CSE 287 Guide Study for Second Midterm Fall 2019.doc

**CSE 287: Practical Learning Objectives**

**Chapter Four (Ray Tracing)**

1. State whether ray tracing is an image-order or object-order algorithm.

Image-Order

2. Write pseudo code that describes the basic ray tracing algorithm.

Compute viewing ray

Find first object hit by ray and the surface normal at the point of intersection

Set pixel color based on

* Surface normal (orientation of the intersection point)
* Relative position of the light
* Color of the object
* Color of the light(s)

3. List or identify the two basic types of projection.

Orthographic and Perspective

4. Identify the principle characteristics of an orthogonal projection.

* All traced rays are parallel to one another
* Parallel lines on objects remain parallel
* Relative dimensions are preserved
* Angles are preserved
* Object size is constant as object distance from viewpoint increases/decreases

5. Identify the principle characteristics of a perspective projection.

* All rays converge on a center-of-projection
* Parallel lines converge to a vanishing point
* Relative dimensions are not preserved
* Angles are not preserved
* Object size increases as we get closer, and decreases as we get further away
* More natural mimicry of real life

6. Write a mathematical expression that defines aspect ratio.

* Width of Screen / Height of Screen

7. Given two rectangles and the dimensions of one or them, set the dimensions of the other to

make the aspect ratios equal.

1. Divide the width by the height of the known rectangle to get a known aspect ratio Q
2. Set the second triangle’s width/height to be multiplied by a variable x.
3. Solve Q = width/height \* x

8. Given the heights, widths, and points of origin for two rectangles, create a mapping from one

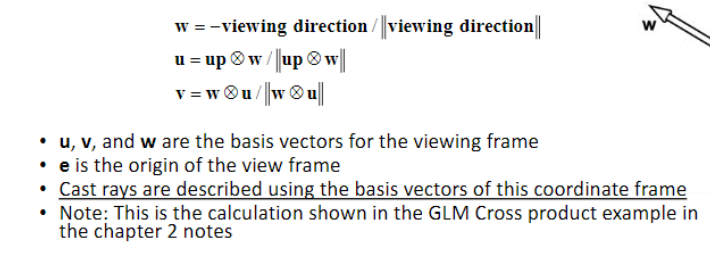
to the other.

If trying to map Rect 1 onto Rect2,

1. Set the origin point of Rect1 equal to the origin point of Rect2.
2. Top-Left point of Rect1 is now equal to the origin point of Rect2.
3. Top-Right point of Rect1 = (origin + Rect1.width, origin.y)
4. Bottom-Left point of Rect1 = (origin.x, origin.y - Rect1.height)
5. Bottom-Right point of Rect1 = origin.x + Rect1.width, origin.y - Rect1.height

9. Given two points, create a ray that will pass through the points going in a specified

directions.



10. Given and origin and a direction, create a corresponding parametric description of a ray.

p(t) = e + t(s - e)

e is the origin, s is a point on the ray

11. Given the parametric description of a ray and a value for the parameter, *t*, describe the

location of the point associated with the parameter value relative to the origin of the ray.

Plug in the desired t for the above equation .

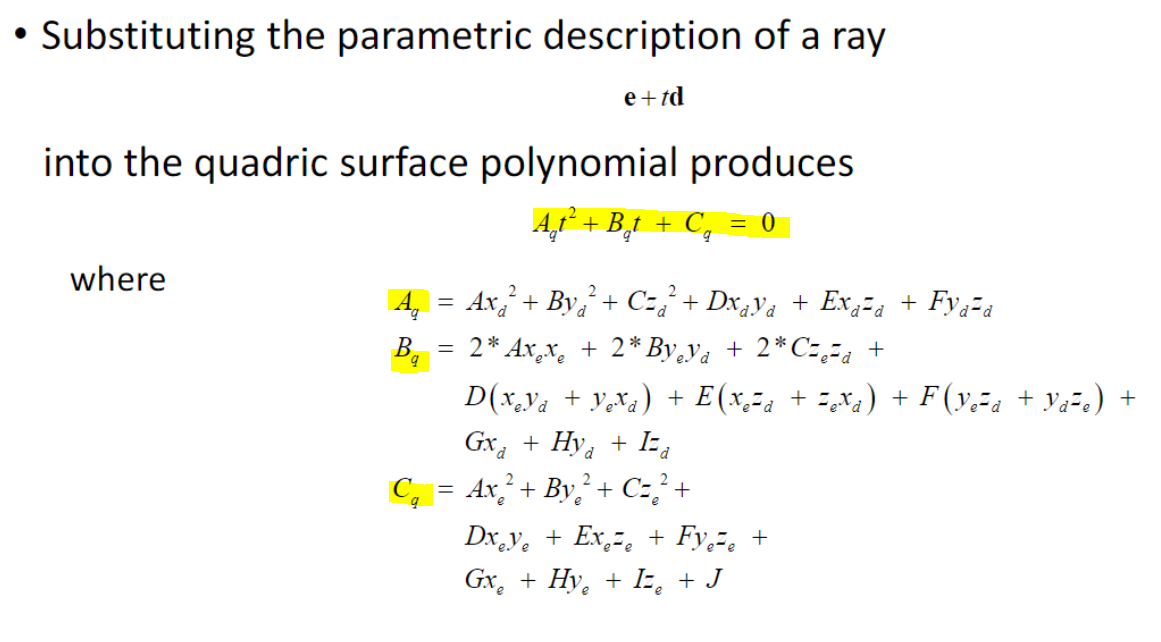
12. Given the parametric description of a ray and solutions for the parameter, *t*, that are

associated the intersections with a surface, identify which surface is closest to the origin of the ray.

Pick the smallest, positive solution

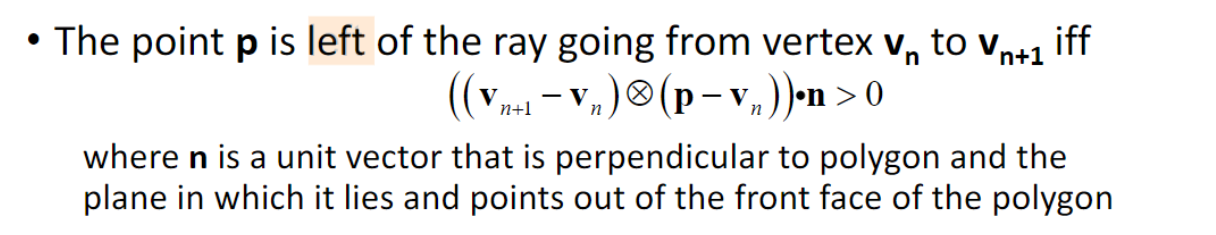
13. Describe the general procedure for determining the point of intersection between a

parametric ray and a surface that is described by a quadratic polynomial.



14. Given two points on a line and a third point that is in the same plane as the other points,

determine whether the third point is “left” or “right” of the line described by the two points.



**Chapter Four (Shading)**

15. Identify or give correct descriptions for ambient, diffuse, and specular reflection.

* Ambient: Bouncing light which is scattered so much it is impossible to tell the direction to its source. “Fills In” areas that do not receive direct illumination.
* Diffuse: Directional light which is brightest on surfaces that are perpendicular to the direction the light is shining. It’s reflection is scattered evenly.
* Specular: Directional light which tends to reflect in a preferred direction(associated with shininess. Dull objects do not give off Specular

16. Describe how emissive materials impact the results of local lighting calculations.

* Emissive Materials provide a large amount of their material’s color to be added into the final lighting calculation. A green sphere that emits green color will look very green.

17. Given the color and intensity of an ambient light source and the material properties of a

surface, correctly include the emissive material properties in the total calculation.

Emissive color of the surface needs to be added to the total amount of light for a given calculation

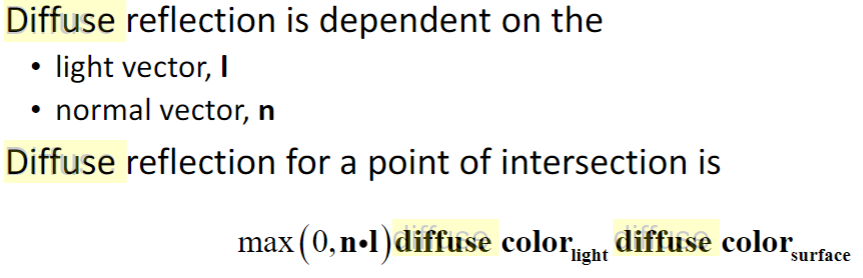
18. Given the color and intensity of an ambient light source and the material properties of a

surface, correctly calculate the ambient reflection for that surface.

AmbientReflection = AmbientColorLight \* AmbientColorSurface

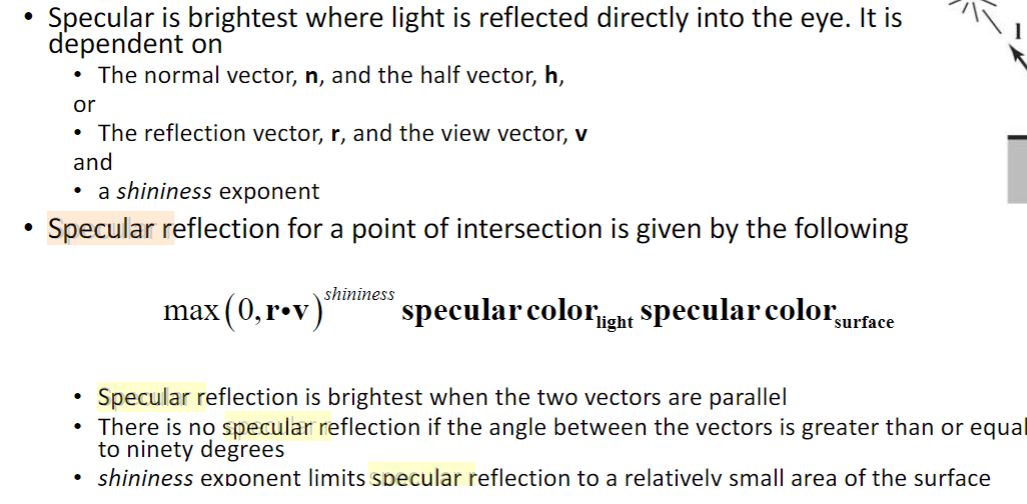
19. Given the color and intensity of a light source, the material properties of a surface, either the

direction to the light source or the position of the light source, and a surface normal, correctly calculate the diffuse reflection for that surface.



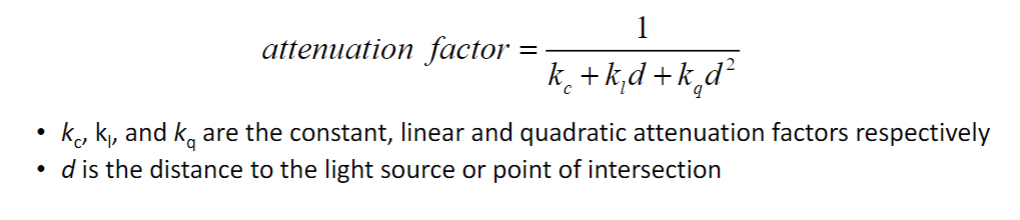
20. Given the color and intensity of specular light, the material properties of a surface, either the

direction to the light source or the position of the light source, either the direction to or the position of the view point, and a surface normal, correctly calculate the specular reflection for that surface.



21. Simulate light attenuation based on constant, linear, and quadratic attenuation factors and

the distance to a point of intersection or the distance to a positional light source.



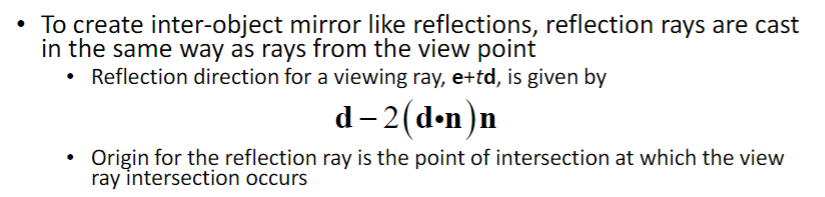
22. Correctly combine the diffuse and specular reflections for multiple light sources, to calculate

the total illumination for a pixel (fragment).



23. Given the direction of a ray that intersects a surface and the surface normal at the point of

intersection, calculate a reflection vector for the ray.



24. Describe the purpose of a shadow feeler. State under what conditions a light source will not

contribute to the illumination for a pixel (fragment) resulting in the rendering of a shadow.

A shadow feeler is used to determine whether or not an object casts a shadow, and if it does, how long that shadow’s length needs to be. If the length of a shadow feeler to a surface is less than the length from light source to surface, the pixel will render a shadow. If it is flipped. No shadow will be rendered.

25. Describe how reflection vectors are used to create mirror-like inter object reflections.

Say an initial ray hits a green ball that reflects and hits a red ball. A proper lighting

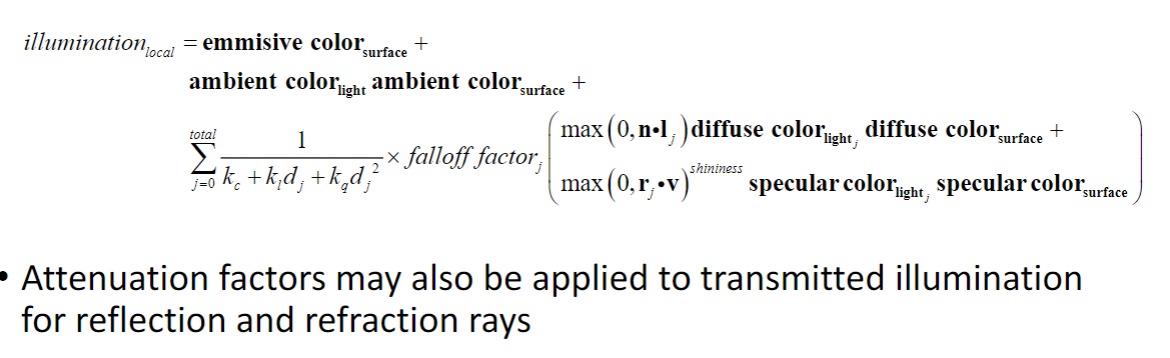
ray-tracer will make the reflection point on the green ball a little bit red, to indicate what the

Ray has hit along it's reflection path. Finding all these reflection hits is what generates a

Mirror like reflection for us

26. Correctly combine the results of tracing a reflection vector with the calculations associated

with direct illumination for a point of intersection or a fragment.



**Refraction and Texture Mapping**

27. State how transparent surfaces differ from opaque surfaces in how they interact with light.

Opaque surfaces do not transfer any light through. Transparent surfaces enable light to travel through and refract at a certain angle.

28. Name the effect that determines how much light is transmitted and how much light is

reflected by a transparent surface.

Fresnel effect

29. State what occurs when the angle of incidence with a transparent surface is exceeded.

No refraction

30. State the name of the law that determines the direction of refraction

Snell’s Law

. 31. State the name of the law that determines how light passing through a material is colored

and absorbed.

Beer’s Law

32. State the name of the discrete elements that a texture is composed of.

Texels

33. Given a picture of a texture image, identify the s and t coordinates of a specified position.

S-T coordinate systems start in the bottom left corner. S is the x-axis, T is the y-axis.

34. Given an image of a texture mapped object, state whether Planar, Cylindrical, or Spherical

Mapping was used to generate the texture coordinates for the object.

* Planar - Top of each object will look like it all the colors are well defined squares/rectangles
* Cylindrical - Top of each object will appear non-square, but with a consistent shape pattern (all triangles of different colors on top).
* Spherical- Top will look exactly as the other sides, shifted over

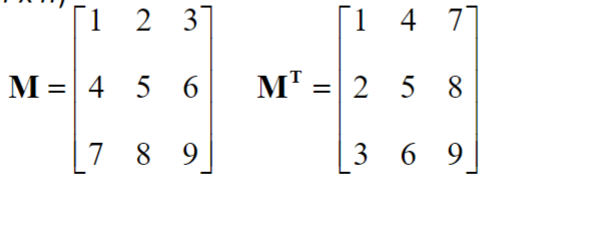
**Chapter Five (Matrices)**

35. Identify the main diagonal of a square matrix.

Top Left corner of the Matrix to the bottom right corner

36. Given a square matrix, create the *transpose* of the matrix.

Swap the rows and columns of the matrix



37. Given two matrices, state whether or not the matrices can be multiplied together based on

their dimensions. If they can be multiplied together, state what the dimensions of the product will be.

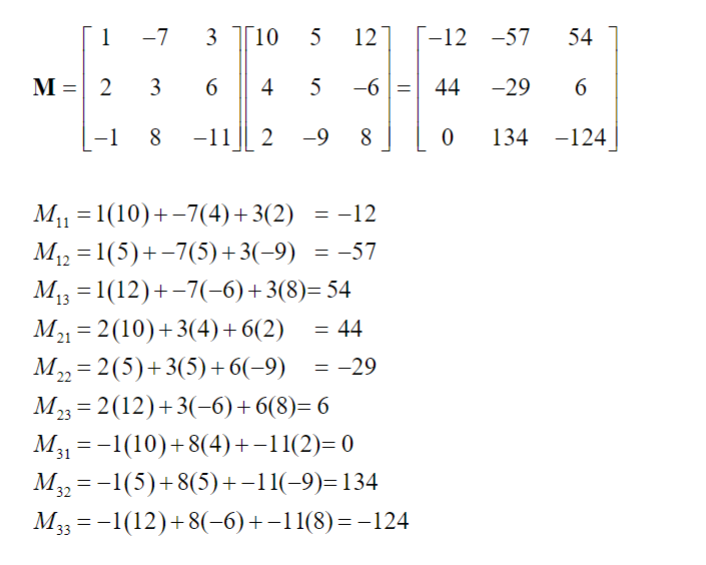
If we are performing operation A x B, the columns of A must match the Rows of B.

The resultant Matrix will have the rows = rows A, and columns = Columns of B.

I.E. [4 x 3] \* [3 x 5] is a valid multiplication and will produce a [4 x 5] matrix.

[3 x 3] \* [2 x 3] is an invalid multiplcation, 3 != 2

38. Given two matrices, calculate their product.



39. Given a matrix and a column vector, calculate their product.

Same process as above

40. State whether or not matrix multiplication is communitive.

Nope!

41. Show how the transpose of the product of two matrices relates to the product of their

transposes.



42. Given the dimension, construct an identity matrix with that dimension.

Slap 1s along the diagonal, fill the rest in with 0s

43. State what the product of the identity matrix and a square matrix with the same dimension will be equal to regardless of the order of the matrices.

IM = MI = M

44. Using matrix multiplication, show how a matrix is related to its *inverse*.

M \* M -1 = I

45. State what condition is necessary for a matrix to be invertible.

Determinant must be non-zero

46. Describe what value the *determinant* of a matrix will be equal to if a matrix is *singular*.

Zero

47. Show how the inverse of the product of two matrices relates to the product of their inverses.



**Chapter Six (Transformations)**

48. Given a description of a desired translation, write a statement that uses a GLM function to

construct the 4 x 4 homogenous transformation matrix that can produce the translation.

glm::translate(dvec3(x, y, z)) glm::rotate(-angle, dvec3(0, y, 0)

Clockwise is negative angle in radians, positive for counter-clockwise. Make 0 everything but the axis of rotation

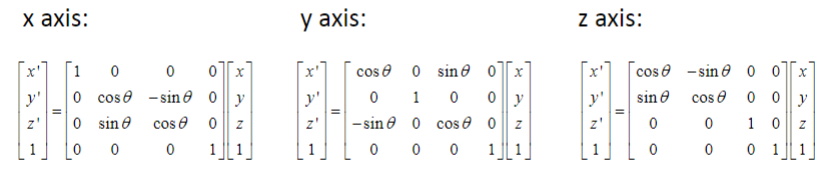
49. Use the right hand rule to correctly determine the positive and negative directions of

rotations in a right handed coordinate system.

I wish I could claim I fully understood this, but I still don’t get it completely.

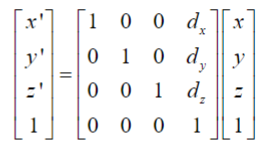
50. Given a description of a desired rotation about the x, y, or z axes, construct the 4 x 4

homogenous transformation matrix that can produce the rotation.

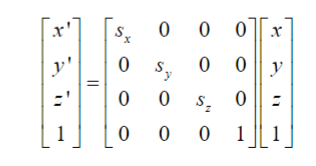


51. Given a description of an angle of rotation and the axis about which the rotation is to be

performed, write a statement that uses a GLM function to construct the 4 x 4 homogenous transformation matrix that can produce the translation.



52. Given a description of a desired uniform or non-uniform scale operation, construct the 4 x 4 homogenous transformation matrix that can produce the desired scaled coordinate system.



53. Given a description of a desired uniform or non-uniform scale operation, write a statement

that uses a GLM function to construct the 4 x 4 homogenous transformation matrix that can produce the translation.

glm::scale(dvec3(scaleX, scaleY, scaleZ)

54. Given a description of a position, orientation, and size, write a code fragment that will

generate the composite (modeling) transformation that will place the object in that desired position and orientation with the desired size.

Reading order is left to right. Multiplication is right to left.

ROTATEX -> TRANSLATION -> ROTATEY

is

ROTATEX \* TRANSLATION \* ROTATEY \* MATRIX