Circle drawing algorithm in computer graphics pdf

Drawing a circle on the screen is a little tricky as drawing a line. There are two popular algorithms for generating a circle – Bresenham's algorithm and the midpoint circle algorithm. These algorithms are based on the idea of determining the following points needed to draw a circle. Let's discuss the algorithms in detail − Equation circle is $X ^ {2} + Y ^ {2} = r ^ {2}, $ where are the radius. Bresenham algorithm We can't display a continuous arc on a raster display. Instead, we need to select the nearest pixel position to complete the arc. In the following image you can see that we put the pixel in (X, Y) place and now we have to decide where to put the next pixel – on N (X+1, Y) or on S (X+1, Y-1). This may be decided by decision-making parameter d. If d < = 0, then N(X+1, Y) is selected as the next pixel. If d > 0, S(X+1, Y-1) is selected as the next pixel. Algorithm Step 1 − Get the coordinates of the center of the circle and radius and save them to x, y and R. Set P=0 and Q=R. Step 2 − Set the decision parameter D = 3 – 2R. Step 3 − Repeat step-8 while P ≤ Q. Step 4 − Call drawing circle (X, Y, P, Q). Step 5 − Increase p. Step 6 − If D < 0 then D = D + 4P + 6. Step 7 − Else Set R = R - 1, D = D + 4(P-Q) + 10. Step 8 − Call draw circle (X, Y, P, Q). Draw a circle method(X, Y, P, Q). Call putpixel (X+P, Y+Q). Call putpixel (X - P, Y + Q). Call putpixel (X+P, Y -Q). Call putpixel (X - P, Y - Q). Call putpixel (X+Q, Y+P). Call Putpixel (X - Q, Y + P). Call putpixel (X+ Q, Y - P). Call putpixel (X - Q, Y - P). Midpoint algorithm Step 1 − Radius of entry r and center of circle $(x_{c,} y_{c})$ to get the first point on the circumference of the circle centered on the origin as (x0, y0) = (0, r) Step 2 − Calculate the starting value of the decision parameter as P_{0} = 5/4 – r (see the following description of the simplification of this equation.) f(x, y) = x2 + y2 - r2 = 0 f(xi - 1/2 + e, yi + 1) = (xi - 1/2 + e)2 + (yi + 1 2 - r2 = (xi- 1/2)2 + (yi + 1)2 - r2 (xi - 1/2)e + e2 = f(xi - 1/2, yi + 1) + 2(xi - 1/2)e + e2 = 0 Let di = f(xi - 1/2, yi + 1) = -2(xi - 1/2)e - e2 So, If e < 0 then di > 0 so select point S = (xi - 1 , yi + 1). di+1 = f(xi - 1 - 1/2, yi + 1 + 1) = ((xi - 1/2) - 1)2 + ((yi + 1) + 1)2 - r2 = di - 2(xi - 1) + 2(yi + 1) + 1 = di + 2(yi + 1 - xi + 1) + 1 If e >= 0 then di <= 0 so choose point T = (xi, yi + 1) di+1 = f(xi - 1/2, yi + 1 + 1) = di + 2yi+1 + 1 The initial value of di is d0 = f(r - 1/2, 0 + 1) = (r - 1/2)2 + 12 - r2 = 5/4 - r {1-r can be used if r is an integer} When point S = (xi - 1, yi + 1) is chosen then di+1 = di + -2xi+1 + 2yi+1 + 1 When point T = (xi, yi + 1) is chosen then di+1 = di + 2yi+1 + 1 Step 3 − At each X_{K} position starting at K=0 , perform the following test − If FK < 0 then there is another point on the circle (0.0) (XK+1,YK) and PK+1 = PK + 2XK+1 + 1 Else PK+1 = PK + 2XK +1 + 1 – 2YK+1 Where, 2XK+1 = 2XK+2 and 2YK+1 = 2YK-2. Step 4 — Determination symmetry of points in the other seven octaves. Step 5 − Drag each calculated pixel position (X, Y) to a circular path centered on $(X_{C,} Y_{C})$ and render the coordinate values. X = X + XC, Y = Y + YC Step 6 − Repeat step-3 to 5 to X > = Y. Page 2 Transformation means changing some graphics to something else by applying rules. We can have different types of transformations such as translation, scaling up or down, rotation, cutting, etc. When a transformation takes place on a 2D plane, it is called a 2D transformation. Transforms play an important role in computer graphics to move graphics on the screen and change their size or orientation. Homogeneous coordinates To perform a sequence of transformations, such as translation, followed by rotation and scaling, we need to proceed sequentially – translate coordinates, rotate translated coordinates, and then scale rotated coordinates to complete the composite transformation. To shorten this process, we need to use the 3×3 transformation matrix instead of the 2×2 transformation matrix. To convert 2×2 to 3×3, we need to add another fictitious W coordinate. In this system we can represent all transformation equations in the multiplication matrix. Any Cardaesian point P(X, Y) may be converted into homogeneous coordinates p' (Xh, Yh, h). Translation Translation moves the object to a different location on the screen. You can translate a point in 2D by adding the translation coordinate (tx, you) to the original coordinate (X, Y) to get a new coordinate (X', Y). From the above image you can write that − X' = X + tx Y' = Y + ty Pair (tx, you) is called the translation vector or displacement vector. The above equations can also be represented by using column vectors. $P = \frac{[X]}{[Y]}$p' = $\frac{[X']}{[Y']}$T = $\frac{[t_{x}]}{[t_{y}]}$ We can write it as − P' = P + T rotation by rotating, we rotate the object at a certain angle θ (theta) from its origin. From the following figure we can see that point P(X, Y) is located at an angle of φ from the horizontal X coordinate with the distance r from the origin. Suppose you want to rotate it at an angle of θ. When you turn to a new location, you will receive a new point P' (X', Y'). Using the standard trigonometry, the original P(X, Y) coordinate can be represented as − $X = r \, cos \, \, \phi ...... (1)$ $Y = r \, sin \, \phi ...... (2)$ In the same way, we can represent point P' (X', Y') as − ${x}'= r \: cos \: \: \left (\phi \: + \: \theta \right) = r\: cos \: \phi \: cos \: \: \theta \: − \: r \: sin \: \phi \: sin \: \: \: \theta .... (3)$ ${y}'= r \: sin \: \left (\phi \: + \: \theta \right) = r\: cos \: \phi \: sin \: \theta \: + \: r \: sin \: \phi \: cos \: \: \theta .... (4)$ Substituting equation (1) & (2) in (3) & (4) respectively, we get ${x}'= x \: cos \: \theta − \: y \: sin \: \: \: \theta $${y}'= \: sin \: \theta + \: y \: cos \: \: \theta $ Represents the above equation in the form of a matrix, $$[X' Y'] = [X Y] \begin{bmatrix} cos\theta & sin\theta \\ −sin\theta & cos\theta \end{bmatrix}OR $$ P' = P . R Where R is the rotation matrix $$R = \begin{bmatrix} cos\theta & sin\theta \\ −sin\theta & cos\theta \end{bmatrix}$$ The angle of rotation can be positive and negative. For a positive angle of rotation, we can use the above rotary matrix. However, for negative angle rotation, the matrix changes as below screenshot shown − $R = \begin{bmatrix} cos(−\theta) & sin(−\theta) \\ -sin(−\theta) & cos(−\theta) \end{bmatrix}. $$ $$=\begin{bmatrix} cos\theta & −sin\theta \\ sin\theta & cos\theta \end{bmatrix} \left (\because cos(−\theta) = cos \theta \; a\; sin(−\theta) = −sin \theta \right)$$ Scale To resize the object, scaling is used. In the scaling process, you can expand or compress the dimensions of an object. You can achieve a scale by multiplying the object's original coordinates by a scale factor to get the result you want. Supposing the original coordinates are (X, Y), the scaling factors are (SX, SY), and the produced coordinates are (X', Y). This can be mathematically illustrated as shown below − X' = X . SX and Y' = Y . SY Scaling Factor SX, SY scales object in direction X and Y, respectively. The above equations can also be represented in the form of a matrix as below − $$\bin{X'}{Y'} = \binom{X}{Y} \begin{bmatrix} S_{x} & 0\\ 0 & S_{y} \end{bmatrix}$$ OR P' = P . S Where S is the scaling nut. The scaling process is shown in the following figure. If we provide values less than 1 per scaling factor S, then we can reduce the size of the object. If we provide values greater than 1, then we can increase the size of the object. Reflection reflection is a mirror image of the original object. In other words, we can say that this is a rotational operation with 180°. When you transform a reflection, the size of the object does not change. The following data show reflections concerning the X and Y axis and the origin. The cut transformation that oblique shape of the object is called the cut transformation. There are two editing transformations of X-Shear and Y-Shear. One shifts the X coordinates of the value and the other shifts the Y coordinate values. However; in both cases, only one coordinate changes its coordinates and the other retains its values. Shearing is also called Skewing. The X-Shear X-Shear preserves the Y coordinate and changes to the X coordinates, causing vertical lines to tilt right or left, as shown in the figure below. The transformation matrix for X-Shear can be represented as − $$X_{sh} = \begin{bmatrix} 1& shx& 0\\ 0& 1& 0\\ 0& 0& 1 \end{bmatrix}$$ Y' = Y + Shy . X X' = X Y-Shear Y-Shear preserves the X coordinates and changes the Y coordinates, which cause the horizontal lines to be transformed into lines that slope up or down as shown in the following figure. In the event that the can be represented in matrix from – $Y_{sh} \begin{bmatrix} 1& 0& 0\\ shy& 1& 0\\ 0& 0& 1 \end{bmatrix}$$X' = X + Shx . Y Y' = Y Composite transformation If the transformation of plane T1 is followed by a second transformation of plane T2, then the result itself may be represented by one transformation T, which is the composition T1 and T2 taken respectively. This is written as T = T1 T2. Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix. Combined matrix − [T][X] = [X] [T1] [T2] [T3] [T4] .... No, it's okay. Where [Ti] is any combination of translation Cutting Cutting reflection rotation Changing the order of transformation would lead to different results because generally matrix multiplication is not cumulative, i.e. [A] . [B] ≠ [B] . [A] and multiplication order. The basic purpose of folding transformations is to gain efficiency by applying one composed transformation to a point and not by applying a series of transformations, one by one. For example, to rotate an object by any point (Xp, Yp), we need to perform three steps − Translate point (Xp, Yp) to origin. Turn it around about origin. Finally, translate the center of rotation where it belonged. Turning a 33D page is not the same as turning 2D. For 3D rotation, we must specify the angle of rotation together with the axis of rotation. We can perform 3D rotation on axes X, Y and Z. They are represented in matrix form as below − $$R_{x}(\theta) = \begin{bmatrix} 1& 0& 0& 0\\ 0& cos\theta & −sin\theta& 0& 0\\ 0& sin\theta & cos\theta& 0\\ 0& 0& 0& 1\\ \end{bmatrix} R_{y}(\theta) = \begin{bmatrix } cos\theta& 0& sin\theta& 0\\ 0& 1& 0\ 0\\ −sin\theta& 0& cos\theta& 0\ 0& 0& 1\1\\ \ \end{bmatrix} R_{z}(\theta) =\begin{bmatrix} cos\theta & −sin\theta & 0& 0\\ sin\theta & cos\theta & 0& 0\\ 0& 0& 1& 0\\ 0& 0& 0& 1 \end{bmatrix}$$ The following figure explains the rotation of different axes − Scaling You can change the size of an object by scaling. In the scaling process, you can expand or compress the dimensions of an object. You can achieve a scale by multiplying the object's original coordinates by a scale factor to get the result you want. The following figure shows the effect of 3D scaling − Three coordinates are used for 3D scaling. Supposing the original coordinates are (X, Y, Z), the scaling factors are $(S_{X,} S_{Y,} S_{z})$, and the coordinates created are (X', Y', Z'). This can be mathematically illustrated as shown below − $S = \begin{bmatrix} S_{x}& 0& 0\ 0&S_{y}& 0\\ 0& 0& 0& S_ {z}& 0\\ 0& 0& 0&1 \end{bmatrix}$P' = P*S $[{X}' \:\:\: {Y}' \:\:\: {Z}' \:\:::1] = [X \:\:\:Y \:\:\: Z \:\:\: 1] \:\: \begin{bmatrix} S_{x}& 0& 0& 0\\ 0& S_ 0\\ 0& 0& 0& S_{z}& 0\\ 0& 0& 0& 1 \end{bmatrix}$ $ = [X.S_{x} \:\:\ : Y.S_{y} \:\:\: Z.S_{z} 1]$ Cut transformation that oblique shape shape object is called a shear transformation. As in 2D editing, we can cut an object along the X, Y-axis or Z-axis axis in 3D. As shown in the above figure, there is a P coordinate. You can cut to get a new P' coordinate that can be represented in the 3D matrix form as below screenshot shown − $Sh = \begin{bmatrix} 1 & sh_{x}^{y} & sh_{x}^{z} & 0 \\ sh_ sh_ {y}^{x} & 1 & sh_{y}^{z} & 0 \\ sh_{z}^{x} & sh_{z}^{y} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ P' = P ≥ Sh $X' = X + Sh_{x}^{y} Y +Sh_ {x}^{z} Z$$Y' = Sh_{y}^{x}X + Y +sh_{y}^{z}Z$$Z = Sh_ of}^{x}X + Sh_{z}^{y}Y + Z$Transformation Matrix Is an essential transform tool. The matrix of dimensions n × m shall be multiplied by the coordinate of the objects. Typically, 3 x 3 or 4 x 4 nuts are used for transformation. For example, consider the following matrix for different operations. $T = \begin{bmatrix} 1& 0& 0\\ 0& 1& 0\ 0& 0& 1& 0\\ t_{x}& t_{y}& t_{z }& 1\\ \end{bmatrix}$ $S = \begin{bmatrix} S_{x}& 0& 0\ 0& S_{y}& 0& 0\\ 0& 0& 0& S_{z}& 0\\ 0& 0& 0& 1 \end{bmatrix}$ $Sh = \begin{bmatrix} 1& sh_{x}^{y}& sh_{x}^{z} & 0\\ sh_{1 ^{x}& 1 & sh_{y}^{z}& 0\\ sh_{z}^{x}& sh_{z}^{y}& 1& 0\\ 0& 0& 0& 1 \end{ bmatrix}$ Matrix Scaling Matrix Translation Shear Matrix $R_{x}(\theta) = \begin{bmatrix} 1& 0& 0& 0\\ 0& cos\theta & -sin\theta& 0\\ 0& sin\theta & cos\theta& 0\\ 0& 0& 0& 1\\ \end{bmatrix}$ $R_{y}(\ theta) = \begin{bmatrix} cos\theta A 0& sin\theta& 0\\ 0& 1& 0\ 0\\ -sin\theta& 0& cos\theta& 0\ 0& 0& 1\\ \end{bmatrix}. $$R_{z}(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0\ sin\theta & cos\theta & 0& 0\\ 0&0& 0& 0& 1& 0\\ 0&0&1 \end{bmatrix}$ Rotation Matrix Page 4 In a 2D system, we use only two X and Y coordinates, but 3D adds an additional Z. 3D graphics coordinate and their use is the basis for entertainment , games and computer-assisted design industry. It is an ongoing field of research into scientific visualization. In addition, 3D graphics components are now part of almost every personal computer, and although traditionally designed for graphics-intensive software such as games, they are increasingly used by other applications. Parallel projection The parallel projection removes the coordinates of the z-coordinates and parallel lines from each peak on the object until they intersect the plane of view. In parallel with the projection, we specify the direction of the projection instead of the center of the projection. In parallel projection, the distance from the center of the projection to the plane of the project is infinite. With this type of projection, we combine planned peaks by line segments that correspond to the joins on the original object. Parallel projections are less realistic, but are good for accurate measurements. For this type of projection, parallel lines remain parallel and angles are Different types of parallel projections are displayed in the following hierarchy. Orthographic projection In orthographic projection, the direction of projection is to the projection of the plane. There are three types of orthographic projections − Front projection Upper projection Lateral projection Oblique projection In the oblique projection, the direction of projection is not normal to the projection of the plane. In oblique projection, we can display an object better than an orthographic projection. There are two types of oblique projections - Cavalier and Cabinet. Cavalier protrusion is a 45° angle with projection plane. The projection of a line perpendicular to the plane of view shall be the same length as the line itself in the Cavalier projection. In cavalier projections, the foreshortening factors for all three main directions are the same. The projection of the housing is a 63,4° angle with a projection plane. In the cabinet projection, the lines perpendicular to the observation area are projected to 1/2 of their actual length. Both projections are shown in the following figure − Isometric projections Orthographic projections that show more than one side of an object are called axonometric orthographic projections. The most common axonometric projection is an isometric projection in which the projection plane crosses each coordinate line in the model coordinate system at the same distance. In this projection of parallelism of lines are preserved, but angles are not preserved. The following figure shows the isometric projection − Prospective projections In a prospective projection, the distance from the center of the projection to the plane of the project is final, and the size of the object changes inversely with a distance that looks more realistic. Distance and angles are not maintained and parallel lines do not remain parallel. Instead, they all converge in one place called the centre of the projection or projection reference point. There are 3 types of forward-looking projections that are shown in the chart below. One point of perspective projection is easy to draw. The two-point perspective projection gives a better depth impression. The three-point projection perspective is the hardest to draw. The following figure shows all three types of forward-looking projection – Translation In 3D translation, we transmit the Z coordinate together with the X and Y coordinates. Translation moves the object to a different location on the screen. The following illustration shows the effect of translation − The point can be translated into 3D by adding translation coordinates $(t_{x,} t_{y,} t_{z})$ to the original coordinate (X, Y, Z) to get a new coordinate (X', Y', Z). $T = \begin{bmatrix} 1& 0& 0\\ 0& 1& 0\ 0& 0& 1& 0\\ t_{x}& t_{y}} & t_{z}& 1\\ \end{bmatrix}$ P' = P*T $[X' \:\: Y' \:\: Z' \:\: 1] \: = \: [X \:\: Y \\\ :\: Z \:\: 1 \: \begin{bmatrix} 1& 0& 0\ 0\ 0& 1& 0\ 0& 0& 1& 0\\ t_{x}. & t_{y}& t_{z}& 1\\ \end{bmatrix}$ $= [X + t_{x} \:\:\: Y + t_{y} \:\:\ : Z + t_{z} \:\:\: 1]$ 1]$