

# Realismo Visual

MATERIAL PREPARADO PELOS PROFESSORES:

RODOLFO AYALA LOPES COSTA

SAUL DELABRIDA

# Agenda

---



CONCEITOS BÁSICOS



RENDERING



RASTERIZAÇÃO

# Introdução

Criar imagens sintéticas realistas é o objetivo final da computação gráfica

O realismo visual é composto pelas técnicas de tratamento computacional aplicadas aos objetos sintéticos gerados, com o objetivo de criar-lhes uma imagem o mais próxima da realidade que se teria se eles fossem construídos e filmados

O realismo é fundamental nas simulações, no entretenimento, na educação e muitas outras aplicações

# Realismo Visual

## - Classificação

Podemos classificar o realismo em:

### Estática:

- Objetos são tratados em cenas estáticas
- Chamado realismo fotográfico
- Usado quando se quer prover imagens realistas e estáticas

### Dinâmica:

- Relacionada ao movimento da cena e seus personagens
- Muito utilizado em animações e cenas de jogos que contém interação do usuário

# Rendering - Realismo Visual

---



*Rendering*, é uma das etapas de criação do realismo fotográfico

Não existe uma tradução fora da área computacional adequada para essa palavra

Basicamente, podemos descrevê-la por “realismo visual”



A computação gráfica trata da síntese de imagens através do computador



Sintetizar uma imagem (uma cena ou objeto) é criá-la em termos da definição dos dados dos objetos que a compõem



Isso se faz a partir da geometria da cena, das informações sobre os materiais de que são feitos os objetos (suas cores e texturas), das condições de iluminação ambiente e da posição de observação da cena

# Etapas do Realismo Visual

---

- Este processo envolve sete etapas distintas
- Nem todas usadas em todas as aplicações
- O mais usual é que um trabalho seja apresentado em um nível de realismo adequado ao seu uso
- Principalmente porque ao aumentar-se o realismo de uma cena aumenta-se também seu tempo de processamento e custo de geração

Construção do Modelo

Transformações Lineares

Eliminação de polígonos ou faces escondidas

Recortes

Representação Tridimensional

Eliminação de pontos oclusos

Tratamento Realismo

# Etapa 1 - Construção do Modelo



Etapa relacionada com a construção do modelo que conterá todas as informações para o processo de realismo visual



Consiste na utilização de alguma técnica de modelagem



OpenGL, Blender, Autodesk Maya, 3ds Max, Cinema 4D, Zbrush, SketchUp, entre outros são exemplos de ferramenta para esta etapa

# Etapa 2 - Transformações Lineares

---



Essa fase aplica transformações lineares no modelo de modo que ele tenha aparência tridimensional nos diversos dispositivos (geralmente bidimensionais)



Essa fase consiste, então, na utilização de projeções e perspectivas adequadas



## Etapa 3 - Eliminação de polígonos ou faces escondidas



Considera a eliminação de polígonos ou faces escondidas devido a posição relativa entre os objetos da cena e o observador



O algoritmo Culling back-faces é utilizado para determinar se um polígono em um objeto gráfico é visível pela perspectiva do usuário



Um método para implementação do back-face culling é descartar todos os triângulos onde o produto escalar de sua superfície normal e o vetor de câmera para triângulo for maior ou igual a zero.

# Etapa 4 - Recortes

---

- São desconsideradas as partes das cenas que não serão mostradas, são chamados “recortes”
- Imagine que você tenha uma infinidade de coisas para ver, para algumas sua visão será limitada, por que estão encobertas ou não chamam atenção, logo podem deixar de entrar no pipeline de processamento
- Clipping é uma técnica utilizada para renderização somente de pontos que estiverem no campo de visão da câmera

■ Algoritmos para realização desta tarefa:

**Line clipping algorithms:**

- Cohen–Sutherland
- Liang–Barsky
- Fast-clipping
- Cyrus–Beck
- Nicholl–Lee–Nicholl
- Skala
- $O(\lg N)$  Algorithm

**Polygon clipping algorithms:**

- Greiner-Hormann
- Sutherland–Hodgman
- Weiler–Atherton
- Vatti

## Etapa 5 - Representação Tridimensional



Essa fase se encarrega em converter a representação tridimensional para pixels



Seja lá qual for o sistema de coordenadas que se esteja usando, os dados serão levados para um conjunto de coordenadas do dispositivo em que será mostrado



Essa conversão de coordenadas leva os dados do modelo para o mundo digital



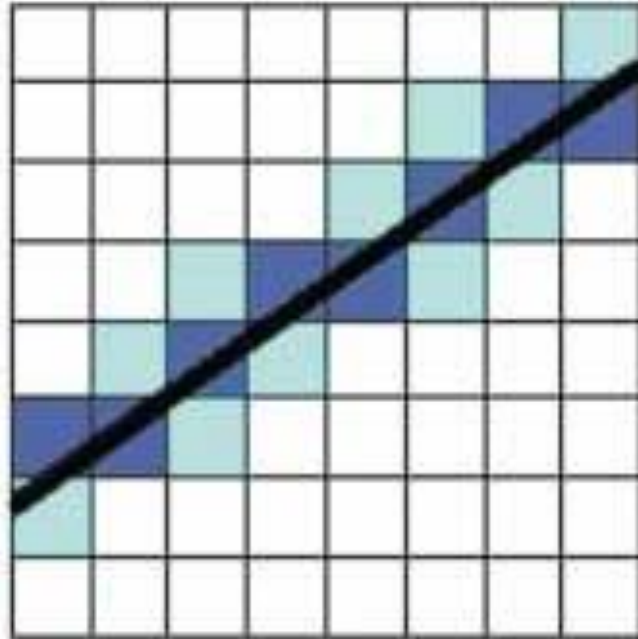
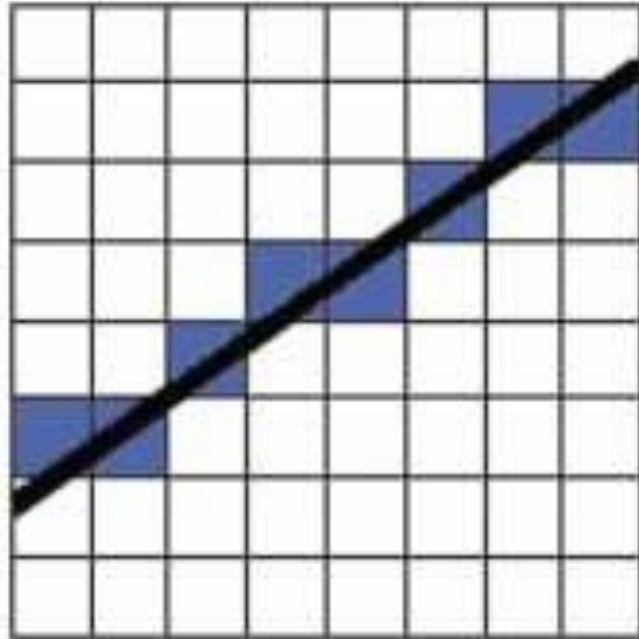
Linhas e áreas serão transformadas em conjuntos de pixels



Esse processo é chamado rasterização

# Etapa 5 - Rasterização

---



- Rasterização é o processo de conversão da representação vetorial para a matricial
- Ela permite realizar a conversão de um desenho tridimensional qualquer em uma representação inteira possível de ser armazenada na memória (de vídeo ou impressão)

# Etapa 5 - Rasterização



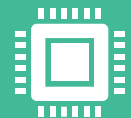
Rasterização é um processo de amostragem

Domínio contínuo -> discreto  
Problemas de aliasing são esperados



Cada primitiva (retas, quadrados, polígonos, ...) pode gerar um grande número de pixels:

Rapidez é essencial



Em geral, rasterização é realizada pelo hardware



Pode ser feita de duas formas:

Rasterização de retas  
Rasterização de polígonos

# Rasterização de Retas

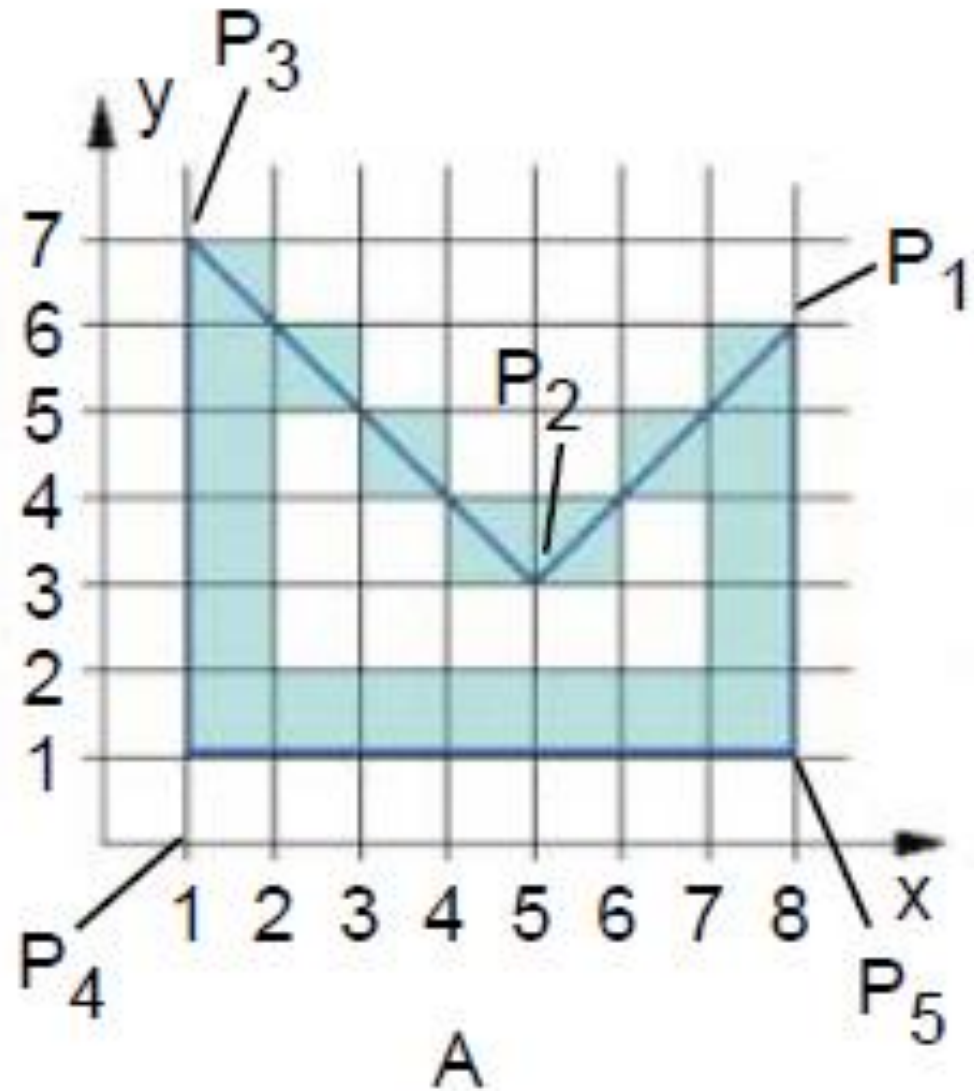
---

- Algoritmo de Bresenham para Traçado de Linhas
- Dados de entrada, dois pixels (x1; y1) e (x2; y2) da reta a ser rasterizada

```
void line(int x0, int y0, int x1, int y1) {  
  
    int dx = abs(x1-x0), sx = x0<x1 ? 1 : -1;  
    int dy = abs(y1-y0), sy = y0<y1 ? 1 : -1;  
    int err = (dx>dy ? dx : -dy)/2, e2;  
  
    for (;;) {  
        setPixel(x0,y0);  
        if (x0==x1 && y0==y1) break;  
        e2 = err;  
        if (e2 >= dx) { err -= dy; x0 += sx; }  
        if (e2 < dy) { err += dx; y0 += sy; }  
    }  
}
```

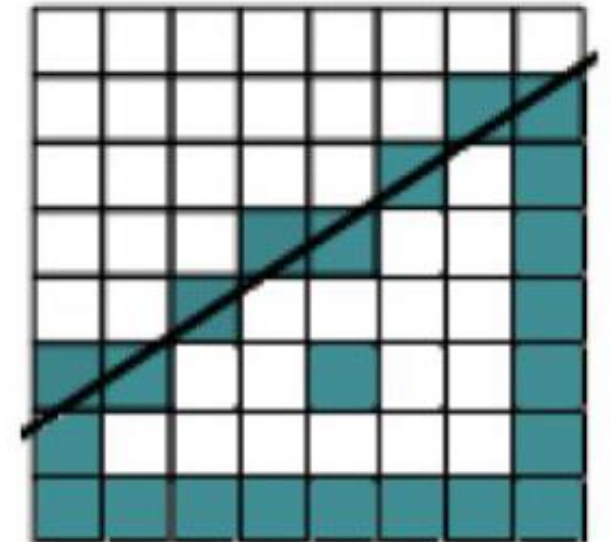
# Rasterização de Polígonos

- Os polígonos são geralmente representados na memória pela lista de seus vértices
- São projetados na tela dos dispositivos pela projeção de cada um desses vértices



# Rasterização Polígono – Técnica de Alastramento

- Técnica recursiva de alastramento
- Considere que as bordas já estão traçadas
- Problemas:
  - Funciona apenas para preenchimento de cor constante
  - Pode consumir muita memória devido o contexto da pilha de recursão



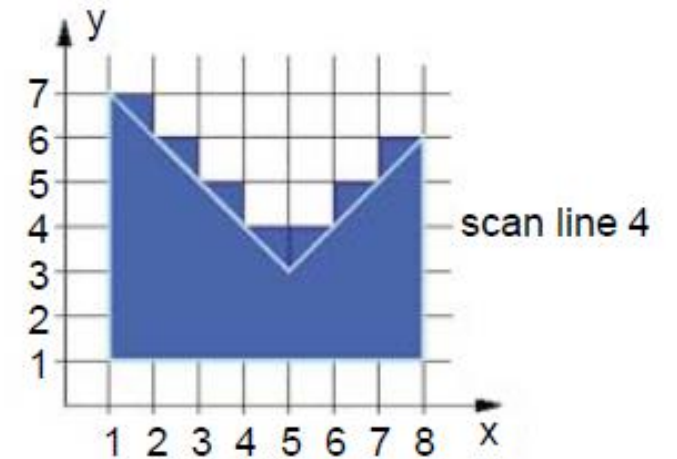


# Rasterização de Polígonos - Técnica de Varredura (scan- line)

Exceto nas bordas, pixels adjacentes em um polígono possuem as mesmas características

Essa propriedade é chamada coerência espacial

Os pixels de uma dada linha (scan line) variam somente nas bordas do polígono

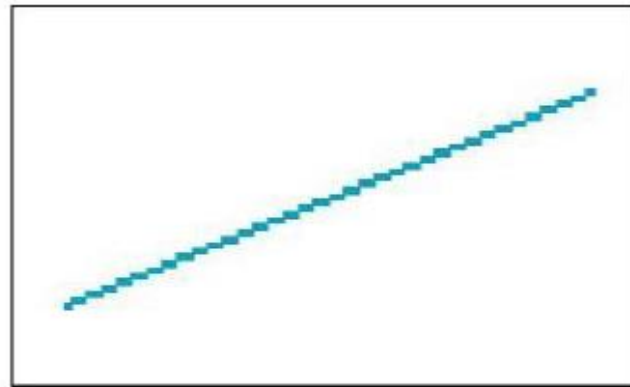


B

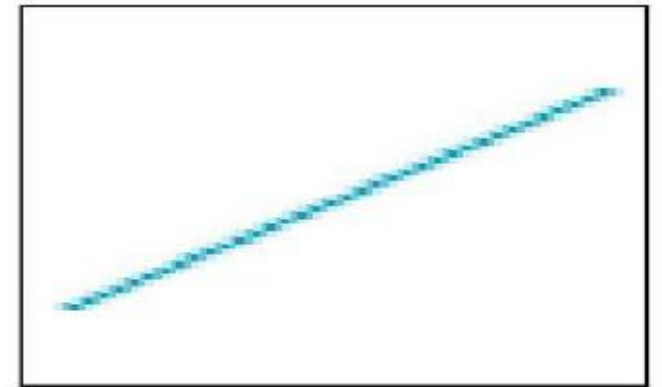
# Rasterização - Antialiasing:

Utiliza ponderação por  
área

A cor de um pixel é  
determinada pela  
porcentagem de sua área  
ocupada por uma linha de  
largura igual a 1 pixel



sem antialiasing



com antialiasing

## Etapa 6 - Eliminação de pontos oclusos



Continuação da terceira fase, pois trata também da eliminação de partes de um objeto que devem ser removidas

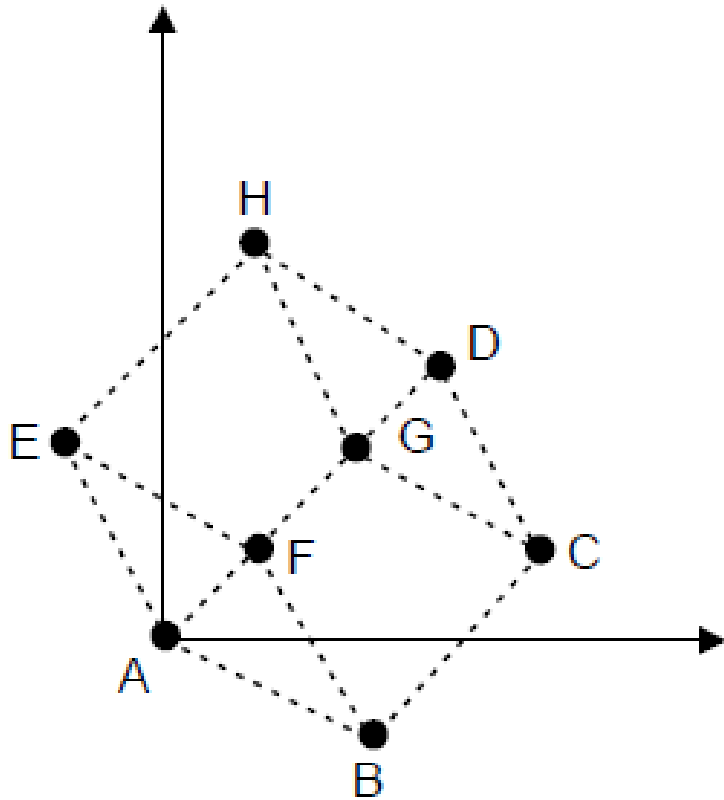


Agora essas partes são devidas à interferência dos diversos objetos presentes na cena, onde, devido a sua posição relativa, pode ocorrer que uns fiquem na frente de partes de outros



Muitos autores englobam estas duas partes no que se chama tratamento de partes escondidas (hidden)

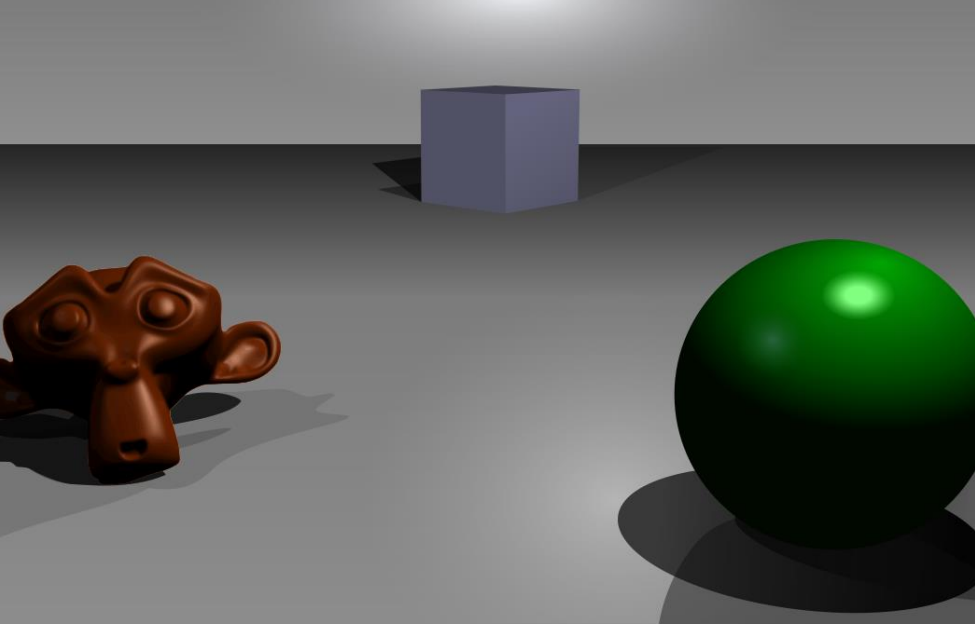
# Remoção de Linhas e Superfícies Escondidas



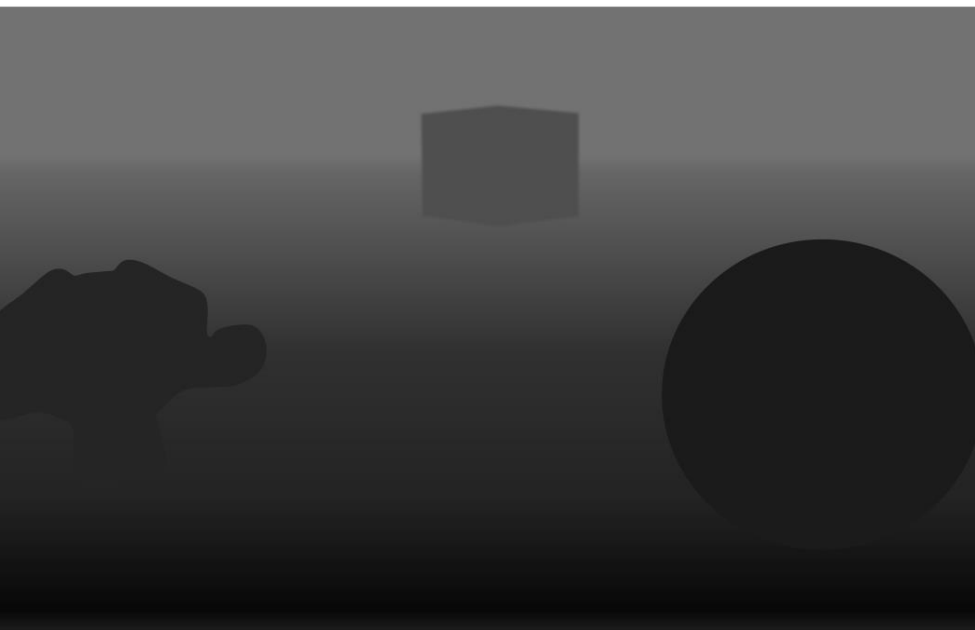
- A solução eficiente de problemas de visibilidade é o principal passo do processo de criação de cenas realísticas
- Esse problema lida frequentemente com a determinação da visibilidade de linhas e superfícies
- Fase foi denominada de eliminação de polígonos ou faces escondidas (culling back-faces)
- Algoritmos:
  - Algoritmo de Visibilidade por Prioridade ou Algoritmo do Pintor
  - Algoritmo Z-Buffer

# Algoritmo de Visibilidade por Prioridade

- Dê prioridade a visualização dos objetos mais próximos ao observador
- Também recebe o nome de algoritmo do pintor por simular a forma como um pintor faria para tratar a visibilidade de uma tela
- Se um objeto “A” bloqueia a visão de um objeto “B” e ambos os objetos encontram-se na mesma linha de visão do observador, então o objeto “B” está mais distante do observador que o objeto “A”



A simple three-dimensional scene



Z-buffer representation

# Algoritmo Z-Buffer

---

- É o algoritmo mais simples e popular
- Requer a alocação de até dois buffers, ou matrizes, em memória
- Dimensões dos buffers são idênticas à tela de apresentação
- Normalmente denominados buffers de Imagem e de Profundidade
- O uso do buffer de Imagem, é opcional, devendo possuir a mesma profundidade de cor da tela de apresentação final
- Utiliza-se o buffer de Imagem como “rascunho” durante os cálculos de visibilidade dos objetos
- O buffer de Profundidade destina-se a armazenar a distância de cada pixel, da tela rascunho fictícia inicial, ao plano de projeção, sendo também chamado de z-Buffer

# Z-Buffer – Vantagens e Desvantagens



## Vantagens:

Simples e comumente implementado em Hardware

Objetos podem ser desenhados em qualquer ordem



## Desvantagens:

Lento se o número de polígonos é grande

Dificulta o uso de transparência ou técnicas de antiserrilhado

## Etapa 7 - Tratamento Realismo

Realismo “fotográfico” começa a ser realmente tratado e percebido

E para que seja feito adequadamente deve-se levar em conta as luzes presentes na cena, suas intensidades e direções (em relação aos objetos)

Também devem ser consideradas todas as características das superfícies representadas: transparência, brilho, reflexão e textura

Ainda devem ser consideradas as sombras que os diversos objetos fazem entre si e nas superfícies em que se apoiam

O nível de realismo dessa fase pode ser tão sofisticado quanto a aplicação precisar



# Formas de se fazer o Rendering



Realismo por Passadas



A construção de uma cena realística é um processo incremental



Essa forma de criação de cenas realísticas é denominada de realismo por passadas



É a forma utilizada pela grande maioria dos sistemas para geração de cenas realísticas



Esse processo permite que os atributos de uma cena sejam renderizados separadamente da sua geração ou modelagem



Diversas técnicas e softwares participam do processo de inclusão do grau de realismo desejado a cena

# Realismo por passadas - Benefícios



Economia de memória: pois nessa técnica não precisam ser colocados todos os objetos de uma vez para o render



Facilidade da introdução de modificações: se precisarmos alterar somente alguns elementos de uma cena como luzes, sombras ou cores, não será necessário perder ou refazer todo o processo de renderização anterior



Um outro motivo que torna mais eficiente essa forma de renderização é a possibilidade de maior utilização das imagens estáticas

Por exemplo, se já temos em uma animação uma cena observada a partir de um ponto estático (como um filme com uma câmera fixa), não será necessário renderizar toda a cena, apenas os pontos que sofrerem modificações



Essa técnica permite também a mixagem com objetos ou texturas reais obtidas por captura, ou seja, o render por passadas permite uma integração com imagens fotográficas, acrescentando sombras, texturas complexas ou elementos do mundo real (como rostos) à cena