28/02/07

Una delle più diffuse applicazioni dell'Informatica

Visualizza oggetti mediante il computer.

1) grafica per applicazioni economiche
2) animazione
3) progettazione

1 - Presenta risultati tramite grafici semplici, algoritmi non complicati
2 - Sfrutta l'elevata velocità dei computer per animare disegni, sensazione di movimento continuo (videogiochi, analisi cinematica di un fenomeno)
3 - Progettazione elettronica (circuiti o schede interne)
architettonica (abitabilità, disegno, vista)
industriale (progetto tecnico, studio estetico)
meccanica (CAD) rende automatico il controllo delle macchine utensili
(CAM) e la progettazione delle forme da realizzare

Sistema CAD esprimere come insieme di numeri la forma di un oggetto e operare su di essa

1 - Rappresentazione numerica
2 - Calcoli semplici
3 - Indipendenza dei sistemi coordinati
4 - Utilizzabili non matematici

Vogliamo trasformare curve e superfici, hanno queste 4 proprietà
X questo usiamo: RAPPRESENTAZIONE PARAMETRICA:
- esplicita
- forma vettoriale
- esprime una curva come combinazione lineare a coefficienti scalari
- indipendenza del sistema e riduco il coefficiente

CAGD: disciplina nuova che vuole mettere in evidenza gli aspetti matematici sviluppati negli anni '60

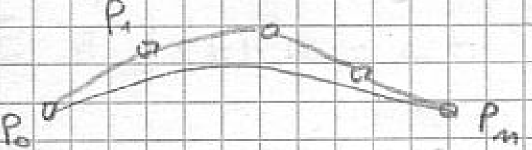
1° esempio di curve si è sviluppato negli anni '70

CURVE DI BEZIER (+ significativa)

$$B_n(t) = \sum_{k=0}^{n} B_{n,k}(t)\, P_k \qquad P_k \in R^3 \quad \text{punti di controllo}$$

$B_{n,k}$ → Blending

Poligono che si ottiene unendo i punti = poligono di controllo



Punti nello spazio, $n+1$ punti

$t \in [0,1]$ (variabile)

$$B_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k}$$ funzioni di base ($n+1$ addendi)

=> Curva Smooth (dolce)

PROPRIETÀ

1) $\sum_{k=0}^{n} B_{n,k}(t) = 1 \quad \forall t$ Ci garantisce che curva sia ben definita

Curva BEN DEFINITA = se dipende solo dai punti del poligono di controllo e non dal sistema di riferimento usato

DIM

$$\sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k} = (a+b)^n \qquad \forall a,b \quad \text{identità notevole}$$

$$\sum_{k=0}^{n} \binom{n}{k} t^k (1-t)^{n-k} = 1 \; ?$$ Voglio verificare che vale questa uguaglianza

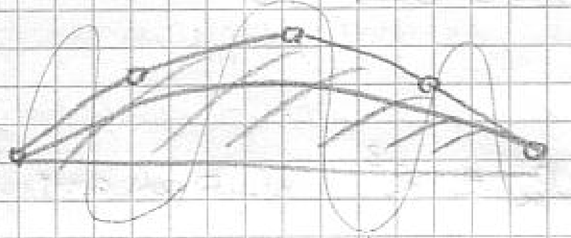
$a = t \;;\; b = 1-t$

$$\sum_{k=0}^{n} \binom{n}{k} t^k (1-t)^{n-k} = (t+1-t)^n = 1^n = 1 \quad \text{ok!}$$

=> Verificato che la Σ delle funzioni di Blending è 1

2) Proprietà del dominio convesso = CONVEX HULL

Dato un poligono di controllo, la curva che vado a costruire non si sposta molto dal poligono, giace nel dominio convesso del poligono di controllo



$B_{n,k} \geq 0 \quad \forall t \Rightarrow$ la curva si trova vicino al polig di controllo

$$B_{n,k}(t) = \underbrace{\binom{n}{k} t^k (1-t)^{n-k}}_{\geq 0} \qquad t \in [0,1] \quad t^k \overset{\text{è}}{>} 0 \,;\, 1-t \overset{\text{è}}{>} 0$$

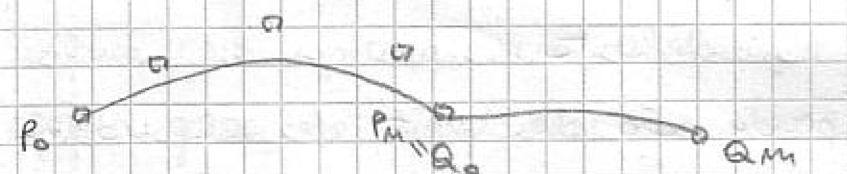
$\Rightarrow$ proprietà è verificata per le curve di Bezier

$\Rightarrow$ la curva giace nel convex hull

3) La curva è regolare = le funzioni che appaiono sono derivabili

$\Rightarrow$ curve smooth, non avremo spigolosità, sarà estremamente dolce

4) Interpolazione agli estremi



Per raccordare tratti $\neq$, legare curve $\neq$

A patto che la curva passi per i pti iniziali e finali del poligono

Il 1° punto della curva coincide con l'ultimo pto della 1ª curva

$$B_{n,k}(0) = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases} \qquad B_{n,k}(1) = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases}$$

Verificate dalle funzioni Blending di Bezier.

5) Simmetria: se inverto l'ordine dei pti del poligono, la curva non cambia

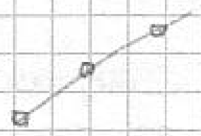
$$B_{n,k}(t) = B_{n,n-k}(1-t)$$ Verificata per Bezier

6) Riprodurre punti e rette

Punti: Se tutti i punti del poligono di controllo si riconducono ad un unico punto => curva degenera nel pto

$$\sum B_{n,k}(t) = 1$$

Rette: Condizione limite (non comune per la grafica)

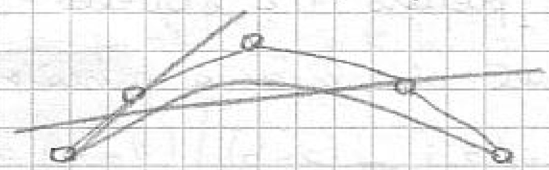


Se pti del polig di contr descrivono una retta, la curva deve essere una retta

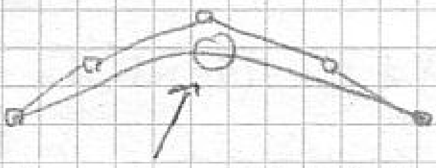
$$\sum B_{n,k}(t)\, k = nt$$

7) VARIATION DIMINUITING (riduzione delle oscillazioni)

Se traccio qualsiasi retta che taglia il polig di contr, il # delle intersez. di una qualsiasi retta della curva è ≤ al # delle intersez. della retta con il polig di controllo



8) Controllo locale (globale): 1° proprietà che le curve di Bezier non soddisfano



Spostando di poco 1 o 2 punti ne risente soltanto una parte locale della curva

disegno non soddisfatto di questa parte

Per le curve di Bezier ne risente tutta la curva

9) Il grado della curva dipende dal # di pti di controllo

Se pti da P_0 a P_n => grado $n-1$

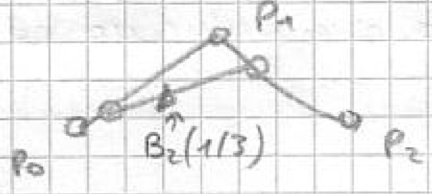
Svantaggio della curva di Bezier => non soddisfa il punto 9

Algoritmo x come disegnare le curve di Bezier. n alto

$$B_n(t) = \sum_{k=0}^{n} \binom{n}{k} t^k (1-t)^{n-k} P_k \qquad t \in [0,1] \qquad \binom{n}{k}! \; t^k (1-t)^{n-k}$$

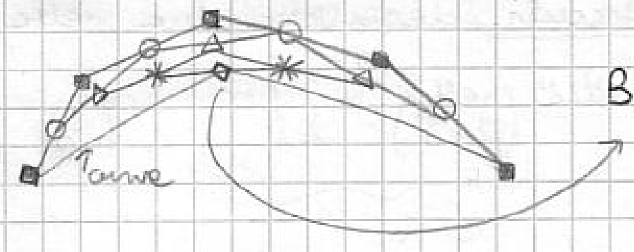
non è una rappresent comoda

Algoritmo di De Casteljau (ricorsivo)



$B_2(1/3)$

Traccio segmento tra 2 punti (P_0, P_1) e poi (P_1, P_2).
Individuo 1/3 dei due segmenti => trovato valore per la curva di Bez in quel punto



$B_4(1/2)$

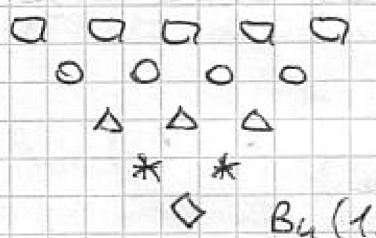
Sono passata da 5 pti di partenza a 4 punti
Unisco cerchietti con segmenti
Con triangolo trovo 1/2 dei risultanti segmenti
Trovo ancora 1/2 => unisco i punti ...
Quando trovo un unico segmento, prendo 1/2 e quello sarà il valore di $B_4(1/2)$ della curva in quel pto, la curva passa x quel pto

Algoritmo

b_i^0 punti di controllo di partenza

$b_i^r = (1-t)\, b_i^{r-1} + t\, b_{i+1}^{r-1}$ $\qquad b_0^r = b^r(t)$

$r = 1, \dots, n \qquad i = 0, \dots, n-2$



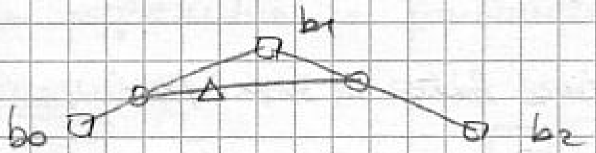
$B_4(1/2)$ valore della curva per $t = 1/2$

Metto t su base i pti di partenza
Come prima base i punti b^1 e poi b^2 ...

$$
\begin{array}{ccccccccc}
 & & & & b_0^4 & & & & \\
 & & & b_0^3 & & b_1^3 & & & \\
 & & b_0^2 & & b_1^2 & & b_2^2 & & \\
 & b_0^1 & & b_1^1 & & b_2^1 & & b_3^1 & \\
b_0 & & b_1 & & b_2 & & b_3 & & b_4
\end{array}
$$

Triangolo o piramide di Casteljau

Esempio $\quad t$ fissato



applico x la 1ª volta l'algoritmo

$$b_0^1 = (1-t)\, b_0 + t\, b_1$$

$$b_1^1 = (1-t)\, b_1 + t\, b_2$$

$$b_0^2 = (1-t)\, b_0^1 + t\, b_1^1 = B_2(t) \ ?$$

$$b_0^2 = (1-t)\left[(1-t)\, b_0 + t\, b_1\right] + t\left[(1-t)\, b_1 + t\, b_2\right] = \ldots =$$

$$= B_2(t) = \sum_{k=0}^{2} \binom{2}{k} t^k (1-t)^{2-k}\, b_k =$$

$$= b_0 (1-t)^2 + 2t(1-t)\, b_1 + t^2 b_2$$