

DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE SÍNTESIS DE TEXTURAS PARA APLICACIONES CON IMÁGENES DIGITALES

D. Serna¹, J.P. García¹, D.S. Fernández^{1,2}

*1 Grupo de investigación GEPAR, Facultad de Ingeniería, Universidad de Antioquia UdeA, Calle 70 No. 52-21, Medellín, Colombia. jpablo.garcia@udea.edu.co.

2 Departamento de Ingeniería Electrónica y de Telecomunicaciones, Facultad de Ingeniería, Universidad de Antioquia UdeA, Calle 70 No. 52-21, Medellín, Colombia. david.fernandez@udea.edu.co.

Palabras clave: síntesis de textura, textura, procesamiento digital de imágenes.

Resumen

En el desarrollo de este trabajo se creó un algoritmo de síntesis de textura basada en píxeles teniendo en cuenta el trabajo previo de Li-Yi Wey y Mark Levoy sobre este tema. Se explicará brevemente que es la síntesis de textura y sus posibles aplicaciones, a su vez se explicará el funcionamiento del algoritmo [2]. Finalmente se mostrarán ejemplos de diferentes resultados de síntesis realizadas con su respectiva comparación con trabajos previos.

1 Introducción

Desde finales de los años sesenta del siglo XX se ha empezado a dar importancia a la síntesis de texturas en el procesamiento de imágenes y en los campos relacionados con gráficos por computadora como la producción y simulación 3D. Esto ha llevado a la creación de métodos de síntesis de textura eficientes que cada vez son más exigentes en cuanto a que buscan mantener baja carga en el procesamiento para la máquina y alta calidad en las texturas sintetizadas para resoluciones según la necesidad.

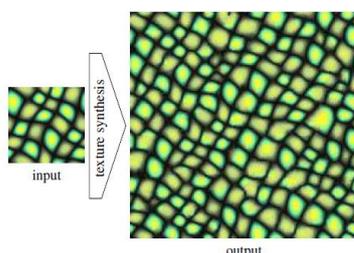


Figura 1. A la izquierda la textura muestra. A la derecha una textura de mayor resolución sintetizada a partir de la textura inicial [4].

La síntesis de textura basada en ejemplos, que es el objeto de estudio en el presente trabajo, permite que, a partir de una textura de muestra, normalmente de menor resolución, se sintetice una nueva textura en la salida, de resolución arbitraria que mantenga las propiedades de la textura inicial (ver Figura

1) [4]. En los últimos años se han realizado progresos significativos en algoritmos de síntesis de textura basadas en ejemplos, actualmente existen numerosos métodos entre los que destacan la síntesis por escala arbitraria (Arbitrary-scale), por copia de píxeles, (Based-pixels), bloques de píxeles (Blocks of pixels), o basadas en parches (Patch-based), estas últimas a diferencia de la síntesis basadas en píxeles no cumplen con la eficiencia requerida en aplicaciones ejecutadas en tiempo real [6].

Los autores Alexei Efros y Thomas Leung, fueron pioneros en este tema al crear una síntesis de texturas basada en ejemplos con resultados aceptables con un algoritmo simple, sin embargo el procedimiento era lento y solo funcionaba para una gama de texturas reducida. Basándose en el anterior trabajo, Li-Yi Wei y Marc Levoy, crearon un algoritmo más eficiente, modular, con baja complejidad computacional y con resultados aceptables en una amplia gama de texturas.

El algoritmo de Wei & Levoy se convirtió en la base de trabajos posteriores en síntesis de textura. Podemos citar, entre otros, el trabajo de Seunghyup Shin quien creó un algoritmo iterativo con análisis en frecuencia, que sacrifica procesamiento computacional para mejorar los resultados en la calidad visual de las síntesis, y el trabajo de Ashikmin que modificó el algoritmo original de Wei y Levoy para mejorar los resultados en texturas naturales manteniendo la eficiencia inicial.

Estos son algunos de los muchos autores a los que se les puede atribuir investigaciones sobre síntesis de textura basadas en ejemplos que proponen técnicas basadas en el método de copia de píxeles (Pixel-Based Texture Synthesis), método que ha demostrado una sorprendente capacidad para sintetizar una amplia gama de texturas que van desde texturas simples regulares hasta texturas estocásticas [8].

El enfoque del presente trabajo toma como base el trabajo realizado por Wei y Levoy, el cual tiene resultados muy convincentes en calidad visual y rendimiento, por lo que ha sido base de múltiples investigaciones recientes como ya se mencionó. Se concentra en mejorar este algoritmo para su utilización en el desarrollo de imágenes digitales para su contenido 3D o 2D, donde utilizar mapas de texturas es indispensable para mejorar el realismo y/o detalle visual de los

personajes, objetos o superficies que pueden ser usados para diversos contenidos digitales. El uso de algoritmos de síntesis de texturas puede proporcionar una herramienta ágil para crear texturas de una resolución y tamaño óptimo para ser usadas en diferentes aplicaciones.

2 Metodología

El dibujo hecho a mano, las fotografías y la síntesis de textura son algunas de las fuentes donde podemos obtener texturas, sin embargo esta última se destaca por su fácil uso. Se debe dar un ejemplo o muestra de textura y un conjunto de parámetros de síntesis los cuales serán procesados y arrojarán como resultado una imagen de una resolución requerida sin repeticiones antinaturales, produciendo imágenes escalables mediante el manejo de las condiciones de frontera de la imagen.

El proceso de la síntesis de textura está compuesto por dos componentes principales: Análisis y Síntesis. La primera se refiere a la caracterización de una imagen finita mediante modelos estructurales y estocásticos, y la segunda se refiere al proceso de usar la caracterización obtenida en el análisis para generar la nueva imagen de nuevas dimensiones [4].

Existen dos tipos de algoritmos que actualmente son estudiados en los diferentes procesos de la síntesis de textura, estos son (ver Figura 2):

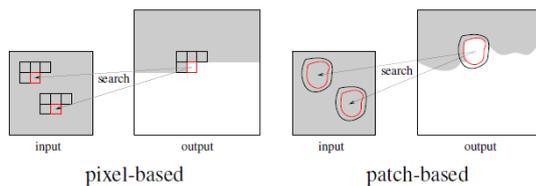


Figura 2. A la izquierda, esquema del funcionamiento de la síntesis basada en píxel. A la derecha, esquema de la síntesis basada en parches.

2.1 Síntesis de textura por píxel:

Así como otras muchas técnicas, se basa en criterios de similitud. La imagen se va creando píxel a píxel y el criterio a la hora de seleccionar el color del píxel es examinando el resto de la imagen y escoger el píxel que tenga un entorno conocido más similar al que se está a punto de llenar. Cuando se identifica el píxel, se copia el color.

2.2 Síntesis de textura por parches:

Es similar a los algoritmos de síntesis de texturas por píxeles, pero esta reemplaza los píxeles por parches que abarcan varios píxeles. En estos algoritmos se debe definir un criterio de similitud para comparar el parche que se va a llenar con todos los posibles parches en la imagen de muestra [5].

Basándose en una simple estrategia de copia de un píxel a la vez, estas técnicas han demostrado su sorprendente capacidad de sintetizar una amplia variedad de texturas de alta calidad que van desde texturas regulares a estocásticas. Sin embargo, algunas texturas resultantes presentan artefactos visuales, tales

como borrosidad (blurring) y basura creciente (garbage growing), como se puede verificar en los casos de Levoy, Efros y Ashikhim [8]. Es decir, un método basado en píxeles tiene una tendencia a desdibujar características o crecer estructuras de pequeña escala en las texturas sintetizadas.

Los inconvenientes que presentan estos dos tipos de metodologías de síntesis de textura se pueden reducir añadiendo complejidad en los algoritmos y sacrificando eficiencia en la mayoría de los casos. [8].

3 Algoritmo Wei & Levoy

Como se mencionó anteriormente, el presente trabajo se basará principalmente en el trabajo realizado por los investigadores Li-Yi Wei y Marc Levoy quienes realizaron un algoritmo rápido computacionalmente, con resultados visualmente aceptables, compatible con gran variedad de texturas, con posibilidad de generar texturas aptas para mosaico (tiled) y que han sido base de nuevas investigaciones gracias a la flexibilidad de su algoritmo [2].

El algoritmo de Wei y Levoy utiliza un enfoque de síntesis basada en modelos de textura de Campos Aleatorios de Markov (Markov Random Field o MRF) que han sido ampliamente implementados en los algoritmos actuales. Esto se logra mediante la generación de la imagen de salida pixel por pixel en forma de Scanline, asignando a cada pixel en la imagen de salida, el color del pixel en la imagen muestra con el vecindario más similar al vecindario del pixel que se está asignando. En la versión más básica del algoritmo, se realizan los siguientes pasos:

Paso 1: La imagen de salida se inicializa con ruido blanco gaussiano.

Paso 2: La imagen de salida se va generando en forma de Scanline:

- En la imagen de salida, se posiciona un vecindario en forma de L de tamaño fijo, sobre el pixel que se va a sintetizar, véase la Figura 3.

- Se realiza una búsqueda en la muestra de entrada para un píxel con un vecindario lo más similar posible a los vecindarios en la textura sintetizada.

- El valor actual del píxel en la imagen de salida es copiado desde la posición en la imagen de muestra identificada como la más similar por la búsqueda realizada (Ver Figura 4) [3].

La similitud entre dos vecindarios N_1 y N_2 son procesados de acuerdo a la distancia entre ellos (L_2) la cual es una suma de las diferencias al cuadrado sobre todos los píxeles del vecindario para el valor del pixel en la posición en particular (ver ecuación 1).

$$D(N_1, N_2) = \sum_{p \in N} \{(R_1(p) - R_2(p))^2 + (G_1(p) - G_2(p))^2 + (B_1(p) - B_2(p))^2\} \quad (1)$$

Donde R, G, y B se refieren a los valores del pixel en los canales rojo, verde y azul respectivamente en la posición p, y

los subíndices 1 y 2 hacen referencia a la imagen ya sea la de muestra o la sintetizada.

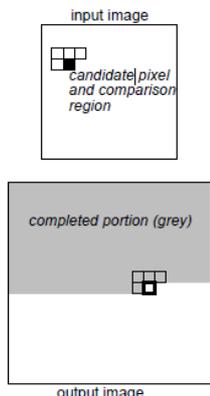


Figura 3. Forma en L usada en el algoritmo de Wei y Levoy y barrido tipo Scanline (Zona gris).

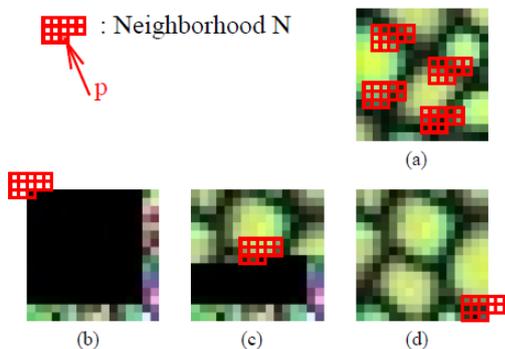


Figura 4. (a) Es la textura muestra. Entre (b) y (d) se muestran diferentes etapas de la síntesis de la imagen de salida. Los píxeles de la textura sintetizada se asignan en forma Scanline [2].

En el proceso de construcción de los vecindarios las fronteras de la imagen son tratadas en forma toroidal, esto permite que los bordes de la imagen sintetizada coincidan verticalmente y horizontalmente, este tipo de imágenes son conocidas como “tiled image”, característica necesaria para texturizar objetos tridimensionales de gran extensión donde la textura se mapea en forma de mosaico para evitar el uso de imágenes en alta resolución (mayores a 1024x1024) y ahorrar espacio de almacenamiento.

El tamaño del vecindario debe estar a una escala mayor que la estructura regular que compone la textura para capturar de forma óptima sus componentes y no se pierda información a la hora de realizar la síntesis. Dado que la cantidad de trabajo para evaluar la ecuación (1) es directamente proporcional al número de píxeles del vecindario escogido, el tiempo de ejecución del algoritmo se ve afectado radicalmente si la imagen es de una resolución mayor.

Este aspecto es mejorado con una síntesis en multiresolución, que consiste en realizar el procedimiento anterior en varias versiones de la imagen muestra en resoluciones más bajas

mediante el uso de pirámides, cuyas síntesis se unen para formar la imagen de salida. Esto se logra al utilizar la inicialización con ruido blanco solo en el nivel de resolución más bajo, los niveles siguientes se inicializan con la síntesis obtenida en el nivel inmediatamente anterior. De esta forma con los niveles de resolución más bajos obtenemos la síntesis de los patrones más grandes de la textura inicial y con los niveles más altos obtenemos la síntesis de los patrones más pequeños. Con este método es posible la utilización de vecindarios más pequeños lo cual mejora el rendimiento del algoritmo [2] [3].

4 Modificación al algoritmo

El algoritmo anterior tiene un buen desempeño con texturas naturales, y presenta algunos artefactos de borrosidad en texturas simples. Esto se debe al método de selección de similitud usado (distancia L2), el cual al ser una medida estadística poco robusta no tiene la capacidad para diferenciar entre vecindarios similares de una textura simple (por ejemplo dos esquinas de dos ladrillos diferentes en una misma imagen), lo que se traduce a una copia sucesiva del primer patrón encontrado en la textura, esto genera borrosidad y crecimiento de estructuras de pequeña escala en las texturas sintetizadas.

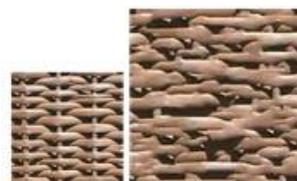


Figura 5. A la derecha la textura muestra. A la izquierda la textura generada con el algoritmo de Wei y Levoy, donde se puede apreciar cómo se expanden algunas zonas de la textura original debido a la selección repetitiva de un mismo píxel. Esto ocurre debido a que esta textura simple contiene varios vecindarios muy similares que la medida L2 no logra diferenciar.

Es posible mejorar esta selección de píxeles agregando a la medida L2, una segunda medida estadística que aumente la diferenciación entre vecindarios y evite que se tomen como iguales. Esta medida adicional debe ser simple matemáticamente para evitar aumentar en gran medida el costo computacional del algoritmo. La varianza de los errores entre píxeles de dos vecindarios que son comparados es una medida que cumple estas especificaciones ya que utiliza una medida ya obtenida por el algoritmo (la distancia L2 a partir de la cual se encuentra el error medio) y añade separación a la descripción estadística de la similitud de varios vecindarios (vecindarios en la textura muestra) con un vecindario referente (el vecindario del píxel que será pintado).

Lo anterior se incluyó en el algoritmo original al establecer como criterio de similitud una ecuación de energía que contiene el cálculo del error medio y la varianza de los errores entre los píxeles de los vecindarios, tal como se muestra a continuación:

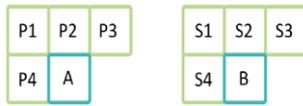


Figura 6. Vecindarios a comparar, donde A es cualquier píxel en la imagen muestra y B es el píxel que será pintado en la textura de salida copiando el valor del píxel A que tenga el vecindario con mayor similitud al vecindario de B.

$$E_n = (P_n - S_n)^2 \quad (2)$$

$$\bar{E} = \frac{\sum_{n=1}^k E_n}{k} \quad (3)$$

$$\sigma^2 = \sum_{n=1}^k (E_n - \bar{E})^2 \quad (4)$$

$$Dif(A, B) = C_1 * \bar{E} + C_2 * \sigma^2 \quad (5)$$

Donde A y B son los píxeles cuyos vecindarios serán comparados, k es el número de píxeles de un vecindario, y los escalares C1 y C2 representan el peso del error medio y la varianza respectivamente. El valor arrojado por la ecuación de energía (5) representa la similitud entre los vecindarios A y B, donde valores bajos equivalen a mayor similitud entre los vecindarios comparados. Aplicando este criterio se obtienen menos y mejores píxeles candidatos para la textura de salida.

Este criterio de similitud agrega complejidad computacional y disminuye el rendimiento del algoritmo, por esta razón solo es usado en el nivel de imagen de resolución más bajo de la síntesis. En los pasos siguientes para las resoluciones mayores se usa el criterio original de distancia (L2) (Ver Figura 7).

El desempeño de este algoritmo está limitado al tamaño del vecindario y el nivel de pirámide seleccionado además del valor de los parámetros C1 y C2 de la ecuación 5.

5 Experimento y Resultados

Como se mencionó en la sección 4, se realizó una modificación al algoritmo original de Wei y Levoy, modificando el criterio de similitud entre vecindarios para la elección de píxeles candidatos que pueden ser copiados a la textura sintetizada, dicha modificación se realiza para la síntesis de nivel más bajo, es decir, la imagen de menor resolución. Comparando los resultados del algoritmo modificado y el original se verifica que la modificación consigue disminuir la borrosidad producida por el algoritmo original en una amplia gama de texturas como se puede observar en las imágenes de la Figura 8.

Para todas las síntesis mostradas en este trabajo se usan los valores de 0.7 para C1 y 0.3 para C2 en la ecuación 5, elegidos por su buen desempeño tras varias pruebas. Cabe resaltar que a pesar de que para muchas texturas el algoritmo presenta resultados aceptables, para las texturas determinísticas con patrones muy repetitivos sigue presentando algunos artefactos visuales.

Los resultados se realizaron en base a un banco de texturas compartido por los autores Wei y Levoy mediante una página

web. Estas mismas texturas son usadas por diversos autores como Seunghyup Shin y Ashikmin para mostrar los resultados de sus algoritmos, y muchos otros que investigan nuevas metodologías en el tema de síntesis de texturas basadas en ejemplos [13].

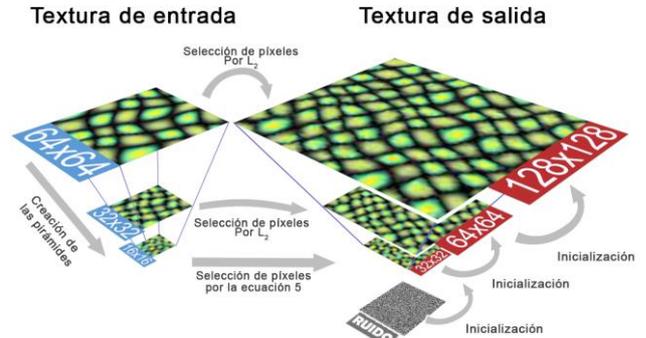
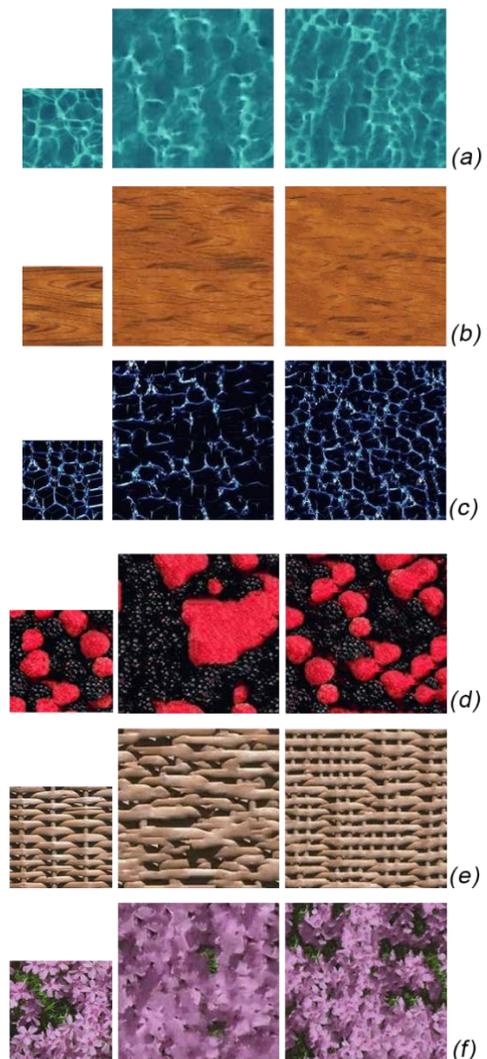


Figura 7. La síntesis se realiza comenzando en los niveles más bajos de resolución donde los píxeles son seleccionados usando la ecuación 1 y continúa hacia los niveles más altos de resolución usando el criterio de distancia sugerido por Wei y Levoy.



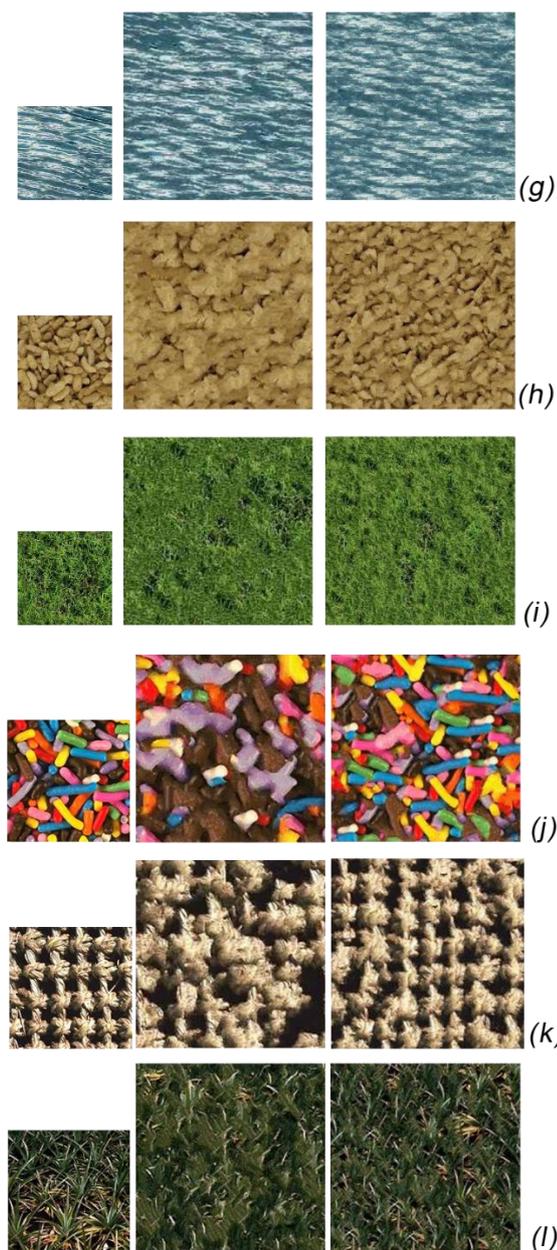


Figura 8. Se exhibe las diferentes síntesis realizadas por el método propuesto (Imágenes a la derecha) con sus respectivas muestras (imágenes a la izquierda) y a su vez se comparan con los resultados obtenidos por Wei y Levoy en su investigación (imágenes del medio).

De igual forma se probó la respuesta del algoritmo con texturas fuera del banco de muestras inicial. Se tomaron algunas imágenes con una cámara digital común, de algunas texturas que se encuentran en espacios como oficinas o exteriores sin ajuste de las condiciones de luz ni retoques o ajustes de color. El proceso anteriormente descrito fue realizado y se obtuvieron los resultados expuestos en la Figura 9.

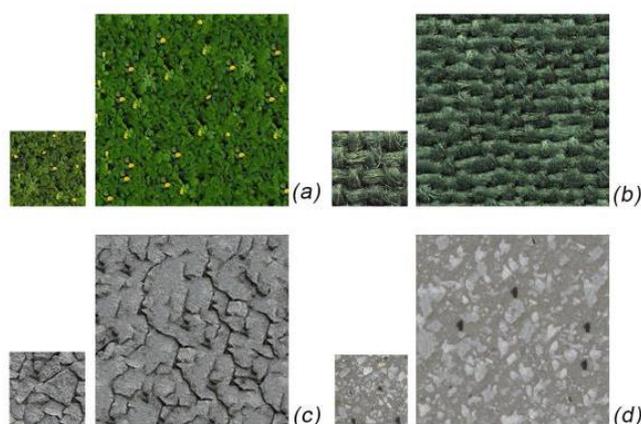


Figura 9. Las muestras de (a), (b), (c) y (d) tienen una resolución de 128×128 , y a partir de estos se generan texturas a una resolución 512×512 .

El algoritmo tiene un buen rendimiento, característica común en las síntesis basadas en píxeles manteniendo los tiempos de ejecución cortos presentados por el algoritmo original de Wei y Levoy, este hecho se puede observar en la tabla 1, donde se exponen los tiempos de acuerdo a las figuras sintetizadas y en las resoluciones presentadas en la Figura 10.

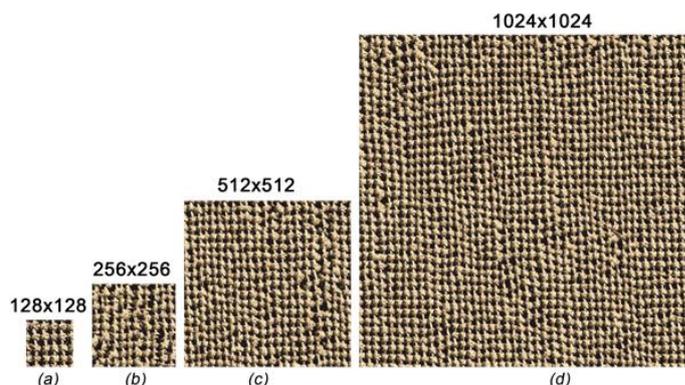


Figura 10. Síntesis desde una muestra de 128×128 a diferentes resoluciones de salida.

Los tiempos de ejecución dependen del tamaño del vecindario, la resolución de salida, la resolución de la muestra, y el nivel máximo de la pirámide, y técnicamente del hardware en donde se ejecute el algoritmo.

6 Conclusiones

El algoritmo propuesto sintetiza correctamente una amplia gama de texturas, algunas de ellas con resultados observables sobresalientes a los obtenidos por Wei y Levoy, debido a la modificación realizada en el método de comparación de vecindarios que permitió la selección de mejores píxeles candidatos en las texturas determinísticas y en algunas texturas naturales (ver Figura 12).

TABLA I
RENDIMIENTO DE LA HERRAMIENTA DE SÍNTESIS

Textura	Resolución Muestra	Resolución Salida	Nivel Pirámide	Tiempo
Fig. 10(b)	128	256	2	4min 37s
Fig. 10(c)	128	512	2	10min 19s
Fig. 10(d)	128	1024	2	39min 51s
Fig. 9(a)	128	256	2	2min 25s
Fig. 9(b)	128	256	2	2min 23 s
Fig. 9(d)	128	256	1	8min 41s
Fig. 8(e)	128	256	2	2min 35s
Fig. 8(f)	128	256	1	10min 26s
Fig. 8(a)	128	256	1	10 min 52s
Fig. 8(g)	128	256	2	5min 25s

Unidades en el Sistema MKS: s= segundos, min= minutos.

Las distintas pruebas que se ejecutaron para probar el algoritmo presentado en este estudio se realizaron en un equipo con procesador Intel Core i5 de primera generación a 3.1 GHz con 8GB de memoria RAM DDR3 ejecutado sobre C#.

Este método de síntesis no es útil como método de compresión de texturas, algo deseado en aplicaciones como los videojuegos, debido a que los tiempos de síntesis superan el segundo, lo cual ocasiona tiempos de carga muy elevados cuando estos tienen varias texturas, sin embargo el método provee una herramienta fuerte que ahorra tiempo en el desarrollo de aplicaciones gráficas.

Los algoritmos de síntesis de texturas representan una solución efectiva para construir texturas estocásticas que disminuyan la sensación de repetición de una textura pequeña mapeada en forma de mosaico en un objeto 3D, característica que es comúnmente reconocida como antinatural.

7 Trabajo Futuro

La aplicación de métodos de optimización para la elección de los mejores valores de los escalares C1 y C2 en la ecuación 4, utilizada como criterio de similitud entre dos vecindarios y que amplió la gama de texturas que puede sintetizar el algoritmo de Wei y Levoy, puede mejorar aún más la calidad de la síntesis y posiblemente ampliar más dicha gama.

El uso de métodos de optimización de código podría mejorar el rendimiento del algoritmo que si bien es aceptable aún es lento para la exigencia en tiempos de desarrollo de la industria

digital que cada día busca producir contenido de calidad más rápido.

Además la optimización del método podría permitir el uso del criterio de similitud propuesto en todos los niveles de la pirámide aumentando la calidad de la síntesis sin sacrificar tiempo de ejecución.

Referencias

- [1] A. Efros and T. Leung. Texture synthesis by nonparametric sampling. In International Conference on Computer Vision, vol. 2, pp. 1033–1038, 1999.
- [2] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. Proceedings of SIGGRAPH 2000, 479–488, ISBN 1–58113–208–5, July 2000.
- [3] M Ashikhmin. Synthesizing natural textures[C] // Proc of ACM Symposium on Interactive 3D Graphics, North Carolina: ACM Press: 217-226, 2001.
- [4] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk. State of the art in example-based texture synthesis. In EG-STAR, 2009. 1, 4
- [5] N. B. Cortés, “Restauración de imágenes mediante un modelo matemático basado en las técnicas de detección de bordes y propagación de texturas Restauración de imágenes mediante un modelo matemático basado en las técnicas de detección de bordes y propagación de texturas,” 2011.
- [6] H. F. Cui, X. Zheng, and T. Ruan, “An efficient texture synthesis algorithm based on WT,” Proc. 7th Int. Conf. Mach. Learn. Cybern. ICMLC, vol. 6, no. July, pp. 3472–3477, 2008.
- [7] A. Wong and W. Bishop, “Practical Perceptually Adaptive Color Texture Map Compression for 3D Video Games.”
- [8] Shin, “On Pixel-Based Texture Synthesis by Non-parametric Sampling,” pp. 1–24.
- [9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. 2nd ed. New Jersey: Prentice Hall, 2001.
- [10] M. Petrou and C. Petrou, *Image Processing: The Fundamentals*. West Sussex, United Kingdom: John Wiley & Sons, 2010.
- [11] B. Burley and D. Lacewell, “Ptex: Per-face texture mapping for production rendering,” Comput. Graph. Forum, vol. 27, no. 4, pp. 1155–1164, 2008.
- [12] Kwatra, Vivek. Turk, Greg. Wei, Li-Yi. State of the Art in Example-based Texture Synthesis. Eurographics (2009).
- [13] L.-Y. Wei and M. Levoy, “Texture Synthesis Examples,” 2000. [Online]. Available: <http://graphics.stanford.edu/projects/texture/demo>.