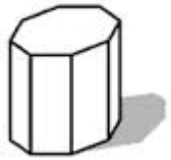


Realismo Visual

Aula 11

UFF - 2018

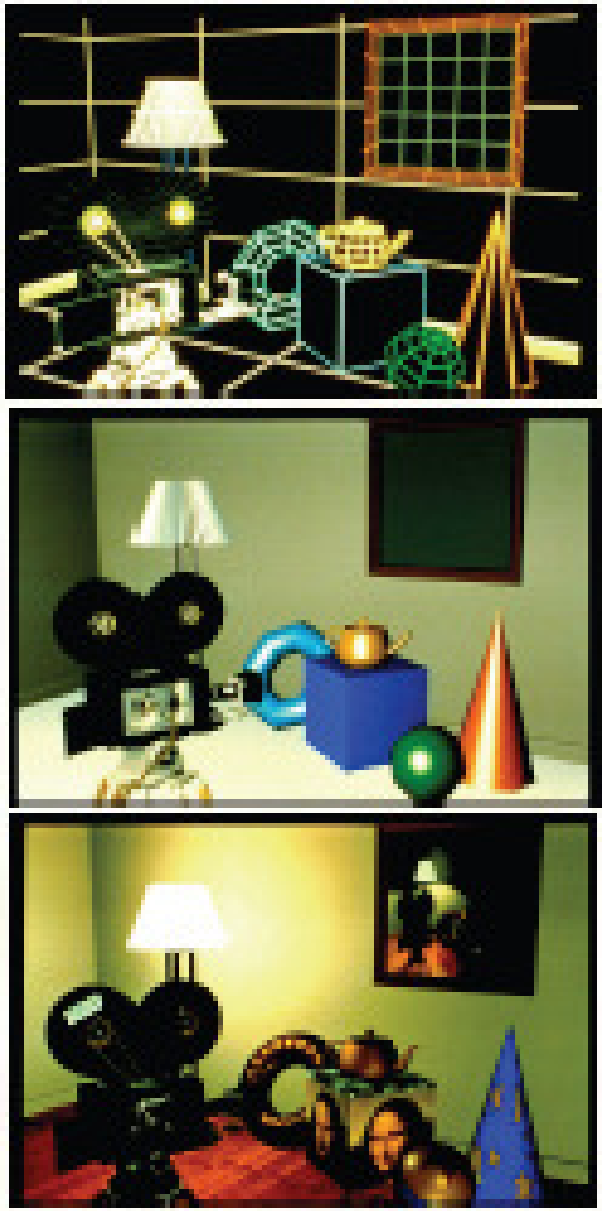
Objetivos

Melhorar o entendimento das cenas e objetos criados

Possibilidade de representação de dados, objetos e cenas complexas

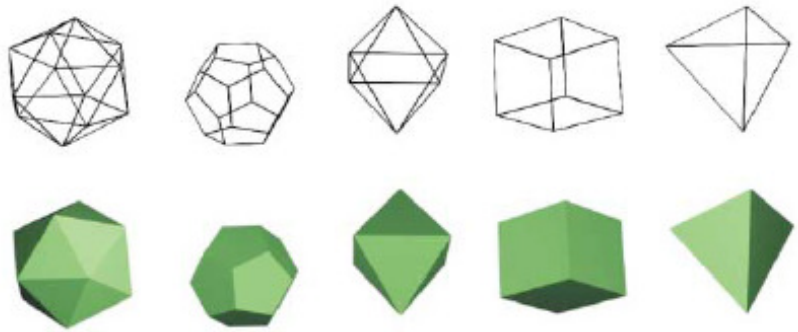
Realismo até o nível desejado da forma adequada para a aplicação

(real time x perfeição física da cena)



Nível adequado do realismo

Remoção de partes invisíveis do objeto (linhas, superfícies e oclusões por outros objetos)



Sombreamento das diversas superfícies ou *Shading* :

reflexão difusa,

reflexão especular

Demais níveis de detalhes:

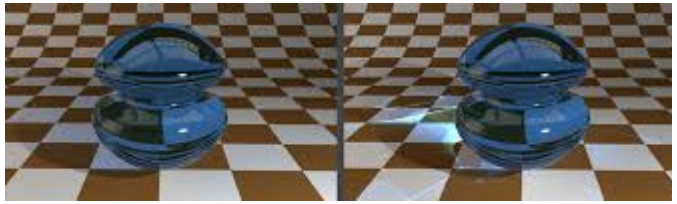
Sombras (*shadows*)

Reflexão,

transparências,

refração,

Texturas



Wire frame : adequado para posicionamentos e desenho, mas não realístico

Todas as linhas são mostradas.

Passo seguinte do realismo eliminar **partes da cena que não são vistas quando objetos opacos são vistos de determinada direção.**

Tratamento de *hiddens* ou *Hidden Line/surface problem*

Eliminação de linhas:
caso particular da

definição de que faces ou superfícies são ocultas por outras do objeto ou cena.

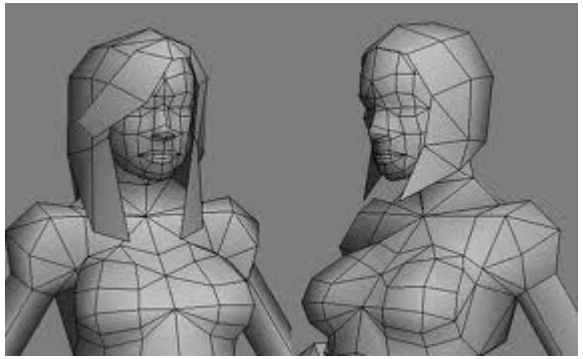
Técnicas de visibilidade

Back face culling

Priority fill ou *painter's algorithm*

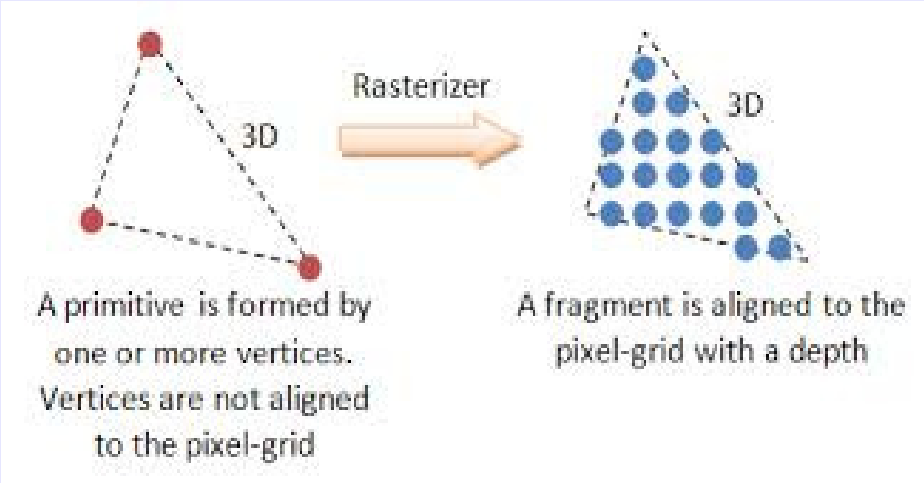
Z- buffer

Ray casting
(*Ray tracing simplificado*
ou aproximado)



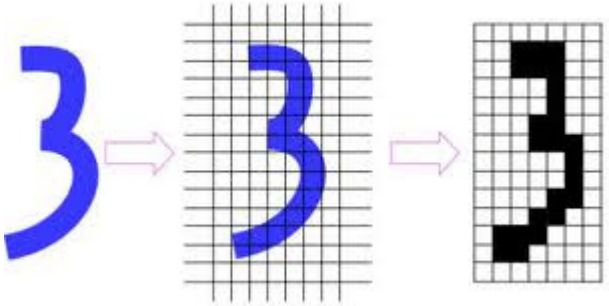
HÁ ALGORITMOS NA FORMA **VETORIAL** E **RASTER**

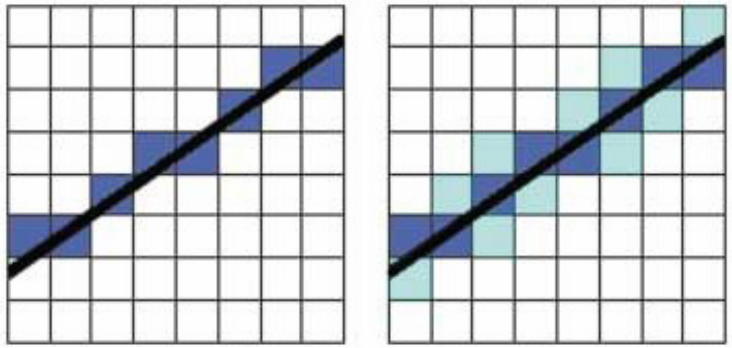
RASTER: o objeto em 3D é tratado na forma final quando já "*discretizado*" em pixels.



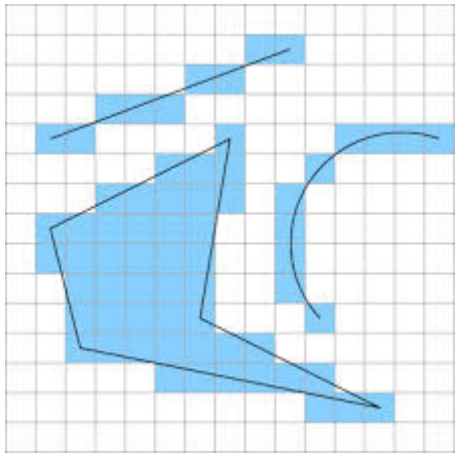
Rasterisation
(ou **rasterization**)
converte uma imagem descrita como vector format para a forma de pixels (dots) para representação em video, printer ou storage in a bitmap file format.

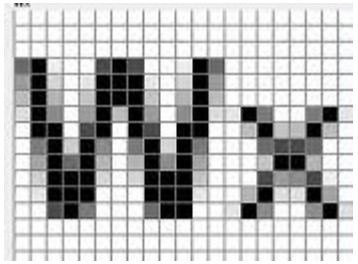
Alising → antialising





Rasterizar = Usar a malha de pixels para descrever os objetos!





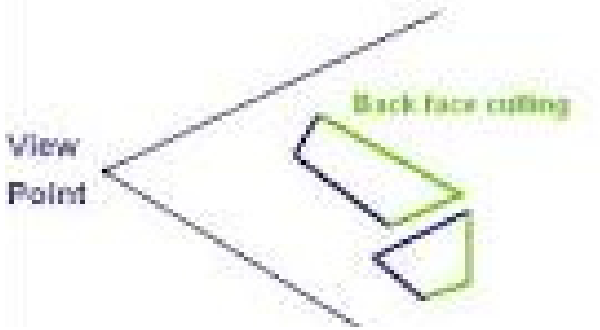
Back face culling, método de Roberts ou teste da normal

Algoritmo posiciona o **objeto** e o **observador** no mesmo sistema de coordenadas (SRU ou WC).

Não considera projeções ou perspectivas inicialmente.

Isso entra em uma outra etapa no processo de visualização (pipeline)

Back face culling



Demo: em javascript:

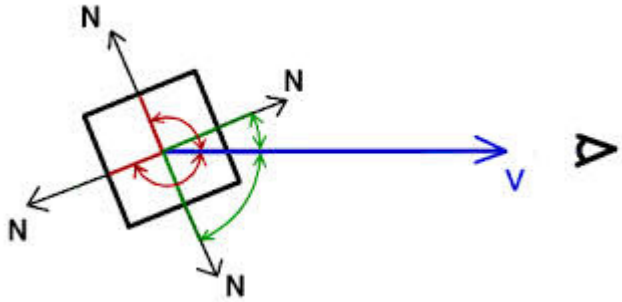
http://echolot-1.github.io/back-face-culling-demo/

echolot-1/**back-face-culling-demo**

*Em CG **back-face culling** determina quando a face de um objeto será **visivel de um ponto de vista**.*

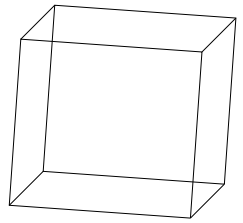
*Esse processo torna o **rendering** mais **eficiente pois reduz o número de poligonos** a ser desenhado.*

Back face culling



Idéia básica:

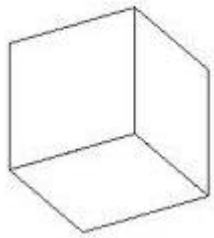
Remover faces traseiras dos objetos em relação ao observador



Adequadas para objetos convexos.

OBS :

Ser **não convexo** ≠ ser **côncavo**

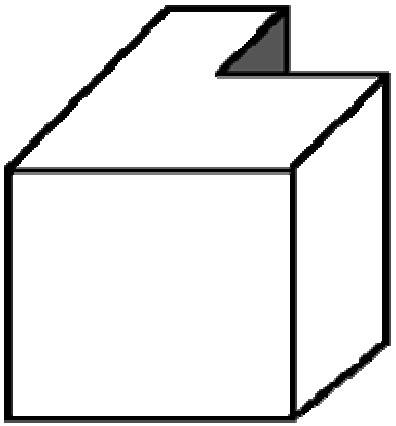


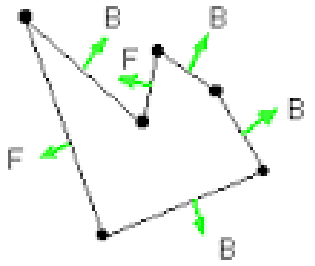
Objetos convexos

Definição:

Formado por faces convexas.

i.e. Formado por polígonos convexos: nos quais a **ligação entre quaisquer 2 pontos** internos nunca passa por uma parte externo a face:

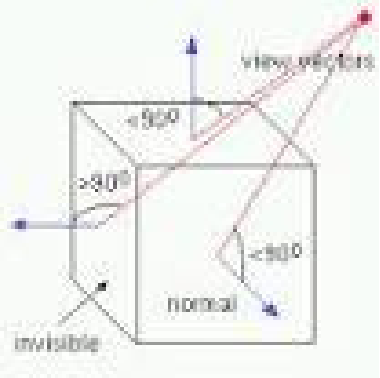


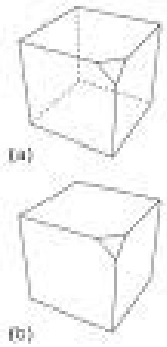


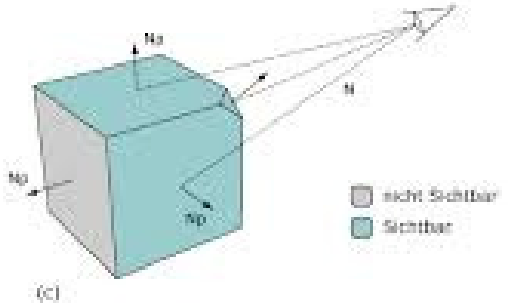
Algoritmo posiciona o objeto e o observador no mesmo sistema de coordenadas (SRU ou WC)

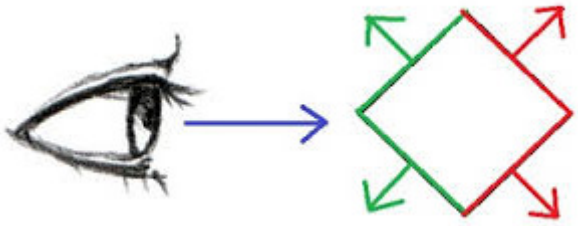
Usa-se a **direção que as normais** às faces fazem com a direção de visualização.

Entre **-90** graus e **90** graus a **face é visível** pelo observador (ou a face é de frente) .



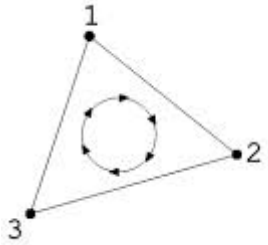


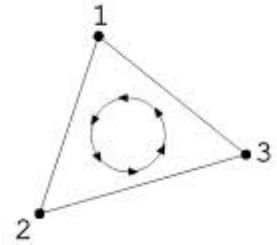


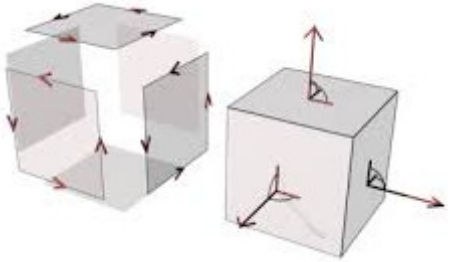


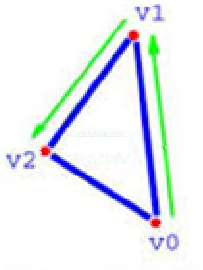
1-Obtêm a normal às faces

Através do cálculo do **produto vetorial** de dois vetores da face: a ordem dos vértices é importante!

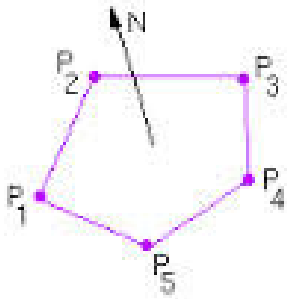








$$N = (V_1 - V_0) \times (V_2 - V_0)$$



$$(V_1 - V_0) \times (V_2 - V_0) = -(V_2 - V_0) \times (V_1 - V_0)$$

2 - Define-se o vetor da direção de visão

3- Verifica-se o ângulo!

Através do **produto interno** entre as normais e a direção de visão, (não é preciso calcular o ângulo) apenas ver se o resultado **é maior que zero** → ângulo entre -90 ° e 90 ° !

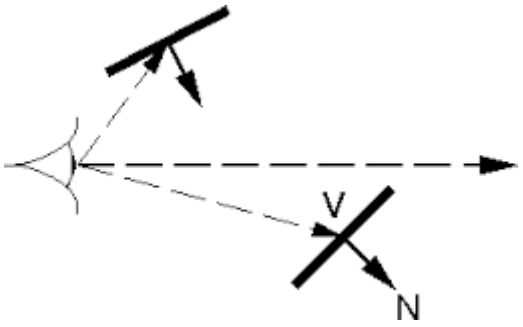
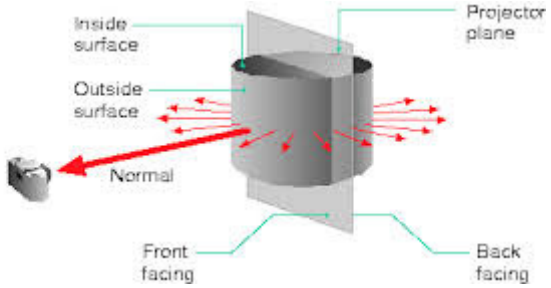


figure 206
Inside and outside surface



$$(V_0 - P) \cdot N \geq 0$$

Algoritmo

4- Só desenha a face se ele é visivel **!**

OBS- Se for visível ai se preocupa em projetar o objeto de 3D para 2D e em posiciona-lo no dispositivo

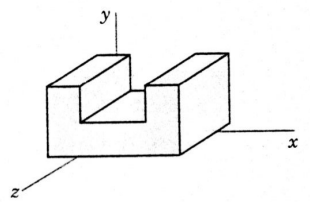
Viewing pipeline / Ações para ver uma cena

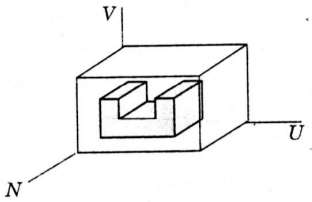
Modelagem dos objetos que compõem a Cena –SRO)

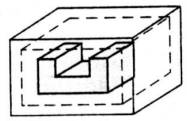
Sua posição no SRU (World Coordinates - WC), sua visão de maneira realística por um observador .

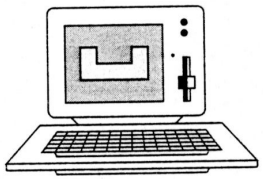
Sua vista em perspectiva e projeção em 2D.

E posicionamento na window ou no canvas de desenho (DC - SRD).



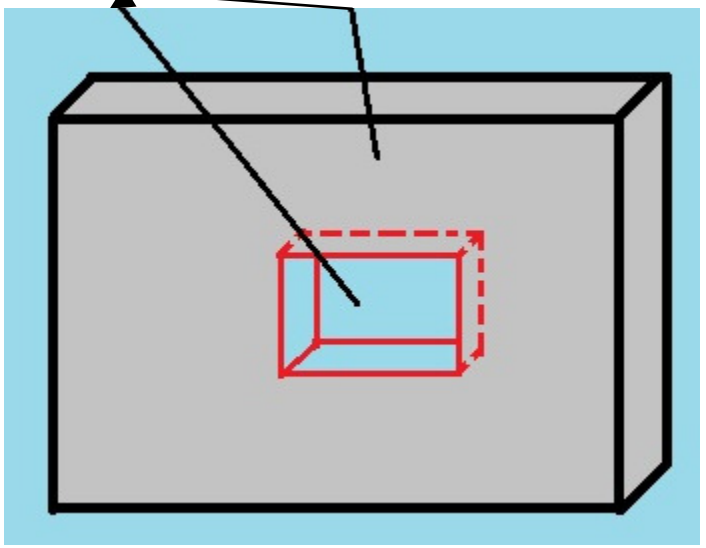






Fórmula de Euler V – A + F = 2

Genus G de um objeto : menor **número de furos** que trespassam o objeto.

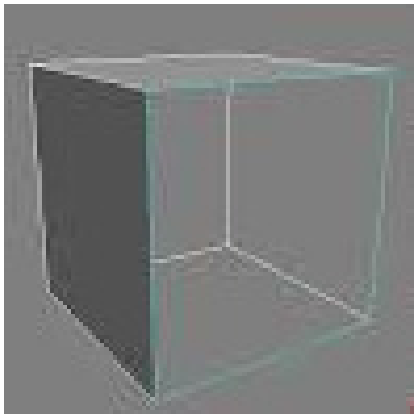


Genus G=**1**

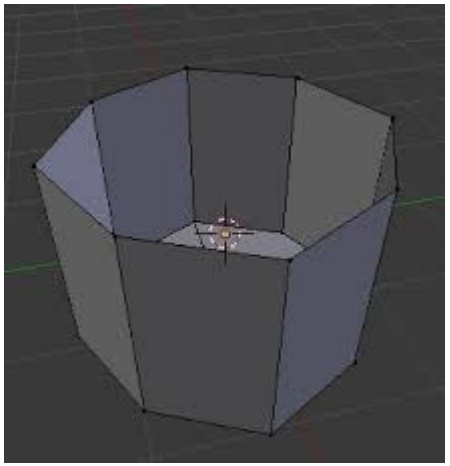
Qual o genus de uma tubulação ?
Resposta: Veja o vídeo no Breno onde ele mostra isso por deformação!
Segue o link do vídeo no youtube: http://youtu.be/QkcryL4f6hE

Fórmula de Euler :V – A + F = 2

Buracos H : menor **número de furos** que não trespassam ou loops fechados de faces.



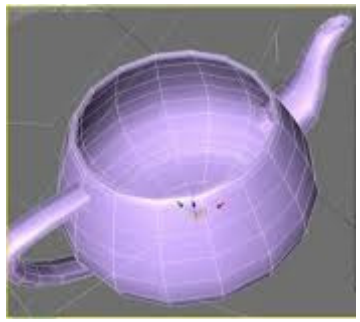
Buracos H=**1**



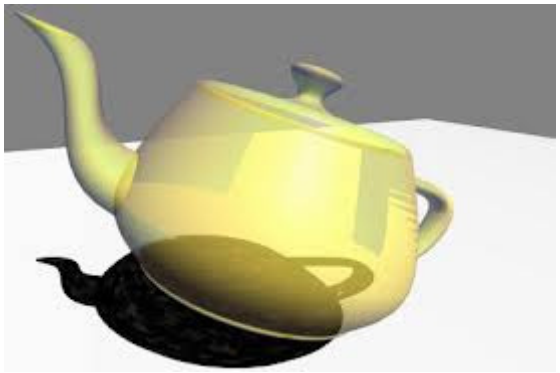
Formula de Euler→**Euler-Poincaré**:

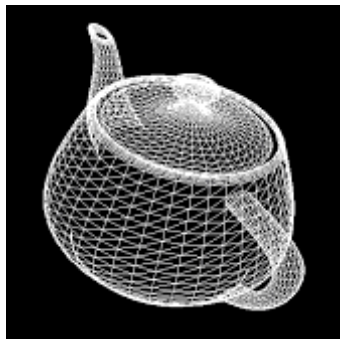
Componentes separáveis ou partes conectadas: C

formula de Euler - Poincaré: V-A+F-H = 2(C-G)



H=1 e G=?





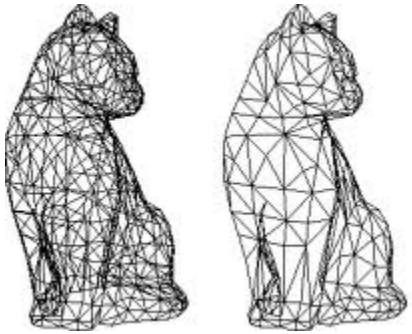
Utah teapot

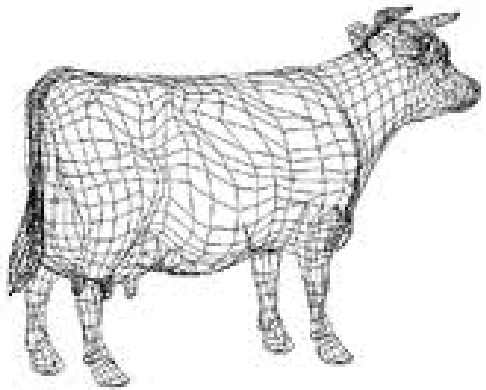
Um **teapot** não é uma **chaleira** ! Nunca é usado para por água no fogo e a ferver!

V – A + F = 2

Importante da modelagem correta para o de uso do objeto adequadamente

Já definir se há **buracos H**, ou furos **trespassantes G** ou **partes conectadas C**, na modelagem inicial do objeto é mais complexo.





Qual o ***Geno*** de um corpo humano para uma modelagem que o tratasse por dentro, como para uma endoscopia?

Bibliografia:

E. Azevedo, A. Conci, Computação Gráfica: teoria e prática, Campus ; - Rio de Janeiro, 2003

J.D.Foley,A.van Dam,S.K.Feiner,J.F.Hughes. Computer Graphics- Principles and Practice, Addison-Wesley, Reading, 1990.

H. Watt, F. Policarpo - The Computer , Addison-Wesley Pub Co (Net); 1998

http://en.wikipedia.org/wiki/Shadow_mapping

https://noppa.oulu.fi/noppa/kurssi/521493s/luennot/521493S_3-d_graphics_vi.pdf

http://graphics.stanford.edu/papers/rad/