Introduction to Transformations

Graphics 1 CMP-5010B

Dr. David Greenwood

david.greenwood@uea.ac.uk

SCI 2.16a University of East Anglia

February 15, 2022

Content

- What is a transformation?
- Types of transformations
- Translation
- Rotation

What is a transformation

. . . in computer *graphics*?

Transformations

. . . in 2D Computer Graphics

- Two spatial dimensions
- Planar world or the "plane"
- Usually represented by Cartesian **coordinates**
- x and y for objects
- s and t for textures
- u and v for images

Transformations

Geometric transformations will map points in one *space* to points in another space:

$$(x', y') = f(x, y)$$

Transformations

The mapping function uses elementary operations, which include:

- Translation
- Rotation
- Scaling
- Shear
- Reflection
- Projection
- Warp

Transformations

Types of transformation preserve **geometric properties** of the object.

Rigid transformations

- *Translation* and *Rotation*
- preserves the Euclidean distance between every pair of points
- preserves “handedness” of the object

Euclidean transformations

- Translation, Rotation and *Reflection*
- Also known as **Isometries**
- preserves the Euclidean distance between every pair of points

Similarity transformations

- Translation, Rotation, Reflection and *Uniform Scaling*
- preserves the shapes of the objects
- Examples of similar shapes include all squares, all circles, but not all triangles.

Affine transformations

- Translation, Rotation, Reflection, Scaling and *Shear*
- Scaling can be uniform or non-uniform
- preserves *lines* and *parallelism* of objects

Projective transformations

- *Projection* from N dimensions to a lower dimension
- useful in 3D graphics but not in 2D
- *Perspective* or *Orthographic* projection

Non-linear transformations

- *Warp*: non-linearly deform the object.
- Example: for images we may talk about lens *distortion*.

Object Representation

- How do we represent objects in computer graphics?

Object Representation

In graphics, we represent objects using points or *vertices*, which are connected to form polygons or *faces*.

Object Representation

Only the *vertices* are subjected to the transformations.

Object Representation

Question: How do we represent a vertex mathematically?

- A column **vector** of the vertex coordinates.

Tools for transformations

Transformations of an object are applied to **each** individual vertex of that object.

Tools for transformations

The mathematical entity used to perform a transformation to the vector of n vertex coordinates is:

A *square matrix* of size $n \times n$, where n is the dimension of the vertex vector.

2D Transformations

- Assume a 2D *plane* with coordinates x and y.
- Polygonal object is a triangle with 3 *vertices*.
- All vertices are subject to transformations we apply.

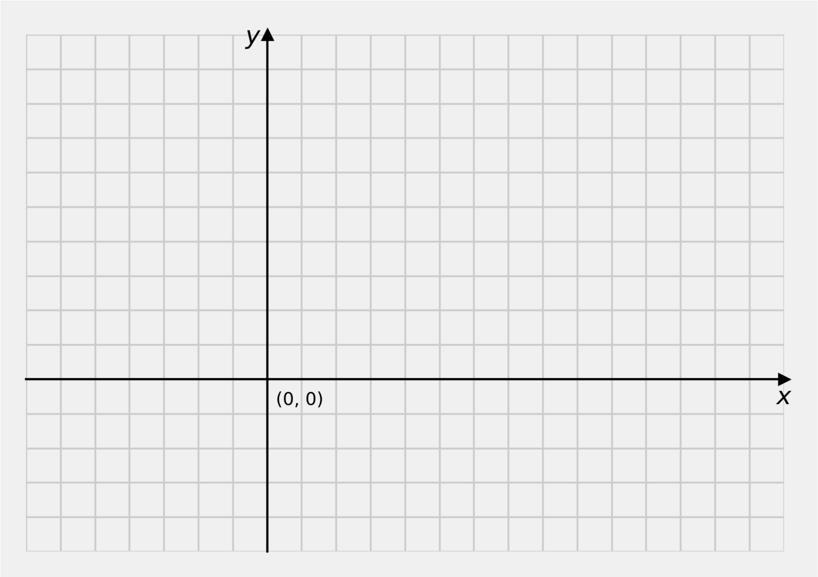

Figure 1: The coordinate system

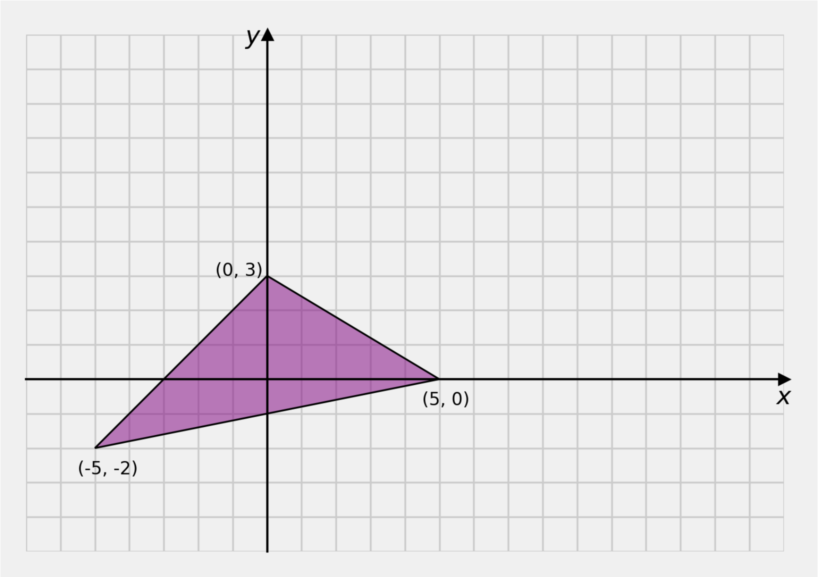

Figure 2: A model in the plane.

Translation

Formally, we will represent vertex coordinates as a column vector:

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

Translation

Translation is performed by **adding** a *translation vector*.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Translation

Example: Consider the vertex with coordinates $(-5, -2)$, that we wish to translate in the x direction 9 units, and in the y direction 5 units.

$$\begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} -5 \\ -2 \end{bmatrix} + \begin{bmatrix} 9 \\ 5 \end{bmatrix}$$

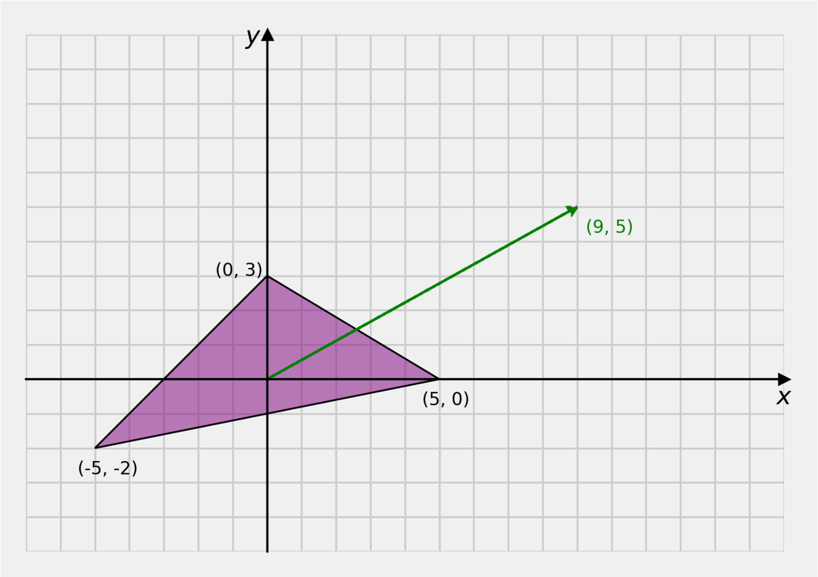

Figure 3: Translation as a vector.

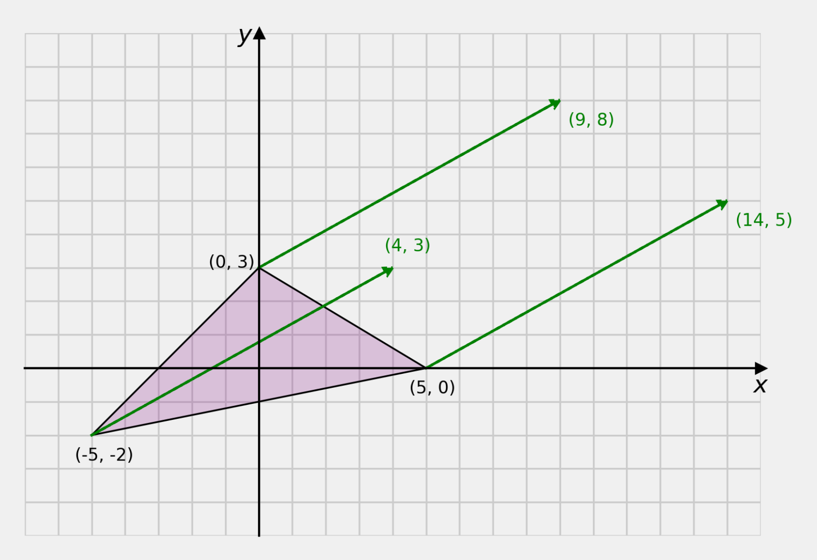

Figure 4: Add translation vector to each vertex.

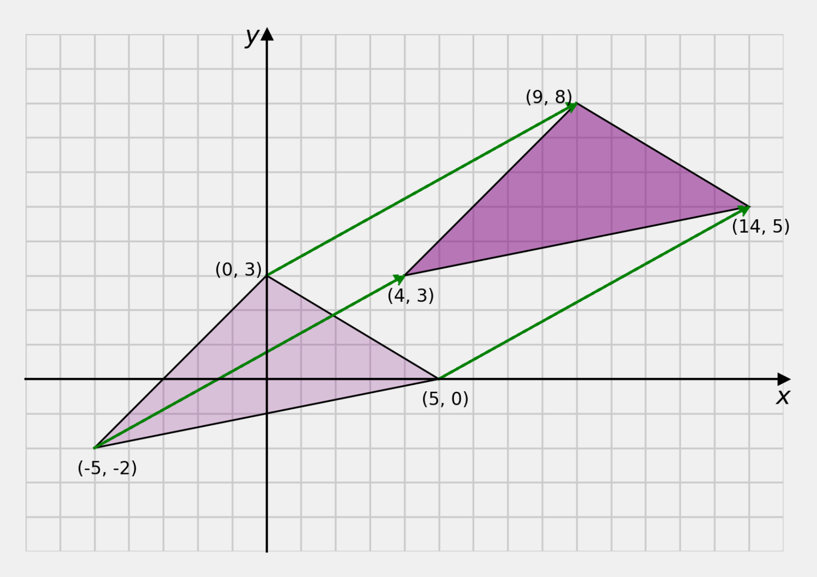

Figure 5: All vertices are translated.

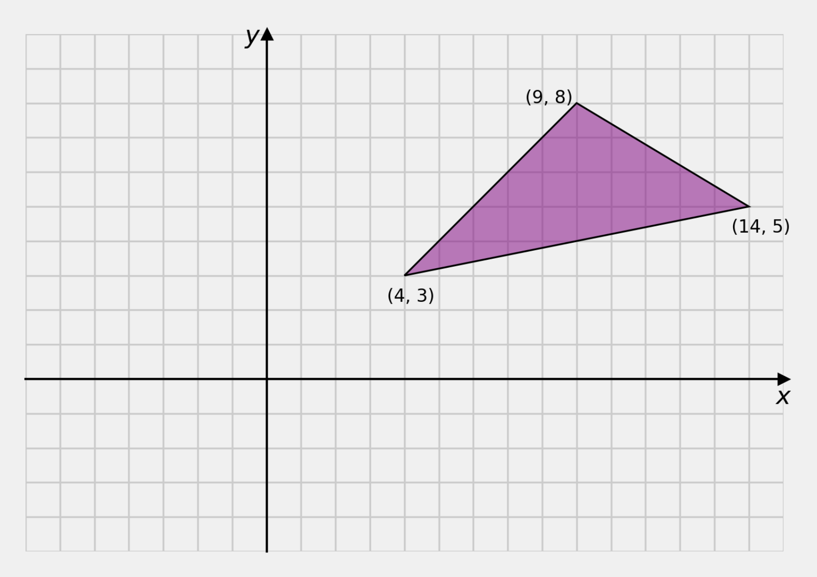

Figure 6: The model is in a new position.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Rotation

We stated earlier, that the mathematical entity used to perform a transformations to the vector coordinates is a *matrix*.

How do we use a matrix to perform a transformation?

Rotation

- translation moves a single point
- rotation of a point is meaningless
- we need to perform rotations about an axis

Rotation

A mnemonic for rotation is *SOH*, *CAH*, *TOA*.

$$sin(\theta) = \frac{O}{H},\ cos(\theta) = \frac{A}{H},\ tan(\theta) = \frac{O}{A}$$

Rotation

We can fit this mnemonic to our 2D plane:

$$sin(\theta) = \frac{O}{H} = \frac{y}{r},\ cos(\theta) = \frac{A}{H} = \frac{x}{r},\ tan(\theta) = \frac{O}{A} = \frac{y}{x}$$

where r is the radius of a circle.

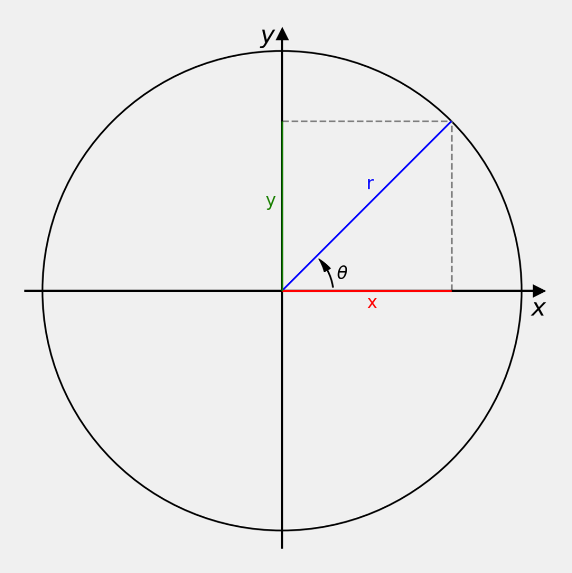

Figure 7: rotation in unit circle

Matrix multiplication

Matrix multiplication is performed row by column:

$$\begin{bmatrix} ax + by \\ cx + dy \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}$$

- Number of columns in the first operand **must** equal the number of rows in the second operand.

Rotation Matrix

Deriving the rotation matrix using trigonometric identities.

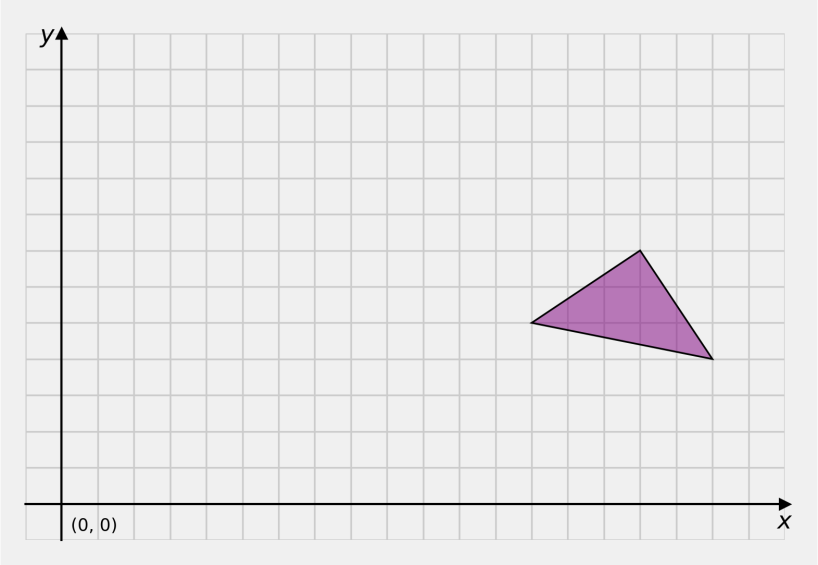

Figure 8: A model in the plane

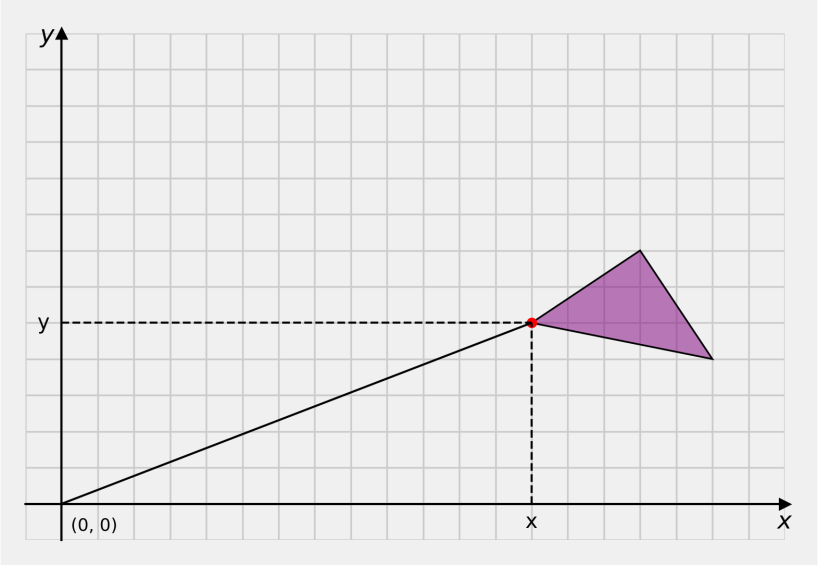

Figure 9: consider one vertex

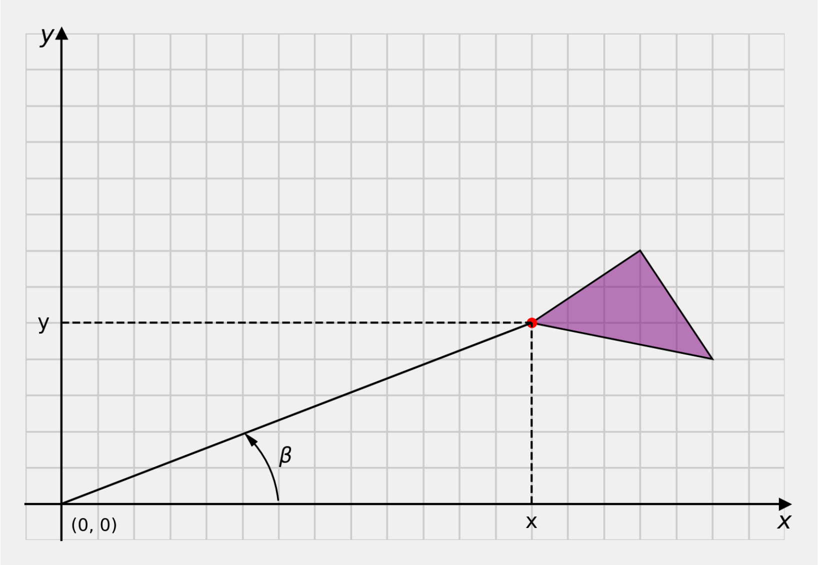

Figure 10: angle between the *x* axis

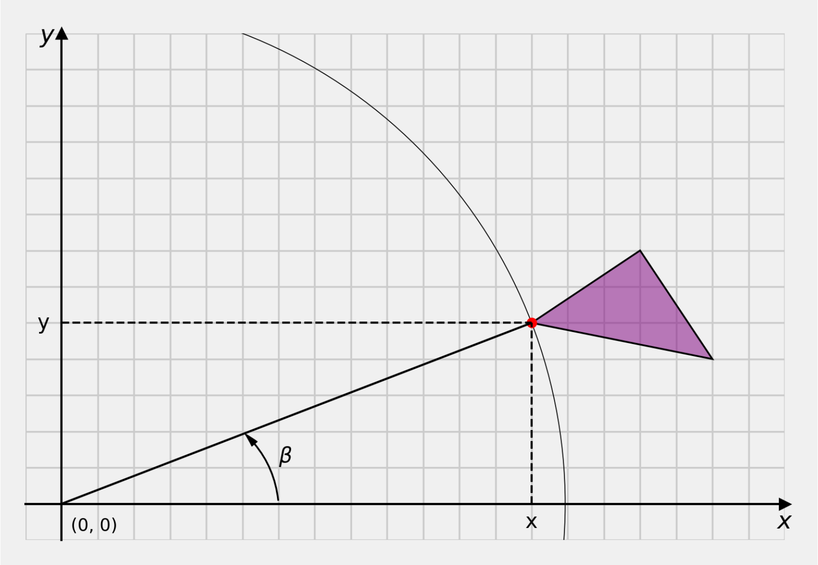

Figure 11: rotation about the origin

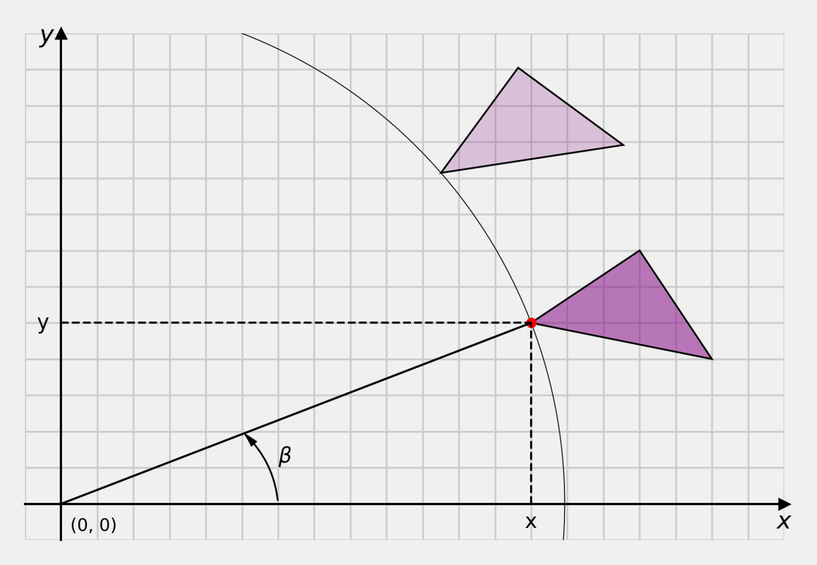

Figure 12: a second rotation

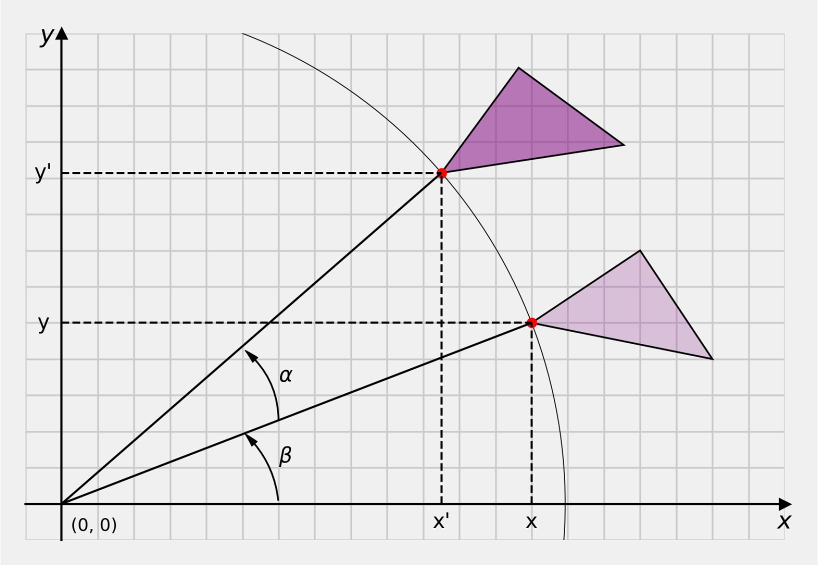

Figure 13: sum of two angles

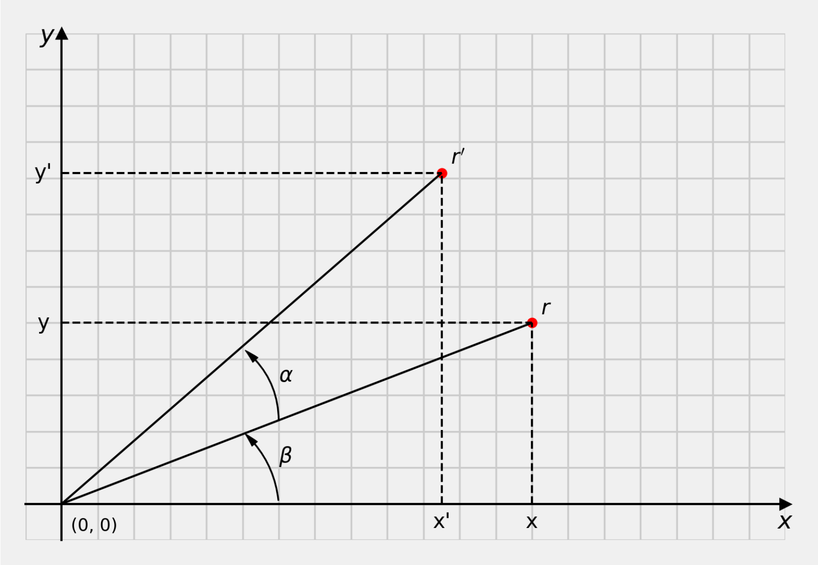

Figure 14: r and r'

$$r = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix}$$

$$r' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \end{bmatrix}$$

Ptolomy's identity

The sum of two angles:

$$\cos(\alpha + \beta) = \cos\alpha\ \cos\beta - \sin\alpha\ \sin\beta$$
$$\sin(\alpha + \beta) = \sin\alpha\ \cos\beta + \cos\alpha\ \sin\beta$$

Rotation Matrix Derivation

using the identities:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha \ \cos\beta - \sin\alpha \ \sin\beta \\ \sin\alpha \ \cos\beta + \cos\alpha \ \sin\beta \end{bmatrix}$$

Rotation Matrix Derivation

recall:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix}$$

substitute x and y:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x\cos\alpha - y\sin\alpha \\ x\sin\alpha + y\cos\alpha \end{bmatrix}$$

Rotation Matrix Derivation

as a matrix multiplication:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

Summary

- Types of transformations
- Translation
- Rotation

Reading:

- Hearn, D. et al. (2004). Computer Graphics with OpenGL.