Computer Graphic and Rasterization

李丞弘

Outline

- Introduction
- Application
- Homogeneous coordinates and Transform
- Texture and BRDF
- Rasterization

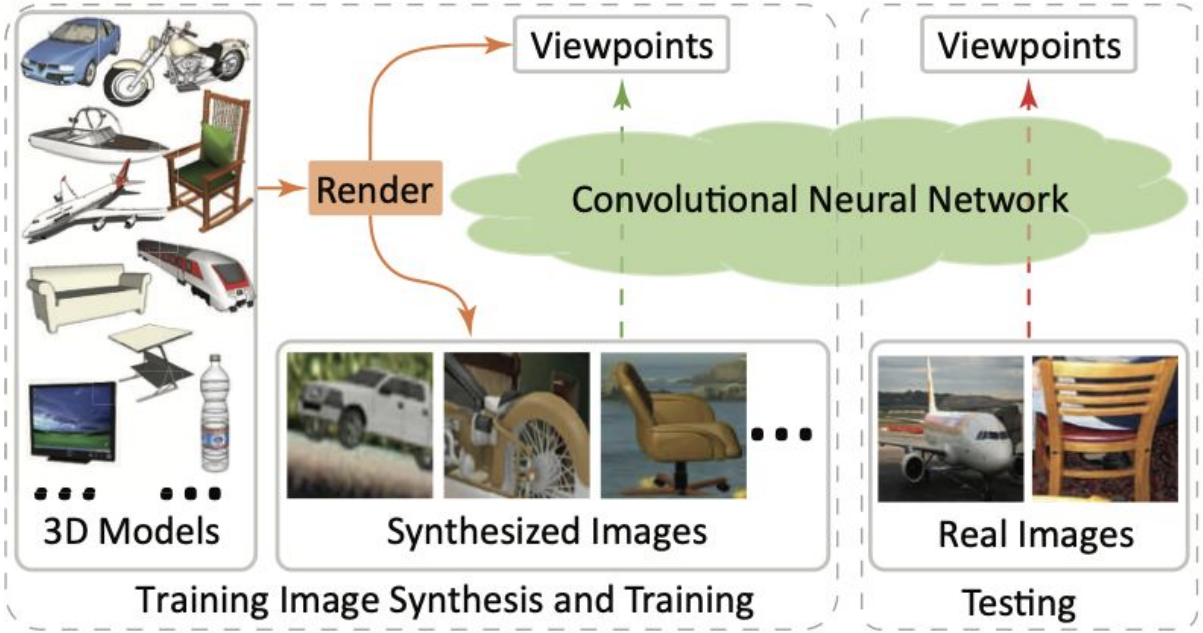
Introduction

Given a scene description, generate the 3d scene.

- Camera
  - Position & Direction
  - Lens
    - Focal length
    - Aperture
    - Depth of field
  - Motion blur
- Object
  - Geometry
  - Material
- Light Source

Application

- Computer Vision



Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views [Su et al. ICCV 2015]

Application

- Computer Vision
- Virtual Reality

Application

- Computer Vision
- Virtual Reality
- Animation & Game

Homogeneous coordinates and Transform

In 3d space, we can do linear transform like rotation and sheer with 3x3 matrix.

i.e. (x, y, z) >> (a1*x + b1*y + c1*z, a2*x + b2*y + c2*z, a3*x + b3*y + c3*z)

Displacement : (x, y, z) >> (x + x0, y + y0, z + z0)

Project to z=1: (x, y, z) >> (x / z, y / z, 1)

Homogeneous coordinates and Transform

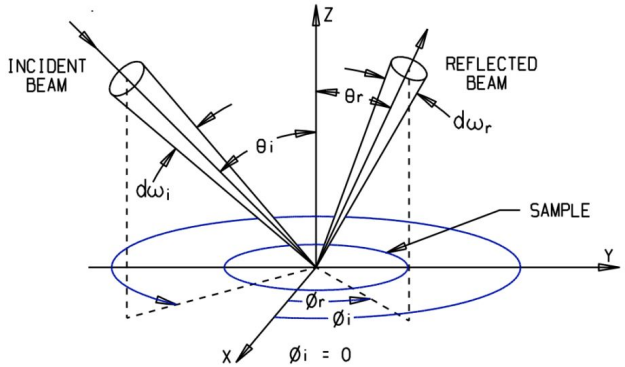
Add 4th dimension: (x, y, z, 1) (last element normalize to 1 after transform)

Displacement : (x, y, z, 1) >> (x + x0, y + y0, z + z0, 1)

Project to z = 1 : (x, y, z, 1) >> (x, y, z, z) >> (x / z, y / z, 1, 1)

Texture and BRDF

- Specular reflection
  - angle of incident = angle of reflection
  - Engery focus on only one direction
  - Cannot describe diffuse reflecation
- BRDF
  - Input: input and output direction in hemi-sphere (4d)
  - Output: Energy passed from input light to output light
  - Not always circular symmetric
    - e.g. Nylon



Rasterization

Just project objects to screen of camera

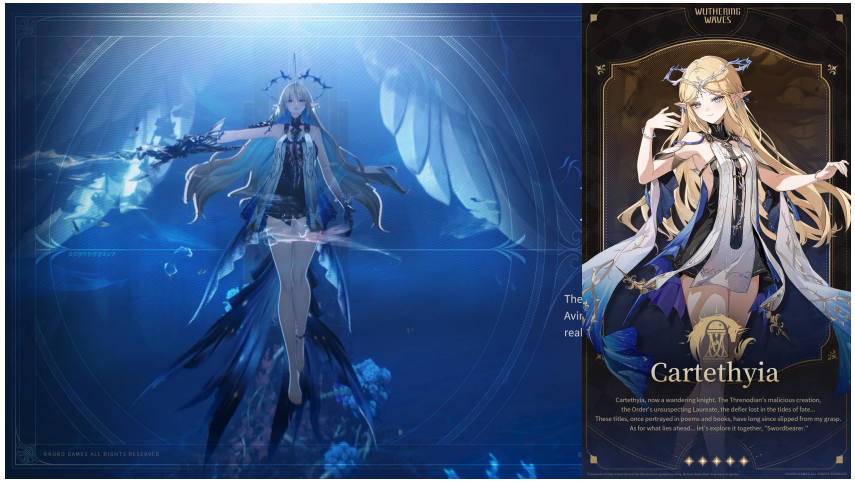
Simple & efficient >> Popular in the times without GPU.

- Painting algorithm
- z-buffer

Painting algorithm

Render scenes like stacking layers in photoshop.

Example : 2 objects



Painting algorithm

Render scenes like stacking layers in photoshop.

Example : 2 objects

Middle first >> backward

Right second >> Forward

Painting algorithm

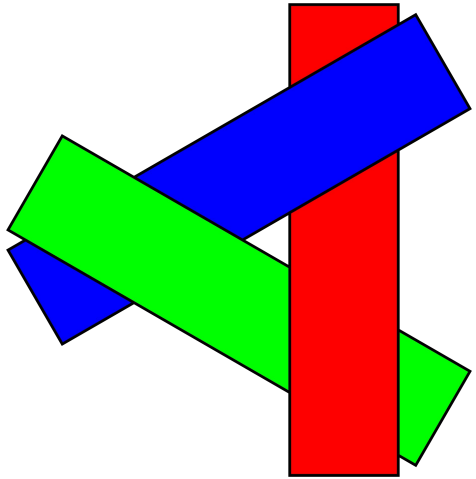
Sort objects by z-axis

for obj in objects:

    Project obj onto screen

Painting algorithm

Painting algorithm cannot handle cyclical overlapping and piercing polygons.
We should cut the polygons.

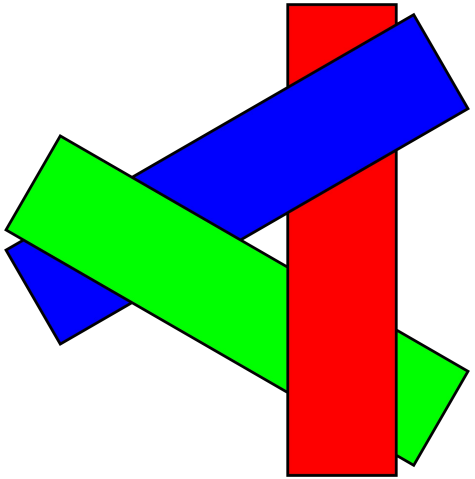


Painting algorithm cannot handle cyclical overlapping and piercing polygons.
We should cut the polygons.

Z-buffer

Cut every objects into pixels. Use a buffer to record current depth of each pixel.

```
for obj in objects:
    for pixel in screen:
        if pixel & obj overlap and depth < buffer[pixel]:
            update pixel and z-buffer
```



Demo

Reference

https://arxiv.org/abs/1505.05641%20

https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/438100/header.jpg?t=1762366454

https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcR0k2aPJk8dnfjT8dtr9HPqfNrSsXgHkVgUIz_sFmMyz5Q54xLQ

https://www.threads.com/@grey_official28/post/C60teBNrR-i

https://truth.bahamut.com.tw/s01/202510/forum/74860/8d9ecf57068350eb973a08e08729eac8.JPG

https://surfaceoptics.com/bsdf-brdf-and-btdf-a-review-of-measurement-approaches/%20

https://www.youtube.com/watch?v=t7ZVCOjXBd8

https://www.facebook.com/WutheringWaves.Official/posts/post-lament-anthropocene-enemies-fleurdelysthe-otherworldly-being-that-slumbers-/665851926001037/

https://wutheringwaves.gg/profile-reveal-feathered-tempest-cartethyia/

https://en.wikipedia.org/wiki/Painter%27s_algorithm

https://www.youtube.com/watch?v=a_9Tsbduk9E