Lezione 4: Grafica 3D*(II)

Informatica Multimediale

Docente:

_Umberto Castellani_

*I lucidi sono tratti da una lezione di Maura Melotti (m.melotti@cineca.it)*
Rendering

- What is rendering?
- Rendering methods.
- What color is the object at that pixel?
- Alpha Buffering
- Texture mapping.
Cos’è il rendering

- Il rendering è il processo di conversione degli elementi descrittivi della scena in una immagine 2D
Metodi di rendering

- Wire-frame
- Painter’s algorithm
- Z-Buffer
- Ray Tracing
- Radiosity
Wire frame

- Only the edges of objects are drawn.
- Faster and useful for quick previews.
- Some mechanical drawings are clearer when only the edges of objects are drawn.
Algoritmo del pittore

- Problem: how to form an image from a scene such that object mask one other correctly?
Algoritmo del pittore

- This technique is similar to the way a painter creates a painting:
- start painting the most distant elements, such as the sky, and over-paint these with objects that are closer.

1. sort objects by distance from eye
2. render objects farthest from the eye first
3. continue rendering all objects from back to front
 painter’s algorithm fails if object intersect!
An object that intersects the grid looks like this:

But it should look like this!

Solution: Z-Buffer algorithm
Z-Buffer

- At each pixel is associated a Z-value that is the distance from the eye to the point on the object represented by the pixel.
- A Z-buffer is used to record the Z-value of pixels.
- Before any polygons are drawn all the Z-value in the Z-Buffer are set to indicate the farthest distance from the eyes.
- Before a new value is written in the Z buffer the pixel’s Z-value, stored in the Z-buffer, is compared to the new Z-value.
- If the new Z-value represents a closer distance to the eyes the new value is written in the Z Buffer.
Z-Buffer

Important properties:

Z-buffer is order independent.
Z-buffer rendering can be done incrementally.
Z-buffer

- Z-values are usually stored in 16 bits, which results in more than 65000 different depth values.
- It is the most common solid surface rendering technique today.
- It is simple enough to be implemented directly in inexpensive hardware.
- But …. it doesn’t address how light really interacts between object (shadows, refraction, ….) .
Ray-tracing

Forward Propagation:
• A ray of light is thrown from each light sources.
• If the ray hits an object, refraction and reflection effects must be calculated.
• This can happen many times, depending on the number of objects in the scene.
• Rays that reach the camera (the eye point) are few in comparison with the total rays calculated (too much computational cost!!!).

Back Propagation
Ray-tracing

Back Propagation:

• A ray is shot from the eye point in the view direction in order to calculate the color of a pixel.

• This process doesn’t stop when the first object is encountered.

• Unless the object is a light source more rays are shot in each direction where light might be come from.
Ray-tracing

Fig.1: Z-Buffer rendering

Fig.2: Ray Tracing rendering
Ray-tracing

Fig.1: Z-Buffer rendering

Fig.2: Ray Tracing rendering
Ray-tracing vs. Z-buffer

Ray tracing is **slow**, no real-time:
- the algorithm is **very complex**, generally computed at software level
- it works **per pixel** instead of **per object**.

Z-Buffer algorithm is **faster**, real-time:
- **visit each polygon once**, generally the hardware does the per pixel operations
- for most of today’s scenes, pixel greatly outnumber objects.
Ray-tracing
Radiosity

With ray tracing shadows have sharp edges because the test ray to the light source either gets there or it doesn’t.

In real life we can see lots of soft shadows because light is reflected by many objects (secondary light).

Radiosity can model the diffuse reflection (secondary scatter).
Radiosity

The radiosity algorithm computes what portion of the light from each polygon reaches each of the others.

The light from the bright polygons causes other polygons to be lit.

These polygons, in turn, shine on other polygons, which then shine on more polygons…

This process of lighting up polygons continues until things are settle down.
Radiosity
Radiosity
Esempi di rendering

Ray tracing
Esempi di rendering

Ray tracing + ombre sfumate
Esempi di rendering + caustiche
Esempi di rendering

+ global illumination
What color is the object at that pixel?

The process of eventually coming up with pixel color values is called **shading**.

- Flat Shading
- Linear (Gourand) Shading
What color is the object at that pixel?

Flat Shading

The color values from the polygon vertices are averaged, and all the polygon’s pixels are set to this fixed (flat) color value.

Each pixel is exactly the same color throughout the polygon.
What color is the object at that pixel?

**Linear (Gourand) Shading**

The color values from the polygon vertices are linear interpolated, and the rate of change of the color value is held constant for all the polygon’s pixels.
In **alpha buffering** an **alpha value** is kept per pixel (in addition to color value – rgbα – and Z-value).

- The alpha value (**transparency value**) is an opacity fraction that indicates how opaque (or transparent) the pixel is.

- The alpha value is commonly stored in **8 bits**: 0 means fully transparent, 255 means fully opaque.
Alpha Buffering
Alpha Buffering

+ 

=
Texture Mapping

The process of mapping (wrap) an image over an object.
Texture Mapping

During the modeling step the user defines texture coordinates across the object.

The texture coordinates are interpolated across each polygon instead of the final color values.
Texture Mapping

Texture mapping adds significant visual detail to the image without the need for modeling more complex geometry.
Rendering review

3D scene description ➔ Image

Texture Mapping

Antialiasing

Shading: figuring out pixel color