

# FONDAMENTI DI COMPUTER GRAPHICS LM

## VISUALIZZAZIONE MODELLI GEOMETRICI

---

Questa esercitazione puo' essere eseguita sia in ambiente Windows che Linux. Dopo aver scaricato i file necessari dalla pagina web del docente, compilare ed eseguire il programma fornito.

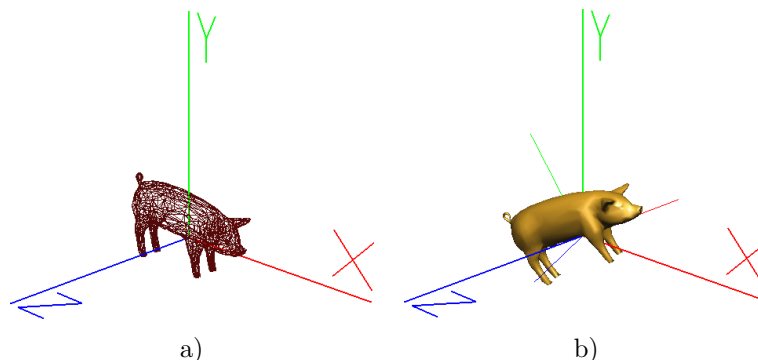


Figure 1: b) Uno screenshot dell'applicazione model\_viewer; c) il rendering in model\_viewer del modello pig.m in modalita' smooth dopo aver sviluppato il calcolo delle normali ai vertici, attivato lo shading smooth, cambiato il materiale e ruotato il modello

### 1. model\_viewer - Caricamento e visualizzazione modelli geometrici

L'applicazione model\_viewer carica e visualizza oggetti mesh poligonali memorizzati in file con estensione .m. La directory **data** contiene vari file di modelli a mesh poligonali.

A partire dal programma fornito model\_viewer.c, che presenta un menu a tendina al click del bottone destro del mouse sulla finestra, creare un ambiente interattivo di caricamento e visualizzazione dei modelli geometrici sia primitive GLU sia mesh poligonali (in formato .m) secondo le seguenti direttive:

- Caricamento e visualizzazione modelli geometrici di tipo mesh in formato .m
- Visualizzazione superfici quadriche dalla libreria GLU (es. Sfere, cilindri, tori)
- Verificare la gestione della visualizzazione dei modelli poligonali a mesh tramite display list.
- Gestire con display list anche la visualizzazione di almeno due quadriche.
- Calcolo e memorizzazione delle normali ai vertici per i modelli mesh poligonali. Visualizzazione con normali ai vertici in modalita' smooth ( `glShadeModel(GL_SMOOTH)` ).

### 2. model\_viewer - Controllo interattivo della scena

Alla pressione del tasto destro del mouse compare un menu pop up. Si puo' interagire con la scena selezionando gli item del menu e successivamente, tramite input da tastiera, modificare i valori associati. Le funzionalita' che il sistema interattivo deve fornire sono elencate nella tabella 1,

<i><b>Funzionalita'</b></i>	<i><b>Descrizione</b></i>	<i><b>Tasti</b></i>	<i><b>Default</b></i>
<b>Change eye point</b>	modifica le coordinate del punto di vista	x,X,y,Y,z,Z	(8.8, 4.9, 9.0)
Change ref. point	modifica le coordinate del punto di riferimento del sistema	x,X,y,Y,z,Z	(0.0, 0.0, 0.0)
Change up vector	modifica le coordinate del vettore "alto" della camera	x,X,y,Y,z,Z	(0.0, 1.0, 0.0)
Change light position	modifica le coordinate del vettore posizione della fonte luminosa	x,X,y,Y,z,Z	(5.0, 5.0, 5.0)
<b>Rotate model</b>	ruota il modello	x,X,y,Y,z,Z	(0.0, 0.0, 0.0)
<b>Zoom in/out</b>	modifica l'angolo (field of view) al vertice della piramide di vista	f,F	20.00
<b>Proiezione</b>	modifica il tipo di proiezione e quindi il volume di vista	-	prospettica
<b>Culling</b>	abilita/ disabilita il back face culling	-	off
<b>Wireframe</b>	cambia modalita' di rendering dei poligoni	-	GL_LINE
<b>Shading</b>	Cambia modalita' di shading	-	GL_FLAT
Materials	Cambia il materiale associato agli oggetti	-	red.plastic
<b>Trackball</b>	implementare una trackball virtuale per il controllo della camera. Vedi nota <sup>1</sup>	-	-
<b>Camera Motion</b>	movimento della camera lungo un percorso. Vedi nota <sup>2</sup>	-	-
Print system status	stampa sul terminale i dati correnti del sistema	-	-
Reset	riporta il sistema allo stato iniziale	-	-
Quit	chiude il programma	Esc	-

Table 1: Tabella delle funzionalita' sviluppate e da sviluppare (indicate in grassetto) nell'applicazione model\_viewer

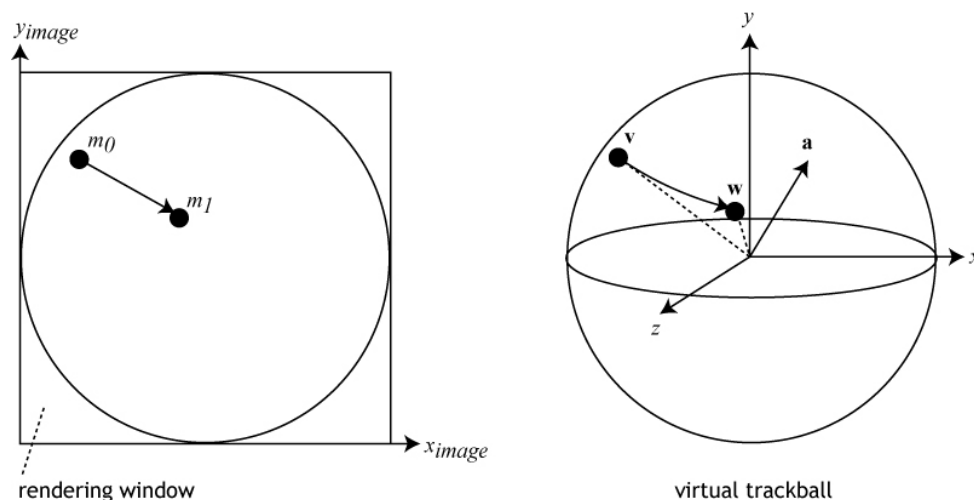


Figure 2: La trackball virtuale traduce il movimento del mouse in una matrice di rotazione

alcune sono già realizzate (prendete visione di come) e altre invece sono lasciate a voi da realizzare (indicate in grassetto).

### 3. **OPZIONALE: Manipolazione dello stack delle matrici**

Modificare l'applicazione model\_viewer sviluppando i seguenti punti:

- posizionare almeno tre diversi modelli geometrici nella scena sfruttando le funzioni OpenGL: `glPushMatrix()`, `glPopMatrix()`.
- permettere la traslazione e rotazione dei **singoli** oggetti rispetto ai sistemi di riferimento WCS, OCS e VCS ( World, Object e View Coordinate Systems ). I singoli oggetti devono essere selezionabili tramite i tasti 1, 2, 3, etc. . Le trasformazioni di traslazione o rotazione devono essere applicate tramite i tasti 'x', 'X', 'y', 'Y', 'z', 'Z'. La selezione del tipo di trasformazione (traslazione / rotazione) e del sistema di riferimento rispetto al quale eseguire tale trasformazione deve essere selezionabile tramite i tasti:
  - 'o' traslazione rispetto all'OCS
  - 'O' rotazione rispetto all'OCS
  - 'w' traslazione rispetto al WCS
  - 'W' rotazione rispetto al WCS
  - 'v' traslazione rispetto al VCS
  - 'V' rotazione rispetto al VCS (con assi di rotazione aventi direzione del VCS e passanti per l'origine del VCS oppure direzione del VCS e passanti per l'origine dell'OCS)

<sup>2</sup>La trackball virtuale permette di ruotare un oggetto interattivamente utilizzando il mouse. La figura 2 illustra intuitivamente come tradurre il movimento del mouse in un asse ed un angolo di rotazione.  $m_0$  ed  $m_1$  sono due posizioni consecutive del mouse e definiscono due punti  $v$  e  $w$  sulla sfera 3D virtuale. Il loro prodotto vettoriale definisce l'asse di rotazione  $a = v \times w$ , mentre l'angolo di rotazione può essere calcolato dal loro prodotto scalare.

<sup>2</sup>Muovere la camera virtuale lungo un percorso in modalità look at (mediante curva chiusa di Bézier) in modo che la camera si muova lungo un percorso con centro di interesse l'oggetto in scena