Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Computação Gráfica
AD1 - 2° semestre de 2019.

1) Faça uma pesquisa sobre o uso de Computação Gráfica na indústria cinematográfica. (1.0 ponto)

A Computação Gráfica (CG) esteve presente desde muito cedo na indústria cinematográfica. Assim que a área se tornou mais amadurecida, e os dispositivos capazes de realizar efeitos visuais mais sofisticados, os métodos de CG começaram a ser utilizados na produção de efeitos especiais e, posteriormente, de películas completas.

Em muitos casos, o custo da produção é muito reduzido, ao se trocar efeitos criados através de processos físicos (por exemplo, explosões) por efeitos gerados por computador. Em outros casos, os efeitos são inviáveis sem uso de animação digital.

O filme TRON, da Walt Disney Productions, foi um dos primeiros a utilizar efeitos de Computação Gráfica. Vale ressaltar que Ken Perlin, um renomado pesquisador da área de Computação Gráfica, recebeu o prêmio da *Academy Award for Technical Achievement*, por invertar o *Perlin Noise* para o filme TRON.

A primeira produção feita completamente em CG foi o filme "Toy Story" da Pixar, e atualmente, cada vez mais a animação por computador tem sido utilizada na Indústria cinematográfica. Dentre as produtoras mais famosas podemos citar a Disney e a Pixar.

Com relação ao processo de animação por computador, destaca-se o surgimento da técnica de *Motion Capture*, um processo de captura, registro e transferência de movimentos, que facilitou enormemente a animação por computador de personagens articulados, com características humanoides ou de animais.

Outra tendência é o uso de combinação de cenas geradas por computador com cenas e atores reais como, por exemplo, no filme Avatar da 20th Century Fox.

Referências:

https://en.wikipedia.org/wiki/Tron, acessado em 03/09/2019.
https://en.wikipedia.org/wiki/Toy_Story_(franchise), acessado em 03/09/2019.
https://en.wikipedia.org/wiki/Pixar, acessado em 03/09/2019.
https://en.wikipedia.org/wiki/The_Walt_Disney_Company, acessado em 03/09/2019.
https://en.wikipedia.org/wiki/Avatar_(2009_film) , acessado em 03/09/2019.
https://en.wikipedia.org/wiki/Motion_capture, acessado em 03/09/2019.
https://en.wikipedia.org/wiki/Perlin_noise, acessado em 03/09/2019.

2) Faça uma pesquisa sobre o impacto do avanço da tecnologia das GPUs na computação. (1.0 ponto)

As GPUs foram desenvolvidas inicialmente para aplicações gráficas, entretanto, programadores e pesquisadores logo encontraram formas de utilizar as GPUs para programação de propósito geral chamada GPGPUs (*General Purpose Computing on Graphics Processing Units*). Neste interim, a programação de aplicações genéricas em GPUs não era natural, visto que era necessário mapear os modelos para objetos gráficos manipuláveis pelo pipeline da GPU.

Com o tempo, surgiram APIs que permitem programar GPUs usando extensões da linguagem C/C++ como CUDA e OpenCL. As GPUs são dispositivos de hardware especializados, voltados para a solução de problemas nos quais uma mesma operação (uma *thread*) pode ser aplicada simultaneamente sobre múltiplos dados.

Atualmente, não só a Computação Gráfica usufrui do poder das GPUs. Áreas como Computação de Alto Desempenho, Modelagem e Simulação numérica, Otimização Combinatória, Inteligência Artificial, Ciência de Dados e muitas outras requerem o uso de GPUs de alto processamento para resolver seus problemas.

3) Cite aplicações de transformações geométricas em Computação Gráfica. (1.0 ponto)

Anulada.

4) Defina as operações de uma transformação geométrica que compute a rotação de θ graus de um ponto $(x,y) \in R^2$ em torno de um ponto (x_0,y_0) no plano. Dica: atente para o fato que as matrizes de rotação efetuam rotações em torno da origem (0,0). (1.0 ponto)

Anulada.

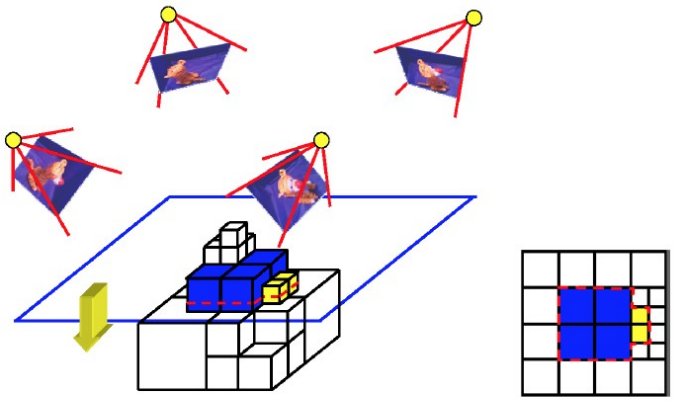
5) Explique a diferença entre transformações lineares e transformações afins. (1.0 ponto)

Anulada.

6) Descreva uma aplicação das Octrees em Computação Gráfica. (1.0 ponto)

As octrees são estrturas de dados do tipo árvore, usadas para decomposição adaptativa do espaço. Um dos objetivos de toda estrutura hierárquica, como são as Octrees e Quadtrees, é lidar adequadamente com a complexidade do problema.

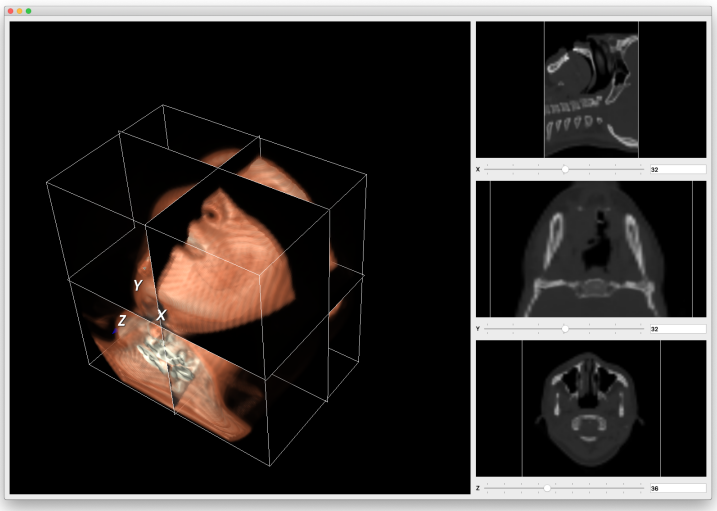
Octrees são utilizadas em muitas aplicações na Computação Gráfica como, por exemplo, na reconstrução de objetos a partir de imagens, tomadas de diferentes pontos de vista, onde são conhecidos os parâmetros da câmera. O produto final da reconstrução, neste caso, é um modelo descrito por um volume adaptativo.



(a) Adaptive Space Carving (b) Selection

Referência: *Montenegro, A.A., Carvalho, P.C., Gattass, M., & Velho, L. (2004). Adaptive space carving. Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004., 199-206.*

7) Considere um dado volumétrico representado por uma matriz de valores escalares V_{ijk} , para $0 \leq i < dimx$, $0 \leq j < dimy$, $0 \leq k < dimz$, onde dimx, dimy e dimz são as dimensões da matriz. O fatiamento ou *slicing* é o processo que extrai uma imagem de uma fatia do volume. Descreva como produzir uma imagem dada por uma matriz I correspondente a uma fatia no volume. Considere por simplificação o caso de images I_{ij} de dimensão dimx × dimy, associadas a fatias ao longo do eixo z (índice k = constante). A figura abaixo mostra um programa escrito em Python e Vtk que ilustra o processo de *slicing* ao longo dos eixos x (imagem superior à direita), y (imagem do meio à direita e z (imagem inferior à direita). Assuma que o volume V e as imagens I são matrizes de números reais.(1.0 ponto)



Dado volumétrico obtido de https://vtk.org/vtk-textbook-examples-and-data/

O processo de *slicing* por planos ortogonais é relativamente simples. Basta determinar a fatia do volume intersectada pelo plano desejado e amostrar os valores dos voxels intersectados pelo plano. Para cada pixel (i,j) de I, obteremos o valor do voxel correspondente V_{ijk}, onde k é o índice da fatia intersectada pelo plano desejado.

Em outras palavras, o *slicing* é uma subamostragem do volume dado por uma imagem representando os voxels em uma dada fatia. Slicing em direções não ortogonais, isto é, em eixos distintos dos eixos x,y e z requerem um processo de reamostragem mais sofisticado, já que não necessariamente existe um mapeamento 1-a-1 dos voxels nos pixels da imagem destino. Entretanto, o procedimento em geral não é muito diferente.

8) Escreva um algoritmo que construa o campo escalar associado a um dado volumétrico V_{ijk}, onde $0 \le i < dimx$, $0 \le j < dimy$, $0 \le k < dimz$, (dimx, dimy e dimz são as dimensões da matriz). O campo escalar é definido por uma função implícita dada abaixo, onde p_m é um ponto com coordenadas $(p_m.x, p_m.y, p_m.z)$ pertencente a um conjunto de pontos $P = \{p_0, p_1, ..., p_{n-1}\}$, r_m^2 é o quadrado da distância de um voxel na posição (i,j,k) a um ponto p_m, e o termo exponencial define uma função Gaussiana de altura b_m e desvio padrão a_m. (1.0 ponto).

$$V_{ijk} = \sum_{m=0}^{n-1} b_m e^{-a_m r_m^2},$$

$$r_m^2 = (i - p_m.x)^2 + (j - p_m.y)^2 + (k - p_m.z)^2$$

```
Algoritmo CampoEscalar
Entrada: dimx,dimy,dimz (dimensões)
         Lista com n primitivas pontuais (p0,p1,p2,...pn-1)
         Lista de parâmetros (b0,b1,...,bm) e (a0,a1,...,an-1) por primitiva.
Saida: Volume Vijk

Para i=0 até dimx faça
  Para j=0 até dimy faça
    Para k = 0 até dim z faça
      Vijk ← 0.0;
    Fim_para
  Fim_para
Fim_para

Para i=0 até dimx faça
  Para j=0 até dimy faça
    Para k = 0 até dim z faça
       Soma ← 0.0;
       Para m = 0 até n-1 faça
          r2 ← (i-pm.x)*(i-pm.y)+(j-pm.y)*(j-pm.y)+(k-pm.z)*(k-pm.z);
          sum ← sum + bm*exp(-am*r2);
       Fim_para
       Vijk ← sum;
    Fim_para
  Fim_para
Fim_para
Retorne Vijk
```

Referência: James F. Blinn. 1982. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.* 1, 3 (July 1982), 235-256. DOI=http://dx.doi.org/10.1145/357306.357310James F. Blinn. 1982. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.* 1, 3 (July 1982), 235-256. DOI=http://dx.doi.org/10.1145/357306.357310

9) Mostre que não é possível definir um círculo usando uma curva de Bézier quadrática (curva de segundo grau). Dica: considere a expressão dada pela fórmula de Bézier de segundo grau no parâmetro t e substitua na equação do círculo $x(t)^2 + y(t)^2 = R^2$; em seguida, analise o que ocorre com os pontos (x(t), y(t)) definidos em função de t. (1.0 ponto)

Primeiramente, define-se uma curva de Bézier quadrática através da expressão:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^{3} \begin{pmatrix} 2 \\ i \end{pmatrix} (1-t)^{2-i} t^i \begin{pmatrix} x_i \\ y_i \end{pmatrix}, 0 \le t \le 1$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = (1-t)^2 \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + 2t(1-t) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + t^2 \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}, 0 \le t \le 1$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_1 - 2(x_2 - x_1)t + (x_3 - 2x_2 + x_1)t^2 \\ y_1 - 2(y_2 - x_1)t + (y_3 - 2y_2 + y_1)t^2 \end{pmatrix}$$

Pode-se observar que a segunda equação é uma equação de segundo grau nos coeficientes *a*,*b*,*c*,*d*,*e* e, *f*.

$$\begin{cases} x = at^2 + bt + c \\ y = dt^2 + et + f \end{cases}$$

Substituindo a definição de x e de y na equação do círculo tem-se

$$(x - x_0)^2 + (y - y_0)^2 = R^2$$
$$(at^2 + bt + (c - x_0))^2 + (dt^2 + et + (f - y_0))^2 = R^2, \forall t, 0 \le t \le 1$$
$$a^2t^4 + 2abt^3 + 2a(c - x_0)t^2 + b^2t^2 + 2b(c - x_0)t + (c - x_0)^2 + d^2t^4 +$$
$$2det^3 + 2d(f - y_0)t^2 + e^2t^2 + 2e(f - y_0)t + (f - y_0)^2 - R^2 = 0$$
$$(a^2 + d^2)t^4 + 2(ab + de)t^3 + (b^2 + e^2 + 2a(c - x_0) + 2d(f - y_0))t^2$$
$$+2(b(c - x_0) + e(f - y_0))t + ((c - x_0)^2 + (f - y_0)^2 - R^2) = 0$$

Como os termos nos coeficientes t^4,t^3,t^2, e t devem se anular temos então as seguintes equações.

$$a^2 + d^2 = 0 \ (1)$$

$$ab + de = 0 \ (2)$$

$$b^2 + e^2 + 2a(c - x_0) + 2d(f - y_0) = 0 \ (3)$$

$$b(c - x_0) + e(f - y_0) = 0 \ (4)$$

$$(c - x_0)^2 + (f - y_0)^2 - R^2 = 0 \ (5)$$

Por (1) tem-se que a = 0 e d = 0. Substituindo os valores de a e d em (3) tem-se que b = 0 e e = 0.

Finalmente, substituindo os valores de *a*,*b*,*d* e *e* na equação

$$(at^2 + bt + (c - x_0))^2 + (dt^2 + et + (f - y_0))^2 = R^2$$

chega-se ao resultado

$$(c - x_0)^2 + (f - y_0)^2 = R^2$$

satisfeita por apenas um ponto (x_1,y_1), o que é uma contradição.

Baseado na referência https://math.stackexchange.com/questions/1941230/why-it-is-impossible-to-draw-a-circle-by-bezier-curve acessado em 19-08-2019 às 13:15 AM.

10) Explique a diferença entre uma B-Spline e uma Nurbs (*Non-uniform Rational B-Splines*). (1.0 ponto)

Uma B-Spline é uma curva suave formada por uma série de *m-2* segmentos de curva $Q_3, Q_4, \ldots, Q_m$ definidos por *m+1* pontos de controle $P_0, P_1, \ldots, P_m$, $m \geq 3$. No caso de B-Splines cúbicas, cada segmento de curva é definido por quatro pontos de controle e 4 funções de mistura. Cada ponto de controle influencia somente 4 segmentos de curva.

A formulação de um segmento $Q_i(u)$ *B-spline cúbica* é dada pela seguinte expressão:

$$Q_i(u) = UB_s P = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}$$

$$Q_i(u) = \sum_{k=0}^{3} P_{i-3+k} B_{i-3+k}(u)$$

onde *i* é o número do segmento e *k* é o índice do ponto de controle correspondente ao segmento *i*. O valor de *u* para um único segmento está contido no intervalo $0 \leq u \leq 1$. Isto é, *u* é um parâmetro local que varia entre 0 e 1 para definir um único segmento.

Os pontos em que dois segmentos de uma B-Spline se acoplam é denominado nó e o valor de u correspondente é o valor do nó.

Um único polinômio pode ser utilizado para expresser a combinação dos segmentos de uma B-Spline, conforme abaixo:

$$Q(u) = \sum_{k=0}^{m} P_i B_i(u)$$

onde i e u são parâmetros globais e não locais a cada segmento.

Diferentemente das B-Splines, as *NURBS* são *B-Splines* não uniformes dadas pela razão entre dois polinômios.

$$\frac{\sum_{i=0}^{n} P_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)}$$

Os valores w_i associados a cada ponto de controle são pesos, que podem ser vistos como parâmetros extras. Os w_i afetam a curva apenas localmente. A curva é atraída para um ponto P_i se o w_i correspondente aumenta e é afastada de P_i se w_i diminui. Os w_i podem ser compreendidos como parâmetros de acomplamento da curva aos pontos de controle.

Essencialmente, Nurbs são B-Splines definidas em coordenadas homogêneas projetadas em um plano, através da pela divisão pelo polinômio no denominador.