estos dos últimos pasos se describen en la Sección 6. En las diferentes secciones se indica la manera en que estos elementos se combinan para dar lugar a la nube final.

4. Generación de la forma inicial de la nube

Este módulo del sistema se encarga de generar la forma inicial de la nube, lo cual nos permitirá generar varios tipos de nubes. Nuestra propuesta incluye la generación de cúmulos individuales, paisajes de cúmulos y paisajes de cirros. A continuación describimos el proceso propuesto para cada una de estas formas. Durante el desarrollo usaremos la siguiente notación. Dado que el dominio para la generación de la nube es un prisma rectangular, podemos considerarlo como el producto cartesiano de tres intervalos, $\Omega = \mathbf{I}_x \times \mathbf{I}_y \times \mathbf{I}_z$. Un punto $\mathbf{x} \in \Omega$ tendrá coordenadas $\mathbf{x} = (x, y, z)$, $x \in \mathbf{I}_x$, $y \in \mathbf{I}_y$, $z \in \mathbf{I}_z$. Usaremos la dirección del eje z como dirección vertical. Sin pérdida de generalidad, supondremos que el dominio está centrado en el origen.

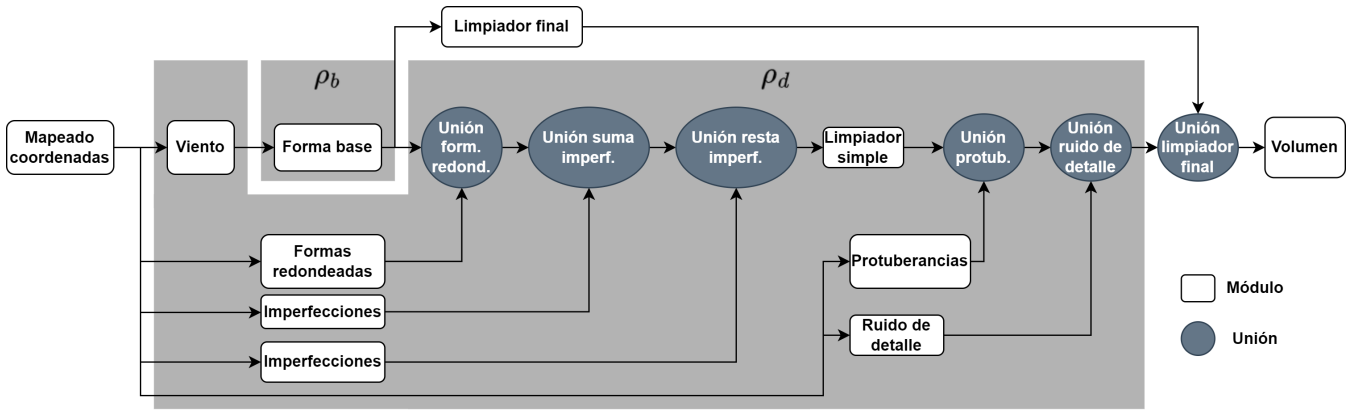


Figure 2: Grafo del modelo por módulos.

4.1. Cúmulo

Los cúmulos son nubes de desarrollo vertical que se forman, principalmente, por convección. Suelen tener formas definidas y ser relativamente compactas. Otra característica típica es que el borde inferior suele ser más plano que el borde superior, coincidiendo con la altitud a la que empieza a producirse la condensación. Para la creación de cúmulos tomaremos como forma básica un gradiente esférico tridimensional. Es decir, utilizaremos una función $g(|\mathbf{x}|)$, monótona decreciente y derivable, que asignará un valor entre 0 y 1 a cada punto $\mathbf{x} \in \Omega$ en función de su distancia al centro del dominio, alcanzando el máximo en el centro. Aunque la derivabilidad de g es una propiedad prescindible, es deseable para obtener formas suaves.

Dado un punto $\mathbf{x} = (x, y, z) \in \Omega$, para determinar su densidad obtendremos en primer lugar $\mathbf{x}' = (s_x \cdot x, s_y \cdot y, d(z))$, donde $d: \mathbb{I}_z \rightarrow \mathbb{I}_z$ es una función monótona creciente y $s_x, s_y > 0$ son dos factores de escala que controlan la forma de la nube. A continuación, calculamos la densidad de \mathbf{x} como $g(|\mathbf{x}'|)$. Para conseguir una nube con borde inferior plano a una altitud dada z_0 , utilizaremos una función d que tome valor z_0 en el extremo inferior del intervalo y que tenga derivada nula o casi nula en ese extremo. La Figura 3 muestra dos ejemplos de generación de la forma. En el ejemplo superior no se ha aplicado ningún escalado a las coordenadas x, y y derivada $d' = 1$ en la segunda mitad del intervalo hace que la parte superior de la nube conserve el degradado del gradiente esférico original. En el ejemplo inferior, al tener d una derivada mayor en la segunda mitad del intervalo la parte superior de la nube queda escalada, formándose una nube más alta.

4.2. Paisaje de cúmulos

Denominamos *paisaje de cúmulos* a un cielo más o menos cubierto con cúmulos. Para generar este tipo de nubosidad se genera, en primer lugar, una capa de nube uniforme en cierta franja de altitud y, a continuación, se elimina densidad en ciertas regiones del volumen. Para la definición de la región ocupada por las nubes se emplea una función meseta h . Esta función puede computarse, por ejemplo, restando dos funciones gradiente monótonas crecientes, como se muestra en la Figura 4, parte central. El valor preliminar

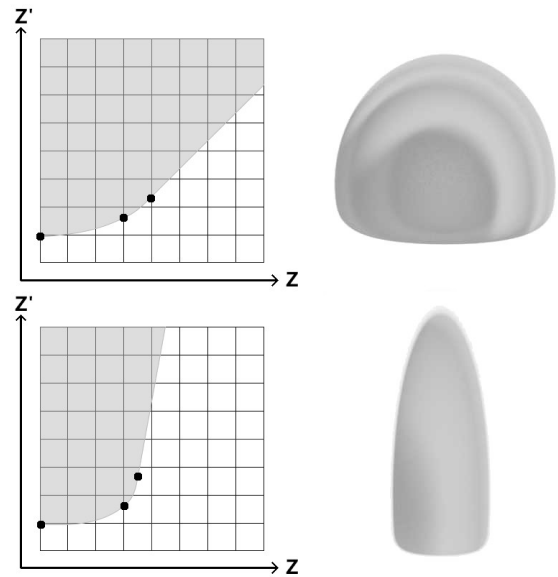


Figure 3: Densidad base para un cúmulo individual. Dado un punto \mathbf{x} , se escalan sus coordenadas x, y y se transforma su coordenada z a partir de $d(z)$. El valor de densidad se calcula con un gradiente esférico en el punto desplazado. La parte derecha muestra el efecto de dos funciones de desplazamiento vertical $d(z)$.

de la densidad para un punto se obtiene evaluando la función meseta en su coordenada vertical, $\rho_0(\mathbf{x}) = h(z)$. A continuación, se evalúa un ruido de Perlin p , normalizado al intervalo $[0, \mu]$, que se resta a la densidad preliminar. El coeficiente $\mu > 0$ determina el grado de cobertura que se obtendrá. La densidad final para este tipo de nubes se obtiene como

$$\rho(\mathbf{x}) = \text{máx}(0, \rho_0(\mathbf{x}) - \mu \cdot p(\mathbf{x})) = \text{máx}(0, h(z) - \mu \cdot p(\mathbf{x})).$$

El ruido de Perlin tiene como parámetros un factor de escala y un desplazamiento que permiten cambiar el aspecto del resultado. La escala del ruido de Perlin, junto con el factor μ permiten controlar la cantidad y tamaño de los claros que se formarán entre las nubes.

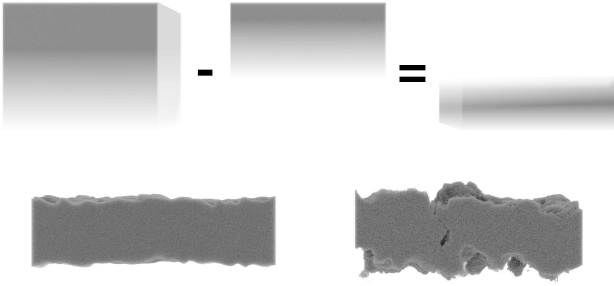


Figure 4: Ejemplo de la creación de la capa en el módulo de forma inicial de tipo paisaje de cúmulos. Las dos capas de la fila de abajo ilustran la diferencia a la hora de aplicar el resto de efectos con un gradiente definido o no definido.

4.3. Paisaje de cirros

Una de las características principales de las nubes de tipo cirro es que se organizan en bandas delgadas. Por este motivo, la forma básica que emplearemos para este tipo de nubosidad será una capa formada por barras o prismas rectangulares con densidad 1 en el dominio Ω . En este caso no emplearemos funciones de densidad derivables y quedará para los diferentes módulos de distorsión la obtención de formas suaves y difuminadas. Podemos emplear una función periódica f , como una función trigonométrica y un umbral c para determinar las regiones con nube. La función de densidad podría definirse como

$$\rho(\mathbf{x}) = \begin{cases} 1 & z \in [z_{min}, z_{max}] \text{ y } f(x) < c; \\ 0 & \text{en otro caso.} \end{cases}$$

En este caso, los cirros se extenderán en la dirección del eje y y con la anchura de las bandas controlada por el parámetro c y la altura de la capa de nubes por el intervalo $[z_{min}, z_{max}]$. La Figura 5 muestra un ejemplo de la creación de la forma base para un paisaje de cirros que, posteriormente, será distorsionada para crear el resultado final.

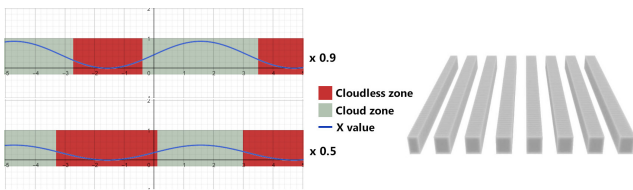


Figure 5: Ejemplo de la creación de bandas en el módulo de forma inicial de tipo paisaje de cirros.

5. Distorsión de la forma base

Como hemos indicado, tras la generación de la densidad que define la forma base de la nube, ρ_b , generaremos una densidad ρ_d para añadir detalles. La forma base se distorsionará siguiendo dos estrategias; por una parte se introducirá una distorsión para simular el efecto del viento, con desplazamientos direccionales, y, por otra

parte, se añadirán diferentes funciones de ruido a la densidad para generar imperfecciones en la forma base.

5.1. Simulación de viento

Este módulo se encarga de alterar la función de densidad de la forma base de tal manera que dé la apariencia de un desplazamiento producido por viento. Para ello se aplica un desplazamiento a las coordenadas de cada punto antes de introducirlo al módulo que calcula la densidad de la forma base. Dado un punto $\mathbf{x} = (x, y, z) \in \Omega$, se ha de obtener $\mathbf{x}' = \mathbf{x} + \mathbf{w}(\mathbf{x})$. El valor del desplazamiento \mathbf{w} se calcula combinando varias funciones de ruido de Perlin de la siguiente forma. Se consideran dos texturas de ruido de Perlin tridimensional, $\mathbf{p}_1, \mathbf{p}_2 : \mathbb{R}^3 \rightarrow [-\frac{1}{2}, \frac{1}{2}]^3$. Recordemos que un ruido de Perlin tiene como parámetros un factor de escala y un desplazamiento inicial, que serán los que definirán cada una de las texturas. A partir de estas funciones de ruido, el desplazamiento lo obtendremos por medio de la expresión

$$\mathbf{w}(\mathbf{x}) = \lambda \otimes (\mu_1 \cdot \mathbf{p}_1(\mathbf{x} + \mathbf{p}_1(\mathbf{x})) + \mu_2 \cdot \mathbf{p}_2(\mathbf{x} + \mathbf{p}_2(\mathbf{x}))),$$

donde μ_1, μ_2 son dos coeficientes que permiten controlar el efecto de cada una de las funciones de ruido y $\lambda \in \mathbb{R}^3$ determina la dirección del viento. El resultado de este desplazamiento se introduce en el proceso de cálculo de la forma base de la nube, de tal forma que, para una función de densidad $\rho(\mathbf{x})$ de las descritas en la Sección 4 la función que generará la nube se calcula como

$$\rho_b(\mathbf{x}) = \rho(\mathbf{x} + \mathbf{w}(\mathbf{x})).$$

La Figura 6 muestra un esquema del cálculo de esta distorsión y un ejemplo de su aplicación al calcular una función de densidad de nube.

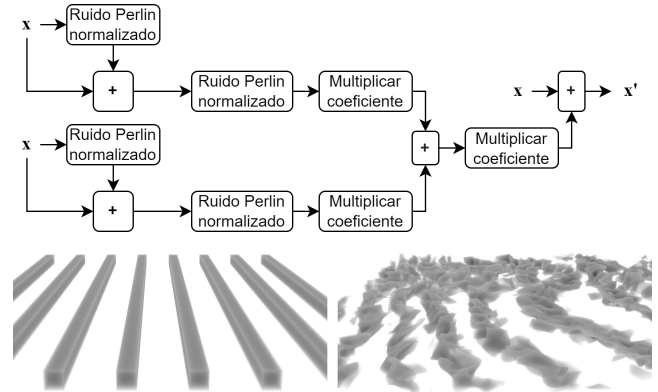


Figure 6: La aplicación de un desplazamiento a las coordenadas de los puntos del dominio permite distorsionar la forma base, haciendo el efecto de un viento que arrastra partes de la nube. La parte superior de la figura muestra un esquema de la generación de este efecto. En la parte inferior de la figura se muestra un ejemplo de aplicación de este efecto.

5.2. Imperfecciones de gran escala

A partir de la forma base generada, el siguiente paso es generar estructuras irregulares que den un aspecto realista a la nube. En pri-

mer lugar, generamos estructuras de una escala comparable a la de la propia nube y hasta tamaños de 1/10 respecto al de ésta. Estas estructuras de gran escala se obtienen a través de una función de densidad de detalle $\rho_d^g : \Omega \rightarrow [0, 1]$. Esta densidad se obtiene a través de la combinación de dos ruidos de Perlin con el mismo factor de escala pero diferente desplazamiento de referencia, $p_1, p_2 : \Omega \rightarrow [0, 1]$. La combinación de ambos ruidos se realiza por medio de la función

$$\rho_d^g(\mathbf{x}) = 1 - \frac{1 - p_1(\mathbf{x})}{p_2(\mathbf{x})} \quad (1)$$

y el resultado se suma a la densidad base multiplicado por un coeficiente, $\mu_g \in [-1, 1]$, que modula el efecto de estas imperfecciones. Puede ser conveniente generar imperfecciones de varias escalas y sumarlas o restarlas a la densidad base. De esta manera, dada una densidad de forma base, ρ_b , y una serie de densidades de imperfección $\rho_d^{g_1}, \dots, \rho_d^{g_n}$, la densidad que se obtiene tras aplicar las imperfecciones de gran escala será

$$\rho(\mathbf{x}) = \rho_b + \sum_k \mu_{g_k} \rho_d^{g_k}(\mathbf{x}).$$

La Figura 7 muestra un esquema de la generación de imperfecciones de gran escala y el resultado de aplicar dos términos de este tipo de distorsión a una forma base de tipo cúmulo. La ecuación (1) implementa lo que comúnmente se conoce como la mezcla de tipo *burn* en composición de imágenes [Jac15]. En este caso, la imagen es la forma base de la nube y las imperfecciones actúan como máscara.

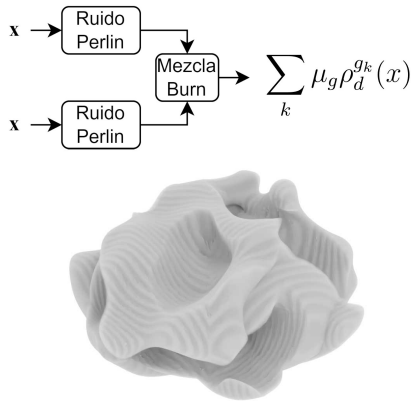


Figure 7: La generación de imperfecciones de gran escala permite dar a la nube una forma irregular a partir de la forma base.

5.3. Imperfecciones redondeadas

A continuación describimos un esquema de generación de densidad de detalle que permite generar nubes con aspecto sintético, del estilo típico de los dibujos animados. Para ello utilizaremos un ruido de Worley que genera texturas con celdas esféricas distribuidas de manera irregular. En nuestro caso, emplearemos una función de ruido de Worley [Wor96], $w : \Omega \rightarrow [0, 1]$ y calcularemos la densidad como

$$\rho_d^w(\mathbf{x}) = \rho_b(\mathbf{x})(\rho_b(\mathbf{x}) + 2w(\mathbf{x})(1 - \rho_b(\mathbf{x}))). \quad (2)$$

Esta expresión corresponde a la aplicación de una mezcla de tipo *overlay* que reduce el valor final en puntos que corresponden a densidades medias de ρ_d [Jac15]. Esto hace que se preserven las celdas generadas por w que están en la zona de la superficie de la nube. La densidad obtenida de esta forma se combina con la de la forma base, ρ_b , realizando una interpolación de ambos valores:

$$\rho(\mathbf{x}) = a^w \rho_b(\mathbf{x}) + (1 - a^w) \rho_d^w(\mathbf{x}); \quad a^w \in [0, 1].$$

La Figura 8 muestra un ejemplo de la formación de una nube con volúmenes redondeados generados a partir de ruido de Worley.

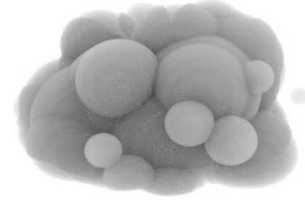


Figure 8: Por medio de ruido de Worley y la aplicación de una máscara de tipo *burn* es posible generar nubes muy regulares, con aspecto de dibujo animado clásico.

5.4. Imperfecciones de pequeña escala

La última fase de la generación de la nubosidad consiste en la adición de detalle de pequeña escala en la superficie de la nube. Para ello tendremos en cuenta que las nubes presentan a menudo una forma con estructura fractal, en la que determinados patrones se repiten a diferentes escalas.

La adición de detalles de pequeña escala parte de la densidad resultante de los pasos anteriores, $\rho(\mathbf{x})$, que contendrá la forma base combinada con los efectos de viento y de deformaciones de gran escala. Previamente al cálculo de las imperfecciones realizamos un filtrado de esta densidad por medio de un umbral v . Es decir, trabajaremos con la densidad ρ' definida como

$$\rho'(\mathbf{x}) = \begin{cases} \rho(\mathbf{x}) & \rho(\mathbf{x}) > v; \\ 0 & \text{en otro caso.} \end{cases}$$

Sobre esta función de densidad generaremos, en primer lugar, un conjunto de protuberancias sobre la superficie y, por último, añadiremos ruido de detalle.

5.4.1. Protuberancias

En primer lugar generamos una serie de protuberancias sobre la superficie de la nube empleando la misma estrategia que se ha usado en la Sección 5.3 para crear formas redondeadas. Crearemos una serie de texturas de ruido de Worley con escalas decrecientes que se combinan por medio de máscaras de tipo *overlay*, según la ecuación (2). La combinación de las texturas se realiza en cascada: en primer lugar se combinan las dos texturas de mayor escala, w_1 y w_2 ; el resultado se combina con la textura siguiente en escala, w_3 ; el resultado con la siguiente, y se prosigue hasta la última textura de ruido de Worley.

Este proceso da lugar a una función de densidad de protuberancias, ρ_d^p . Esta función se combina con la densidad de entrada por medio de la expresión

$$\rho(\mathbf{x}) = a^p \rho' + (1 - a^p) \rho'(\mathbf{x})(1 - \rho_d^p(\mathbf{x})); \quad a^p \in [0, 1].$$

Todo este proceso puede combinarse con un desplazamiento de las coordenadas antes de empezar los cálculos, empleando el efecto de viento descrito en la Sección 5.1. De esta forma se obtendrá un efecto más irregular. La Figura 9 muestra un esquema de la generación de protuberancias en la superficie de un cúmulo.

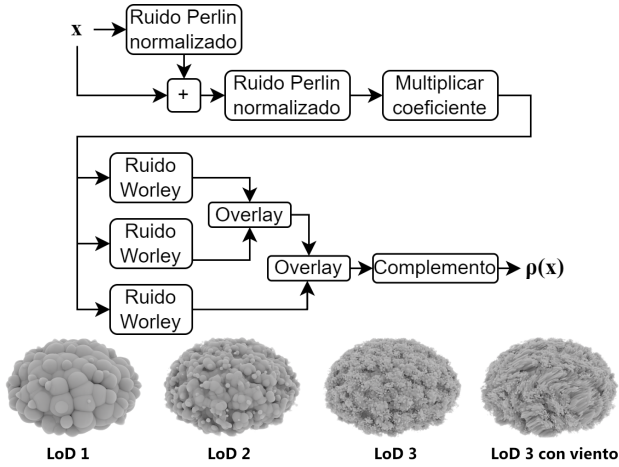


Figure 9: Añadiendo protuberancias de diferentes escalas, junto con efecto de viento, se consigue una superficie irregular.

5.4.2. Ruido de detalle

El último paso consiste en crear un ruido de detalle fino con múltiples escalas. Para nubes muy cercanas a la cámara este efecto aporta realismo pues evita la observación de zonas muy planas ausentes de ruido. Se genera un único ruido de Perlin de escala muy pequeña y con varias octavas para simular el efecto fractal de las nubes. Para combinar este ruido con la densidad de la nube ρ' se aplica de nuevo una máscara de tipo *overlay*, que genera la densidad de ruido de detalle, ρ_d^d . Por último, esta densidad de ruido se combina linealmente con la densidad obtenida tras la adición de protuberancias. La Figura 10 muestra el efecto de introducir ruido de detalle sobre una nube base sin ninguna transformación.

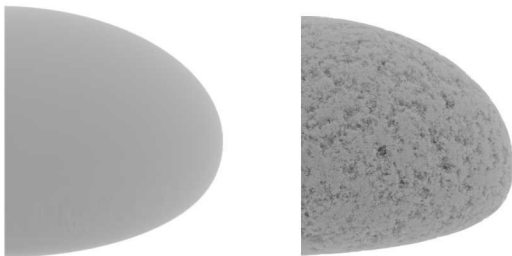


Figure 10: Ejemplo antes y después de aplicar ruido de detalle.

6. Limpieza de artefactos y binarización

Durante el proceso de generación de detalle es posible que se creen zonas con densidad no nula en regiones alejadas de la forma inicial de manera indeseada. El último paso del proceso de generación de la función de densidad limpia de artefactos nubosos las zonas exteriores de la nube. Para ello se parte de la premisa de que no debe haber densidad de nube en regiones alejadas de la forma base inicial. Por tanto, se utilizará la densidad de la forma base ρ_b como máscara y se reducirá o anulará la densidad en las zonas en las que no hay nube de acuerdo con ρ_b .

Con el fin de controlar la intensidad de este proceso, se empleará una transformación lineal del intervalo $[0, 1]$ en un intervalo $[a, b] \subseteq [0, 1]$, definida por

$$f(x, a, b) = \begin{cases} 0 & x \leq a; \\ (x - a)/(b - a) & x > a \text{ y } x < b; \\ 1 & x \geq b. \end{cases}$$

Tras aplicar esta función a la densidad complementaria de ρ_b se obtiene la densidad final de la nube, ρ_f , como

$$\rho_f = \rho_d' - f(1 - \rho_b, a, b)$$

donde ρ_d' es la densidad tras calcular la aplicación de los efectos de las diferentes escalas y ρ_b es la densidad de la forma base. Esta operación limpia los valores de densidad altos en las regiones que están lejos de la forma original. La Figura 11 muestra un ejemplo de aplicación de este proceso de limpieza.

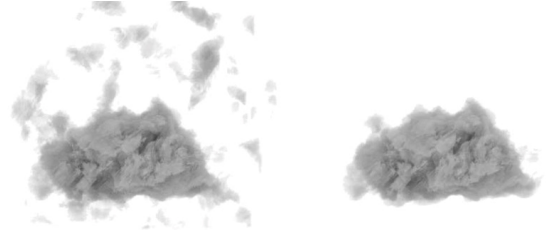


Figure 11: Ejemplo de aplicación del limpiador final.

El resultado final obtenido contiene valores en el intervalo $[0, 1]$, lo que da lugar a nubes muy poco definidas, con bordes muy difuminados. El último paso antes de renderizar la nube consiste en convertir la función de densidad a una función discreta, ρ_v , que únicamente toma valores 0 y δ , con $\delta \in [0, 1]$ un valor constante. Para ello se aplica un umbral, v , a los valores de la densidad

$$\rho_v(\mathbf{x}) = \begin{cases} \delta & \rho_f(\mathbf{x}) > v; \\ 0 & \text{en otro caso.} \end{cases}$$

7. Implementación

Como se ha indicado, el proceso de cálculo de la densidad para cada punto es independiente del de cualquier otro punto del dominio. Esto permite implementar el modelo de forma altamente paralelizable empleando lenguajes de programación de sombreado o *shaders*. En nuestro caso, el modelo se ha implementado utilizando el flujo de trabajo basado en nodos del programa de animación Blender [Ble20] y se ha desarrollado un complemento o *add-on* que

expone los parámetros del sistema a través de la interfaz gráfica de usuario del programa. El sistema se ha desarrollado empleando únicamente nodos nativos de esta aplicación, sin desarrollar nodos nuevos. Muchos de los parámetros del modelo, tales como umbrales o escalas, se han expuesto al usuario a la hora de diseñar la interfaz gráfica, de tal manera que la creación de las nubes pueda llevarse a cabo de forma interactiva. Estos parámetros, sin embargo, se filtran en algunos casos de forma que se respeten ciertas relaciones de escala, de tal manera que, por ejemplo, las imperfecciones de gran escala, definidas en la Sección 5.2 tengan, al menos, un orden de magnitud mayor que las protuberancias de pequeña escala propuestas en la Sección 5.4.

La implementación en Blender permite añadir nubes desde el menú del *Viewport 3D* como un objeto más de la escena. El objeto nube tiene sus propias configuraciones en el panel de propiedades. Dependiendo de si se trata un cúmulo individual, paisaje de cúmulos o de cirros cambian algunas de las configuraciones disponibles. La Figura 12 muestra una vista del complemento desarrollado.

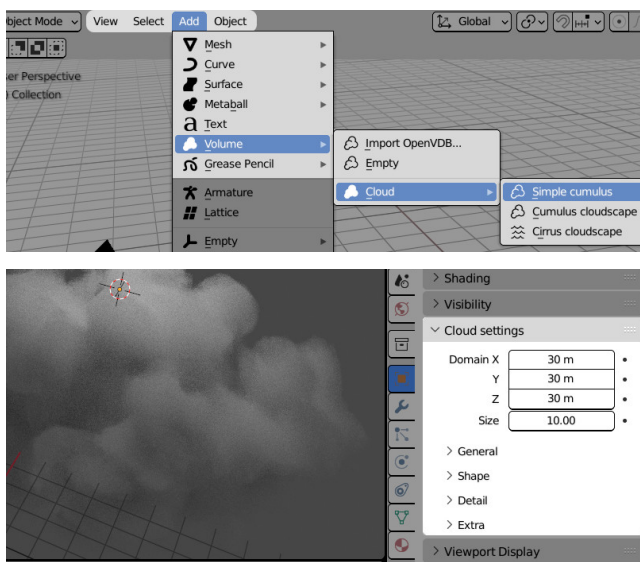


Figure 12: Interfaz del add-on implementado.

Además, se ha implementado, por un lado, la posibilidad de elegir entre un modo simple o avanzado de configuración simplificando la cantidad de parámetros. Por otro lado, se han predefinido dos configuraciones de renderizado: una que disminuye la calidad visual pero mejora la respuesta, facilitando la edición de las nubes, y otra con parámetros adaptados para un renderizado de calidad. Se trata únicamente de preconfiguraciones que pueden ser personalizadas según las necesidades de la producción. El add-on está disponible para descarga en [RHGF22].

8. Resultados

El sistema desarrollado permite generar, a partir de una representación volumétrica, una gran variedad de tipos de nubes. Las Figuras 1 y 13 presentan varios ejemplos de cobertura que muestran

las opciones que ofrece el modelo en cuanto al control del resultado final. El sistema se basa en la construcción de una forma básica que posteriormente se modifica. Según se muestra en la Figura 14, la forma básica de cúmulo permite controlar las proporciones de la nube, así como la forma de la parte inferior, que puede ser más o menos plana. Para la generación de paisajes con una cobertura de cúmulos es posible controlar la cantidad y tamaño de los claros, empleando el parámetro μ según se describe en la Sección 4.2. La Figura 13 muestra varias configuraciones que incluyen nubes con aspecto de altocúmulos (arriba a la izquierda e imagen inferior).

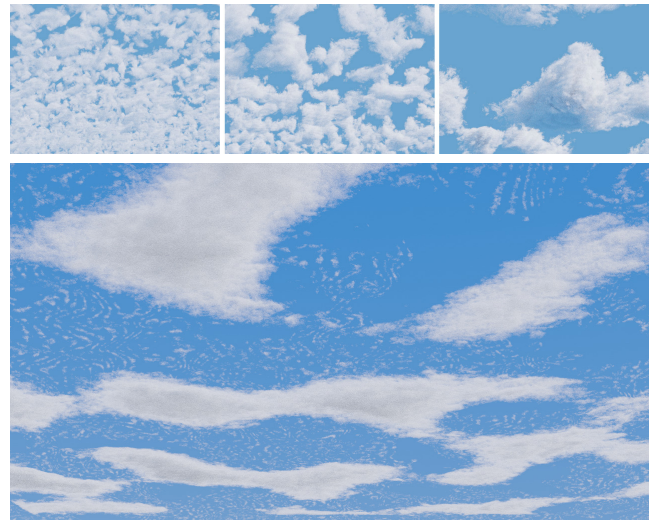


Figure 13: Controlando las dimensiones de la capa de nubes, la escala de los claros y el nivel de cobertura es posible generar tipos muy variados de nubosidad.

Las nubes de tipo cirro permiten generar nubes altas. En este caso, uno de los parámetros principales es el efecto de viento descrito en la Sección 5.1 que difuminará las bandas de nubes hasta darles aspecto realista, como se ve en la Figura 15 y en la imagen inferior derecha de la Figura 1. Este tipo de nubes, combinado con capas de nubes bajas, permite obtener resultados especialmente realistas, como se muestra en la imagen de la derecha de la Figura 15.

Para dar realismo a las nubes obtenidas es importante conseguir formas irregulares y dotarlas del detalle adecuado en su superficie. La distorsión de gran escala descrita en la Sección 5 permitirá generar nubes aleatorias con formas irregulares. La Figura 16 muestra la aplicación de esta deformación sobre una forma base regular. Las diferentes protuberancias se obtienen sumando y restando texturas de ruido de Perlin de diferente escala a la densidad base ρ_b .

Además de las modificaciones de la forma general de la nube, se introducen efectos de pequeña escala que permiten proporcionar a las nubes un aspecto más realista y la característica forma fractal que se asocia a su superficie. La Figura 17 muestra cómo a medida que se van añadiendo efectos de escalas cada vez más pequeñas se obtiene un resultado de mayor realismo. Los efectos mostrados en esta figura corresponden a la adición de protuberancias sobre una cúmulo con deformación de gran escala. La técnica corresponde al



Figure 14: En las nubes de tipo cúmulo es posible controlar las proporciones del resultado. También es posible conseguir nubes que tengan la parte inferior más o menos plana. Las tres nubes mostradas se han generado usando imperfecciones redondeadas con $a^w = 0,6$, tres niveles de protuberancias con $a^p = 0,2$, dos de imperfecciones con $\mu_{g1} = 0,5$, $\mu_{g2} = -0,25$ y una densidad final de $\delta = 1$. La única diferencia es el valor de las escalas s_x, s_y y la función de desplazamiento $d(z)$.



Figure 15: Los paisajes de cirros permiten simular nubes altas. La imagen de la derecha muestra un paisaje formado por una capa de cirros sobre una capa de cúmulos. En ambas imágenes, los cirros usan un viento de intensidad $\lambda = 4,2$, perturbaciones con $\mu_{g1} = 0,7$ y $\mu_{g2} = -0,4$ y un nivel de protuberancias con $a^p = 0,1$. A la derecha, la capa de cúmulos usa una cobertura de $\mu = 0,5$, viento con $\lambda = 1$, e imperfecciones con $a^w = 0,6$, $\mu_{g1} = 0,6$, $\mu_{g2} = -0,2$ y tres niveles de protuberancias con $a^p = 0,3$.

método descrito en la Sección 5.4.1 en el que se propone la adición de protuberancias de varias escalas por medio de ruido de Worley (véase también la Figura 9).

El ruido de detalle y el efecto de viento proporcionan el último nivel de adición de estructura a la superficie de la nube. El ruido de detalle, consistente en un ruido de Perlin de varias octavas, da lugar a la escala más pequeña de deformaciones aplicadas a la función de densidad. El efecto de viento descrito en la Sección 5.1 permite mo-

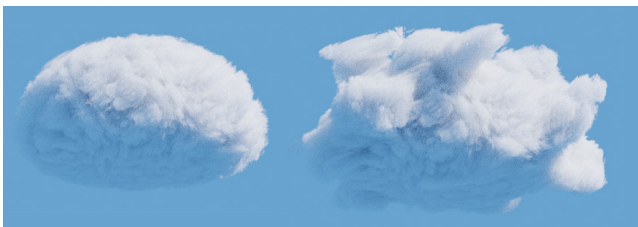


Figure 16: Aplicación de deformación de gran escala sobre una nube base de tipo cúmulo. Ambas nubes emplean $a^w = 0$, $a^p = 0,3$ con tres niveles de imperfección, ruido de imperfecciones con $\lambda = 0,6$ y densidad final $\delta = 2$. Las imperfecciones de la nube de la derecha usan $\mu_{g1} = 0,7$ y $\mu_{g2} = -0,4$.



Figure 17: Efecto de la aplicación de pequeñas protuberancias a la superficie de una nube de tipo cúmulo. De izquierda a derecha y de arriba a abajo, se muestra la nube sin protuberancias y con uno, dos y tres niveles de ruido usando $a^p = 0,3$. El resto de parámetros están fijos en $a^w = 0,6$, $\mu_{g1} = 0,3$, $\mu_{g2} = -0,1$ y $\delta = 1$.

dificar cualquiera de las transformaciones anteriores introduciendo cierta direccionalidad, lo que hace el efecto de partículas que son arrastradas por corrientes de aire. La Figura 18, derecha, muestra una nube a la que se han aplicado los diferentes tipos de detalle antes (izquierda) y después (derecha) de aplicarle ruido de viento.

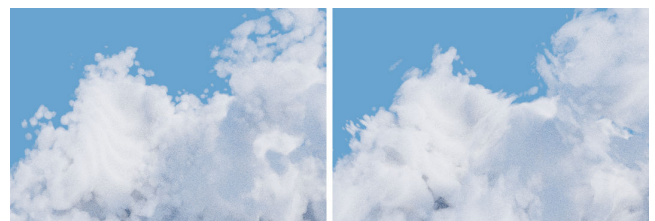


Figure 18: Aplicación de ruido de viento (derecha) con $\lambda = 0,6$ sobre una nube generada con $a^p = 0,3$ y tres niveles de ruido de protuberancias, $a^w = 0,6$, $\mu_{g1} = 0,3$, $\mu_{g2} = -0,1$ y $\delta = 2$ (izquierda).

Dado que la representación que se emplea para las nubes es una función volumétrica de densidad, permite su uso en flujos de trabajo en los que sea factible emplear este tipo de información en el proceso de render. Además, esto permite combinar el resultado con otras técnicas, incluida la aplicación de máscaras (Figura 19, parte superior). De esta forma, por ejemplo, es sencillo hacer que la forma base de la nube se genere a partir de un modelo 3D permitiendo generar nubes de formas arbitrarias o elaborar efectos como letras. La Figura 19, parte inferior, muestra un ejemplo de nubes generadas empleando esta técnica.

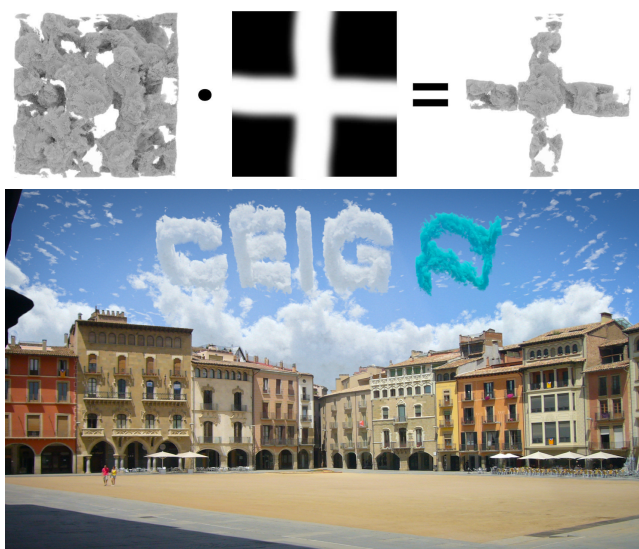


Figure 19: La representación de la nube como una función de densidad volumétrica permite el uso máscaras para la creación de nubes de formas arbitrarias. La imagen inferior muestra un ejemplo de la aplicación de esta técnica junto con nubes de aspecto realista. Foto de Jordi Ferrer, reproducida bajo licencia CC BY-SA 3.0.

Nuestro método presenta una variante de la propuesta de [Sch15], utilizando igualmente gradientes para delimitar las zonas de nubes y varias capas de los ruidos mencionados para dar forma. Pero además, proveemos técnicas para la generación de varias formas de nube base sobre la que aplicar los efectos, un desarrollo más detallado de la generación de efectos nubosos y su aplicación y una estructura modular detallada que facilita la implementación del modelo.

8.1. Análisis de prestaciones

No resulta sencillo realizar un análisis sistemático del coste computacional asociado al modelo propuesto. El módulo desarrollado para Blender permite generar renders por trazado de rayos en baja resolución a una velocidad suficiente como para manipular la escena de forma interactiva, usando la previsualización integrada en el programa [Ble20]. También se han llevado a cabo pruebas de render, aunque son difíciles de valorar ya que dependen mucho de los parámetros de configuración del motor Cycles. Las pruebas

se han realizado en un equipo portátil con las siguientes especificaciones: AMD Ryzen 7 4800HS, NVIDIA GeForce RTX2060-6GB Max-Q, 16 GB RAM corriendo Windows 10. Se ha utilizado Blender 2.83 LTS y el motor de render Cycles, configurado con 82 muestras y un máximo de 12 rebotes. Los tiempos de render para las figuras mostradas, a una resolución de 640×360 píxeles, van desde los 3 minutos para la capa de cirros de la Figura 15, izquierda, hasta los casi 20 minutos en la Figura 18 o alrededor de 17 minutos en la Figura 17 con máximo detalle. En esta última figura destaca el progresivo aumento del coste del render a medida que la complejidad de la superficie de la nube aumenta, ya que la nube sin detalle requiere 7 minutos.

También se han identificado los solapamientos de los dominios de diferentes nubes como un factor que añade tiempo de render. Por ello es conveniente ajustar los dominios a la región que queremos que ocupe cada una de las nubes.

9. Conclusiones y trabajo futuro

Este trabajo presenta un sistema para la creación de nubes atmosféricas que permite una edición rápida y con mucho control sobre el resultado a partir de un juego reducido de parámetros. El sistema se basa en una jerarquía de funciones de densidad que lo hace muy flexible y extensible. Además, esto permite su integración con la mayoría de motores de render. El sistema se ha implementado como un complemento para el software de animación Blender 2.83 y se han presentado diversas pruebas junto con un análisis del coste de render en función de los parámetros empleados.

Como trabajo futuro se plantea mejorar el efecto de viento para obtener resultados más realistas, aplicando diferentes magnitudes en función de la altitud y el uso de ruido de tipo *curl* [BHN07]. También se plantea la realización de un análisis más exhaustivo tanto del comportamiento del modelo como del coste del render en diferentes escenarios.

References

- [AFGC07] ABDUKADIR M., FUJIMOTO T., GENI M., CHIBA N.: An efficient hybrid method for animating the growth of large-scale cumulus-type cloud. *The Journal of the Society for Art and Science* 6 (2007), 179–196. doi:10.3756/artsci.6.179. 2
- [ALM*14] ALLDIECK T., LUNDTFT D. H., MONTANARI N., NIKOLOV I., VLAYKOV I. G., MADSEN C. B.: Modelling of Clouds from a Hemispherical Image. In *Computer Graphics and Visual Computing (CGVC)* (2014). doi:10.2312/cgvc.20141202. 2
- [BDY15] BARBOSA C., DOBASHI Y., YAMAMOTO T.: Adaptive cloud simulation using position based fluids. *Computer Animation and Virtual Worlds* 26 (2015). doi:10.1002/cav.1657. 2
- [BHN07] BRIDSON R., HOURIHAM J., NORDENSTAM M.: Curl-noise for procedural fluid flow. *ACM Trans. Graph.* 26, 3 (2007), 46–es. doi:10.1145/1276377.1276435. 9
- [Ble20] BLENDER ONLINE COMMUNITY: *Blender 2.83 Reference Manual*. Blender Foundation, Blender Institute, Amsterdam, 2020. URL: <https://docs.blender.org/manual/en/2.83/>. 6, 9
- [BN04] BOUTHORS A., NEYRET F.: Modeling clouds shape. In *Proc. of EUROGRAPHICS, Short papers* (2004). 2
- [DG17] DUARTE R. P., GOMES A. J.: Real-time simulation of cumulus clouds through skewt/logp diagrams. *Computers and Graphics* 67 (2017), 103–114. doi:https://doi.org/10.1016/j.cag.2017.06.005. 2

- [DNYO99] DOBASHI Y., NISHITA T., YAMASHITA H., OKITA T.: Using metaballs to modeling and animate clouds from satellite images. *The Visual Computer* 15 (1999), 471–482. doi:10.1007/s003710050193. 2
- [DSY10] DOBASHI Y., SHINZO Y., YAMAMOTO T.: Modeling of clouds from a single photograph. *Comput. Graph. Forum* 29 (2010), 2083–2090. doi:10.1111/j.1467-8659.2010.01795.x. 2
- [DYN06] DOBASHI Y., YAMAMOTO T., NISHITA T.: A controllable method for animation of earth-scale clouds. In *Proc. of CASA* (2006), pp. 43–52. 2
- [DYN10] DOBASHI Y., YAMAMOTO T., NISHITA T.: An interactive rendering system using hierarchical data structure for earth-scale clouds. *SCIENCE CHINA Information Sciences* 53 (2010), 920–931. doi:10.1007/s11432-010-0070-4. 2
- [EES*16] ELHADDAD A., ELHADDAD F., SHENG B., ZHANG S., SUN H., WU E.: Real-time cloud simulation using lennard-jones approximation. In *Proceedings of the International Conference on Computer Animation and Social Agents* (2016), CASA '16, p. 131–137. doi:10.1145/2915926.2915942. 2
- [EP90] EBERT D. S., PARENT R. E.: Rendering and animation of gaseous phenomena by combining fast volume and scanline a-buffer techniques. *SIGGRAPH Comput. Graph.* 24, 4 (1990), 357–366. doi:10.1145/97880.97918. 2
- [ES00] ELINAS P., STUERZLINGER W.: Real-time rendering of 3d clouds. *Journal of Graphics Tools* 5, 4 (2000), 33–45. doi:10.1080/10867651.2000.10487531. 2
- [Gar85] GARDNER G. Y.: Visual simulation of clouds. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1985), SIGGRAPH '85, p. 297–304. doi:10.1145/325334.325248. 2
- [GN17] GOSWAMI P., NEYRET F.: Real-time Landscape-size Convective Clouds Simulation and Rendering. In *Workshop on Virtual Reality Interaction and Physical Simulation* (2017). doi:10.2312/vrphys.20171078. 2
- [Gos21] GOSWAMI P.: A survey of modeling, rendering and animation of clouds in computer graphics. *The Visual Computer* 37, 7 (Jul 2021), 1931–1948. doi:10.1007/s00371-020-01953-y. 1
- [HBSL03] HARRIS M. J., BAXTER W. V., SCHEUERMANN T., LASTRA A.: Simulation of cloud dynamics on graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware* (2003), HWWS '03, p. 92–101. 2
- [HHS07] HUFNAGEL R., HELD M., SCHRÖDER F.: Large-scale, realistic cloud visualization based on weather forecast data. In *Proceedings of the Ninth IASTED International Conference on Computer Graphics and Imaging* (2007), CGIM '07, p. 54–59. 2
- [HKA06] HASAN M., KARIM M., AHMED E.: Generating and rendering procedural clouds in real time on programmable 3d graphics hardware. In *2005 Pakistan Section Multitopic Conference* (2006), pp. 1–6. doi:10.1109/INMIC.2005.334442. 2
- [IND11] IWASAKI K., NISHINO T., DOBASHI Y.: Real-time Rendering of Endless Cloud Animation. In *Pacific Graphics Short Papers* (2011). doi:10.2312/PE/PG/PG2011short/073-076. 2
- [Jac15] JACKSON W.: *Digital Image Compositing Fundamentals*. Apress, Berkeley, CA, 2015. 5
- [KK09] KOWSUWAN N., KANONGCHAIYOS P.: 3d cloud animation using ca based method. *ISPACS 2009 - 2009 International Symposium on Intelligent Signal Processing and Communication Systems, Proceedings* (2009). doi:10.1109/ISPACS.2009.5383819. 2
- [KVH84] KAJIYA J. T., VON HERZEN B. P.: Ray tracing volume densities. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques* (1984), SIGGRAPH '84, p. 165–174. URL: <https://doi.org/10.1145/800031.808594>, doi:10.1145/800031.808594. 2
- [LHW*07] LIU S., HUANG R., WANG Z., PENG Q., ZHANG J., SUN J.: Texture advection based simulation of dynamic cloud scene. In *IEEE International Conference on Computer-Aided Design and Computer Graphics* (2007), pp. 144–149. doi:10.1109/CADCG.2007.4407871. 2
- [MDN02] MIYAZAKI R., DOBASHI Y., NISHITA T.: Simulation of cumuliform clouds based on computational fluid dynamics. In *Proc. of EUROGRAPHICS, Short papers* (2002). 2
- [MMPZ12] MILLER B., MUSETH K., PENNEY D., ZAFAR N. B.: Cloud modeling and rendering for "puss in boots". *ACM SIGGRAPH Talks* (2012). 2
- [Per02] PERLIN K.: Improving noise. *ACM Trans. Graph.* 21, 3 (2002), 681–682. doi:10.1145/566654.566636. 2
- [RHGF22] RODRÍGUEZ-HERNÁNDEZ D., GARCÍA-FERNÁNDEZ I.: Atmospheric clouds add-on. <https://rhdariorh.github.io/proyectos/proyectoAddonNubesBlender.html>, 2022. [Online; accessed 29-April-2022]. 7
- [Sak93] SAKAS G.: Modeling and animating turbulent gaseous phenomena using spectral synthesis. *Vis. Comput.* 9, 4 (1993), 200–212. doi:10.1007/BF01901724. 2
- [SBRS10] STIVER M., BAKER A., RUNIONS A., SAMAVATI F.: Sketch based volumetric clouds. In *International Symposium on Smart Graphics* (2010), pp. 1–12. doi:10.1007/978-3-642-13544-6_1. 2
- [Sch15] SCHNEIDER A.: The Real-time Volumetric Cloudscapes of Horizon: Zero Dawn. In *Advances in Real-Time Rendering in Games. SIGGRAPH Courses* (2015). 2, 9
- [Set97] SETHI M.: *Generating clouds in computer graphics*. Master's thesis, McGill University, 1997. 2
- [SSEH03] SCHPOK J., SIMONS J., EBERT D. S., HANSEN C.: A real-time cloud modeling, rendering, and animation system. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), SCA '03, p. 160–166. 2
- [TB02a] TREMBILSKI A., BROSSLER A.: Transparency for polygon based cloud rendering. In *Proceedings of the 2002 ACM Symposium on Applied Computing* (2002), SAC '02, p. 785–790. doi:10.1145/508791.508943. 2
- [TB02b] TREMBILSKI A., BROSSLER A.: Surface-based efficient cloud visualisation for animation applications. In *Proc. of WSCG* (2002), pp. 453–460. 2
- [WBC08] WITHER J., BOUTHORS A., CANI M.-P.: Rapid Sketch Modeling of Clouds. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2008). doi:10.2312/SBM/SBM08/113-118. 2
- [WCGG18] WEBANCK A., CORTIAL Y., GUÉRIN E., GALIN E.: Procedural cloudscapes. *Comput Graph Forum* 37 (2018). doi:10.1111/cgf.13373. 2
- [Wor96] WORLEY S.: A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996), SIGGRAPH '96, p. 291–294. doi:10.1145/237170.237267. 2, 5
- [WYMS12] WENKE W., YUMENG G., MIN X., SIKUN L.: Automatic generation of large scale 3d cloud based on weather forecast data. In *2012 International Conference on Virtual Reality and Visualization* (2012), pp. 69–73. doi:10.1109/ICVRV.2012.19. 2
- [YLH*14] YUAN C., LIANG X., HAO S., QI Y., ZHAO Q.: Modeling Cumulus Cloud Shape from a Single Image. *Comput Graph Forum* (2014). doi:10.1111/cgf.12350. 2
- [YLHY13] YUAN C., LIANG X., HAO S., YANG G.: Modeling Large Scale Clouds from Satellite Images. In *Pacific Graphics Short Papers* (2013). doi:10.2312/PE.PG.PG2013short.047-052. 2
- [YW11] YU C.-M., WANG C.-M.: An effective framework for cloud modeling, rendering, and morphing. *J. Inf. Sci. Eng.* 27 (2011), 891–913. 2