Grafica Computazionale

Texturing

Fabio Ganovelli
fabio.ganovelli@isti.cnr.it
a.a. 2005-2006

Texture Mapping

- Nelle operazioni per frammento si può accedere ad una RAM apposita
  - la Texture RAM
  - strutturata in un insieme di Textures ("tessiture")

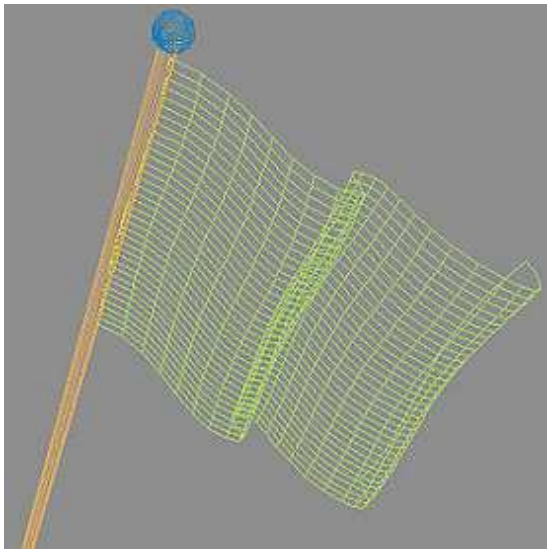
- Ogni tessitura è un array

  1D, 2D o 3D

  di Texels (campioni di tessitura)

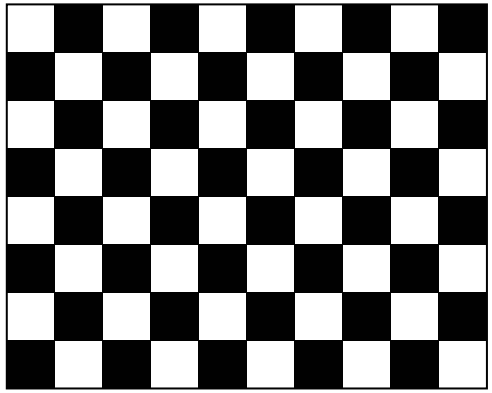
  dello stesso tipo

Tipica applicazione:
rimappare immagini sulla geometria



geometria 3D
(mesh di quadrilateri)

+

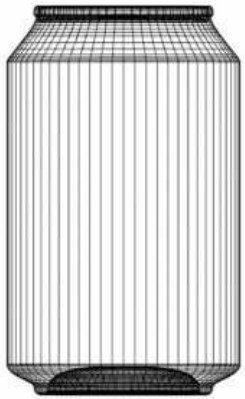


RGB texture 2D

(color-map)

=

Altro esempio

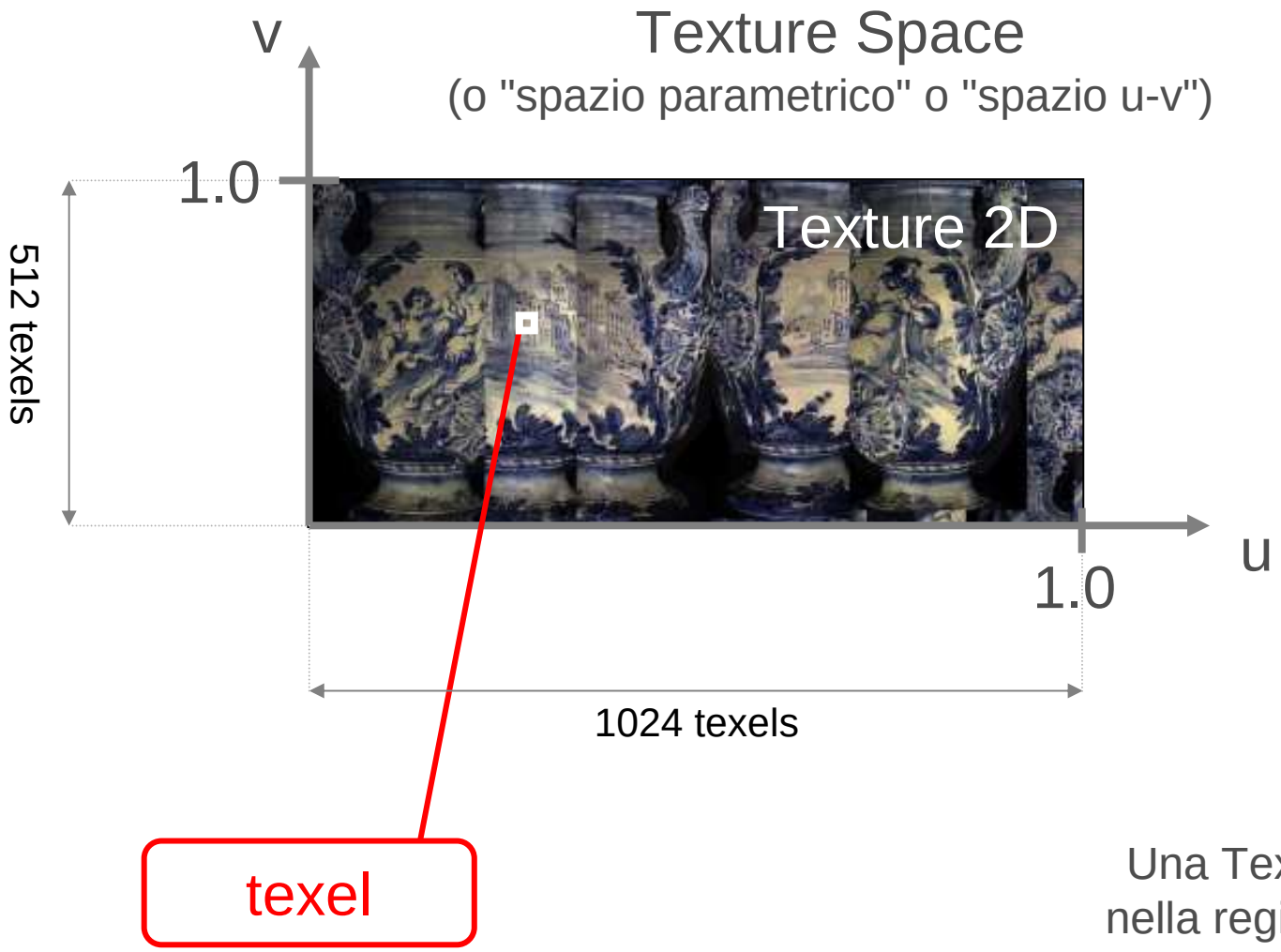


Altro esempio

+

=

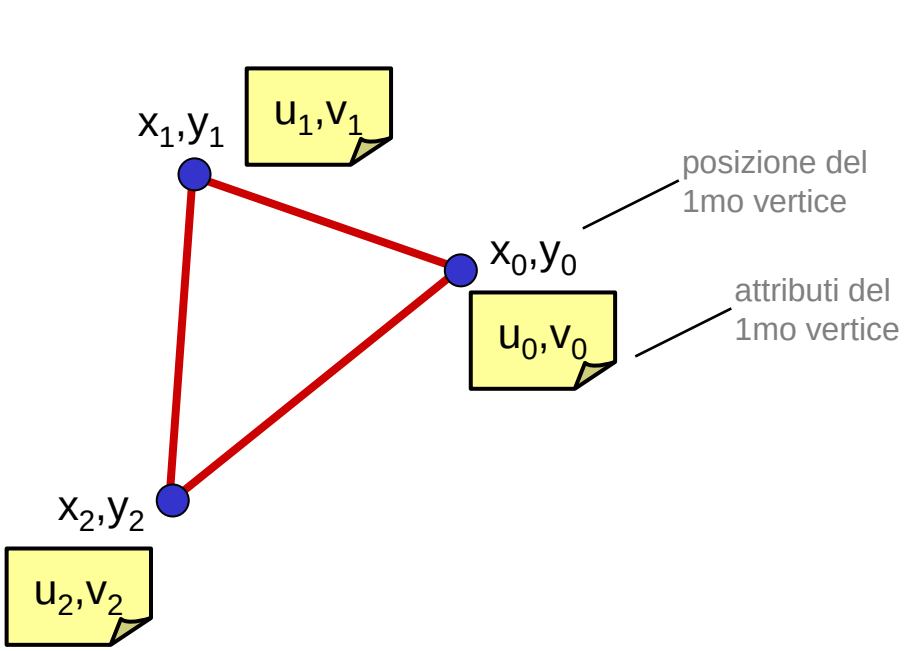
Notazione



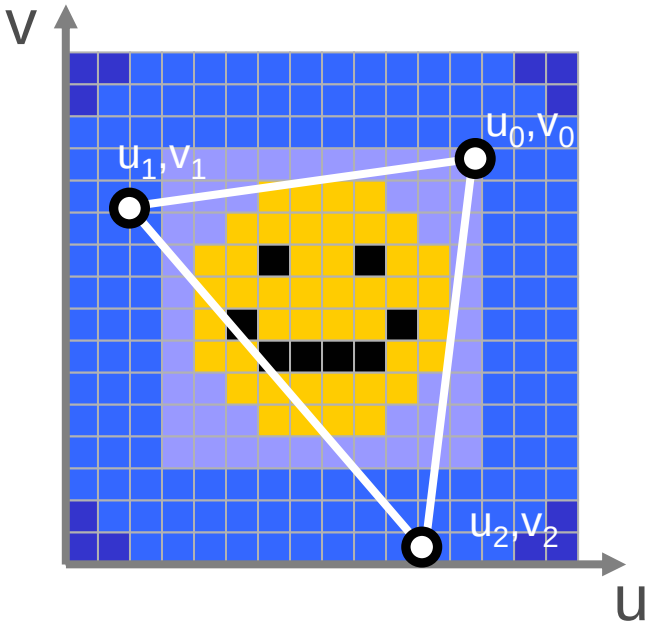
Una Texutre e' definita
nella regione [0,1] x [0,1]
dello "spazio parametrico"

Texture Mapping

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



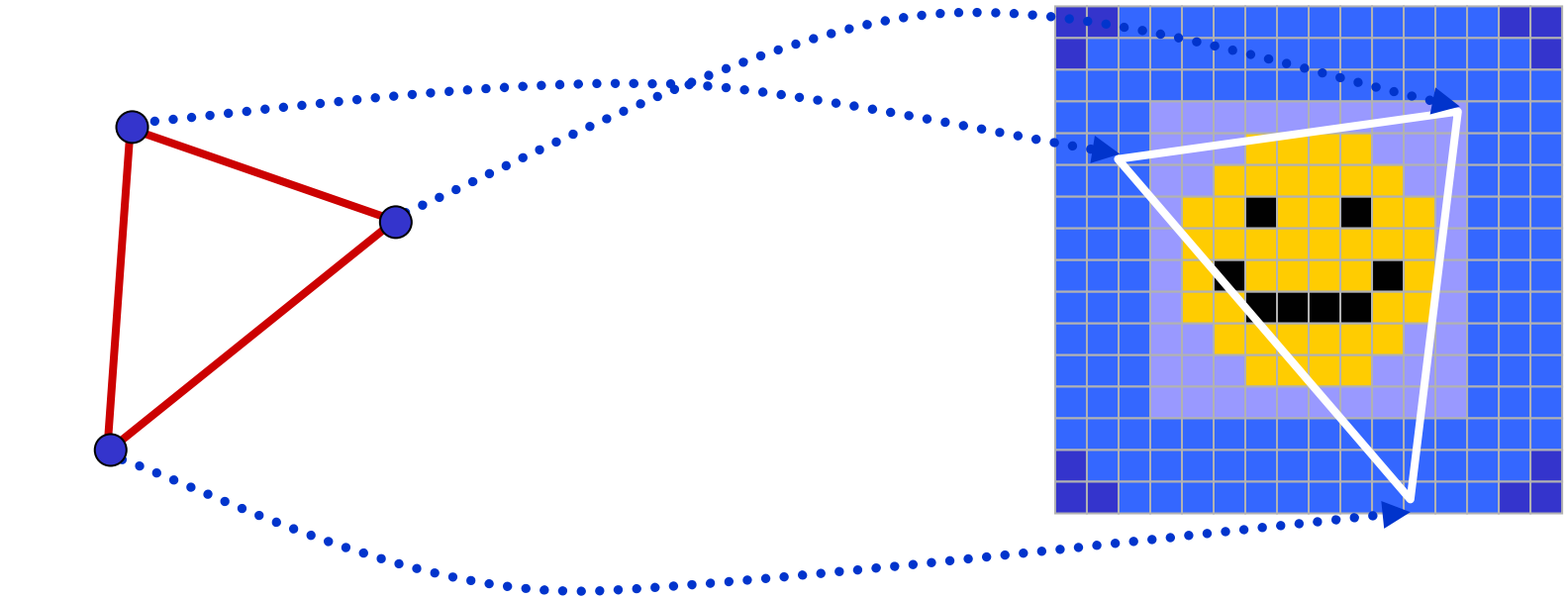
Screen Space



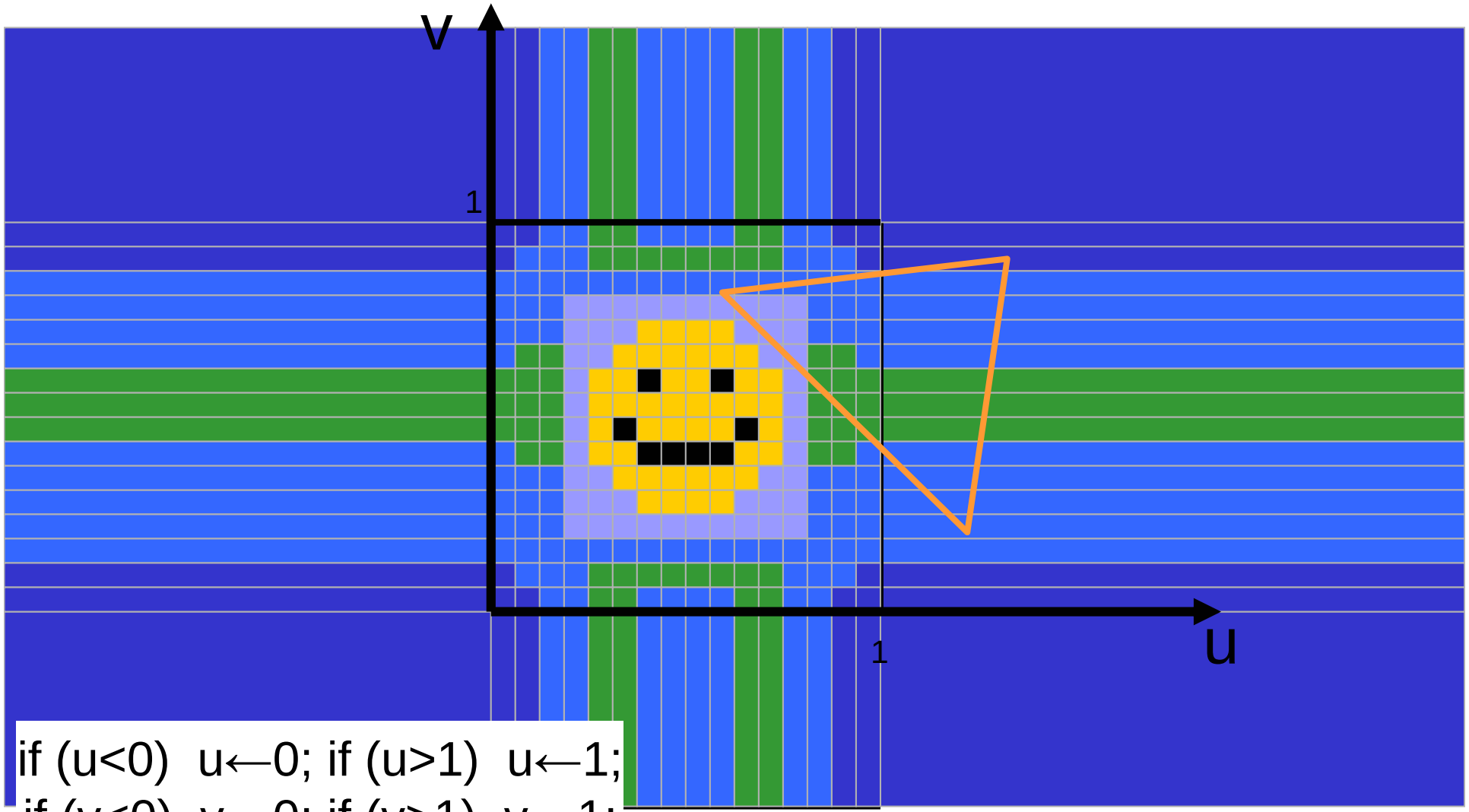
Texture Space

Texture Mapping

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

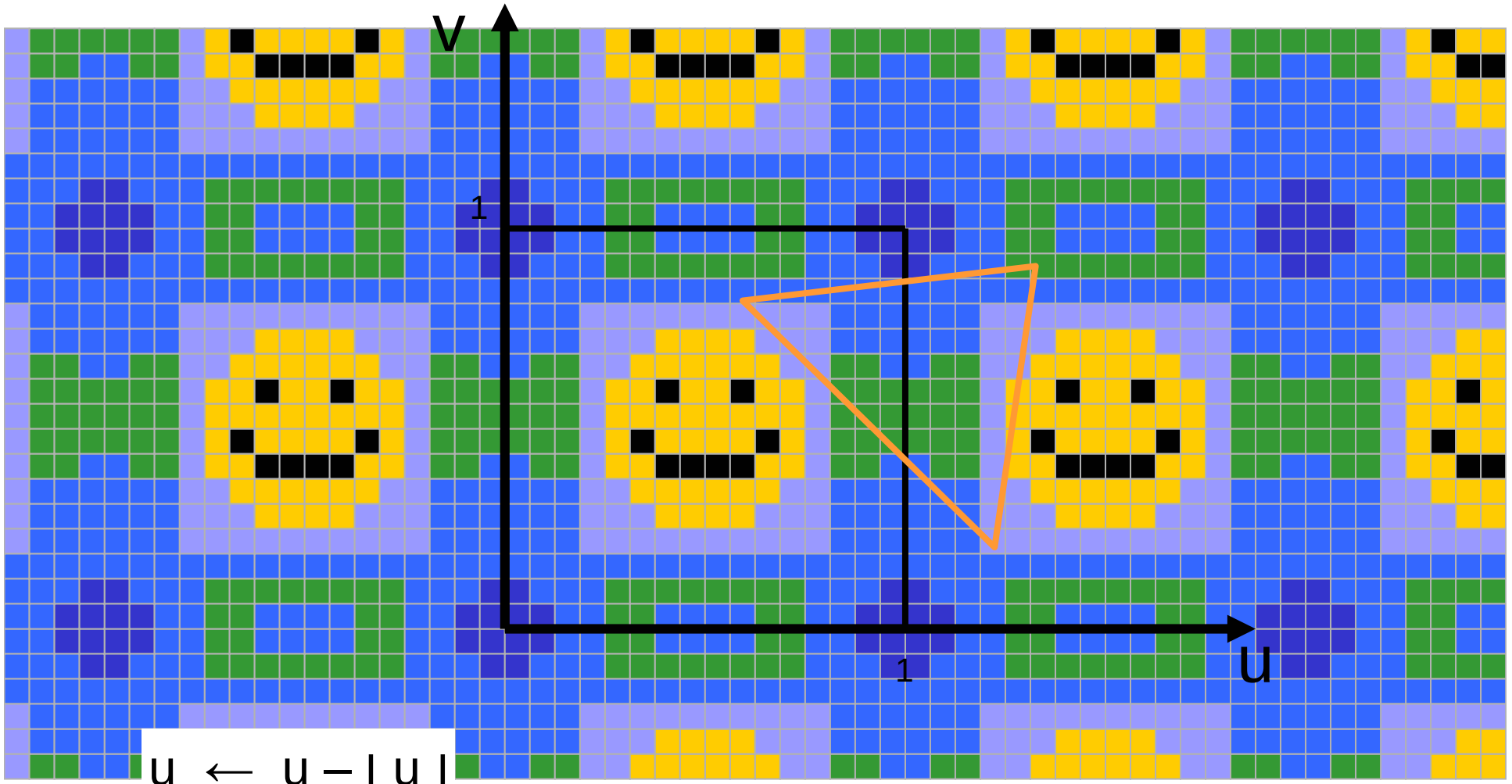


Texture Look-up:
fuori dai bordi: modo "clamp"



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

Texture Look-up:
fuori dai bordi: modo "repeat"



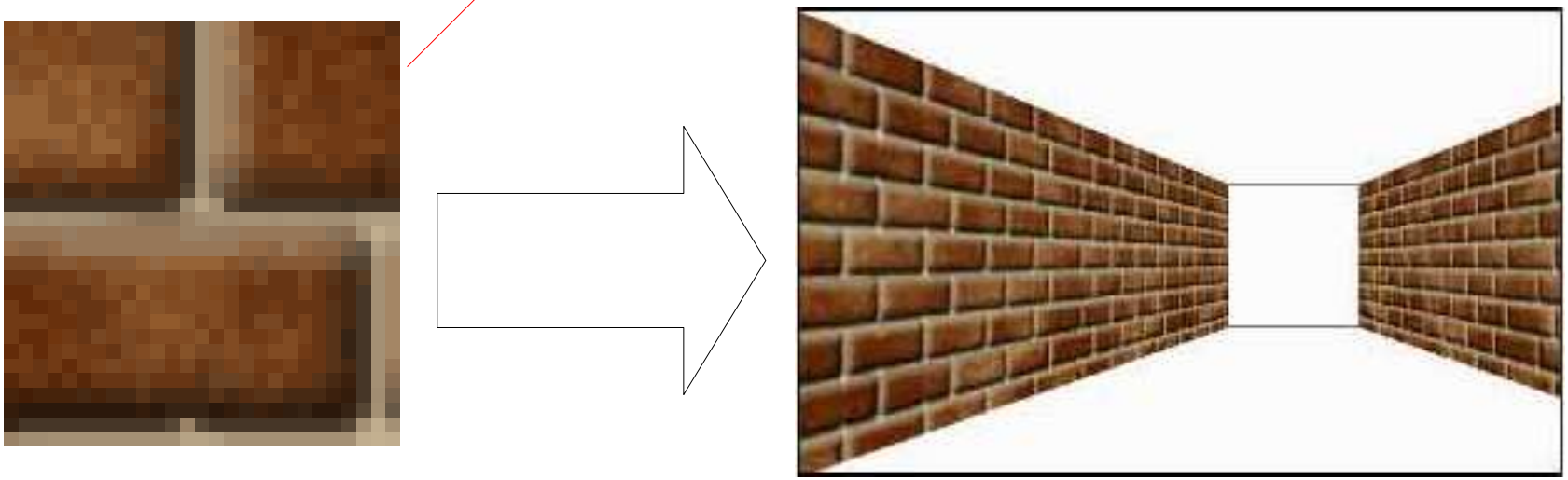
$u \leftarrow u - \lfloor u \rfloor$

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

Tessiture **ripetute**

- Tipico utilizzo:

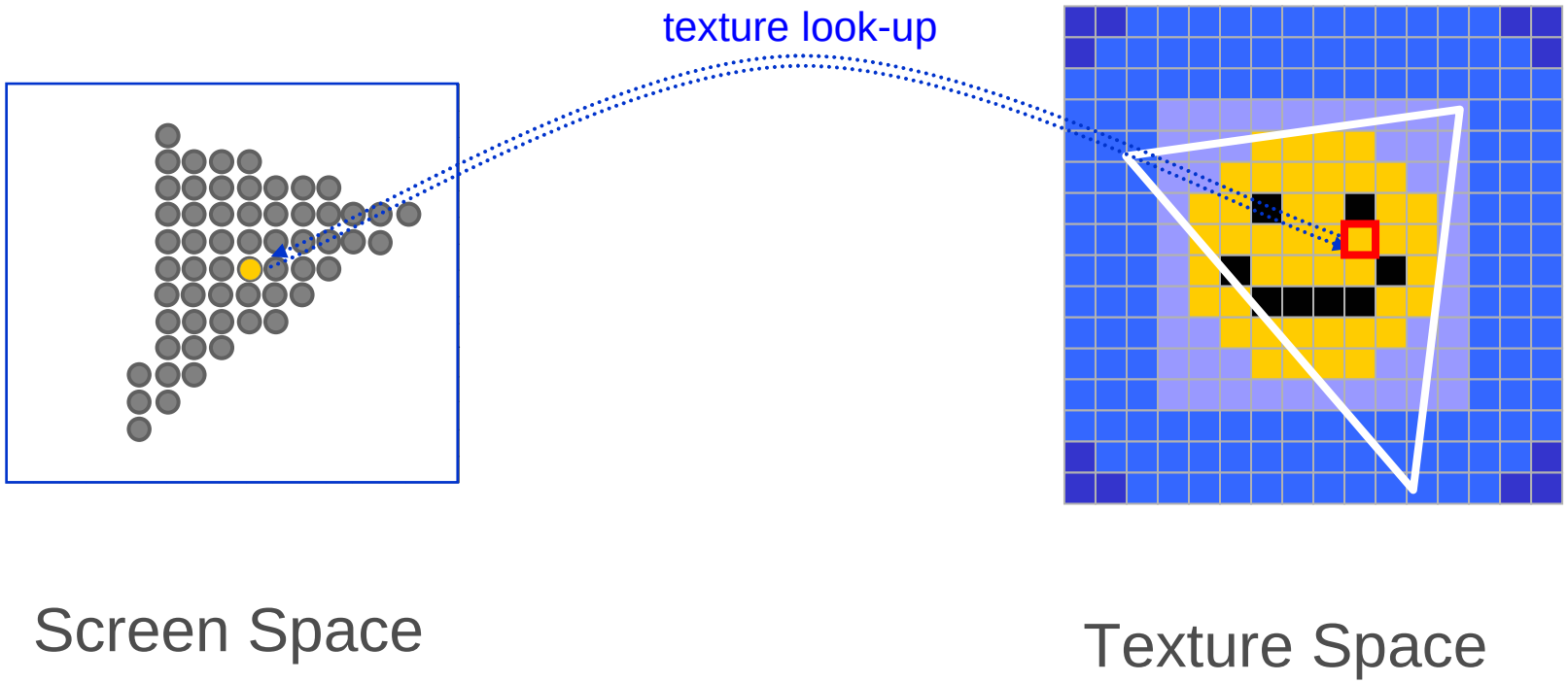
Nota: deve essere TILABLE



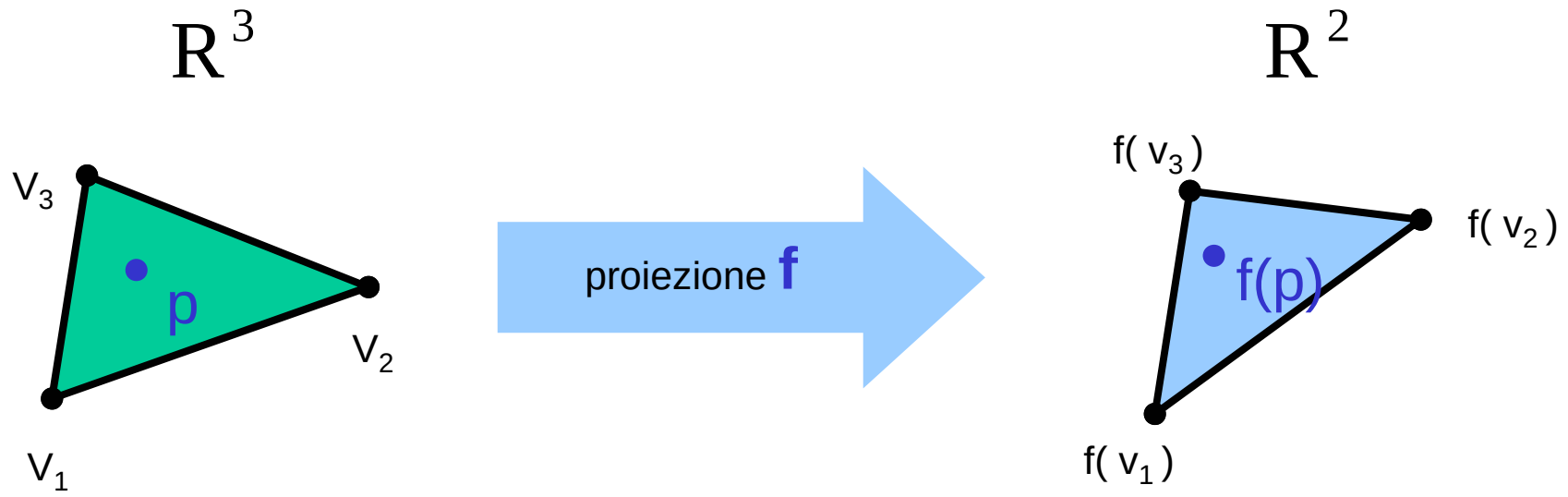
Molto efficiente in spazio!

Texture Mapping

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



Problema: interpolazione lineare coordinate texture



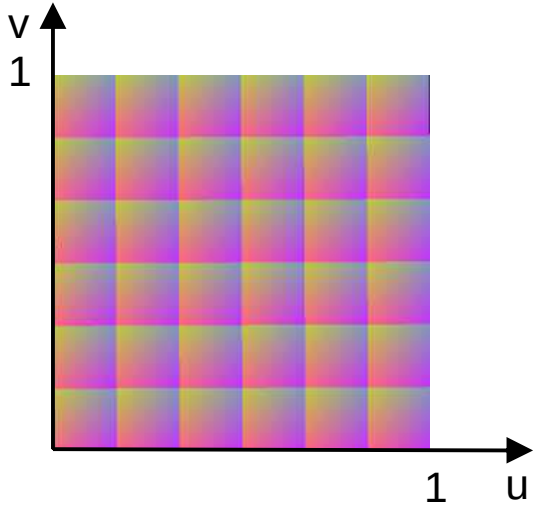
p ha coord. baricentriche a,b,c nel triangolo v_1 v_2 v_3

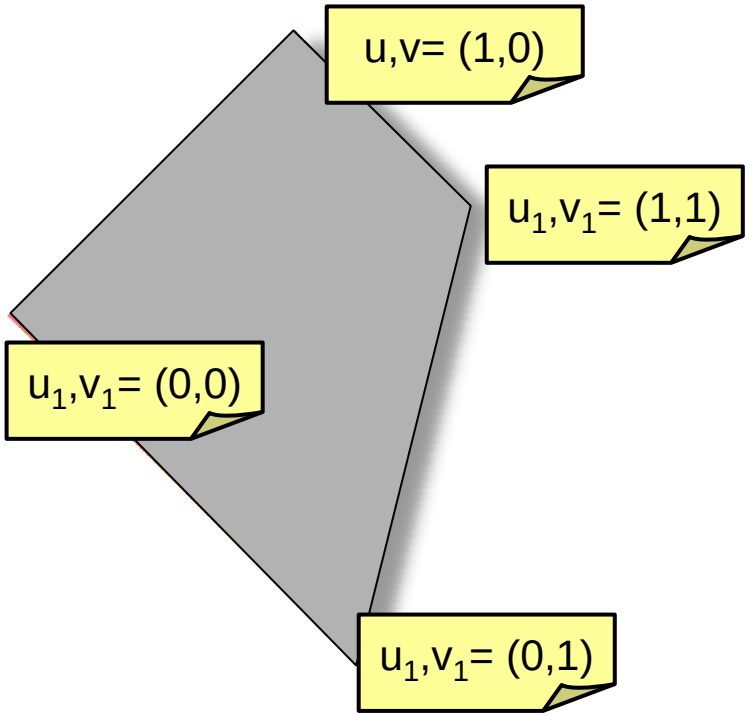
f(p) ha coord. baricentriche a,b,c nel triangolo $f(v_1)$ $f(v_2)$ $f(v_3)$

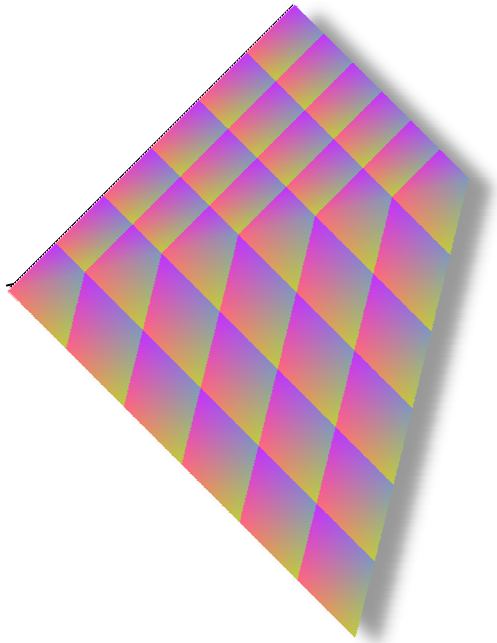
- Non vale per la proiezione prospettica!
  - era solo una approssimazione
  - andava bene quando interpolavamo colori, normali
  - non va bene quando interpoliamo coordinate texture...

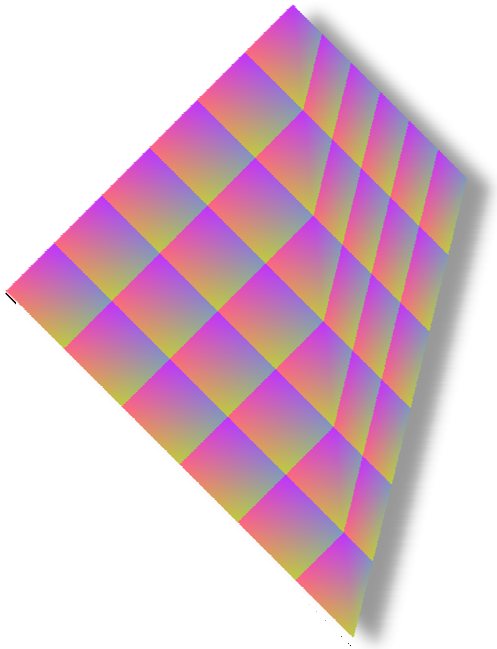
Problema: interpolazione lineare coordinate texture

- Esempio:



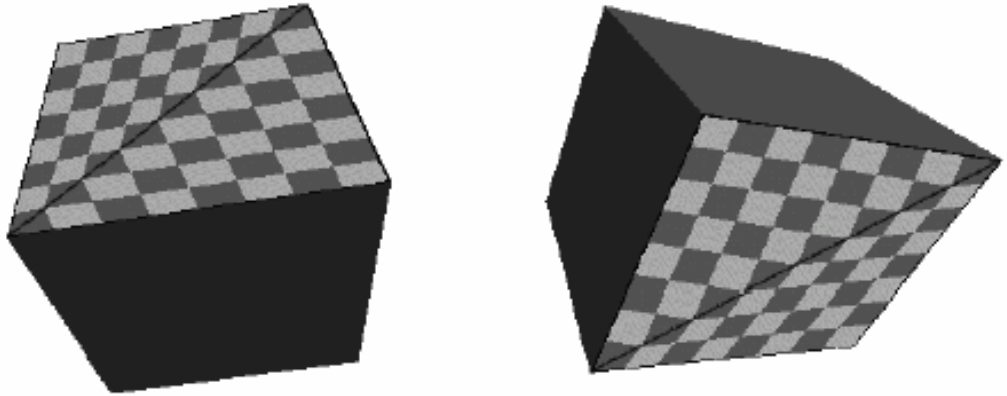






Problema:
interpolazione lineare coordinate texture

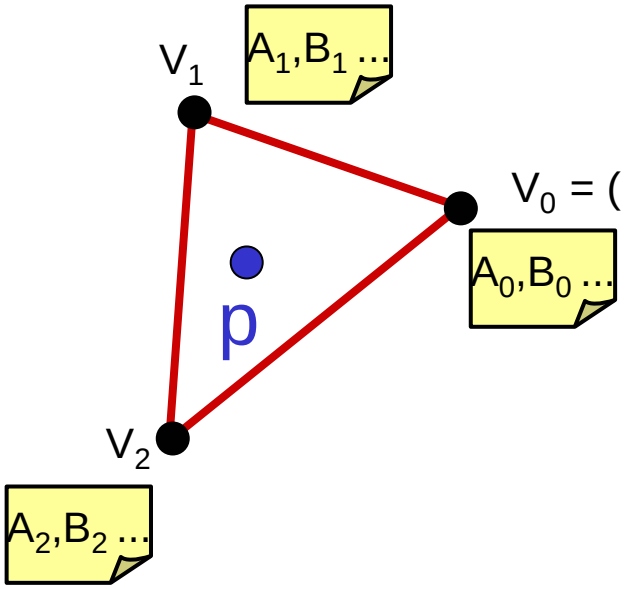
- Esempio:



Soluzione:
Correzione Prospettica

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

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



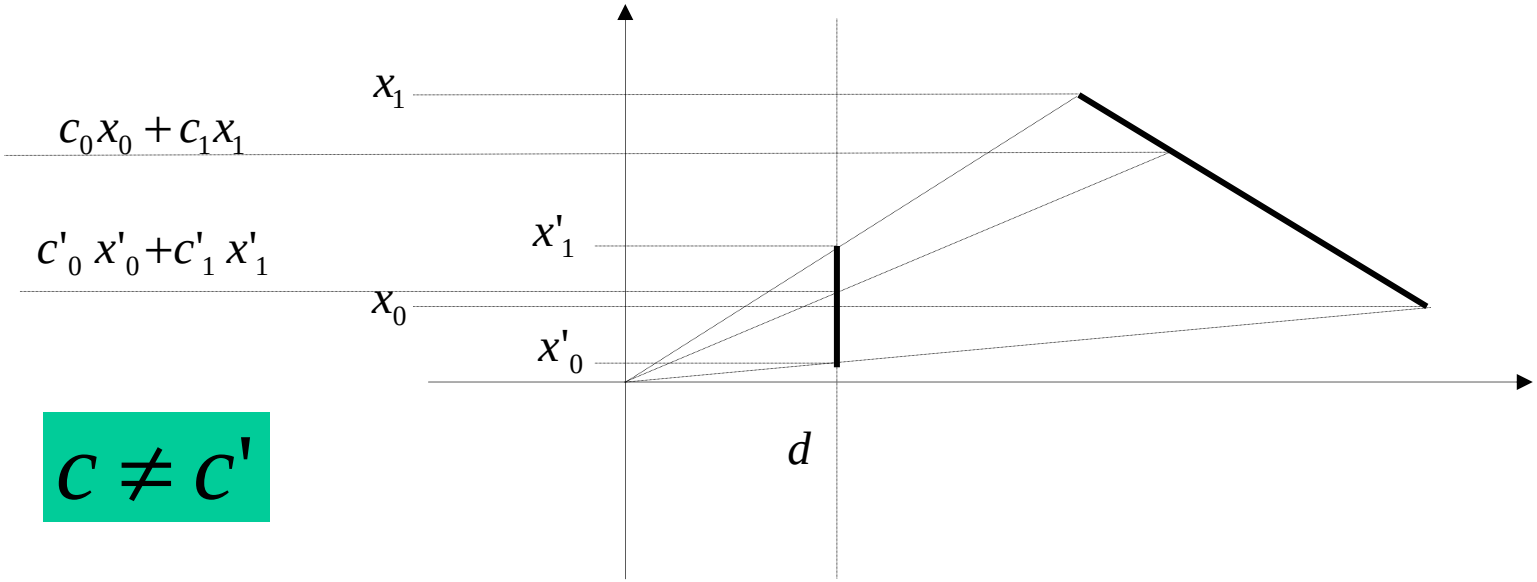
attributi di p:
(senza considerare
la "correzione prospettica")

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

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

Perchè non vale?

- La proiezione prospettica non è una trasformazione affine: non preserva i rapporti fra le distanze
- Esempio per un segmento:



In generale $c \neq c'$

Problema: quale punto del segmento proietta su del segmento proiettato?

$c_0 x_0 + c_1\, x_1$

$c'_0\, x'_0 + c'_1\, x'_1$

Qualche passaggio …

$$\begin{pmatrix} c'_0\, x'_0 + c'_1\, x'_1 \\ 1 \end{pmatrix} = \begin{pmatrix} c'_0 \dfrac{x_0}{(x_{0_z}/d)} + c'_1 \dfrac{x_1}{(x_{1_z}/d)} \\ 1 \end{pmatrix}$$

Ricordiamoci che $W_0 = x_{0_z}/d$ *e* $W_1 = x_{1_z}/d$

$$= \begin{pmatrix} \dfrac{c'_0}{W_0} x_0 + \dfrac{c'_1}{W_1} x_1 \\ 1 \end{pmatrix}$$

Ricordiamoci che $\begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} xw \\ w \end{pmatrix}$

$$= \begin{pmatrix} \dfrac{c'_0}{W_0} x_0 + \dfrac{c'_1}{W_1} x_1 \\ 1 \end{pmatrix} \cdot \frac{1}{\dfrac{c'_0}{W_0} + \dfrac{c'_1}{W_1}} = \begin{pmatrix} \dfrac{\dfrac{c'_0}{W_0} x_0 + \dfrac{c'_1}{W_1} x_1}{\dfrac{c'_0}{W_0} + \dfrac{c'_1}{W_1}} \\ \dfrac{1}{\dfrac{c_1}{W_0} + \dfrac{c_1}{W_1}} \end{pmatrix} = \begin{pmatrix} \dfrac{\dfrac{c'_0}{W_0}}{\dfrac{c'_0}{W_0} + \dfrac{c'_1}{W_1}} x_0 + \dfrac{\dfrac{c'_1}{W_1} x_1}{\dfrac{c'_0}{W_0} + \dfrac{c'_1}{W_1}} x_1 \\ \dfrac{1}{\dfrac{c_1}{W_0} + \dfrac{c_1}{W_1}} \end{pmatrix}$$

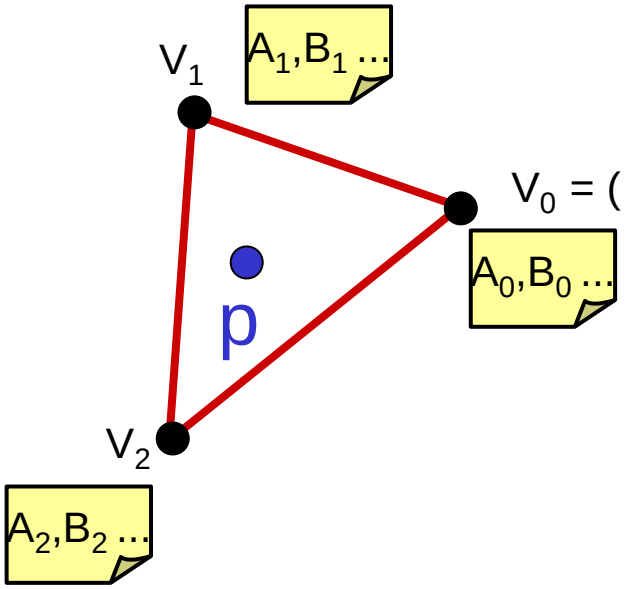
c_0 c_1

Abbiamo espresso le coordinate baricentriche del punto $c_0 x_0 + c_1 x_1$ in termini delle coordinate baricentriche della sua proezione!

Soluzione: Correzione Prospettica

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

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

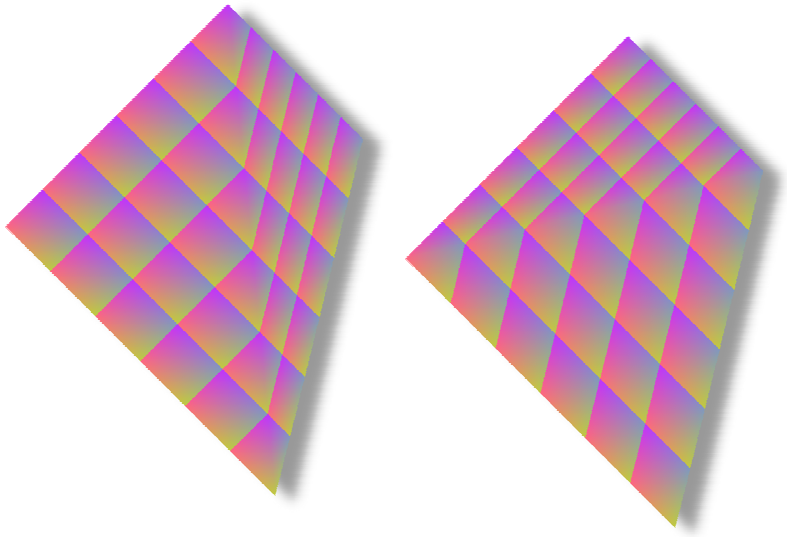


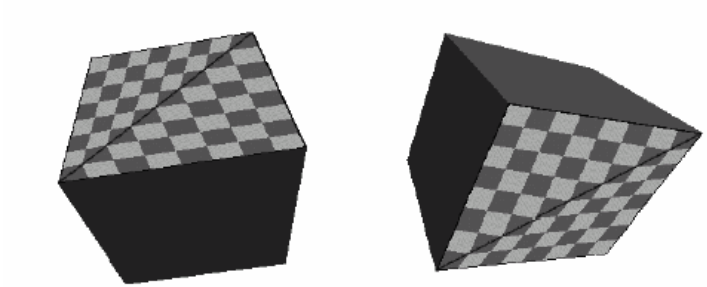
attributi di p:
(considerando
la "correzione prospettica")

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

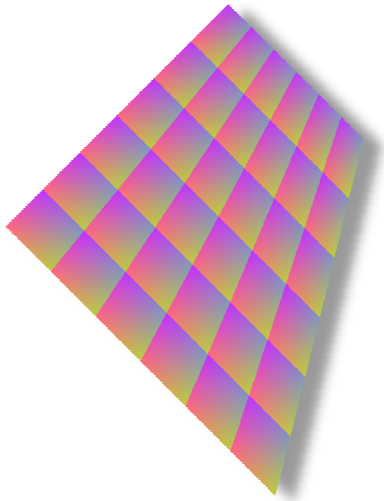
Correzione Prospettica

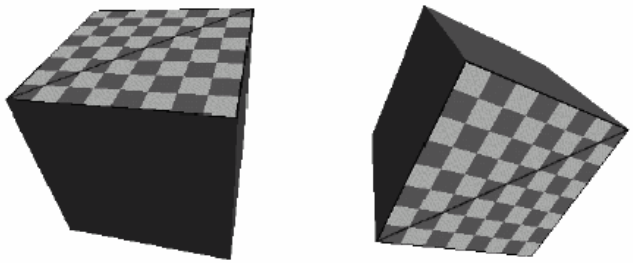
- Senza





- Con



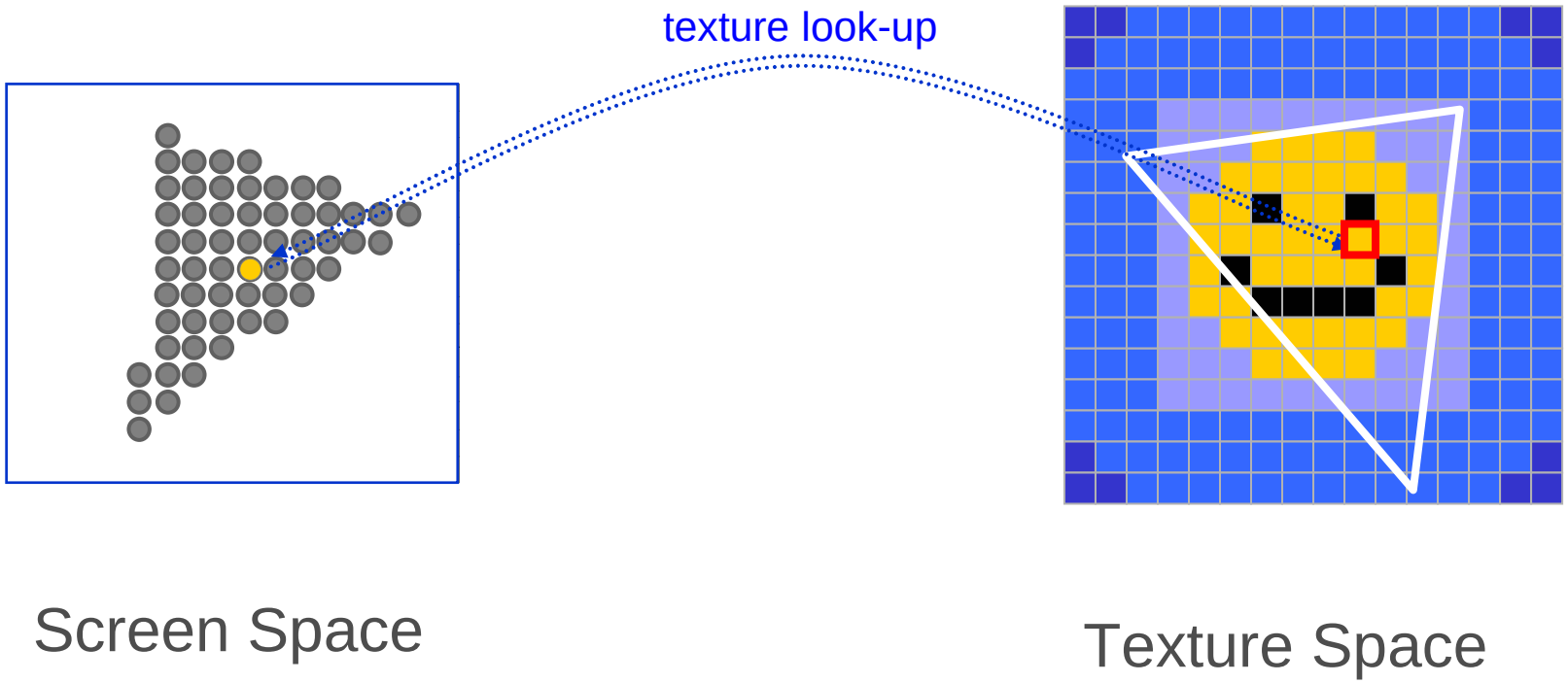


Correzione Prospettica

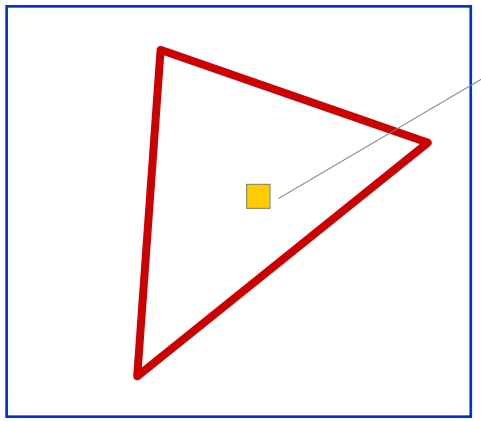
- Texture mapping con correzione prospettica
  - anche conosciuto come:
    - texture mapping perfetto
    - metodo dei 3 vettori magici *(arcaico)*
    - metodo dei 9 numeri magici *(arcaico)*

Texture Look-up

- Un frammento ha coordinate non intere (in texels)

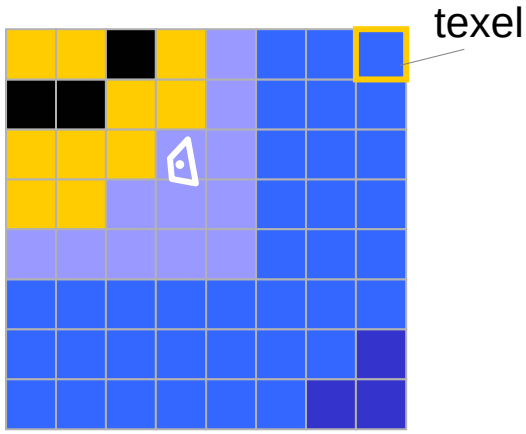


Texture Look-up



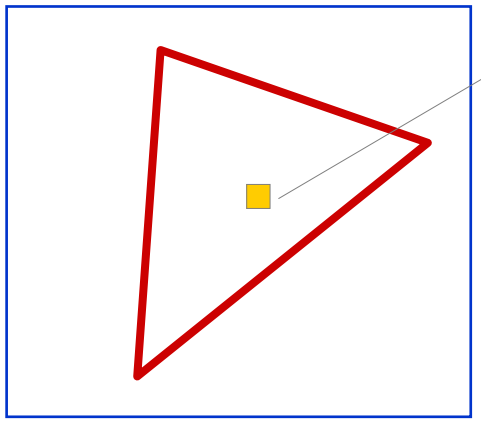
un pixel = meno di un texel

magnification



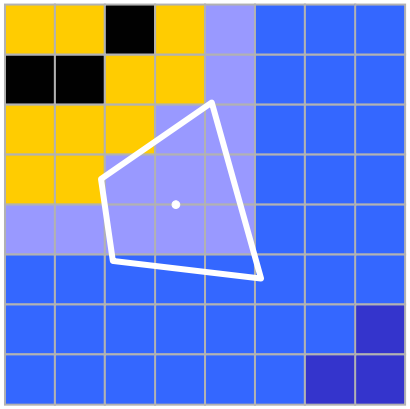
Screen Space

Texture Space



un pixel = più di un texel

minification



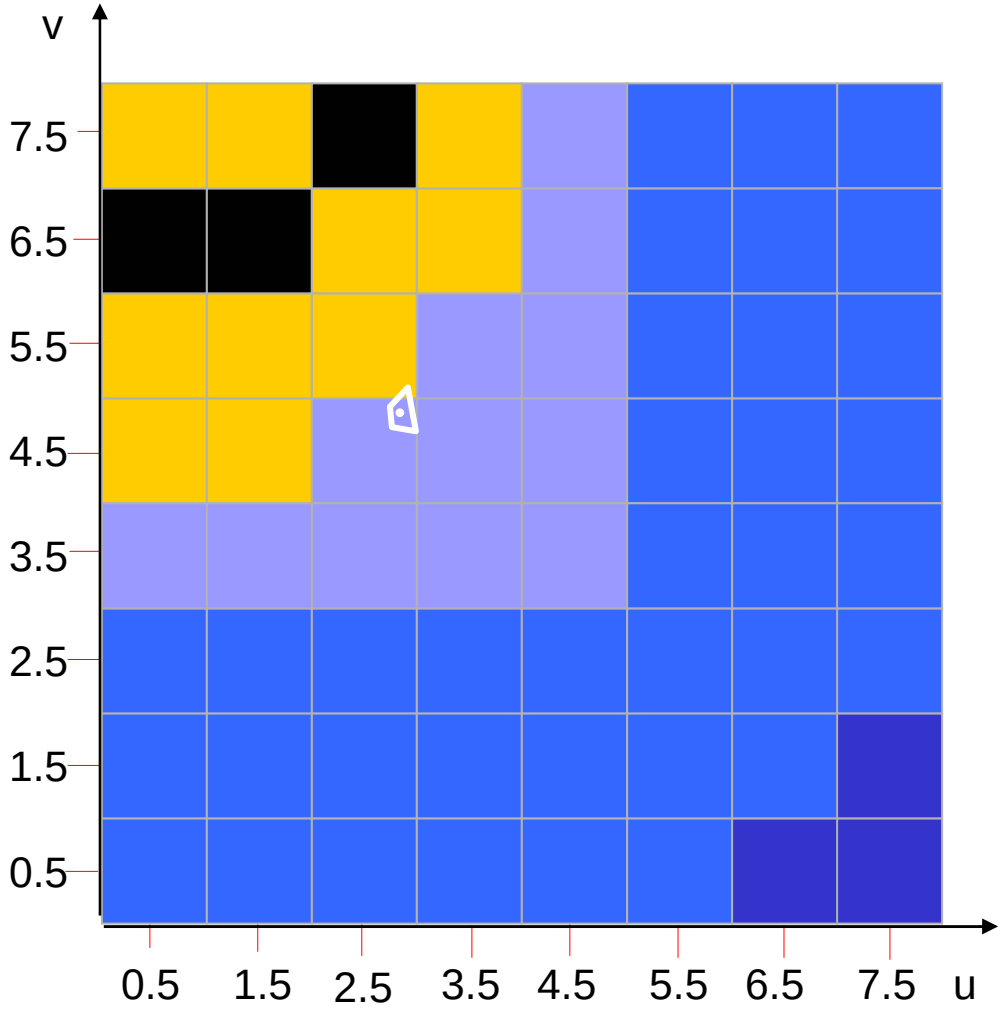
Caso Magnification

Soluzione 1:
prendo il texel in cui sono

(equivale a prendere
il texel più vicino)

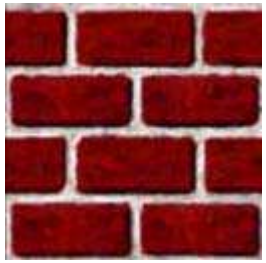
equivale ad arrotondare
alle coordinate texel
intere

"Nearest Filtering"



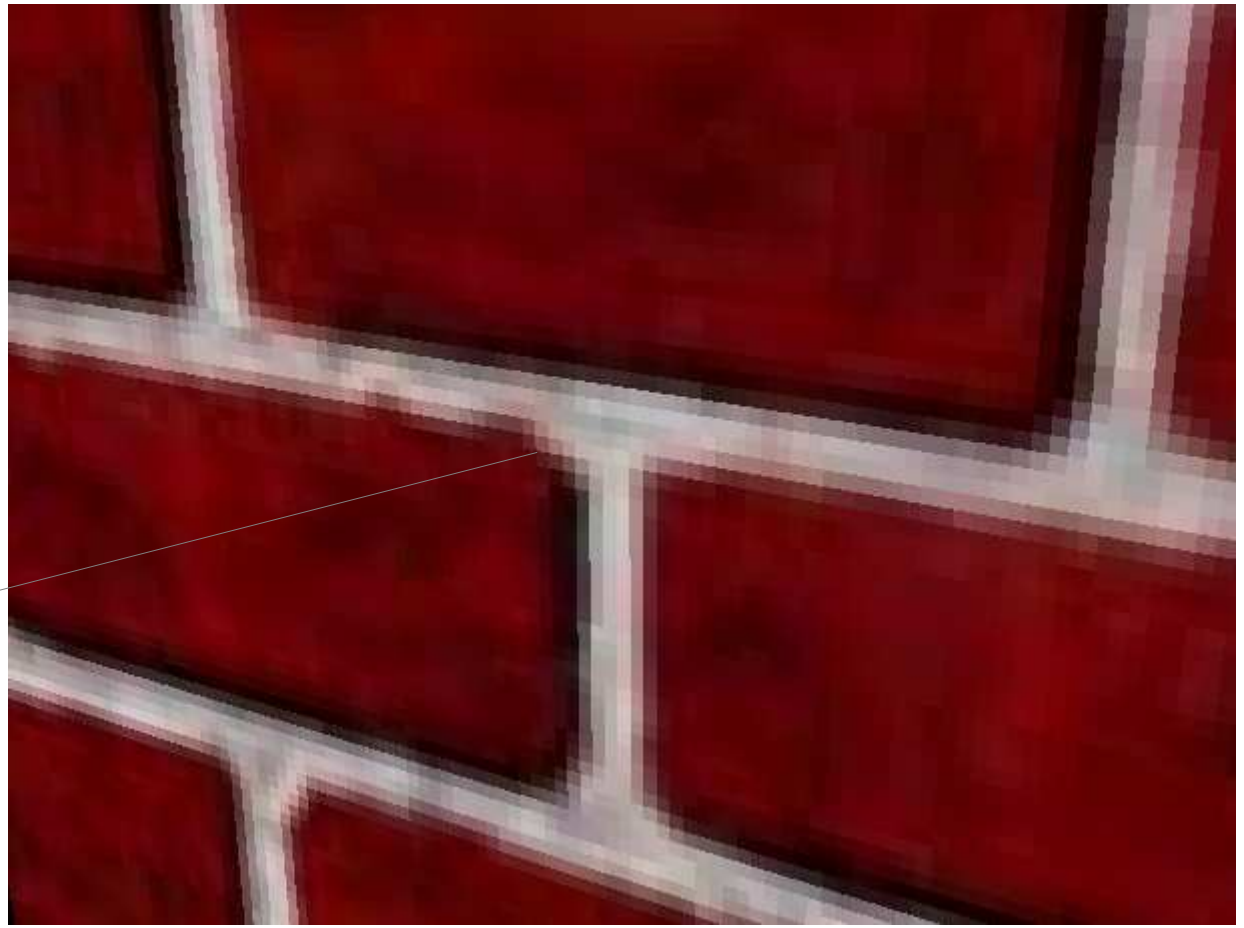
Caso Magnification

Nearest Filtering: risultato visivo



texture 128x128

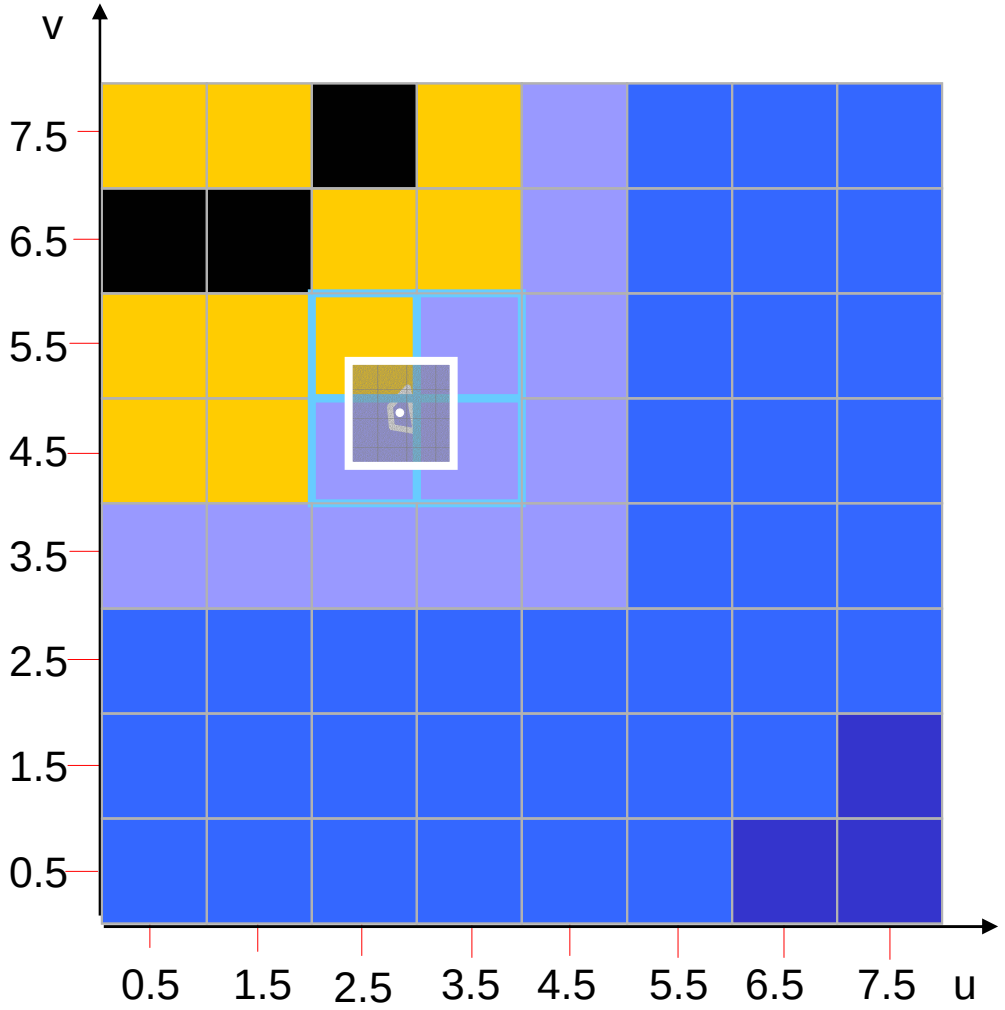
"si vedono i texel !"



Caso Magnification

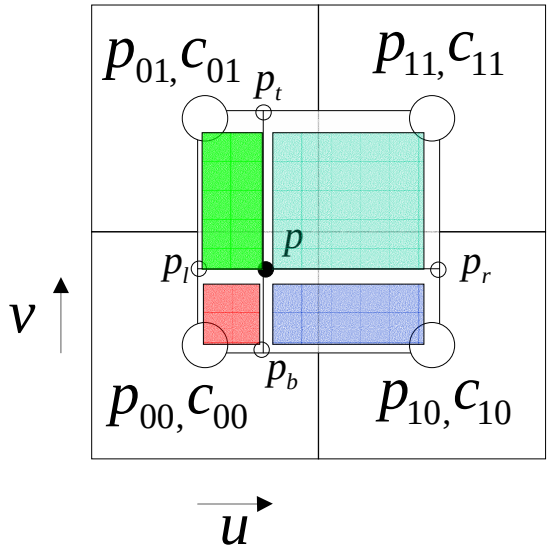
Soluzione 2:
Medio il valore dei quattro texel più vicini

Interpolazione Bilineare



Intrpolazione bilineare

$$p_b = (1-u)\,p_{00} + u\,p_{10}$$
$$p_t = (1-u)\,p_{01} + u\,p_{11}$$
$$p_l = (1-v)\,p_{00} + v\,p_{01}$$
$$p_r = (1-v)\,p_{10} + v\,p_{11}$$
$$p = (1-u)\,p_l + u\,p_r$$



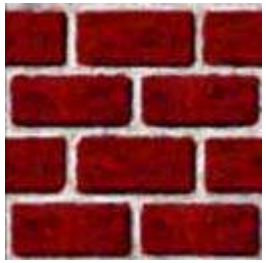
oppure $p = (1-v)\,p_b + v\,p_t$

In entrambi i casi:

$$p = (1-u)(1-v)\,p_{00} + (1-u)v\,p_{01} + (1-v)u\,p_{10} + uv\,p_{11}$$

Caso Magnification

Bilinear Interpolation: risultato visivo



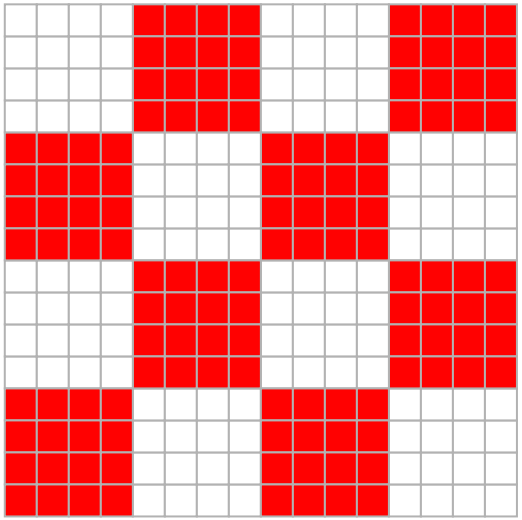
texture 128x128

Caso Magnification

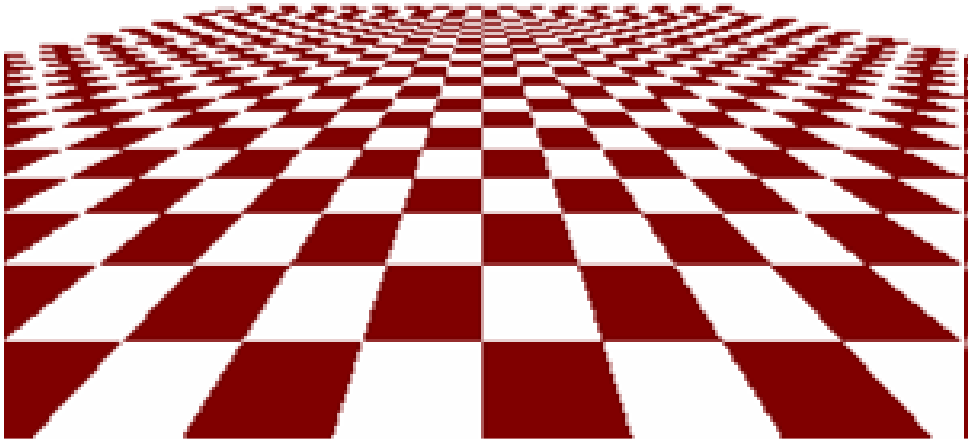
- Modo Nearest:
  - si vedono i texel
  - va bene se i bordi fra i texel sono utili
  - più veloce

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

Caso Minification

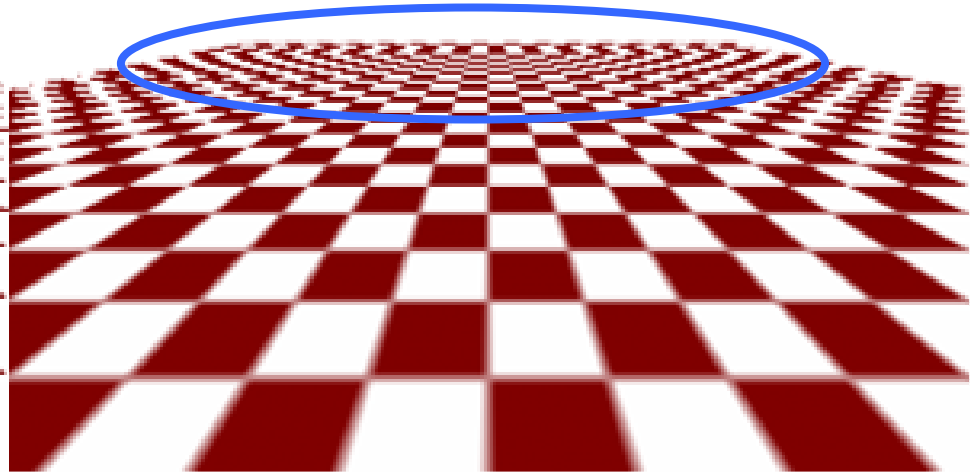


Nearest Filtering



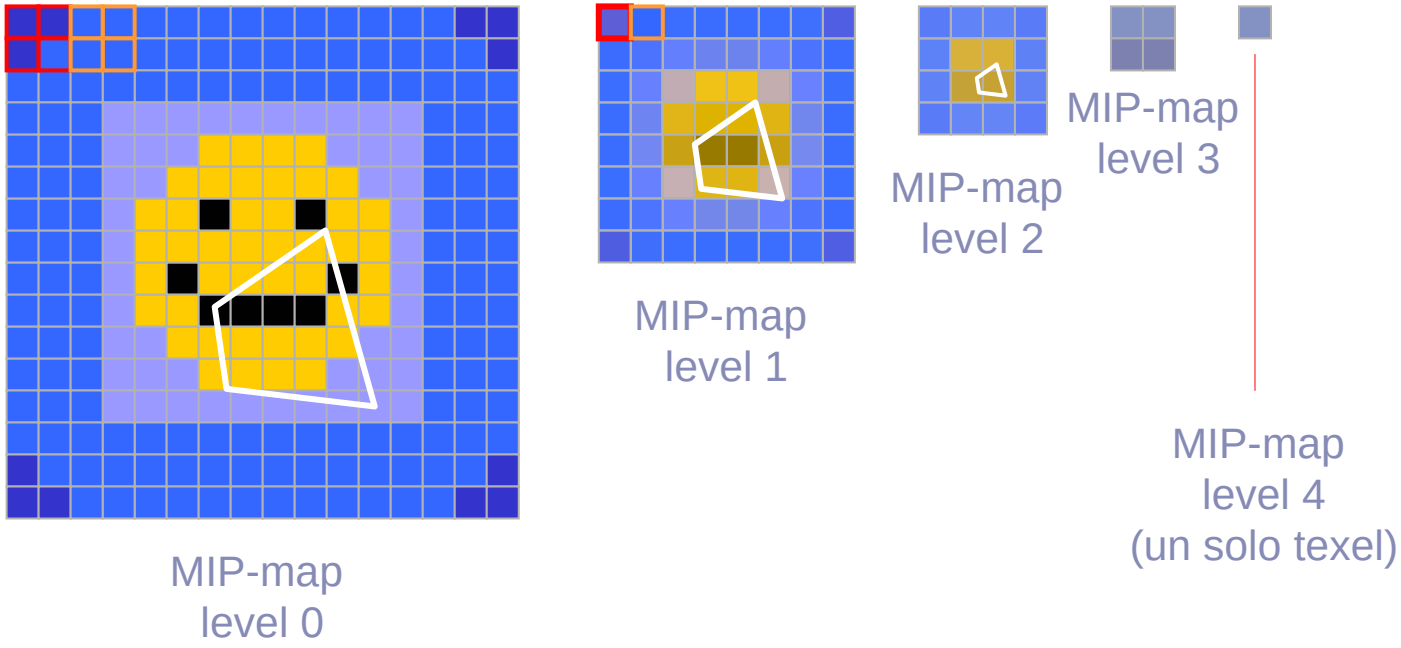
Bilinear interpolation

non risolve il problema



Caso Minification: MIP-mapping

MIP-mapping: "Multum In Parvo"

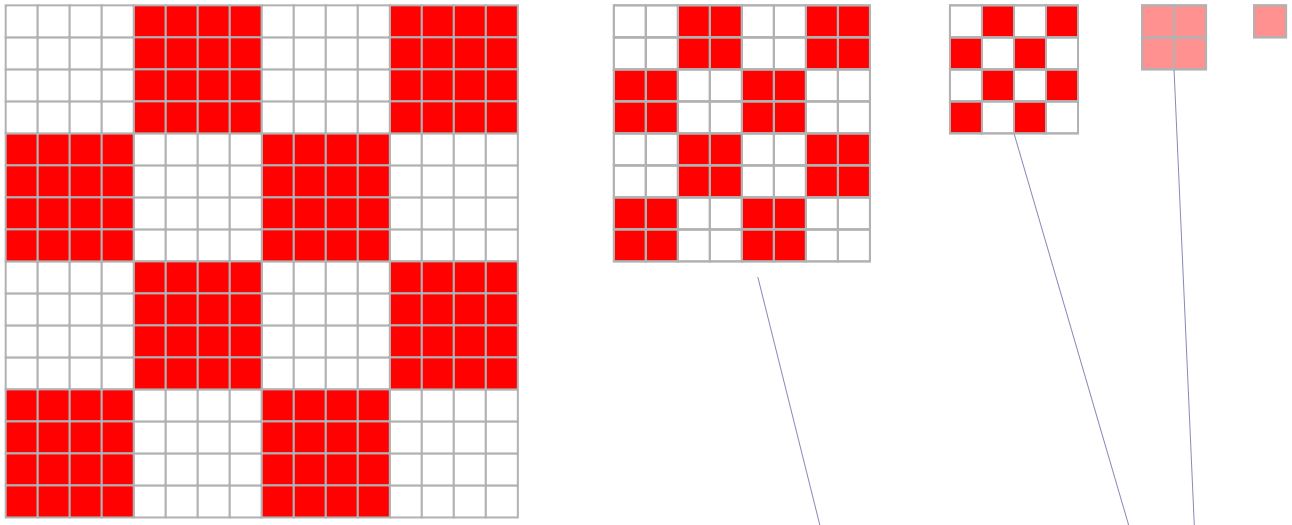


Matematica del Mipmapping

- Definiamo un fattore di scala, ρ=texels/pixel
  - ρ è il massimo fra ρ_x e ρ_y
  - può variare sullo stesso triangolo
  - può essere derivato dalle matrici di trasformazione
  - calcolato nei vertici, interpolato nei frammenti

- il livello di mipmap da utilizzare è: $\log_2 \rho$
  - livello 0 = massima risoluzione
  - se livello<0 cosa significa?
  - nota: il livello non è necess. un numero intero

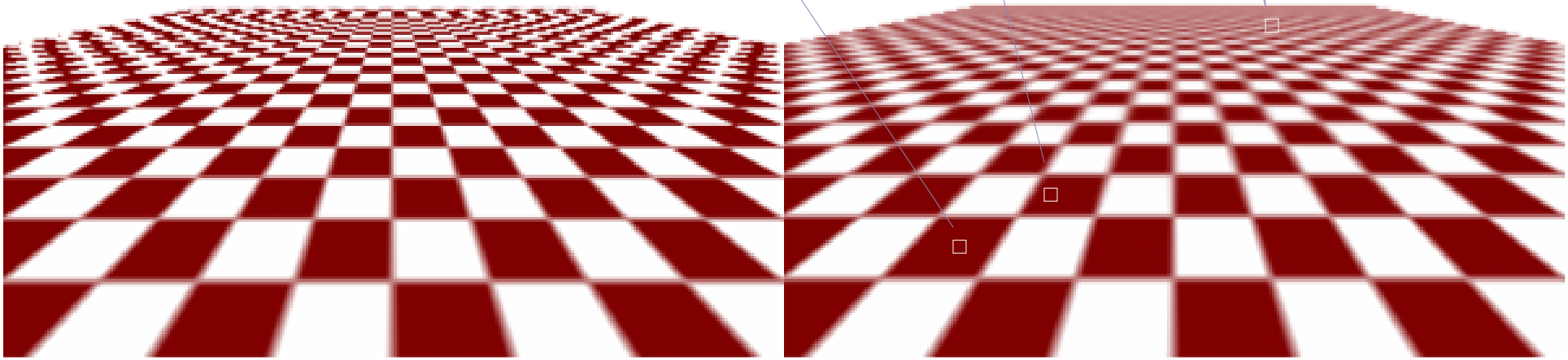
Caso Minification: MIP-mapping



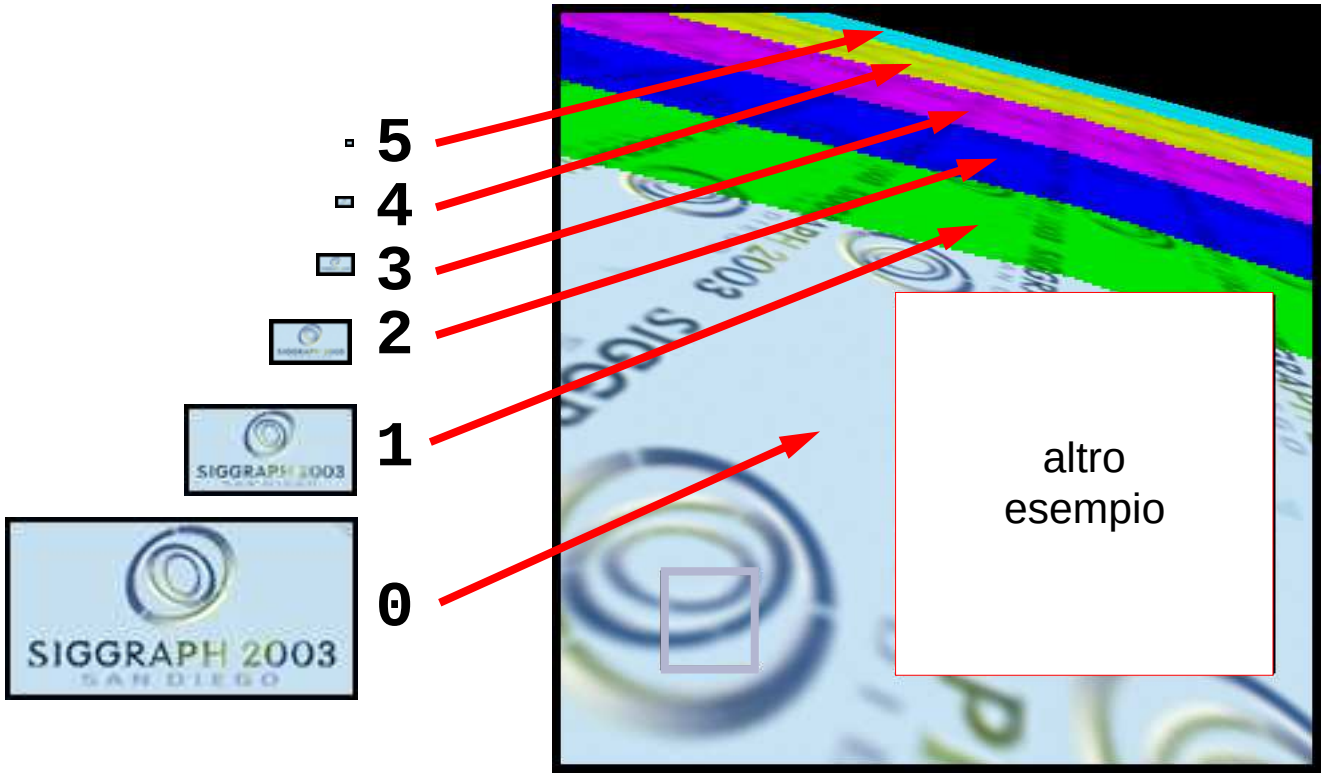
Bilinear interpolation

non risolve il problema

MIP-mapping

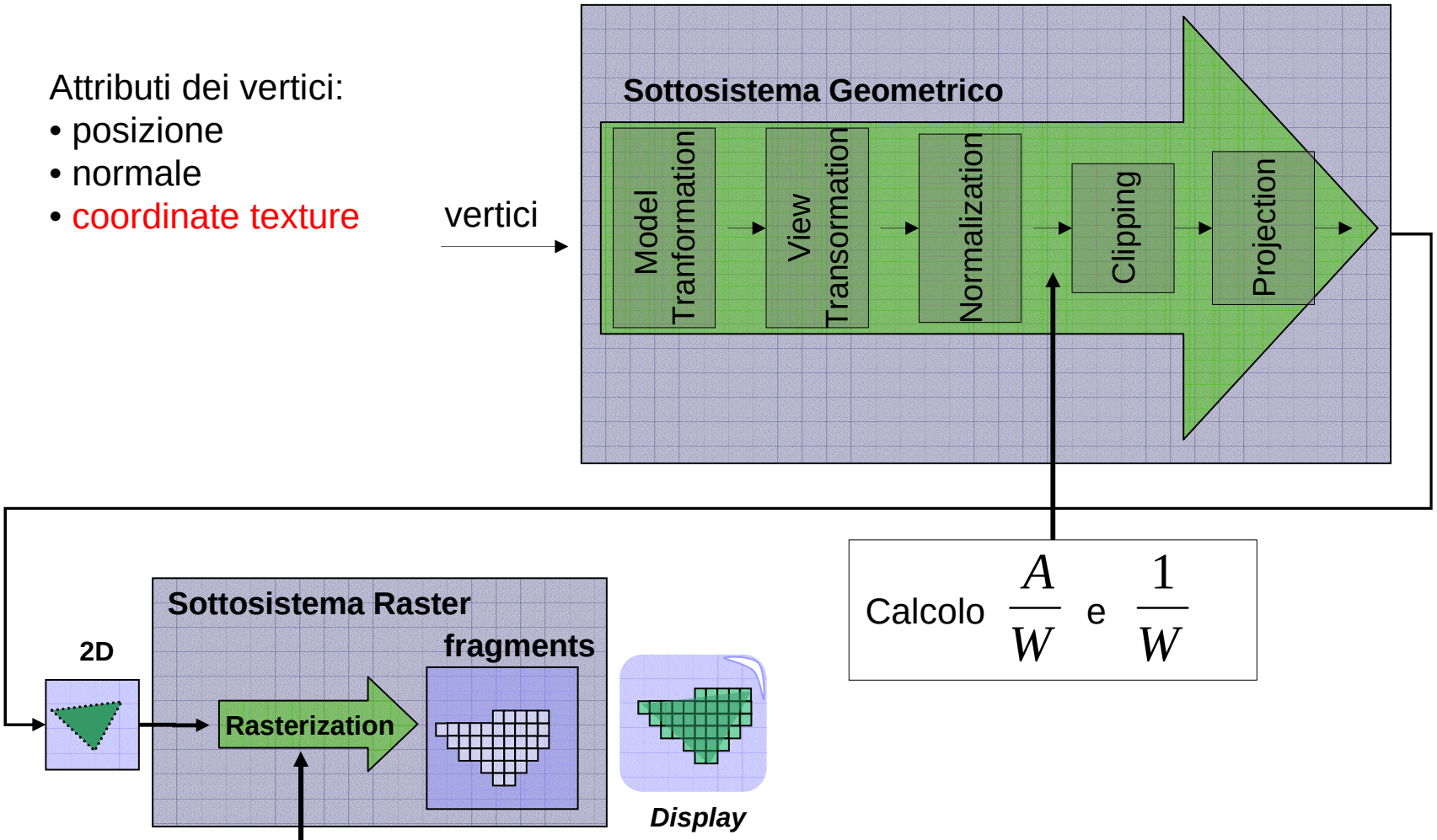


Caso Minification: MIP-mapping



Nella pipeline

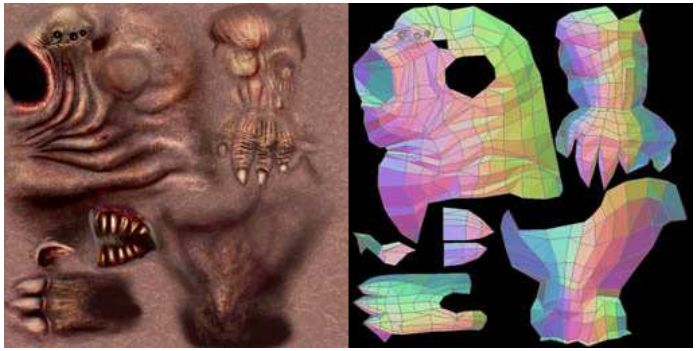
Attributi dei vertici:
- posizione
- normale
- coordinate texture



Le coordinate texture: da dove vengono?

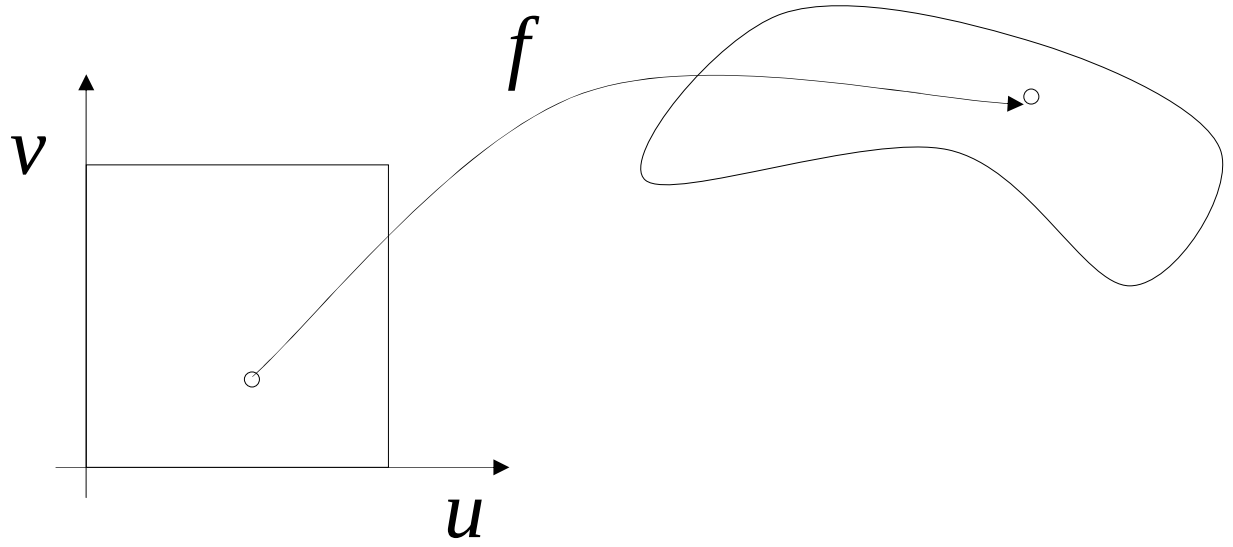
- Nella modellazione CAM sono calcolate a mano o comunque in maniera semiautomatica
- Si colora il modello in 3D e poi si costruiscono le texture prendendo la superficie e “stirandola” su un quadrato [0,1]x[0,1]
- Ogni vertice finisce nella sua coordinata texture

- Quando è difficile fare lo “stiramento” ?
- abbastanza….



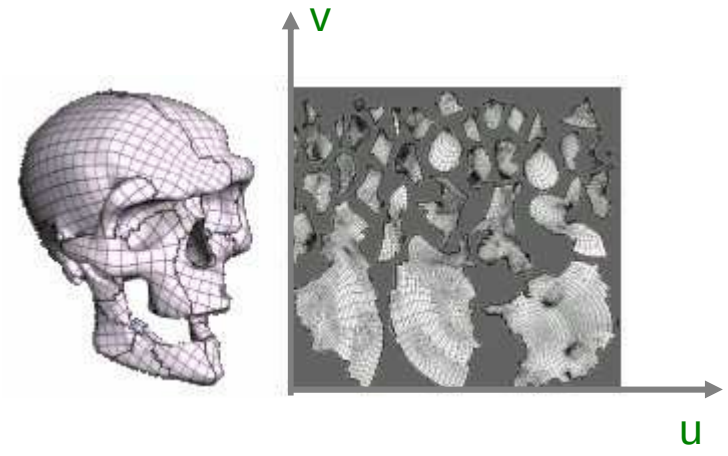
UV Mapping

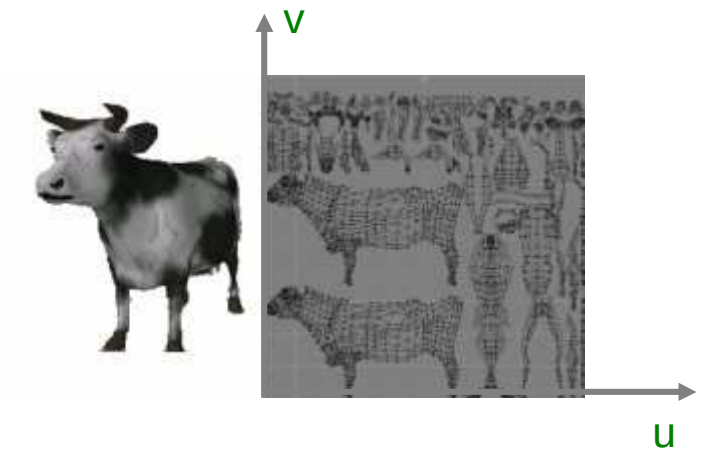
- Trovare una funzione cha mappa [0,1]x[0,1] nella superficie
  - Esiste sempre?..NO
  - Se esiste è unica? ..NO
  - Ce ne è una migliore?..No, dipende dalla metrica
  - e una volta stabilita la metrica?..In generale NO, ci sono minimi locali



Atlas

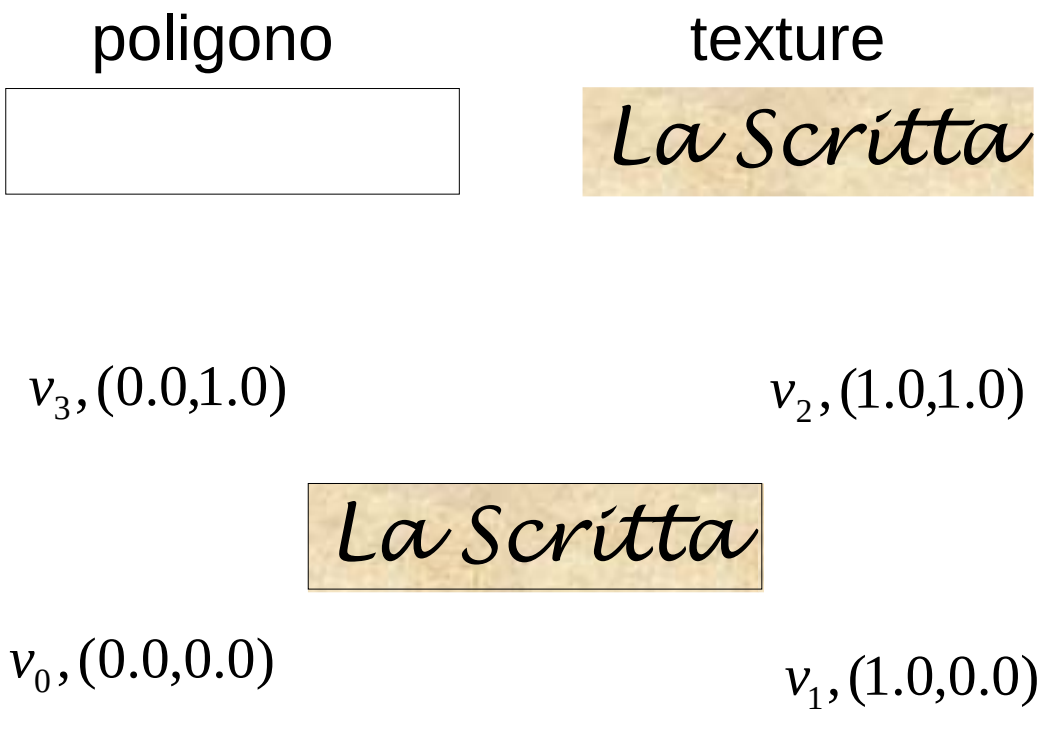
- Costruire un atlas: partizionare la superficie in modo che le singole parti siano facilmente parametrizzabili





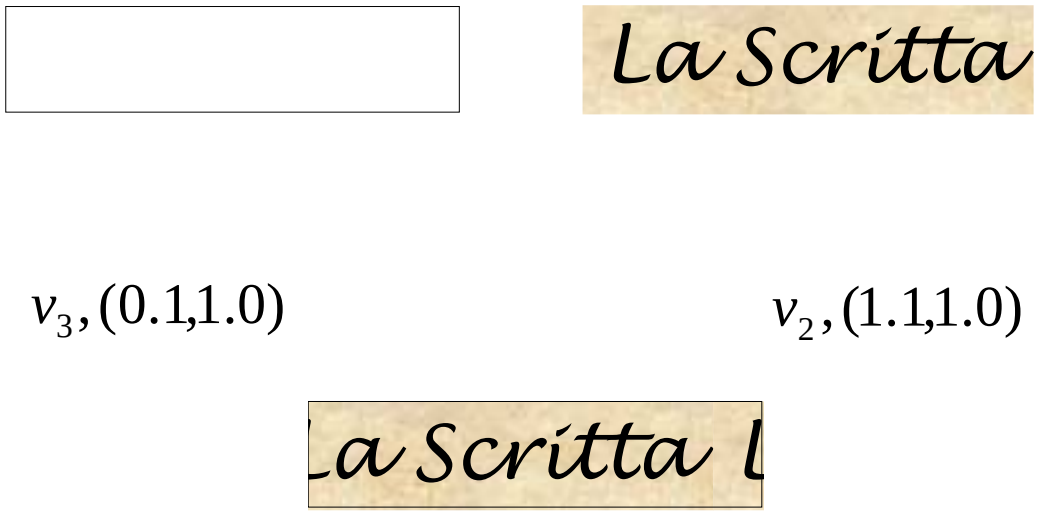
Coordinate texture calcolate nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



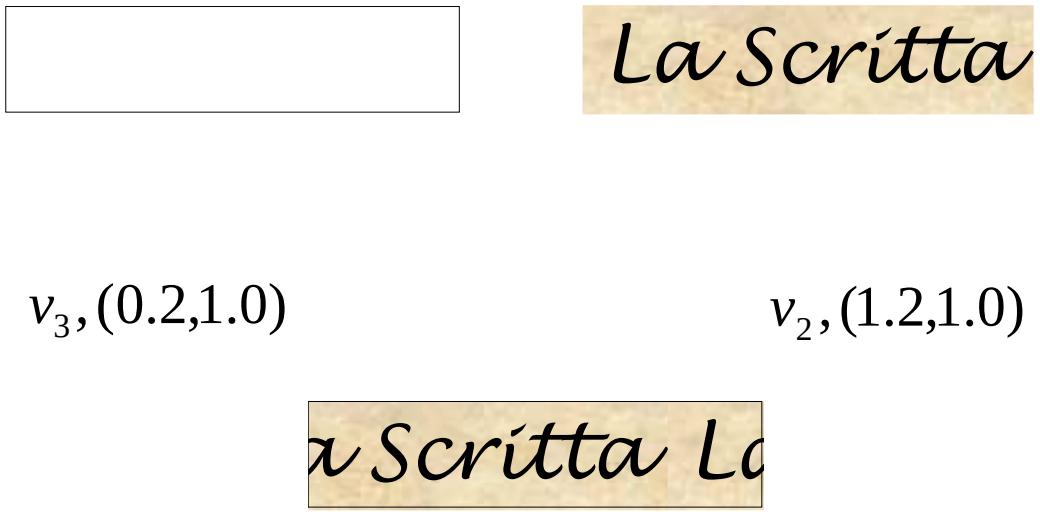
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



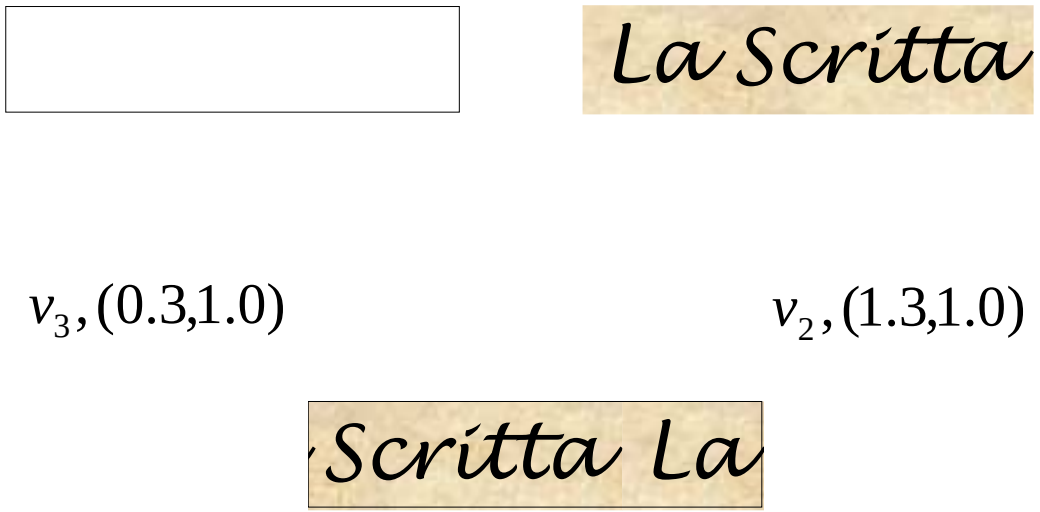
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



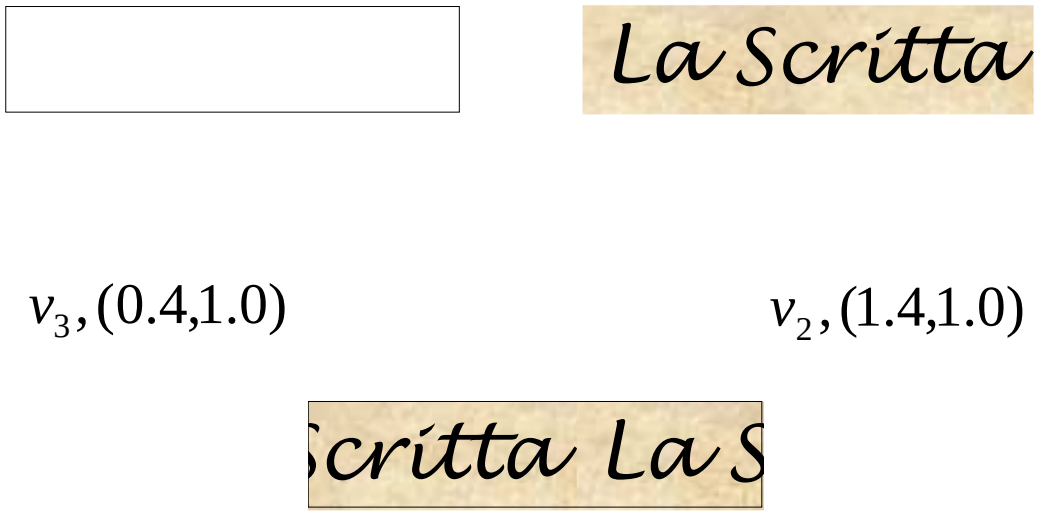
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



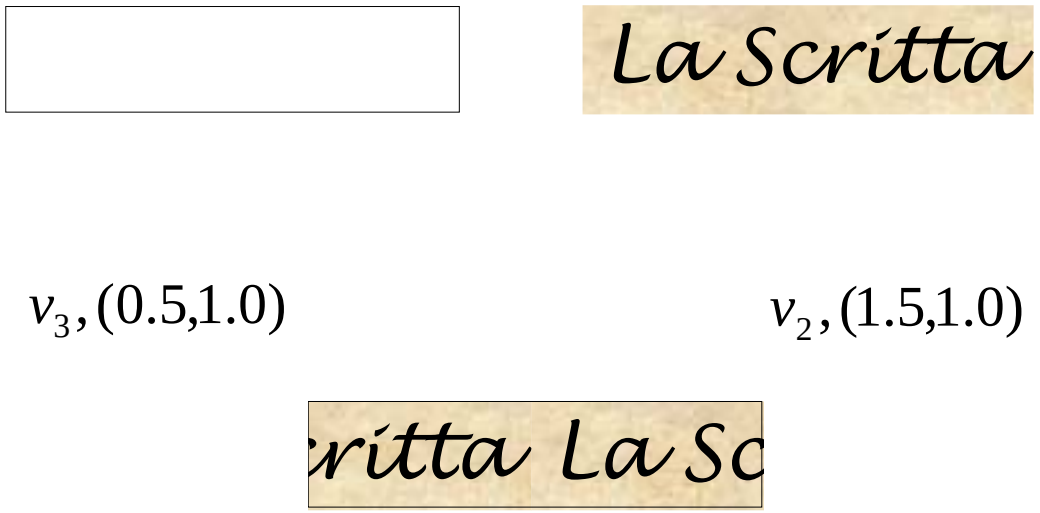
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



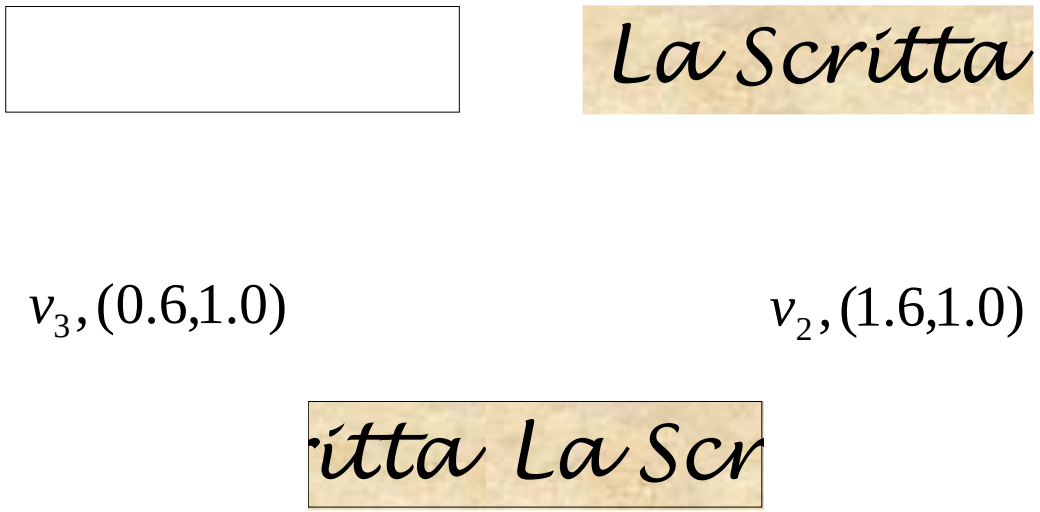
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



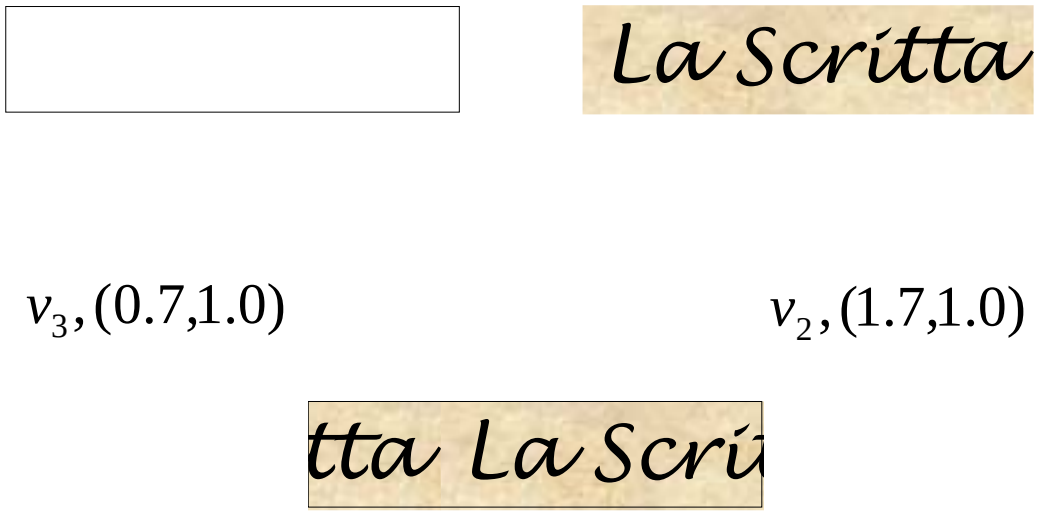
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



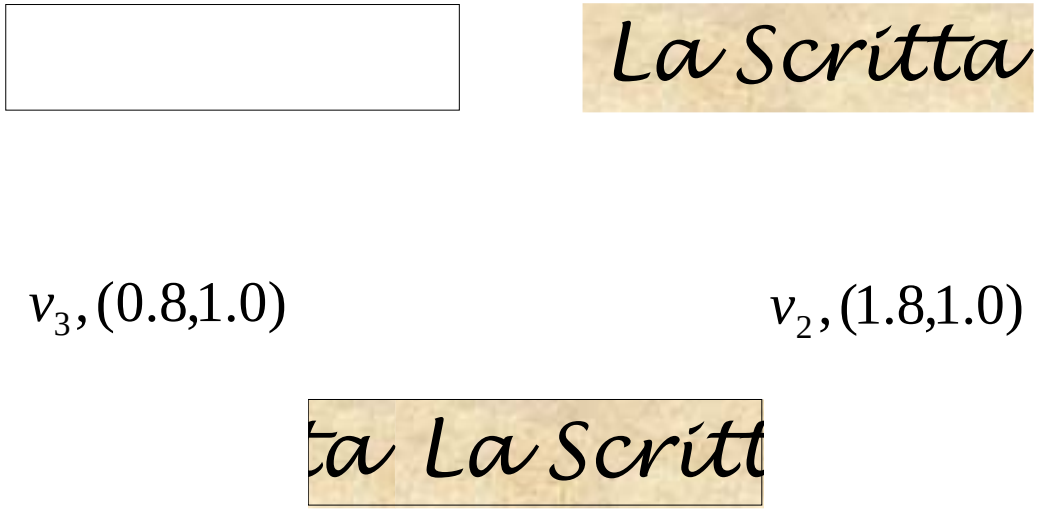
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



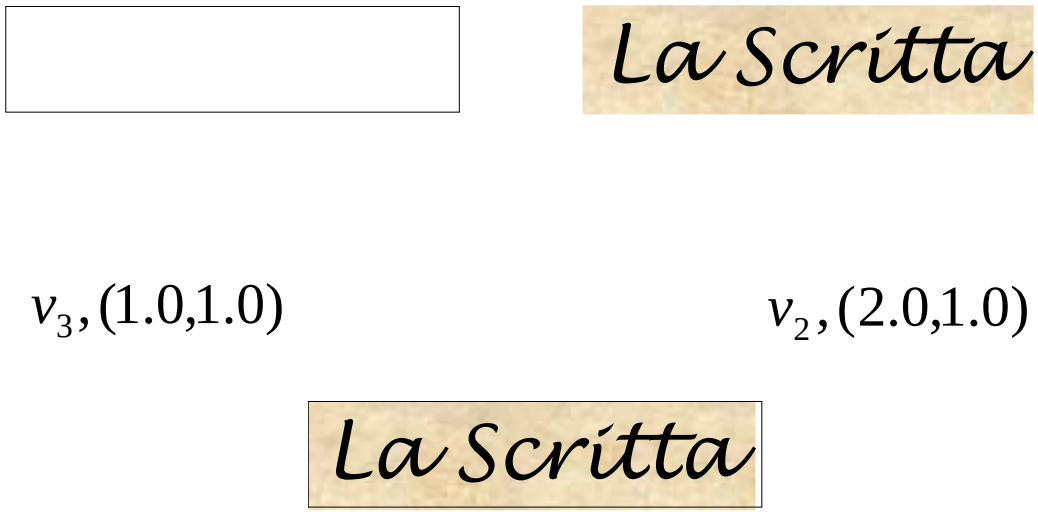
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



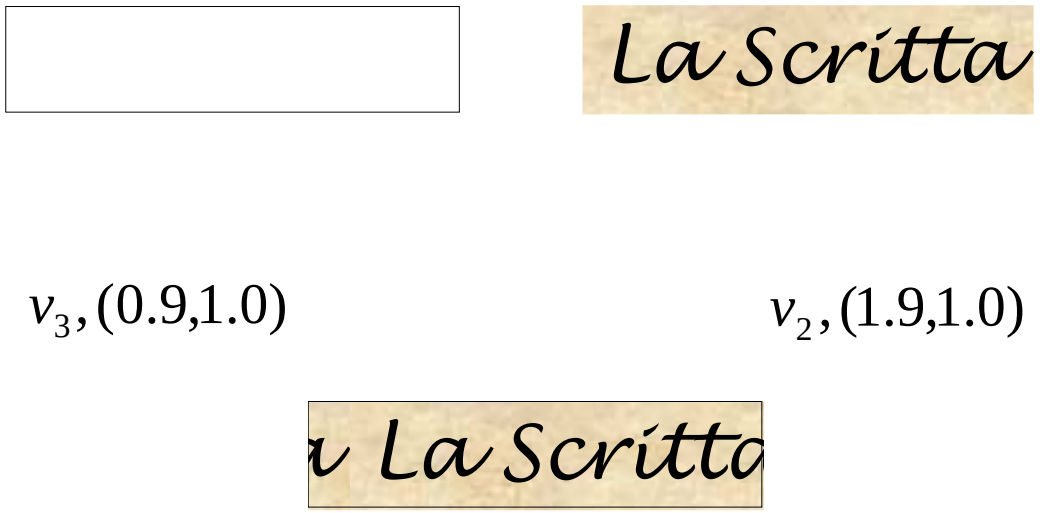
Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?



Calcolarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione del tempo
- A che serve?

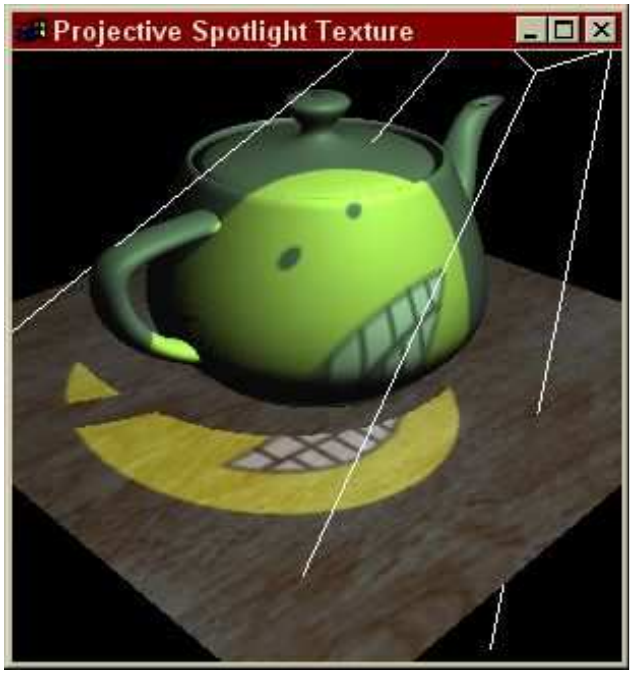


Matrice di trasformazione coordinate texture

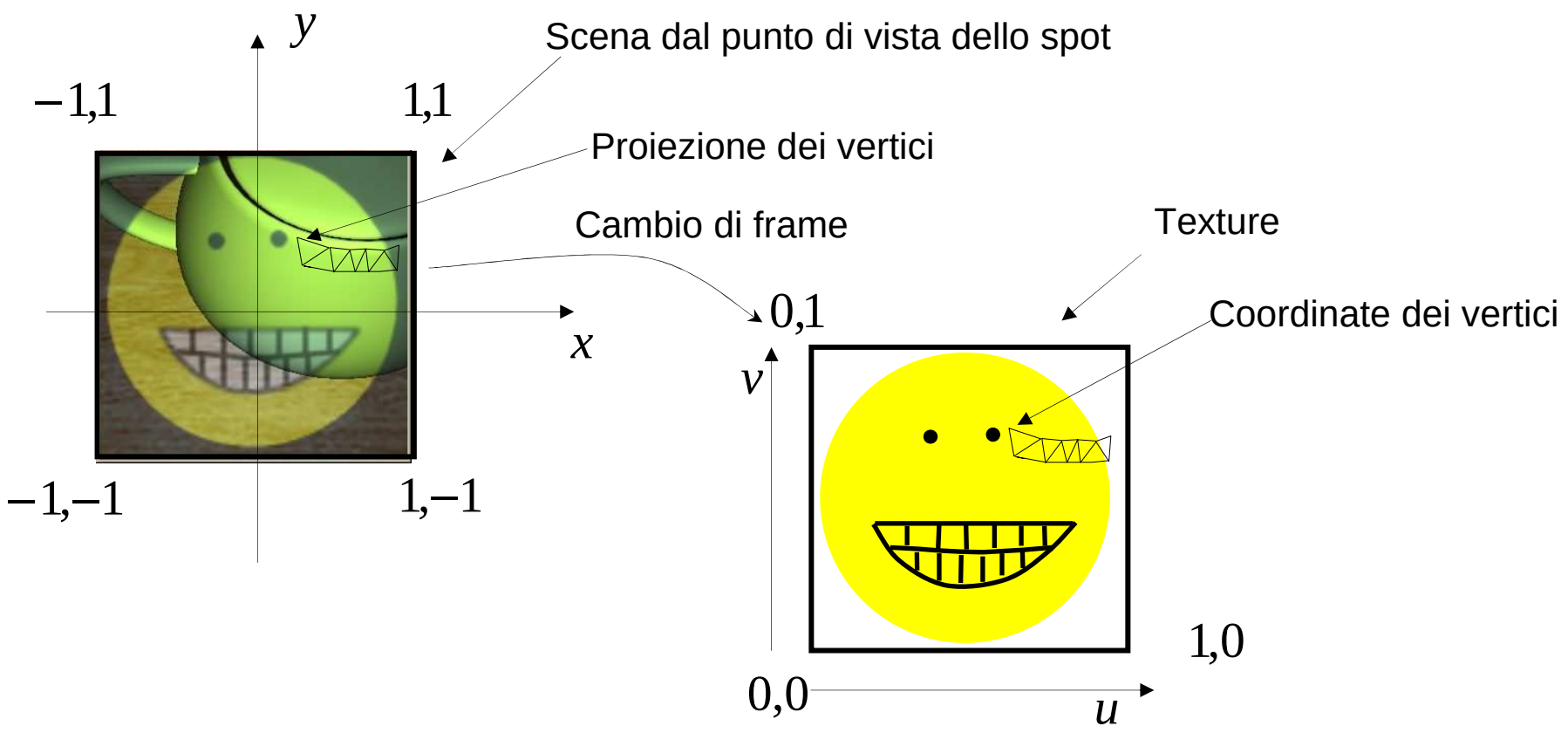
- Occorre cambiare a mano i valori delle coordinate texture vertice per vertice?
- No, le coordinate texture vengono moltiplicate per la *matrice di texture* così come le posizioni dei vertici vengono moltiplicate per la matrice modelview

Calcorarle nella pipeline

- Cambiare le coordinate texture dei vertici in funzione delle coordinate della loro posizione
- A che serve?
- Es: spotlight



Spotlight


y
−1,1
1,1
x
−1,−1
1,−1
Scena dal punto di vista dello spot
Proiezione dei vertici
Cambio di frame
Texture
Coordinate dei vertici
0,1
v
1,0
0,0
u

Sphere Environment Mapping

- Il problema: guardando un oggetto di superficie speculare ci voglio vedere il mondo (environment) riflesso
- L'idea: memorizzo in una texture come sarebbe il mondo riflesso su una semisfera perfettamente speculare
- La texture sarà tipo questa:

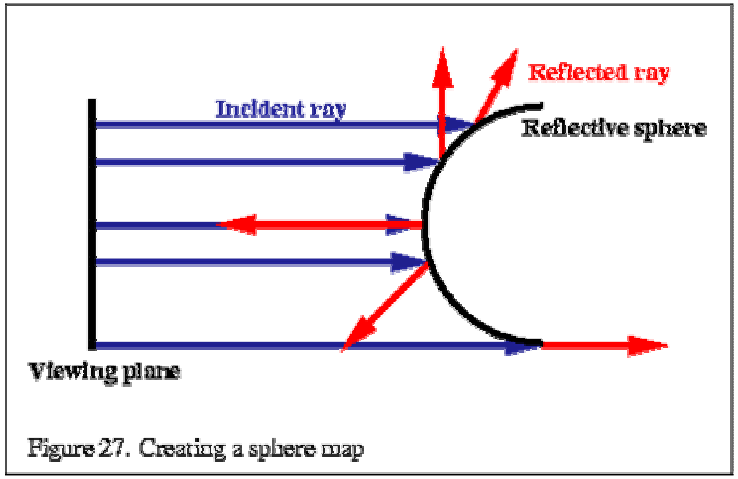


Figure 27. Creating a sphere map

Sphere Environment Mapping

$$f' = 2N \cdot (N \cdot f) - f$$

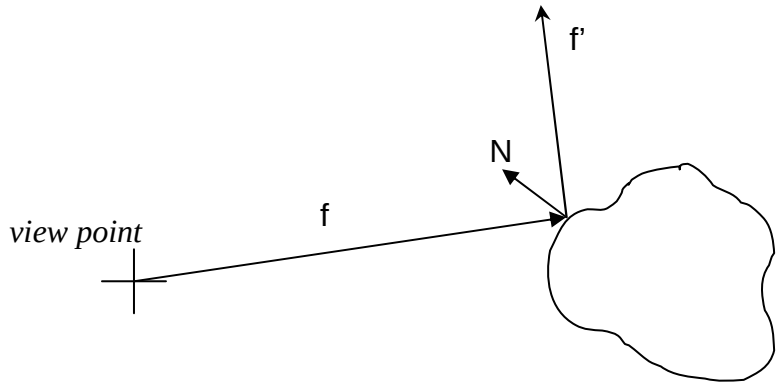
$$p = 2 \cdot \sqrt{f'^2_x + f'^2_y + (f'_z + 1)^2}$$

Sphere mapping

$$t = \frac{f'_x}{p} + 0.5$$

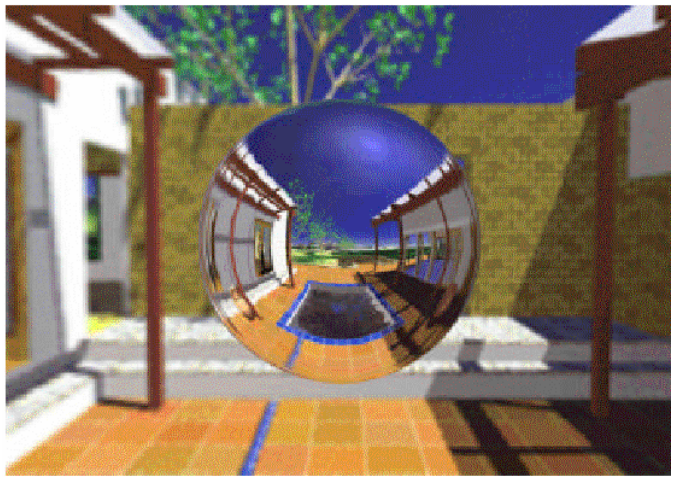
$$s = \frac{f'_y}{p} + 0.5$$

Si applica una trasformazione che mappa tutte le direzioni nella circonferenza di raggio 0.5 centrata in (0.5,0.5)

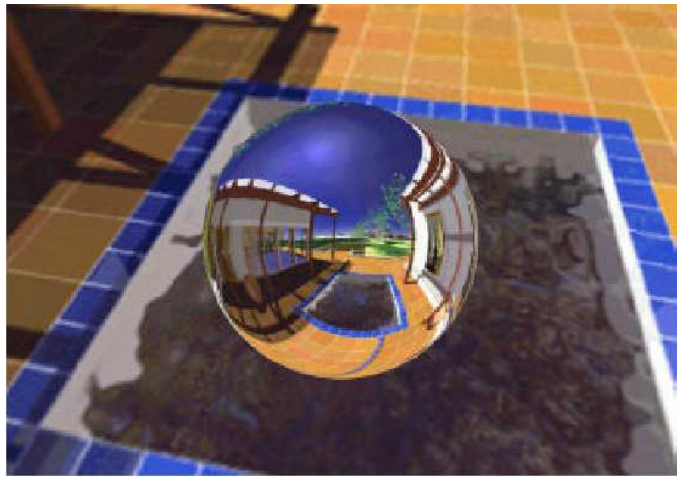


Sphere Env.Mapping.: problemi

- È view dependent: la mappa viene costruita da un punto di vista preciso ed è corretta solo per quello



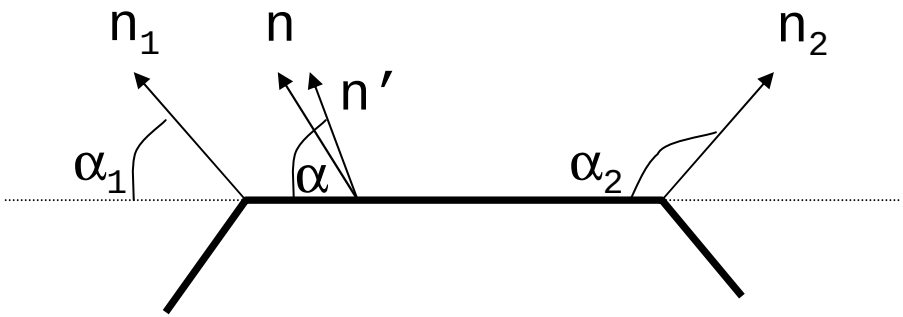
Corretto



Errato: vedo la piscina e il riflesso della piscina

Sphere Env.Mapping.: problemi

- interpolazione:
  - l'interpolazione delle normali è lineare ma la sphere map non è uno spazio lineare

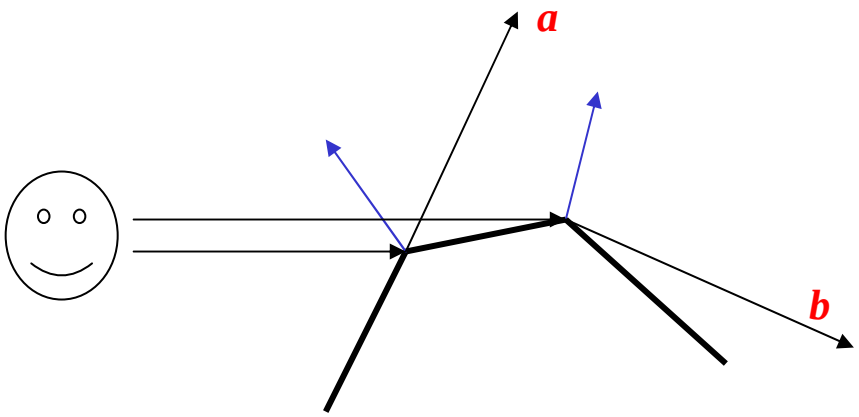


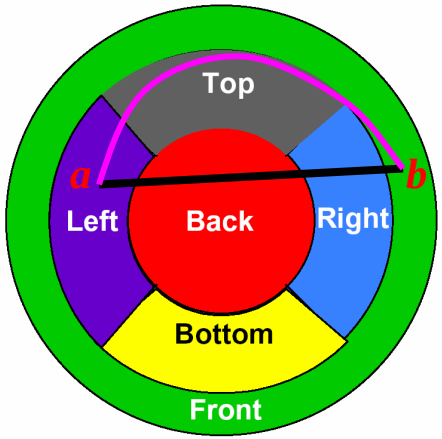
$n = n_1 (1-\lambda)+n_2 \lambda$:come è

$\alpha = \alpha_1 (1-\lambda)+\alpha_2 \lambda$:come dovrebbe essere

Sphere Env.Mapping.: problemi

- Sul bordo di un oggetto puo’ capitare che coordinate calcolate per I vertici di un triangolo corrispondano
  a punti molto distanti nella texture

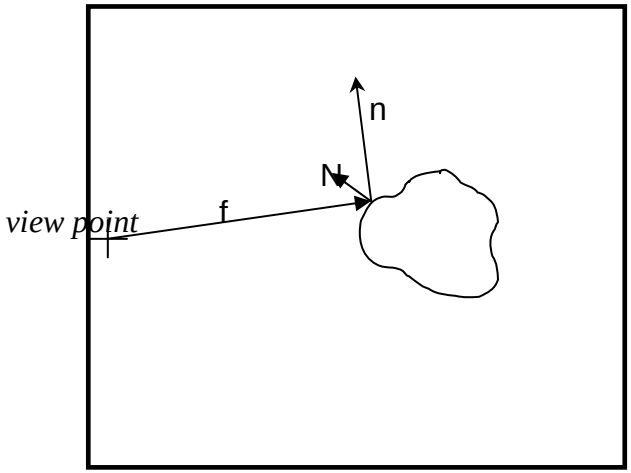




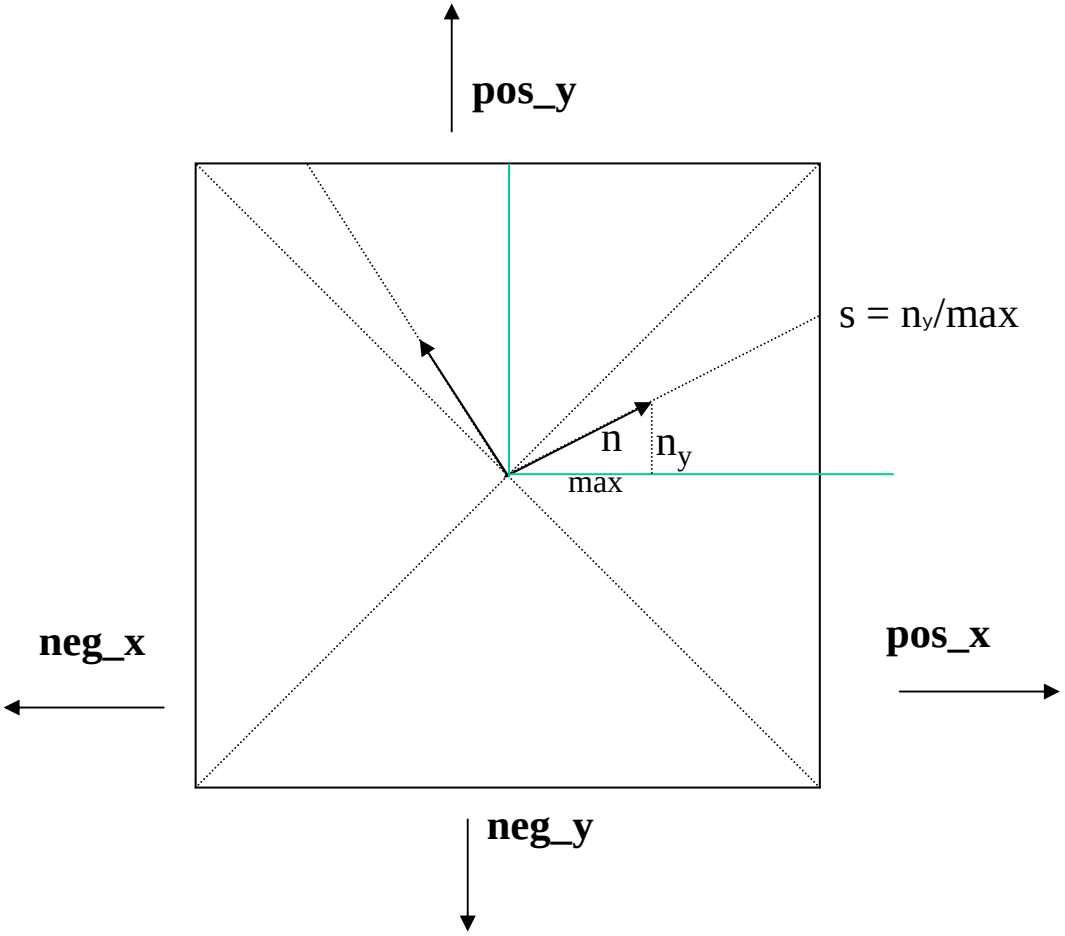
Cube Maps

- Stesso principio, però
  - più semplice
  - più robusta
  - view independent
  - meno distorsione
- L’idea: metto un cubo di grandezza infinita intorno alla scena. Ogni faccia è l’environment in quella direzione. Per il resto procedo come per lo sphere mapping

Come si fa?



Cube Maps: accesso alle textures



La texture viene scelta in base alla componente di massimo valore assoluto *max*

le coordinate texure sono:
s = (n1 / max+1)/2
t = (n2 / max +1)/2
dove n1 e n2 sono le altre due componenti di n

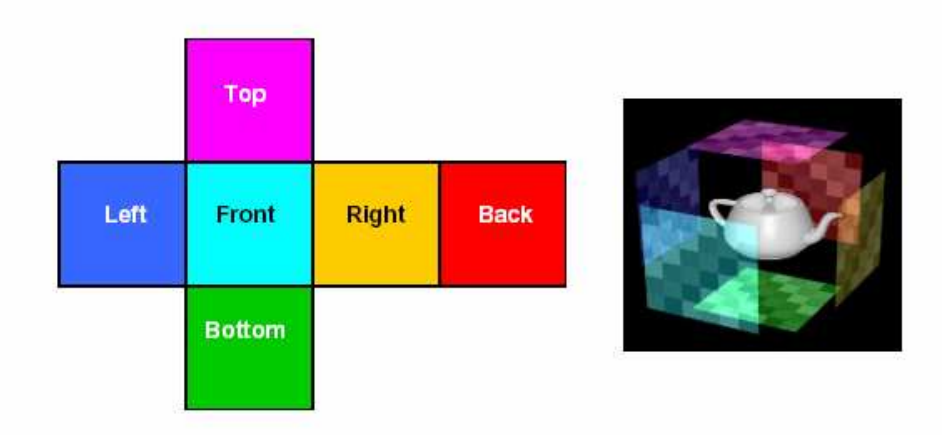
.....e cosa è *n*?

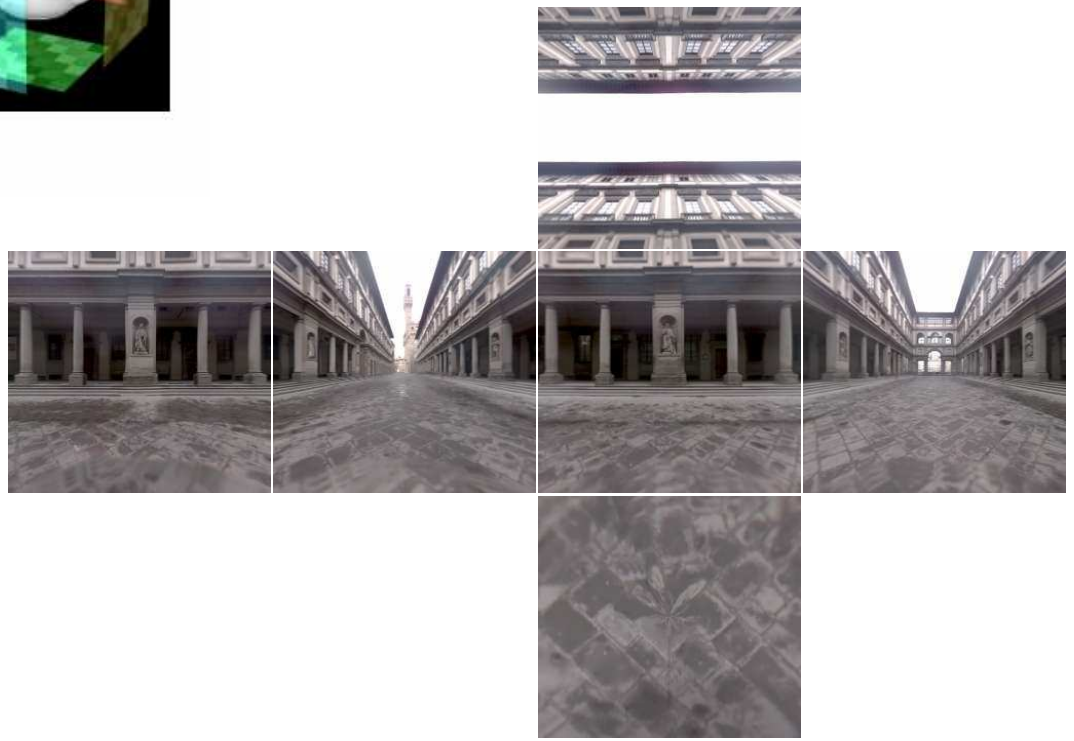
Cube Maps: accesso alle textures

- n è una tripla di coordinate texture associate ad ogni vertice
- potreste specificarle voi con **`glCoord3f(nx,ny,nz)`** e la cube map funzionerebbe
- Noi però le vogliamo dipendenti dalla normale alla superficie in *eye space*, cioè nel frame di vista
- Si generano automaticamente (di nuovo)

Cube Maps

- Esempio di textures per il cupe mapping:

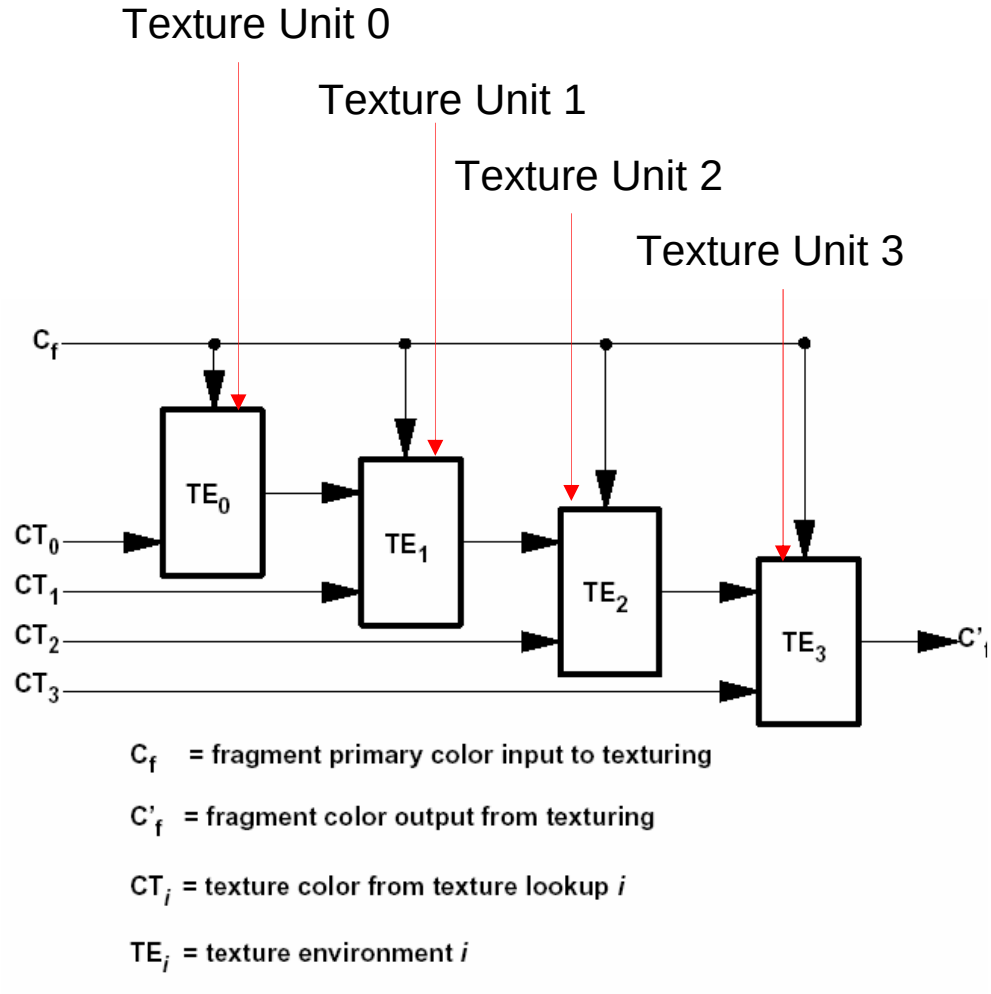




demo..

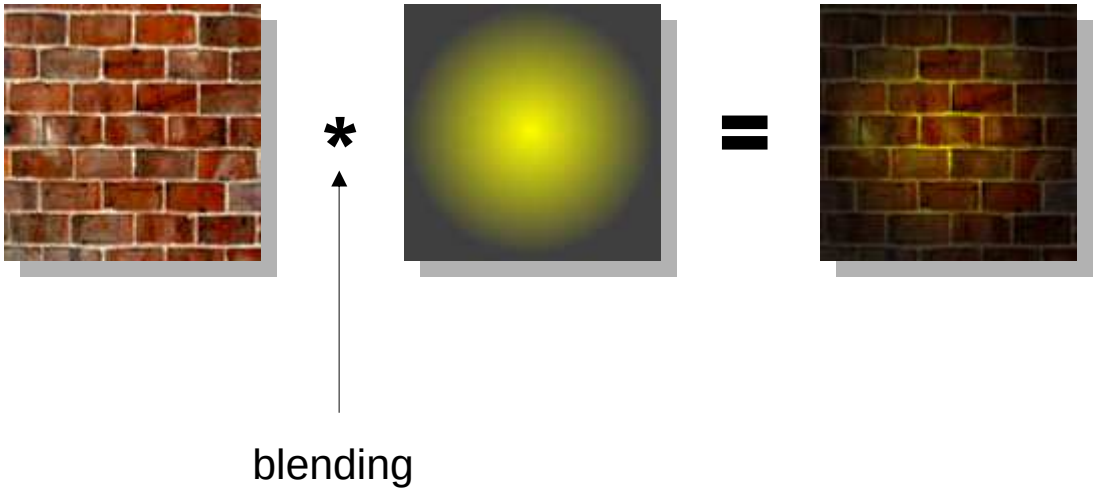
Multi-texturing

- È possibile specificare più di una texture per una singola primitiva
- Le varie texture sono applicate in sequenza usando il risultato del precedente texturing per mixarlo con la texture corrente secondo un proprio texture environment



Light Maps

- Uso del Multitexturing

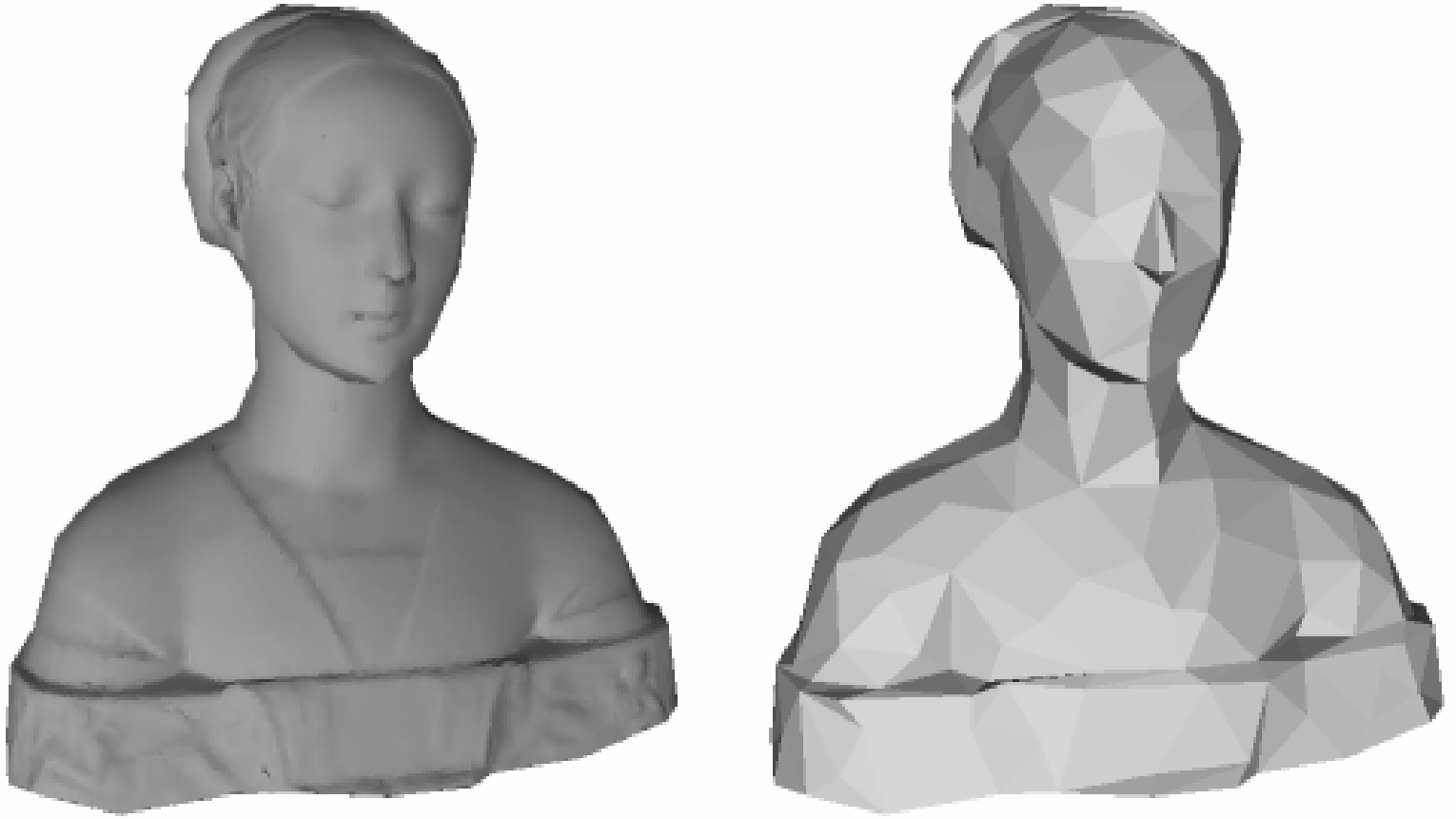


Texture

- Non solo colore
- Texture mapping è utilizzato anche per *spargere* altri tipi di attibuti sulla superficie di un oggetto:
  - Normali (bump/normal mapping)
  - Posizione (displacement mapping)
  - Trasparenza (alpha)
  - Shininess
  - Ombre portate.

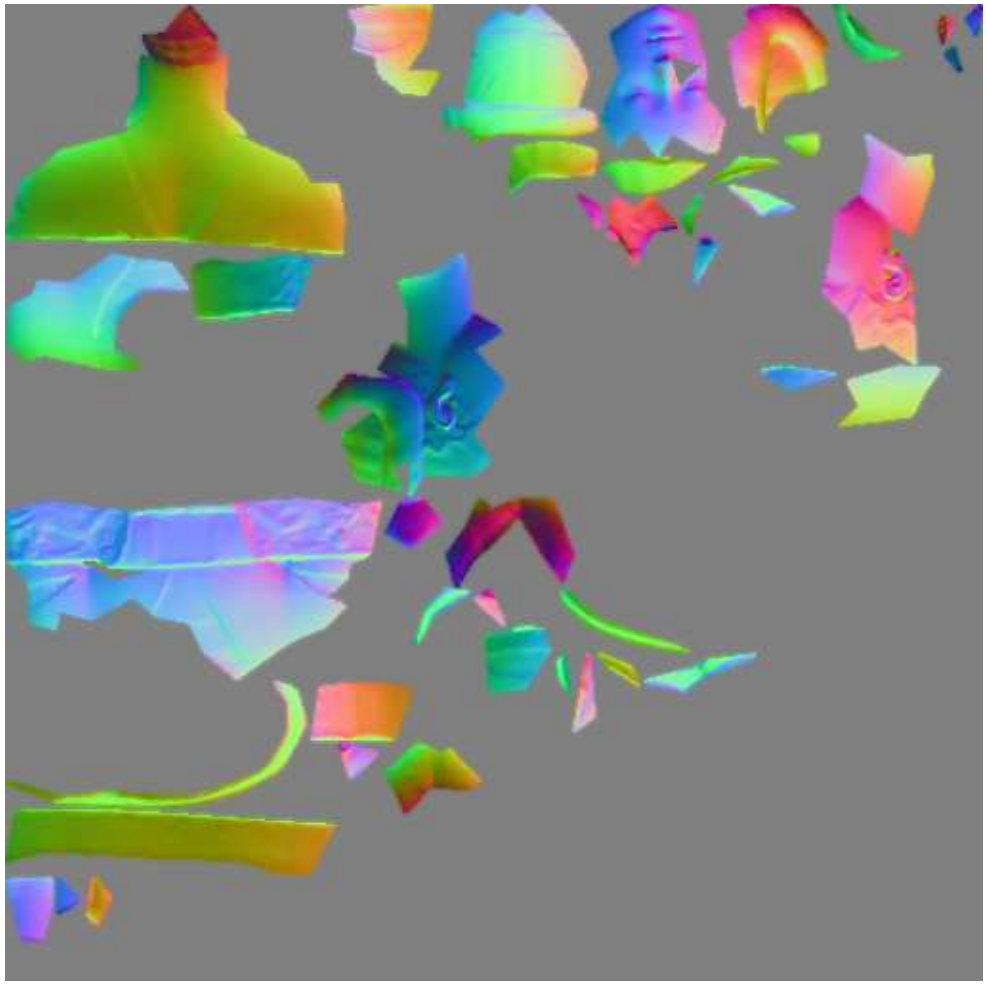
Bump Mapping

Bump Map



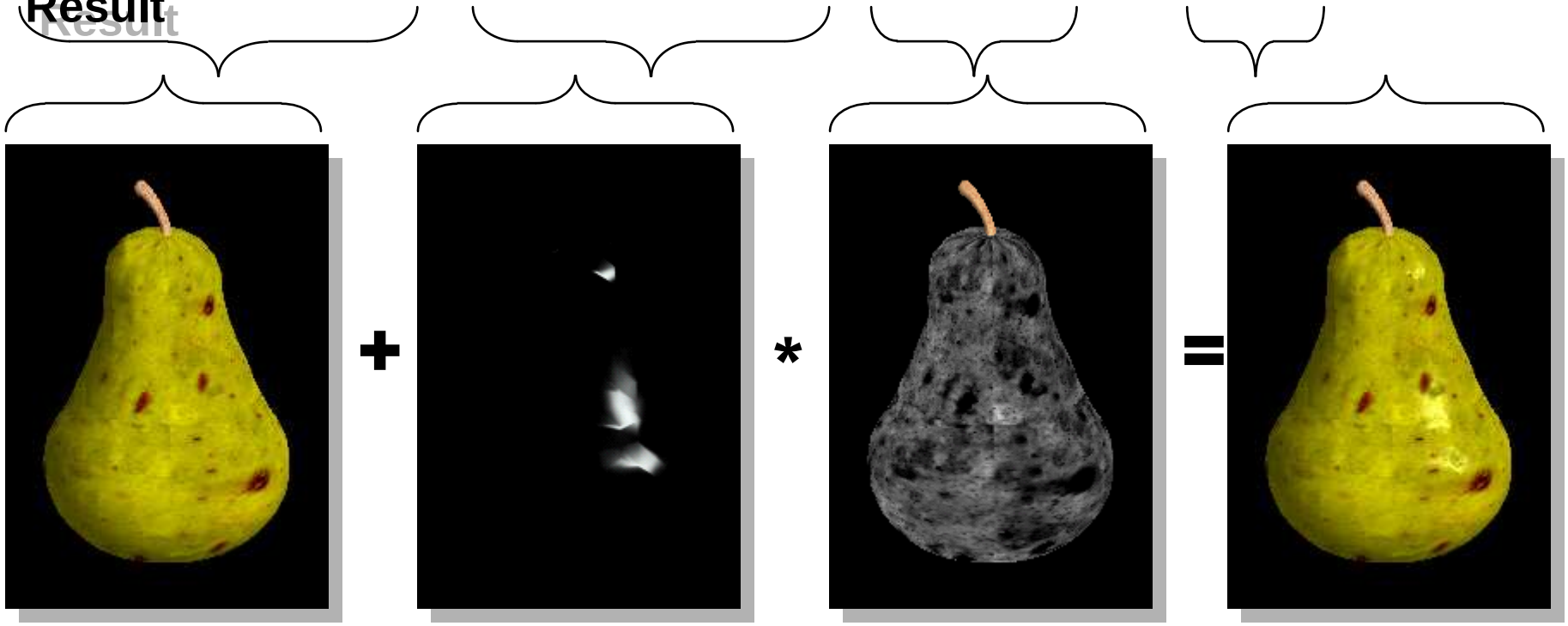
Bumpmapping

- Le normali sono codificate nello spazio rgb signed
- La direzione della luce viene passata come colore corrente



Gloss mapping

Base * Diffuse Light + Environment Map * Gloss Map = Result



Gouraud * Base Texture

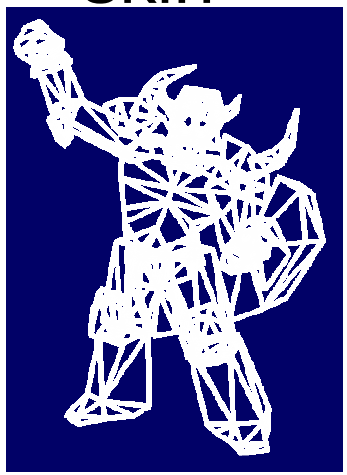
Environment or Specular Map

Gloss Map

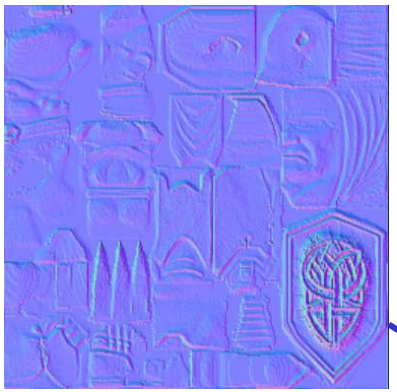
Gloss Mapped Object

Per pixel lighting

- Bump-map model with bump, gloss, & decal skin



Interpolated keyframe of model geometry



Normal map

Decal skin (not pre-lit)

Gloss skin

Per pixel lighting

- diffuse & specular bump mapping in tre “passate” di rendering

1st pass
Diffuse

2nd pass
Decal

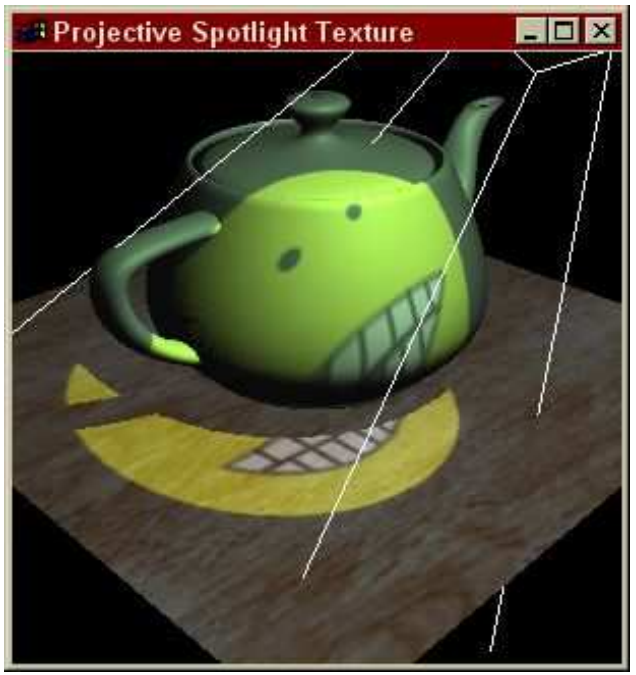
3rd pass
Glossed Specular

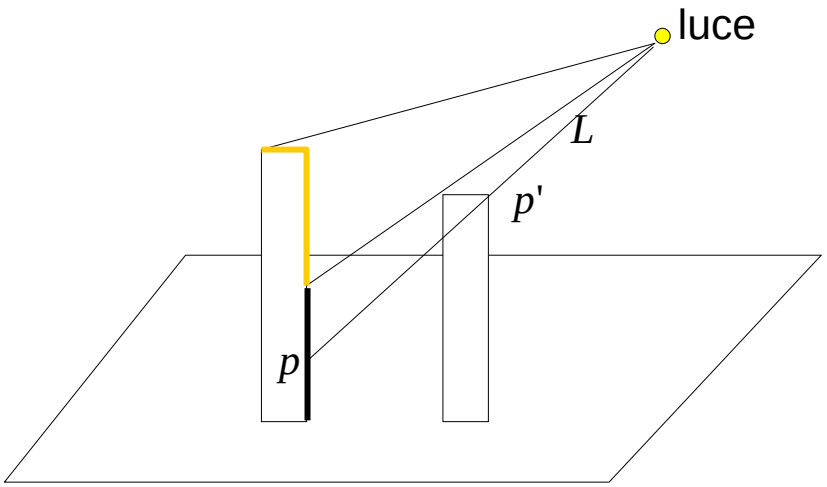
Final result!

Ombre portate: Shadow Mapping

C’erano anche nell’esempio della spotlight!

• lo *shadow mapping* UN modo di realizzarlo





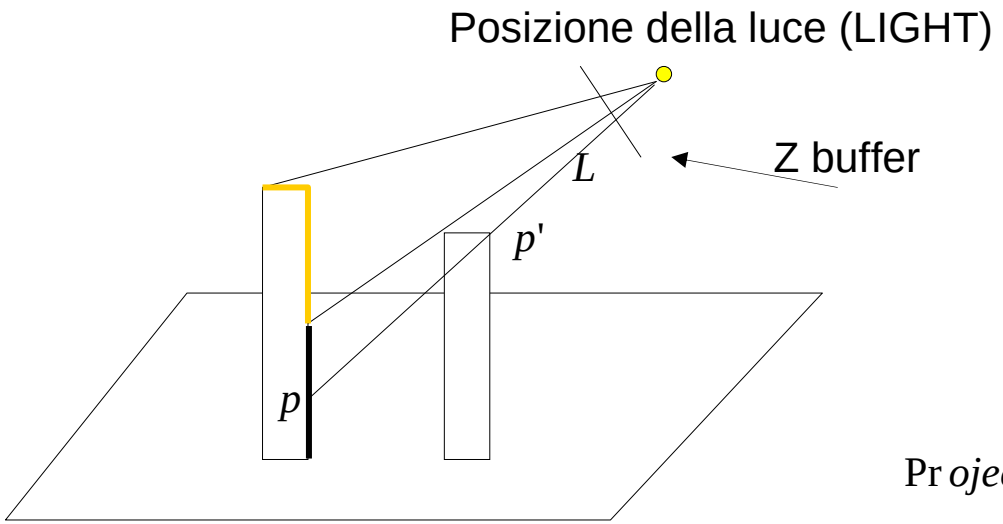
- Punto di vista

Il principio:
Il punto p è in ombra perché c’è un punto p’ nella linea da p alla luce che è più vicino di p alla luce

Ombre portate: Shadow Mapping

- L’idea: visualizzo la scena dal punto di vista della luce e mi tengo lo z-buffer
- Visualizzato la scena dal punto di vista voluto e per ogni frammento che produco
  - Riproietto il frammento in coordinate mondo
  - Riproietto il risultato come visto dalla luce
  - Confronto la z della proiezione con quello che ho scritto nel depth buffer: se è maggiore vuol dire che è in ombra
  - Se è in ombra posso “scurire” un po’ il pixel

Ombre portate: Shadow Mapping



$$\Pr oject(Un\Pr oject(p_w, POV), LIGHT)_z$$

- Punto di vista (POV)

È un algoritmo multipass:
Un primo passo per il rendering dalla luce
Il secondo per il rendering della scena

Domanda: Se ho N luci?

Texture e scheda grafica