

OSG Visibility Groups

Renato Leriche Vázquez

Franco Serrano Victor Hugo

UNAM, DGSCA, Departamento de Realidad Virtual

24 de abril de 2009

rleriche@unam.mx

vhfranco@unam.mx

1. Áreas de aplicación y Palabras clave

Ambientes virtuales, Optimización,

2. Índice del reporte.

1. Áreas de aplicación y Palabras clave.....	1
2. Índice del reporte.....	1
3. Resumen (abstract, summary).....	1
4. Introducción.....	2
5. Marco teórico.....	2
6. Parte Experimental - Método.....	3
7. Resultados y discusión de Resultados.....	8
8. Conclusiones y Recomendaciones.....	8
9. Bibliografía.....	9

3. Resumen (abstract, summary).

En la búsqueda de métodos de optimización geométrica y de cámara se encontró la necesidad de carga y descarga de modelos según la ubicación de la cámara, esto con el fin de optimizar geometrías interiores de gran extensión y que estuvieran divididas en cuartos.

Las pruebas de exportación llevaron a los siguientes resultados, al exportar el rayo de intersección mide una unidad, lo que no se ajusta a todos los escenarios y debe ser ajustado en cada escenario, exportar en IVE (Binario) no en OSG (Ascii) puesto que la Versión del OpenSceneGraph Max exporter 0.9.5 no lo permite, una vez exportado a OSG se debe pasar por un programa (en desarrollo) que corrija el rayo de intersección.

4. Introducción.

La aplicación directa de esta investigación beneficiará al proyecto IXTLI de temática Mercado Abelardo L. Rodríguez¹ pues la extensión de este mercado es amplia y compleja, por lo que deberán ser usadas técnicas como visibility groups, shader y campos de alturas para optimizar el despliegue en pantalla.



5. Marco teórico.

Los OSG Visibility Groups son ayudantes usados para definir zonas de carga dentro de las aplicaciones tridimensionales que pueden ser usadas para aminorar la carga de modelos.

Los OSG Visibility Groups permiten asignarle un objeto a un grupo de objetos geométricos, que funciona como envoltorio y permite a la cámara que se encuentre dentro cargar los objetos que se decidan que se carguen y que cuando la cámara se encuentre fuera desaparezcan.

¹ Mercado Abelardo L. Rodríguez (Milenio Online, 2009) (Gómez Flores, 2008) (Cordero, 2008)

Dentro del algoritmo que resuelve el OSG Visibility Group existe un rayo que nace desde la cámara en dirección del vector del punto de vista, que se extiende tan largo como un valor numérico asignado, si este rayo intersecta una cara del objeto envolvente, en dirección de la normal, los objetos del grupo de visibilidad son desplegados.

6. Parte Experimental - Método.

Solución 1:

Programa que construía geometrías y realizaba una navegación que mostraba los visibility groups.

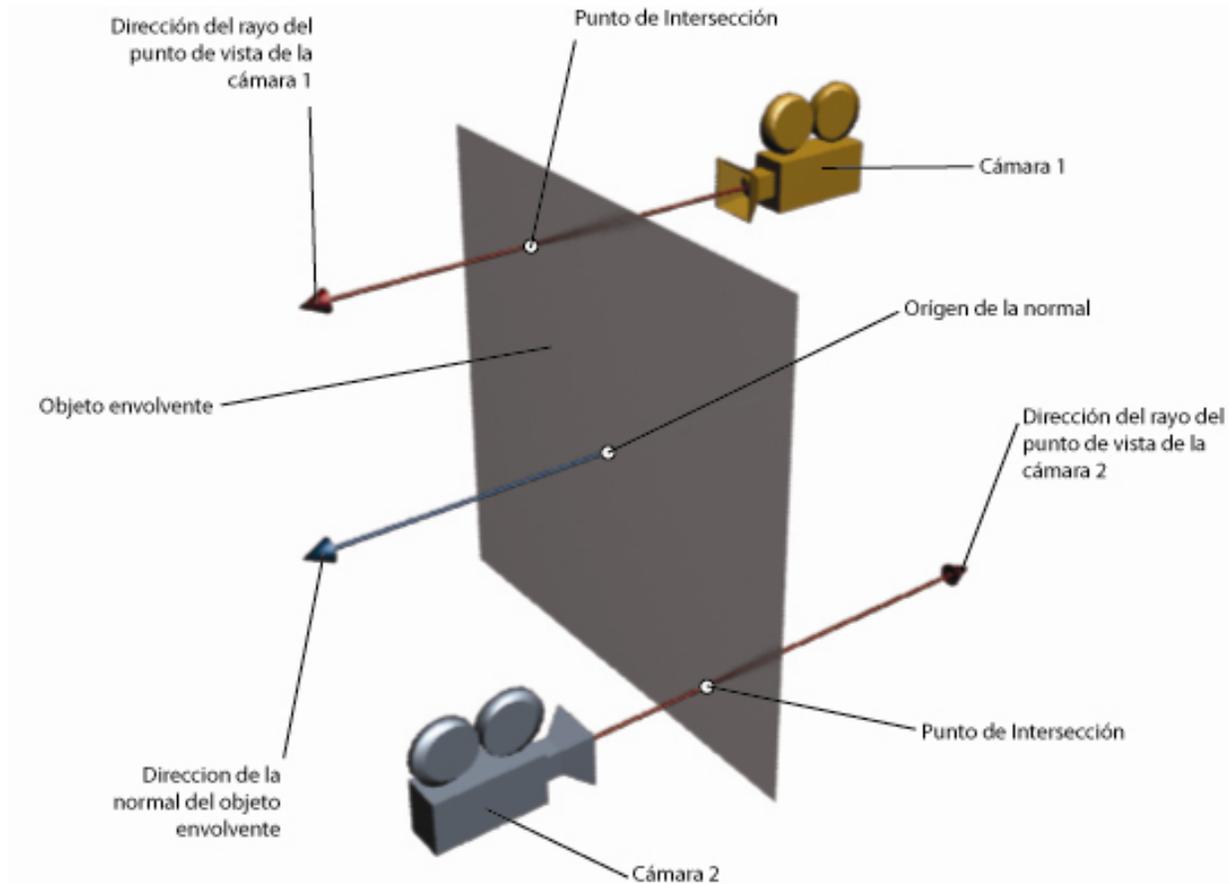
Solución 2:

Exportación directa de Visibility Groups desde 3D Studio Max

1.- Creación de escenario usable para visibility groups

Para la creación del escenario es necesario tomar en cuenta el cómo deben crearse las geometrías, uno de los aspectos más importantes es la dirección en la que las normales son creadas, por ende es necesario el análisis de ellas, otro de los aspectos son los elementos que aparecerán dentro del visibility group.

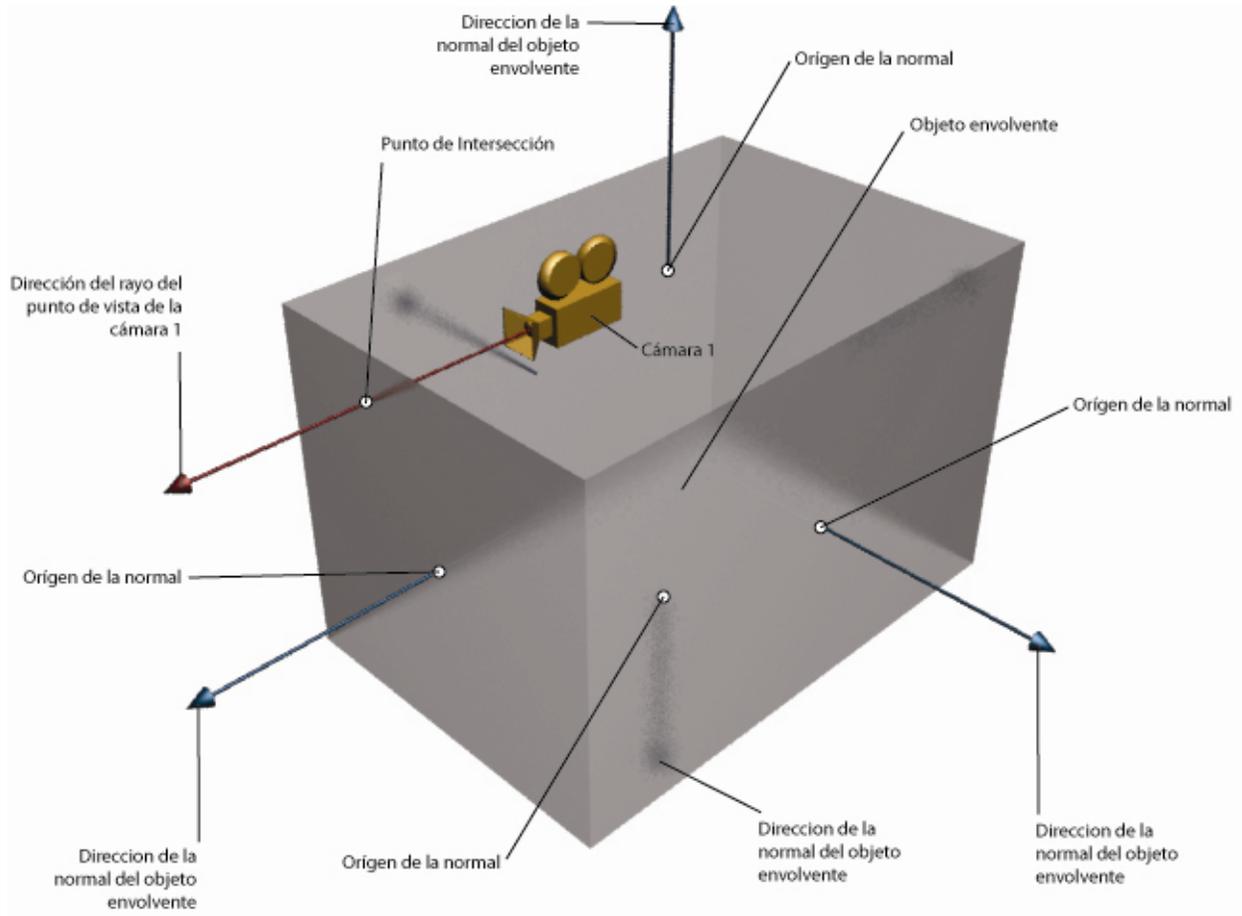
1a.- Orientación de normales



La orientación de la normal debe ser la misma que la de la cámara intersectante, de lo contrario no serán desplegados los objetos.

En la imagen superior podemos observar al centro el origen de la normal y la flecha que señala la dirección de la normal, por ende la cámara 1 será viable de uso ya que la dirección del rayo del punto de vista de la cámara sigue la dirección de la normal del objeto envolvente

Posición de las normales

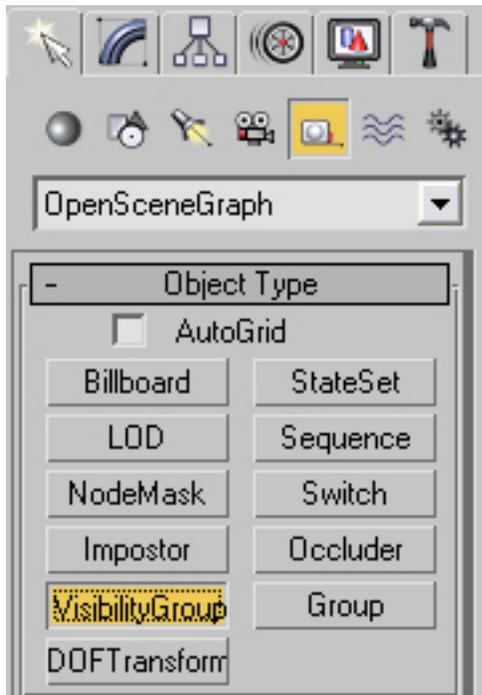


Las normales del objeto envolvente deben estar apuntando hacia afuera para que el rayo de la cámara que se encuentra adentro pueda llegar a hacer la intersección con alguno de los lados, en caso contrario, si las normales apuntaran hacia adentro el rayo de la cámara realizaría una intersección pero el ayudante de OSG Visibility Groups no lo detectaría.

2.- Implementación del ayudante de OSG Visibility Groups

Para implementar el ayudante de OSG Visibility Group dentro de la escena es necesario ocupar la versión instalada de OSG Max Exporter (la Versión 0.9.5 se encuentra en la dirección <http://osgmaxexp.wiki.sourceforge.net>).

2a.- Ubicación del ayudante



Dentro de 3D Studio Max, es posible ubicar al ayudante Visibility Group dentro de los ayudantes de Open Scene Graph.

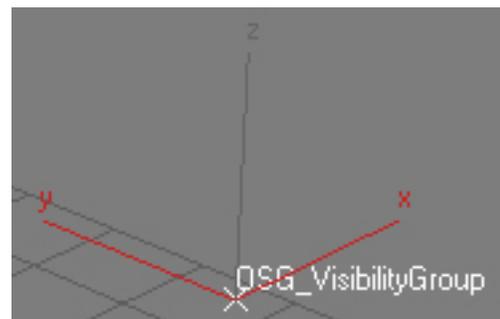
Para acceder a ellos es necesario seleccionar la pestaña Create para luego buscar el botón Helpers, dentro del menú desplegable se debe seleccionar la que dice OpenSceneGraph.

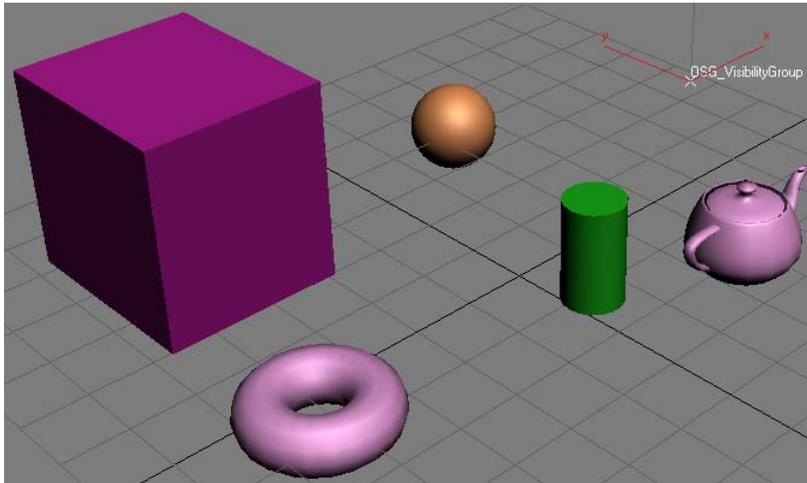
Posteriormente dentro de los botones se debe ubicar el botón de Visibility Group y dar click en el.

2b.- Implementación del ayudante

Para colocar el ayudante es necesario dar click sobre el botón Visibility Group para posteriormente dar click en algún área del viewport.

Aparecerá el diminuto ayudante de OSG_VisibilityGroup, posteriormente se debe configurar.

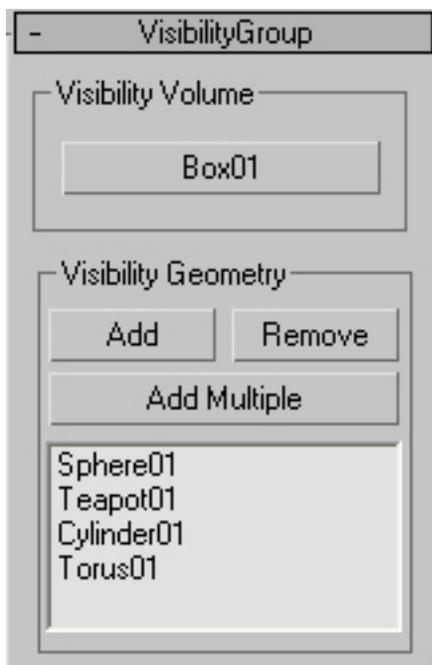




Se deberán genera los lementos del escenario y el objeto envolvente, en este caso la caja (Box1) será asignada como envolvente y los demas elementos geometricos seran el grupo de visibilidad.

Nota: Los objetos del grupo de visibilidad no necesitan estar dentro del envolvente para que este funcione.

2c.- Definición de geometrías



Para definir el comportamiento de cada geometria u objeto en la escena es necesario, en una primera instancia, dar clic en el boton None que se encuentra en la imagen asignado al Box1, dentro del Visibility Volume.

Continuando con la configuración se deben ubicar los objetos restantes (geometricos o no) a la lista del Visibility Geometry, una vez hecho esto es posible corroborar por medio de la exportación o previsualización de los modelos.

3.- Pruebas de exportación

Las pruebas de exportación fueron sistemáticas, en donde se exporto primeramente en formato OSG, tratando de ubicar el porqué de los errores que este denotaba se encontró que la exportación debía ser en formato IVE y posteriormente (fuera de 3D Studio Max) ser convertido en OSG a través del osgconv.

7. Resultados y discusión de Resultados.

Las pruebas de exportación llevaron a los siguientes resultados:

a.- Al exportar el rayo de intersección mide una unidad, lo que no se ajusta a todos los escenarios y debe ser ajustado en cada escenario.

Dependiendo del tamaño del escenario será necesario ajustar el rayo de intersección que proviene de la cámara pues por lo regular el valor de una unidad es demasiado bajo y deberá ser incrementado de forma manual dentro del código de OSG.

b.- Exportar en IVE (Binario) no en OSG (Ascii)
Versión OpenSceneGraph Max exporter 0.9.5

Puesto que el formato IVE es el formato adecuado para la exportación desde 3D Studio Max, los archivos deberán ser exportados en este formato y posteriormente (fuera de 3D Studio Max) ser convertido en OSG a través del `osgconv`, una vez hecho esto es posible cambiar manualmente el valor del rayo de intersección.

c.- Una vez exportado a OSG se debe pasar por un programa (en desarrollo) que corrija el rayo de intersección.

Será en un futuro conveniente utilizar un programa que se encuentra en desarrollo que permitirá al usuario aumentar el valor del rayo de colisión adaptándolo así a las necesidades específicas de cada proyecto.

8. Conclusiones y Recomendaciones.

Es conveniente planear completamente el escenario que se va a realizar definiendo las áreas de carga y descarga de modelos, a fin de evitar errores comunes, como modelos que aparecen de la nada o modelos que en algún punto de vista desaparezcan, puesto que crean una inconsistencia en la percepción de los elementos integrantes el entorno virtual.

Tomar en cuenta la dirección de las normales antes de realizar la exportación de modelos, exportarlos en formato IVE y convertirlos con el `osgconv` en formato OSG.

El uso de esta técnica beneficiara en gran medida a los proyectos que posean construcciones arquitectónicas con cuartos o con extensiones interiores grandes y con divisiones por pared o puertas, facilitando el despliegue continuo de las imágenes en pantalla.

9. Bibliografía.

Comunidad de desarrollo de Open Scene Graph. (2009). *osgSim VisibilityGroup Class Reference*.

Recuperado el 24 de marzo de 2009, de

<http://www.openscenegraph.org/documentation/OpenSceneGraphReferenceDocs/a01058.html>

Cordero, P. (06 de noviembre de 2008). *Recuperan esplendor en el mercado Abelardo L. Rodríguez*.

Recuperado el 24 de Febrero de 2009, de

<http://www.exonline.com.mx/XStatic/excelsior/template/content.aspx?se=nota&id=405086>

Gómez Flores, L. (19 de mayo de 2008). *Remodelan el histórico mercado Abelardo L. Rodríguez como parte del rescate del Centro*. Recuperado el 24 de febrero de 2009, de

<http://www.jornada.unam.mx/2008/05/19/index.php?section=capital&article=039n1cap>

Milenio Online. (18 de enero de 2009). *Restauran murales del mercado Abelardo Rodríguez*. Recuperado el 24 de febrero de 2009, de <http://impreso.milenio.com/node/8521183>