

Investigación bibliográfica 1:

# Estructuras de datos para gráficos en tres dimensiones, con énfasis en algoritmos de comparación.

Jean Carlos Chavarría Hughes

Universidad de Costa Rica

*jeancarlos.chavarria@ucr.ac.cr*

30 de septiembre de 2014

# Objetivos Específicos

- 1 Comparar las características de los formatos de gráficos en 3D.
- 2 Presentar un análisis de representaciones de mallas poligonales.
- 3 Describir los tipos de visualización científica de conjuntos de datos:

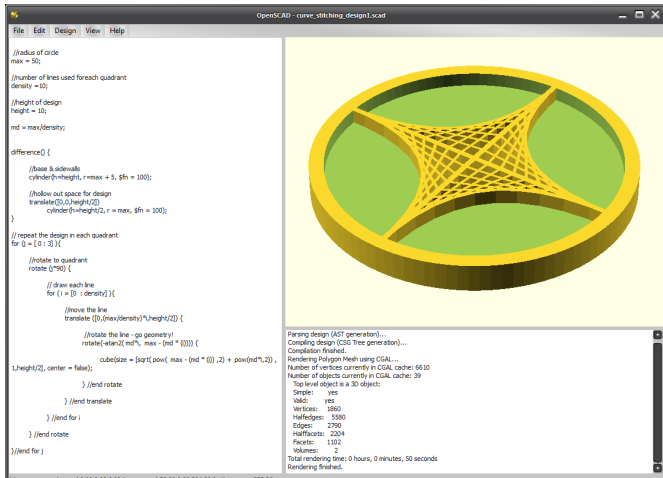
# Justificación

Why 3D objects?

- Vivimos en un mundo 3D.
- Análisis científico e industrial.
- Implementación de álgebra lineal a aplicaciones reales.

# Introducción

- Relación entre figuras, imágenes, objetos 3D y estructuras de datos.
- Visualización científica.



# Visualización Científica. Qué es y para que sirve

El mapeo de representaciones hechas por la computadora a representaciones preceptúales, con técnicas de codificación para maximizar el entendimiento con los seres humanos.

## Campos Escalares

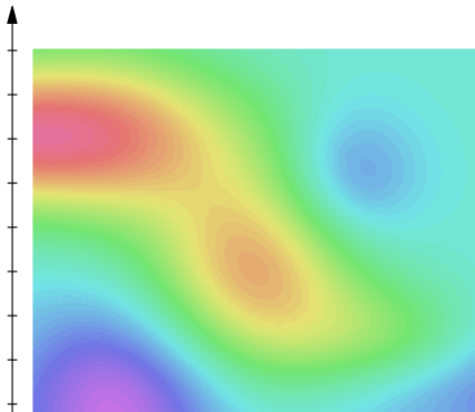
Se refiere a conjuntos de datos que se pueden distribuir en el tiempo o en posiciones del espacio. Temperatura, presión, resistencia, reflectividad, densidad.



# Qué es y para que sirve

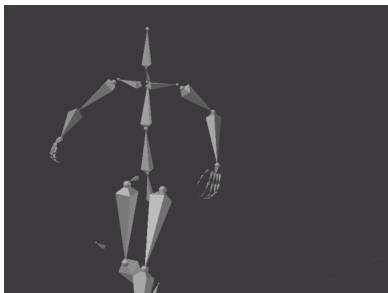
## Campos Tensoriales

Posee 9 componentes y se representa con una matriz  $3 \times 3$ . Ejemplos pueden ser el tensor de presión de materiales anisotrópicos.

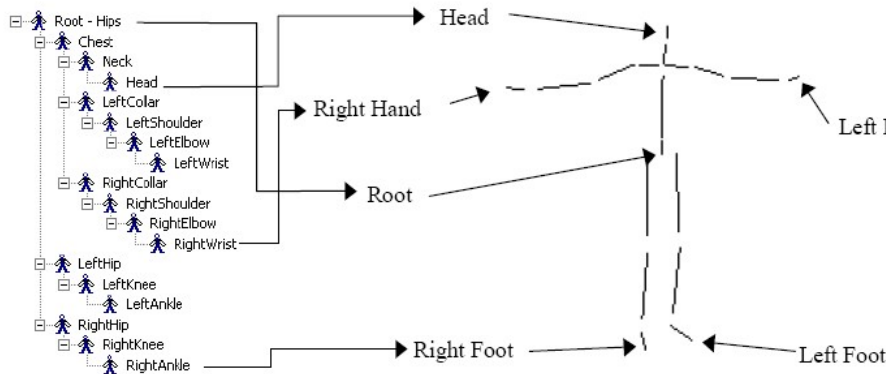


## Formatos: BVH

- Desarrollado por BioVision y enfocado en movimientos humanos.
- Dos partes principales, el encabezado **HIERARCHY** y el **MOTION**.
- La primera define los segmentos: **OFFSET, CHANNELS y JOINT**.
- La segunda define el número de *frames*, el *frame time*, y la información capturada de movimiento.

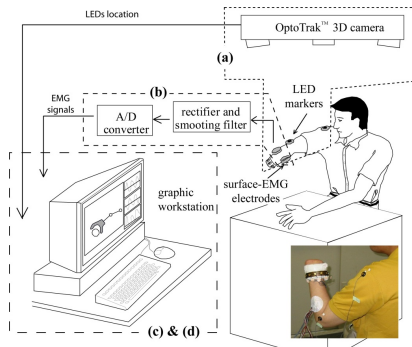






## Formatos: C3D

- Doctor Andrew Dainis en 1987, consiguió aceptación en laboratorios de biomedicina en Bethesda, Maryland.
- Preserva información que describe el tipo de diseño físico utilizado del laboratorio, tal como posiciones de platos, conjunto de marcas y tipos de canales empleados y EMG.
- Almacena información del paciente como nombre, edad, peso, longitud de piernas, etc.



## Formatos: FBX

- Desarrollado por Kaydara. Ahora dueño Autodesk desde 2006.
- Similar al BVH en tanto que tiene dos secciones importantes: el ROOT y los Hijos.
- Dirigido a las aplicaciones de simulación de objetos y animación en 3D, debido a que trabaja con muchas propiedades que sirven para caracterizar los objetos físicos.
- Datos: Límites de transformaciones, espacios y herencia, luz, cámara, null data, mesh data, armadura, textura, etc.



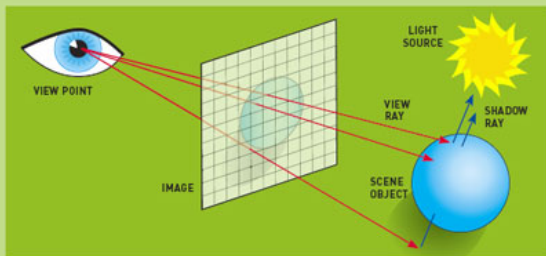
# Formatos: POV

- Vigente desde el año 1993.
- Utiliza la técnica llamada **Ray Tracing**.
- Permite la descripción de escenas de manera matemática y utiliza efectos como la reflexión, transparencia y luminosidad. Además, tiene la capacidad de crear imágenes muy realistas utilizando esta técnica y generar imágenes tipo TGA o GIF.
- La información almacenada en un POV es un conjunto de descriptores de escenas: cámaras, objetos y fuentes de luces.

# Ray Tracing

HOW IT WORKS

## Rendering Pixels via Ray Tracing



In the real world, light comes from a source and bounces off surfaces. Some of the photons that reach your eye hit your retina, allowing you to see. Ray tracing works the other way, by casting rays back from the observer's viewpoint to determine which pixels it should render.

# Polygon Mesh: Elementos

## Caras

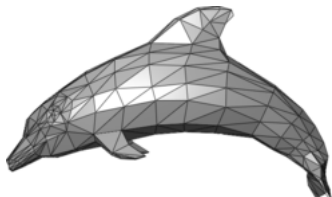
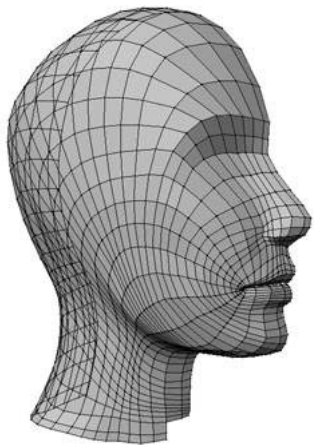
Se refiere a un conjunto cercano de bordes que conforman un polígono establecido, generalmente un triángulo pero también puede ser un cuadrilátero u otro.

## Vértices

Contiene coordenadas en 3D de cada uno de los vértices que conforman los polígonos limitadores de la superficie. Puede contener información como color, vector normal y textura.

## Bordes

Contiene definiciones de la conexión de cada borde en términos de índices de nodos y especifica las conexiones de vértices.



# Polygon Mesh: Representaciones

## Cara Vértice

Representa un conjunto de caras y vértices y típicamente es aceptado por los hardwards de procesamiento gráfico actual debido a su gran precisión y rendimiento.

## Face-Vertex Meshes

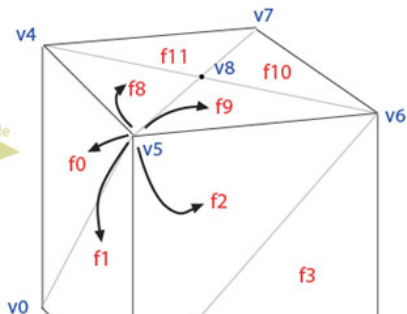
Face List

f0	v0 v4 v5
f1	v0 v5 v1
f2	v1 v5 v6
f3	v1 v6 v2
f4	v2 v6 v7
f5	v2 v7 v3
f6	v3 v7 v4
f7	v3 v4 v0
f8	v8 v5 v4
f9	v8 v6 v5
f10	v8 v7 v6
f11	v8 v4 v7

Vertex List

v0	0,0,0	f0 f1 f12 f15 f7
v1	1,0,0	f2 f3 f13 f12 f1
v2	1,1,0	f4 f5 f14 f13 f3
v3	0,1,0	f6 f7 f15 f14 f5
v4	0,0,1	f6 f7 f0 f8 f11
v5	1,0,1	f0 f1 f2 f9 f8
v6	1,1,1	f2 f3 f4 f10 f9
v7	0,1,1	f4 f5 f6 f11 f10
v8	.5,.5,0	f8 f9 f10 f11
v9	.5,.5,1	f12 f13 f14 f15

example





# Polygon Mesh: Representaciones

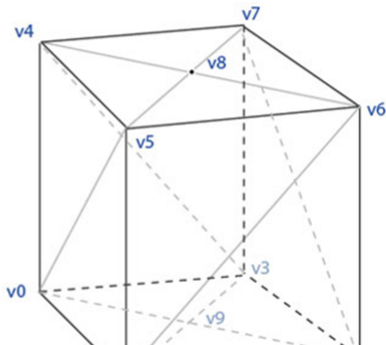
## Vértice Vértice

Representa a objetos conjuntos de vértices conectados con vértices y es la manera más simple y ligera, pero no la más precisa.

## Vertex-Vertex Meshes (VV)

Vertex List

v0	0,0,0	v1 v5 v4 v3 v9
v1	1,0,0	v2 v6 v5 v0 v9
v2	1,1,0	v3 v7 v6 v1 v9
v3	0,1,0	v2 v6 v7 v4 v9
v4	0,0,1	v5 v0 v3 v7 v8
v5	1,0,1	v6 v1 v0 v4 v8
v6	1,1,1	v7 v2 v1 v5 v8
v7	0,1,1	v4 v3 v2 v6 v8
v8	5,5,1	v4 v5 v6 v7
v9	5,5,1	v4 v5 v6 v7



# Polygon Mesh: Representaciones

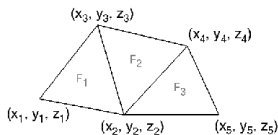
## Eje alado

Baumgart en 1975. Provee información sobre los tres elementos fundamentales: caras, bordes y vértices.

### Winged Edge



- Example:



VERTEX TABLE					
v <sub>1</sub>	x <sub>1</sub>	y <sub>1</sub>	z <sub>1</sub>	e <sub>1</sub>	
v <sub>2</sub>	x <sub>2</sub>	y <sub>2</sub>	z <sub>2</sub>	e <sub>6</sub>	
v <sub>3</sub>	x <sub>3</sub>	y <sub>3</sub>	z <sub>3</sub>	e <sub>3</sub>	
v <sub>4</sub>	x <sub>4</sub>	y <sub>4</sub>	z <sub>4</sub>	e <sub>5</sub>	
v <sub>5</sub>	x <sub>5</sub>	y <sub>5</sub>	z <sub>5</sub>	e <sub>6</sub>	

EDGE TABLE									
				11	12	21	22		
e <sub>1</sub>	v <sub>1</sub>	v <sub>3</sub>	F <sub>1</sub>	e <sub>2</sub>	e <sub>2</sub>	e <sub>4</sub>	e <sub>3</sub>		
e <sub>2</sub>	v <sub>1</sub>	v <sub>2</sub>	F <sub>1</sub>	e <sub>1</sub>	e <sub>1</sub>	e <sub>3</sub>	e <sub>6</sub>		
e <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>	F <sub>1</sub> F <sub>2</sub>	e <sub>2</sub>	e <sub>5</sub>	e <sub>1</sub>	e <sub>4</sub>		
e <sub>4</sub>	v <sub>3</sub>	v <sub>4</sub>	F <sub>2</sub>	e <sub>1</sub>	e <sub>3</sub>	e <sub>7</sub>	e <sub>5</sub>		
e <sub>5</sub>	v <sub>2</sub>	v <sub>4</sub>	F <sub>2</sub> F <sub>3</sub>	e <sub>3</sub>	e <sub>6</sub>	e <sub>4</sub>	e <sub>7</sub>		
e <sub>6</sub>	v <sub>2</sub>	v <sub>5</sub>	F <sub>3</sub>	e <sub>5</sub>	e <sub>2</sub>	e <sub>7</sub>	e <sub>7</sub>		
e <sub>7</sub>	v <sub>4</sub>	v <sub>5</sub>	F <sub>3</sub>	e <sub>4</sub>	e <sub>5</sub>	e <sub>6</sub>	e <sub>6</sub>		

FACE TABLE		
F <sub>1</sub>	e <sub>1</sub>	
F <sub>2</sub>	e <sub>3</sub>	
F <sub>3</sub>	e <sub>5</sub>	

# Referencias

- Foundations of Computer Graphics: Berkeley.
- Geometric Modeling for Computer Graphics: Princeton.
- Computer Graphics: MIT.
- Solid Modeling: Berkeley.

# Thanks!