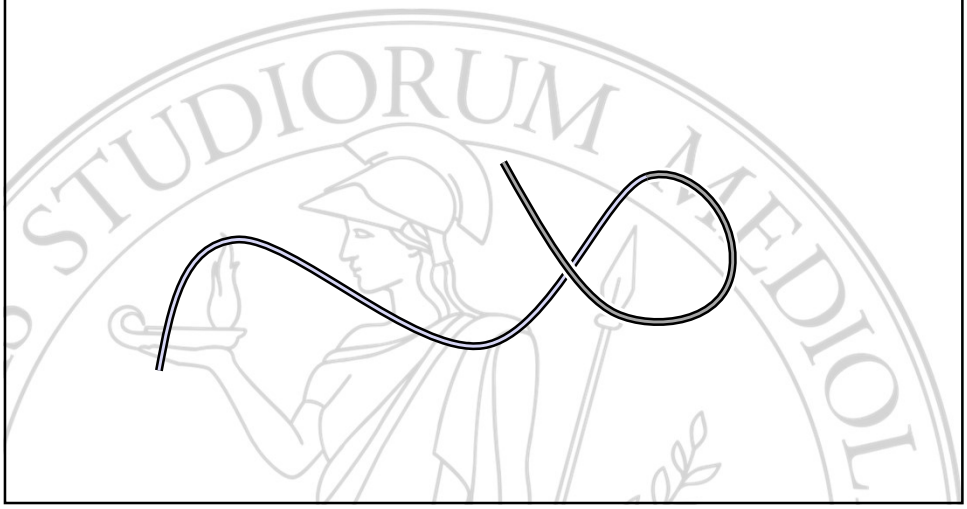
Marco Tarini - Computer Graphics 2024/2025
Università degli Studi di Milano

Parametric Curves 2/2



28

Curva di Bezier di grado 3 in forma di codice

L'algoritmo di De Casteljau ci fornisce un modo per codificare la funzione f in modo semplice:

```
vec3 bezier( float t, // posizione parametrica
             vec3 p0, vec3 p1,
             vec3 p2, vec3 p3)
{
   vec3 q0 = mix(p0,p1, t); // cioè
                            // p0*(1-t)+p1*t
   vec3 q1 = mix(p1,p2, t);
   vec3 q2 = mix(p2,p3, t);
   vec3 r0 = mix(q0,q1, t);
   vec3 r1 = mix(q1,q2, t);
   return mix( r0, r1, t);
}
```

29

Formulazione equivalente di Bèzier (grado 2)

✓ Per De Casteljau: $f(t) = (1-t)\mathbf{q}_0 + t\,\mathbf{q}_1$

con $\mathbf{q}_0 = (1-t)\mathbf{p}_0 + t\,\mathbf{p}_1$

e $\mathbf{q}_1 = (1-t)\,\mathbf{p}_1 + t\,\mathbf{p}_2$

✓ Sostituendo, otteniamo questa riscrittura:

$$f(t) = (1-t)^2\mathbf{p}_0 + 2\,(1-t)\,t\,\mathbf{p}_1 + t^2\mathbf{p}_2$$

⇒che è di grado 2 (t compare al quadrato: t^2)

✓ Lo riscrivo, usando $\bar{t} = 1 - t$, come:

$$f(t) = \bar{t}^2\mathbf{p}_0 + 2\,\bar{t}\,t\,\mathbf{p}_1 + t^2\mathbf{p}_2$$

⇒dove $\bar{t} = 1 - t$ ($\bar{t}$ è il «complementare» di t)

⇒cioè $\bar{t} + t = 1$

30

Formulazione equivalente di Bèzier (grado 3)

✓ Per De Casteljau: $f(t) = \bar{t}\,\mathbf{r}_0 + t\,\mathbf{r}_1$ $\quad \boxed{\bar{t} = 1 - t}$

con $\mathbf{r}_0 = \bar{t}\,\mathbf{q}_0 + t\,\mathbf{q}_1$

e $\mathbf{r}_1 = \bar{t}\,\mathbf{q}_1 + t\mathbf{q}_2$

con $\mathbf{q}_0 = \bar{t}\,\mathbf{p}_0 + t\,\mathbf{p}_1$

e $\mathbf{q}_1 = \bar{t}\,\mathbf{p}_1 + t\,\mathbf{p}_2$

e $\mathbf{q}_2 = \bar{t}\,\mathbf{p}_2 + t\,\mathbf{p}_3$

✓ Sostituendo tutto, otteniamo questa riscrittura:

$$f(t) = \bar{t}^3\mathbf{p}_0 + 3\,\bar{t}^2 t\,\mathbf{p}_1 + 3\,\bar{t}\,t^2\,\mathbf{p}_2 + t^3\mathbf{p}_3$$

…che è un polinomio cubico in t

31

Curve Bézier cubiche: codice ottimizzato

```
vec3 bezier( float t,
             vec3 p0, vec3 p1,
             vec3 p2, vec3 p3)
{
   float k = 1.0-t;
   return   (k*k*k)*p0 +
        (3.0*k*k*t)*p1 +
        (3.0*k*t*t)*p2 +
            (t*t*t)*p3 ;
}
```

33

Gli end-point degli archi di Bèzier e loro tangenti

- ✓ Gli end-point di arco di Bèzier coincidono con il primo e l'ultimo punto di controllo
  - ⇒ E' banale da verificare usando le formule sopra!
  - ⇒ Gli endpoints sono con $t = 0$, $\bar{t} = 1$ e con $t = 1$, $\bar{t} = 0$
- ✓ Come sono le direz. tangenti nei due end-point?
  - ⇒ Quanto vale la derivata f' nei punti parametrici $t = 0$ e $t = 1$
- ✓ La risposta (calcolando le derivate) è :
  - ⇒ all'inizio dell'arco, la tangente è orientata come il vettore che connette il 1mo pt. di controllo al 2do
  - ⇒ alla fine dell'arco, la tangente è orientata come il vettore che connette il penultimo pt. di controllo all'ultimo
- ✓ Vale per tutti i gradi (verificare per i gradi 2 e 3)

35

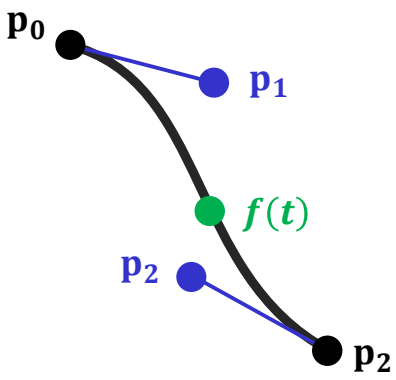
Curve di Bézier cubiche

- ✓ Il grado 3 (curve di Bézier cubiche) rappresenta il grado più conveniente da usare
- ✓ Ho quattro punti *distinti* che controllano la curva in modo intuitivo:
  - ⇒ Posizione e orientamento all'inizio della curva (i primi due)
  - ⇒ Posizione e orientamento alla fine della curva (gli ultimi due)
  - ⇒ Posso ottenere curve di Bézier con continuità G1 (e C1)
- ✓ Per paragone:
  - ⇒ I gradi > 3 non sono altrettanto utili e più difficili da manipolare, (cosa controlla il punto intermedio)?
  - ⇒ Coi gradi <3, non posso manipolare le due direzioni tangenti in modo indipendente una dall'altra
- ✓ Le curve cubiche sono ideali per fondere gli archi in una sola spline, detta «Bezièr curve» (o Bezièr path)
  - ⇒ Posso far coincidere le direzioni tangenti per ottenere una curva smooth

36

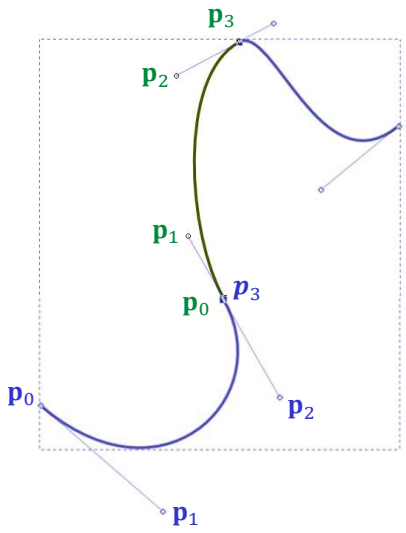
Bézier cubiche : recap

$$\begin{aligned} f(t) \quad = \quad & (1-t)^3 \, \mathbf{p_0} \\ + \, & 3\,t \, (1-t)^2 \, \mathbf{p_1} \\ + \, & 3\,t^2 \, (1-t) \, \mathbf{p_2} \\ + \, & t^3 \, \mathbf{p_3} \end{aligned}$$



37

Concatenare archi Bezier arch: *Bezier curve (o path)*



Se end-points coincidono ($\mathbf{p}_3 = \mathbf{p}_0$): curva continua C0

Se, inoltre ($\mathbf{p}_2$, $\mathbf{p}_3 = \mathbf{p}_0$, $\mathbf{p}_1$) sono allineati: curva continua con direz di tangente continua, cioè C1 (curva “smooth”-“liscia”)

38

Curve di Bezier

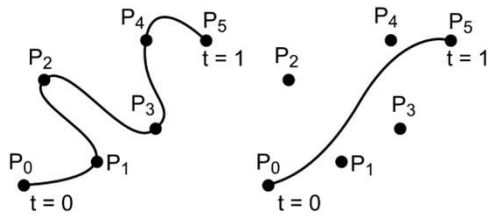
- ✓ Usati per:
  - ⇒ design 2D (molte applicazioni)
  - ⇒ Formato SVG (per immagini vettoriali 2D) (è anche un formato standard del Web!)
  - ⇒ specifica della forma delle lettere in un font
  - ⇒ Ideale per path di animazione: basta specificare posizione e velocità in una serie di punti sul percorso!
- ✓ Programma Open Source per sperimentare:

  - ⇒ Inkscape (http://inkscape.org)
  - ⇒ Test: disegnare una curva (chiamata path in Inkskape), salvare il progetto in formato SVG, e analizzare il file risultante come file di testo

39

Curve paramteriche con punti di controllo

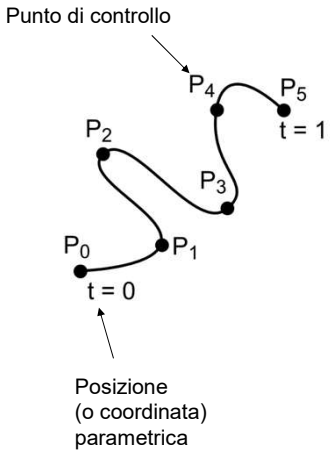
- ✓ La curve di Bèzier è solo un'esempio di schema
- ✓ In generale, uno schema consiste nel definire N punti di controllo e definire una curva che passa da questi punti (schema «interpolativo») oppure ci si avvicina (schema «approssimativo»)

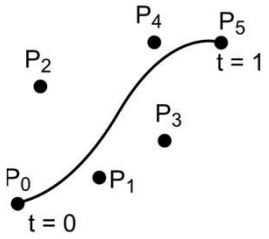


40

Curve paramteriche (terminologia)

- ✓ Schema interpolativo
- ✓ Schema approssimativo

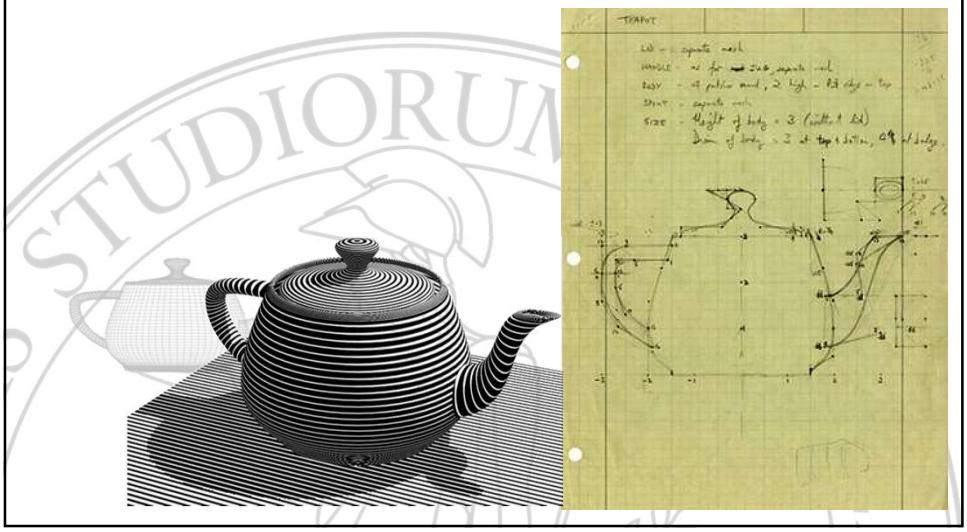




41

Marco Tarini - Computer Graphics 2024/2025
Università degli Studi di Milano

Parametric **Surfaces**



42

Superfici parametriche

✓ Siamo ora pronti per estendere le curve parametriche alle superfici parametriche

$$f: A \rightarrow B$$

dominio parametrico — immagine di f

$A \subseteq \mathbb{R}$ $B \subseteq \mathbb{R}^2$	$A \subseteq \mathbb{R}$ $B \subseteq \mathbb{R}^3$	$A \subseteq \mathbb{R}^2$ $B \subseteq \mathbb{R}^3$
B è una curva parametrica nel piano	B è una curva parametrica nello spazio	B è una **superficie** parametrica nello spazio

43

Superficie Parametrica

$f: A \to \mathbb{R}^3$

$A \subseteq \mathbb{R}^2$ dominio parametrico

$B \subseteq \mathbb{R}^3$ immagine di f

Superficie Parametrica:
immagine di una funzione da un dominio in R^2 a R^3

per definirne una, scegliere una funzione (e il suo dominio A) (il "dominio parametrico")

$$f\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

" x,y,z sono calcolate come formule di s,t "

44

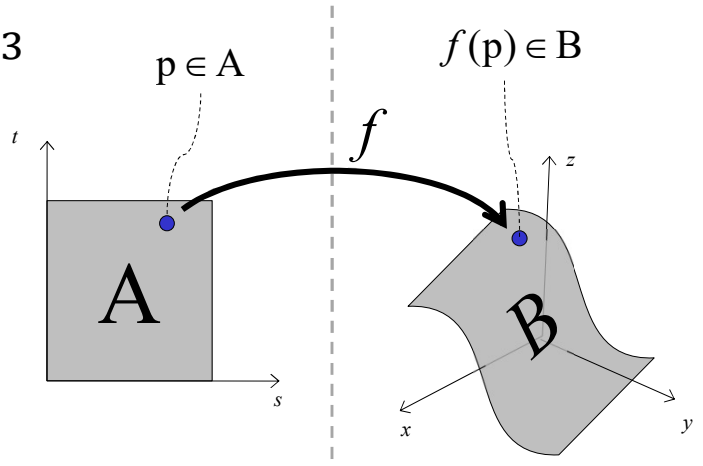
Superficie Parametrica

$f: A \to \mathbb{R}^3$

$A \subseteq \mathbb{R}^2$

$B \subseteq \mathbb{R}^3$

$$f\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



2D
dominio parametrico

3D

45

Sup. parametriche: convenzioni e terminologia

- ✓A = Dominio Parametrico = il dominio di f
  - ⇒Definito come regione di uno spazio bimensionale
  - ⇒Spesso, per convenzione, definito come il quadrato [0..1]x[0..1]
  - ⇒Per distinguere le sue coordinate da quelle usate come spazio oggetto (*x*,*y*,*z*), le chiamiamo *s*, *t* (a volte, *u* e *v*)
  - ⇒Come per le coordinate delle tessiture
- ✓B = immagine di f
  - ⇒Il luogo di punti raggiungibili da f a partire da A
  - ⇒Definisce la superficie parametrica

47

Bézier patches di grado 2 (o quadratiche)

- ✓Ho una griglia di (N+1)×(N+1) pts di controllo…
  - ⇒Es: con N = 2 ho: $p_{0,0}$ $p_{1,0}$ $p_{2,0}$ / $p_{0,1}$ $p_{1,1}$ $p_{2,1}$ / $p_{0,2}$ $p_{1,2}$ $p_{2,2}$
- ✓Con N = 2, algoritmo per trovare $p = f(s,t)$
  - ⇒cioè la pos 3D del punto di coord parametriche (s, t)
- ✓procedo così:
  - ⇒trovo a come $f(t)$ sulla Bézier curve $p_{0,0}$ $p_{1,0}$ $p_{2,0}$
  - ⇒trovo b come $f(t)$ sulla Bézier curve $p_{0,1}$ $p_{1,1}$ $p_{2,1}$
  - ⇒trovo c come $f(t)$ sulla Bézier curve $p_{0,2}$ $p_{1,2}$ $p_{2,2}$
  - ⇒trovo p come $f(s)$ sulla Bézier curve a,b,c

51

Bézier patches grado 3 «Bi-cubic Bézier patches»

✓ Algoritmo per trovare $\mathrm{p} = f(s,t)$:

⇒trovo **a** come $f(t)$ sulla Bèzier curve $\mathbf{p}_{0,0}$ $\mathbf{p}_{1,0}$ $\mathbf{p}_{2,0}$ $\mathbf{p}_{3,0}$

⇒trovo **b** come $f(t)$ sulla Bèzier curve $\mathbf{p}_{0,1}$ $\mathbf{p}_{1,1}$ $\mathbf{p}_{2,1}$ $\mathbf{p}_{3,1}$

⇒trovo **c** come $f(t)$ sulla Bèzier curve $\mathbf{p}_{0,2}$ $\mathbf{p}_{1,2}$ $\mathbf{p}_{2,2}$ $\mathbf{p}_{3,2}$

⇒trovo **d** come $f(t)$ sulla Bèzier curve $\mathbf{p}_{0,3}$ $\mathbf{p}_{1,3}$ $\mathbf{p}_{2,3}$ $\mathbf{p}_{3,3}$

⇒trovo **p** come $f(s)$ sulla Bèzier curve **a,b,c,d**

✓ Oppure, equivalentemente :

⇒trovo **a** come $f(s)$ sulla Bèzier curve $\mathbf{p}_{0,0}$ $\mathbf{p}_{0,1}$ $\mathbf{p}_{0,2}$ $\mathbf{p}_{0,3}$

⇒trovo **b** come $f(s)$ sulla Bèzier curve $\mathbf{p}_{1,0}$ $\mathbf{p}_{1,1}$ $\mathbf{p}_{1,2}$ $\mathbf{p}_{1,3}$

⇒trovo **c** come $f(s)$ sulla Bèzier curve $\mathbf{p}_{2,0}$ $\mathbf{p}_{2,1}$ $\mathbf{p}_{2,2}$ $\mathbf{p}_{2,3}$

⇒trovo **d** come $f(s)$ sulla Bèzier curve $\mathbf{p}_{3,0}$ $\mathbf{p}_{3,1}$ $\mathbf{p}_{3,2}$ $\mathbf{p}_{3,3}$

⇒trovo **p** come $f(t)$ sulla Bèzier curve **a,b,c,d**

52

Bezier patches grado 3 «Bi-cubic Bézier patches»

✓ N = 3

✓ Griglia di 4x4 vertici di controllo

⇒ $\mathbf{p}_{0,0}$ $\mathbf{p}_{1,0}$ $\mathbf{p}_{2,0}$ $\mathbf{p}_{3,0}$
$\mathbf{p}_{0,1}$ $\mathbf{p}_{1,1}$ $\mathbf{p}_{2,1}$ $\mathbf{p}_{3,1}$
$\mathbf{p}_{0,2}$ $\mathbf{p}_{1,2}$ $\mathbf{p}_{2,2}$ $\mathbf{p}_{3,2}$
$\mathbf{p}_{0,3}$ $\mathbf{p}_{1,3}$ $\mathbf{p}_{2,3}$ $\mathbf{p}_{3,3}$

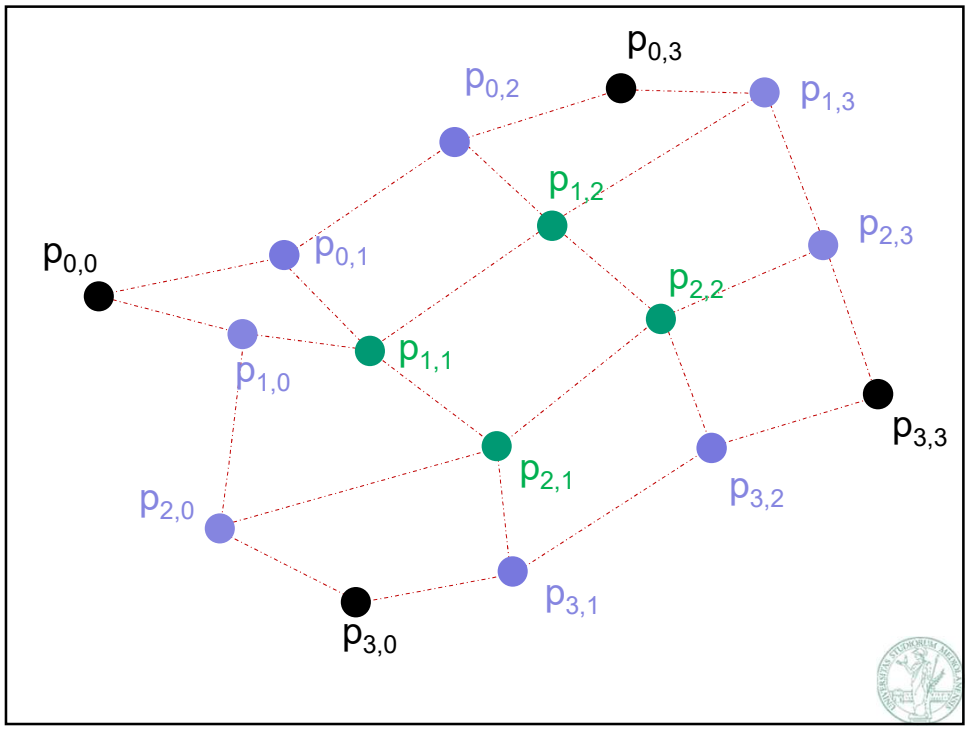
✓ Passa per i 4 angoli (è interpolante su loro)

p0,0 p0,3 p3,0 p3,3

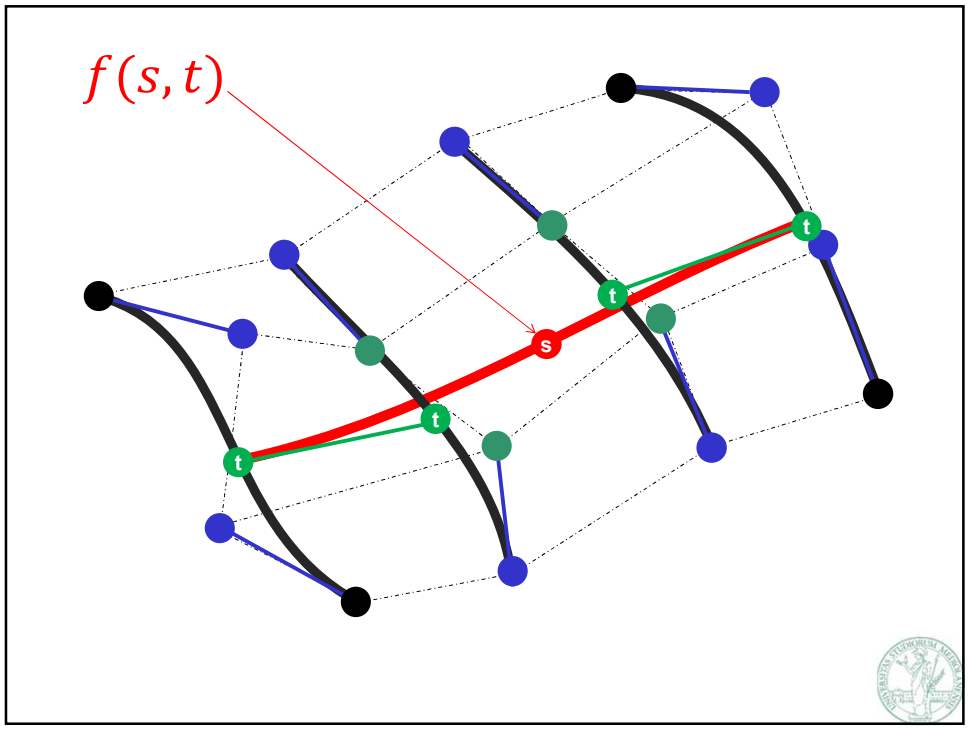
✓ E' approssimante sugli altri punti

✓ E detto anche «Bi-Cubico» perché è cubico sia sulla s, che sulla t

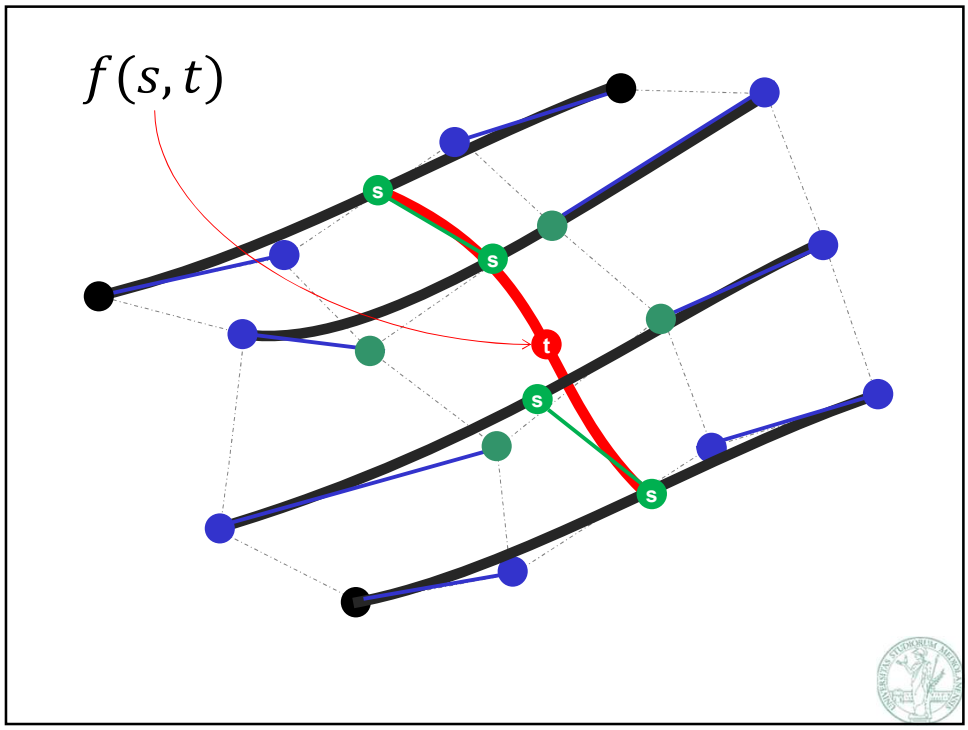
53


$p_{0,3}$
$p_{0,2}$
$p_{1,3}$
$p_{1,2}$
$p_{2,3}$
$p_{0,0}$
$p_{0,1}$
$p_{2,2}$
$p_{1,1}$
$p_{1,0}$
$p_{3,3}$
$p_{2,1}$
$p_{2,0}$
$p_{3,2}$
$p_{3,1}$
$p_{3,0}$

54

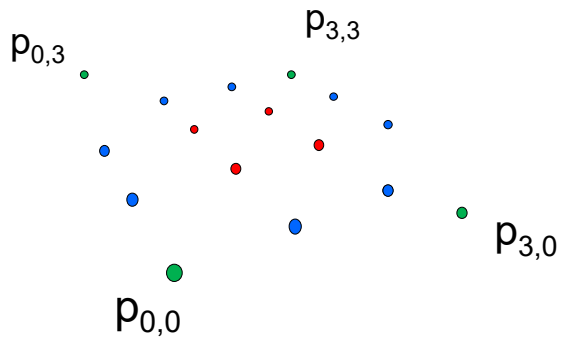

$f(s,t)$
s
t
t
t
t

55



56

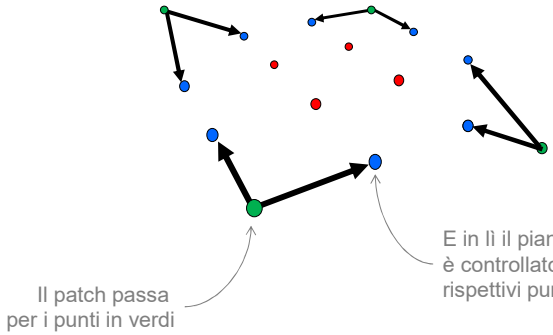
Bi-cubic Bézier patches: esempio



4x4 punti di controllo

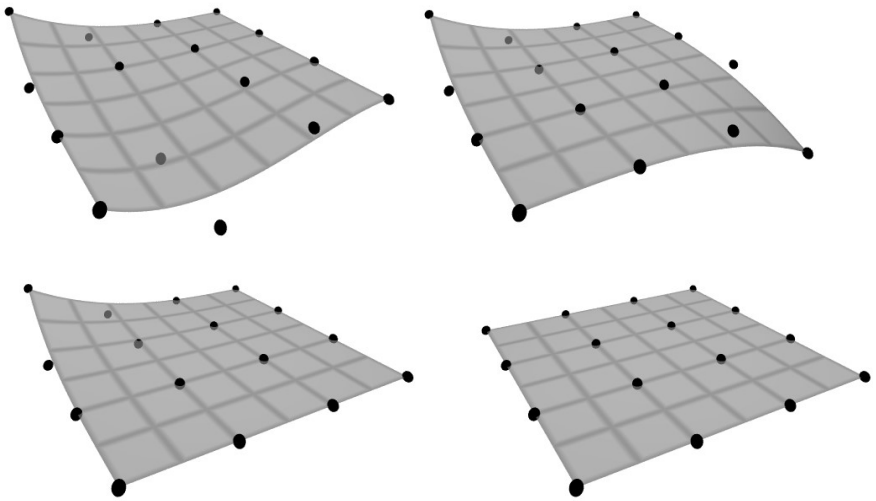
58

Bi-cubic Bézier patches



59

Cubic Bézier patches: esempi



Modificando I punti di controllo, modifico il patch. Prova, con:

http://kovacsv.github.io/JSModeler/documentation/examples/bezier.html

61

Superifici di Bézier

- ✓Così come una **curva** (o un **path**) di Bézier è costituito da molti **archi** connessi, una **superficie** di Bézier è costituita da molti **patch** connessi sui loro lati
  - ⇒Nuovamente, il matching dei punti di controllo sul bordo garantisce che la superficie sia continua, senza gaps (cioè G0)
  - ⇒Inoltre, imporre la colinearità fra i punti di controllo interni garantisce che la superficie non abbia crease (discontinuità di normale), cioè che sia smooth (lisca) (cioè G1) – laddove questo sia l'effetto desiderato!

62

Lo zoo delle superfici parametriche

- ✓Anche gli altri tipi di splines sono estendibili a patch, nello stesso modo:
  - ⇒Hermite Splines
  - ⇒Bezier splines
  - ⇒B-splines
  - ⇒NURBS
- ✓Le NURBS in particolare sono molto utilizzate in contesto CAD 3D

63

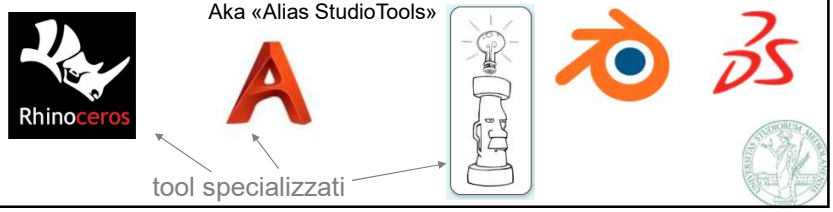
Superifici parametriche: benefici

- ✓ Controllabili, intuitive da modellare, ottime per il design
  - ⇒ Queste superfici so ideali per il design
  - ⇒ Soprattutto di tipo industriale / artificiale (CAID – Computer-aided industrial design)
  - ⇒ Controllo di curvatura e normale
- ✓ E' una rappresentazione compatta ed espressiva
  - ⇒ Basta memorizzare pochi pts di controllo
  - ⇒ Rappresentazione matematica di superfici dalla geometria *realmente* curva
  - ⇒ Paragona: mesh: lineari a tratti (curvatura solo simulata con normali o normal map)
- ✓ Risoluzione: numero di patch che compone la sup
  - ⇒ equivalentemente a meno di fattore, del num. di pt. controllo
  - ⇒ E' una res adattiva!

64

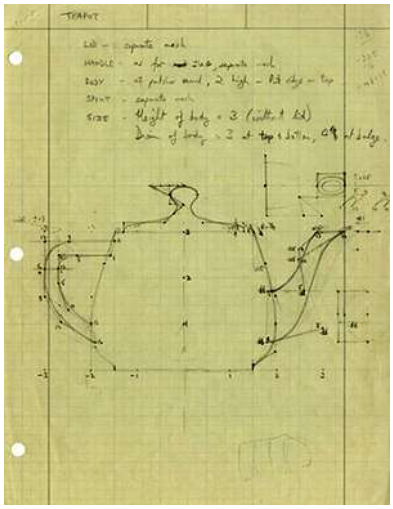
Superifici parametriche : benefici

- ✓ La superficie «nasce» già con un dominio parametrico e un uv-mapping incluso nella sua definizione
  - ⇒ per es, è triviale applicare il texture mapping ad un patch
  - ⇒ paragona col caso Mesh: parametrizzazione difficile
- ✓ Sono intuitive da manipolare per il designer del modello
  - ⇒ Esempio: la teiera dello Utha, Il famoso modello 3D, simbolo della Computer Graphic, è stato costruito (anni '70) come superficie di Bézier completamente «a mano», scrivendo i punti di controllo con carta e penna su un foglio (guardando una teiera fisica)
  - ⇒ Oggi, moltissime suite di modellazione 3D sono basate o includono superfici parametriche (NURBS, di Bézier, e/o altro)
  - ⇒ Es: Rhinoceros3d, Autodesk Alias, Aka «Alias StudioTools», moi3D, Blender, Solidworks



65

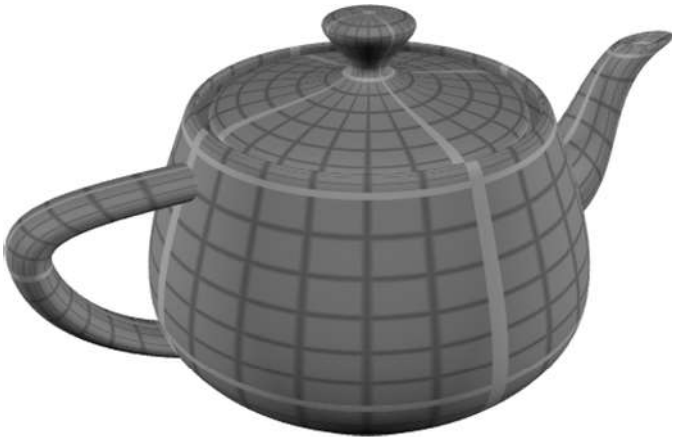
Bézier patches: per il design



The Utah Teapot, by Martin Newell (1975)

66

Bézier patches: utili per il desing



The Utah Teapot, by Martin Newell (1975)

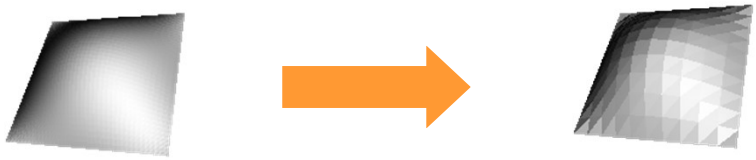
67

Processing su una superficie parametrica

- ✓ In molti contesti, la superficie parametrica è l'input o l'output diretto di task di geometry processing, incluso:
  - ⇒ Semplificazione: riduzione del numero di patch
  - ⇒ Costruzione automatica (a partire da un'altra rappresentazione, quale una mesh). Task difficile.
- ✓ Per molti usi, compreso il rendering, è necessario, viceversa, convertire la superficie parametrica in una mesh poligonale
  - ⇒ Fortunatamente, questo procedimento è semplice!
  - ⇒ Nota: può essere fatto «on demand», per es solo in fase di rendering, usando la rappresentazione originale (assai più compatta) per lo storing del modello 3D e qualsiasi altro uno

68

Da superficie parametrica a mesh



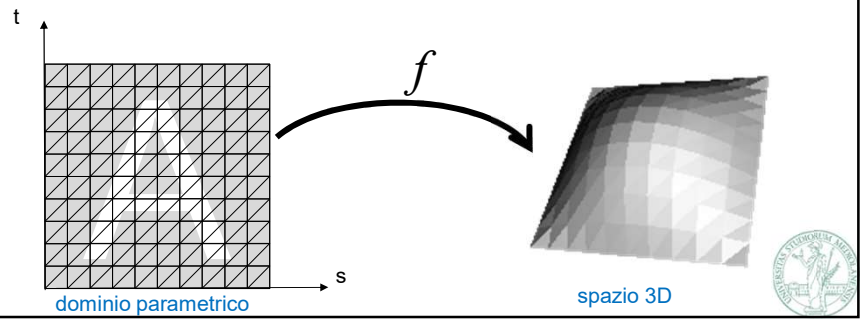
una superficie curva parametrica?

Mesh poligonale Poligoni

69

Da superficie parametrica a mesh poligonale

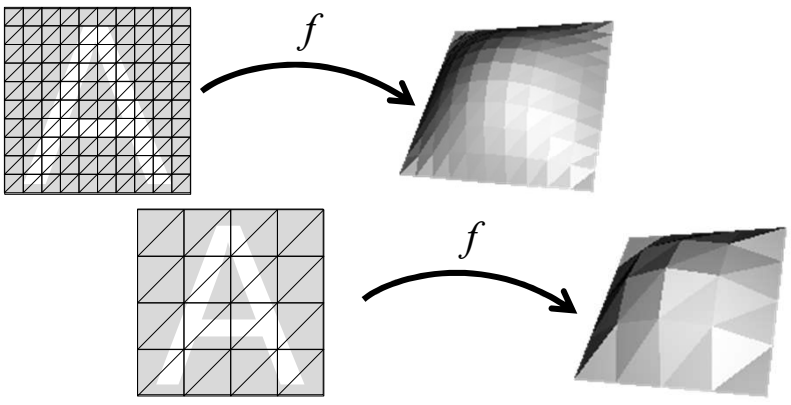
- ✓ Banale: campiono il dominio parametrico A su una griglia regolare (es.: 10x10)
  - ⇒ (s,t) = (0.0,0.0), (0.1,0.0), … (0.0,0.1), (0.1,0.1)… , (1.0, 1.0)
- ✓ Per ogni posizione campionata (s,t)
  - ⇒ creo un vertice della mesh nel punto $f(s,t)$
  - ⇒ doto ogni vertice della sua normale, ottenuta con le derivate di f
- ✓ Diagonal split so ogni quad ottenuto
- ✓ Risultato: tri-mesh regolare!



70

Da superficie parametrica a mesh

- ✓ Osservazione: quanto densamente tassellare può essere deciso liberamente, anche in fase di rendering
  - ⇒ La superficie rappresentata è curva: solo durante il rendering la approssimiamo con superfici triangolate



71