Computer Vision

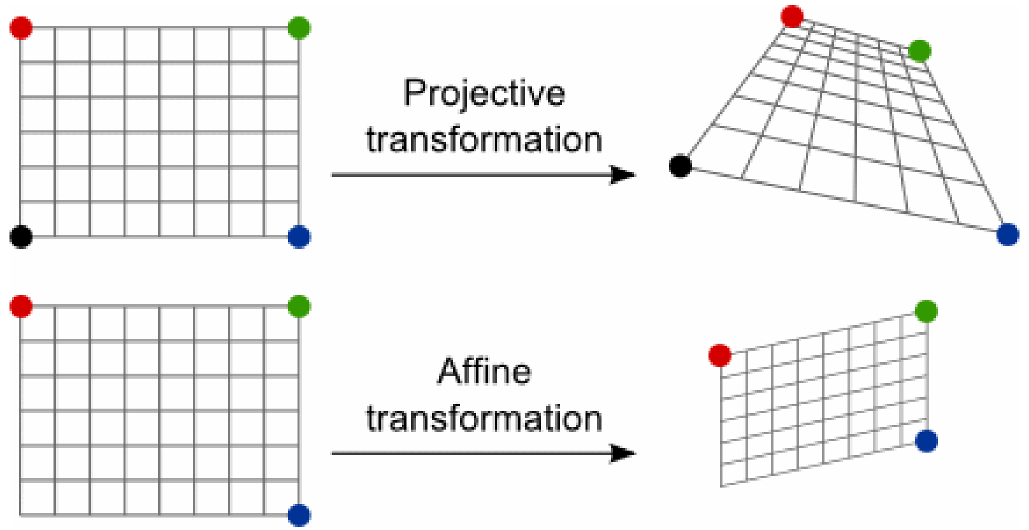
(Course Code: 4047)

Module-2:Lecture-8: Affine Transformations

Gundimeda Venugopal, Professor of Practice, SCOPE

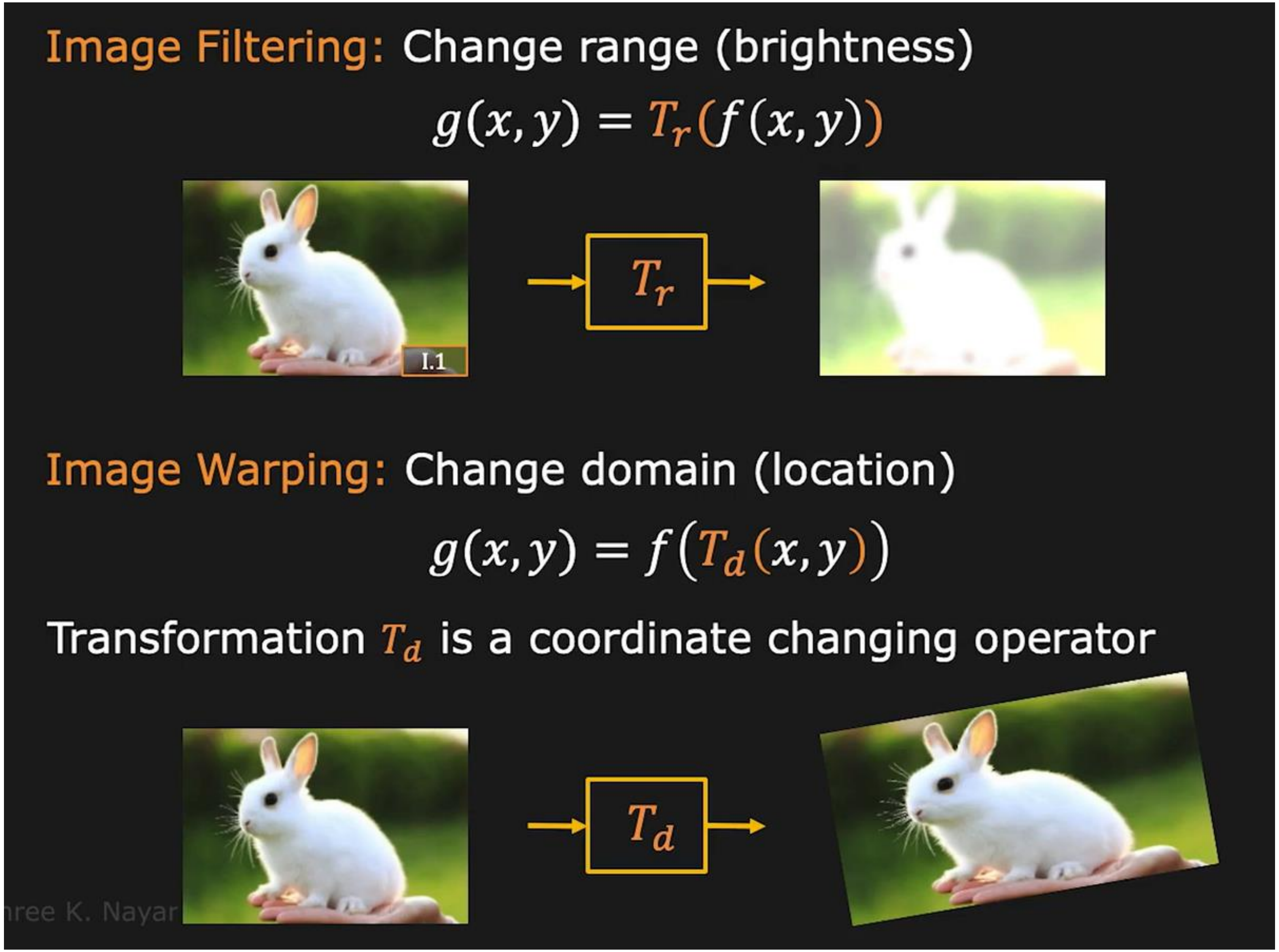
Affine and Projective Transformations

- An affine transformation is any transformation that preserves collinearity, parallelism as well as the ratio of distances between the points (e.g. midpoint of a line remains the midpoint after transformation).
  - All the geometric transformations such as translation, rotation, scaling, etc are all affine transformations as all the above properties are preserved in these transformations.
  - In simple terms, one can think of the affine transformation as a composition of rotation, translation, scaling, and shear.
  - Affine transformations preserve parallelism
  - It doesn't necessarily preserve distances and angles.

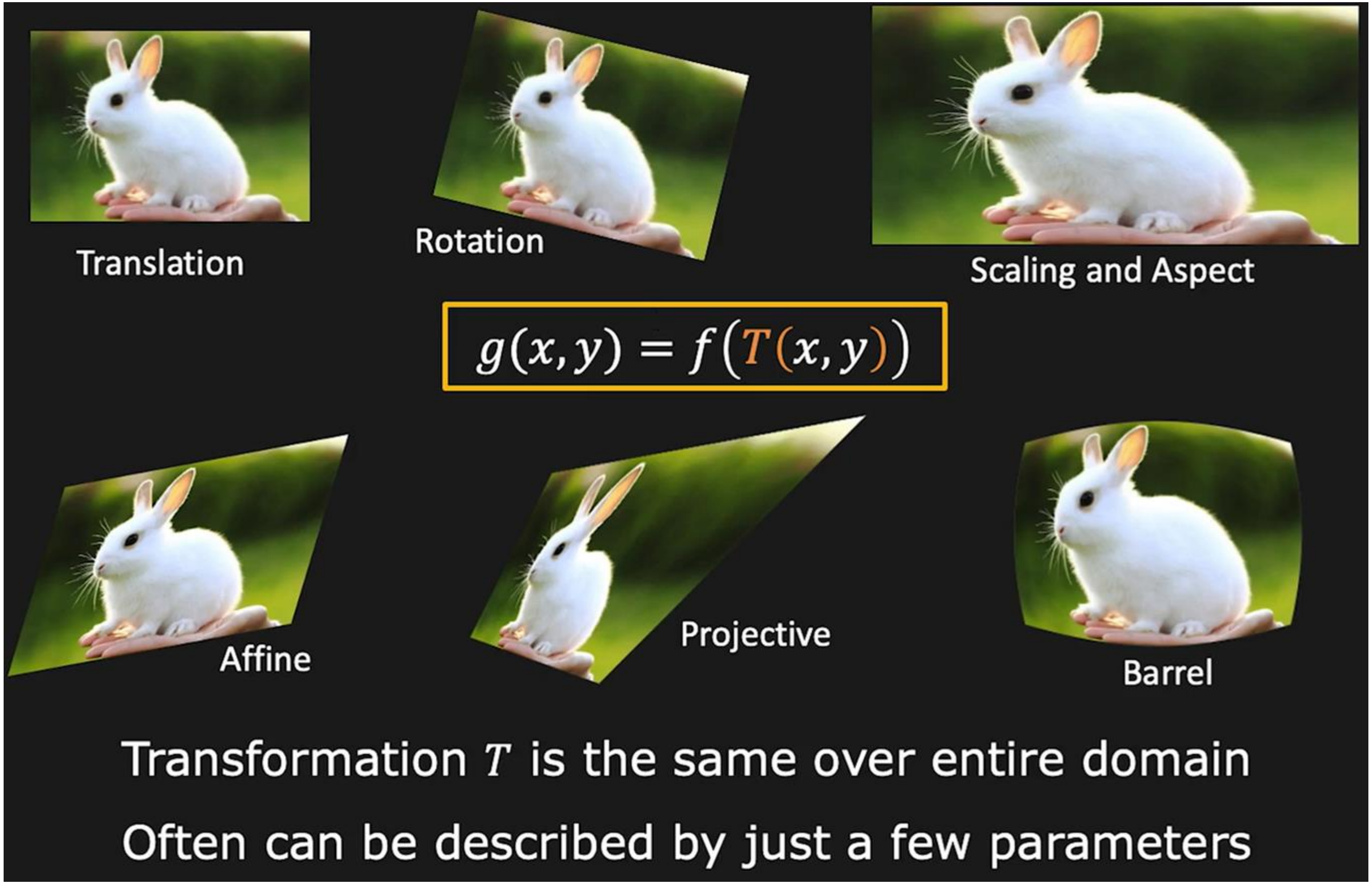


- A projective transformation shows how the perceived objects change as the observer's viewpoint changes.
  - These transformations allow the creating of perspective distortion.
  - Projective transformations do not preserve parallelism, length, and angle.
  - A projective transformation can be represented as the transformation of an arbitrary quadrangle (that is a system of four points) into another one.

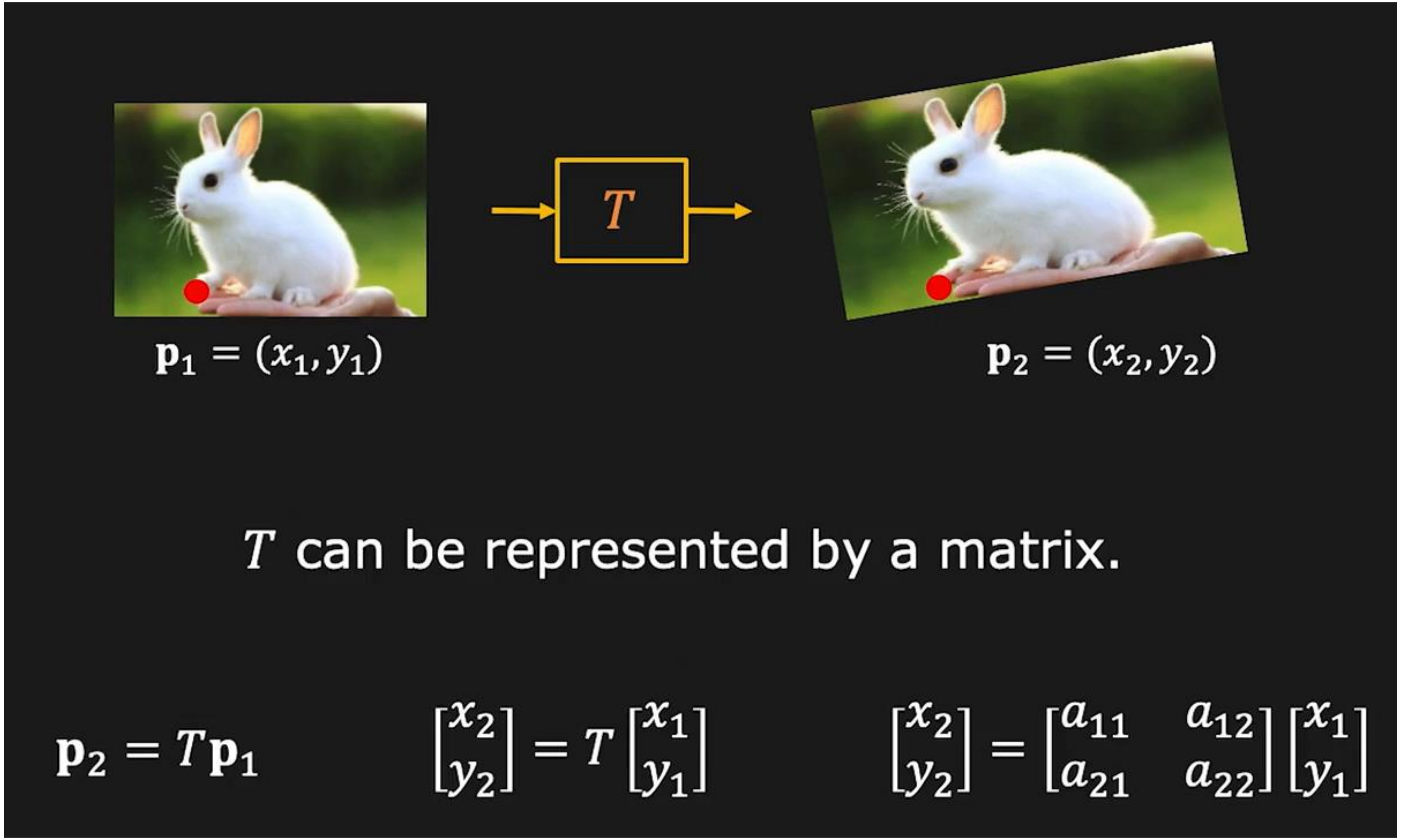
Image Manipulation



Global Warping/Transformation



2x2 Linear Transformations



Linear Transformations

- A linear transformation is a function that maps one vector space into another and these transformations are often implemented by a matrix.
- A transformation is considered to be linear if it preserves vector addition and scalar multiplication.

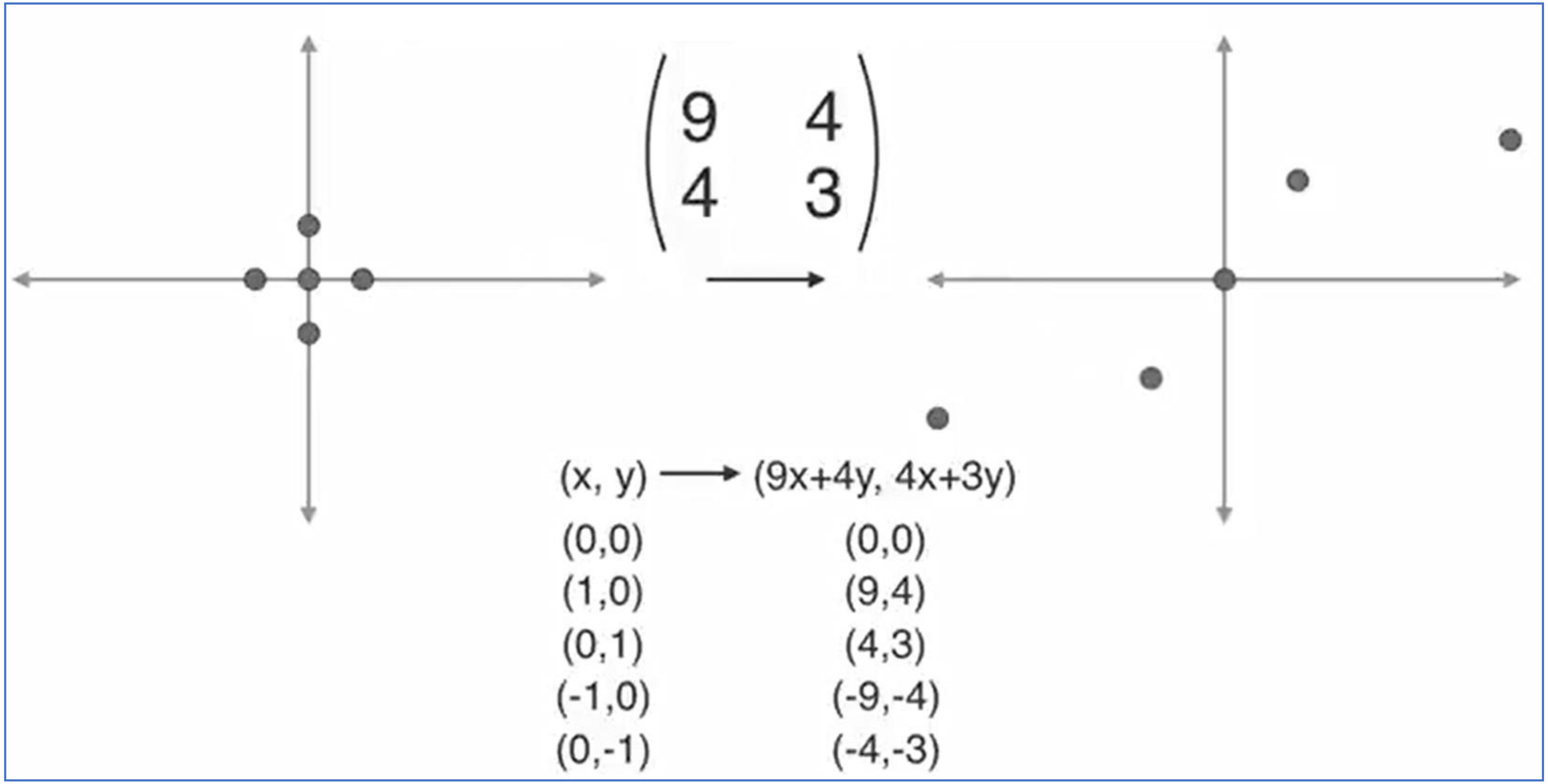
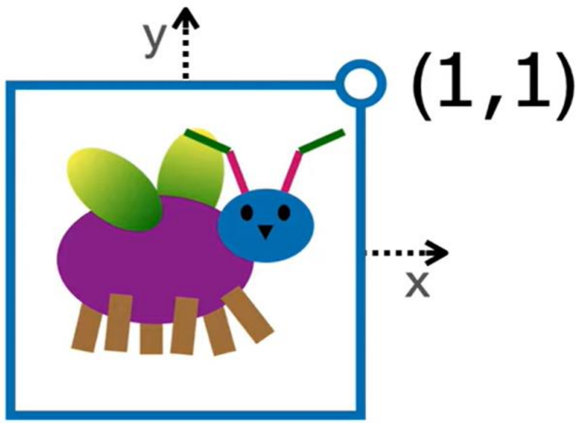


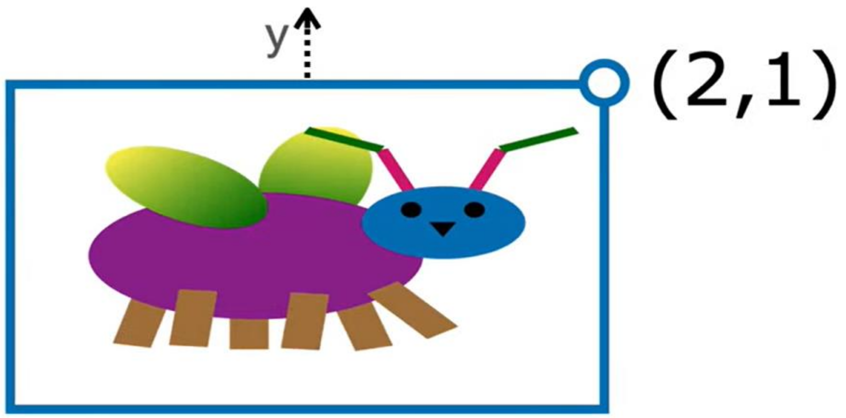
Image Transformation Example

The following matrix helps the scale the image in X-direction

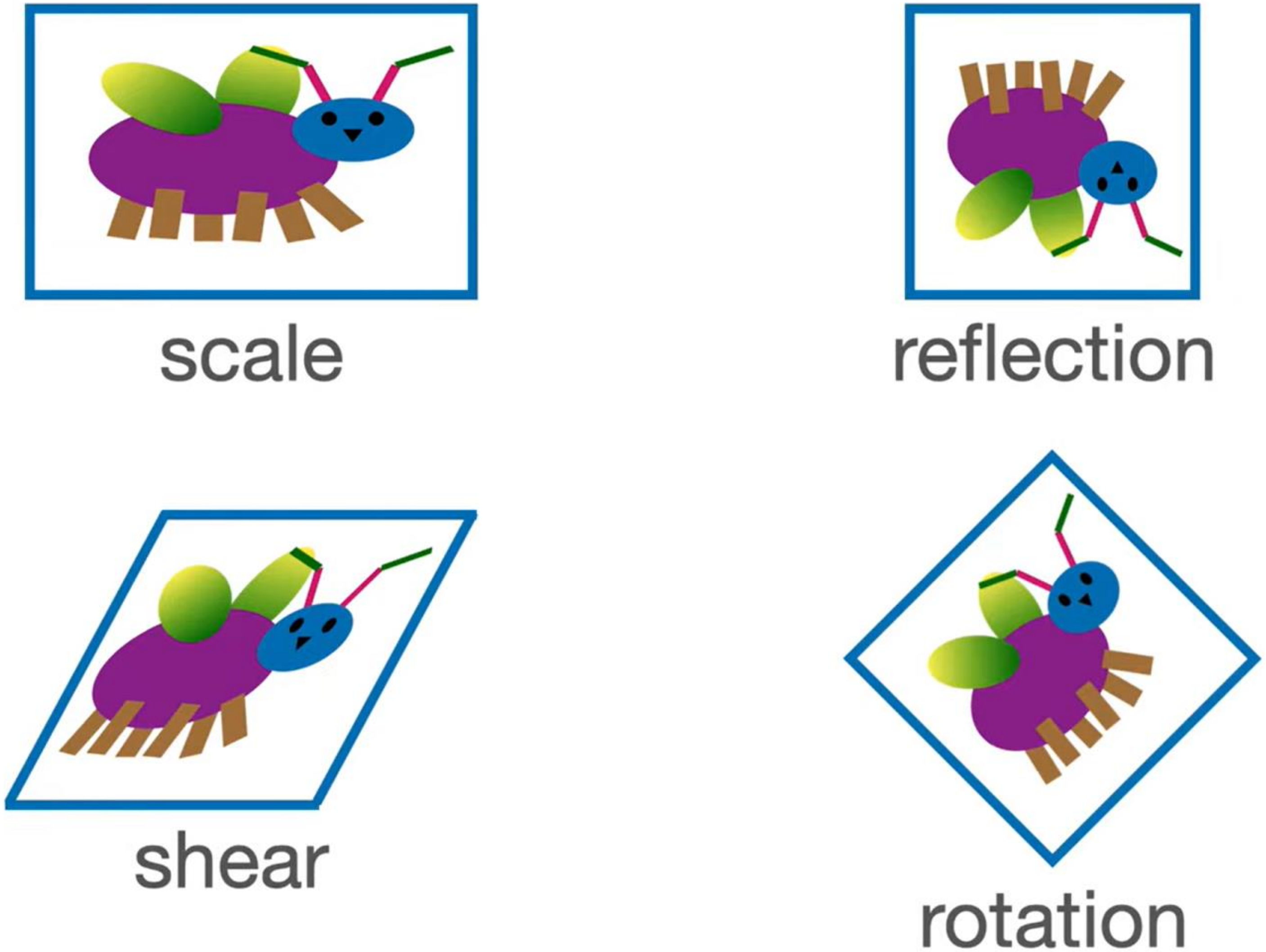
$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$



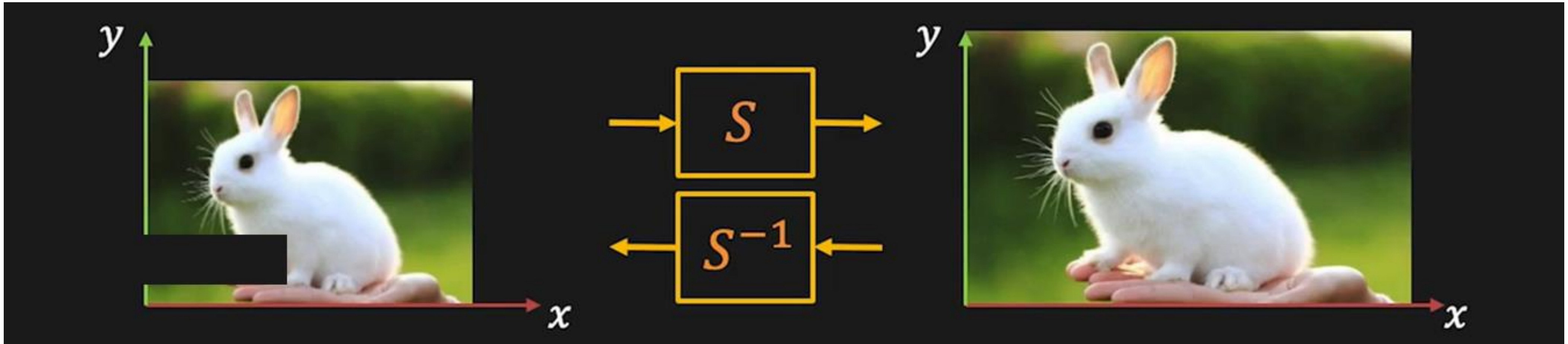
$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$



Affine Transformations



Scaling (Stretching or Squishing)



Forward:

$$x_2 = ax_1 \qquad y_2 = by_1$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

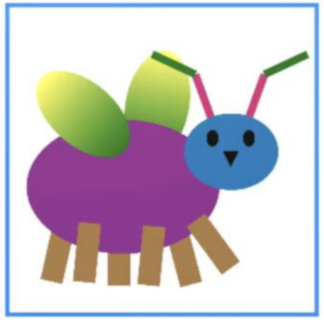
Inverse:

$$x_1 = \frac{1}{a} x_2 \qquad y_1 = \frac{1}{b} y_2$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = S^{-1} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1/a & 0 \\ 0 & 1/b \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

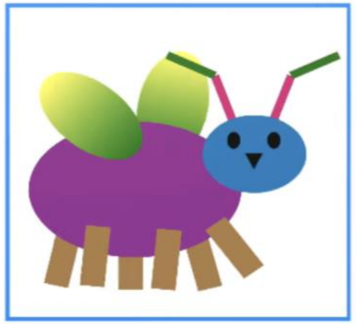
Horizontal and Vertical Scaling

$$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$



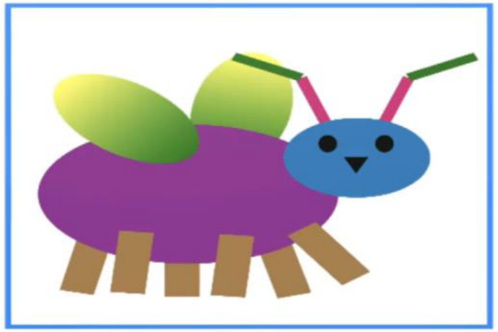
No Scaling

$$\begin{bmatrix} \boxed{1.1} & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$



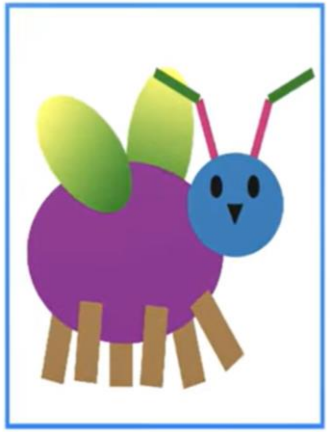
Horizontal Scaling = 1.1x

$$\begin{bmatrix} \boxed{1.5} & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$



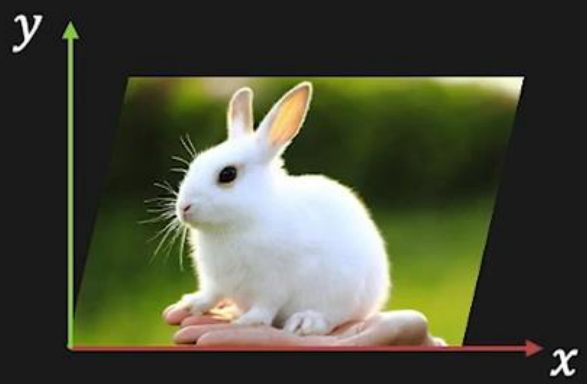
Horizontal Scaling = 1.5x

$$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & \boxed{1.3} \end{bmatrix}$$



Vertical Scaling = 1.3x

Skew



Horizontal Skew:

$$x_2 = x_1 + m_x y_1$$

$$y_2 = y_1$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S_x \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & m_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Vertical Skew:

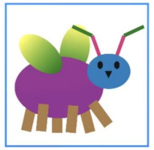
$$x_2 = x_1$$

$$y_2 = m_y x_1 + y_1$$

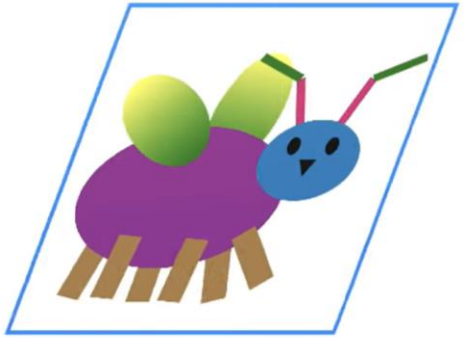
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S_x \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ m_y & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Shear

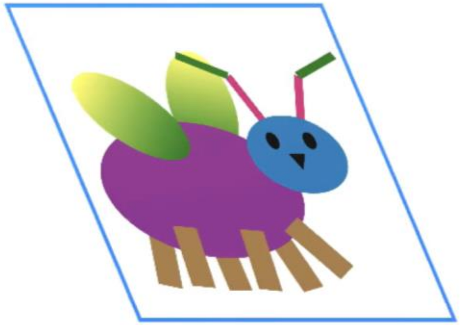
$$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$



$$\begin{bmatrix} 1.0 & \boxed{0.4} \\ 0.0 & 1.0 \end{bmatrix}$$



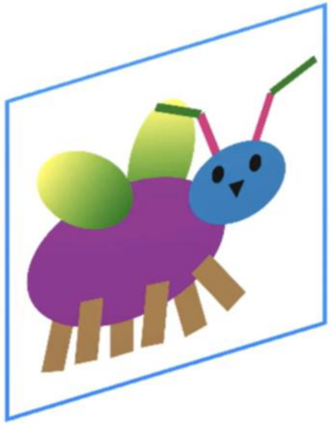
$$\begin{bmatrix} 1.0 & \boxed{-0.4} \\ 0.0 & 1.0 \end{bmatrix}$$



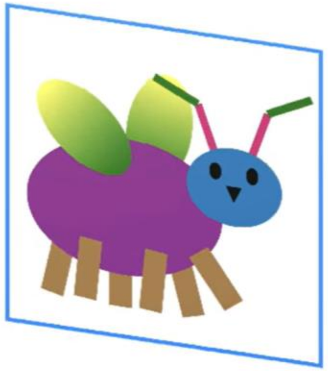
Horizontal Shear = 0.4

Horizontal Shear =- 0.4

$$\begin{bmatrix} 1.0 & 0.0 \\ \boxed{0.3} & 1.0 \end{bmatrix}$$



$$\begin{bmatrix} 1.0 & 0.0 \\ \boxed{-0.1} & 1.0 \end{bmatrix}$$

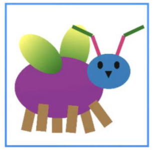


Vertical Shear = 0.3

Vertical Shear = -0.1

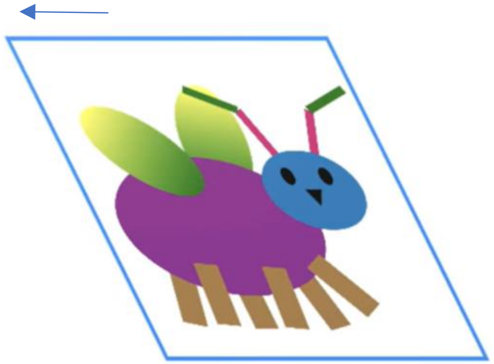
Rotation

$$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$



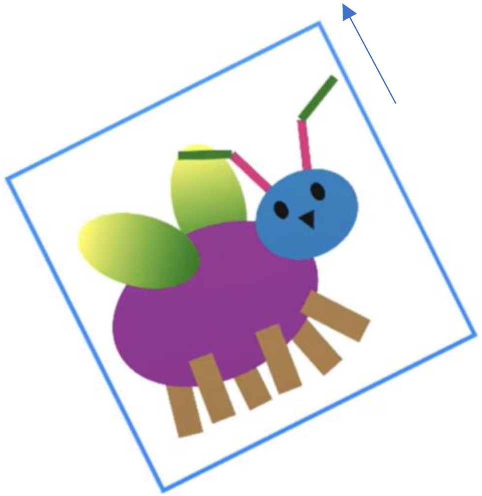
- As two shear operations (← + ↖)

$$\begin{bmatrix} 1.0 & \boxed{-0.5} \\ 0.0 & 1.0 \end{bmatrix}$$



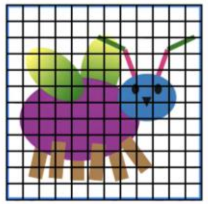
+

$$\begin{bmatrix} 1.0 & -0.5 \\ \boxed{0.5} & 1.0 \end{bmatrix}$$



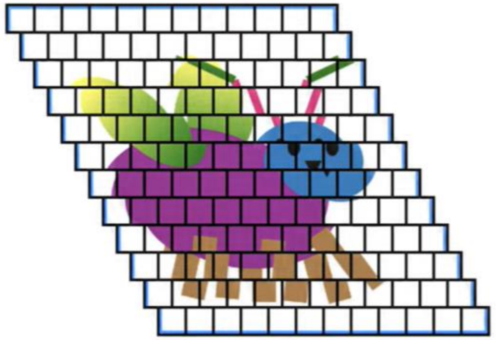
Rotation: 2d rotation with 1d translation

$$\begin{bmatrix} 1.0 & -0.5 \\ 0.0 & 1.0 \end{bmatrix}$$

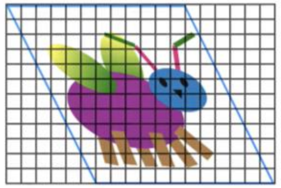


Translate each row by different amount

$$\begin{bmatrix} 1.0 & -0.5 \\ 0.0 & 1.0 \end{bmatrix}$$

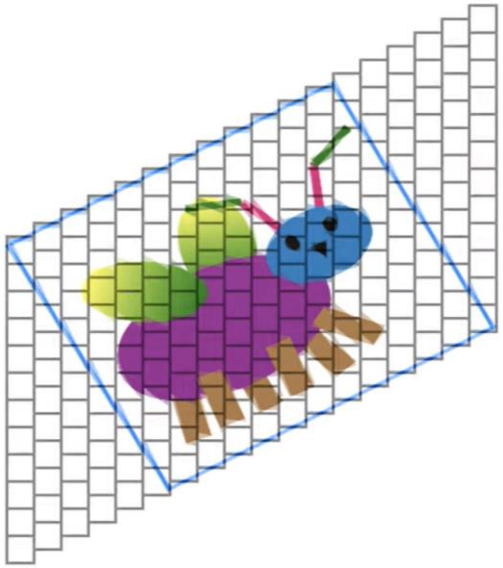


$$\begin{bmatrix} 1.0 & 0.0 \\ 0.5 & 1.0 \end{bmatrix}$$

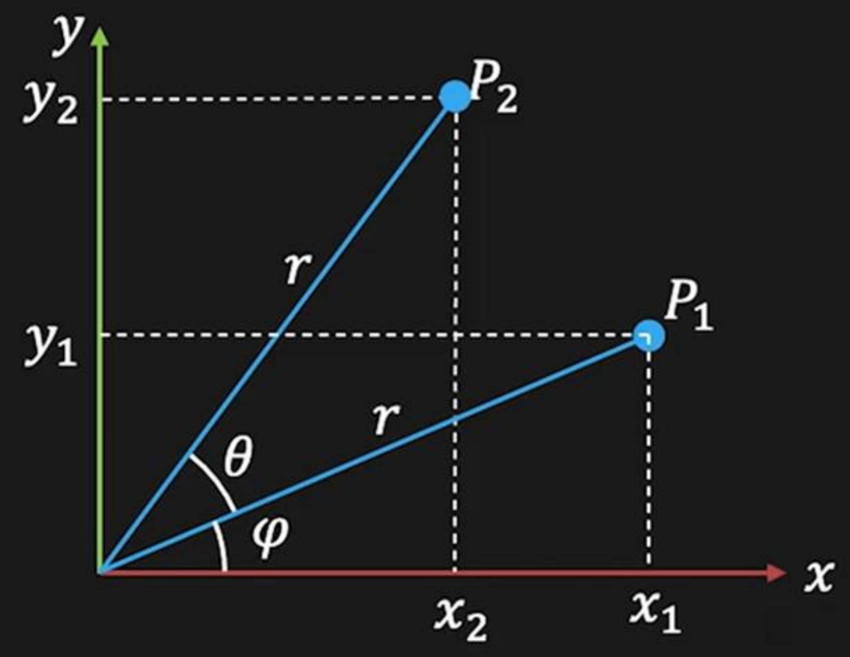


Then by Translate each column by different amount

$$\begin{bmatrix} 1.0 & 0.0 \\ 0.5 & 1.0 \end{bmatrix}$$



2D Rotation



$$x_1 = r\cos(\varphi)$$

$$y_1 = r\sin(\varphi)$$

$$x_2 = r\cos(\varphi + \theta)$$

$$x_2 = r\cos\varphi\cos\theta - r\sin\varphi\sin\theta$$

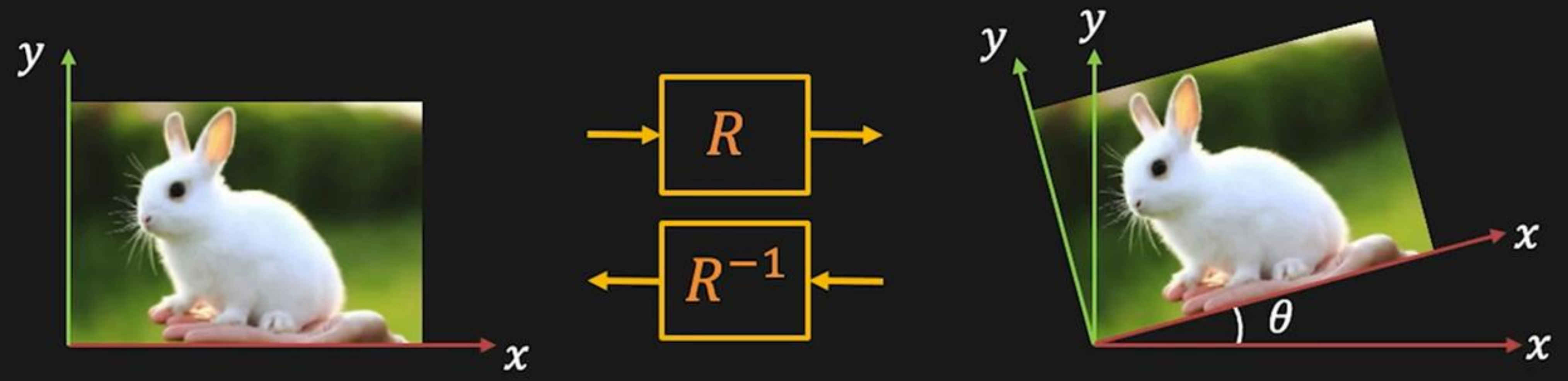
$$x_2 = x_1\cos\theta - y_1\sin\theta$$

$$y_2 = r\sin(\varphi + \theta)$$

$$y_2 = r\cos\varphi\sin\theta + r\sin\varphi\cos\theta$$

$$y_2 = x_1\sin\theta + y_1\cos\theta$$

2D Rotation



Forward:

$$x_2 = x_1 cos\theta - y_1 sin\theta$$

$$y_2 = x_1 sin\theta + y_1 cos\theta$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = R \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Inverse:

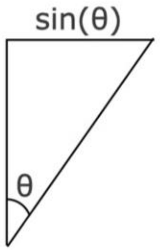
$$x_1 = x_2 cos\theta + y_2 sin\theta$$

$$y_1 = -x_2 sin\theta + y_2 cos\theta$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = R^{-1} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

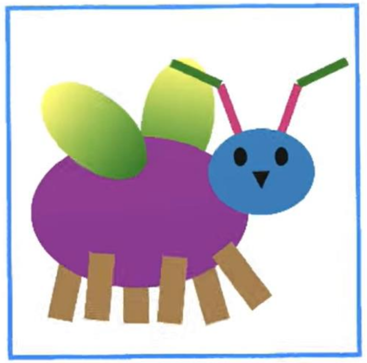
Rotation using Sine and Cosine

$$\begin{bmatrix} 1.0 & -a \\ a & 1.0 \end{bmatrix}$$



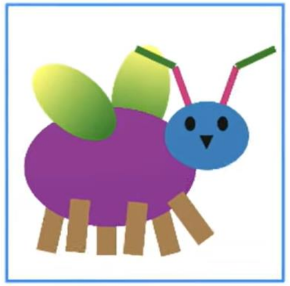
$$\begin{bmatrix} 1.0 & -\sin(\theta) \\ \sin(\theta) & 1.0 \end{bmatrix}$$

$$\begin{bmatrix} 1.0 & -\sin(0.0\pi) \\ \sin(0.0\pi) & 1.0 \end{bmatrix}$$



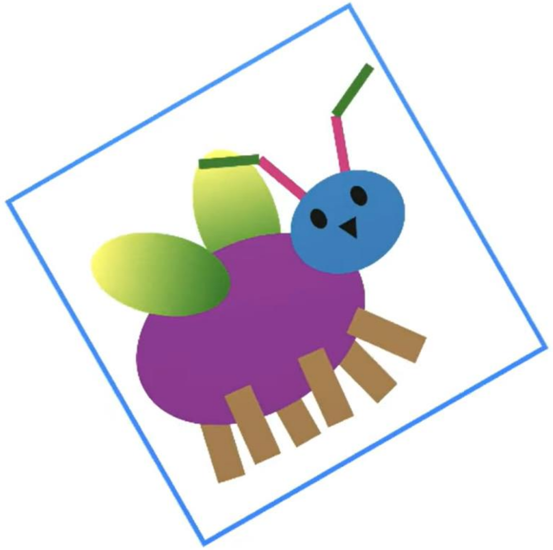
Original Image

$$\begin{bmatrix} \cos(0.2\pi) & 0.0 \\ 0.0 & \cos(0.2\pi) \end{bmatrix}$$



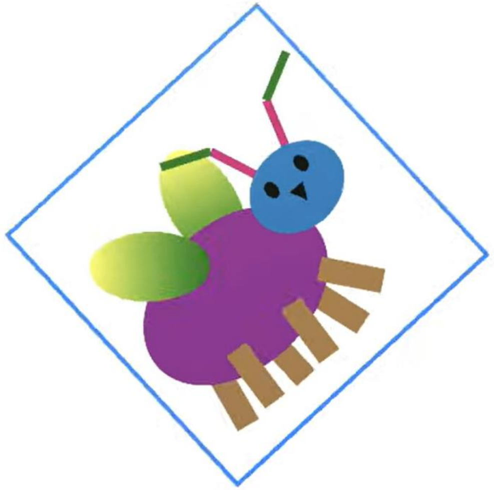
Scales by a factor 1/x

$$\begin{bmatrix} 1.0 & -\sin(0.2\pi) \\ \sin(0.2\pi) & 1.0 \end{bmatrix}$$



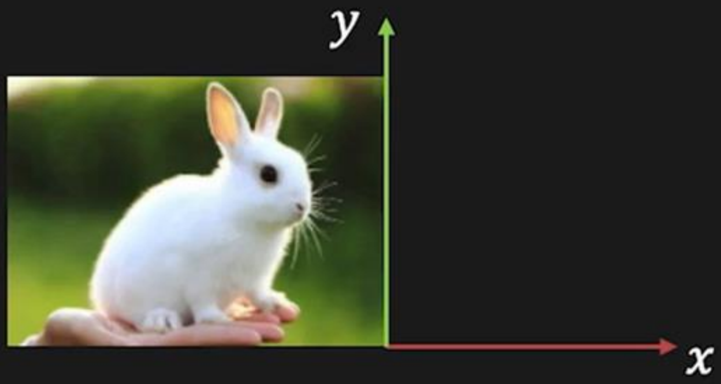
Rotates by 0.2Π + Scales by a factor x

$$\begin{bmatrix} \cos(0.2\pi) & -\sin(0.2\pi) \\ \sin(0.2\pi) & \cos(0.2\pi) \end{bmatrix}$$



Rotates by 0.2Π

Mirror



Mirror about Y-axis:

$$x_2 = -x_1$$

$$y_2 = y_1$$

$$M_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Mirror about line $y = x$:

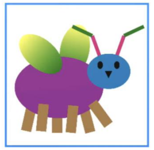
$$x_2 = y_1$$

$$y_2 = x_1$$

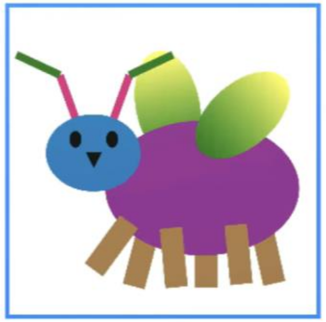
$$M_{xy} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Reflection

$$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$



$$\begin{bmatrix} \boxed{-1.0} & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

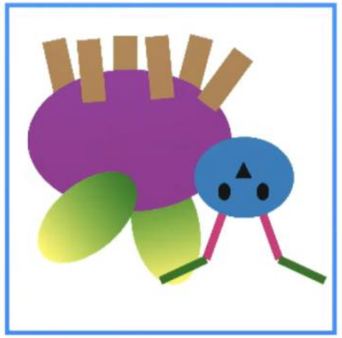


Reflection (around y axis)

$$\begin{bmatrix} \boxed{-0.5} & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

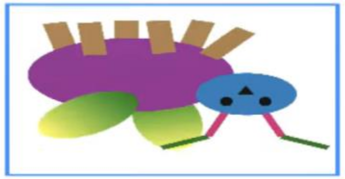
Reflection(around y-axis) + [Horizontal Scaling = 0.5x]

$$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & \boxed{-1.0} \end{bmatrix}$$



Reflection (around x axis)

$$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & \boxed{-0.5} \end{bmatrix}$$



Reflection (around x-axis) + [Vertical Scaling = 0.5x]

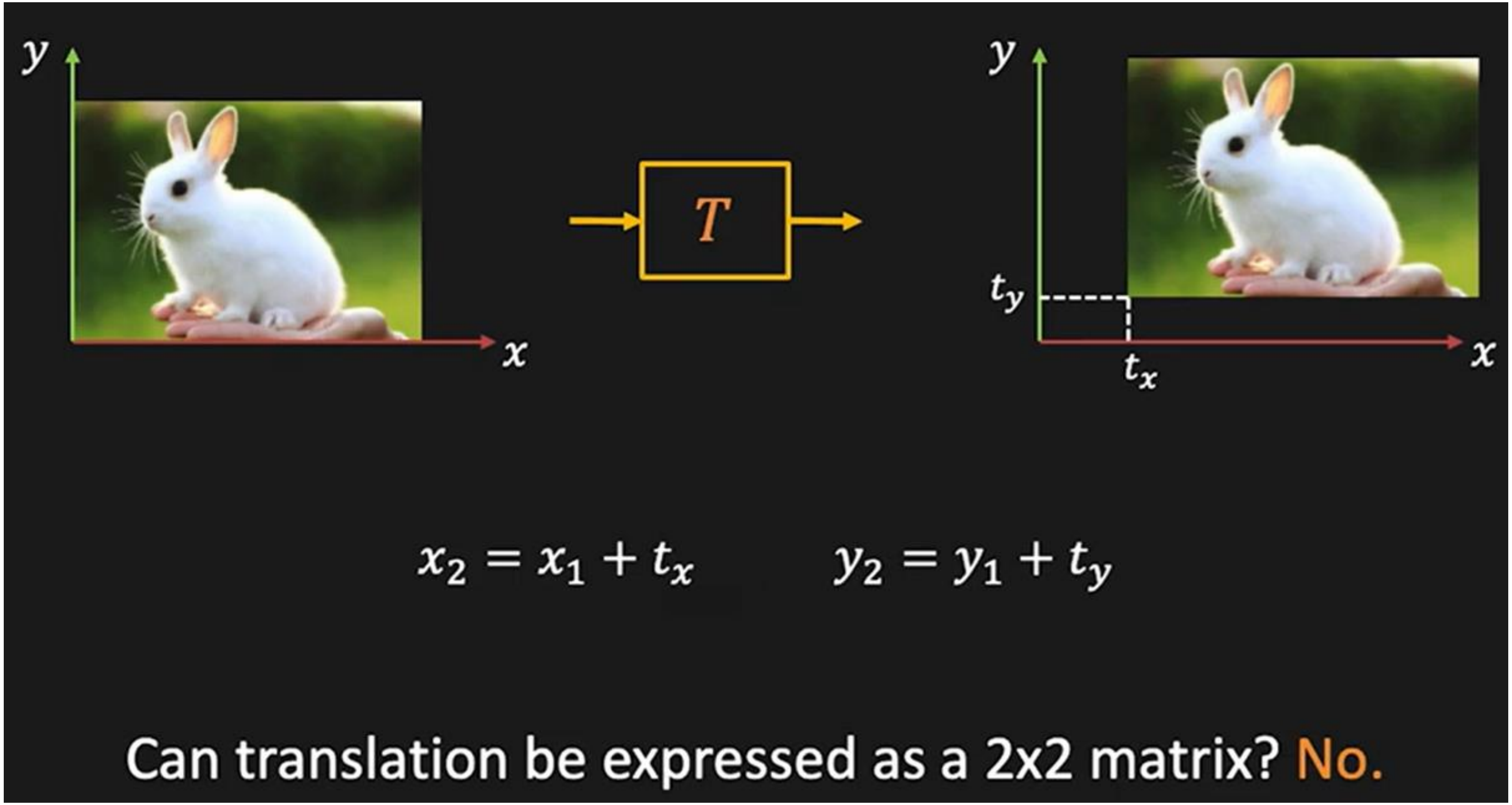
2x2 Matrix Transformations

Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

- Origin maps to the origin
- Lines map to lines
- Parallel lines remain parallel
- Closed under composition

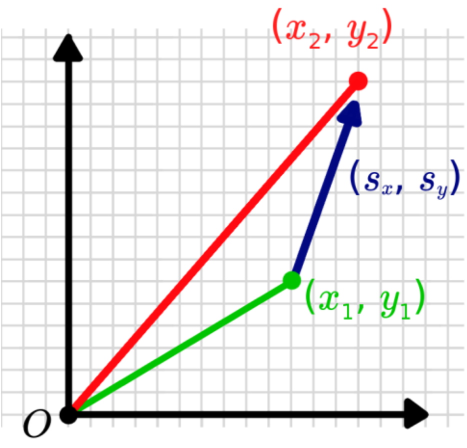
Translation



Translation Problem

❖ We simply need to add the appropriate amount to the x and y coordinates. Say, for example, that we had the point (x,y) and we wanted to shift it by s_x units in the x direction and s_y units in the y direction. We simply perform the following addition:

$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} s_x \\ s_y \end{bmatrix} = \begin{bmatrix} x + s_x \\ y + s_y \end{bmatrix}$$



❖ The problem is that this operation is *non-linear*.

❖ To solve this problem we're going to introduce a slightly modified representation of our coordinates. This new system is called homogeneous coordinates.

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 + s_x \\ y_1 + s_y \\ 1 \end{bmatrix}$$

❖ The first thing we have to do is modify our coordinates, which simply involves taking a "1" onto the end of our point vector. we are looking for a 3×3 matrix

$$\begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 1 \end{bmatrix}$$

❖ Firstly, we need to guarantee a 1 in the bottom element of the result.

$$\begin{bmatrix} 1 & 0 & ? \\ 0 & 1 & ? \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 + ? \\ y_1 + ? \\ 1 \end{bmatrix}$$

❖ Secondly, we know that for the first element of our result, there is one x_1 and no y_1 , and vice versa for the second element.

❖ Lastly, the top of the right column of our matrix will contain the column vector we want to translate by.

$$\begin{bmatrix} 1 & 0 & s_x \\ 0 & 1 & s_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 + s_x \\ y_1 + s_y \\ 1 \end{bmatrix}$$

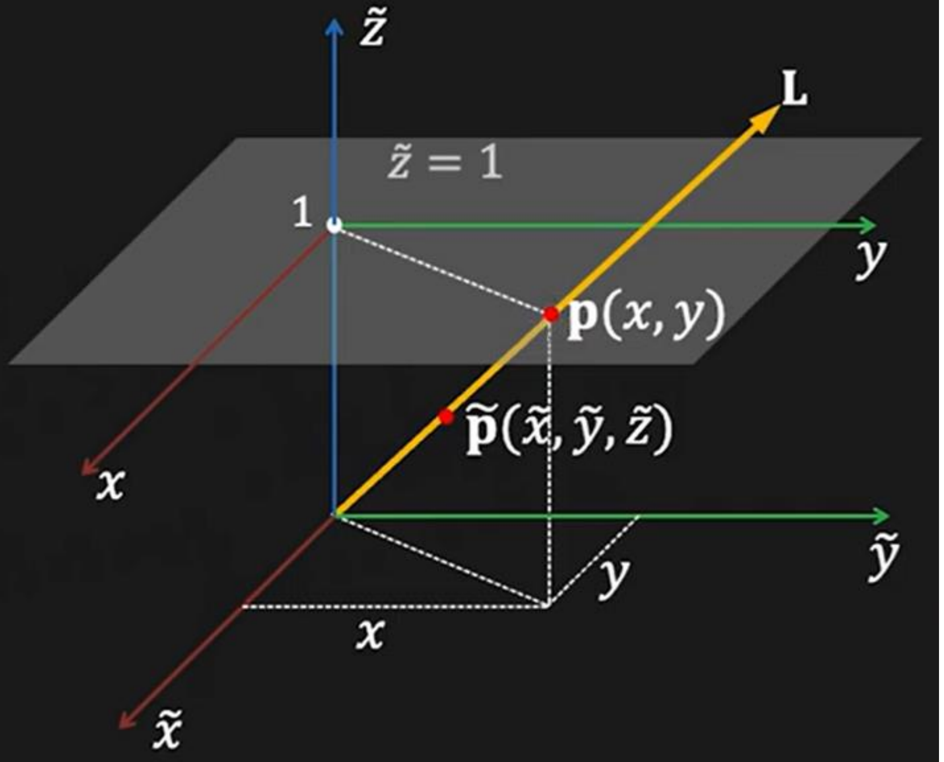
By using homogeneous coordinates, we can represent our non-linear translation as a linear transformation.

Homogeneous Representation

The homogenous representation of a 2D point $\mathbf{p} = (x, y)$ is a 3D point $\tilde{\mathbf{p}} = (\tilde{x}, \tilde{y}, \tilde{z})$. The third coordinate $\tilde{z} \neq 0$ is fictitious such that:

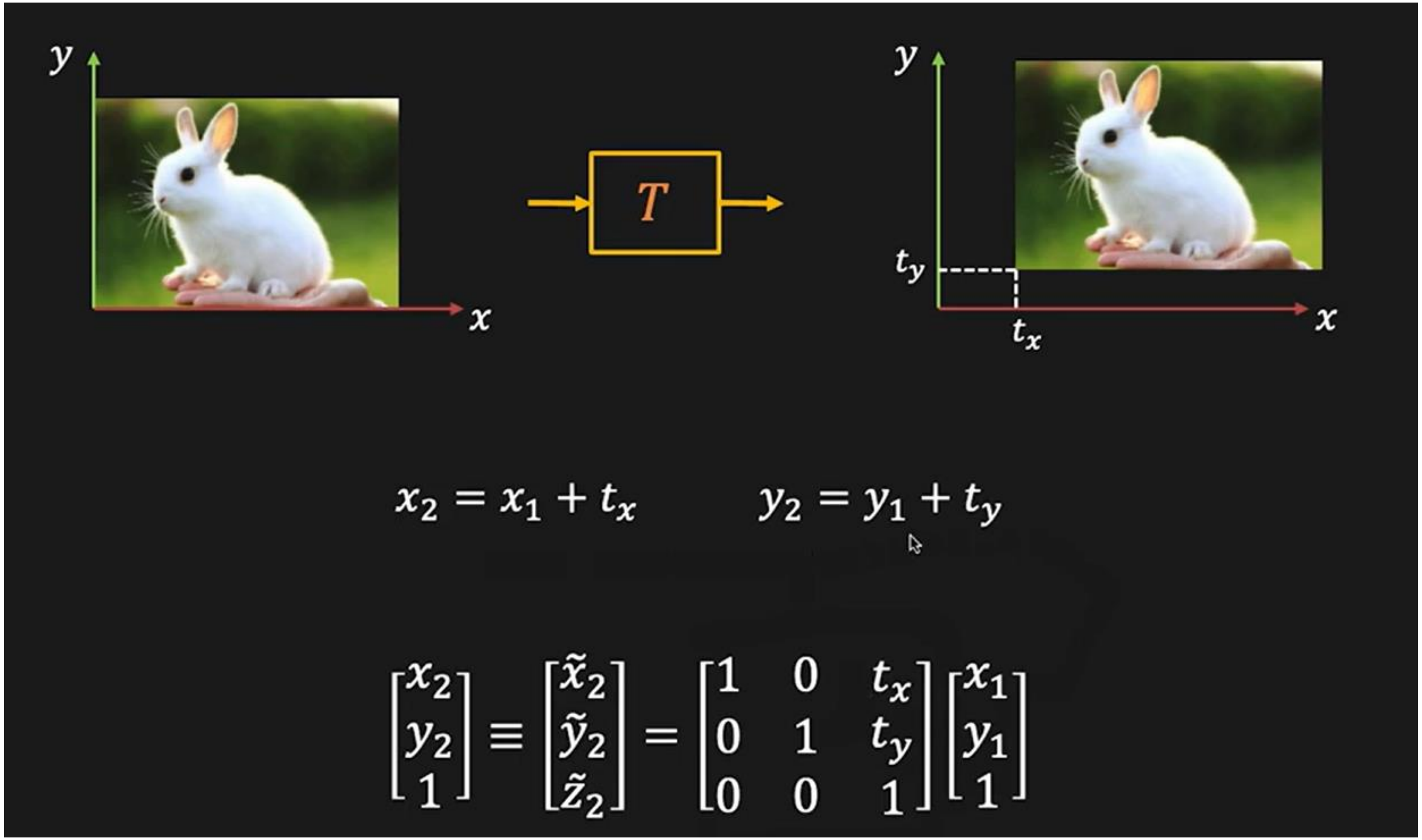
$$x = \frac{\tilde{x}}{\tilde{z}} \qquad y = \frac{\tilde{y}}{\tilde{z}}$$

$$\boxed{\mathbf{p} \equiv \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{z}x \\ \tilde{z}y \\ \tilde{z} \end{bmatrix} \equiv \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \tilde{\mathbf{p}}}$$



Every point on line $\mathbf{L}$ (except origin) represents the homogenous coordinate of $\mathbf{p}(x, y)$

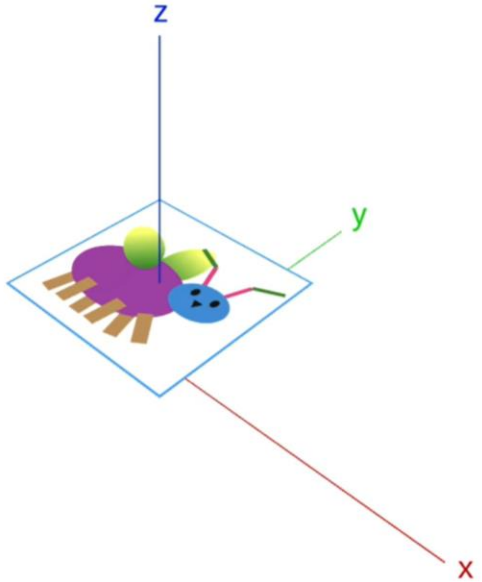
Translation



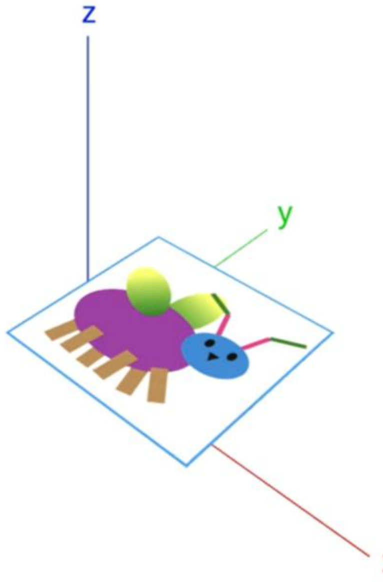
Translation

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

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$



$$\begin{bmatrix} 1.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$



Scaling, Rotation, Skew and Translation

$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Scaling

$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} 1 & m_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Skew

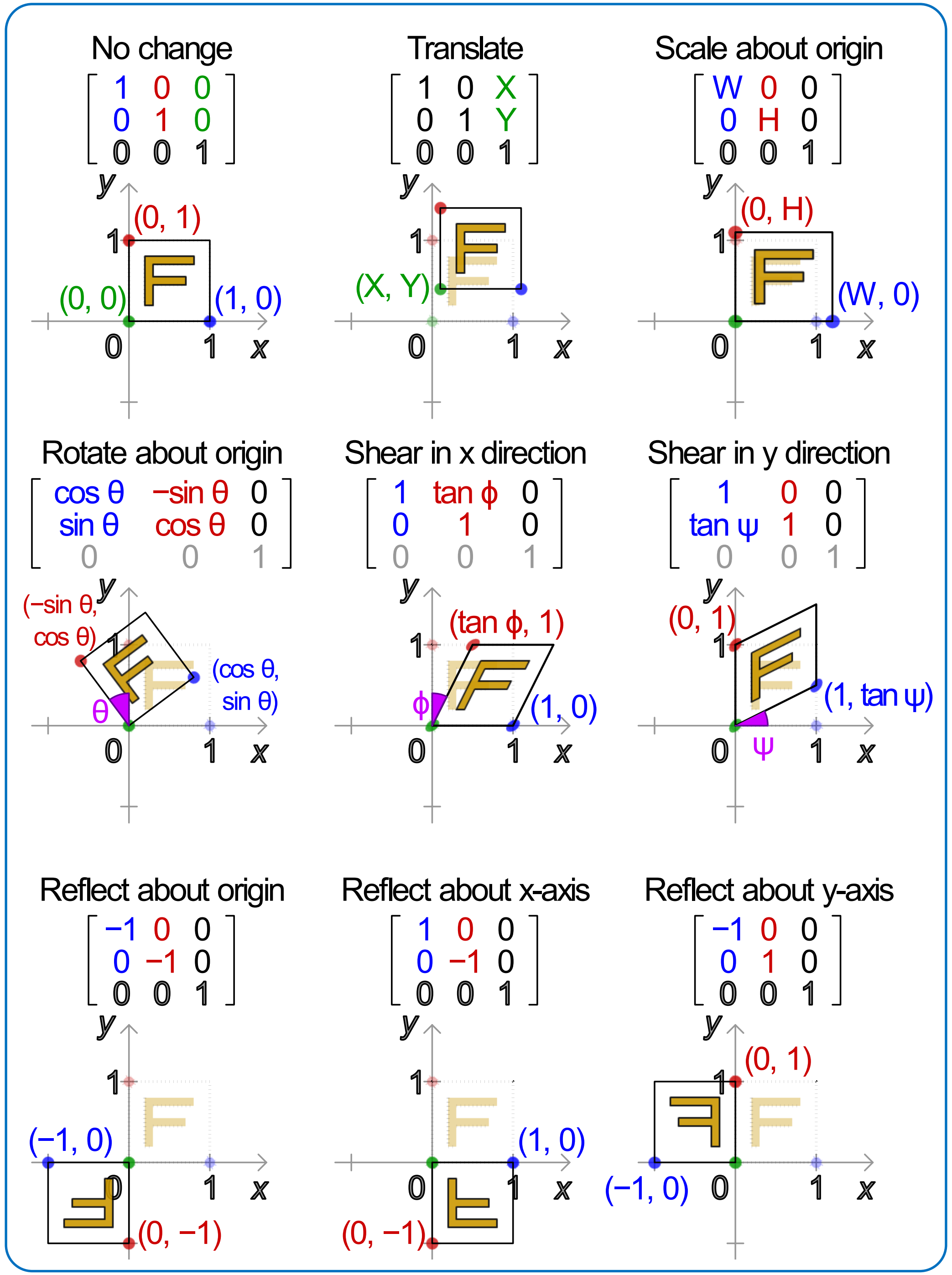
$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Rotation

Affine Transformation Matrices



Affine Transformation

Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix}$$

Transformation Matrix: Take a closer look

We can simplify the representation of this matrix as shown below (note the use of a bold $\mathbf{0}$ to indicate it is a zero vector).

$$\mathbf{T} = \left[\begin{array}{cc|c} \cos(\theta) & -\sin(\theta) & s_x \\ \sin(\theta) & \cos(\theta) & s_y \\ \hline 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} \mathbf{R} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}$$

❖ Take a look at the Transformation matrix:

- The top left corner contains the original rotation matrix
- The top right hand corner contains the translation offset as a column vector
- The very bottom right hand corner contains a 1
- The rest of the bottom row (to the left of the 1) is all zeros.

❖ Properties of transformation matrices:

➢ Chainability

- Affine transformation matrix is *linear* (using homogeneous coordinates), we are able to chain them together using multiplication, just like with the rotation matrices
- A rotation matrix multiplied by another rotation matrix produces a rotation matrix, in the same way a transformation matrix multiplied by a transformation matrix will always result in a transformation matrix.

$$\begin{aligned} \mathbf{T}_1\mathbf{T}_2 &= \begin{bmatrix} \mathbf{R}_1 & \mathbf{b}_1 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_2 & \mathbf{b}_2 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_1\mathbf{R}_2 & \mathbf{R}_1\mathbf{b}_2 + \mathbf{b}_1 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_3 & \mathbf{b}_3 \\ \mathbf{0} & 1 \end{bmatrix} \end{aligned}$$

➢ Inverse

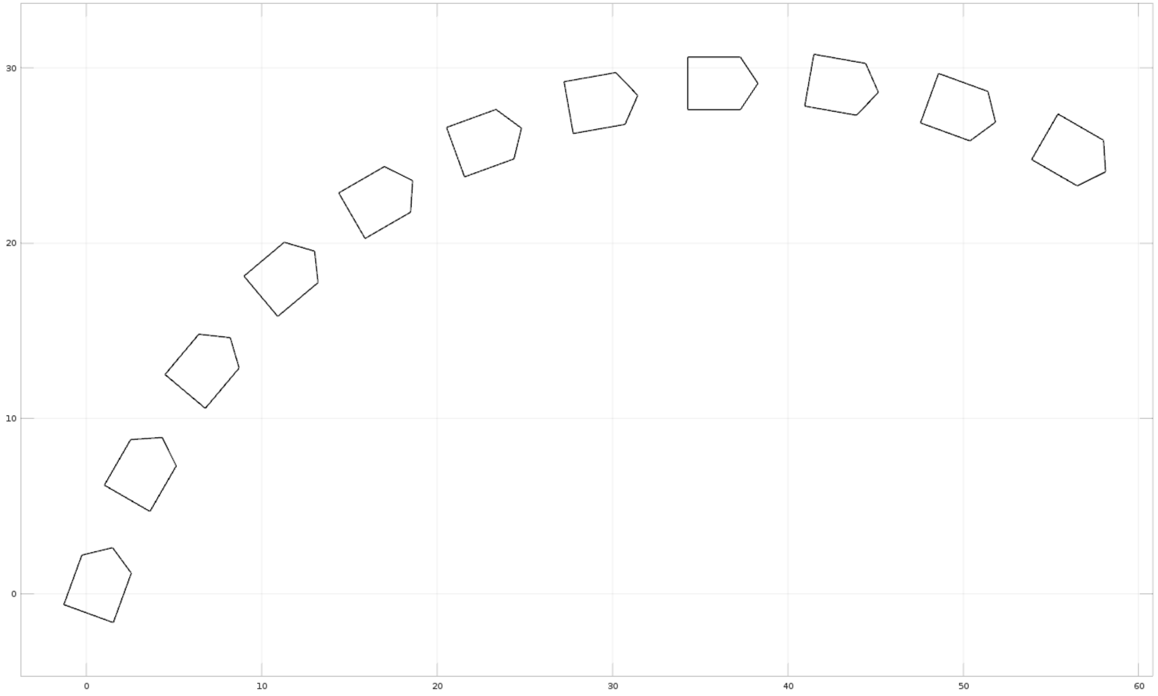
- The second useful property of the transformation matrix is Inverse and taking inverses is relatively easy. Inverse of the rotation matrix is its transpose.
- The formula for the inverse of a transformation matrix **T** and its verification is given below (without the derivation):

$$\begin{aligned} \mathbf{T}^{-1} &= \begin{bmatrix} \mathbf{R} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}^{\mathrm{T}} & -\mathbf{R}^{\mathrm{T}}\mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \mathbf{T}\mathbf{T}^{-1} &= \begin{bmatrix} \mathbf{R} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}\mathbf{R}^{-1} & \mathbf{R}(-\mathbf{R}^{-1})\mathbf{b} + \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & -\mathbf{I}\mathbf{b} + \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \end{aligned}$$

Note that if you use this structure to represent other affine transformations (e.g. a shear matrix instead of a rotation), you can use the middle result, but you have to take inverse for R.

Affine Transformation Trajectory Demo

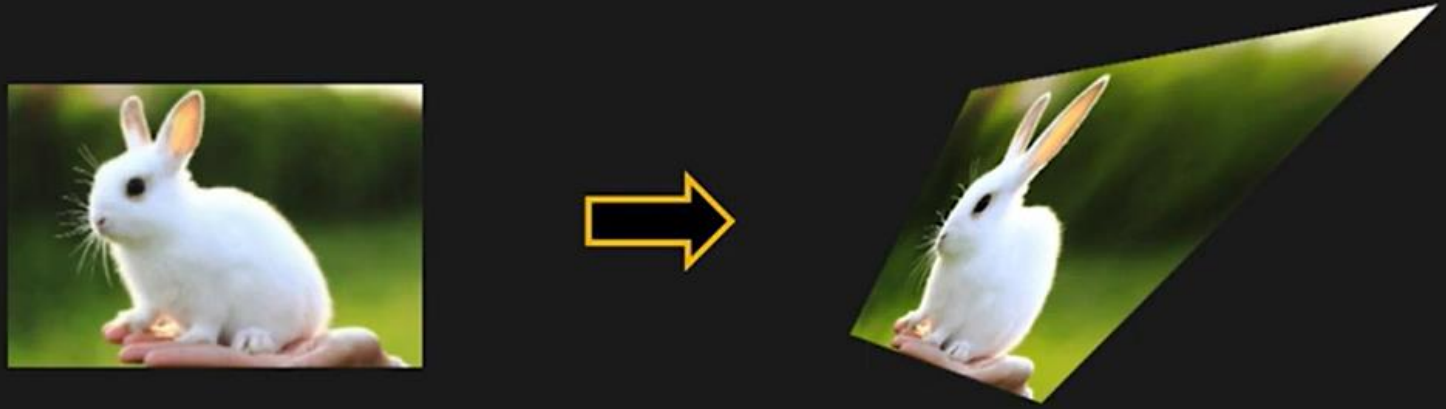



```
%% TRANSFORMATION TRAJECTORY DEMO
% Demonstrates the use of affine transformation matrices
% to plot an object moving along a trajectory

% Set up an array of points
x_points = [2, 2, 0.5, -1, -1, 2];
y_points = [-1, 2, 3, 2, -1, -1];
points = [x_points; y_points; ones(1, length(x_points))];

% Initial Conditions
sx = 0; sy = 0; theta = -20;

clf;
for t = 0:10
% Compute the transformation matrix
transf_mat = [cosd(theta), -sind(theta), sx; ...
sind(theta), cosd(theta), sy; ...
0, 0, 1];

% Compute the new points
transf_pts = transf_mat*points;

% Plot the points
plot(transf_pts(1,:), transf_pts(2,:), '-k' );
hold on;

% Update the state for the next plot
sx = sx + 7*cosd(theta+90);
sy = sy + 7*sind(theta+90);
theta = theta - 10;
end

axis equal; grid on;
```

Projective Transformation

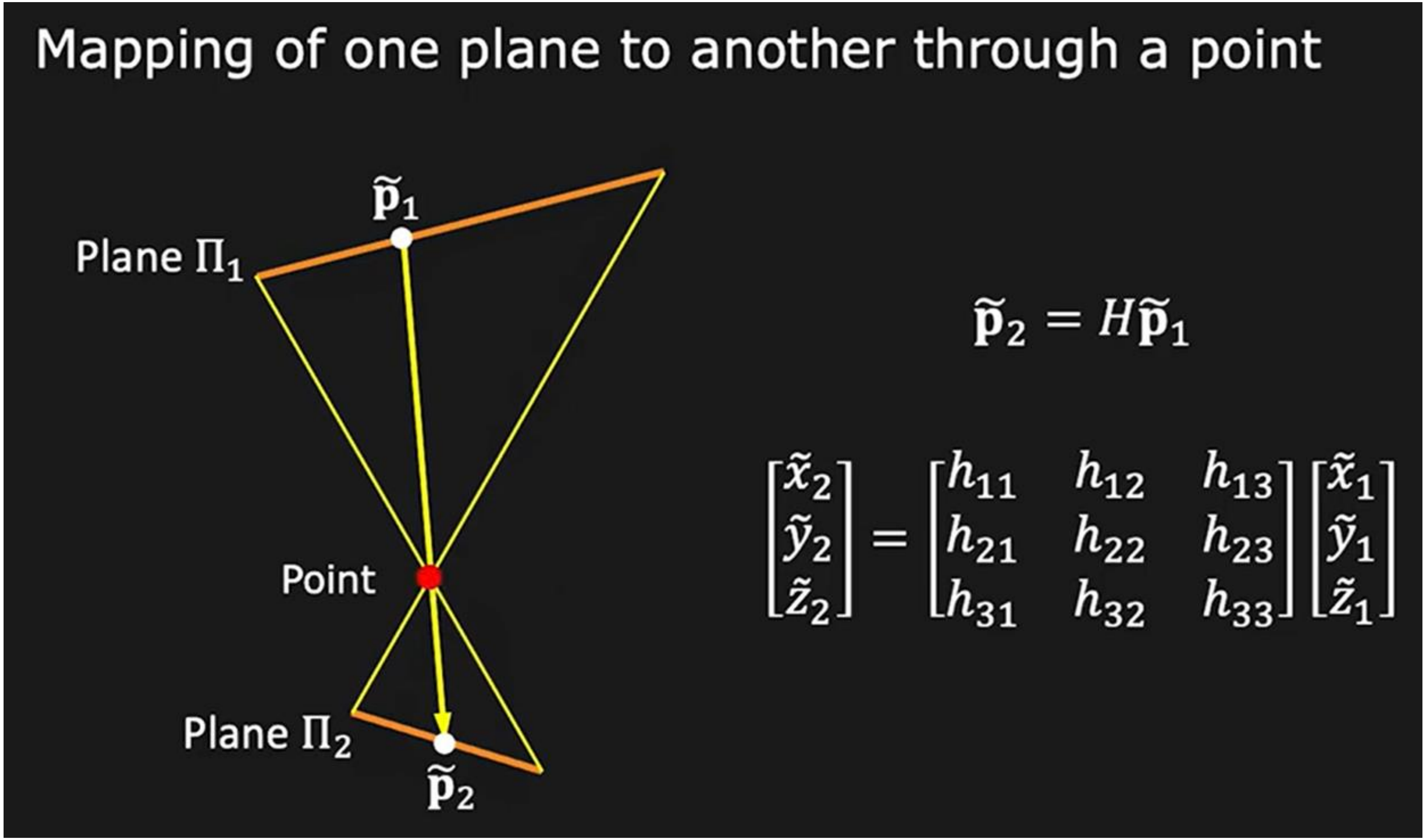
Any transformation of the form:

$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \qquad \tilde{\mathbf{p}}_2 = H\tilde{\mathbf{p}}_1$$



Also called **Homography**

Projective Transformation



Projective Transformation

Homography can only be defined up to a scale.

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} \equiv k \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix}$$

If we fix scale such that $\sqrt{\Sigma(h_{ij})^2} = 1$ then 8 free parameters

- Origin does not necessarily map to the origin
- Lines map to lines
- Parallel lines do not necessarily remain parallel
- Closed under composition

References

- ❖ Affine transformations in 5 minutes
- ❖ Image Stitching | Face Detection
- ❖ What are affine transformations?
- ❖ https://articulatedrobotics.xyz/tutorials/coordinate-transforms/transformation-matrices/
- ❖ https://articulatedrobotics.xyz/tutorials/coordinate-transforms/translations
- ❖ https://www.algorithm-archive.org/contents/affine_transformations/affine_transformations.html