Representação de Objetos 3D

Prof. Antonio L. Apolinário Junior
Estagiária Docente: Rafaela Alcantara

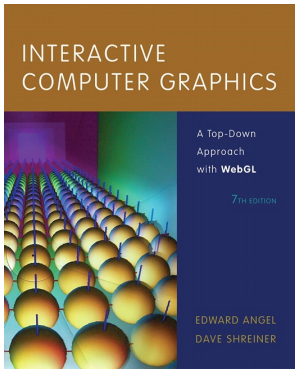
UFBA/IM/DCC/BCC - 2018.1

Roteiro

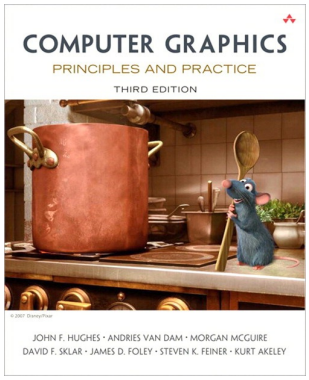
- Representação de Objetos Gráficos
- Representação de formas 3D
  - Esquema de representação por fronteira
- Representação de Objetos em Three.JS/WebGL

Leitura de referencia

- Capitulo 3 (até a seção 3.7)
  Interactive Computer Graphics -
  A top-down approach with OpenGL
  7th Edition
  Angel, Edward.
  Addison-Wesley. 2012.

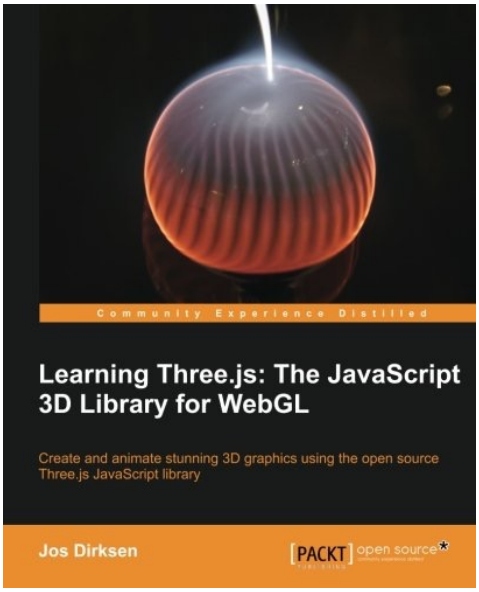


- Capítulos 8
  Computer Graphics : Principles and Practice
  Third Edition in C
  John F. Hughes / Andries van Dam
  Morgan McGuire / David F. Sklar
  James D. Foley / Steven K. Feiner
  Addison-Weslley. 2013.



Leitura de referencia

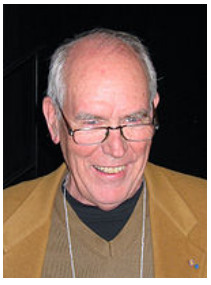
- Capitulo 5, 6 e 8
  Learning Three.js: The JavaScript
  3D Library for WebGL
  Jos Dirksen
  Packt Publishing - 2013.



Modelos de representação de Objetos Gráficos

História

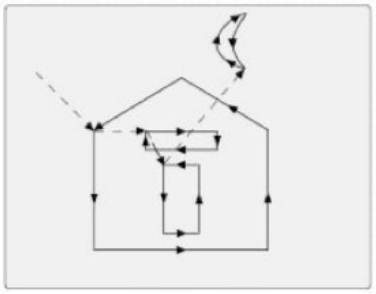
- 1963 - Sketchpad



- Ivan Sutherland, Sketchpad, A Man-Machine Graphical Communication System. Ph.D. Thesis, Massachusetts Institute of Technology, 1963.

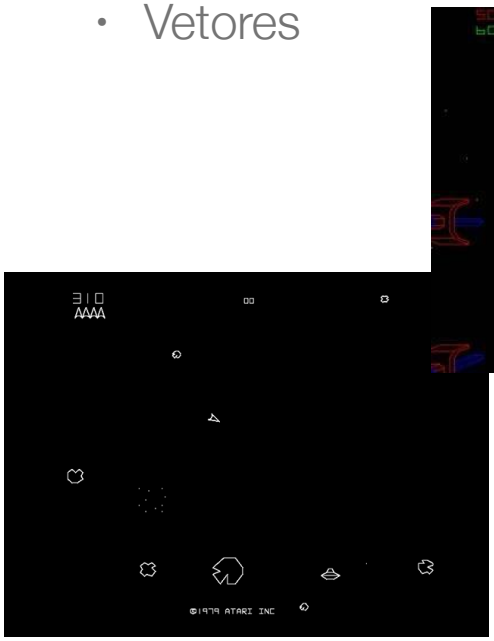
Representação Vetorial

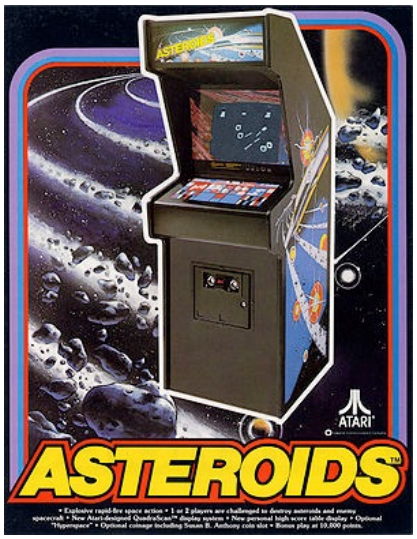
- Descreve os objeto a partir de primitivas geométricas simples



Representação Vetorial

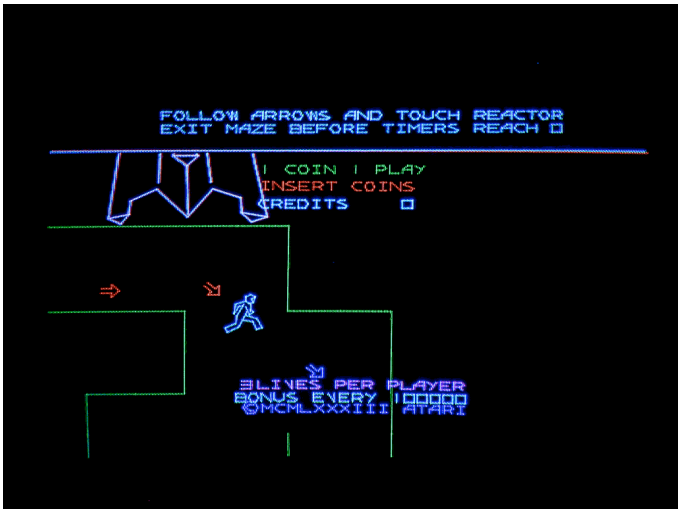
- Descreve os objeto a partir de primitivas geométricas simples
  - Vetores

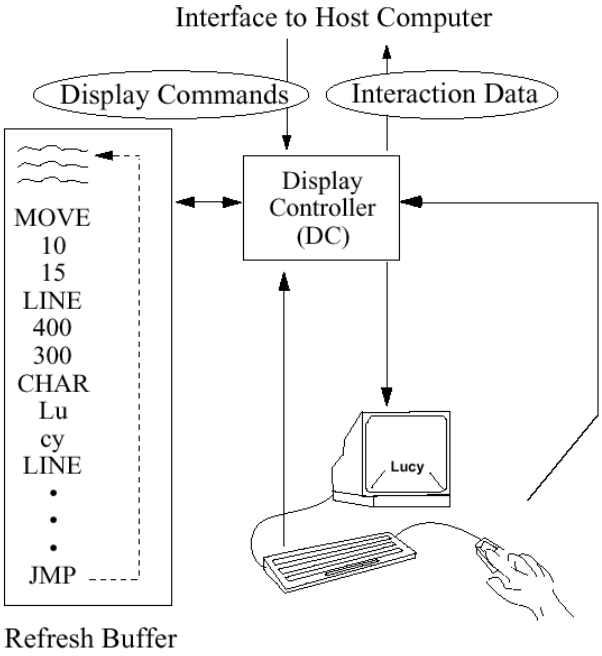




Representação Vetorial

- Uma imagem é gerada a partir de um conjunto de comandos de desenho de primitivas (vetoriais)

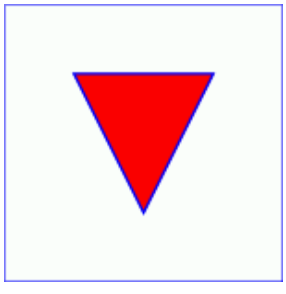




Representação Vetorial

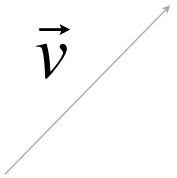
- Utilizado até hoje
  - Exemplo: formato SVG

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102//EN"
  "http://www.w3.org/TR/2000/CR-SVG-20001102/DTD/
svg-20001102.dtd">
<svg width="4cm" height="4cm" viewBox="0 0 400 400">
  <title>Example triangle01- simple example of a 'path'</title>
  <desc>A path that draws a rectangle</desc>
  <rect x="1" y="1" width="398" height="398"
        style="fill:none; stroke:blue"/>
  <path d="M 100 100 L 300 100 L 200 300 z"
        style="fill:red; stroke:blue; stroke-width:3"/>
</svg>
```



Representação Vetorial

- Como representar vetores?
  - Direção
  - Sentido
  - Tamanho



- Como descrever um vetor matemáticamente?
  - Como um segmento de reta

Representação Vetorial

- Como um segmento de reta
  - Sistema de referencia

    $P_i = x_i.\vec{i} + y_i.\vec{j}$
  - Dois pontos

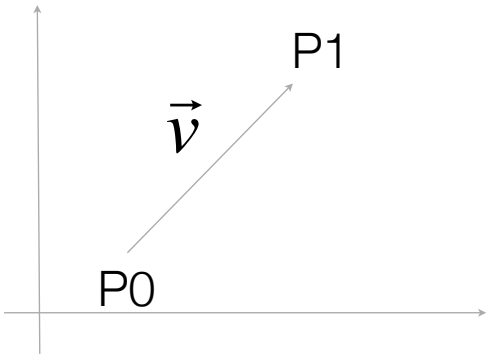
    $\vec{v} = (P_1 - P_0)$
  - Pontos sobre o vetor

    $P = P_0 + t.\vec{v}$

    $= P_0 + t.(P_1 - P_0)$

    $= (1-t).P_0 + t.P_1$

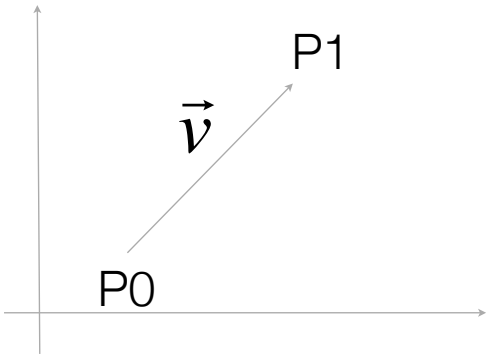
    $t \in [0,1]$



Representação Vetorial

- Algoritmo para desenhar um vetor:

```
registro Ponto { real x, y };
Algoritmo DesenhaVetor(Ponto P0,
Ponto P1, real delta)
  Inicio
    move(P0);
    Para t=0.0 até 1.0 passo delta
      Ponto P=(1-t)*P0 + t*P1;
      desenha(P);
    fim-para;
  fim-Algoritmo.
```



Representação Vetorial

- Redimensionamento:
  - Trivial
  - Recalculo com novos limites

Representação Vetorial

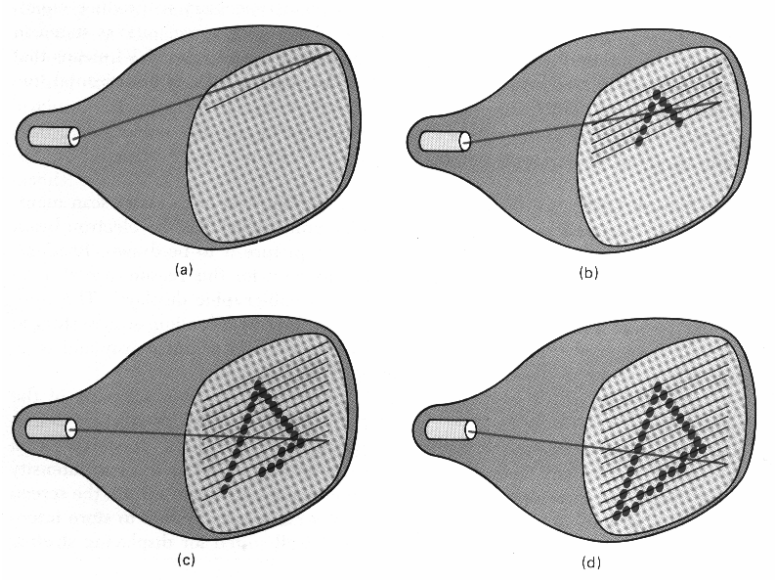
- Representação baseada em primitivas geométricas simples como:
  - Linhas
  - Polígonos
  - Poliedros
- Associa atributos a cada primitiva
  - Cor
  - material
- Armazena os dados necessários a representação matemática de cada primitiva

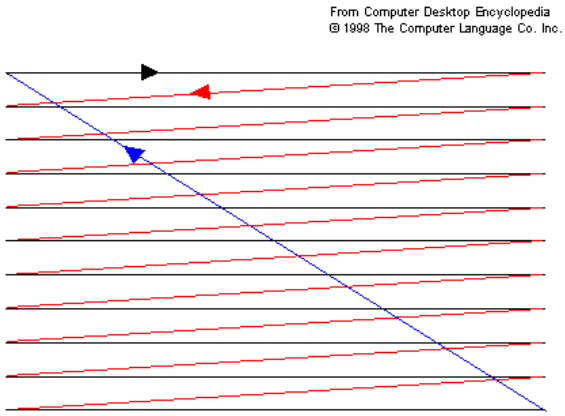
Representação Vetorial

- Precisão de um modelo vetorial:
  - “Infinita”
    - Limitada pela precisão da máquina
- Mudança de resolução:
  - Trivial
- Custo de Armazenamento
  - Relativamente baixo
    - apenas as informações para construção da primitiva geométrica

Representação Matricial

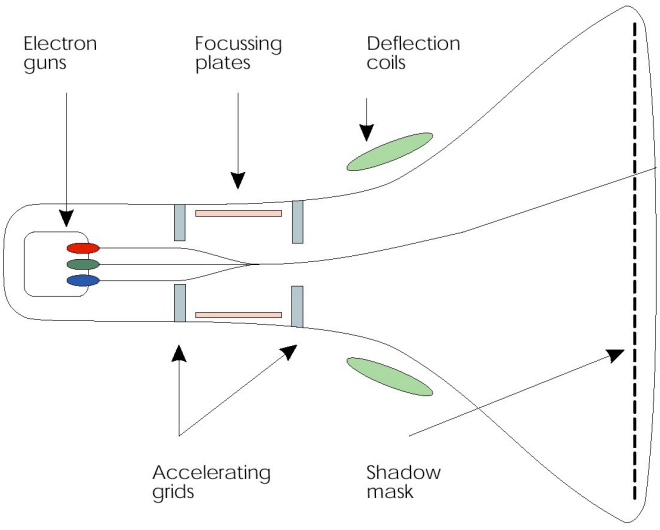
- Tecnologia vinda da TV
  - Tubo de Raios Catódicos (CRT)

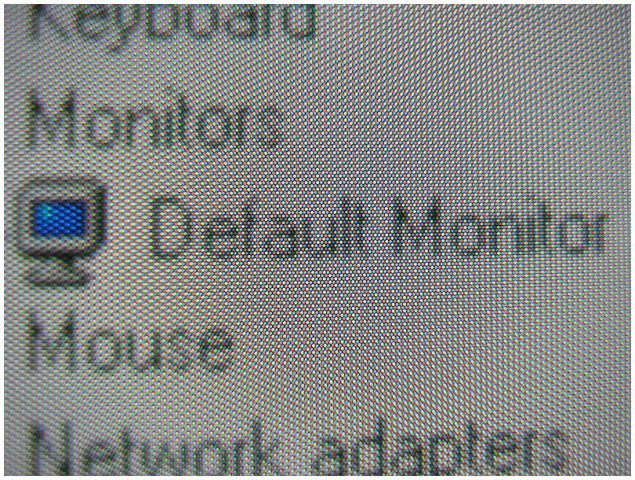




Representação Matricial

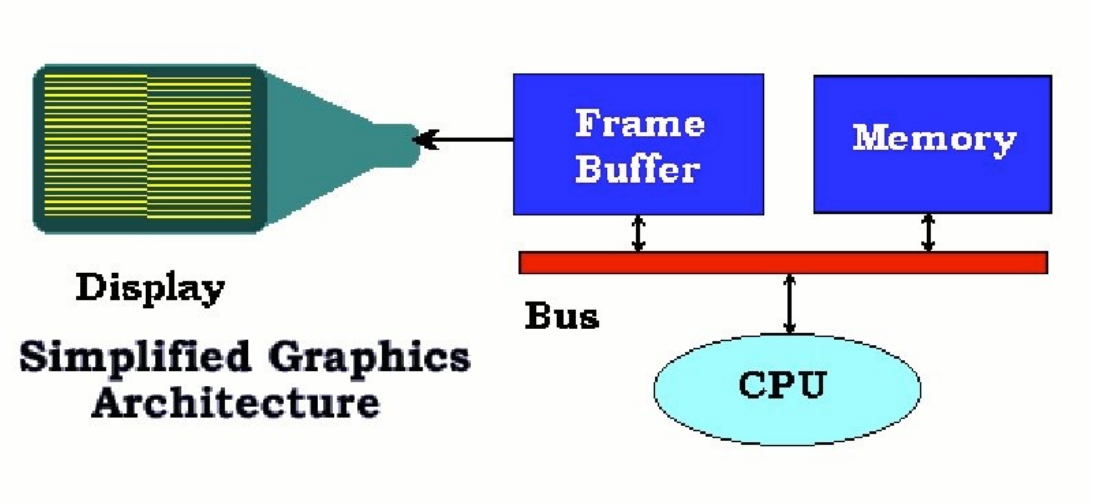
- Tecnologia vinda da TV
  - Tubo de Raios Catódicos (CRT)





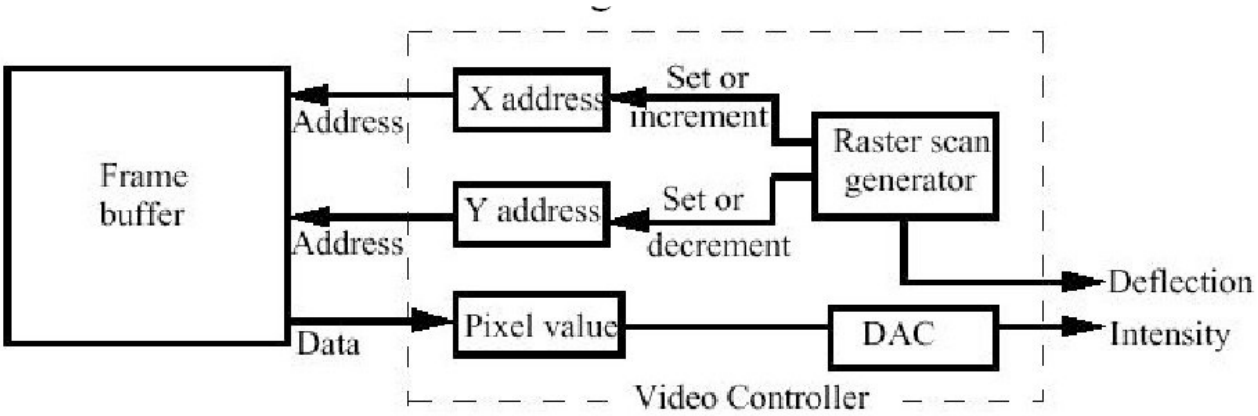
Representação Matricial

- Processo de varredura para a geração da imagem
  - Armazenamento da informação em cada pixel (picture element)



Representação Matricial

- Arquitetura de controle simples



Representação Matricial

- Gerar uma representação matricial envolve:
  - Amostragem
  - Reconstrução

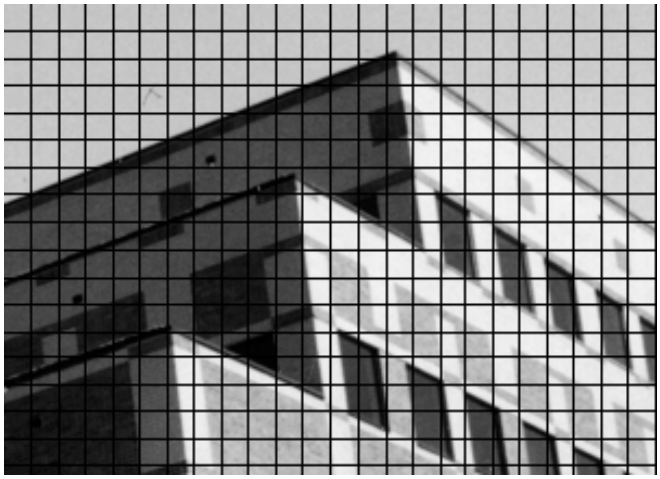
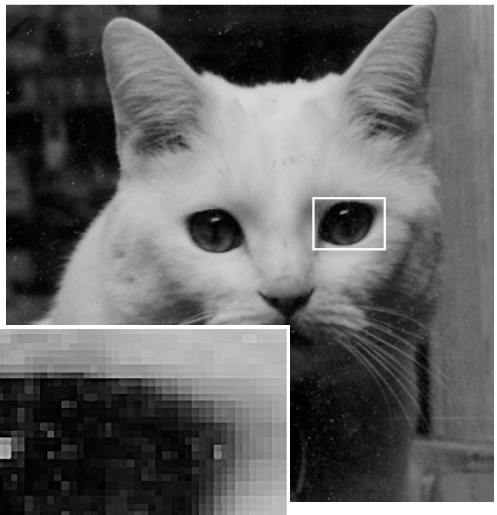


Imagem Digital

- Conjunto de amostras discretas
  - ***Pixel - picture element***
  - Cada pixel possui :
    - Cor
    - Opacidade
    - Outros atributos
  - Natureza Discreta



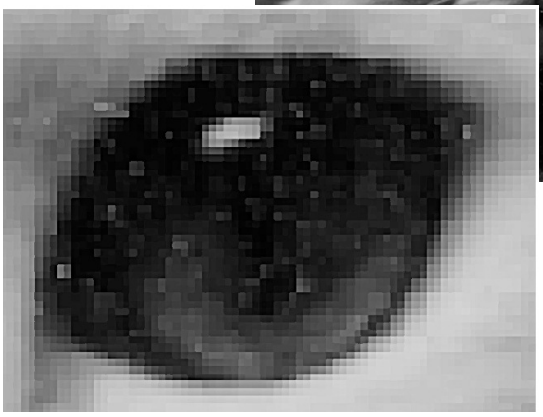


Imagem Digital

- Matriz de intensidades
  - Amostras

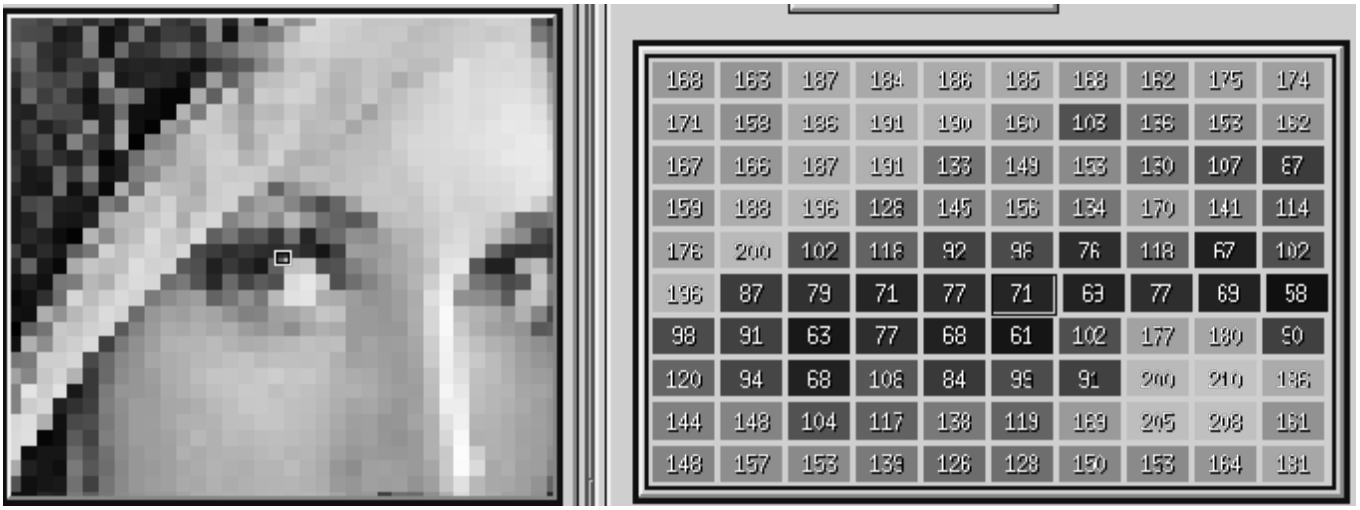
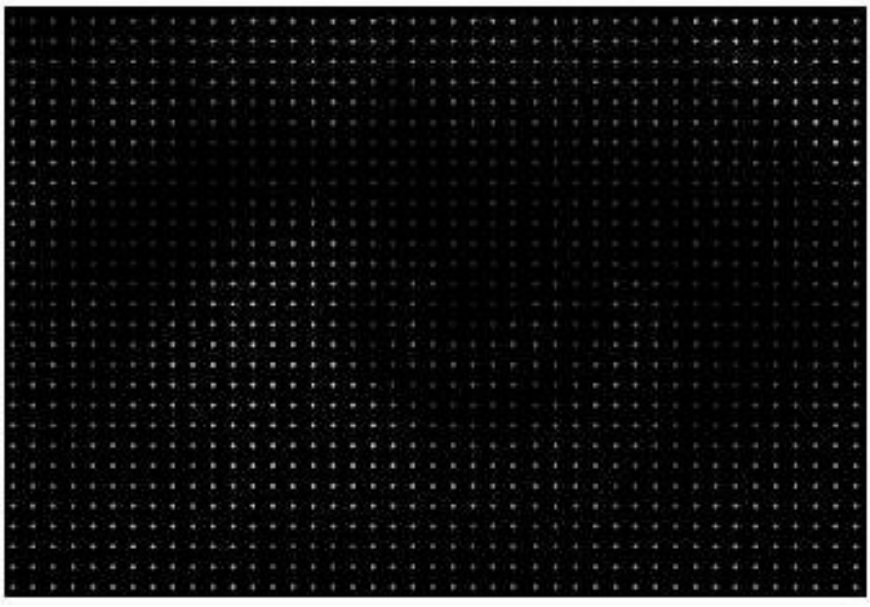


Imagem Digital

- Representação Discreta



Representação Matricial

- Gerar uma representação matricial envolve:
  - Amostragem
  - Reconstrução

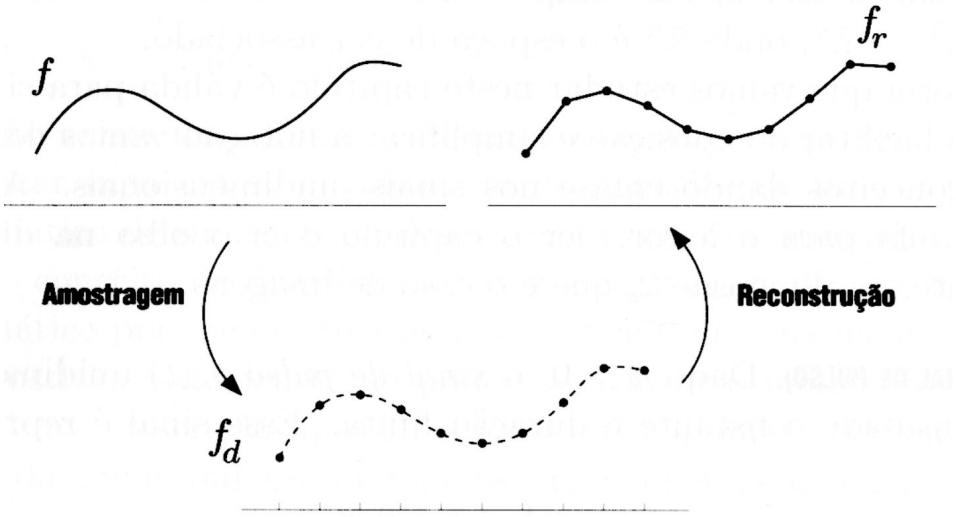


Imagem Digital

- Reconstrução
  - Interpolação

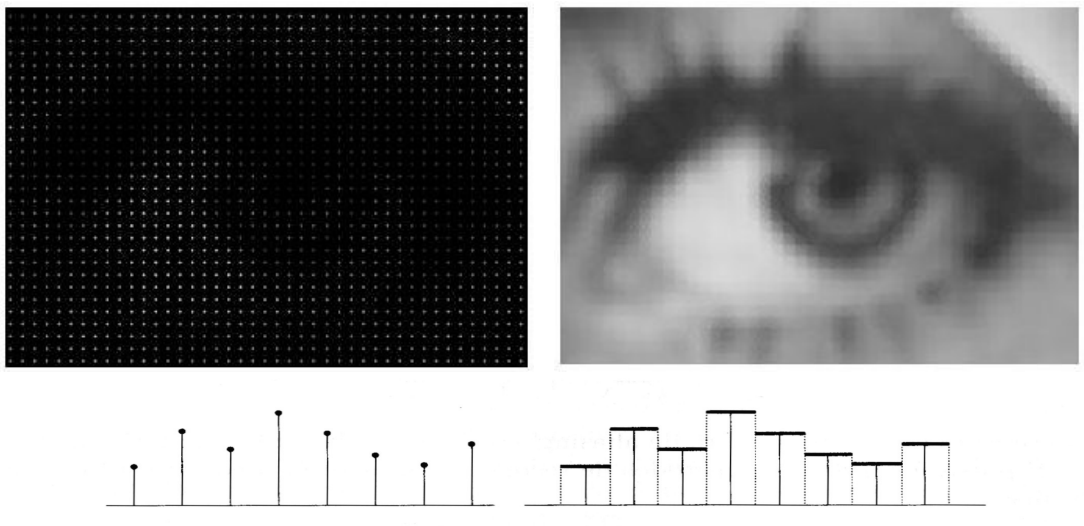
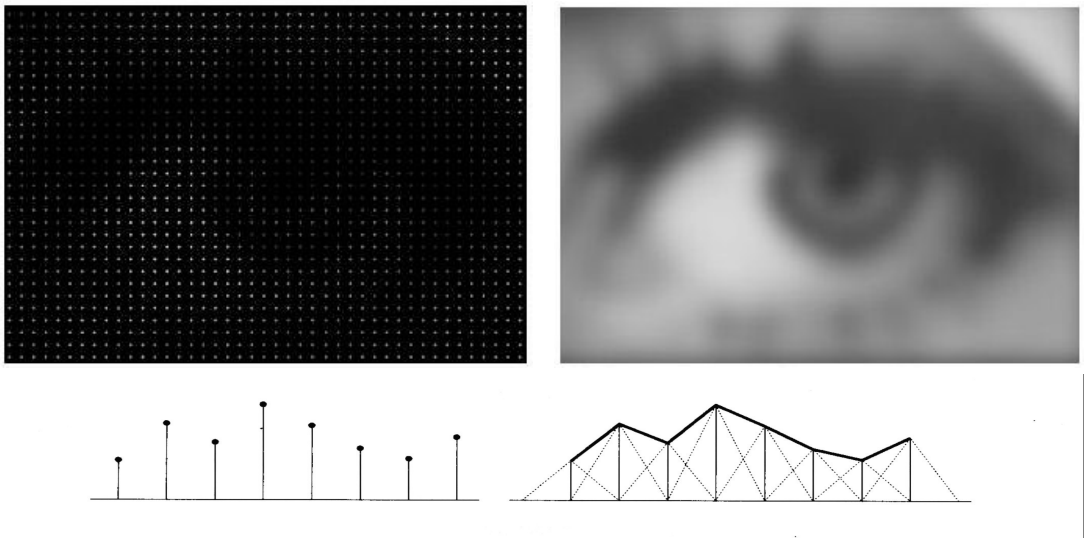


Imagem Digital

- Reconstrução
  - Interpolação

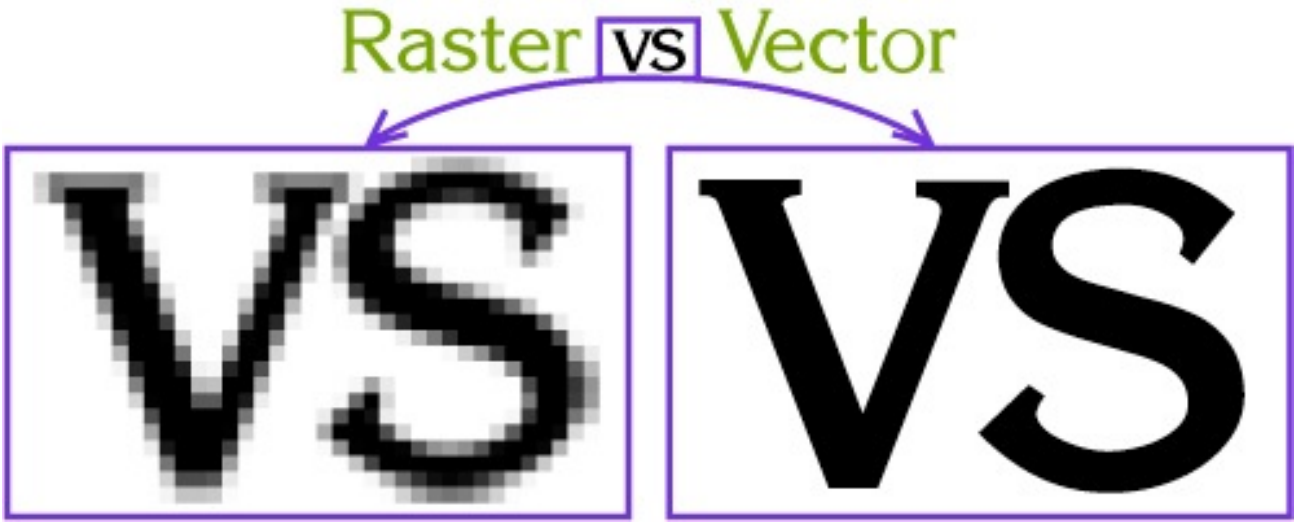


Representação Matricial

- Baseada em um conjunto de elementos discreto
  - Pixels
- Precisão
  - Limitada pela resolução da amostragem
- Mudança de resolução
  - Processo de reconstrução
- Custo de Armazenamento
  - Relativamente alto
  - Da ordem do número de pixels e de sua profundidade

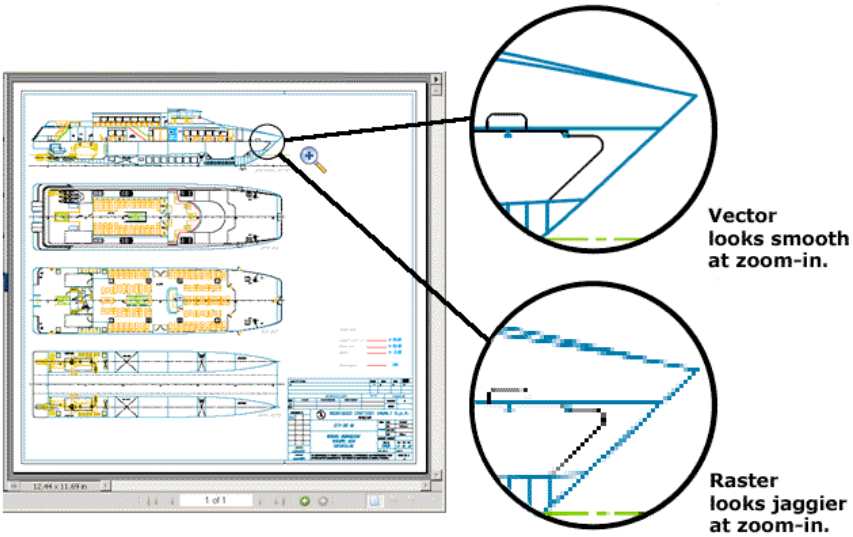
Vetorial x Matricial

- Representação contínua x representação discreta



Vetorial x Matricial

- Mudança de escala



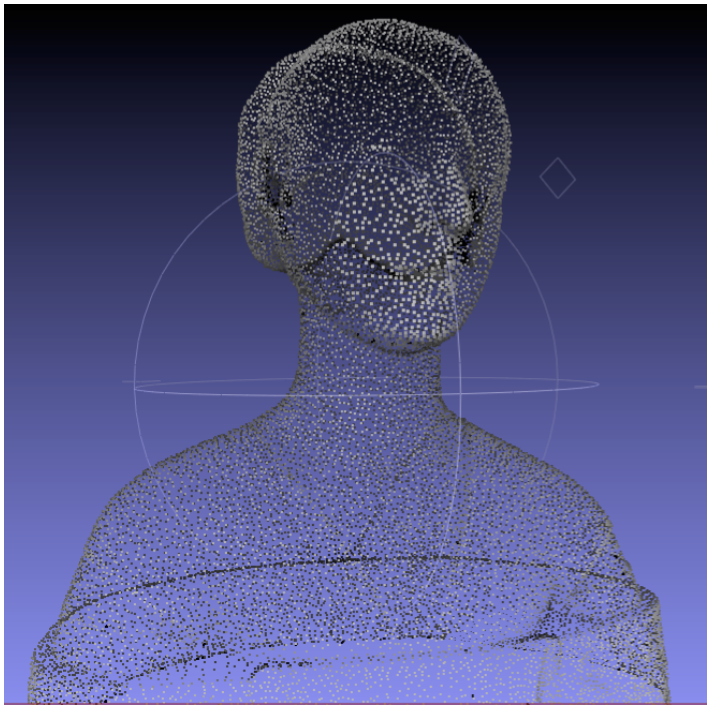
Vetorial x Matricial

- Representação Discreta
  - Imagem digital
    - Armazena uma matriz de pixels (amostras)
    - Cada pixel contem uma informação de intensidade
  - Formatos: BMP, GIF, JPEG, TIFF, PNG, PPM,....
- Representação Contínua
  - Linhas Poligonais e Curvas
    - Armazena pontos de controle
  - Formatos: SVG, CDR, 3DS/MAX, OBJ, PDF, ...

Representação de formas 3D

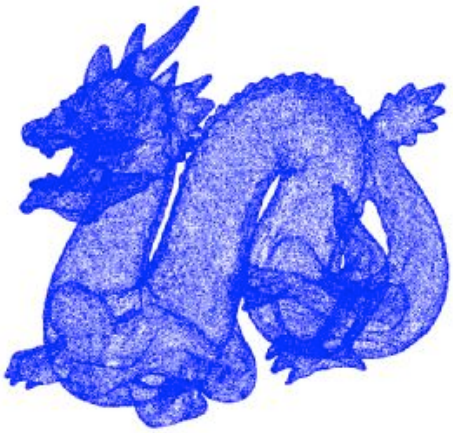
Esquema de Representação

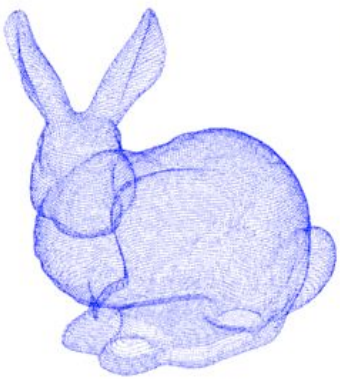
- Como desenhar objetos 3D?
  - Pontos?



Esquema de Representação

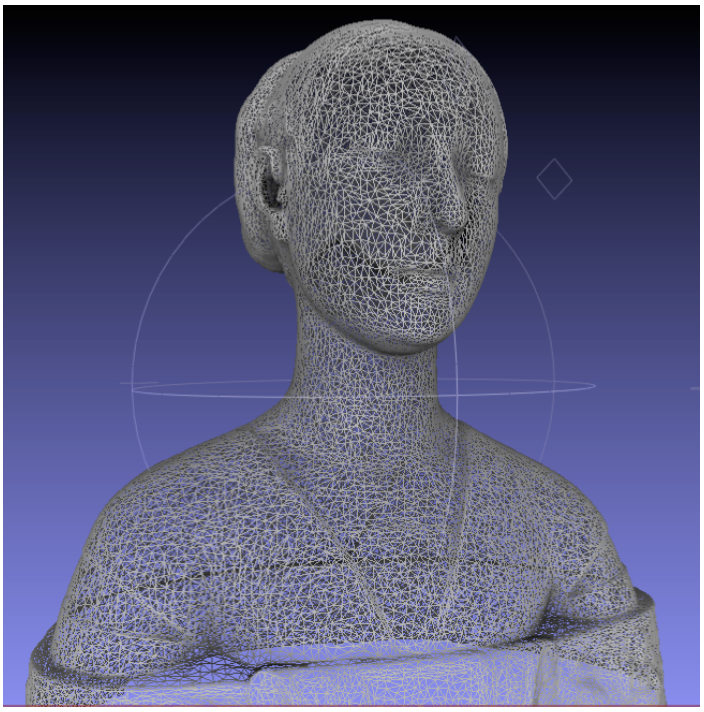
- Como desenhar objetos 3D?
  - Pontos?
    - Densidade <=> Qualidade
    - Fornece pouca informação forma do objeto





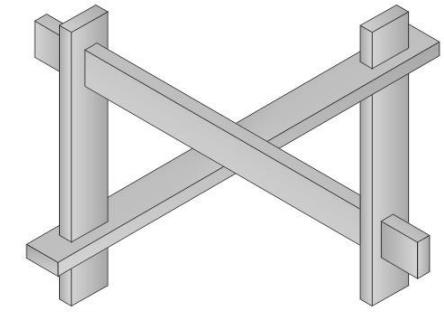
Esquema de Representação

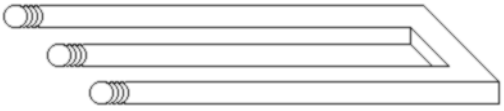
- Como desenhar objetos 3D?
  - Linhas?

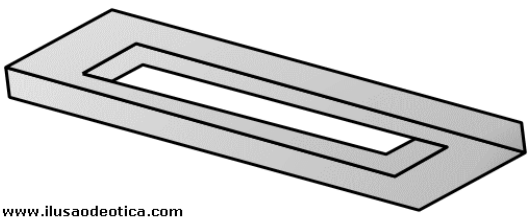


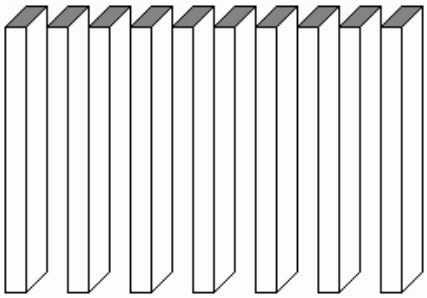
Esquema de Representação

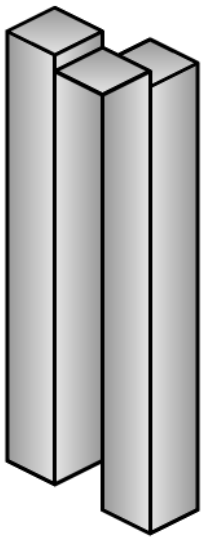
- Como garantir consistência dos objetos formados apenas por linhas?





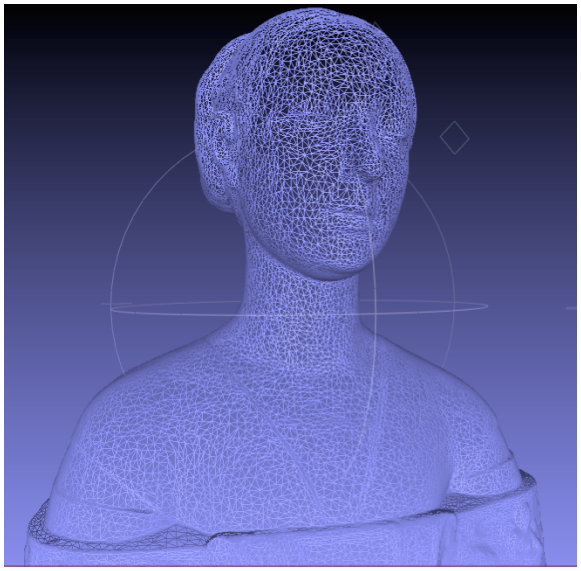






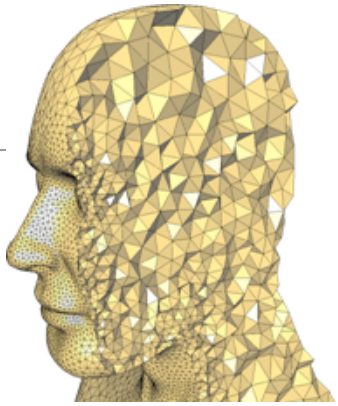
Esquema de Representação

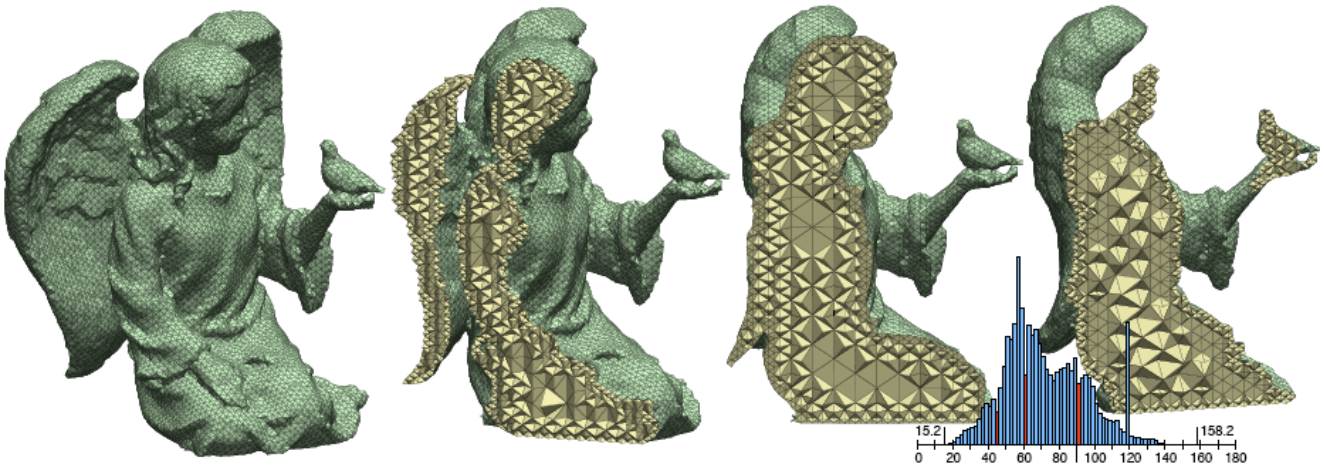
- Como desenhar objetos 3D?
  - Polígonos?



Esquema de Representação

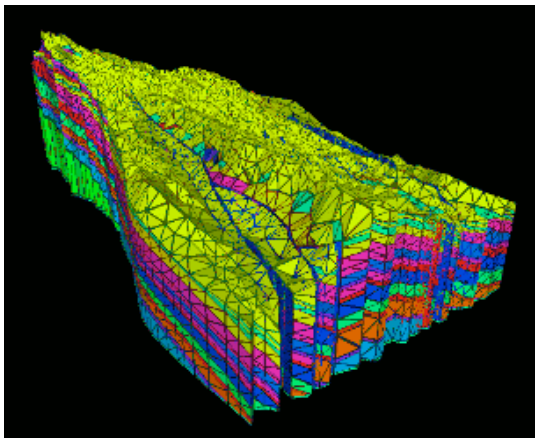
- Como desenhar objetos 3D?
  - Poliedros?

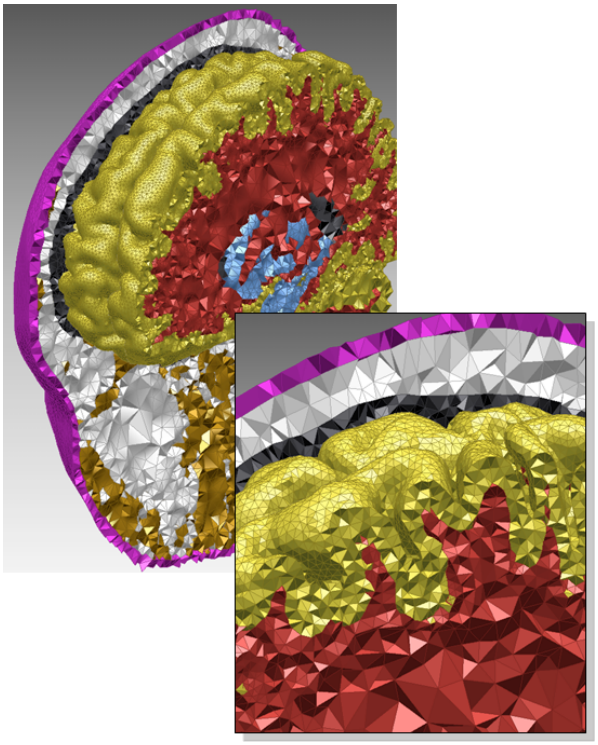




Esquema de Representação

- Como desenhar objetos 3D?
  - Poliedros?
    - Permite calcular propriedades físicas do sólido

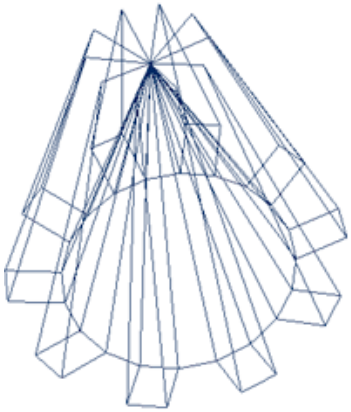


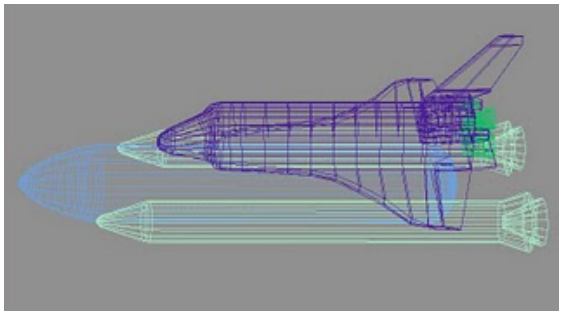


Esquema de Representação Por Fronteira

Wireframe

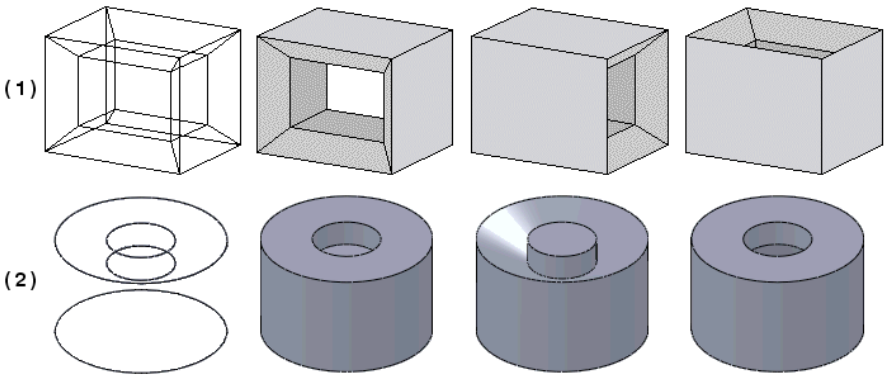
- Primeiro tipo de representação para desenho de modelos 3D
  - Simples
  - Não são representadas as relações topológicas entre vértices e arestas





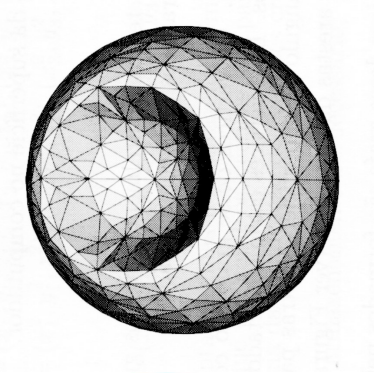
Wireframe

- Ambiguidade
  - Visual e de representação



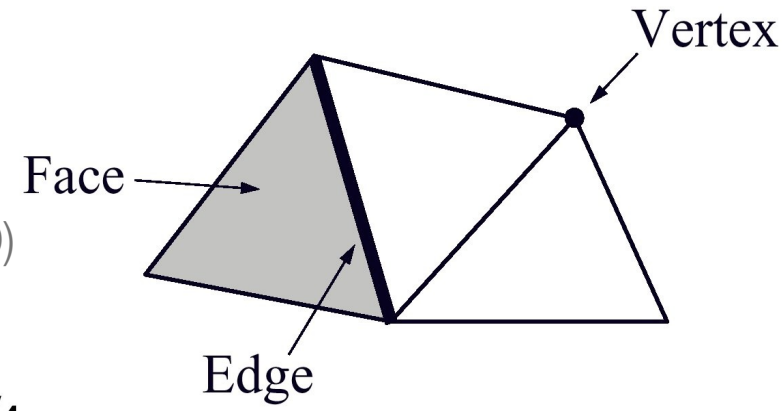
Representação Por Fronteira

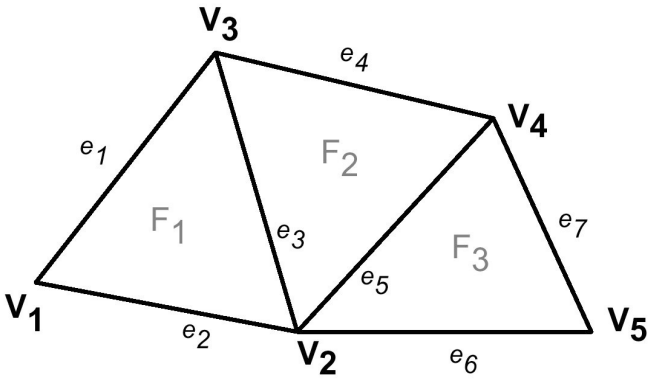
- B-Rep (Boundary Representation)
  - Objeto é descrito pela sua fronteira
    - Geometria
      - Amostras (pontos)
    - Topologia
      - Conectividade entre as amostras



Representação Por Fronteira

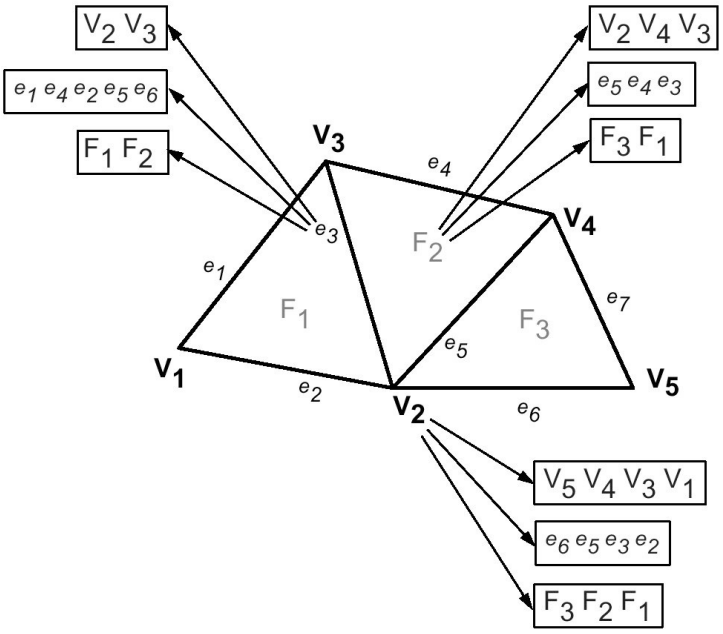
- Elementos básicos:
  - Vértices (Pontos 0D)
  - Arestas (Linhas 1D)
  - Faces (Polígonos 2D)

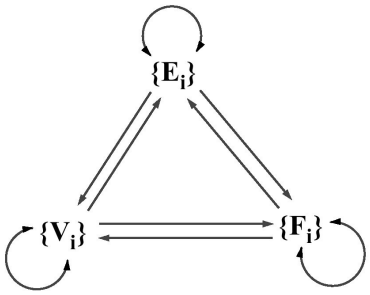




Representação Por Fronteira

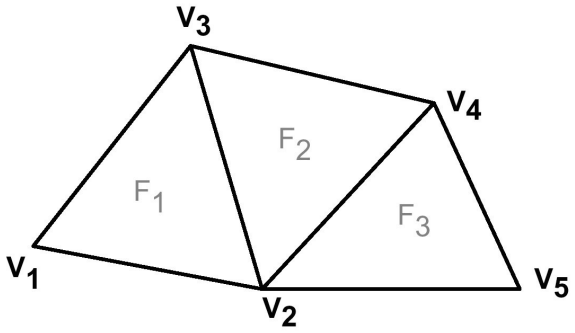
- Representar todas as relações topológicas
  - Rapidez
  - Estrutura + pesada
  - Custo de armazenam





Representação Por Fronteira

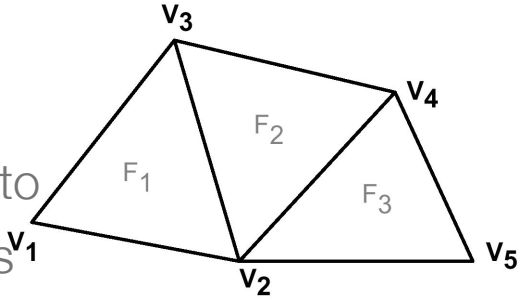
- Baseada em faces
  - Simples
  - Redundância
  - Fornece relações topológicas básicas



FACE TABLE	
F_1	(x_1, y_1, z_1) (x_2, y_2, z_2) (x_3, y_3, z_3)
F_2	(x_2, y_2, z_2) (x_4, y_4, z_4) (x_3, y_3, z_3)
F_3	(x_2, y_2, z_2) (x_5, y_5, z_5) (x_4, y_4, z_4)

Representação Por Fronteira

- Representação Por Fronteira
  - Baseada em vértices
    - Simples
    - Baixo custo de armazenamento
    - Não fornece todas as relações topológicas

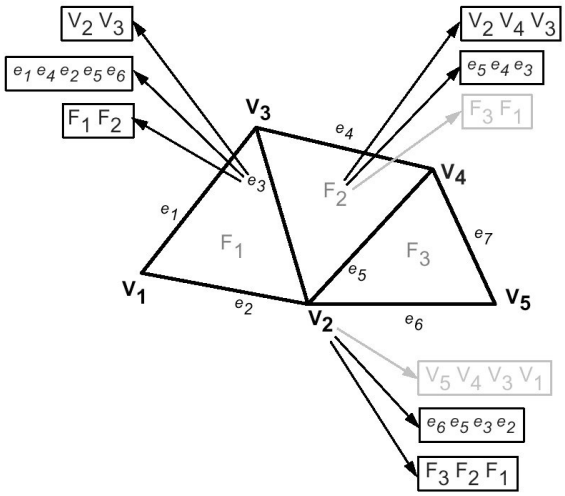


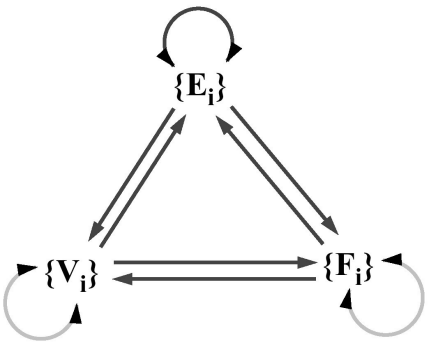
VERTEX TABLE			
V_1	X_1	Y_1	Z_1
V_2	X_2	Y_2	Z_2
V_3	X_3	Y_3	Z_3
V_4	X_4	Y_4	Z_4
V_5	X_5	Y_5	Z_5

FACE TABLE			
F_1	V_1	V_2	V_3
F_2	V_2	V_4	V_3
F_3	V_2	V_5	V_4

Representação Por Fronteira

- Baseada em arestas
  - Informações são derivadas
  - Rapidez
  - Estrutura menos pesada
  - Menor custo de armazenar

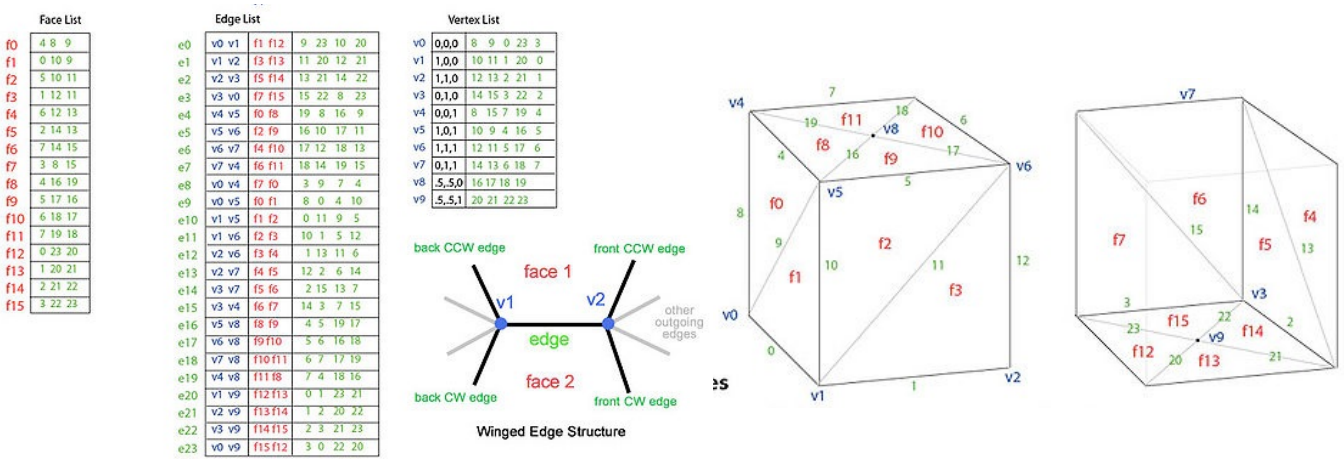




Representação Por Fronteira

- Várias estruturas baseadas em arestas:
  - *Winged-Edge*

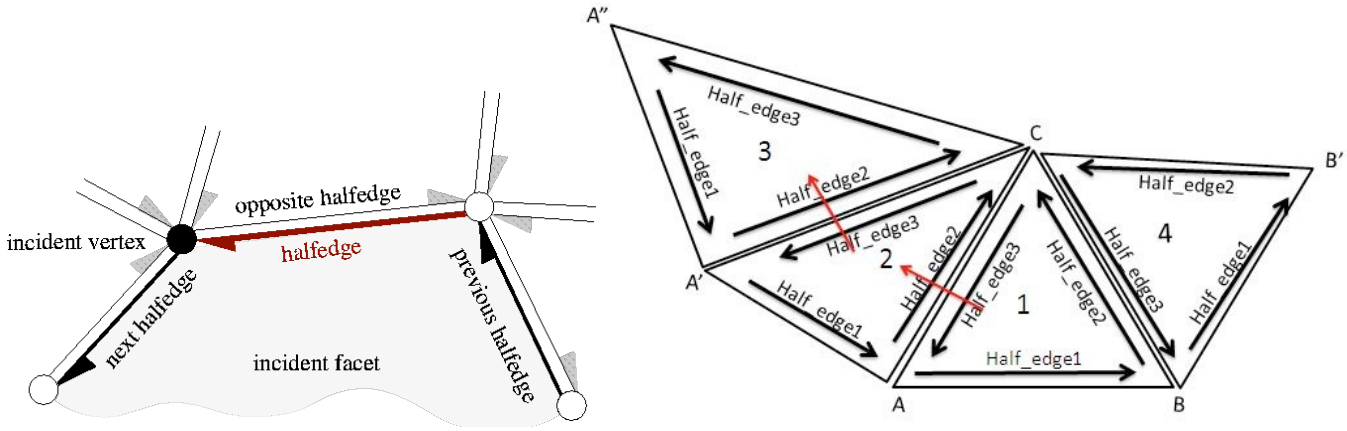
Baumgart,B., Winged-Edge Polyhedron Representation for Computer Vision. National Computer Conference,1975.



Representação Por Fronteira

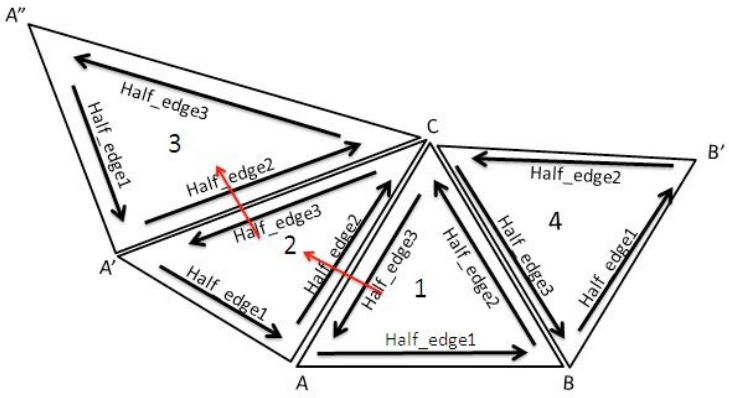
- Várias estruturas baseadas em arestas:
  - *Half-Edge*

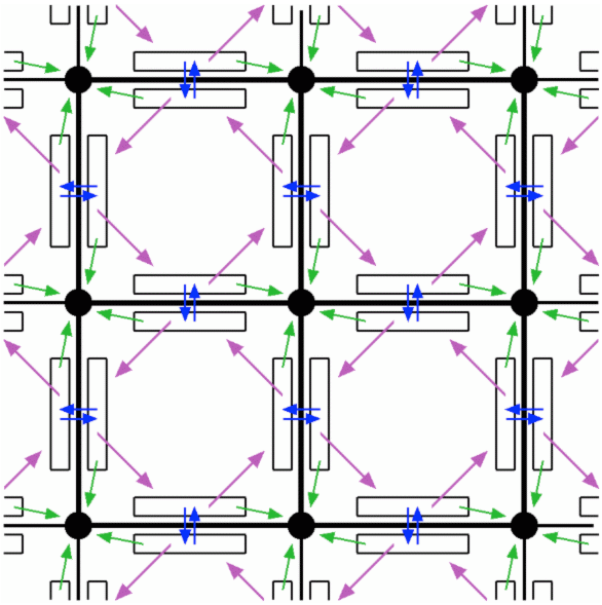
Weiler, K.. Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments. IEEE Computer Graphics and Application, 1985..



Representação Por Fronteira

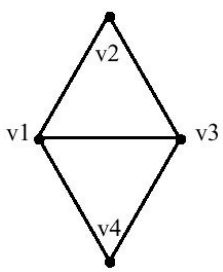
- Várias estruturas baseadas em arestas:
  - *Half-Edge*





Representação Por Malha Poligonal

- Estrutura simplificada
  - Malha Poligonal
    - Representa apenas
      - Vértices (Geometria)
      - Faces (Topologia)



- Simples
  - Armazenamento/ manutenção
  - visualização
- Difícil para aferição de propriedades da malha.

points/polygons format

vertices:
(x1, y1, z1)
(x2, y2, z2)
(x3, y3, z3)
(x4, y4, z4)

triangle: (1, 2, 3)
triangle: (1, 3, 4)

Representação de Objetos em Three.JS/WebGL

Representação de Objetos em Three.JS/WebGL

- Objetos geométricos são representados por conjuntos de:
  - Vértices
  - Faces

Representação de Objetos em Three.JS/WebGL

- Objetos simples podem ser parametrizados e gerados automaticamente:
  - Axis

    ```
    new THREE.AxisHelper( 1.0 );
    ```
  - Box

    ```
    new THREE.BoxGeometry( 0.2, 0.2, 0.2 );
    ```
  - Sphere

    ```
    new THREE.SphereGeometry( 0.2, 10, 10 );
    ```
  - Ring

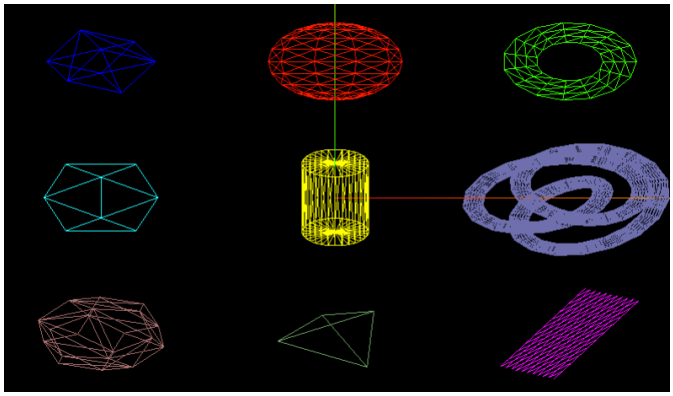
    ```
    new THREE.RingGeometry( 0.2, 0.1, 15, 3 );
    ```
  - TorusKnot

    ```
    new THREE.TorusKnotGeometry( 0.2, 0.03, 60, 20, 3, 2 );
    ```
  - Cilinder

    ```
    new THREE.CylinderGeometry( 0.1, 0.1, 0.5, 30);
    ```
  - Plane

    ```
    new THREE.PlaneBufferGeometry(0.2, 0.5, 10, 10);
    ```
  - Tetrahedron

    ```
    new THREE.TetrahedronGeometry(0.2);
    ```



Representação de Objetos em Three.JS/WebGL

- Criação explicita de uma malha simples com apenas uma face:

```
var triangleGeometry = new THREE.Geometry();

triangleGeometry.vertices.push(new THREE.Vector3( 0.5,  0.5, 0.0));
triangleGeometry.vertices.push(new THREE.Vector3(-0.5, -0.5, 0.0));
triangleGeometry.vertices.push(new THREE.Vector3( 0.5, -0.5, 0.0));

triangleGeometry.faces.push(new THREE.Face3(1, 2, 0));

triangleGeometry.faces[0].vertexColors[0] = new THREE.Color( 1.0, 0.0, 0.0);
triangleGeometry.faces[0].vertexColors[1] = new THREE.Color( 0.0,  1.0, 0.0);
triangleGeometry.faces[0].vertexColors[2] = new THREE.Color( 0.0,  0.0, 1.0);

var triangleMaterial = new THREE.MeshBasicMaterial({
     color:0xffffff,
     vertexColors:THREE.VertexColors,
     side:THREE.DoubleSide,
     wireframe:false
     });
var triangleMesh = new THREE.Mesh(triangleGeometry, triangleMaterial);
scene.add( triangleMesh );
```

Representação de Objetos em Three.JS/WebGL

- Criação explicita de uma malha representando um círculo:

```
var triangleGeometry = new THREE.Geometry();
var numVertices = 60;
var raio = 0.8;
triangleGeometry.vertices.push(new THREE.Vector3( 0.0,  0.0, 0.0));
for (i = 0 ; i < 2*Math.PI ; i+= (2*Math.PI)/numVertices) {
      var x = raio * Math.cos(i);
      var y = raio * Math.sin(i);
      triangleGeometry.vertices.push(new THREE.Vector3( x,  y, 0.0));
      }

for (i = 0 ; i <= numVertices ; i++) {
      triangleGeometry.faces.push(new THREE.Face3(0, i, i+1));
      triangleGeometry.faces[i].vertexColors[0] = new THREE.Color( 1.0, 1.0, 1.0);
      triangleGeometry.faces[i].vertexColors[1] = new THREE.Color( 0.0,  i/numVertices, 1.0 - i/numVertices);
      triangleGeometry.faces[i].vertexColors[2] = new THREE.Color( 0.0,  (i+1)/numVertices, 1.0 - (i+1)/numVertices);
      }

var triangleMaterial = new THREE.MeshBasicMaterial({
      color:0xffffff,
      vertexColors:THREE.VertexColors,
      side:THREE.DoubleSide,
      wireframe:false
      });

var triangleMesh = new THREE.Mesh(triangleGeometry, triangleMaterial);
```

Representação de Objetos em Three.JS/WebGL

- Criação explicita de uma malha representando um cubo:

```
var triangleGeometry = new THREE.Geometry();

triangleGeometry.vertices.push(new THREE.Vector3( 0.5,  0.5,  0.5));
triangleGeometry.vertices.push(new THREE.Vector3(-0.5, -0.5,  0.5));
triangleGeometry.vertices.push(new THREE.Vector3( 0.5, -0.5,  0.5));
triangleGeometry.vertices.push(new THREE.Vector3(-0.5,  0.5,  0.5));
triangleGeometry.vertices.push(new THREE.Vector3( 0.5,  0.5, -0.5));
triangleGeometry.vertices.push(new THREE.Vector3(-0.5, -0.5, -0.5));
triangleGeometry.vertices.push(new THREE.Vector3( 0.5, -0.5, -0.5));
triangleGeometry.vertices.push(new THREE.Vector3(-0.5,  0.5, -0.5));
```

Representação de Objetos em Three.JS/WebGL

- Criação explicita de uma malha representando um cubo (cont.):

```
// Front
triangleGeometry.faces.push(new THREE.Face3(1, 2, 0));
triangleGeometry.faces.push(new THREE.Face3(1, 0, 3));
triangleGeometry.faces[0].materialIndex =
triangleGeometry.faces[1].materialIndex = 0;
// Back
triangleGeometry.faces.push(new THREE.Face3(5, 4, 6));
triangleGeometry.faces.push(new THREE.Face3(5, 7, 4));
triangleGeometry.faces[2].materialIndex =
triangleGeometry.faces[3].materialIndex = 1;
// Top
triangleGeometry.faces.push(new THREE.Face3(3, 0, 4));
triangleGeometry.faces.push(new THREE.Face3(3, 4, 7));
triangleGeometry.faces[4].materialIndex =
triangleGeometry.faces[5].materialIndex = 2;
// Bottom
triangleGeometry.faces.push(new THREE.Face3(1, 6, 2));
triangleGeometry.faces.push(new THREE.Face3(1, 5, 6));
triangleGeometry.faces[6].materialIndex =
triangleGeometry.faces[7].materialIndex = 3;
// Right
triangleGeometry.faces.push(new THREE.Face3(2, 6, 4));
triangleGeometry.faces.push(new THREE.Face3(2, 4, 0));
triangleGeometry.faces[8].materialIndex =
triangleGeometry.faces[9].materialIndex = 4;
// Left
triangleGeometry.faces.push(new THREE.Face3(5, 1, 3));
triangleGeometry.faces.push(new THREE.Face3(5, 3, 7));
triangleGeometry.faces[10].materialIndex =
triangleGeometry.faces[11].materialIndex = 5;
```

Representação de Objetos em Three.JS/WebGL

- Criação explicita de uma malha representando um cubo (cont.):

```
var boxMaterials = [   new THREE.MeshBasicMaterial({color:0xFF0000}),
                       new THREE.MeshBasicMaterial({color:0x00FF00}),
                       new THREE.MeshBasicMaterial({color:0x0000FF}),
                       new THREE.MeshBasicMaterial({color:0xFFFF00}),
                       new THREE.MeshBasicMaterial({color:0x00FFFF}),
                       new THREE.MeshBasicMaterial({color:0xFFFFFF})
                   ];

var triangleMaterial = new THREE.MeshFaceMaterial(boxMaterials);
var triangleMesh = new THREE.Mesh(triangleGeometry,triangleMaterial);

scene.add( triangleMesh );
```

Representação de Objetos em Three.JS/WebGL

- Carregando uma malha de um arquivo:
  - Inúmeros formatos de representação 3D:
    - OBJ
    - PLY
    - VTK
    - STL
    - Json
    - …
  - Three.JS fornece leitores dos principais modelos

Representação de Objetos em Three.JS/WebGL

- Como é “a cara” de um modelo armazenado em um arquivo?

CUBE.OBJ

```
# cubeTriangle.obj
# a cube drawn with triangles
mtllib cubeMultiColor.mtl
v 1.0 -1.0 -1.0
v 1.0 -1.0 1.0
v -1.0 -1.0 1.0
v -1.0 -1.0 -1.0
v 1.0 1.0 -1.0
v 1.0 1.0 1.0
v -1.0 1.0 1.0
v -1.0 1.0 -1.0
```

```
g Top
f 7 6 5 8
#f 7 6 5
#f 7 5 8
g Bottom
f 2 3 4 1
#f 3 4 1
#f 3 1 2
g Front
f 3 2 6 7
#f 2 7 3
#f 2 6 7
g Back
f 1 4 8 5
#f 1 4 8
#f 1 8 5
```

```
g Right
f 4 3 7 8
#f 7 4 3
#f 7 8 4
g Left
f 2 1 5 6
#f 1 6 2
#f 1 5 6
```

```
group Top
usemtl Top
group Bottom
usemtl Bottom
group Front
usemtl Front
group Back
usemtl Back
group Left
usemtl Left
group Right
usemtl Right
```

Representação de Objetos em Three.JS/WebGL

- Como é “a cara” de um modelo armazenado em um arquivo?

```
CUBE.MTL
# Blender3D MTL File:
newmtl Top
Kd 0.0 1.0 0.0
newmtl Bottom
Kd 1.0 1.0 0.0
newmtl Front
Kd 0.0 0.0 1.0
newmtl Back
Kd 0.0 1.0 1.0
newmtl Left
Kd 1.0 0.0 0.0
newmtl Right
Kd 1.0 0.0 1.0
```

Representação de Objetos em Three.JS/WebGL

- Como é “a cara” de um modelo armazenado em um arquivo?

```
<html>
  <head>
      <title>MATA65 - Computacao Grafica</title>
      <h1>Malhas Poligonais em Three.js/WebGL.</h1>
      <h3>Carrega um modelo .obj de um arquivo.</h3>
  </head>

  <script type="text/javascript" src="../libs/three.js"></script>
  <script type="text/javascript" src="../libs/examples/js/loaders/OBJLoader.js"></
script>
   <script type="text/javascript" src="simpleLoadOBJ.js"></script>

   <body onload="init();">
      <div id="WebGL-output" ></div>
  </body>
</html>
```

Representação de Objetos em Three.JS/WebGL

- Leitura de um arquivo OBJ:

```
   // Load Mesh
var loader = new THREE.OBJLoader();
loader.load('../../Assets/Models/
soccerball.obj', loadMesh);

   renderer.clear();
   render();
};

function render() {
   if (mesh) {
   renderer.render(scene, camera);
        }
   requestAnimationFrame(render);
   }
```

```
function loadMesh(loadedMesh) {
var material = new
THREE.MeshBasicMaterial({color: 0xffffff,
wireframe: true});

loadedMesh.children.forEach(function (child){
        child.material = material;
        });

mesh = loadedMesh;
scene.add(loadedMesh);

// Global Axis
var globalAxis = new THREE.AxisHelper( 5.0 );
   scene.add( globalAxis );
   };
```

A seguir...
Cores