



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Computação Gráfica**  
**AD2 - 2º semestre de 2007.**

- 1) Defina uma transformação afim. Como uma transformação afim pode ser implementada computacionalmente? (1.0 ponto).

Um espaço afim é um espaço determinado por um conjunto de pontos  $S$ , mais um conjunto de vetores  $V$ , munido de duas operações:

Diferença entre dois pontos  $P-Q = v$ , onde  $P$  e  $Q$  são pontos e  $v$  é um vetor.

Soma de um ponto com um vetor:  $P+v = Q$

Denominamos *combinação afim* de dois pontos  $P$  e  $Q$ , por um valor escalar real  $t$ , como  $P - t(Q-P)$ , que resulta em um novo ponto dado pela soma de  $P$  com o vetor  $t(Q-P)$ , que tem comprimento dado por uma fração  $t$  do comprimento original de  $Q-P$ .

Observe que apesar de não termos definido a operação de multiplicação de um ponto por escalar, podemos reescrever  $P-t(Q-P)$  como  $aP+bQ$ , onde  $a$  e  $b$  são dois escalares, desde que  $a+b = 1$ .

Uma transformação  $T$  é dita afim se é uma transformação invariante a uma combinação afim, isto é:

$$T[P+t(Q-P)] = T(P) + t[T(Q)-T(P)], \text{ ou escrevendo de outro modo:}$$

$$T[aP+bQ] = aT(P)+bT(Q), \text{ } a+b=1.$$

Uma transformação afim consiste de um mapeamento  $T(P): P \rightarrow AP+b$ , determinado por uma transformação linear  $A$  seguida de uma translação dada por um vetor  $b$ .

Como uma transformação afim pode ser caracterizada como uma transformação linear seguida de uma translação, é natural que se busque uma implementação similar à utilizada para transformações lineares, isto é, utilizando álgebra matricial. Infelizmente, a translação adicional impede que possamos representar transformações afins de forma integrada através de operações com matrizes.

Entretanto, tal limitação pode ser removida se trabalharmos em coordenadas homogêneas. Sem perda de generalidade, já que as afirmações a seguir valem para dimensões maiores, consideraremos o caso do plano afim. Em coordenadas homogêneas, um ponto do plano é representado por uma tripla  $(x,y,w)$  tal que, para todo escalar  $\lambda$ ,  $(x,y,w) \approx (\lambda x, \lambda y, \lambda w)$ , o que significa que tais pontos no espaço homogêneo são identificados com retas passando pela origem excetuando-se a própria origem no  $\mathbb{R}^3$ . Consequentemente, é possível identificar as coordenadas afins com as coordenadas homogêneas  $(x/w, y/w, 1)$ , o que significa que o plano  $w=1$  no espaço homogêneo corresponde à imersão do plano afim no  $\mathbb{R}^3$ . Sabemos que transformações lineares no  $\mathbb{R}^3$ , podem ser representadas através de matrizes  $3 \times 3$ .

$$T(P) = \begin{bmatrix} a & b & c \\ c & d & e \\ f & g & h \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Considerando um ponto  $P$ , originalmente afim com coordenadas  $(x,y,1)$ , podemos afirmar que a matriz correspondente a  $T$  será uma transformação afim puder ser decomposta em um transformação linear  $A$  seguida de uma translação dada por um vetor  $b$ , que leve um ponto afim em outro ponto afim, isto é que preserve a coordenada  $w=1$ . Podemos obter tal transformação a partir da matriz  $3 \times 3$  original, fazendo  $f=g=0$ . Consequentemente, temos que  $T(x,y,1) = (ax+by+c, cx+dy+e, fx+gy+1) = (ax+by+c, cx+dy+e, 1)$  que pode ser descrita através de uma matriz  $M$  da forma:

$$M = \begin{bmatrix} a & b & c \\ c & d & e \\ 0 & 0 & 1 \end{bmatrix},$$

onde a submatriz  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$  corresponde a componente linear da transformação afim e o vetor  $[c \ d]^T$  à componente de translação.

2) Seja a matriz

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

representando um movimento rígido do espaço em coordenadas homogêneas. Mostre que a transformação inversa é dada por

$$T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -\langle p, n \rangle \\ o_x & o_y & o_z & -\langle p, o \rangle \\ a_x & a_y & a_z & -\langle p, a \rangle \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dica: as três primeiras colunas de  $T$  são formadas por vetores ortonormais (1.0 ponto).

A matriz  $T$  pode ser escrita como o produto

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

logo,

$$\begin{aligned} T^{-1} &= \left( \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \\ &= \begin{bmatrix} n_x & n_y & n_z & 0 \\ o_x & o_y & o_z & 0 \\ a_x & a_y & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & -(n_x p_x + n_y p_y + n_z p_z) \\ o_x & o_y & o_z & -(o_x p_x + o_y p_y + o_z p_z) \\ a_x & a_y & a_z & -(a_x p_x + a_y p_y + a_z p_z) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} n_x & n_y & n_z & -\langle p, n \rangle \\ o_x & o_y & o_z & -\langle p, o \rangle \\ a_x & a_y & a_z & -\langle p, a \rangle \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

- 3) Descreva a principal deficiência da matriz de projeção abaixo. Sugira outra matriz de projeção que corrija tal deficiência (1.0 ponto).

$$T = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

O problema com tal matriz de projeção é o de que, dado um ponto do espaço com coordenadas homogêneas  $[x,y,z,1]$ , então  $T[x,y,z,1]=[nx,ny,nz,z]$ , o que significa que após a divisão pela coordenada homogênea, a profundidade  $z$  do ponto resultante é levada em  $n$ , que é a coordenada  $z$  do plano de projeção. Isto faz com que as profundidades originais dos pontos sejam perdidas após a divisão pela coordenada homogênea, o que significa que não será possível ordenar os pontos do espaço em função de suas profundidades, o que é fundamental para a solução do problema de visibilidade.

A solução é definir uma outra matriz que não achate os valores de profundidade, preservando, deste modo, a ordem relativa das profundidades dos pontos.

Isto pode ser feito determinando-se uma matriz, tal que a profundidade  $z'' = z'/w$ , resultante da coordenada projetada  $z'$ , seja uma função monotônica da profundidade  $z$  original do ponto. Para isto precisamos determinar os coeficientes  $\alpha$  e  $\beta$  na matriz abaixo, cujo produto da terceira linha pelo vetor de coordenadas especifica a equação à direita, que descreve uma função monotônica (verificar o gráfico da função ou se não há troca de sinais na primeira derivada).

$$P' = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$z'' = \frac{z'}{w'} = \frac{\alpha z + \beta}{-z}$$

Para determinar  $\alpha$  e  $\beta$  lembremos que os valores de profundidade dos pontos no plano *near* e *far* devem ser preservados ( $-n$  e  $-f$ , respectivamente), assim chegamos ao seguinte sistema:

$$\begin{aligned} \frac{\alpha(-n) + \beta}{-(-n)} &= -n, \text{ valor de } z' \text{ para o plano near} \\ \frac{\alpha(-f) + \beta}{-(-f)} &= -f, \text{ valor de } z' \text{ para o plano far} \end{aligned}$$

A solução do sistema acima indica que  $\alpha = n+f$  e  $\beta = nf$ , donde chegamos a matriz:

$$P' = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & nf \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

- 4) Faça uma pesquisa sobre o algoritmo de Weiler-Atherton para recorte de polígonos (1.0 ponto).

O algoritmo Weiler-Atherton é um algoritmo de recorte capaz de recortar um polígono côncavo, com buracos, em relação à borda de um outro polígono com as mesmas características. O polígono a ser recortado é denominado **polígono alvo** e a região de recorte é dada pelo **polígono de recorte**. Ambos os polígonos são descritos por uma lista circular de vértices.

A borda exterior dos polígonos é especificada no sentido horário enquanto que as bordas interiores, isto é, os buracos, são especificadas no sentido anti-horário. O lado interno dos polígonos, desta forma está sempre à direita das bordas.

As interseções, quando existem, ocorrem aos pares. Uma delas, quando uma aresta do polígono alvo penetra o interior do polígono de recorte, e a outra, quando uma aresta do polígono alvo sai do polígono de recorte.

Idéia do algoritmo:

1. Começamos com uma interseção de entrada.
2. Percorremos a borda externa do polígono alvo no sentido horário até encontrar uma nova interseção.
3. Neste momento, viramos para a direita e percorremos a borda externa do polígono de recorte no sentido horário, até encontrar uma nova interseção.
4. Neste ponto, retornamos ao polígono alvo e continuamos o processo até que o ponto de início seja alcançado.
5. Obs: as bordas internas do polígono alvo são percorridas no sentido anti-horário.

Inicialização:

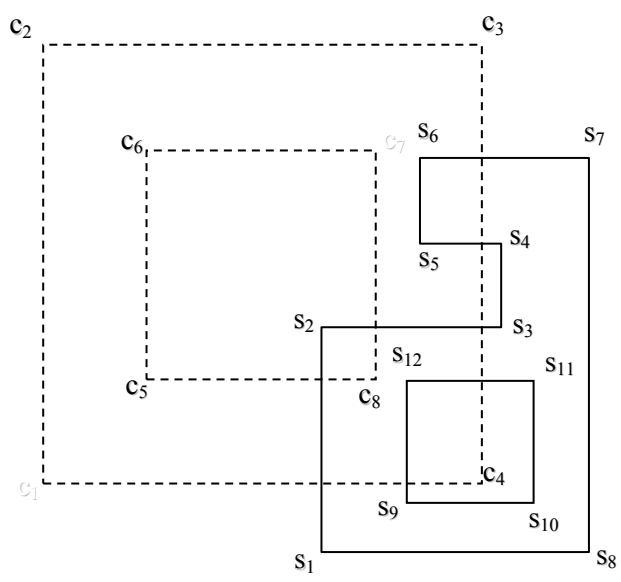
- Determine as interseções entre os polígonos alvo e de recorte.
- Adicione cada interseção às listas de vértices de ambos os polígonos.
- Crie uma ligação bidirecional entre as duas listas de vértices para cada interseção.

Crie duas listas de armazenamento.

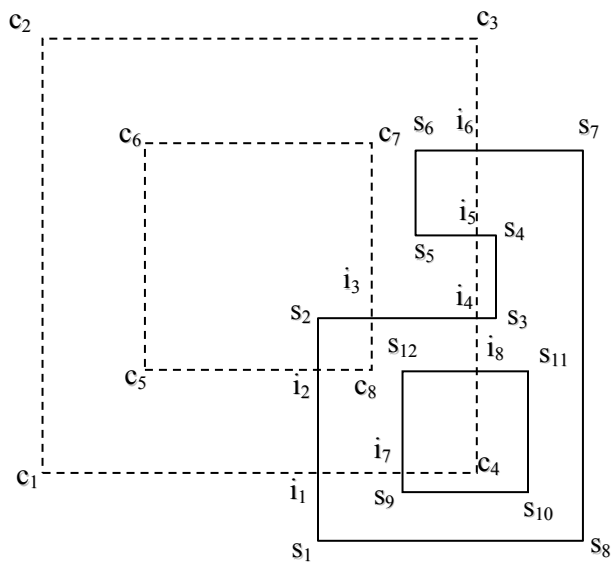
- Uma para as bordas que pertençam ao interior do polígono de recorte e a outra para as bordas que pertençam ao seu exterior.
- Bordas do polígono de recorte no interior do polígono alvo formam buracos. Neste caso, tais bordas pertencerão às duas listas.

Algoritmo de recorte:

1. Remova um vértice da lista de interseções de entrada. Se não houver nenhum o processo terminou.
2. Siga a lista de vértices do polígono alvo até que uma interseção seja encontrada.
3. Neste momento, copie os vértices percorridos até o momento para a lista de bordas internas.
4. Salte para a posição correspondente na lista de vértices do polígono de recorte.
5. Siga a lista de vértices do polígono de recorte até que uma interseção seja encontrada.
6. Neste momento, copie os vértices percorridos do polígono de recorte, até o momento, para a lista de bordas internas.
7. Salte para a posição correspondente na lista de vértices do polígono alvo.
8. Repita o processo até que o ponto inicial seja alcançado. Nesse momento, um polígono interno foi fechado.



PA	PR	
S1	C1	Borda externa
S2	C2	
S3	C3	
S4	C4	
S5	C1	
S6		
S7		
S8		
S1		
S9	C5	Borda interna
S10	C8	
S11	C7	
S12	C6	
S9	C5	



PA

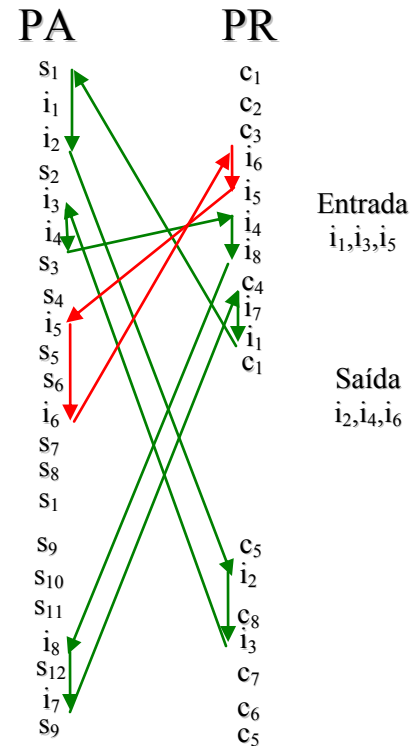
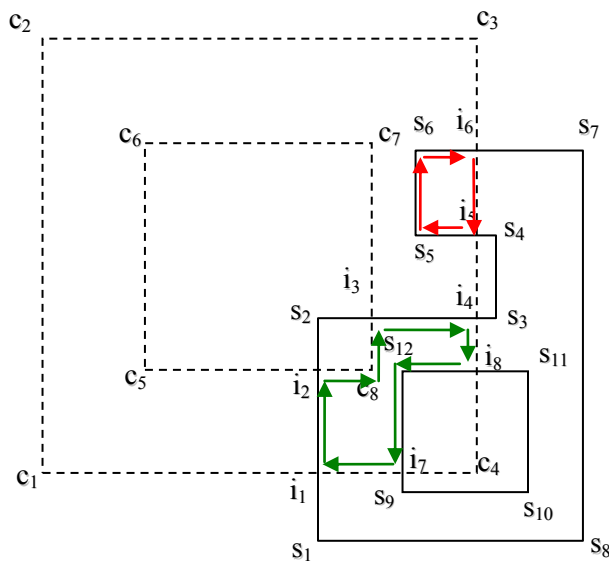
s<sub>1</sub>  
i<sub>1</sub>  
i<sub>2</sub>  
s<sub>2</sub>  
i<sub>3</sub>  
i<sub>4</sub>  
s<sub>3</sub>  
s<sub>4</sub>  
i<sub>5</sub>  
s<sub>5</sub>  
s<sub>6</sub>  
i<sub>6</sub>  
s<sub>7</sub>  
s<sub>8</sub>  
s<sub>1</sub>  
s<sub>9</sub>  
s<sub>10</sub>  
s<sub>11</sub>  
i<sub>8</sub>  
s<sub>12</sub>  
i<sub>7</sub>  
s<sub>9</sub>

PR

c<sub>1</sub>  
c<sub>2</sub>  
c<sub>3</sub>  
i<sub>6</sub>  
i<sub>5</sub>  
i<sub>4</sub>  
i<sub>8</sub>  
c<sub>4</sub>  
i<sub>7</sub>  
i<sub>1</sub>  
c<sub>1</sub>  
c<sub>5</sub>  
i<sub>2</sub>  
c<sub>8</sub>  
i<sub>3</sub>  
c<sub>7</sub>  
c<sub>6</sub>  
c<sub>5</sub>

Entrada  
i<sub>1</sub>, i<sub>3</sub>, i<sub>5</sub>

Saída  
i<sub>2</sub>, i<sub>4</sub>, i<sub>6</sub>



- 5) Descreva em detalhes a componente difusa do modelo de iluminação Phong, justificando porque não é necessário levar em consideração para este a posição do observador (0.5 ponto).

A iluminação difusa segue a lei de Lambert, que afirma que a intensidade da radiação refletida por unidade de área independe da direção de reflexão. Assim a iluminação difusa chega na superfície em uma direção determinada, sendo refletida em todas as direções de igual maneira. O resultado implica que a iluminação da superfície é o mesmo independente do posicionamento do observador.

A equação phong para varias fontes de luz é a seguinte:

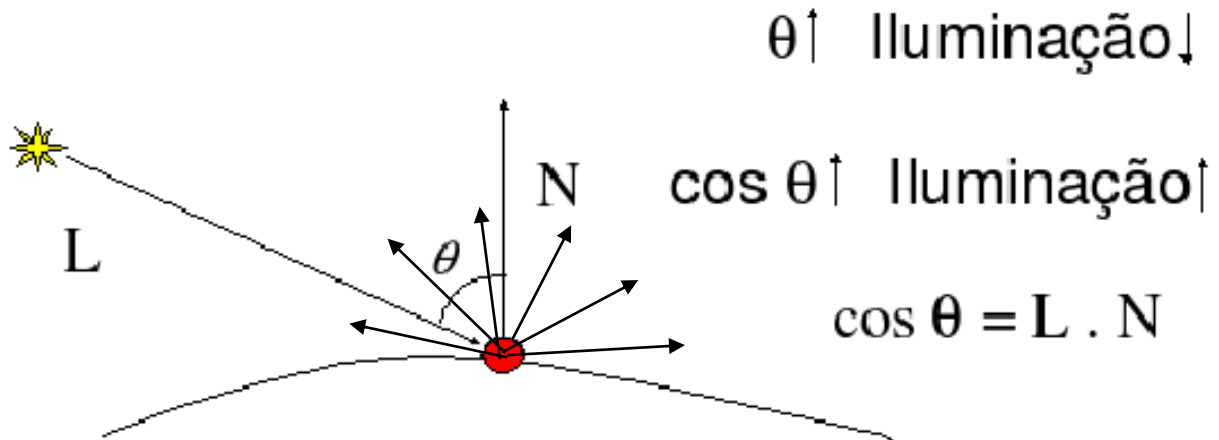
$$I_{total} = I_{ambiente} + \sum_{i=1}^n (I_{difusa} + I_{especular})$$

Onde a componente difusa é dada por

$$I_{difusa} = I_i K_d (N \cdot L_i)$$



Sendo  $I_i$  a intensidade de cada uma das luzes.



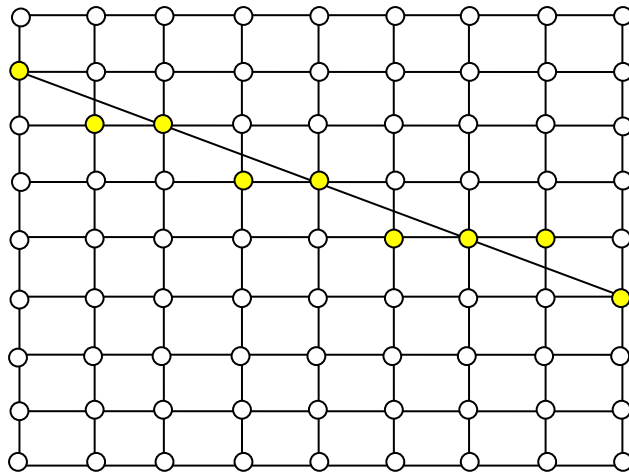
- 6) Para que a iluminação ambiente fosse correta, o que deveria ser feito? Porque isto é computacionalmente caro? (0.5 ponto)

Para que esta iluminação fosse correta deveria-se analisar o que ocorre com cada um dos raios de luz lançados ao ambiente e por ele devolvidos novamente à cena. Estes raios também podem ser denominados de energia da fonte de luz. Este processo é computacionalmente caro devido ao fato de que há infinitos raios de luz e para um cálculo razoável deveria-se fazer uma amostragem muito grande.

- 7) Descreva as diferenças entre o modelo de iluminação per pixel e per vertex. (1.0 ponto).

Na iluminação per vertex a função de cálculo da iluminação é aplicada sobre os vértices, interpolando os valores na parte interior do polígono. A iluminação per pixel calcula a iluminação para cada ponto da superfície, calculando a normal para cada um dos pontos. Enquanto na iluminação per vertex apenas são feitos 3 cálculos de iluminação para cada polígono, na iluminação per pixel o cálculo é feito dezenas, centenas ou milhares de vezes, dependendo do número de pixels que um polígono ocupa na tela. É por esta razão que em tempo real se costuma usar o cálculo per vertex.

- 8) Considerando o algoritmo do ponto-médio para rasterização de retas, determine quais *pixels* devem ser acessos no reticulado abaixo. Considere os *pixels* como sendo os elementos na interseção das retas do reticulado conforme ilustrado na figura abaixo (0.5 ponto).



9) O que ocorre se o rendering for efetuado diretamente no front-buffer? (0.5 ponto).

O Front Buffer é uma área da memória de vídeo que contém a imagem que está sendo mostrada no monitor. Se o rendering ocorrer diretamente nesta área, será possível ver a imagem sendo formada na tela. Como neste processo são plotados polígonos que depois serão sobrepostos por outros, devido ao Z-Buffer, será possível observar um efeito de ruído muito forte na tela. (polígonos que aparecem e desaparecem, para a mesma imagem)

10) Descreva os processos de mapeamento direto e inverso. Quais as vantagens do mapeamento inverso em relação ao mapeamento direto (1.0 ponto).

No mapeamento direto, os valores dos pixels em uma imagem destino ID são calculados a partir dos valores da imagem de origem ID através da transformação  $T: ID(i,j) \rightarrow IS(i,j)$ . No mapeamento inverso, percorre-se os pixels da imagem destino, buscando os valores na imagem origem através da transformação inversa  $T^{-1}: ID(i,j) \rightarrow IS(i,j)$ .

A vantagem do mapeamento inverso é de que não surgem buracos na imagem destino transformações que envolvam um fator de ampliação. Além disso, quando houver um fator de redução, é possível utilizar esquemas de filtragem considerando-se uma vizinhança em torno do pixel correspondente na imagem origem, efetuando-se deste modo uma amostragem mais apropriada.

11) Explique como é a recursão referente ao reflexo para o algoritmo de Ray-tracing (0.5 ponto)

1. Trace um “raio” a partir do observador até a cena a ser representada através de um pixel da tela;
2. Determine qual é o primeiro objeto a interceptar esse raio;

3. Calcule a cor da superfície do objeto no ponto de interseção baseado nas características do objeto e na luz;
4. Se a superfície do objeto for reflexiva, calcule um novo raio a partir do ponto de interseção e na “direção de reflexão”;
5. Multiplique o resultado da cor obtida no reflexo pelo coeficiente de reflexão do objeto e some-o com a cor obtida pela equação Phong, previamente.

12) Porque na recursão do Ray-tracing, caso as superfícies não sejam 100% reflexivas, chega-se a um momento, depois de muitas recursões, em que praticamente não há mais contribuição da cor? (0.5 ponto)

Para cada recursão, o valor do reflexo é multiplicado pelo coeficiente de reflexo da superfície. Não ser 100% reflexiva, significa que este coeficiente está entre 0 e 1. Ao multiplicar vários coeficientes com este valor, o resultado tenderá a zero. Num determinado momento, mesmo que não seja zero absoluto, este coeficiente acumulado será tão pequeno, que a diferença será menor do que a capacidade representativa de cores RGB, o que não provocará mais resultado visível na imagem.

13) O que são *key-frames*. Explique em que consiste o processo de *in-betweening* (0.5 ponto).

Key-frames são os quadros que especificam o início e fim de uma transição suave em uma animação. O processo de in-betweening consiste na geração dos quadros intermediários, entre dois key-frames, necessários para a criação da ilusão de movimento.

14) O que é quantização e quais são suas aplicações. Descreva de forma sucinta o Algoritmo da Populosidade (0.5 ponto).

Quantização é uma transformação que visa discretizar o gamute de cores de uma imagem, além de reduzir o número de bits necessário para armazenar a informação de cor. As principais aplicações estão na compressão e transmissão de imagens.

Algoritmo da Populosidade:

1. Calcule o histograma da imagem.
  2. Escolha os k níveis de quantização como sendo as k cores mais frequentes (mais populosas).
  3. A função de quantização  $q(c)$  atribui a cada cor  $c$  o nível de quantização mais próximo de  $c$  segundo o quadrado da métrica euclidiana.
- Obs.: Em caso de empate fazer uma escolha aleatória ou considerar a vizinhança do pixel.