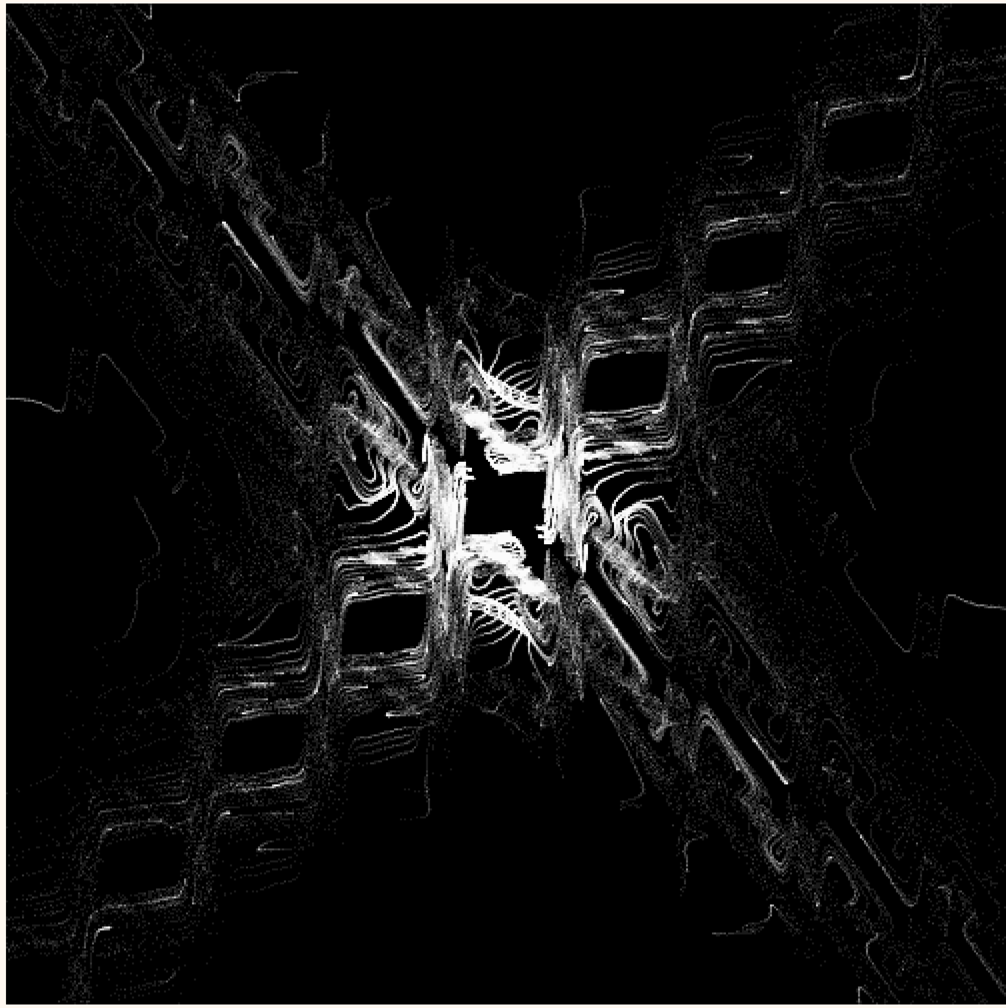
2D Transformations

2D Transformations

2D transformations are essential tools in the world of computer graphics, digital design, and artistic expression. These mathematical operations allow us to manipulate and reshape flat, two-dimensional objects in captivating ways, opening up new realms of creativity and problem-solving. From simple translations and rotations to complex scaling and shearing, the versatility of 2D transformations empowers designers, artists, and engineers to bring their visions to life and explore innovative visual concepts.



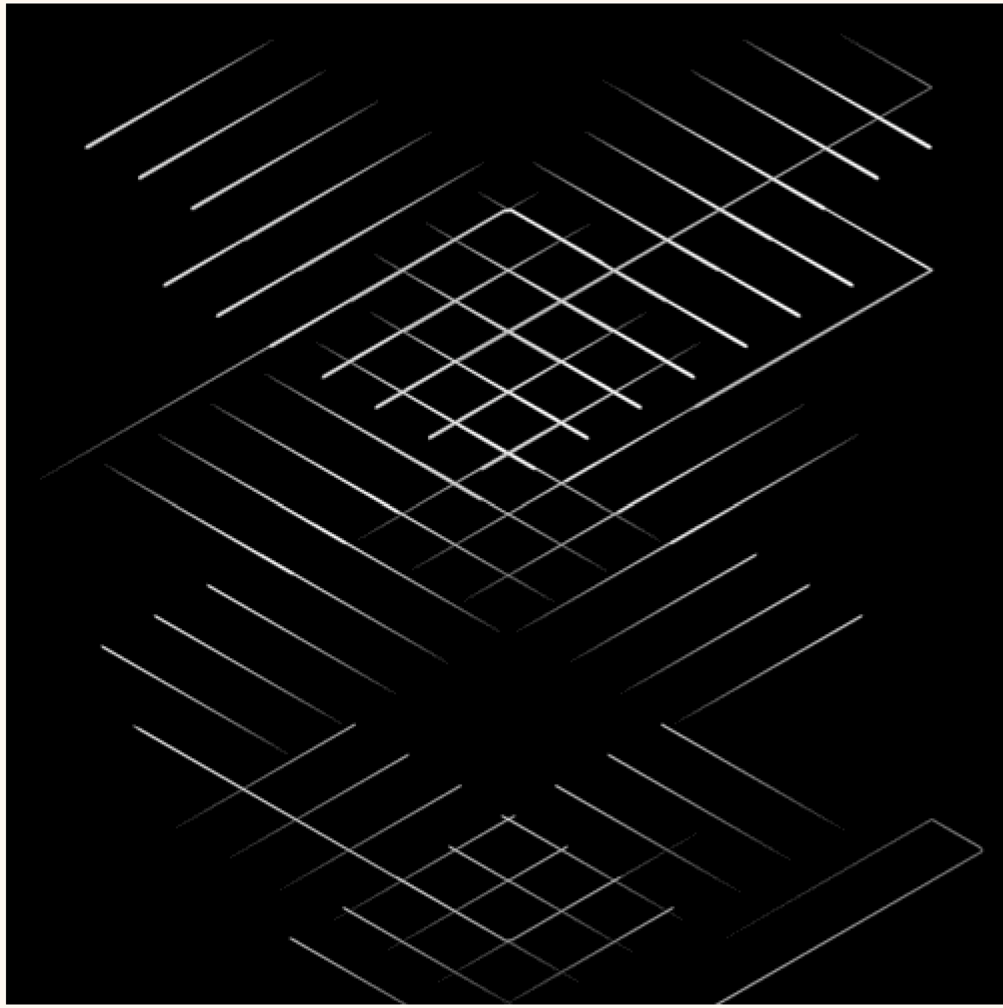
Attributes

In the context of 2D transformations in computer graphics, attributes refer to the properties of an object that can be changed or manipulated.

Example: Position(Translation), Size (Scaling), Shape (Shear)

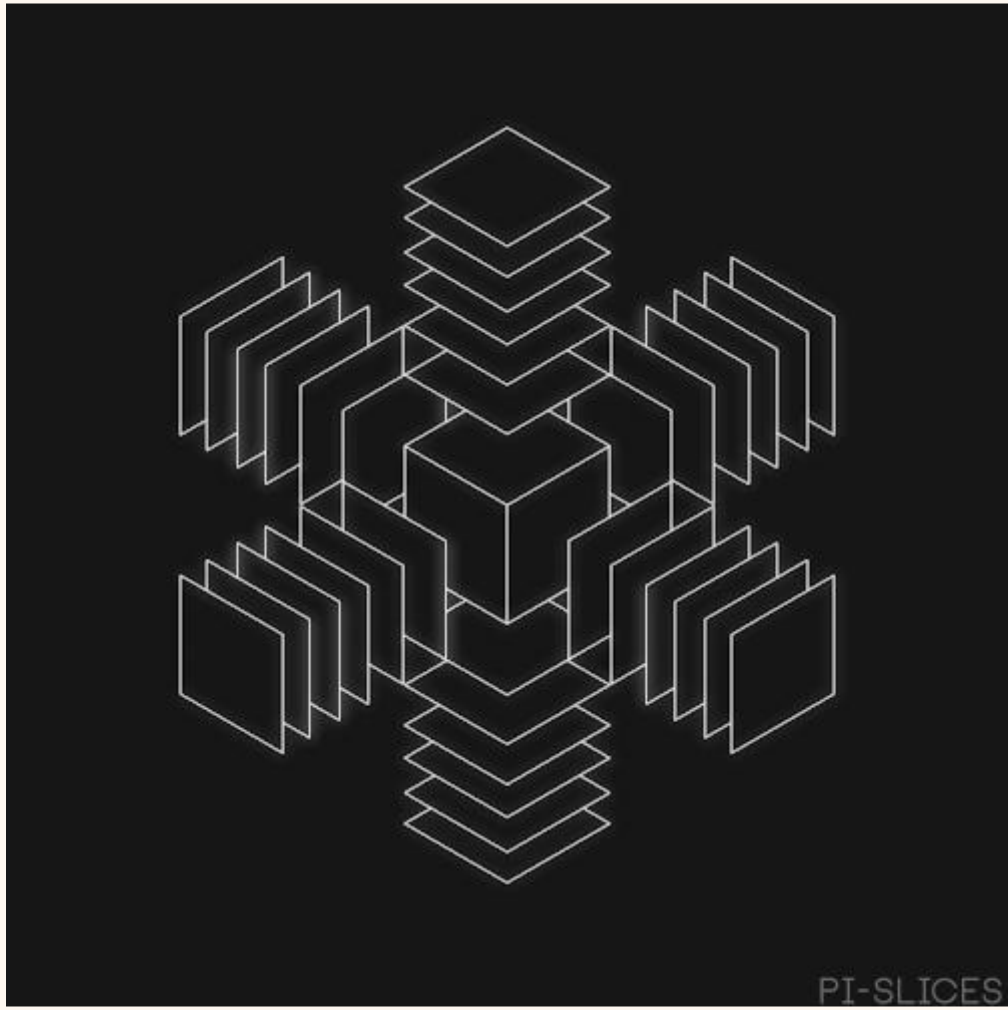
Attributes – Line attributes

Line attributes define the appearance of lines in 2D graphics. These attributes include line type, line width, and line color. When applying 2D transformations such as translation, rotation, and scaling, these attributes help maintain the visual consistency of the lines.



Attributes – Line attributes

Line Width



The thickness or width of a line is a crucial attribute that can be adjusted to create different visual effects. Varying the line width can help emphasize certain elements, guide the viewer's eye, or establish a specific aesthetic.

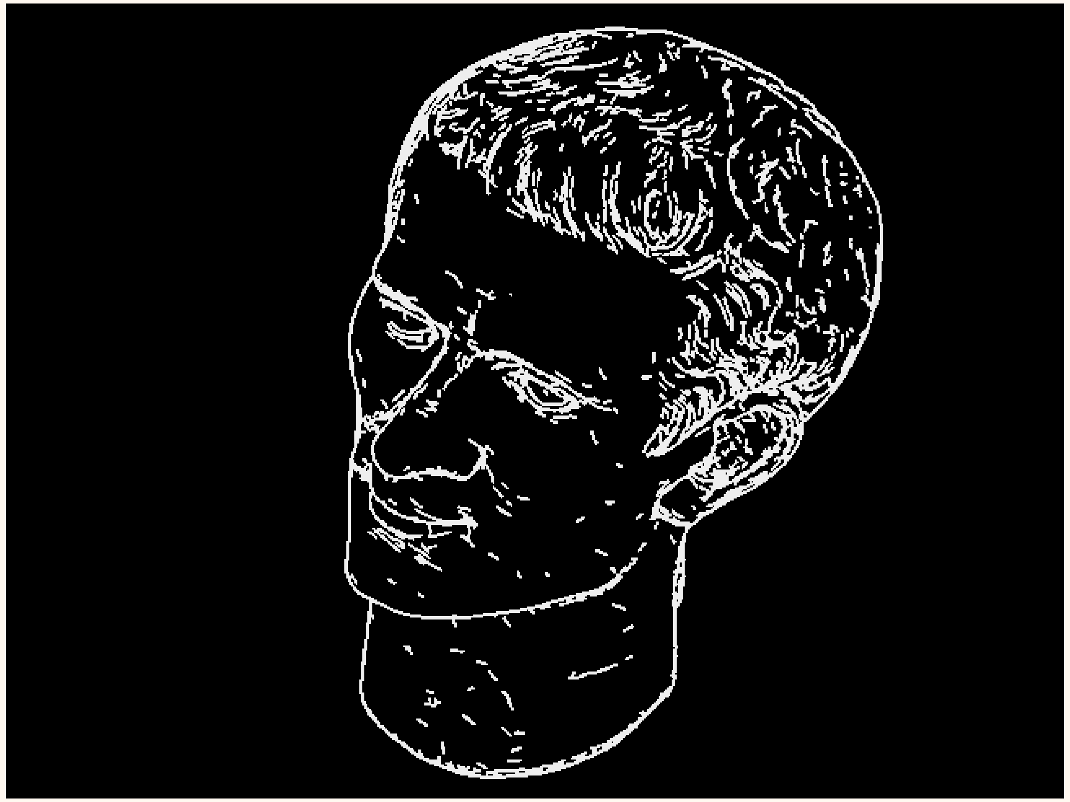
Example: A line with a width of 3 pixels will appear thicker than a line with a width of 1 pixel.

Attributes – Line attributes

Line Style

The style of a line, whether solid, dashed, dotted, or a custom pattern, can convey different meanings and moods. Selecting the appropriate line style can enhance the overall design and communicate specific information.

Example: A dashed line might be represented as '- - - - - -'.

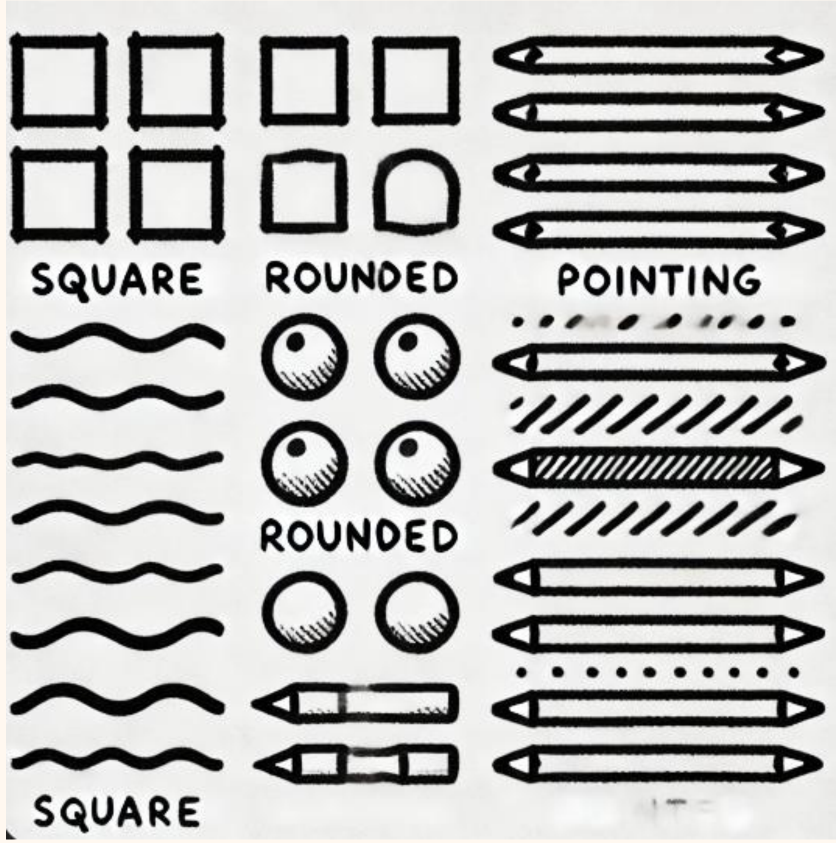


Attributes – Line attributes

Line Ending

The termination of a line, known as the line ending, can be square, rounded, or pointed. These endpoints can impact the visual impact and create a sense of finality or continuity.

Example:



In the image above:

- The first line has a square ending.
- The second line has a rounded ending.
- The third line has a pointed ending.

Attributes – Curve attributes

In 2D transformations, curve attributes are essential for defining the characteristics and behavior of curves. These attributes influence how curves are rendered and manipulated in computer graphics. Key curve attributes include:

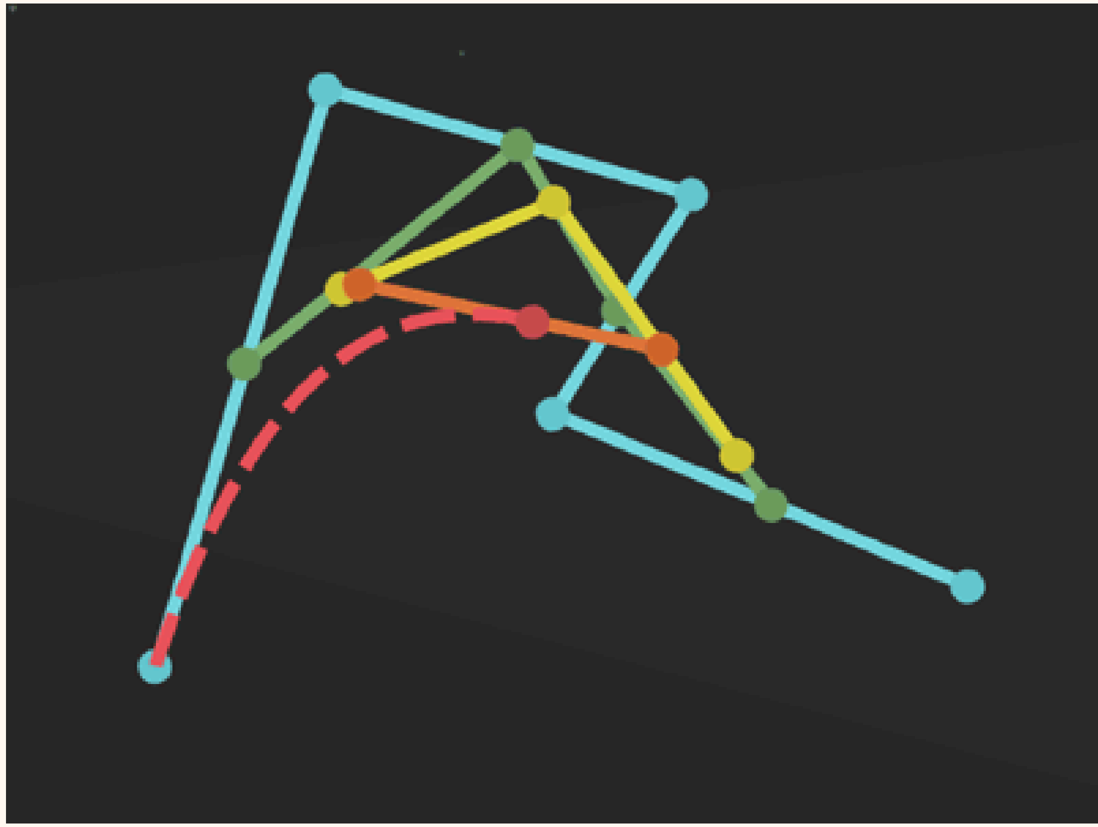
Type of Curve:

- **Implicit Curves**: Defined by an equation $f(x,y)=0$.
- **Explicit Curves**: Defined by a function $y=f(x)$.
- **Parametric Curves**: Defined by parameter equations $x=f(t)$ and $y=g(t)$.

Attributes – Curve attributes

Control Points:

Points that determine the shape and position of the curve. These are used in spline curves (A spline is a smooth curve that passes through or near a set of points that influence the shape of the curve.) such as Bezier and B-splines.



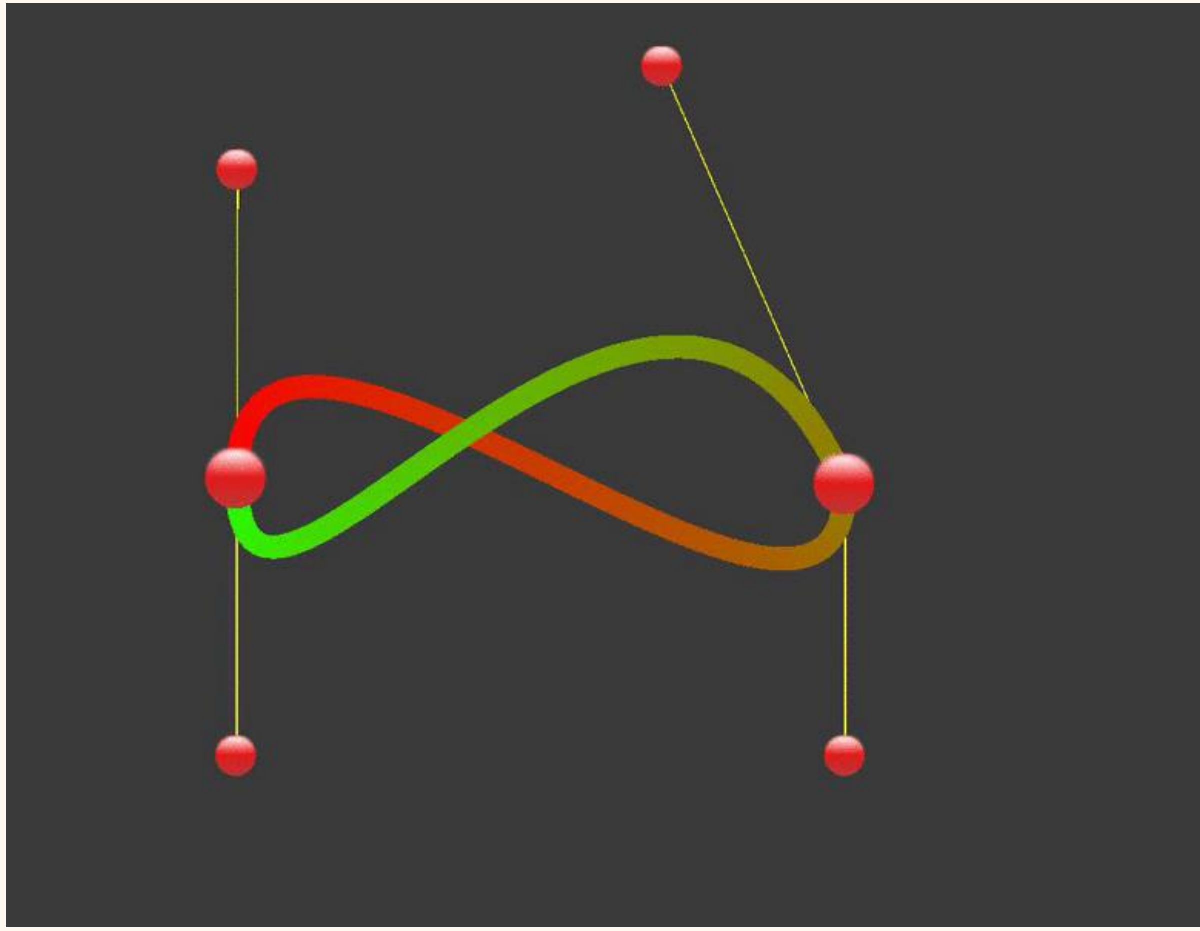
Continuity:

Describes the smoothness of the curve at the control points. For instance, G0, G1, and G2 continuity ensure that the curve is connected and smooth to varying degrees at the joints.

Attributes – Curve attributes

Curve Equation Coefficients:

Coefficients that define the mathematical equation of the curve. These coefficients change when the curve is transformed, such as through translation, scaling, or rotation.



Attributes – Colour and Grayscale Levels

1

Color Attributes

Color in digital graphics is typically represented using the RGB color model. In this model:

- **Red, Green, and Blue (RGB)** components are combined to create various colors.
- Each component can have values ranging from 0 to 255, allowing for over 16 million possible colors.
- Colors can be stored as numerical values in a frame buffer, used for rendering images on screens.

Attributes – Colour and Grayscale Levels

2

Grayscale Levels

Grayscale images differ from color images as they do not contain color information but instead represent different shades of gray. Each pixel in a grayscale image has:

- **Luminance Value**: This determines the brightness of the pixel. The values range from 0 (black) to 255 (white), with various shades of gray in between.
- **Single Intensity Level**: Unlike RGB images, each pixel in a grayscale image is represented by a single intensity value that indicates its brightness.

Attributes – Colour and Grayscale Levels

Key Differences

- **Color Images**: Use multiple channels (usually RGB) to represent different colors.
- **Grayscale Images**: Use a single channel to represent varying intensities of gray.

Attributes – Colour and Grayscale Levels

Applications

- **Grayscale**: Commonly used in medical imaging, document scanning, and certain artistic effects where color is not necessary.
- **Color**: Essential for natural images, videos, and most graphical applications where color differentiation is crucial.

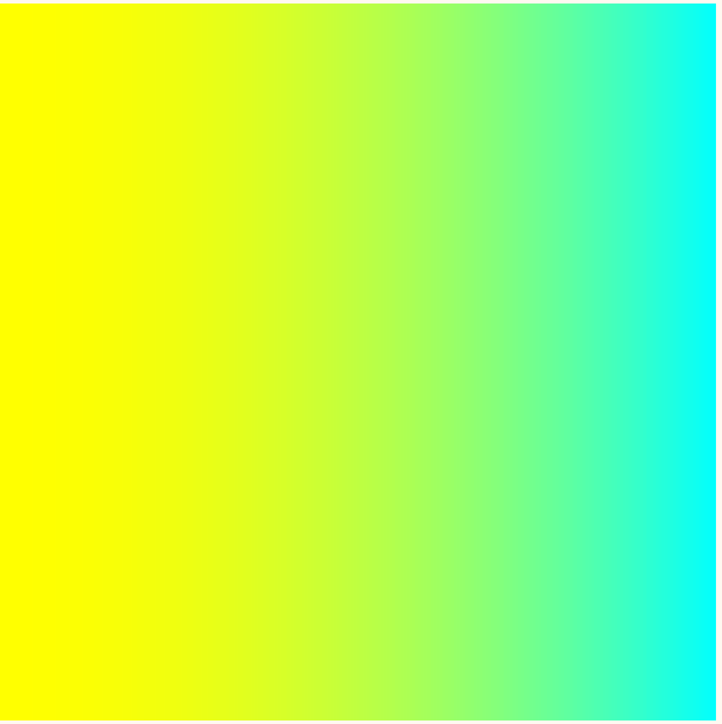
Attributes – Area Fill Attributes

Solid Fill

A solid fill applies a single, uniform color to an enclosed area, creating a clean, uninterrupted visual effect.

Gradient Fill

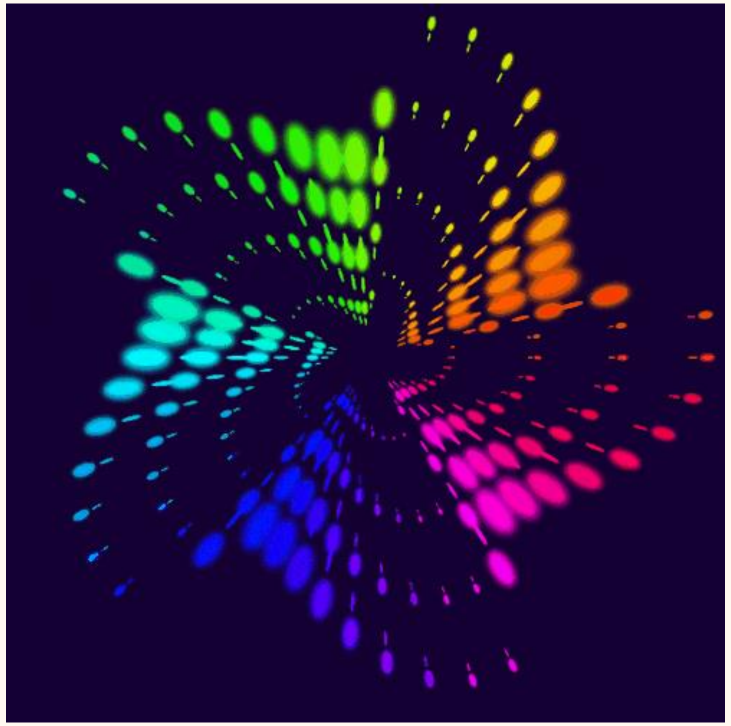
Gradient fills allow for a smooth transition between two or more colors, creating a sense of depth, movement, or emphasis.



Attributes – Area Fill Attributes

Pattern Fill

Pattern fills use repeating images, textures, or shapes to fill an area, adding visual interest and depth to a design.



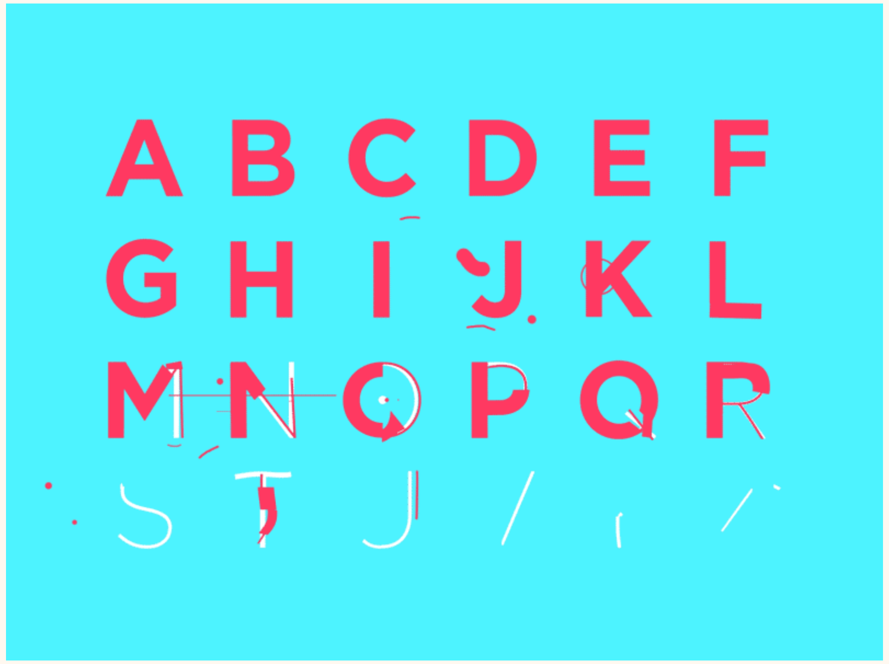
Transparency

Adjusting the transparency of an area fill can create layering effects, reveal underlying elements, or establish a sense of depth.

Attributes – Character Attributes

Font

The font, or typeface, used for text can dramatically impact the overall aesthetic and tone of a design, conveying different moods and styles.



Size

The size of text, measured in points or pixels, can be used to establish hierarchy, emphasize important information, or create visual balance.

Attributes – Character Attributes

Color

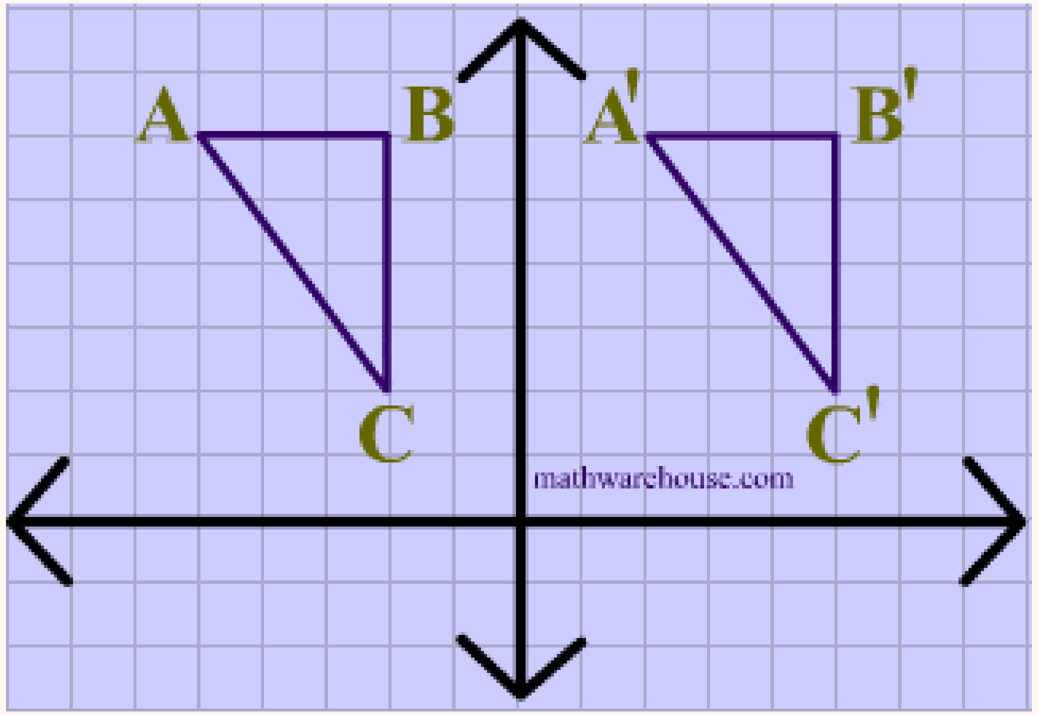
The color of text can be used to draw attention, create contrast, or establish a specific mood or theme within a design.

Alignment

The alignment of text, whether left, right, center, or justified, can impact the overall layout and readability of a design.

Two Dimensional Transformation – Basic Translate

Translation in 2D transformations involves moving every point of an object a certain distance in the x and y directions. This process shifts the position of the object without altering its shape, size, or orientation.



Translation Formula

For a point $P(x,y)$ that needs to be translated to a new point $P'(x',y')$ by a translation vector (tx,ty), the translation equations are:

$x' = x + tx$

$y' = y + ty$

Two Dimensional Transformation – Basic Translate

Steps of Translation:

Identify the Translation Vector: Determine the distances tx and ty by which the object needs to be moved in the x and y directions, respectively.

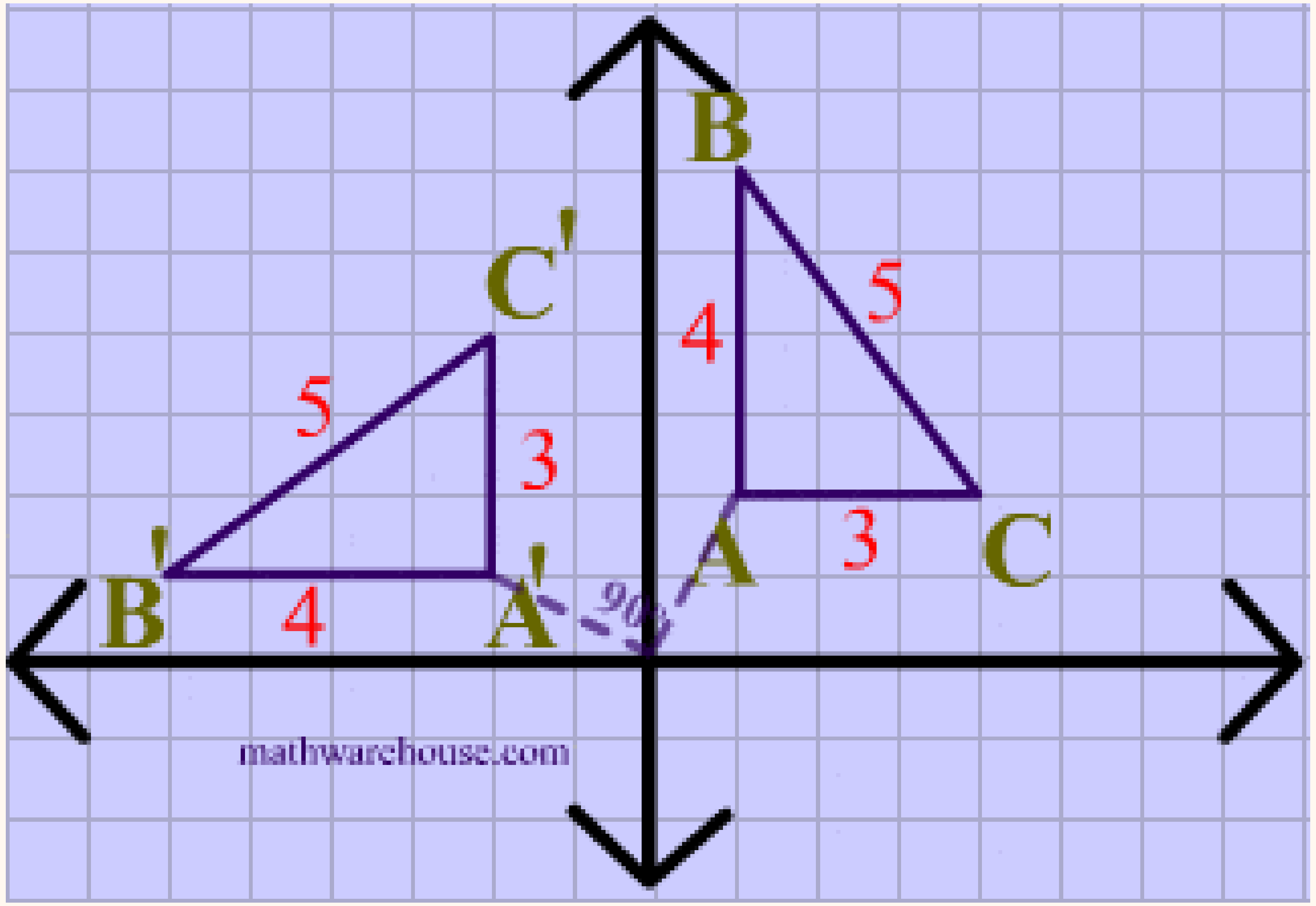
Apply the Translation: Add tx to the x-coordinates and ty to the y-coordinates of all points defining the object.

Update the Object's Position: The object is now repositioned at the new coordinates.

Two Dimensional Transformation – rotate

Rotation

Rotation is a fundamental 2D geometric transformation that involves rotating a point or shape around a fixed point, usually the origin, by a specified angle. The rotation is typically counterclockwise, and the angle is measured in degrees or radians.



Two Dimensional Transformation – rotate

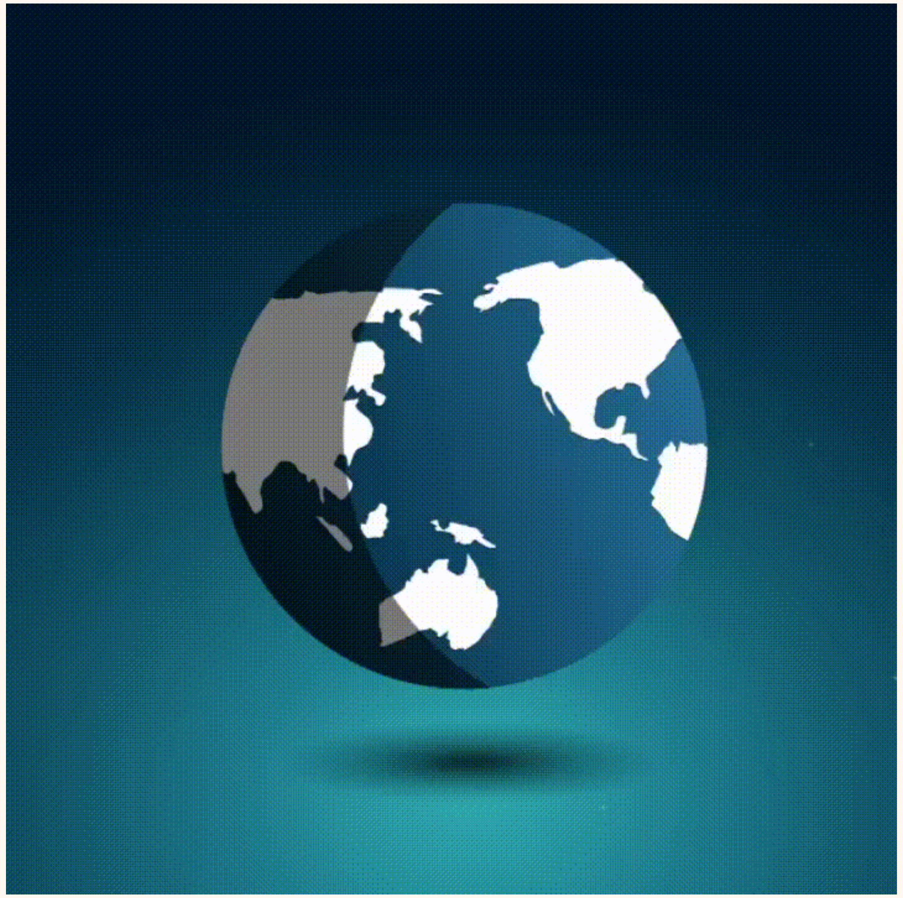
Rotation Matrix

The mathematical representation of 2D rotation is expressed using a rotation matrix. Given a point (x,y) and an angle θ/theta, the new coordinates (x′,y′) after rotation are calculated as follows:

$x' = x\cos(\theta) - y\sin(\theta)$

$y' = x\sin(\theta) + y\cos(\theta)$

This can be compactly written using matrix multiplication:



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Two Dimensional Transformation – rotate

Steps for Rotation

- **Determine the Rotation Angle (θ/theta)**: Specify the angle by which you want to rotate the shape.
- **Construct the Rotation Matrix**: Use the angle to form the rotation matrix.
- **Apply the Rotation Matrix**: Multiply the coordinates of each point in the shape by the rotation matrix to get the new coordinates.

Two Dimensional Transformation – rotate

Example

To rotate a point (1, 0) by 90 degrees counterclockwise:

1. **Angle (θ/theta)**: 90 degrees (π/2 radians).
2. **Rotation Matrix**:

$$\begin{bmatrix} \cos(90^\circ) & -\sin(90^\circ) \\ \sin(90^\circ) & \cos(90^\circ) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

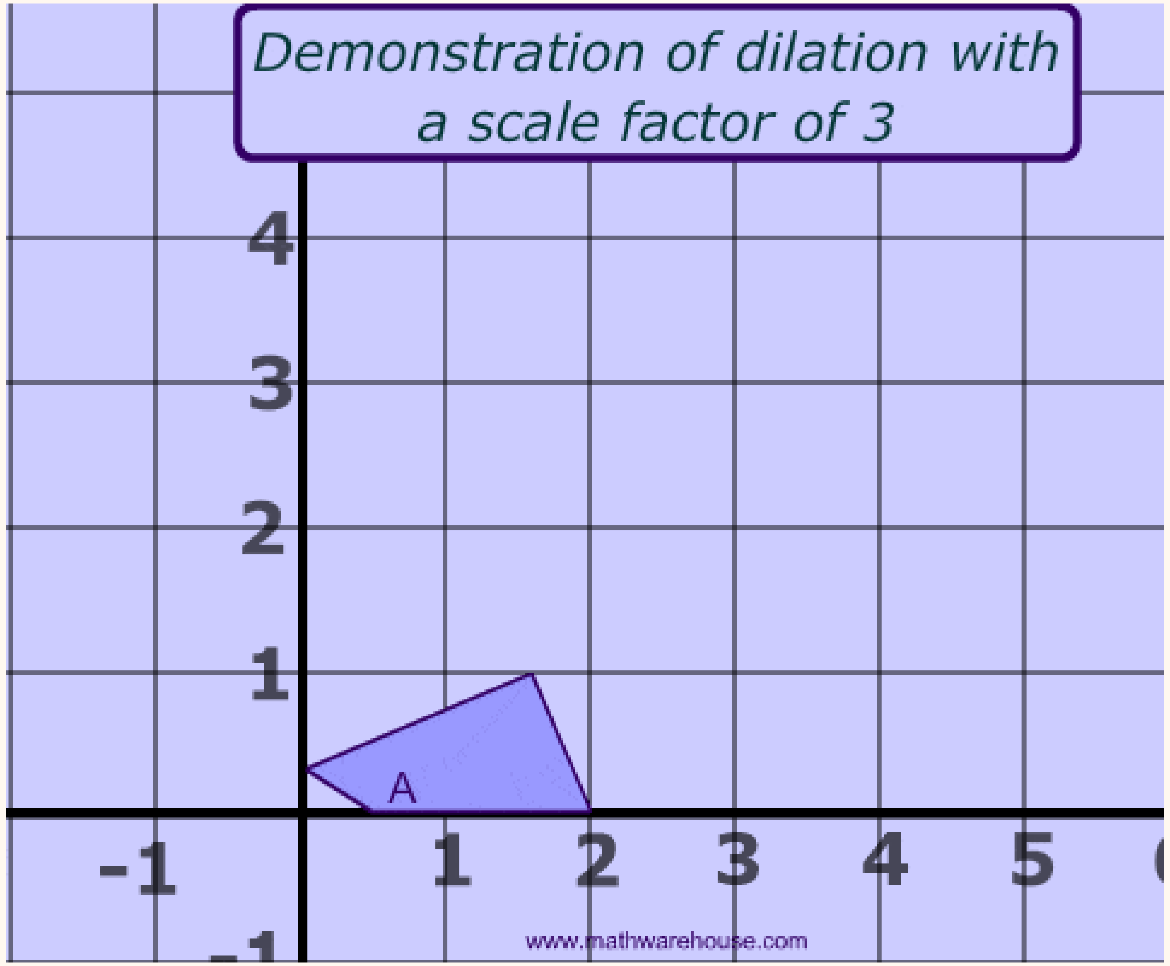
3. **New Coordinates**:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Thus, the point (1, 0) becomes (0, 1) after a 90-degree counterclockwise rotation.

Two Dimensional Transformation – scaling

Scaling is a fundamental geometric transformation in computer graphics that changes the size of an object in a two-dimensional plane. It can either enlarge or shrink the object along the x-axis, y-axis, or both.



Two Dimensional Transformation – scaling

Scaling Transformation Matrix

The scaling transformation in 2D can be represented using a scaling matrix S:

$$S = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$$

Where:

Sx is the scaling factor along the x-axis.

Sy is the scaling factor along the y-axis.

Two Dimensional Transformation – scaling

Applying the Scaling Transformation

Given a point P(x,y) in 2D space, applying the scaling transformation will give a new point P′(x′,y′) using the following equations:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} S_x \times x \\ S_y \times y \end{pmatrix}$$

So, the new coordinates after scaling will be:
$x' = S_x \times x$
$y' = S_y \times y$

Example:
Consider a point P(2,3) and scaling factors Sx = 2 and Sy =3:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \times 2 \\ 3 \times 3 \end{pmatrix} = \begin{pmatrix} 4 \\ 9 \end{pmatrix}$$

So, the new point P′(4,9) is the scaled version of P(2,3).

Two Dimensional Transformation – Matrix representations

1. ## Matrix Representation

   2D transformations can be represented using 2x2 or 3x3 matrices, which encode the scaling, rotation, and translation operations. These matrices can be multiplied with the coordinates of points to apply the desired transformation.

2. ## Advantages

   Matrix representations are powerful because they allow multiple transformations to be combined through matrix multiplication, enabling complex composite transformations. They also provide a compact, efficient way to store and apply transformations in computer graphics and visualization algorithms.

3. ## Homogeneous Coordinates

   Homogeneous coordinates are often used in computer graphics to simplify the mathematics of transformations, such as translation, rotation, and scaling, by incorporating them into a single matrix operation.

Homogenous Coordinates Other transformations

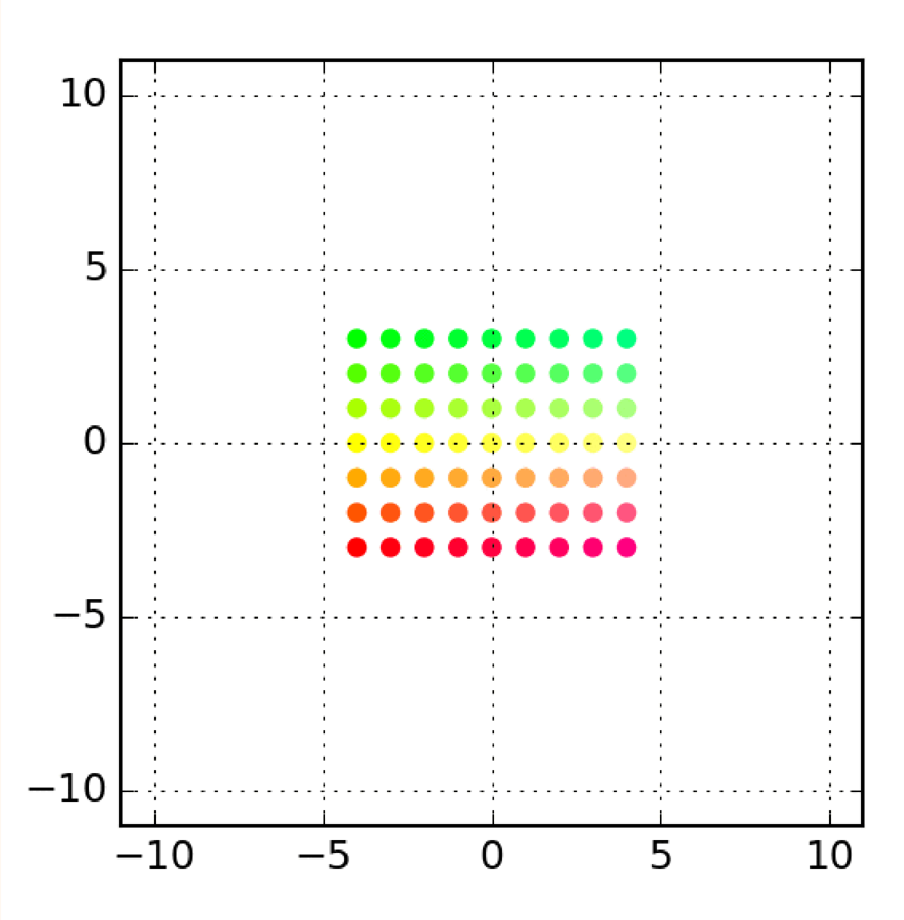
Other Transformations

Homogeneous coordinates also allow for representing 2D transformations like shearing and reflection, which are useful for mirroring, or applying complex deformations to visual elements.

Efficiency and Flexibility

The use of homogeneous coordinates and 3x3 transformation matrices simplifies the implementation of 2D transformations, making it easier to combine and apply multiple transformations in sequence. This flexibility and efficiency is crucial for real-time graphics, animation, and interactive data visualization applications.

Homogenous Coordinates Other transformations - Shear



1

Shear Transformation

Shear is a 2D transformation that slants an object by shifting its corners along one axis while keeping the other axis unchanged. It can be used to create depth, skew visuals, or apply special effects to images.

2

Matrix Representation

Shear is represented by a 3x3 homogeneous transformation matrix with 1s on the diagonal and non-zero values in the off-diagonal positions. The amount of shear is determined by the values in these off-diagonal elements.

Homogenous Coordinates Other transformations - Shear

1. Horizontal Shear (Shear along the X-axis):

A horizontal shear transformation shifts the y-coordinates of points horizontally. The matrix for a horizontal shear by a factor of sh_x is:

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If a point (x, y) is transformed using this matrix, the new coordinates (x', y') are given by:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + sh_x \cdot y \\ y \\ 1 \end{bmatrix}$$

Homogenous Coordinates Other transformations - Shear

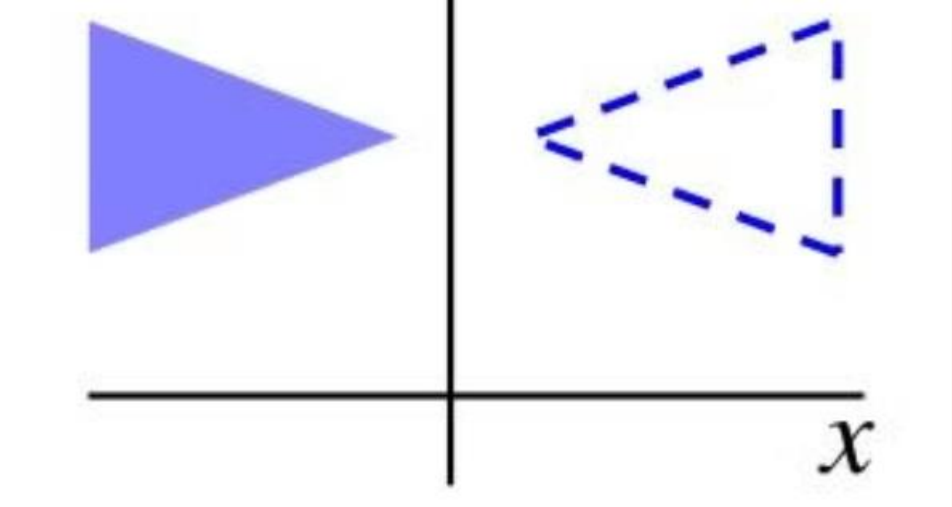
2. Vertical Shear (Shear along the Y-axis):

A vertical shear transformation shifts the x-coordinates of points vertically. The matrix for a vertical shear by a factor of sh_y is:

$$\begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If a point (x, y) is transformed using this matrix, the new coordinates (x', y') are given by:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y + sh_y \cdot x \\ 1 \end{bmatrix}$$



Homogenous Coordinates Other transformations - reflection

Reflection

Reflection is a 2D transformation that flips or mirrors an object along a specified axis. This can be useful for creating symmetrical designs, mirroring user interface elements, or visualizing data from different perspectives.

Axis of Reflection

The axis of reflection can be either the x-axis, y-axis, or a line with a specific slope. The transformation matrix for reflection depends on the chosen axis.

Applications

Reflection transformations are commonly used in graphic design, data visualization, and user interface design to create symmetrical layouts, mirror interface elements, or visualize data from multiple viewpoints.