Visibility

Hidden-line removal

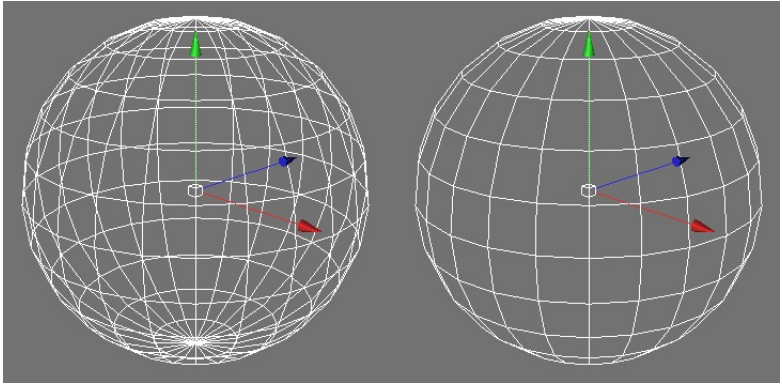
Marcel Makovník

Department of Algebra and Geometry
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava, Slovakia

5 December 2019

Main idea

- Erase those faces (edges), which cannot be seen by a viewer.



How?

Roberts' algorithm

- Roberts, Lawrence G., Machine perception of three-dimensional solids, 1963

- **INPUT:**
  - A mesh, which is:
    - non-oriented – e.g. represented by a face-based data structure,
    - without a boundary.
  - The position of the camera V and the viewing direction $\vec{v}$.
- **OUTPUT:**
  - a list of the visible faces,
  - a list of the visible edges.

Roberts' algorithm – processing

For each face:

1. find the coinciding plane,
2. find the outer-pointing normal of the face,
3. determine (in)visible faces,
4. classify edges.

Roberts' algorithm – processing – coinciding plane

- A plane ρ determined by points $P_1 = (x_1, y_1, z_1)^\top,\ P_2 = (x_2, y_2, z_2)^\top,\ P_3 = (x_3, y_3, z_3)^\top$ is determined by

$$\langle (P_2 - P_1) \times (P_3 - P_1),\ X - P_1 \rangle = (X - P_1,\ P_2 - P_1,\ P_3 - P_1) =$$

$$\det \begin{pmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{pmatrix} = \det \begin{pmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{pmatrix} = 0.$$

Roberts' algorithm – processing – coinciding plane

- By using the Laplace expansion along the first row of the determinant

$$\det \begin{pmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{pmatrix}$$

we obtain the general equation of ρ, more precisely

$$\rho : \; Ax + By + Cz + D = 0,$$

where

$$A = \det \begin{pmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{pmatrix}, \; B = \det \begin{pmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{pmatrix},$$

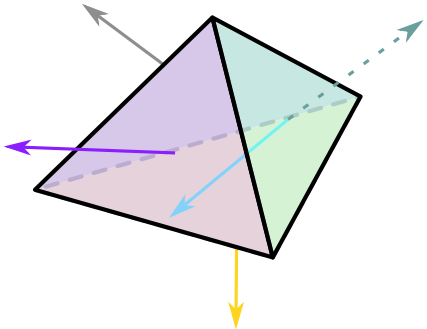
$$C = \det \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix}, \; D = \det \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix}.$$

Roberts' algorithm – processing – outward-pointing normal vectors

- We are working in the positively oriented Euclidean space.
- The computed normal vector $\vec{n} = (A, B, C)^\top$ does not have to be pointing outwards by default.
- Take a point inside the object, e.g. a center

$$T = \frac{1}{n} \sum_{i=1}^{n} P_i.$$

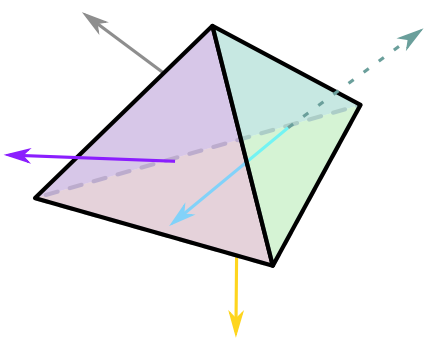
- if $\rho(T) < 0 \rightsquigarrow$ outwards
- if $\rho(T) > 0 \rightsquigarrow$ inwards $\rightsquigarrow$ invert



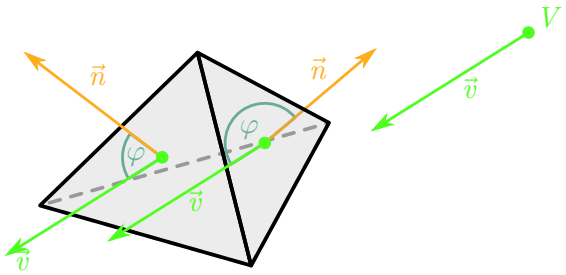
Roberts' algorithm – processing – where is the camera?

- We need to check, if the camera V is not inside the object (in such case no face would be visible).
- The camera is inside the object iff for each plane ρ (defined by the outward-pointing normal), passing through the faces of the object, $\rho(V) \leq 0$.

- Thus it is enough to check if for **some** plane ρ is $\rho(V) > 0$, if we want to check whether the camera is outside the object.



Roberts' algorithm – processing – front/back face



- if $\langle \vec{v}, \vec{n} \rangle < 0 \rightsquigarrow \varphi > \frac{\pi}{2} \rightsquigarrow$ front,
- if $\langle \vec{v}, \vec{n} \rangle > 0 \rightsquigarrow \varphi < \frac{\pi}{2} \rightsquigarrow$ back,
- if $\langle \vec{v}, \vec{n} \rangle = 0 \rightsquigarrow \varphi = \frac{\pi}{2} \rightsquigarrow \rho$ is parallel with $\vec{v} \rightsquigarrow$ back.

Roberts' algorithm – processing – visible faces/edges

Convex objects:

- front face $\rightsquigarrow$ visible
- back face $\rightsquigarrow$ invisible

- front face $\cap$ front face $\rightsquigarrow$ visible edge
- back face $\cap$ front face $\rightsquigarrow$ visible *(contour)* edge
- back face $\cap$ back face $\rightsquigarrow$ invisible edge

Roberts' algorithm – processing – visible faces/edges

Non-convex objects:

- otherwise visible faces/edges might be "shaded" by the others

- front face $\rightsquigarrow$ potentially visible
- back face $\rightsquigarrow$ invisible

- front face $\cap$ front face $\rightsquigarrow$ potentially visible edge
- back face $\cap$ front face $\rightsquigarrow$ potentially visible *(contour)* edge
- back face $\cap$ back face $\rightsquigarrow$ invisible edge