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

Curso de Tecnologia em Sistemas de Computação

Disciplina: Computação Gráfica

AD1 - 1° semestre de 2010 - Gabarito.

1) "As curvas paramétricas podem ser usadas para descrever a trajetória de um objeto numa animação". Explique mais detalhadamente esta afirmação (1.0 ponto).

Uma curva paramétrica pode ser compreendida como um modelo matemático que descreve a trajetória percorrida por um ponto em movimento ao longo do tempo. Deste modo, é possível caracterizar o movimento de um objeto ou parte dele (ou de uma junta), em uma animação, através de uma curva paramétrica que descreva a sua trajetória no tempo.

2) Como um cilindro pode ser descrito de forma paramétrica? (1.0 ponto)

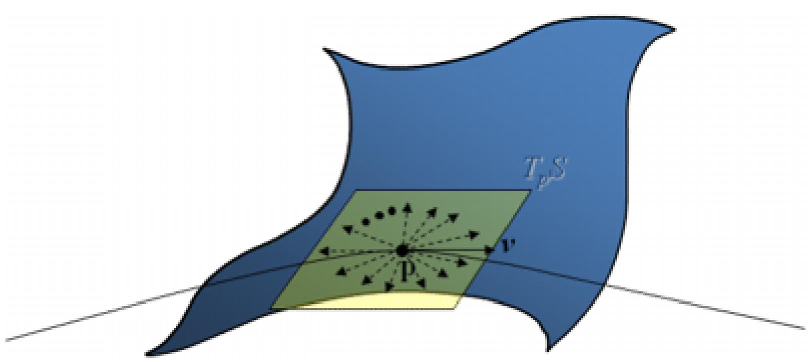
Um cilindro pode ser descrito de forma paramétrica através da função

$$f : R^2 \rightarrow R^3, f(u,v) = (\cos u, \sin u, v)$$

que corresponde a um círculo que se desloca ao longo do eixo parametrizado por v.

3) O que é um plano tangente a uma superfície? (1.0 ponto).

Um plano tangente em um ponto p de uma superfície S é um plano T_pS que contém os vetores tangentes às curvas contidas em S que passam por $p \in S$.



4) Quais são as vantagens de termos faces triangulares? (1.0 ponto).

   Em geral, as faces utilizadas em computação gráfica são triângulos devido às seguintes propriedades:

   - Os vértices de um triângulo sempre definem um plano, ao contrário de outros polígonos. A decomposição intrínseca de uma superfície em elementos triangulares garante uma representação linear por partes.

   - É possível definir um sistema de coordenadas a partir de qualquer triângulo através de coordenadas baricêntricas.

   - A normal de uma face triangular é bem definida, sendo determinada pelo produto vetorial de dois vetores sobre dois lados adjacentes a um vértice.

   - As propriedades de um triângulo são válidas para elementos que são generalizações de triângulos em dimensões arbitárias: os simplexos. Um simplexo de dimensão *n* é um politopo (objeto geométrico com faces planares) com *n+1* vértices, que corresponde ao fecho convexo (conjunto convexo minimal) de tal conjunto de vértices.

5) O que são os *patches* ou retalhos? (0.5 ponto).

   Um retalho é um pedaço de uma superfície *S* definido localmente através de uma parametrização $f:R^2 \to R^3$. Retalhos combinados de forma apropriada são capazes de expressar superfícies mais complexas. É importante observar a forma como os retalhos são colados para gerar a superfície global.

6) O que são *voxels*? (0.5 ponto).

   Voxels são as células do reticulado correspondente a representação matricial de um objeto volumétrico. Cada *voxel* define um paralelepípedo no espaço e traz diversas informações sobre os atributos do objeto na região como, por exemplo, a sua densidade. Os *voxels* desempenham, para a representação matricial de um objeto volumétrico, o mesmo papel que os *pixels* para representação matricial de uma imagem bidimensional.

7) Quais são as vantagens de uma representação matricial de objetos gráficos? (1.0 ponto)

   As vantagens são:

   - Indexação direta de cada um dos elementos que fazem parte da representação do espaço.

   - Possibilidade do uso de operações sobre matrizes como, por exemplo, operações de filtragem.

8) O que é uma curva Bézier? Cite algumas propriedades. (1.0 ponto).

Uma curva de Bézier é uma curva interativa descrita pela combinação (mistura) das coordenadas de pontos de controle através das funções de Bernstein. A forma geral de uma curva de Bézier de grau *n* é dada por

$$f(u) = \sum_{i=0}^{n} B_{i,n}(u) P_i(u), B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}$$

O caso mais comum de curvas de Bérzier são as curvas cúbicas onde *n*=3, para as quais temos as seguintes expressões para a base de Bernstein:

$$B_{i,3}(u) = \binom{3}{i} u^i (1-u)^{3-i}$$

$$B_{0,3} = (1-u)^3$$
$$B_{1,3} = 3u(1-u)^2$$
$$B_{2,3} = 3u^2(1-u)$$
$$B_{3,3} = u^3$$

Dentre as propriedadas das curvas de Bézier citamos as seguintes:

- A curva se restringe ao fecho convexo uma vez que as funções de base somam 1 (um) para todo valor de *u*.
- *Os pontos de controle não exercem controle local*. Mover um ponto de controle move toda a curva (As funções de base são diferentes de *0* em todo o domínio exceto em *u=0* e *u=1*).
- Os vetores tangentes à curva nos pontos extremos coincidem com a primeira e última aresta do polígono de controle.
- A curva não oscila sobre nenhuma reta mais do que oscila o polígono de controle (*propriedade de minimização de variação*).
- A curva pode ser transformada por *transformações afins* (translações e rotações) definidas sobre os pontos de controle.
- O controle exercido pelos pontos de controle não é local. A movimentação de um ponto altera toda curva, apesar de sua influência ser maior na vizinhança de tal ponto.

- Não é possível definir uma curva de Bérzier cúbica para aproximar ou representar um conjunto de *n* pontos sem utilizar múltiplos segmentos de curva.

O algoritmo de deCastlejau é um algoritmo utilizado para calcular um ponto sobre uma curva de Bérzier correspondente a um valor de parâmetro $t = t0$, $0 \leq t0 \leq 1$, baseado em aplicações repetidas de interpolações lineares. Considere, por exemplo, uma curva de Bézier de grau n=3 contendo os pontos *P0*,*P1*,*P2* e *P3*. A parametrização do segmento *P0P1* é dada por

$(1 - t)P0 + tP1$,

Para um dado valor de $t=t0$ podemos calcular um novo ponto de controle sobre P0P1 da seguinte forma:

$Q0 = (1 - t0)P0 + t0P1$.

Podemos aplicar o mesmo processo calculando para os segmentos P1P2 e P2P3 os pontos :

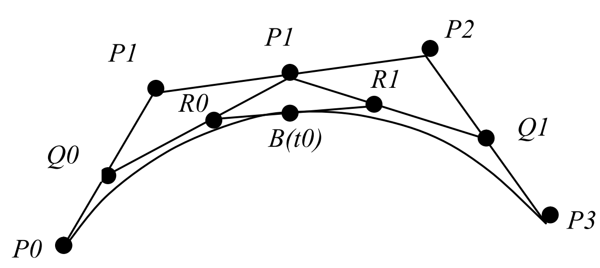
$Q1 = (1 - t0)P1 + t0P2$ e $Q2 = (1 - t0)P2 + t0P3$.

Em seguida, obtemos os pontos R0 e R1, interpolando para t=t0, respectivamente, os extremos dos segmentos Q0Q1 e Q1Q, conforme abaixo:

$R0 = (1 - t0)Q0 + t0Q1$ and $R1 = (1 - t0)Q1 + t0Q2$.

Finalmente, o ponto $B(t0)$ na curva é obtido interpolando-se *R0* e *R1*

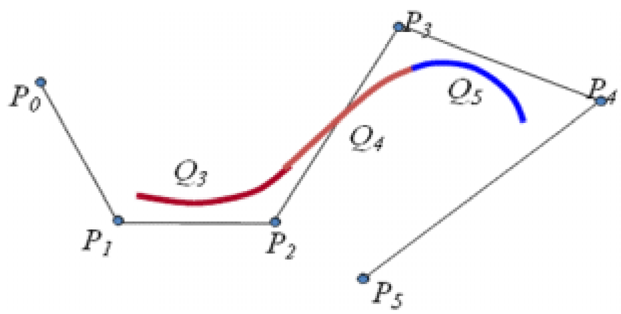
$B(t0) = (1 - t0)R0 + t0R1$



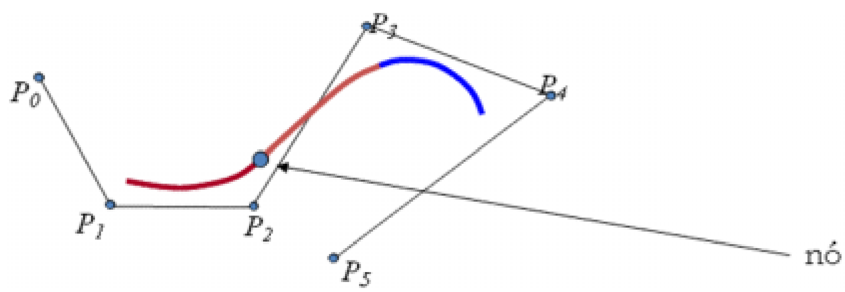
9) Defina uma curva B-Spline? (1.0 ponto).

Uma B-Spline é uma curva completa polinomial por partes consistindo de um conjunto de m-2 segmentos de curva $Q_3, Q_4, \ldots, Q_m$ definidos por $m+1$ pontos de controle $P_0, P_1, \ldots, P_m$, $m \geq 3$.

Cada segmento de curva é definido por quatro pontos de controle e quatro funções de mistura. Cada ponto de controle influencia somente quatro segmentos de curva.

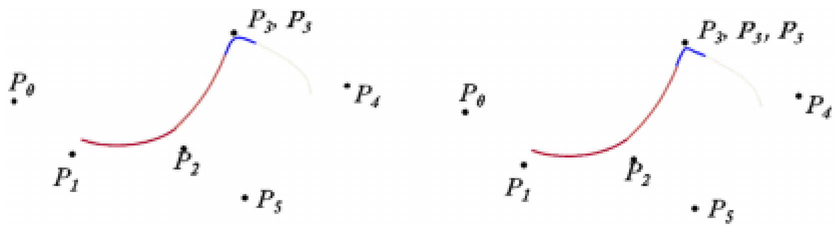


Os pontos associados a junções de dois segmentos adjacentes são denominados nós. O valor de u correspondente a um nó é denominado valor do nó. Uma B-spline é dita uniforme se valores dos nós são igualmente espaçados no espaço do parâmetro u.



Em geral, B-Splines não interpolam seus pontos de controle. Entretanto, é possível alcançar tal efeito de duas formas:

- Replicando pontos de controle, o que pode causar perdas de continuidade na curva



- Aumentando a multiplicidade dos nós, o que não causa perda de continuidade, mas que gera curvas B-Spline não uniformes, já que o intervalo no espaço de parâmetros entre os valores de nós não é mais o mesmo.

10) O que são NURBS? (1.0 ponto)

NURBS são B-Splines não uniformes dadas pela razão de 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 que regem o acomplamento da curva aos pontos de controle.

11) O que são *callbacks* de desenho, na OpenGL? (1.0 ponto).

As *callbacks* fazem parte fundamental do mecanismo de funcionamento da biblioteca de interface GLUT, normalmente vinculada a OpenGL.

Callbacks são funções criadas pelo programador que implementam uma função com certa assinatura, que são chamadas por um programa ou biblioteca (no caso a GLUT) quando da necessidade de se tratar algum evento. As callbacks são registradas através de funções que recebem um ponteiro para função com a assinatura especificada.