Grafica al calcolatore

-

Computer Graphics

9 – Tecniche di Mapping

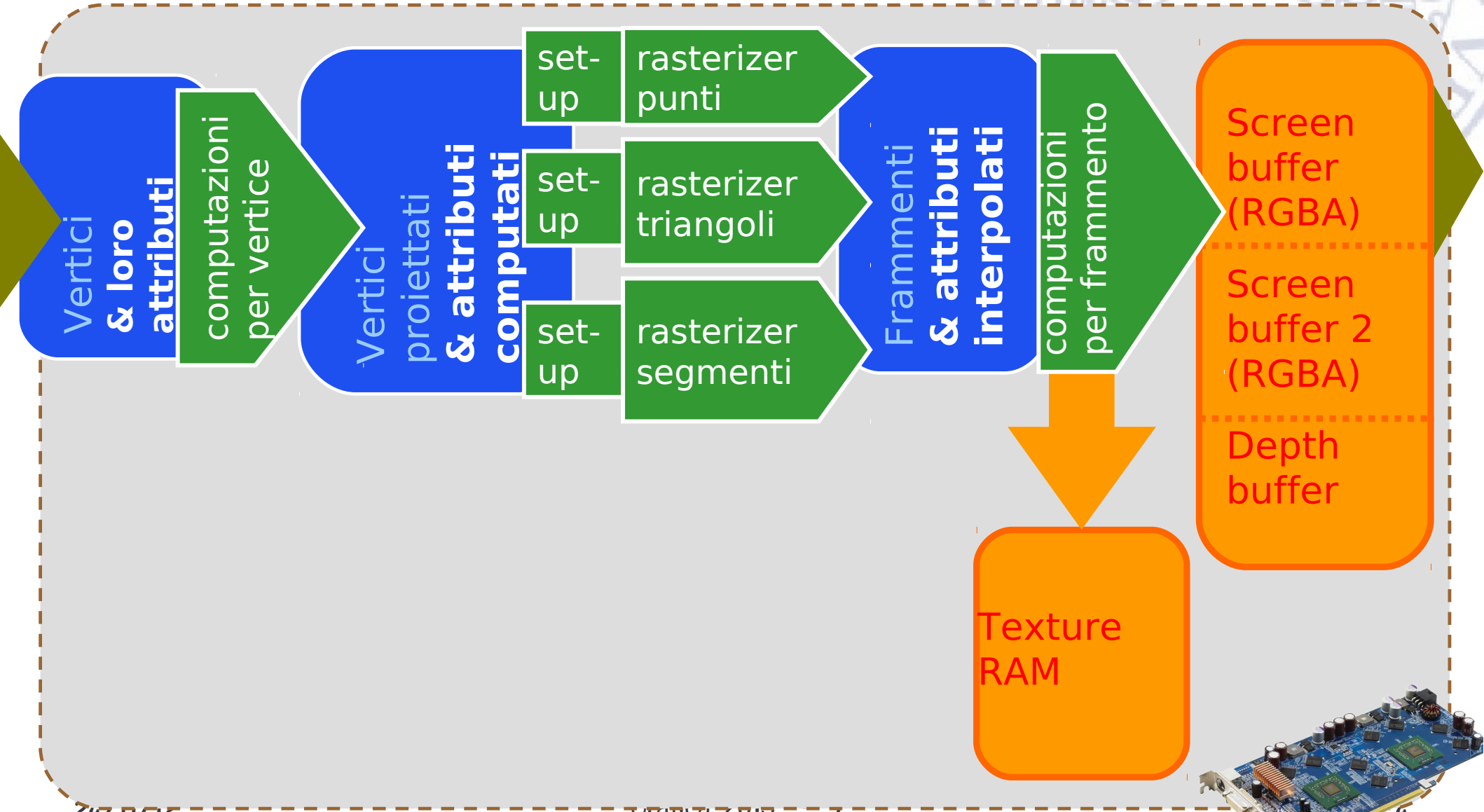
Introduzione

- Il modello di illuminazione di Phong è abbastanza versatile: con una scelta opportuna dei vari parametri si possono imitare diversi materiali in modo abbastanza realistico (per esempio la plastica)
- E' comunque limitato: non si possono simulare i dettagli di un materiale, a meno di non introdurli nella geometria (ovvero aumentare il numero dei poligoni).
- Invece di incrementare la complessità del modello, si aggiungono particolari come parte del processo di rendering.
- Le tecniche di mappatura (texture mapping) interagiscono con lo shading usando una mappa bidimensionale o texture (tessitura) come tabella di lookup, in modo da aggiungere dettagli alla superficie.
- Si possono ottenere risultati ottimi dal punto di vista del fotorealismo con un limitato carico computazionale.
- Il texture mapping è ampiamente supportato dalle schede grafiche.

Texture mapping

- Il texture mappping, nella sua forma più semplice (color mapping) consiste nell'applicare una immagine su una superficie, come una decalcomania.
- Esempi: una etichetta su una lattina, una foto su un cartellone pubblicitario, oppure tessiture regolari come legno o marmo su una superficie.
- Una texture map una matrice bidimensionale di dati indirizzati da due coordinate s e t comprese tra 0 ed 1. Il dato contenuto è tipicamente un colore (la mappa è una immagine), ma potrebbe essere qualcos'altro.
- Più in generale una texture map può contenere qualunque tipo di informazione che incide sull'apparenza di una superficie: la texture map è una tabella ed il texture mapping consiste nel recuperare dalla tabella (lookup) l'informazione che serve per effettuare il rendering di un certo punto.

Memoria RAM nelle schede grafiche



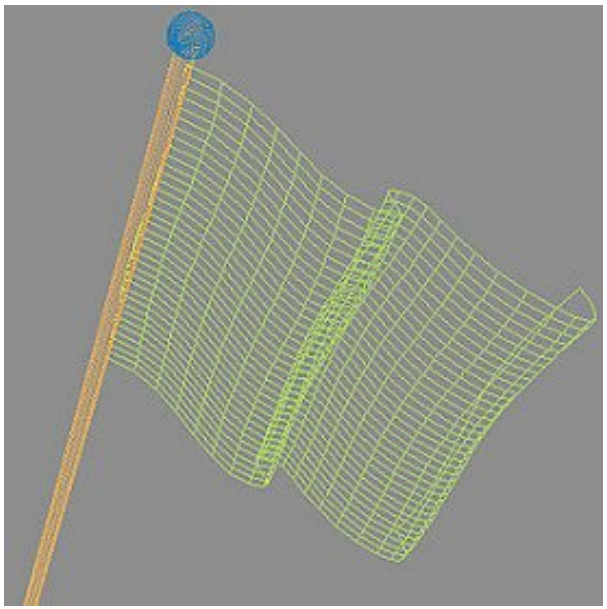
Texture Mapping

- Nelle operazioni per frammento si può accedere ad una RAM apposita: la Texture RAM strutturata in un insieme di Textures (“tessiture”)
- Ogni tessitura è un array 1D, 2D o 3D di Texels (campioni di tessitura) dello stesso tipo

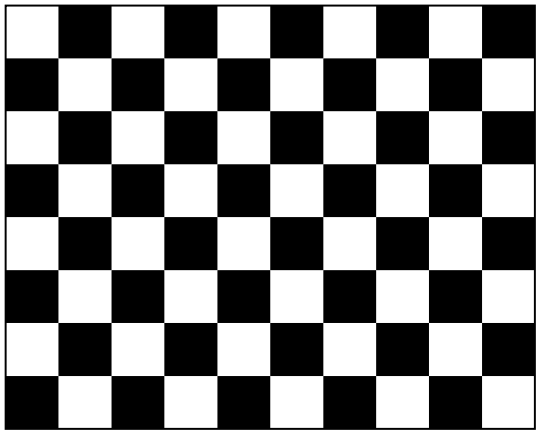
Texels

- Sono esempi di texels:
  - Ogni texel un colore (componenti: R-G-B, o R-G-B-A): la tessitura è una "color-map"
  - Ogni texel una componente alpha: la tessitura è una "alpha-map"
  - Ogni texel una normale (componenti: X-Y-Z): la tessitura è una "normal-map" o "bump-map"
  - Ogni texel contiene un valore di specularità: la tessitura è una "shininess-map"

Rimappare immagini sulla geometria



+

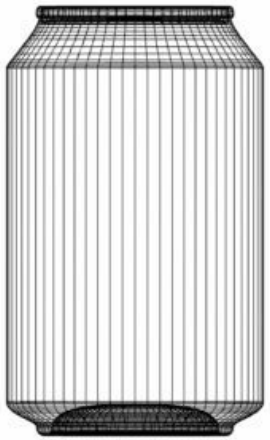


=

geometria 3D
(mesh di quadrilateri)

RGB texture 2D
(color-map)

Rimappare immagini sulla geometria



Rimappare immagini sulla geometria

+

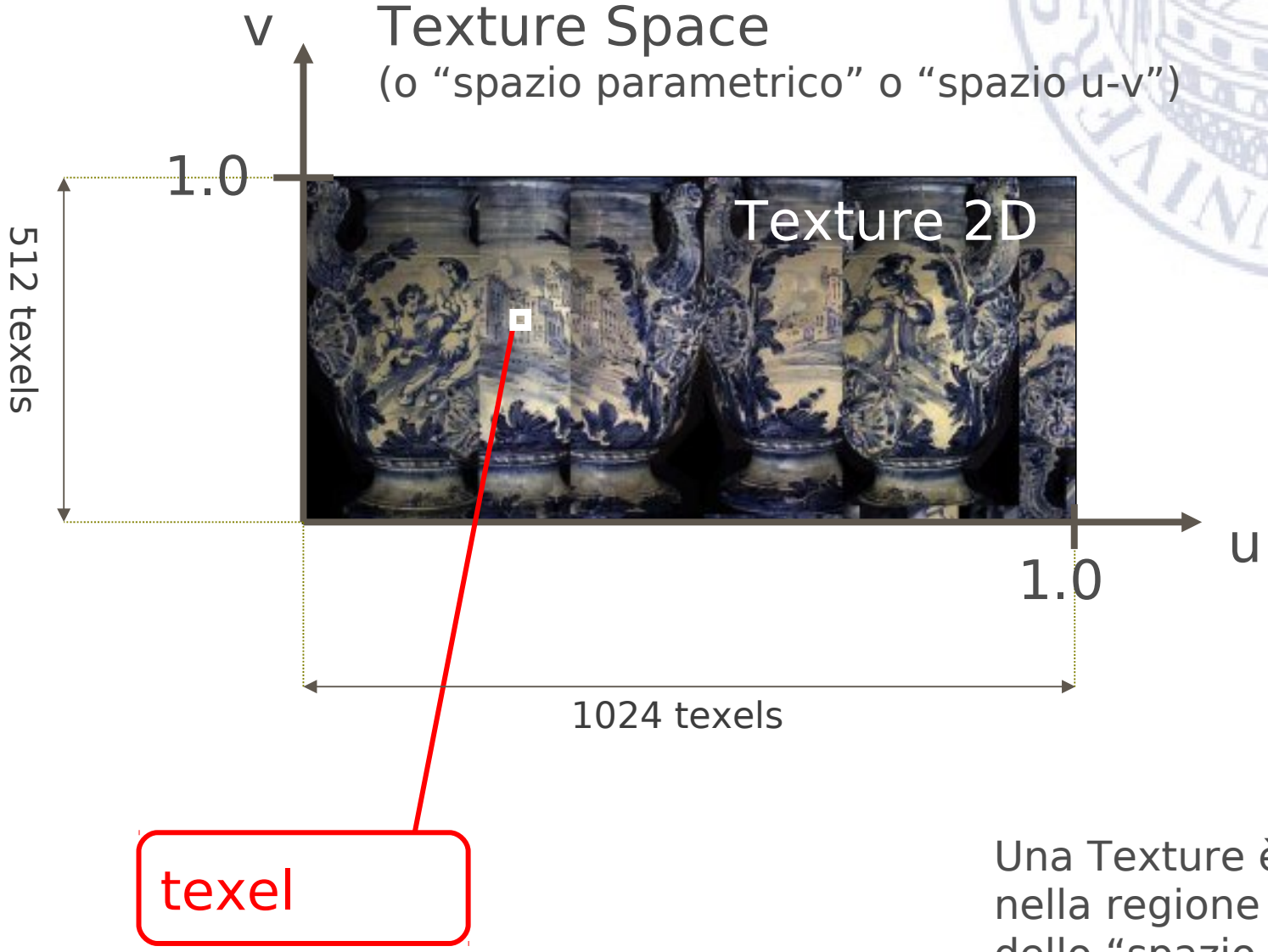
=

Texture Mapping: storia

- 1974 introdotto da Ed Catmull
  - nella sua Phd Thesis
- Solo nel 1992 (!) si ha texture mapping in hardware
  - Silicon Graphics RealityEngine
- Dal '92 a oggi ha avuto aumento rapidissimo della diffusione
  - strada intrapresa soprattutto da low end graphic boards
- Oggi è una delle più fondamentali tecniche di rendering
  - Il re indiscusso delle tecniche image based

Ed Catmull

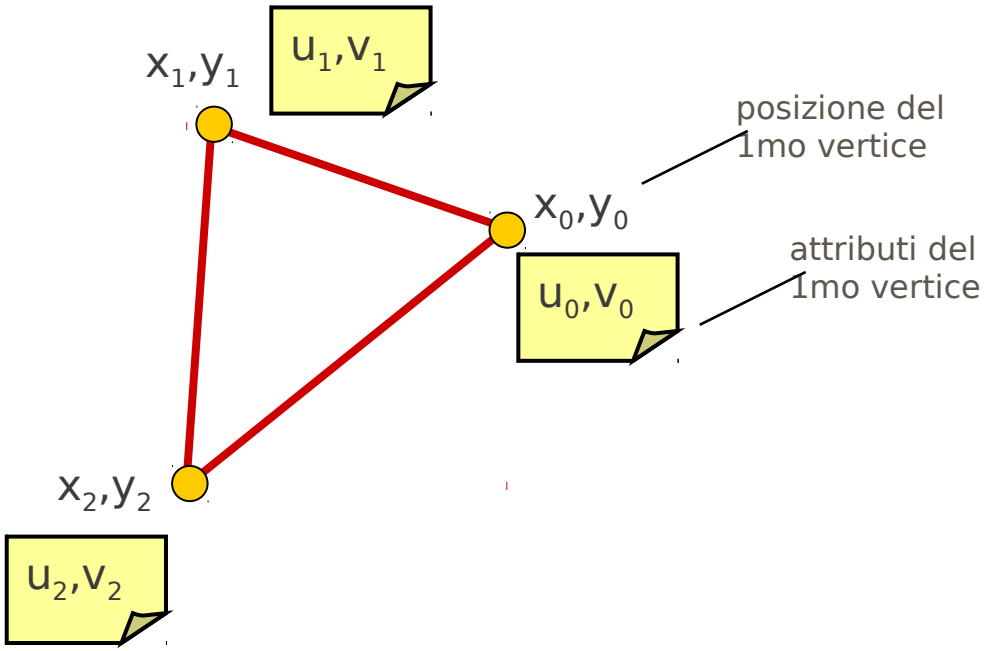
Notazione



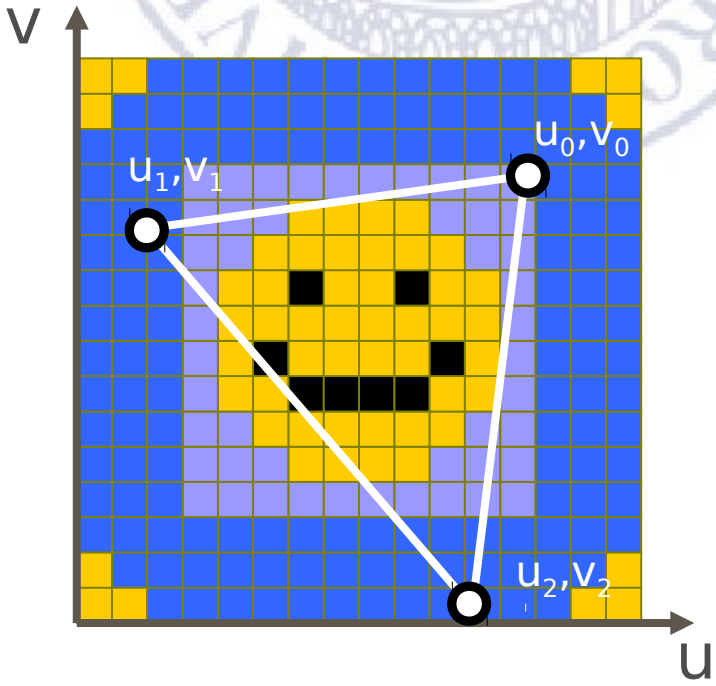
Una Texture è definita nella regione [0,1] x [0,1] dello “spazio parametrico”

Texture Mapping

- Ad ogni vertice (di ogni triangolo) assegno le sue coordinate u,v nello spazio tessitura



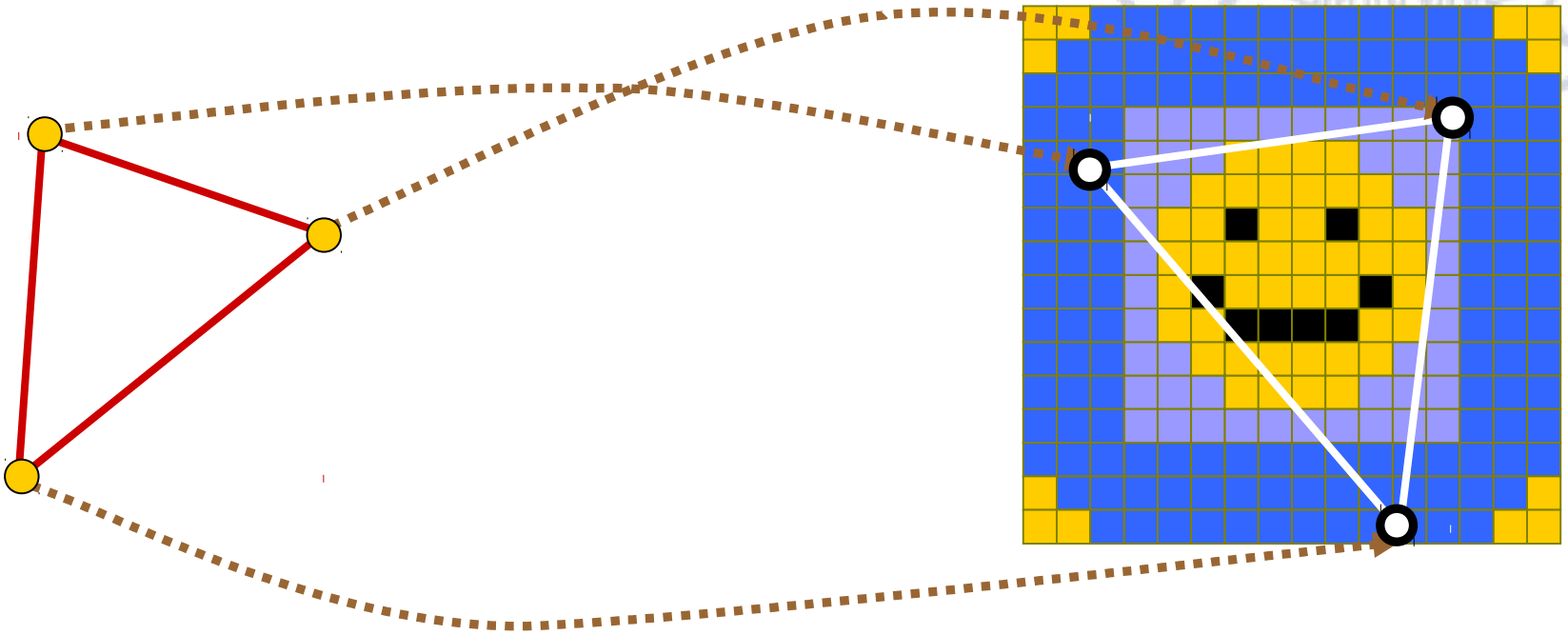
Screen Space



Texture Space

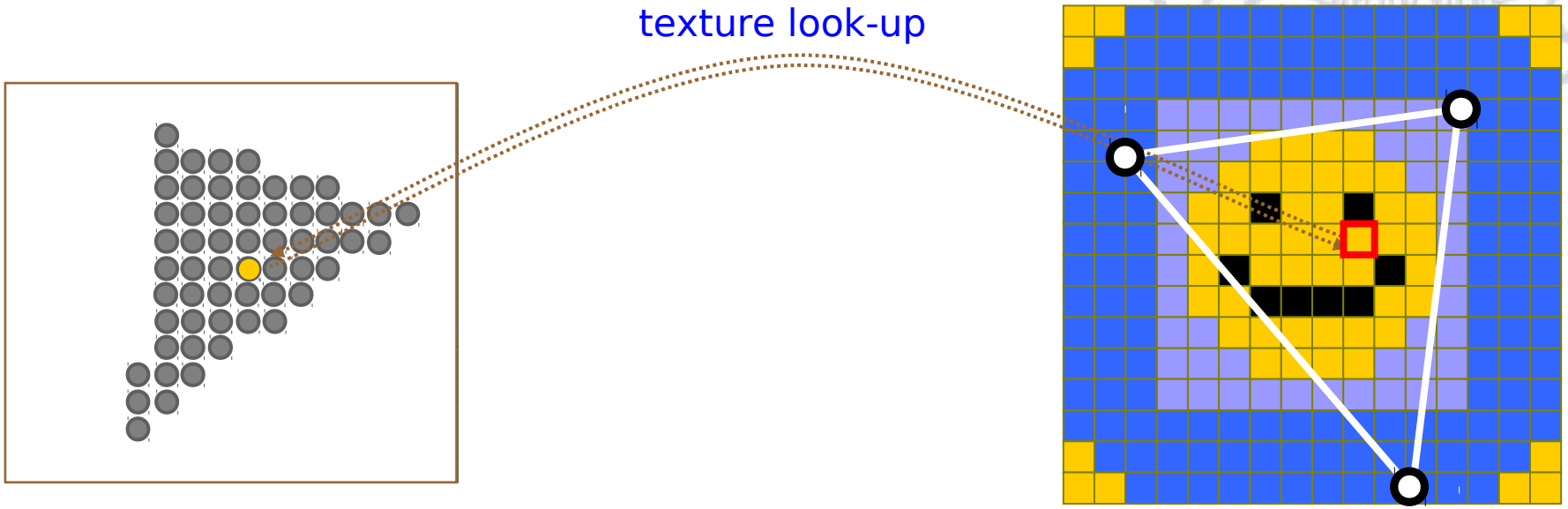
Texture Mapping

- Così in pratica definisco un mapping fra il triangolo e un triangolo di tessitura



Texture Mapping

- Ogni vertice (di ogni triangolo) ha le sue coordinate u,v nello spazio tessitura



Texture Mapping

Vertici **& loro attributi**

computazioni per vertice

Vertici poriettati **& attributi computati**

set-up

set-up

set-up

rasterizer punti

rasterizer triangoli

rasterizer segmenti

Frammenti **& attributi interpolati**

computazioni per frammento

Screen buffer

texture look-up

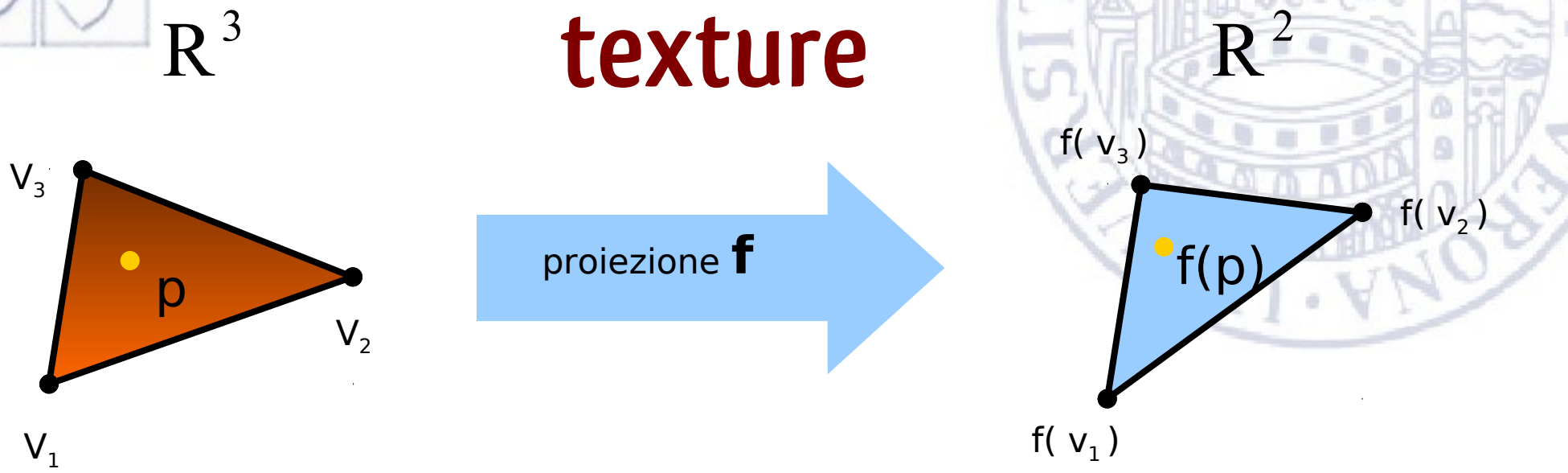
compreso: coordinate texture (per vertice!)

interpolo coordinate texture

coordinate texture interpolate

Texture RAM

Interpolazione delle coordinate texture



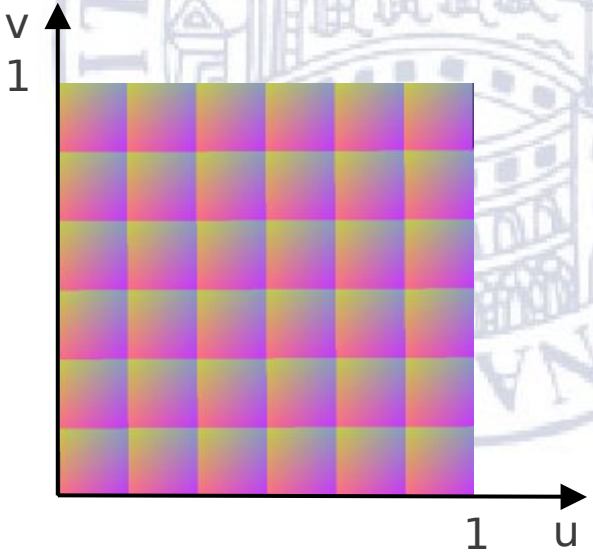
p ha coordinate baricentriche a,b,c nel triangolo v_1 v_2 v_3

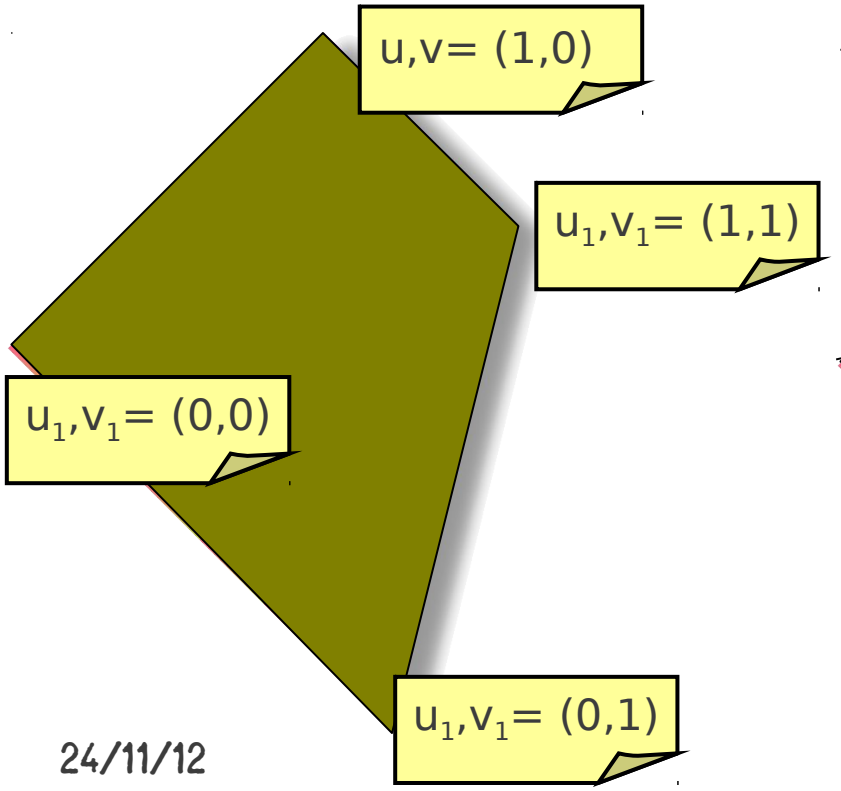
f(p) ha coordinate baricentriche a,b,c nel triangolo $f(v_1)$ $f(v_2)$ $f(v_3)$

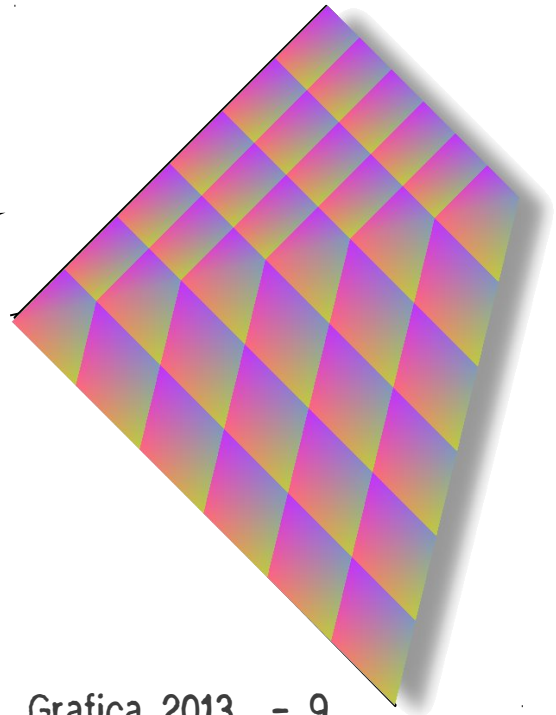
- Non vale per la proiezione prospettica poiché è solo una approssimazione che è utile per colori e normali ma non funziona quando interpoliamo coordinate texture...

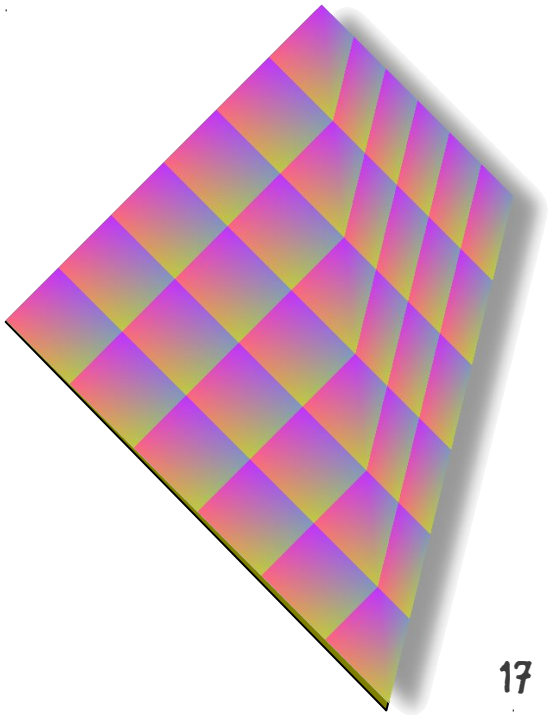
Interpolazione delle coordinate texture

- Esempio:





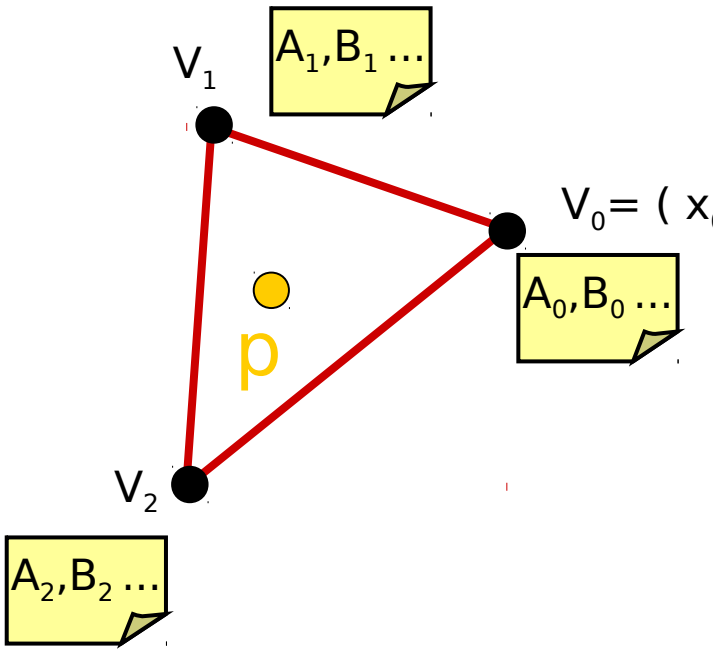




Correzione Prospettica

- p ha coordinate baricentriche $c_0\ c_1\ c_2$

$$p = c_0 v_0 + c_1 v_1 + c_2 v_2$$



attributi di p:
(senza considerare
la correzione prospettica)

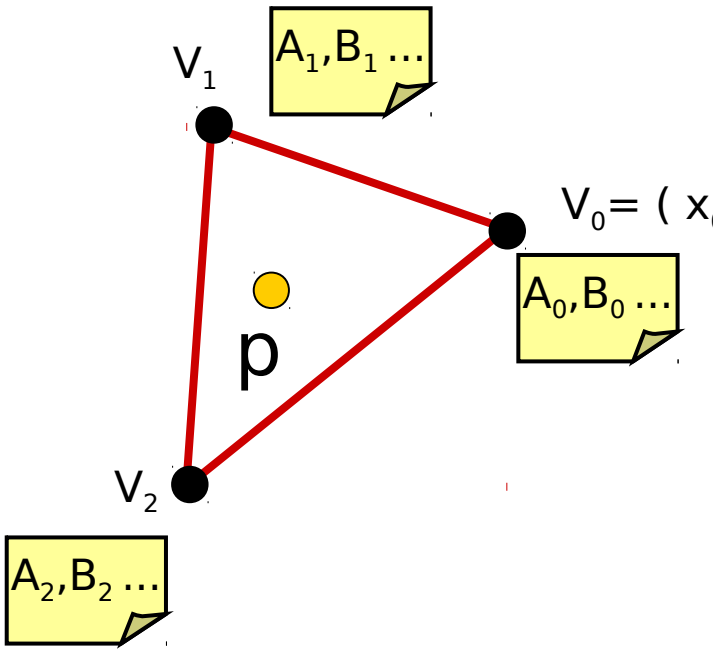
$$A_p = c_0 A_0 + c_1 A_1 + c_2 A_2$$

$$B_p = c_0 B_0 + c_1 B_1 + c_2 B_2$$

Correzione Prospettica

- **p** ha coordinate baricentriche $c_0\ c_1\ c_2$

$$p = c_0 v_0 + c_1 v_1 + c_2 v_2$$

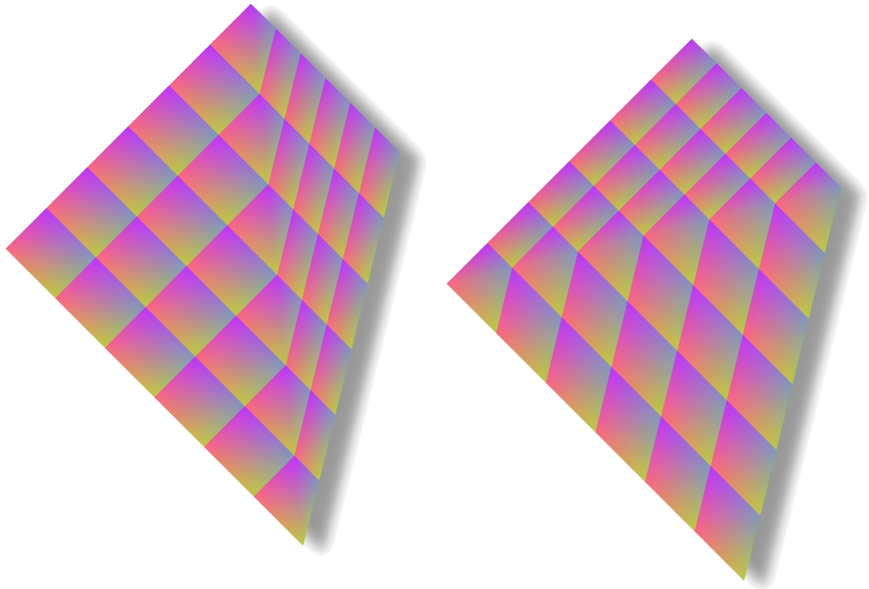


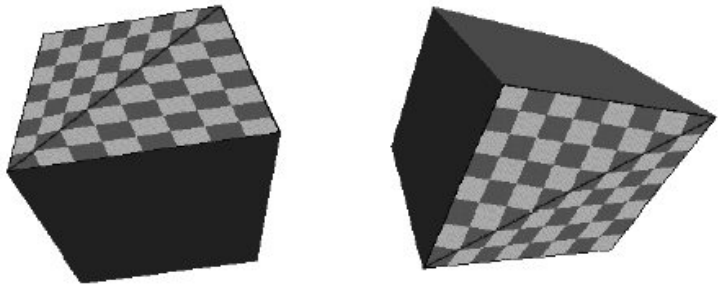
attributi di p:
(con la correzione prospettica)

$$A_p = \frac{c_0 \frac{A_0}{w_0} + c_1 \frac{A_1}{w_1} + c_2 \frac{A_2}{w_2}}{c_0 \frac{1}{w_0} + c_1 \frac{1}{w_1} + c_2 \frac{1}{w_2}}$$

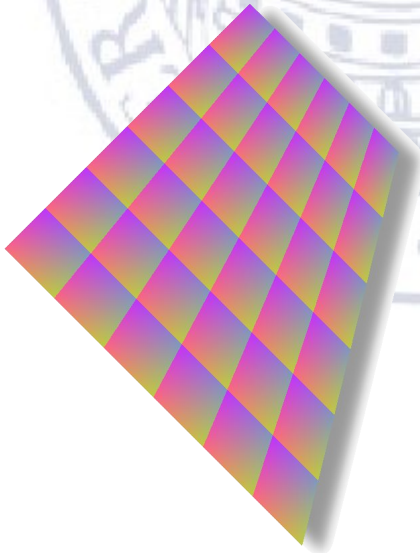
Correzione Prospettica

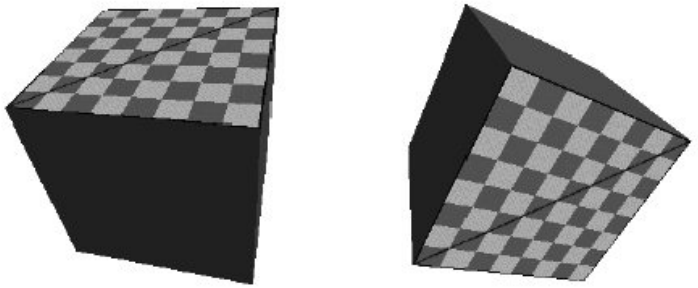
- Senza





- Con

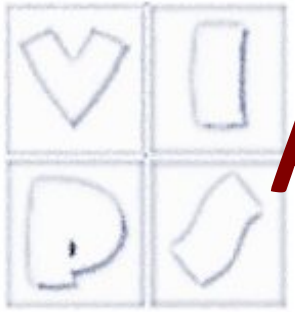




Nota: la tessitura va caricata

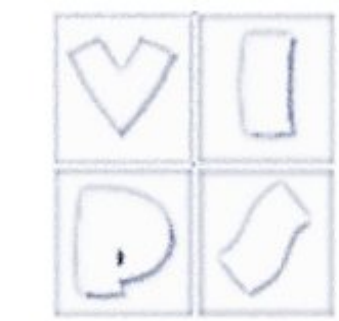
Nota: la tessitura va caricata

- Da disco a memoria RAM main (sulla scheda madre)
- Da memoria RAM main a Texture RAM (on board dell'HW grafico)
- Entrambe le operazioni sono piuttosto lente e sono impossibili da fare una volta per frame quindi nel progetto dell'applicazione si devono utilizzare strategie per la gestione delle texture



Assegnazione delle coordinate texture

- Due classi di soluzioni:
  - Calcolare le coordinate textures on-the-fly durante il rendering…
  - Precomputarle (e salvarle insieme alla mesh)
- Non esiste una soluzione ideale, dipende dall'applicazione che stiamo progettando
- Modelli con una sola texture l'avranno precomputata, per altri che variano dinamicamente l'assegneremo in rendering



Texture mapping

- Da Angel

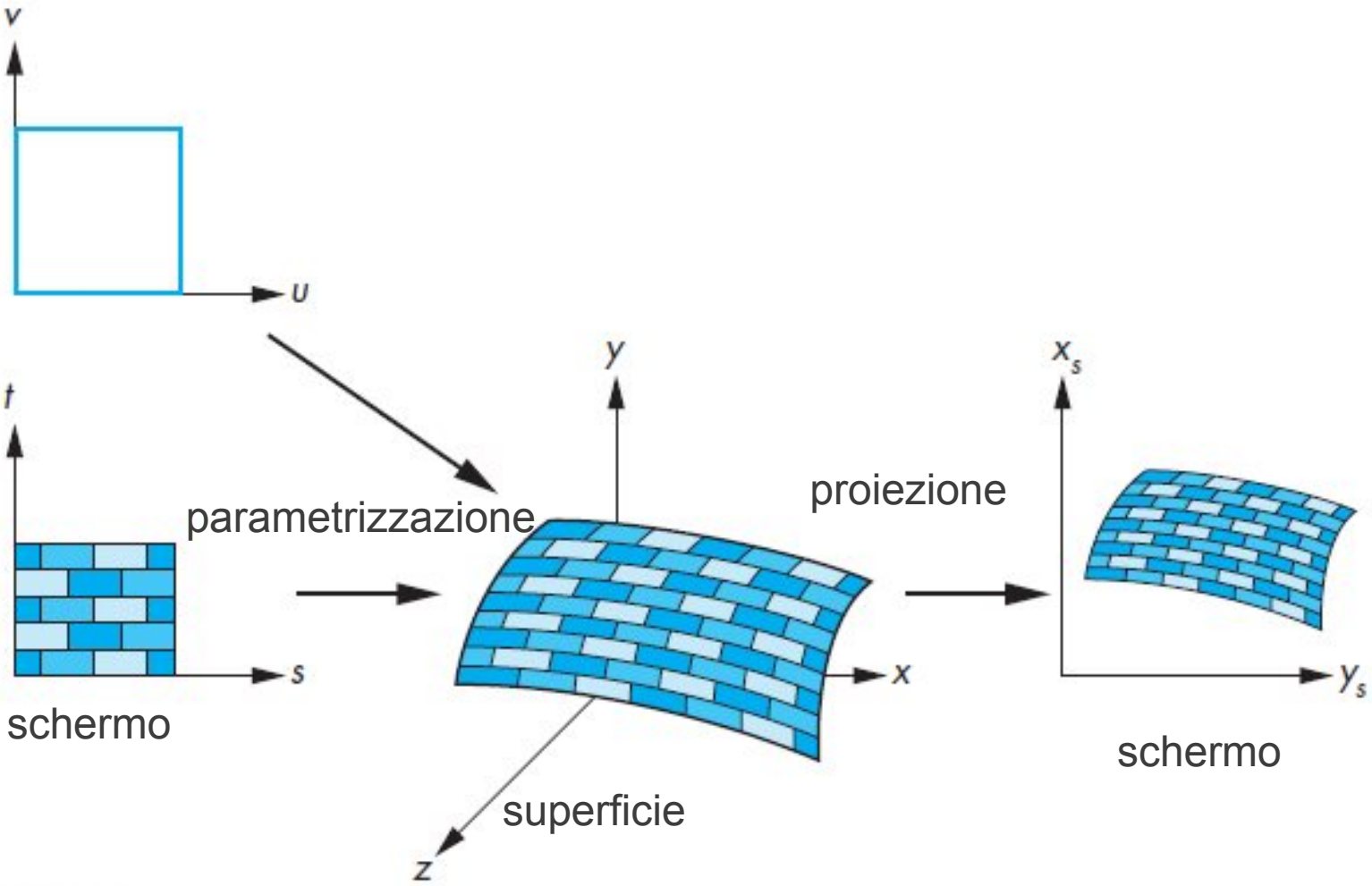


FIGURE 7.9 Texture maps for a parametric surface.

Texture mapping

- La texture può essere applicata dopo il calcolo della illuminazione con Phong (per modificare attributi come colore, luminosità o trasparenza) oppure può modificare i parametri (come le normali) che entrano nel modello di Phong.
- Un passaggio chiave è stabilire una corrispondenza univoca tra superficie dell'oggetto e texture.
- Occorre definire la funzione di parametrizzazione $W()$ che associa un punto $(s; t)$ della texture ad un punto $\mathbf{P}$ della superficie dell'oggetto 3D (è una funzione che "spalma " la texture sulla superficie).
- Il punto P viene poi mappato dalla proiezione in un punto $(x_s; y_s)$ dello schermo. Il rendering della texture si occupa poi di stabilitre il valore di texture da associare a ciascun pixel.

Parametrizzazione

- Nella mappatura in un passo si definisce (in forma analitica) la funzione W che definisce la corrispondenza tra i pixel della texture ed i punti della superficie.
- Questo si può fare quando si ha la descrizione parametrica della superficie soggiacente alla maglia poligonale.
- Altrimenti si specifica tabularmente la corrispondenza W^{-1} tra vertici della maglia e punti della texture.
- Se la superficie è data in forma parametrica, ad ogni suo punto P sono associate due coordinate (parametri) $(u; v)$.
- Per ottenere W basta specificare la mappa che va da $(u; v)$ sulla superficie a $(s; t)$ nella texture.
- E' opportuno che W sia invertibile. Spesso è l'identità (con qualche fattore di normalizzazione).

Parametrizzazione

- Ad esempio si consideri un cilindro di altezza h. per il quale si definisce la funzione

$$W : (\theta, z) \rightarrow (s, t) = (\frac{m}{2\pi}\theta, \frac{n}{h}z)$$

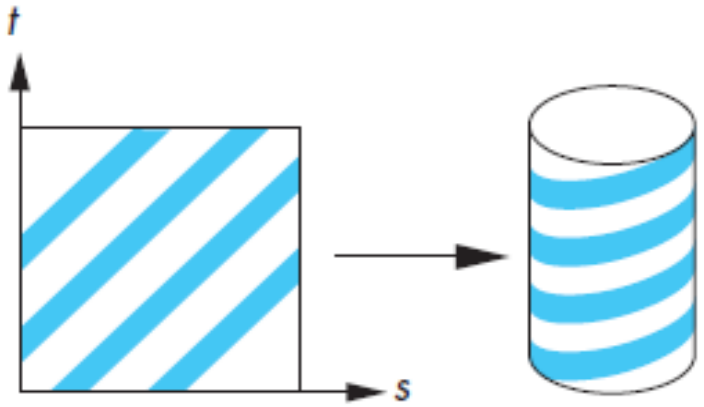


FIGURE 7.13 Texture mapping with a cylinder.

- dove (m; map.

S/O - Mapping

- Una tecnica più generale, che si può usare senza conoscere l'equazione parametrica della superficie, è la mappatura in due passi
- Si mappa la texture su una superficie intermedia semplice, in modo che la parametrizzazione (corrispondenza punti-superficie con pixel-texture) sia immediata; questa prende il nome di S-mapping
- Quindi si mappa ogni punto della superficie intermedia in un punto della superficie in esame; questa prende il nome di O-mapping
- La concatenazione dei due mapping genera la corrispondenza W tra i pixel della texture ed i punti dell'oggetto

- Il primo passaggio (S-mapping) è in genere semplice; basta scegliere superfici facili da parametrizzare
- Ad esempio si può prendere come superficie intermedia un cilindro (vedi slide precedente)
- Oltre al cilindro è facile fare l'S-mapping con cubi, piani e sfere.
- In genere si considera la superficie intermedia come esterna all'oggetto da tessiturare.

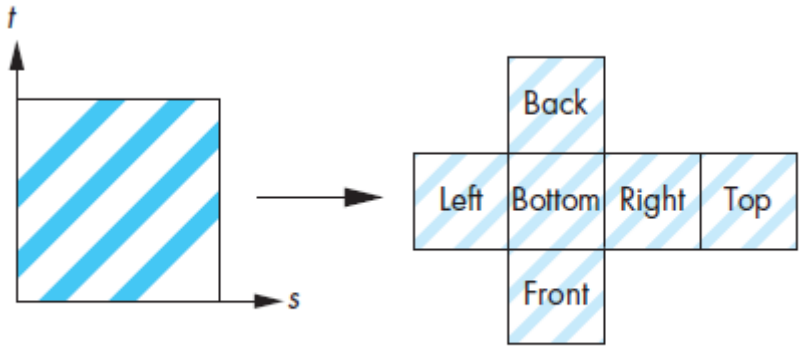
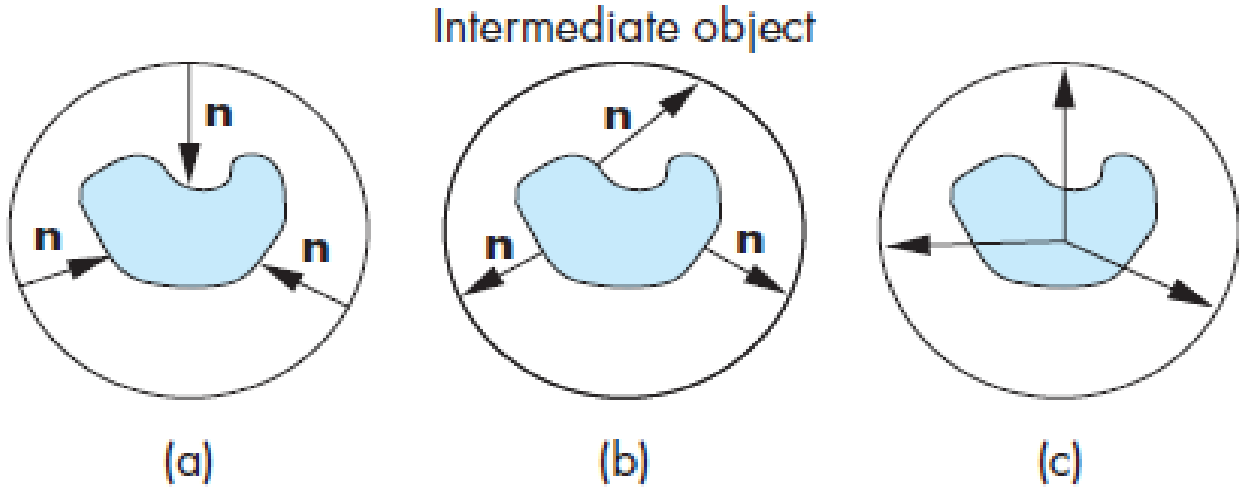


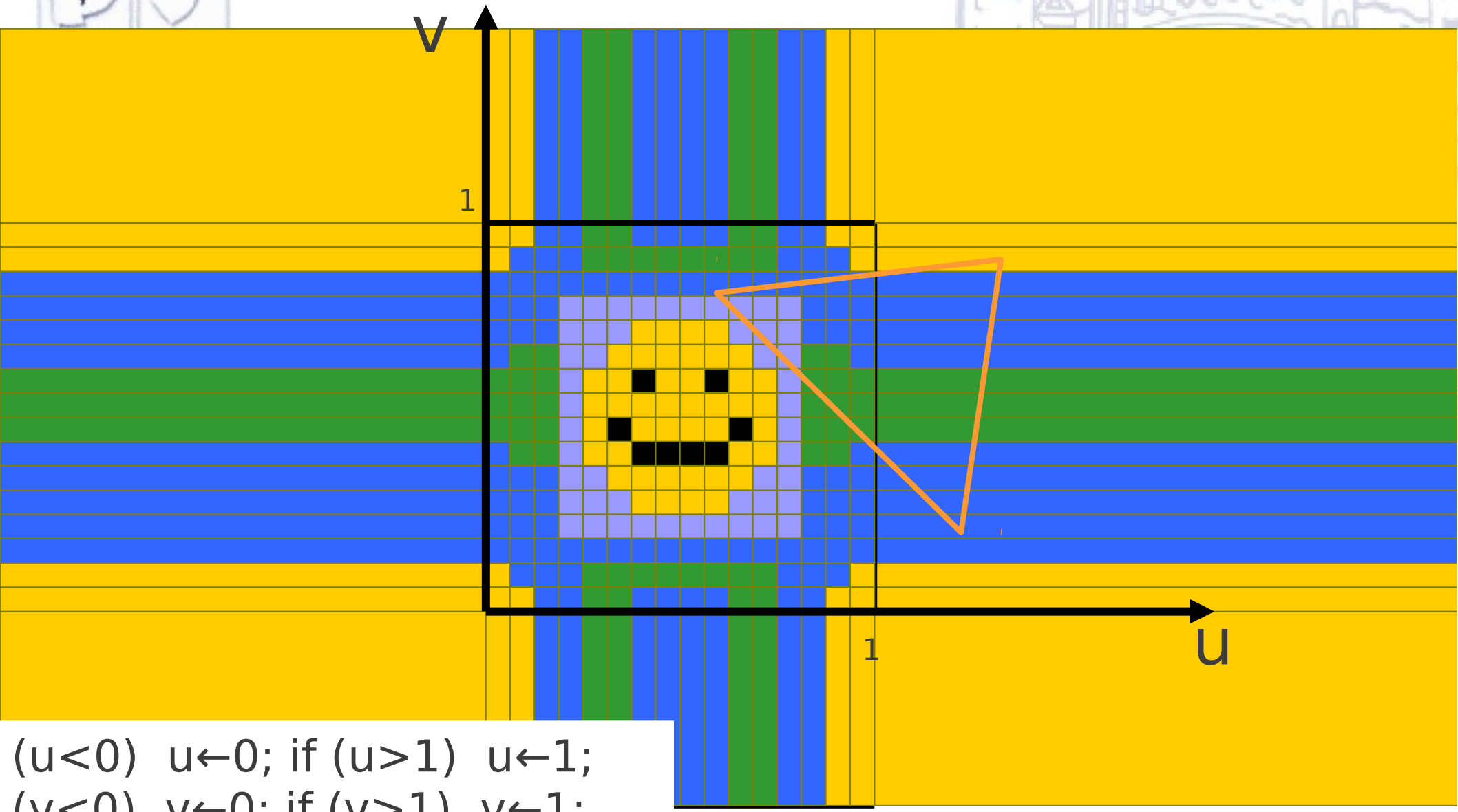
FIGURE 7.14 Texture mapping with a box.

- Per l'O-mapping ci sono varie scelte
  - Si considera la normale uscente da un punto dell'oggetto; il raggio che passa per tale punto e con direzione tale normale intersecherà la superficie intermedia in un punto, stabilendo così l'O-mapping
  - Anziché usare la normale si può usare la retta che congiunge il centroide dell'oggetto con il punto considerato
  - Si può considerare la normale in un punto della superficie intermedia e la retta che passa per tale punto e con direzione questa normale, intersecherà la superficie dell'oggetto in un punto, stabilendo un altro possibile O-mapping

Esempi di O-mapping.
(a) Usando la normale alla superficie intermedia.
(b) Usando la normale dalla superficie dell'oggetto.
(c) Usando i raggi dal centro dell'oggetto.
(©Angel)

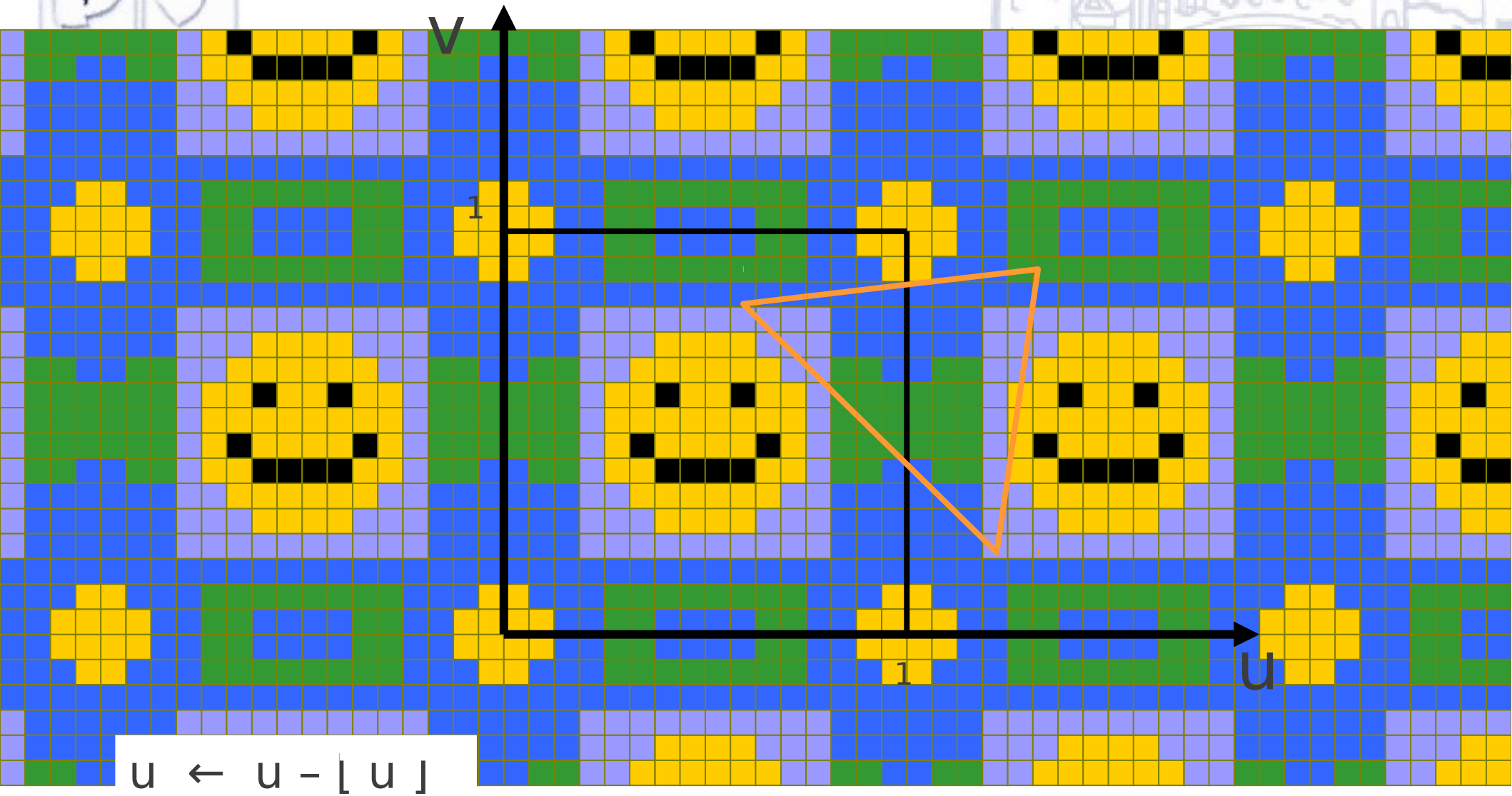


Texture fuori dai bordi: modo *clamp*



(u<0) u←0; if (u>1) u←1;
(v<0) v←0; if (v>1) v←1;

Texture fuori dai bordi: modo *repeat*



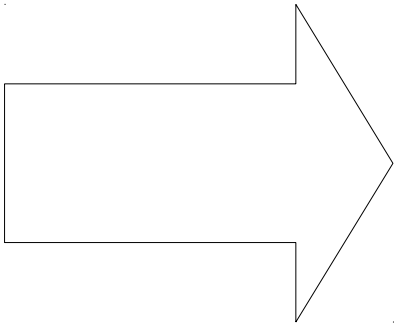
$u \leftarrow u - \lfloor u \rfloor$

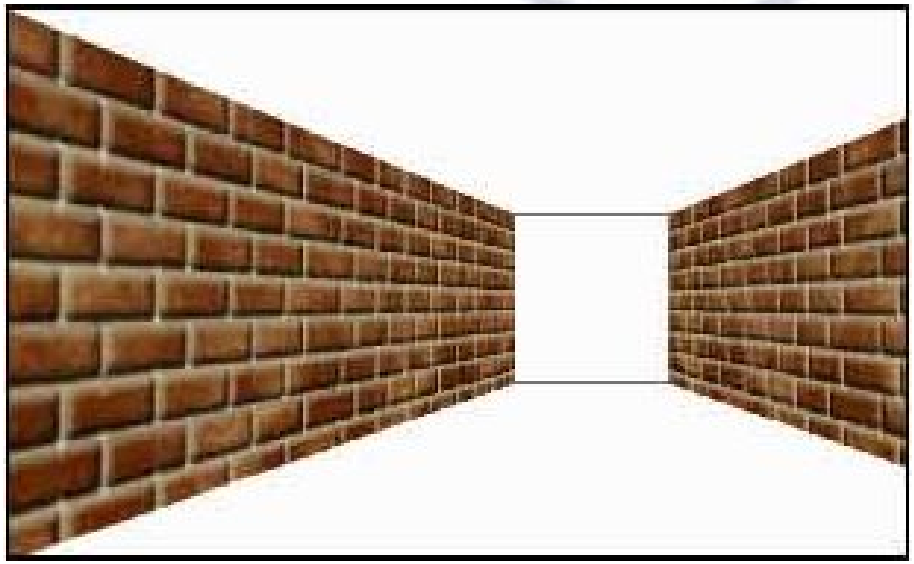
$v \leftarrow v - \lfloor v \rfloor$

Tessiture ripetute

- Tipico utilizzo:

Nota: deve essere TILABLE



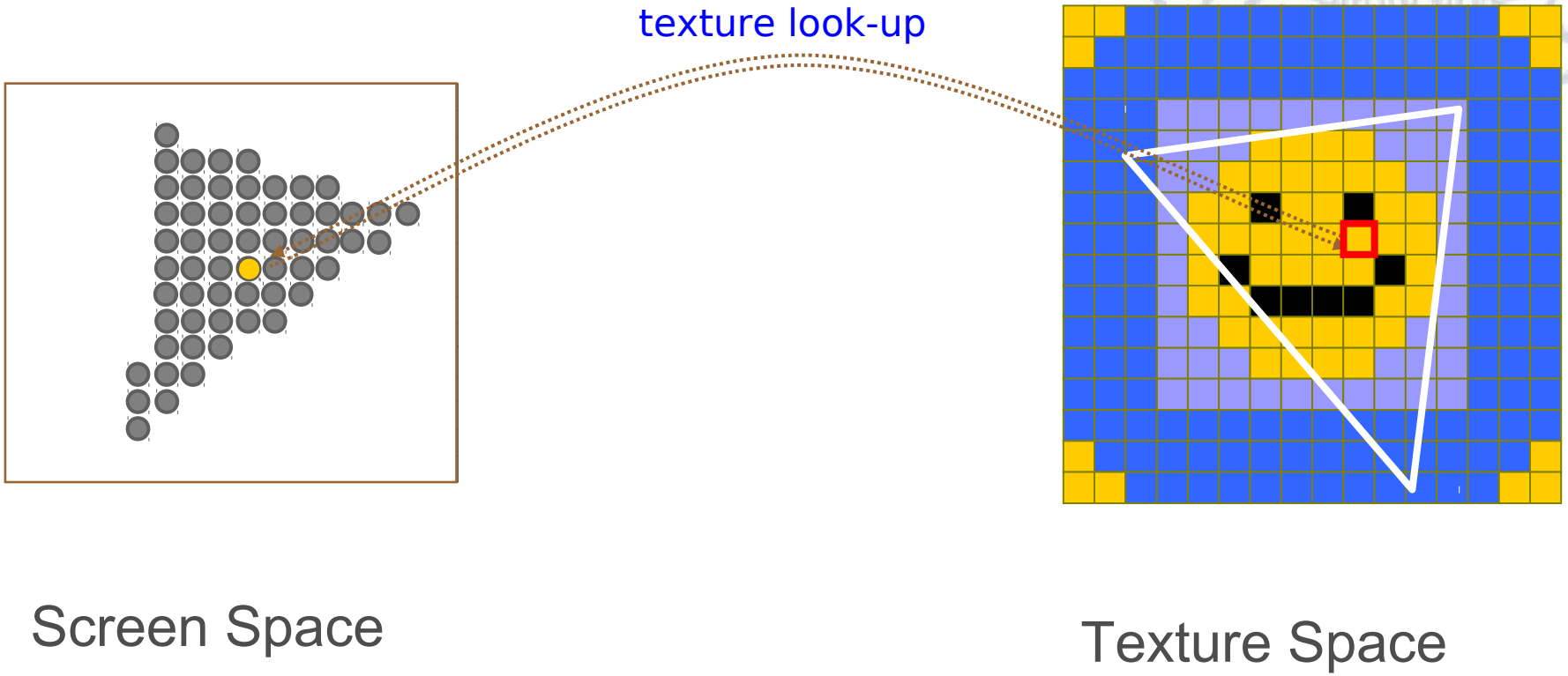


Molto efficiente in spazio: una sola texture mappa su molti triangoli

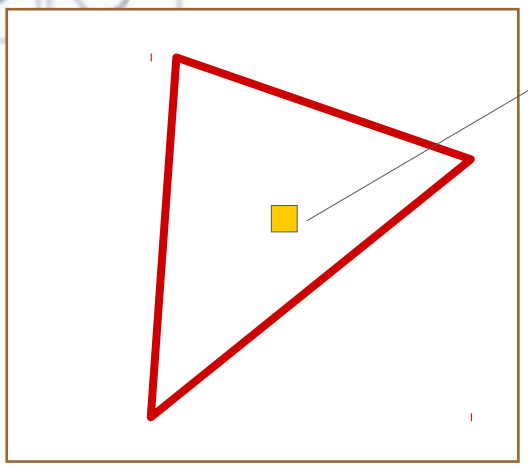
Texture Mapping

Texture Look-up

- Un frammento ha coordinate non intere (in texels)

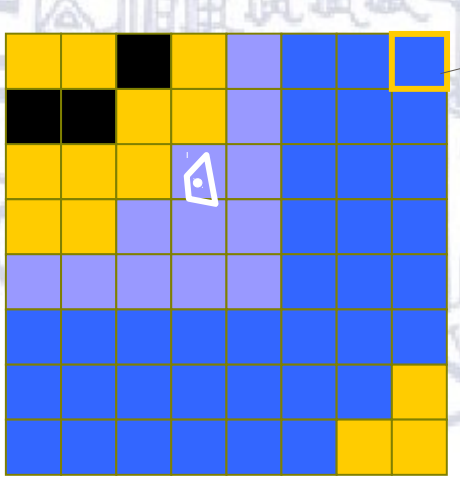


Texture Look-up



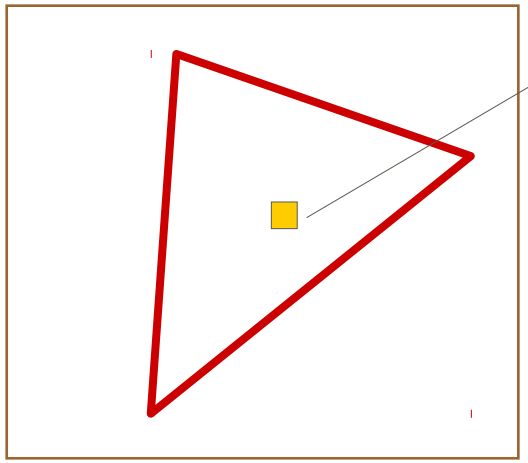
un pixel = meno di un texel

magnification



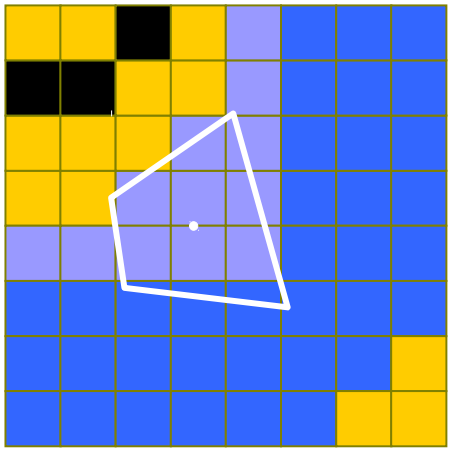
Screen Space

Texture Space



un pixel = più di un texel

minification



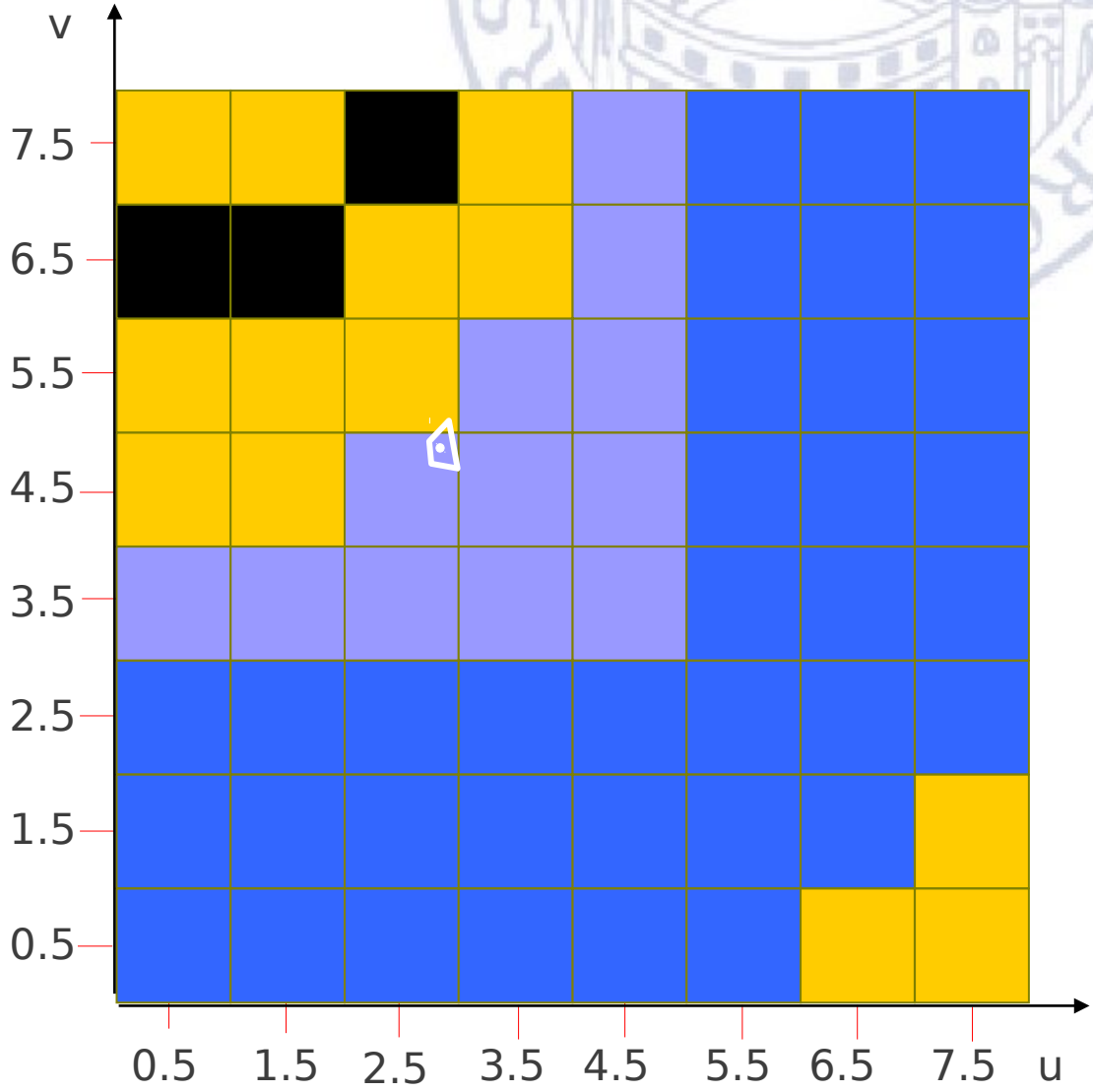
Caso Magnification

Soluzione 1:
prendo il texel in cui sono

(equivale a prendere
il texel più vicino)

equivale ad arrotondare
alle coordinate texel
intere

"Nearest Filtering"



Caso Magnification

Nearest Filtering: risultato visivo

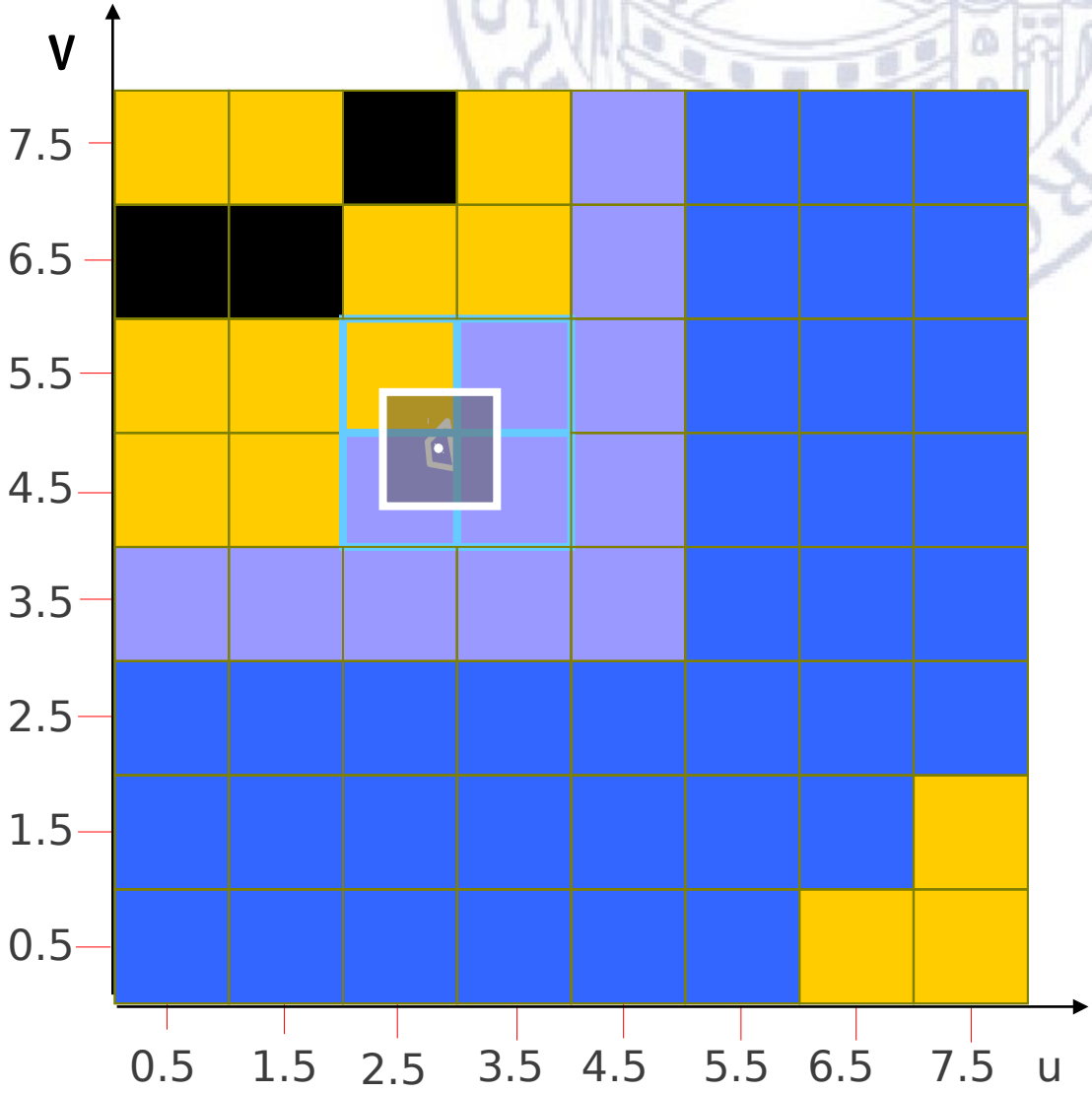
texture 128x128

"si vedono i texel !"

Caso Magnification

Soluzione 2:
Medio il valore dei quattro texel più vicini

Interpolazione Bilineare



Caso Magnification

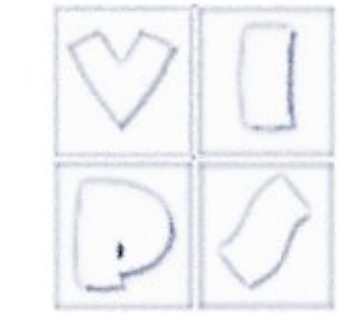
Bilinear Interpolation: risultato visivo

texture 128x128

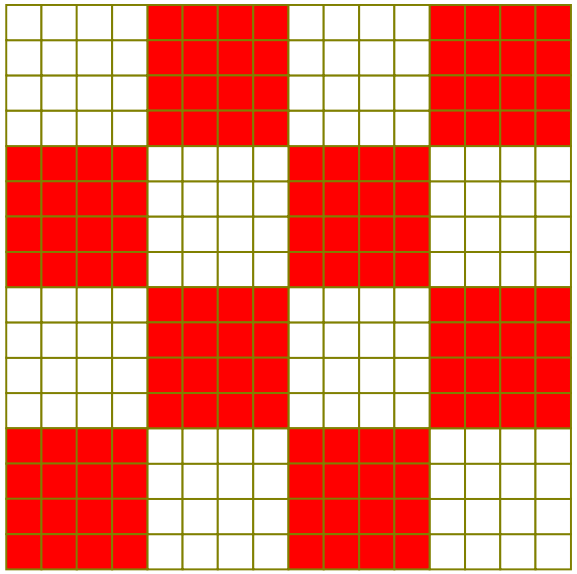
Caso Magnification

- Modo Nearest:
  - si vedono i texel
  - va bene se i bordi fra i texel sono utili
  - più veloce

- Modo Interpolazione Bilineare
  - di solito qualità migliore
  - può essere più lento
  - rischia di avere un effetto "sfuocato"



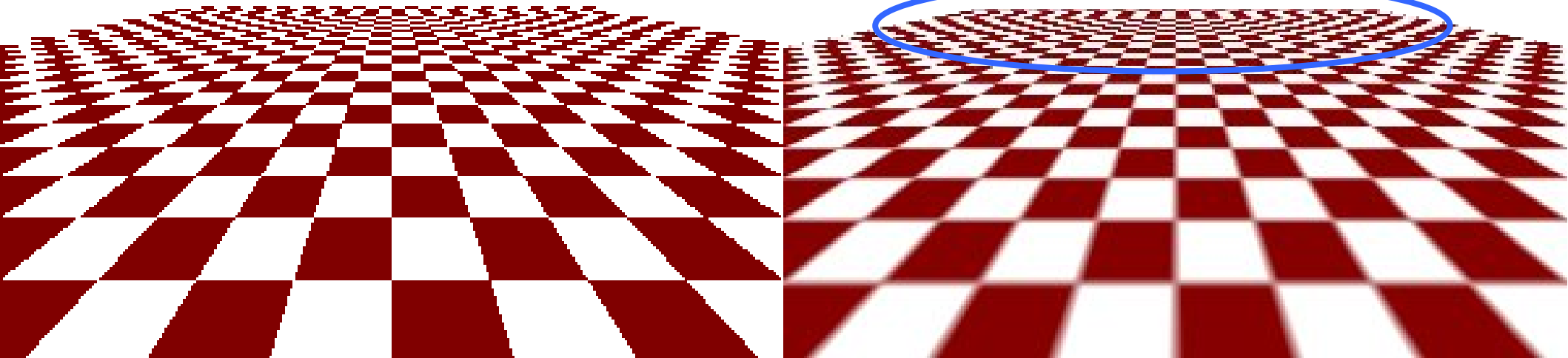
Caso Minification



Nearest Filtering

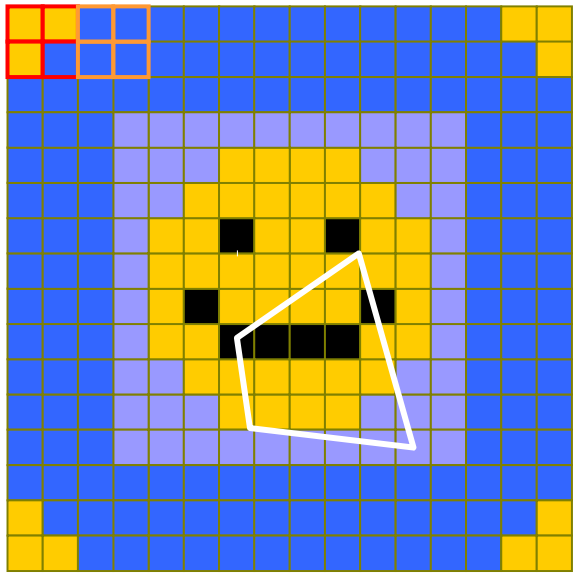
Bilinear interpolation

non risolve il problema

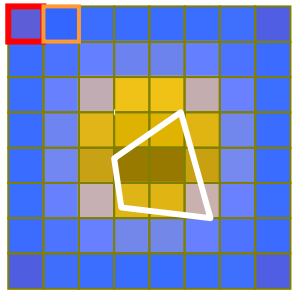


Caso Minification: MIP-mapping

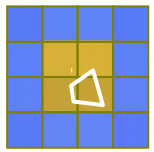
MIP-mapping: "Multum In Parvo"



MIP-map
level 0



MIP-map
level 1



MIP-map
level 2

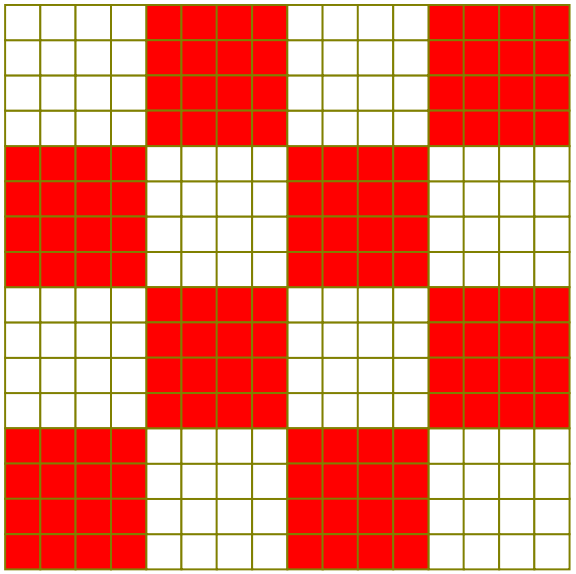
MIP-map
level 3

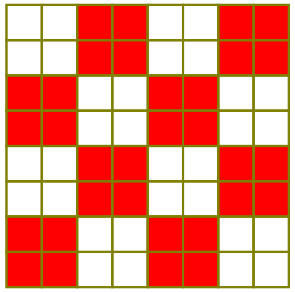
MIP-map
level 4
(un solo texel)

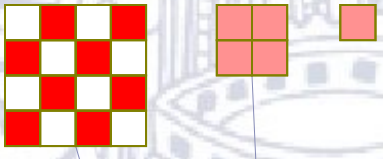
Mipmap Math

- Definiamo un fattore di scala, ρ=texels/pixel come valore massimo fra ρ_x e ρ_y, che può variare sullo stesso triangolo, può essere derivato dalle matrici di trasformazione ed è calcolato nei vertici, interpolato nei frammenti
- Il livello di mipmap da utilizzare è: $\log_2 \rho$ dove il livello 0 indica la massima risoluzione
- Il livello non è necessariamente un numero intero e può quindi essere arrotondato

Caso Minification: MIP-mapping



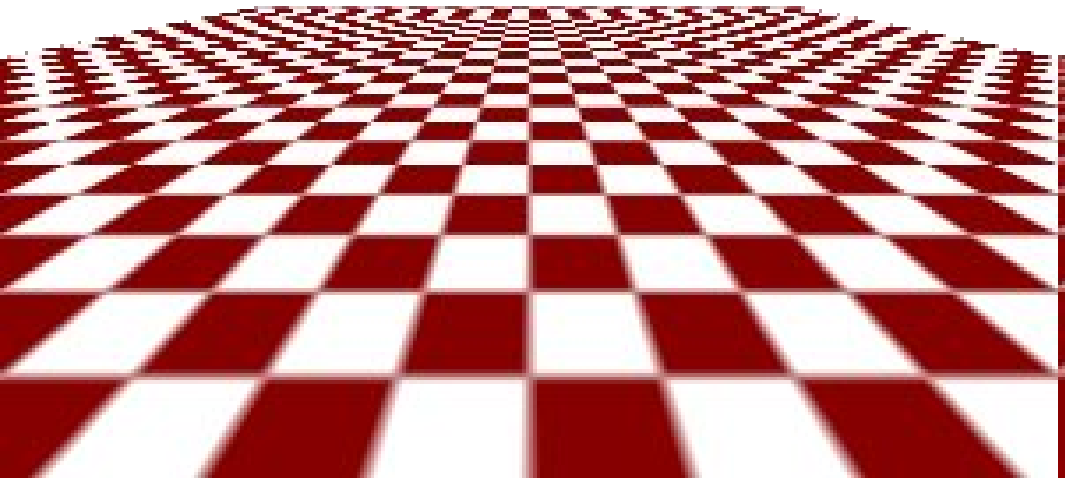


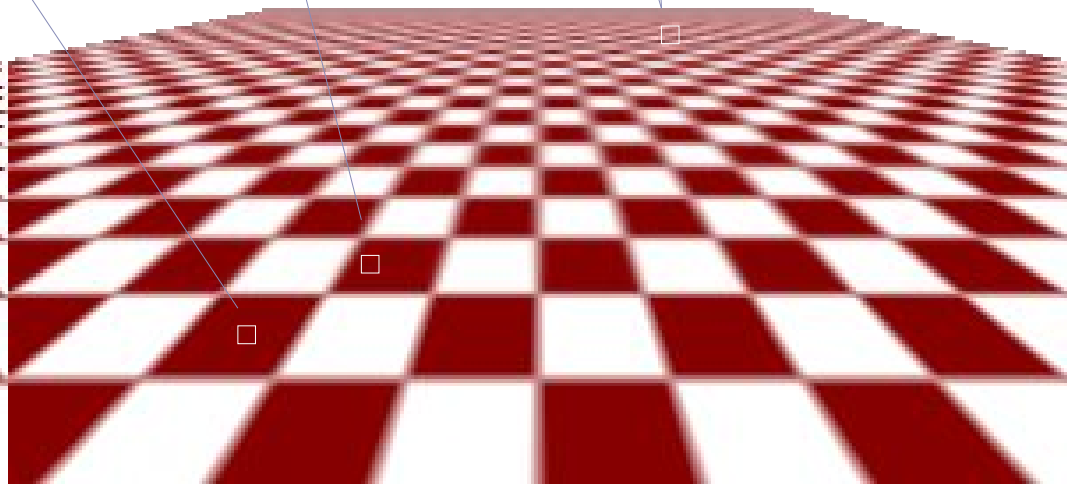


Bilinear interpolation

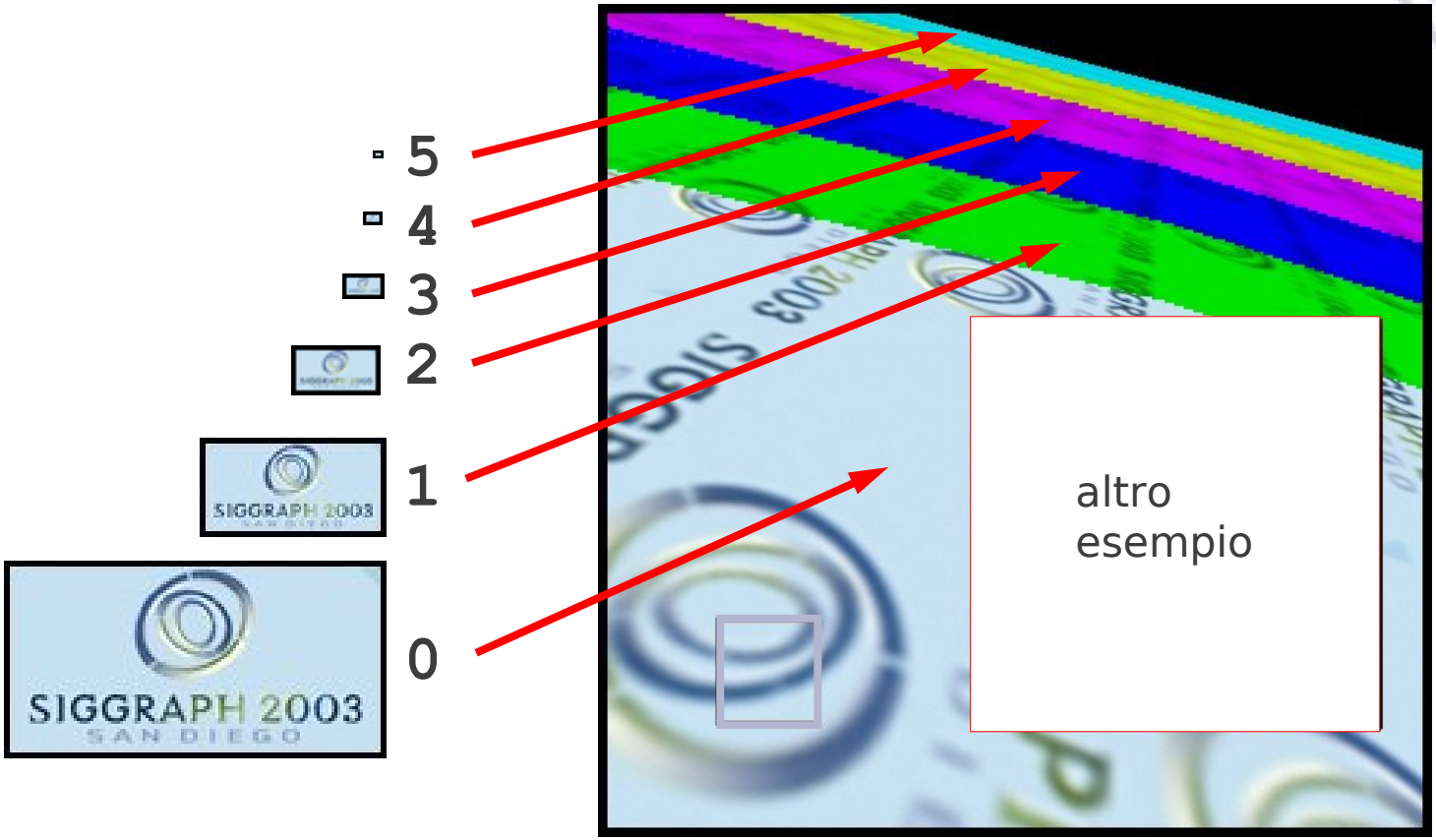
non risolve il problema

MIP-mapping



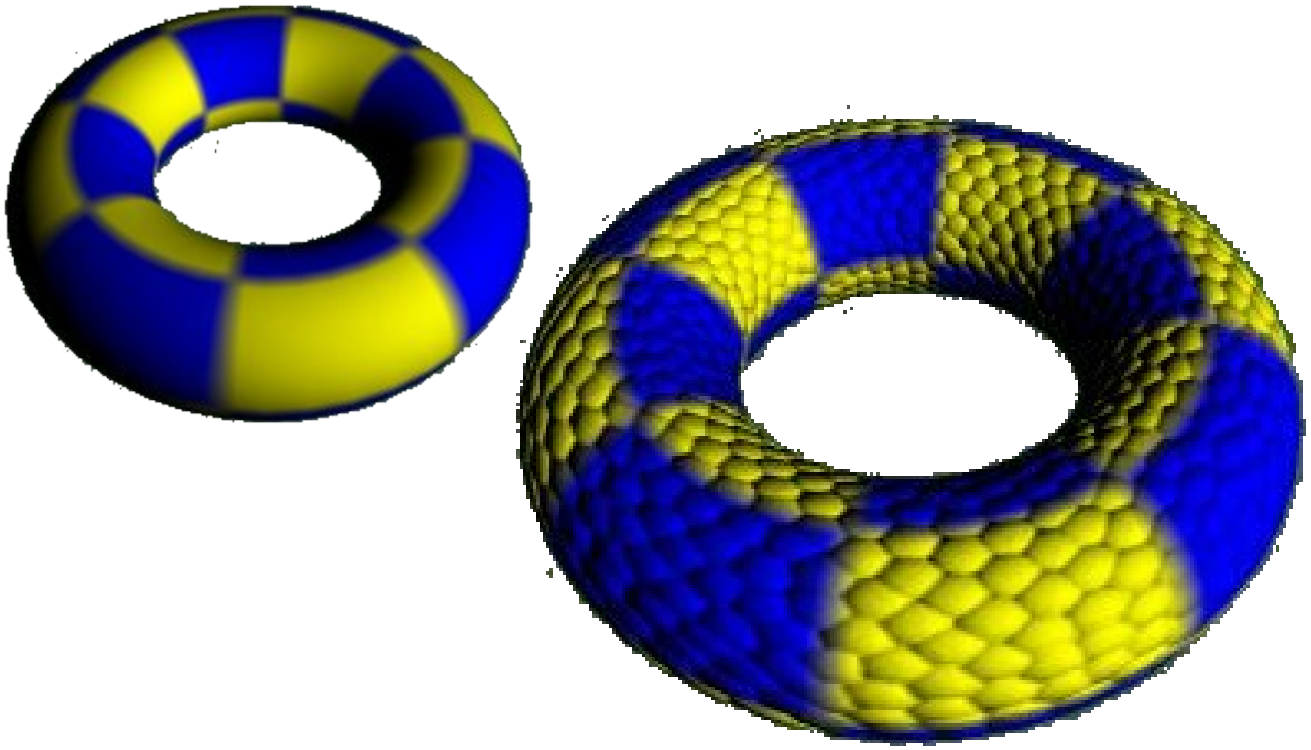


Caso Minification: MIP-mapping



Bump mapping

- Un ulteriore modifica all'apparenza del rendering può essere effettuata usando il bump mapping



Bump mapping

- Il metodo prevede di variare la normale alla superficie pixel per pixel utilizzando la formula:

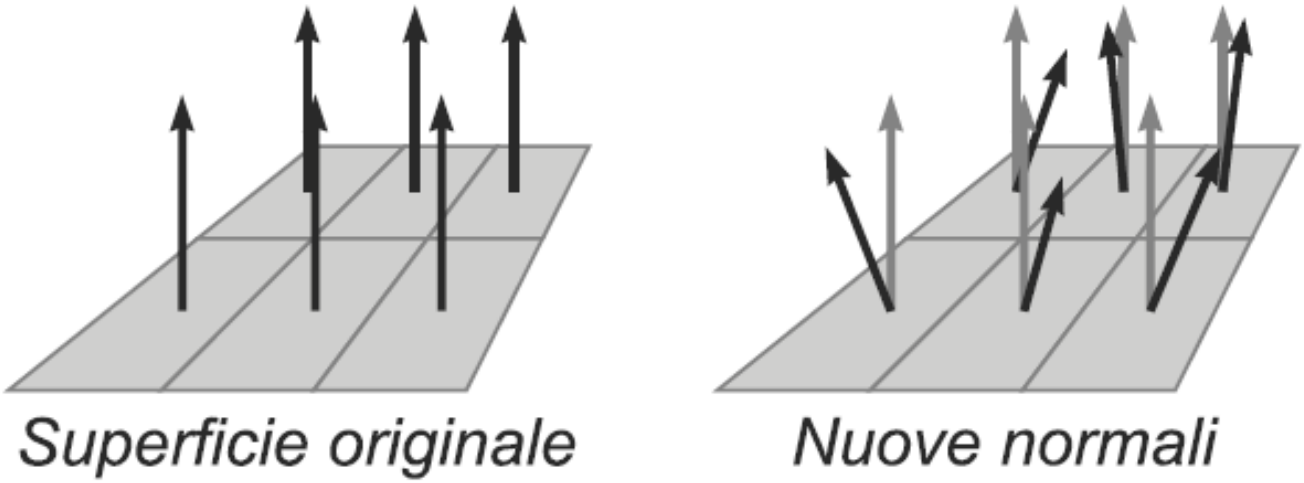
$$\vec{\mathbf{N}}_{new} = \vec{\mathbf{N}}_{old} + \vec{\mathbf{D}};$$

- I texel in questo caso sono utilizzati ad uno stadio diverso rispetto ai color texel, prima del calcolo dell'equazione di illuminazione

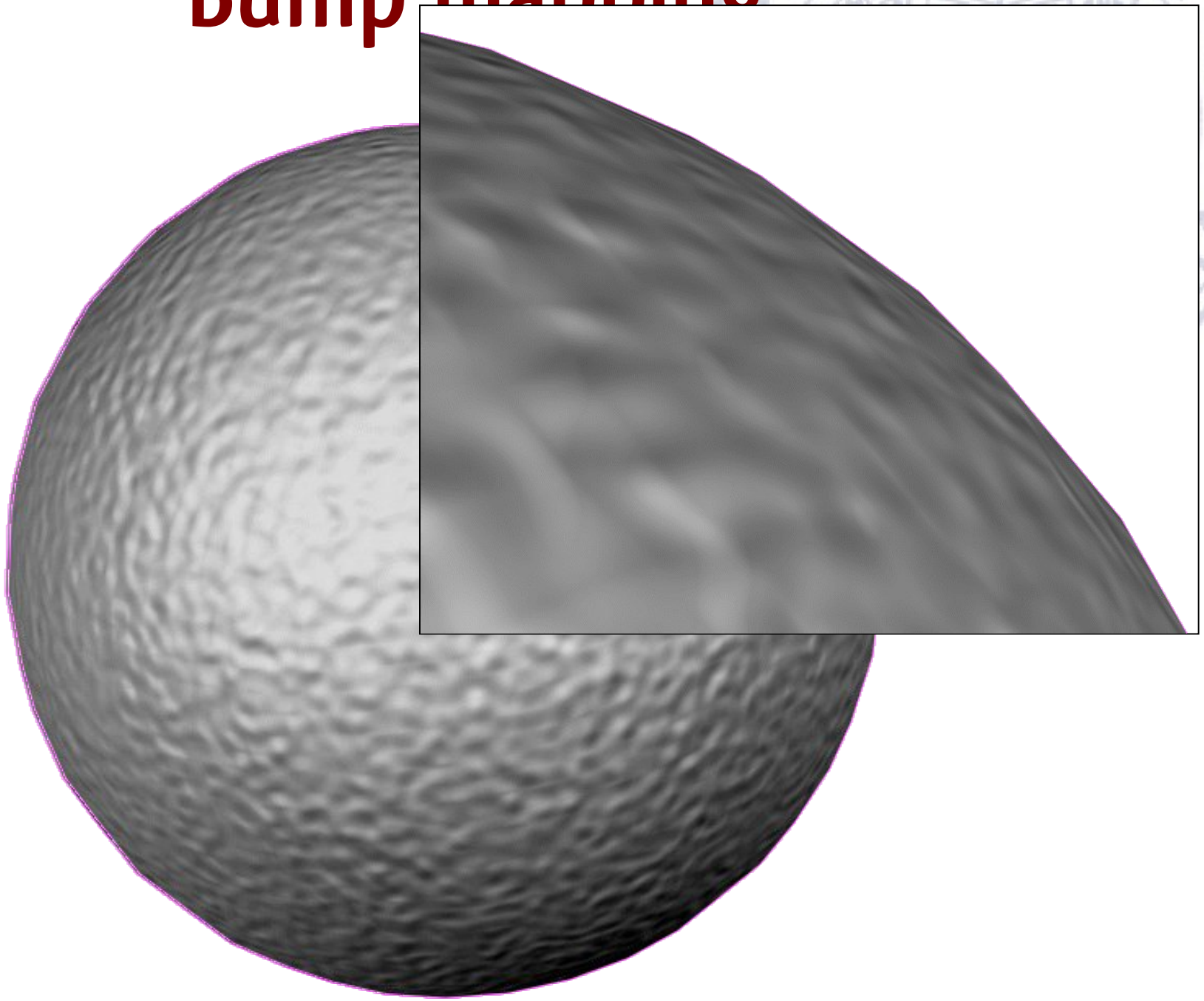
$$\vec{\mathbf{D}} = (\Delta x, \Delta y, \Delta z)$$

Bump mapping

- L'effetto che si ottiene è una perturbazione del valore delle normali che altera il rendering senza modificare la geometria



Bump mapping



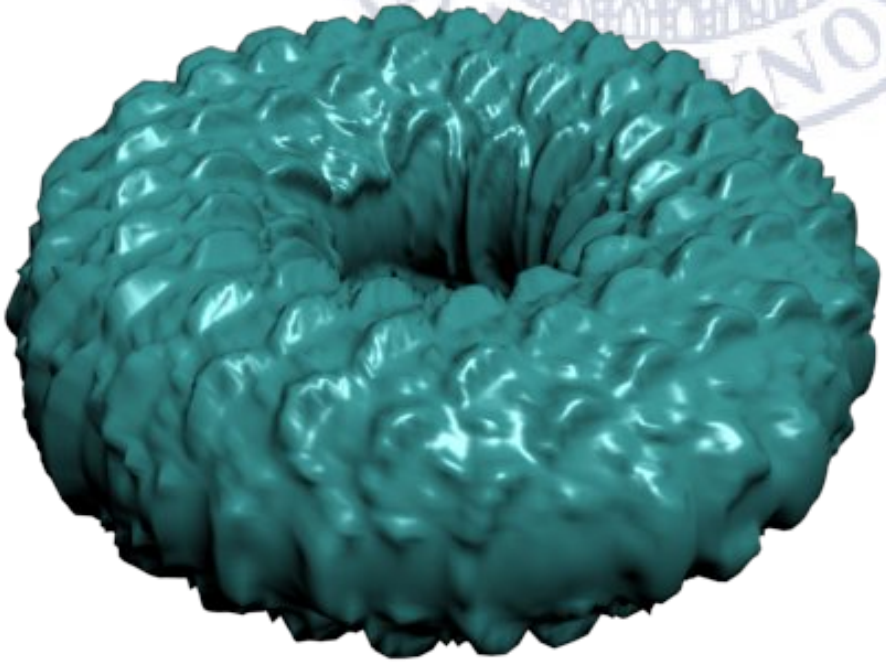
Displacement mapping

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

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

- Il displacement m di rendering e non modifica stabilme na
- Rispetto al bump mapping anche la silhouette del modello mostra le corrette deformazioni

Displacement mapping



Riferimenti

- Scateni et al. Cap 6.4-6.5
- Angel cap. 7.4

Domande di verifica

- Perché il texture mapping si fa tipicamente con mappatura inversa?
- Che differenza c'è tra gli artefatti nella mappatura delle tessiture che possono verificarsi nel caso si usi la proiezione prospettica o quella ortografica per creare la scena?
- Si indichino differenti applicazioni del texture mapping
- Qual è la differenza tra bump mapping e displacement mapping?