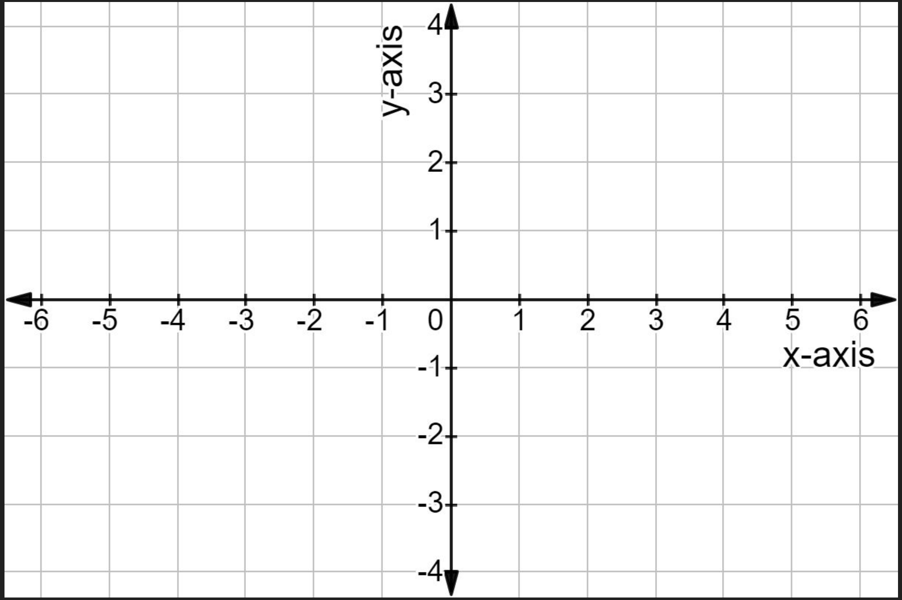
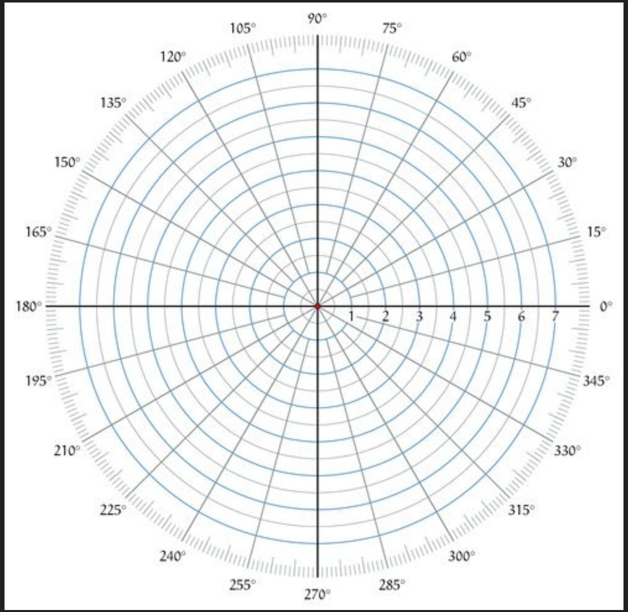
Transformations

Coordinate systems:

cartesian coordinates

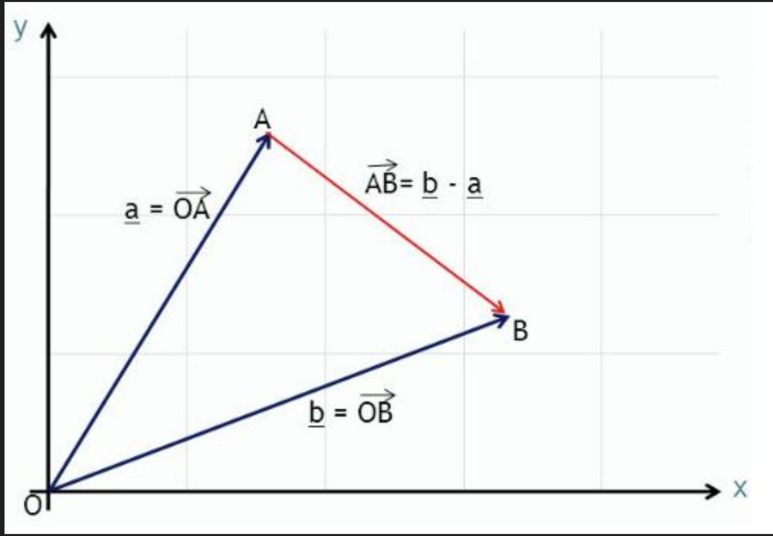


polar coordinates



Euclidean Vector

a quantity having direction as well as magnitude, especially as determining the position of one point in space relative to another.



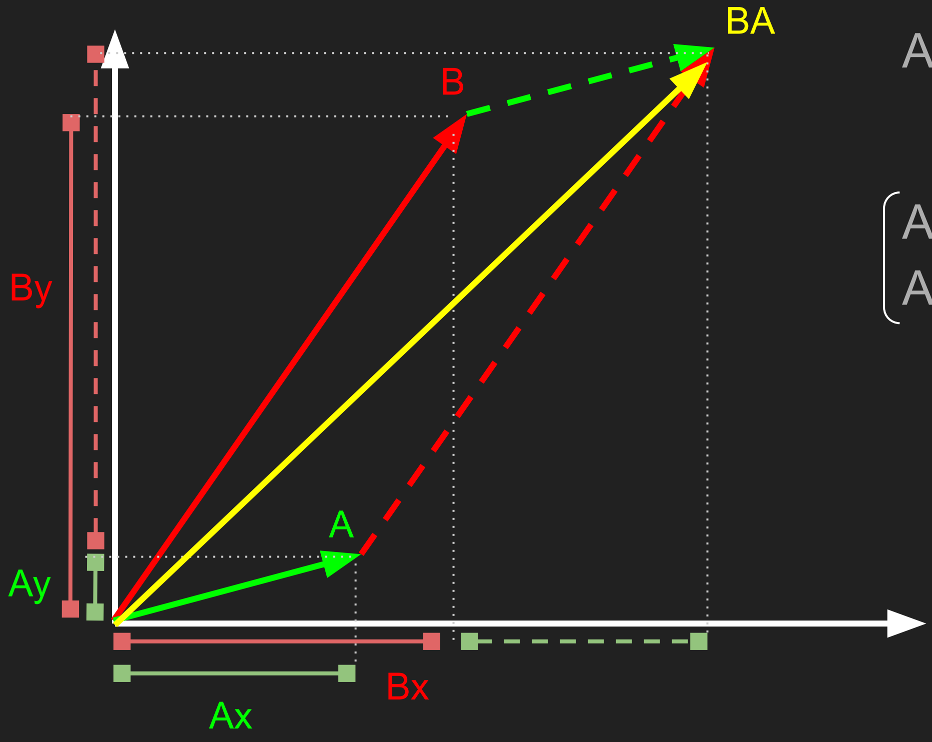
A, B, O: Points

$\overrightarrow{OA}$, $\overrightarrow{OB}$, $\overrightarrow{AB}$: Directions

Vector Operators

Add, Substract (+, -)	Vector ± Vector = Vector
Multiply Scalar (*)	Vector * Real Number = Vector
Dot Product (·)	Vector · Vector = Real Number
Cross Product (×)	Vector × Vector = Vector
Normalize	normalize (Vector) = Vector

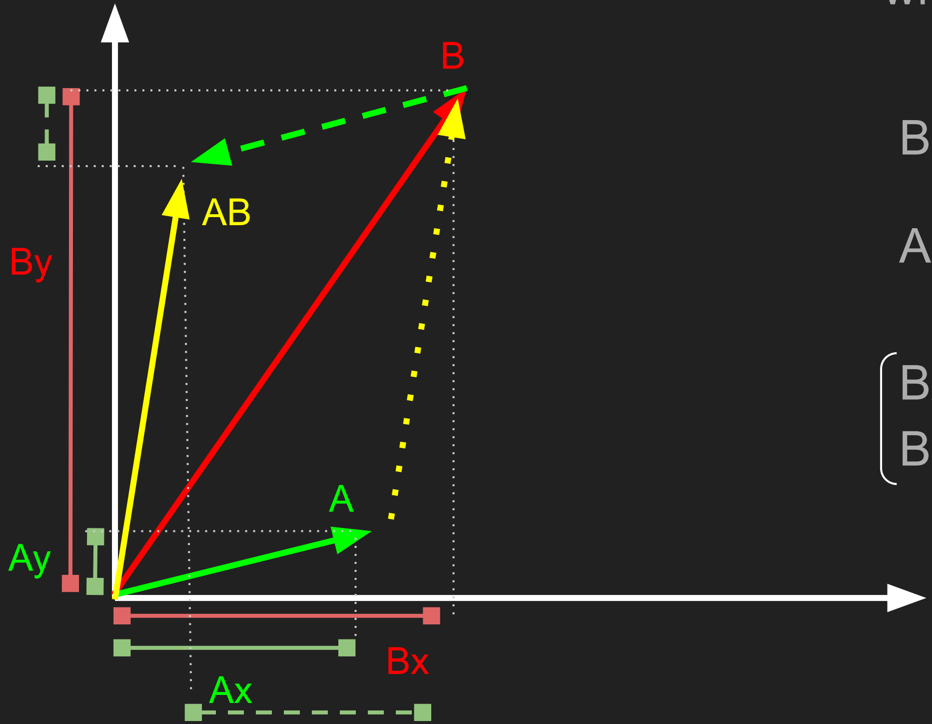
Vector Addition



$$A + B = B + A = BA$$

$$\begin{pmatrix} Ax \\ Ay \end{pmatrix} + \begin{pmatrix} Bx \\ By \end{pmatrix} = \begin{pmatrix} Ax + Bx \\ Ay + By \end{pmatrix} = \begin{pmatrix} BAx \\ BAy \end{pmatrix}$$

Vector Subtraction



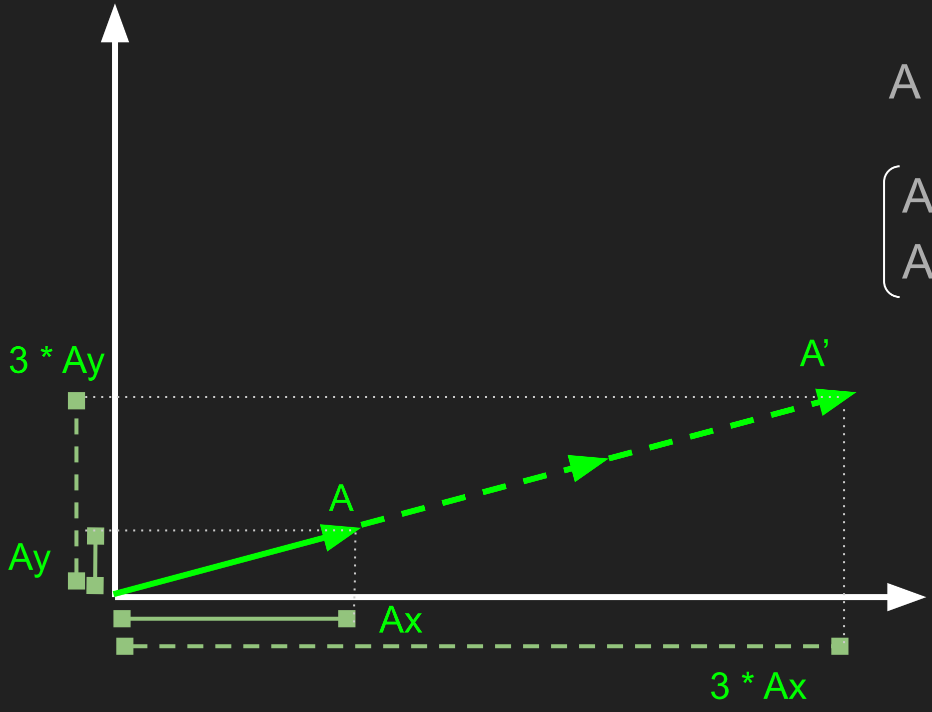
AB or B - A is the vector from A to B, where A is the origin and B is the end.

B - A = AB

A - B != B - A

$$\begin{pmatrix} Bx \\ By \end{pmatrix} - \begin{pmatrix} Ax \\ Ay \end{pmatrix} = \begin{pmatrix} Bx - Ax \\ By - Ay \end{pmatrix} = \begin{pmatrix} ABx \\ ABy \end{pmatrix}$$

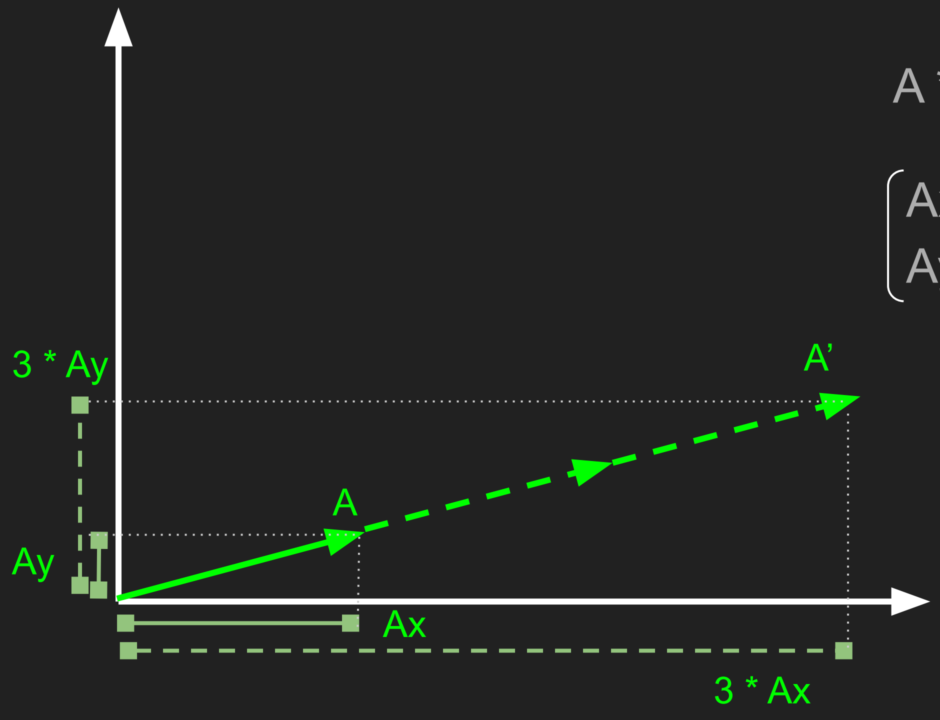
Scalar Multiplication



$$A * 3 = A'$$

$$\begin{pmatrix} Ax \\ Ay \end{pmatrix} * \quad 3 \quad = \begin{pmatrix} Ax * 3 \\ Ay * 3 \end{pmatrix} = \begin{pmatrix} A'x \\ A'y \end{pmatrix}$$

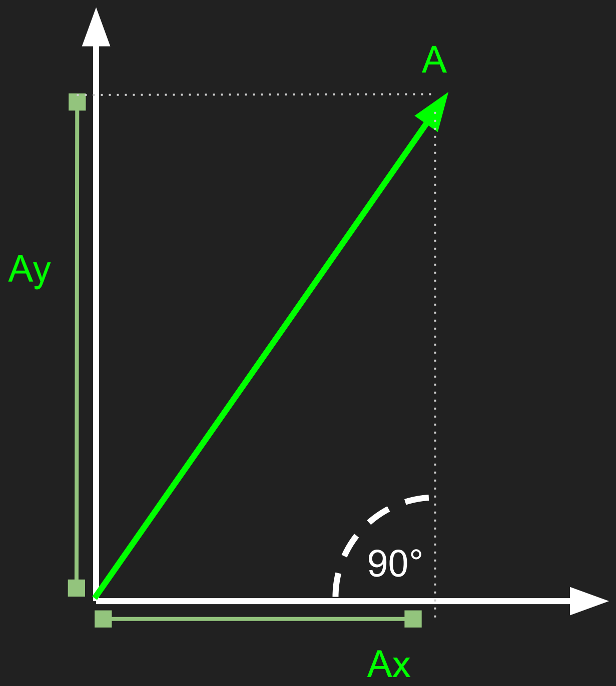
Scalar Multiplication

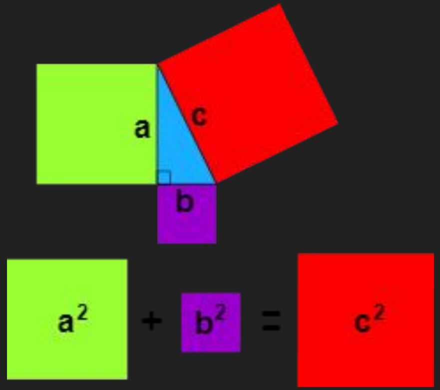


A * 3 = A'

$$\begin{pmatrix} Ax \\ Ay \end{pmatrix} * 3 = \begin{pmatrix} Ax * 3 \\ Ay * 3 \end{pmatrix} = \begin{pmatrix} A'x \\ A'y \end{pmatrix}$$

Magnitude (length): Pythagoras theorem

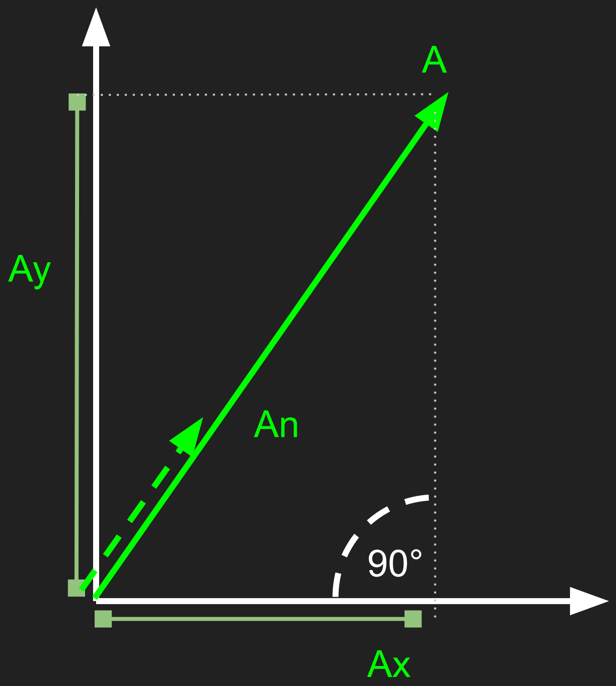




$$\left\| \begin{pmatrix} Ax \\ Ay \end{pmatrix} \right\| = \text{length}(A)$$

$$length \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \sqrt{2^2 + 3^2} = \sqrt{4 + 9} = \sqrt{13}$$

Normalization

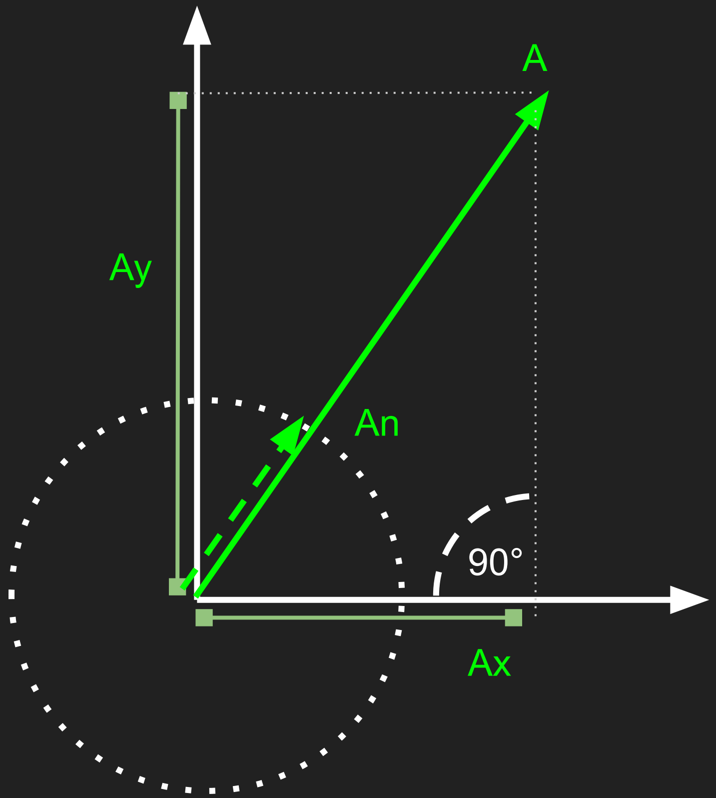


An is the vector with length 1, that faces the same direction of A

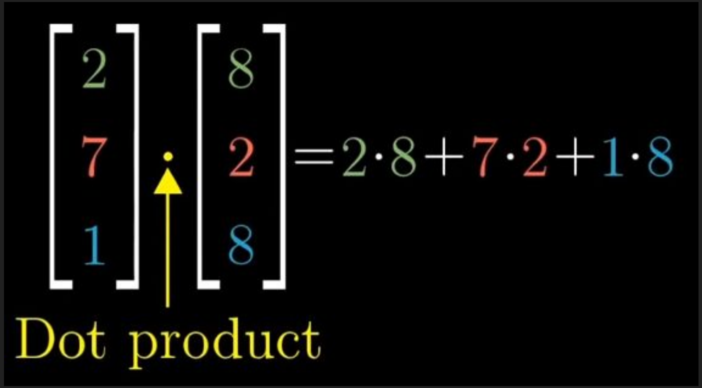
$$\left\| \begin{pmatrix} Ax \\ Ay \end{pmatrix} \right\| = \text{length } (A) = \ell$$

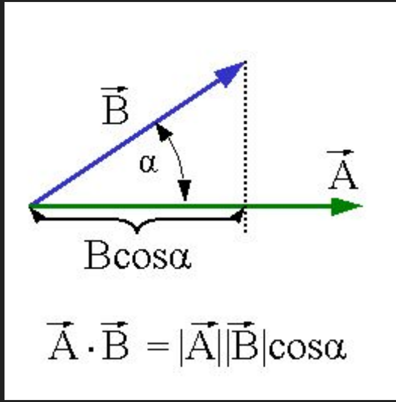
$$An = A * \frac{1}{\ell}$$

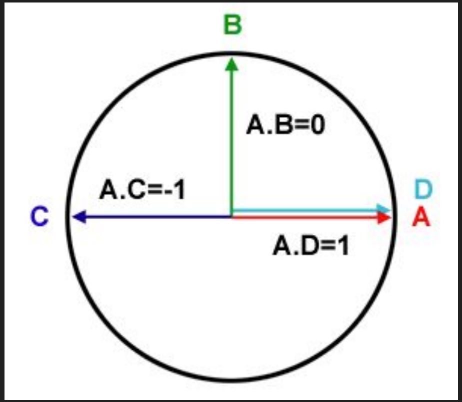
Normalization, the unit circle



Dot Product: Calculating angles

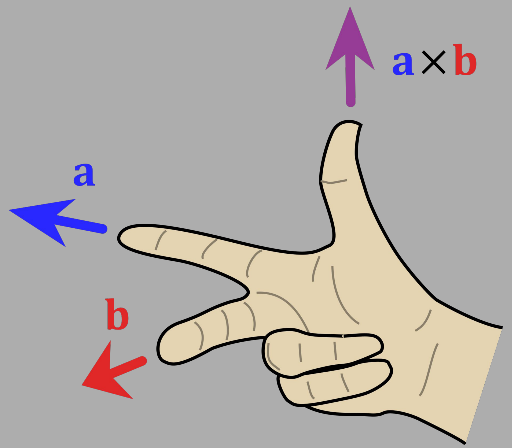


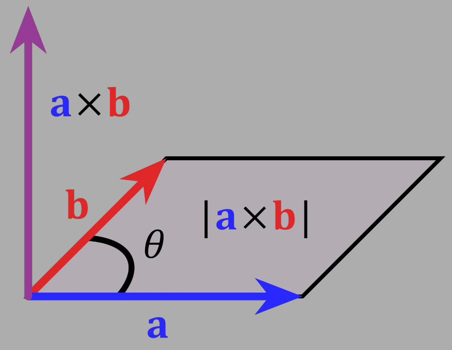




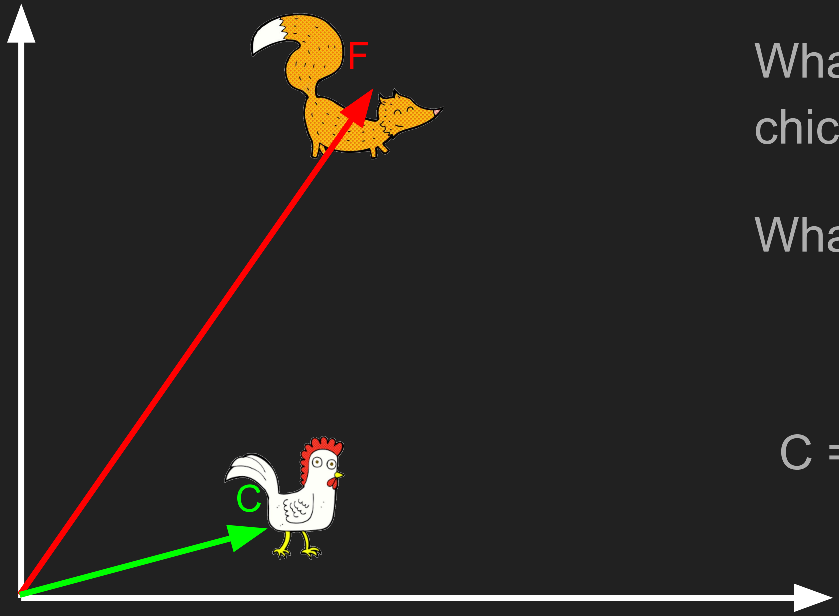
Cross Product

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - b_y a_z \\ a_z b_x - b_z a_x \\ a_x b_y - b_x a_y \end{pmatrix}$$





Mini Exercise



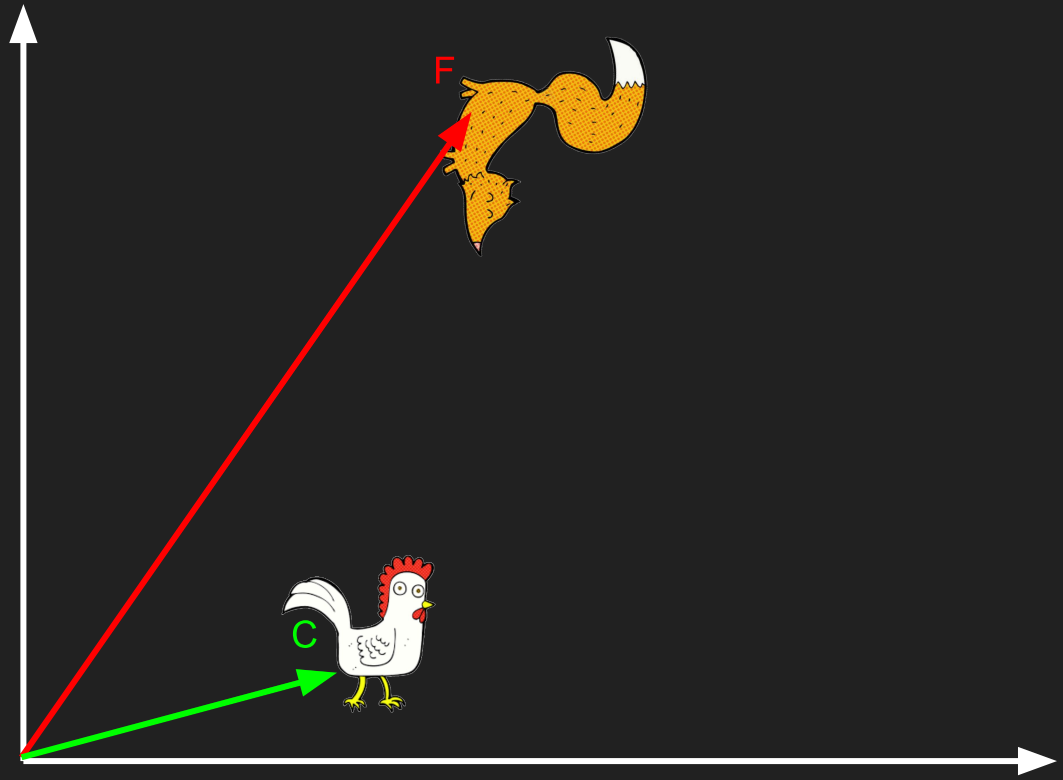
In which direction would the fox have to look to see the chicken?

What is the distance of the fox to the chicken?

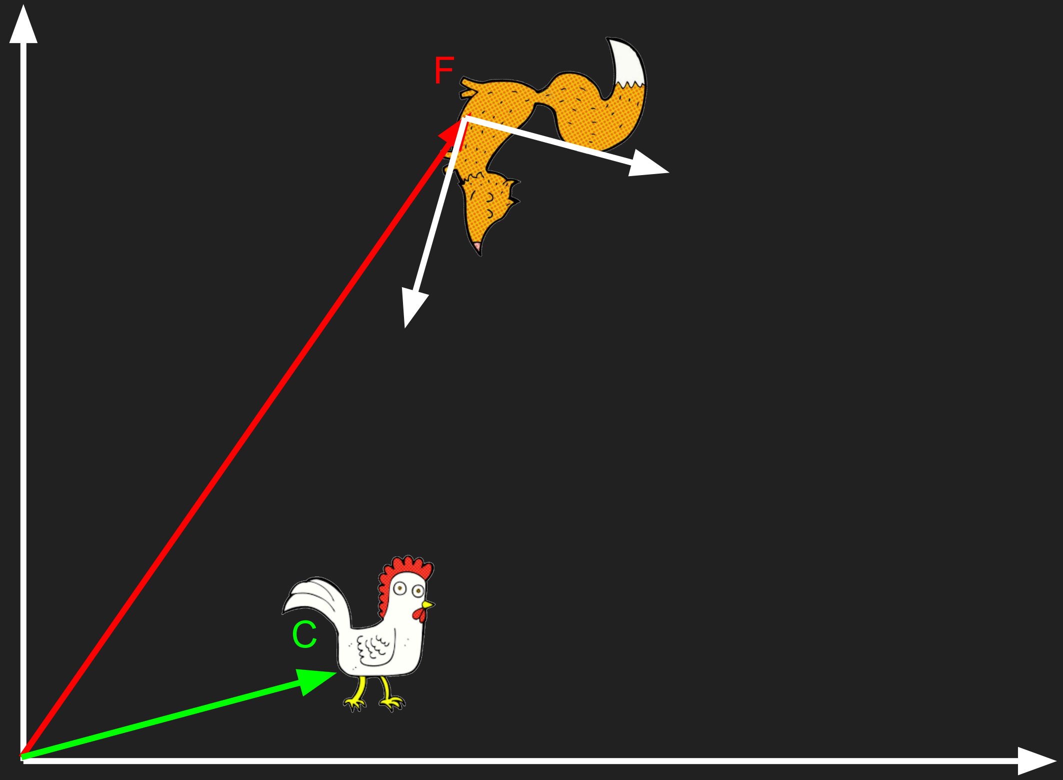
What is the unit vector from fox to chicken?

$C = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ $F = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$

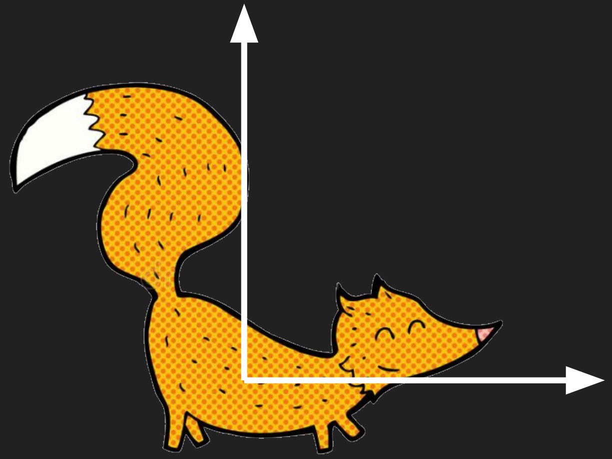
Affine Spaces and linear transformations

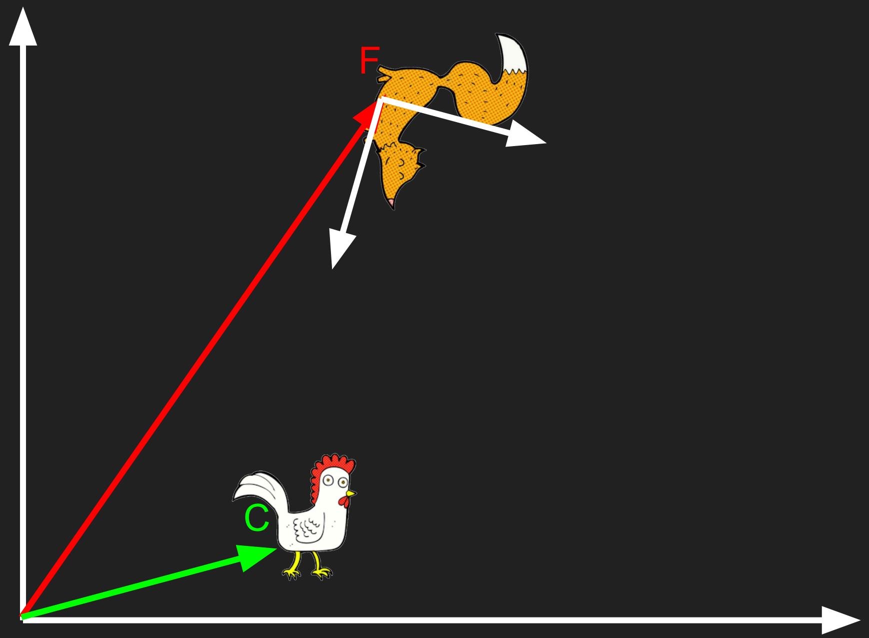


Affine Spaces and linear transformations

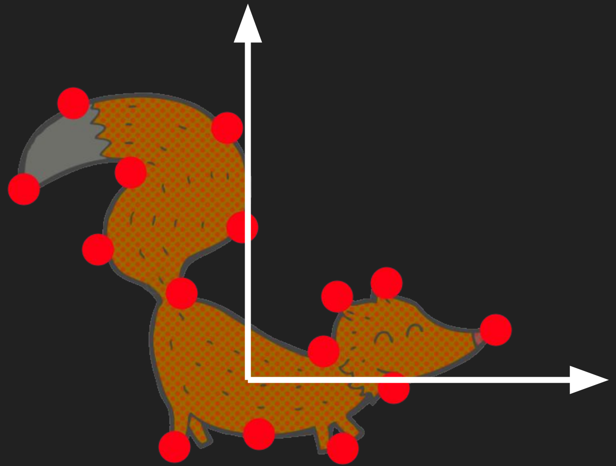


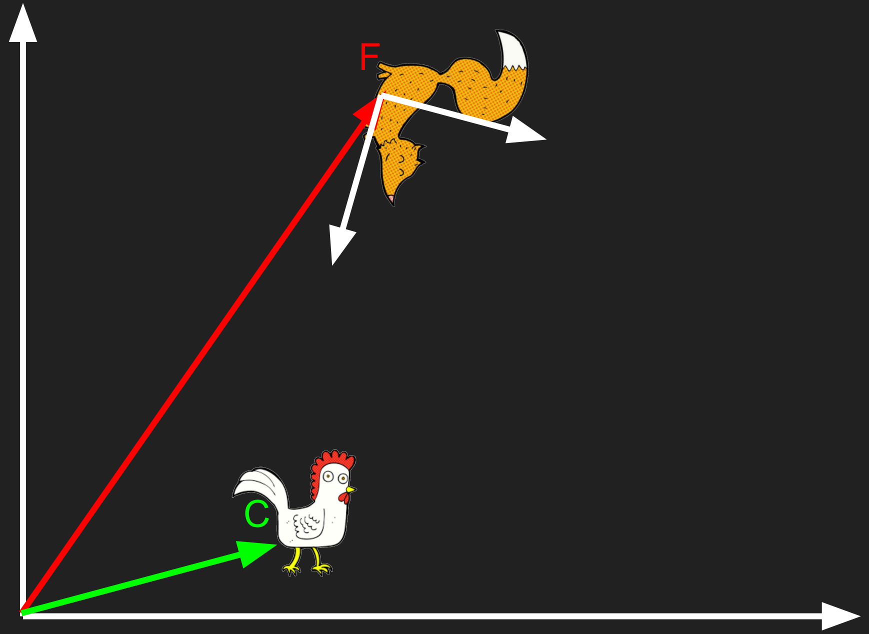
Affine Spaces and linear transformations



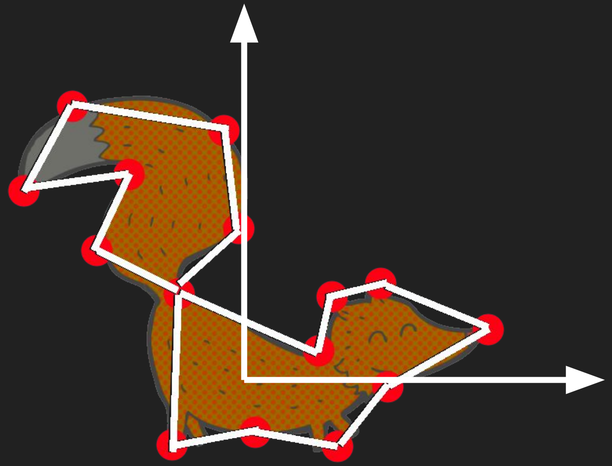


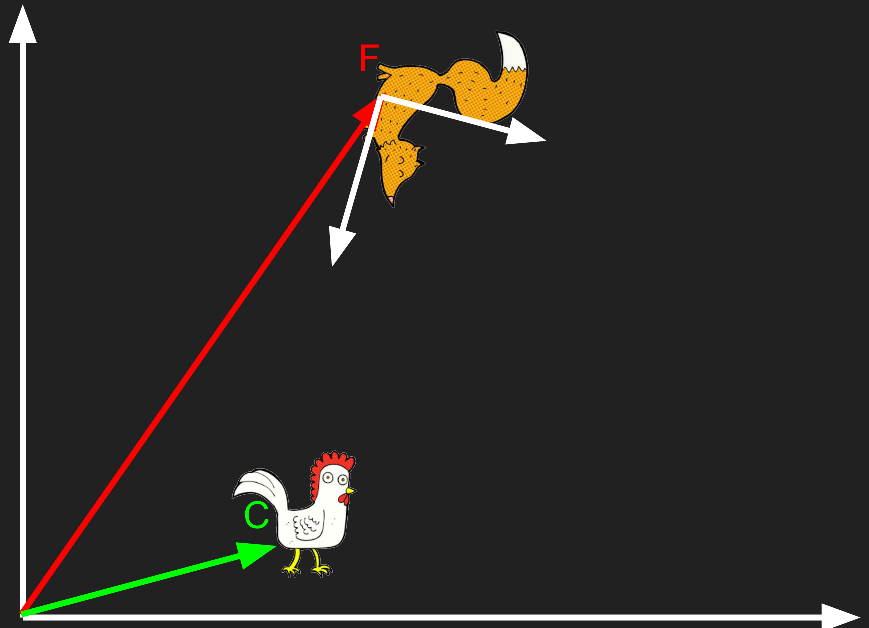
Affine Spaces and linear transformations



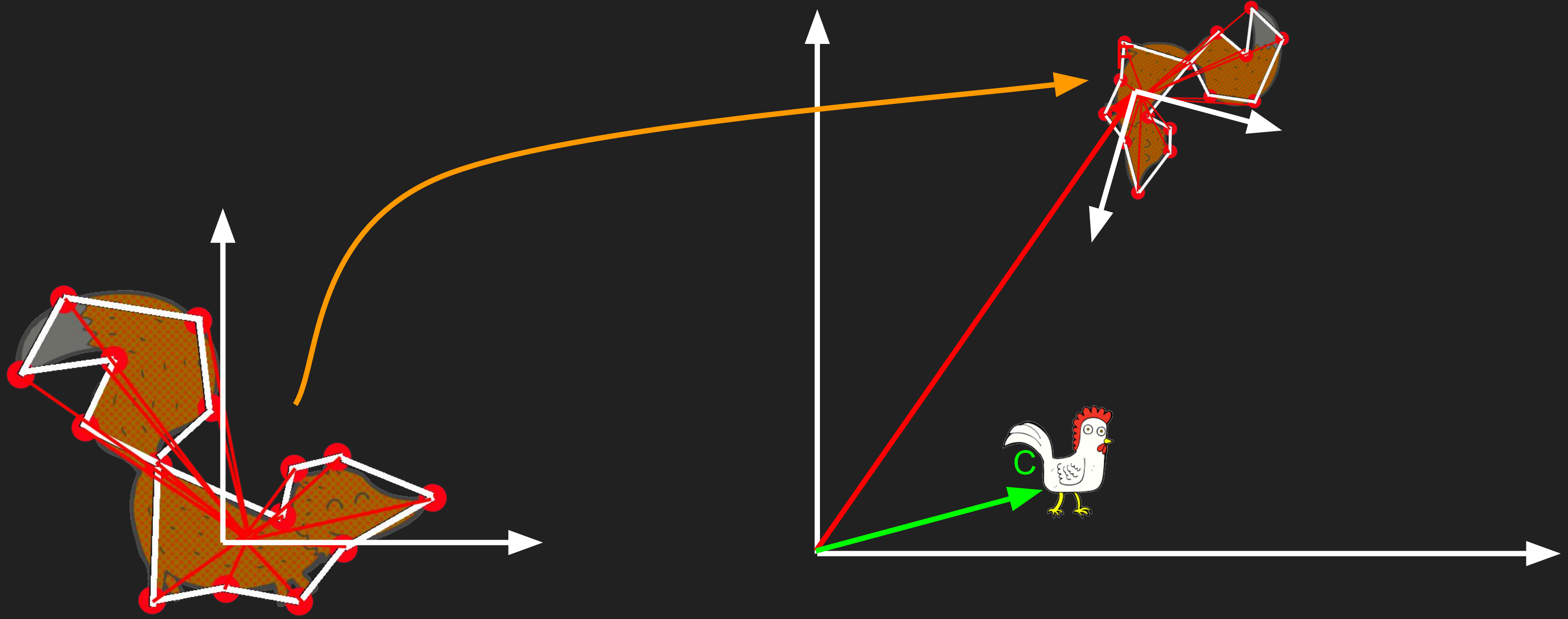


Affine Spaces and linear transformations



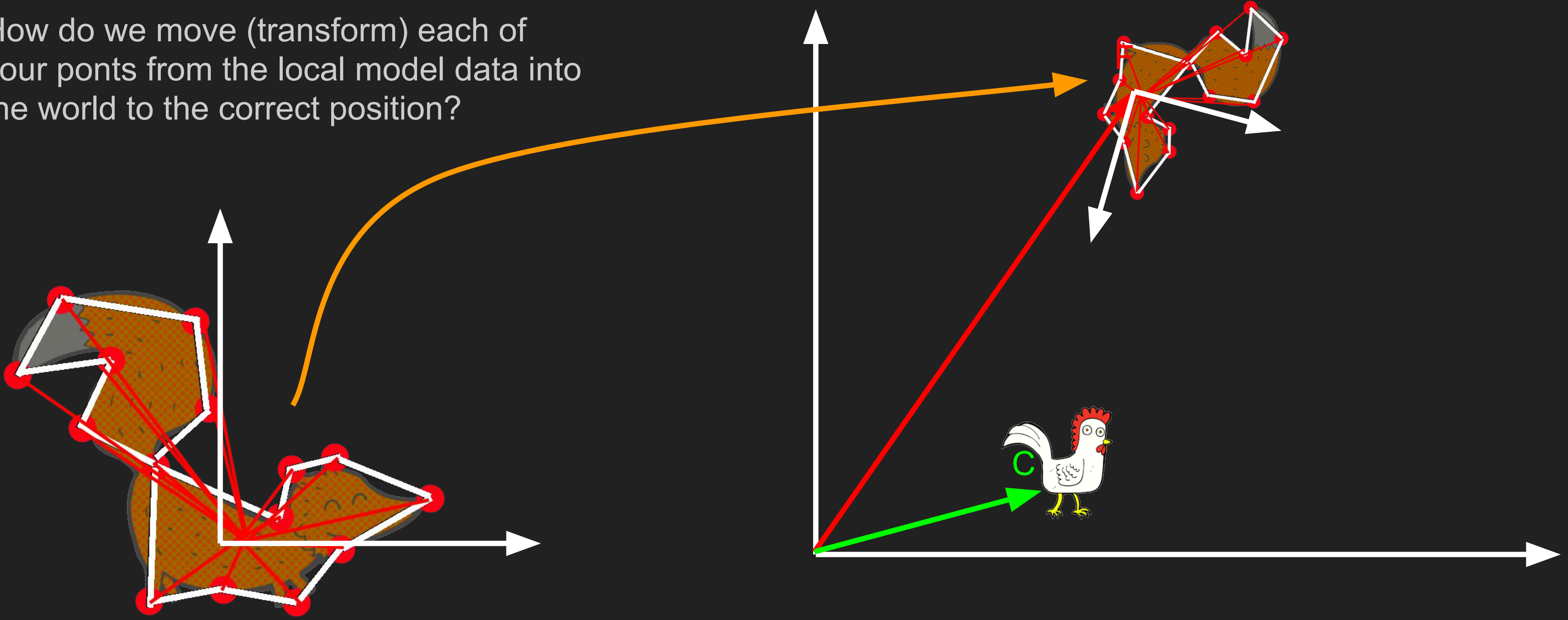


Affine Spaces and linear transformations



Affine Spaces and linear transformations

How do we move (transform) each of your ponts from the local model data into the world to the correct position?



Local Space of the Fox Model

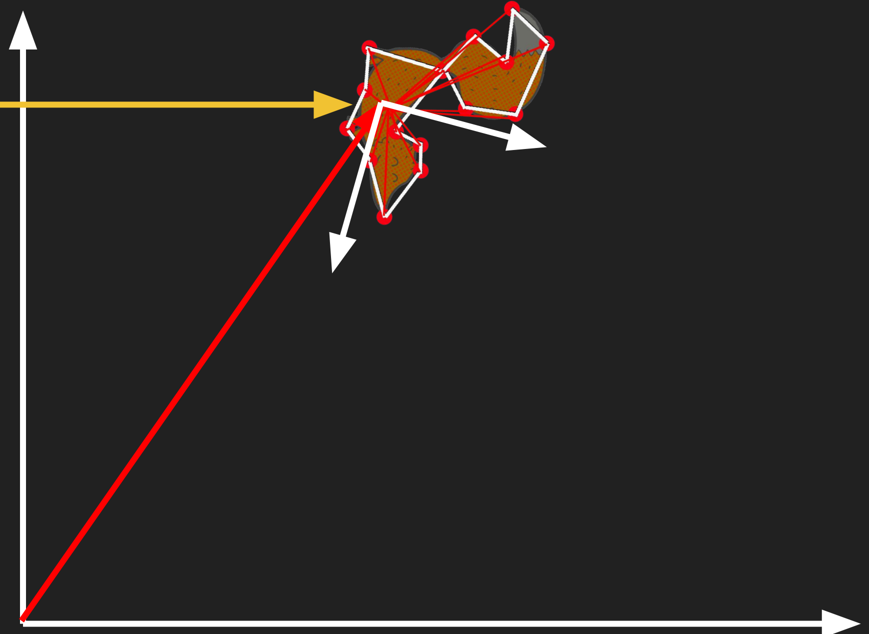
World Space

The Transform

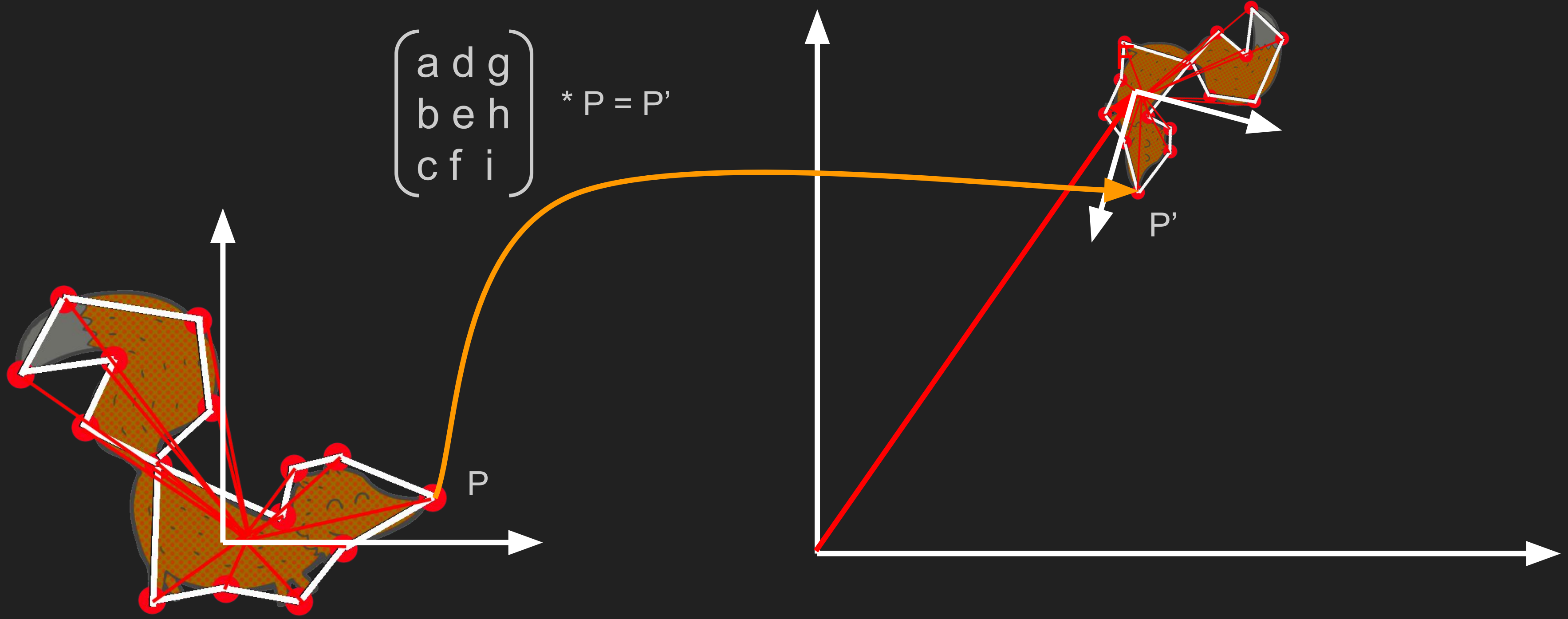
Fox Transform:
-Position
-Rotation
-Scale

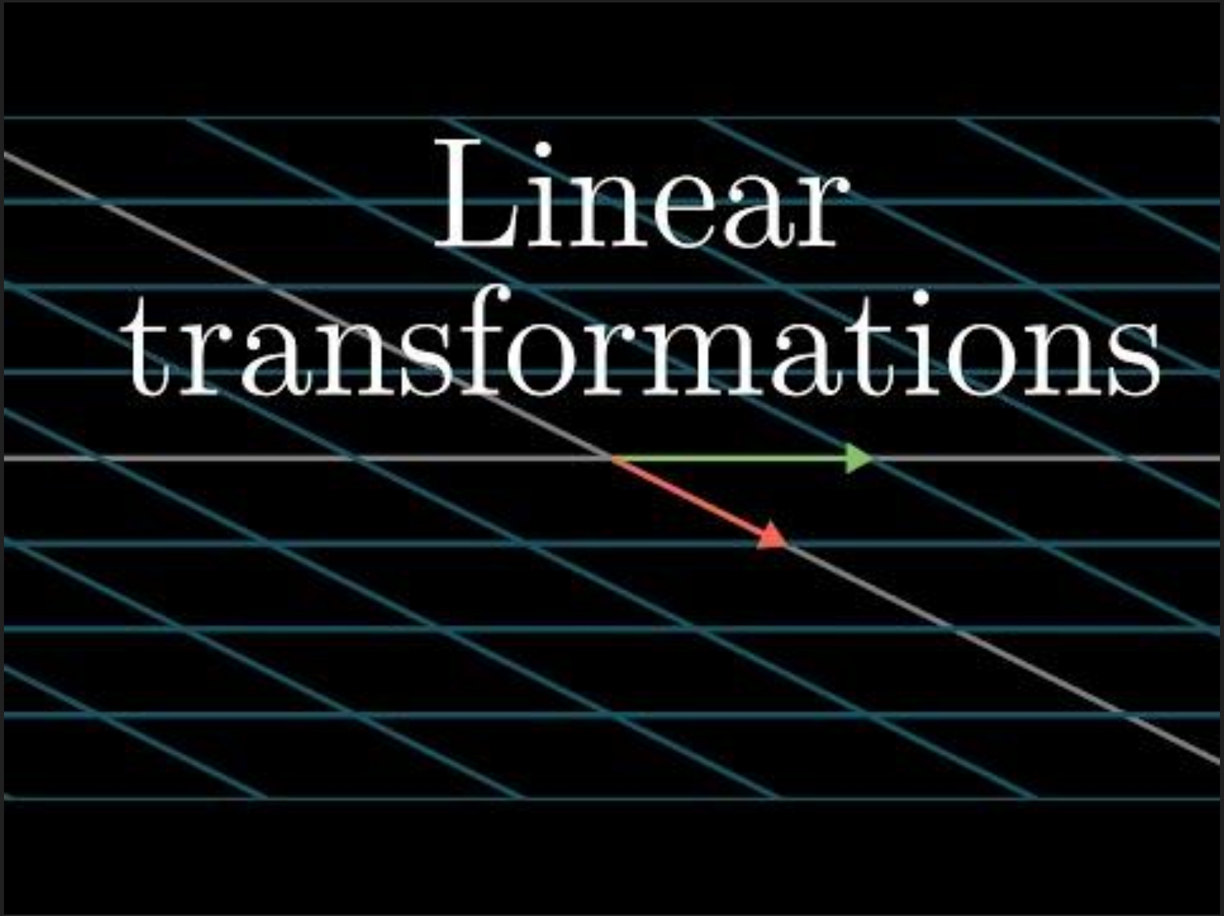
Fox Transform Matrix

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$



Transformation to World Space using Matrix

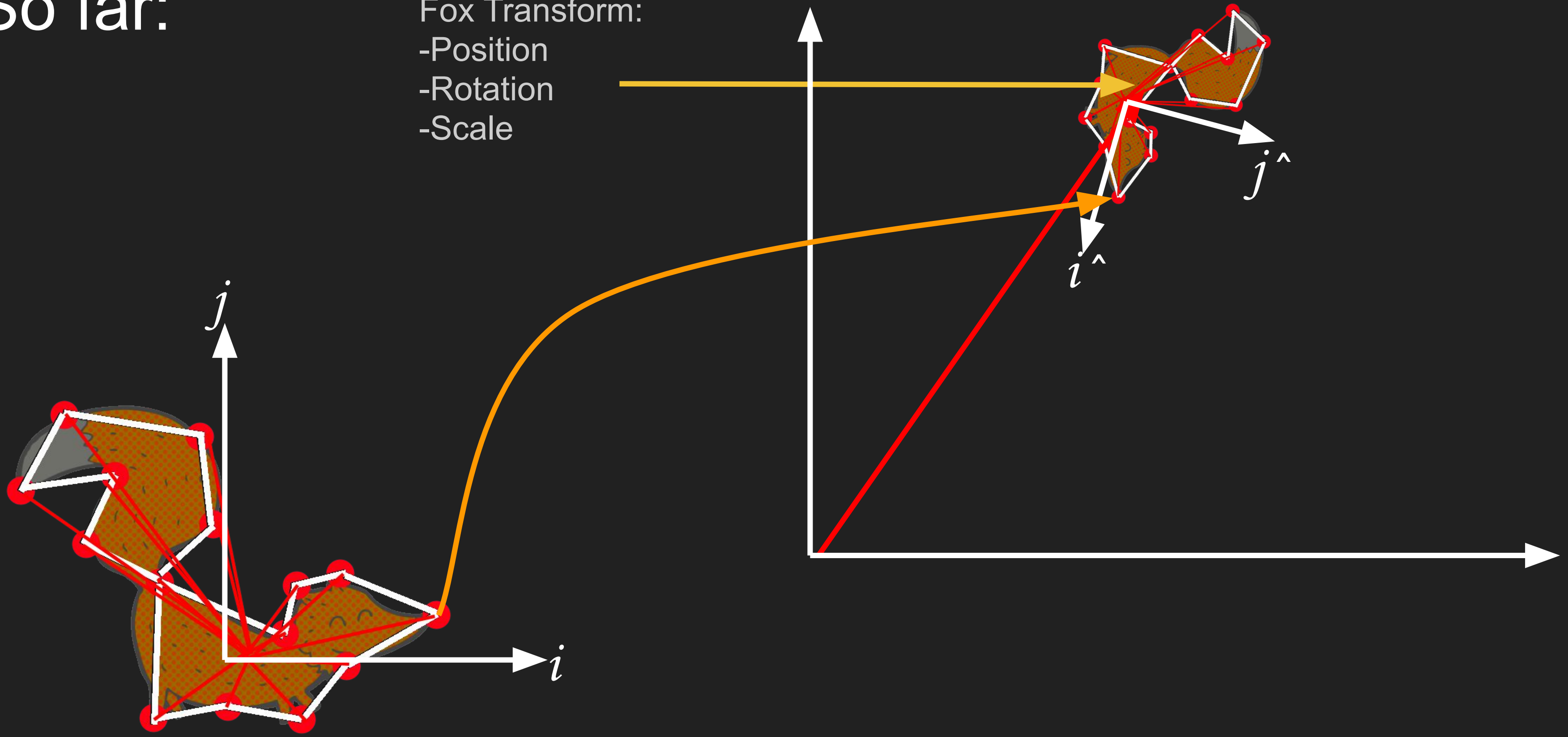




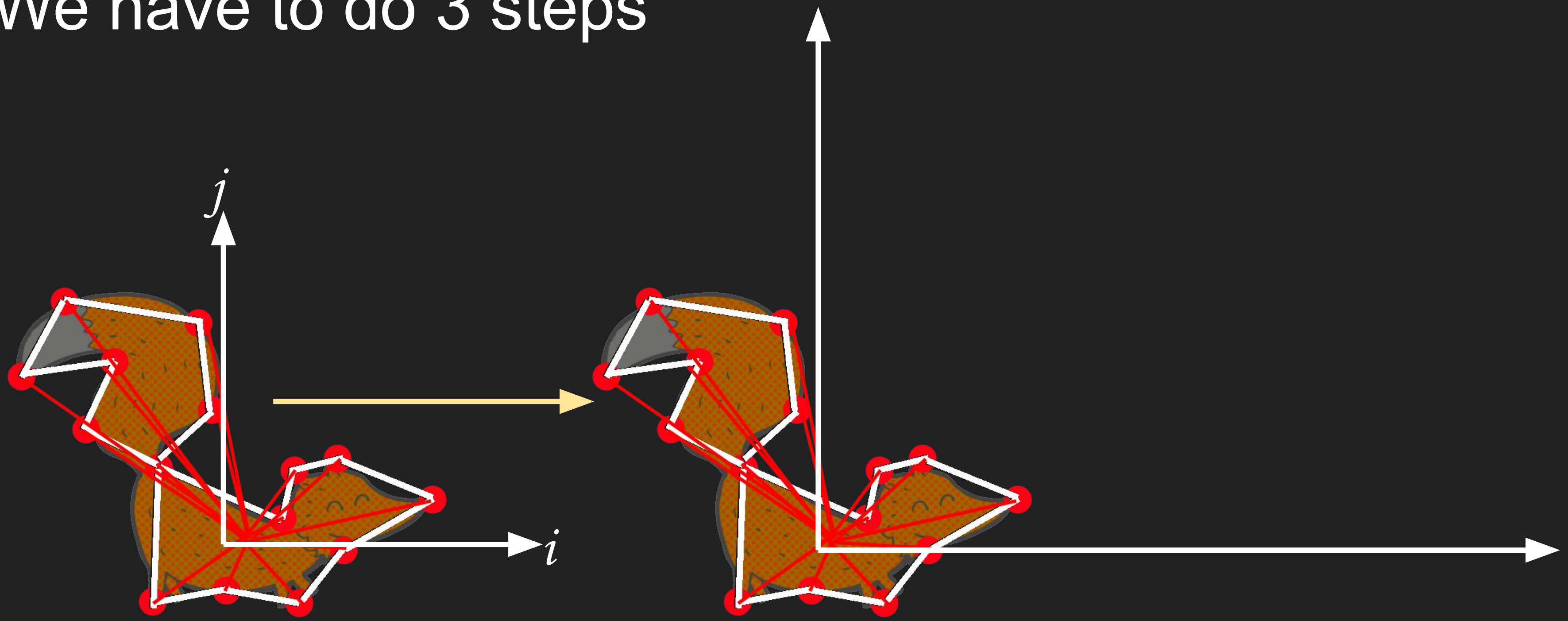
https://www.youtube.com/watch?v=kYB8IZa5AuE&feature=emb_logo

So far:

Fox Transform:
-Position
-Rotation
-Scale



We have to do 3 steps

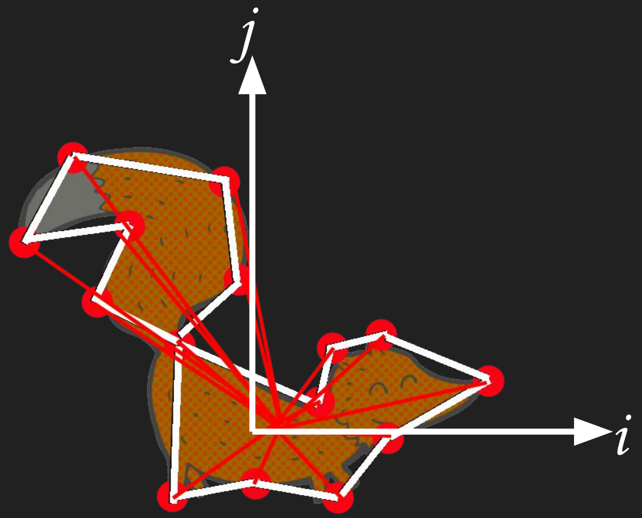


Local Space of the Fox Model

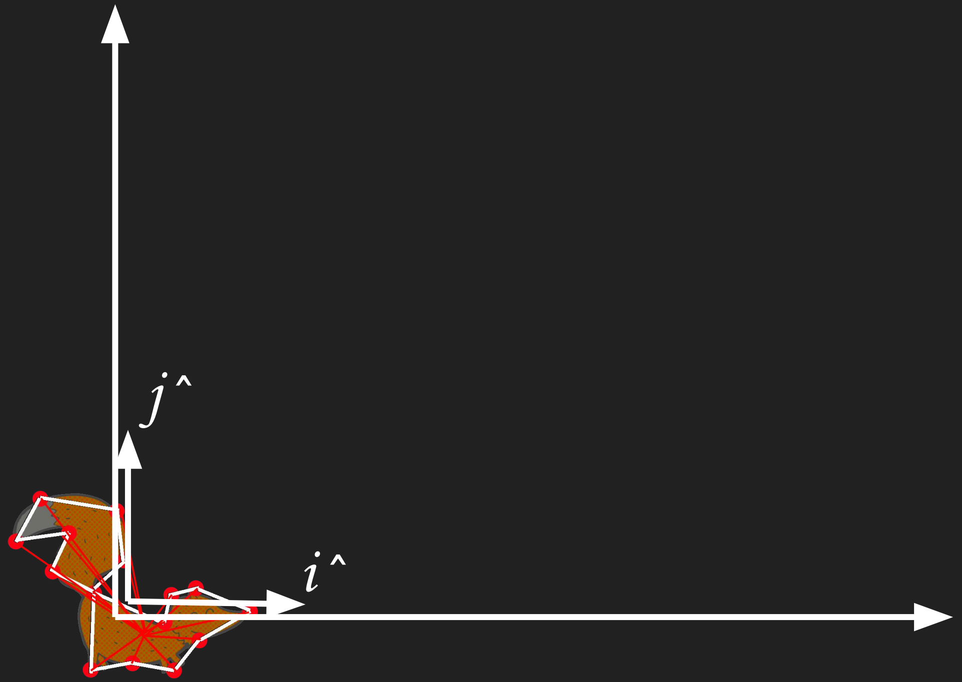
World Space

We have to do 3 steps

1: Scale

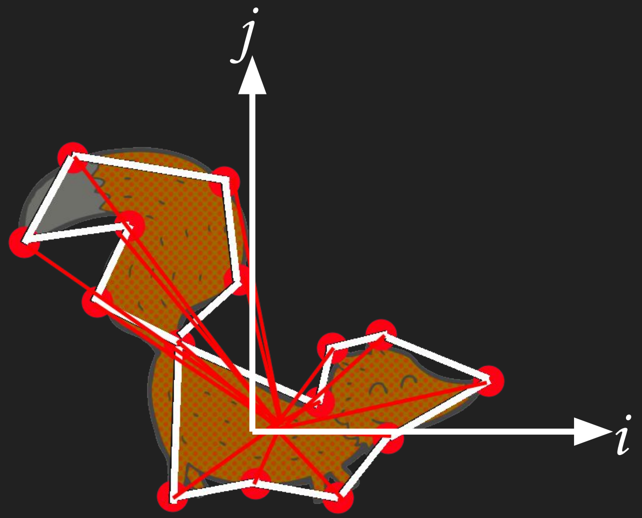


Local Space of the Fox Model

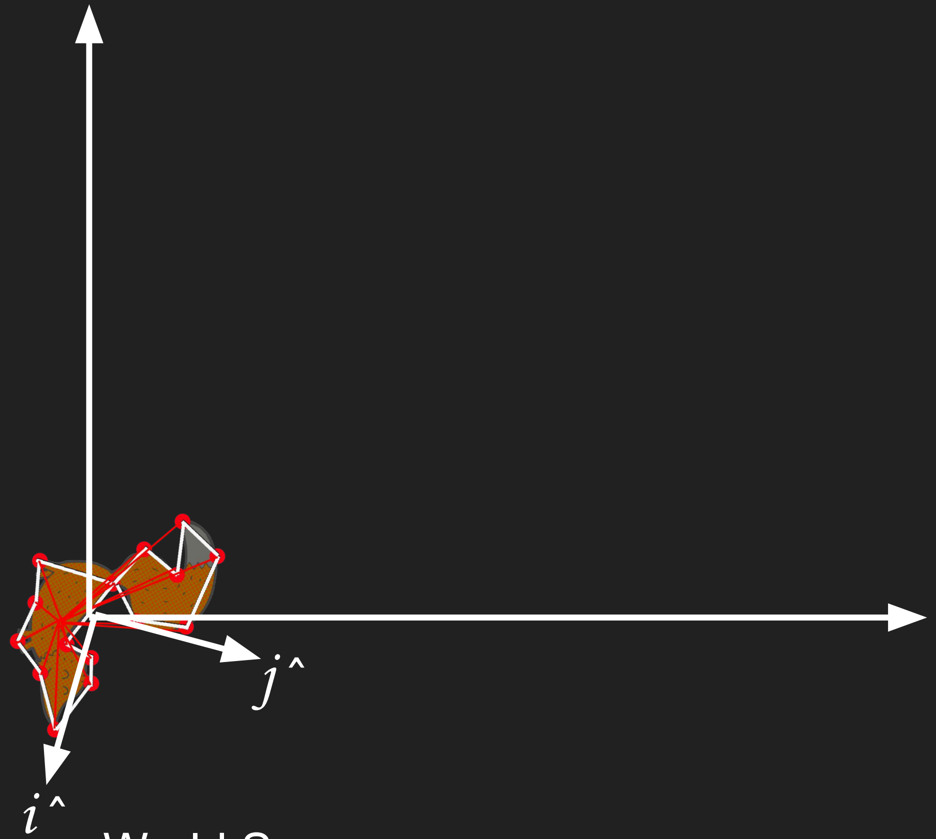


World Space

We have to do 3 steps
2: Rotate

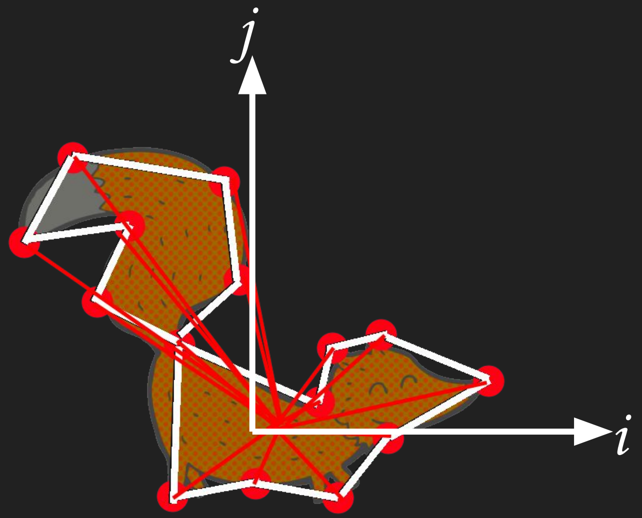


Local Space of the Fox Model

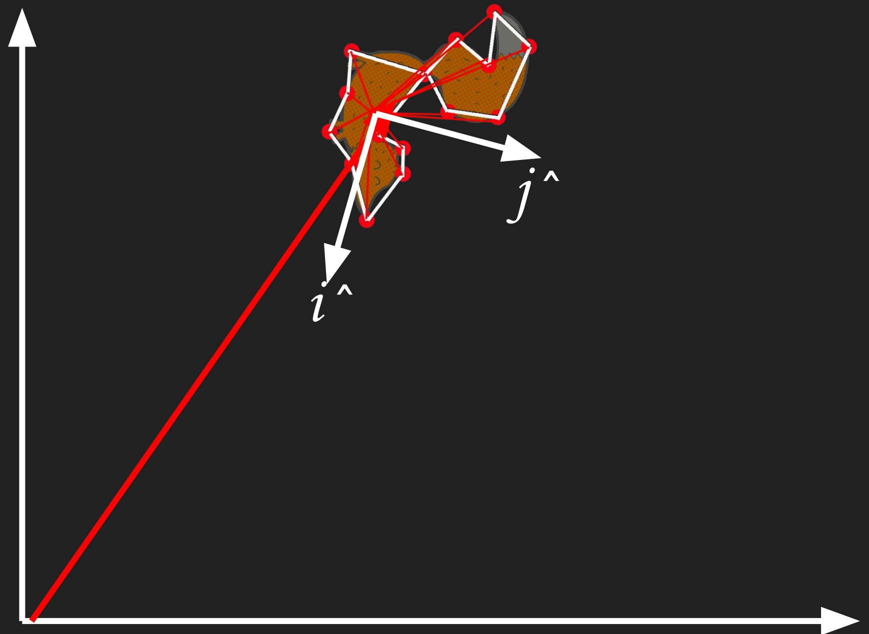


World Space

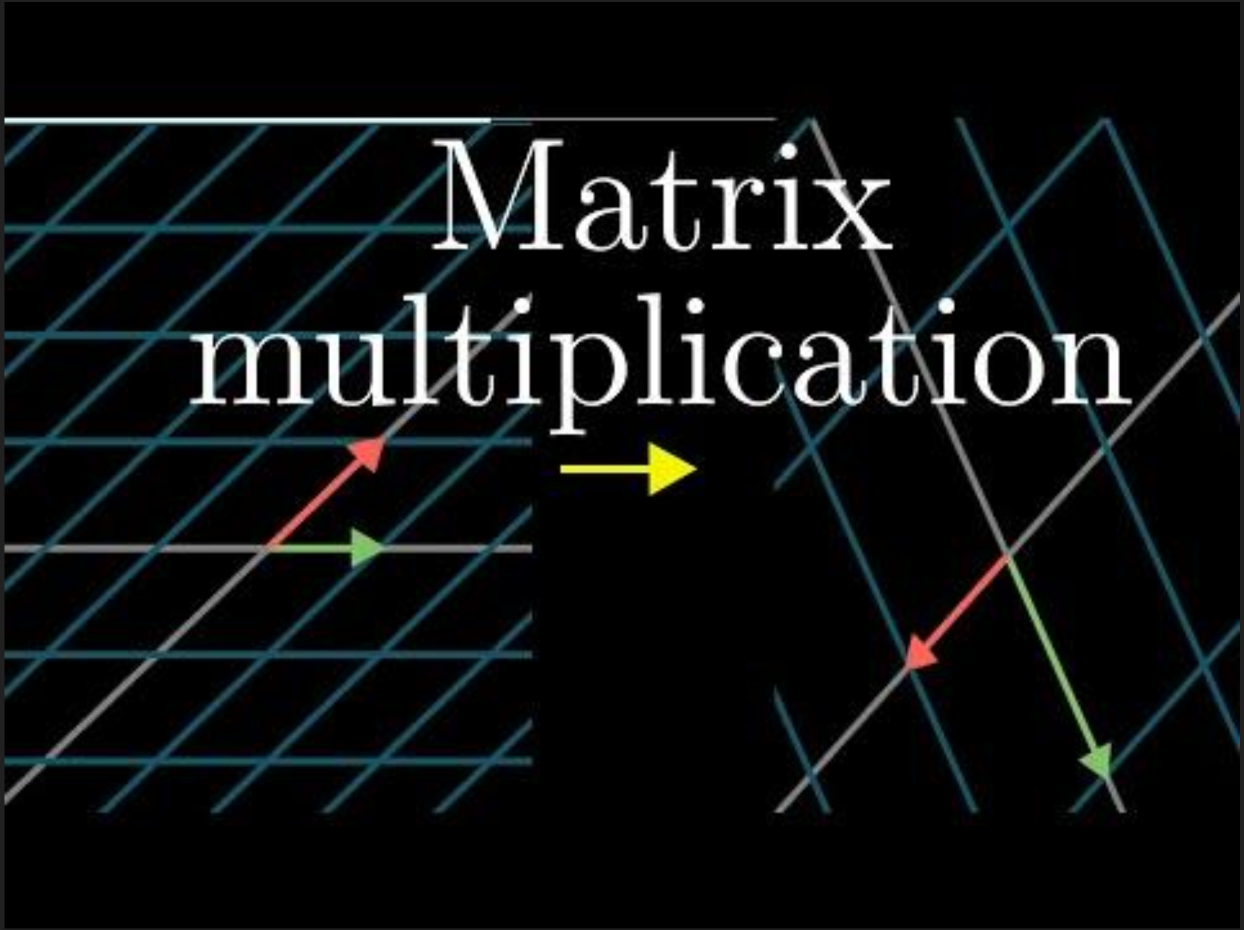
We have to do 3 steps
3: Translate



Local Space of the Fox Model

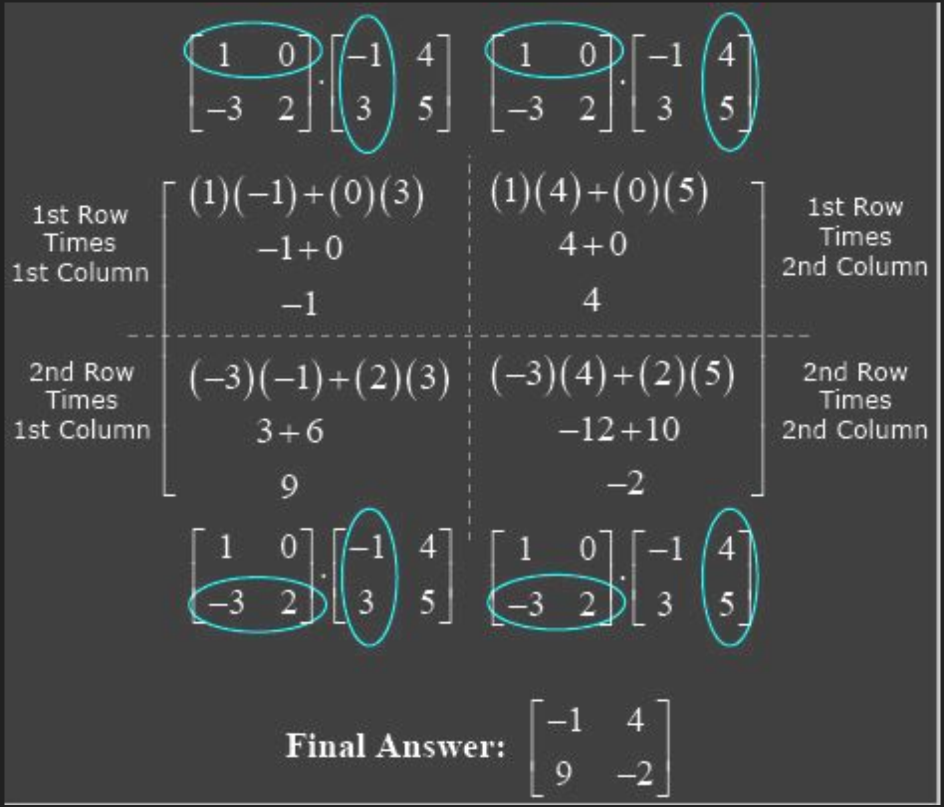


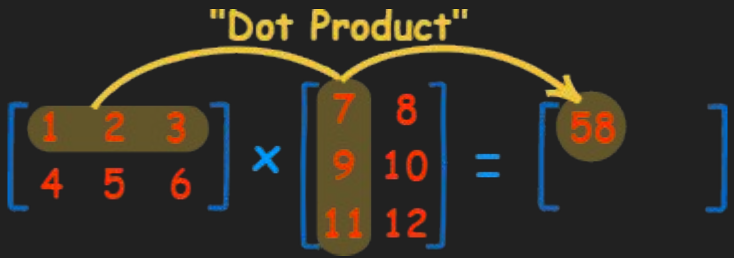
World Space



https://www.youtube.com/watch?time_continue=1&v=XkY2DOUCWMU&feature=emb_logo

Matrix multiplication



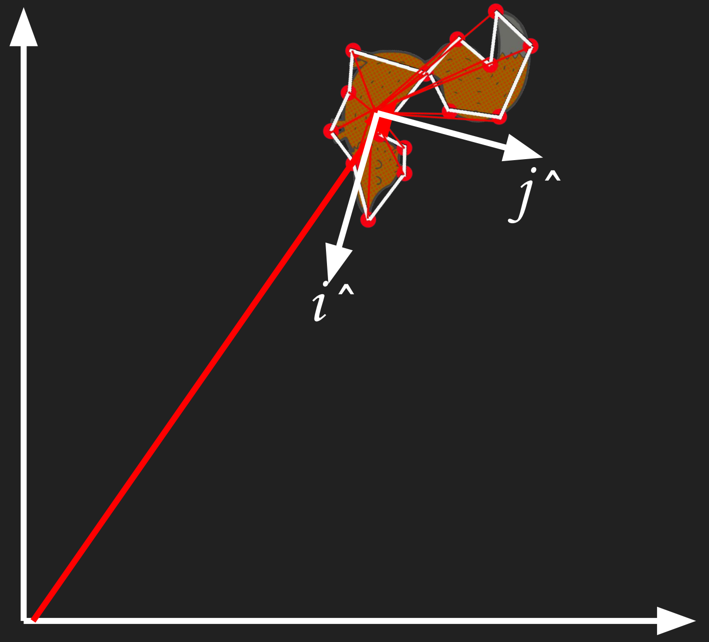


Transform a Vector by a Matrix

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

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

Because of the **Translation,** we actually need a 3x3 Matrix for 2D transformations.



The holy trinity of Linear Transformations

The scaling Matrix

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation Matrix

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The translation Matrix

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

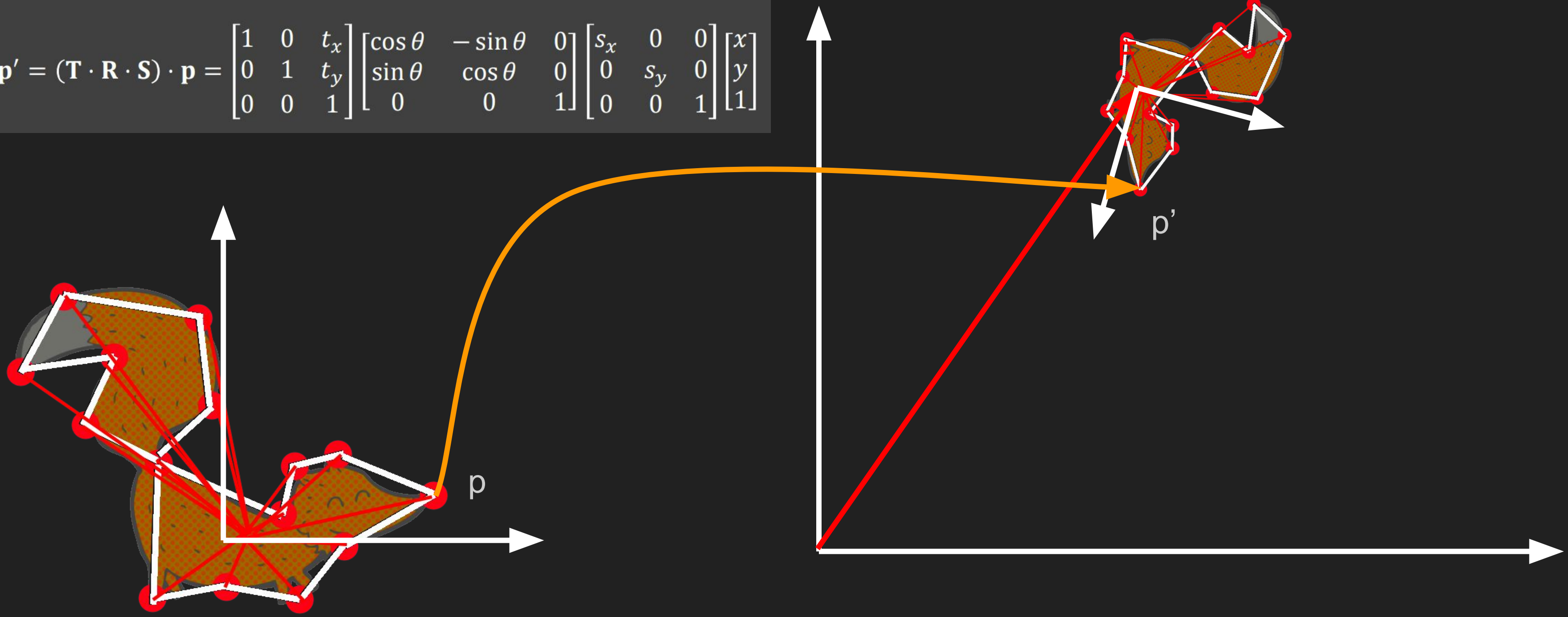
All together:

$$\mathbf{p}' = (\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S}) \cdot \mathbf{p} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ = The Transform Matrix of our object

Transformation to World Space using Matrix

$$\mathbf{p}' = (\mathbf{T}\cdot\mathbf{R}\cdot\mathbf{S})\cdot\mathbf{p} = \begin{bmatrix}1 & 0 & t_x\\ 0 & 1 & t_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}s_x & 0 & 0\\ 0 & s_y & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ 1\end{bmatrix}$$



The model matrix in WebGL

Model matrix: Matrix to transform the vertices of a model from local to world space

3x3 model matrix in the vertex shader:

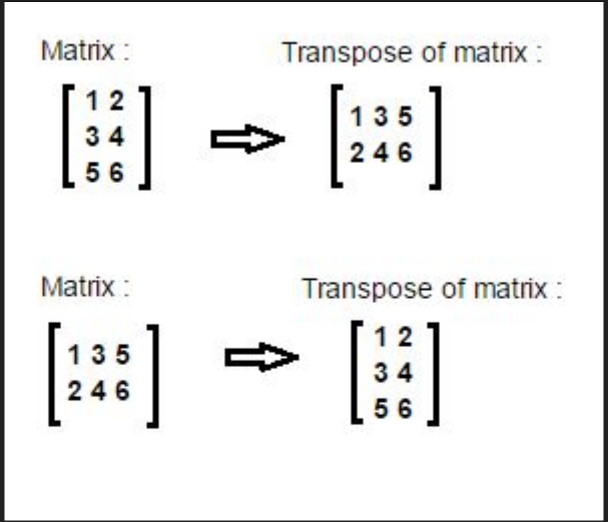
```
attribute vec2 a_position;
uniform mat3 u_modelMatrix;
void main() {
    vec3 pos = u_modelMatrix * vec3(a_position, 1);
    gl_Position = vec4(pos, 1);
}
```

Setting the data for the uniform in the draw() or render() functions:

```
let modelMatrixLocation = gl.getUniformLocation(shaderProgram, "u_modelMatrix");
gl.uniformMatrix3fv(modelMatrixLocation, false, this.modelMatrix.toFloat32());
```

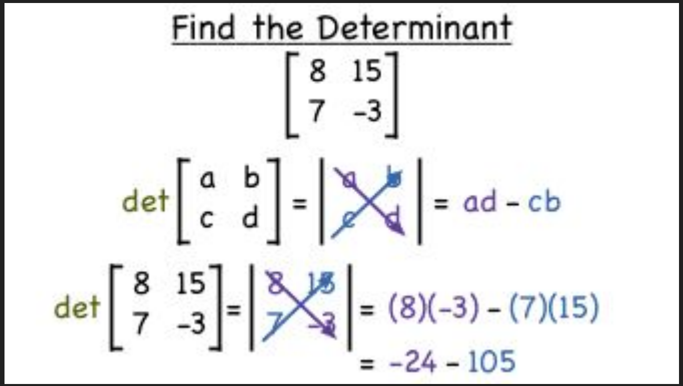
Matrix Transpose

A transpose of a matrix means flipping its rows and columns (useful for more advanced calculations)



Matrix determinants

In linear algebra, the determinant is a scalar value that can be computed from the elements of a square matrix and encodes certain properties of the linear transformation described by the matrix.



Matrix determinants 3x3

Find the determinant of each 2x2 submatrix

For a 3×3 matrix (3 rows and 3 columns):

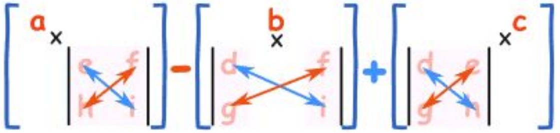
$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

The determinant is:

$$|A| = a(ei - fh) - b(di - fg) + c(dh - eg)$$

"The determinant of A equals ... etc"

It may look complicated, but **there is a pattern**:



$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}$$

$$= -\left(a_1 \begin{vmatrix} c_2 & b_2 \\ c_3 & b_3 \end{vmatrix} + b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} - c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}\right) = -\begin{vmatrix} a_1 & c_1 & b_1 \\ a_2 & c_2 & b_2 \\ a_3 & c_3 & b_3 \end{vmatrix}$$

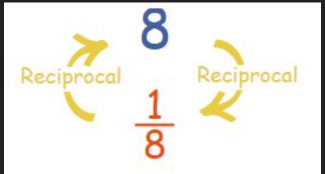
$$= -\left(-a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} + b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} b_2 & a_2 \\ b_3 & a_3 \end{vmatrix}\right) = -\begin{vmatrix} b_1 & a_1 & c_1 \\ b_2 & a_2 & c_2 \\ b_3 & a_3 & c_3 \end{vmatrix}$$

$$= -\left(a_1 \begin{vmatrix} c_2 & b_2 \\ c_3 & b_3 \end{vmatrix} - b_1 \begin{vmatrix} c_2 & a_2 \\ c_3 & a_3 \end{vmatrix} + c_1 \begin{vmatrix} b_2 & a_2 \\ b_3 & a_3 \end{vmatrix}\right) = -\begin{vmatrix} c_1 & b_1 & a_1 \\ c_2 & b_2 & a_2 \\ c_3 & b_3 & a_3 \end{vmatrix}.$$

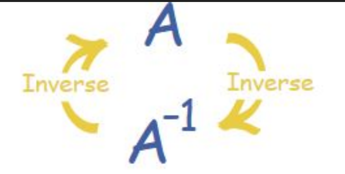
The determinant can be used to calculate the inverse of a Matrix

The inverse of a matrix can be found by other means (eg. Gaussian Elimination), however using the determinant is the best for use in programming.

An inversion of a Matrix, multiplied by the original always results in the identity Matrix:



$8 \times (1/8) = 1$



$A^{-1} \times A = \mathbf{I}$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix 3x3 inversion using Determinant and Adjugate

The Inverse of a Matrix can be gained by dividing its Adjugate with its determinant.

The Adjugate is the Transpose of its Cofactor Matrix.

The Cofactor Matrix can be optained using the Matrix of Minors.

The Matrix of Minors is a 3x3 Matrix of the 9 determinants of the 2x2 sub-Matrices of the 3x3 Matrix.

$$A^{-1} = \frac{1}{\det(A)} adj(A)$$

Or

$$A^{-1} = \frac{1}{\det(A)} (cofactor\ of\ matrix\ A(A))^T$$

Matrix 3x3 Adjugate

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

For this example, we shall consider a 3x3 matrix named A

$$\operatorname{adj}(\mathbf{A}) = \begin{pmatrix} +\begin{vmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{vmatrix} & -\begin{vmatrix} A_{12} & A_{13} \\ A_{32} & A_{33} \end{vmatrix} & +\begin{vmatrix} A_{12} & A_{13} \\ A_{22} & A_{23} \end{vmatrix} \\ -\begin{vmatrix} A_{21} & A_{23} \\ A_{31} & A_{33} \end{vmatrix} & +\begin{vmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{vmatrix} & -\begin{vmatrix} A_{11} & A_{13} \\ A_{21} & A_{23} \end{vmatrix} \\ +\begin{vmatrix} A_{21} & A_{22} \\ A_{31} & A_{32} \end{vmatrix} & -\begin{vmatrix} A_{11} & A_{12} \\ A_{31} & A_{32} \end{vmatrix} & +\begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} \end{pmatrix} = \begin{pmatrix} +\begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} & -\begin{vmatrix} 2 & 3 \\ 8 & 9 \end{vmatrix} & +\begin{vmatrix} 2 & 3 \\ 5 & 6 \end{vmatrix} \\ -\begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} & +\begin{vmatrix} 1 & 3 \\ 7 & 9 \end{vmatrix} & -\begin{vmatrix} 1 & 3 \\ 4 & 6 \end{vmatrix} \\ +\begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix} & -\begin{vmatrix} 1 & 2 \\ 7 & 8 \end{vmatrix} & +\begin{vmatrix} 1 & 2 \\ 4 & 5 \end{vmatrix} \end{pmatrix}$$

A's adjutant matrix

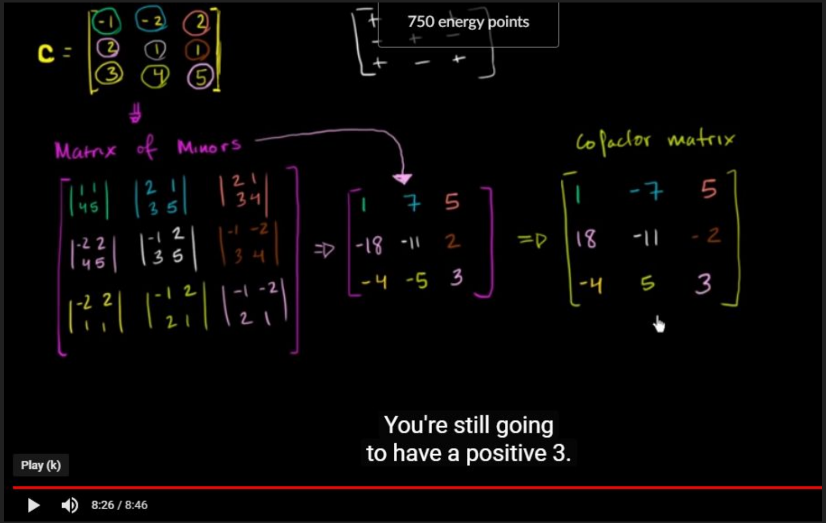
Matrix 3x3 Inversion

$$A^{-1} = \frac{1}{\det(A)} adj(A)$$

Or

$$A^{-1} = \frac{1}{\det(A)} (cofactor\ of\ matrix\ A(A))^T$$

Matrix 3x3 inversion



https://www.khanacademy.org/math/algebra-home/alg-matrices/alg-determinants-and-inverses-of-large-matrices/v/inverting-3x3-part-1-calculating-matrix-of-minors-and-cofactor-matrix