

Procedural Modeling of Terrain from GPS Routes

C. Gasch, M. Chover, I. Remolar and C. Rebollo

Univesitat Jaume I, Castellón, Spain

Abstract

Procedural modeling techniques provide an easy way for realistic terrain synthesis. There are many methods for this purpose, but few of them provide the necessary constraints to control the final result. The current controlling methods are slow and inaccurate.

The goal of this paper is to present a new procedural method to synthesis realistic terrain that fulfills the constraint of a set of consecutive coordinates from GPS routes.

In the new method, random terrain generation is based on Perlin noise algorithm. Instead of getting a random value for all points of each octave, the method presents the novelty of solving the needed values of each point to fulfill the coordinates of the GPS route when Perlin algorithm is applied. Points that are not included in the constraint, keep the randomness provided by the original Perlin method.

The results show that the method allows, at low computational cost, to integrate different complexity paths with high accuracy while conserving the natural aspect of the procedural generation.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics, Three-Dimensional Graphics and Realism]: Fractals.

1. Introducción

El modelado procedural de terrenos ha sido un tema de investigación durante más de treinta años según el estudio de Smelik et al. [STBB14]. La simulación de los entornos naturales, de los cuales el terreno es un componente vital, tiene gran importancia en una amplia gama de aplicaciones tales como simulaciones, juegos y sistemas de información geográfica [RRCR04]. Hay una gran cantidad de trabajos anteriores sobre modelado de terreno, además de herramientas que permiten modelar, generar y modificar terrenos de forma manual, como el Vterrain (<http://vterrain.org>) y el Terragen (<http://planetside.co.uk>), pero el diseño controlable y los métodos eficientes siguen siendo un problema importante, ya que el terreno suele requerir ser grande y detallado, con lo que el modelado manual no siempre es la mejor opción [STBB14].

Smelik et al. clasifican y describen estos trabajos de modelado de terreno. Al principio se basaron en métodos de subdivisión, donde el mapa de altura se va creando de forma iterativa y en cada iteración se va añadiendo una cantidad controlada de aleatoriedad para crear la elevación. El primer algoritmo de subdivisión se conoce como el método de desplazamiento del punto medio [FFC82][Mil86], donde cada vez que se genera un nuevo punto, su elevación se obtiene con el promedio de sus vecinos añadiendo además un desplazamiento aleatorio, que con cada iteración va disminuyendo. Hay muchos otros métodos, pero la mayoría se basan en generadores de ruido, como Perlin [Per85], donde el mapa de alturas se obtiene mediante una función de turbulencia. La ventaja de los métodos basados en generadores de ruidos es que el coste computacional es bajo, ya

que cada punto puede calcularse independientemente de sus vecinos.

Los métodos descritos tienen el problema de que no se puede tener ningún control sobre el resultado obtenido, excepto por las pequeñas variaciones que se pueden aplicar en los parámetros iniciales, como podría ser la altura máxima en el caso del método de Perlin. Pero aun así no son suficientes para permitir la generación de un terreno al que haya que aplicarle elementos que necesiten de alguna restricción. Algunos de estos elementos que requieren tener un control con precisión al generar un terreno serían los caminos y los ríos, ya que al incluirlos sobre el terreno da resultados poco realistas debido a que estos tienen formas y límites precisos y por tanto deben condicionar la forma del terreno. Por ejemplo, los caminos no deben tener pendientes demasiado inclinadas, los ríos no pueden fluir hacia arriba, etc.

Según Smelik et al. la solución que aportan los trabajos que intentan resolver este problema, se basan en su mayoría en el método de prueba y error para obtener el resultado deseado o al menos una aproximación. Algunos incluso necesitan que se indique donde se han de hacer las modificaciones y en qué medida, algo que puede no ser trivial para los usuarios, ya que es casi imposible predecir de forma fiable cómo afectará cada operación.

Por esta razón, en este trabajo se propone un método de creación de un terreno con una serie de restricciones de adaptación a una ruta determinada, sin necesidad de la interacción con el usuario y sin necesidad de modificar la ruta. El método creará el terreno mediante generación procedural,

pasando por los puntos de una ruta GPS. Para ello, extraerá los valores de altura que deben tener en los puntos de la ruta y, mediante un sistema de ecuaciones, calculará el valor del ruido en los puntos que influyen en la ruta. Posteriormente, se aplicará el método de generación de ruido de Perlin al resultado obtenido y al añadir un valor aleatorio al resto de puntos del terreno, se obtendrá un mapa de alturas que seguirá incluyendo la aleatoriedad de la naturaleza, pero con el camino perfectamente integrado en él.

Este artículo está organizado en cinco partes. La sección dos considera el estado del trabajo previo y una clasificación de estos trabajos según su forma de generar elementos restrictivos en un terreno. La sección tres proporciona los detalles de la implementación del método propuesto. La sección cuatro ofrece los resultados obtenidos con el método propuesto. Y por último, la sección cinco contiene algunas observaciones sobre la solución presentada y los resultados obtenidos, además del trabajo futuro.

2. Estado del arte

Durante los últimos años se han desarrollado diferentes métodos que podemos agrupar en la generación de terrenos con restricciones y en métodos que trabajan en su adaptación.

En el primer caso, la restricción correspondiente se genera a la vez que el terreno, es decir, el terreno generado ya incluye el elemento deseado. En cambio, en el segundo caso, los métodos necesitan un terreno ya creado, al cual se le aplican las modificaciones para que el terreno incluya el elemento que se necesita.

2.1. Generación del terreno con restricciones

Mediante la modificación del método de generación de terrenos utilizando el desplazamiento del punto medio, Kamal et al. [KU07] presenta un método que genera una sola montaña con restricción de la elevación y la propagación base y por otro lado, Belhadj utiliza un método más general [Bel07], donde un conjunto de valores limitan el desplazamiento del punto medio.

Un método más a gran escala, es el propuesto por Doran y Parberry [DP10], donde mediante el uso de agentes, añaden restricciones durante la generación del terreno, pudiendo crear así, accidentes geográficos como una montaña, una playa, etc. Este método, aunque permite controlar el relieve del terreno no permite saber la posición exacta de dónde se producirán estos accidentes geográficos [STBB14].

2.2. Adaptación del terreno

Los métodos que adaptan el terreno para que estén presentes elementos, tales como caminos o ríos, se pueden dividir entre los que no necesitan interacción con el usuario y los que sí.

Sin interacción del usuario: Una buena forma de modificar los terrenos de forma realista es el uso de algoritmos basados en la física para simular la erosión del terreno, ya sea producido por el aire o por el agua [MKM89]. Este método disminuye los cambios bruscos en las zonas elevadas mediante iteraciones hasta conseguir un equilibrio. En el caso del agua, la cantidad de agua y material disuelto se calcula a partir de la pendiente de la

elevación. El resultado final depende de la configuración inicial y aunque el resultado es bastante realista tiene un alto coste computacional, algo que se ha resuelto gracias al uso de la GPU, permitiendo que esta erosión se realice en tiempo real en terrenos a gran escala [KBK09]. Estas modificaciones del terreno permiten dar un aspecto más realista e incluso sirven para la creación de ríos aunque no de caminos. Su inconveniente es que los cambios que ocurran no pueden saberse de forma precisa antes de ver los resultados.

Un método que sí puede servir para la generación de caminos es el propuesto por Stachniak y Stürzlinger [SS05], el cual integra las restricciones en el terreno utilizando una imagen de máscara. Para ello utilizan un algoritmo que busca un conjunto aceptable de operaciones de deformación a aplicar al terreno. El problema es que este método, como se indica en el propio trabajo, es computacionalmente caro.

Por otra parte, el método de Zhou et al. [ZSTR07] puede servir para generar tanto caminos, ríos o simplemente crestas y valles. El método utiliza un dibujo lineal proporcionado por el usuario que al unirlo con el mapa de alturas del terreno pregenerado generará el terreno con dicho dibujo en forma de cresta o valle. Este método permite la modificación de grandes terrenos de forma sencilla, aunque no permite pequeños cambios y la altura final de la modificación no se puede predecir con exactitud.

Mediante interacción del usuario: Gain et al. [GMS09] permiten a través de una interfaz, que el usuario dibuje la silueta y los límites de una cadena montañosa para que mediante la propagación del ruido, el terreno crezca hasta cumplir la restricción. Para facilitar la interacción y hacerlo más intuitivo, Smelik et al. [STdB10] proponen un método en el que se pinta una vista superior del mapa, que abarca tanto las alturas, como el material del suelo. También entra en esta categoría el método para la creación de islas de Puig et al. [RRCR04], donde mediante una interfaz se puede ver la isla resultante a la vez que la dibujas.

Por otro lado, Hnaidi et al. [HGA*10] permiten que un diseñador dibuje curvas en 3D para controlar la forma del terreno generado. Otros métodos utilizan interfaces para que el usuario pueda modificar los terrenos [BMV*11] o hasta esculpirlos enteros en tiempo real [dB09].

Las modificaciones hechas mediante la interacción del usuario tienen la ventaja de que es mucho más fácil saber cómo será el resultado final o al menos facilita la prueba y error. Pero la desventaja es que para obtener unos resultados precisos el usuario ha de invertir el tiempo en crear el terreno [STBB14].

3. Método de Perlin

Para generar el terreno se ha utilizado el método de generación de terreno de Perlin, por tanto, para poder entender el método propuesto, en este punto se va a explicar su funcionamiento. Este método, para conseguir el aspecto realista de un terreno, cumple con el principio de la naturaleza. A una mayor escala del ruido habrá menos variaciones pero estas tendrán más influencia. Por esto, Perlin genera ruido a diferentes niveles, de forma que a medida que se aumenta el nivel, la frecuencia de puntos con valores aleatorios suben y la amplitud de onda baja, como se puede ver en la Figura 1. Al unir todos los niveles se obtiene la turbulencia, como se muestra en la Figura 2.

Para obtener el valor de estos puntos, Perlin busca los cuatro puntos conocidos que lo rodean, es decir, los puntos que se les ha dado un valor aleatorio en ese nivel según la frecuencia de puntos. Además, como se conoce la distancia entre el valor desconocido con cada uno de los cuatro puntos de ruido conocidos, aproxima su valor mediante la función de interpolación bilineal.

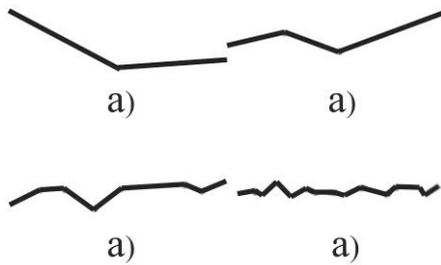


Figura 1: Ejemplo de ruido de Perlin para los cuatro primeros niveles, representados en una dimensión. a) frecuencia 2 y amplitud 128. b) Frecuencia de 4 y amplitud de 64. C) Frecuencia de 8 y amplitud de 32. D) Frecuencia de 16 y amplitud de 16.



Figura 2: Ejemplo de resultado de aplicar el método de Perlin en una dimensión.

Esta función, primero hace una interpolación lineal en la dirección x, obteniendo así los dos puntos P1 y P2, como se muestra en la Figura 3. El punto P1 se conoce que está a una distancia x1 de P01 y a una distancia x2 de P11, representados por un valor entre 0 y 1, además, esta distancia será la misma que de P2 a P00 y P10.

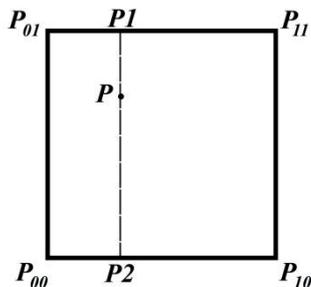


Figura 3: Puntos P1 y P2 obtenidos al aplicar la interpolación lineal en la dirección x entre el punto P con los valores de ruido conocidos.

A continuación se realiza otra interpolación lineal, esta vez en la dirección y pero utilizando a P1 y P2, conociendo además que la distancia que hay entre P con P1 es y1 y la distancia con P2 es y2. Dando como resultado el valor de P, siendo la ecuación utilizada la que se muestra a continuación (1), donde P00, P01, P10 y P11 son los cuatro puntos conocidos y P el punto a obtener. Hay que tener en cuenta que el resultado será el mismo si la interpolación lineal se hace primero en la dirección y.

$$P = y_1(x_1P_{00} + x_2P_{01}) + y_2(x_1P_{10} + x_2P_{11}) \quad (1)$$

Esta ecuación permite conocer el valor de cada punto en un nivel, pero para saber el valor final del punto se ha de ver su valor en cada uno de los niveles. Esta operación en cada uno de los puntos de un nivel se conoce como el ruido de Perlin, de forma que se puede reducir la ecuación (1) a:

$$P = \text{Ruido}(P) \quad (2)$$

Pero el método de Perlin lo que quiere obtener no es el ruido, si no la turbulencia completa. Es decir, la suma del ruido en los diferentes niveles, teniendo en cuenta que la frecuencia aumenta en potencia de dos y la amplitud se reduce en potencia de dos, a medida que sube el nivel. Si se parte de la ecuación anterior (2), la ecuación para obtener la turbulencia en un punto sumando el ruido de cada nivel n sería:

$$\sum_n \frac{\text{Ruido}(P \cdot 2^n)}{2^n} \quad (3)$$

Y por último, si se aplica la turbulencia a toda la matriz se obtiene el mapa de alturas con el que se representa el terreno, tal y como se muestra en el ejemplo de tres niveles en la Figura 4. En este ejemplo se puede ver como a medida que se aumenta el nivel, la frecuencia de puntos del ruido aumenta y la amplitud se reduce, siendo cada imagen una ampliación de un cuadrante de la siguiente. Al sumar todos los niveles de ruido se obtiene la turbulencia.

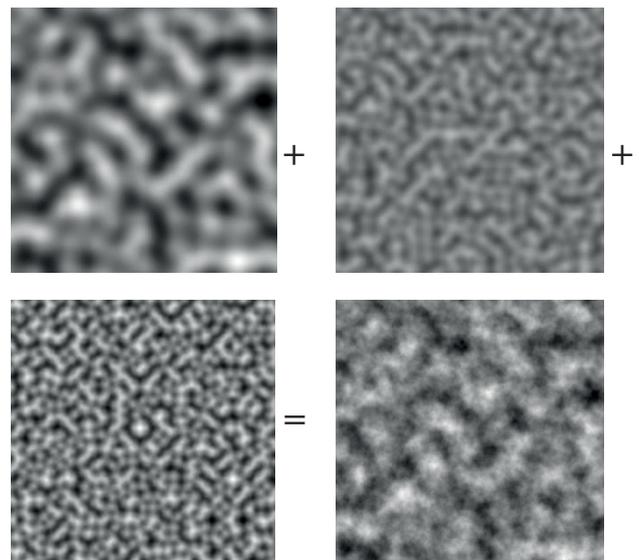


Figura 4: En las tres primeras imágenes, los tres niveles de ruido. Al sumarlos genera la turbulencia de la cuarta imagen.

4. Modelado del terreno

Como se ha visto en la sección anterior, no se ha encontrado en la literatura un método que permita la generación de un terreno con poco coste, al que se le pueda integrar una ruta con la posición y altura exacta, de forma que esto no elimine el realismo del terreno final.

Uno de los principales objetivos de nuestro modelo es poder generar un terreno con una ruta específica integrada y que esta no se vea modificada para poder adaptarse al terreno. Como puede verse en la Figura 5, el modelo que presentamos se divide en tres partes, detalladas por separado en esta sección. El primer paso explica cómo se extrae la ruta para poder utilizarla, a continuación, a partir de la ruta se extraen las ecuaciones para poder integrar la ruta en el terreno y se resuelve el sistema. Y por último, con el resultado se aplica el método de Perlin obteniendo el mapa de alturas final.



Figura 5: Diagrama de las partes en las que se divide el método propuesto.

3.1. Ruta GPS

En primer lugar, el método necesita que se le incluya un fichero con una serie de coordenadas, correspondientes a la ruta GPS. De este fichero, el método extrae estas coordenadas para obtener la posición y la altura de cada punto de la ruta. Hay que tener en cuenta que en una ruta GPS real los puntos de la ruta no tienen por qué estar a la misma distancia entre todos ellos. Esto puede ocasionar que entre dos puntos el método añada un valor aleatorio y modifique la ruta. Por eso, antes de poder pasar la ruta al método es necesario realizar un preproceso.



Figura 6: Representación de los puntos de una ruta real de 12 km.

En la Figura 6 se puede ver un ejemplo de este problema en una ruta GPS real vista desde arriba, donde al visualizar los puntos de la ruta se ve como hay zonas donde los puntos están mucho más juntos y otros donde entre punto y punto hay un gran vacío.

Este preprocesado, busca la mínima distancia entre puntos y completa el resto de la ruta con nuevos puntos para que se mantenga la distancia entre cada par. Para añadir nuevos puntos, el método los calcula a partir de una interpolación lineal con el punto anterior y el siguiente. Por otro lado, en el preprocesado también se eliminan aquellos puntos que están repetidos, para evitar que un mismo punto tenga dos valores distintos causados cuando la ruta incluye alguna parte con ida y vuelta por el mismo sitio. En la Figura 7 se muestra cómo al aplicar el preproceso completo, esta ya no tiene ninguna parte vacía, pasando de 100 puntos que tenía la ruta original a 290.

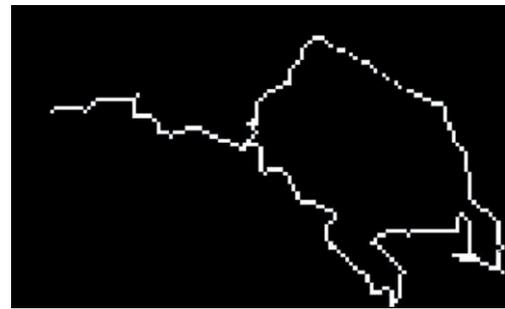


Figura 7: Representación de la ruta real de la Figura 6 preprocesada.

3.2. Sistema de ecuaciones

Con la ruta ya definida se puede pasar a obtener el terreno, el cual, como se ha dicho se hará mediante el método de generación de terrenos de Perlin. Pero si se aplica este método directamente el terreno obtenido es completamente aleatorio y lo que se quiere en este método es integrar una ruta con valores de altura exactos, por lo tanto, se necesita un paso previo donde se invertirá el proceso. Es decir, que como lo que se tiene es el valor final de los puntos que representarán la ruta y lo que se necesita calcular es el valor base de cada punto de la matriz, se invertirá el método de Perlin para obtenerlos y con el resultado se aplicará Perlin dando ya el resultado deseado.

Hay que aclarar que no tiene por qué calcularse todos los puntos de la matriz, solo aquellos que influyen en el valor final de la ruta, los demás seguirán manteniendo un valor aleatorio, que será lo que le dará realismo al terreno. Para ello, se recorre cada punto de la ruta para obtener los cuatro puntos con los que se realizaría la interpolación bilineal para cada nivel de la turbulencia.

Para ayudar a la explicación se va a utilizar un ejemplo con una ruta de un punto y mediante tres niveles de ruido. En este ejemplo se mostrará el proceso para obtener el valor de los puntos que tienen influencia sobre el punto de la ruta, en adelante llamado P . En la Figura 8 se puede ver el primer nivel, donde los puntos en rojo marcados con un círculo, diferenciados entre ellos por la coordenada que ocupan en la matriz, son los valores del ruido que se quieren calcular y que intervienen en la interpolación bilineal con P en ese nivel.

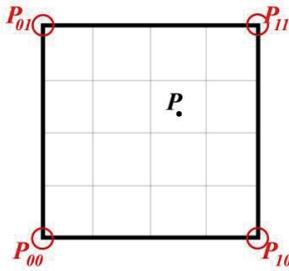


Figura 8: Puntos que influyen en el valor de P en el primer nivel en un ejemplo de tres niveles.

Al ser el primer nivel, esos cuatro puntos son los únicos que hay, por lo que el resto de puntos deberán interpolarse a partir de ellos una vez se haya obtenido su valor. En el siguiente nivel, la matriz pasa a tener nueve puntos, pero como puede verse en la Figura 9, cuatro de ellos son los mismos que los del nivel anterior por lo que pasan a estar en negro para diferenciarlos. De estos cinco nuevos puntos, solo influyen en P en este nivel tres de ellos, marcados con un círculo junto al punto que también influye del nivel anterior. Los dos puntos restantes, como en este ejemplo la ruta solo tiene un punto, no influyen así que se les dará un valor aleatorio, que será lo que permitirá al terreno final tener un aspecto realista.

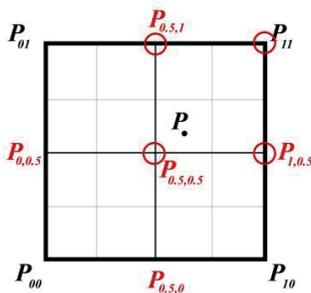


Figura 9: Puntos que influyen en el valor de P en el segundo nivel en un ejemplo de tres niveles.

En el siguiente nivel, al ser el último, aparecen todos los puntos restantes de la matriz. Diferenciados en la Figura 10 de los niveles anteriores marcándose en rojo. De estos nuevos puntos, al igual que en el nivel anterior solo tres de ellos influyen en el punto de la ruta junto con uno de los puntos del nivel anterior, marcados con un círculo.

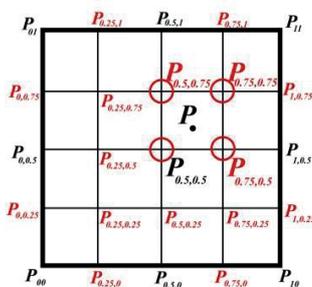


Figura 10: Puntos que influyen en el valor de P en el tercer nivel en un ejemplo de tres niveles.

Teniendo ya claro cómo se eligen los puntos con influencia en cada nivel se puede pasar a explicar cómo se obtienen los

valores de estos puntos. Aquí es donde se invierte el método de Perlin, de forma que a partir del valor del punto de la ruta, es decir, el valor de turbulencia de ese punto, se quiere obtener el valor de todos los puntos con influencia en cada nivel. La ecuación para este proceso sería igual que la utilizada por Perlin mostrada en la ecuación (3), sólo que esta vez se conoce el resultado y se desconoce el valor del ruido.

Cabe destacar que en el ejemplo solo se muestra como se obtiene la ecuación para calcular los puntos de influencia para un solo punto de la ruta. Pero en una ruta real hay muchos otros puntos, los cuales dependerán también de los mismos puntos o algunos de ellos. Al tratar cada punto de la ruta independiente de los demás, se consigue una ecuación de turbulencia por cada punto de la ruta, obteniendo así un sistema de ecuaciones (5) en la que las incógnitas serán todos los puntos que tienen alguna influencia en al menos un punto de la ruta.

$$\begin{cases} \sum_n \frac{\text{Ruido}(P_1 * 2^n)}{2^n} = P_1 \\ \vdots \\ \sum_n \frac{\text{Ruido}(P_x * 2^n)}{2^n} = P_x \end{cases} \quad (5)$$

Este sistema de ecuaciones, donde n es el nivel de ruido, permite al resolverlo, que se obtenga el valor que deben tener todos los puntos de influencia para que al usar con ellos el método de turbulencia de Perlin se obtenga la ruta. Es decir, el valor que deben tener para que al calcular el ruido de cada punto en cada nivel, sin olvidar la reducción de la amplitud a medida que este aumenta, al sumarlos dé como resultado el valor de la turbulencia de todos los puntos de la ruta correcto.

Hay que tener en cuenta que si se utiliza un tamaño de terreno y una ruta pequeños, es posible que haya más incógnitas que ecuaciones. Es decir, como el método obtiene una ecuación por cada punto de la ruta, si el número de puntos de esta es inferior al número de puntos de los que estos dependen, el número de incógnitas será mayor que al de las ecuaciones. Para resolver este problema en estos casos se añaden varios puntos en la ruta entre cada par, interpolando su valor. De esta forma, se aumenta el número de ecuaciones sin aumentar el número de incógnitas y resolviendo el problema.

Otro problema que puede ocurrir es que aparezcan ecuaciones reducibles entre ellas, lo cual hace que el sistema no tenga una sola solución. Pero como el resultado al que se quiere llegar incluye valores aleatorios para darle un aspecto realista de terreno, se le pueden dar valores aleatorios a las incógnitas necesarias para llegar a tener una sola solución, de forma que el resto se resolverán teniendo en cuenta estos valores. Con lo que el resultado seguirá manteniendo el valor correcto de los puntos de la ruta y se resolverá el problema.

3.3. Método de Perlin

Una vez obtenidos los valores base de la matriz, añadiendo además para aquellos puntos que no intervienen en la ruta un valor aleatorio, se pasa al último paso. Mediante el método de generación de ruido de Perlin, saltándose el paso de dar valor al ruido ya que eso es lo que ha hecho el método, se obtiene el mapa de alturas final.

Niveles	Tamaño de ruta (puntos)	Coste en segundos de cada tipo de ruta			
		Pendiente	Serpenteante	Circular	Ruta real
6	176	0.35	0.36	0.35	2.7
7	176	3.31	3.33	3.37	12.8
8	176	8.20	8.20	8.06	53.35
6	290	0.74	0.74	0.74	2.87
7	290	3.41	3.41	3.44	13.03
8	290	12.23	12.23	12.2	61.32

Tabla 1: Eficiencia del método a partir del coste en segundo usando diferentes rutas, diferentes niveles de ruido y diferentes tamaños.

Tipo de ruta	Tamaño de ruta (puntos)	Niveles de ruido	
		6	7
Pendiente	96	99.92%	99.92%
Pendiente	192	99.92%	99.93%
Pendiente	384	99.2%	99.4%
Serpenteante	96	100%	100%
Serpenteante	192	100%	100%
Serpenteante	384	100%	100%
Circular	96	100%	100%
Circular	192	100%	100%
Circular	384	100%	100%
Ruta real	176	80.32%	99.80
Ruta real	246	82.24%	99.82
Ruta real	290	88.12%	99.84

Tabla 2: Precisión de los resultados con rutas sintéticas de 96, 192 y 384 puntos y una ruta real con 176,246 y 284 puntos

4. Resultados

En esta sección se van a estudiar los resultados obtenidos en un equipo Intel Xeon E5 v2 3.70GHz, al aplicar el método a diferentes rutas: una sintética con pendientes, una sintética circular, una sintética serpenteante y una ruta real de 12 km. Además de aplicarlo con diferentes niveles de ruido y tamaños de las rutas. El estudio se ha realizado bajo los aspectos de eficiencia, precisión y el realismo obtenido.

4.1. Eficiencia

El método se ha probado en cuatro tipos de ruta, en tres niveles de ruido diferentes y con dos tamaños para cada ruta, una con 176 puntos y la otra con 290. Las tres primeras rutas son sintéticas, es decir, se han creado de forma ideal, sin cambios bruscos y sin necesidad de preprocesado, pero la última es una ruta GPS real. Para aumentar el tamaño de las rutas sintéticas se han alargado, pero en la real, para evitar modificar la forma de la ruta se ha usado para la pequeña un preprocesado más simple quedándose con algunas zonas vacías entre los puntos y para la grande, se ha realizado el preprocesado completo, es decir, se han añadido todos los puntos necesarios para evitar las zonas vacías entre puntos.

Los resultados obtenidos se pueden ver en la Tabla 1. Como se puede ver, el coste no se ve afectado por el tipo de ruta sintética, pero aumenta en la real. Por otro lado, al aumentar el número de puntos en la ruta también aumenta el coste, ya que cada punto implica una ecuación del sistema de ecuaciones, causando este aumento. Pero el aumento de coste más significativo viene dado al aumentar el número de niveles de ruido.

4.2. Precisión

Para ver la precisión de los resultados se ha comparado el valor de altura del mapa resultante en cada punto de la ruta, con los valores reales de la ruta. Esta comparación se ha realizado en cuatro tipos de ruta diferentes, con seis y siete niveles de ruido y con tres tamaños de ruta para cada tipo.

Los resultados se recogen en la Tabla 2, donde se puede ver que para rutas sintéticas la precisión es en su mayoría exacta, exceptuando el caso de la ruta con pendientes, en la que la precisión ha bajado un poco. Este error se reduce cuando el tamaño del camino aumenta, ya que es causado por el punto donde empieza y acaba cada pendiente, donde el valor difiere con el deseado varios decimales. La razón de esto es que al haber un cambio brusco en el camino, durante la interpolación bilineal, en alguno de los niveles el valor se ha suavizado un poco. Para la ruta real, la precisión baja un poco causado en gran parte en las zonas donde hay puntos muy cercanos y con alturas muy diferentes. Aunque aun así tiene un error inferior al 20% al usar un tamaño de la ruta pequeño y 6 niveles de ruido, pero al usar un nivel más de ruido este error pasa a ser de un 0.2%.

4.3. Realismo

Para poder ver la representación de cada mapa de alturas obtenido por el método, se ha utilizado el motor de juego UNITY3D (<https://unity3d.com>). Para las rutas sintéticas se ha utilizado un tamaño de ruta de 192 puntos y 7 niveles de ruido y el mapa de alturas resultante para cada una se puede ver en las figuras 9, 11 y 13. Y para la ruta real, se ha utilizado un tamaño de 290 puntos y 7 niveles de ruido, pudiendo ver el resultado en la figura 15. En las figuras 10,12,14 y 16 se muestra el resultado al visualizar los mapas de alturas.

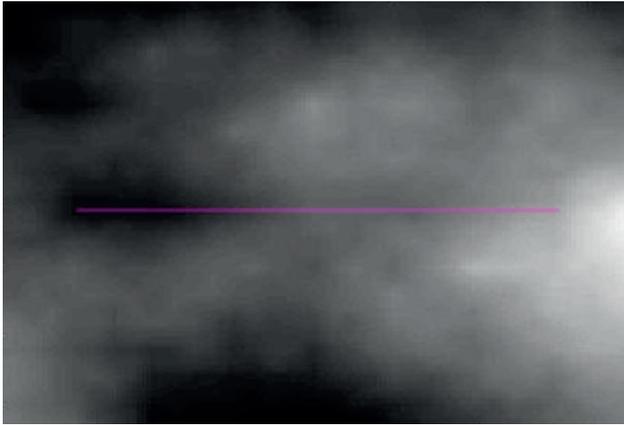


Figura 11: Mapa de alturas de una ruta con varias pendientes.

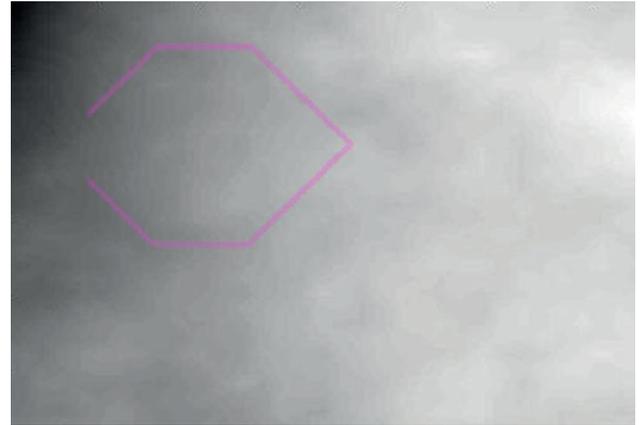


Figura 15: Mapa de alturas de una ruta circular.

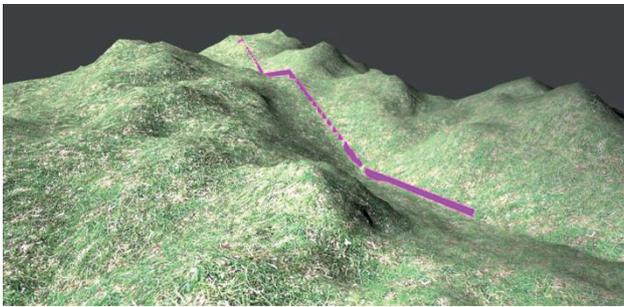


Figura 12: Terreno obtenido al utilizar las coordenadas de la ruta con varias pendientes de la Figura 9.

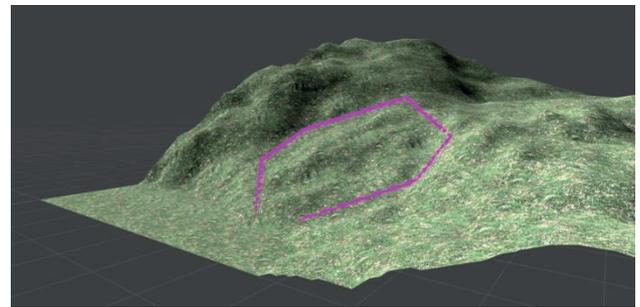


Figura 16: Terreno obtenido al utilizar las coordenadas de la ruta circular de la Figura 13.

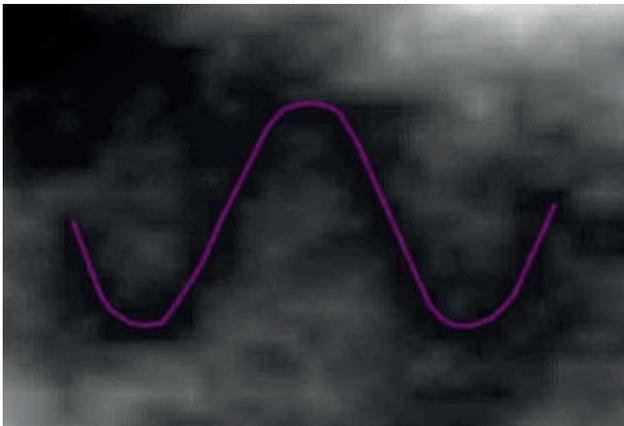


Figura 13: Mapa de alturas de una ruta serpenteante.

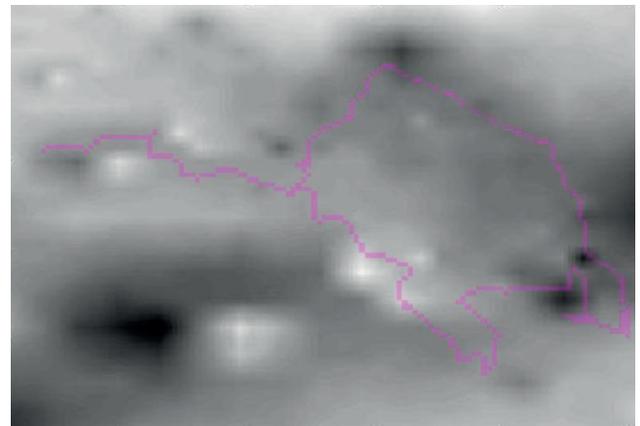


Figura 17: Mapa de alturas de una ruta real.

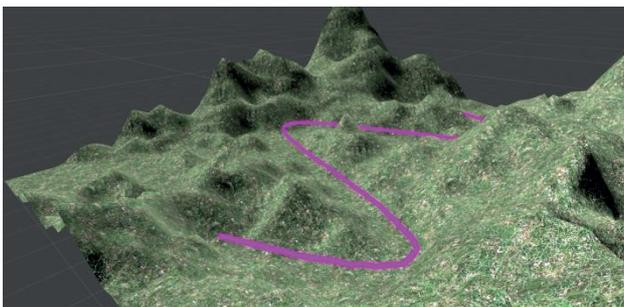


Figura 14: Terreno obtenido al utilizar las coordenadas de la ruta serpenteante de la Figura 11.



Figura 18: Terreno obtenido al utilizar las coordenadas de la ruta real de la Figura 15.

En los terrenos resultantes se ha añadido también la ruta original, al igual que en los mapas de alturas, representada por una línea rosa para mostrar así por donde pasa la ruta y como el método ha creado el terreno para poder integrarlas. El resultado además, muestra como el terreno no contiene pendientes bruscas ni formas antinaturales para poder adaptarse a la ruta, dando así un efecto visual realista.

5. Conclusiones

Este artículo presenta un método para la generación de terrenos con una ruta específica integrada de forma precisa, sin necesidad de la interacción con el usuario.

Los resultados obtenidos cumplen los objetivos iniciales. Las pruebas realizadas muestran que nuestro método es capaz de ofrecer un buen realismo del terreno al integrarle una ruta de cualquier tipo, sin que esta quede modificada y con un bajo coste. Este último se ve incrementado cuando la ruta es más compleja y con un mayor nivel de detalle, aunque el resultado visual y la precisión, para rutas complejas, no contienen mejoras significativas a partir de siete niveles de ruido y para las rutas más simples puede reducirse este nivel, permitiendo así mantener un bajo coste.

Las pruebas se han realizado con rutas ficticias y con una ruta real con un número de puntos suficientes que permiten valorar los resultados. En un futuro se quiere aplicar el método en rutas reales más grandes y terrenos de gran tamaño, además de no limitar el método a una ruta, sino utilizar varias rutas en diferentes partes del terreno. Por otro lado, como a medida que el tamaño del mapa y el número de niveles aumentan también aumenta el coste se quiere ampliar el método para que, dependiendo de la ruta y del tamaño deseado, busque las especificaciones mínimas necesarias para obtener un resultado satisfactorio y así, agilizar el proceso al máximo sin perder calidad en el resultado.

6. Referencias

- [Bel07] BELHADJ F.: Terrain modeling: a constrained fractal model. In *AFRIGRAPH '07: Proceedings of the 5th Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (2007), 197–204.
- [BMV*11] BERNHARDT A., MAXIMO A., VELHO L., HNAIDI H., CANI M.-P.: Real-time terrain modeling using cpu-gpu coupled computation. In *SIBGRAPI '11: Proceedings of the 24th Conference on Graphics, Patterns and Images* (2011), 64–71.
- [dB09] DE CARPENTIER G. J., BIDARRA R.: Interactive gpu based procedural heightfield brushes. In *FDG '09: Proceedings of the 4th International Conference on the Foundations of Digital Games* (2009).
- [DP10] DORAN J., PARBERRY I.: Controlled procedural terrain generation using software agents. *IEEE Transactions on Computational Intelligence and AI in Games* 2, 2 (2010), 111–119.
- [FFC82] FOURNIER A., FUSSELL D., CARPENTER L.: Computer rendering of stochastic models. *Communications of the ACM* 25, 6 (1982), 371–384.
- [GMS09] GAIN J., MARAIS P., STRASSER W.: Terrain sketching. In *I3D '09: Proceedings of the Symposium on Interactive 3D Graphics and Games* (2009), 31–38.
- [HGA*10] HNAIDI H., GUÉRIN E., AKKOUCHE S., PEYTAVIE A., GALIN E.: Feature based terrain generation using diffusion equation. In *Computer Graphics Forum: Proceedings of Pacific Graphics 2010* 29, (2010), 2179–2186.
- [KBK09] KRIŠTOF P., BENES B., KRIVANEK J., ŠTAVA O.: Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum* 28, 2 (2009).
- [KU07] KAMAL K. R., UDDIN Y. S.: Parametrically controlled terrain generation. In *GRAPHITE '07: Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia* (2007), 17–23.
- [Mil86] MILLER G. S. P.: The definition and rendering of terrain maps. In *SIGGRAPH '86: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* 20, 4 (1986), 39–48.
- [MKM89] MUSGRAVE F. K., KOLB C. E., MACE R. S.: The synthesis and rendering of eroded fractal terrains. In *SIGGRAPH '89: Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques* (1989), 41–50.
- [Per85] PERLIN K.: An Image Synthesizer. In *SIGGRAPH '85: Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques* 19 (1985), 287–296.
- [PVR14] PUIG-CENTELLES A., VARLEY P. A. C., RIPOLES O., CHOVER M.: Automatic Terrain Generation with a Sketching Tool. *Multimedia Tools and Applications* 70, (2014), 1957–1986.
- [RRCR04] REBOLLO C., REMOLAR I., CHOVER M., RAMOS J. F.: A Comparison of Multiresolution Modelling in Real-Time Terrain Visualisation. *Springer-Verlag Berlin Heidelberg* 3044, (2004), 703–712.
- [SS05] STACHNIAK S., STÜRZLINGER W.: An algorithm for automated fractal terrain deformation. *Computer Graphics and Artificial Intelligence* 1 (2005), 64–76.
- [STBB14] SMELIK R., TUTENEL T., BIDARRA R., BENES B.: A survey on procedural modeling for virtual worlds. *Computer Graphics Forum*, 33, 6 (2014), 31–50.
- [STdB10] SMELIK R. M., TUTENEL T., DE KRAKER K. J., BIDARRA R.: Interactive creation of virtual worlds using procedural sketching. In *Proceedings of Eurographics 2010: Short Papers* (2010).
- [ZSTR07] ZHOU H., SUN J., TURK G., REHG J. M.: Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics* 13, 4 (2007), 834–848.