SESSION 11

Subdivision Surfaces

- Adaptative representation,
- Polygonal but smooth
- Not parametric "Based on algorithm"
- Diference schemes.

Steps

1. Refinement: Create and reconect new vertexes from Control Net.
2. Smooth: Calculate position of new vertexes with a scheme. The new mesh is called Limit Surface.

Scheme

Polyhedral: Limit surface dont change the vertexes.

Butterfly: Limit surface is bigger than original.

Catmull: Create new surface inside original mesh.

Steps of Catmull Alg

1. For each face in the control net add a face point as the average of all.
2. For each edge, add a new edge point with the average of the 2 vertexes of the edge.
3. Move the original vertexes of the control net. This form new edges.

$$v^{i+1} = \frac{N}{N+1} v^i + \frac{1}{N^2-1} \left(\sum_{j=0}^{N-1} (e_j^i) \right)$$

N = Number of edges of each vertex

v_i = Vertex at recursion level i

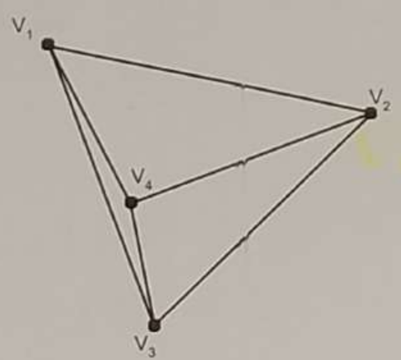
e_j^i = Neighbor edge vertexes of v

f_j^i = Neighbor face vertexes of v

Computer Graphics

Subdivision Surfaces Exercise

Given the tetrahedron of the next figure and the vertexes $V_1=(-1,-1,1)$, $V_2=(1,1,1)$, $V_3=(1,-1,-1)$ and $V_4=(1,-1,1)$ which define three faces (the face $V_1V_2V_3$ does not exist, it is an empty face), calculate the result of applying an iteration of Catmull-Clark subdivision method. The particular case of the Catmull-Clark algorithm is defined as follows:

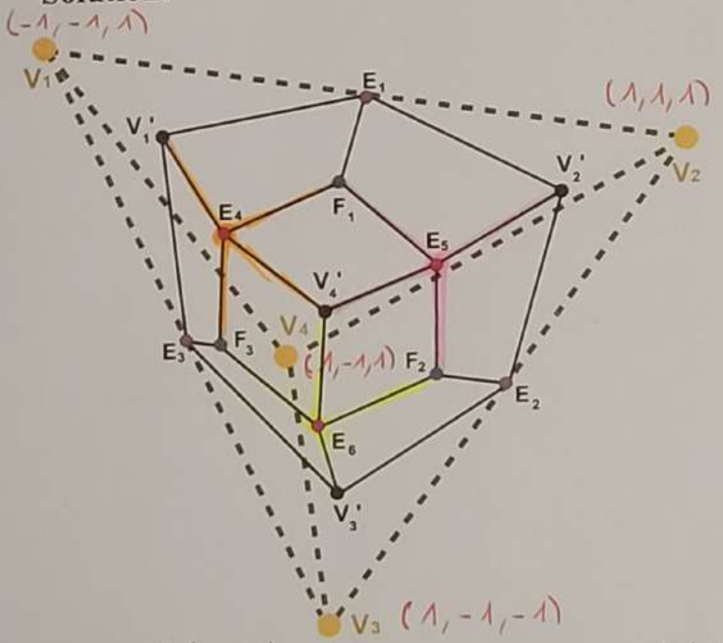


a) If an edge does not have two adjacent faces (i.e. it is part of an empty face), the new vertex of the edge is calculated as the average point of the original edge (do not take into account the face vertexes).

b) To compute the new position of the vertexes, if some of the adjacent faces are empty, use only the adjacent vertexes that share the empty face. In this case, utilize the next version of the equation (see also Slide 8):

$$v^{i+1} = \frac{N}{N+1} v^i + \frac{1}{N^2-1}\left(\sum_{j=0}^{N-1} \left(e_j^i\right)\right)$$

Solution:



Repositioned Control Net:

$V_1' = (-6/8, -7/8, 7/8)$

$V_2' = (7/8, 6/8, 7/8)$

$V_3' = (7/8, -7/8, -6/8)$

$V_4' = (8/9, -8/9, 8/9)$

New Edge Vertexes:

$E_1 = (0, 0, 1)$

$E_2 = (1, 0, 0)$

$E_3 = (0, -1, 0)$

$E_4 = (2/3, -10/3, 10/3)/4$

$E_5 = (10/3, -2/3, 10/3)/4$

$E_6 = (10/3, -10/3, 2/3)/4$

New Face Vertexes:

$F_1 = (1/3, -1/3, 1)$

$F_2 = (1, -1/3, 1/3)$

$F_3 = (1/3, -1, 1/3)$

$$F_x = \frac{V_1 + V_2 + V_n}{n} \qquad E_{1,2,3} = \frac{V_x + V_y}{2}$$

$$E_{4,5,6} = \frac{F_x + F_y + V_z + V_a}{4} \qquad V'_{1,2,3} = \frac{3}{4} \cdot V_1 + \frac{1}{8} \cdot (E_1 + E_3)$$

$$V_4 = \frac{3-2}{3} \cdot (1,-1,1) + \frac{1}{9}(E_4 + E_5 + E_6 + F_1 + F_2 + F_3)$$

SESSION 12

Implicit Surfaces

Metasurfaces

Implicit Surface: A point p belongs to surface if:

$f(p) = T$ T: Threshold value

Blobs

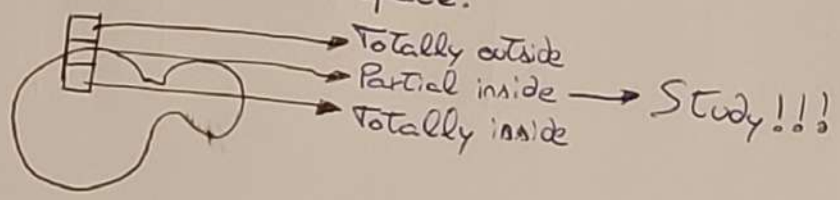
```
for each blob in the scene{
    d= distance (P, blop.pos);
    if (d < blop.radious) {
        fall off = 1-(d/blob.radius);
        energy += blob.strength · falloff²;
    }
}
```

Maching Cubes

Is Known as 3D mesh generating algorithm

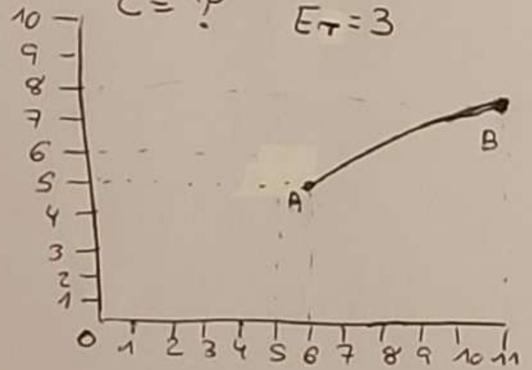
Problem: How to draw the surface of the continuous energy field.

Solution: Discretize the space.



Linear interpolation

$A = (6,5)$ $E_A = 1$
$B = (11,8)$ $E_B = 6$
$C = ?$ $E_T = 3$



$\overrightarrow{a,b} = (5,3)$

$$S = \frac{E_T - E_A}{E_B - E_A} = \frac{3-1}{6-1} = \frac{2}{5}$$

$C = A + \overrightarrow{AB} \cdot S$;

$C = (6,5) + (5,3) \cdot \frac{2}{5} = (8, 6'2)$

Grid size & Resolution

The grid size defines the resolution

Grid size = 0.8 → 100 Triangles // Grid size = 0.4 → 500 Triangles

Tethrahedron approach

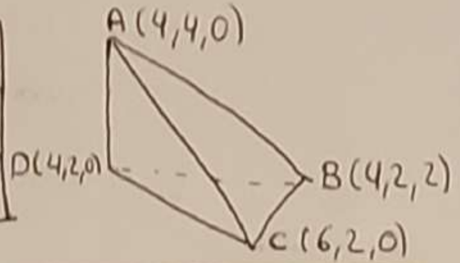
1 vertex inside
Define 1 Triangle

3 vertex inside
Define 1 Triangle

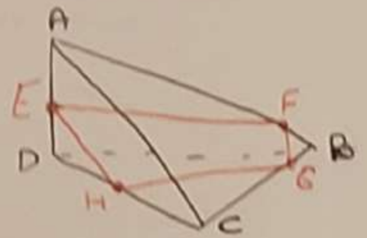
2 vertex inside
Define 2 Triangles

Exercise

	Position	Strength	Radius
Blob1	(0,0,0)	7	6
Blob2	(8,5,0)	5	4



	dB_1	dB_2	E_1	E_2	E_T	
A	$(4,4,0) \rightarrow \sqrt{4^2+4^2} = 5'65$	$(-4,-1,0) \rightarrow \sqrt{4^2+1^2} = 4'12$	$7 \cdot \left(1-\left(\frac{5'65}{6}\right)\right)^2 = 0'024$	$\emptyset$	0'025	<T
B	$(4,2,2) \rightarrow \sqrt{4^2+2^2+2^2} = 4'89$	$(-4,-3,-2) \rightarrow \sqrt{4^2+3^2+2^2} = 5'38$	0'235	$\emptyset$	0'235	>T
C	$(6,2,0) \rightarrow \sqrt{6^2+2^2} = 6'32$	$(-2,-3,0) \rightarrow \sqrt{2^2+3^2} = 3'60$	$\emptyset$	0'05	0'05	<T
D	$(4,2,0) \rightarrow \sqrt{4^2+2^2} = 4'47$	$(-4,-3,0) \rightarrow \sqrt{4^2+3^2} = 5$	0'455	$\emptyset$	0'455	>T



$$S = \frac{E_T - E_x}{E_y - E_x}$$

$$E = D + \overrightarrow{DA} \cdot S = (4,2,0) + (0,2,0) \cdot \frac{0'2 - 0'455}{0'025 - 0'455} = (4, 3'186, 0)$$

$$F = A + \overrightarrow{AB} \cdot S = (4,4,0) + (0,0,2) \cdot \frac{0'2 - 0'025}{0'23 - 0'025} = (4, 4, 1'7073)$$

$$G = B + \overrightarrow{BC} \cdot S = (4,4,2) + (2,-2,2) \cdot \frac{0'2 - 0'23}{0'05 - 0'23} = (4'\hat{3}, 3'\hat{6}, 2'\hat{3})$$

$$H = C + \overrightarrow{CD} \cdot S = (6,2,0) + (-2,0,0) \cdot \frac{0'2 - 0'05}{0'455 - 0'05} = (5'26, 2, 0)$$

SESSION 13

Advanced Modeling, Scanning & 3D printing

STL file format

- Boundary surface file format
- Not constructed solid geometry
- 2 versions: ASCII and binary.

3D Scanning Techniques

- Structured light: Uses projection light patterns and multiple cameras
- Laser scanner: Use laser that analizes the envitoment around the camera.
- Mechanical scanner: A mechanical arm that is used to analize an object, moving a pointer that is used to obtain the shape of the object.
- Photogrammetry Based on taking multiple photos and calculate where the camera taking the photos. Not necessary to see the complete object, at least 70% of the photo's pixels should be the object.

SESSION 14

Materials

Graphics Pipeline: Review

Concept. Pipeline: Application → Transf to screen → Rasterization

Program. Pipeline: Application → Vertex Shader → Triangle setup → Fragment shader

Basic Terminology

Material: Constant properties. Describe how an object looks.

Textures: Variable properties. Add details without geometrical data.

Shading: Determinate the optical qualities of a surface such as color and shininess.

Local illumination

Light interacts with surfaces and return into the camera, One interaction. Low cost.

Global illumination

Multiple lights interaction with multiple surfaces. A lot render time. High cost.

A Simple Lighting Model

Ambient: Independent from light position and observer position.
Emissive: "" Constant component that avoid total darkness.
Diffuse: Based on microroughness of surfaces and reflect light on every direction.
Speculer: Simulates concentrated lightness of surfaces
Result: The addition of all previus.

A very simple equation

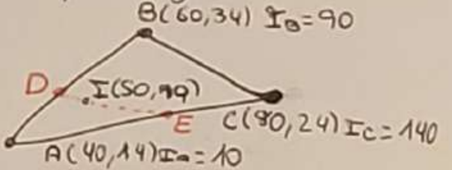
$$C = A + E + \underbrace{\sum \left(D(\vec{l}) + S(\vec{l}, \vec{v}) \right)}_{\text{for each light source}}$$

Final color: Ambient + Emissive + Diffuse + Speculer
Difuse Reflection: Reflect light in all directions. $c \propto n \cdot l$
Speculat Reflection: Depends on viewer position. $c \propto (r \cdot e)^h$

Shading

Shading normal: Geometrical normals and shading normals
Surface shading: Flat shading and Smooth shading (simulate curved surfaces).
Flat shading: One normal vector per polygon.
Smooth shading: Gouraud Shading: normal vector per polygon and per vertex.
Phong Shading: Gouraud Shading + multiple normal vectors per edge.

Exercise



$D = A + \overrightarrow{AB} \cdot S = (40,14) + (20,20) \cdot \frac{19-14}{34-14} = (45, 19)$

$E = A + \overrightarrow{AC} \cdot S = (40,14) + (40,10) \cdot \frac{19-14}{24-14}$

$I_D = I_A + S \cdot (I_B - I_A) = 10 + \frac{1}{4} \cdot 80 = 30$

$I_E = I_A + S \cdot (I_C - I_A) = 10 + \frac{1}{2} \cdot 130 = 75$

$I_{(50,19)} = I_D + S \cdot (I_E - I_D) = 30 + \frac{50-45}{60-45} \cdot (75-30) = 45$

SESSION 15

Texturing

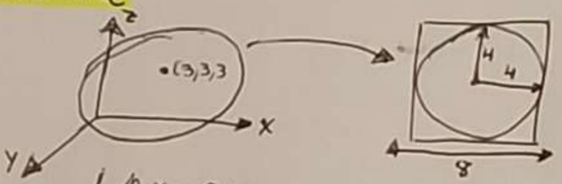
Textures

Textures allow to change the properties of the material along the surface.
Procedural Textures: Based on algorithmic or ecuation.
Noise Functions: Add realism, most used: Perlin noise
Space: 2D textures or 3D textures.
Method: Procedural or images.
Using Images: 2D img to 3D object, need mapping technig to apply.
Texel coordinate generation: Function or UV Mapping.
Texel: Texture element pixel.

Function-based Mapping

Use 3D object coordinates normalized
Flat Mapping: The object must be normalized in a unit cube centered in the origin.
Scale proportional without the z coordinate.
Cube Mapping: Scale without taking account the biggest cord.
Cylinder Mapping: x-y: angle as a vector, z: Height.
Sphere Mapping: x-y: angle as a vector. z: Cos Height.

Exercise



$r=4$ $d=8$ $C=(3,3,3)$
TextureSize $=(1024 \times 512)$
$p=(6'07, 4'27, 5'22)$

$\vec{n} \equiv p' = \left(\frac{px-cx}{d}, \frac{py-cy}{d}, \frac{pz-cz}{d}\right) = (0'3837, 0'16, 0'28)$

a) Cube Map

1ist Case: $(x>y)$ and $(x>z)$

$py = p'y + 0'5 = 0'66$ $\quad pz = p'z + 0'5 = 0'78$

Texel $= (1024 \cdot py, 512 \cdot pz) = (1024 \cdot 0'66, 512 \cdot 0'78) = (676, 399)$

b) $p1 = \frac{0'5 \cdot \text{Arctg}\left(\frac{px}{py}\right)}{\pi} + 0'5 = 0'69$ $\quad p2 = p.z + 0'5 = 0'78$

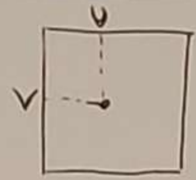
Texel $= (1024 \cdot py, 512 \cdot p2) = (707, 399)$

c) $p1 = \frac{0'5 \cdot \text{Arctg}\left(\frac{px}{py}\right)}{\pi} + 0'5 = 0'69$ $\quad p2 = \frac{\cos\left(\frac{pz}{d}\right)}{\pi} = 0'32$

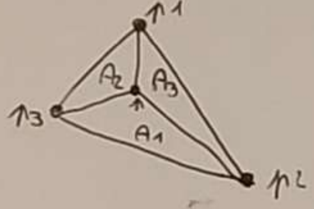
Texel $(1024 \cdot py, 512\, p2) = (707, 164)$

UV Maping

Use Two parametrics cordinates (u,v) per vertex



v: -1'56, 3'48, 2'3
vt: 0'0001, 0'6789
f: 1/1 2/3 4/5 6/7

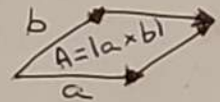


$$p = \lambda_1 \cdot p_1 + \lambda_2 \cdot p_2 + \lambda_3 \cdot p_3$$

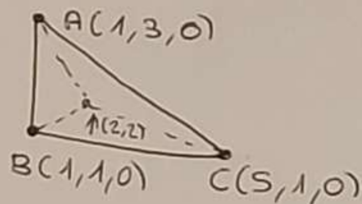
$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

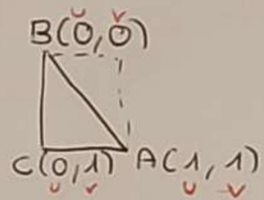
$$u(p) = \lambda_1 u(p_1) + \lambda u(p_2) + \lambda u(p_3)$$

$$v(p) = \lambda_1 v(p_1) + \lambda v(p_2) + \lambda v(p_3)$$



Exercise: UV coordinate





$$A = \frac{|\vec{BA} \times \vec{BC}|}{2} = \frac{|(0,2)\times(4,0)|}{2} = \frac{\cancel{2}\cdot 4-0}{\cancel{2}} = 4$$

$$A_A = \frac{|\vec{Bp} \times \vec{BC}|}{2} = \frac{|(1,1)\times(4,0)|}{2} = \frac{1\cdot 4-1\cdot 0}{2} = 2$$

$$\lambda_A = \frac{A_A}{A} = \frac{2}{4} = 0'5$$

$$A_B = \frac{|\vec{AC} \cdot \vec{Ap}|}{2} = \frac{|(4,-2)\times(1,-1)|}{2} = \frac{4\cdot 1-(-2)\cdot(-1)}{2} = 1$$

$$\lambda_B = \frac{A_B}{A} = \frac{1}{4} = 0'25 \qquad \lambda_C = 1-\lambda_A-\lambda_B = 1-0'5-0'25 = 0'25$$

$$u(p) = \lambda_A \cdot u(A) + \lambda_B \cdot u(B) + \lambda_C \cdot u(C) = 0'5\cdot 1 + 0'25\cdot 0 + 0'25\cdot 0 = 0'5$$

$$V(p) = \lambda_A \cdot v(A) + \lambda_B \cdot v(B) + \lambda_C \cdot v(C) = 0'5\cdot 1 + 0'25\cdot 0 + 0'25\cdot 1 = 0'75$$

Normal Mapping

Technique that used special types of images that store the normal value for each texel instead of the color

Types: Object space / Tangent space

Calculation: Range = {0 : 255} normal {-1 : 1} $128 + 128 \cdot x = v$

Morado oscuro = (128, 39, 219) = v

$$x = \frac{128-128}{128} = 0$$

$$y = \frac{39-128}{128} = -0'69 \qquad \vec{n} = (0, -0'69, 0'71)$$

$$z = \frac{219-128}{128} = 0'71$$

SESSION 16

Animation

Animation:

Presentation of images on the computer screen repeatedly replaced by a new image produces illusion of movement. The more the frames the better.
Wagon-wheel effect: Temporal Aliasing. Due to a limited frame rate.

Animation Methods

Low level: Scripts, Keyframing and Splines.
Medium level: Kinematics, physics and procedural.
High level: Motion capture, morphing and automatic.
Keyframe Animation: using animation curves, then define key propierties.
Spline Interpolation: its gets better continuity.

SESSION 17

Kinematics

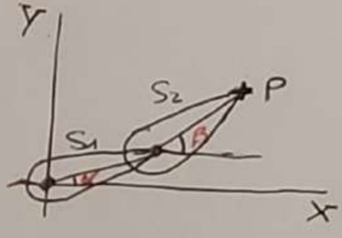
Hierarchical Structures

Scene graph: Graph without cycles that defines hierarchical relationships.

- Efficient Data Management
- Ease of Use
- High level interface
- Extensibility

Direct Kinematics: The final position is defined specifying the angles of the joints.
Inverse Kinematics: The angles are automatically calculated.
Kinematic chains: Armature   Bone: pose mode

Exercise 1



$S_1 = 3 \quad S_2 = 4$

$\alpha = 30° \quad \beta = 45°$

$$P_x = S_1 \cdot \cos(\alpha) + S_2 \cdot \cos(\beta)$$

$$P_x = 3 \cdot \cos(30) + 4 \cdot \cos(45) = 2'6 + 2'82 = 5'43$$

$$P_y = S_1 \cdot \text{sen}(\alpha) + S_2 \cdot \text{sen}(\beta)$$

$$P_y = 3 \cdot \text{sen}(30) + 4 \cdot \text{sen}(40) = 1'5 + 2'82 = 4'33$$

Inverse Kinematics

The method we are studying is CCD (iterative method for comp graph).

Cyclic Coordinate Descent:

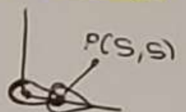
D - Desired position

E - Position of effector

R - Base of the bone That we are moving To get the rotation.

The idea is To change The position of the effector for every bone per iteration. The effector will be always at The end of the arm.

Exercise 2



$S_1 = 3$ $S_2 = 4$ $MinDis = 0'1$ $N^o IT = 500$

a) $\vec{AB} = (4,0)$ $\vec{AP} = (2,5)$

$\theta = \arccos\left(\frac{\vec{AB} \cdot \vec{AP}}{|\vec{AB}| \cdot |\vec{AP}|}\right) = \frac{4 \cdot 2 + 5 \cdot 0}{\sqrt{4^2+0^2} \cdot \sqrt{2^2+5^2}} = \frac{8}{4 \cdot 5'38} = 68'2^o$

$E_x = (S_1 \cdot \cos(\alpha) + S_2 \cdot \cos(68'2^o)) = 3 \cdot 1 + 4 \cdot 0'3714 = 4'48$

$E_y = S_1 \cdot sen(\alpha) + S_2 \cdot sen(68'2^o) = 3 \cdot 0 + 4 \cdot 0'93 = 3'7140$

$\vec{CE_1} = (4'48, 3'71)$ $\vec{CP} = (5,5)$

$\theta = \arccos\left(\frac{\vec{CE_1} \cdot \vec{CP}}{|\vec{CE_1}| \cdot |\vec{CP}|}\right) = \frac{5 \cdot 4'48 + 5 \cdot 3'71}{\sqrt{5^2+5^2} \cdot \sqrt{4'48^2+3'71^2}} = 5'3710^o$

$E_x = S_1 \cdot \cos(5'37) + S_2 \cdot \cos(68'2^o) = 4'18$

$E_y = S_1 \cdot sen(5'37^o) + S_2 \cdot sen(68'2^o) = 4'117$

Distance To $P = (5,5) - (4'18, 4'117) = (0'89, 0'89)$

$Module = \sqrt{0'89^2 + 0'89^2} = 1'25 > 0'1$

b) Distance from center To $P = \sqrt{4^2+6^2} = 7'21$; $7'21 - 7 = 0'21$

The min distance can get is 0'21 so if The Min Dis is 0'1 The exercise canot be solve.

Physics Simulation SESSION 18

1.- Physical Process → 2.- Model → 3.- Simulation Algorithm

→ 4.- Program → 5.- Simulation

Common Collision Shapes = Sphere, Box, Cylinder, Convex Volume, Polygonal Mesh, Compound Shape.

Considerations = Predictibility, Test, Integration, Realism, Export, GUI.

Optimizations = Temporal consistency, Spatial Partitioning and Multi-level detection

CAMERA

SESSION 19

Getting a Camera Matrix

$$\underbrace{\overset{3\times 4}{P}}_{\text{Camera Matrix}} = \underbrace{\overset{3\times 3}{K}}_{\text{Intrinsic Matrix}} \times \underbrace{\overset{3\times 4}{[R|T]}}_{\text{Extrinsic Matrix}}$$

$T = -RC$

P is (3×4) matrix

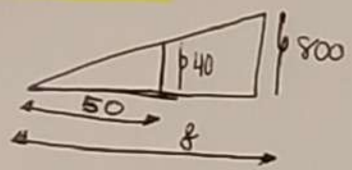
$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

f = Focal lenght → Distance between the pinhole and the film.

s = Axis Skew → In ideal cameras $s = 0$

x/y = Principal point Offset → center of the image normally

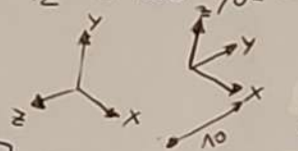
Exercise 1



$$\frac{800}{f} = \frac{40}{50} \quad f = \frac{800 \cdot 50}{40} = 1000$$

$$x_0 = \frac{800}{2} = 400 \quad y_0 = \frac{600}{2} = 300$$

$P = [K] \times [R|T]$



$x = x = (1,0,0)$

$y = z = (0,0,1)$

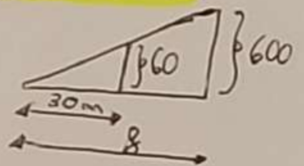
$z = -y = (0,-1,0)$

$$[K] = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1000 & 0 & 400 \\ 0 & 1000 & 300 \\ 0 & 0 & 1 \end{bmatrix} \quad [R|T] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 10 \end{array}\right]$$

$$P = \begin{bmatrix} 1000 & 0 & 400 \\ 0 & 1000 & 300 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 10 \end{array}\right] = \begin{bmatrix} 1000 & 400 & 0 & 4000 \\ 0 & 300 & -1000 & 3000 \\ 0 & 1 & 0 & 10 \end{bmatrix}$$

$$P \cdot v = P \cdot \begin{bmatrix} -3 \\ 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1000 \\ 1000 \\ 10 \end{bmatrix} \rightarrow (1000/10,\ 1000/10) = (100, 100)$$

Exercise 2



$$\frac{600}{f} = \frac{60}{30} \quad f = \frac{600 \cdot 3\cancel{0}}{6\cancel{0}} = 300$$

$$x_0 = \frac{600}{2} = 300 \quad y_0 = \frac{400}{2} = 200$$

$$K = \begin{bmatrix} 300 & 0 & 300 \\ 0 & 300 & 200 \\ 0 & 0 & 1 \end{bmatrix} \quad [R|T] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 \end{array}\right]$$

$$P = K \cdot [R|T] = \begin{bmatrix} 300 & 0 & 300 & 1500 \\ 0 & 300 & 200 & 1000 \\ 0 & 0 & 1 & 5 \end{bmatrix}$$

$$P \cdot v = P \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{pmatrix} 1800 \\ 1300 \\ 5 \end{pmatrix} \rightarrow (1800/5,\ 1300/5) = (360, 260)$$

double the resolution) $f = 2f = 600$

$P' = P \cdot 2 = (360, 260) \cdot 2 = (720, 520)$

RayTracing **SESSION 20**

Rendering review

GPU (Hardware Pipeline): The pipeline raster stored the distance of objects from the view plane

Ray Casting: We trace rays from the camera and calculate if it bounces with an object.

Raycasting

The algorithm starts in the camera traces a ray per pixel in the image

Every pixel traces a shadow ray to determinate how the light source interacts with the object.

When hitting an object, it bounces.

Reflected: due to mirror properties of the object's surface.

Refracted: Same as Reflected, but if surface has transparency.

Ray Casting Algorithm

Ray rep: $p = o + t \cdot d$

$o \rightarrow$ Origin point $\quad d \rightarrow$ Direction $\quad t \rightarrow$ Distance from origin

Ray object intersection

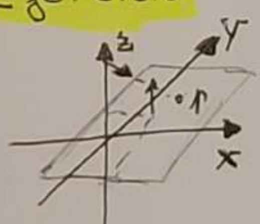
light ray: $r(t) =$ Ray Rep formula.

plane: $(x - p0) \cdot n = 0$ if dot product $== 0$ x belongs to the plane.

Ray equation on a Plane

$$t = \frac{-\vec{n} \cdot (o - p)}{\vec{n} \cdot \vec{d}}$$

Ejercicio



$o = (0, 0, 5) \quad p = (1, 1, 0)$

$d = (1, 1, -1) \quad \vec{n} = (0, 0, 1)$

$x > 0 \quad y > 0 \quad z = 0$

$$t = \frac{-\vec{n} \cdot (o - p)}{\vec{n} \cdot \vec{d}} = \frac{-(0,0,1) \cdot (-1,-1,5)}{(0,0,1) \cdot (1,1,-1)} = \frac{+5}{+1} = 5$$

$$r(5) = o + 5 \cdot d = (0,0,5) + 5 \cdot (1,1,-1) = (5,5,0)$$