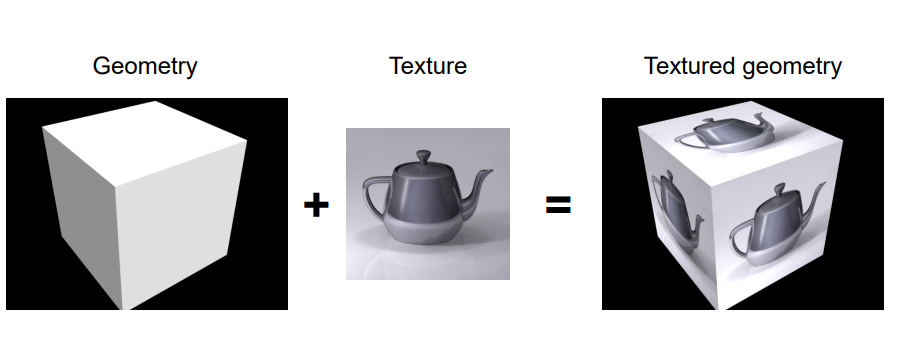
**Texture mapping**

Geometry+texture=textured geometry

La texture viene moltiplicata per il modello di illuminazione



Bisogna definire le coordinate di texturizzazione

Dei punti di ancoraggio dell’immagine che vanno messe sulla mesh

In fase di rasterizzazione viene aggiunta la texture sulla mesh.

La vram principalmente serve a contenere i dati delle texture

Texel sono i dati/pixel della texture e una volta sullo schermo si hanno i pixel

Durante la rasterizzazione, ogni colore texel\* viene moltiplicato per il colore calcolato dal modello di illuminazione o specificato direttamente dal programmatore: – È possibile modificare questa impostazione predefinita tramite glTexEnv\*();

le texture possono essere rettangolari, in passato le dimensioni dovevano essere potenze di 2, ora è stata tolta questa limitazione, ma in certi casi ha dei vantaggi.

Le texture vanno normalizzate tra 0 e 1, questo consente di avare le texture indipendenti dal modello

Consente inoltre di avere la stessa texture ma in risoluzioni diverse.

Immagine che contiene testo, pentola, utensile da cucina

Descrizione generata automaticamente

La coordinata (0,0) delle texture è in basso a sinistra, cambia in base alla libreria

Immagine che contiene testo

Descrizione generata automaticamente

**Interpolazione durante la rasterizzazione**

Non usa un interpolazione lineare

Immagine che contiene testo, clipart

Descrizione generata automaticamente

Se si interpolasse linearmente si avrebbe il risultato centrale.

Nelle coordinate screen coordinate la z oltre alla profondità per lo z buffer serve alle texture per capire come mettere la texture in un contesto tridimensionale

Immagine che contiene testo, orologio, arancia

Descrizione generata automaticamente

OpenGL calcola l’interpolazione con la formula sopra, che sfrutta la coordinata z.

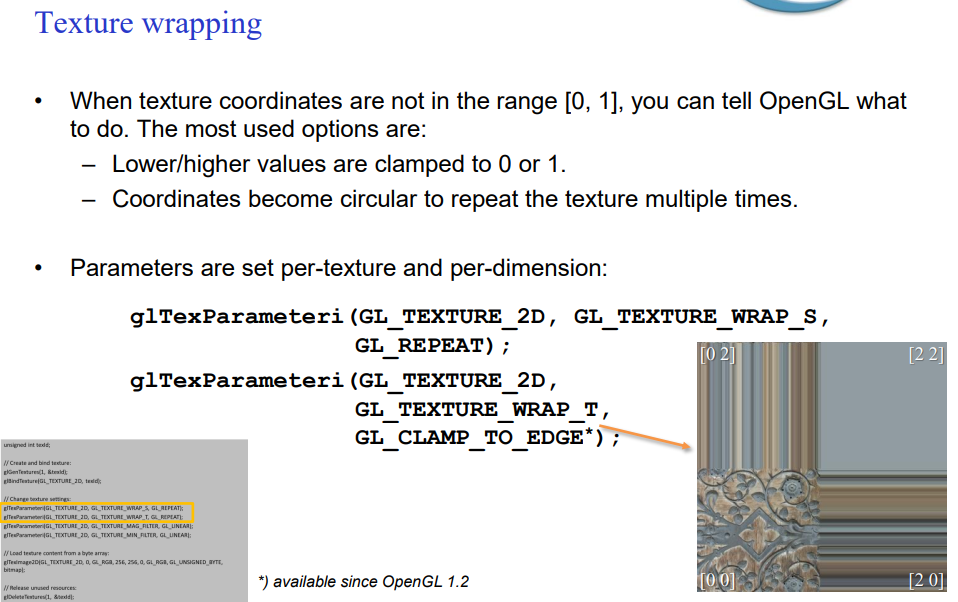
**Caricare immagini sulla GPU**

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente



Con gl\_repeat guarda solo la parte decimale ed ignora la parte intera, utile per ripetere la texture

Con gl\_clamp\_to\_edge stira l’immagine l’ultima parte renderizzata sul lato i valori fuori dal range 0-1.

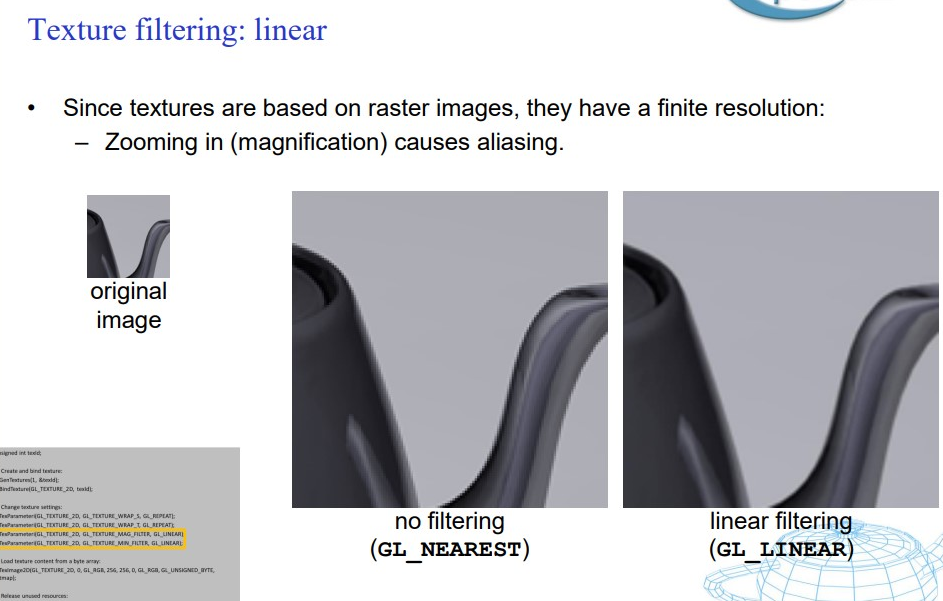
**Texture tile**: sono texture disegnate in modo da non distinguere la ripetizione di essa

**Filtering**

Le texture sono già rasterizzati.

Se si ha una texture nxm e la applico ad un oggetto se si ci allontana con la camera essa sarà più piccola, minificazione

Se mi avvicino ottengo una magnificazione, ho bisogno di più pixel per replicare l’immagine

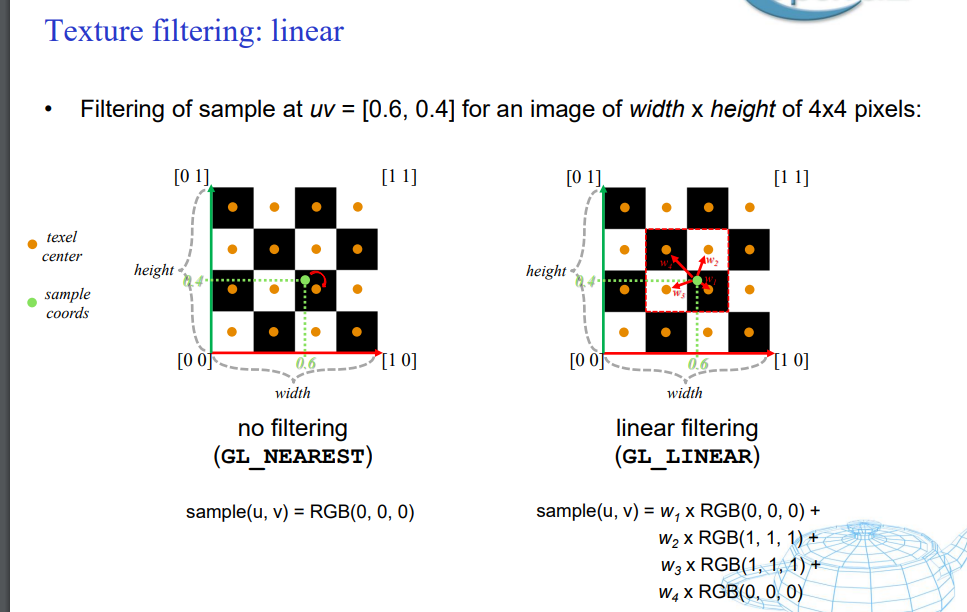


Quest’ultima avviene tramite un’interpolazione lineare dei pixel attorno

Immagine che contiene testo, utensile da cucina, pentola, interni

Descrizione generata automaticamente

Opengl prende il centro del pixel più vicino nel nearest mentre per il linear fa una media pesata dei 4 pixel sommati e pesati per i pesi calcolati.



Di default è nearest

**Mipmapping**

Vengono creati dei livelli di texturizzazione già filtrati e vengono applicati da opengl in base alla dimensione del modello su cui bisogna applicare la texture.

Cerca la mipmap più vicina alla dimensione su schermo del modello da texturizzare

Le texture sono pre-filtrate diverse volte per ottenere un miglio LOD (levels of details)

Il mipmapping si può fare in maniera procedurale partendo dall’immagine più definita oppure precaricare i livelli già renderizzati.

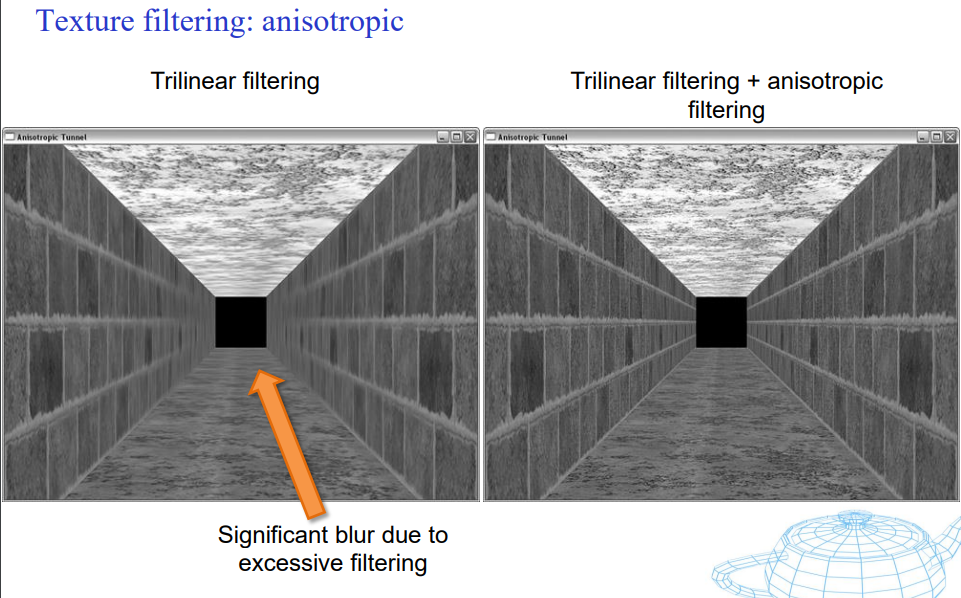
Non c’è un eccessivo consumo di memoria.

Immagine che contiene testo

Descrizione generata automaticamente

Si può dire a openGL di usare entrambe le immagini se si trova in una dimensione che sta tra due mipmap e fa la media ed interpola i valori rgb delle due per ottenere la texture corrispondente.

**Filtro anisotropico**



Consente di mantenere la stabilità del filtro trilineare e mantenere la qualità.

È un filtro parecchio pesante perché aumenta i texel per fare le medie.

E ciò dipende anche da quanto è grande il triangolo.