

Algorithm for Bresenham Line Drawing and its Rotation

Step 1:

Input the starting and ending points of the line and enter rotation/translation values:

- Starting point (x_0, y_0)
- Ending point (x_1, y_1)

Step 2:

Calculate the absolute differences between the coordinates:

- $\Delta x = |x_1 - x_0|$
- $\Delta y = |y_1 - y_0|$

Step 3:

Determine the step directions for x and y:

- $s_x = 1$ if $x_1 > x_0$, otherwise -1
- $s_y = 1$ if $y_1 > y_0$, otherwise -1

Step 4:

Initialize lists to store the points:

- $xes = [x_0]$
- $yes = [y_0]$

Step 5:

Iteratively calculate points for the line using Bresenham's algorithm:

- If $\Delta x > \Delta y$, update x coordinates, otherwise update y coordinates.
- Append each new point to the xes and yes lists.

Step 6:

Define a rotation matrix to rotate the line by a specified angle around the origin:

- Use $\theta = 45^\circ$ ($\pi/4$ radians).

Step 7:

Define translation matrices to move the line:

- Translation matrix: Translate by x_0, y_0 units.
- Reverse translation matrix: Translate back by $-x_0, -y_0$ units.

Step 8:

Calculate the composite transformation matrix:

- Composite = Translation * Rotation * Reverse Translation.

Apply this transformation to the original line.

Step 9:

Plot the original line and the transformed line on the same graph:

- Use matplotlib to display the lines with different styles and colors.