Chapter 1: A Survey of Computer Graphics - CAD,CAM,CAT

1. CAD (Computer-Aided Design)

Definition: Computer-Aided Design (CAD) is the use of computer technology to assist in the creation, modification, analysis, and optimization of designs for a wide range of applications. CAD software is used to create 2D and 3D models of physical objects, buildings, and machines.

Explanation: CAD is used extensively in industries such as architecture, engineering, manufacturing, and construction to create detailed designs of products and structures. CAD tools allow designers to create, modify, and test designs before they are physically built, thereby reducing the time and cost of development.

Example: Examples of CAD software include AutoCAD, SolidWorks, and SketchUp.

2. CAM (Computer-Aided Manufacturing)

Definition: Computer-Aided Manufacturing (CAM) is the use of computer technology to automate manufacturing processes. CAM software is used to create instructions for machines to produce parts and components, based on the designs created in CAD software.

Explanation: CAM software helps to automate the production process, allowing for greater precision, consistency, and speed. CAM software can be used in a range of industries, including aerospace, automotive, and medical device manufacturing.

Example: Examples of CAM software include Mastercam, GibbsCAM, and EdgeCAM.

3. CAT (Computer-Aided Tomography)

Definition: Computer-Aided Tomography (CAT) is a medical imaging technique that uses X-rays and computer technology to create detailed, cross-sectional images of the body.

Explanation: CAT scans are used to diagnose a range of medical conditions, including cancer, cardiovascular disease, and neurological disorders. CAT scans provide detailed images of the body's internal structures, allowing doctors to see inside the body without the need for surgery.

Example: A common example of a CAT scan is a CT scan of the head, which can be used to diagnose brain tumors, bleeding, or other abnormalities.

Chapter 2: Overview of Graphics systems -

CRT - Flat Panel Display - LCD,LED

Sure, here are the definitions, explanations, and examples of CRT, flat panel displays, LCD, and LED:

1. CRT (Cathode Ray Tube)

Definition: A Cathode Ray Tube (CRT) is a type of display device that uses an electron beam to produce images on a phosphorescent screen.

Explanation: CRTs were once the most common type of display technology used in computer monitors and televisions. The electron beam in a CRT is directed by magnetic fields to scan across the screen and illuminate the phosphorescent dots on the screen, creating an image.

Example: CRT displays were commonly used in older computer monitors and televisions, but have largely been replaced by flat panel displays.

2. Flat Panel Display

Definition: A flat panel display is a type of display device that is thin and flat, with a low profile and minimal depth.

Explanation: Flat panel displays are commonly used in modern computer monitors, laptops, and televisions. They are typically lighter and more energy-efficient than CRT displays, and have a wider viewing angle.

Example: Flat panel displays include LCD, LED, and OLED displays.

3. LCD (Liquid Crystal Display)

Definition: A Liquid Crystal Display (LCD) is a type of flat panel display that uses liquid crystals to control the amount of light that passes through the display.

Explanation: LCD displays are commonly used in computer monitors, televisions, and mobile devices. The liquid crystals in an LCD display are controlled by electrical signals to produce images on the screen.

Example: Examples of LCD displays include those used in laptops, tablets, and smartphones.

4. LED (Light Emitting Diode)

Definition: A Light Emitting Diode (LED) is a type of flat panel display that uses an array of light-emitting diodes to produce images on the screen.

Explanation: LED displays are commonly used in outdoor signs, billboards, and high-definition televisions. LED displays are known for their brightness, contrast, and energy efficiency.

Example: Examples of LED displays include those used in high-definition televisions, large outdoor signs, and sports scoreboards.

Chapter 3: Output Primitives -

Points and Lines - Line Drawing Algo(DDA, Bresenham) - Midpoint Circle Drawing Algorithm

Sure, here are the definitions, explanations, and examples of Points and Lines, Line Drawing Algorithms (DDA and Bresenham), and Midpoint Circle Drawing Algorithm:

1. Points and Lines

Definition: In computer graphics, points and lines are the simplest output primitives that can be used to create more complex images.

Explanation: Points are represented by a single pixel on the screen, while lines are created by connecting multiple points together. Points and lines can be used to create basic shapes and patterns, and are the foundation of more complex geometric forms.

Example: Points and lines are used to draw the basic shapes and outlines in most computer-generated images, from simple line drawings to complex 3D models.

2. Line Drawing Algorithms

a. DDA (Digital Differential Analyzer)

Definition: The Digital Differential Analyzer (DDA) algorithm is a line drawing algorithm that calculates the color and intensity values of pixels along a line segment between two points.

Explanation: The DDA algorithm uses the slope of the line to determine the color and intensity values of pixels along the line. It is a relatively simple algorithm that is easy to implement, but can be slow for large images or complex lines.

Example: The DDA algorithm can be used to draw straight lines in computer graphics.

b. Bresenham's Line Drawing Algorithm

Definition: Bresenham's line drawing algorithm is a more efficient algorithm for drawing lines than the DDA algorithm.

Explanation: Bresenham's algorithm uses integer calculations to determine the color and intensity values of pixels along a line, rather than floating-point calculations used by the DDA algorithm. This results in faster and more efficient line drawing, especially for lines with steep slopes.

Example: Bresenham's algorithm can be used to draw straight lines in computer graphics, and is often used in video game programming and other real-time graphics applications.

3. Midpoint Circle Drawing Algorithm

Definition: The Midpoint Circle Drawing Algorithm is a graphics algorithm used to draw circles on a computer screen.

Explanation: The Midpoint Circle Drawing Algorithm uses integer calculations to determine the positions of pixels along the circumference of a circle. The algorithm is relatively fast and efficient, and produces accurate results.

Example: The Midpoint Circle Drawing Algorithm can be used to draw circles and curves in computer graphics, such as in logo design or in 2D video games.

Sure, here are the formulas for the Line Drawing Algorithms and Midpoint Circle Drawing Algorithm:

1. DDA Algorithm:

The equation for calculating the color and intensity values of pixels along a line using the DDA algorithm is:

$0<m<1$, $x=x+1$, $y=y + m$

$-1<= m<=0$, $x=x-1$, $y=y - m$

$m>1$, $x= x+(1/m)$, $y=y+1$

$m<-1$, $x= x- (1/m)$, $y = y-1$

Where:

- `y` is the y-coordinate of the pixel
- `m` is the slope of the line
- `x` is the x-coordinate of the pixel

2. Bresenham's Line Drawing Algorithm:

The Bresenham's line drawing algorithm uses the following formula to determine which pixel to turn on or off:

Pk = 2Δy - Δx

If Pk<0, P(k+1) = Pk +2dy

X++

Else if Pk>0, P(k+1) = Pk+2dy-2dx

X++

Y++

Where:

- `Δx` is the difference between the x-coordinates of the two endpoints of the line
- `Δy` is the difference between the y-coordinates of the two endpoints of the line
- `x` and `y` are the current coordinates of the pixel being plotted
- `d` is the decision parameter used to determine which pixel to turn on or off

3. Midpoint Circle Drawing Algorithm:

The Midpoint Circle Drawing Algorithm uses the following formula to calculate the positions of pixels along the circumference of a circle:

```
```

```
x = 0

y = r

d = 1 - r


while (x < y) {

    plot(x, y)


    if (d < 0) {

        d = d + 2x + 3

    } else {

        d = d + 2(x - y) + 5

        y = y - 1

    }


    x = x + 1

}
```

Where:

- `r` is the radius of the circle

- `x` and `y` are the current coordinates of the pixel being plotted

- `d` is the decision parameter used to determine which pixel to turn on or off

- `plot(x, y)` is a function that turns on the pixel at position `(x, y)` on the screen.

Chapter 5: Two-Dimensional Geometric Transformations - Translation - Scaling - Rotation - Composite Transformations - General Pivot Point Rotation - General Fixed Point Scaling - Reflection – Shear

Sure, here are the definitions, explanations, and examples of the two-dimensional geometric transformations including Translation, Scaling, Rotation, Composite Transformations, General Pivot Point Rotation, General Fixed Point Scaling, Reflection, and Shear:

1. Translation

Definition: Translation is a geometric transformation that moves an object from one position to another without changing its orientation or shape.

Explanation: Translation is performed by adding a translation vector to the coordinates of each vertex of the object. The translation vector specifies the amount of movement in the x and y directions.

Example: Translation can be used to move objects on the screen, such as when dragging and dropping files or icons on a computer desktop.

2. Scaling

Definition: Scaling is a geometric transformation that changes the size of an object.

Explanation: Scaling is performed by multiplying the coordinates of each vertex of the object by a scaling factor. The scaling factor can be different for the x and y directions, which can result in the object being stretched or compressed.

Example: Scaling can be used to zoom in or out of an image or to change the size of objects in a 2D game.

3. Rotation

Definition: Rotation is a geometric transformation that rotates an object around a fixed point.

Explanation: Rotation is performed by multiplying the coordinates of each vertex of the object by a rotation matrix. The rotation matrix specifies the angle of rotation and the fixed point around which the object is rotated.

Example: Rotation can be used to create spinning or rotating animations in a 2D game or to adjust the orientation of objects in a drawing program.

4. Composite Transformations

Definition: Composite transformations are multiple transformations applied to an object in sequence.

Explanation: Composite transformations are performed by applying multiple transformations, such as translation, scaling, and rotation, in a specific order. The order of transformations can affect the final outcome of the transformation.

Example: Composite transformations can be used to create complex animations or to transform an object in a specific sequence.

5. General Pivot Point Rotation

Definition: General pivot point rotation is a geometric transformation that rotates an object around an arbitrary point.

Explanation: General pivot point rotation is performed by translating the object so that the arbitrary point is at the origin, rotating the object, and then translating it back to its original position.

Example: General pivot point rotation can be used to rotate an object around a specific point, such as a joint in a robot arm or a hinge in a door.

6. General Fixed Point Scaling

Definition: General fixed point scaling is a geometric transformation that scales an object around an arbitrary point.

Explanation: General fixed point scaling is performed by translating the object so that the arbitrary point is at the origin, scaling the object, and then translating it back to its original position.

Example: General fixed point scaling can be used to scale an object around a specific point, such as the center of a circle or the pivot point of a shape.

7. Reflection

Definition: Reflection is a geometric transformation that mirrors an object across a line or plane.

Explanation: Reflection is performed by multiplying the coordinates of each vertex of the object by a reflection matrix. The reflection matrix specifies the axis of reflection.

Example: Reflection can be used to create a mirror image of an object or to create a symmetrical pattern.

8. Shear

Definition: Shear is a geometric transformation that skews an object in a specific direction.

Explanation: Shear is performed by multiplying the coordinates of each vertex of the object by a shear matrix. The shear matrix specifies the amount of skew in the x and y directions.

Example: Shear can be used to create a slanted or distorted effect on an object or to create a 3D perspective effect in a 2D image.

Sure, here are the formulas for each of the two-dimensional geometric transformations:

1. Translation:

New $x = x + tx$

New $y = y + ty$

where tx and ty are the translation factors for the x and y directions, respectively.

2. Scaling:

New $x = sx * x$

New $y = sy * y$

where sx and sy are the scaling factors for the x and y directions, respectively.

3. Rotation:

New $x = x * \cos\theta - y * \sin\theta$

New $y = x * \sin\theta + y * \cos\theta$

where θ is the angle of rotation in radians.

4. Composite Transformations:

Multiple transformations can be combined using matrix multiplication:

New position $= T * S * R * P$

where T, S, and R are the matrices for the translation, scaling, and rotation transformations, respectively, and P is the position vector of the object.

5. General Pivot Point Rotation:

The steps for general pivot point rotation are as follows:

- Translate the object so that the pivot point is at the origin.
- Rotate the object around the origin.
- Translate the object back to its original position.

The transformation matrix for general pivot point rotation is:

T' * R * T

where T and T' are the matrices for the translation transformations to move the object to and from the origin, and R is the matrix for the rotation transformation.

6. General Fixed Point Scaling:

The steps for general fixed point scaling are similar to those for general pivot point rotation:

- Translate the object so that the fixed point is at the origin.
- Scale the object around the origin.

- Translate the object back to its original position.

The transformation matrix for general fixed point scaling is:

T' * S * T

where T and T' are the matrices for the translation transformations to move the object to and from the origin, and S is the matrix for the scaling transformation.

7. Reflection:

The transformation matrix for reflection across the x-axis is:

[1 0]

[0 -1]

and the transformation matrix for reflection across the y-axis is:

[-1 0]

[0 1]

To reflect across a line with a slope of m, the transformation matrix is:

[(1 - m^2)/(1 + m^2) 2m/(1 + m^2)]

[2m/(1 + m^2) (m^2 - 1)/(1 + m^2)]

8. Shear:

The transformation matrix for shear in the x direction is:

[1 shx]

[0 1]

where shx is the shear factor for the x direction.

The transformation matrix for shear in the y direction is:

[1 0]

[shy 1]

where shy is the shear factor for the y direction.

Chapter 6:Two-Dimensional Viewing

- Definition (Window, View Port, World Coordinate, Viewing Coordinate, Viewing Transformation, Clipping, Schilding)
- Window to Viewport Coordinate Transformation (Till equation 6.4)
- Point Clipping
- Line Clipping (Cohen Sutherland Algo)
- Polygon Clipping (Sutherland Hodgeman polygon Clipping)
- Weiler Atherton Polygon Clipping

Two-Dimensional Viewing

- Window: A world-coordinate area selected for display is called a window

- Viewport: A rectangular region of the device coordinate system where the objects are mapped for display.

- World Coordinate: A coordinate system that defines the location of objects in the scene.

- Viewing Coordinate: A coordinate system that defines the location of the observer and the orientation of the view.

- Viewing Transformation: The process of transforming objects from world coordinates to viewing coordinates.

- Clipping: The process of removing objects or portions of objects that are outside the view window or viewport.

- Schilding: The process of moving the objects to the viewport coordinates. It is done after the objects have been transformed into the viewing coordinate system and clipped to remove the parts that are outside the viewport.

Window to Viewport Coordinate Transformation:

The transformation from window to viewport coordinates can be defined by the following equation:

xv = xvmin + (xw - xwmin) * (xvmax - xvmin) / (xwmax - xwmin)

yv = yvmin + (yw - ywmin) * (yvmax - yvmin) / (ywmax - ywmin)

where (xw, yw) are the coordinates of a point in the window, and (xv, yv) are the coordinates of the corresponding point in the viewport. The variables xmin, ymin, xmax, and ymax represent the boundaries of the window and viewport.

Point Clipping:

The process of point clipping involves checking if a point is inside or outside the window. If the point is outside the window, it is clipped, and if it is inside, it is not. The point is represented by its (x, y) coordinates.

Line Clipping (Cohen-Sutherland Algorithm):

The Cohen-Sutherland algorithm is a line clipping algorithm that checks whether a line segment is inside, outside, or partially inside the window. The algorithm divides the window into nine regions and checks the endpoints of the line segment. If both endpoints are in the same region, the line segment is entirely inside or outside the window. If the endpoints are in different regions, the algorithm clips the line segment to the boundary of the window.

Polygon Clipping (Sutherland-Hodgeman Polygon Clipping):

The Sutherland-Hodgeman algorithm is a polygon clipping algorithm that clips a polygon against the window boundary. The algorithm works by successively clipping the polygon against each edge of the window. The clipped polygon is then used as input for the next clipping operation until all edges have been clipped.

Weiler-Atherton Polygon Clipping:

The Weiler-Atherton algorithm is a more complex polygon clipping algorithm that is capable of handling both concave and convex polygons. The algorithm works by creating a doubly linked list of polygon vertices and intersecting edges with the window boundary. The resulting clipped polygons are then stored in separate lists.

Chapter 10: Three-Dimensional Object Representation

Plane Equation, Polygon Meshes - - Sphere - Ellipsoid - Torus - Super Ellipse, Super Ellipsoid - Blooby Objects - Spline Representation (Interpolation and Approximation) - Parametric Continuity Conditions - Spline Specification (Upto Figure: 10-25) - Cubic Spline Interpolation method - Natural Cubic Spline - Hermite Interpolation (upto 10.34) -

Blending Function - Cardinal Spline (upto 10.38) - Bezier Curve (upto 10.42) - Properties of Bezier Curve - Fractal Geometry - Types of Fractals - Fractal Dimension - Deterministic Self Similar Fractals - Snow Flack Patterns - Random Midpoint displacement method - Self Squaring fractals, Self inverse fractal (short question) - Definition (Interpolation & Approximation Spline, Convex Hull, Control Graph, Control Polygon)

Three-Dimensional Object Representation

Plane Equation, Polygon Meshes:

A plane in three-dimensional space can be defined by an equation of the form Ax + By + Cz + D = 0, where A, B, and C are the coefficients of the plane's normal vector, and D is a constant.

Polygon meshes are collections of planar polygons that are used to represent three-dimensional objects.

Sphere:

A sphere is a three-dimensional geometric shape that is defined as the set of points in space that are equidistant from a given point called the center. The equation of a sphere centered at (a, b, c) with radius r is (x - a)^2 + (y - b)^2 + (z - c)^2 = r^2.

Ellipsoid:

An ellipsoid is a three-dimensional geometric shape that is defined as the set of points in space whose distances from the center along the three principal axes are proportional to a, b, and c. The equation of an ellipsoid centered at (a, b, c) with semi-axes a, b, and c is $(x - a)^2/a^2 + (y - b)^2/b^2 + (z - c)^2/c^2 = 1$.

Torus:

A torus is a three-dimensional geometric shape that is defined as a donut-shaped object formed by revolving a circle about an axis in the plane of the circle. The equation of a torus centered at (a, b, c) with radii R and r is $(x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) = 0$.

Super Ellipse, Super Ellipsoid:

A superellipse is a two-dimensional geometric shape that is defined by the equation $(|x/a|^n + |y/b|^n)^{(1/n)} = r$, where a, b, and r are constants, and n is a parameter that determines the shape of the curve. A superellipsoid is a three-dimensional geometric shape that is defined by the equation $(|x/a|^n + |y/b|^n + |z/c|^n)^{(1/n)} = r$.

Blooby Objects:

A blooby object is a three-dimensional geometric shape that is defined by a set of implicit surface equations. The surface of the object is defined as the zero-set of the equations.

Spline Representation:

Spline representation is a method of representing smooth curves and surfaces using a set of polynomial functions that are defined piecewise over smaller subintervals. The curves and surfaces are defined by a set of control points that are used to specify the shape of the spline.

Parametric Continuity Conditions:

Parametric continuity conditions are constraints that are imposed on the derivatives of the spline functions at the boundaries of the subintervals to ensure that the curve or surface is smooth and continuous.

Cubic Spline Interpolation Method:

Cubic spline interpolation is a method of fitting a curve to a set of data points by using a set of cubic polynomial functions that are defined piecewise over smaller subintervals. The spline functions are required to pass through each of the data points and to have continuous first and second derivatives.

https://youtu.be/f4iNbNRKZKU

Natural Cubic Spline:

A natural cubic spline is a type of cubic spline that is used to interpolate data points with zero second derivatives at the endpoints of the interval.

Hermite Interpolation:

Hermite interpolation is a method of fitting a curve to a set of data points by using a set of cubic polynomial functions that are defined piecewise over smaller subintervals. The spline functions are required to pass through each of the data points and to have specified first derivatives.

Blending Function:

- Cubic Spline Interpolation method: The cubic spline interpolation method is used to interpolate a set of data points to obtain a smooth curve. It involves dividing the data points into small intervals and fitting a cubic polynomial to each interval. The polynomials are then combined to form a single smooth curve that passes through all the data points. The formula for a cubic spline is:

$$S(x) = a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3$$

where $x_i \le x \le x_{i+1}$, a_i, b_i, c_i, d_i are constants.

- Natural Cubic Spline: The natural cubic spline is a type of cubic spline that has additional constraints at the endpoints to ensure that the curve is smooth and continuous. The constraints are that the second derivative of the curve is zero at the endpoints. This produces a more natural-looking curve than a standard cubic spline.

- Hermite Interpolation: Hermite interpolation is a method of interpolation that uses the function value and its derivative at a given point to determine the polynomial that passes through that point. It is particularly useful for interpolating data that has sharp corners or discontinuities. The formula for Hermite interpolation is:

$P(x) = f(x0)H0(x) + f'(x0)H1(x) + f(x1)H2(x) + f'(x1)H3(x)$

where H0(x), H1(x), H2(x), H3(x) are the Hermite basis functions.

- Blending Function: A blending function is a mathematical function that is used to blend two or more functions or curves together. It is commonly used in computer graphics to smoothly interpolate between two or more shapes or colors. The most common blending function is the linear blending function.

- Cardinal Spline: The cardinal spline is a type of spline curve that is defined by a set of control points and a tension parameter. The tension parameter controls the tightness of the curve, with higher tension producing a tighter curve. The formula for a cardinal spline is:

$P(t) = \Sigma i\ PiBi,n(t)$

where Pi are the control points, Bi,n(t) are the B-spline basis functions, and n is the degree of the spline.

- Bezier Curve: The Bezier curve is a type of curve that is commonly used in computer graphics and CAD/CAM applications. It is defined by a set of control points and a blending function that determines the shape of the curve. The formula for a Bezier curve is:

$$P(t) = \Sigma i{=}0nPibi(t)$$

where Pi are the control points, bi(t) are the Bernstein polynomials, and n is the degree of the curve.

- Fractal Geometry: Fractal geometry is a branch of mathematics that deals with the study of self-similar patterns and structures. Fractals are complex shapes that exhibit self-similarity at different scales. They are commonly used in computer graphics to create realistic textures, landscapes, and other natural phenomena.

- Types of Fractals: There are several types of fractals, including deterministic fractals, random fractals, and self-similar fractals. Deterministic fractals are generated using a fixed set of rules, while random fractals are generated using a stochastic process. Self-similar fractals are generated by repeating the same pattern at different scales.

- Fractal Dimension: Fractal dimension is a measure of the complexity of a fractal shape. It is a non-integer value that describes the degree of self-similarity and the amount of detail in the fractal.

- Deterministic Self Similar Fractals: Deterministic self-similar fractals are generated by repeating the same pattern at different scales using a fixed set of rules.

Chapter 11: Three-Dimensional Geometric and Modeling Transformations - Translation - Rotation - General 3D Rotation (Till Fig: 11-8) - Scaling - Reflections – Shears

Sure, here are the formulas for 3D geometric transformations:

1. Translation:

```

T = [ 1 0 0 Tx ]

   [ 0 1 0 Ty ]

   [ 0 0 1 Tz ]

   [ 0 0 0 1  ]
```

Where Tx, Ty, and Tz are the translation distances in the x, y, and z directions, respectively.

2. Rotation:

```
Rx = [ 1 0       0       0 ]
     [ 0 cos(a) -sin(a)  0 ]
     [ 0 sin(a) cos(a)   0 ]
     [ 0 0       0       1 ]

Ry = [ cos(a)  0 sin(a) 0 ]
     [ 0       1 0      0 ]
     [-sin(a)  0 cos(a) 0 ]
     [ 0       0 0      1 ]

Rz = [ cos(a) -sin(a) 0 0 ]
     [ sin(a) cos(a)  0 0 ]
     [ 0      0       1 0 ]
     [ 0      0       0 1 ]
```

Where a is the rotation angle in radians and Rx, Ry, and Rz are the rotation matrices for the x, y, and z axes, respectively.

3. Scaling:

```
S = [ Sx 0  0  0 ]
    [ 0  Sy 0  0 ]
    [ 0  0 Sz  0 ]
    [ 0  0  0  1 ]
```

Where Sx, Sy, and Sz are the scaling factors in the x, y, and z directions, respectively.

4. Reflection:

```
R = [ -1 0  0  0 ]
    [ 0 -1  0  0 ]
    [ 0  0 -1  0 ]
    [ 0  0  0  1 ]
```

5. Shear:

```
Shx = [ 1   Shy Shz 0 ]
```

```
[ Shy 1   Shz 0 ]

[ Shz Shy 1   0 ]

[ 0   0   0   1 ]
```

Where Shy and Shz are the shearing factors in the y and z directions, respectively.

In addition to these basic transformations, there are also composite transformations, which involve combining two or more transformations to produce a single transformation that achieves the desired effect. These can be achieved by multiplying the transformation matrices together in the appropriate order.

Finally, there is also the concept of a viewing transformation, which is used to transform objects in 3D space into a 2D image that can be displayed on a screen. This involves specifying a camera position and orientation, as well as a projection matrix that maps 3D points onto a 2D plane.

Chapter 12: Three-Dimensional Viewing –

Viewing Pipeline - Line of Sight - World Coordinate to viewing coordinate transformation - Projection - Vanishing point - Definition (Orthographic Parallel Projection, axonometric drawing, View Window, Projection window, view volume) - Frustum

Three-dimensional viewing is an essential aspect of computer graphics that involves creating a three-dimensional representation of an object or a scene on a two-dimensional medium such as a computer screen. The process of viewing a three-dimensional scene involves a series of transformations and projections.

The viewing pipeline is a sequence of stages that an image must pass through to be displayed on a two-dimensional screen. It consists of several stages such as world-coordinate to viewing-coordinate transformation, projection, clipping, and rasterization.

The line of sight is an imaginary line that passes through the eye or the viewpoint and intersects with the object or scene being viewed. It helps to determine which parts of the object are visible and which are hidden.

The world-coordinate to viewing-coordinate transformation is the process of mapping a three-dimensional object from its world coordinate system to the viewing coordinate system. This transformation is necessary to create a perspective view of the object.

Projection is the process of creating a two-dimensional image from a three-dimensional object. There are two types of projections: parallel projection and perspective projection. Orthographic parallel projection is a type of parallel projection where all the projection lines are parallel to each other and perpendicular to the projection plane.

Axonometric drawing is a type of parallel projection where the object is projected onto the viewing plane along three axes at equal angles.

View window is the rectangular region on the projection plane that defines the portion of the scene that will be visible. Projection window is the region of the projection plane onto which the view window is projected.

View volume is the three-dimensional region of space that is visible from a given viewpoint.

Frustum is the truncated pyramid that results from the intersection of the view volume and the projection plane. It defines the volume of space that is visible in the final image.

Chapter 13: Visible-Surface Detection Methods

- Classification of visible surface detection algorithm - Back-face detection - Depth buffer method - Depth Sorting method - BSP Tree method - Painter's Algorithm - Area Subdivision method - Definition (Object space methods, Image space methods)

Chapter 13: Visible-Surface Detection Methods

Visible-surface detection is a critical aspect of computer graphics, as it determines which surfaces in a 3D scene are visible to the viewer and need to be rendered. The goal of visible-surface detection algorithms is to identify the parts of a scene that are visible from a particular viewpoint and to draw only those parts.

Some common visible-surface detection methods are:

1. Classification of visible surface detection algorithm: This includes object space and image space methods. Object space methods are based on analyzing objects in the 3D scene, while image space methods analyze the image produced by the scene.

2. Back-face detection: This method is used to determine which surfaces of an object are facing the viewer. If a surface is facing away from the viewer, it is not visible and can be ignored.

3. Depth buffer method: This is a fast and efficient algorithm that uses a depth buffer to keep track of the depth of each pixel in the scene. The depth buffer is compared with the depth of new pixels being rendered, and only the visible pixels are drawn.

4. Depth Sorting method: This method sorts all the objects in the scene based on their depth from the viewer. The objects are rendered in order from the farthest to the nearest, so that the visible surfaces are drawn last.

5. BSP Tree method: This method involves constructing a binary space partition tree that subdivides the 3D scene into smaller regions, each containing a set of polygons. The tree is then traversed to determine which regions are visible and need to be rendered.

6. Painter's Algorithm: This is a simple visible-surface detection method that involves rendering objects in the order of their distance from the viewer. However, it can produce artifacts when objects overlap.

7. Area Subdivision method: This method involves dividing the screen into smaller areas and determining which objects intersect each area. Only the visible portions of each object are then rendered.

In addition to these methods, there are other visible-surface detection techniques such as ray casting and ray tracing, which are more computationally expensive but can produce more realistic images.

Overall, choosing the right visible-surface detection method depends on the complexity of the scene and the desired level of realism.

Chapter 15: Color Models and Color Applications

- Chromaticity - Color - Visual Spectrum Range - Characteristic of light - Intuitive Color Concept (Shade,tint,tone) - Color Model (RGB,CMYK,HSV) - Definition (Color,Color Model, Dominant Frequency, Hue, Brightness,Intensity, Purity/Saturation, Chromaticity, Complementary color, Color Gammate, Primary Color, Shade, tint, tone)

Color models are mathematical models used to describe the way colors are represented and displayed on computer systems. There are several color models used in computer graphics, such as RGB, CMYK, and HSV.

RGB stands for Red Green Blue and is an additive color model, which means that colors are created by adding various amounts of red, green, and blue light. This color model is widely used in computer graphics and digital imaging, as it is used to represent colors on computer screens and other electronic displays.

CMYK stands for Cyan Magenta Yellow Black and is a subtractive color model used in printing. This color model is based on the idea that the colors we see are created by subtracting certain wavelengths of light from white light. In printing, colors are created by combining various amounts of cyan, magenta, yellow, and black inks.

HSV stands for Hue Saturation Value and is a color model that is based on the way that humans perceive color. Hue refers to the color of the light, saturation refers to the intensity or purity of the color, and value refers to the brightness of the color.

Chromaticity is a term used to describe the quality of a color that is independent of its brightness. It is often used to describe the hue and saturation of a color.

Brightness is a term used to describe the amount of light in a color. It is often used to describe the overall intensity of a color.

Intensity is a term used to describe the strength of a color. It is often used to describe the strength of a color in relation to other colors.

Purity/Saturation refers to the intensity or purity of a color. A pure color is one that contains no white light, while a less pure color contains more white light.

Complementary color refers to the color that is opposite to a given color on the color wheel. For example, the complementary color of red is green.

Color Gamma is a term used to describe the way that colors are displayed on computer screens. It refers to the relationship between the input values and the output brightness of a color.

Primary colors are colors that cannot be created by mixing other colors. In the RGB color model, the primary colors are red, green, and blue, while in the CMYK color model, the primary colors are cyan, magenta, and yellow.

Shade, tint, and tone are terms used to describe variations of a given color.

Shade refers to a darker version of a color,

Tint refers to a lighter version of a color, and

Tone refers to a variation of a color that is created by adding gray to the color.

Indexed color is a technique used to reduce the number of colors in an image while preserving its visual appearance. In this technique, a color palette of up to 256 colors is defined, and each pixel in the image is represented by an index into this palette. This reduces the amount of memory required to store the image, and allows for faster processing and display. Indexed color is often used in computer graphics and animation applications where fast processing and small file sizes are important, such as in video games and web graphics.

YIQ model: The YIQ color model is used in television broadcast systems. It separates the color information from the brightness information. The Y component represents the brightness level, while the I and Q components represent the color information. The I component represents the amount of redness or greenness in the color, and the Q component represents the amount of blueness or yellowness in the color.

The Y component is the grayscale to drive old black-and-white TVs. The I component goes from orange to blue and the Q component goes from purple to green.