

Screen Space Obscurances

Nicolau Sunyer¹ and Jesus Gumbau² and Miguel Chover² and Mateu Sbert¹

¹Graphics and Imaging Laboratory, Instituto de Informática y Aplicaciones, Universitat de Girona

²Computer Graphics Group, Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, Castellón

Abstract

La simulación de la iluminación indirecta en una escena suele ser muy costosa por lo que normalmente ésta se usa de forma precalculada para escenas estáticas. En este artículo proponemos un algoritmo que aproxima el cálculo de la iluminación indirecta enteramente en GPU y permite escenas dinámicas pues se recalcula para cada fotograma. La solución propuesta se basa en la técnica de Obscurances y permite simular el efecto de sangrado de color (color-bleeding). Al estar el muestreo basado en espacio de imagen no se usan ni reproyecciones ni transformaciones, simplificando los calculos. Nuestra solución está formada por la fusión de tres mapas diferentes de Obscurances, de baja, media y alta frecuencia, que permiten simular los distintos niveles de interacción de la iluminación indirecta, resaltando los detalles sin sacrificar la interacción con los ocluidores a distancia media y lejana. El sistema no requiere del uso de filtrado del mapa de Obscurances pues se ha optimizado el muestreo de la escena y se obtienen imágenes con muy poco ruido usando pocas muestras por píxel.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introducción

El algoritmo planteado en esta publicación intenta generar la visualización más realista posible de escenas tridimensionales pero manteniendo la interactividad. Está basado en el cálculo de obscurancias (Obscurances) [ZIK98] [IKSZ03] generado en espacio de imagen y usando técnicas de GPGPU, por lo que se puede implementar en el hardware gráfico de forma eficiente. Estos algoritmos generan un mapa de luces llamado mapa de obscurancias que aproxima la iluminación indirecta de una escena aumentando el realismo en la visualización de ésta (ver Figura 1).

Las obscurancias son una variante de los métodos llamados de oclusión ambiental (Ambient Occlusion) [Lan02] [Bun05] [KA06] pero, a diferencia de estos, permiten simular el sangrado de color (color-bleeding) [MSC03] [MFS09], normalmente asociados a gráficos pre-calculados. Al usar una técnica en espacio de imagen sólo calculamos el valor de obscurancias para los elementos visibles de la escena permitiendo un dinamismo total en todos los elementos de la escena pues el mapa de obscurancias se vuelve a calcular para cada fotograma de la visualización. El cálculo de las oclusiones se ha dividido en tres frecuencias en función del tipo de oclusiones que queremos detectar en cada una de ellas.

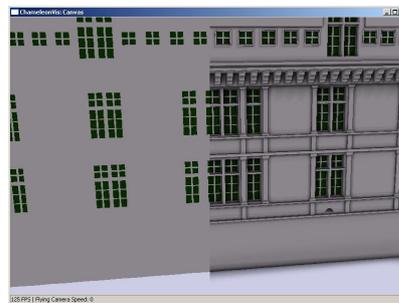


Figura 1: Ejemplo de obscurancias aplicadas al modelo pared. Nótese la sensación de volumen que se transmite y el realismo que aportan a la escena. Resultado obtenido con nuestra técnica basada en espacio de imagen.

Las altas frecuencias detectan oclusiones por los cambios de relieve en las superficies de la escena así como las esquinas. Las frecuencias medias detectan las oclusiones por los elementos más grandes de un objeto. Finalmente las bajas frecuencias detectan las interacciones entre objetos de la escena o elementos muy grandes de un objeto complejo. El

mapa de obscurancias final es el resultado de combinar estas tres capas.

Este artículo presenta la siguiente estructura. La sección 2 resume el trabajo previo en obscurancias y oclusión ambiental en espacio de imagen por GPU. La sección 3 explica el método propuesto y sus distintas partes. Posteriormente, la sección 4 muestra los resultados obtenidos tanto en rendimiento como en calidad visual. Finalmente la sección 5 resume las aportaciones de nuestro trabajo y esboza líneas de trabajo futuro.

2. Trabajo previo

Los métodos de aproximación de iluminación indirecta en espacio de imagen por GPU se basan normalmente en oclusión ambiente [SA07] [Qui] [BSD08] [RGS09]. Muchas de ellas se utilizan en juegos comerciales [Mit07] [FM08].

Normalmente se basan en la generación de muestras aleatorias en espacio de cámara, la reproyección de estas muestras para obtener las coordenadas que les correspondería en espacio de imagen y finalmente en el muestreo de las texturas generadas en el primer paso con estas coordenadas para obtener los valores de la muestra. Estos métodos funcionan bien con las oclusiones de alta frecuencia pero no permiten la detección de oclusiones de baja frecuencia. La mayoría de publicaciones proponen el uso de otros métodos no basados en imagen para la computación de las oclusiones de baja frecuencia [BSD08] [SA07]. Nosotros proponemos un método que nos permite usar el mismo algoritmo para generar cada uno de estos tipos de oclusiones. También hemos evitado usar reproyecciones y tomamos las muestras simplemente entre los vecinos, en espacio de imagen, del punto a iluminar.

3. Metodo

Las técnicas basadas en espacio de imagen se basan en el uso de iluminación diferida (deferred lighting) [DWS*88] (ver figura 2). En ella la visualización de la escena se realiza en dos pasos. En un primer paso se visualiza la geometría una vez, desde el punto de vista del observador, para generar una serie de texturas que contienen información como la posición, normal, color difuso o demás datos para cada punto visible de la escena. Todas estas texturas se generan en un solo paso utilizando una técnica llamada Multiple Render Targets (MRT) en inglés. En un segundo paso, se procesan estas texturas y se computa el valor de obscurancia únicamente para aquellos puntos visibles para el observador (que son los contenidos en las texturas).

A partir de ahora la explicación se va a referir a cómo calculamos el valor de uno de esos píxeles de la imagen final. Para empezar, sólo calculamos la obscurancia para los píxeles que son válidos (no son parte del fondo de la imagen). Dado un fragmento válido P con una posición P_p y

una normal P_n en espacio de cámara aplicamos el siguiente algoritmo:

- Muestreo: Obtenemos muestras de posibles ocluidores alrededor del fragmento P . Tomamos estas muestras de forma aleatoria en espacio de imagen. Cada una de estas muestras contendrá la posición y normal del posible ocluidor, así como su color difuso si queremos habilitar sangrado de color.
- Criterio de aceptación: Para cada muestra, se determina si es válida. Para hacerlo, tenemos en cuenta la posición del posible ocluidor respecto a la normal P_n de la superficie en el punto a iluminar. Sólo aceptamos muestras que estén desviadas un ángulo respecto a la normal menor a un ángulo límite y que estén a una distancia menor a d_{max} del punto a iluminar. Variando este ángulo límite y la distancia de aceptación d_{max} obtenemos los distintos tipos de oclusiones. Aquí cabría indicar que se deben de usar normales planas y no suavizadas pues pueden inducir a aceptar falsas oclusiones. Se puede usar un geometry shader para generar las normales planas al vuelo a partir de la geometría al procesar esta.
- Función de transferencia: Para el conjunto de muestras válidas, se calcula el valor de obscurancia que aportan al valor final, que se aplicará sobre el fragmento P . Esta función se corresponde con \sqrt{dist}/d_{max} en nuestro caso pero se puede usar una función distinta.
- Cómputo final: Es un proceso que toma el valor acumulado por las aportaciones de las distintas muestras tomadas para el fragmento P . El valor final de obscurancias se calcula como la media de estas aportaciones.

Los valores generados por cada muestra válida están formados por cuatro canales de información, donde los tres primeros contienen la información de color y el cuarto canal contiene el número de muestras válidas que se han usado para obtener este color.

El color final tiene en cuenta tanto el valor de oclusión como la cantidad de sangrado a aplicar sobre la muestra P . Al final, se divide la suma de todas las muestras acumuladas por el divisor, siempre que éste no sea cero.

4. Resultados

Los siguientes resultados se han obtenido en un ordenador con un procesador Intel Quad-core a 2.4GHz con 4GB de RAM usando Windows XP Profesional. La GPU usada es una GeForce GTX280 con 1GB de RAM. La resolución de las imágenes es de 800x600 píxeles.

En la Figura 3 mostramos un modelo de un volquete compuesto por 197,843 triángulos. Mostramos el mapa final de obscurancias, los tres mapas por frecuencias que lo componen, así como la combinación del mapa final con el color difuso. La tasa de refresco para el modelo final está en el rango de 13-15 fps permitiendo interactividad.

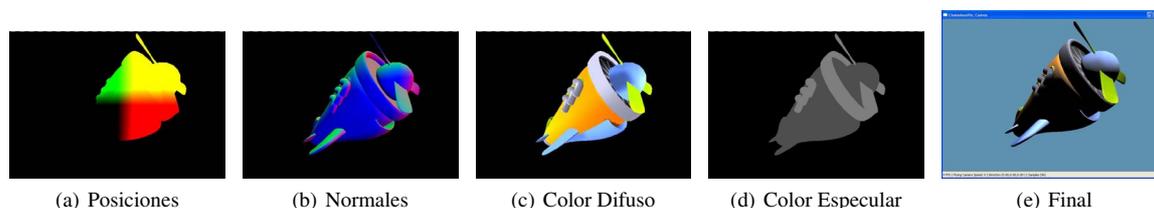


Figura 2: Diferentes pasos del deferred shading.

La Figura 4 contiene el castillo Hikone que está compuesto de 379,542 triángulos. Aquí la tasa de refresco es la misma mostrando que el método es bastante independiente de la complejidad de la escena.

La Figura 5 muestra una vista de cerca de la escena de la ciudad mostrando el sangrado de color y su efecto en el resultado final. Observar que la escena va más lenta aunque muestra mucho menos geometría porque la velocidad de refresco viene determinada por la cantidad de pantalla que contiene geometría, siendo éste un ejemplo del caso peor.

5. Conclusiones

En esta publicación proponemos un método para aproximar la iluminación indirecta de una escena usando obscurancias en la tarjeta gráfica. El método propuesto tiene en cuenta la distancia a la que están los ocluidores así como su color. También se propone una forma de calcular las distintas frecuencias de oclusión de forma separada pero usando el mismo algoritmo. Este método no usa reproyección. El muestreo se ha optimizado dependiendo de la distancia de cada punto de la imagen a la cámara y de la frecuencia de las oclusiones que queremos capturar. Finalmente, se propone un sistema que usa geometry shaders para eliminar problemas con superficies curvadas. El sistema pretende obtener la mayor calidad con el menor número de muestras. Para ello no se ha implementado ningún sistema de filtrado del mapa de obscurancias. Un algoritmo de filtrado podría ayudar a reducir el número de muestras necesarias para tener un buen resultado, mejorando el rendimiento del sistema. Al disponer de los mapas separados por frecuencias, podríamos aplicar un filtrado distinto a cada frecuencia evitando así suavizar demasiado las altas frecuencias como ocurre en los métodos usados habitualmente. Los números aleatorios para generar las muestras se leen actualmente de textura pero se podrían calcular al vuelo en la GPU. Finalmente se podría usar la coherencia temporal de los mapas de obscurancias para reducir aún más el número de muestras a tomar.

6. Agradecimientos

Este trabajo ha sido subvencionado en parte por el MCT (proyectos TIN2007-68066-C04-01 y TIN2007-68066-C04-02).

References

- [BSD08] BAVOIL L., SAINZ M., DIMITROV R.: Image-space horizon-based ambient occlusion. In *SIGGRAPH '08: ACM SIGGRAPH 2008 talks* (New York, NY, USA, 2008), ACM, pp. 1–1. 2
- [Bun05] BUNNELL M.: *GPU Gems 2*. Addison-Wesley Publishers., 2005, pp. 223–233. 1
- [DWS*88] DEERING M., WINNER S., SCHEDIWY B., DUFFY C., HUNT N.: The triangle processor and normal vector shader: a vlsi system for high performance graphics. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1988), ACM, pp. 21–30. 2
- [FM08] FILION D., MCNAUGHTON R.: Effects & techniques. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes* (New York, NY, USA, 2008), ACM, pp. 133–164. 2
- [IKSZ03] IONES A., KRUPKIN A., SBERT M., ZHUKOV S.: Fast, realistic lighting for video games. *IEEE Comput. Graph. Appl.* 23, 3 (2003), 54–64. 1
- [KA06] KONTKANEN J., AILA T.: Ambient occlusion for animated characters. In *Rendering Techniques 2006 (Eurographics Symposium on Rendering)* (jun 2006), Wolfgang Heidrich T. A.-M., (Ed.), Eurographics. 1
- [Lan02] LANDIS H.: *Production-ready global illumination*. Tech. rep., Technical report, SIGGRAPH Course notes 16, 2002. 1
- [MFS09] MÉNDEZ-FELIU L., SBERT M.: From obscurances to ambient occlusion: A survey. *Vis. Comput.* 25, 2 (2009), 181–196. 1
- [Mit07] MITTRING M.: Finding next gen: Cryengine 2. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (New York, NY, USA, 2007), ACM, pp. 97–121. 2
- [MSC03] MÉNDEZ A., SBERT M., CATÁ J.: Real-time obscurances with color bleeding. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics* (New York, NY, USA, 2003), ACM, pp. 171–176. 1
- [Qui] QUILEZ I.: Screen-space ambient occlusion. <http://iquilezles.org/www/articles/ssao/ssao.htm>. 2
- [RGS09] RITSCHER T., GROSCH T., SEIDEL H.-P.: Approximating dynamic global illumination in image space. In *ISD '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2009), ACM, pp. 75–82. 2
- [SA07] SHANMUGAM P., ARIKAN O.: Hardware accelerated ambient occlusion techniques on gpus. In *ISD '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2007), ACM, pp. 73–80. 2
- [ZIK98] ZHUKOV S., IONES A., KRONIN G.: An ambient light illumination model. In *Rendering Techniques* (1998), Drettakis G., Max N. L., (Eds.), Springer, pp. 45–56. 1

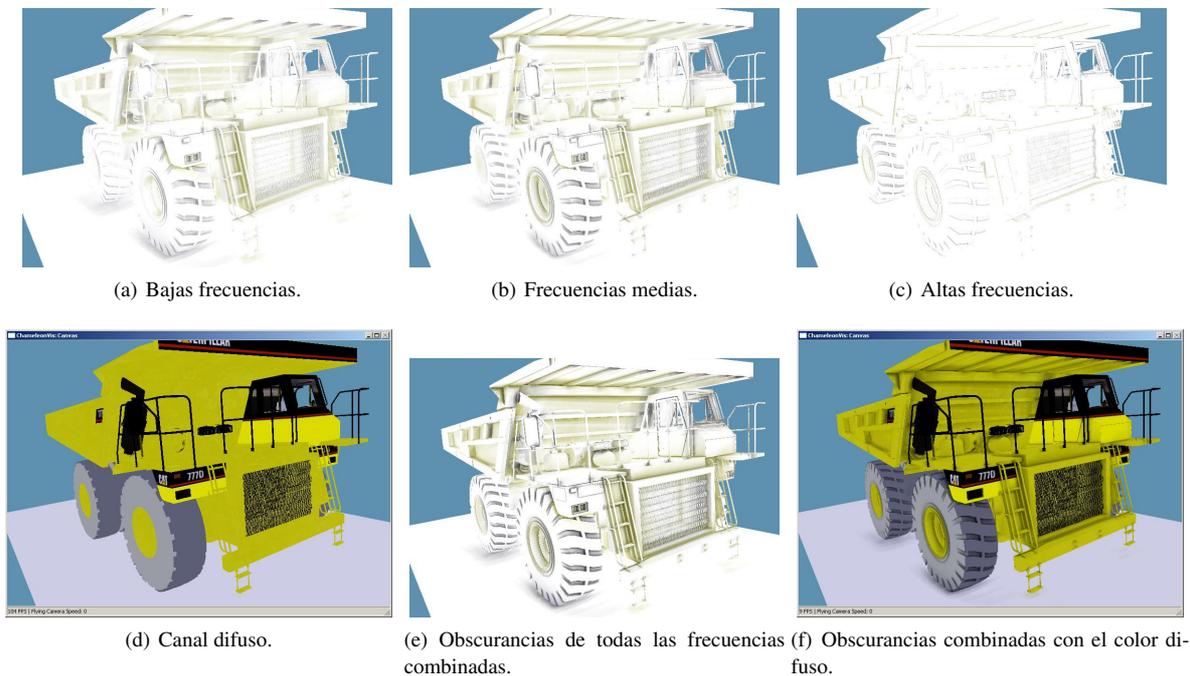


Figura 3: Obscurancias de diferentes frecuencias (bajas, medias y altas) y resultado final al combinar todas las frecuencias con el color difuso.

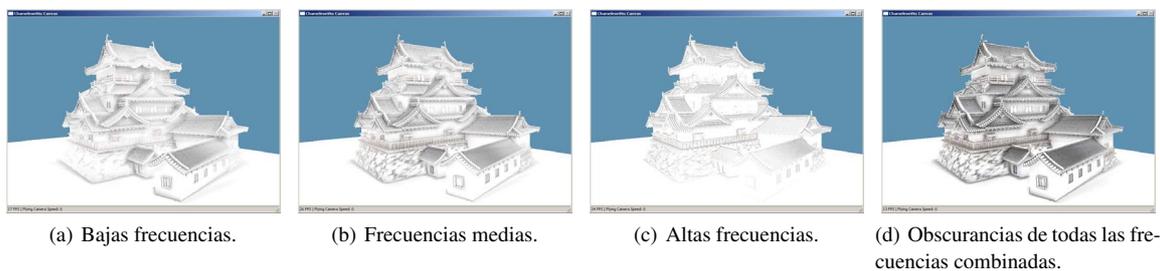


Figura 4: Resultados de obscurancias de diferentes frecuencias (bajas, medias y altas) y resultado final usando la escena del castillo de Hikone.

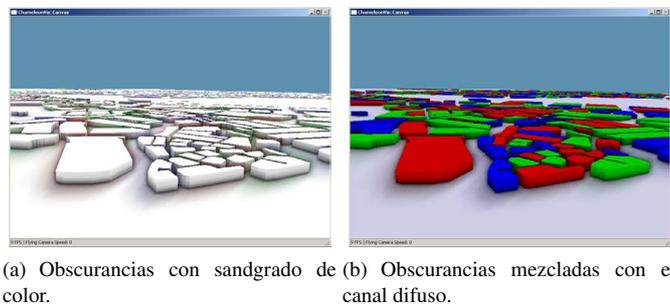


Figura 5: Mapa de obscurancias con sandgrado de color (a) y su efecto en el resultado final (b).