Računarska grafika

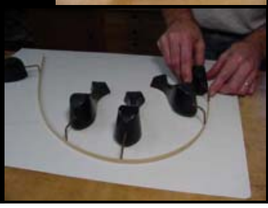
predavanja
v.prof.dr. Samir Lemeš
slemes@unze.ba

26. Parametarske krive

- Bézier krivulje
- Bézier funkcije miješanja
- Uniformni B-splajnovi
- NeUniformni, Racionalni B-Splajnovi (NURBS)
- Konverzija između splajnova
- Rasterizacija splajnova

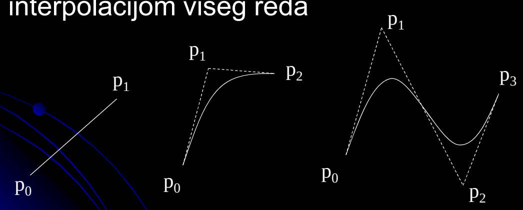
Splajnovi - historija

- Kontinuirane krivulje (*splines*) koje prolaze kroz zadane kontrolne tačke
- Kontinuitet drugog reda (zajedničke tačke, tangente (prvi) i drugi izvodi



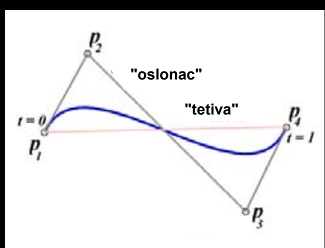
Bézier krivulje

- Bézier krivulje se mogu posmatrati kao proširenje linearne interpolacije interpolacijom višeg reda



Bézier krivulje

- Slične Hermitovim, ali imaju intuitivniju definiciju izvoda krajnjih tačaka
- Četiri kontrolne tačke, od kojih su dvije čvorovi



Bézier krivulje

- Vrijednosti izvoda Bezier krivulje u čvorovima zavise od susjednih tačaka

$$\nabla p_1 = 3(p_2 - p_1)$$
$$\nabla p_4 = 3(p_4 - p_3)$$

- Skalar 3 je izabran samo za ovu krivu, kao primjer, može imati bilo koju vrijednost.

Osnovna matrica Bézier krivulje

- Matrična forma

$$\begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{M}_{Bezier}} \underbrace{\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}}_{\mathbf{G}_{Bezier}}$$

- M_{Bezier} je dobra osnovna matrica jer ima dosta 0.

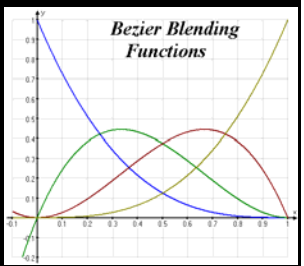
Bézier funkcije miješanja

- Funkcija miješanja definiše utjecaj kontrolne tačke na svaku tačku krivulje.
- Vrijednost 0 znači da kontrolna tačka nema utjecaja na tačku krivulje.
- Ako funkcija miješanja ima vrijednost 1, krivulja siječe kontrolnu tačku.
- Ova familija polinoma se naziva Bernstein polinomima 3. reda
  - $C(3, k)\ t^k\ (1-t)^{3-k}$; $0 <= k <= 3$
  - Svi su pozitivni na intervalu [0,1]
  - Njihov zbir iznosi 1

$$p(t) = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

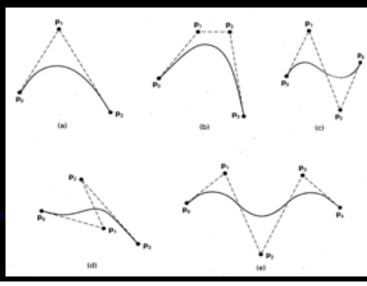
Bézier funkcije miješanja

- Svaka tačka na krivoj je linearna kombinacija kontrolnih tačaka
- Sve vrijednosti kombinacija su pozitivne
- Zbir vrijednosti je 1
- Krivulja je konveksna kombinacija kontrolnih tačaka

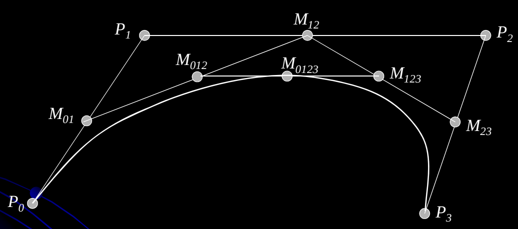


Konveksna kombinacija kontrolnih tačaka

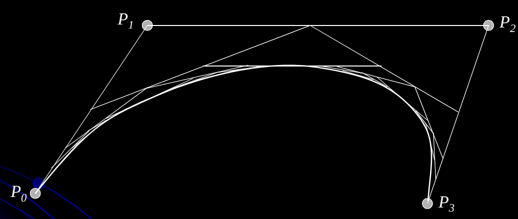
- Uvijek ostaje unutar graničnog regiona (konveksna ljuska) definisanog kontrolnim tačkama



Podjela Bézier krivulja

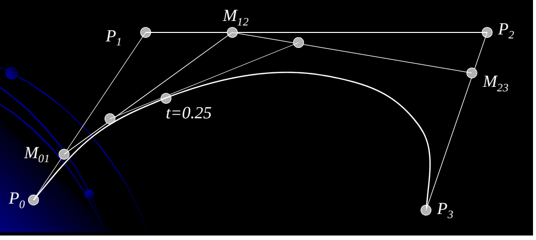


Podjela Bézier krivulja



de Castellijev algoritam

- Može se naći tačka na Bézier krivulji za svaku vrijednost parametra *t* sličnim algoritmom
- Ako se npr. želi *t=0.25*, umjesto da se uzmu srednje tačke, uzimaju se tačke na 0.25 dužine

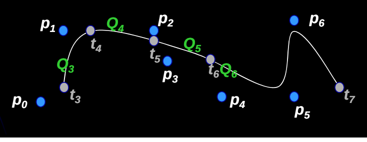


Zašto je potrebno još vrsta splajnova?

- Bézier i Hermitovi splajnovi imaju globalni utjecaj
  - Može se kreirati Bézier krivulja kojoj je trebalo 15 tačaka za definisanje krivulje…
    - Pomjeranjem jedne kontrolne tačke utječe se na cijelu krivulju
  - Kod diskontinuiranih Bézier ili Hermitovih krivulja se to ne dešava, ali zato kod njih nema obaveznog kontinuiteta izvoda u tačkama dodira
- B-splajnovi se sastoje od zakrivljenih segmenata čiji koeficijenti polinoma zavise samo od nekoliko kontrolnih tačaka
  - Lokalna kontrola

B-Splajn krive (kubne periodične)

- Polazi se od niza kontrolnih tačaka
- Izaberu se 4 tačke od sredine niza (p_{i-2}, p_{i-1}, p_i, p_{i+1}) d
  - Bézier i Hermite idu između p_{i-2} i p_{i+1}
  - B-Splajn ne interpolira (ne dodiruje) nijednu od njih ali aproksimira prolazeći kroz p_{i-1} i p_i



Uniformni B-splajnovi

- **Aproksimirajući** splajnovi
- Aproksimiraju n+1 kontrolnih tačaka
  - $P_0, P_1, \ldots, P_n$, n ¸ 3
- Krivulja se sastoji od n –2 segmenata kubnih polinoma
  - $Q_3, Q_4, \ldots Q_n$
- t se mijenja duž B-splajna kao Q_i: $t_i <= t < t_{i+1}$
- t_i (i = cijeli broj) su čvorne tačke koje spajaju segment Q_i sa segmentom Q_{i+1}
- Kriva je *uniformna* jer se čvorovi nalaze na jednakim intervalima parametra t

Uniformni B-splajnovi

- Prvi segment krivulje, Q_3, je definisan pomoću prve 4 kontrolne tačke
- Posljednji segment krivulje, Q_m, je definisan pomoću posljednje 4 kontrolne tačke, P_{m-3}, P_{m-2}, P_{m-1}, P_m
- Svaka kontrolna tačka utječe na 4 segmenta krivulje

Osnovna matrica B-splajna

- Formulisati 16 jednačina da se nađe 16 nepoznatih
- 16 jednačina forsiraju C_0, C_1, i C_2 kontinuitet između susjednih segmenata Q

$$M_{B-spline} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

B-splajn

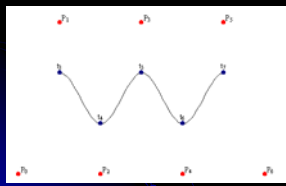
- Tačke duž B-splajna se izračunavaju kao i kod Bézier krivulje

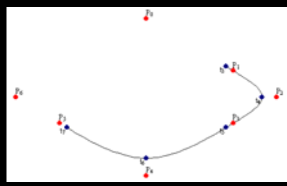
$$Q_i(t) = UM_{B-Spline}P$$

$$Q_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_i \\ p_{i+1} \\ p_{i+2} \\ p_{i+3} \end{bmatrix}$$

B-splajn

- Zasad je ovo najpopularniji splajn koji se koristi
- C_0, C_1, i C_2 kontinuitet

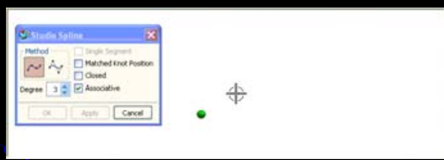




NeUniformni, Racionalni B-Splajnovi (NURBS)

- Osnovni element geometrije u 3D programu Maya
- Modeli se sastoje od površina definisanih pomoću NURBS-a, a ne poligona
- NURBS su glatki
- NURBS zahtijevaju poseban napor da bi se postiglo da ne budu glatki

NeUniformni, Racionalni B-Splajnovi (NURBS)



Konverzija između splajnova

- Posmatraju se dvije osnovne formulacije splajnova za dvije vrste splajnova

$$P = T \times M_{spline_1} \times G_{spline_1}$$

$$P = T \times M_{spline_2} \times G_{spline_2}$$

$$T \times M_{spline_1} \times G_{spline_1} = T \times M_{spline_2} \times G_{spline_2}$$

Konverzija između splajnova

- Moguće je transformisati kontrolne tačke iz jedne osnove splajna u drugu

$$P = T \times M_{spline_1} \times G_{spline_1}$$

$$P = T \times M_{spline_2} \times G_{spline_2}$$

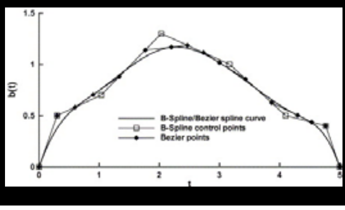
$$T \times M_{spline_1} \times G_{spline_1} = T \times M_{spline_2} \times G_{spline_2}$$

$$M_{spline_1} \times G_{spline_1} = M_{spline_2} \times G_{spline_2}$$

$$G_{spline_1} = M^{-1}_{spline_1} \times M_{spline_2} \times G_{spline_2}$$

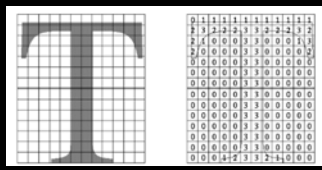
Konverzija između splajnova

- Ovom konverzijom, može se konvertovati B-splajn u Bézier splajn
- Bézier splajnovi su lakši za rasterizaciju (renderovanje) od B-splajnova



Rasterizacija splajnova

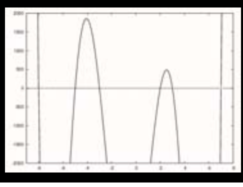
- Hornerov metod
- Inkrementalni metod (*Forward Difference*)
- Metod podjele na podsegmente



Hornerov metod

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$
$$x(t) = [(a_x t + b_x)t + c_x]t + d_x$$

- Numerički metod za približno računanje korijena polinoma.
- Tri množenja
- Tri sabiranja

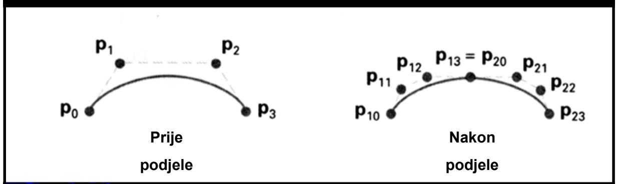


Forward Difference

$$x_{k+1} = x_k + \Delta x_k$$
$$x_k = a_x t^3 + b_x t^2 + c_x t + d$$
$$x_{k+1} = a_x (t_k + \delta)^3 + b_x (t_k + \delta)^2 + c_x (t_k + \delta) + d_x$$
$$x_{k+1} - x_k = \Delta x_k = 3a_x \delta t_k^2 + (3a_x \delta^2 + 2b_x \delta) t_k + (a_x \delta^3 + b_x \delta^2 + c_x \delta)$$

- Zahtjevno za izračunavanje
  - Traži se rješenje za promjenu za k (Δ_k) i promjenu za k+1 (Δ_{k+1})
  - Nastavak proračuna sa početnim vrijednostima x_0, Δ_0, i Δ_1
  - Izračunati x_3 dodavanjem $x_0 + \Delta_0 + \Delta_1$

Metod podjele na podsegmente



Rasterizacija Bézier splajna

```
public void spline(ControlPoint p0, ControlPoint p1,
                   ControlPoint p2, ControlPoint p3, int pix) {
   float len = ControlPoint.dist(p0,p1) + ControlPoint.dist(p1,p2)
             + ControlPoint.dist(p2,p3);
   float chord = ControlPoint.dist(p0,p3);
   if (Math.abs(len - chord) < 0.25f) return;
   fatPixel(pix, p0.x, p0.y);
   ControlPoint p11 = ControlPoint.midpoint(p0, p1);
   ControlPoint tmp = ControlPoint.midpoint(p1, p2);
   ControlPoint p12 = ControlPoint.midpoint(p11, tmp);
   ControlPoint p22 = ControlPoint.midpoint(p2, p3);
   ControlPoint p21 = ControlPoint.midpoint(p22, tmp);
   ControlPoint p20 = ControlPoint.midpoint(p12, p21);
   spline(p20, p12, p11, p0, pix);
   spline(p3, p22, p21, p20, pix);
}
```