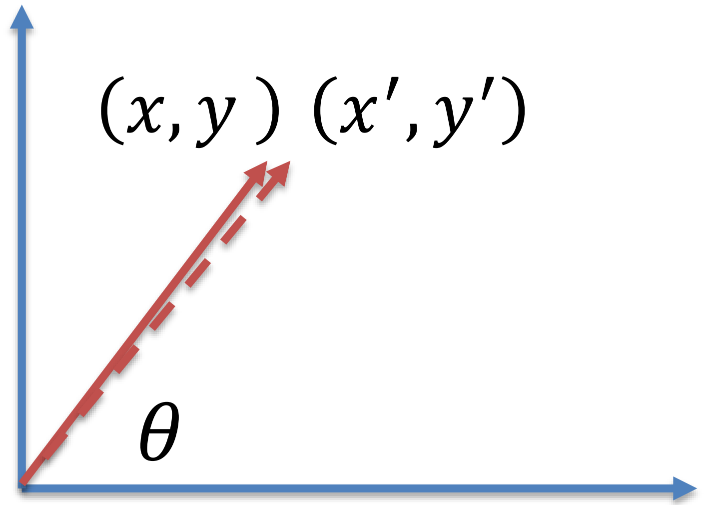
Deformation Matrix and Spatial Reasoning

John R. Williams and Abel Sanchez
MIT 1.00

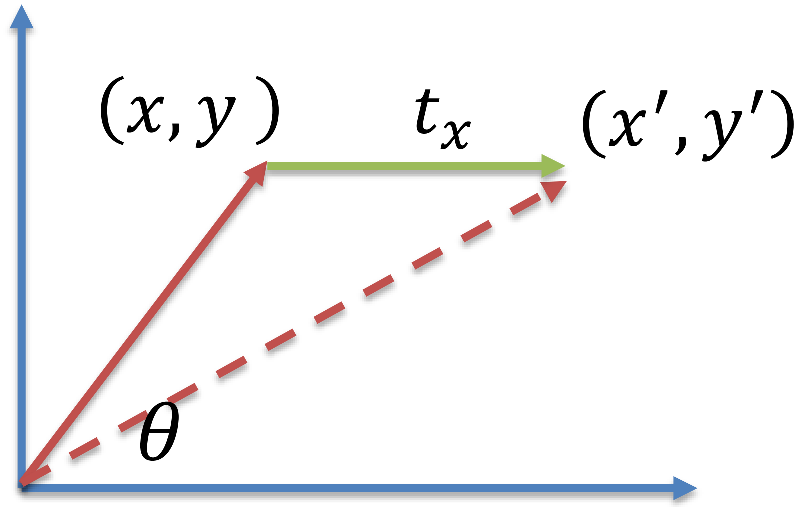
Idenity

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



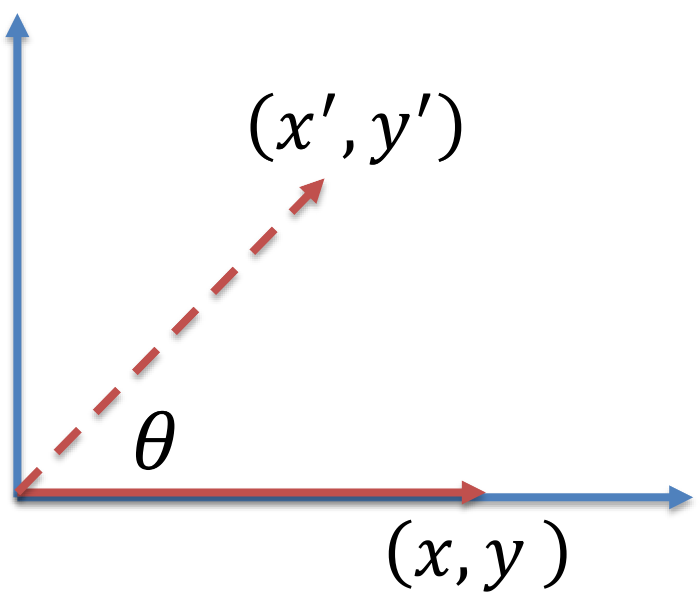
Translate a Point

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} + \begin{Bmatrix} t_x \\ t_y \end{Bmatrix}$$



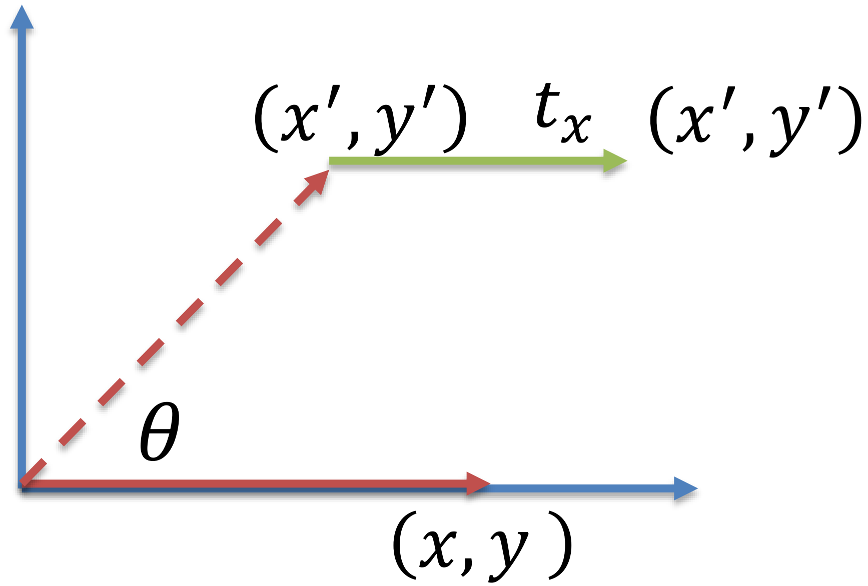
Rotate a Point

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



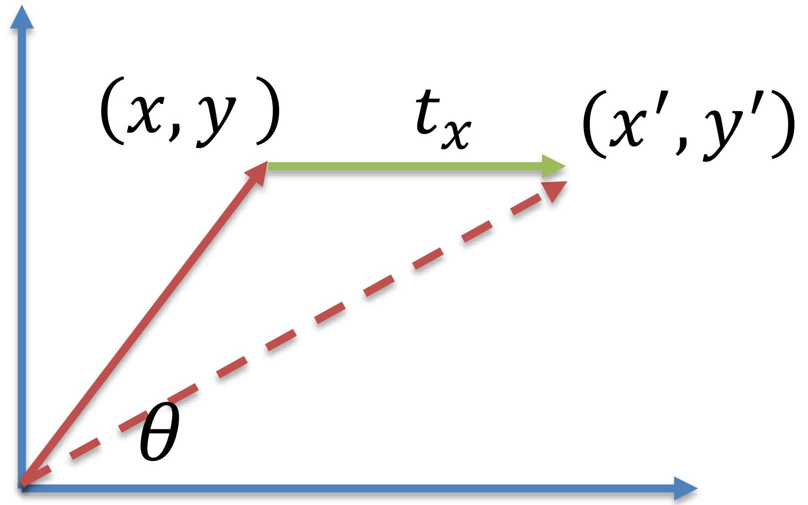
Rotate a Point and Translate it

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} + \begin{Bmatrix} t_x \\ t_y \end{Bmatrix}$$



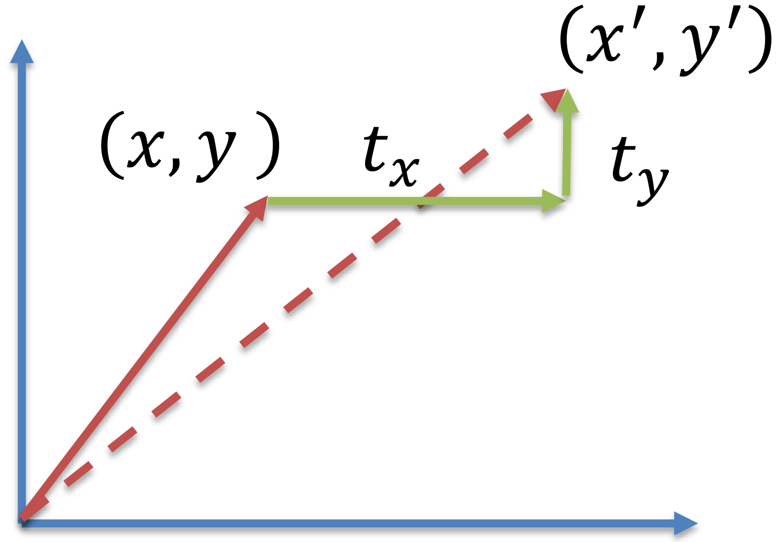
Translate a Point

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



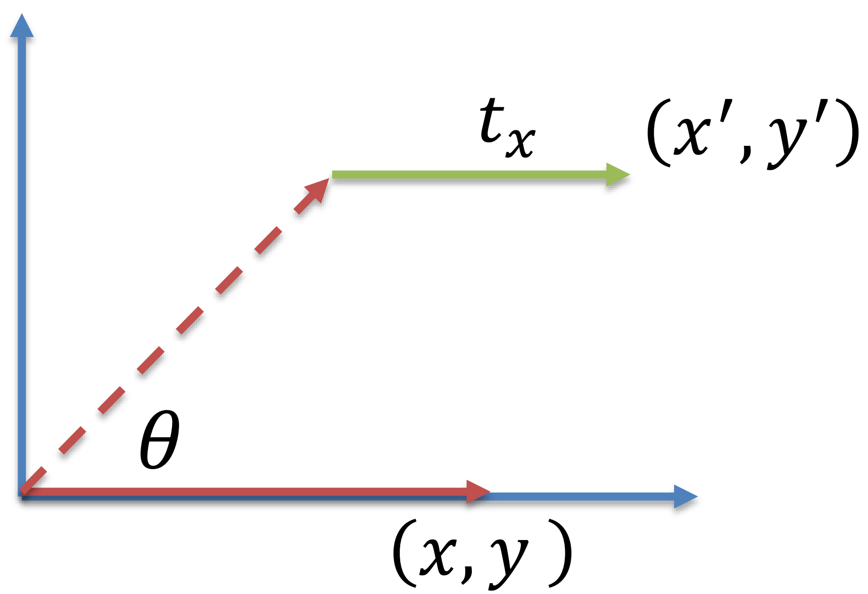
Translate a Point

$$\begin{Bmatrix} x' \\ y' \\ 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$



Rotate + Translate

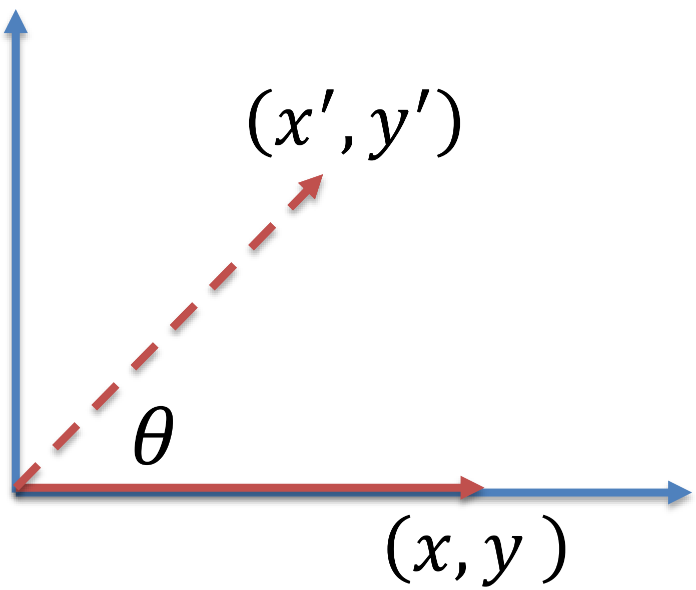
$$\begin{Bmatrix} x' \\ y' \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$



Rotate + Translate

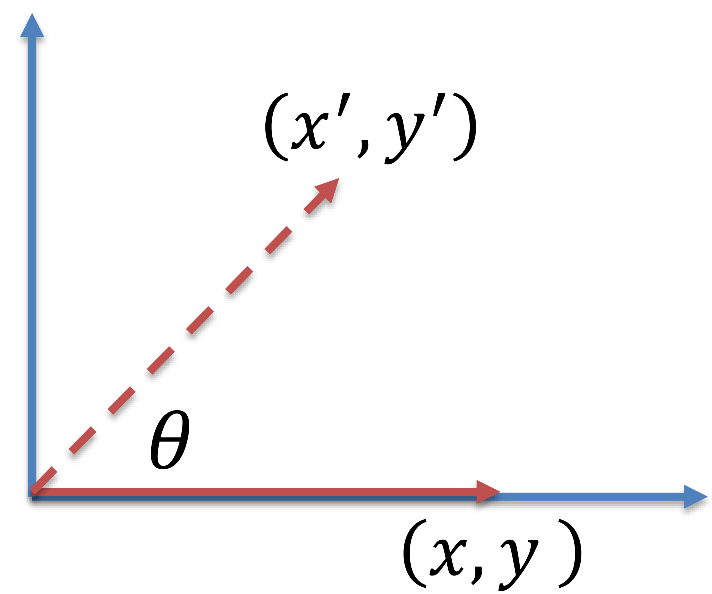
$$\begin{Bmatrix} x' \\ y' \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$

$x = 1, \mathrm{y} = 0, \theta = \frac{\pi}{4}, t_x$=1,



T Matrix- Defines Everything

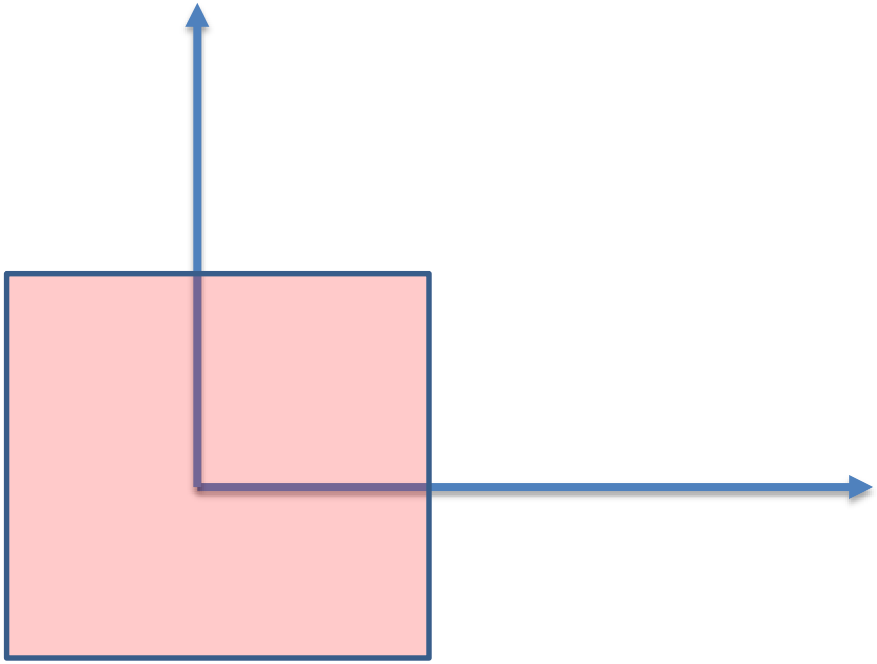
$$[T] = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



In Class - Rotate + Translate Square

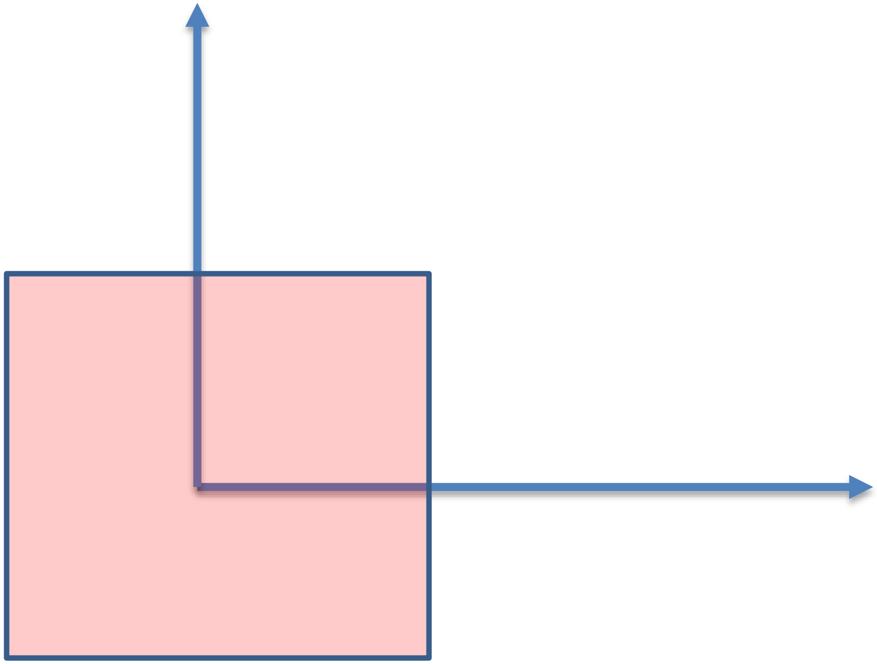
$$\begin{Bmatrix} x' \\ y' \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$

$\theta = \frac{\pi}{4}, t_x$=0



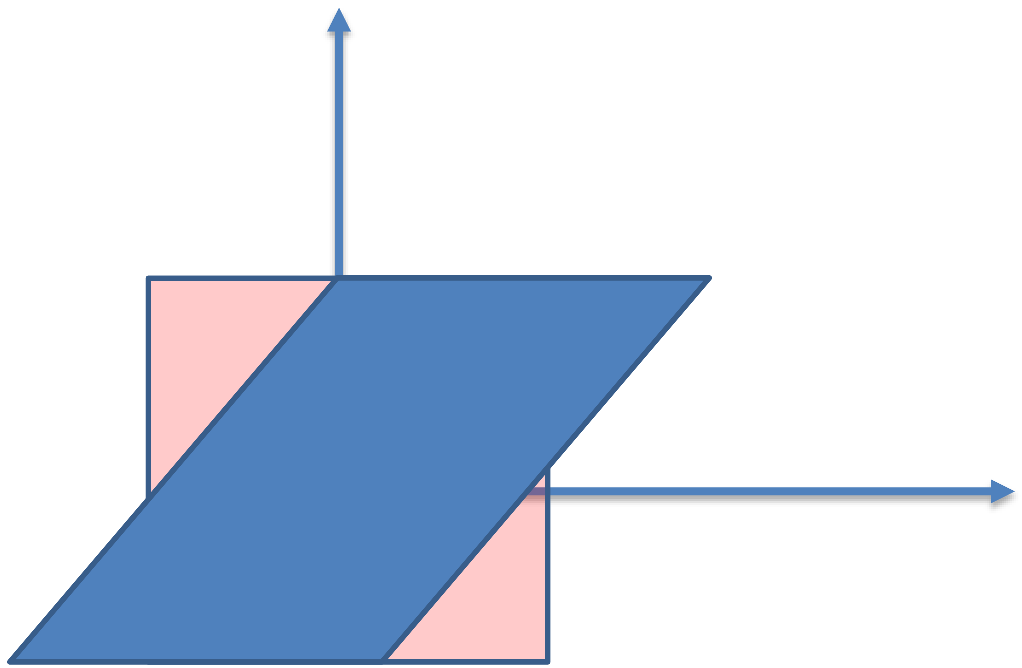
In Class – Draw the Result

$$T=\begin{bmatrix} 1 & -0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



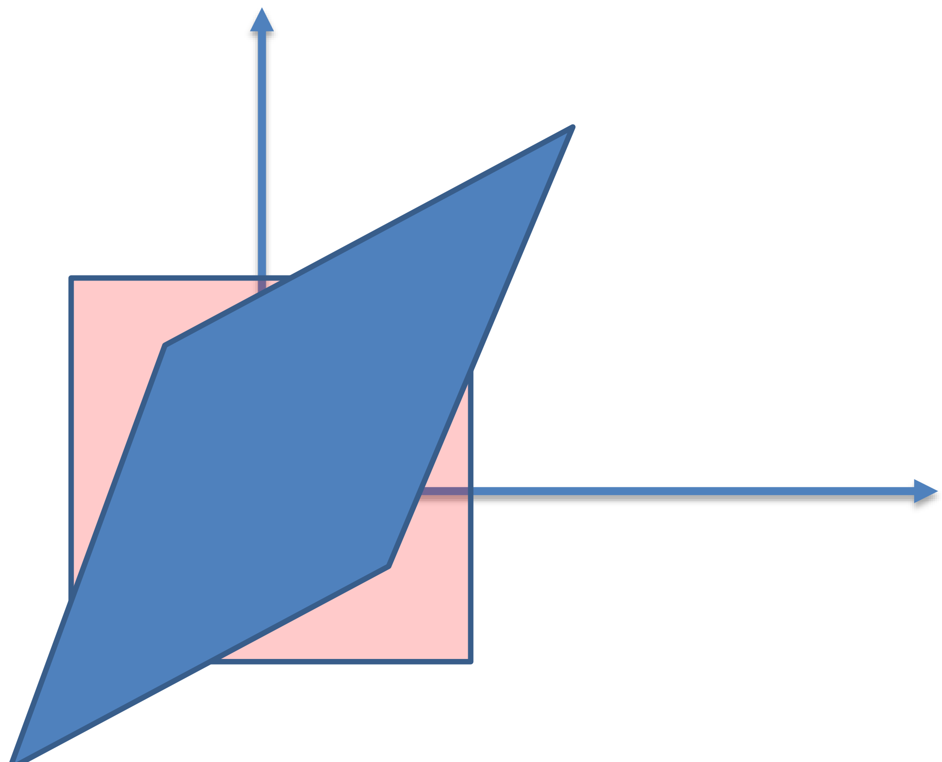
In Class – Find the Matrix which makes the figure look like

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



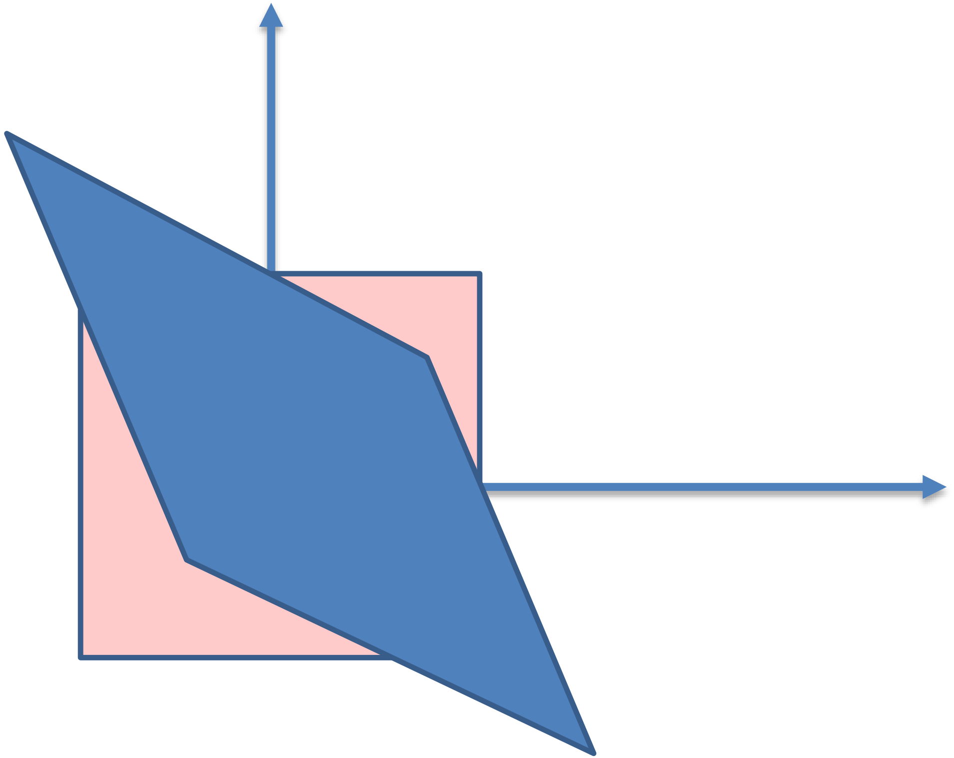
In Class – Find the Matrix which makes the figure look like

$$T=\begin{bmatrix} ? & ? & 200 \\ ? & ? & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

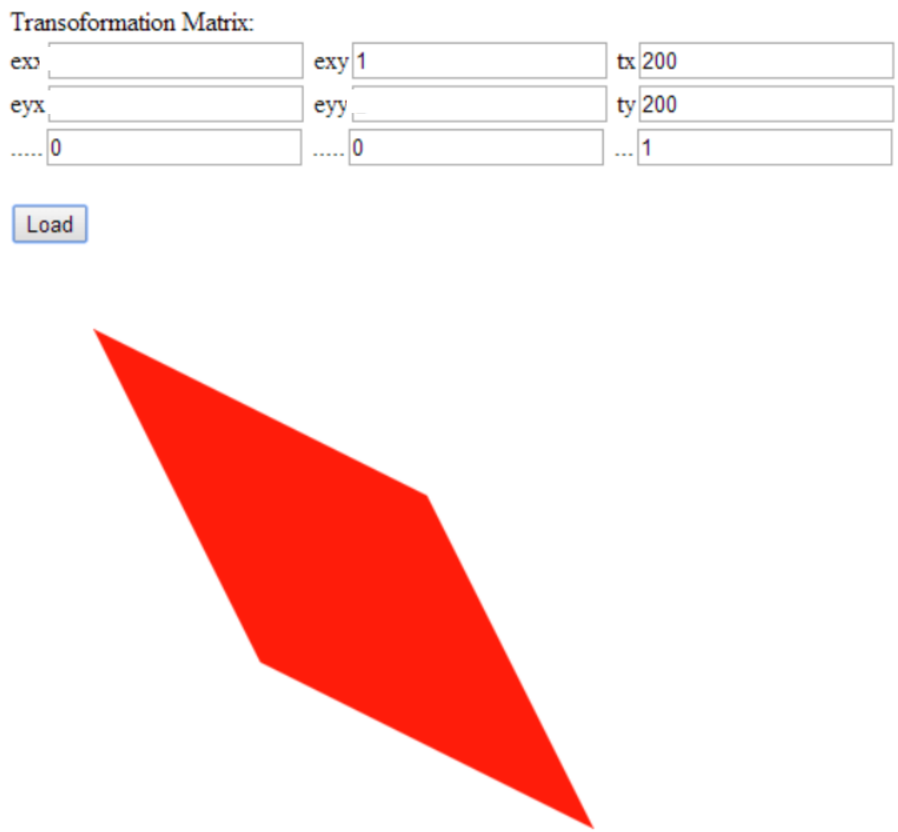


In Class – Find the Matrix which makes the figure look like

$$T=\begin{bmatrix} ? & ? & 200 \\ ? & ? & 200 \\ 0 & 0 & 1 \end{bmatrix}$$



In Class – Check your results with the Starter Code



Hand in screen shots of the figures and the Matrix similar to above.

Animate Figure

```
function Repeat(){
  setInterval(Run,100);
}
```

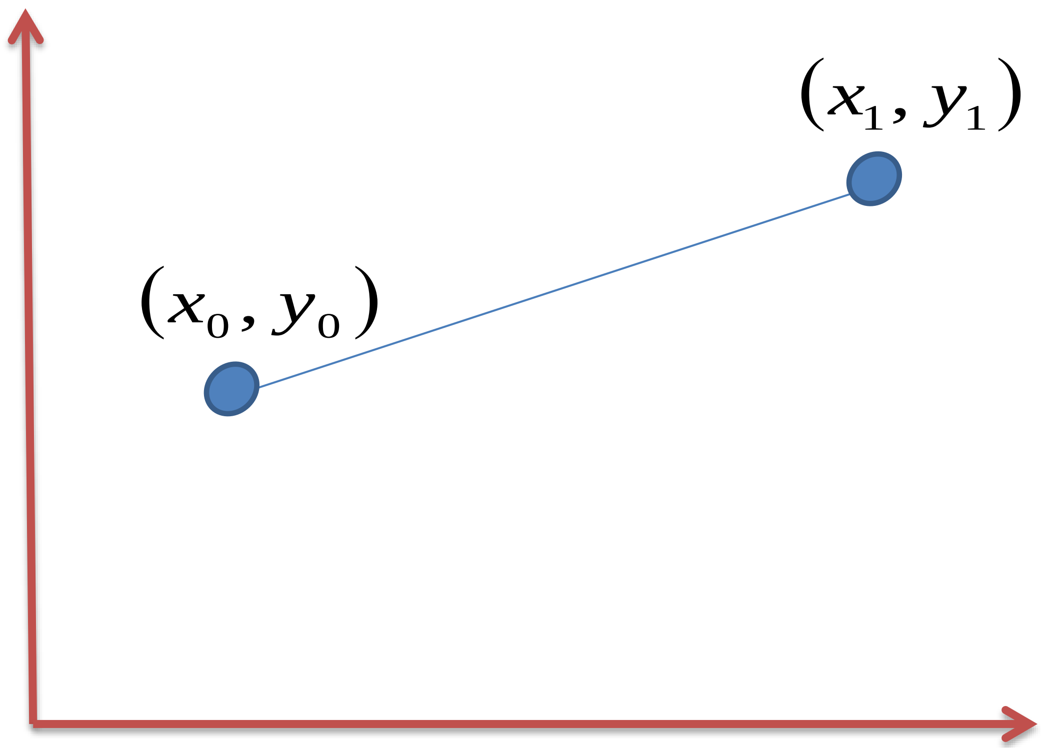
```
function Run() {
  GetInput();
  context.clearRect(0, 0, WIDTH, HEIGHT);
  var t = [];
  var r;
  for( var i = 0; i < vertices3.length; i++ ) {
    v = vertices3[i];
    r = T.MultiplyBy(v);
    t.push(r);
  }
  DrawRect(t,'Red');
  vertices3 = t;

}
```

```
<input type="button" value="1st Step" onclick="Load()">
<input type="button" value="Repeat" onclick="Repeat()">
```

Animation Strategy

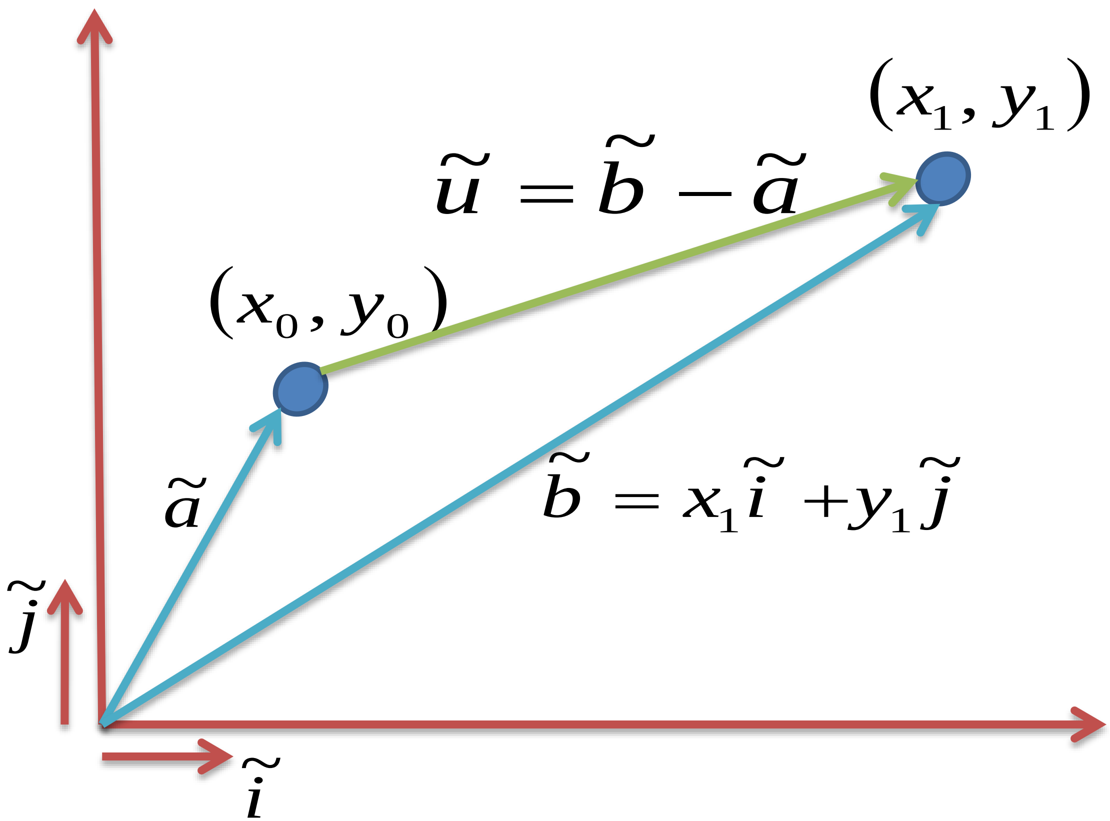
```
function GetRotationMatrix(rad){
  v1 = Vector(2*Math.cos(rad),-2*Math.sin(rad),tx);
  v2 = Vector(2*Math.sin(rad), 2*Math.cos(rad),ty);
  v3 = Vector(0,0,1);
  return Matrix(v1,v2,v3);
}
function GetStrainMatrix(rad){
  v1 = Vector(1+exx, exy*Math.sin(rad), tx);
  v2 = Vector(eyx*Math.sin(rad), 1+eyy, ty);
  v3 = Vector(0,0,1);
  return Matrix(v1,v2,v3);
}
function GetDilationMatrix(rad){
  v1 = Vector(1+ exx*Math.sin(rad),0, tx);
  v2 = Vector(0, 1 + eyy*Math.sin(rad), ty);
  v3 = Vector(0,0,1);
  return Matrix(v1,v2,v3);
}
function GetTranslationMatrix(rad){
  v1 = Vector(1+ exx,0, tx + 10*Math.cos(rad));
  v2 = Vector(0, 1 + eyy,ty + 10*Math.cos(rad));
  v3 = Vector(0,0,1);
  return Matrix(v1,v2,v3);
}
```

A Line is Defined by 2 Ordered Points



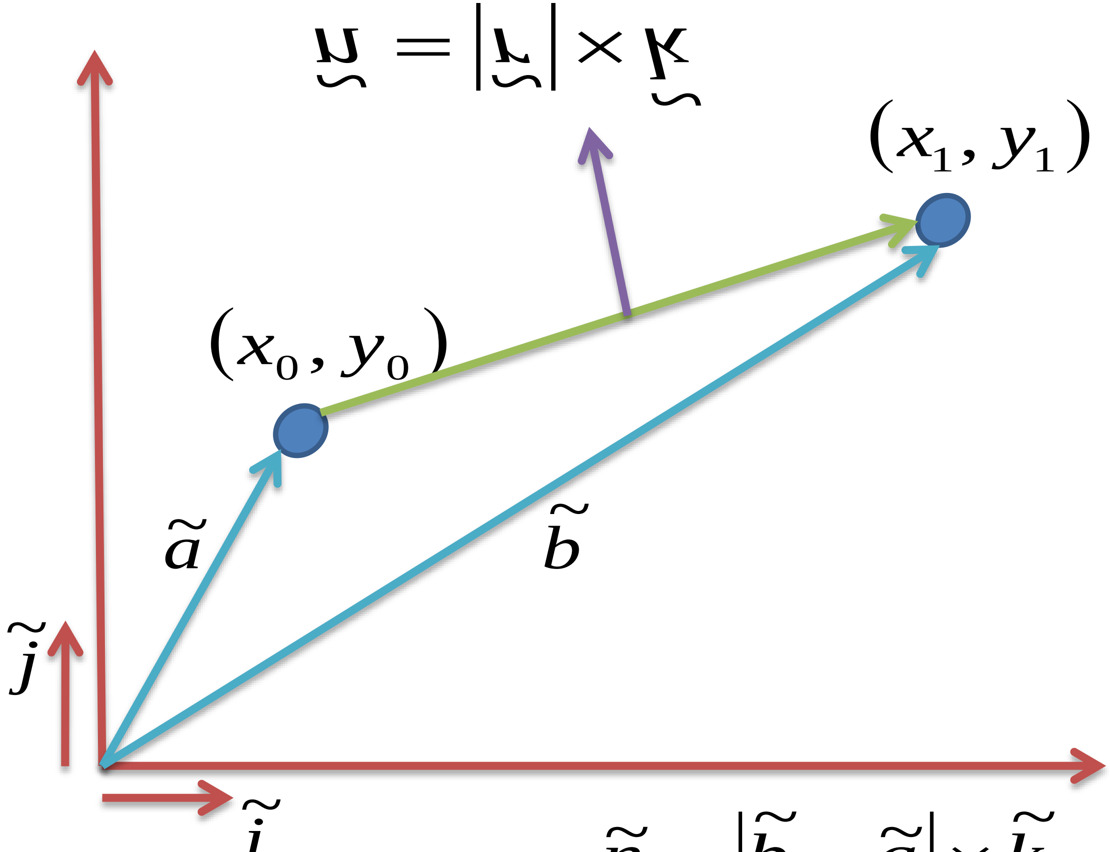
public Point[] points = new Point[2];

A Line is Defined by 2 Vectors



We take i and j as unit vectors

Normal to Line



$$\tilde{n} = \left|\tilde{b} - \tilde{a}\right| \times \tilde{k}$$

$$\tilde{n} = \left|(x_1\tilde{i} + y_1\tilde{j}) \times \tilde{k} - (x_0\tilde{i} + y_0\tilde{j}) \times \tilde{k}\right|$$

Normal to Line

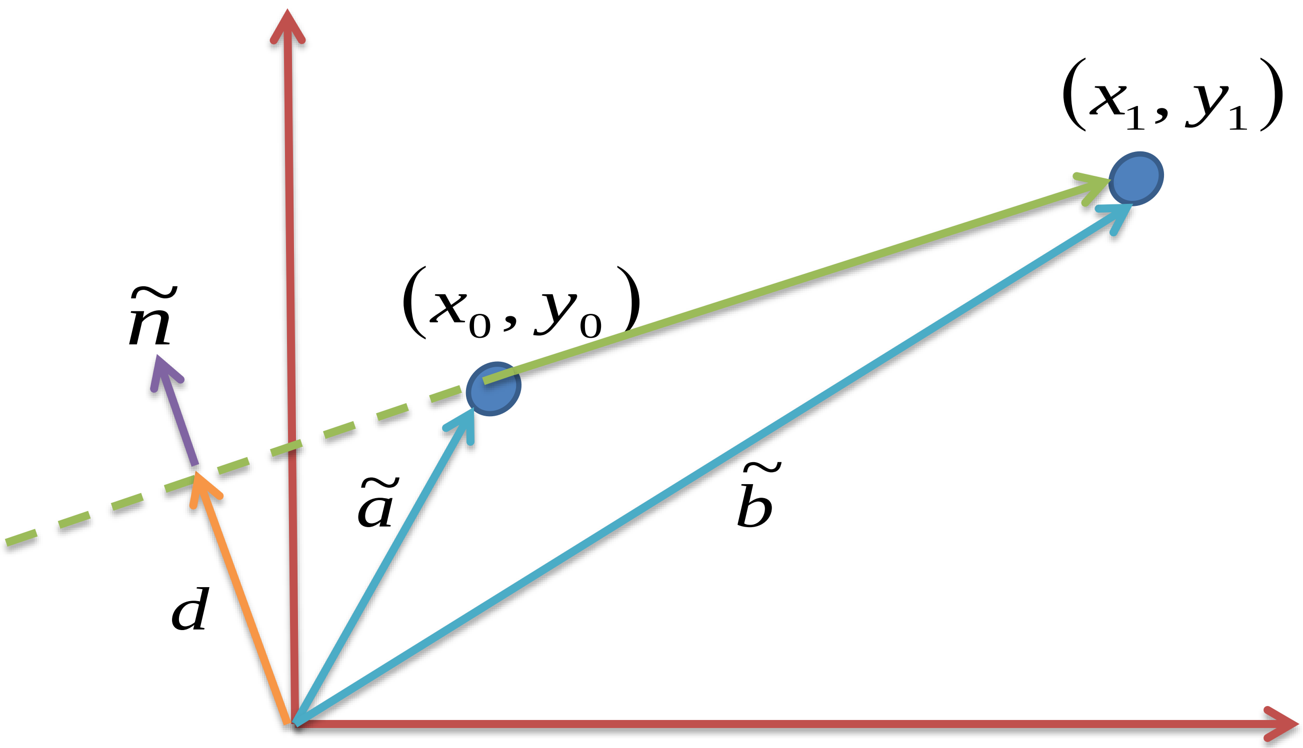
$$\tilde{n} = \left| \tilde{b} - \tilde{a} \right| \times \tilde{k}$$

$$\tilde{n} = \left| (x_1 \tilde{i} + y_1 \tilde{j}) \times \tilde{k} - (x_0 \tilde{i} + y_0 \tilde{j}) \times \tilde{k} \right|$$

$$\begin{bmatrix} i & j & k \\ x_1 & y_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = y_1 \tilde{i} - x_1 \tilde{j} \qquad \begin{bmatrix} i & j & k \\ x_0 & y_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = y_0 \tilde{i} - x_0 \tilde{j}$$

$$\tilde{n} = \left| (y_1 - y_0) \tilde{i} - (x_1 - x_0) \tilde{j} \right|$$

Perpendicular Distance from Origin to Line

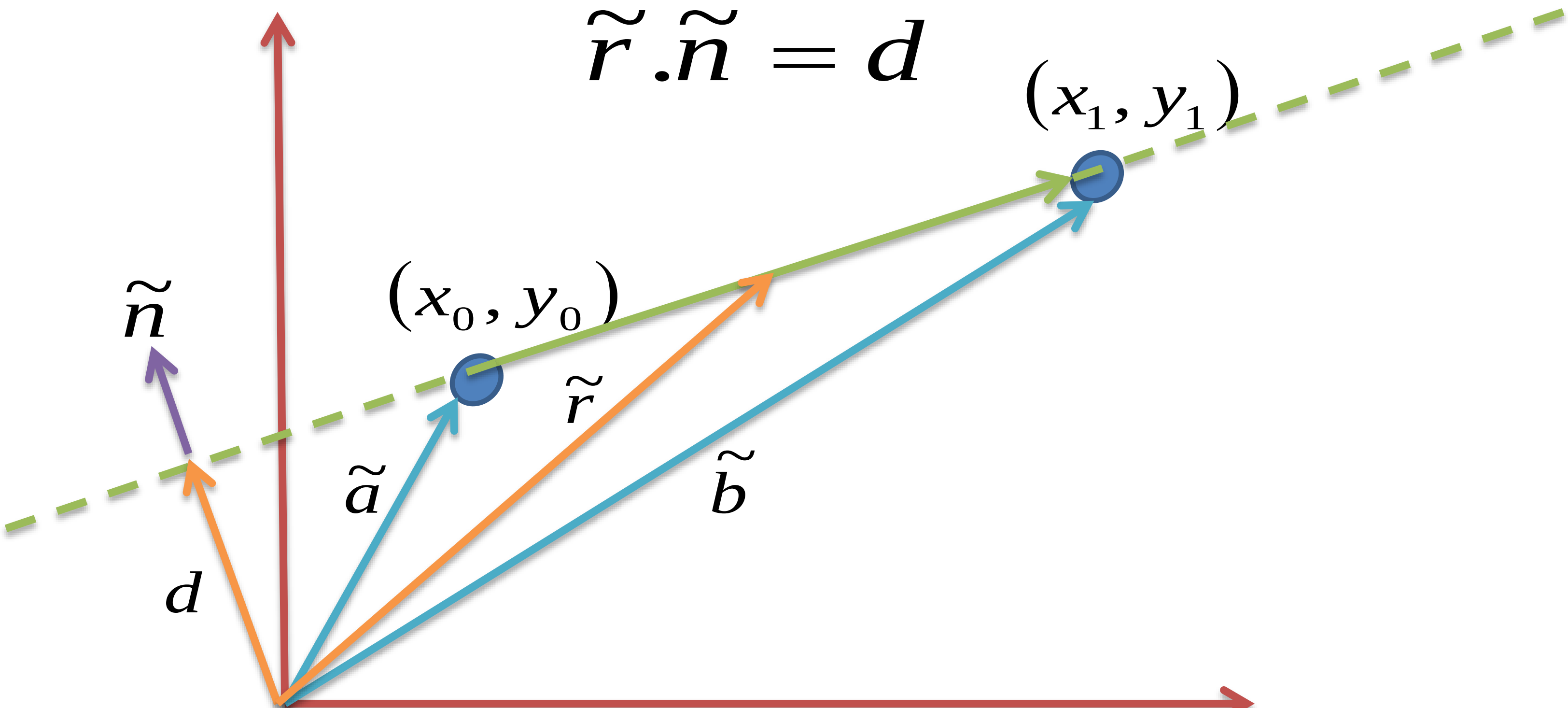


$$d = \tilde{a}.\tilde{n}$$

$$d = \tilde{b}.\tilde{n}$$

Vector Equation of an Infinite Line

$$\tilde{r}.\tilde{n} = d$$

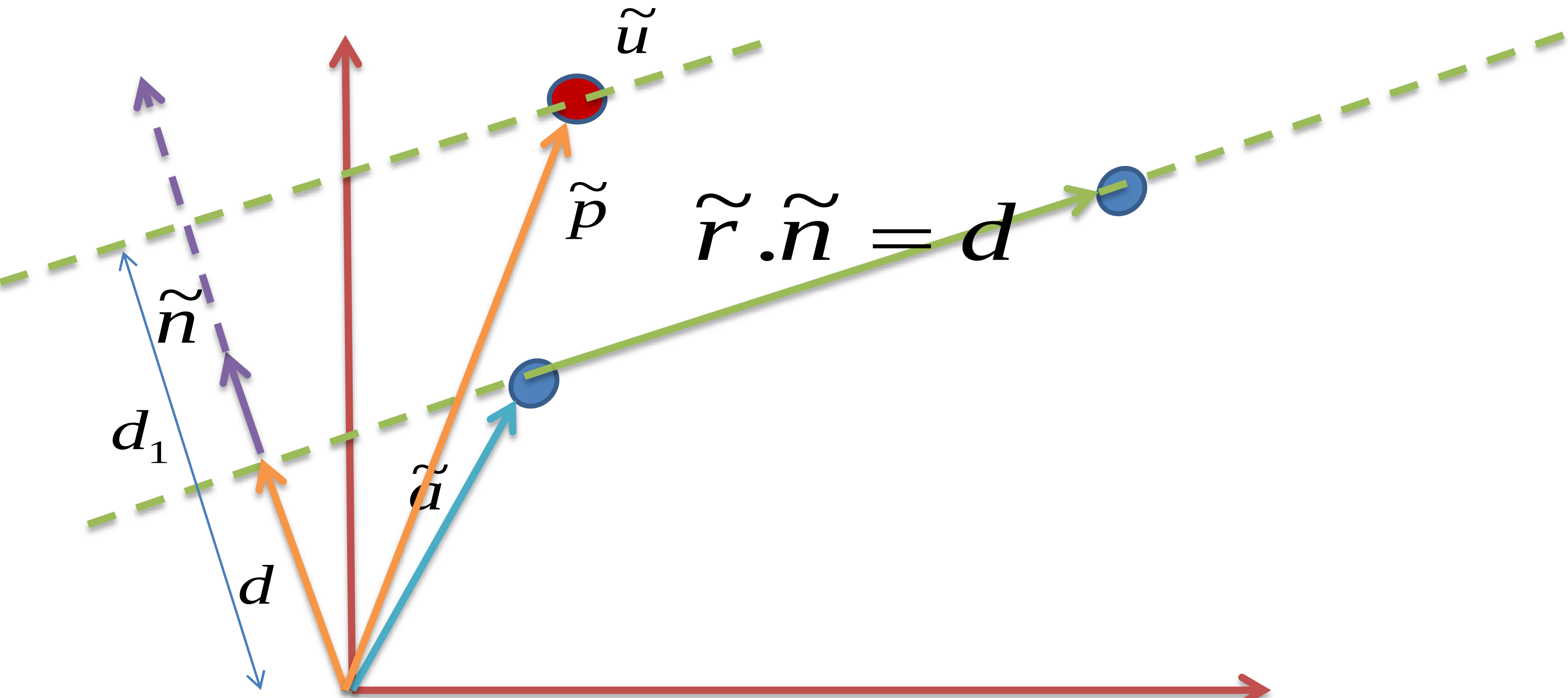


We know it is true for the end points. It is true for any point r

$$d = \tilde{a}.\tilde{n}$$

$$d = \tilde{b}.\tilde{n}$$

Point to Line



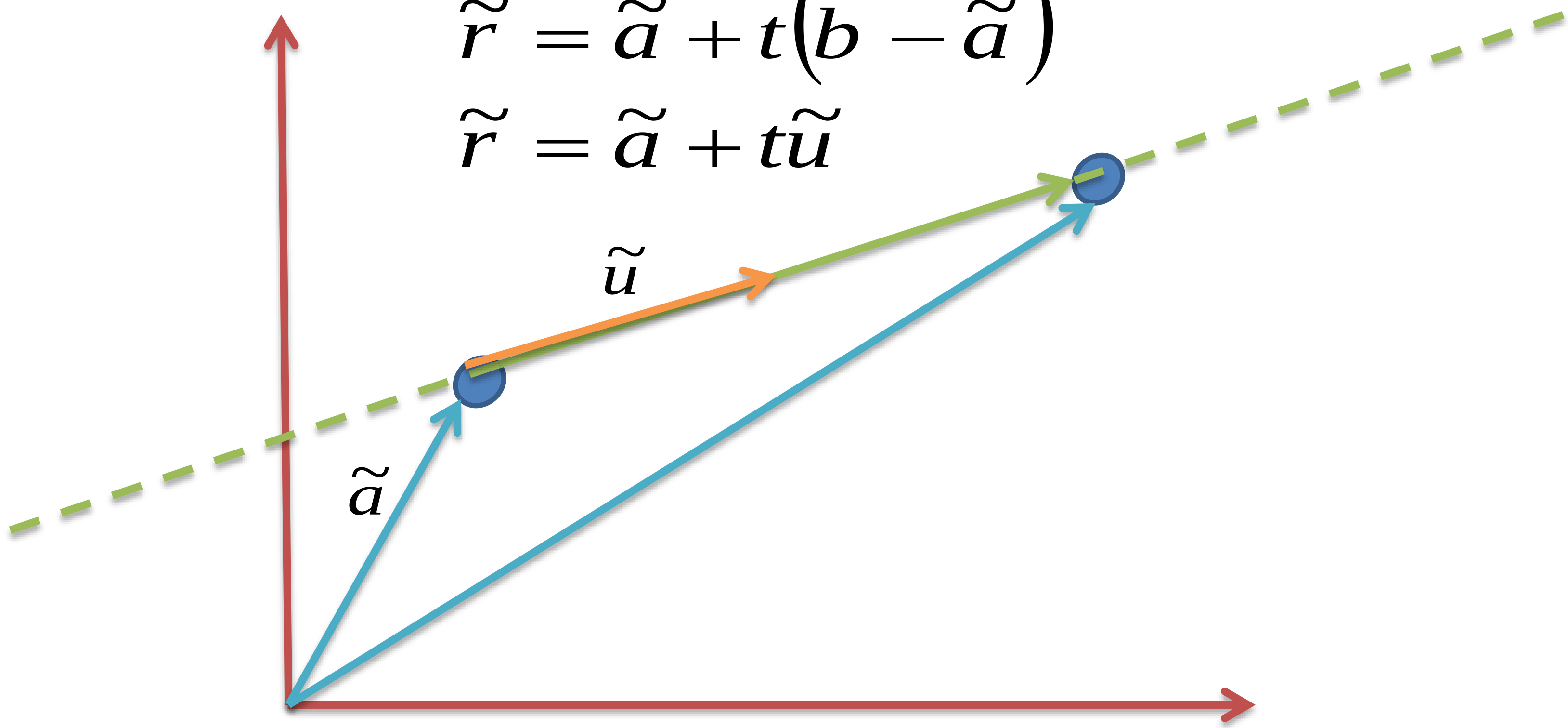
Line through p
parallel to given line

$$d_1 - d = \tilde{p}.\tilde{n} - \tilde{r}.\tilde{n} = (\tilde{p} - \tilde{r}).\tilde{n}$$

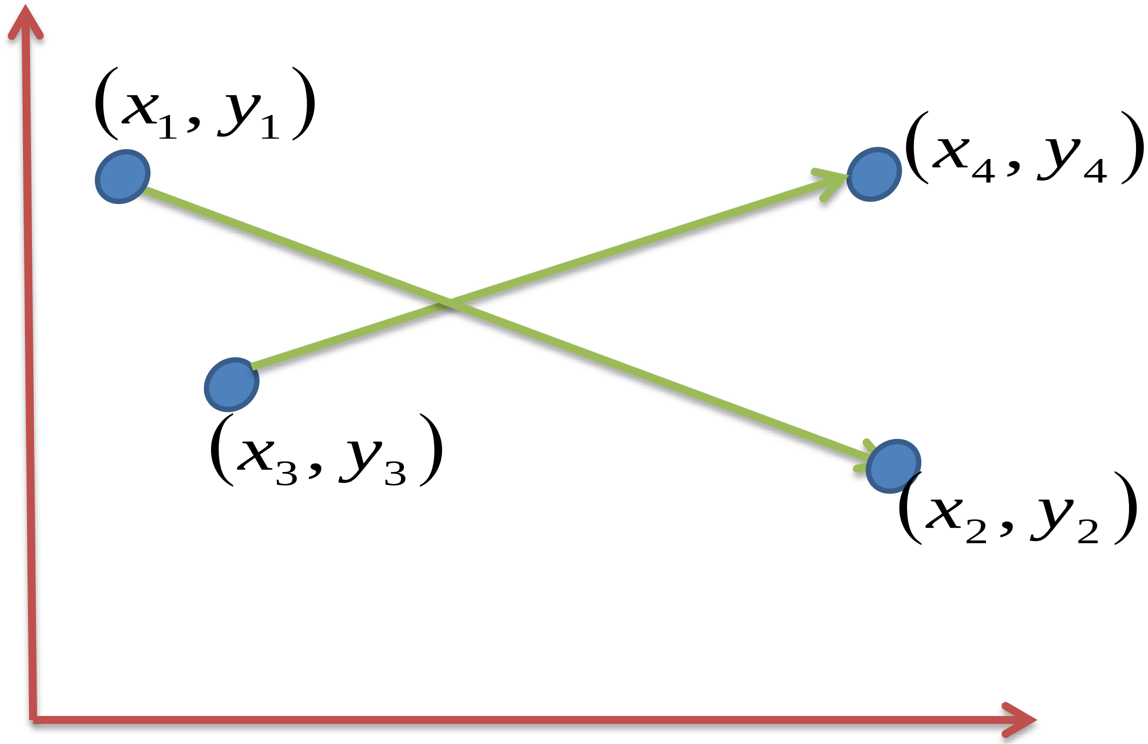
Parameterized Equation

$$\tilde{r} = \tilde{a} + t\left(\tilde{b} - \tilde{a}\right)$$

$$\tilde{r} = \tilde{a} + t\tilde{u}$$



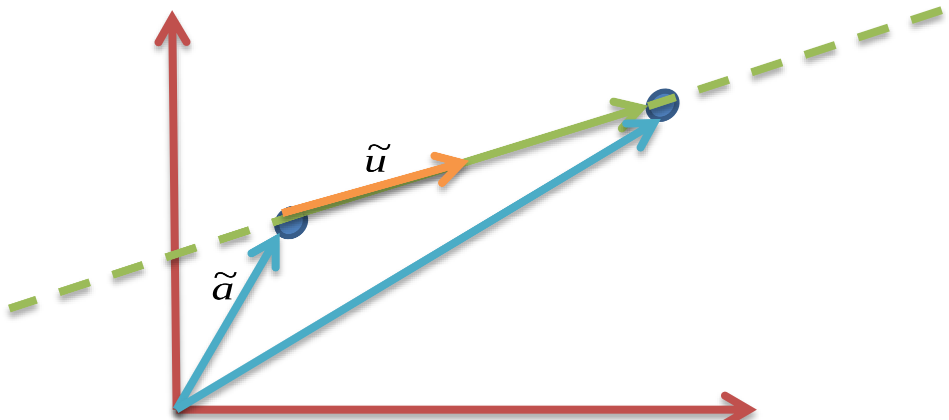
Line-Line Intersection



http://local.wasp.uwa.edu.au/~pbourke/geometry/lineline2d/

Intersection Math

$$\tilde{r}_1 = \tilde{a}_1 + t_1\left(\tilde{b}_1 - \tilde{a}_1\right)$$
$$\tilde{r}_2 = \tilde{a}_2 + t_2\left(\tilde{b}_2 - \tilde{a}_2\right)$$



At the intersection point r1 = r2 →

$$\tilde{a}_1 + t_1\left(\tilde{b}_1 - \tilde{a}_1\right) = \tilde{a}_2 + t_2\left(\tilde{b}_2 - \tilde{a}_2\right)$$

This gives us 2 equations for the unknowns t1 and t2

$$x_1 + t_1\left(x_2 - x_1\right) = x_3 + t_2\left(x_4 - x_3\right)$$
$$y_1 + t_1\left(y_2 - y_1\right) = y_3 + t_2\left(y_4 - y_3\right)$$

Standard Form and Soln Using Determinants

$$t_1(x_2 - x_1) - t_2(x_4 - x_3) = x_3 - x_1$$
$$t_1(y_2 - y_1) - t_2(y_4 - y_3) = y_3 - y_1$$

Standard form

$$ax + by + p = 0$$

$$cx + dy + q = 0$$

$$x = \frac{\left\| \begin{bmatrix} p & b \\ d & q \end{bmatrix} \right\|}{\left\| \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right\|} \qquad y = -\frac{\left\| \begin{bmatrix} a & p \\ c & q \end{bmatrix} \right\|}{\left\| \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right\|}$$

Intersection Point

$$t_1 = \left(\frac{(x_4 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_4 - y_3)}{(y_4 - y_3)(x_2 - x_1) - (y_2 - y_1)(x_4 - x_3)} \right)$$

$$t_2 = \left(\frac{(x_2 - x_1)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_1)}{(y_4 - y_3)(x_2 - x_1) - (y_2 - y_1)(x_4 - x_3)} \right)$$

The denominators for the equations for t1 and t2 are the same.
If the denominator for the equations for t1 and t2 is 0 then the two lines are parallel.
If the denominator and numerator for the equations for t1 and t2 are 0 then the two lines are coincident.

Solving by Elimination

$$t_1(x_2 - x_1) - t_2(x_4 - x_3) = x_3 - x_1$$

$$t_1(y_2 - y_1) - t_2(y_4 - y_3) = y_3 - y_1$$

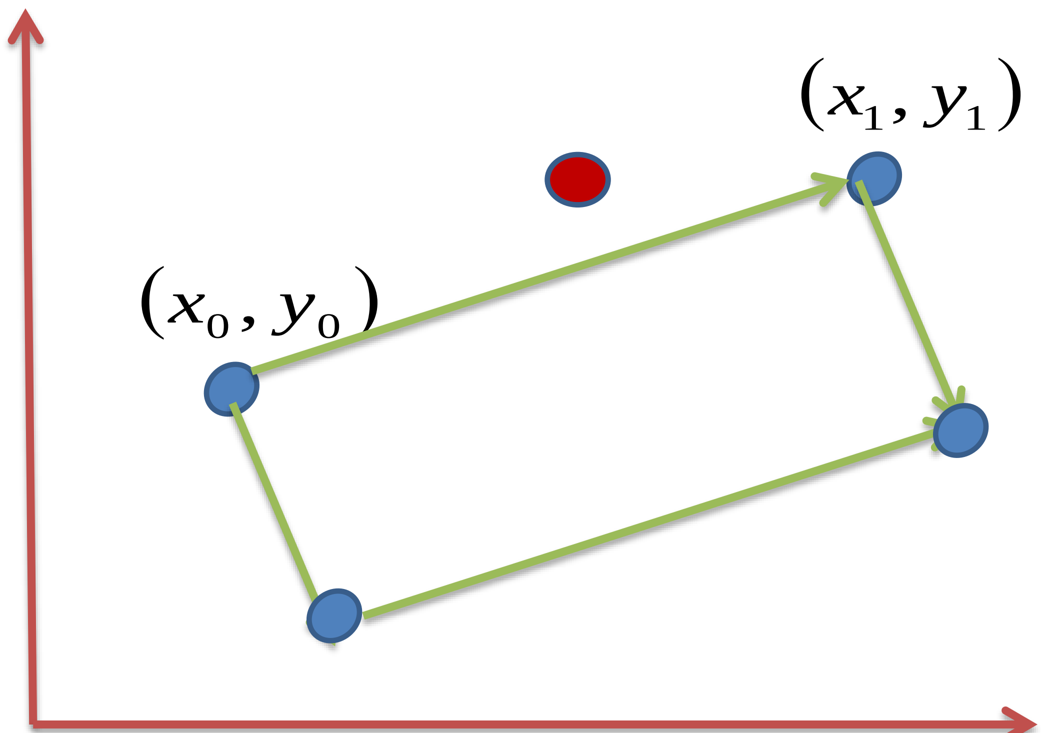
$$t_1 - t_2 \frac{(x_4 - x_3)}{(x_2 - x_1)} = \frac{x_3 - x_1}{(x_2 - x_1)}$$

$$t_1 - t_2 \frac{(y_4 - y_3)}{(y_2 - y_1)} = \frac{y_3 - y_1}{(y_2 - y_1)}$$

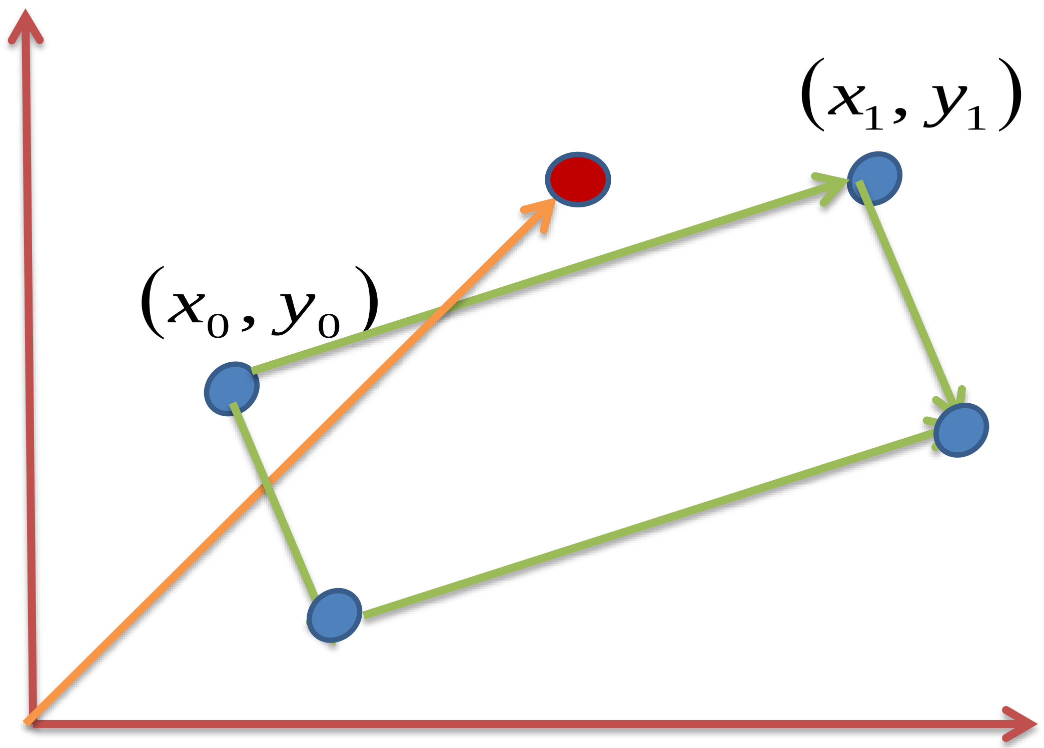
$$t_2\left(\frac{(y_4 - y_3)}{(y_2 - y_1)} - \frac{(x_4 - x_3)}{(x_2 - x_1)} \right) = \frac{y_3 - y_1}{(y_2 - y_1)} - \frac{x_3 - x_1}{(x_2 - x_1)}$$

$$t_2 = \frac{B}{A}$$

Inside or Outside Polygon



Inside or Outside Polygon



Inside or Outside Polygon

$$(x_0, y_0)$$

Random Vector