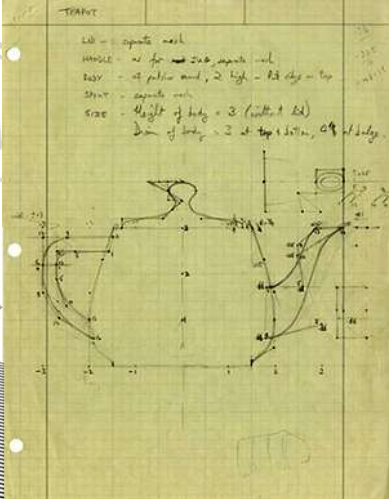


Marco Tarini - Computer Graphics 2020/2021
Università degli Studi di Milano

Parametric Surfaces

TELEDIDATTICA!



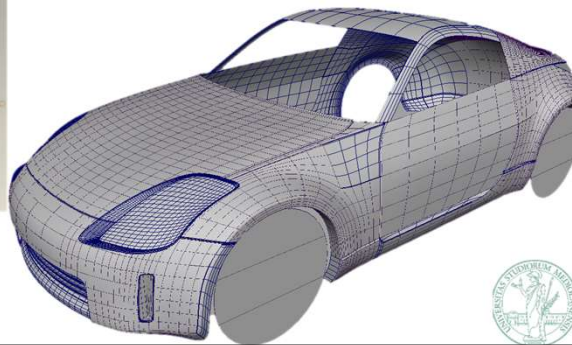
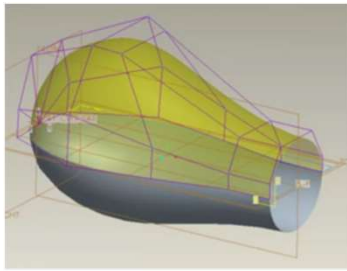
2

Una (imperfetta) categorizzazione dei tipi di modelli digitali 3D					
		ELEMENTI DISCRETI			CONTINUI
		regolari <i>«a griglia»</i>	semi-regolari o irregolari		
			elementi simpliciali	elementi non simpliciali	
SUPERFICIALI	2-manifold <i>«rappresenta una vera superficie»</i>	Height Field Range Scan Geometry Images	Triangle Mesh	Polygonal Mesh Quad Mesh Quad dominant Mesh	Subdivision surfaces Parametric Surfaces
	non-manifold <i>«non rappresenta una sup»</i>	Set di Range Scan		Point Cloud	
	VOLUMETRICI	(3-manifold)	Voxelized Volume Volumetric Textures	Tetra Mesh	Hexa Mesh

3

Superfici parametriche

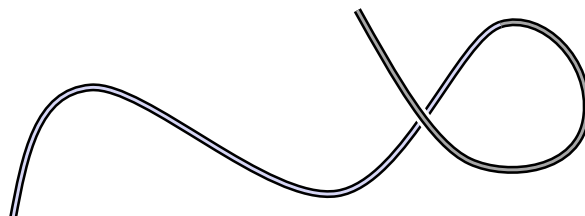
- ✓ Modo di rappresentare superfici curve
- ✓ Non solo «lineari a tratti» (come le mesh) ma «quadratiche (o cubiche, o quartiche) etc. a tratti»
- ✓ Molto utilizzato per CAD/CAM



4

Curve parametriche

- ✓ Vediamo prima un caso più semplice (ma pure molto usato):
le **curve parametriche**
 - ⇒ Una rappresentazione per linee curve (in 2D o 3D)
- ✓ Il caso **superfici** sarà una sua generalizzazione

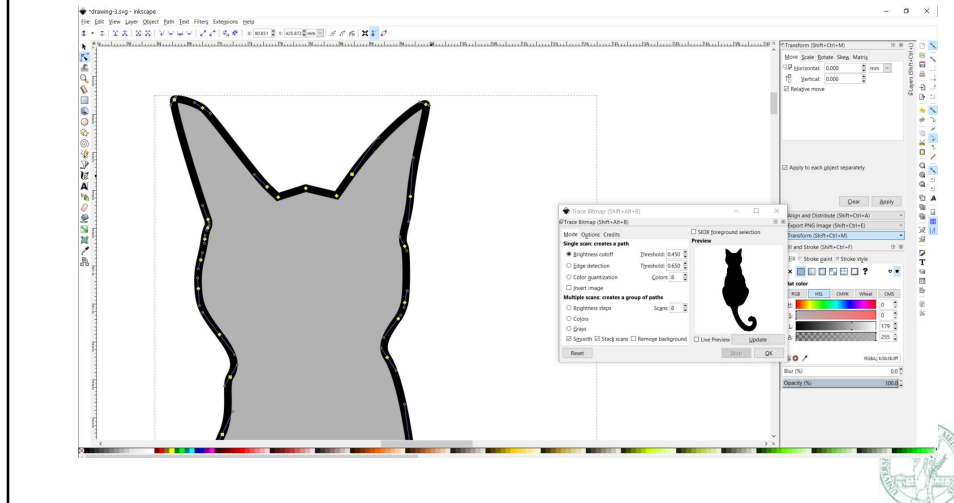


6

Curve parametriche

✓ Usi:

⇒ in 2D: grafica vettoriale



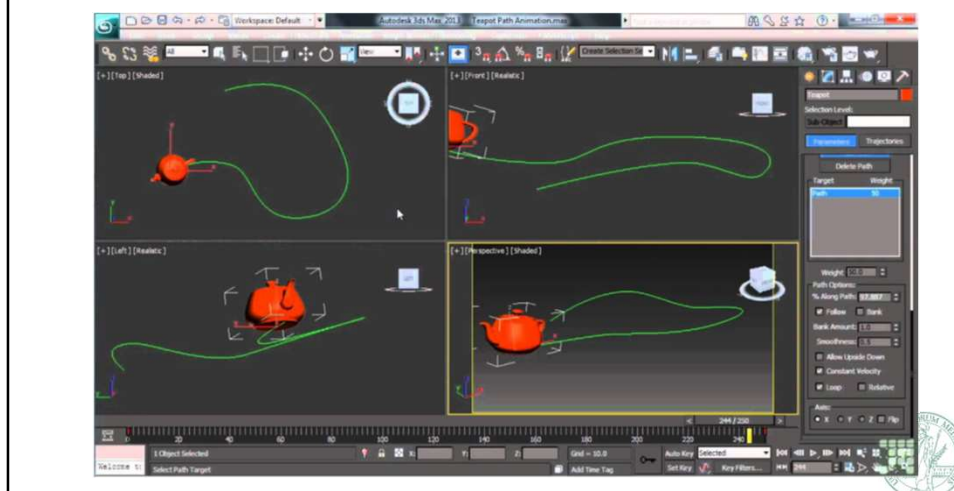
7

Curve parametriche

✓ Usi:

⇒ in 2D: grafica vettoriale

⇒ in 3D: percorso di un'animazione (qui: la coord parametrica t)



8

Curva parametrica

- ✓ Curva descritta da una funzione

$$f: A \rightarrow B$$

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

$$B \subseteq \mathbb{R}^2 \text{ o } \mathbb{R}^3$$

- ✓ $B \subseteq \mathbb{R}^2$ curve in 2D
(linee disegnate su di un foglio)

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

- ✓ $B \subseteq \mathbb{R}^3$ curve in 3D
(linee che “volano” nello spazio)

$$f(t) = \begin{pmatrix} x \\ y \end{pmatrix}$$

⇒ La curva è quindi “l’immagine di una funzione da $\mathbb{R}$ a $\mathbb{R}^2$ (o $\mathbb{R}^3$)”

- ✓ t è detto il parametro, o la posizione parametrica
- ✓ A , il dominio di f , è detto il “dominio parametrico”
- ✓ B , l’immagine di f , la nostra curva (un luogo di punti)
- ✓ Se $f(t)$ è un polinomio di t , la curva si dice polinomiale



9

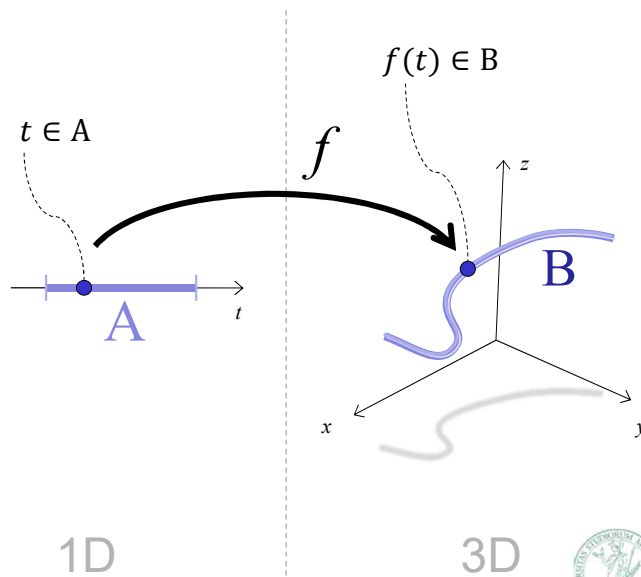
Curva parametrica

$$f: A \rightarrow B$$

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

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

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



10

Curve parametriche: quale direzione tangente?

- ✓ Dato un parametro t , il punto $f(t)$ è sulla curva (per costruzione).
- ✓ Quale dir tangente della curva in quel punto?
- ✓ Basta chiedersi di quanto si sposta il punto $f(t)$ se t aumenta di un piccolo delta dt .
- ✓ Risposta:

- ⇒ si sposta nel punto $f(t + dt)$
- ⇒ cioè si sposta del vettore $f(t + dt) - f(t)$
- ⇒ divido per dt , e ottengo lo spostamento (un vettore) per *unità di spostamento* nello dominio parametrico

$$\frac{f(t+dt) - f(t)}{dt}$$

- ⇒ il limite con $dt \rightarrow 0$ è la direzione tangente che cerco!
- ⇒ cioè la derivata f' di f calcolata in t

$$\frac{f'(t)}{\|f'(t)\|} = \text{vettore (unitario) tangente della curva nel punto } f(t)$$



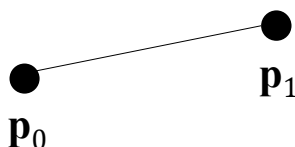
11

Esempio giocattolo:

- ✓ Segmento come “curva” parametrica
- ⇒ (anche se non è «curva» in questo caso)
- ⇒ Il segmento fra due punti p_0, p_1 è il luogo di punti raggiunti dalla funzione

$$f(t) = p_0 \cdot (1 - t) + p_1 \cdot t$$

$$= p_0 + (p_1 - p_0) \cdot t$$



Derivata in t ?
 Tangente?



13

Note sull'esempio giocattolo...

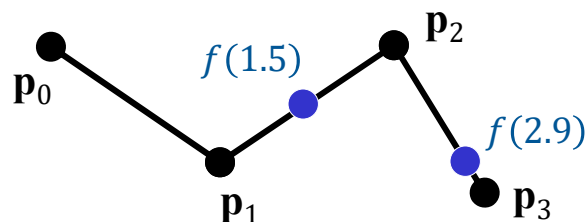
- ✓ In questo esempio, **ed anche in generale...**
 - ⇒ i punti $\mathbf{p}_0$ e $\mathbf{p}_1$ sono costanti nella formula di f
 - ⇒ sono i «**punti di controllo**» della curva: modificandoli, modifico la forma della curva
- ✓ in questo esempio... (ma in generale, non sarà sempre così)
 - ⇒ La curva è un polinomio lineare, di grado 1 (in generale, si usano polinomi di grado superiore)
 - ⇒ la curva passa attraverso i punti di controllo
 - ⇒ i punti di controllo sono solo 2
 - ⇒ i punti di controllo sono all'inizio e alla fine della curva
 - ⇒ la direzione tangente è costante per ogni t , (calcolarla!) cioè la «curva» non è... curva (è dritta, non è incurvata)
- ✓ Vedremo la prossima lezione esempi di grado superiore



14

Es: Linea Spezzata come "curva" parametrica

- ✓ Come concatenazione di "curve" lineari
 - ⇒ lineari = f polinomiali di grado 1 (in t)
- ✓ Cioè come curva «lineare a tratti»
 - ⇒ f è lineare "su ogni tratto"
- ✓ Esempio con tre tratti:



Dominio
 parametrico $A = [0, 3]$

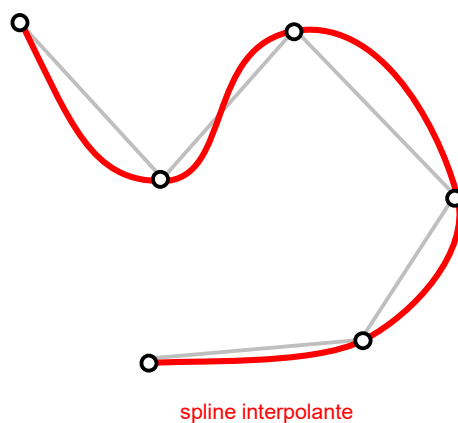
$$f(t) = \begin{cases} \mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0)(t - 0) & \text{se } t \in [0, 1] \\ \mathbf{p}_1 + (\mathbf{p}_2 - \mathbf{p}_1)(t - 1) & \text{se } t \in [1, 2] \\ \mathbf{p}_2 + (\mathbf{p}_3 - \mathbf{p}_2)(t - 2) & \text{se } t \in [2, 3] \end{cases}$$



15

Spline: una classe di curve parametriche

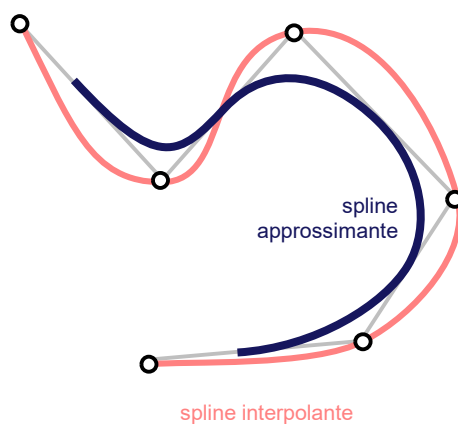
✓ Spline = curva *polinomiale* a tratti



16

Spline

✓ Spline = curva *polinomiale* a tratti



17

Spline

- ✓ E' una curva polinomiale a tratti
 - ⇒ Di un certo grado n
- ✓ Una spline è controllata attraverso un insieme **punti di controllo**
 - ⇒ Si dice "approssimante" se ci passa "vicino"
I punti di controllo attirano a sè la curva
 - ⇒ Si dice "interplante se ci passa attraverso"
La curva "interpola" i punti di controllo
- ✓ Una spline è divisa in **archi**.
- ✓ Ogni punto su ogni arco sarà una certa interpolazione lineare dei punti di controllo
 - ⇒ i pesi sono un polinomio $P(t)$, di grado $n > 1$
 - ⇒ n è il grado della spline
(che quindi può essere quadratica, cubica, etc)
- ✓ Vediamo ora un esempio importante: le curve di Bézier

Nell'esempio
giocattolo della linea
spezzata, $n = 1$,
ma normalmente, $n > 1$

nota: non si tratta di
una interpolazione
lineare
come quelle
abbiamo visto



18

Lo zoo delle splines

- ✓ Sono stati inventati diversi tipi di spline, ad es:
 - ⇒ Hermite Splines
 - ⇒ Bézier splines
 - ⇒ B-splines
 - ⇒ NURBS (polinomi razionali)
- ✓ Differiscono in:
 - ⇒ La formula che definisce f
 - ⇒ Se la spline è interpolativa o approssimativa
 - ⇒ Il grado di continuità raggiungibile
alle giunture fra due curve

vediamo queste



19

Curve di Bézier: note 1/2

- ✓ Sono una famiglia di splines di grado n qualsiasi
 - ⇒ La linea spezzata vista è una curva di Bézier di grado 1,
 - ⇒ Vediamo ora come si generalizza
- ✓ Alcune caratteristiche: ← Verificare per il grado 1
 - ⇒ Una curva di Beziér è composta da archi di Beziér (giunti alle estemità)
 - ⇒ Un arco di Beziér di grado n è controllato da $n+1$ punti di controllo
 - ⇒ L'arco è interpolante (passa attraverso) il primo e l'ultimo di questi punti, ma in generale non gli altri (la curva è solo approssimante per loro)
 - ⇒ Quindi, per rendere la curva continua («disegnabile senza staccare la penna dal foglio») basta far coincidere l'ultimo punto di controllo di un arco col primo dell'arco successivo



20

Curve di Bézier: note 2/2

- ✓ Le curve di Beziér cubiche ($n=3$) sono le più usate
 - ⇒ Vedremo perché
- ✓ Ci sono due modi equivalenti di definire un arco di curva di Beziér (di grado n):
 - ⇒ Metodo «grafico»: algoritmo di DeCasteljau
 - ⇒ Riformulazione matematica, come polinomio (anni 50, Pierre Beziér)
- ✓ Cenni storici:
 - ⇒ le curve di Beziér sono nate per progettare le curve delle fusoliere delle automobili.



21

Formulaz. di De Casteljau: esempio per grado 2

- ✓ Ho tre «control point» $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$
- ✓ Data la coordinata parametrica $t \in [0..1]$, trovo :
 - ⇒ il punto $\mathbf{q}_0$ interpolando fra $\mathbf{p}_0$ e $\mathbf{p}_1$ (con t)
 - ⇒ il punto $\mathbf{q}_1$ interpolando fra $\mathbf{p}_1$ e $\mathbf{p}_2$ (con t)
- ✓ Infine
 - ⇒ ottengo il punto finale $f(t)$ interpolando fra $\mathbf{q}_0$ e $\mathbf{q}_1$ (di nuovo con t)

(Esercizio: seguire le istruzioni su un disegno)

Demo: <http://www.malinc.se/m/DeCasteljauAndBezier.php>



22

Formulaz. di De Casteljau: esempio per grado 3

- ✓ 4 «control point» $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$
- ✓ data una posizione parametrica t , trovo:
 - ⇒ il punto $\mathbf{q}_0$ interpolando fra $\mathbf{p}_0$ e $\mathbf{p}_1$ (con t)
 - ⇒ il punto $\mathbf{q}_1$ interpolando fra $\mathbf{p}_1$ e $\mathbf{p}_2$ (con t)
 - ⇒ il punto $\mathbf{q}_2$ interpolando fra $\mathbf{p}_2$ e $\mathbf{p}_3$ (con t)
- ✓ poi
 - ⇒ il punto $\mathbf{r}_0$ interpolando fra $\mathbf{q}_0$ e $\mathbf{q}_1$ (con t)
 - ⇒ il punto $\mathbf{r}_1$ interpolando fra $\mathbf{q}_1$ e $\mathbf{q}_2$ (con t)
- ✓ e infine
 - ⇒ il punto finale $f(t)$ interpolando fra $\mathbf{r}_0$ e $\mathbf{r}_1$ (con t)

(Esercizio: seguire le istruzioni su un disegno)



23

Curva di Bezier di grado 3 in forma di codice

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

```
vec3 bezier( float t, // posiz. paramet.
             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);
}
```

24

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, troviamo un f come polinomio di t

$$f(t) = \mathbf{p}_0 + t(\mathbf{p}_1 - 2\mathbf{p}_0) + t^2(\mathbf{p}_0 - \mathbf{p}_1 + \mathbf{p}_2)$$
 - ⇒ avete i pts di controllo $\mathbf{p}_{0,1,2}$ come costanti
 - ⇒ di grado 2, (t compare al quadrato: t^2)
- ✓ Se raccogliamo $\mathbf{p}_{0,1,2}$ otteniamo questa riscrittura:

$$f(t) = (1-t)^2\mathbf{p}_0 + (1-t)t\mathbf{p}_1 + t^2\mathbf{p}_2$$
- ✓ Lo riscrivo, usando $\bar{t} = 1-t$, come:

$$f(t) = \bar{t}^2\mathbf{p}_0 + \bar{t}t\mathbf{p}_1 + t^2\mathbf{p}_2$$
 - ⇒ dove $\bar{t} = 1-t$ ($\bar{t}$ è il complemento a 1 di t)
 - ⇒ cioè $\bar{t} + t = 1$

25

Formulazione di Bèzier (grado generico n)

- ✓ Modo pratico di costruire (*pro tip*: e ricordare) la formula di Bezier di qualsiasi grado n
 - ⇒ Step 1: scomporre l'unità in t e $\bar{t}$: $\bar{t} + t = 1$
 - ⇒ Step 2: elevare alla n entrambi i lati $(\bar{t} + t)^n = 1^n = 1$
 - ⇒ Step 3: ottengo una scomposizione dell'unità in $n + 1$ termini
Esempi, per $n = 2, 3, 4$:
$$(\bar{t} + t)^2 = \bar{t}^2 + 2 \bar{t} t + t^2 = 1$$
$$(\bar{t} + t)^3 = \bar{t}^3 + 3 \bar{t}^2 t + 3 \bar{t} t^2 + t^3 = 1$$
$$(\bar{t} + t)^4 = \bar{t}^4 + 4 \bar{t}^3 t + 6 \bar{t}^2 t^2 + 3 \bar{t} t^3 + t^4 = 1$$
 - ⇒ Step 4: questi termini sono i pesi degli $n + 1$ control points

- ✓ Esempi:

- ⇒ Grado 2: $f(t) = \bar{t}^2 \mathbf{p}_0 + 2 \bar{t} t \mathbf{p}_1 + t^2 \mathbf{p}_2$
- ⇒ Grado 3: $f(t) = \bar{t}^3 \mathbf{p}_0 + 3 \bar{t}^2 t \mathbf{p}_1 + 3 \bar{t} t^2 \mathbf{p}_2 + t^3 \mathbf{p}_3$
- ⇒ Grado 4: $f(t) = \bar{t}^4 \mathbf{p}_0 + 4 \bar{t}^3 t \mathbf{p}_1 + 6 \bar{t}^2 t^2 \mathbf{p}_2 + 3 \bar{t} t^3 \mathbf{p}_3 + t^4 \mathbf{p}_4$

Verificare
anche per
il grado 1

26

Curve Bézier cubiche: codice ottimizzato

```
vec3 bezier( in float t,
             in vec3 p0, in vec3 p1,
             in vec3 p2, in 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 ;
}
```

27

Gli endpoint degli archi di Bèzier

- ✓ Gli endpoint di arco di Bezier coincidono con il primo e l'ultimo punto di controllo (ma non dagli altri)
 - ⇒ E' banale verificare con 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 endpoints?
 - ⇒ Quanto vale la derivata f' nei punti parametrici $t=0$ e $t=1$
- ✓ La risposta (calcolando le derivate) è :
 - ⇒ all'inizio dell'arco, è sempre orientata come il vettore che connette il 1mo pt. di controllo al 2do
 - ⇒ alla fine dell'arco, è sempre orientata come il vettore che connette il penultimo pt. di controllo all'ultimo
- ✓ Vale per tutti i gradi (verificare per i gradi 1 e 2)



28

Curve di Bèzier cubiche

- ✓ Gli endpoint di arco di Bèzier coincidono con il primo e l'ultimo punto di controllo (ma non dagli altri)
 - ⇒ E' banale verificare con 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 endpoints?
 - ⇒ Quanto vale la derivata f' nei punti parametrici $t=0$ e $t=1$
- ✓ La risposta (calcolando le derivate) è :
 - ⇒ all'inizio dell'arco, è sempre orientata come il vettore che connette il 1mo pt. di controllo al 2do
 - ⇒ alla fine dell'arco, è sempre orientata come il vettore che connette il penultimo pt. di controllo all'ultimo
- ✓ Vale per tutti i gradi (verificare per i gradi 1 e 2)



29

Curve di Bézier cubiche

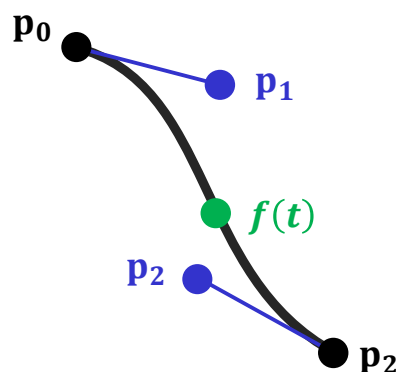
- ✓ Il grado 3 (curve di Bézier cubiche) rappresenta il grado più conveniente da usare
- ✓ Ho quattro punti *distinti* che controllano la curva in modo intuitivo:
 - ⇒ Posizione e orientamento all'inizio della curva (i primi due)
 - ⇒ Posizione e orientamento alla fine della curva (gli ultimi due)
 - ⇒ 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



30

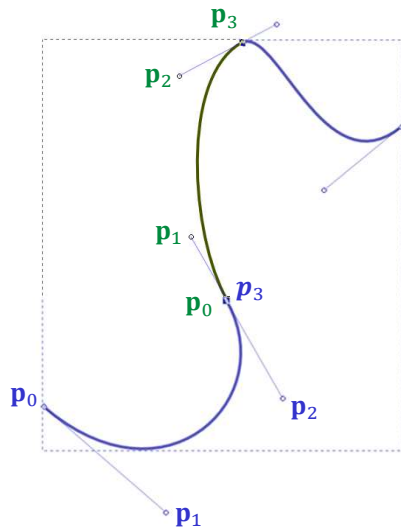
Bézier cubiche : recap

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



31

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



Se end-points
coincidono ($p_3 = p_0$):
curva continua C0

Se, inoltre
($p_2, p_3 = v_0, p_1$)
allineati:
curva continua
con direz di tangente
continua, C1
(curva “smooth”-
“liscia”)



32

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:



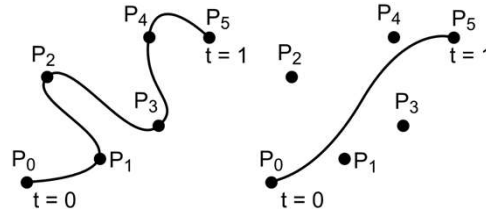
- ⇒ Inkscape (<http://inkscape.org>)
- ⇒ Test: disegnare una curva
(chiamata path in Inkscape),
salvare il progetto in formato SVG,
e analizzare il file risultante
come file di testo



33

Curve parametriche con punti di controllo

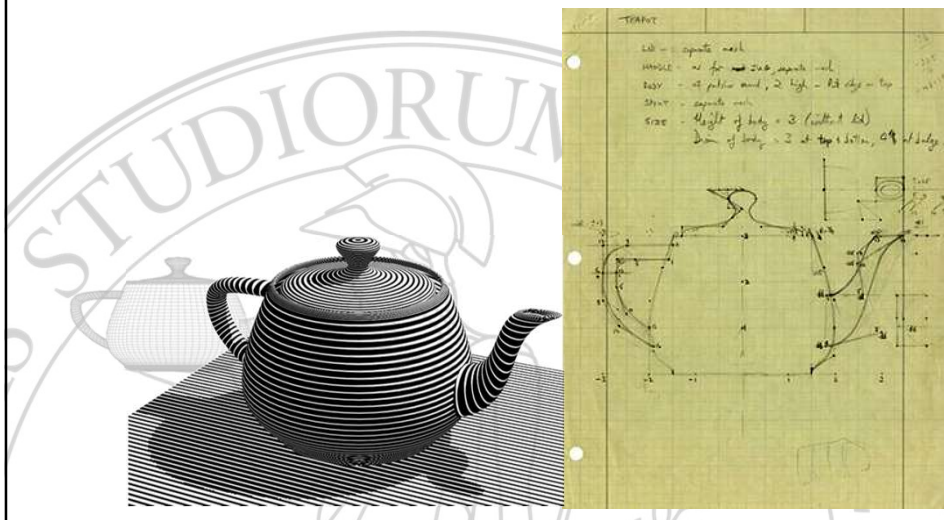
- ✓ La curve di Bèzier è solo un'esempio di schema
- ✓ In generale, uno schema consiste nel definire N punti di controllo e definire una curva che passa da questi punti (schema interpolativo) oppure ci si avvicina (schema approssimativo)



34

Marco Tarini - Computer Graphics 2019/2020
 Università degli Studi di Milano

Parametric Surfaces



35

Superfici parametriche

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

$$f: A \rightarrow B$$

dominio parametrico immagine di f

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

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

B è una
curva
parametrica
nel piano

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

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

B è una
curva
parametrica
nello spazio

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

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

B è una
superficie
parametrica
nello spazio

36

Superficie Parametrica

$$f: A \rightarrow B$$

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

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

Superficie Parametrica:

immagine di una
funzione da
un dominio in $\mathbb{R}^2$
a $\mathbb{R}^3$

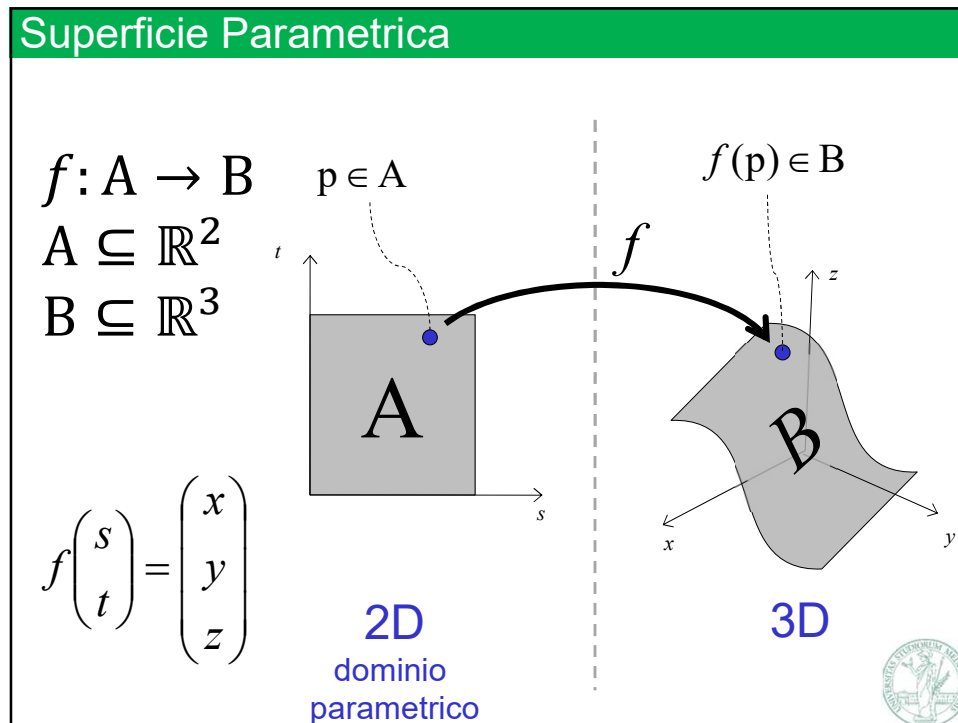
per definirla una,
scegliere una funzione
(e il suo dominio A)
(il "dominio parametrico")

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

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



37



38

Sup. parametriche: convenzioni e terminologia

- ✓ **A = Dominio Parametrico = il dominio di f**
 - ⇒ Definito come regione di uno spazio bidimensionale
 - ⇒ Spesso, per convenzione, definito come il quadrato $[0..1] \times [0..1]$
 - ⇒ 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 tessiture!
- ✓ **B = immagine di f**
 - ⇒ Il luogo di punti raggiungibili da f a partire da A
 - ⇒ Costituisce la superficie parametrica

40

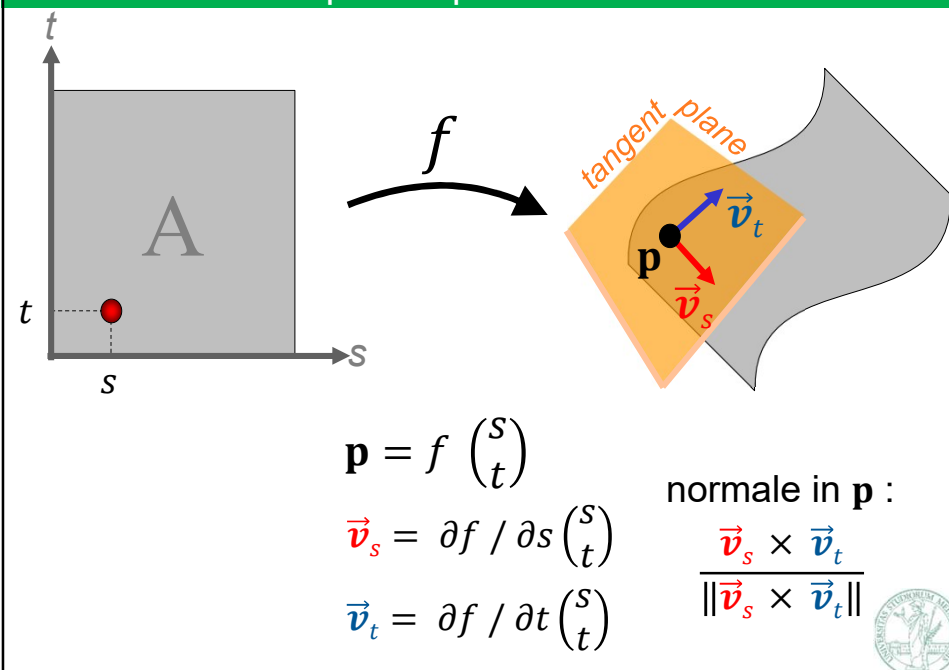
Normale di una superficie parametrica

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

- ✓ «Muovendomi» in direzione s e t sul dominio parametrico A , mi sposto sulla superficie B
- ✓ Posso derivare f sia rispetto a s , sia rispetto a t
 - ⇒ Derivate “parziali” di f , $\partial f / \partial s$ e $\partial f / \partial t$
 - ⇒ Sono gli spostamenti in 3D che effettuo incrementando s o t
- ✓ Ottenengo due vettori tangenti alla superficie
 - ⇒ Quindi due vettori sul piano tangente alla superficie
- ✓ La normale della superficie in un punto è data dal prodotto cross fra questi due vettori! (rinormalizzato)

41

Normale di una superficie parametrica



42

Superficie parametriche e punti di controllo

- ✓ Come nel caso delle spline, una classe popolare di superfici parametriche è costituita
 - ⇒ $f \begin{pmatrix} s \\ t \end{pmatrix}$ è una interpolazione lineare di un certo numero di punti 3D «di controllo»
 - ⇒ i pesi sono dati da polinomi (ora, in s e t), secondo una formula predeterminata
 - ⇒ Memorizzare la superficie consiste nel memorizzare i punti
- ✓ Posso usare lo schema di una qualsiasi spline per costruire il corrispondente «patch» (una superf.)
 - ⇒ Quindi esistono: patch di Beziér, patch NURBS, ...
- ✓ Idea intuitiva: la superficie è un tessuto di linee parametriche
 - ⇒ come la trama e l'ordito in un tessuto



43

Bezier patches di grado N

- ✓ Ho una griglia di $(N+1) \times (N+1)$ 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 bezier curve $p_{0,0}$ $p_{1,0}$ $p_{2,0}$
 - ⇒ trovo b come $f(t)$ sulla bezier curve $p_{0,1}$ $p_{1,1}$ $p_{2,1}$
 - ⇒ trovo c come $f(t)$ sulla bezier curve $p_{0,2}$ $p_{1,2}$ $p_{2,2}$
 - ⇒ trovo p come $f(s)$ sulla bezier curve a, b, c



44

Bezier patches grado 3 «Bi-cubic Bézier patches»

- ✓ Algoritmo per trovare $\mathbf{p} = f(s, t)$:
 - ⇒ trovo **a** come $f(t)$ sulla bezier curve $\mathbf{p}_{0,0}$ $\mathbf{p}_{1,0}$ $\mathbf{p}_{2,0}$ $\mathbf{p}_{3,0}$
 - ⇒ trovo **b** come $f(t)$ sulla bezier curve $\mathbf{p}_{0,1}$ $\mathbf{p}_{1,1}$ $\mathbf{p}_{2,1}$ $\mathbf{p}_{3,1}$
 - ⇒ trovo **c** come $f(t)$ sulla bezier curve $\mathbf{p}_{0,2}$ $\mathbf{p}_{1,2}$ $\mathbf{p}_{2,2}$ $\mathbf{p}_{3,2}$
 - ⇒ trovo **d** come $f(t)$ sulla bezier curve $\mathbf{p}_{0,3}$ $\mathbf{p}_{1,3}$ $\mathbf{p}_{2,3}$ $\mathbf{p}_{3,3}$
 - ⇒ trovo **p** come $f(s)$ sulla bezier curve **a,b,c,d**
- ✓ Oppure, equivalentemente
 - ⇒ trovo **a** come $f(s)$ sulla bezier curve $\mathbf{p}_{0,0}$ $\mathbf{p}_{0,1}$ $\mathbf{p}_{0,2}$ $\mathbf{p}_{0,3}$
 - ⇒ trovo **b** come $f(s)$ sulla bezier curve $\mathbf{p}_{1,0}$ $\mathbf{p}_{1,1}$ $\mathbf{p}_{1,2}$ $\mathbf{p}_{1,3}$
 - ⇒ trovo **c** come $f(s)$ sulla bezier curve $\mathbf{p}_{2,0}$ $\mathbf{p}_{2,1}$ $\mathbf{p}_{2,2}$ $\mathbf{p}_{2,3}$
 - ⇒ trovo **d** come $f(s)$ sulla bezier curve $\mathbf{p}_{3,0}$ $\mathbf{p}_{3,1}$ $\mathbf{p}_{3,2}$ $\mathbf{p}_{3,3}$
 - ⇒ trovo **p** come $f(t)$ sulla bezier curve **a,b,c,d**



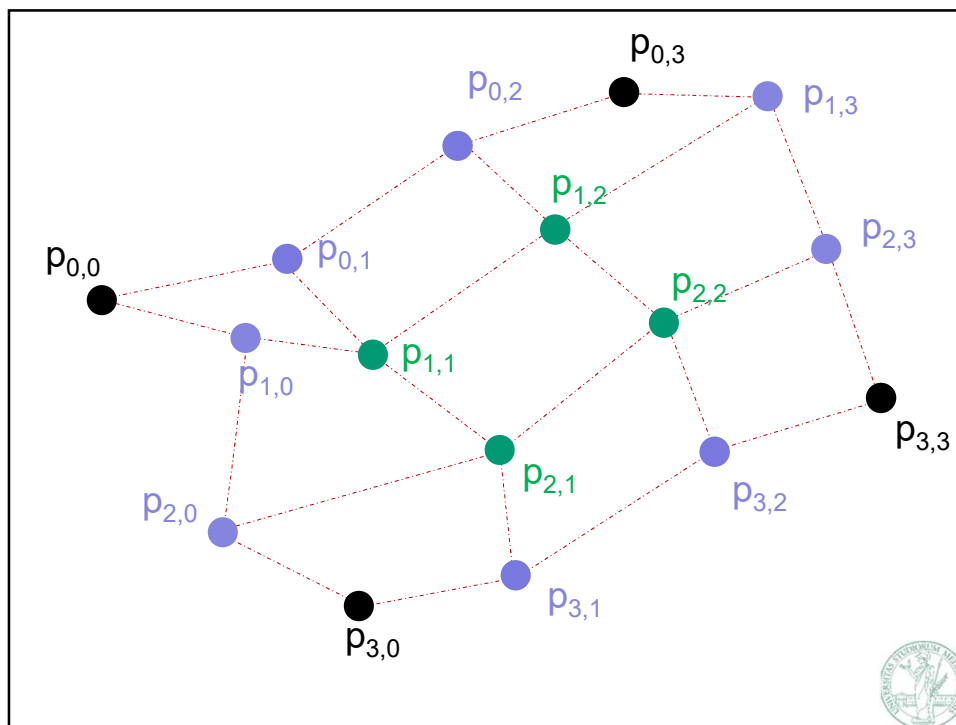
45

Bezier patches grado 3 «Bi-cubic Bézier patches»

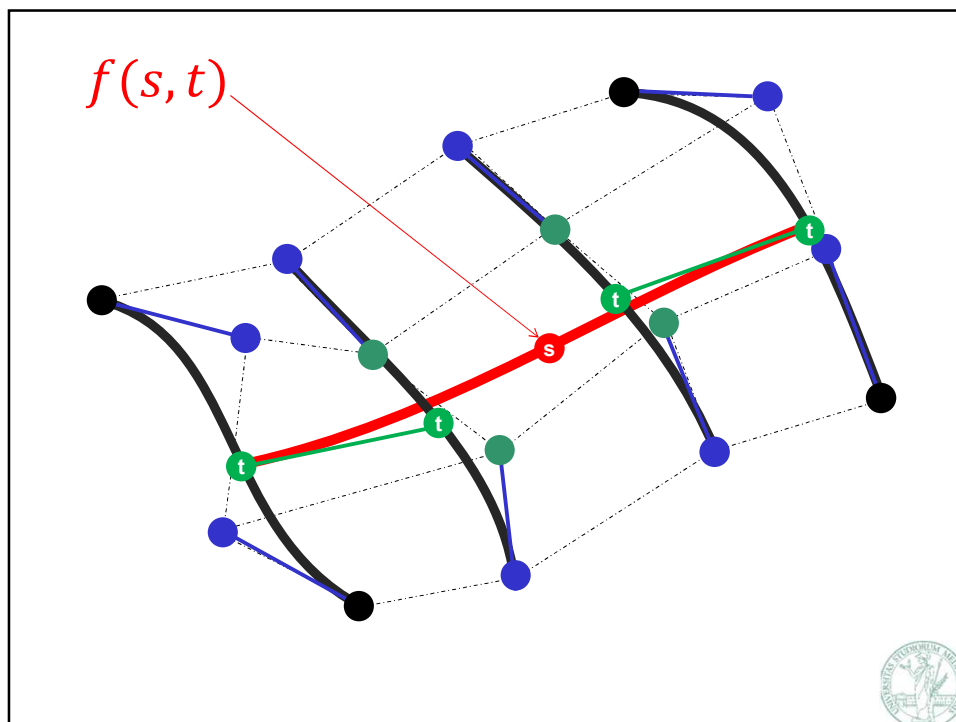
- ✓ $N = 3$
- ✓ Griglia di 4x4 vertici di controllo
 - ⇒ $\mathbf{p}_{0,0}$ $\mathbf{p}_{1,0}$ $\mathbf{p}_{2,0}$ $\mathbf{p}_{3,0}$
 - $\mathbf{p}_{0,1}$ $\mathbf{p}_{1,1}$ $\mathbf{p}_{2,1}$ $\mathbf{p}_{3,1}$
 - $\mathbf{p}_{0,2}$ $\mathbf{p}_{1,2}$ $\mathbf{p}_{2,2}$ $\mathbf{p}_{3,2}$
 - $\mathbf{p}_{0,3}$ $\mathbf{p}_{1,3}$ $\mathbf{p}_{2,3}$ $\mathbf{p}_{3,3}$
- ✓ Passa per i 4 angoli (è interpolante su loro)
 - $\mathbf{p}_{0,0}$ $\mathbf{p}_{0,3}$ $\mathbf{p}_{3,0}$ $\mathbf{p}_{3,3}$
- ✓ E' approssimante sugli altri punti
- ✓ E detto anche «Bi-Cubico» perché è cubico sia sulla s , che sulla t



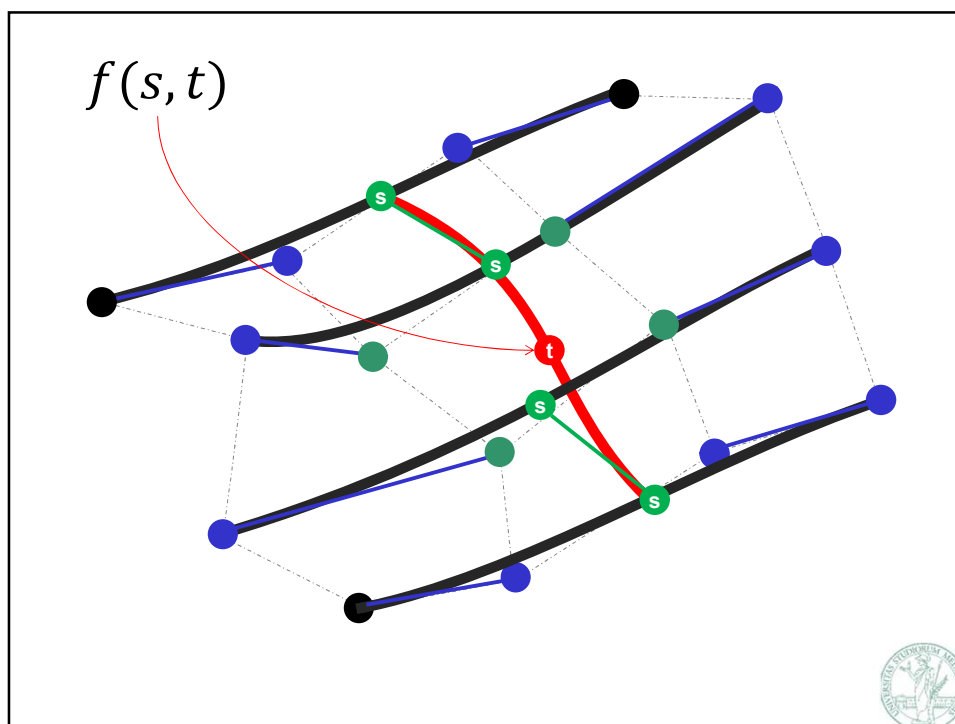
46



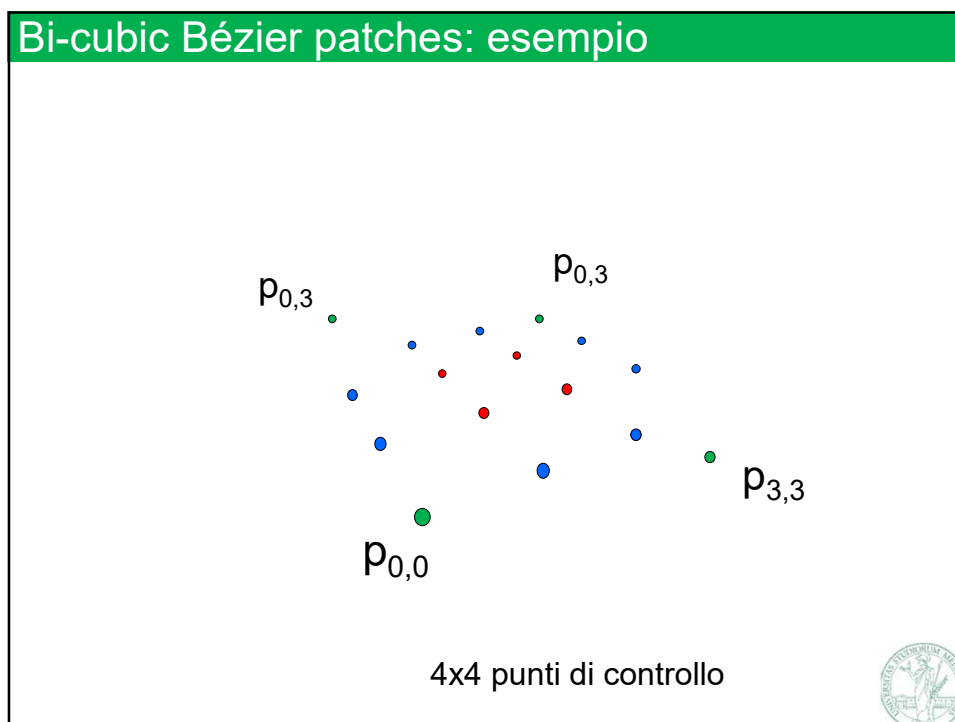
47



48

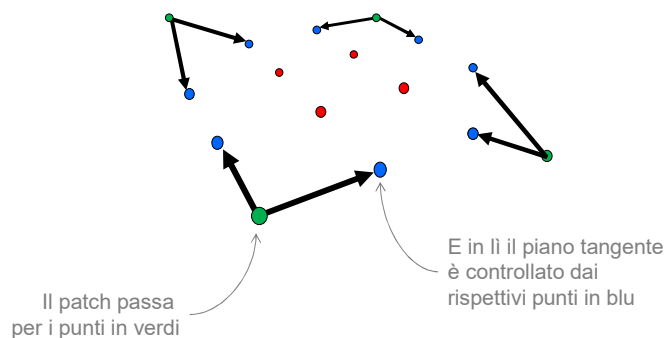


49



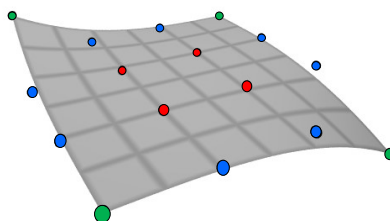
51

Bi-cubic Bézier patches



52

Bi-cubic Bézier patches

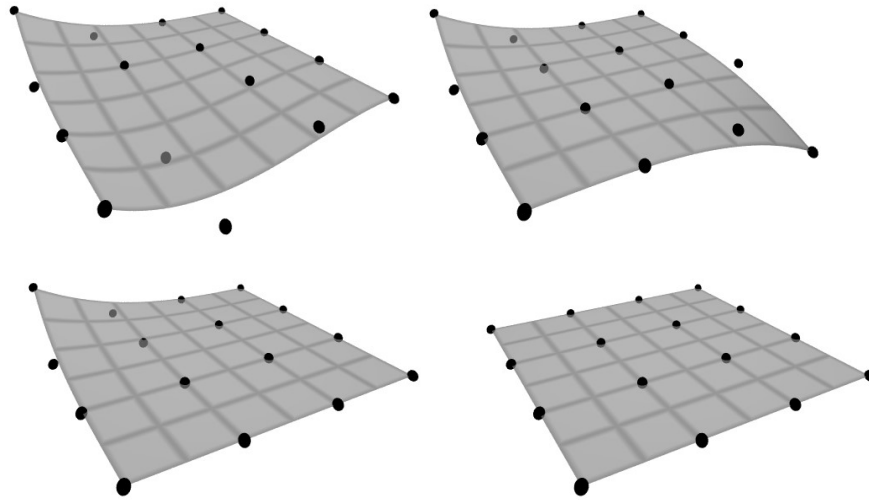


Patch di bezier



53

Cubic Bézier patches: esempi



Modificando i punti di controllo, modifico il patch

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



54

Superfici di bezier

- ✓ 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
 - ⇒ Inoltre, imporre la colinearità fra i punti di controllo interni garantisce che la superficie non abbia crease (discontinuità di normale), cioè che sia smooth (liscia) – se questo è desiderato!



55

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



56

Superfici parametriche: benefici

- ✓ Controllabili, intuitive da modellare, ottime per il design
 - ⇒ Queste superfici sono 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!



57

Superfici parametriche : benefici

- ✓ La superficie «nasce» già con un dominio parametrico e un 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, moi3D, Blender, Solidworks

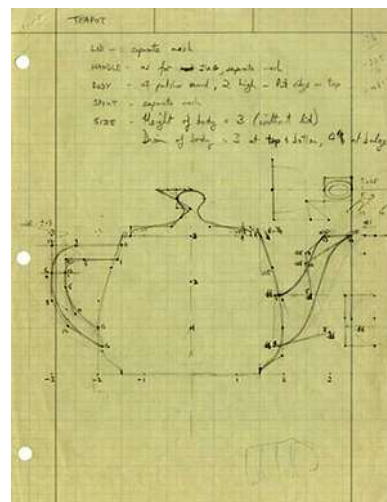


tool specializzati



58

Bézier patches: per il design

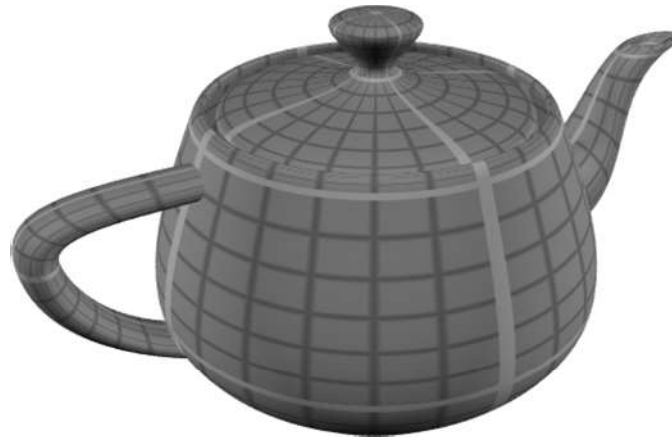


The Utah Teapot, by Martin Newell (1975)



59

Bézier patches: utili per il desing



The Utah Teapot, by Martin Newell (1975)



60

Processing su una superficie parametrica

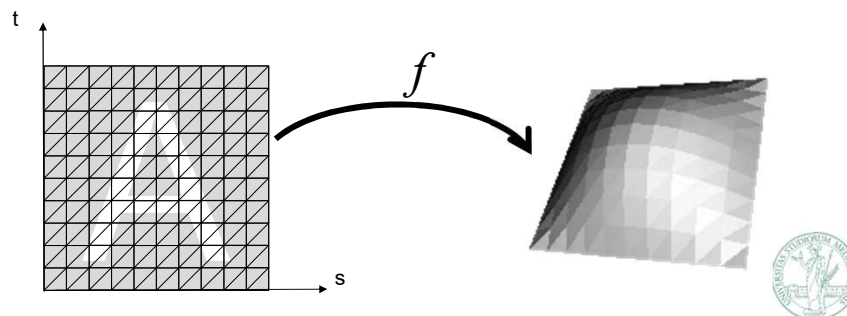
- ✓ In molti contesti, la superficie parametrica è l'input o l'output diretto di task di geometry processing, incluso:
 - ⇒ Semplificazione: riduzione del numero di patch
 - ⇒ Costruzione automatica (a partire da un'altra rappresentazione, quale una mesh). Task difficile.
- ✓ Per molti usi, compreso (spesso) 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 uso



61

Da superficie parametrica a mesh poligonale

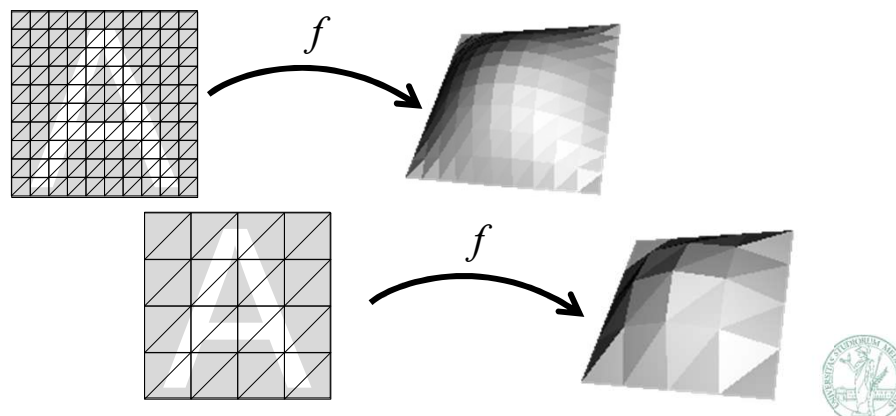
- ✓ Banale: campiono il dominio parametrico A su una griglia regolare (es.: 10×10)
 - ⇒ $(s,t) = (0.0,0.0), (0.1,0.0), \dots (0.0,0.1), (0.1,0.1) \dots, (1.0, 1.0)$
- ✓ Per ogni posizione campionata (s,t)
 - ⇒ creo un vertice della mesh nel punto $f(s,t)$
 - ⇒ do a ogni vertice della sua normale, ottenuta con le derivate di f
- ✓ Creo triangoli che connettono i vertici nel modo banale
- ✓ Risultato: mesh regolare!



63

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 poligonate



64