

Representación de sombras suaves en tiempo real usando mapas de distancia

J. Gumbau, M. Chover

Universitat Jaume I, Castellón, España

Abstract

Este trabajo presenta una nueva técnica para la representación de sombras suaves en tiempo real mediante mapas de sombras. Para ello se extiende el algoritmo de filtrado por proximidad (percentage-closer filtering o PCF) de generación de sombras suaves mediante multi-muestreo de forma que se varía la distancia entre muestras por píxel dependiendo de la distancia entre el ocluser y la sombra proyectada. Además, se usa un filtro gaussiano de tamaño variable por píxel para suavizar la penumbra de la sombra de forma que ésta sea más precisa cuanto más cerca esté del ocluser. Finalmente, el mapa de distancias permite que la intensidad de la sombra se atenúe al alejarse del objeto que la proyecta.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display algorithms

1. Introducción

Las sombras son un elemento indispensable para transmitir realismo a las imágenes generadas por ordenador. Sin ellas se pierde en gran parte la sensación de profundidad. La generación de sombras en tiempo real puede dividirse en dos grandes subapartados: las basadas en geometría y las basadas en imágenes.

Las técnicas basadas en geometría utilizan volúmenes geométricos que representan la forma de la sombra pudiendo proyectarse con una precisión por píxel sobre la escena. El problema de esta técnica es que hay que calcular un volumen geométrico que puede cambiar cada fotograma y que la representación de la sombra requiere generalmente la evaluación de todos los píxeles de la pantalla una vez la escena ya está renderizada, por lo que se sobrecarga bastante la etapa de procesado de píxeles. Además, introducen geometría extra que hay que procesar cada fotograma, lo que hace que sean poco prácticas en escenas con una gran carga poligonal. Por último, al estar basadas en geometría no funcionan bien con objetos con transparencias. Por ejemplo, en la representación de escenas naturales la vegetación suele dibujarse como primitivas con mapas con transparencia.

Por otra parte, las técnicas basadas en imágenes, también

conocidas como mapas de sombra (*shadow maps*), capturan la profundidad de la escena desde el punto de vista de la fuente de luz. Esta información se almacena en una textura a la que se accede cuando se están dibujando los objetos de la escena susceptibles de recibir la sombra, de forma que se pueda decidir con una simple comparación si un determinado píxel está en sombra o es afectado por la fuente de luz. Este tipo de técnicas son las más utilizadas en aplicaciones interactivas debido a que se realizan totalmente por *hardware* y a que son muy escalables.

El problema de este tipo de métodos es que, al estar basados en texturas, están limitados por la resolución y la precisión de la imagen donde se almacena el mapa de sombras. Esto causa que las sombras generadas presenten problemas de dentado, que pueden paliarse en gran medida usando técnicas como las aportadas en los trabajos [MT04], [SD02], [WSP04] o [ZSXL06].

Otro problema de las sombras generadas por estos métodos es que generan bordes demasiado acusados. El método básico del mapa de sombras se encarga de decidir qué zonas de la pantalla tienen que recibir sombra y cuáles no, pero no en qué proporción. En la realidad, las sombras presentan bordes difuminados (o penumbras), causados por que todas las luces son luces de área, que son las responsables de que



Figure 1: Utilización de nuestro método para el cálculo de sombras suaves y la simulación de penumbras realistas.

cualquier sombra tenga una penumbra, en mayor o menor medida. Sin embargo, en las aplicaciones interactivas, los cálculos necesarios para usar luces de área son demasiado costosos, de forma que se recurre a otras técnicas que permiten dibujar sombras con penumbra sin tener que simular la interacción de las luces de área para generar este tipo de sombras.

Estos métodos de antidentado para mapas de sombras, como los basados en filtro de porcentaje de proximidad (en inglés *percentage-closer filtering* o PCF), realizan múltiples accesos al mapa de sombras y realizan un promedio para determinar la cantidad de luz que debe recibir un punto dado. Gracias a esto puede recrear una penumbra artificial, pero al no tener en cuenta el comportamiento real de la luz esto le resta realismo a las sombras: el tamaño de la penumbra es constante, independientemente de su proximidad al ocluidor, algo totalmente incoherente con la realidad.

El método que se presenta en este artículo se basa en una mejora a los algoritmos basados en mapas de sombras que consiste en la implementación de un algoritmo basado en mapas de sombras de penumbra variable, dependiendo de la distancia de la sombra al ocluidor, y del tamaño de la fuente de luz. Además la sombra puede recibir componentes de iluminación indirecta alterando su intensidad dependiendo de esta misma distancia. Por último, este método da pie a optimizaciones que reducen el número de accesos a textura necesarios para representar la sombra. Además, funciona completamente integrado en el *hardware* gráfico sin que se resienta la tasa de fotogramas por segundo, comparándolo con las técnicas actuales de PCF combinado con filtrado gaussiano. En la figura 1 puede observarse el resultado de la aplicación de nuestro método para la generación de una penumbra realista, nótese cómo la definición de la sombra disminuye según la distancia al ocluidor.

En la sección 2 se proporciona un estado del arte del campo de la representación de sombras mediante *shadow*

mapping y de los distintos tipos de filtrado que se emplean para reducir el dentado de este tipo de sombras. La sección 3 muestra detalladamente el algoritmo de sombreado utilizado. La sección 4 analiza los resultados obtenidos mediante la implementación de nuestra técnica. Finalmente, la sección 5 ofrece un resumen de las aportaciones realizadas en este documento desde un punto de vista crítico.

2. Trabajo previo

La generación de sombras en tiempo real es un tema que ha generado una gran cantidad de trabajos en la literatura, debido a que son de vital importancia en la representación de imágenes realistas. Además, la generación de sombras realistas en tiempo real es una tarea muy difícil de implementar en *hardware* gráfico. Por esa razón existe una gran cantidad de técnicas con muchas variantes que mejoran las sombras en uno u otro aspecto.

Como se ha comentado en la introducción, existen dos tipos de técnicas: las basadas en geometría (volúmenes de sombras), y las basadas en imágenes (mapas de sombras). Aunque los artículos basados en volúmenes de sombra son numerosos, este estado del arte no se extenderá en ellos ya que pertenecen a otro tipo de algoritmo de generación de sombras diferente al tratado en este artículo.

En 1977, F. Crow [Cro77] mostró cómo se podían generar volúmenes de sombra al extraer la silueta de un objeto desde el punto de vista de la luz. A este trabajo le han seguido otros que han mejorado esta técnica. T. Heidmann [Hei91] mostró cómo implementar el test de sombra por píxel en el *hardware* gráfico mediante el *stencil buffer*. Finalmente, Brabec y Seidel [BS03] desarrollaron una técnica para generar la silueta del objeto desde la luz por *hardware*, haciendo posible que todo el proceso se ejecutara internamente en la GPU.

Por otra parte, la primera referencia a la técnica de mapas de sombra la aporta Williams [Wil78] en 1978. Esta primera

versión del algoritmo proporcionaba un método de cálculo de sombras fácil de implementar y potencialmente compatible con el *hardware* gráfico, al estar basado en capturar el punto de vista de la luz para determinar los objetos que deben recibir la sombra. Sin embargo, como se ha comentado anteriormente, el método introduce varios problemas de dentado inherentes a la naturaleza discreta de las imágenes. A Partir de ese momento una gran cantidad de autores han aportado ideas para mejorar esta técnica básica para solucionar estos problemas.

Fernando et al. [FFBG01] proponen una técnica llamada Adaptive Shadow Maps para reducir este *aliasing* creando un mapa de sombras jerárquico, de forma que se refine en mayor o menor medida en las zonas con mayor detalle. El problema de esta técnica es que su naturaleza jerárquica la hace inadaptable al *hardware* gráfico. Arvo [Arv04] propone usar una estructura de rejilla para teselar el punto de vista de la luz, como una versión simplificada de [FFBG01]. Cada celda en la rejilla contiene una densidad de muestreo dado por un análisis heurístico.

Otros autores se decantan por alterar la perspectiva al capturar el mapa de sombras de forma que se dé más importancia en la textura a las partes más cercanas al observador, como un intento para ahorrar gran cantidad de memoria y reducir el dentado. En esta línea de trabajos se pueden encontrar los trabajos de Wimmer et al. [WSP04] [SDD03] [MT04]. Los autores en [ZSXL06] utilizan una filosofía similar con la diferencia de que se separa el rango de profundidad del observador en diferentes capas de menor precisión, aumentando la precisión en capas muy cercanas al punto de vista.

Por otra parte, otros autores se concentran en el tipo de filtrado que se le aplica al mapa de sombras para simular la penumbra (puntos que están parcialmente iluminados) y además para reducir el dentado. Reeves et al. [RSC87] presenta una técnica denominada Percentage-Closer Filtering (o PCF) que permite suavizar los bordes de la sombra y generar una cierta penumbra artificial. Hasenfratz et al. [HLHS03] hacen una recopilación sobre técnicas de representación de sombras con penumbra en tiempo real, utilizando las técnicas más importantes en este campo y ofreciendo una discusión de qué método es el más acertado para cada tipo escenas.

Otros autores han hecho hincapié en la representación de penumbras. Haines [Hai01] consigue una representación bastante realista de la penumbra, pero tiene importantes limitaciones: solo funciona para proyectar la sombra sobre un plano y que las sombras generadas siempre tendrán partes totalmente oscuras, sin importar su distancia al plano receptor. Además, se aprecian fallos en las partes de la penumbra con concavidades. Chan et al. [CD03] y Akenine-Möller et al. [AMA02] se basan en una extrusión de la silueta del objeto desde el punto de vista de la luz por lo que tiene las

mismas desventajas que los volúmenes de sombras en este aspecto.

Finalmente, Donelli et al. [DL06] introducen una nueva técnica que permite filtrar el mapa de sombras de la misma forma en que se filtraría cualquier mapa de color. Esto permite mejorar la calidad de las sombras reduciendo el dentado e incluso aplicando filtrados anisotrópicos, que dotan a las sombras de gran detalle, incluso en ángulos muy forzados.

3. Método

El método presentado en este documento aporta una nueva técnica para la representación de sombras suaves en tiempo real.

Existen técnicas como el *percentage closer filtering* (PCF) o el *variance shadow maps* (VSM) que permiten la representación de sombras con bordes difusos en tiempo real, que además, para disminuir el *aliasing* (de precisión de textura y de número discreto de escala de grises), utilizan un filtro gaussiano de desenfoque que opera a nivel de píxel que homogeneiza el aspecto de esta penumbra artificial.

El problema de estas técnicas es que la penumbra artificial generada no se comporta como lo haría en el mundo real, difuminando los bordes de la sombra de forma completamente homogénea. Tras simples observaciones del comportamiento de las sombras en entornos reales, es fácil llegar a la conclusión que la intensidad penumbra de la sombra en un punto dado depende de:

1. Tamaño de la fuente de luz (las luces puntuales no producen penumbra).
2. Distancia de la sombra al objeto que la proyecta.

Dicho de forma simple, la penumbra de una sombra será más fina y definida cuanto más cerca esté del ocluidor y cuanto menor sea la fuente de luz, y viceversa.

Nuestro método se basa en aplicar la técnica de mapa de sombras con un filtrado de tipo PCF combinado con un desenfoque gaussiano que varía la amplitud del suavizado por píxel, de forma que se aproxime una penumbra real de una forma muy rápida. Para ello se tendrá en cuenta tanto el tamaño de la fuente de luz como de la distancia a nivel de píxel del punto que recibe la sombra al punto que la proyecta. Para conocer esta distancia necesitamos introducir el concepto de **mapa de distancias**. Este mapa, del mismo tamaño que el área de visualización, se calculará después del mapa de sombras y permitirá almacenar para cada píxel de la pantalla, cuál es la distancia al ocluidor.

A continuación se detalla cada uno de los pasos necesarios del algoritmo.

3.1. Mapa de sombras "engordado"

Tras la captura del mapa de sombras tradicional, nuestro algoritmo genera lo que llamamos el mapa de sombras "engordado". Se trata del mapa de sombras correspondiente aplicar

una operación de desplazamiento por vértice en la dirección de la normal a todos los objetos que van a proyectar sombra y está capturado desde el mismo punto de vista que el mapa de sombras tradicional. Éste mapa nos servirá para delimitar el área donde potencialmente va a dibujarse la sombra en pantalla y servirá como base para calcular el mapa de distancias. Además, como se verá más adelante, también servirá para aplicar importantes optimizaciones a nivel de píxel en el cálculo de sombras, cuyos resultados pueden verse en las tablas de la sección de resultados.

En la práctica, la operación de engordado se puede implementar completamente por *hardware* en el *vertex shader*. Esta operación desplaza cada vértice en la dirección de su normal una distancia dada. Por lo tanto es necesario que se cumpla que todos los vértices que compartan la misma posición en el espacio tengan la misma dirección de engordado. De no cumplirse esta condición, aparecerán agujeros al realizar la operación. Una forma sencilla de solucionarlo, si las normales actuales del modelo no cumplen esta condición, es calcular los vectores normales correspondientes en un pre-proceso y pasarlos como atributo al *vertex shader* para que use éstos en vez de las normales de vértice.

Finalmente recalcar que usando el *geometry shader*, disponible como última tecnología en *hardware* gráfico, es posible calcular tanto el mapa de sombras tradicional como el engordado en la misma pasada de render, replicando y desplazando los vértices de la geometría y almacenando cada mapa en su *buffer* correspondiente.

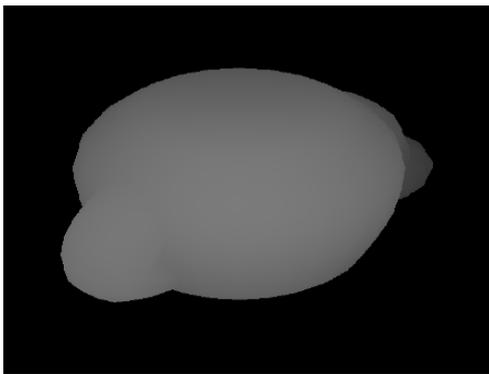


Figure 2: Ejemplo de mapa de sombras "engordado".

La figura 2 muestra el mapa de sombras desplazado necesario para generar la sombra de la figura 1. Nótese como la forma de la tetera aparece "engrosada" por haber aplicado el paso de desplazamiento de vértices en la dirección de la normal.

3.2. Mapa de distancias

El mapa de sombras "engordado" se utiliza como entrada para calcular el mapa de distancias. Este mapa se calcula

desde el punto de vista del observador con un tamaño equivalente al del área de dibujado actual.

Para calcular el mapa de distancias se dibujan los objetos que van a recibir sombras proyectando el mapa de sombras engordado desde la fuente de luz, tal y como si fuera un mapa de sombras tradicional. Para cada píxel se accede al valor del mapa de sombras engordado en ese punto y se compara con la profundidad del píxel actual desde el espacio de coordenadas de la luz que ha generado el mapa de sombras. El resultado es la distancia por píxel entre el objeto que recibe la sombra y el ocluidor que la ha generado.

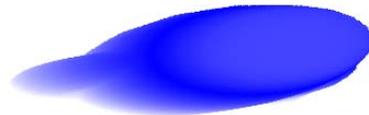


Figure 3: Ejemplo de mapa de distancias. Los canales rojo y verde guardan la distancia de la sombra al ocluidor. El canal azul guarda la máscara donde la sombra del objeto es válida.

La figura 3 muestra un ejemplo del mapa de distancias necesario para generar la sombra de la figura 1. Los canales rojo y verde guardan la distancia de la sombra al ocluidor. El canal azul guarda la máscara donde la sombra del objeto es válida. Un valor de 0 en el canal azul en un punto indica que el punto pertenece a la sombra (incluyendo la penumbra) del objeto, mientras que un 1 la invalida. El valor de los canales rojo y verde que están fuera de la máscara válida se ignoran. Nótese que los mapas de las figuras 2 y 3 no se corresponden píxel a píxel porque están generados desde diferentes puntos de vista: el mapa de sombras se genera desde el punto de vista de la fuente de luz, mientras que el mapa de distancias de hace desde el punto de vista del observador.

3.3. Modificación del PCF

Nuestro método se basa en el algoritmo del *percentage closer filtering* (PCF) para representar la sombra con penumbras a partir de un mapa de sombras. PCF realiza múltiples lecturas del mapa de sombras para conseguir bordes de la sombra suavizados. Cuantas más lecturas se realicen más niveles de gris podrán utilizarse para suavizar el borde de la sombra. Por ejemplo, un PCF de un tamaño de 3x3 permitirá representar la sombra con 9 variaciones de color, mientras que uno de 4x4 permitirá representarla con 16. Además,

estas muestras usan un factor de aleatoriedad por píxel para esconder patrones repetitivos y aliviar el dentado propio del mapa de sombras.

Nuestra aportación al PCF permite modificar el espaciado de las muestras realizadas al mapa de sombras a nivel de píxel, dependiendo de la distancia entre la sombra y el oclisor en ese punto, que viene dada por el mapa de distancias en ese punto. El usuario puede definir un parámetro que controla la velocidad con la que aumenta el espaciado entre las muestras con la distancia al oclisor. De esta forma se pueden definir sombras tanto de exteriores (generalmente más marcadas) como de interiores (con bordes más difusos dependiendo del tipo de iluminación).

Como se ha explicado anteriormente, el mapa de distancias acota la zona potencialmente susceptible de recibir una sombra. Por lo tanto se puede utilizar el mapa de distancias para no realizar ningún acceso al mapa de sombras en las zonas donde no se puede recibir sombra y optimizar accesos a textura desde el *pixel shader*.

El resultado de aplicar el PCF sobre todos los objetos en pantalla se almacena en una textura para ser utilizado en la siguiente fase del algoritmo: la fase de difuminado de bordes.

3.4. Difuminado de bordes

El algoritmo de PCF se suele combinar con una fase de difuminado o desenfocado de bordes, ya que utilizar directamente el resultado del PCF para sombrear una escena suele producir sombras con degradados de color discretos (más acusado cuanto menos muestras se realicen sobre el mapa de sombras) y con mucho ruido ya que se han utilizado muestras aleatorias.

Para solucionar estos problemas se añade una etapa de difuminado de la sombra utilizando un filtro de desenfoque gaussiano. Este filtro opera en dos dimensiones a nivel de píxel sobre las sombras representadas obteniendo múltiples muestras que son pesadas de acuerdo a una distribución gaussiana.

De igual forma que en el apartado anterior, el usuario puede definir un parámetro que controla la velocidad con la que aumenta el espaciado entre las muestras del filtro gaussiano en función de la distancia al oclisor, para configurar las sombras al tipo de iluminación deseada. Aunque pueda configurarse este valor manualmente, generalmente vendrá determinado por el tamaño de la luz que proyecta las sombras. Cuanto mayor sea la luz, mayor será este valor, y con mayor velocidad crecerá la penumbra.

Nuestra aportación en este apartado es la de utilizar el mapa de distancias para determinar la amplitud del filtro de suavizado, es decir, la distancia que separa las muestras entre sí. De esta forma, las partes de la sombra que estén más cerca del oclisor usarán una amplitud del filtro muy baja,



Figure 4: Mapa de sombras después de la aplicación del filtro gaussiano.

causando así bordes de sombra muy definidos y precisos. De igual forma, las partes de la sombra que estén más alejadas del oclisor usarán un filtro con muestras muy separadas espacialmente, lo que repercutirá en una sombra con bordes mucho más difusos en esos puntos, tal y como puede observarse en la realidad.

La figura 4 muestra el resultado de aplicar el efecto del filtro gaussiano de desenfoque sobre el mapa de sombras calculado mediante el PCF. En la figura se puede ver como las zonas más alejadas del oclisor se desenfocan más que las más cercanas.

3.5. Aportación de la iluminación indirecta a la sombra

Las sombras obtenidas por los métodos tradicionales basados en mapas de sombras provienen de la aportación principal de la escena definidas por las fuentes de luz explícitas. Sin embargo, una iluminación más realista tendría en cuenta la aportación proveniente de la iluminación indirecta. En el caso de escenas de interior, esta iluminación indirecta proviene de otros objetos de la escena que reflejan cierta cantidad de luz. En escenas de exterior, este tipo de iluminación proviene principalmente de la luz reflejada en la atmósfera y en las nubes.

Esta iluminación también afectará a la intensidad de la sombra dependiendo de la cantidad de luz indirecta que llegue a un punto dado de la sombra. La cantidad de iluminación indirecta en un punto dado será potencialmente mayor cuanto más alejado esté del oclisor. Por lo tanto, cuanto más cerca esté la sombra de éste, menor cantidad de luz indirecta va a recibir.

Igual que en las etapas anteriores, el usuario puede definir un parámetro que permite variar la velocidad con la que au-

menta la aportación de este tipo de iluminación sobre la sombra a medida que se aleja del ocluser.

Para implementar este comportamiento en nuestro esquema utilizamos el mapa de distancias para aclarar la sombra cuanto más alejada esté del ocluser. La cantidad de aclarado que recibe la sombra se pesa por un valor definido por el usuario. De esta forma se pueden simular desde sombras en escenas de interior con mucha iluminación indirecta con iluminación difusas que producen sombras muy poco definidas a sombras de escenas de interior muy definidas.

3.6. Paso final

Después del proceso de difuminado de bordes, las sombras resultantes están almacenadas en una textura cuyo contenido representa la intensidad de la sombra en cada punto. El último paso del algoritmo es aplicar sobre la escena las sombras calculadas. Esto se realiza mediante una última pasada de *render* donde la geometría susceptible de recibir sombras accede a la textura de sombras y pesa la intensidad de la iluminación por píxel dependiendo de la sombra calculada para ese punto.

4. Resultados

Las pruebas se han realizado en un equipo Athlon +3500 equipado con 1 Gb de RAM y una tarjeta gráfica nVidia GeForce 8800 GT. Se ha utilizado un tamaño de PCF de 3x3 y para el paso de difuminado de bordes se ha usado un filtro de tamaño 11. Éste se ha programado como un filtro separable de forma que no se necesitan 121 accesos sino dos pasadas (horizontal y vertical) de 11 accesos cada uno. Esto se realiza una sola vez al terminar cada fotograma.

La figura 5 muestra el resultado de la aplicación de nuestro método sobre una malla típica en videojuegos. La figura muestra además la diferencia de calidad entre nuestro método y la aplicación de un PCF tradicional con una fase de desenfoque gaussiano. Nótese cómo nuestro método simula mejor el comportamiento de la luz al representar la sombra, haciendo los bordes de la sombra más precisos y definidos en las partes más cercanas al ocluser (la sombra del pie en la figura) y más desenfocados en las más alejadas.

La figura 6 muestra como varía el grosor y la definición de la penumbra dependiendo de la distancia de la sombra al ocluser. En la figura se muestran tres fotogramas de la escena con una luz en movimiento que genera sombras más largas o más cortas, con penumbras más definidas o más borrosas, dependiendo del ángulo de incidencia de la luz.

Las tablas 1 y 2 muestran el rendimiento obtenido mediante la aplicación de esta técnica, comparado con el rendimiento obtenido con la técnica tradicional de PCF con filtro gaussiano. Se han medido los tiempos con diferentes resoluciones de pantalla, ya que las técnicas de mapas de

Resolución	800x600		1280x1024	
	nuestro	tradicional	nuestro	tradicional
FPS	217	212	135	134
	218	215	110	112
	230	224	116	115

Table 1: Tabla de rendimientos, medidos en fotogramas por segundo (FPS), tomados desde tres puntos de vista diferentes. Escena de la tetera.

Resolución	800x600		1280x1024	
	nuestro	tradicional	nuestro	tradicional
FPS	215	224	124	128
	218	230	120	124
	203	211	130	125

Table 2: Tabla de rendimientos, medidos en fotogramas por segundo (FPS), tomados desde tres puntos de vista diferentes. Escena del personaje.

sombras de este tipo son muy sensibles a los cambios de resolución, ya que realizan muchos accesos a textura por cada píxel en pantalla. Las técnicas utilizadas son las mismas que las usadas para obtener las imágenes de comparación de la figura 5.

En las tablas puede verse que los resultados de rendimiento de nuestra técnica son muy parecidos a los obtenidos con una técnica tradicional de PCF con filtro gaussiano. Esto es debido a que, como se comenta más adelante en la sección de conclusiones, el sobrecoste añadido por píxel no representa una gran sobrecarga, siendo el proceso añadido más pesado la nueva fase de cálculo del mapa de sombras desplazado. Sin embargo, gracias a las optimizaciones realizadas mediante el mapa de distancias (ver apartado de conclusiones), esta nueva etapa no causa un gran impacto sobre el rendimiento final de la aplicación.

Tras estudiar estos resultados concluimos que nuestra técnica permite la representación de sombras más realistas que con el método de PCF con filtrado gaussiano ofreciendo un rendimiento parejo, por lo que la convierte en una buena opción para la generación de penumbras más realistas.

5. Conclusiones

El método presentado en este artículo aporta una nueva técnica de cálculo de penumbra en tiempo real usando mapas de sombra. La técnica se basa en la generación de un mapa de distancia que permite variar la distancia entre muestras del filtro gaussiano de desenfoque aplicado como post proceso. Esto permite variar la distancia entre muestras a nivel de píxel, dependiendo de la distancia de la sombra al ocluser que la ha generado. Además, el mapa de distancias también se usa para determinar la intensidad de la sombra simulando así la iluminación recibida de rayos de segundo orden.

Figure 5: Esta figura muestra la diferencia entre nuestro método (a) y el PCF con desenfoque gaussiano tradicional (b). Nótese cómo la sombra es más definida cuánto más cerca está el ocluidor.



(a) Ejemplo de sombras con nuestro método.



(b) Ejemplo de sombras con PCF tradicional.



(c) Detalle de sombra con nuestro método.



(d) Detalle de sombra con PCF tradicional.

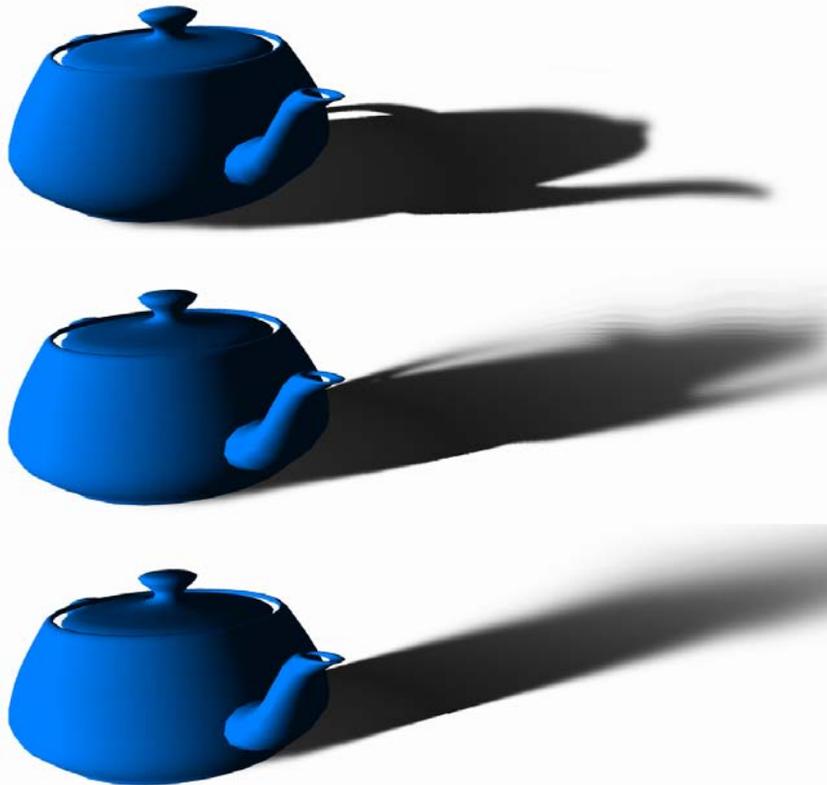
La nueva técnica ofrece representación de sombras con penumbra en tiempo real ofreciendo una buena tasa de fotogramas por segundo. A continuación se analiza como puede afectar cada etapa de nuestro método al rendimiento, comparándolo con una implementación estándar de PCF con una etapa de desenfoque gaussiano:

- **Fase de cálculo del mapa de sombras "engordado".** Esta fase requiere dibujar la geometría que arroja sombras. El engrosamiento de la geometría se realiza en el *vertex shader* emdiante una operación trivial y el *pixel shader* es el más trivial posible (una sola instrucción). El mayor sobre coste de esta etapa es el de volver a dibujar la geometría que ya se ha usado para calcular el mapa de sombras original en una pasada de *render* extra. Sin embargo, esta etapa podría eliminarse utilizando el *geometry shader* para duplicar la geometría y engordar una de las dos versiones, de forma que cada una se volcara en una

textura diferente (haciendo uso de la capacidad de *Multiple Render Targets* de las GPU). De esta forma tanto el mapa de normales como su versión engrosada podrían calcularse en una única pasada de *render*.

- **Fase de cálculo del mapa de distancias.** Esta fase dibuja la geometría receptora de sombras para calcular el mapa de distancias. A nivel computacional, los *shaders* utilizados en esta nueva etapa son muy simples y no representan una gran carga computacional comparándolos con los utilizados para cálculos de iluminación, *skinning* o aplicación de sombras comúnmente utilizados en aplicaciones en tiempo real como videojuegos.
- **Fase de aplicación del mapa de sombras con PCF.** Un acceso a textura, dos multiplicaciones y una suma extra en la fase de aplicación del mapa de sombras con PCF. La aplicación de un PCF de 3x3 sobre un objeto requiere de 9 accesos a textura, siendo 16 los accesos necesarios en el

Figure 6: Ejemplo de penumbra adaptativa dependiendo de la distancia de la sombra al ocluidor ante una luz en movimiento.



caso de usar un PCF de 4x4. Así pues, un acceso a textura a textura es irrelevante en este caso.

- **Fase de aplicación del filtro gaussiano.** La aplicación de un filtro gaussiano con un *kernel* de tamaño 11 usado comúnmente para difuminar los bordes de las sombras uniformemente requiere dos pasadas de render donde cada píxel realiza 11 accesos a textura en cada paso. Nuestro método añade simplemente un acceso a textura y una operación de multiplicación en cada paso.

Al poder utilizarse el mapa de distancias como una máscara para determinar que zonas están dentro de la sombra y cuales no, se consigue una mayor optimización de accesos al mapa de sombras necesarios que con el algoritmo de PCF tradicional. Por esta razón, en la mayor parte de los casos, y a no ser que las sombras ocupen gran parte de la pantalla, nuestra técnica es tan rápida como el PCF tradicional, como también se ha mostrado en la sección de resultados.

Una opción para mejorar el rendimiento de la aplicación a costa de perder un poco de calidad es eliminar la etapa del PCF y representar la sombra utilizando el algoritmo original de *shadow mapping* [Wil78]. De esta forma se reduce el gran número de accesos al mapa de sombras necesarios

por el PCF a uno solo, dejando la tarea de generación de la penumbra al filtro gaussiano. Nuestro método permite también eliminar esta etapa, permitiendo también optimizar el *pipeline* de la misma forma.

Como trabajo futuro se propone el estudio de la utilización de un filtro gaussiano en el espacio de coordenadas del mapa de sombras, no de la pantalla. De esta forma podría evitarse el efecto de *ghosting* asociado a la aplicación de un filtro en el espacio de coordenadas de la pantalla. Además, cabría la posibilidad de estudiar como repercute la utilización del *geometry shader* para mejorar la etapa de cálculo de los mapas de sombras. Otra forma de mejorar la calidad de las penumbras sería adaptar el número de muestras realizadas en función del tamaño de la distancia al ocluidor, lo que ayudaría a aliviar los efectos de patrones en las sombras. De esta forma se conseguirían penumbras con degradados de color más suaves, incluso en sombras muy forzadas.

Agradecimientos

Este trabajo ha sido financiado por la Universidad Jaume I (PREDOC/2006/54), el gobierno español (TIN2005-08863-

C03-03, TIN2004-07451-C03-03) y la fundación Bancaixa (IST-2-004363).

References

- [AMA02] AKENINE-MÖLLER T., ASSARSSON U.: Approximate soft shadows on arbitrary surfaces using penumbra wedges. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2002), Eurographics Association, pp. 297–306.
- [Arv04] ARVO J.: Tiled shadow maps. In *Proceedings of Computer Graphics International 2004* (2004), pp. 240–247.
- [BS03] BRABEC S., SEIDEL H.: Shadow volumes on programmable graphics hardware. *Computer Graphics Forum (Eurographics 2003)*, 25(3), September 2003. 8 (2003).
- [CD03] CHAN E., DURAND F.: Rendering fake soft shadows with smoothies. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 208–218.
- [Cro77] CROW F. C.: Shadow algorithms for computer graphics. In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1977), ACM, pp. 242–248.
- [DL06] DONNELLY W., LAURITZEN A.: Variance shadow maps. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), ACM, pp. 161–165.
- [FFBG01] FERNANDO R., FERNANDEZ S., BALA K., GREENBERG D.: Adaptive shadow maps. In *SIGGRAPH '01* (New York, USA, 2001), pp. 387–390.
- [Hai01] HAINES E.: Soft planar shadows using plateaus. *J. Graph. Tools* 6, 1 (2001), 19–27.
- [Hei91] HEIDMANN T.: Real shadows real time. 28–31.
- [HLHS03] HASENFRATZ J.-M., LAPIERRE M., HOLZSCHUCH N., SILLION F.: A survey of real-time soft shadows algorithms. *Computer Graphics Forum* 22, 4 (dec 2003), 753–774.
- [MT04] MARTIN T., TAN T.: Anti-aliasing and continuity with trapezoidal shadow maps. In *Rendering Techniques* (2004), pp. 153–160.
- [RSC87] REEVES W. T., SALESIN D. H., COOK R. L.: Rendering antialiased shadows with depth maps. *SIGGRAPH '87* 21, 4 (July 1987), 283–291.
- [SD02] STAMMINGER M., DRETTAKIS G.: Perspective shadow maps. *ACM Transactions on Graphics*, 21(3):557–562, July 2002. (2002).
- [SDD03] STAMMINGER M., DRETTAKIS G., DACHSBACHER C.: Perspective shadow maps. In *Game Programming Gems IV*. 2003.
- [Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *SIGGRAPH '78* (New York, USA, 1978), pp. 270–274.
- [WSP04] WIMMER M., SCHERZER D., PURGATHOFER W.: Light space perspective shadow maps. In *Rendering Techniques 2004 (Proceedings Eurographics Symposium on Rendering)* (June 2004), Keller A., Jensen H. W., (Eds.), Eurographics, pp. 143–151.
- [ZSXL06] ZHANG F., SUN H., XU L., LUN L. K.: Parallel-split shadow maps for large-scale virtual environments. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications* (New York, NY, USA, 2006), ACM, pp. 311–318.