Fundação CECIERJ - **Vice Presidência de Educação Superior a Distância**

Curso Superior de Tecnologia em Sistemas de Computação
Disciplina: Computação Gráfica
AD1 1° semestre de 2016.

1) Faça uma pesquisa sobre algum dispositivo utilizado em Realidade Virtual (1.0 ponto).

Head-mounted displays (dispositivos de exibição montados sobre a cabeça) são equipamentos muito utilizados para exibir imagens de um ambiente virtual. Muita das vezes se parecem com um capacete com pequenas telas com um aspecto similar a um óculos. O capacetes oferecem mobilidade permitindo o usuário se mover. A ilusão de profundidade é criada através da projeção de duas imagens de diferentes pontos de vista. Além disso, conforme o usuário se move, todo o ambiente virtual é transformado consoantemente. As telas de um *head-mounted display* podem ser opacas ou transparentes. Na primeira modalidade todo os objetos devem ser gerados virtualmente, incluindo o próprio usuário. No segundo modo, é possível combinar objetos do mundo real com objetos virtuais criando a ideia de uma realidade combinada.

Referência: M. Mihelj et al., Virtual Reality Technology and Applications, 97. Intelligent Systems, Control and Automation: Science and Engineering 68, DOI: 10.1007/978-94-007-6910-6_5, © Springer Science+Business Media Dordrecht 2014.

2) Imagens são exemplos de objetos gráficos planares bidimensionais representados por decomposição espacial. Considere o problema de detectar o movimento em uma sala onde uma câmera captura imagens continuamente. Forneça uma solução para o problema(1.0 pontos).

Uma possível solução consiste em capturar um conjunto de imagens da sala sem nenhum objeto dinâmico e construir um modelo de fundo. O modelo de fundo é uma matriz que armazena para cada pixel p, uma cor (r,g,b) para o fundo e um valor de tolerância (t_r,t_g,t_b). A cor do modelo de fundo pode ser obtida a partir de uma média das cores (μ_r,μ_g,μ_b) dos vários quadros obtidos em um intervalo de tempo *t*. Por sua vez, a tolerância, descrita por um intervalo de confiança, pode ser estimada através do desvio padrão das cores dos quadros $(\sigma_r, \sigma_g, \sigma_b)$. O intervalo de confiança é importante porque de um quadro para o outro de uma cena estática sempre existe ruído que faz com que duas imagens não sejam exatamente as mesma. Com base no modelo de fundo M, é possível verificar se um pixel *p* de um novo quadro corresponde a alguma mudança na cena verificando se $p_r \in [\mu_r-\sigma_r, \mu_r+\sigma_r]$, $p_g \in [\mu_g-\sigma_g, \mu_g+\sigma_g]$ e $p_r \in [\mu_b-\sigma_b, \mu_b+\sigma_b]$. Essa solução simples considera que os canais r, g e b sejam independentes. Além disso, não contempla variações de iluminação. Uma solução mais robusta deve utilizar um espaço de cor diferente que represente cores em termos de luminância e crominância. Assim, somente a "qualidade de cor" deve ser considerada e não a intensidade de seu brilho que pode variar conforme as horas do dia, condições climáticas e etc.

3) O que é uma representação CSG (Constructive Solid Geometry)? (1.0 ponto).

Uma representação CSG é um modelo para representação de sólidos baseado em três elementos básicos: primitivas geométricas, transformações do espaço e operações booleanas.

As primitivas são em geral formas muito simples de descrever no computador. No plano são regiões como: um quadrado unitário ou um disco unitário. Já no espaço citamos como exemplo uma esfera unitária e um cubo unitário.

As transformações são utilizadas tanto para posicionar as primitivas no espaço quanto deformá-las. Em geral utiliza-se para posicionamento translações e rotações (movimentos rígidos) para transformar o sistema de coordenadas local da primitiva no sistema de coordenadas global. As transformações de deformação são utilizadas para gerar diferentes formas a partir das básicas.

As operações booleanas valem-se de operadores de união, interseção e diferença para gerar novas formas, considerando os pontos que compõem cada primitiva como conjuntos.

Finalmente, um modelo CSG pode ser descrito de forma hierárquica, através de uma árvore onde os nós podem ser primitivas ou operações de transformação ou booleanas.

4) Considere dois sólidos A e B descritos por duas funções implícitas f(x,y,z) <=0 e g(x,y,z) <= 0. Explique como determinar os sólidos abaixo de modo implícito:

a) $A \cup B$ (1.0 ponto)
b) $A \cap B$ (1.0 ponto)

Um ponto pertencerá a união dos dois sólidos se satisfizer ou f(x,y,z) <=0 ou g(x,y,z) <= 0. Logo, o sólido $A \cup B$ pode ser dado por h(x,y,z) = min{f(x,y,z),g(x,yz)}. Por sua vez o sólido $A \cap B$ consiste dos pontos tais que f(x,y,z) <=0 e g(x,y,z) <= 0. Portanto, pode ser descrito por h(x,y,z) = max{f(x,y,z),g(x,y,z)}.

5) Determine a expressão para interpolação trilinear. Use o mesmo processo utilizado para derivar a expressão correspondente à interpolação bilinear(1.0 ponto).

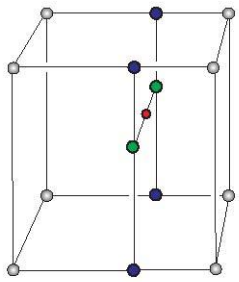
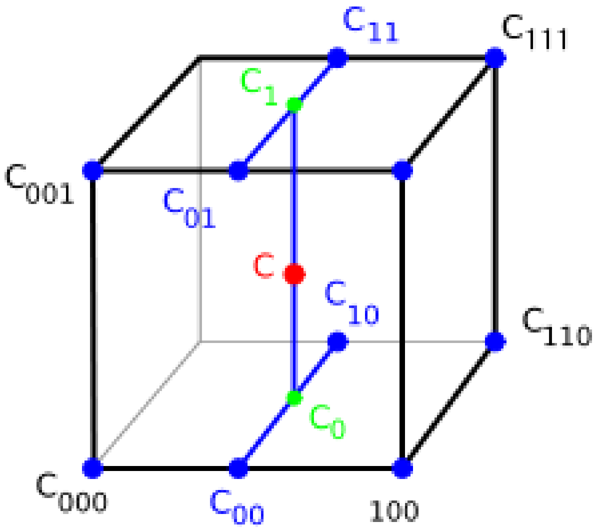


Figura 11. Reconstrução por interpolação trilinear.

Fonte: http://www.visgraf.impa.br/Publications/fundamentos/cap16/fig11.jpg

Considere a figura abaixo extraída de https://upload.wikimedia.org/wikipedia/commons/thumb/9/97/3D_interpolation2.svg/230px-3D_interpolation2.svg.png



Ainda, considere que o ponto central p de cor C tenha coordenadas (x,y,z) e que os pontos de cor C_{000} e C_{111} tenham, respectivamente, coordenadas (x_0,y_0,z_0) e (x_1,y_1,z_1). Defina as distâncias u,v e w, entre p e cada uma dos planos do cubo como:

$$u=(x-x_0)/(x_1-x_0)$$
$$v=(y-y_0)/(y_1-y_0)$$
$$w=(z-z_0)/(z_1-z_0)$$

Deste modo, podemos calcular os valores C_{00}, C_{01}, C_{10} e C_{11} através de interpolação linear em *u*.

$$C_{00}=C_{000}(1-u)+C_{100}u$$
$$C_{01}=C_{001}(1-u)+C_{101}u$$
$$C_{10}=C_{010}(1-u)+C_{110}u$$
$$C_{10}=C_{011}(1-u)+C_{111}u$$

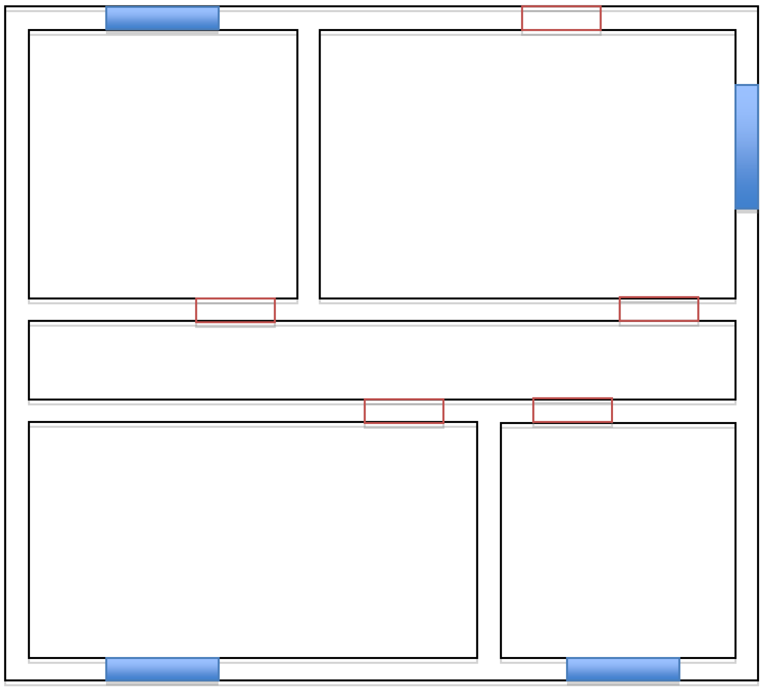
Em seguida, podemos obter a cor C_0 no ponto correspondente via interpolação linear em *v* entre C_{00} e C_{10} e a cor C_1 de modo análogo.

$$C_0=C_{00}(1-v)+C_{10}v$$
$$C_1=C_{01}(1-v)+C_{11}v$$

Finalmente a cor em C é dada pela interpolação linear entre C_0 e C_1 usando a distância w.

$$C=C_0(1-w)+C_1w$$

6) Considere a planta baixa de uma construção determinada por um conjunto de curvas poligonais fechadas representando paredes, portas e janelas.



a) Desenvolva um algoritmo que recebe como entrada um conjunto de linhas poligonais fechadas e erga a planta baixa, gerando as paredes e tetos dos cômodos. Faça os espaços vazios correspondentes as paredes e portas (2.0 pontos).

Uma ideia para construir um modelo 3d da construção a partir da planta baixa é usar uma modelagem inspirada pelo modelo CSG, conforme visto na questão 3. Usaremos como primitivas sólidos gerados a partir das curvas poligonais.

Para uma dada curva poligonal **c**=$\{v_0,v_1,...,v_n\}$ geramos um sólido **s** conforme abaixo:

Para cada par de vértices consecutivos ve_i e ve_{i+1} criamos uma face $\mathbf{f_i} = \{fv_i,fv_{i+1},fv_{2i},fv_{2(i+2)}\}$, com quatro vértices consecutivos, com coordenadas x, y e z definidas da seguinte forma:

$$fv_i = (ve_i.x, ve_i.y, 0)$$
$$fv_{i+1} = (ve_{i+1}.x, ve_{i+1}.y, 0)$$
$$fv_{2i+2} = (ve_{i+1}.x, ve_{i+1}.y, h)$$
$$fv_{2i} = (ve_i.x, ve_i.y, h)$$

Construímos mais duas faces **fc** e **ft**. A face f**c** representa o chão e é determinada por todos os vértices das faces $\mathbf{f_i}$ com coordenadas z =0, seguindo o sentido contrário da ordem dos vértices em **c**. Já a face **ft** representa o teto, e contem os vértices das faces **fi** que tem coordenada z=h, vértices estes que devem ser ordenados no sentido original dos vértices **c**. Deste modo, descrevemos um modelo B-Rep de um sólido **s**, atentando para o fato de que as normais de todas as faces apontam para fora do modelo.

Aplicamos o método acima para gerar o sólido **se** associado a curva externa **ce** que descreve o limite da área de construção. Em seguida, aplicamos o mesmo método para cada uma das curvas internas $\mathbf{ci_i}$ (que delimitam o espaço dos cômodos), gerando modelos si_0, si_1,...,si_n.

Construímos também sólidos, na forma de paralelepípedos, a partir das curvas correspondentes às paredes e às portas. Chamemos, respectivamente os sólidos das portas de spo_i e sj_i.

Definimos o sólido final **Sf** em dois passos. Primeiro, aplicamos o operador booleano de diferença entre o sólido externo se e os sólidos internos si_i, gerando o modelo intermediário da construção com as paredes M = se - $\bigcup_i si_i$. Aplicamos a operação de diferença entre o modelo M e a união dos sólidos das portas e paredes Mf = M- ($\bigcup_i spo_i \cup \bigcup_i sj_i$).

Para utilizar o modelo CSG precisamos definir as operações de união, interseção e diferença entre as primitivas sólidas. A definição desses operadores no caso geral requer cálculo de interseção entre polígonos que descrevem as faces e regiões volumétricas. Uma formulação mais robusta pode ser obtida usando operadores de Euler (http://repository.cmu.edu/cgi/viewcontent.cgi?article=2586&context=compsci) que podem ser vistos em livros de modelagem de sólidos.

No caso do exercício acima, como os sólidos correspondentes ao espaço vazio dos cômodos estão completamente contidos no interior do sólido maior, calcular a diferença implica em simplesmente adicionar as faces de cada sólido interior com inversão de sua orientação ao conjunto final. No caso da diferença entre os sólido M e as portas e janelas, é necessário um pouco mais de cuidado, pois faces do modelo M vão passar a ter buracos determinados pelas faces geradas pelas arestas das curvas poligonais, correspondentes às portas e às janelas que tem interseção com alguma aresta curva poligonal **c**, que descreve a área total da construção. Neste caso, é necessário uma representação especial que permita descrever polígonos com furos. Cada polígono passa a ter então uma lista de polígonos internos que descrevem furos, podendo essa lista ser vazia.

Uma outra possibilidade é representar os sólidos de forma implícita usando decomposição espacial. Isto produz um modelo voxelizado do sólido onde os voxels internos tem valor 1 e os externos 0. Deste modo calcular uniões, interseções e diferenças pode ser feito de forma simples, conforme a questão 4.

b) Caracterize o objeto gráfico que você construiu com seu algoritmo (1.0 ponto)

O objeto por construção é um sólido (espacial e tridimensional)

7) Sejam dadas as coordenadas de 4 pontos p0,p1,p2 e p3. Mostre como derivar a expressão do polinômio cúbico de Bézier p(t) sabendo que p(0)=p0, p(1)=p3 e p'(0) = 3(p1-p0) e p'(1) = 3(p3-p2) (1.0 ponto).

ANULADA POR ERRO NO ENUNCIADO: p(1) = p3

Um polinômio cúbico é dado por:

$$p(x) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \qquad (1)$$

e sua primeira derivada por

$$p'(x) = a_1 + 2a_2 t + 3a_3 t^2 \qquad (2)$$

Impondo as restrições fornecidas acima temos o seguinte sistema:

$$\begin{aligned}
p(0) &= a_0 + a_1.0 + a_2.0 + a_3.0 = p_0 \\
P(1) &= a_0 + a_1.1 + a_2.1 + a_3.1 = p3 \\
p'(0) &= a_1 + 2a_2.0 + 3a_3.0 = 3(p_1 - p_0) \\
p'(1) &= a_1 + 2a_2.1 + 3a_{3.}.1^2 = 3(p_3 - p_2)
\end{aligned} \qquad (3)$$

Diretamente, podemos afirmar que:

$$\begin{aligned}
a_0 &= p_0 \\
a_1 &= 3(p_1 - p_0)
\end{aligned} \qquad (4)$$

Substituindo (3) na segunda e quarta equações de (2) e resolvendo o sistema temos:

$$\begin{aligned}
&a_0 + a_1 + a_2 + a_3 = p_3 \\
&a_1 + 2a_2 + 3a_3 = 3(p_3 - p_2)(5), \\
&p_0 + 3(p_1 - p_0) + a_2 + a_3 = p_3 \\
&3(p_1 - p_0) + 2a_2 + 3a_3 = 3(p_3 - p_2)(6), \\
&a_2 + a_3 = (p_3 - p_0) - 3(p_1 - p_0) \\
&2a_2 + 3a_3 = 3(p_3 - p_2) - 3(p_1 - p_0)(7), \\
&a_3 = (p_3 - p_0) - 3(p_2 - p_1) \\
&a_2 = 3(p_2 - p_1) - 3(p_1 - p_0)(8)
\end{aligned}$$

Substituindo os coeficientes em (1) e manipulando a expressão chegamos ao polinômio de Bézier cúbico:

$$p(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + p_3 t^3$$