

# Representaciones Espaciales Para Escenas Constituidas Mediante Agrupaciones de Objetos

N. Aguilera, J. Revelles, M. Lastra, R. Montes, R.J. García  
Dpt. Lenguajes y Sistemas Informáticos  
E.T.S. Ingeniería Informática  
e-mail: noelio@fedro.ugr.es  
{jrevelle,mلاstral,rosana,ruben}@ugr.es  
Web: <http://giig.ugr.es>

## Resumen

En este trabajo se presentan varias representaciones espaciales eficientes que ofrecen un ajuste más adecuado a un determinado tipo de escenas formadas por grandes agrupaciones de objetos. Además, se presenta una representación espacial basada en jerarquía de octrees que ofrece mejores resultados que las inicialmente propuestas. Dichos resultados mejoran cuanto mayor es la distancia entre las agrupaciones de objetos a tenor de las escenas utilizadas en la realización de las pruebas empleando para ello un algoritmo de síntesis basado en fotosimulación.

**Palabras clave:** Ray-Tracing, Fotosimulación, Métodos de Optimización, Indexación espacial.

## 1. Introducción

En todo sistema de síntesis de imágenes es necesario incluir un núcleo dedicado a la optimización del test de intersección rayo-escena. Esta incorporación conlleva la disponibilidad de varios tipos de representaciones espaciales todos ellos aplicables a cualquier escena pero con mejoras sensibles según dos parámetros: el *tamaño* y la *distribución* de los objetos. La mejora en cuanto a rendimiento se consigue cuando, para cualquier rayo, se intente realizar el test de intersección rayo-objeto únicamente con aquellos objetos de la escena que estén en el camino del rayo.

Según se ha mostrado en los trabajos [14, 13], una representación espacial es un conjunto de objetos denominados volumétricos que deben cumplir la propiedad de

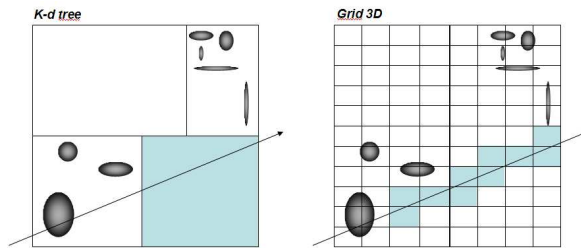


Figura 1: Ajuste de una Representación Espacial a una Escena (La izquierda es un K-d tree y la derecha es un Grid 3D).

que en su volumen total debe contener la totalidad de la escena, es decir, cualquier objeto de la escena debe estar alojado total o parcialmente en, al menos, un objeto volumétrico.

En [14] también se expone que, básicamente, existen dos tipos de representaciones espaciales aplicables al espacio de los objetos de la escena:

- **Representaciones Espaciales Simples:** Aquellas que emplean una vez un único tipo concreto de método de indexación espacial para la consecución de la misma.
- **Representaciones Espaciales Compuestas:** Aquellas que requieren dos o más tipos métodos de indexación espacial. Si estos tipos son idénticos, entonces se aplicaría dos o más veces.

En escenarios en los que el tamaño de los objetos varía muy poco entre ellos y la distribución de éstos no es muy variable, es posible aplicar una representación espacial basada en subdivisión espacial uniforme o Grid 3D [3, 1].

Cuando el anterior tipo de escenario no se da ya sea porque el tamaño de los objetos es muy variable (objetos muy grandes junto a otros muy pequeños) o bien la distribución de éstos no es homogénea, es más adecuado aplicar representaciones espaciales basadas en subdivisiones espaciales adaptativas. En este caso se han propuesto bastantes que, en su mayoría, tienen en común algún tipo de agrupación jerárquica. Destacamos la Jerarquía de Volúmenes Envolventes [9, 6], el Octree [5], BSP tree [17], K-d tree [4] y el Octree-R [18].

Aunque muchos autores han presentado trabajos relativos a mejoras de estas representaciones espaciales, el principal problema que puede presentar la elección de una en concreto es que no se ajuste adecuadamente a la escena. Varios parámetros podrían intervenir para medir el ajuste de una representación espacial a una escena. Uno de esos parámetros lo podría representar el número de objetos volumétricos de la misma que queden vacíos, es decir, sin albergar objetos de la escena. Es importante

tener en cuenta este factor ya que en el recorrido se puede perder mucho tiempo en atravesarlos hasta llegar a alcanzar alguno que incluya objetos con los que hacer el test de intersección rayo-objeto.

En la figura 1 se pueden apreciar dos representaciones espaciales aplicadas a la misma escena. La de la izquierda se basa en un K-d tree cuyo ajuste resulta mejor que el del Grid 3D (la de la derecha) para la escena según se puede apreciar para el rayo que se muestra.

Además de este tipo de escenas nos podríamos encontrar con otras más complejas en las que si bien es posible aplicar las representaciones espaciales anteriores, se podría evitar el problema anteriormente ilustrado. Un ejemplo de este tipo de escenarios en los que la distribución de los objetos está formada por agrupaciones de objetos son las escenas urbanas.

Existen representaciones espaciales compuestas que mejoran dicho ajuste en virtud del parámetro mostrado. Éstas son mostradas en la sección 2. En la sección 3 se presenta la representación espacial compuesta basada en jerarquía de octrees. Para mostrar las prestaciones de la jerarquía de octrees se proponen diversas escenas de prueba para mostrar en qué circunstancias la representación espacial propuesta ofrece mejores resultados. Se finaliza el artículo con las conclusiones y trabajos futuros en la sección 5.

## 2. Trabajos Previos

Cuando se diseña e implementa un sistema de síntesis de imágenes hay que tener en cuenta dos aspectos que son primordiales a la hora de obtener una imagen a partir de una escena:

- Selección del algoritmo de síntesis a emplear. Ésto conlleva la inicialización de los parámetros que caracterizan dicho algoritmo.
- Selección del optimizador que intentará acelerar el proceso de test de intersección rayo-escena. Ésto conlleva además la fijación de los parámetros característicos del mismo.

En cuanto a la determinación del optimizador a emplear, esta labor no suele estar automatizada ya que, actualmente, aún no se ha propuesto un selector automático que lo determine en función del algoritmo de síntesis a emplear y la escena de entrada. Varios trabajos en los que se muestran ciertas comparativas entre distintas representaciones espaciales están propuestos en [9, 11, 12, 7].

Tampoco suele estar automatizado el proceso de fijación de los parámetros característicos del mismo en función de la escena de partida, si bien, algunos autores han estado trabajando al respecto proponiendo nuevas ideas al respecto. Goldsmith y Salmon [6] propusieron una estrategia para determinar de forma automática el nivel de profundidad de una representación espacial basada en agrupaciones jerárquicas por

proximidad, es decir, jerarquía de volúmenes envolventes. Una generalización de esta estrategia pero para cualquier representación espacial de índole jerárquico (Octree, BSP tree, etc.) fue propuesta por Subramanian y Fussell en [16].

En resumen, mientras un grupo de autores han actuado sobre la automatización de los parámetros característicos de un optimizador concreto, otros han seguido estudiando la posibilidad de proponer nuevas representaciones espaciales que intenten ajustarse mejor a determinado tipo de escenas. En este sentido, uno de los trabajos más relevantes sobre escenas complejas fue el propuesto por Cazals y otros [2]. En este trabajo se presenta una representación espacial basada en una jerarquía de Grids 3D Uniformes.

La principal característica de dicha estrategia reside en que se ajusta mejor en escenas complejas cuya distribución de objetos esté repartida en agrupaciones. En una primera etapa se realiza una clasificación de los objetos por tamaños discretos (tres o cuatro tamaños a lo sumo según indican los autores). Posteriormente se realiza la fase de agrupación por tamaños, es decir, se forman grupos según los tamaños. En cada agrupación se construye un Grid 3D Uniforme y, finalmente, se construye una jerarquía de éstos atendiendo a la proximidad de las agrupaciones. Una ventaja notoria de esta representación espacial es que, además de ofrecer mejores prestaciones en escenas orientadas a grandes agrupaciones de objetos, los parámetros referentes a los niveles de subdivisión de los Grids 3D que se contruyen en cada agrupación y los tamaños selectivos de los objetos están totalmente automatizados. Ésto indica que, según la escena de entrada, la representación espacial es construida sin la intervención directa del usuario.

La representación espacial que se presenta en este trabajo, la cual será expuesta con detalle en la sección 3, toma la misma filosofía del trabajo anterior a excepción de que se construye un Octree en lugar del Grid 3D Uniforme.

### **3. Representación Espacial Basada en Jerarquía de Octrees**

A continuación se presenta la representación espacial empleada para la aceleración del test de intersección rayo-escena. Básicamente se basa en la misma concepción de representación espacial que Cazals y otros propusieron en su trabajo [2]. La principal diferencia es que aquí se emplean Octrees en lugar de Grids 3D Regulares.

Entendiendo que este tipo de representaciones espaciales compuestas son más adecuadas para escenas cuya distribución de los objetos sigue un determinado patrón de agrupamiento, también debemos aceptar que un Octree ofrece mejores prestaciones en cuanto a ajuste que un Grid 3D. La idea del trabajo de Cazals residía en que un Grid 3D era más eficiente que un Octree cuando los objetos poseían tamaños muy similares. De ahí su inicial clasificación por tamaños y su posterior agrupación en un Grid 3D.

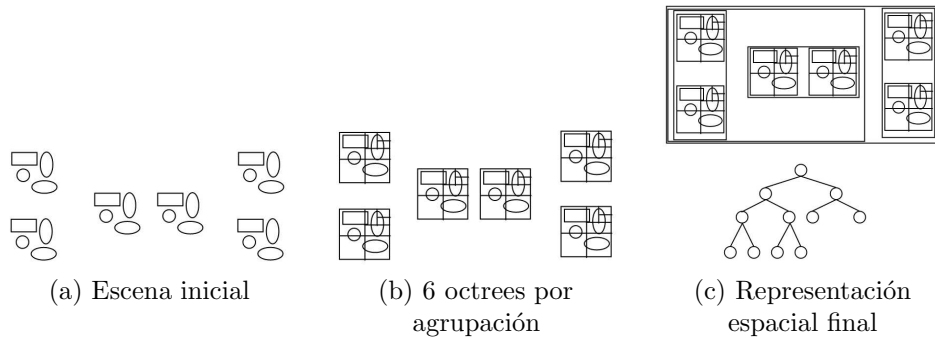


Figura 2: Ejemplo de escena con 6 agrupaciones.

---

**Algoritmo 1 . ConstruirArbol (S : Array de Envoltentes de Agrupaciones)**

---

```

N = vacío; // Lista que contiene nuevas envoltentes
if (NumeroAgrupaciones(S) == 1) then
  ConstruyeNodoRaiz (S[1])
end if
while (NumeroAgrupaciones(S) > 1) do
  {A, B} = SeleccionaAgrupacionesMasCercanas (S);
  S = S - {A, B};
  N = N + Envoltente (A, B);
end while
ConstruirArbol (N);
for i=1 hasta NumeroAgrupaciones(S) do
  ConstruyeNodo (S[i]) // Construye los nodos
end for

```

---

El método de Cazals tiene estas etapas en la construcción de la jerarquía de Grid 3D:

- Clasificación de los objetos por tamaños (tres o cuatro a lo sumo a indicación de los autores).
- Determinación automática de las agrupaciones de objetos por proximidad tomando como referencia los tamaños de los objetos de la clasificación del paso anterior.
- Construcción automática de un Grid 3D en cada agrupación obtenida.
- Construcción de la jerarquía en función de las agrupaciones obtenidas.

Dos son las principales diferencias del método de Cazals con respecto al que se propone. La primera de ellas reside en que partimos de que los objetos de la escena

---

**Algoritmo 2 . SelecAgrupacionesMasCercanas (S : Array de Envoltentes de las Agrupaciones)**

---

```
M = NumeroAgrupaciones(S) × NumeroAgrupaciones(S); // Producto
cartesiano de matrices
for i=1 hasta NumeroAgrupaciones(S) do
  for j=i hasta NumeroAgrupaciones(S) do
    M[i][j] = M[j][i] = DistanciaEntre (S[i], S[j]);
  end for
end for
DistanciaMinima = +∞;
for i=1 hasta NumeroAgrupaciones(S) do
  for j=i+1 hasta NumeroAgrupaciones(S) do
    if (DistanciaMinima > M[i][j]) then
      DistanciaMinima = M[i][j];
      A = i;
      B = j;
    end if
  end for
end for
return (S[A], S[B]);
```

---

poseerán tamaños muy similares, es decir, si se aplicase exhaustivamente dicho método a las escenas usadas para las pruebas, se obtendría un único grupo. Para este caso por tanto, no es necesario distinguir los tamaños de los objetos. El sistema de síntesis en el que se implementa esta representación espacial está basado en técnicas de estimación de densidades: Photon Maps [8] y DETP (Density Estimation on the Tangent Plane) [10].

La otra de las diferencias es que en la jerarquía de Octrees los nodos terminales del árbol binario balanceado están formados por Octrees en lugar de Grids 3D Regulares. El equilibrio en el árbol binario se consigue determinando las agrupaciones por proximidad y por pares bajo una caja envolvente repitiéndolo hasta que sólo quede una, la cual, será el nodo raíz del árbol.

Un ejemplo en el que se aplica el proceso anterior se puede ver en la figura 2(a) en la que existen claramente diferenciadas 6 agrupaciones. Tras la aplicación del método anterior se obtendrían las 6 agrupaciones. En cada una de ellas se construirá un Octree 2(b). Por último, al agruparlas por distancias se obtendrá el árbol binario balanceado que se muestra en la figura 2(c).

En los algoritmos 1 y 2 se esbozan el proceso de construcción de la Jerarquía de Octrees.

Además del proceso de construcción de la Jerarquía de Octrees, es necesario un algoritmo que simule el recorrido del rayo por ésta. En ese caso, el recorrido es tan

simple como el recorrer un árbol binario. Cuando se alcance un nodo terminal, el recorrido se basará en atravesar el octree que tenga asociado dicho nodo terminal.

Un pseudocódigo del algoritmo de recorrido de la Jerarquía de Octrees puede verse en los algoritmos 3 y 4. Para el recorrido del octree se emplea el algoritmo paramétrico propuesto en [15].

Un problema que se puede encontrar el algoritmo de determinación de las agrupaciones de objetos es que no se detecten las que realmente existen o no llegue a detectar ninguna. En esas situaciones, el resultado final de dicho proceso es que únicamente detecta una agrupación. Inicialmente no es problemático porque la representación final seguirá funcionando pero sin poder actuar con las mejores prestaciones sobre la escena.

---

**Algoritmo 3 . RecorridoPrincipal (JDO : Jerarquia de Octrees)**

---

```
intersecciones = vacío;  
for all (rayo) do  
  if (Interseca con la escena) then  
    RecorrerJerarquiaOctrees (JDO.NodoRaiz, rayo, intersecciones);  
  end if  
end for
```

---

---

**Algoritmo 4 . RecorrerJerarquiaDeOctrees (nodo, rayo, intersecciones)**

---

```
if (nodo es terminal) then  
  RecorrerOctree (nodo.octree, rayo, intersecciones);  
else  
  if (Interseca con nodo.izquierdo) then  
    RecorrerJerarquiaDeOctrees (nodo.izquierdo, rayo, intersecciones);  
  end if  
  if (Interseca con nodo.derecho) then  
    RecorrerJerarquiaDeOctrees (nodo.derecho, rayo, intersecciones);  
  end if  
end if
```

---

## 4. Resultados

Las prestaciones o mejoras que puede aportar la representación espacial que se presenta debe demostrarse en casos concretos en los que se pueda apreciar su beneficio. Para ello se han propuesto dos casos de pruebas:

1. Comparativa entre un Octree y una Jerarquía de Octrees.

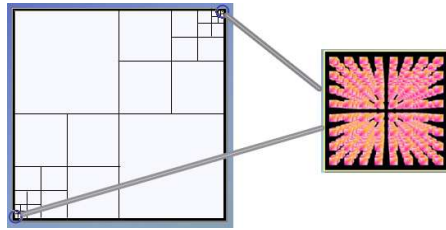


Figura 3: Escena con 2 agrupaciones muy separadas entre sí (cada agrupación es un array de  $8 \times 8 \times 8$  cubos).

## 2. Comparativa entre una Jerarquía de Octrees y una Jerarquía de Grids Uniformes.

Todos los resultados fueron obtenidos usando un ordenador PC con procesador Pentium 4 a 1.7 Ghz., 1GB de memoria RAM y Linux 2.4.21. En la tabla de resultados se considera únicamente el tiempo empleado en el lanzamiento de fotones y no el empleado en la estimación de densidades.

En la escena de la figura 3 existen dos agrupaciones muy separadas entre sí. Cada una de ellas consta, según se puede apreciar en la parte derecha de dicha figura, de un array de  $8 \times 8 \times 8$  cubos y una fuente de luz extendida situada justo encima. Cada cubo dispone de 12 triángulos (dicha cifra es el máximo número de objetos que se pueden alojar en un nodo hoja para no tener que seguir subdividiendo).

Si aplicásemos una representación espacial basada en una Jerarquía de Octrees se detectarían dos agrupaciones y, tomando como referencia que 12 es el número de objetos que se alojarán como máximo en un nodo hoja, necesitaríamos alcanzar un nivel de octree igual a 3. Sin embargo, si se aplicase un Octree que albergase ambas agrupaciones y siguiendo el criterio de número de objetos máximos por nodo hoja, necesitaríamos alcanzar un nivel igual a 10. Se podría razonar claramente que los requerimientos de memoria para alojar el Octree serían más exigentes que los de una Jerarquía de Grids Uniformes. Además, el beneficio en tiempo es bastante considerable ya que se ha llegado a alcanzar un beneficio de hasta un 87% usando la Jerarquía de Octrees.

La ganancia de la Jerarquía de Octrees reside principalmente en que no pierde demasiado tiempo en alcanzar las agrupaciones ya que, a través de la jerarquía, únicamente debe realizar un único descenso. Para el caso del Octree son necesarios 7 descensos hasta alcanzar la agrupación.

La escena anterior tiene la facilidad de que es previsible el nivel de profundidad de la Jerarquía de Octrees necesario para obtener los mejores resultados. Generalmente las escenas no suelen prestarse a este tipo de comodidades y, por contra, es necesario aventurar un nivel de profundidad adecuado para alcanzar un equilibrio entre consumo de memoria y la reducción del número de tests de intersección rayo-objeto a realizar.



Nivel Oct.	Tiempo (sec.)	%	Tiempo (sec.)	Nivel Grid
1	2.727,17	-6,85%	2.540,46	2
3	264,77	-15,95%	222,55	8
4	107,53	3,03%	110,89	14
5	63,16	21,62%	80,58	22
6	51,55	26,62%	70,25	37
7	52,37	35,71%	81,46	60
8	59,98	49,17%	118,00	102
9	75,87	61,59%	197,55	176

Escena de la figura 4(a)

Cuadro 1: Resultados obtenidos.

Nivel Oct.	Tiempo (sec.)	%	Tiempo (sec.)	Nivel Grid
1	2.204,75	-4,00%	2.116,47	2
3	216,28	-14,44%	185,05	8
4	90,93	11,96%	103,28	13
5	56,17	23,69%	73,61	20
6	46,84	29,08%	66,05	29
7	45,18	31,05%	65,53	40
8	45,82	39,38%	75,58	54
9	47,13	46,36%	87,87	71
10	48,14	55,23%	107,53	94
11	50,04	65,24%	143,95	133

Escena figura 4(b)

Nivel Oct.	Tiempo (sec.)	%	Tiempo (sec.)	Nivel Grid
1	1.802,86	-3,54%	1.738,97	2
2	557,85	-40,91%	329,65	5
3	184,23	-14,52%	157,48	8
4	81,65	15,20%	96,28	12
5	53,78	22,32%	69,23	19
6	46,54	26,27%	63,12	28
7	44,97	31,23%	65,39	37
8	45,70	36,90%	72,42	49
9	46,57	44,69%	84,20	64
10	47,41	53,80%	102,61	85
11	48,98	64,32%	137,26	121

Escena figura 4(c)

Cuadro 2: Resultados obtenidos.

Dado que la Jerarquía de Grids Uniformes ha sido ampliamente utilizada para una multitud de escenas compuestas por una gran cantidad de objetos, hemos considerado oportuno comparar ambas representaciones espaciales tomando como referencia el tamaño necesario para albergar ambas estructuras en memoria, es decir, para tamaños equivalentes.

Para esta comparativa se han diseñado cuatro escenas en las que varían tanto el número de agrupaciones como la distribución y tamaño de éstas. Las escenas 4(a), 4(b) y 4(c) representan un suelo y 5 tetras variando su ubicación para detectar respectivamente 2, 3 y 4 agrupaciones. El número total de triángulos es de 30,000. Las escenas 4(d) y 4(e) representan 5 tetras y otra en el centro muy grande y muy pequeña, respectivamente. El número total de triángulos es de 35,000.

Bajo estas circunstancias y según se puede apreciar en las tablas de resultados, en la mayoría de los casos es más favorable la utilización de una Jerarquía de Octrees. La información mostrada en las tablas 1, 2 y 3 es la siguiente: la primera columna

Nivel Oct.	Tiempo (sec.)	%	Tiempo (sec.)	Nivel Grid
1	909,22	-4,67%	866,77	2
2	239,71	-36,06%	153,28	5
3	90,05	-8,92%	82,02	8
4	49,07	15,32%	57,95	12
5	37,32	21,65%	47,63	18
6	34,36	26,92%	47,02	26
7	33,87	31,93%	49,76	35
8	33,39	40,34%	55,97	47
9	34,38	47,46%	65,44	62
10	34,79	57,23%	81,35	85

Escena figura 4(d)

Nivel Oct.	Tiempo (sec.)	%	Tiempo (sec.)	Nivel Grid
1	1.956,44	-20,18%	1.561,63	2
2	650,12	-47,95%	338,37	5
3	243,60	-26,15%	179,89	8
4	129,11	-15,20%	109,49	14
5	102,24	-6,18%	95,92	21
6	97,05	0,25%	97,29	29
7	99,16	7,69%	107,42	40
8	101,36	17,86%	123,40	52
9	104,91	31,49%	153,13	69
10	106,59	48,67%	207,65	98

Escena figura 4(e)

Cuadro 3: Resultados obtenidos.

muestra el nivel de profundidad de los octrees en la Jerarquía de Octrees. La siguiente columna muestra el tiempo empleado por el proceso de lanzamiento de fotones. La columna central muestra el porcentaje de ganancia (un resultado negativo muestra que la Jerarquía de Grids Regulares es mejor para ese caso). Las siguientes columnas son relativas a la Jerarquía de Grids en cuanto a tiempo y nivel de subdivisión de los Grids Uniformes.

## 5. Conclusiones y Trabajos Futuros

La principal aportación de esta representación espacial es que ofrece mejores resultados en términos de tiempo de ejecución para el proceso de lanzamiento de fotones según se ha podido mostrar en la sección anterior. En general, en cualquier algoritmo de síntesis basado en seguimiento de rayos se obtendrían similares beneficios.

En cuanto al tipo de escenarios, éstos deben estar formados por agrupaciones de objetos y, a la vista de los resultados, los beneficios aumentan cuanto mayor separación existe entre ellos. Esta ventaja es propia de las representaciones espaciales orientadas a estas agrupaciones.

Como trabajo futuro se pretende aplicar esta representación espacial a escenas más complejas y en concreto a escenas urbanas las cuales siguen estos principios. Añadir un procesado que determine el nivel del octree de forma automática también forma parte de los próximos esfuerzos. También sería interesante proponer múltiples tipos de escenarios en los que los objetos posean diferentes tamaños y medir las prestaciones de la Jerarquía de Octrees y la de Grids Uniformes con respecto a representaciones espaciales simples como son el BSP tree y el K-d tree de cara a poder tomar acuerdos que puedan determinar qué tipo de optimizadores son los más adecuados para un determinado tipo de escenarios.

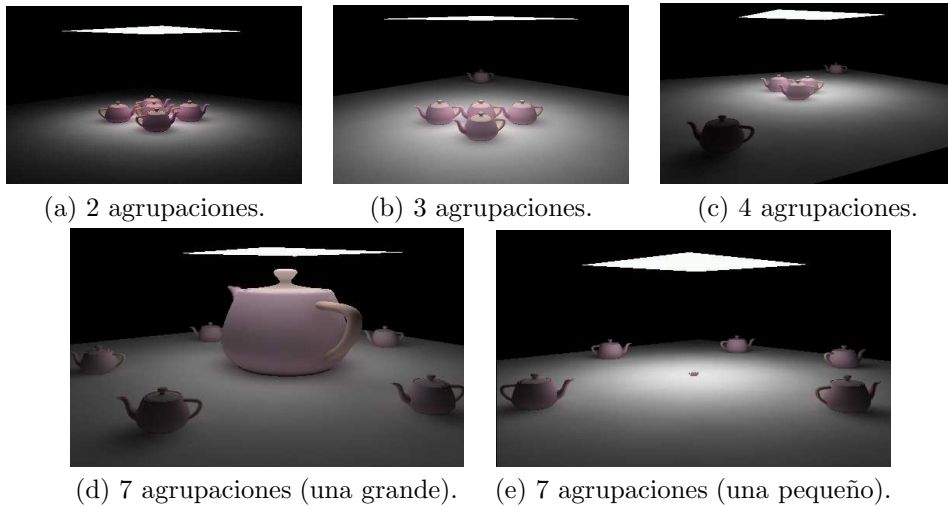


Figura 4: Escenas de pruebas (incluyendo el suelo como agrupación).

## Agradecimientos

Este trabajo ha sido realizado gracias a la financiación del Proyecto de Investigación TIC2001-2932-C03-03 de la Comisión Interministerial de Ciencia y Tecnología (CICYT).

## Referencias

- [1] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *EUROGRAPHICS'87*, pages 3–10, Amsterdam, 1987.
- [2] F. Cazals, G. Drettakis, and C. Puech. Filtering, clustering and hierarchy construction: A new solution for ray-tracing complex scenes. In *EUROGRAPHICS'95*, pages 371–382, 1995.
- [3] A. Fujimoto and K. Iwata. Arts: Accelerated ray tracing system. *IEEE Computer Graphics & Applications*, 6(4):16–26, 1986.
- [4] D.S. Fussel and K.R. Subramanian. Fast ray tracing using k-d trees. Technical Report TR-88-07, U. of Texas, Austin, Dept. Computer Science, March 1988.
- [5] A.S. Glassner. Space subdivision for fast ray tracing. *IEEE Computer Graphics & Applications*, 4(10):15–22, 1984.

- [6] J. Goldsmith and J. Salmon. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics & Applications*, 7(5):14–20, 1987.
- [7] V. Havran. *Heuristic Ray Shooting Algorithms*. Ph.d. thesis, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, 2001.
- [8] H. W. Jensen. Global illumination using photon maps. In *Rendering Techniques'96*, pages 21–30, 1996.
- [9] T.L. Kay and J. Kajiya. Ray tracing complex scenes. *Computer Graphics*, 20(4):269–278, 1986.
- [10] M. Lastra, C. Ureña, J. Revelles, and R. Montes. A particle-path based method for monte carlo density estimation. In *Eurographics Workshop on Rendering (short paper) 2002, Pisa (Italy)*, 2002.
- [11] J.D. MacDonald and K.S. Booth. Heuristics for ray tracing using space subdivision. *The Visual Computer*, 6(3):153–166, 1990.
- [12] M.D.J. McNeill, B.C. Shah, M.P. Hebert, P.F. Lister, and R.L. Grimsdale. Performance of space subdivision techniques in ray tracing. *Computer Graphics Forum*, 11(4):213–220, 1992.
- [13] J. Revelles, M. Lastra, R.J. García, and R. Montes. A formal framework approach for ray-scene intersection test improvement. In *11th. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2003 (WSCG'2003)*, ISBN: 80-903100-2-8, 2003.
- [14] J. Revelles and C. Ureña. A formalization of ray casting optimization techniques. In *XI Congreso Español de Informática Gráfica, CEIG'2001*, ISBN: 84-8458-061-X, 2001.
- [15] J. Revelles, C. Ureña, and M. Lastra. An efficient parametric algorithm for octree traversal. *Journal of WSCG (Copyright UNION Agency-Science Press)*, 8(2):212–219, 2000.
- [16] K.R. Subramanian and D.S. Fussell. Automatic termination criteria for ray tracing hierarchies. In *Proceedings of Graphics Interface'91*, pages 93–100, June 1991.
- [17] K. Sung. *Ray Tracing with the BSP Tree*, pages 271–274. Academic Press, 1992.
- [18] K. Whang, J. Song, J. Chang, J. Kim, W. Cho, C. Park, and I. Song. Octree-r: An adaptive octree for efficient ray tracing. *IEEE Transaction on Visualization and Computer Graphics*, 1(4):343–349, 1995.