Corso di grafica 3D con C++ e OpenGL

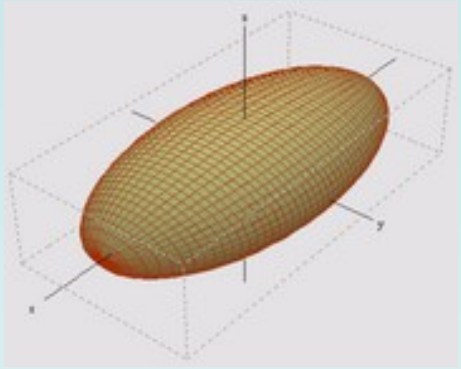
Alessandro Bugatti – ITIS B. Castelli

Utilizzo di quadriche

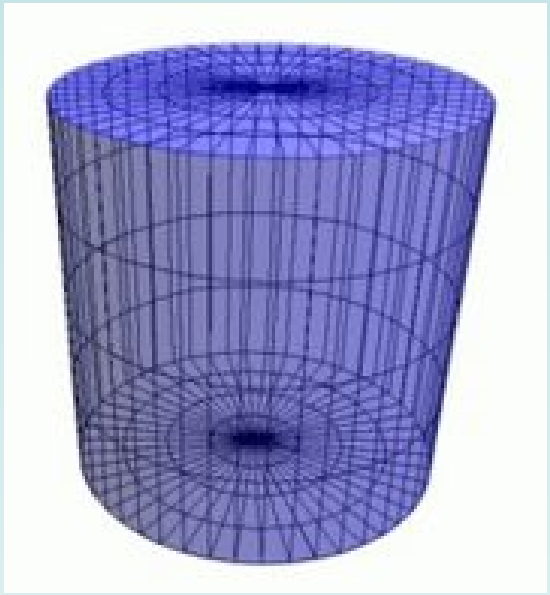
- Una quadrica in 3D è ogni superficie rappresentata da un'equazione polinomiale del secondo ordine nelle variabili spaziali (coordinate).

$$\pm\frac{x^2}{a^2} \pm \frac{y^2}{b^2} \pm \frac{z^2}{c^2} = 1$$

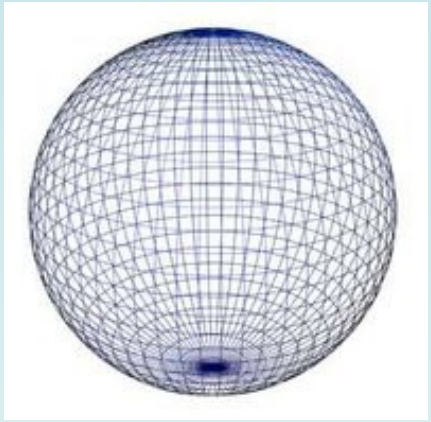
Esempi di quadriche



Ellissoide



Cilindro



Sfera

Libreria GLU

- La libreria GLU (GL Utility) racchiude una serie di funzioni “comode”
- Prima si crea una quadrica
  - GLUquadricObj* **gluNewQuadric**( ) che crea memoria per i parametri della superficie e ritorna un puntatore
- Poi si sceglie il tipo

Esempio delle sfera

- void **gluSphere**(GLUquadricObj ***qobj*, GLdouble *radius*, GLint *slices*, GLint *stacks*)

dove:

  - radius è il raggio della sfera
  - slices sono le fette in direzione longitudinale
  - stacks sono le fette in direzione latitudinale

Esempio dei cilindri

- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks);

dove:

  - base è il raggio nella parte inferiore
  - top è il raggio nella parte superiore
  - height è l'altezza
  - slices sono le fette in direzione longitudinale
  - stacks sono le fette in direzione latitudinale

Texture mapping

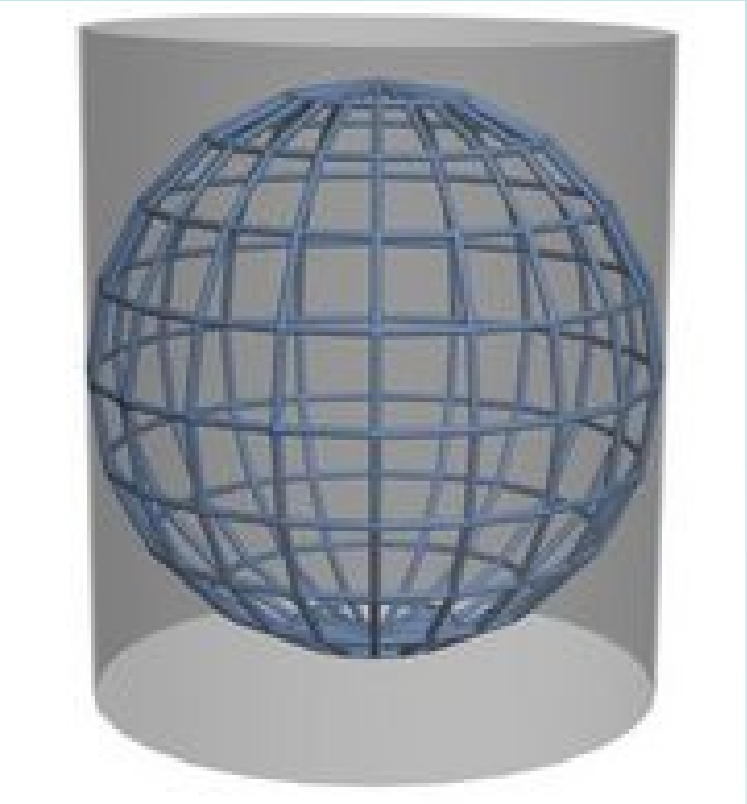
- Per texture mapping si intende l'applicazione di una "tessitura" su un oggetto per dargli un aspetto più realistico
- Generalmente si tratta di un'immagine bitmap che viene applicata su di un modello come fosse una "pelle"

Caricamento della bitmap

- Il primo passo è quello di caricare un'immagine in memoria. Questo può essere fatto " a mano" oppure usando una qualsiasi libreria che lo permetta (noi useremo la SDL)
- L'immagine deve essere quadrata con dimensioni potenza del 2 (per motivi di efficienza)

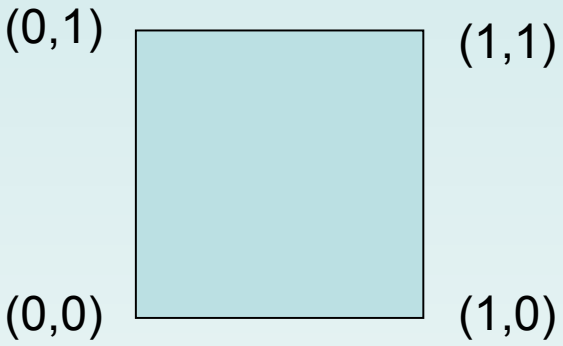
Proiezione della bitmap

- Bisogna proiettare una superficie su di un volume.
- E' lo stesso problema che si presenta ai cartografi quando devono rappresentare la Terra



Mapping

- In generale noi dovremo posizione la bitmap su una superficie piana
- Dovremo far corrispondere ogni angolo della bitmap con un vertice del nostro modello



Funzioni per le texture

glEnable(GL_TEXTURE_2D);

- Funzione che abilita il texturing: senza questa chiamata non funziona niente
- Le texture 2D sono quelle più utilizzate, ma esistono anche texture 1D (hanno altezza uguale a 1) e 3D (vengono utilizzate soprattutto in campo medico)

Funzioni per le texture

glGenTextures(Gluint n, Gluint *v);

- Crea n identificativi per le texture (possono essere più di una) e li memorizza nel vettore puntato da v

  glBindTexture(GL_TEXTURE_2D, Gluint i);
- Associa ad un identificativo i una texture (in questo caso 2D)

Funzioni per le texture

void **glTexImage2D**(GLenum *target*, GLint *level*, GLint *components*, GLsizei *width*, GLsizei *height*, GLint *border*, GLenum *format*, GLenum *type*, const GLvoid **pixels*)

- Target è il tipo di di texture (1D, 2D, 3D)
- Level è il livello di dettaglio
- Components è il numero di colori della texture
- Width e heigth sono la larghezza e l'altezza della bitmap

Funzione per le texture

- Border specifica la larghezza del bordo (0 o 1)
- Format è il formato in cui sono memorizzati i colori
- Type è il tipo di dati in cui sono memorizzati i colori
- Pixels è il puntatore ai pixel della bitmap

Funzione per le texture

glTexCoord2f(float u, float v);

- Associa ad ogni spigolo dell'immagine un vertice del modello
- Deve essere seguita da una chiamata ad una glVertex, che sarà il vertice cui associare lo spigolo dell'immagine
- Eventualmente adatta l'immagine alla forma del modello