

# Tecniche avanzate



*Quello che avanza...*

- **Image-based rendering**
- **Quaternioni e rotazioni 3D**
- **Intersezioni**

# Image-based rendering

- Il problema è della grafica interattiva è la complessità dei modelli: sono laboriosi da costruire e pesanti da elaborare.
- Scopo: disaccoppiare complessità della scena da tempo di rendering e aumentare il realismo.
- Tecniche di rendering che usano immagini (+ informazioni geometriche): **image-based rendering**
- Impostori planari: Oggetti 3D vengono approssimati con piani su cui è mappata (come texture) una immagine pre-calcolata dell'oggetto.
- Non rendono il parallasse; per averlo bisogna associare all'immagine pre-calcolata (o acquisita) anche la profondità dei punti (depth map).
- La tecnica del 3D-warping consente di generare viste dell'oggetto con parallasse. L'effetto è 3D, ma la complessità dipende solo dal numero di pixel.
- Sposta nella fase off-line la maggior parte del carico computazionale, dipendente dalla complessità del modello (come le lightmap).

## 3D warping

- Le tecniche 2D non riproducono il parallasse; per farlo serve la profondità dei punti.
- Off-line: genero una immagine della scena (o dell'oggetto);
- Ad ogni pixel associo la sua profondità (come lo z-buffer, ma quest'ultimo contiene la la pseudo-depth, mentre serve la z nello spazio vista)
- On-line: per creare una immagine della scena (o dell'oggetto), dato il movimento della telecamera virtuale rispetto alla posizione di riferimento (quella da cui è presa l'immagine di riferimento) e la z posso calcolare il warping da applicare alla immagine di riferimento.



## Derivazione della formula del warping

Sia  $\mathbf{P} = \begin{bmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [\mathbf{A}|\mathbf{0}]$  la matrice di proiezione prospettica della telecamera e

supponiamo che lo scostamento rispetto alla posizione di riferimento sia data dalla

trasformazione rigida  $\mathbf{G} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$ .

Siano  $\mathbf{w} = [x, y, z, 1]^T$  e  $\mathbf{m}^T = [u, v, 1]$  le coordinate omogenee di un punto 3D e le coordinate della sua proiezione sulla telecamera, rispettivamente. Dunque

$$z\mathbf{m} = \mathbf{P}\mathbf{w} = [\mathbf{A}|\mathbf{0}]\mathbf{w}, \quad (1) \quad \text{e} \quad z'\mathbf{m}' = \mathbf{P}\mathbf{G}\mathbf{w} = [\mathbf{A}\mathbf{R}|\mathbf{A}\mathbf{t}]\mathbf{w} \quad (2)$$

dove  $z$  è la profondità di  $\mathbf{w}$  rispetto alla telecamera nella posizione di riferimento,  $z'$  è la profondità di  $\mathbf{w}$  rispetto alla telecamera dopo la rotazione e traslazione e  $\mathbf{m}'$  sono le coordinate omogenee del punto nella nuova immagine.

Dalla (2) otteniamo:

$$z'\mathbf{m}' = [\mathbf{AR}|\mathbf{At}] \left( \begin{pmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{pmatrix} = \mathbf{AR} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{At}, \quad (3)$$

e dalla (1) otteniamo:

$$z\mathbf{A}^{-1}\mathbf{m} = [\mathbf{I}|\mathbf{0}] \mathbf{w} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (4)$$

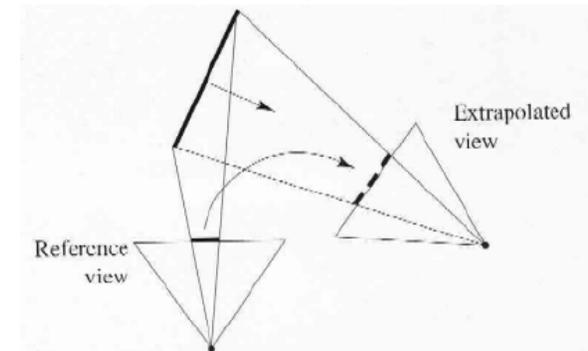
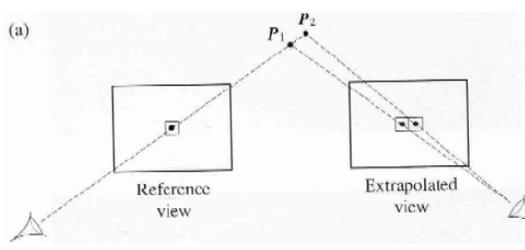
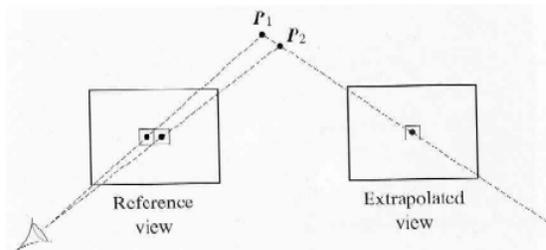
Sostituendo quest'ultima nella (3) si ottiene

$$\boxed{z'\mathbf{m}' = z\mathbf{ARA}^{-1}\mathbf{m} + \mathbf{At}.} \quad (5)$$

Questa è una mappa  $\mathbf{m} \rightarrow \mathbf{m}'$ . Il risultato del calcolo della parte destra di (5) fornisce una tripla, dividendo per la terza componente si ottiene  $\mathbf{m}'$ .

## Problemi relativi all'image warping:

- Image folding (ripiegamento): accade quando più di un pixel della immagine di riferimento si mappa su un pixel della immagine estrapolata. Quale scegliere?
- Buchi da occlusioni: accade quando è disponibile nella immagine di riferimento l'informazione per riempire un pixel della immagine estrapolata.
- Buchi da ingrandimento: accade quando l'area proiettata di una superficie aumenta nella vista estrapolata.



© A. Watt

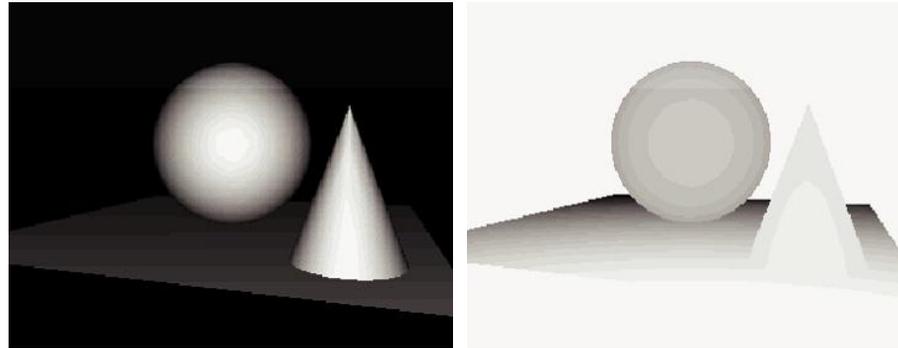


Figura 1: Semplice scena e depth map corrispondente.

© A. Watt

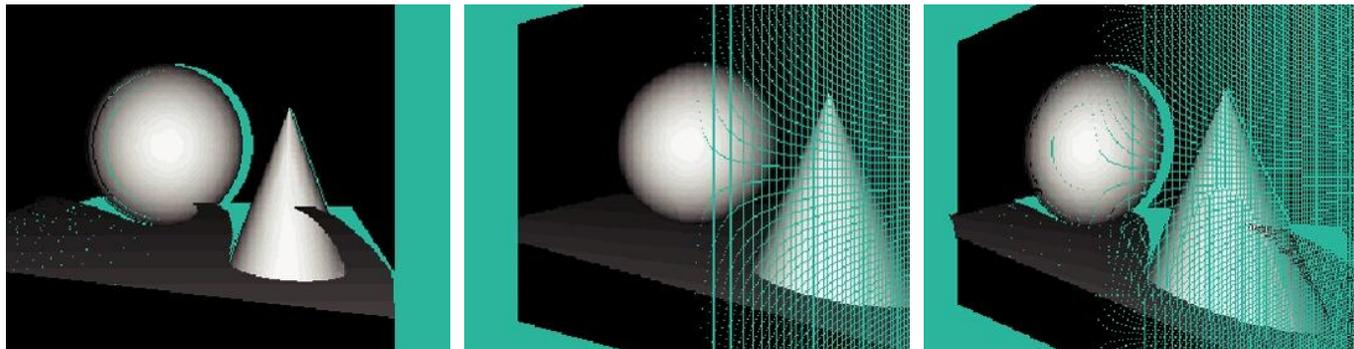


Figura 2: Artefatti dovuti a traslazione (occlusioni) e rotazione (ingrandimento) ed entrambe.

© A. Watt

## McMillan's ordering algorithm

I buchi da ripiegamento (folding) si possono evitare seguendo nel warping un opportuno ordine di valutazione, che garantisce che i pixel più vicini all'osservatore vengono considerati per ultimi (e quindi sovrascrivono gli altri).

Per esempio, se la camera trasla a sinistra i pixel devono venire processati da destra a sinistra.

In generale l'ordine dipende dalla posizione del nuovo punto di vista nella immagine di riferimento. McMillan ha enumerato tutti i possibili casi, che sono nove.

## Layered Depth Images

Le **Layered Depth Images** risolvono il problema dei buchi da occlusione associando a ciascun pixel molte profondità, tutte quelle di ciascuna superficie incontrata da un raggio (ideale) uscente dal pixel.

Le profondità sono discretizzate, dando origine così ad una struttura a strati (layer).

Il rendering si fa back to front un layer alla volta usando il warp-ordering di Mc Millan.

Questo consente di non avere buchi nelle zone occluse, ma occupa più memoria.

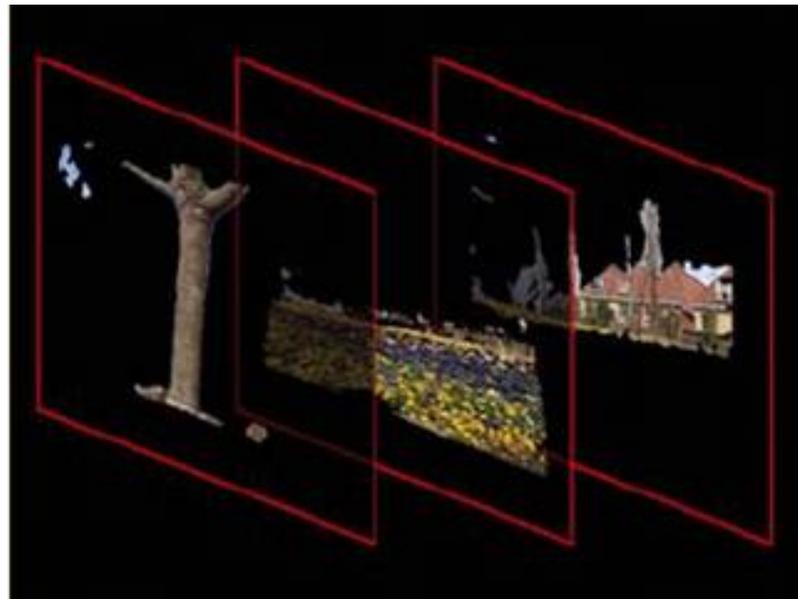
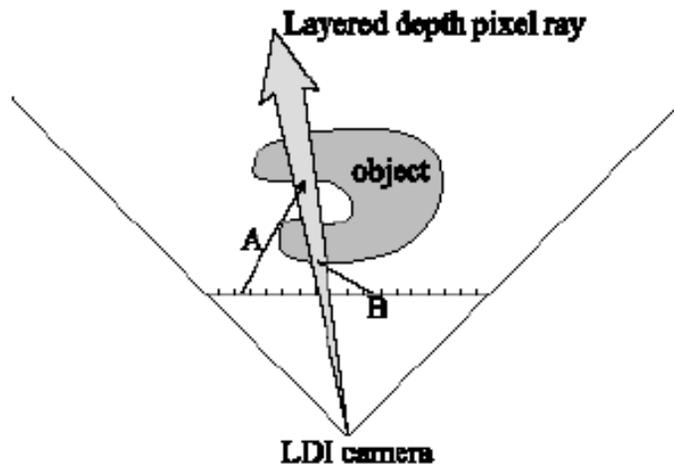


Figura 3: Le profondità A e B sono assegnate allo stesso pixel.

© Shade et al.

## Splatting

I buchi da ingrandimento si risolvono con interpolazione oppure disegnando pixel più grossi (splatting).

La dimensione delo pixel (splat size) dipende dalla profondità del punto e dal moto (virtuale) della telecamera.

## Interpolazione di viste

Un caso particolare di warping è quello della interpolazione.

Si parte da due immagini + informazione di profondità e si generano viste intermedie.

In questo caso si evitano naturalmente i buchi (anche quelli da occlusione).

## Quaternioni e rotazioni 3D

- Nell'animazione la decisione di come rappresentare le rotazioni nello spazio 3D è cruciale.
- In applicazioni dove esiste una direzione "up" privilegiata (es. orientazione di una telecamera) si usa la rappresentazione RPY.
- In applicazioni dove non esiste una direzione "up" privilegiata (es. orientazione di una astronave nello spazio) o dove sia necessario interpolare rotazioni, i quaternioni forniscono una buona rappresentazione.

- **Gimbal Lock**

- Si consideri per esempio la matrice  $R_y(\alpha)R_x(-\pi/2)R_z(\beta)$ : cambiare  $\alpha$  o  $\beta$  sortisce lo stesso effetto. In effetti dipende da un solo parametro, dunque un grado di libertà è perso.
- una rappresentazione con asse di rotazione e angolo (come i quaternioni) evita questo inconveniente.

- **Interpolazione**

- se si interpolasse tra la rotazione  $\Omega_1 = (0, 0, 0)$  e la  $\Omega_2 = (90, -90, -90)$  si otterrebbe a metà strada la  $\Omega_{1/2} = (45, -45, -45)$  che non è affatto intuitivamente a metà tra  $\Omega_1$  e  $\Omega_2$  (sempre a causa del problema delineato sopra).
- la tecnica SLERP basata sui quaternioni evita questo effetto

- **Indipendenza dal riferimento;**

- la rappresentazione delle rotazioni con angoli di Eulero non è indipendente dal sistema di riferimento (es. se cambio nome agli assi la rotazione cambia).
- una rappresentazione con asse di rotazione e angolo (come i quaternioni) è intrinseca (dipende da quantità geometriche).

- Un **quaternione** è una coppia ordinata formata da una parte **scalare** ed una **vettoriale**

$$q = (s, \mathbf{v})$$

- Può essere visto come una generalizzazione dei numeri complessi;  $s$  è la parte reale e le componenti di  $\mathbf{v}$  sono tre parti complesse

$$q = s + ia + jb + kc$$

dove  $(a, b, c)$  sono le componenti del vettore  $\mathbf{v}$  ed i **simboli**  $i, j, k$  soddisfano le seguenti proprietà, diretta generalizzazione delle proprietà dei numeri complessi

$$i^2 + j^2 + k^2 = -1$$

$$ij = -ji = k$$

- si dimostra che dalle proprietà precedenti valgono le seguenti relazioni

$$jk = -kj = i$$

$$ki = -ik = j$$

- Sui quaternioni sono definite, in modo naturale, le operazioni di **prodotto per uno scalare**, **somma** di due quaternioni e **prodotto** di due quaternioni; per le prime due si ha

$$\alpha q = \alpha s + i\alpha a + j\alpha b + k\alpha c$$

$$q_1 + q_2 = s_1 + s_2 + i(a_1 + a_2) + j(b_1 + b_2) + k(c_1 + c_2)$$

o, usando la notazione vettoriale

$$\alpha q = (\alpha s, \alpha \mathbf{v})$$

$$q_1 + q_2 = (s_1 + s_2, \mathbf{v}_1 + \mathbf{v}_2)$$

- Usando le proprietà viste sopra per gli elementi  $i$ ,  $j$  e  $k$ , si può derivare la seguente formula per il prodotto o di due quaternioni  $q_1$  e  $q_2$ :

$$q_1 q_2 = (s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2)$$

- Se due quaternioni con parte scalare nulla sono moltiplicati, la parte scalare del prodotto il prodotto scalare della parte vettoriale cambiato di segno, mentre la parte vettoriale del prodotto il prodotto vettoriale.
- il prodotto di quaternioni è associativo ma non commutativo.
- Si definisce anche il modulo di un quaternione

$$|q|^2 = s^2 + \mathbf{v} \cdot \mathbf{v}$$

ed un inverso

$$q^{-1} = \frac{1}{|q|^2} (s, -\mathbf{v})$$

(infatti si può verificare che  $qq^{-1} = q^{-1}q = (1, 0)$ )

- Un quaternione si dice **unitario** se ha modulo pari a 1
- I quaternioni unitari sono in corrispondenza biunivoca con i punti della superficie sferica in  $\mathbb{R}^4$ .

Come si usano i quaternioni per rappresentare rotazioni in uno spazio 3D?

- Data una rotazione di un angolo  $\theta$  intorno ad un asse passante per l'origine del riferimento, identificato da un versore  $\mathbf{u}$ , si definisce il seguente quaternione unitario

$$q = (s, \mathbf{v}) = \left( \cos \frac{\theta}{2}, \mathbf{u} \sin \frac{\theta}{2} \right)$$

- Dato un punto  $P$ , che vogliamo ruotare, di coordinate  $(p_1, p_2, p_3, 1)$  (ovvero  $(\mathbf{p}, 1)$ ), definiamo il seguente quaternione che lo rappresenta

$$p = (0, \mathbf{p})$$

- Definiamo quindi un **operatore rotazione** nel seguente modo

$$p \rightarrow qpq^{-1}$$

- Si dimostra che la parte scalare di  $p' = qpq^{-1}$  è nulla (quindi è un punto) e che la parte vettoriale è data da

$$\mathbf{p}' = s^2 + \mathbf{v}(\mathbf{p} \cdot \mathbf{v}) + 2s(\mathbf{v} \times \mathbf{p}) + \mathbf{v} \times (\mathbf{v} \times \mathbf{p})$$

dove  $s$  e  $\mathbf{v}$  sono quelli definiti sopra.

- Si ottiene così un nuovo punto  $P'$  di coordinate  $(\mathbf{p}', 1)$  e si dimostra che tale punto è esattamente il punto ottenuto ruotando  $P$  intorno a  $\mathbf{u}$  di un angolo  $\theta$
- Da notare che la rotazione così ottenuta è attorno ad un asse che passa per l'origine del riferimento; per una rotazione qualsiasi bisogna prima traslare in modo da portare l'asse a passare per l'origine, effettuare la rotazione e quindi traslare indietro nella posizione originale, come già visto prima

# Interpolazione di rotazioni

- Date due orientazioni di un oggetto (rappresentate da due rotazioni), si vuole generare una traiettoria nello spazio delle rotazioni che unisca le due orientazioni.
- l'uso degli angoli di Eulero non è l'ideale (per i due problemi appena visti), usiamo i quaternioni
- **Slerp: spherical linear interpolation**
  - Siano dati due quaternioni unitari  $p_1 = (s_1, \mathbf{v}_1)$  e  $p_2 = (s_2, \mathbf{v}_2)$
  - Una buona interpolazione tra i due quaternioni si ottiene con la seguente formula:

$$\text{slerp}(p_1, p_2, t) = \frac{p_1 \sin((1-t)\theta) + p_2 \sin(t\theta)}{\sin \theta} \quad t \in [0, 1]$$

dove

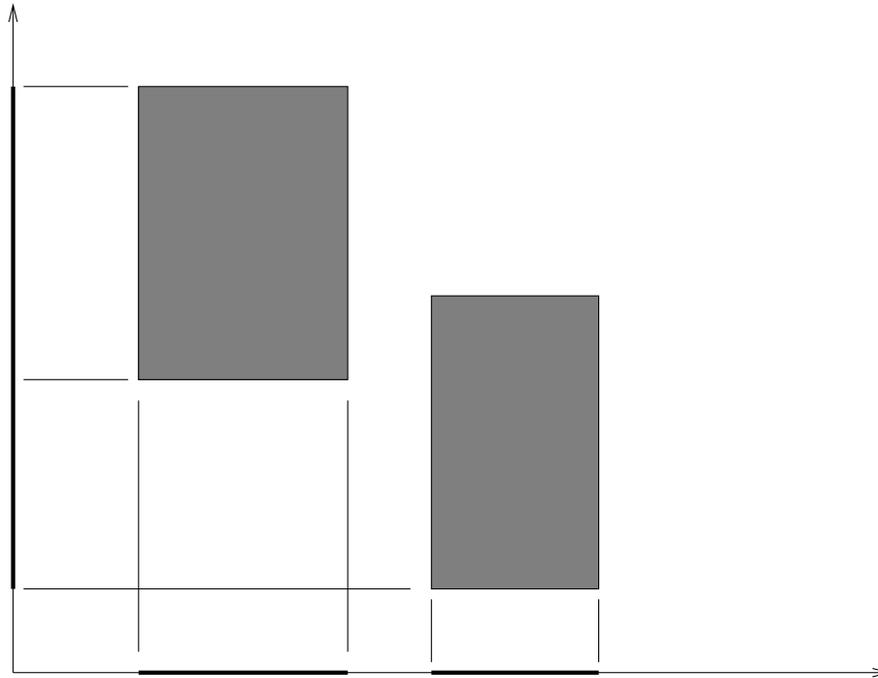
$$\cos \theta = s_1 s_2 + \mathbf{v}_1 \cdot \mathbf{v}_2$$

- questo corrisponde ad interpolare due punti sulla sfera 4D lungo un arco di cerchio massimo.

# Intersezioni

## Intersezione di AABB

- Determinare se due AABB si intersecano è molto più semplice che fare il test di intersezione per poliedri.
- L'algoritmo che vedremo è simile (ma molto più semplice) alla tecnica di sweep per intersezione di segmenti. L'idea base è che due AABB si intersecano se e solo se le loro proiezioni su ciascun asse coordinato si intersecano.



- Il problema si scompone quindi in tre problemi unidimensionali: trovare se un insieme di intervalli sulla retta si intersecano.
- Si ordinano gli estremi degli intervalli in una lista, mantenendo l'informazione circa il tipo di estremo (inferiore o superiore) e l'identità dell'intervallo.
- Si scandisce la lista:
  - ogni volta che si incontra un estremo inferiore si pone l'intervallo corrispondente in una lista di intervalli attivi e si stabilisce che gli intervalli attivi si sovrappongono.
  - ogni volta che si incontra un estremo superiore si toglie l'intervallo dalla lista.

# Intersezione Poliedro-Poliedro

- Si vuole rilevare se due poliedri convessi si intersecano.
- controllo se uno dei vertici di uno è contenuto nell'altro (deve stare nel semipiano negativo di tutte le facce).
- controllo se i lati di uno intersecano le facce dell'altro
- controllo se sono due poliedri identici e coincidenti.
- Il punto 2 si fa nel seguente modo:
  - test di intersezione segmento (lato) con piano contenente la faccia (gli estremi del segmento devono avere distanza segnata discorde rispetto al piano)
  - determino la posizione del punto di intersezione rispetto alla faccia (poligono)

Teaser: di più sulle intersezioni nel corso di **Geometria Computazionale**.