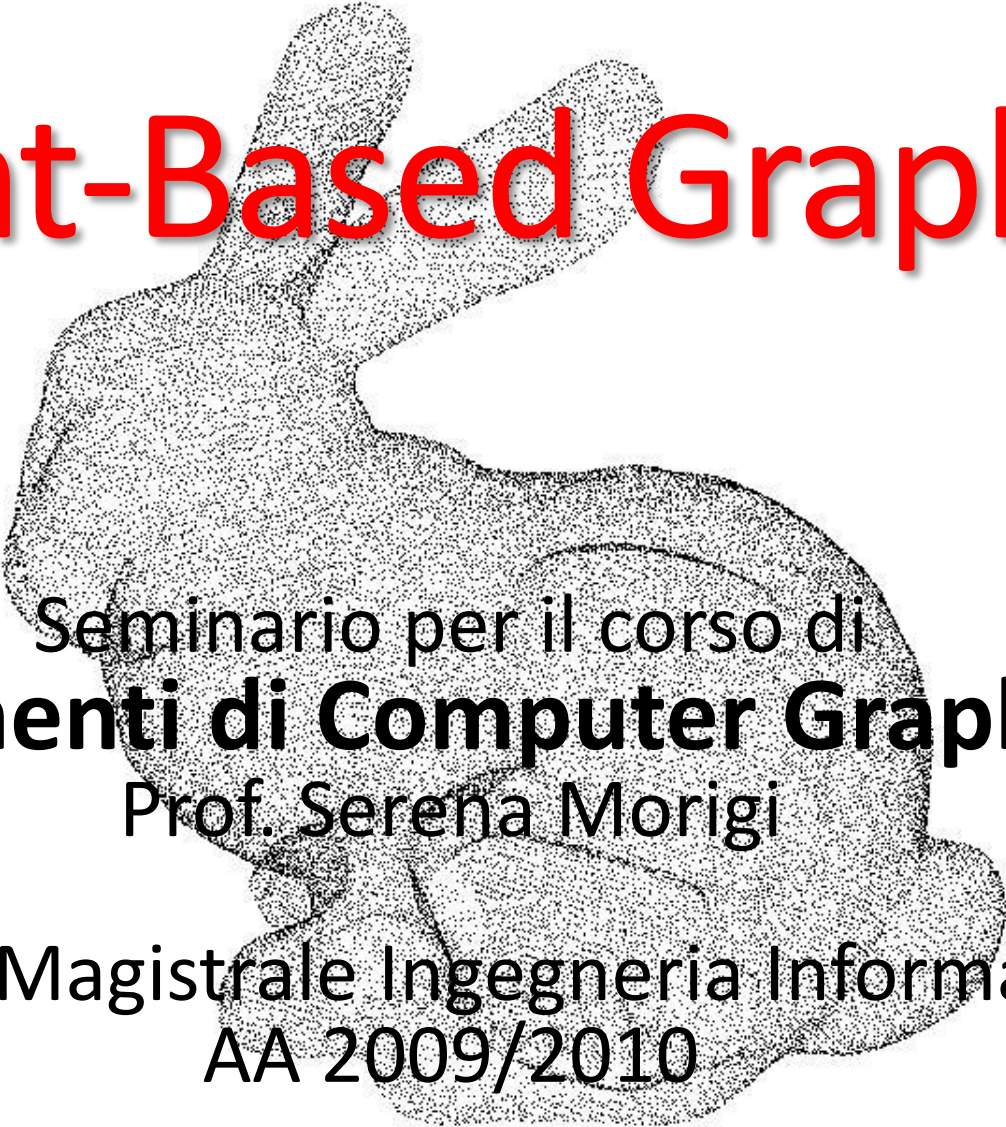


Point-Based Graphics



Seminario per il corso di
Fondamenti di Computer Graphics di
Prof. Serena Morigi

Laurea Magistrale Ingegneria Informatica
AA 2009/2010

Outline

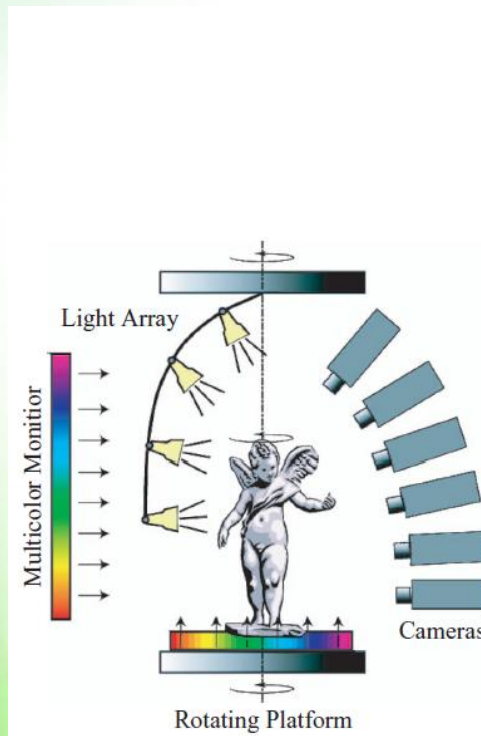
- Motivazioni
- Point-based Graphic features
- Rappresentazione di superfici
 - Surfels
 - Moving Least Squares
- Point-based rendering
 - Proiezione
 - Ray – tracing
 - Shading
- Editing



Motivazioni

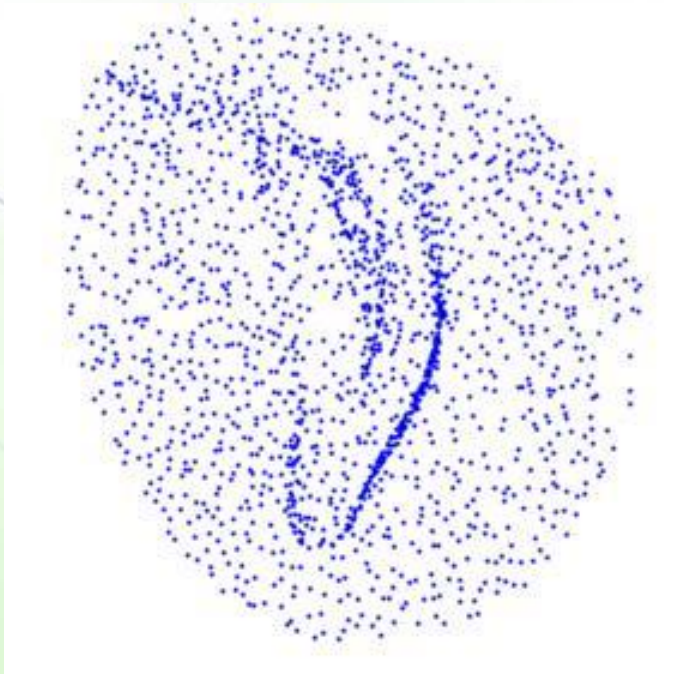
Alcuni sistemi di acquisizione, come gli scanner 3D producono un enorme quantità di dati:

- rumore
- casualità
- gestione



Point-based Graphics features

La grafica basata è congeniale per trattare le nuvole di punti (**point clouds**) senza dover usare algoritmi di triangolazione (mesh).



Il modello a disposizione è costituito da un insieme di punti nello spazio privi di qualsiasi informazione topologica:

$$\{p_i\} \in \mathbb{R}^3$$

Point-based Graphics features

- Implicito il concetto di LOD, come supporto ad un progressivo rendering di nuvole di punti



Downsampling



Algoritmi di decimazione

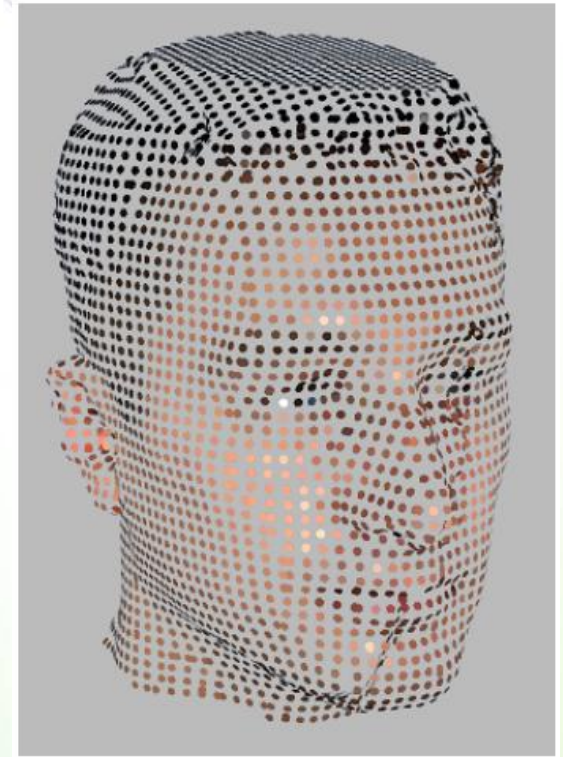
Upsampling



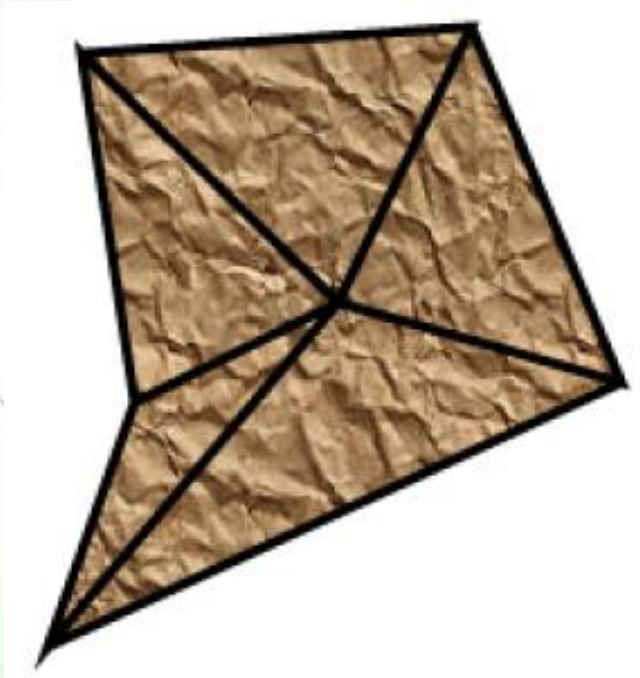
LOD crescenti

Point-based Graphic features

- Non vi è separazione tra superfici e volumi
- Il punto rappresenta l'elemento di rendering principale a cui si associano tutte le informazioni necessarie.
- Mancanza di connettività e topologia



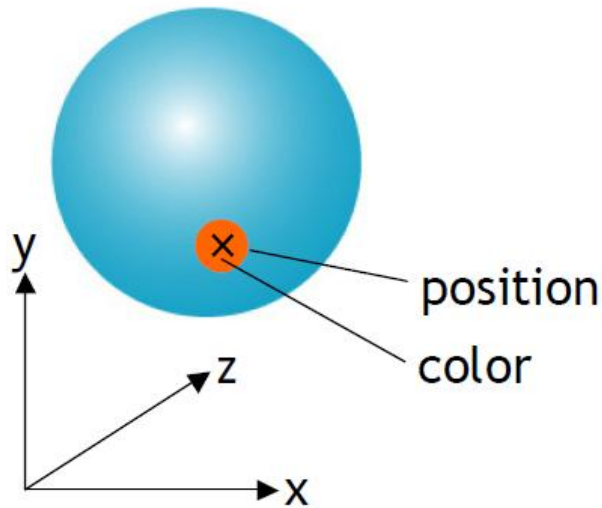
Point clouds vs triangles mesh



Mesh con texture



Punti e superfici



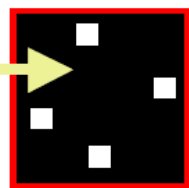
- Ogni punto corrisponde ad un elemento della superficie, descrivendola in un intorno locale
- Un punto ha dimensioni limitate e può essere

paragonato direttamente al pixel

- I punti non sono acquisiti con perfetta regolarità e continuità
 - holes

Rappresentazione delle superfici

holes between
the points



E' necessario interpolare (o approssimare) la superficie mediante punti.

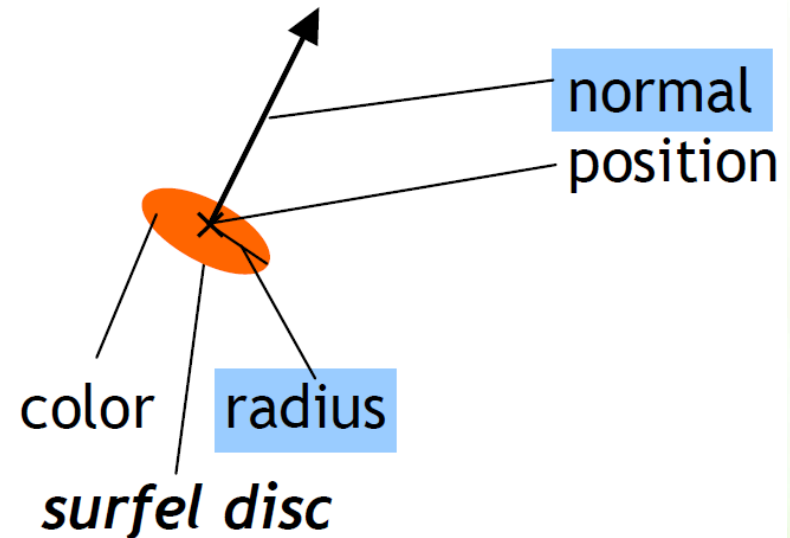
Ad ogni punto viene associata una certa superficie locale descritta da un polinomio g .

La superficie viene rappresentata da un insieme di polinomi associati ai distinti surfel.

Surfels

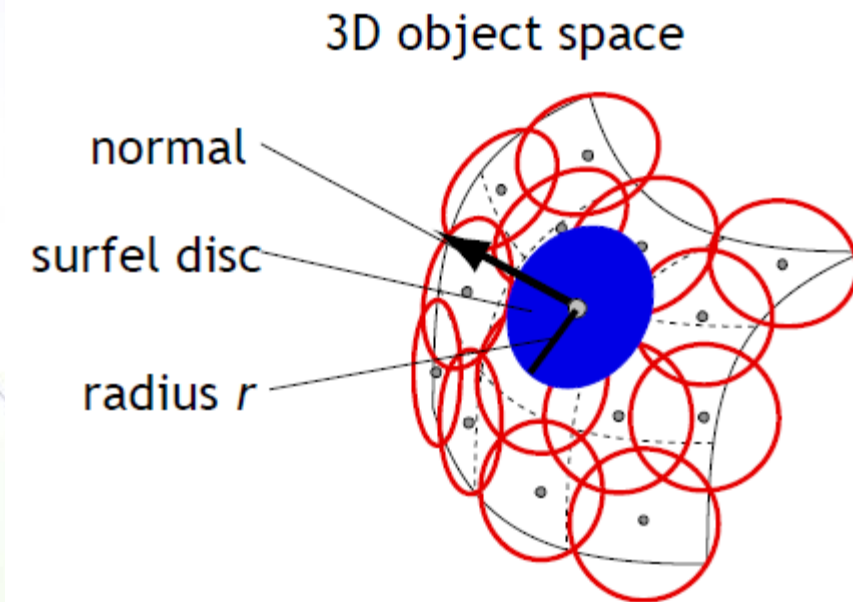
Surfel: elemento primitivo rappresentante localmente una superficie, incapsulando le proprietà:

- posizione
- colore
- normale
- estensione
- ...



Il Surfel è concepito come primitiva di rendering puntiforme e approssima localmente la superficie.

Surfels



Per ovviare al fenomeno degli “holes” si impiega un insieme di surfel di raggio r in modo tale da ricoprire la superficie dell’oggetto completamente

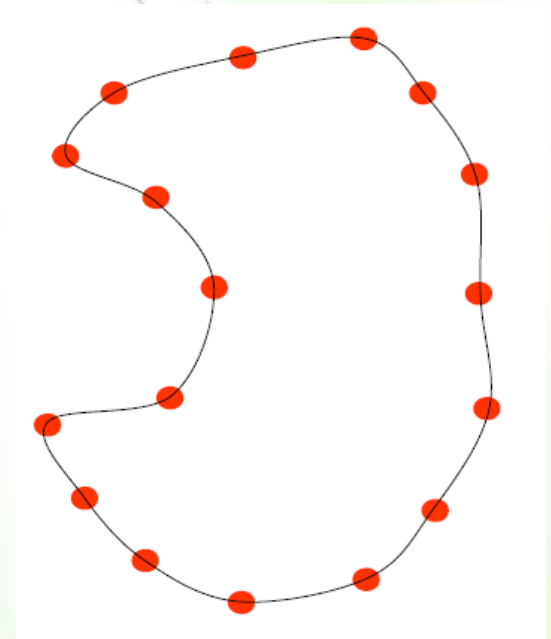
Ricostruzione della superficie

A seconda delle applicazioni potrebbe essere necessario disporre di una superficie basata su campioni puntiformi. Eventualmente che soddisfi particolari proprietà:

- Manifold
- Smooth
- Locale

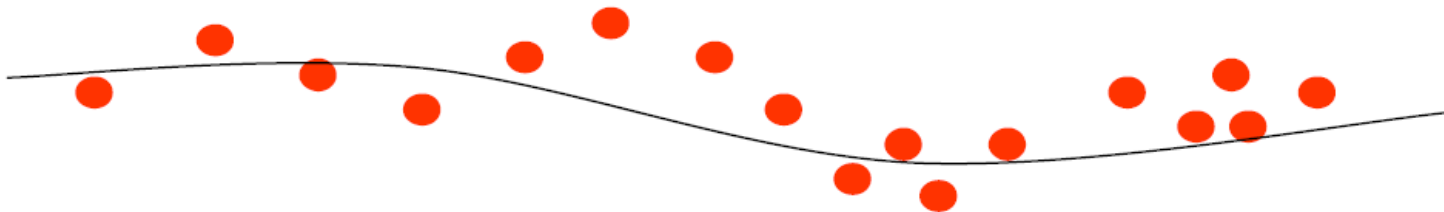
Diversi approcci:

- Approssimazione: LS, MLS, ...
- Interpolazione

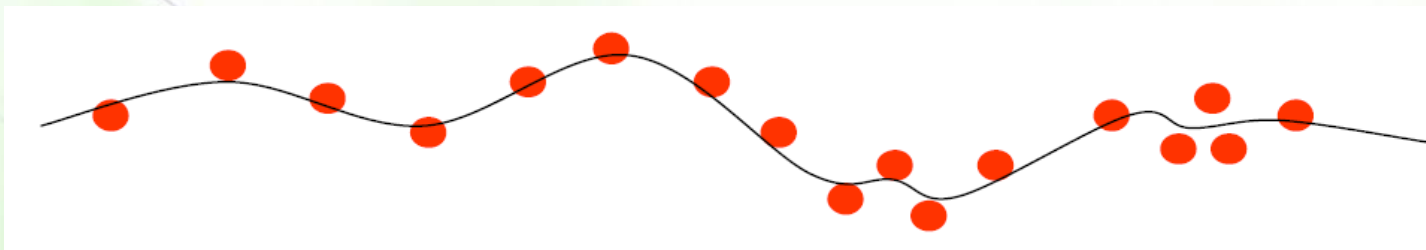


Approssimazione

- Campioni con rumore
- Approssimazione globale



- Approssimazione locale



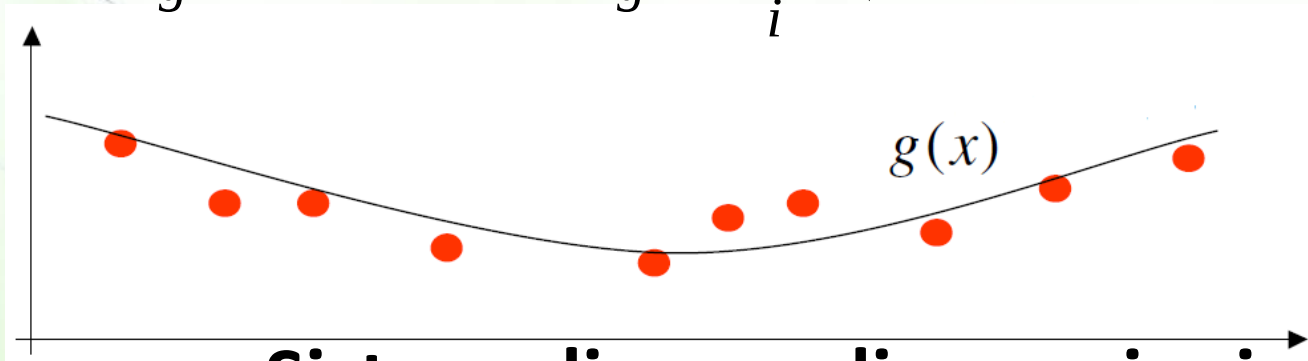
Least Squares

$\{p_i\}$ Punti campione

$g(x)$ Funzione approssimante

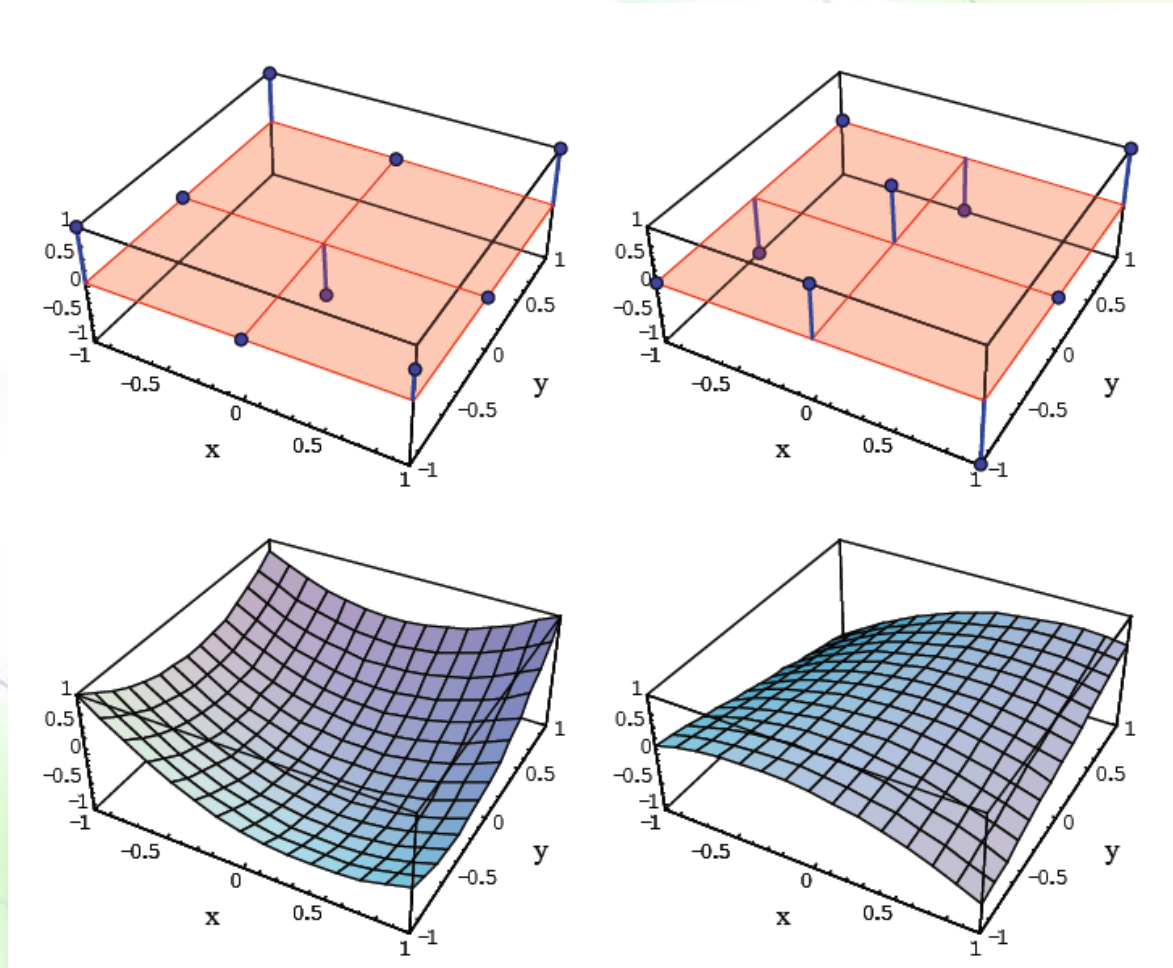
Obiettivo: minimizzare la somma dei quadrati delle distanze tra i p_i e $g(x)$

$$\min_g h(x) = \min_g \sum_i \left(p_{i_y} - g(p_{i_x}) \right)^2$$



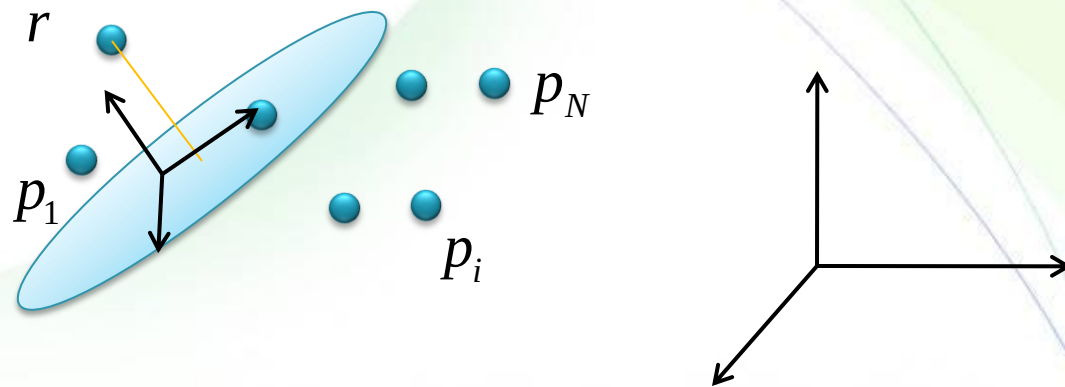
Sistema lineare di equazioni

LS: esempio



Controllo locale vs. controllo globale

Moving Least Squares



Siano $p_i \in \mathbb{R}^3$ con $i=1, \dots, N$ un insieme di punti nello spazio e sia r un punto su $\mathbb{R}^3$ vicino alla superficie S . Si vuole proiettare r su una superficie bidimensionale S_p che approssimi i p_i localmente

MLS: passo 1

Determinare il dominio di riferimento locale per il punto r .

Il piano locale è definito come

$$H = \{x \mid \langle n, x \rangle - D = 0, x \in \mathbb{R}^3\}, \quad n \in \mathbb{R}^3, \quad \|n\| = 1$$

Ed è calcolato in modo tale che minimizzi la somma pesata e locale di distanze al quadrato dei punti p_i dal piano H :

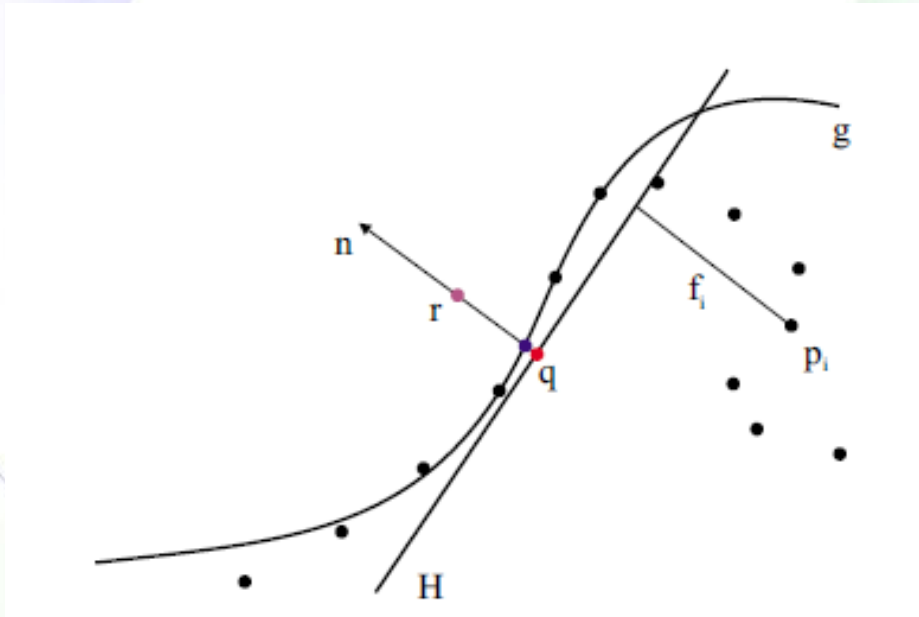
$$\min \sum_{i=1}^N (\langle n, x \rangle - D)^2 \theta(\|p_i - q\|)$$

Passo 1: Note

$$\min \sum_{i=1}^N (\langle n, x \rangle - D)^2 \theta(\|p_i - q\|)$$

- H funge da piano di regressione locale sul quale proiettare il punto r . Sia q tale proiezione
- $\theta(s)$ è una funzione tale per cui $\lim_{s \rightarrow \infty} \theta(s) = 0$
 - $\theta(s) = e^{-\frac{s^2}{h^2}}$
- Si ha a disposizione un **sistema di riferimento locale** con origine in q

MLS: passo 2



q_i : proiezione di p_i su H

f_i : distanza di p_i da H :

$$f_i = n \cdot (p_i - q)$$

Il piano H è utilizzato per calcolare un polinomio g , biavariato, approssimante la superficie in un intorno di r

Polinomio in un intorno locale

Problema del calcolare i coefficienti del polinomio approssimante g :

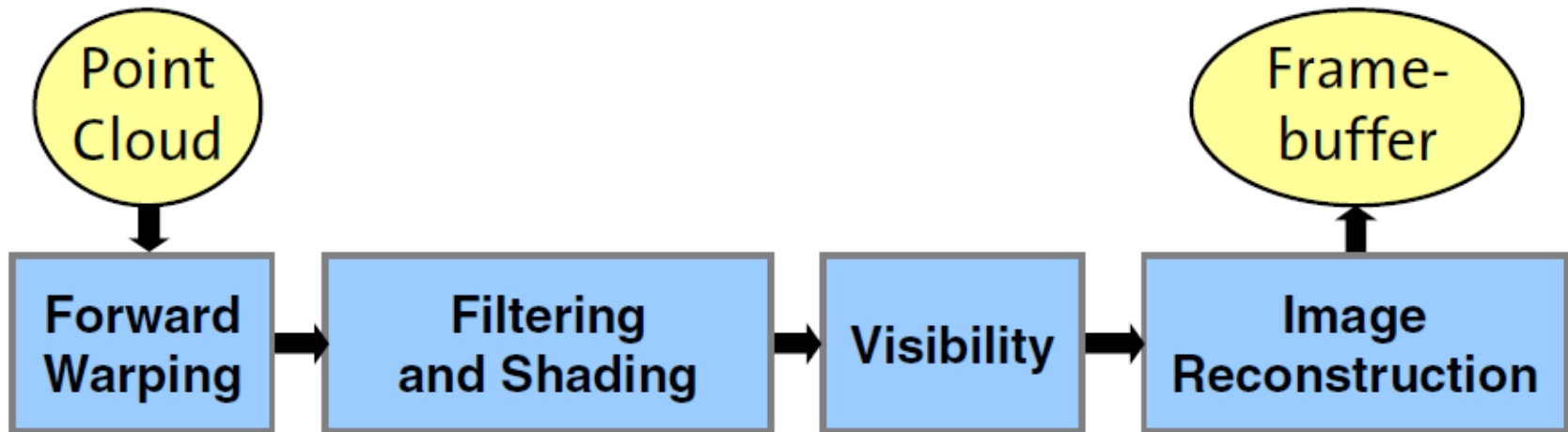
→ minimizzare l'errore dei minimi quadrati pesato:

$$\min \sum_{i=1}^N \left(g(x_i, y_i) - f_i \right)^2 \theta(\|p_i - q\|)$$

(x_i, y_i) è la rappresentazione di q_i nel sdr locale ad H .

La proiezione di r su S_p è definito dal valore del polinomio nell'origine: $P(r) = q + g(0,0)n$

Point rendering pipeline



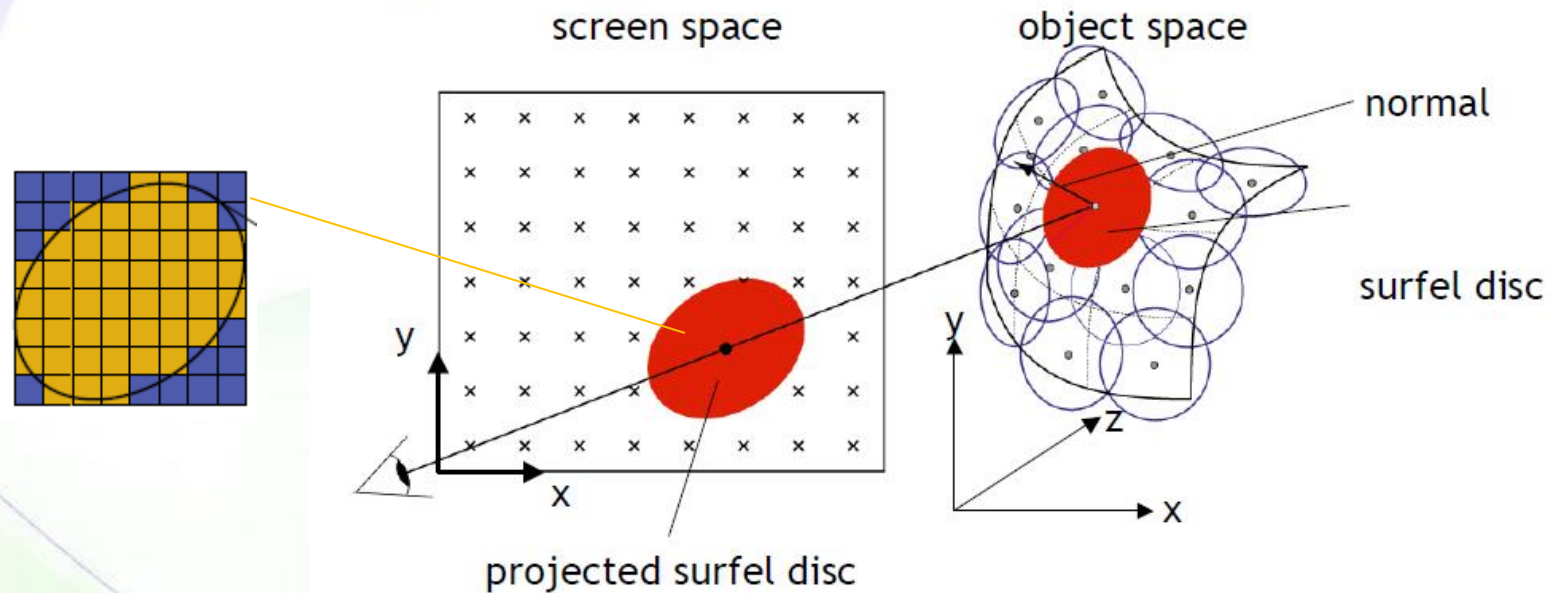
I surfel portano l'informazione necessaria lungo la pipeline (surfel stream)

Il framebuffer memorizza il colore RGB, alfa e Z
Proiezione di ogni punto indipendente

Point rendering pipeline

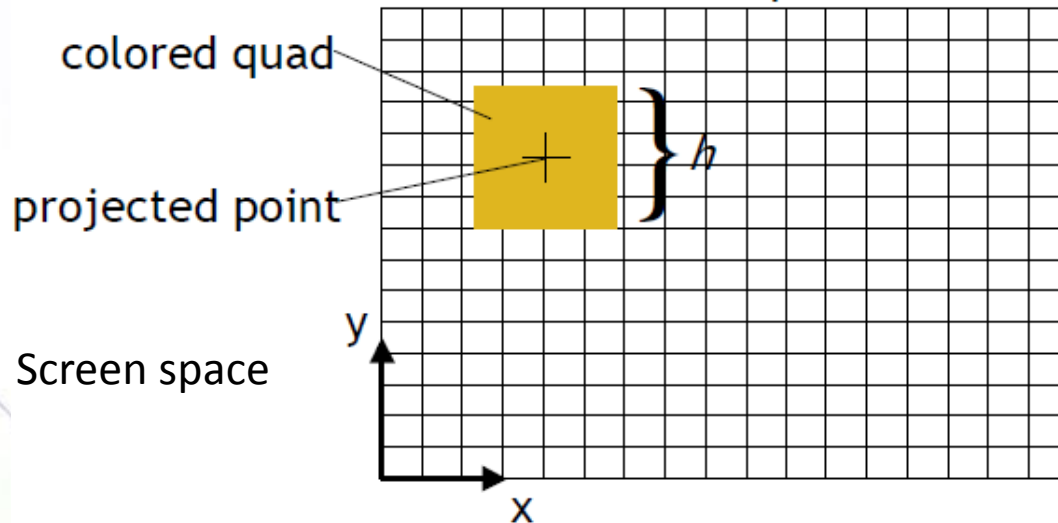
- Proiezione prospettica per ogni punto della nuvola, similmente alla proiezione di triangoli:
 - divisione prospettiva
- Shading per-point
 - Modelli convenzionali di shading (Phong, riflessione...)
- Visibilità e ricostruzione dell'immagine
 - i punti occlusi dal corrente punto di vista vengono scartati
- ricostruzione di una superficie continua a partire dai punti proiettati

Proiezione prospettica



I surfels vengono proiettati dall'object space nello screen space, divenendo ellissi che si adattano all'orientamento della superficie (normale alla superficie).

Quad Rendering primitive



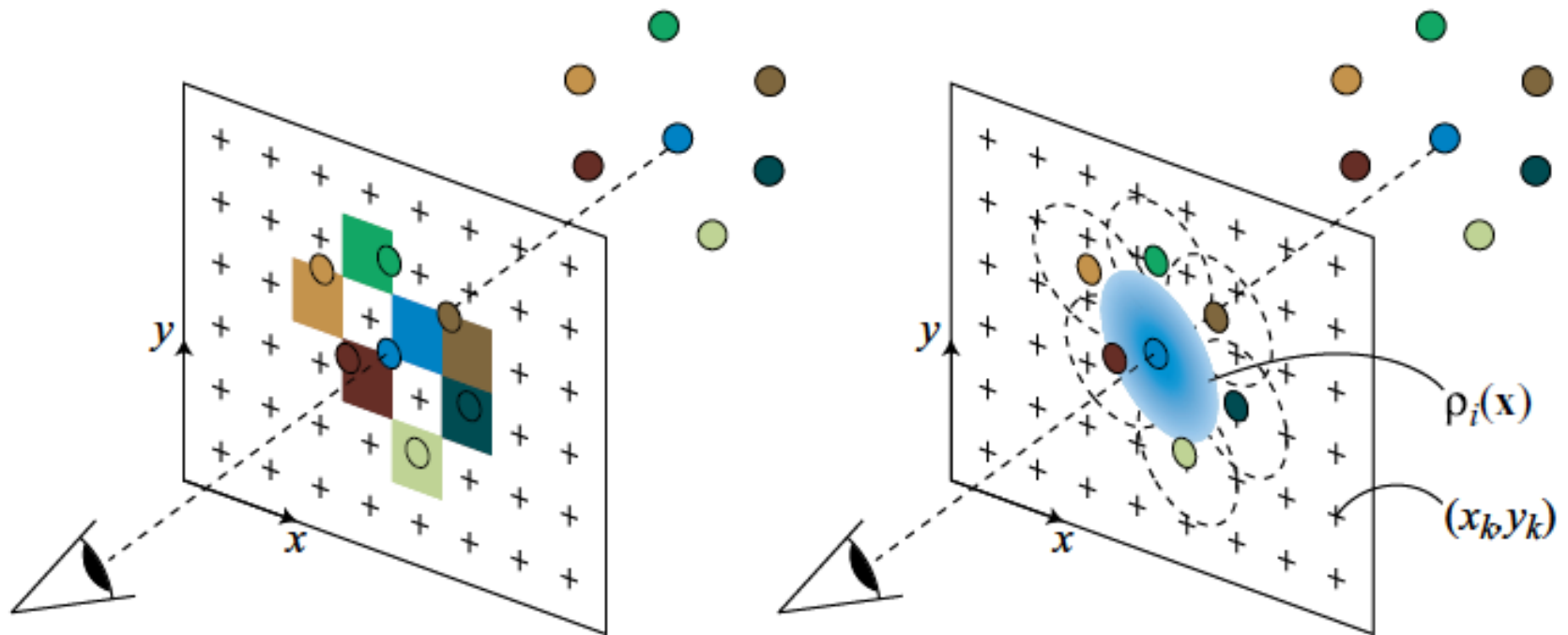
Alternativa per i
projected disc

Si disegna un quadrato colorato centrato nella proiezione del corrispondente punto nello screen space, di dimensione $h = 2 * r * s$, con s fattore di scala fornito dalla proiezione prospettiva e dalla trasformazione di vista

Confronto

Projected Disc	Quad primitive
Qualità dell'immagine superiore (adattamento alla normale della superficie)	Qualità dell'immagine più scarsa per il mancato adattamento all'orientamento
Non vi è supporto diretto da parte dell'hardware	Rendering efficiente
Alto costo computazionale	Supportato dagli acceleratori grafici (OpenGL: GL_POINTS)

Confronto



Point-based Ray tracing

- Ray-tracing convenzionale: da una collezione di surfel rappresentante la superficie, si intersecano i raggi di luce con le ellissi.
 - Discontinuità tra le diverse ellissi
- Ray-surface intersection algorithm

Ray-surface intersection algorithm

Sia $P = \{p_i\}$ un insieme di campioni di una superficie S .

Sia definito il vicinato di P come l'unione degli intorni locali sferici centrati in p_i

$$B = \bigcup_i B_i, \quad B_i = \{x \mid \|x - p_i\| < r_B\}$$

Si assume che B contenga quindi la superficie S , così come la sua

approssimazione $\tilde{S} = \{x \in B \mid f(x) = n(x)^T (x - a(x)) = 0\}$

Ray-surface intersection algorithm

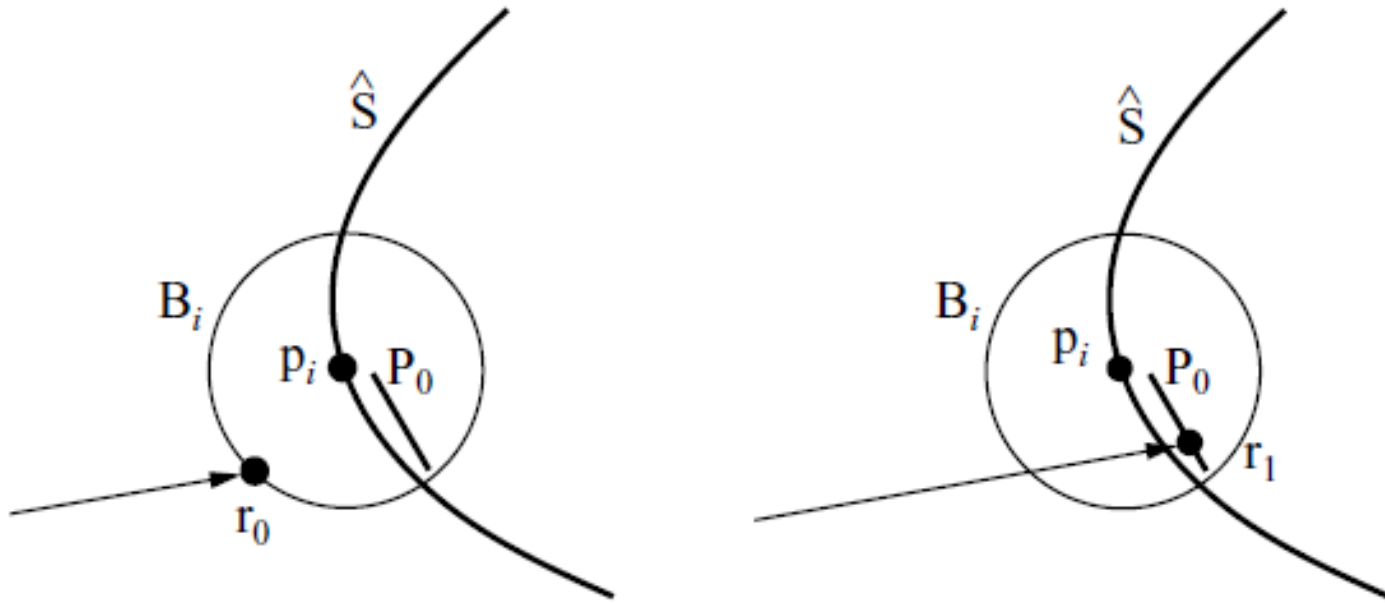
$$\tilde{S} = \left\{ x \in B \mid f(x) = n(x)^T (x - a(x)) = 0 \right\}$$

$a(x)$ Weighted average function

$n(x)$ Normal function

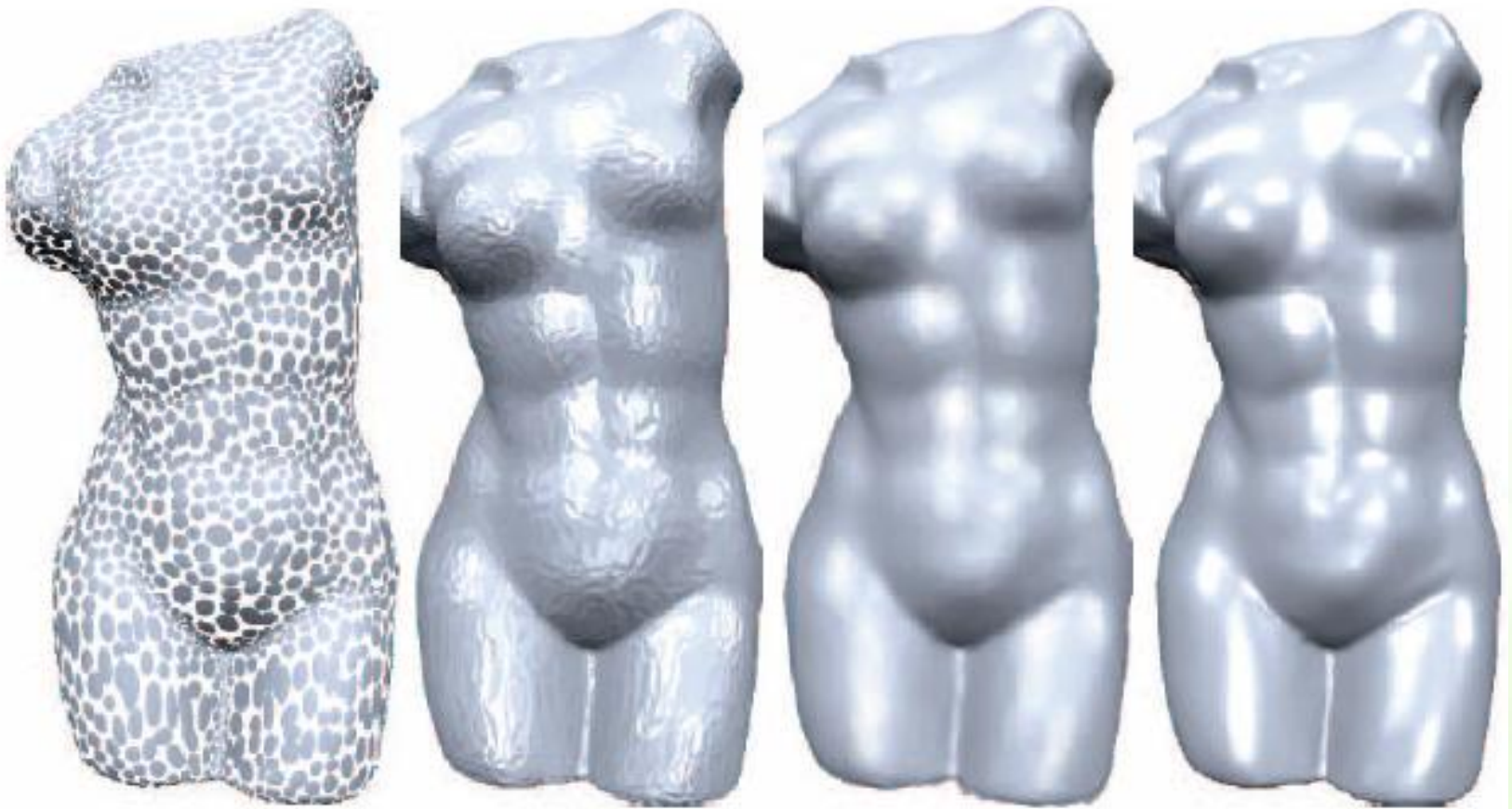
$$a(x) = \frac{\sum_{i=1}^N \theta \left(\frac{\|x - p_i\|}{h} \right) p_i}{\sum_{i=1}^N \theta \left(\frac{\|x - p_i\|}{h} \right)}$$

Intersezione



Iterativamente, fino alla convergenza, si trova un nuovo piano P aggiornando la funzione di media pesata e la funzione normale.

Shading



SPLAT, FLAT SHADING, GOURAUD SHADING, PHONG SHADING

Editing

Texture mapping

Brushing

Distorsione

Boolean operations

Riferimenti

- Markus Gross – Computer Graphics Laboratory: Point-based Graphics – State of the Art and Recent Advances
- Markus Gross & Hanspeter Pfister – Point-based Graphics, The Morgan Kaufmann Series in Computer Graphics
- Point Set Surfaces – Mark Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C.T. Silva