

# Grafica computazionale

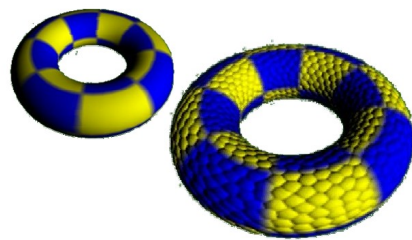
## Lezione 4

(slide parzialmente basate su  
Computer Graphics - MIT OpenCourseware  
Grafica Computazionale - Massimiliano Corsini – Università di Siena)

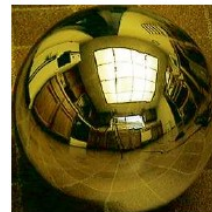
1

### Textures Riviste

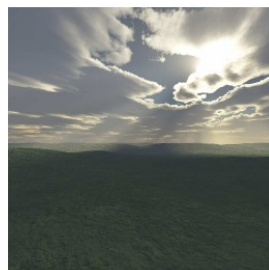
- Abbiamo visto Texture come tecnica per modulare proprietà spaziali
  - In particolare: Texture Mapping modula il colore di una superficie
- Le texture possono essere usate anche per...
  - Alterate la Geometria o lo shading
    - Bump/Normal mapping
    - Displacement mapping



- Emulare effetti non locali
  - ShadowMaps
  - Environment Mapping



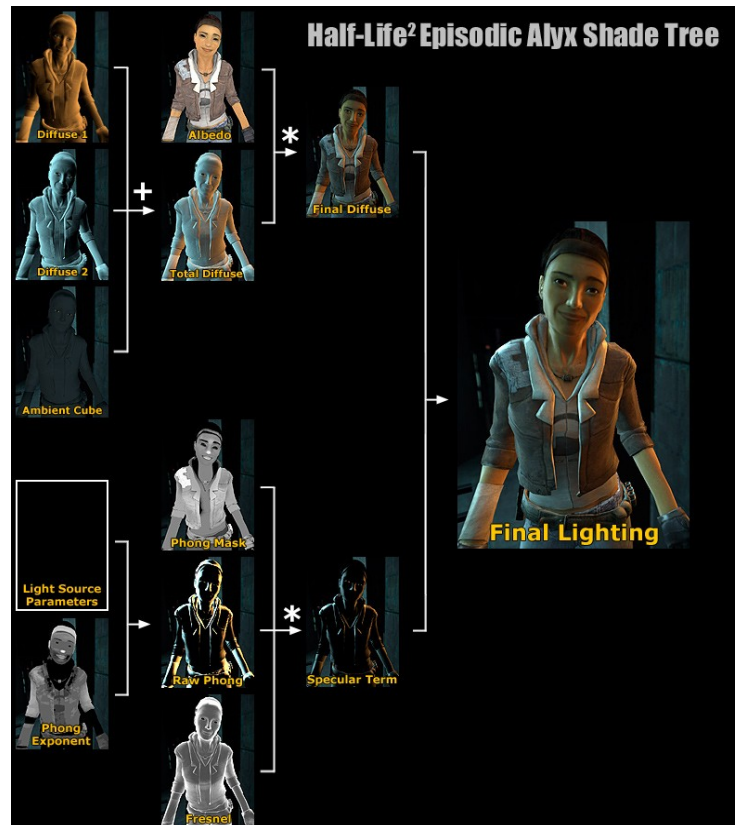
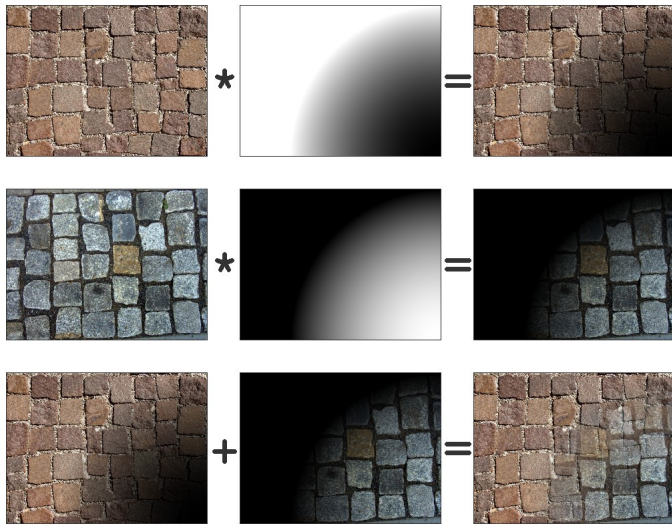
- Semplificare geometrie complesse
  - Billboards/Impostors
  - Skybox



2

# Multitexturing/Texture Splatting

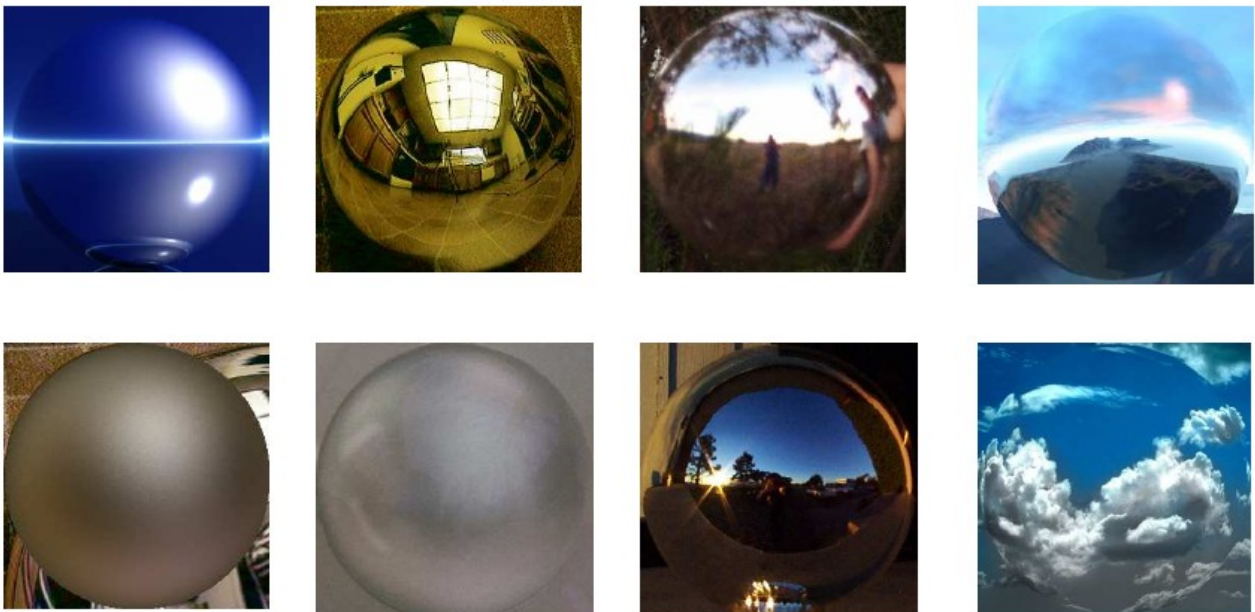
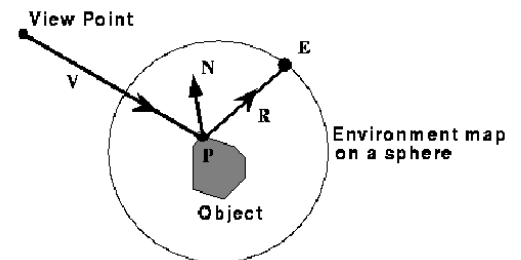
- Per ottenere effetti complessi più texture vengono combinate in un singolo passo (texture splatting)
- Le texture vengono composte con un secondo un canale alfa separato



3

## Environment Mapping

- Possiamo emulare le riflessioni utilizzando la direzione del raggio riflesso per indicizzare una texture map posta "all'infinito"
- Assume che tutti i raggi riflessi inizino dallo stesso punto
  - No parallasse (oggetti lontani)
  - Illuminazione fissa
- Environment map: una tessitura che memorizza il colore dell'ambiente "riflesso"



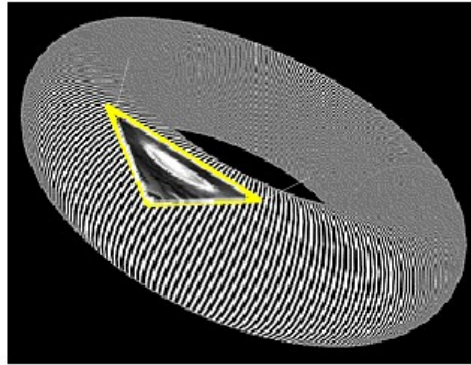
4

## Sphere Mapping

- Memorizziamo in una texture il colore dell'ambiente "riflesso" da ogni normale della semisfera.
- Come coordinate uv, basta utilizzare il vettore riflessione:

$$u = \text{atan2}(r_y, r_x)$$

$$v = \arccos(r_z)$$

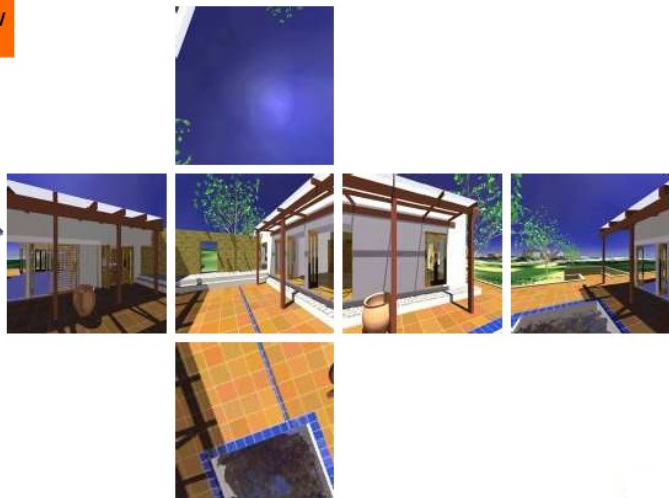
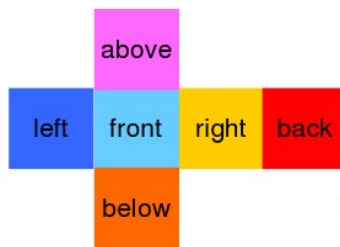
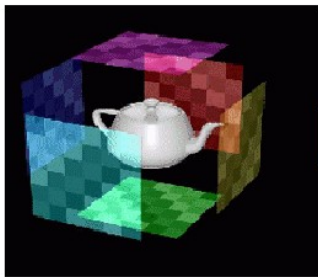


- Mappatura non planare: introduce distorsioni!

5

## Cube Mapping

- Memorizziamo 6 viste dell'ambiente: una per ogni faccia di un cubo



- Campionamento più uniforme dello sphere mapping
- Possibilità di generare real-time l'environment in modo semplice

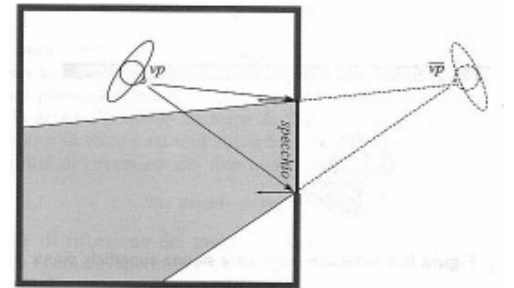
6

## Cube Mapping

- Generazione coordinate Uvm, sia  $n=(n_x, n_y, n_z)^T$  il vettore normale
  - Si sceglie la coordinata massima in valore assoluto di  $n$ , la coordinata ed il segno indicano la faccia:
    - Esempio  $(-3.2, 5.1, -8.4)^T$ , la faccia è -Z
  - Le rimanenti coordinate, divise per il valore massimo, danno le coordinate UV nel range  $[-1; 1]$ 
    - Esempio  $(-3.2/8.4, 5.1/8.4)^T=(-0.38, 0.61)^T$
  - Per passare da un range di valori in  $[-1, 1]$  ad uno in  $[0, 1]$  si aggiunge 1 e si divide per 2
    - Esempio:  $((-0.38 + 1) / 2, (0.61 + 1) / 2) = (0.31, 0.80)$

## Specchi Planari

- Per rendere degli specchi planari esiste una tecnica molto più semplice
- Rendiamo la scena dalla posizione speculare rispetto a quella dell'osservatore
- Applichiamo la scena così resa come texture al posto dello specchio.
- Richiede più passi di rendering (lento), ma la texture può essere riutilizzata se
  - la scena non cambia
  - il punto di vista cambia poco.

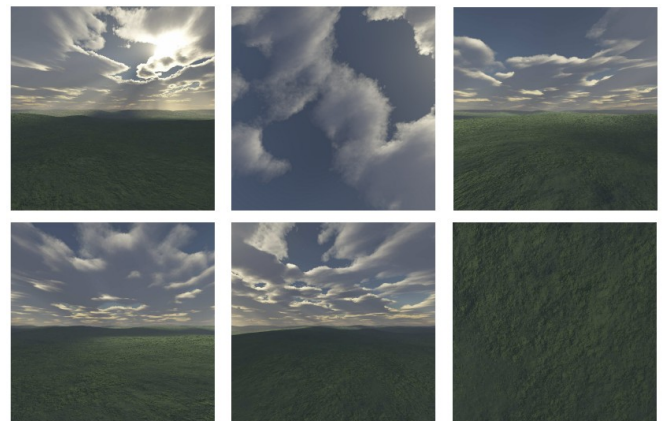


7

## Skybox

- Uno skybox è un metodo per creare lo sfondo di una scena tridimensionale in modo da dare l'illusione dei dintorni (distanti) della scena.
- Si utilizza una serie di texture opportune che rappresentano gli oggetti distanti nella scena (cielo, nuvole, montagne, ecc).
- La tecnica prevede che le texture siano disposte su un cubo di lato unitario con l'osservatore si trova al centro del cubo.
- Come Cube Mapping visto direttamente invece che attraverso l'oggetto riflettente.

- Lo skybox deve essere disegnato per primo
- Prima di disegnare lo skybox bisogna disabilitare il lighting, lo Z-Buffer, ed in generale il blending.
- La camera deve essere posizionata in  $(0,0,0)$  ed orientata secondo l'orientazione dell'osservatore, della camera principale

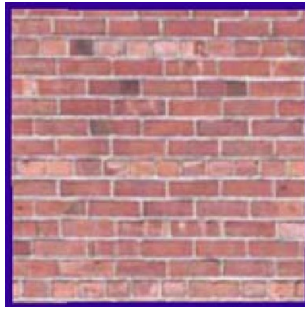


8



## Bump Mapping

- Qual'è la differenza tra un muro di mattoni ed una fotografia di un muro di mattoni mappata su di un piano?
- Cosa succede se cambiamo la illuminazione o la posizione dell'osservatore?

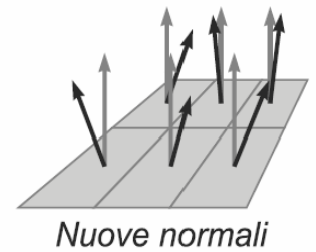
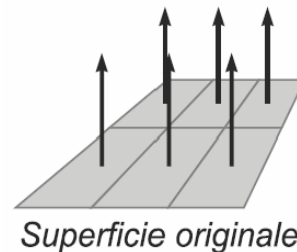


- Le texture non sono efficaci nel modellare superfici ruvide
- Il modello di shading di Phong funziona perché siamo ricettivi a variazioni nelle normali
- Perturbiamo la superficie, in questo modo alteriamo anche l'illuminazione e questo ci dà l'illusione di dettaglio.

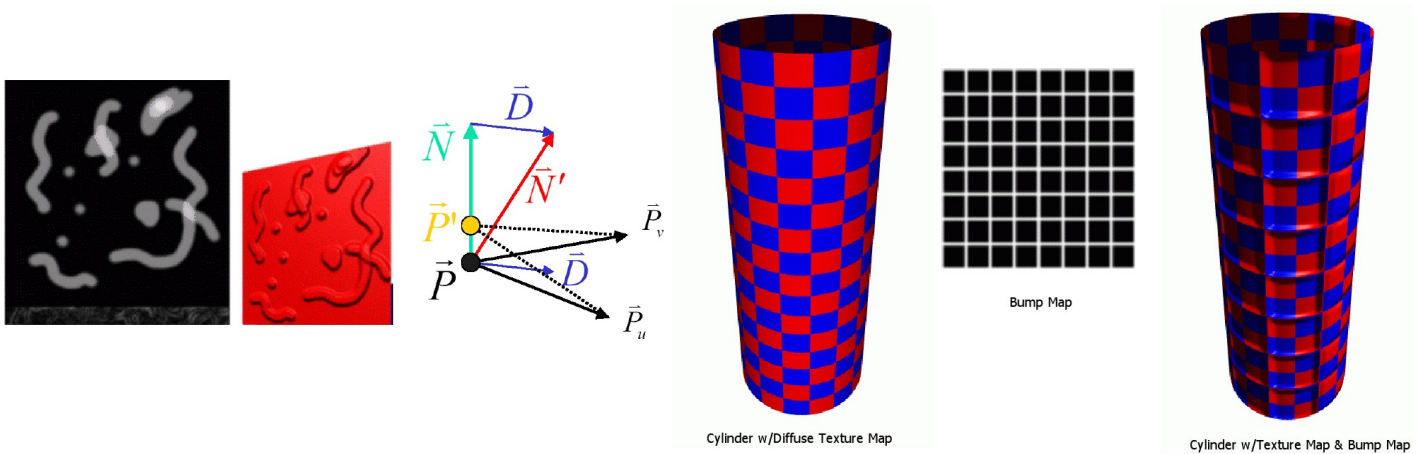
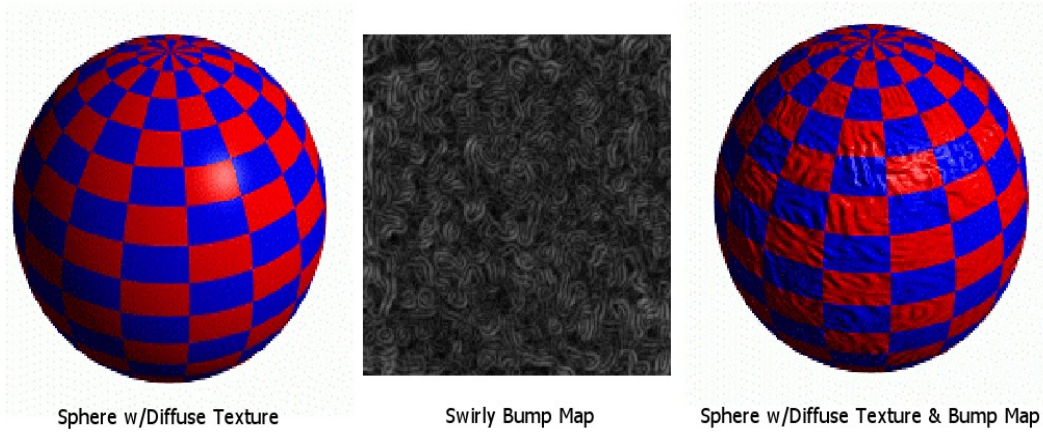
9

## Bump Mapping

- Usiamo le texture per modulare le normali.
- Bump mapping cerca di definire una superficie generica al posto di quella definita dai triangoli
  - La posizione è data dai triangoli
  - L'illuminazione da questa superficie
- Una bump map è una funzione  $b(u,v)$  che rappresenta la variazione della superficie rispetto al piano del triangolo.
  - È interpretabile come una mappa di altezze
- Sia
  - $\mathbf{p}$  un punto sul triangolo di coordiante texture  $(u,v)$
  - $\mathbf{n}$  la normale in  $(u,v)$
  - Allora la superficie passa per  $\mathbf{p} + b(u,v)\mathbf{n}$
  - La normale nel punto perturbato sarà
 
$$\mathbf{n}' = \mathbf{n} - b_v(u,v)(\mathbf{n} \times \mathbf{p}_u) + b_u(u,v)(\mathbf{n} \times \mathbf{p}_v)$$
    - $b_v(u,v)$  e  $b_u(u,v)$  sono le derivate della bump map rispetto a  $v$  e  $u$
    - $\mathbf{p}_u$  e  $\mathbf{p}_v$  sono le derivate della posizione del punto rispetto a  $u$  e  $v$
- Quindi data una bump map, calcoliamo  $\mathbf{n}'$  per ogni pixel e quindi calcoliamo l'illuminazione di conseguenza



## Bump Mapping



11

## Normal Mapping

- Calcolare le derivate per ogni sample è dispendioso e introduce errori: pre-computare!
- Invece che mantenere una mappa delle altezze, manteniamo una mappa delle derivate.
- Il metodo prevede di variare la normale alla superficie utilizzando la formula:

$$\vec{N}_{new} = \vec{N}_{old} + \vec{D};$$

$$\vec{D} = (\Delta x, \Delta y, \Delta z)$$

- Il vettore D viene specificato dai 3 canali e può essere definito in
  - *Object space*: sistema di riferimento unico per tutto l'oggetto
  - *Tangent space*: il sistema di riferimento ruota con la superficie mantenendo la normale allineata con la z della texture
- Object Space: semplice ed efficiente, ma le mappe non si possono riutilizzare o comporre per aggiungere dettaglio
- Tangent space: bisogna ri-orientare la variazione prima di applicarla, ma richiedono meno spazio (z inutile)

12

## Displacement Mapping

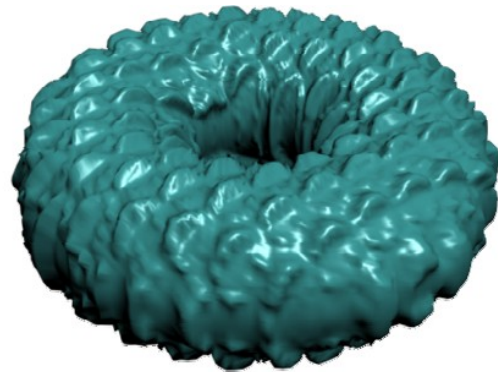
- Nel displacement mapping si modifica effettivamente la geometria dell'oggetto spostando i punti della superficie:

$$P_{new} = P_{old} + h \cdot \vec{N}$$

- Il displacement mapping è eseguito in fase di rendering e non modifica stabilmente la geometria della scena
- Rispetto al bump mapping anche la silhouette del modello mostra le corrette deformazioni
- Richiede alta densità di triangoli (altera posizione in vertex shader)



**Bump  
Mapping**

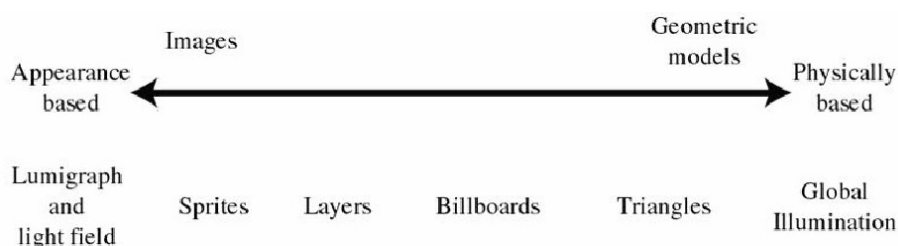


**Displacement  
Mapping**

13

## Image-Based Rendering

- L'idea base è utilizzare (in vari modi) immagini al posto dei dati degli oggetti (geometria + materiali).
- Un oggetto complesso può essere visualizzato più velocemente rimpiazzandolo (in modo coerente) con una sua immagine.
  - Già visto per riflessioni e skybox
- Si ottiene maggior fotorealismo con minore sforzo computazionale.
- Si va dalle tecniche di rendering completamente basate su immagini (nessun modello 3D della scena)
- Ad ibridi scena 3D-immagini (billboards)
- Fino ad arrivare a rendering dove nessun'altra immagine viene utilizzata per generare l'immagine finale (algoritmi di illuminazione globale)



14

## Billboards

- Molti effetti visivi, come il lens flares ad esempio, sono basati sulla tecnica del billboard.
- Il billboard consiste nel renderizzare un'immagine su un quadrilatero orientato secondo il punto di vista.
- Insieme all'alpha texturing questa tecnica può essere utilizzata per visualizzare fenomeni come la nebbia, il fumo, il fuoco, le nuvole, oppure per rimpiazzare con immagini oggetti complessi, come ad esempio un albero.
- Tipicamente, l'orientazione del quadrilatero desiderata è caratterizzata dalla sua normale ( $n$ ) e dalla direzione di "alto" (up direction  $u$ ).
- Grazie a questi due vettori è possibile definire una base ortonormale alla superficie e quindi una matrice di rotazione per orientare il quadrilatero nel modo desiderato.
- In tutte le tecniche di billboarding uno di questi due vettori è fisso e deve mantenere una determinata direzione. L'altro deve essere reso perpendicolare a questo.
- Per ottenere la base ortonormale si procede nel seguente modo:
  - Si calcola il vettore "right"  $r = n \times u$  e lo si normalizza
  - Se  $n$  è il vettore fisso si ottiene la nuova up direction ( $u'$ ) calcolando il prodotto vettoriale  $u' = n \times r$



15

## Billboards

- Esistono diverse tipologie di billboarding a seconda del modo di orientazione:
  - Screen-Aligned Billboarding
  - World-Oriented Billboarding
    - View-plane aligned
    - Viewpoint aligned
  - Axial Billboarding

### Screen-Aligned Billboard

- È la forma più semplice di billboarding.
- L'obiettivo di questo tipo di billboarding è ottenere un effetto "sprite", ossia l'immagine sempre parallela allo schermo e con il vettore up costante.
- Per questo tipo di billboarding la normale desiderata è la normale del view plane cambiata di segno ed  $u$  è l'*up vector* della camera ( $u_{camera}$ ).
- Sprite

- Usati nei vecchi videogiochi 2D
- Esempio: cursore del mouse
- Definiti in un rettangolo
- Mapping 1-to-1 con i pixels dello schermo



background

+



sprite image

=



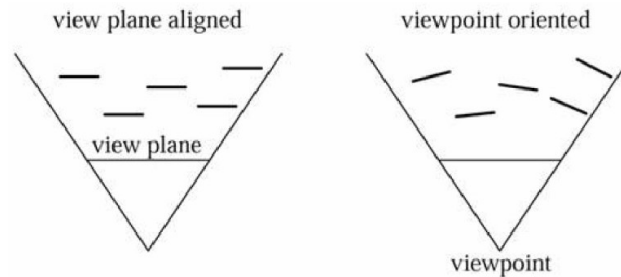
result

16



## World-oriented Billboarding

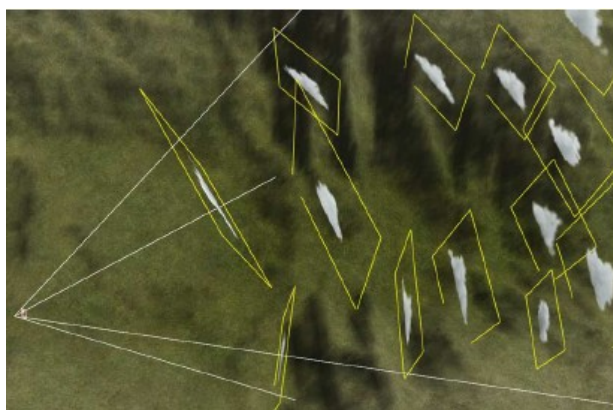
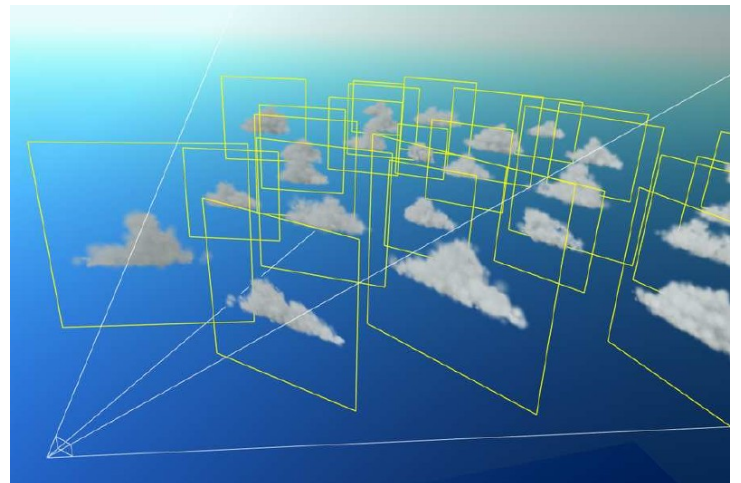
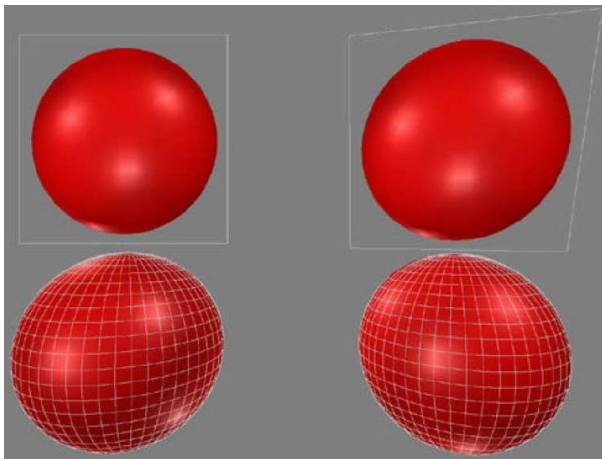
- Screen-aligned billboarding funziona bene per oggetti a simmetria circolare, ma in generale, se la camera ruota intorno alla direzione di vista questo può causare problemi.
- Si utilizza quindi al posto di  $u_{\text{camera}}$  l'*up vector* della scena ( $u_{\text{world}}$ ). In questo caso il vettore  $n$  rimane lo stesso del caso precedente
  - VIEWPLANE ALIGNED BILLBOARDING.
- Un altro modo di allineare gli oggetti è quello di utilizzare come vettore  $u=u_{\text{world}}$  e come vettore  $n$  la direzione che connette il centro della billboard al punto di vista
  - VIEWPOINT ORIENTED BILLBOARDING.



- *La viewpoint orientation* introduce distorsione questo conferisce un prospettiva nella billboard → maggiore realismo all'oggetto. Risulta una buona tecnica per visualizzare impostor.
- *View plane-aligned* risulta una buona tecnica per la realizzazione di particle system.

17

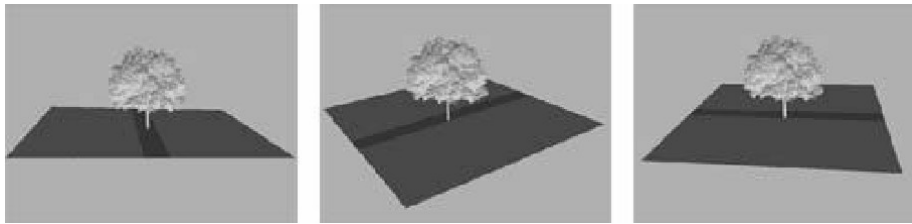
## Viewplane vs Viewpoint



18

## Axial Billboarding

- Questa tecnica è molto utile per rappresentare oggetti a simmetria cilindrica.
  - Per esempio è molto usato per visualizzare alberi od altri effetti grafici (raggi laser).
- In questo caso l'oggetto deve essere ruotato intorno ad un asse fisso fino ad allineare la billboard il più possibile con l'osservatore (tipicamente l'asse fisso è il vettore  $u=u_{world}$ ).
- Una volta che la billboard è orientata secondo l'osservatore può essere traslata nella sua posizione.
- Vantaggio: effettuo il rendering di un'immagine anziché di una geometria complessa.
- Svantaggio: se osservassi l'albero dall'alto avvicinandomi vedrei quest'ultimo assottigliarsi (pensate ad una carta da gioco).
  - Questo viene parzialmente ovviato mettendo più billboards in varie posizioni



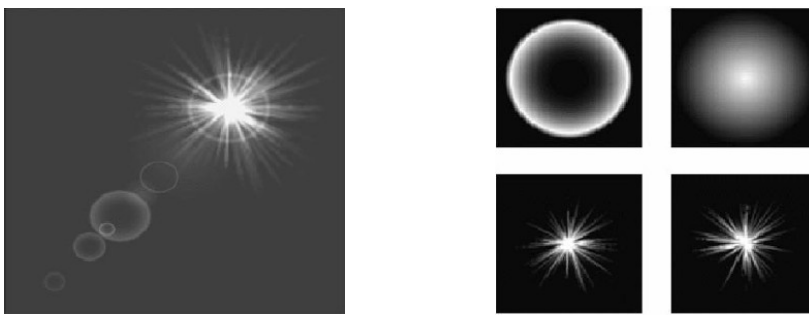
19

## Impostor

- Un impostor è una billboard creata on-the-fly
- Si usa per velocizzare il rendering di oggetti complessi multipli, se ad esempio ho più istanze di uno stesso oggetto creo l'impostore e rimpiazzo la geometria al volo
- Utilissimo per oggetti statici e distanti (skybox)
- Per creare un impostor
  - Scelgo un punto di vista appropriato
  - Renderizzo l'impostor su una texture (tenendo conto di settare lo sfondo trasparente)
  - Uso la texture come immagine per la billboard
- Per la visualizzazione dell'impostor solitamente si utilizza la tecnica di billboarding viewpoint-aligned

## Lens Flares

- Effetto risultante da un insieme di effetti visivi causati dalla riflessioni della luce diretta sulla lente della camera (glare effects)



20

## Particle systems

- I sistemi particellari vengono utilizzati per renderizzare una grande quantità di effetti quali il fuoco, il fumo, una fontana, un'esplosione, ecc.
- In realtà si tratta di una tecnica di animazione e non di visualizzazione.
- Si parte da un insieme di piccoli oggetti (punti, linee o texture di piccole dimensioni) che poi vengono visualizzati (view-plane aligned billboarding) ed animati a centinaia per ottenere il fenomeno desiderato.
- Ogni oggetto ha la sue proprietà di velocità, accelerazione, decadimento, colore che possono essere funzioni delle particelle vicine, funzioni del tempo, ecc. Ad esempio:
  - $\text{pos}(t+1) = f(\text{pos}(t), \dots)$
  - $\text{vel}(t+1) = f(\text{vel}(t), \dots)$
  - $\text{colore}(t+1) = f(\text{colore}(t), p(t), \text{vel}(t), \dots)$
- Spesso le particelle si combinano tramite blending

