

• 1400191304  
ceplic

**Análisis de la coherencia temporal  
en escenas dinámicas  
iluminadas con radiosisdad**

Ignacio Martín

Report LSI 94 18 T

 **UPC**  
Facultat d'Informàtica  
de Barcelona - Biblioteca  
21 FEB. 1995

# Análisis de la coherencia temporal en escenas dinámicas iluminadas con radiosidad

**Ignacio Martín**

6 de Octubre de 1994

Universidad Politécnica de Catalunya  
Departamento de Lenguajes y Sistemas Informáticos  
Sección de Informática Gráfica  
Av. Diagonal 647 planta 8 (ETSEIB)  
08029 Barcelona

## Abstract

Este report analiza los antecedentes existentes en el uso de la coherencia temporal para la aceleración del cálculo de secuencias de animación en escenas dinámicas iluminadas mediante radiosidad. La importancia de estas técnicas reside en el alto coste computacional del método de la radiosidad, y por ello se hace uso de la existencia de una gran coherencia entre imágenes sucesivas de una secuencia de animación.

## 1 Introducción

En la última década, en informática gráfica se ha trabajado en la obtención de imágenes fotorrealistas, es decir, imágenes que desde el punto de vista de la percepción humana serían indistinguibles de unas que se hubiesen obtenido con una cámara fotográfica. Así, han surgido una serie de modelos de iluminación que pretenden simular en una determinada escena el comportamiento real de la luz, y su efecto sobre los elementos que la componen. En [27] Hall da una clasificación de los modelos de iluminación conforme ha ido evolucionando la informática gráfica.

Los primeros modelos de iluminación, los denominados incrementales, fueron completamente empíricos. Los cálculos de iluminación no tenían en cuenta las interreflexiones de las superficies de la escena ([11], [28], [32], [51]).

Posteriormente, y basándose en principios físicos y ópticos, aparecieron los modelos transaccionales. Fundada en dichos principios surgió la técnica del *ray tracing* ([53]), especialmente diseñada para trabajar con superficies muy reflectantes (especulares). A pesar de ello, ha sido ampliada para tratar superficies difusas ([10], [1], [33]).

Recientemente se han desarrollado modelos analíticos partiendo de principios termodinámicos, dando como resultado la técnica de la radiosidad ([42], [23], [7], [40]). Inicialmente trataba solo objetos cuyas superficies eran idealmente difusas, aunque pronto aparecieron ampliaciones que permitían efectos especulares ([32]).

Por último hay que destacar la integración de las técnicas de *ray tracing* y de radiosidad, dando lugar a modelos híbridos ([51]) que tratan tanto superficies difusas como especulares. En los últimos años, y debido al espectacular aumento de la potencia de cálculo de las workstations y del desarrollo de hardware gráfico, se ha abordado el problema de la animación realista de escenas. También ha contribuido a este auge el hecho de que los resultados de estas animaciones han llegado al gran público, por ejemplo a través de la industria cinematográfica o la realidad virtual, lo que ha hecho que se aumenten los recursos dedicados a la investigación en este campo.

Debido al alto coste computacional que requiere una animación de calidad, el desarrollo del hardware gráfico no ha hecho posible que se alcancen resultados óptimos (p.e. animación interactiva de escenas complejas) y eso ha llevado al desarrollo de técnicas que aprovechan

la coherencia temporal existente en una animación, para obtener mejores resultados.

El objetivo principal del report es estudiar esta coherencia temporal en la animación de escenas iluminadas mediante radiosidad. Los precedentes se encuentran en las animaciones de entornos estáticos (*walkthroughs*) y modificaciones interactivas de escenas. Por otra parte, existen una serie de trabajos de coherencia temporal que no abordan la técnica de la radiosidad, aunque desarrollan métodos que pueden aplicarse a distintas fases de dicha técnica. Tratan sobre coherencia temporal en el cálculo de la visibilidad, estructuración de escenas, manejo de memoria en animaciones interactivas, etc... Solo en los últimos años han comenzado a aparecer intentos de aprovechar la coherencia temporal en escenas dinámicas que utilizan la radiosidad como método de iluminación.

El resto del report constará de tres partes. Primero, explicar la técnica de la radiosidad y sus diferentes fases, y definir el concepto de animación con sus diversas variantes. A continuación, definir de forma general qué es la coherencia temporal, cómo puede ser aplicada en cada fase de la radiosidad, y clasificarla según diferentes criterios. Por último, dados todos los precedentes existentes sobre coherencia temporal, clasificarlos según el tipo de animación que tratan y según dónde se hace uso de la coherencia.

## **2 Definiciones**

### **2.1 Radiosidad**

Uno de los grandes problemas de la síntesis de imágenes realistas ha sido el modelado del comportamiento de las superficies al ser alcanzadas por la luz. De manera general puede decirse que hay dos tipos de comportamientos ideales, que constituyen los extremos de un intervalo en el que se sitúan todas las superficies "reales"; estas superficies ideales son : superficies especulares y superficies difusas (**Figura 1**).

Las superficies especulares son aquellas que al recibir un rayo de luz, lo reflejan en una sola dirección, siguiendo las leyes de Lambert. Este enfoque dio como resultado la técnica del *ray tracing* ([53], [2]), que ya se ha mencionado anteriormente. Las superficies difusas, por el contrario, al recibir un rayo de luz, lo reflejan por igual en todas las direcciones (siempre hacia "fuera" de la superficie). Esto hace posible tratar el problema de la iluminación como un problema termodinámico ([34]) en el que se tiene un entorno cerrado (no se pierde energía), y dadas unas fuentes de energía (emisores de luz) se calcula el punto de equilibrio de intercambio de energía entre todas las superficies, de manera que al final se conoce para

cada punto de la superficie la energía (radiosidad) que irradia. De este modo se puede



**Figura 1.** A la izquierda el comportamiento de una superficie perfectamente difusa, y a la derecha una superficie de tipo especular.

calcular la iluminación de cada superficie de la escena una sola vez, y posteriormente visualizar la escena desde cualquier punto de vista con un coste mínimo.

El método básicamente consiste en la determinación del valor de una función (radiosidad) sobre una superficie ([9]). La expresión de dicha función es la siguiente :

$$[\text{Eq. 1}] \quad B(x) = E(x) + \rho(x) \int_S B(x') \frac{G(x, x') V(x, x')}{\pi} dA'$$

donde :

$B(x)$  : valor de la radiosidad en un punto  $x$  de una superficie de la escena.

$E(x)$  : energía emitida por un punto  $x$  de una superficie de la escena. Si la superficie no es una fuente luminosa, este valor es cero.

$\rho(x)$  : reflectividad de un punto  $x$  de una superficie de la escena. Es la fracción de la energía recibida que es reflejada.

$S$  : es la superficie de todos los objetos de la escena. La integral es sobre esta área.

$G(x, x')$  : es la atenuación que se produce al viajar la luz desde el punto  $x'$  al  $x$ .

$V(x, x')$  : es la función de visibilidad; vale 1 si los puntos  $x$  y  $x'$  son visibles mutuamente y cero en otro caso.

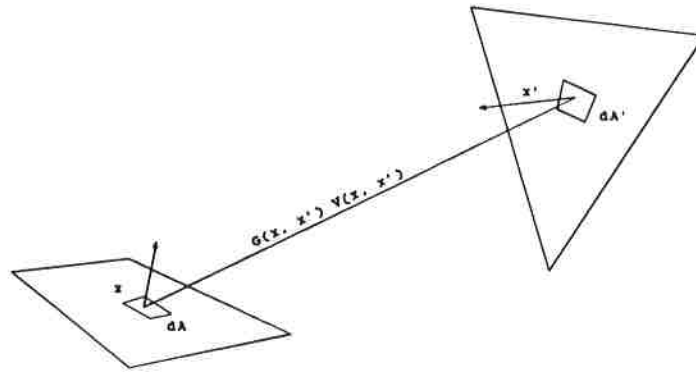


Figura 2. Elementos que intervienen en la ecuación de la radiosidad.

Este tipo de ecuaciones raramente tienen soluciones analíticas y deben utilizarse métodos numéricos para resolverlas. Concretamente se aplica el método de los elementos finitos ([29]) que consiste en subdividir las superficies del entorno en un número finito de pedazos (elementos), y en cada uno de ellos aproximar la función original por una polinomial más simple. Es lo que se denomina proyección de una función en un espacio finito de funciones. Discretizando la ecuación se obtiene la siguiente expresión :

$$[\text{Eq. 2}] \quad B(x_i) = E(x_i) + \rho(x_i) \sum_{j=1}^n B(x_j) F_{ij}$$

La interpretación física de esta ecuación es que la radiosidad de un pedazo es igual a la que emite (si es una fuente de luz) más la suma de las radiosidades del resto de pedazos multiplicadas por un factor  $F_{ij}$  que representa la fracción de energía del pedazo  $j$  que llega al pedazo  $i$ . Este factor se denomina Factor de Forma (*Form Factor*) y sólo depende de la geometría de los pedazos  $i, j$ . Además, el hecho de que  $i$  y  $j$  sean o no visibles está incluido en el mismo  $F_{ij}$ , de manera que si no son visibles su valor será cero.

La resolución de esta ecuación llevó a los primeros trabajos sobre radiosidad ([11], [28], [32]) que obtenían los  $B(x_i)$  de una manera directa, aplicando métodos iterativos de resolución de sistemas de ecuaciones lineales. La solución del sistema de  $n$  incógnitas se lleva a cabo en tres fases :

1. Determinación de los  $n$  pedazos. Asumiendo una escena compuesta de caras planas, se

reduce a una subdivisión de dichas caras, en función de la precisión buscada.

2. Cálculo de los factores de forma. Junto con la emisividad de cada superficie ( $E(x_i)$ ) y la reflectividad ( $\rho(x_i)$ ), determinan los coeficientes de las  $n$  ecuaciones.

3. Solución del sistema. Después del cálculo de los factores de forma queda un sistema de  $n$  ecuaciones con  $n$  incógnitas y puede resolverse con cualquier método iterativo, con una rápida convergencia debido a las propiedades numéricas de los factores de forma.

Estos tres pasos constituyen el algoritmo básico de la radiosidad. Debido a su alto coste computacional se han desarrollado técnicas para acelerarlo. Como se verá, cada paso tiene una problemática bastante diferente.

El primer paso consiste en subdividir la superficie en pedazos en los que la función de radiosidad será aproximada por una función simple (polinomial), y a partir de estas se construye una aproximación global de la radiosidad sobre toda la superficie ([5]). Cuanto menor sea el tamaño de estos pedazos (y mayor su número), mejor será la aproximación y mayor será el coste.

El cálculo de los factores de forma es la parte más costosa de este método. Como se ha indicado antes, este valor depende solamente de la geometría de la escena y representa, para cada pareja de pedazos ( $i,j$ ) la fracción de energía que viniendo de  $j$ , sale reflejada de  $i$ . Esto implica calcular para todas las parejas posibles, si son visibles o no, lo que inicialmente supone un coste cúbico sobre el número de pedazos; teniendo en cuenta que después de realizar la subdivisión del entorno el número de pedazos es mucho mayor que el número de polígonos de la escena, se observa claramente que este paso es muy costoso. Se han propuesto muchos métodos para el cálculo de los factores de forma, de los que se puede encontrar un survey en [40].

Por último, la resolución del sistema por métodos iterativos tradicionales implica mantener la matriz de coeficientes en memoria principal, y dicha matriz ocupa un tamaño cuadrático en el número de pedazos, con lo que en escenas complicadas puede ser imposible almacenarla en la memoria.

## 2.2 Animación

La animación de una escena tridimensional consta de dos elementos fundamentales: un conjunto de elementos (objetos, fuentes de luz, medios participativos, etc..) que configuran la escena propiamente dicha, y un punto de vista (observador) desde el que se contemplan dichos elementos. Estos componentes poseen cierto grado de dinamismo, como movimiento (tanto de los objetos como del observador), cambios en la geometría (deformaciones, descomposiciones ...), cambios en las características de los materiales (color, rugosidad, transparencia, texturas, emisividad, reflectancia ...), etc... Una secuencia de animación viene definida por una serie de imágenes de una escena tomadas desde diferentes puntos de vista que representan puntos consecutivos en el camino que sigue el observador; entre cada punto de vista transcurre un tiempo determinado en función del cual se actualizan las características dinámicas de los elementos de la escena. En función de la densidad de imágenes por unidad de tiempo se obtiene un mayor o menor grado de continuidad en la animación.

Una posible clasificación de los tipos de animación teniendo en cuenta el grado de dinamismo de los diferentes componentes de la escena es:

1. Animación propiamente dicha en la que los objetos se pueden mover, deformar, modificar sus atributos, y en la que la posición del observador se puede desplazar.
2. Modificación interactiva del punto de vista en un entorno estático. Este tipo de animaciones también se les denomina *walkthrough*.
3. Modificación interactiva del entorno en el proceso de modelado de una escena. No es una animación propiamente dicha (no hay continuidad), pero presenta problemáticas similares.

Abordando la animación utilizando la técnica de radiosidad como método de iluminación, los casos 1 y 3 se hacen particularmente costosos computacionalmente, pues el movimiento de un solo objeto puede afectar en teoría a toda la solución del sistema, lo que significa repetir todos los cálculos de nuevo para cada imagen. Sin embargo, entre dos imágenes de una secuencia suelen haber muy pocos cambios. Teniendo esto en cuenta se han desarrollado una serie de técnicas que aprovechan esta coherencia temporal entre imágenes para reducir los cálculos.



## **3 Coherencia temporal**

### **3.1 Definición.**

El uso de la coherencia temporal a la hora de calcular una secuencia de animación consiste en utilizar información previa que reduzca los cálculos en alguna fase del proceso de construcción de dicha secuencia. La cantidad y tipo de información que puede usarse dependerá de diferentes factores, como por ejemplo, si el movimiento de los objetos es conocido a priori, el modelo de iluminación que se utilice, etc... En [49] se da una definición de coherencia temporal, que es clasificada en dos grandes grupos :

- **Objetiva.** Consiste en el uso de información conocida a priori, es decir, antes de obtener ninguna imagen de la secuencia. Esta información se calcula en preprocesos que suelen generar estructuras de datos que se mantienen sin actualizar a lo largo de la secuencia y que aceleran determinadas fases del proceso de animación. Su aplicación natural son las animaciones de entornos estáticos, o aquellas en las que los movimientos de los objetos están predefinidos.

- **Predictiva.** Aquí se parte de toda la información que se ha acumulado durante el cálculo de las primeras  $n$  imágenes para calcular la  $n+1$ . Suelen ser algoritmos heurísticos y es el tipo de coherencia que se usa en animaciones interactivas.

### **3.2 Radiosidad**

En el primer apartado se ha hecho una breve descripción de esta técnica. Ahora se verá de forma general dónde se puede aplicar la coherencia temporal dentro de cada fase, y en el siguiente apartado se analizarán más detalladamente las distintas propuestas existentes en la literatura.

- **Subdivisión en pedazos.** Para no realizar esta subdivisión desde el principio para cada imagen, se puede reaprovechar la subdivisión de la imagen anterior o hacer subdivisiones que tengan en cuenta el movimiento de los objetos. También pueden aprovecharse técnicas existentes para reducir el número de pedazos y adaptarlas a entornos dinámicos; los métodos jerárquicos ([28], [30], [16]) son un ejemplo.

- Cálculo de los factores de forma. Con el objetivo de no recalcular todos los FF para cada imagen, se puede aplicar la coherencia temporal de diversas formas: identificar que FF no cambian de una imagen a la siguiente para no recalcularlos, o partir de los FF de la imagen anterior para calcular los nuevos de forma incremental.

Para el cálculo de los FF se emplean métodos aproximados como *ray tracing* ([52]), o el hemicubo ([8]). En ambos métodos se ha estudiado la coherencia temporal de diversas formas (*ray tracing* [25], [12], [3], [13], [22], *hemicubo* [14])

- Resolución del sistema. En este caso aprovechar la coherencia temporal supone aplicar técnicas de cálculo numérico a la hora de resolver el sistema de ecuaciones de las diferentes imágenes de una secuencia, de manera que no haya que resolver cada sistema desde el principio.

- Aproximación de la solución. Para obtener una imagen de calidad aceptable no es necesario resolver el sistema de forma global y con total exactitud. Pueden realizarse aproximaciones rápidas a la solución que generen imágenes de calidad suficiente, de manera que al calcular una secuencia pueda partirse del resultado obtenido en la imagen anterior para generar la siguiente mediante una reaproximación que requiera pocos cálculos.

Estos métodos de aproximación, para escenas estáticas, son *radiosidad progresiva* ([6]), *jerárquica* ([45], [28]) e *importance driven* ([43]), que pueden combinarse. Posteriormente se han adaptado estas técnicas para tratar el dinamismo ([14], [20], [16], [37]).

#### **4 Usos de la coherencia temporal**

Los primeros trabajos que han tratado la coherencia temporal lo han hecho fundamentalmente sobre algoritmos EPA. Algunos de estos trabajos fueron [48], [36], [31], [17]. En [49] puede encontrarse una completa recopilación de dichos trabajos.

A la hora de caracterizar la coherencia temporal pueden utilizarse diversos criterios. En adelante se analizarán dos que son el tipo de animación donde se aplica dicha coherencia, y la fase del proceso de animación donde se aplica.

#### 4.1 Análisis de antecedentes según el tipo de animación

En el apartado 2.2 se ha definido la animación y se ha clasificado según el grado de dinamismo de los elementos que toman parte. Ahora se describirán las diferentes técnicas que tratan las diferentes clases de animación.

##### 4.1.1 Modificación interactiva de la escena.

En [14] y [20] se parte de la técnica de radiosidad progresiva ([6]) para adaptarla permitiendo añadir y quitar objetos de una escena, mostrando los cambios casi en tiempo real. En [20] se habla de redistribución de radiosidad, mientras que [14] de radiosidad incremental.

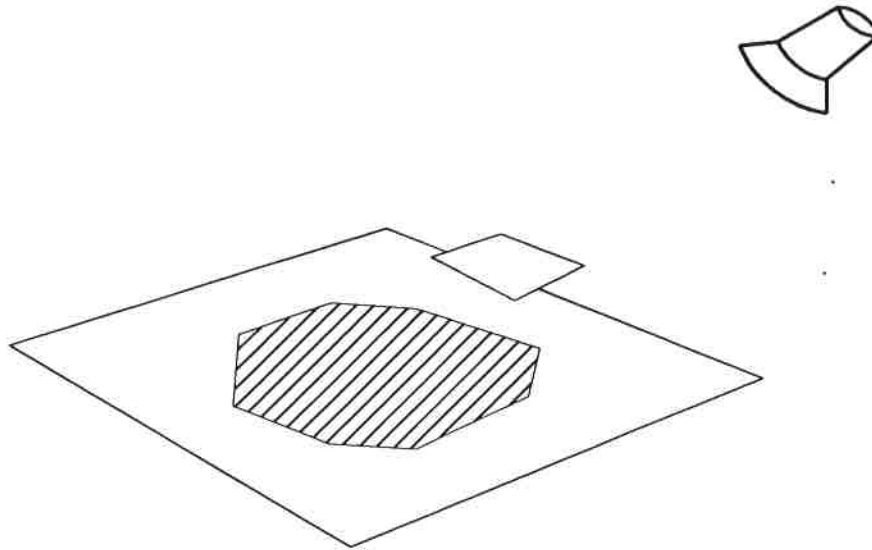
##### *Radiosidad incremental y redistribución de radiosidad*

El objetivo de estas dos técnicas (incremental [14], redistribución [20]) es, partiendo de una solución obtenida mediante un número determinado de iteraciones del método progresivo, realizar una transición gradual hacia una nueva solución en la que se han añadido o eliminado objetos. Esto se consigue reconstruyendo la solución que se ha producido hasta el momento, y para ello se modifican las radiosidades de los pedazos que han sido afectados por los cambios. En el caso de añadir un objeto (el proceso de eliminación es prácticamente simétrico), aquellos pedazos que pertenecen al objeto añadido deben recibir la energía de los pedazos iluminados que le son visibles, mientras que los pedazos que queden total o parcialmente ocultos respecto a un pedazo iluminado (debido al nuevo objeto) deberán restar la energía que habían recibido de dicho pedazo (**Figura 3**). Las dos técnicas describen el mismo método, aunque en [20] se propone una forma más flexible de utilización.

Para calcular la magnitud de la energía que hay que sumar o restar se utiliza Eq. 2, pero teniendo en cuenta que se obtiene una ecuación dependiente del tiempo, puesto que los FF cambiarán con las modificaciones de la escena. Considerando dos instantes de tiempo  $t-1$  y  $t$  entre los que se producen cambios en los FF, se tiene :

$$[\text{Eq. 3}] \quad B_i^{t-1} = E_i + \rho_i \sum_{j=1}^n F_{ij}^{t-1} B_j^{t-1}$$

$$[\text{Eq. 4}] \quad B_i^t = E_i + \rho_i \sum_{j=1}^n F_{ij}^t B_j^t$$



**Figura 3.** Al añadir un objeto, los pedazos que quedan ocultos del foco tienen que restar la radiosidad que habían recibido de él para que la sombra se produzca.

Se denomina al incremento de radiosidad en el pedazo  $i$  y al incremento en el factor de forma de la pareja  $i,j$  :

$$[\text{Eq. 5}] \quad B_i^\Delta = B_i' - B_i'^{-1}$$

$$[\text{Eq. 6}] \quad F_{ij}^\Delta = F_{ij}' - F_{ij}'^{-1}$$

respectivamente. Hay que tener en cuenta que el signo de Eq. 6 depende de si el pedazo pasa a ser visible (signo positivo) o pasa a estar total o parcialmente oculto (signo negativo). Desarrollando Eq. 5 se obtiene:

$$[\text{Eq. 7}] \quad B_i^\Delta = \rho_i \sum_{j=1}^n F_{ij}^\Delta B_j'^{-1} + \rho_i \sum_{j=1}^n F_{ij}' B_j^\Delta$$

Definiendo como término de redistribución:

$$[\text{Eq. 8}] \quad R_i = \rho_i \sum_{j=1}^n F_{ij}^\Delta B_j'^{-1}$$

y sustituyendo Eq. 8 en Eq. 7 se obtiene :

$$[\text{Eq. 9}] \quad B_i^{\Delta} = R_i + \rho_i \sum_{j=1}^n F_{ij}^{\Delta} B_j^{\Delta}$$

Se puede observar que la ecuación de redistribución (Eq. 9) tiene la misma forma que la ecuación de radiosidad normal, exceptuando que el término de emisividad  $E_i$  es sustituido por el de redistribución  $R_i$ , y la incógnita pasa a ser  $B_i^{\Delta}$  en lugar de  $B_i^{\Delta}$ . Se puede solucionar esta ecuación en dos pasos : primero determinar los términos  $R_i$ , y segundo resolver  $B_i^{\Delta}$  con los métodos tradicionales usados para  $B_i$ .

Para determinar los  $R_i$  se puede seguir una estrategia progresiva e ir escogiendo secuencialmente los pedazos a los que les queda más energía por redistribuir, lográndose así una buena aproximación en pocas iteraciones. El algoritmo de redistribución queda de la siguiente manera:

$p = \text{pedazo\_mas\_importante}();$

$B_{\text{lanzada}} = B_p - \Delta B_p$  ; /\* esta es la energía lanzada por  $j$  hasta el momento \*/

para cada pedazo  $i$  en el nuevo objeto hacer

$$R = \rho_i F_{ip}^{\Delta} B_{\text{lanzada}};$$

$$B_i = B_i + R;$$

$$\Delta B_i = \Delta B_i + R;$$

fin para

para cada pareja  $i, p$  oculta por el nuevo objeto hacer

$$R = \rho_i F_{ip}^{\Delta} B_{\text{lanzada}};$$

$$B_i = B_i + R;$$

$$\Delta B_i = \Delta B_i + R;$$

fin para

Esto ha de repetirse hasta que se alcance cierto grado de convergencia; entonces podrá continuarse con la propagación del método progresivo. Sin embargo, si se permiten introducir cambios continuamente, puede ocurrir que el proceso de redistribución sea interrumpido por el usuario al modificar la escena; esto quiere decir que se comenzará la redistribución correspondiente al nuevo objeto sin haber terminado la del anterior. Así, la energía que no haya sido redistribuida cuando se realiza el nuevo cambio, no será redistribuida nunca. Para

evitar esto se guarda una historia de los cambios que se han ido produciendo, y cuando se busca el pedazo al que le queda más energía por redistribuir se tienen en cuenta todos los cambios almacenados en esa historia.

Como resultado se obtiene un método iterativo en que cada paso puede ser de dos clases : redistribución o propagación. Para combinarlos pueden usarse dos estrategias:

1. Redistribuir primero hasta alcanzar un cierto umbral y luego intercalar pasos de redistribución y propagación. Esto hará que los cambios que produce el objeto añadido sean reflejados en la pantalla rápidamente. Hay que tener en cuenta que un paso de redistribución es más rápido que uno de propagación debido a que sólo afecta a los pedazos del nuevo objeto y a los que deja en sombra.

2. Alternar pasos de redistribución y propagación continuamente. Esto hará que los pedazos emisores del nuevo objeto (y los que quedan en sombra) propaguen su energía positiva (y negativa) rápidamente.

Por último, los cambios en la emisividad o en la reflectividad no requieren repropagación, simplemente actualizar las radiosidades de los pedazos correspondientes al objeto modificado de la siguiente manera:

$$[\text{Eq. 10}] \quad \Delta B_i' = E_i' - E_i^{t-1} + \frac{(\rho_i' - \rho_i^{t-1})(B_i' - E_i')}{\rho_i^{t-1}}$$

### ***Radiosidad jerárquica con movimiento de objetos***

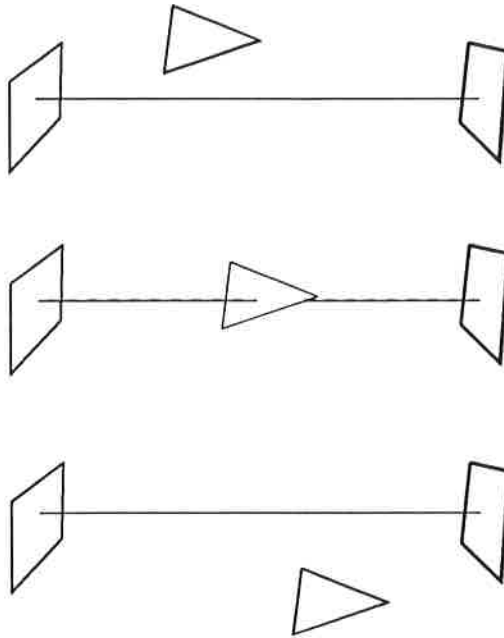
En [16] se parte del método jerárquico de radiosidad ([28]) para ampliarlo permitiendo el movimiento de objetos. Partiendo de una solución inicial obtenida con [28] se estudian los efectos del movimiento de los objetos sobre los enlaces (*links*) establecidos entre pedazos mutuamente visibles de la escena. Al pasar de una imagen a otra, un enlace puede verse modificado de tres maneras :

1. Comienza o termina una oclusión. Se produce cuando los pedazos correspondientes al enlace pasan a estar ocultos (o visibles) debido al movimiento de los objetos. Si el enlace se rompe (los pedazos pasan a ser no visibles) el FF correspondiente a dicho enlace toma el valor cero. En el caso contrario, el enlace vuelve a estar activo (los pedazos son visibles) y el FF

es restaurado con su valor correspondiente (**Figura 4**).

2. Promocionar un enlace. Cuando los objetos se alejan entre si, los FF entre sus pedazos decrecen; al alcanzar la distancia suficiente, el FF desciende lo suficiente como para que el enlace ascienda en la jerarquía. Puede ocurrir que un grupo de pedazos tenga que ser agrupado al quedarse estos sin ningún enlace.

3. Refinar un enlace. Cuando los objetos se acercan unos a otros, los FF se incrementan y puede ser necesario subdividir algún pedazo y establecer nuevos enlaces con los nuevos elementos creados.



**Figura 4.** Ejemplo de cómo el movimiento de un objeto puede causar el inicio de una oclusión y posteriormente su final.

El algoritmo que produce una secuencia de imágenes con este método es el siguiente:

1. Establecer una jerarquía inicial y resolverla para la primera imagen
2. Predecir qué enlaces serán promocionados, refinados, o sufrirán cambios en su visibilidad.

*Repetir*

3. Actualizar los enlaces para reflejar los cambios en la siguiente imagen
4. Predecir la nueva solución y usarla para comenzar la resolución iterativa

*hasta que no hayan más imágenes.*

El problema de este algoritmo reside en la predicción de los cambios que sufrirán los enlaces

a lo largo de la secuencia. La implementación realizada en [16] supone el conocimiento previo de los movimientos de los objetos, lo que permite determinar que enlaces permanecerán estáticos, y cuales no. Además se garantiza que los enlaces estáticos no cambiarán su visibilidad, con lo cual estos enlaces no sufrirán ninguno de los tres tipos de modificaciones descritas anteriormente.

Los enlaces dinámicos son tratados al principio (paso 2); se utiliza extrapolación lineal para determinar el número de imagen en el que se producirá algún cambio. Se almacena este número en el enlace, y cuando se alcanza esa imagen se modifica el enlace de la manera apropiada.

#### **4.1.2 Escena estática con movimiento del observador.**

##### ***Z buffer jerárquico***

En [24] se presenta un algoritmo EPA pensado para entornos complejos, en los que es necesario descartar el máximo de geometría no visible antes de visualizarla. Aprovecha tres tipos de coherencia:

1. Coherencia en espacio objeto mediante un *octree* del tipo de los utilizados para acelerar el *ray tracing*.
2. Coherencia en espacio imagen con una pirámide Z (**Figura 5**). El nivel más fino de dicha pirámide es el *Z buffer* original, de manera que el siguiente nivel se obtiene combinando cuatro valores Z en uno que será el mayor de los cuatro. Así, en la cima de la pirámide hay un solo valor Z que es el máximo de todos los del *Z buffer* inicial.
3. Coherencia temporal usando una lista de los nodos del *octree* que han resultado visibles en una imagen, para usarla como punto de partida de la siguiente.

El algoritmo para visualizar una sola imagen consiste en un recorrido del *octree*, de manera que para cada nodo se efectúan los siguientes pasos:

1. Se comprueba que el cubo correspondiente al nodo intersecta la pirámide de visualización. Si no intersecta se descarta; en caso contrario se continúa.
2. Se hace *scan conversion* de las caras del cubo para determinar si están ocultas o no. Si todas lo están, la geometría que contiene también lo estará, luego se descarta; en caso



contrario se continúa.

3. A cada polígono dentro del cubo se le aplica *scan conversion*.

21			
10		14	
21		7	
4	10	14	3
1	7	2	6
21	17	7	1
19	15	5	6

Figura 5. Pirámide Z correspondiente a un Z *buffer* de 4x4 pixels.

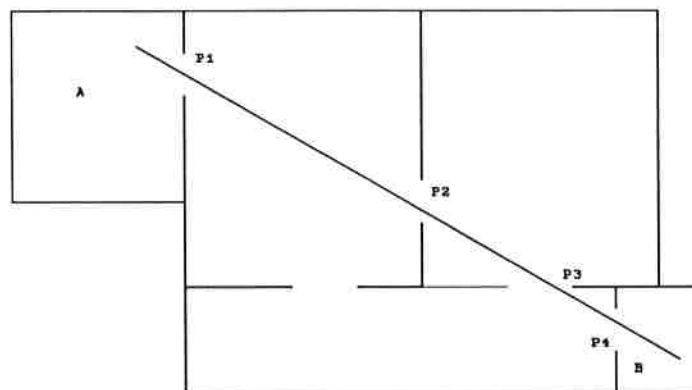
El proceso de *scan conversion* aprovecha la pirámide Z para descartar polígonos que están ocultos mediante un test de visibilidad recursivo. Se compara el valor de la cima de la pirámide con el valor Z más pequeño del polígono. Si el primero es menor que el segundo, el polígono está oculto y se descarta. En caso contrario se aplica el mismo test recursivamente en el nivel inferior. Llega un momento en que en un determinado nivel el polígono se clasifica como oculto, o bien se alcanza el nivel inferior de pixel y se actualiza su valor Z con el del polígono.

En el caso de una animación puede explotarse la coherencia entre imágenes sucesivas guardando en una lista los nodos del *octree* que han resultado visibles en la imagen anterior. Al calcular la nueva se hace *scan conversion* de la geometría de los nodos almacenados en la lista, ya que la mayoría continuará siendo visible y no se perderá mucho trabajo. Esto generará una pirámide Z correspondiente a un Z *buffer* de una imagen casi completa, es decir, con la mayor parte de los pixels con un valor Z; al iniciar el algoritmo se puede descartar mucha geometría desde el principio, sin haber de esperar a que se forme la pirámide Z.

### **Walktroughs**

Hay una serie de artículos que tratan las animaciones tipo *walkthrough*, especialmente pensadas para la navegación por edificios. De las referencias existentes [46], [19], [45] y [18], solo [45] trata la iluminación con el método de radiosidad jerárquica, el resto son simples algoritmos EPA.

En esta última referencia se utiliza la radiosidad dentro de un entorno estático. La principal aportación es la reducción de los cálculos de visibilidad entre los polígonos de la escena mediante un preproceso en el que se realiza una subdivisión del espacio en basada en BSP, con la que se descompone el modelo en un conjunto de celdas. Posteriormente se determina la visibilidad entre estas celdas mediante la construcción de un grafo de adyacencias que establece que dos celdas son visibles si comparten un portal. La idea de portal corresponde



**Figura 6.** Las celdas A y B se clasificarán como mutuamente visibles ya que existe una secuencia de portales P1-P2-P3-P4 a través de la cual puede trazarse una línea entre ambas celdas.

a una ventana o una puerta dentro de un edificio, que es donde mejor rendimiento se obtiene de este método. (Figura 6)

Así pues, la visibilidad entre dos polígonos se calcula sólo si pertenecen a celdas que son mutuamente visibles; si resultan visibles se crea un enlace entre ellas en el que se almacena la lista de polígonos que potencialmente pueden impedir dicha visibilidad. Para que esta lista sea mínima se utiliza una estructura de datos denominada *tube* que codifica el conjunto de todas las líneas que van de un polígono a otro; está basada en el uso de la representación de

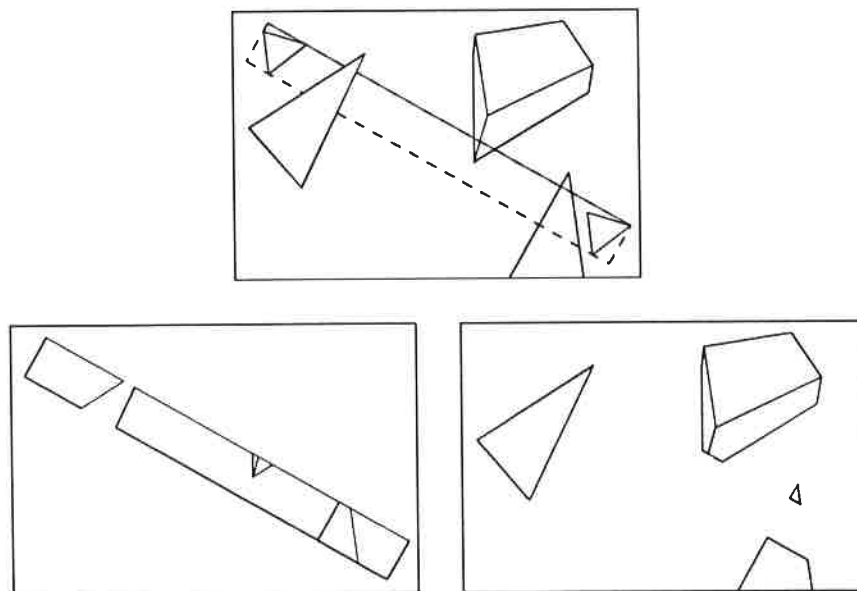
líneas *linespace* ([44], [47]). Esta estructura permite que a la hora de subdividir un polígono en pedazos, el cálculo de las nuevas visibilidades pueda hacerse de forma incremental.

#### 4.1.3 Movimiento de objetos con observador estático.

Al permanecer quieto el observador es posible que partes de la imagen permanezcan sin modificarse a lo largo de la animación, y puede predecirse si se conoce a priori el movimiento que seguirán los objetos. Bajo estas condiciones puede ser útil trabajar sobre espacio imagen.

##### ***Back-buffer***

En [4] se presenta la técnica del *back-buffer* que explota la coherencia temporal que existe desde el punto de vista fijo de un pedazo estático dentro de un entorno dinámico. Este método determina los FF que permanecerán invariantes durante la secuencia, y los calcula en un preproceso una sola vez. Asumiendo la técnica del hemicubo determina que pixels no cambiarán, y cuales estarán temporalmente ocultos por algún objeto dinámico; estos últimos vienen definidos por los volúmenes de barrido de los objetos dinámicos y son almacenados en el *back item buffer* (Figura 7).



**Figura 7.** Arriba se muestra el recorrido de un objeto por la imagen. Debajo a la izquierda está el *back-item buffer* que almacena los pixels de la pantalla que pueden verse ocultos por el paso de un objeto; por último la imagen base que corresponde a los pixels que no se verán modificados.

Así pues el algoritmo consta de dos pasos:

-Preproceso: calcula los FF que no cambiarán durante la secuencia y se inicializa el *back item buffer*.

-Actualización: se calcula la visibilidad para los pedazos temporalmente ocultos, y para los correspondientes a objetos dinámicos. Esto se hace comparando el *back item buffer* con la posición actual de los objetos dinámicos.

#### 4.1.4 Dinamismo completo

Finalmente se encuentran métodos que combinan todos estos factores permitiendo cualquier tipo de movimiento, tanto de objetos como de observador. Exceptuando una referencia que trata la radiosidad en entornos realmente dinámicos, los antecedentes existentes son adaptaciones de métodos usados en entornos estáticos, que en algunos casos podrían aplicarse en diversas fases de la radiosidad.

Suponiendo un conocimiento previo del movimiento de los pedazos de la escena, [25] y [13] utilizan subdivisión adaptativa del entorno para acelerar el trazado de rayos, trabajando sobre espacios de más de tres dimensiones al añadir la componente temporal.

#### *Radiosidad incremental con SFFL*

En [37] se mejora el método de radiosidad incremental ([14], [20]) mediante el uso de unas estructuras de datos denominadas SL (*Shooting List*) y SFFL (*Shadow Form Factor List*) que almacenan una información que permite una repropagación de energía (debida a un cambio en la escena) más eficaz. Sin embargo, como se verá más adelante, no se permiten introducir nuevos cambios hasta que la repropagación del anterior no ha finalizado, a diferencia de lo visto en [14].

Para cada iteración del método progresivo se añade al final de la lista la información del pedazo emisor seleccionado, almacenando la información mostrada en la **Figura 8**. En cada una de estas iteraciones se lanza un rayo desde el punto medio del pedazo emisor hacia los puntos medios del resto de pedazos de la escena, creando la SL correspondiente a esa iteración. En cada entrada de la SL se almacena si se ha encontrado algún obstáculo al lanzar el rayo; si hay oclusión se almacena el identificador del objeto que la produce, y si no se calcula el FF correspondiente y se almacena (**Figura 9**). Cada vez que se produce un cambio

se interrumpe el método progresivo y se llevan a cabo dos pasos:

1. Cálculo de los efectos primarios. Consiste en determinar para cada pedazo de la escena la radiosidad incremental generada al producirse el cambio, que se almacenará en cada campo *diff* de la SL.

	Iteración 1	Iteración 2	Iteración 3		Iteración t		Iteración n
ID-objeto					$s_t$		
$\Delta B$					$\Delta B'$	*	
siguiente				....		....	
SFFL							
diff					$D'$		

**Figura 8.** La SL almacena para cada iteración el identificador del pedazo escogido, la radiosidad no lanzada  $\Delta B$  la siguiente entrada correspondiente al mismo pedazo, la SFFL, y la radiosidad incremental *diff*.

2. Repropagación. Se repiten de nuevo las  $n$  iteraciones que se llevaban efectuadas, propagando las radiosidades incrementales al resto de la escena.

Al finalizar, la solución obtenida tras las  $n$  iteraciones es coherente con el cambio introducido, y se puede reanudar el proceso.

	pedazo 1	pedazo 2	pedazo 3		pedazo M
sombra (si o no)				....	
FF o ID-objeto					

**Figura 9.** La SFFL almacena en cada entrada un campo booleano que indica si el segundo campo es un FF, o un identificador de objeto en el caso de que este pedazo no sea visible desde el pedazo emisor.

Los efectos primarios pueden ser producidos por tres tipos de modificaciones:

1. Cambio de la emisividad de un material. Para cada pedazo  $i$  que referencie ese material, su radiosidad incremental será:

$$[\text{Eq. 11}] \quad D = E^{\text{nueva}} - E^{\text{vieja}}$$

la radiosidad estos pedazos se actualizará así:

$$[\text{Eq. 12}] \quad B_i^{\text{nueva}} = B_i^{\text{vieja}} + D$$

Las radiosidades no emitidas de estos pedazos se actualizan de la misma forma, pero sólo si no han sido escogidos como emisores; si lo han sido se actualizará la primera entrada correspondiente al pedazo  $i$  sumando  $D$  al campo *diff*.

2. Cambio de la reflectividad de un material. Para cada pedazo  $i$  que referencie ese material, su radiosidad incremental será:

$$[\text{Eq. 13}] \quad D_i = (\rho^{\text{nueva}} - \rho^{\text{vieja}}) \frac{B_i - E_i}{\rho^{\text{vieja}}}$$

La radiosidad del pedazo se actualiza como antes, sumando  $D_i$ .

Las radiosidades no emitidas de estos pedazos se actualizan como sigue:

$$[\text{Eq. 14}] \quad \Delta B_i^{\text{nueva}} = \frac{\rho^{\text{nueva}}}{\rho^{\text{vieja}}} \Delta B_i^{\text{vieja}}$$

Para actualizar el campo *diff* de las entradas correspondientes al pedazo  $i$  se procede de la siguiente manera: primero se busca la iteración  $t_1$  donde se produce la primera aparición del pedazo  $i$ ; la radiosidad incremental correspondiente

$$[\text{Eq. 15}] \quad D_{t_1} = (\rho^{\text{nueva}} - \rho^{\text{vieja}}) \frac{\Delta B_{t_1} - E_i}{\rho^{\text{vieja}}}$$

se suma al campo *diff* de  $t_1$ ; para el resto de entradas  $t'$  del pedazo  $i$ , el valor que se suma es

$$[\text{Eq. 16}] \quad D_{t'} = (\rho^{\text{nueva}} - \rho^{\text{vieja}}) \frac{\Delta B_{t'}}{\rho^{\text{vieja}}}$$

3. Cambios en la geometría de la escena. Estos cambios pueden ser de tres tipos : añadir un objeto, eliminar un objeto, y mover un objeto.

3.1 Añadir. Para cada entrada  $t$  de la SL hay que identificar los pedazos que pasan a ser no

visibles debido a la aparición del nuevo objeto. Se construye un *shadow-volume* ([26]) para identificar los pedazos candidatos a ser ocultados. Para cada pedazo  $i$  que la SFFL identifique como visible desde  $s_i$  se traza un rayo hasta él; si el nuevo objeto es alcanzado, se actualiza la entrada  $i$  de la SFFL de  $s_i$  almacenando el identificador del objeto. La radiosidad incremental que se añade a *diff* es:

$$[\text{Eq. 17}] \quad D_i' = -\rho_i \Delta B' F_{is_i}$$

Además de esto, hay que inicializar las radiosidades de los pedazos que forman el nuevo objeto.

3.2 Eliminar. Para cada entrada  $t$  de la SL se recorre la SFFL correspondiente, comprobando si el identificador del objeto eliminado aparece como ocluidor. Si aparece en la posición  $i$  de la SFFL de  $t$ , es posible que esta pareja de pedazos pase a ser visible. Para comprobarlo se lanza un rayo de  $t$  a  $i$ ; si se encuentra otro objeto en el camino, se coloca su identificador y no se produce ninguna radiosidad incremental; si no hay intersecciones la radiosidad incremental que se añade a *diff* es :

$$[\text{Eq. 18}] \quad D_i' = \rho_i \Delta B' F_{is_i}$$

3.3 Movimiento de objetos. En este caso se sabe que determinados objetos son dinámicos y van a ser borrados y añadidos continuamente. Cada vez que se mueve uno de estos objetos se recorre cada pedazo emisor  $s_i$  de la SL, y para cada uno se considera cada pedazo  $i$  del objeto dinámico. Hay que distinguir cuatro casos:

a) El pedazo  $i$  no era visible desde  $s_i$ , pero ahora sí lo es. Hay que calcular el nuevo FF y almacenarlo en la SFFL. La radiosidad incremental es :

$$[\text{Eq. 19}] \quad D_i' = \rho_i \Delta B' F_{is_i}^{\text{nuevo}}$$

b) El pedazo  $i$  era visible antes del cambio, pero ahora no lo es debido a un objeto  $O$ . En la SFFL se reemplaza el FF por el identificador de  $O$ . La radiosidad incremental es :

$$[\text{Eq. 20}] \quad D_i' = -\rho_i \Delta B' F_{is_i}^{\text{viejo}}$$

c) El pedazo  $i$  era visible y continua siéndolo. Hay que calcular el nuevo FF y actualizar la entrada correspondiente de la SFFL. La radiosidad incremental es:

$$[\text{Eq. 21}] \quad D_i' = -\rho_i \Delta B' (F_{is_i}^{\text{nuevo}} - F_{is_i}^{\text{viejo}})$$

d) El pedazo  $i$  no era visible debido a un objeto, y continua siéndolo debido a otro objeto. Se

actualiza la SFFL con el identificador del nuevo objeto. No hay radiosidad incremental.

Después de estos tres tipos de cambios hay que llevar a cabo la repropagación de las radiosidades incrementales que han aparecido. Este paso consiste en un recorrido de la SL en el que se buscan las entradas cuyo campo *diff* sea distinto de cero. Para cada una de estas entradas  $t$  correspondiente a un pedazo  $i$  se realizan las siguientes acciones:

1.  $B_i^{n,nueva} = B_i^n + \rho_i D' F_{is}$

2. Si el pedazo  $i$  no aparece como emisor en una iteración mayor que  $t$ , su radiosidad no emitida puede actualizarse de la misma manera; en otro caso se buscan las iteraciones  $t'$  posteriores a  $t$  en las que aparece  $i$ , y se actualiza la radiosidad incremental igualmente:

$$D'^{t,nueva} = D'^t + \rho_i D' F_{is}$$

3. La nueva radiosidad no emitida correspondiente a la iteración  $t$  se actualiza de la siguiente forma:

$$\Delta B^{t,nueva} = \Delta B^t + D^t$$

4. La radiosidad incremental de  $t$  es borrada:

$$D^t = 0$$

## 4.2 Análisis de antecedentes según su lugar de aplicación

El proceso de animación consta de diferentes fases, no todas ellas relacionadas con el método de la radiosidad, aunque la coherencia temporal puede ser aplicada igualmente en dichas fases. En el apartado 3 hemos visto las fases de este proceso relacionadas con el método de la radiosidad, aunque hay técnicas que se aplican fuera de este método que aceleran el proceso de la animación. El uso de la coherencia temporal en cualquiera de ellas lleva a modos muy diversos de aplicación de dicha coherencia que hay que diferenciar.

### 4.2.1 Visibilidad

Aquí se engloban todos los algoritmos de aceleración de tests de visibilidad. Son aplicables al cálculo de los factores de forma y a la fase final de *rendering*. Pueden clasificarse en dos grandes grupos, dependiendo de si trabajan sobre espacio imagen o sobre espacio objeto:



#### a. Espacio imagen

El problema que tratan estos algoritmos es el siguiente: qué color, y por lo tanto qué superficie, corresponderá a un determinado pixel en una determinada imagen. Hay enfoques muy diferentes de este mismo problema.

La técnica del *back-buffer* [4] también trabaja en espacio imagen. Asume un observador estático y almacena en un *buffer* qué partes de la imagen no van a ser modificadas, y cuales hay que recalcular en cada imagen.

En [24] se utiliza un *Z buffer*, pero jerarquizado en forma de *quadtree*. Esto, añadido a un preproceso en el que se realiza una subdivisión adaptativa de la escena mediante un *octree*, permite descartar mucha geometría eficientemente. Además aprovecha la coherencia entre imágenes para descartar gran parte de la escena antes de aplicar el *Z buffer* clásico.

#### b. Espacio objeto

Este conjunto de algoritmos no tratan el problema de la visibilidad directamente. Son más bien problemas de estructuración del entorno que producen estructuras de datos que posteriormente servirán para acelerar los tests de visibilidad clásicos. Suelen ser preprocesos que aprovechan todo el conocimiento previo existente sobre los objetos y sus movimientos.

Antes se ha mencionado que en [24] se utiliza subdivisión espacial con *octree* para luego poder acelerar el *Z buffer*.

Usando *ray tracing* y conociendo a priori el movimiento de los objetos puede realizarse una subdivisión adaptativa en un espacio de más de tres dimensiones, teniendo en cuenta el tiempo. En [25] se realiza una estructuración de la escena en hipercubos (6D), con la particularidad de no se realiza totalmente en el preproceso, sino que el refinamiento de la subdivisión se realiza conforme se van lanzando los rayos.

La serie de artículos sobre *walkthroughs* utiliza una variante de los BSP para subdividir el espacio en celdas, aprovechando la predisposición que los modelos que representan edificios tienen para producir subdivisiones eficientes de este tipo. Las referencias [46], [19] y [45] tratan desde modelos axiales a cualquier tipo de modelos de polígonos convexos.

#### 4.2.2 Cálculo de los factores de forma.

La coherencia temporal no se suele aplicar para recalcular factores de forma que varían con los movimientos de los objetos.

Al emplear la técnica del hemicubo se ha propuesto recalcular solo la parte del *buffer* que ha sufrido cambios ([14]). En principio esto implicaría conservar todos los *Z buffers* de cada hemicubo, lo cual es muy costoso en términos de almacenamiento.

En [4], la técnica del *back-buffer* propone examinar al principio qué factores de forma van a permanecer constantes a lo largo de la animación para calcularlos una sola vez.

En [37] se mantiene en memoria la SFFL que permite recalcular factores de forma a partir de datos correspondientes a iteraciones anteriores.

#### **4.2.3 Aproximación rápida de la solución del sistema.**

El método de la radiosidad, en un principio, requiere la resolución completa del sistema hasta un cierto grado de precisión. Esto implica un alto coste y el hecho de tener que esperar hasta el final de este proceso para ver los resultados, con el riesgo de que estos no sean los esperados y se haya perdido tiempo. Las dos principales técnicas que intentan paliar este problema son : radiosidad progresiva ([6]) y radiosidad *importance-driven* ([43]).

La radiosidad progresiva obtiene la misma solución final que el método tradicional, pero a través de un método iterativo en el que en cada paso se distribuye la energía de un pedazo; el pedazo que se escoge es el que más energía tiene por distribuir (las fuentes de luz, al principio), de forma que en unas pocas iteraciones se consigue que gran parte de la energía haya sido distribuida, y la solución intermedia obtenida sea una buena aproximación que puede ser visualizada para que el usuario compruebe cómo va a ser el resultado final.

El método *importance-driven* sigue un camino distinto, ya que la solución final no es la misma que obtendría el método tradicional. Se parte del método jerárquico ([28]) y se tiene en cuenta en punto de vista desde el que se visualiza la escena para modificar el criterio de refinamiento de los enlaces. A cada pedazo se le asocia una "importancia"; es un valor que representa la contribución de dicho pedazo a la imagen final. Este valor se inicializa a cero para todos los pedazos, excepto para aquellos que son visibles, siendo el pedazo que ocupa el centro de la imagen el de mayor importancia. Esta importancia se transmite de forma similar a la radiosidad, y la combinación de ambas permite concentrar la precisión de los cálculos en aquellos pedazos que más contribuyen a la escena final. Esto implica que la solución global obtenida es menos precisa; sin embargo este aumento del error se produce de manera que su efecto en la imagen final es pequeño.

En [14] y [20] se presenta la radiosidad incremental (nomenclatura de [14]) que consiste en un método para tratar escenas dinámicas en las que el usuario introduce cambios interactivamente. Cuando se produce un cambio los pedazos que han lanzado energía vuelven a ser tratados repitiéndose el proceso de distribución de energía, pero teniendo en cuenta los cambios en los factores de forma.

En [37] aparece una mejora de la radiosidad incremental en la que se crea una estructura de datos (SFFL) que almacena los datos sobre los factores de forma que se calculan en cada iteración, y permite mejorar la redistribución de energía.

En [45] se propone un método de radiosidad jerárquica en el que se describe una técnica para acelerar el cálculo de la visibilidad entre pedazos mediante una subdivisión de la escena en celdas, lo que permite determinar de entrada muchas posibles oclusiones con un coste reducido. Este método es especialmente eficaz para modelos que representen edificios, en los que se producen numerosas oclusiones debido a la existencia de grandes superficies (suelos, paredes, techos) que forman espacios casi cerrados (habitaciones, pasillos).

En [30] se presenta una estrategia para mejorar la eficiencia de la radiosidad jerárquica. Consiste en no establecer enlaces iniciales entre superficies que no superen un test en el que se estima el factor de forma de una forma rápida, sin tener en cuenta la visibilidad.

En [16] se amplía el método jerárquico a entornos dinámicos, tal y como ha sido descrito en el apartado anterior.

En [50] se propone un método heurístico para escoger el pedazo emisor, dentro de un entorno dinámico. Combina la radiosidad progresiva con el método *importance-driven* ya que tiene en cuenta, aparte de la contribución energética del pedazo, su visibilidad respecto al observador y su movimiento.

### **4.2.3 Acotación del tiempo de visualización**

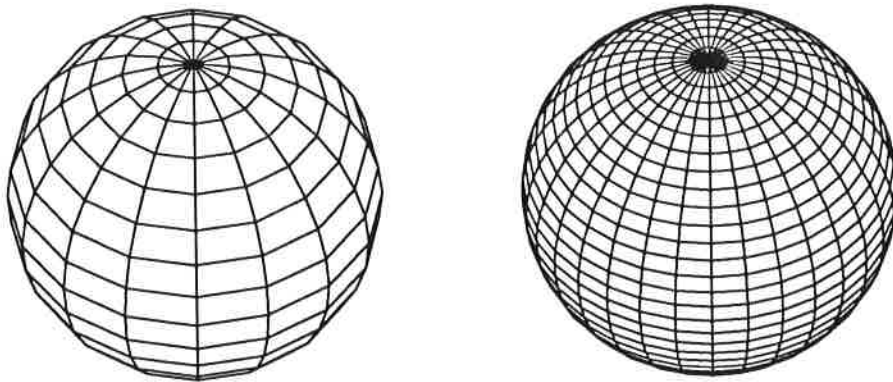
Sin tener relación con la radiosidad, este aspecto es muy importante en las animaciones interactivas, en las que la sensación de continuidad es fundamental. Esto requiere un número mínimo de imágenes por segundo, lo que hace que el tiempo para cada una esté limitado, y tardar más puede ser crítico, pues reduce notablemente la calidad de la animación.

Una de las técnicas fundamentales para acotar el tiempo de visualización consiste en la posibilidad de visualizar un mismo objeto en diferentes niveles de detalle. Esto implica que dado un mismo objeto se puede visualizar con costes computacionales muy diferentes. En [19]

y [18] se proponen diversos usos de esta técnica para mejorar los tiempos de visualización.

### ***Visualización adaptativa***

En [18] se propone un método heurístico que selecciona el nivel de detalle y algoritmo de visualización para cada objeto para producir la "mejor" imagen posible dentro de un tiempo máximo especificado por el usuario. Para garantizar un tiempo de imagen máximo el algoritmo debe de ser predictivo, basado en la complejidad de la escena que va a ser visualizada.



**Figura 10.** El mismo objeto representado en dos niveles de detalle diferentes.

Debe estimar el tiempo requerido para visualizar cada objeto a cada nivel de detalle y seleccionar el tamaño máximo que permita visualizar la imagen por debajo del tiempo máximo. Formalmente se define la tupla de objeto  $(O, L, R)$  como una instancia del objeto  $O$  visualizado con un nivel de detalle  $L$  y con un algoritmo de visualización  $R$ ; se definen también dos heurísticas:  $Coste(O, L, R)$  y  $Beneficio(O, L, R)$ . El  $Coste$  representa el tiempo necesario para visualizar una tupla, mientras que  $Beneficio$  representa la contribución de la tupla a la percepción de la imagen.

Sea  $S$  el conjunto de tuplas que van a ser visualizadas en una imagen; el problema que hemos planteado puede formularse como:

Maximizar  $\sum_s \text{Beneficio}(O, L, R)$

Sujeto a :  $\sum_s \text{Coste}(O, L, R) \leq \text{Tiempo máximo de imagen}$

La heurística  $\text{Coste}(O, L, R)$  es el tiempo necesario para visualizar el objeto  $O$  con un nivel de detalle  $L$  y un algoritmo de visualización  $R$ . Esto depende de un complejo conjunto de factores entre ellos el tipo y características de la *workstation* que se utilice; usando un modelo generalizado de visualización y haciendo algunas simplificaciones se puede conseguir una buena aproximación de esta heurística; en [18] se propone la siguiente:

$$\text{Coste}(O, L, R) = \max \left\{ \begin{array}{l} C_1 \text{Pol}(O, L) + C_2 \text{Vert}(O, L) \\ C_3 \text{Pix}(O) \end{array} \right\}$$

Los dos valores entre los que se escoge el máximo representan el coste de las dos etapas principales del proceso de visualización, que se ejecutan en paralelo.

El cálculo de  $\text{Beneficio}(O, L, R)$  es extremadamente difícil debido a su naturaleza, ya que representa la cantidad y precisión de información que le llega al usuario debido a la visualización de una tupla. Es prácticamente imposible desarrollar un modelo preciso sobre la percepción y la comprensión humanas, así que se ha optado por un método fácil de calcular basado en unos pocos principios intuitivos. Dicho método calcula una serie de parámetros cuyo valor oscila entre 0 y 1:

-*Tamaño*( $O$ ): una estimación de la porción de pantalla (en pixels) que ocupará el objeto  $O$  una vez visualizado.

-*Precisión*( $O, L, R$ ): depende del número de muestras que se utilicen para los cálculos de iluminación (número de pixels si se utiliza *ray tracing*, número de vértices si se utiliza el método de Gouraud, etc), y del método de interpolación que se use (constante, lineal, etc).

Tiene la siguiente expresión:

$$\text{Precisión}(O, L, R) = 1 - \text{ErrorBase}/\text{Muestras}(L, R)^m$$

-*Importancia(O)*: depende de la semántica del objeto, de su importancia para el usuario dentro de la animación.

-*Foco(O)*: los objetos que se encuentran en la parte de la pantalla que el observador está mirando deberían contribuir más a la imagen que los que se encuentran en la periferia. Al no tener disponibles mecanismos que permitan determinar el punto de mira del observador, se considera que el foco es el centro de la pantalla.

-*Movimiento(O)*: los objetos que se mueven con rapidez pueden aparecer borrosos y por ello disminuir su contribución a la imagen.

-*Histeresis(O, L, R)*: hace referencia a la perturbación que supone el cambiar el nivel de detalle o algoritmo de visualización de una imagen a la siguiente.

La expresión final del beneficio se obtiene multiplicando todos estos factores:

$$\text{Beneficio}(O, L, R) = \text{Tamaño}(O) * \text{Precisión}(O, L, R) *$$

$$\text{Importancia}(O) * \text{Foco}(O) * \text{Movimiento}(O) * \text{Histeresis}(O, L, R)$$

Sin embargo, el problema de optimización con restricciones es NP-completo. Se ha optado por una implementación sencilla que consiste en seleccionar las tuplas con un *Valor* ( $\text{Beneficio}(O, L, R)/\text{Coste}(O, L, R)$ ) más elevado, de manera que la suma de sus costes no exceda el tiempo máximo de imagen. En lugar de calcular *Valor* para cada tupla en cada imagen se aprovecha la coherencia de imagen a imagen y se ha desarrollado un algoritmo incremental.

#### 4.2.4 Manejo de memoria

Este es un aspecto bastante diferente de los demás, y al igual que el anterior, cobra especial importancia en animaciones interactivas en las que se visualizan grandes bases de datos que no caben en memoria principal. En estos casos el intercambio de datos y memoria es un punto crítico, sobre todo en aquellas imágenes en se producen variaciones bruscas y hay que traer repentinamente grandes cantidades de datos del disco. La solución está en desarrollar algoritmos predictivos que progresivamente vayan trayendo a memoria la información que puede ser necesitada repentinamente.

En [46] y [19] se estudia como realizar el *swapping* de objetos para que en determinadas imágenes no se dispare el tiempo de visualización.

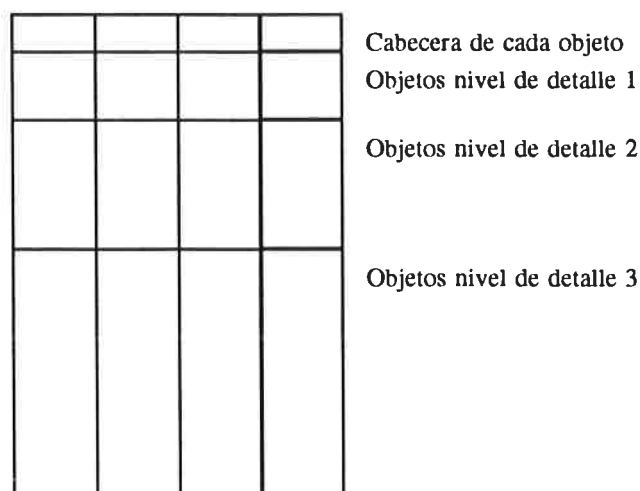
En [19] además se presenta un método completo para el manejo de memoria y disco que permite realizar *walkthroughs* a través de bases de datos moderadamente extensas que representan edificios. Hay dos aspectos fundamentales en dicho método:

- El uso de una base de datos jerárquica que describe el modelo del edificio como un conjunto de objetos representados en múltiples niveles de detalle.

- Un algoritmo de manejo de memoria en tiempo real que se encarga del trasvase de objetos entre memoria y disco mientras el observador se mueve por el edificio. Este algoritmo se basa en una descomposición espacial en celdas y la creación de un grafo de adyacencias, similar a los descritos en el apartado 4.1.2 ([45]).

Todas las entidades son almacenadas en segmentos en la base de datos. Un segmento es un bloque de bytes que puede ser leído mediante una sola operación de E/S. Tanto en memoria como en disco, existe un índice de segmentos que permite conocer el estado y posición de cada uno en un momento determinado. Todas las relaciones (adyacencia, visibilidad, ..) son almacenadas como referencias a segmentos, y para cada uno se mantiene un contador de referencias que nos permite leer y escribir segmentos rápida y transparentemente.

Para cada celda existe la lista de objetos que contiene; la descripción de estos objetos al mismo nivel de detalle se almacena contiguamente en la base de datos (**Figura 11**).



**Figura 11.** Disposición en memoria de los objetos pertenecientes a una misma celda.

En general, se cargan en memoria los contenidos de las celdas que pueden llegar a ser visibles en un corto periodo de tiempo. Para determinar cuales son se utiliza el *view frustum* y el grafo de adyacencias para calcular el tiempo mínimo en el que puede llegar a ser visible para cada celda. Teniendo en cuenta la memoria y el tiempo disponibles se va trayendo a memoria los objetos que forman las celdas; las que tardarán menos en ser visibles se traen primero y a un nivel de detalle alto, mientras que el resto se van trayendo progresivamente y disminuyendo el nivel de detalle.

## **5 Conclusiones**

En este report se ha hecho un estudio el uso de la coherencia temporal para la aceleración de secuencias de animación, centrándose en aquellas producidas mediante la técnica de la radiosidad.

Se han visto las técnicas que tratan la visibilidad global en entornos estáticos, que permiten animaciones en tiempo real de entornos moderadamente complejos. Utilizan estructuras de datos precalculadas que reducen los cálculos de visibilidad tanto a la hora de aplicar radiosidad como de visualizar una imagen en tiempo real.

También se han estudiado diferentes métodos para permitir el movimiento de objetos en entornos de radiosidad, que se basan en la adaptación de técnicas de radiosidad existentes para escenas estáticas. Estos métodos no permiten animación en tiempo real, aunque si permiten modelar escenas con radiosidad de forma interactiva.

Por último también se han estudiado algunos algoritmos adaptativos que permiten acotar los tiempos de cálculo de las imágenes. En concreto el uso de múltiples niveles de detalle en los objetos, y el manejo eficiente de la memoria.

## **6 Problemas abiertos**

La técnica de la radiosidad tiene su cuello de botella en el cálculo de la visibilidad a la hora de determinar los factores de forma. No existen técnicas que permitan tratar con eficiencia este problema, y menos cuando se considera el problema del dinamismo de los objetos. Las estructuras de datos elaboradas para entornos estáticos no son transformables para que puedan tratar el movimiento de forma eficiente, es decir mediante cálculos incrementales que aprovechen la coherencia temporal.



Por otro lado las técnicas que tratan el dinamismo en la radiosidad sólo permiten pequeñas modificaciones interactivas, y no hacen uso extensivo de la coherencia temporal a la hora de recalculer la visibilidad de los factores de forma.

Una de las grandes líneas de estudio es el desarrollo de estructuras de datos y algoritmos que permitan:

- una estructuración eficaz de escenas dinámicas,
- la actualización incremental de las estructuras de datos debido al movimiento,
- el cálculo de la visibilidad global, para cualquier pedazo de la escena, y
- la reconstrucción de la subdivisión del entorno en respuesta a cambios en la escena.

	Radioalidad				No radioalidad			
	Visibilidad		Cálculo de los factores de forma	Aproximación rápida de la solución	Visibilidad		Acotación del tiempo de visualización	Manejo de memoria
	Image Space	Object Space			Image Space	Object Space		
Modificación interactiva de la escena			Incremental [14]	Incremental [14],[20] Jerárquica [16]				
Escena estática con movimiento del observador (walkthrough)		Serie walkthroughs [45]		Z-Buffer jerárquico [24]	Serie walkthroughs [46],[19] ASP [39],[42]	Múltiples LOD's [19]	Swapping inteligente [46],[19]	
Movimiento de objetos con observador estático	Back Buffer [4]		Back-Buffer [4]		Ray tracing [12],[3]			
Variantes (todo tipo de dinamismo)			Incremental (SFLL) [37]	Incremental (SFLL) [37]		Octree [24] BSP [48] Ray tracing [25], [13]	Heurística (LOD's) [18]	

**Tabla 1.** Clasificación de los antecedentes según el tipo de dinamismo y lugar donde se aplica la coherencia temporal, indicando además si es un método de radioalidad o no.

	Motivación interactiva	Walkthrough	Observador estático	Variantes	Visibilidad		Cálculo de FF	Aproximación de solución	Acotación tiempo de imagen	Manejo de memoria
					E. imagen	E. objeto				
					Back buffer					
Radiosidad incremental	X				Si	Si				
Redistribución	X				Si	Si				
Rad. jerárquica dinámica			X			Si				
Rad. progresiva con SFFL				X	Si	Si				
Z-buffer jerárquico		X			Si	Si				
Tiempo de imagen constante		X						Si		

**Tabla 2.** Clasificación de las principales técnicas analizadas, indicando tipo de animación y forma de aplicación de la coherencia.

## Referencias:

- [1] Amanatides J. "Ray tracing with cones". *Computer Graphics (SIGGRAPH'84 Proceedings)*, 18:3 (Julio 1984), pp. 129-135.
- [2] Appel A. "Some techniques for shading machine rendering of solid". *Proceedings AFIPS JSCC vol. 32*, 1968, pp. 37-45.
- [3] Badt S. "Two algorithms for taking advantage of temporal coherence in ray tracing". *The Visual Computer* 4, 1988, pp. 123-132.
- [4] Baum, D.R., Wallace, J.R., Cohen, M.F., Greenberg, D.P. "The back-buffer algorithm: an extension of the radiosity method to dynamic environments". *The Visual Computer* 2:5 (Septiembre 1986), pp. 298-306.
- [5] Bern M., Eppstein D. "Mesh generation and optimal triangulation". *Computing in Euclidean Geometry*, World Scientific, 1992.
- [6] Cohen M.F., Chen S.E., Wallace J.R., Greenberg D.P. "A progressive refinement approach to fast radiosity image generation". *Computer Graphics (SIGGRAPH'88 Proceedings)* 22:4, (Agosto 1988), pp. 75-84.
- [7] Cohen M.F., Greenberg D.P. "The hemicube: A radiosity solution for complex environments". *Computer Graphics (SIGGRAPH'85 Proceedings)* 19:3 (Julio 1985), pp. 31-40.
- [8] Cohen M.F., Greenberg D.P., Immel D.S., Brock P.J. "An efficient radiosity approach for realistic image synthesis". *IEEE Computer Graphics and Applications* 6:3 (Marzo 1986), pp. 26-35
- [9] Cohen M.F., Wallace J.R. "Radiosity and realistic image synthesis". Academic, 1993
- [10] Cook R.L. , Porter T., Carpenter L. "Distributed ray tracing". *Computer Graphics (SIGGRAPH'84 Proceedings)*, 18:3 (Julio 1984), pp. 137-145.
- [11] Cook R.L., Torrance K.E. "A reflection model for computer graphics". *ACM Transactions on Graphics* 1:1 (1982), pp 7-24.
- [12] Chapman J., Calvert T.W., Dill J. "Exploiting temporal coherence in ray tracing". *Graphics Interface'90*, Canadian Information Processing Society Editors, Canada 1990, pp. 196-204.
- [13] Chapman J., Calvert T.W., Dill J. "Spatio-temporal coherence in ray tracing". *Graphics Interface'91*, Canadian Information Processing Society Editors, Canada 1991, pp. 101-108.
- [14] Chen S.E. "Incremental radiosity: An extension of progressive radiosity to an interactive

- image synthesis system". *Computer Graphics (SIGGRAPH'90 Proceedings)* 24:4 (Agosto 1990), pp 135-144.
- [15] Chrysanthou Y., Slater M. "Computing dynamic changes to BSP trees". *Eurographics'92*, 11:3 (1992), pp. 321-332.
- [16] Forsyth D., Yang C., Teo K. "Efficient radiosity in dynamic environments". *Fifth Eurographics Workshop on Rendering*, 1994, pp. 313-324.
- [17] Fuchs H., Abram G., Grant E.D. "Near real-time shaded display of rigid objects". *Computer Graphics, (SIGGRAPH'83 Proceedings)*, 17:3.
- [18] Funkhouser T.A., Sequin C.H. "Adaptative display algorithms for interactiv frame rates during visualization of complex virtual environments". *Computer Graphics (SIGGRAPH'93 Proceedings)*, 27 (Agosto 1993), pp. 247-254.
- [19] Funkhouser T.A., Sequin C.H., Teller S.J. "Management of large amounts of data in interactive building walkthroughs". *Computer Graphics, Special Issue (Proceedings 1992 Symposium on Interactive 3D Graphics, Cambridge, Mass. 29 Marzo, 1992)*, ACM Press, pp. 11-20.
- [20] George D.W., Sillion F.X., Greenberg D.P. "Radiosity redistribution for dynamic environments". *IEEE Computer Graphics and Applications* 10:4 (Julio 1990), pp. 26-34.
- [21] Glassner A.S. "An introduction to ray tracing". Academic Press Ltd., 1989.
- [22] Glassner A.S. "Spacetime ray tracing for animation". *IEEE Computer Graphics and Applications* 4:10 (Octubre 1984), pp. 15-22.
- [23] Goral C.M., Torrance K.E., Greenberg D.P. "Modelling the interaction of light between diffuse surfaces". *Computer Graphics (SIGGRAPH'84 Proceedings)* 18:3 (Julio 1984), pp. 213-222.
- [24] Greene N., Kass M., Miller G. "Hierarchical Z buffer visibility". *Computer Graphics (SIGGRAPH'93 Proceedings)*, 27 (Agosto 1993), pp. 231-238.
- [25] Gröller E., Purgathofer W. "Using temporal and spatial coherence for accelerating the calculation of animation sequences". *Eurographics'91*, Elsevier Science Publishers, Viena 1991, pp. 103-113.
- [26] Haines E.A., Wallace J.R. "Shaft culling for efficient ray traced radiosity". *2nd Eurographics Workshop on Rendering, Barcelona, España, Mayo 1991*, pp. 1-14.
- [27] Hall R. "Illumination and color in computer generated imagery". Springer-Verlag, New York, 1989.
- [28] Hanrahan P., Salzman D., Aupperle L. "A rapid hierarchical radiosity algorithm". *Computer Graphics (SIGGRAPH'91 Proceedings)* 25:4 (Julio 1991), pp. 197-206.
- [29] Heckbert P., Winget J. "Finite element methods for global illumination". Report n°

UCB/CSD 91/643, University of California at Berkeley, 1991.

- [30] Holzschuch N., Sillion F.X., Drettakis G. "An efficient progressive refinement strategy for hierarchical radiosity". Fifth Eurographics Workshop on Rendering, 1994, pp. 343-355.
- [31] Hubschman H., Zucker S.W. "Frame-to-Frame coherence and the hidden surface computation: Constraints for a convex world". ACM Transactions on Graphics, 1:2 (Abril 1982), pp. 129-162.
- [32] Immel D.S., Cohen M., Greenberg D.P. "A radiosity method for non-diffuse environments". Computer Graphics (SIGGRAPH'86 Proceedings) 20:4 (Agosto 1986), pp. 133-142.
- [33] Kajiya J.T. "Ray tracing volume densities". Computer Graphics (SIGGRAPH'84 Proceedings), 18:3 (Julio 1984), pp. 165-175.
- [34] Kajiya J.T. "The rendering equation". Computer Graphics (SIGGRAPH'86 Proceedings) 20:4 (Julio 1986), pp. 143-150.
- [35] Lischinski D., Tampieri F., Greenberg D.P. "Discontinuity meshing for accurate radiosity". IEEE Computer Graphics and Applications 12:6 (Noviembre 1992), pp. 25-39.
- [36] Matsushita Y. "Hidden line elimination for a rotating object". Communications of the ACM, 15:4 (Abril 1972), pp. 245-253.
- [37] Müller S., Schöffel F. "Fast radiosity repropagation for interactive virtual environments using a Shadow-Form-Factor-List". Fifth Eurographic Workshop on Rendering, 1994, pp. 325-342.
- [38] Nishita T., Nakamae E. "Continuous tone representation of three-dimensional objects taking account of shadows and interreflection". Computer Graphics (SIGGRAPH'85 Proceedings) 19:3 (Julio 1985), pp. 23-30.
- [39] Plantinga H., Seales W.B., Dyer Ch.R. "Real-Time hidden-surface elimination for a rotating polyhedral scene using the aspect representation". Graphics Interface'90, Canadian Information Processing Society Editors, Canada 1990, pp. 9-16.
- [40] Pueyo X. "Diffuse interreflections: techniques for form factor computation: a survey", The Visual Computer 7 (1991), pp. 200-209.
- [41] Seales W.B., Dyer Ch.R. "Shaded rendering and shadow computation for polyhedral animation". Graphics Interface'90, Canadian Information Processing Society Editors, Canada 1990, pp. 175-182.
- [42] Siegel R., Howell J.R. "Thermal radiation heat transfer". McGraw Hill Book Co., New York, 1991.

- [43] Smits B.E., Arvo J.R., Salesin D.H. "An importance-driven radiosity algorithm". *Computer Graphics (SIGGRAPH'92 Proceedings)* 26:4 (Julio 1992), pp. 273-282.
- [44] Sommerville, D. "Analytical geometry of three dimensions". Cambridge University Press, 1959.
- [45] Teller S.J., Hanrahan P. "Global visibility algorithms for illumination computations". *Computer Graphics (SIGGRAPH'93 Proceedings)*, 27 (Agosto 1993) pp. 239-246.
- [46] Teller S.J., Sequin C.H. "Visibility preprocessing for interactive walkthroughs". *Computer Graphics (SIGGRAPH'91 Proceedings)* 25:4 (Julio 1991), pp. 61-69.
- [47] Teller, S.J. "Computing the antipenumbra cast by an area light source". *Computer Graphics (SIGGRAPH'92 Proceedings)* 26:2, (Julio 1992), pp. 139-148.
- [48] Torres E. "Optimization of the Binary Space Partition (BSP) for the visualization of dynamic scenes". *Eurographics'90*. Elsevier Science Publishers, North-Holland, pp. 507-518.
- [49] Tost D. "Utilització de la coherència temporal entre imatges successives per a la visualització d'escenes 3D en animació". Tesis doctoral, Universitat Politècnica de Catalunya, Marzo 1991.
- [50] Tost, D., Pueyo X. "Temporal coherence in progressive radiosity". *Visualization and Intelligent Design*, Elsevier Science Publishers, 1993, pp. 77-90.
- [51] Wallace J.R., Cohen M.F., Greenberg D.P. "A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods". *Computer Graphics (SIGGRAPH'87 Proceedings)* 21:4 (Julio 1987), pp. 311-320.
- [52] Wallace J.R., Elmquist K.E., Haines E.A. "A ray tracing algorithm for progressive radiosity". *Computer Graphics (SIGGRAPH'89 Proceedings)*, 23:3 (Julio 1989), pp. 315-324.
- [53] Whitted T. "An improved illumination model for shaded display". *Communications of the ACM*, 23:6 (1980), pp. 343-349.