

# Fundamentos de Computação Gráfica

Nuno Martins  
ncmartin@isec.pt

# Técnicas discretas

- Durante muitos anos, a computação gráfica preocupou-se exclusivamente com primitivas para a modelação de objetos, não se importando em alcançar o fotorealismo em imagens geradas por computador (CGI, na sua sigla em inglês).
  - Não permitia a escrita e leitura direta de pixels individuais no *frame buffer* ou noutros *buffers* existentes.

# Técnicas discretas

- Uma imagens CGI fotorealista assemelha-se a uma fotografia de uma cena real, com detalhes precisos, iluminação e sombras realistas, e uma aparência geral que se aproxima da realidade.
  - Estas imagens são frequentemente usadas em filmes, jogos de vídeo e publicidade para criar mundos virtuais que pareçam reais.

# Técnicas discretas

- Este avanço veio permitir o uso de algumas novas técnicas.
- As principais técnicas que fazem parte desse grupo e que interessam a esta unidade curricular são:
  - *Anti-aliasing*;
  - Mapeamento de texturas;
  - Nevoeiro.

# Técnicas discretas

- Aliasing e anti-aliasing -

- A que se deve a diferença entre as duas imagens abaixo?



- Tecnicamente, à distorção produzida por se representar um sinal de alta resolução numa resolução mais baixa.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- Na prática, a diferença é devida à representação das linhas (um sinal de alta resolução, por ser contínuo) apresentar um aspeto em escada (ou serrilhado).
  - Este efeito, por vezes, acontece com as CGI e pode ser visto em ecrãs de resolução relativamente alta.
  - Estas imperfeições são, também, conhecidas como *jaggies* ou *crawlies* (em animação).

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- O problema referido é chamado de *aliasing espacial* e as técnicas usadas para o reduzir designam-se por *anti-aliasing* (AA).



**Figura 1.** Aliasing e anti-aliasing.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

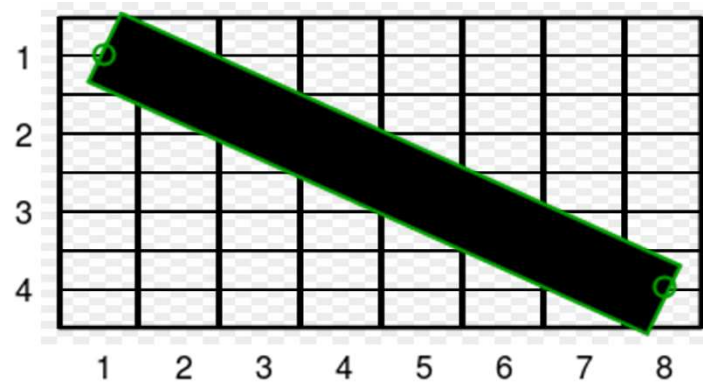
- Mais especificamente, o *aliasing* espacial é causado pelo fato da linha ser aproximada por conjuntos de pixels localizados numa grelha discreta.
  - Isto é, acontece devido ao fato de se estar a representar uma linha reta, cujo modelo tem uma resolução infinita (porque é definido num espaço contínuo), numa representação visual com resolução limitada (discreta).



# Técnicas discretas

- Aliasing e anti-aliasing -

- Matematicamente as linhas são entidades unidimensionais que têm comprimento e não têm espessura.
- As linhas desenhadas num ecrã têm que ter espessura para poderem ser visíveis.



**Figura 2.** Linha a desenhar.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- Na computação gráfica, para se poder renderizar uma linha (como a da figura 2),
  - Define-se uma espessura e uma cor para a linha;
  - Assumindo uma espessura de um *pixel*, como a linha pode atravessar mais do que um *pixel* de cada vez, atribui-se a cada *pixel* por onde a linha passa o valor da respetiva cor.

# Técnicas discretas

- Aliasing e anti-aliasing -

- Como se pode verificar na figura 2, certos *pixels* não são ocupados na sua totalidade, e a linha acaba por ser representada em escada.

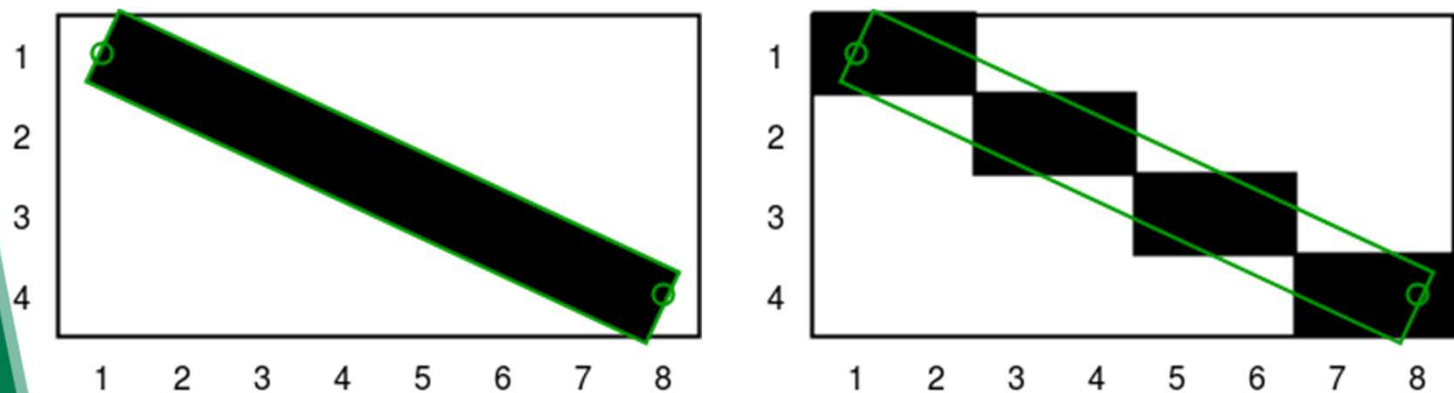


Figura 3. Linha renderizada.

# Técnicas discretas

- Aliasing e anti-aliasing -

- Existem diversas técnicas de AA, sendo as mais conhecidas são a por:
  - Superamostragem
    - É a técnica mais antiga e mais simples.
  - Multi Amostra
    - É essencialmente a técnica de AA por superamostragem, utilizando-se menos amostras e aplicado-se apenas onde é realmente necessário.
      - Resulta num melhor desempenho e qualidade da cena.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- A técnica AA por superamostragem:
  - É, também, conhecida como super sampling anti aliasing (SSAA), anti-aliasing de cena completa ou full sceen anti-aliasing (FSAA).
  - Consiste em renderizar a cena numa maior resolução e, em seguida, misturar as amostras dessa renderização de forma a que a cena final seja do tamanho pretendido, suavizando os *jaggies* das suas linhas.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- A amostragem (sampling) utilizada é, tipicamente, a do vizinho mais próximo (normalmente numa caixa em torno do *pixel*).
  - As amostras são definidas para qualquer posição dentro dessa caixa.
  - Se escolhidas aleatoriamente, a amostragem chama-se estocástica.
  - Geralmente, mais amostras implica melhor resultado da aplicação da AA.

# Técnicas discretas

- Aliasing e anti-aliasing -

- O processo de mistura das amostras para se obter a cor final do *pixel* é feito com base na média aritmética da cor das amostras.



Pixel with sample postions

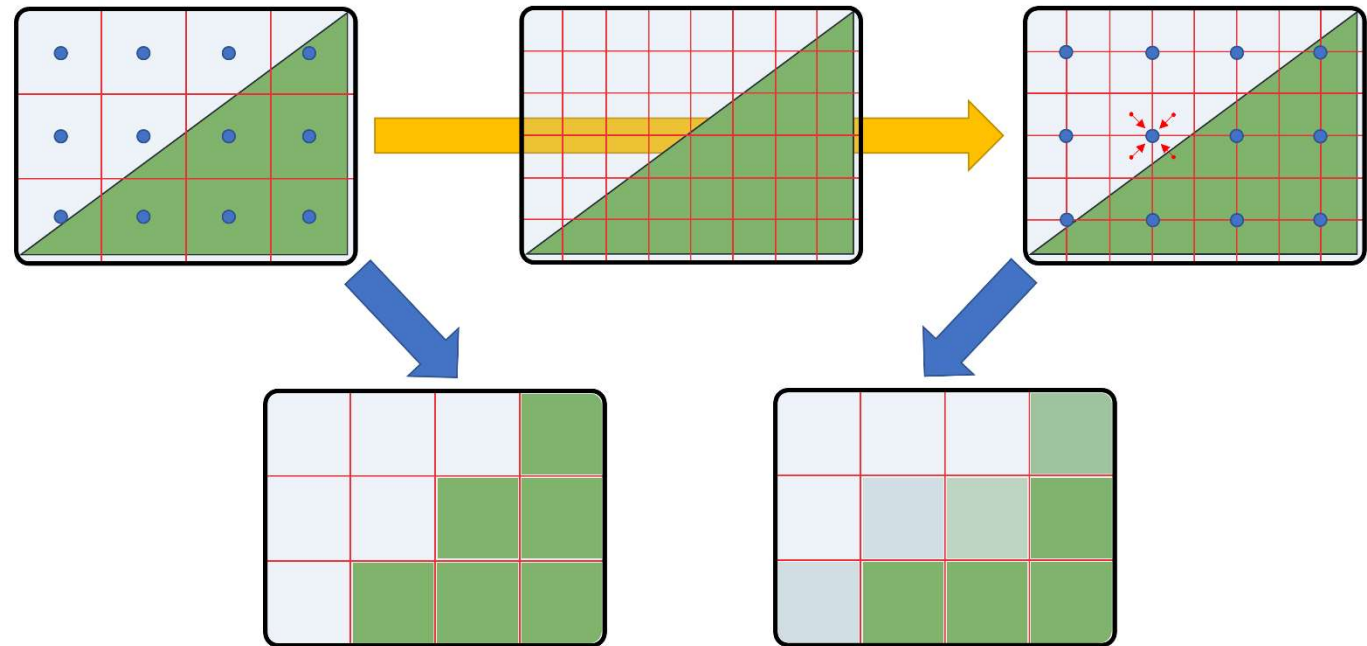
Resulting color

$$\frac{\text{white} + \text{white} + \text{yellow} + \text{brown}}{4} = \text{light yellow}$$

Figura 4. Calculo da cor final do *pixel*.

# Técnicas discretas

- Aliasing e anti-aliasing -



**Figura 5.** Aplicação da técnica 4x SSAA (cada *pixel* vai ser dividido em 4).



# Técnicas discretas

- *Aliasing e anti-aliasing* -

- As desvantagens da SSAA são:
  - Poder afetar negativamente imagens com muitas linhas horizontais ou verticais (essas linhas são nítidas por natureza, esbatendo-se na aplicação).
  - Exigir que toda a imagem seja tratada antes de suavizar as linhas (daí o nome de cena completa).
    - Como os jogos são mostrados em tempo real, o SSAA requer um processamento intenso para funcionar eficientemente (por isso já não é mais muito usado em jogos).

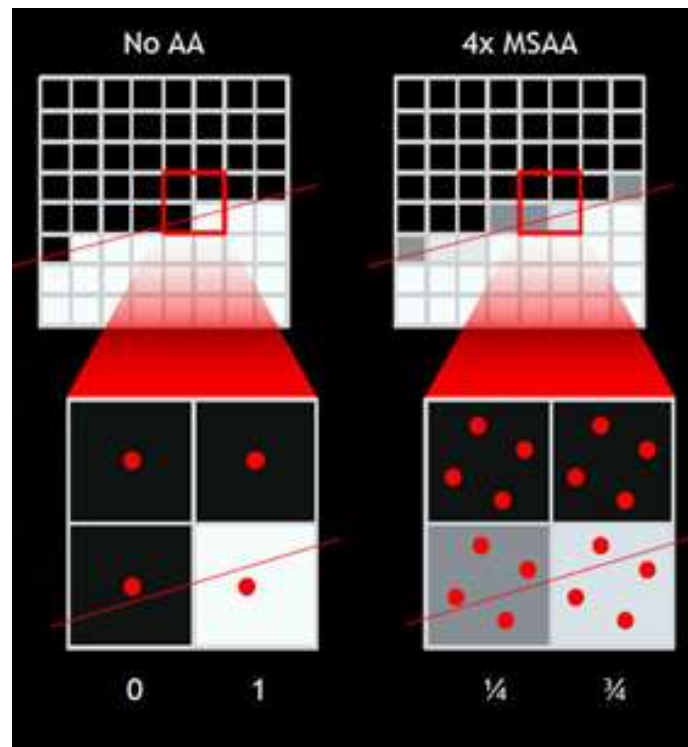
# Técnicas discretas

- *Aliasing e anti-aliasing* -

- A técnica AA por multi amostra:
  - É, também, conhecida como Multi-Sample Anti-Aliasing (MSAA) ou anti-aliasing por média de área.
  - Suaviza apenas as bordas dos polígonos usando múltiplos pontos de amostragem para determinar a cor de cada *pixel*, reduzindo um pouco (não muito) o tempo de processamento.

# Técnicas discretas

- Aliasing e anti-aliasing -



**Figura 6.** Aplicação da técnica MSAA.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- As técnicas AA de amostragem de cobertura ou coverage sampling anti-aliasing (CSAA), da NVIDIA, e AA de qualidade aprimorada ou enhanced quality anti-aliasing (EQAA), da AMD funcionam de forma semelhante.
  - A GPU deteta se um polígono está presente na imagem e, determina quais as suas partes que terão provavelmente *jaggies*. Em seguida, mostra apenas esses *pixels*.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- Assim, as técnicas CSAA e a EQAA necessitam significativamente de muito menos processamento para serem executadas, pois não suavizam a cena inteira.
- Existem, ainda, a técnica subpixel morphological anti-aliasing (SMAA), que combina o MSAA com um filtro de suavização para produzir linhas suaves e sem desfocagem.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- Outro tipo de *aliasing* é o temporal (AT), também, conhecido como *flicker* ou cintilação.
  - É um fenômeno visual que ocorre quando uma sequência de imagens em movimento são exibidas numa frequência abaixo do limite de percepção do olho humano, fazendo com que a imagem pareça estar a piscar ou a tremer.

# Técnicas discretas

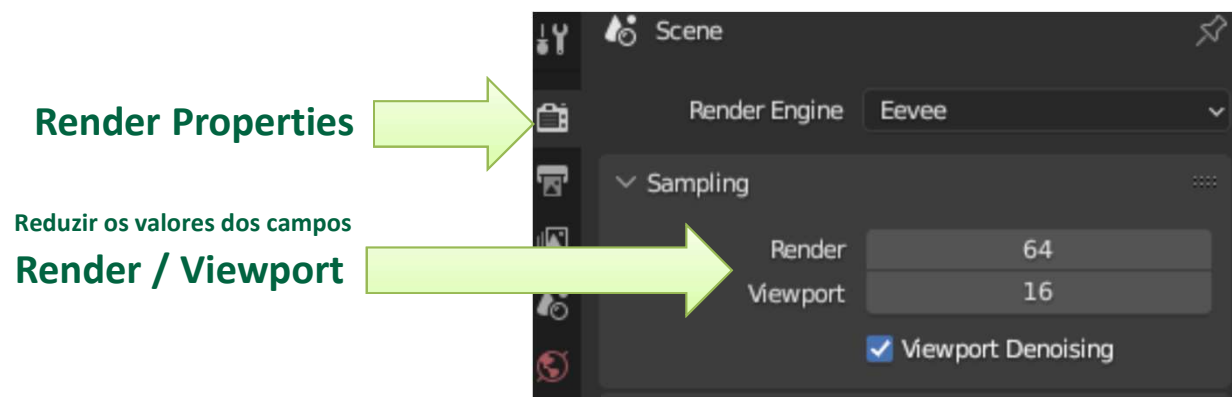
- *Aliasing e anti-aliasing* -

- Uma técnica para resolver o AT é a temporal anti-aliasing (TXAA), que usa a informação contida em várias imagens consecutivas para suavizar a cintilação.
  - Procura manter um nível de movimento suave à medida que o jogador se move no ambiente virtual.
  - É a técnica que exige mais recursos do sistema.

# Técnicas discretas

- *Aliasing e anti-aliasing* -

- Para alterar os parâmetros do *anti-aliasing* no Eevee (*Render Engine*), o Blender tem que se usar os campos do *Sampling*, *Render* e *Viewport*, no painel *Render Properties*.



**Figura 7.** Técnicas de *anti-aliasing* no Blender.



# Técnicas discretas

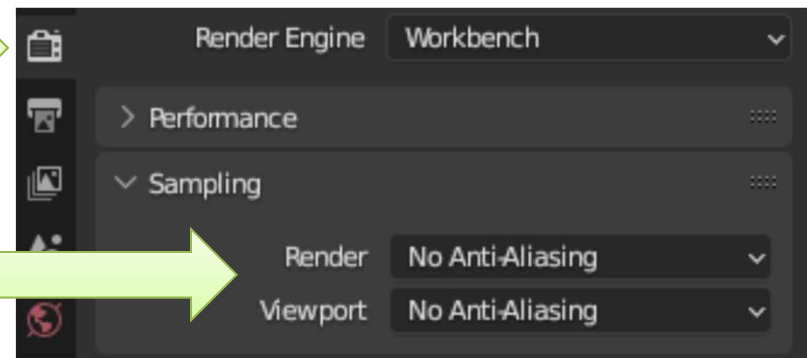
- Aliasing e anti-aliasing -

- Para alterar os parâmetros do *anti-aliasing* no Workbench (*Render Engine*), o Blender tem que se usar os campos do *Sampling*, *Render* e *Viewport*, no painel *Render Properties*.

Render Properties



Reduzir os valores dos campos  
Render / Viewport



**Figura 8.** Técnicas de *anti-aliasing* no Blender.

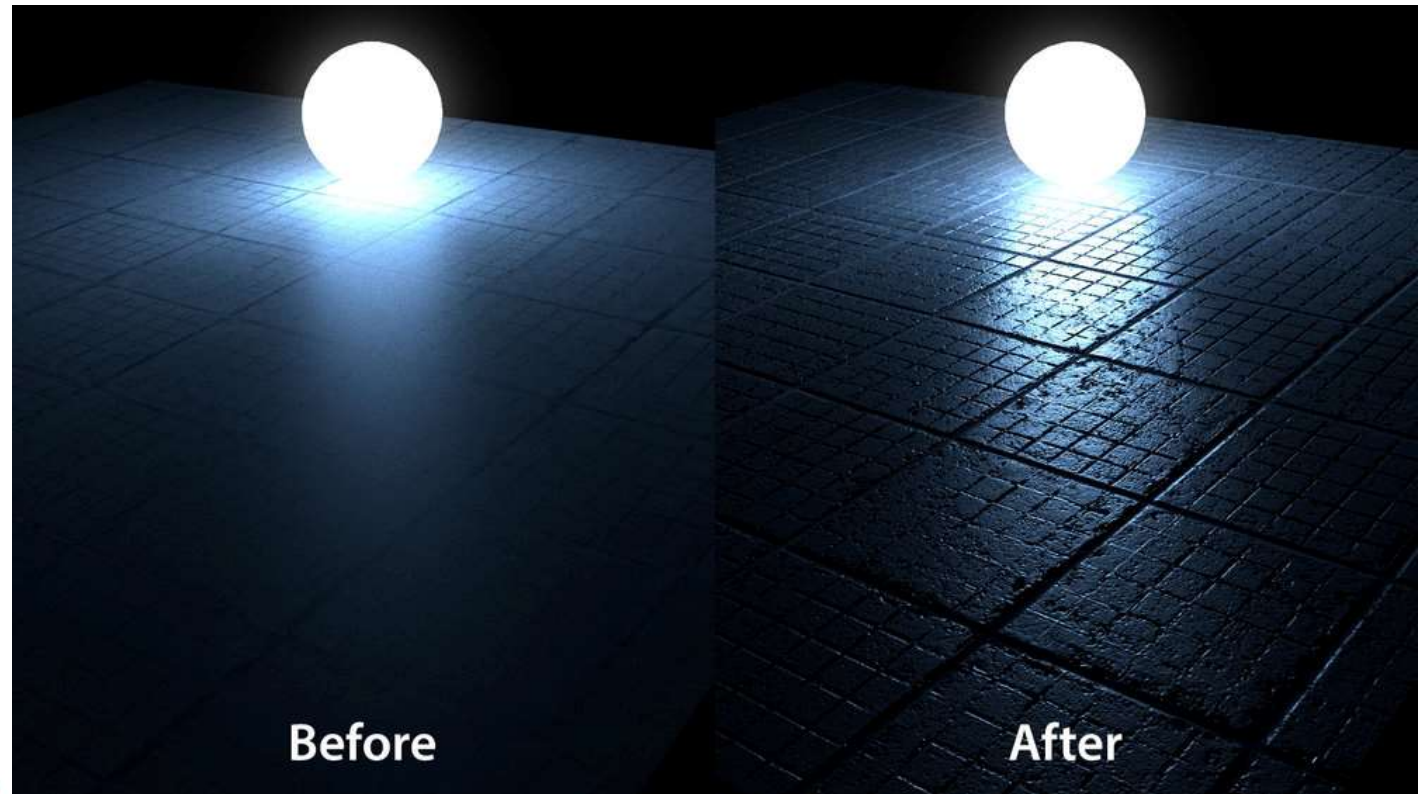
# Técnicas discretas

- *Aliasing e anti-aliasing* -

- O Blender não permite alterar os parâmetros do *anti-aliasing* no Cycles (*Render Engine*).
- Um PC com hardware para jogos:
  - de nível básico, deve usar as técnicas SMAA e CSAA/EQAA.
  - de nível médio, deve usar as técnicas SMAA ou MSAA.
  - de nível superior, deve usar as técnicas SSAA, MSAA e TXAA.

# Técnicas discretas

- Texturas -



# Técnicas discretas

- Introdução às texturas -

- Os modelos de iluminação usados na computação gráfica nem sempre descrevem todas as propriedades da superfície do objeto, limitando a capacidade de se obter uma representação realistas do objeto.
  - É possível obter esses detalhes com o acréscimo de componentes na geometria da superfície ou usando materiais com propriedades óticas distintas.

# Técnicas discretas

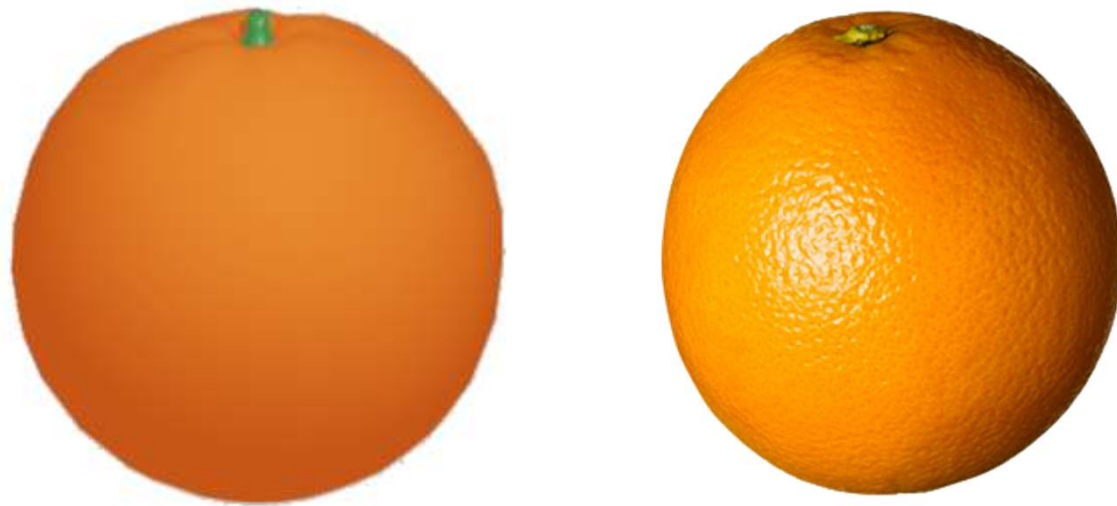
- Introdução às texturas -

- De forma a
  - evitar modelações demoradas e complexas,
  - reduzir o custo computacional para a obtenção dos objetos,
  - aumentar o seu realismo,esses detalhes podem ser obtidos através do uso de:

**Texturas (ou mapas de textura)**

# Técnicas discretas

- Introdução às texturas -



**Figura 9.** Modelo de uma laranja, sem recurso a texturas (à esquerda) e com aplicação de texturas (à direita).



# Técnicas discretas

- Introdução às texturas -

- As texturas, propostas por Catmull em 1975, permitem distinguir objetos geometricamente iguais.



**Figura 10.** Esferas geometricamente iguais, com texturas diferentes.

# Técnicas discretas

- Introdução às texturas -

- Uma textura é um conjunto de valores (padrão ou mapa de *bits*), organizada usualmente em *arrays*, e que é aplicada (ou mapeada) na superfícies dos objetos.
- Os elementos de cada textura são conhecidos como *texels* (*texture elements*), e a origem do sistema de coordenadas de textura é o seu canto superior esquerdo.



# Técnicas discretas

- Introdução às texturas -

- As texturas podem ser classificadas segundo a sua dimensão, como:

- Unidimensionais (1D) 

- Este tipo de texturas é normalmente usada para fazer dégradé em superfícies.

- Bidimensionais (2D) 

- É o tipo de textura mais usado.

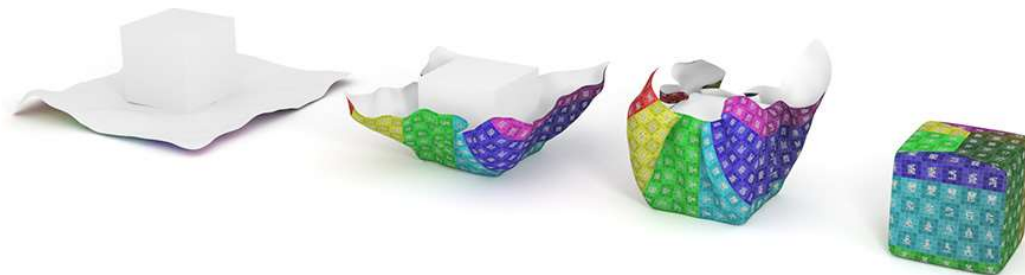
- Tridimensionais (3D) 

- Este tipo de texturas não é muito usado, pois consome muita memória. As coordenadas 3D da textura equivalem às mesmas coordenadas 3D da superfície.

# Técnicas discretas

- Mapeamento de texturas -

- O processo de aplicação de uma textura à superfície de um objeto é chamado de **mapeamento de textura** e, em termos simplistas, é como “embrulhar” (ou *wrap*) a superfície do objeto com a textura.

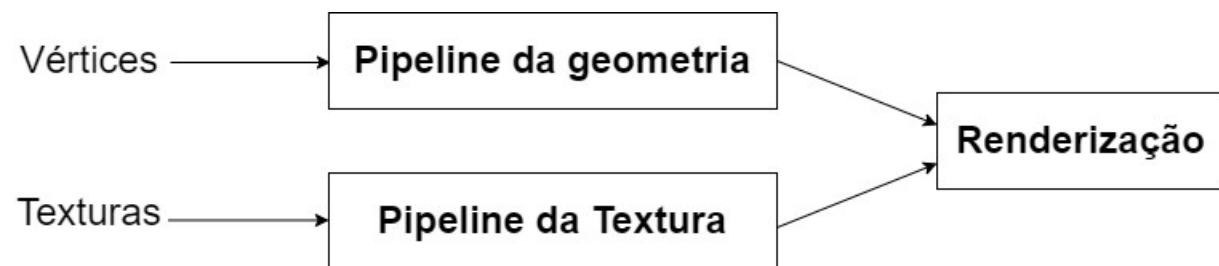


**Figura 11.** Mapeamento de texturas.

# Técnicas discretas

- Mapeamento de texturas -

- O mapeamento de texturas é feito durante a fase de renderização.
  - As texturas e a geometria do objeto seguem caminhos separados no *pipeline* gráfico, encontrando-se na renderização.



**Figura 12.** *Pipeline* gráfico.

# Técnicas discretas

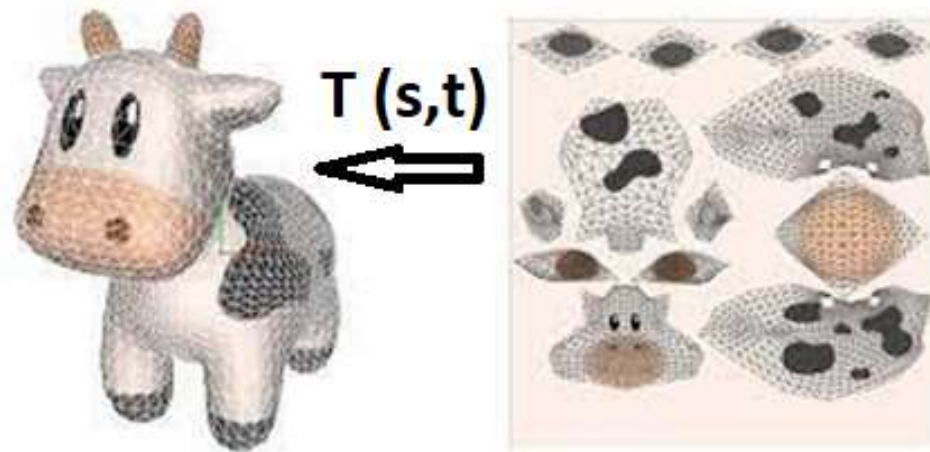
- Mapeamento de texturas -

- O mapeamento de texturas consiste em relacionar os *texels* da textura com os fragmentos da superfície do objeto, de forma a lhes alterar:
  - os coeficientes da reflexão difusa (a cor), da reflexão especular e difusa (o brilho e a refletividade – mapeamento de ambiente) e da transparência;
  - o vetor normal e a geometria da superfície na direção normal.

# Técnicas discretas

- Mapeamento de texturas -

- A relação entre os *texels* da textura com os fragmentos da superfície do objeto é definida com base numa função  $T$ .



**Figura 13.** Função  $(x, y, z) = T(s, t)$ , para uma textura 2D.

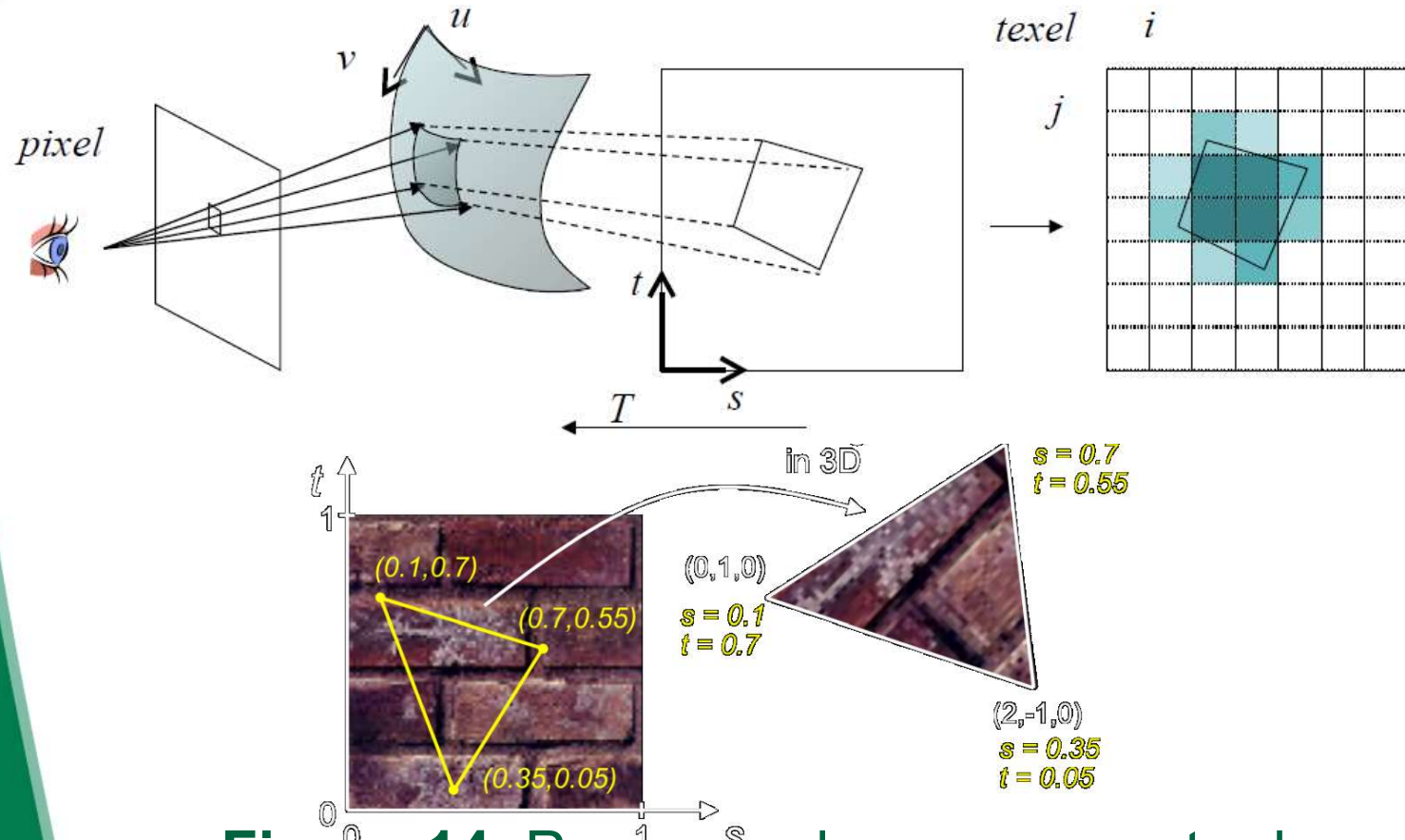
# Técnicas discretas

- Mapeamento de texturas -

- Mais tecnicamente,
  - Durante a renderização, as superfícies dos objetos são projetadas num *buffer* (em coordenadas da câmara);
  - Para cada fragmento das superfícies, as coordenadas dos *texels* são calculadas através de interpolação;
  - O valor na textura, associado às coordenadas dos *texels*, é usado para atribuir/modificar o que está no *buffer* (ou seja, as propriedades da superfície).

# Técnicas discretas

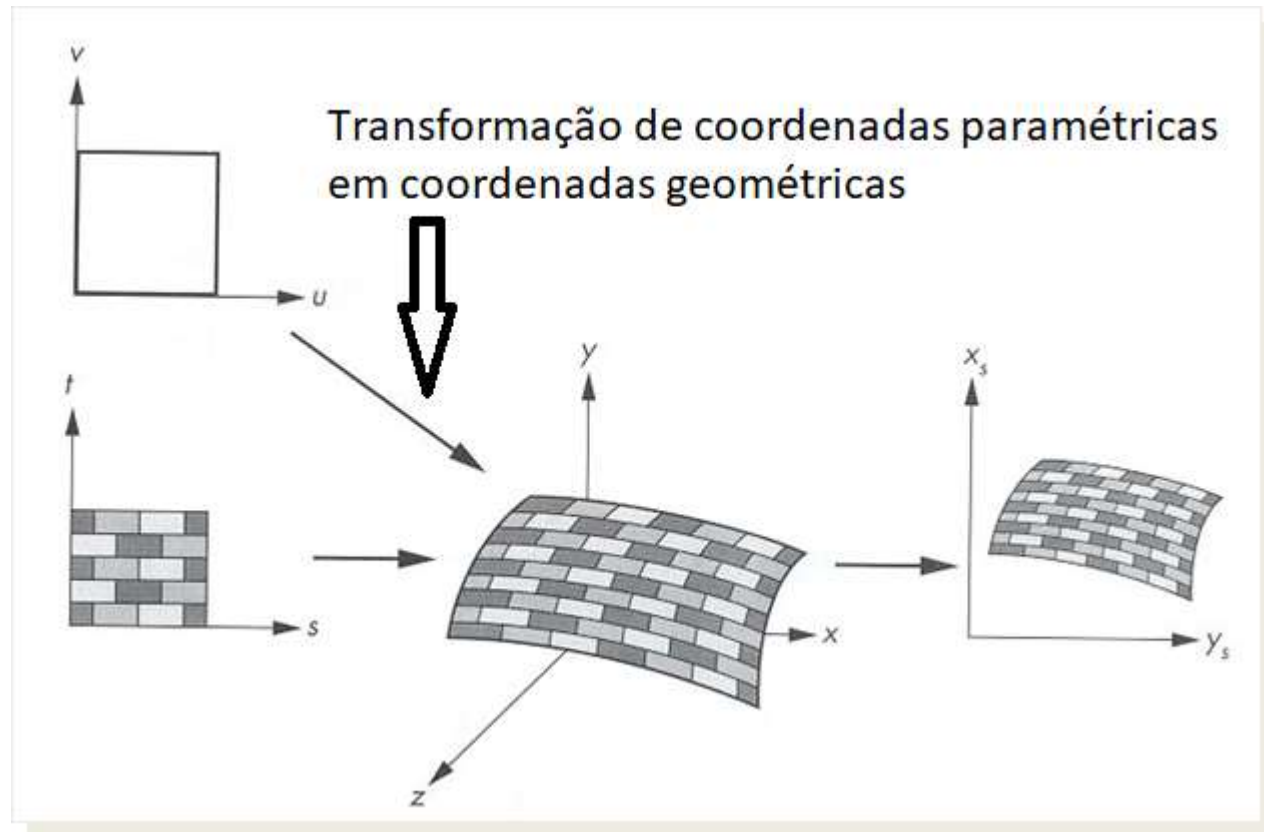
- Mapeamento de texturas -



**Figura 14.** Processo de mapeamento de texturas, na fase de renderização, para uma textura 2D.

# Técnicas discretas

- Mapeamento de texturas -

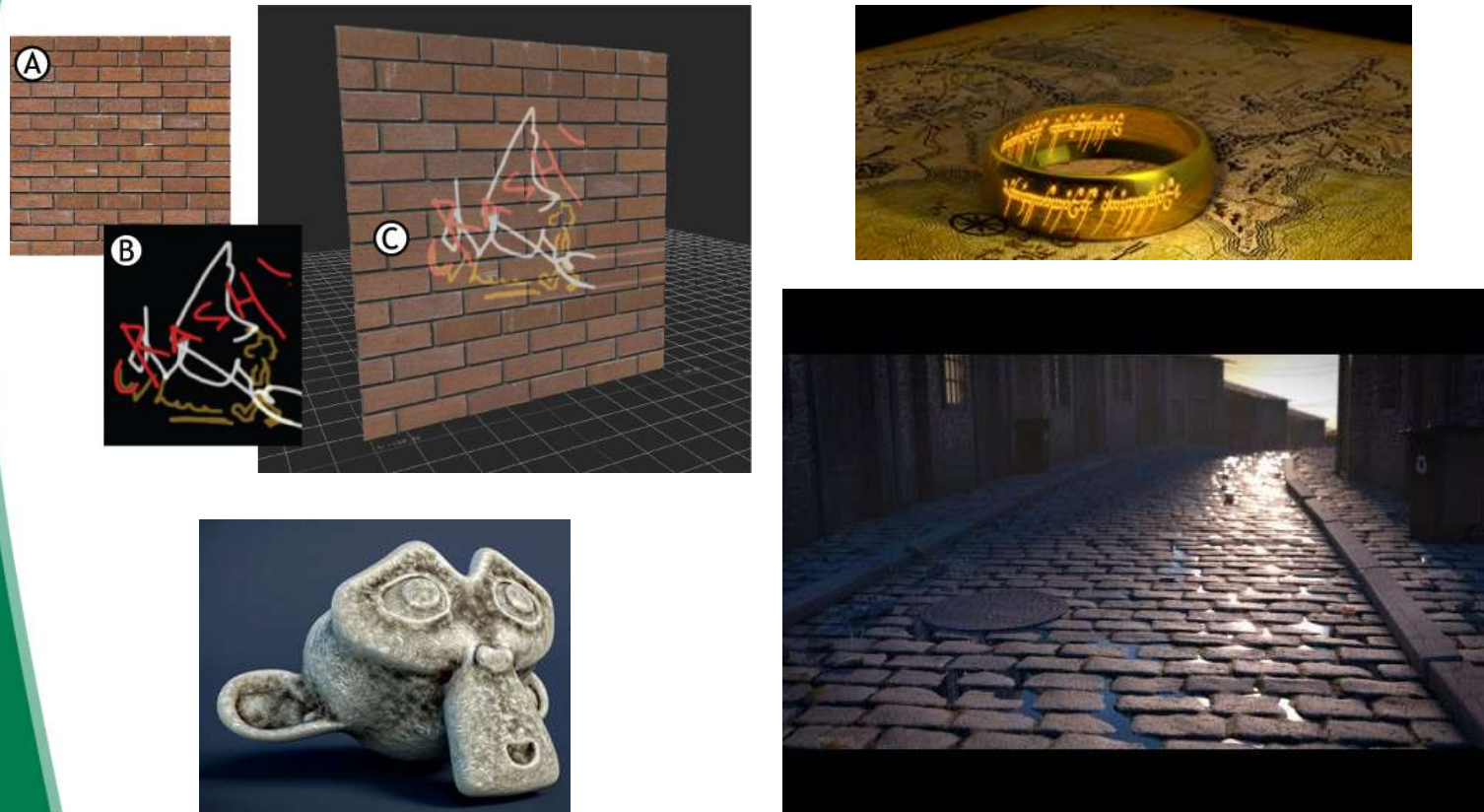


**Figura 15.** Processo de mapeamento de texturas para uma superfície paramétrica, na fase de renderização, para uma textura 2D.



# Técnicas discretas

- Mapeamento de texturas -



**Figura 16.** Exemplos de aplicação de mapeamento de texturas.

# Técnicas discretas

- Mapeamento de texturas -

- Dado que as coordenadas de textura tem um intervalo de variação. em ambas as direções. de  $[0.0, 1.0]$ , o processo de mapeamento deve assumir que o padrão da textura se repete fora desse intervalo, em qualquer direção.
  - Permitindo a repetição dos padrões da textura nas superfícies dos objetos.

# Técnicas discretas

- Mapeamento de texturas -



**Figura 17.** Repetição do padrão da texturas.

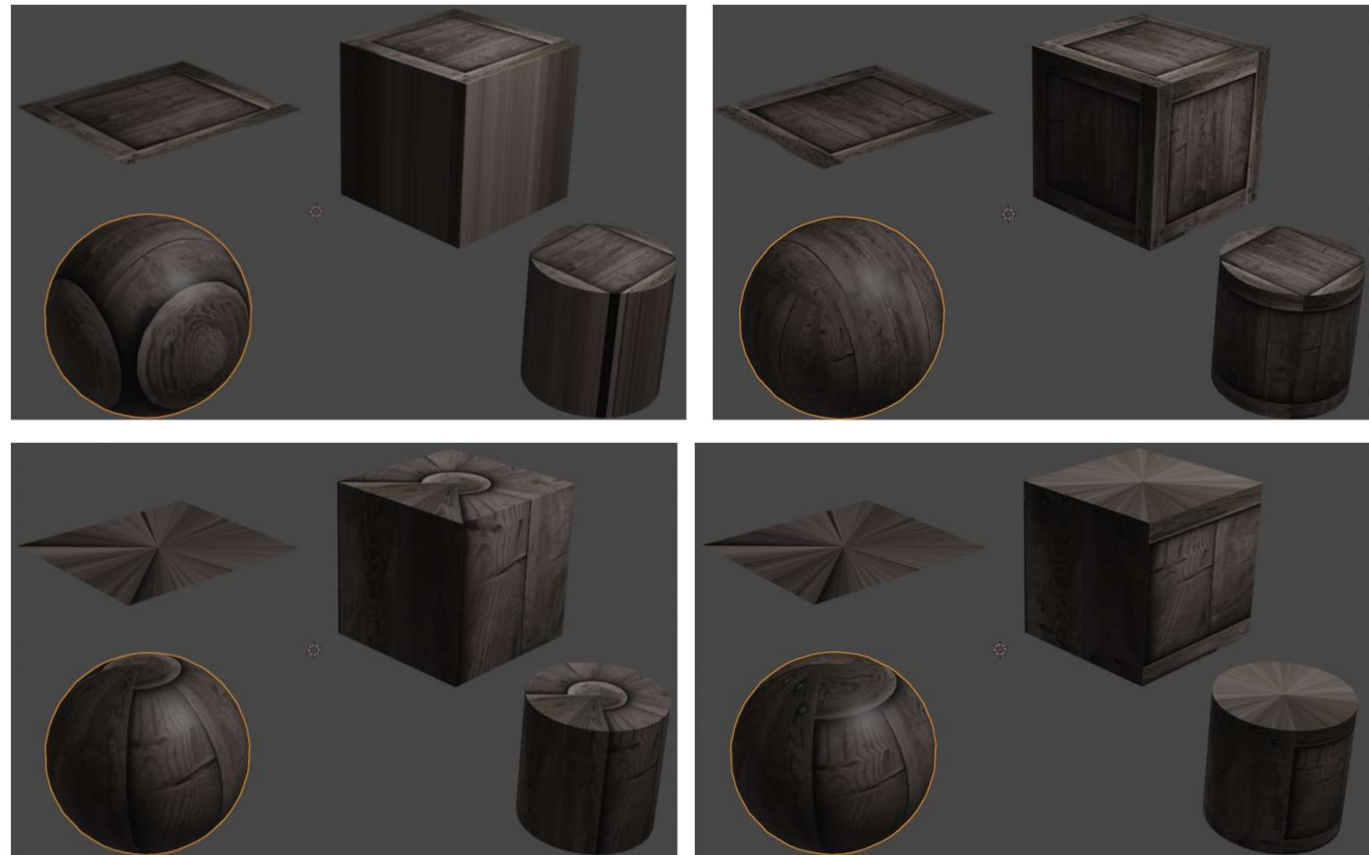
# Técnicas discretas

- Mapeamento de texturas -

- Os mapeamentos podem ser:
  - Planar (as coordenadas de textura são mapeadas ortogonalmente);
  - Cúbico (as coordenadas de textura são mapeadas ortogonalmente nos seis planos de um cubo);
  - Esférico (as coordenadas de textura são mapeadas segundo coordenadas polares esféricas);
  - Cilíndrico (as coordenadas de textura são mapeadas segundo coordenadas polares cilíndricas).

# Técnicas discretas

- Mapeamento de texturas -



**Figura 18.** Mapeamento planar (topo esquerdo), cúbico (topo direito), esférico (baixo esquerdo) e cilíndrico (baixo direito), realizado no blender.



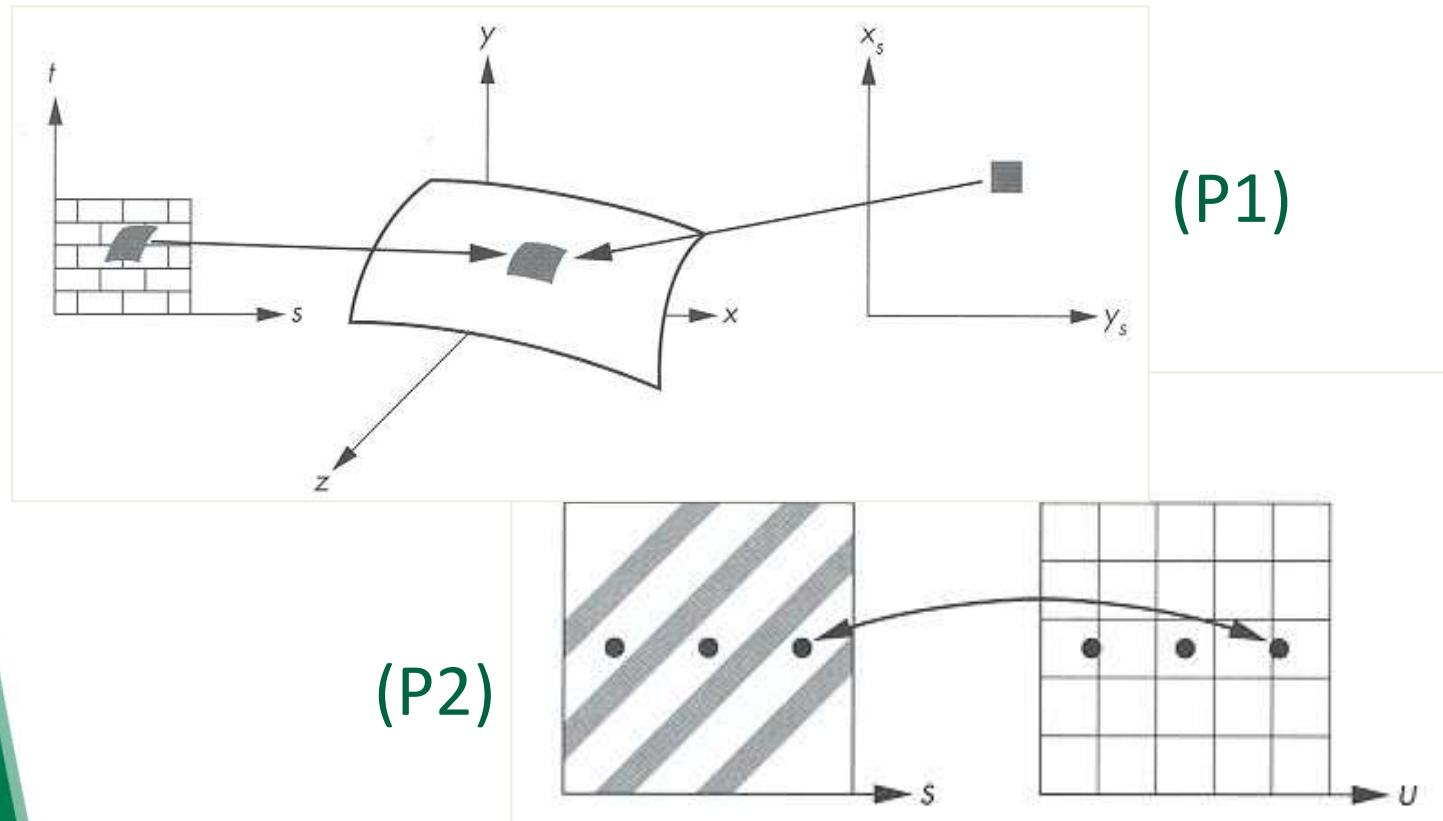
# Técnicas discretas

- Problemas com o mapeamento de texturas -

- Os principais problemas que se colocam com o mapeamento de texturas são:
  - O mapeamento ser baseado no formato retangular, criando para as regiões com forma arbitrária algumas dificuldades (designado por P1);
  - Como se pretende mapear áreas em áreas e não pontos em pontos, existe a possibilidade de ocorrência de *aliasing* (designado por P2).

# Técnicas discretas

- Problemas com o mapeamento de texturas -



**Figura 19.** Principais problemas (P1 e P2) que se colocam com o mapeamento de texturas.

# Técnicas discretas

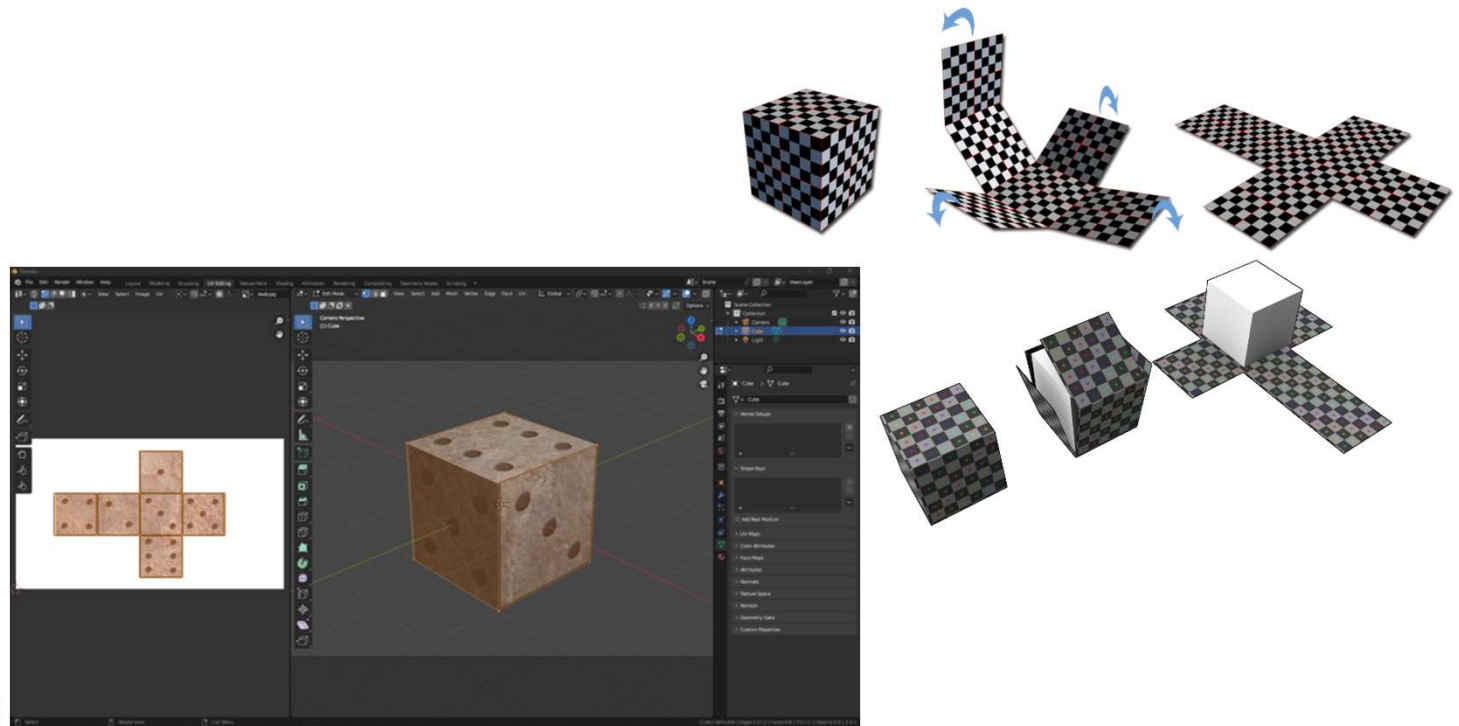
- Problemas com o mapeamento de texturas -

- As soluções para esses problemas podem passar pelo seguinte:
  - *UV Mapping* (para P1)
    - Planificar o objeto a mapear (ou fazer *UV Unwrap* – desembrulhar para ficar planar);
    - Define quais as zonas do objeto e qual a parte da textura que lhe corresponderá;
    - Aplica-se o mapeamento a cada parte.
  - Técnicas de anti-aliasing (para P2)
    - Uma delas é usar Mipmaps (conjunto de texturas em diferentes resoluções, criados a partir da original).



# Técnicas discretas

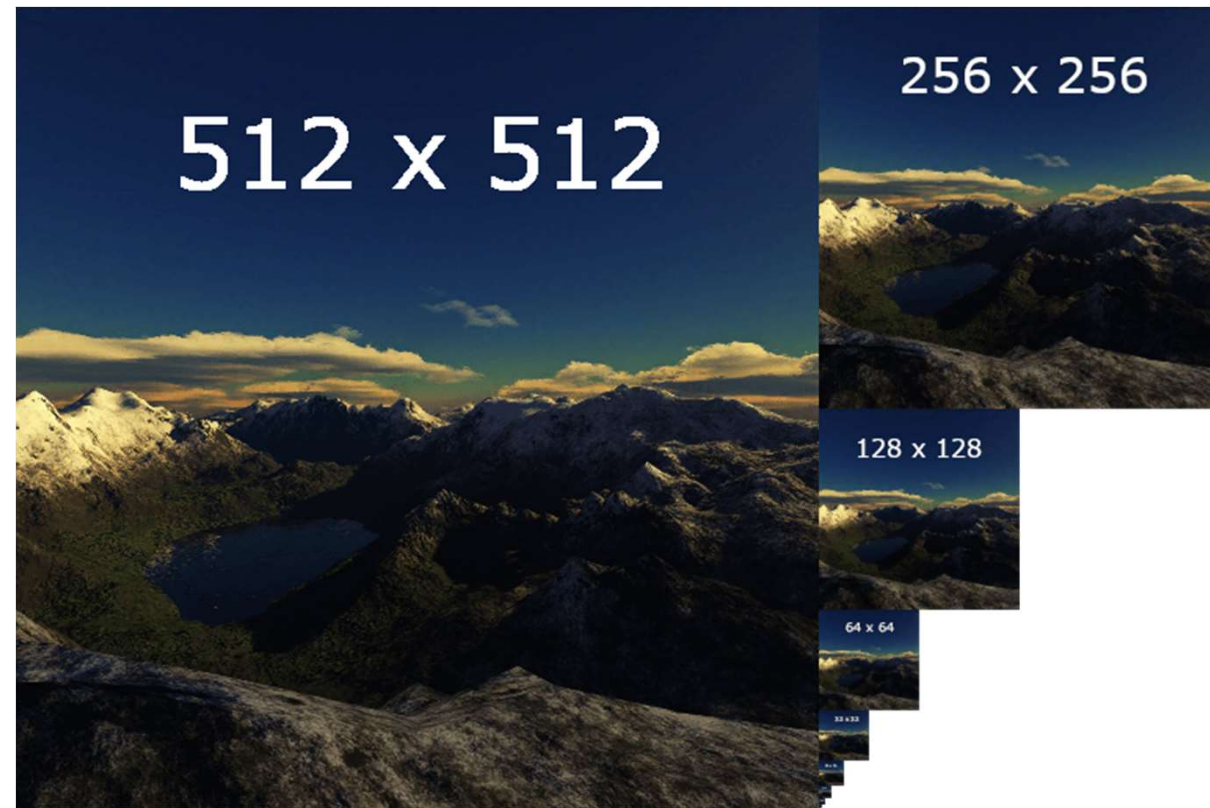
- Problemas com o mapeamento de texturas -



**Figura 20.** Exemplo de *UV mapping*, com o *UV Unwrapping* feito no *Blender*.

# Técnicas discretas

- Problemas com o mapeamento de texturas -



**Figura 21.** Exemplo de *mipmaps*.

# Técnicas discretas

- Problemas com o mapeamento de texturas -

- Os *mipmaps* são:
  - produzidos para várias resoluções da textura original (de resolução total);
  - usados para situações em que a distância entre um objeto e a câmara pode mudar;
  - escolhidos aquando do mapeamento da textura tendo em conta a relação de 1:1 entre os *texels* e os *pixels*.
  - armazenados com a textura original, requerendo mais memória.

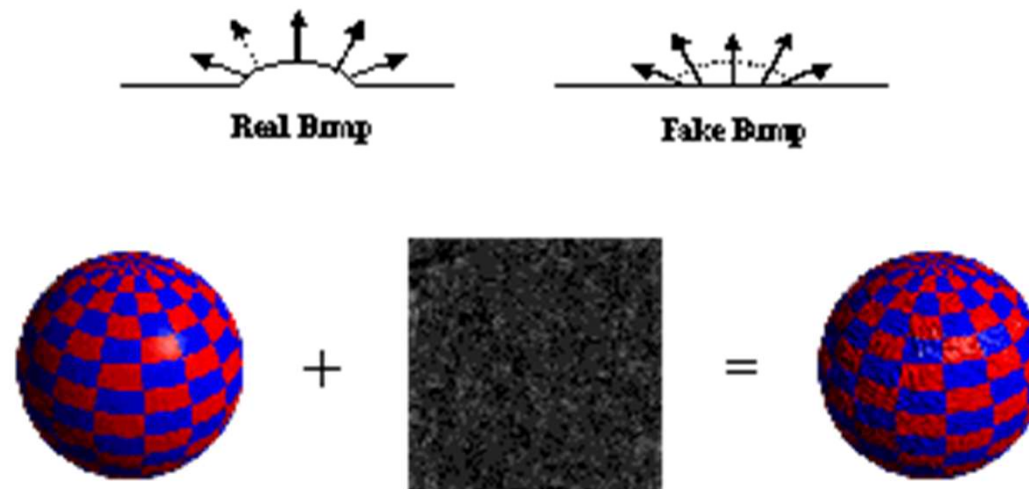
# Técnicas discretas

- Mapeamento *Bump* -

- O mapeamento *Bump* utiliza um padrão para distorcer a geometria da superfície (com base no vetor normal) para criar variações locais.
- A alteração que a textura provoca no vetor normal é feita sem fazer nenhuma mudança real na geometria do objeto (usa apenas a textura).

# Técnicas discretas

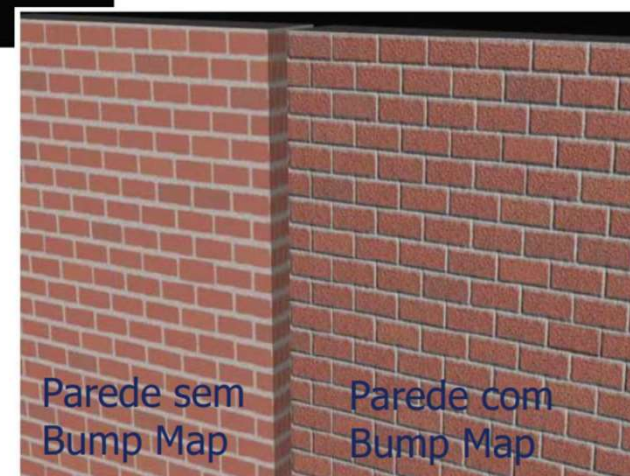
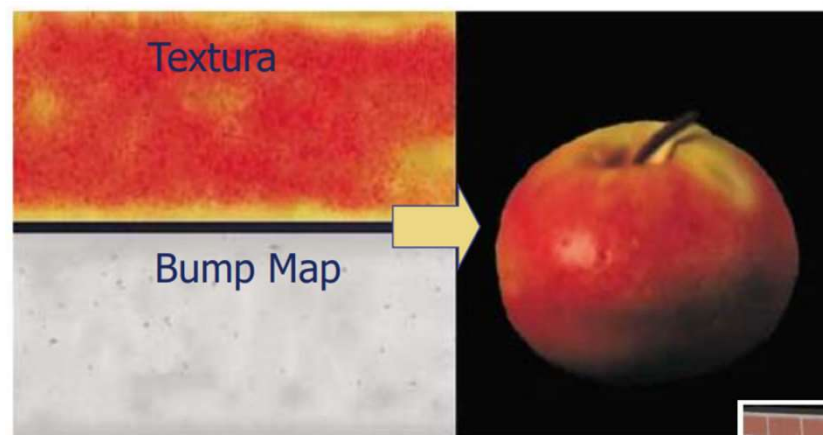
- Mapeamento *Bump* -



**Figura 22.** Mapeamento *Bump*. A superfície não muda. É o sombreamento que faz com que pareça mudada. O bump map causa deslocamentos acima e abaixo da superfície.

# Técnicas discretas

- Mapeamento *Bump* -



**Figura 23.** Exemplos de mapeamento *Bump*.

# Técnicas discretas

- Mapeamento de deslocamento -

- O mapeamento de texturas que usa o padrão para alterar a geometria da superfície na direção normal designa-se por mapeamento de deslocamento (ou *displacement mapping*).
  - É uma técnica usada para adicionar detalhes a uma superfície de forma mais eficiente do que com geometria adicional.



# Técnicas discretas

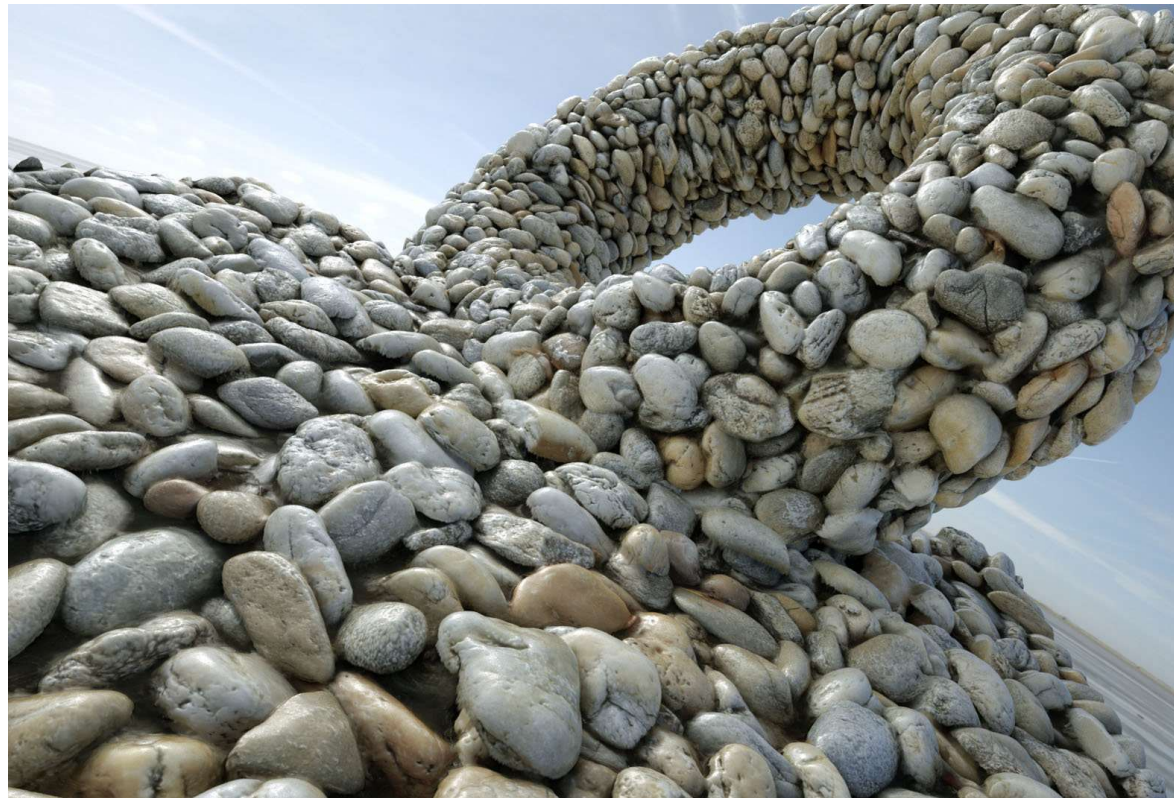
- Mapeamento de deslocamento -

- Nesta técnica, a textura é usada para deslocar a geometria da superfície, dando a ilusão de que tem mais detalhes do que realmente tem.
  - A textura é usada para deslocar os vértices em diferentes direções, criando saliências, depressões e outros detalhes na superfície.
  - Pode ser usada em vários tipos de superfícies, incluindo terrenos, madeira, peles, entre outros.



# Técnicas discretas

- Mapeamento de deslocamento -



**Figura 24.** Exemplo de mapeamento de deslocamento.

# Técnicas discretas

- *Bump mapping vs. Displacement mapping* -

	<b>Bump Mapping</b>	<b>Displacement Mapping</b>
Utilização de memória	Utiliza a memória usada pela textura (idealmente em escala de cinzas).	Utiliza mais memória para 16/32 bits. Usa potencialmente muita memória para os muitos polígonos pequenos.
Velocidade de renderização	Pequena desaceleração, pois requer múltiplas amostras de textura.	Mais lento no raytracing/PBR, porque adiciona polígonos.
Tempo de arranque	Pequeno impacto, devido à carga de textura.	Impacto maior com o Raytracing/PBR, porque o renderizador deve gerar a geometria deslocada.

# Técnicas discretas

- *Bump mapping vs. Displacement mapping* -



Base  
Model



Bump  
Mapping



Displacement  
Mapping

Image courtesy of [www.chromosphere.com](http://www.chromosphere.com)

**Figura 25.** Exemplo de mapeamento *Bump* vs. Mapeamento de deslocamento.

# Técnicas discretas

- Geração de texturas -

- As **texturas** mais usadas na área da computação gráfica são as **baseadas em imagens** (texturas 2D).
- O padrão destas texturas é igual ao conteúdo da imagem.
- Podem gerar problemas quando a sua escala difere da escala da superfície a ser mapeada.
- O *Blender* permite lidar com este tipo de texturas.

# Técnicas discretas

- Geração de texturas -

- As texturas baseadas em imagens têm a vantagem de serem de fácil obtenção (basta tirar uma fotografia ou ir buscar à internet) e de demorarem pouco tempo a serem processadas (por isso, são muito usadas nos jogos).
- As texturas baseadas em imagens têm a desvantagem de ocupar muito espaço.

# Técnicas discretas

- Geração de texturas -

- Também podem existir texturas baseadas em algoritmos, as quais se designam por **texturas procedimentais**.
- As texturas procedimentais funcionam para qualquer dimensão, são fáceis de implementar e são, normalmente, constituídas por uma certa aleatoriedade.

# Técnicas discretas

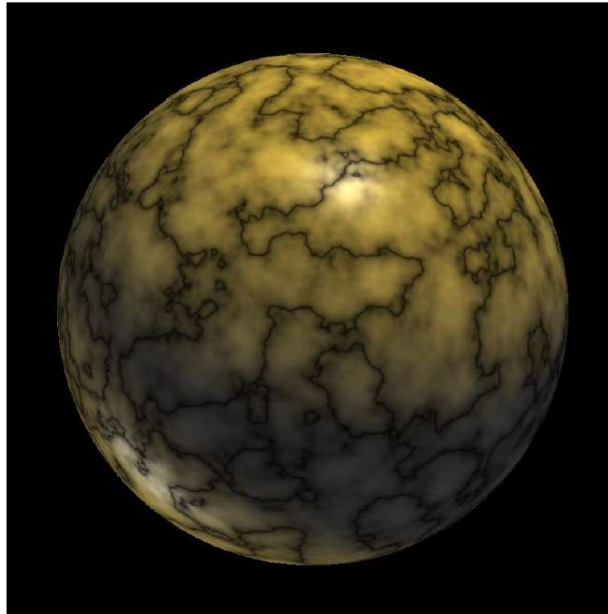
- Geração de texturas -

- As texturas procedimentais têm normalmente padrões de grande complexidade e irregularidade geométrica, semelhantes aos que se vê na natureza (como madeira, nuvens, mármore, etc.).
- O seu padrão pode ser construído por funções matemáticas ou um qualquer algoritmo.

# Técnicas discretas

- Geração de texturas -

2D



3D



**Figura 26.** Exemplos de aplicação de diferentes texturas procedimentais.



# Técnicas discretas

- Geração de texturas -

- Por exemplo, uma aproximação para se criar o padrão de uma textura procedural consiste em utilizar um gerador de ruído (para dar aleatoriedade), seguido da aplicação de um filtro.



**Figura 27.** Geração de uma textura procedural.

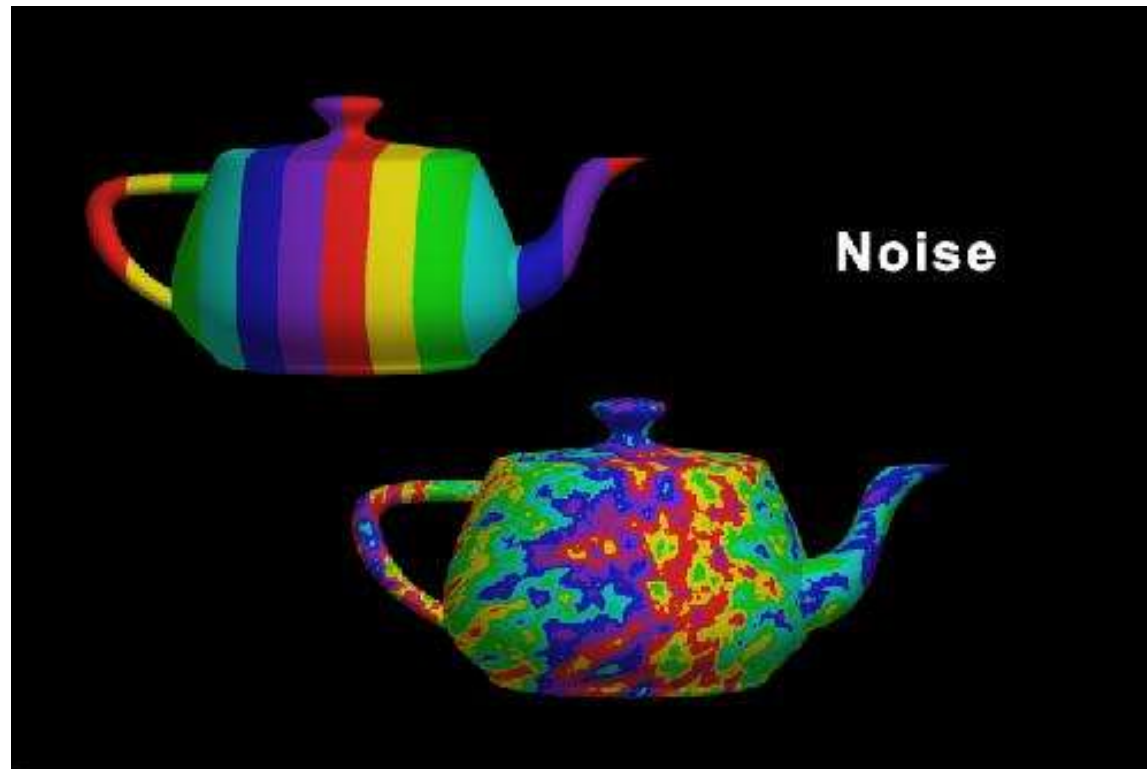
# Técnicas discretas

- Geração de texturas -

- Existem uma série de geradores de ruído, desde o deslocamento para algumas cores até o uso de funções trigonométricas e fratais.
  - Alguns dos métodos de ruído mais conhecidos são os seguintes:
    - *Perlin Noise*;
    - *Worley Noise* (também conhecido como *Voronoi Noise*);
    - *Fractal Noise*;
    - *Gradient Noise*.

# Técnicas discretas

- Geração de texturas -



**Figura 28.** Aplicação de diferentes geradores de ruído na criação de texturas procedimentais.

# Técnicas discretas

- Geração de texturas -

- As texturas procedimentais têm a vantagem de ocupar pouco espaço e de se poder facilmente fazer *zoom in* sem perda de qualidade.
  - Uma textura procedural mal programada pode perder todas essas vantagens.
- As texturas procedimentais têm a desvantagens de demorarem mais tempo a serem processadas.

# Técnicas discretas

- Geração de texturas -

- O *Blender* lida com as seguintes texturas procedimentais:

Brick Texture

Checker Texture

Environment Texture

Gradient Texture

IES Texture

Magic Texture

Musgrave Texture

Noise Texture

Point Density

Sky Texture

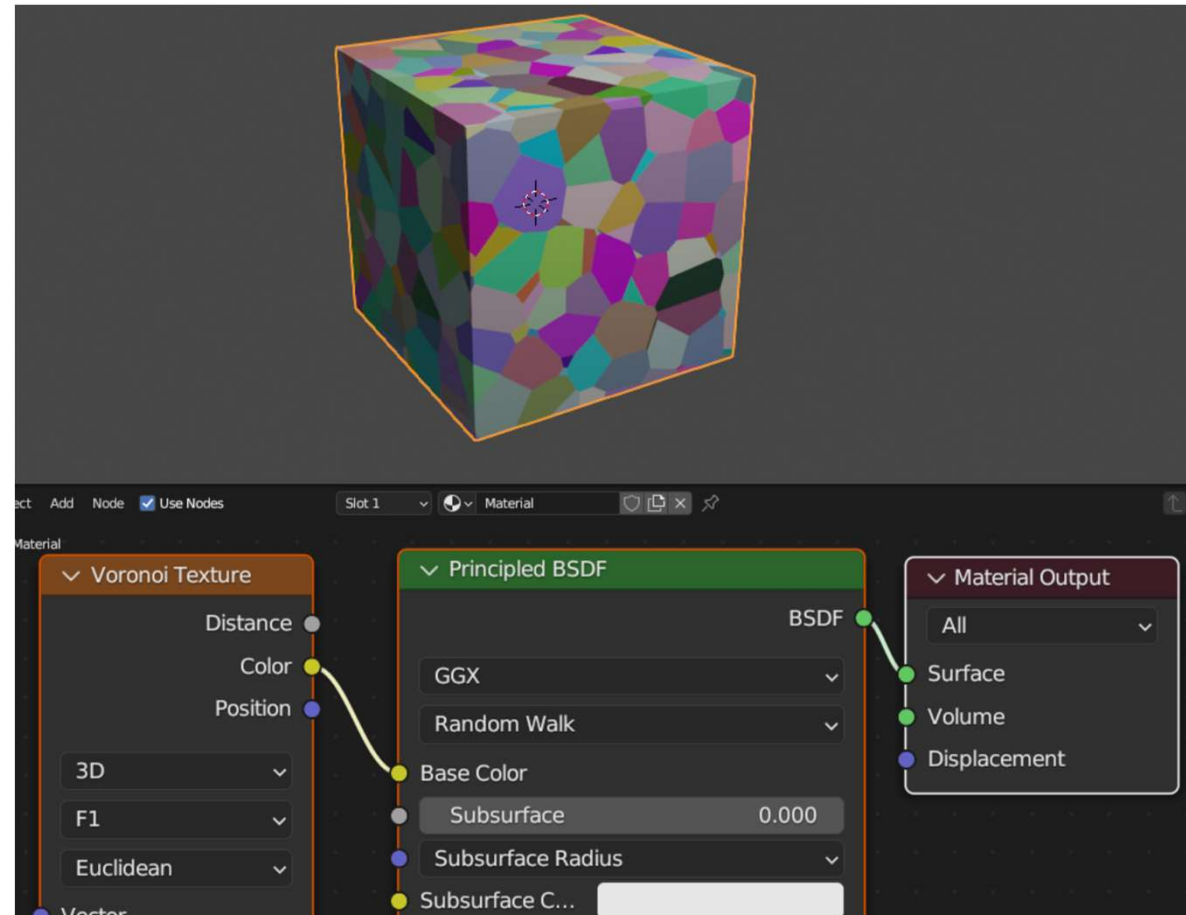
Voronoi Texture

Wave Texture

White Noise Texture

# Técnicas discretas

- Geração de texturas -



**Figura 29.** Aplicação de uma textura procedural (Voronoi) a um cubo, no *Blender*.

# Técnicas discretas

- Geração de texturas -



**Figura 30.** Aplicação de texturas procedimentais a um avião.

# Técnicas discretas

- Nevoeiro -

- Nas imagens em computador, a definição é constante ao longo da cena no sentido da profundidade, isto é, a profundidade de campo é infinita dentro do volume de visualização.
- O aspeto da cena pode tornar-se pouco realista devido a uma definição exagerada para pontos mais distantes.



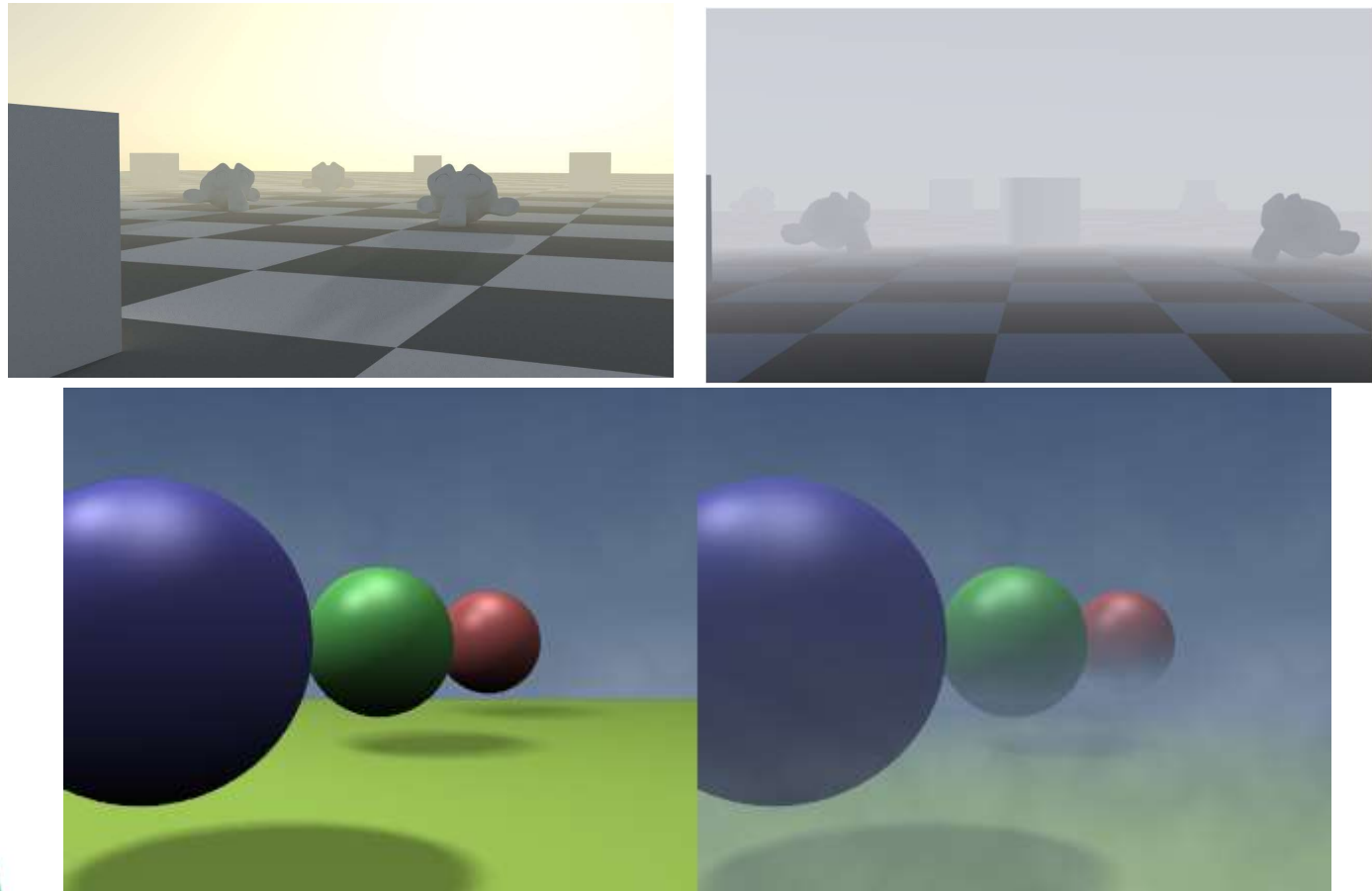
# Técnicas discretas

- Nevoeiro -

- Acrescentando o efeito de nevoeiro (ou *fog*), os objetos na cena vão-se desvanecendo com a distância.
  - Com este efeito ligado, os objetos colocados mais longe do centro de projecção começam a desvanecer para a cor do nevoeiro;
  - Serve para, além de nevoeiro, simular fumo, poluição ou qualquer efeito de limitação de visibilidade.

# Técnicas discretas

- Nevoeiro -



**Figura 31.** Imagens resultantes da aplicação do nevoeiro.

# Técnicas discretas

- Nevoeiro -

- O efeito de nevoeiro é aplicado após
  - as transformações geométricas;
  - os cálculos de iluminação;
  - a aplicação de texturas.
- Assim, afeta os objetos que já sofreram esses efeitos.
- O *Blender* permite lidar com a introdução de nevoeiro.