MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

METAVERSE
CENTER

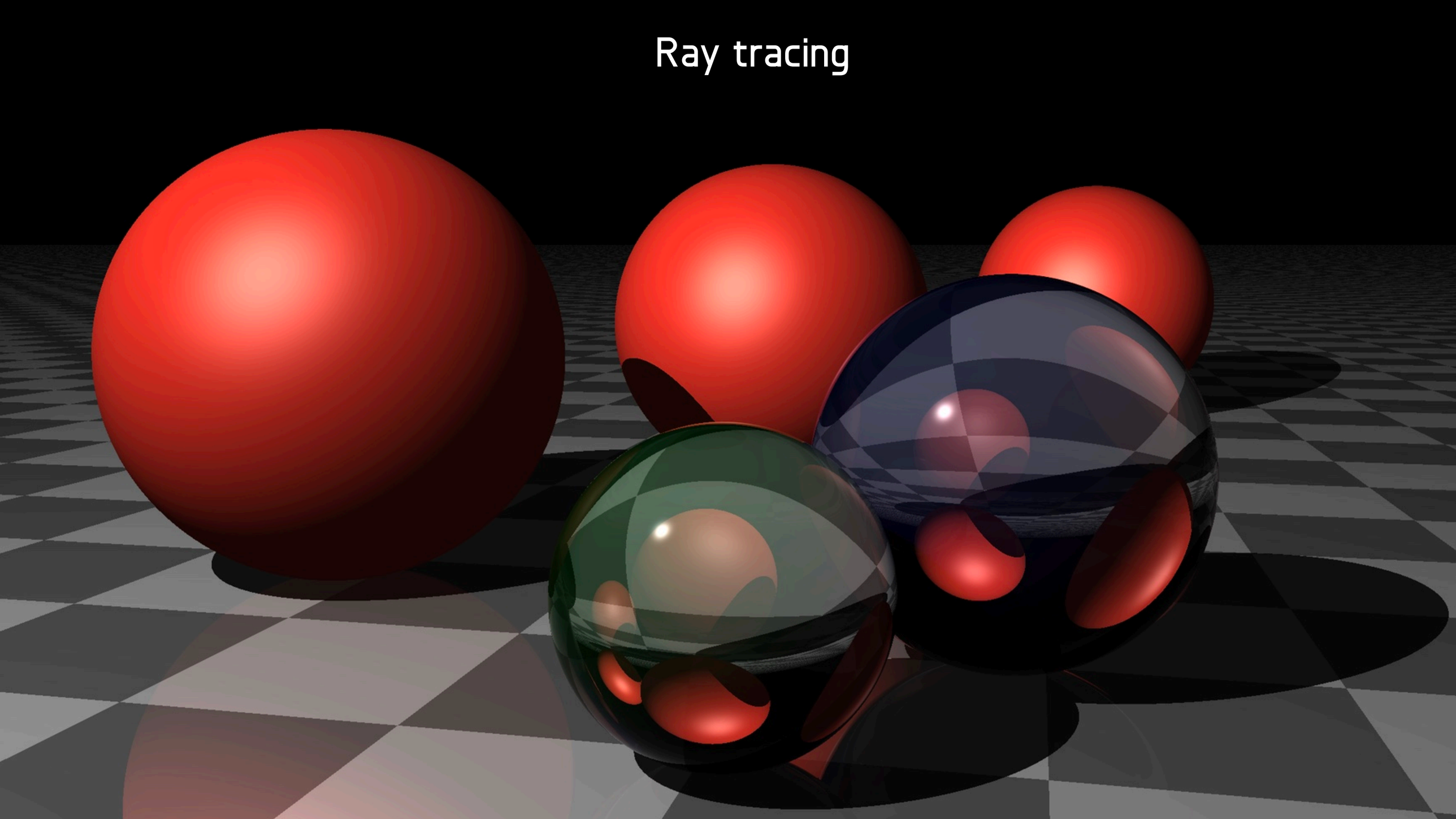
CV 804: 3D RENDERING & GEOMETRY PROCESSING

WEEK 8 / EXERCISE 4: Ray Tracing

HAO LI

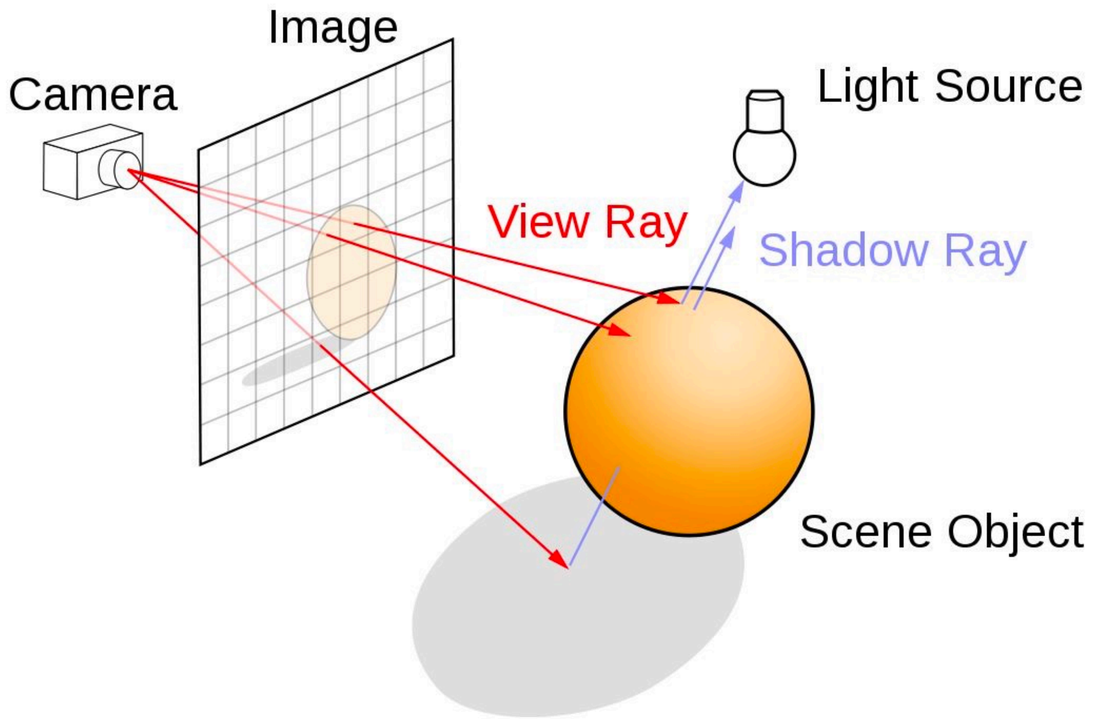
01/31/24

Ray tracing

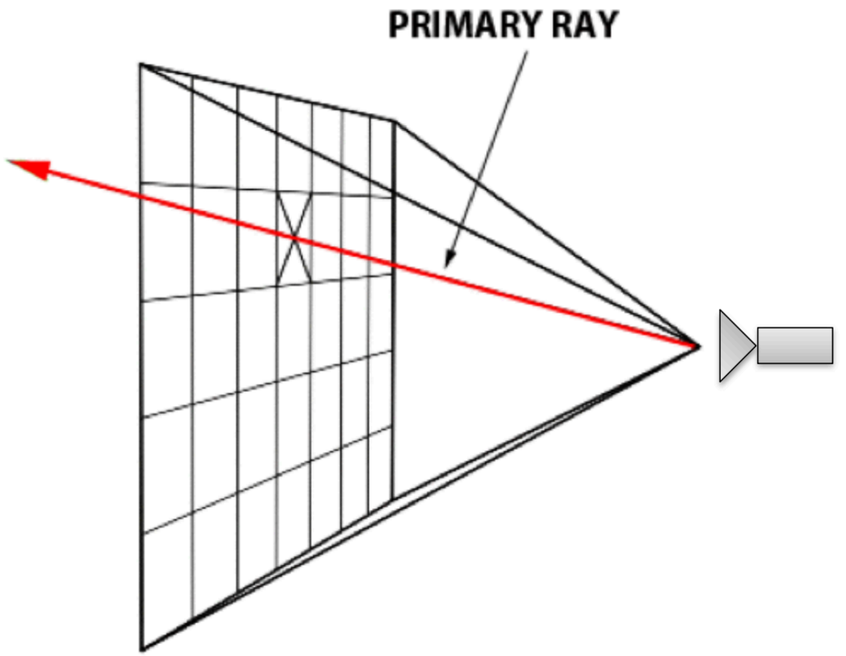


Exercise 4

- Level 1: sent out rays
- Level 2: intersection
- Level 3: illumination



Level 1: sent out rays



Level 1: sent out rays

Camera position: (0, 0, 0)

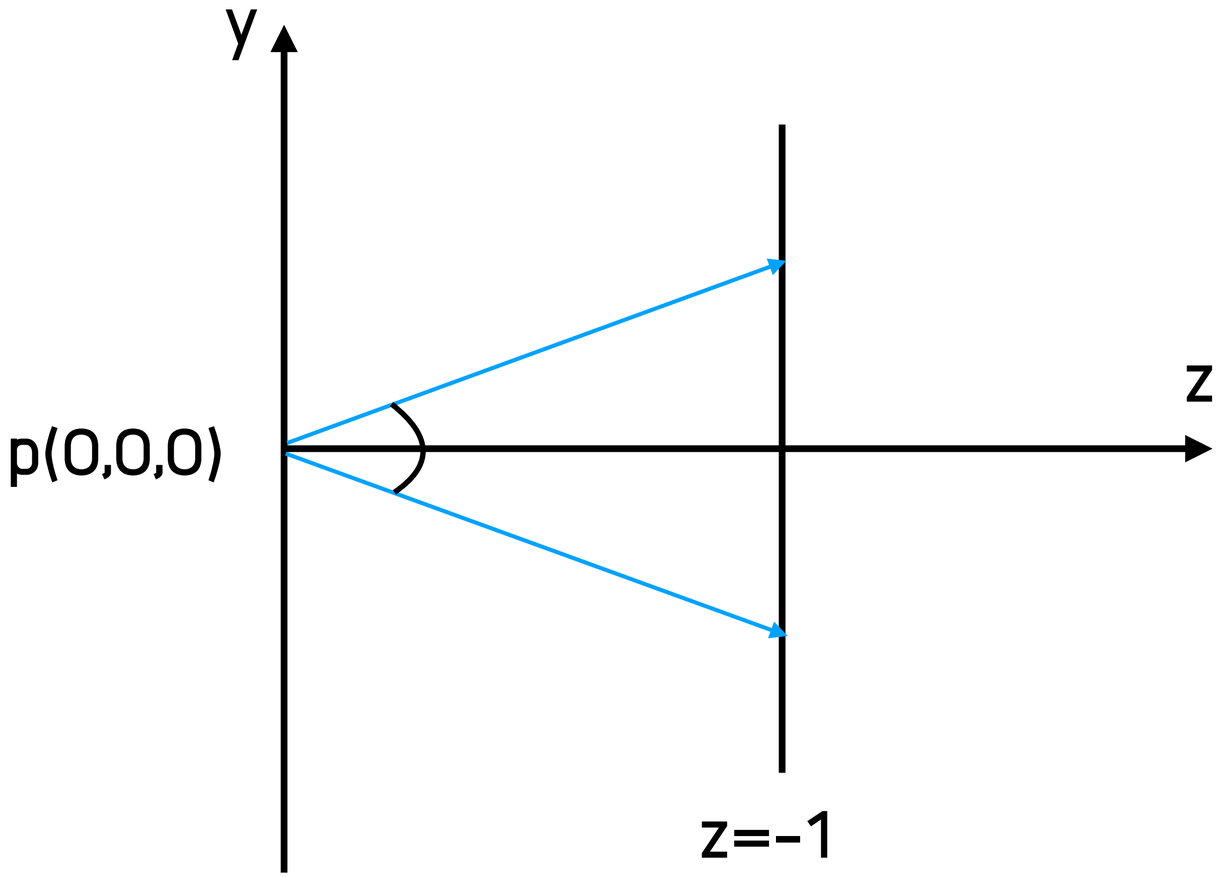
Look at: (0, 0, -1)

Up vector: (0, 1, 0)

Near plane: z=-1

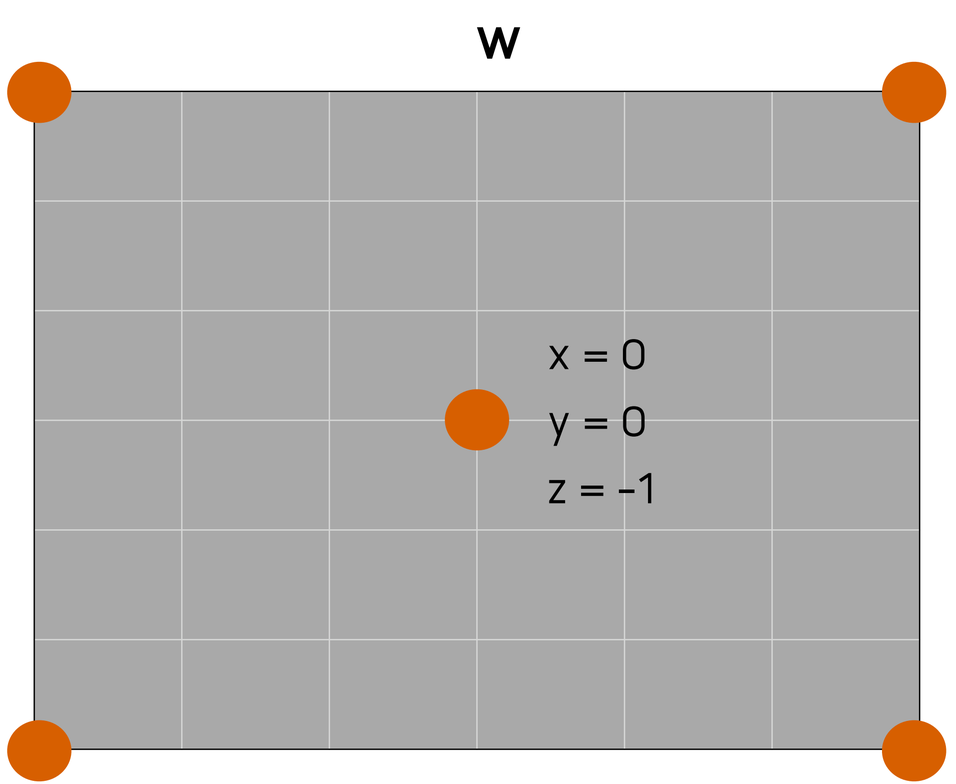
FOV: 60 degree

a = aspect ratio = w/h



Side view

Level 1: sent out rays



FOV: 60 degree

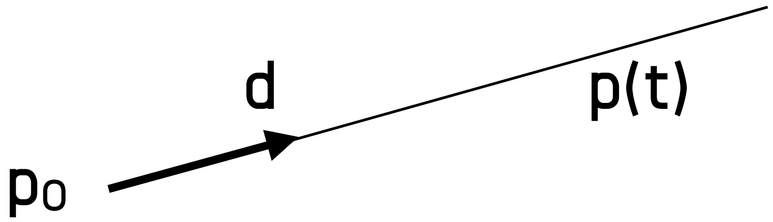
a = aspect ratio = w/h

Frontal view

Level 1: sent out rays

Ray in parametric form:

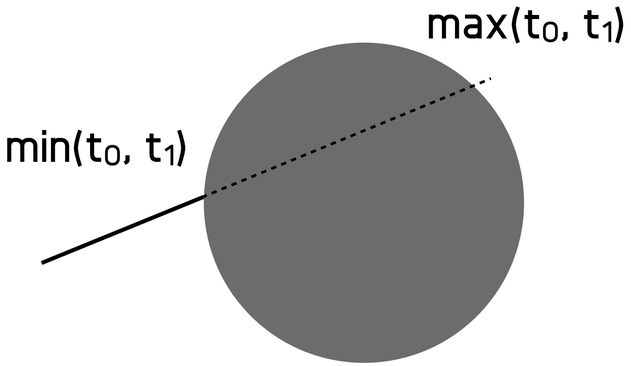
- Origin: $p_0 = [x_0 \quad y_0 \quad z_0]^T$
- Direction: $d = [x_d \quad y_d \quad z_d]^T$
- Assume d is normalized: $x_d * x_d + y_d * y_d + z_d * z_d = 1$
- Ray: $p(t) = p_0 + dt \quad \text{for } t > 0$



Level 2: Ray-Sphere Intersection

Define sphere by:

- Center: $c = [x_c \quad y_c \quad z_c]^T$
- Radius: r
- Implicit surface: $f(q) = (x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2 - r^2 = 0$



Plug in ray equations for x, y, z:

$$x = x_0 + x_d t \qquad y = y_0 + y_d t \qquad z = z_0 + z_d t$$

Obtain a scalar equation for t:

$$(x_0 + x_d t - x_c)^2 + (y_0 + y_d t - y_c)^2 + (z_0 + z_d t - z_c)^2 - r^2 = 0$$

Solve the equation, if the equation has solution, check $t_0, t_1 > 0$. Return $\min(t_0, t_1)$

Level 2: Ray-Triangle Intersection

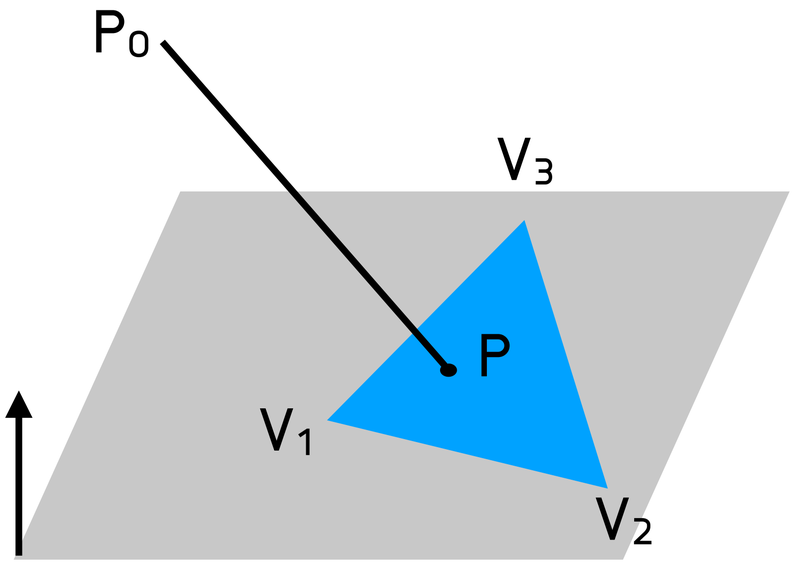
Method 1:

- Find intersection of the ray and the plane which the triangle lies on.
- Determine the ray-plane intersection point is in/out of the triangle

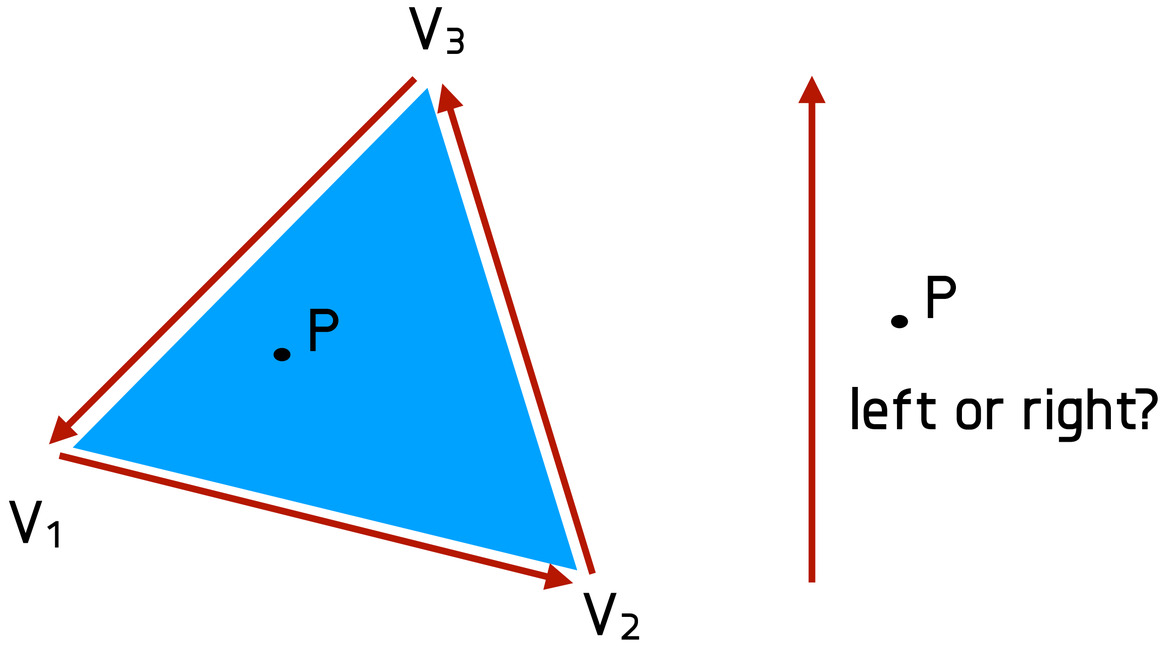
Method 2:

- Fast, Minimum Storage Ray/Triangle Intersection [Moller et al. 1997].

Level 2: Ray-Triangle Intersection

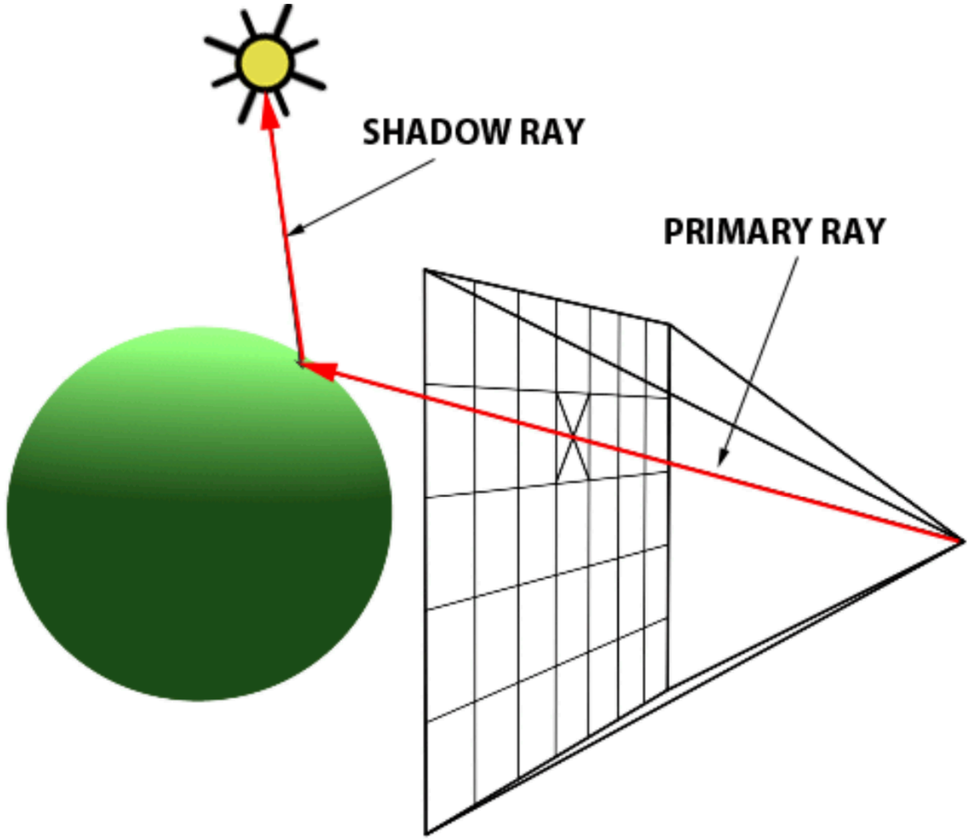


Find intersection of the ray and the plane which the triangle lies on.

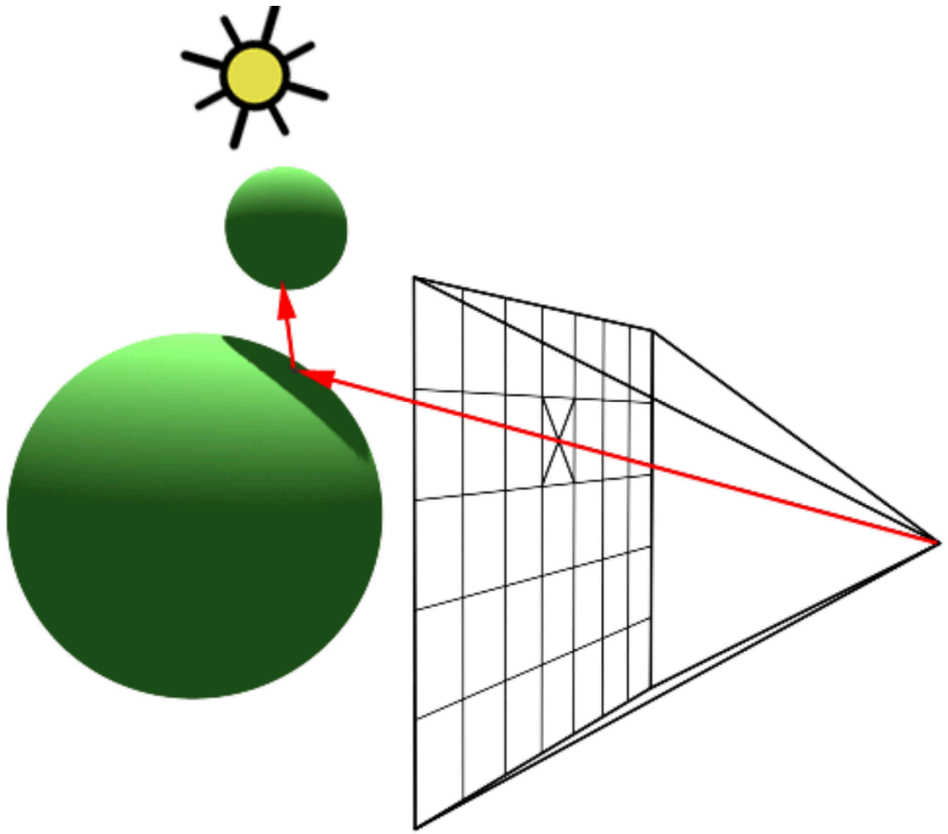


Determine the ray-plane intersection point is in/out of the triangle

Level 3: Illumination – shadow rays

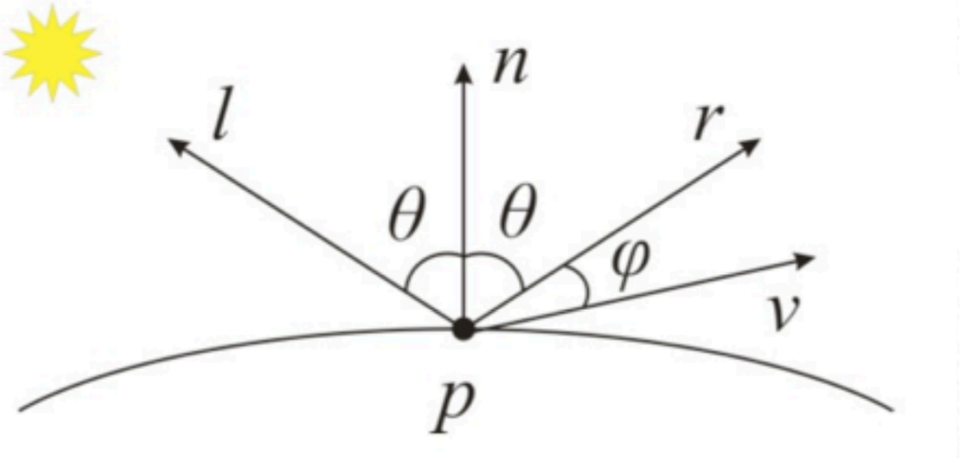


Level 3: Illumination – shadow rays



Level 3: Illumination – Phong shading

$$I = L(k_d(l \cdot n) + k_s(r \cdot n)^{\alpha})$$



Notice

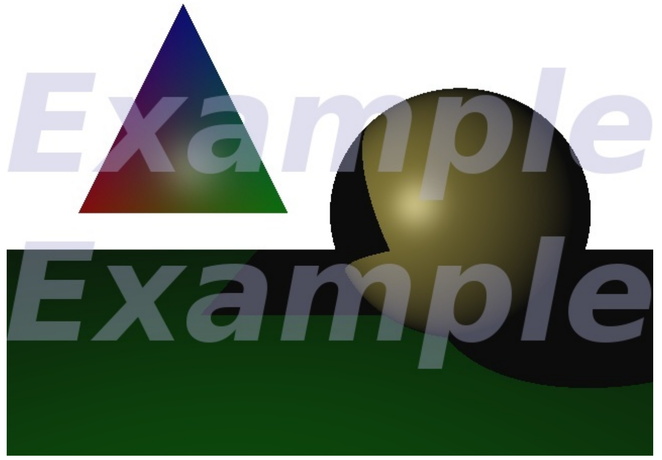
- Ensure B!=0 when doing A/B
- Before call sqrt(A), make sure A>=0
- Remember to normalize direction vector
- Remember to check len(dir)!=0 before normalize dir
- Floating-point operations are not accurate

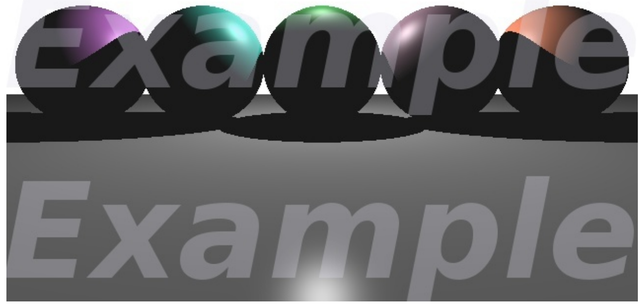
```
if (a>0.0f)
#define EPS 10e-8f
if (a>EPS)
```

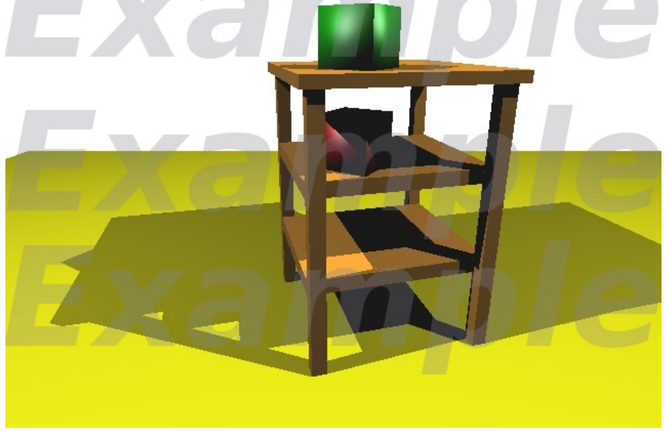
Extra credit

- Recursive reflection
- Recursive refraction
- Antialiasing
- Soft shadows
- Animation
- Motion blur
- Using Spatial structure to accelerate
- Parallel computing to accelerate
- . . .

Demo







Submission

- Deadline: **Tuesday 12th March, 11:59 pm**
- Follow submission instructions on Ex.4 webpage:
  - Upload a .zip compressed file named “Exercise4-YourName.zip” to Moodle
  - Include your code with comments
  - Include a readme file:
    - Describe how you solved each level, using same level numbers.
    - Describe problems you encountered
  - Include JPEG frames or a video

Contact

INSTRUCTOR
- Hao Li, hao.li@mbzuai.ac.ae
- Office: Building 1B, 1st floor (please schedule first)

TEACHING ASSISTANTS
- Phong Tran, the.tran@mbzuai.ac.ae
- Long-Nhat Ho, long.ho@mbzuai.ac.ae

OFFICE HOURS
* Office Hours: TBD, will be posted soon
* Emails (include “CV804” in title)

QUESTIONS?

www.hao-li.com

THANKS!

MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

hao.li@mbzuai.ac.ae