

SIMULACIÓN DE LA VISIÓN ESTEREOSCÓPICA. PARTE II: SIMULACIÓN MEDIANTE UN TRAZADOR DE RAYOS Antonio Fernández Caballero Gabriel Sebastián Rivera Edwin Paredes Calizaya

Antonio Fernández Caballero es profesor del Departamento de Informática, Escuela Politécnica Superior de Albacete, Universidad de Castilla-La Mancha.

Gabriel Sebastián Rivera y Edwin Paredes Calizaya son colaboradores en proyectos de investigación.

RESUMEN

Este artículo presenta la segunda parte de un trabajo sobre simulación de la visión estereoscópica mediante un trazador de rayos. En la primera entrega se ofrecían los aspectos físicos básicos de la visión humana, para pasar seguidamente a introducir las nociones relacionadas con la estereoscopía, haciendo un especial hincapié en la idea de disparidad. Este segundo artículo comenta los pasos concretos a seguir para conseguir simular esta visión estereoscópica merced al uso de un trazador de rayos.

1. INTRODUCCION

La estereopsis es un fenómeno óptico que nos permite apreciar la profundidad de los objetos que estamos observando, al igual que la distancia a la que se encuentra dicho objeto. Los estudios realizados sobre nuestra visión "normal" han llegado a determinar que el hecho de que tengamos dos ojos nos ayuda a poder calcular las distancias a las que se encuentran todos los objetos que nos rodean. Con un solo ojo esto no se puede apreciar.

Tomando como base los estudios de la fisiología del ojo humano, el funcionamiento de la reproducción del mundo que nos rodea a través de nuestros ojos y las propiedades físicas de la óptica, nos hemos planteado simular la visión humana a partir de elementos artificiales. En concreto hemos usado aplicaciones informáticas y funciones de procesamiento de la señal, que nos han servido de material para poder construir unos "ojos" artificiales con los cuales poder "ver" objetos artificiales dentro de un mundo tridimensional artificial.

Se van a ir exponiendo las operaciones a medida en que avancemos en la simulación. En primer lugar haremos una breve descripción de la visión artificial que deseamos generar, después haremos una descripción completa de los medios que se han utilizado para el planteamiento de la simulación. Posteriormente haremos una demostración de cómo se genera movimiento y de la simulación de la visión estereoscópica.

2. DESCRIPCIÓN DE LA VISIÓN ARTIFICIAL QUE SE DESEA GENERAR

La visión artificial que deseamos generar estará compuesta por elementos artificiales dentro de un mundo tridimensional, cuyo sistema de coordenadas podemos observar en la figura 1. Como el objetivo de nuestro estudio es la simulación de la estereopsis, ubicaremos nuestros ojos en el semieje $-z$ a una distancia l del origen como puede apreciarse en la figura 2. La separación entre los ojos será aproximadamente de $d = 5$ cm. y la distancia l se estudiará con una variación entre 1 m. y 5 m. del origen. El movimiento de los objetos dentro de nuestro mundo artificial se hará a lo largo del eje x , permitiendo poder establecer unas bases para los cálculos posteriores y su generalización si así procede.

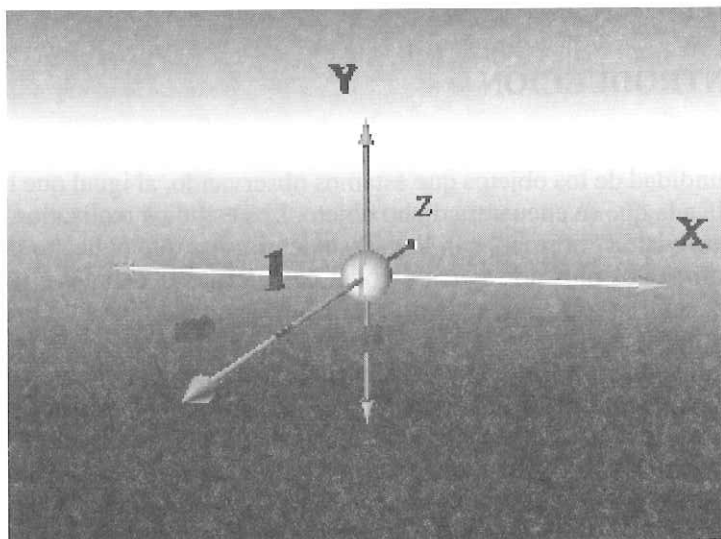


Figura 1. Sistema de coordenadas propuesto.

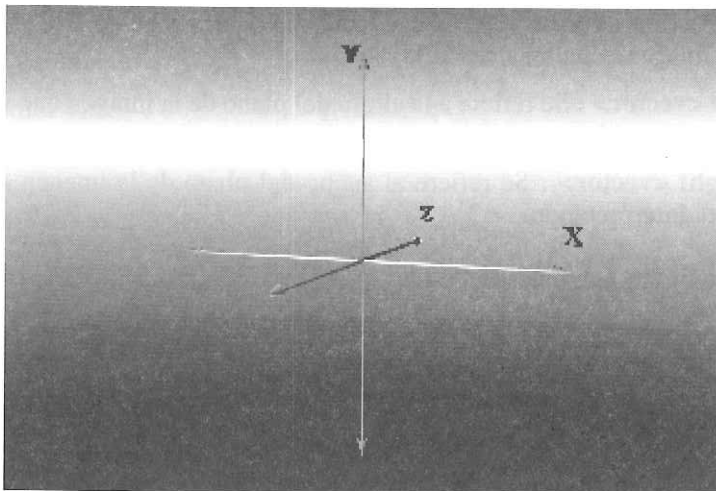


Figura 2. Situación de nuestros ojos dentro del eje de coordenadas planteado.

3. DESCRIPCIÓN DE LOS MEDIOS QUE SE VAN A UTILIZAR

Los medios que se han utilizado para la realización de la simulación son los siguientes:

- un ordenador personal.
- el programa trazador de rayos POV-Ray.

Pasamos a exponer la sintaxis del objeto *Cámara* de POV-Ray, que será sin lugar a dudas el centro de nuestra simulación (ver descripción 1).

```

camera {
location <VECTOR>           //lugar donde se ubica la cámara
look_at <VECTOR>           //lugar a donde se enfoca
right <VECTOR>              //ancho horizontal del plano de la imagen
up <VECTOR>                 //ancho vertical del plano de la imagen
direction <VECTOR>         //longitud focal
}

```

Descripción 1. Sintaxis y parámetros del objeto cámara.

De entre todos los parámetros disponibles para el objeto *Cámara* hemos destacado aquellos que realmente nos interesan para nuestra simulación:

- **location <vector>** : Es el lugar de ubicación de la cámara dentro de la escena tridimensional.
- **look_at <vector>** : Es el lugar dentro del mundo artificial al que se está mirando.

- **direction <vector>** : Es la distancia a la que se encuentra el plano de la imagen calculada desde la cámara.
- **up <vector>** : Se refiere a la altura del plano de la imagen que se está interpretando.
- **right <vector>** : Se refiere al ancho del plano de la imagen que se está interpretando.

En la figura 3 se muestran estos parámetros como función de la cámara.

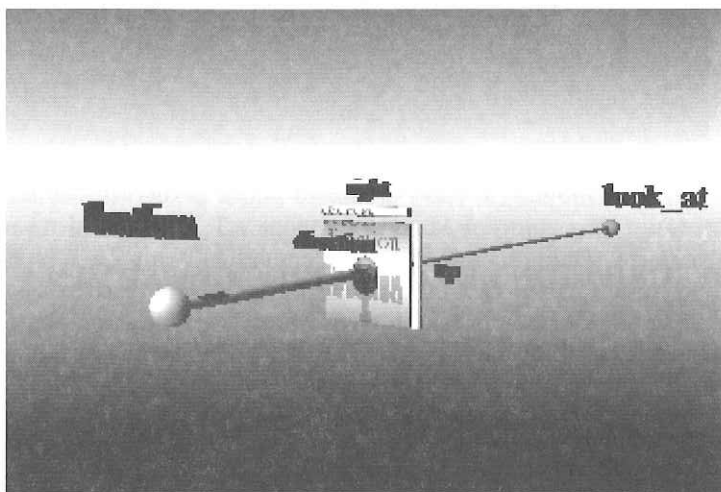


Figura 3. Parámetros de la cámara artificial.

Existe una diferencia básica entre el ojo real y la cámara en cuanto al plano donde se representa la imagen. El ojo lo hace en la zona amarilla y de manera inversa al objeto que está observando, mientras que la cámara lo hace en la parte anterior entre el punto donde se encuentra localizada la cámara y el objeto a observar. Para poder relacionar este comportamiento hemos igualado la distancia en el ojo desde la córnea hasta la zona amarilla con la distancia focal. Así pues:

$$\text{distancia focal} = \text{direction}$$

También hay que tener en cuenta que nuestra cámara es una cámara ideal y que no tiene ningún índice de refracción, por lo que hemos incorporado unos elementos translúcidos con un índice de refracción igual a los que se tiene en el ojo. Sin embargo, POV-Ray no puede procesar todos los medios del ojo humano con todos sus índices de refracción. Por ello, solamente hemos considerado los siguientes datos:

Humor acuoso con	IOR=1.33
Cristalino con	IOR=1.45
Humor vítreo	IOR=1.34

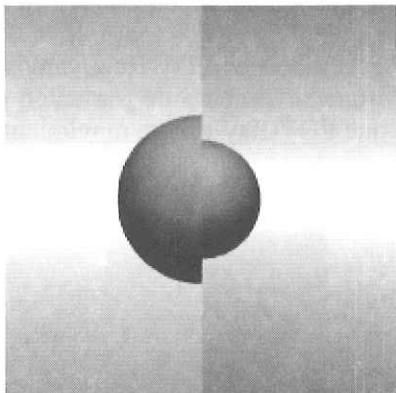


Figura 4. Diferencia entre la visión con y sin el IOR de la cámara.

Vamos ahora a incorporar la descripción del ojo artificial para que POV-Ray lo interprete.

```

/* Esta es una descripción de nuestro ojo artificial mirando hacia una esfera con
sus índices de refracción correspondientes para que se note la diferencia entre la
escena sin los índices de refracción solo hemos cubierto la zona izquierda de la
cámara con los medios , por lo que resaltará la diferencia */
#include "colors.inc"
#include "skies.inc"
#declare Punto_focal=<0,0,0>
#declare Distancia_origen=100 //distancia en centímetros
sky_sphere // fondo de la escena
(
  pigment
  { gradient y
    color_map { [0.01 color White][0.3 color Gray05]}
  }
)
camera( //nuestro ojo
location z*-Distancia_origen
up y*.5
right x*.5
direction z*1.5
look_at Punto_focal
)
box (<-4,-3,-98.3><0,3,-96.7> //humor vitreo
pigment (White filter 1)
finish (refraction 1 ior 1.3)
)
box (<-4,-3,-96.7><0,3,-93.6> //Cristalino
pigment (White filter 1)
finish (refraction 1 ior 1.45)
)
box (<-4,-3,-93.6><0,3,-95.9> //humor acuoso
pigment (White filter 1)
finish (refraction 1 ior 1.33)
)
light_source(<0,10,-15>color White)
sphere (<0,0,0>,5 pigment (color White))

```

4. GENERACIÓN DE ESCENAS TRIDIMENSIONALES EN MOVIMIENTO

Las escenas en movimiento que se van a generar se corresponden con una secuencia de fotogramas obtenidas a razón de 30 fotogramas por segundo. Este número de fotogramas simula una continuidad en las escenas de movimiento.

Como ejemplo de movimiento generado por POV-Ray, vamos a presentar una escena consistente en un movimiento de traslación de una esfera a lo largo del eje x. Para que POV-Ray genere movimiento, se usa la palabra clave **clock**

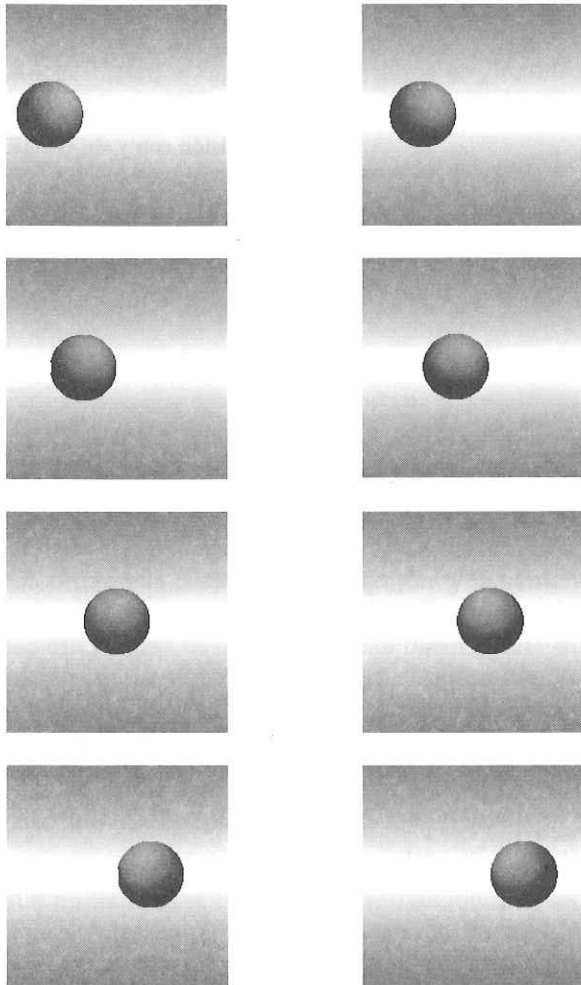


Figura 5. Secuencia de movimiento de una esfera a lo largo del eje x.

```

/*
Este pequeño fragmento de programa genera una escena de una esfera efectuando una
traslación en el eje de la X
*/

#include "colors.inc"
camera {
location <0,0,-20>
look_at 0
}
object {
sphere (0, 1)
pigment (color Blue)
translate x*-10
translate x*20*clock //efecto de movimiento horizontal
}
light_source ( <0, 20, -100> color White )

```

Descripción 3. Objeto trasladándose a lo largo del eje x.

La figura 5 muestra el objeto en movimiento en diferentes instantes de tiempo.

5. SIMULACIÓN DE LA VISIÓN ESTEREOSCÓPICA

La simulación de la visión estereoscópica consistirá en colocar dos "ojos" a una distancia entre sí de 5 centímetros y a una distancia de 1 metro del origen de coordenadas y estudiar la escena desde estos dos ojos.

<pre> #declare Sep_ojos=5 #declare Dist_origen=100 </pre>	
<pre> //....Ojo Izquierdo location <-x*Sep_ojos/2,0,-Dist_origen> //....Fin de la declaración </pre>	<pre> //....Ojo Derecho location <x*Sep_ojos/2,0,-Dist_origen> //....Fin de la declaración </pre>
<pre> sphere (<0,0,-30>,5 pigment (color White)) </pre>	<pre> sphere (<-10,0,0>,5 pigment (color White)) </pre>

Descripción 4. Declaración de los dos ojos.

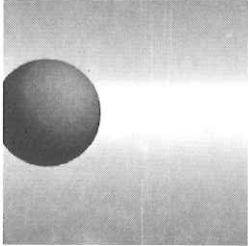
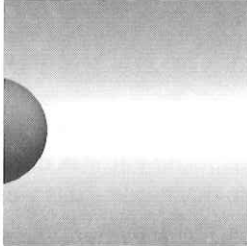
6. PROPOSICIÓN PARA EL CÁLCULO DE LA DISTANCIA Y LA PROFUNDIDAD DE UN OBJETO

Una vez visto todo el planteamiento y la descripción del funcionamiento de las cámaras u ojos, haremos tres proposiciones para el cálculo de la distancia y profundidad de un objeto. Este objeto estará en un punto fijo dentro del mundo artificial, para que a la hora de realizar los cálculos se puedan comparar los resultados obtenidos. Para nuestras proposiciones utilizaremos una figura de geometría básica para poder facilitar los cálculos, a saber una esfera. Estas tres proposiciones son las siguientes:

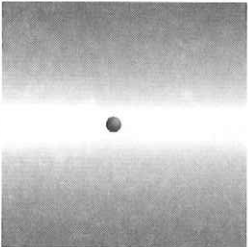
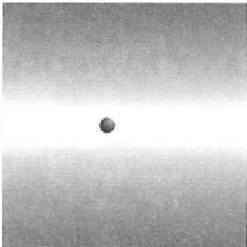
- a) Ojos enfocando hacia un punto fijo en el infinito.
- b) Ojos enfocando a un punto fijo cercano.
- c) Ojos enfocando al objeto.

a) Ojos enfocando hacia un punto fijo en el infinito.

En este caso, las cámaras estarán en el sistema de coordenadas propuesto, el punto focal será el infinito (teóricamente, ya que dentro de nuestro mundo artificial no podemos definir el infinito).

<pre>#declare Punto_focal=z*1000000 //"infinito"=10 Km. #declare Sep_ojos=5 //separación de los ojos en centímetros #declare Dist_origen=100 //distancia al origen en centímetros #declare esfera=sphere(<0,0,0>,5 pigment(color White))</pre>	
<pre>//..Declaración del ojo izquierdo . location <-x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>	<pre>//..Declaración del ojo derecho . location <x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>
	

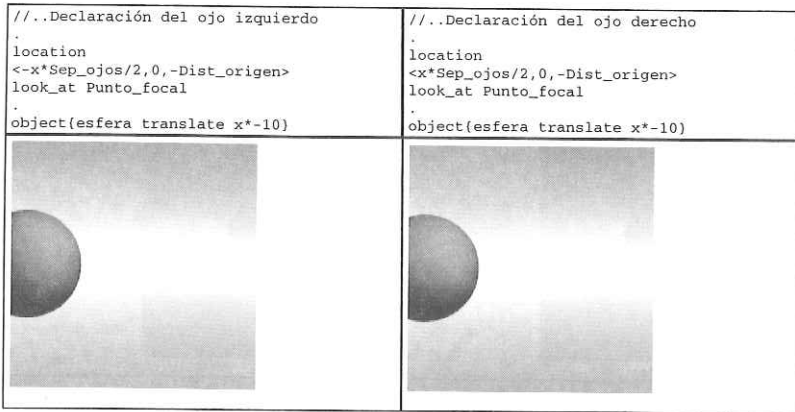
Descripción 5. Vista de una esfera a una distancia de un metro del origen fijando la vista en el infinito.

<pre>//..Declaración del ojo izquierdo . location <-x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>	
<pre>//..Declaración del ojo derecho . location <x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>	<pre>//..Declaración del ojo derecho . location <x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>
	

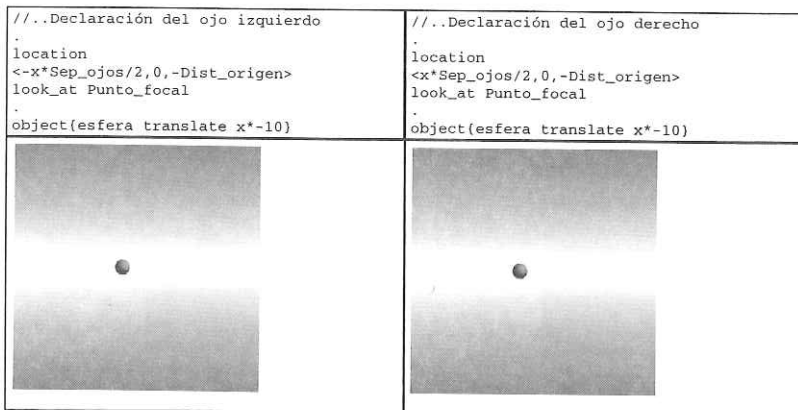
Descripción 6. Vista de una esfera a una distancia de cinco metros del origen fijando la vista en el infinito.

b) Ojos enfocando hacia un punto fijo cercano.

En este caso seguimos manteniendo las cámaras en el mismo sistema de coordenadas, pero enfocando a un punto cercano de nuestro sistema de coordenadas. En concreto, tomamos como punto focal el eje de coordenadas $\langle 0,0,0 \rangle$.



Descripción 7. Vista de una esfera a una distancia de un metro del origen fijando la vista en el centro de coordenadas.



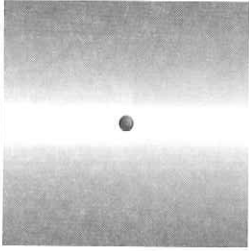
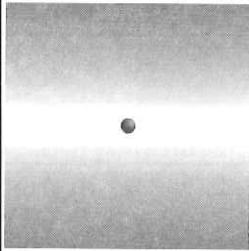
Descripción 8. Vista de una esfera a una distancia de cinco metros del origen fijando la vista en el centro de coordenadas.

c) Ojos enfocando al objeto.

En este caso planteamos que nuestras cámaras miren o enfoquen al objeto que se desea estudiar, manteniendo la posición de las cámaras. Aparece aquí la palabra clave **look_at**.

<pre>///..Declaración del ojo izquierdo . location <-x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>	<pre>///..Declaración del ojo derecho . location <x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>
	

Descripción 9. Vista de una esfera a una distancia de un metro del origen fijando la vista en la esfera.

<pre>///..Declaración del ojo izquierdo . location <-x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>	<pre>///..Declaración del ojo derecho . location <x*Sep_ojos/2,0,-Dist_origen> look_at Punto_focal . object(esfera translate x*-10)</pre>
	

Descripción 10. Vista de una esfera a una distancia de cinco metros del origen fijando la vista en la esfera.

BIBLIOGRAFÍA

- S.D. BARNARD & M.A. FISCHLER (1990). *Computational and Biological Models of Stereo Vision*. Proceedings, Image Understanding Workshop. Morgan-Kaufman.
- A. ENZMANN (1994). *Ray Tracing Worlds with Pov-Ray*.
- R. FAUCHER (1967). *Physique*. Classe de Première Sections C, D et E. Librairie A. Hatier. Paris.
- A.S. GLASSNER (1986). *An Introduction to Ray Tracing*. Andrew S. Glassner, ed.
- A.C. GUYTON (1988). *Tratado de fisiología médica*. 6ª edición. Interamericana /McGraw-Hill. Madrid.
- D. MARR (1985). *La visión*. Alianza Psicología ,12. Alianza Editorial, Madrid.
- C. YOUNG & D. WELLS (1994). *Ray Tracing Creations: Generate 3D Photorealistic Images on the PC*.