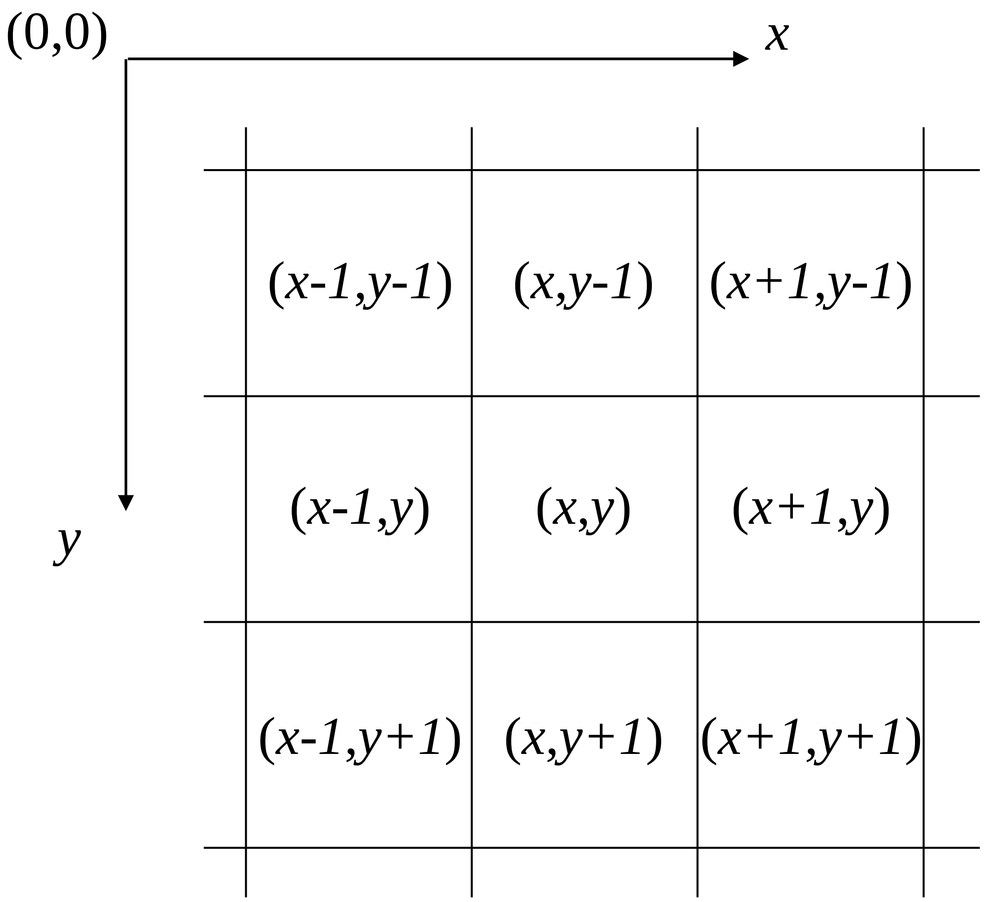
Digital Image Processing:

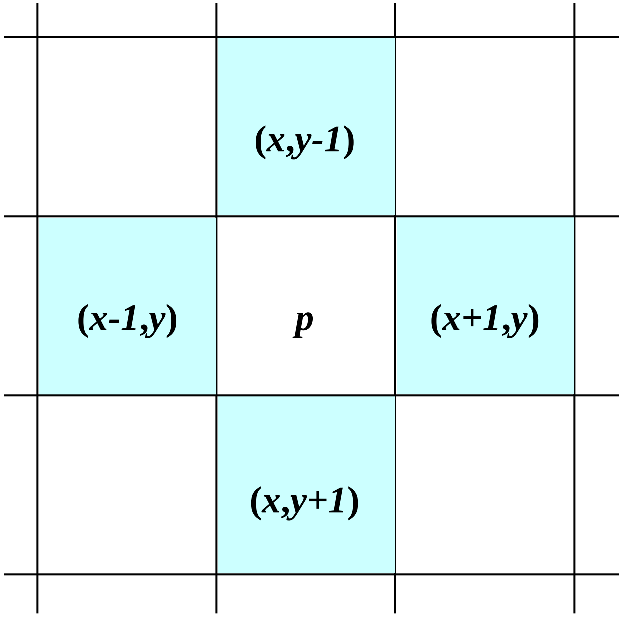
Basic Relationship of Pixels



Conventional indexing method

Neighbors of a Pixel

Neighborhood relation is used to tell adjacent pixels. It is useful for analyzing regions.



4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{array}{c} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{array} \right\}$$

4-neighborhood relation considers only vertical and horizontal neighbors.

Note: $q \in N_4(p)$ implies $p \in N_4(q)$

Neighbors of a Pixel (cont.)

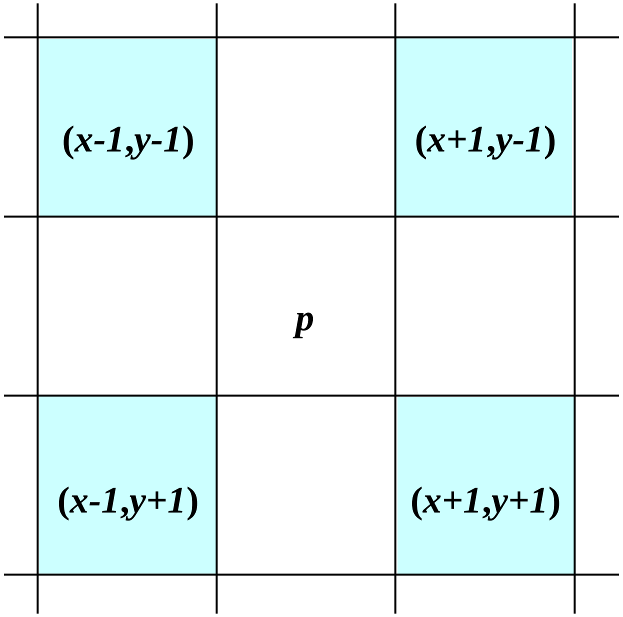
(x-1,y-1)	***(x,y-1)***	***(x+1,y-1)***
(x-1,y)	***p***	***(x+1,y)***
(x-1,y+1)	***(x,y+1)***	***(x+1,y+1)***

8-neighbors of *p*:

$$N_8(p) = \left\{ \begin{matrix} (x-1,y-1) \\ (x,y-1) \\ (x+1,y-1) \\ (x-1,y) \\ (x+1,y) \\ (x-1,y+1) \\ (x,y+1) \\ (x+1,y+1) \end{matrix} \right\}$$

8-neighborhood relation considers all neighbor pixels.

Neighbors of a Pixel (cont.)



Diagonal neighbors of *p*:

$$N_D(p) = \left\{ \begin{matrix} (x-1,y-1) \\ (x+1,y-1) \\ (x-1,y+1) \\ (x+1,y+1) \end{matrix} \right\}$$

Diagonal -neighborhood relation considers only diagonal neighbor pixels.

Connectivity

Connectivity is adapted from neighborhood relation. Two pixels are connected if they are in the same class (i.e. the same color or the same range of intensity) and they are neighbors of one another.

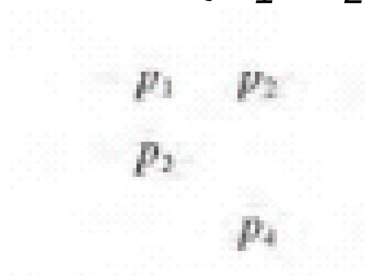
For p and q from the same class

- 4-connectivity: p and q are 4-connected if $q \in N_4(p)$
- 8-connectivity: p and q are 8-connected if $q \in N_8(p)$
- mixed-connectivity (m-connectivity):

  p and q are m-connected if $q \in N_4(p)$ or
  $q \in N_D(p)$ and $N_4(p) \cap N_4(q) = \varnothing$

Relations, Equivalence, and Transitive Closure

- A *binary relation' R* on a set *A* is a setof pairs of elements from *A.* If the pair *(a, b)* is in *R,* the notation often used is *aRb* which, in words, is interpreted to mean *"a* is related to *b."* Take for example, the set of points $A = \{P_1, P_2, P_3\ P_4\}$ arranged as



- and define the relation "4-connected." In this case, *R* is the set of pairs of points from *A* that are 4-connected; that is, *R* = {(PI . *p,),* *(p"* PI)' *(P .. p,), (p" PI)}'* Thus *PI* is related to *p"* and *p,* is related to *p"* and vice versa, but *p* is not related to any other point under the relation "4-connected".

Relations

- A binary relation *R* over set *A is* said to be
  - *reflexive* if for each *a* in *A, aRa;*
  - *symmetric* if for each *a* and *b* in *A* , *aRb* implies *bRa;* and
  - *transitive* if for *a, b,* and c in *A* , *aRb* and *bRc* implies *aRc.*
- A relation satisfying these three properties is called an *equivalence relation.*

Example

- letting $R = \{(a. a), (a, b), (b, d), (d, b). (c, e)\}$ yields the matrix

$$\mathbf{B} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Symmetric calculation

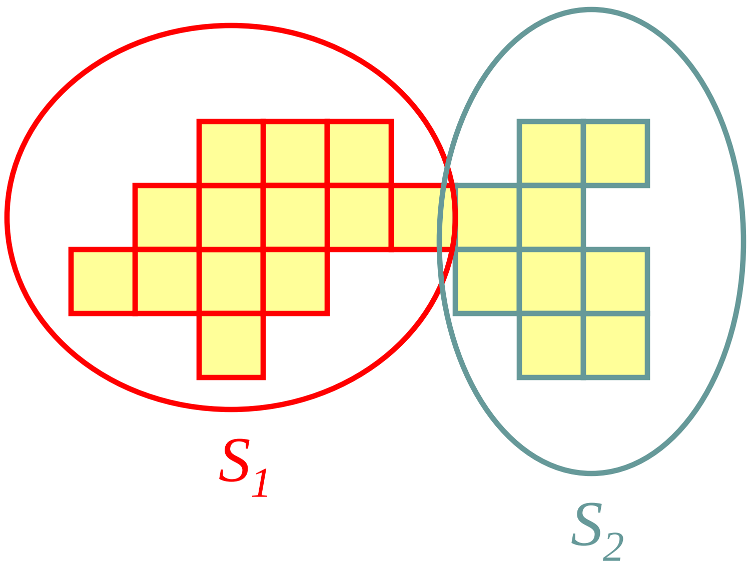
$R^+ = \{(a, a), (a, b), (a, d), (b, b), (b, d), (d, b), (d, d), (c, e)\}$

$$\mathbf{B}^+ = \mathbf{B} + \mathbf{BB} + \mathbf{BBB} + \cdots + (\mathbf{B})^n$$

$$\mathbf{B}^+ = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Adjacency

A pixel *p* is *adjacent* to pixel *q* if they are connected. Two image subsets S_1 and S_2 are adjacent if some pixel in S_1 is adjacent to some pixel in S_2



We can define type of adjacency: 4-adjacency, 8-adjacency or m-adjacency depending on type of connectivity.

Path

A ***path*** from pixel *p* at (x,y) to pixel *q* at (s,t) is a sequence of distinct pixels:

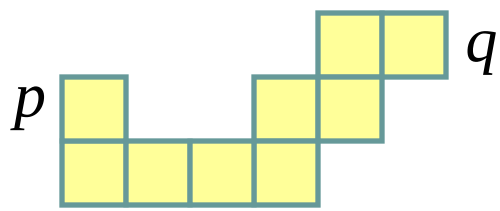
$$(x_0,y_0), (x_1,y_1), (x_2,y_2),\ldots, (x_n,y_n)$$

such that

$$(x_0,y_0) = (x,y) \text{ and } (x_n,y_n) = (s,t)$$

and

$$(x_i,y_i) \text{ is adjacent to } (x_{i-1},y_{i-1}), \qquad i = 1,\ldots,n$$



We can define type of path: 4-path, 8-path or m-path depending on type of adjacency.

Distance

For pixel p, q, and z with coordinates (x,y), (s,t) and (u,v), D is a ***distance function*** or ***metric*** if

- $D(p,q) \geq 0 \quad (D(p,q) = 0$ if and only if $p = q)$
- $D(p,q) = D(q,p)$
- $D(p,z) \leq D(p,q) + D(q,z)$

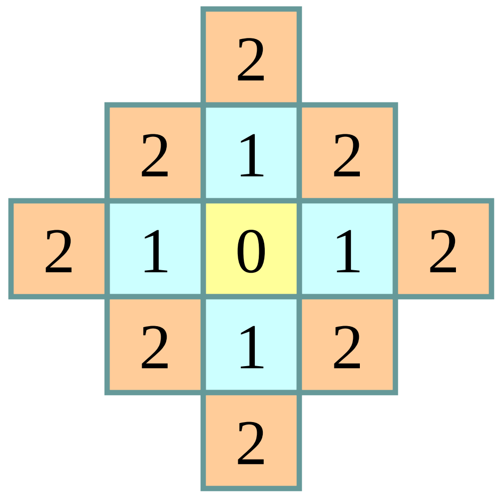
Example: Euclidean distance

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

Distance (cont.)

D_4-distance (*city-block distance*) is defined as

$$D_4(p,q) = |x - s| + |y - t|$$



Pixels with $D_4(p) = 1$ is 4-neighbors of *p*.

Distance (cont.)

D_8-distance (*chessboard distance*) is defined as

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Pixels with $D_8(p) = 1$ is 8-neighbors of p.

Arithmetic/Logic Operations

Addition:	$p + q$
Subtraction:	$p - q$
Multiplication:	$p * q$ (also, pq and $p \times q$)
Division:	$p \div q$

AND:	pANDq (also, $p \cdot q$)
OR:	pORq (also, $p + q$)
COMPLEMENT:	NOTq (also, $\overline{q}$)

Imaging Geometry

Translation

$$X^* = X + X_0$$
$$Y^* = Y + Y_0$$
$$Z^* = Z + Z_0$$

$$\mathbf{v} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{v}^* = \begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation

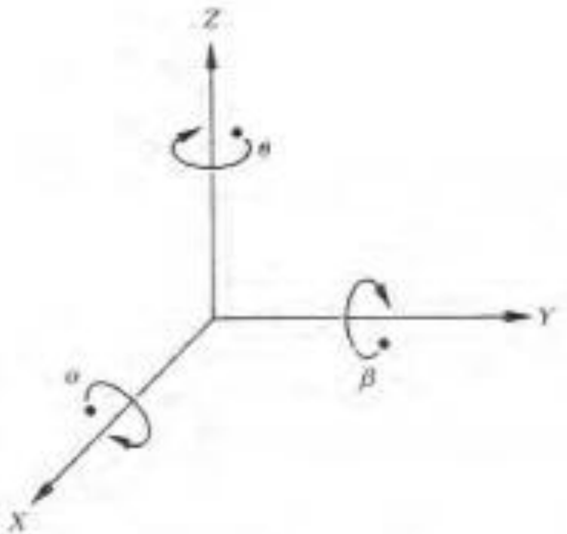
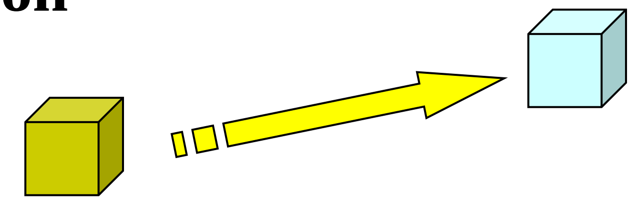


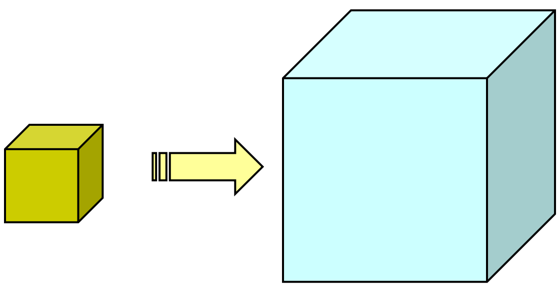
Figure 2.16 *Rotation of a point about each of the coordinate axes. Angles are measured clockwise when looking along the rotation axis toward the origin.*

Imaging Geometry :Transformations

1. Translation



2. Scaling



3. Rotating

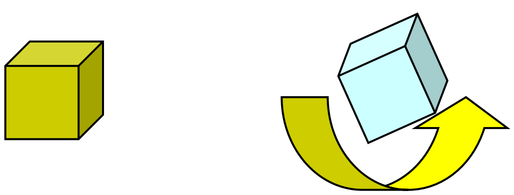
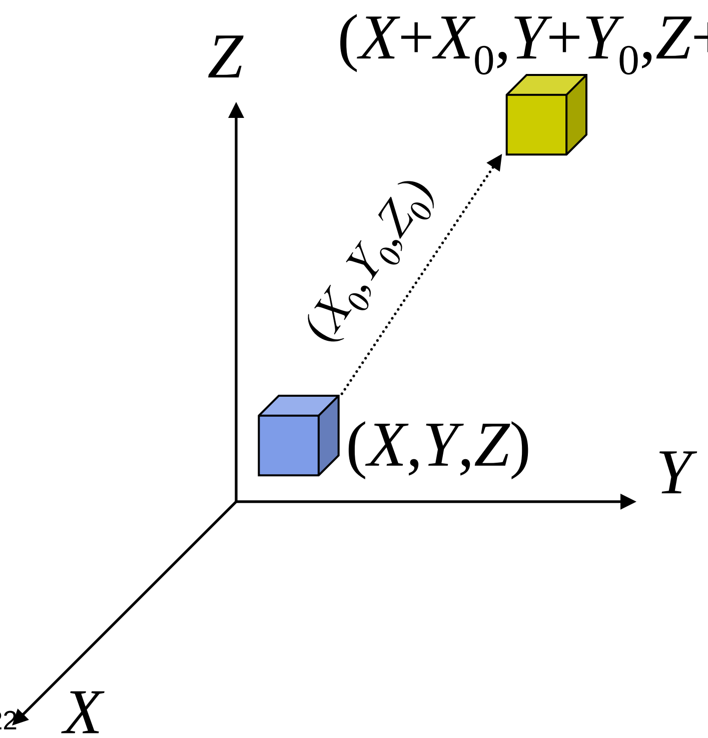


Image Geometry: Translation of Object

Displace the object by vector (X_0, Y_0, Z_0) with respect to its old position.

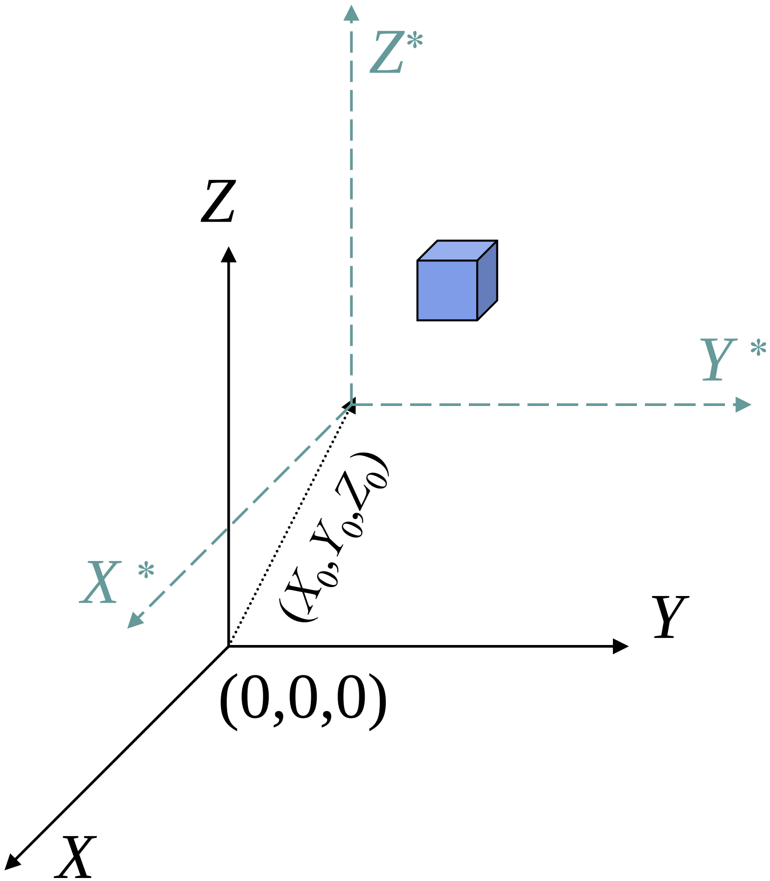
$$X^* = X + X_0$$
$$Y^* = Y + Y_0$$
$$Z^* = Z + Z_0$$



$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Image Geometry: Translation of Frame

Translate the origin point of the frame by (X_0, Y_0, Z_0) with respect to the old frame



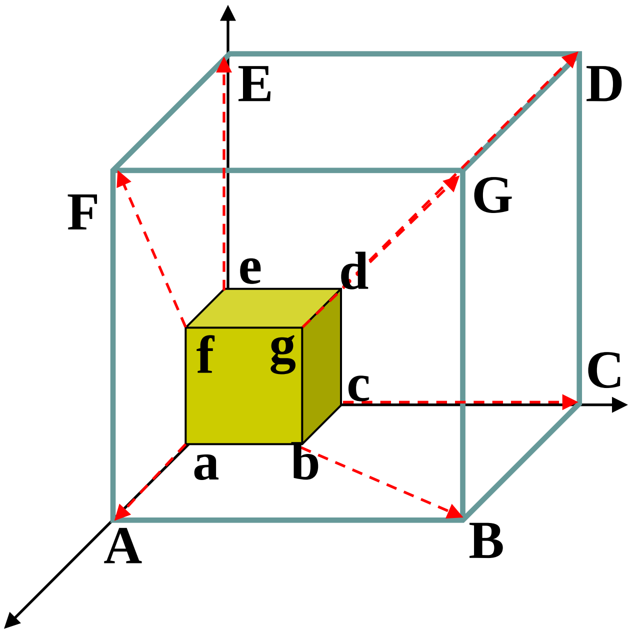
$$X^* = X - X_0$$
$$Y^* = Y - Y_0$$
$$Z^* = Z - Z_0$$

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The object still stays at the same position. Only the frame is moved.

Image Geometry: Scaling

Scale by factors S_x, S_y, S_z along X, Y, and Z axes.



$$X^* = S_x X$$

$$Y^* = S_y Y$$

$$Z^* = S_z Z$$

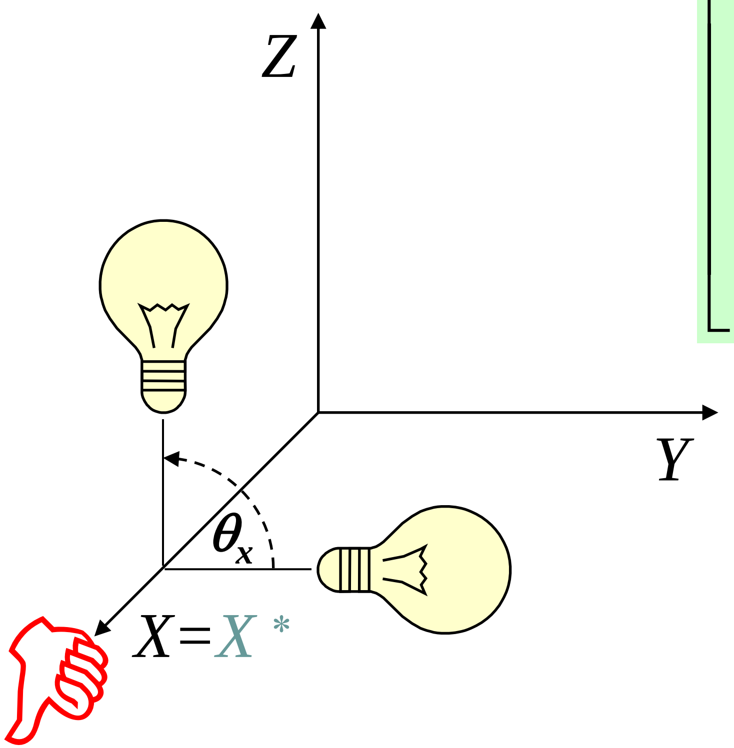
$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Note: Origin point is unchanged.

Image Geometry: Rotating an object about X-axis

Rotate an object about X-axis by θ_x in a counterclockwise direction.

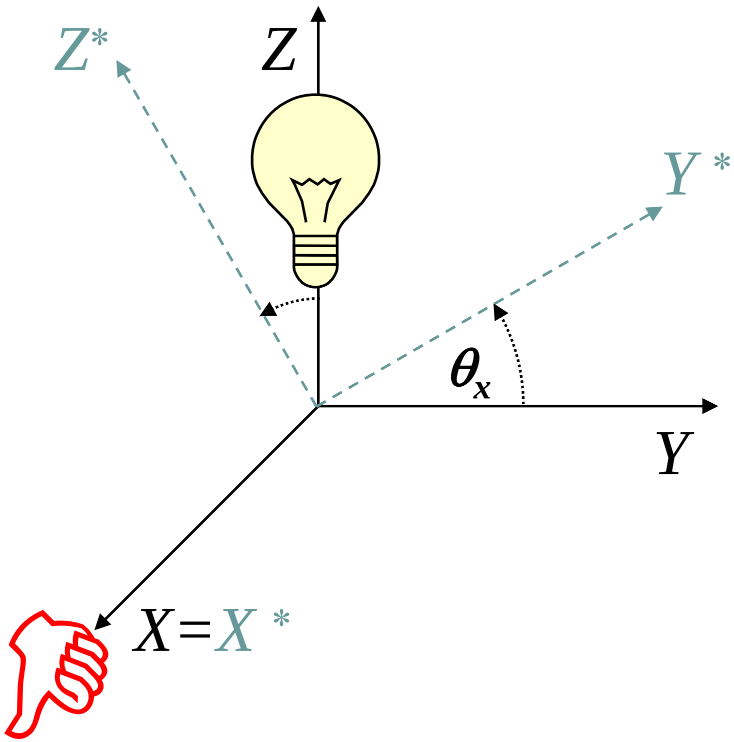
$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



Note : In this case the object is moved. Only y and z are changed while x stills the same.

Image Geometry: Rotating a frame about X-axis

Rotate the frame about X-axis by θ_x in a counterclockwise direction.



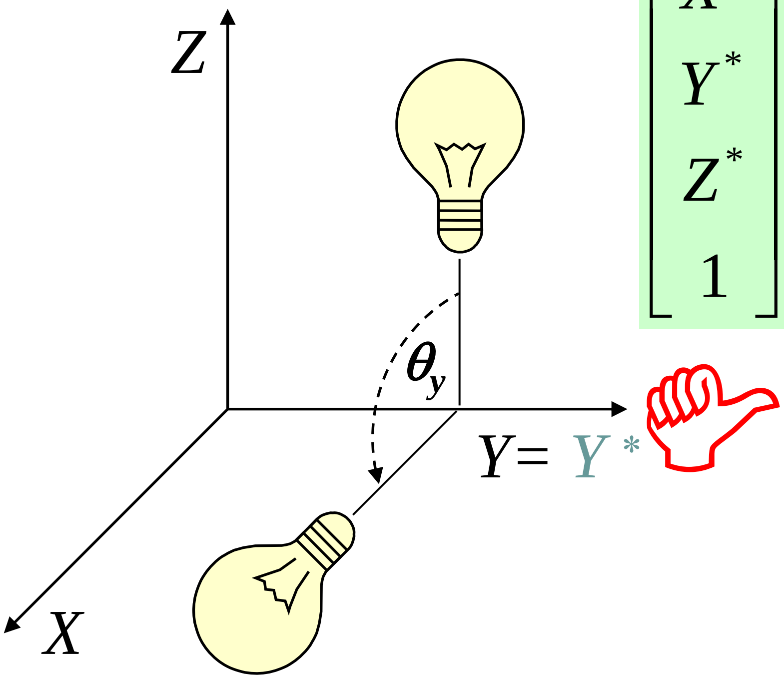
$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x & 0 \\ 0 & -\sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Note : In this case the object is not moved. The frame is rotated instead.

Image Geometry: Rotating an object about Y-axis

Rotate an object about Y-axis by θ_y in a counterclockwise direction.

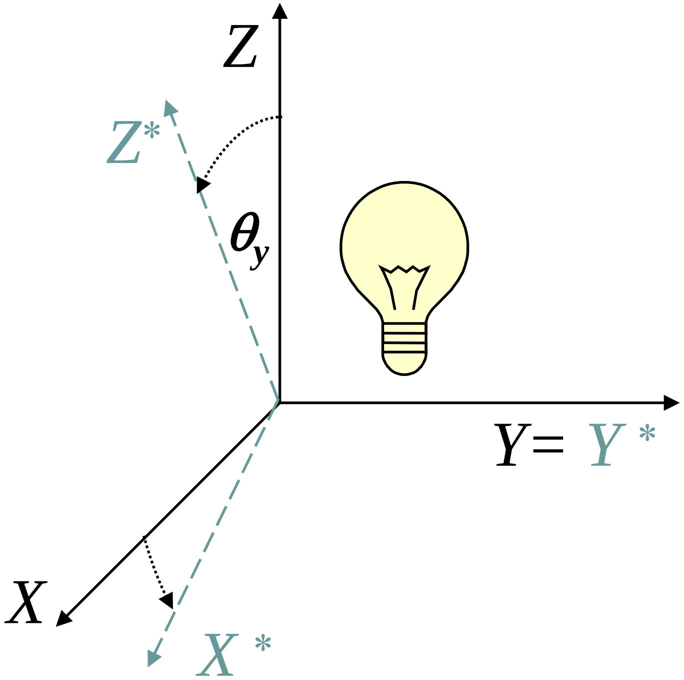
$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



Note : In this case the object is moved. Only x and z are changed while y stills the same.

Image Geometry: Rotating a frame about Y-axis

Rotate the frame about Y-axis by θ_y in a counterclockwise direction.

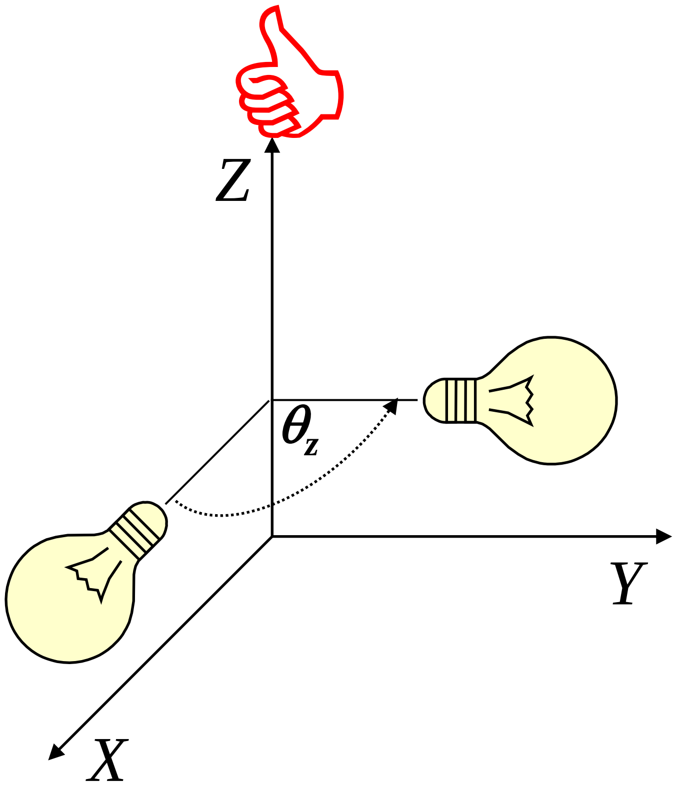


$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Note : In this case the object is not moved. The frame is rotated instead.

Image Geometry: Rotating an object about Z-axis

Rotate an object about Z-axis by θ_z in a counterclockwise direction.

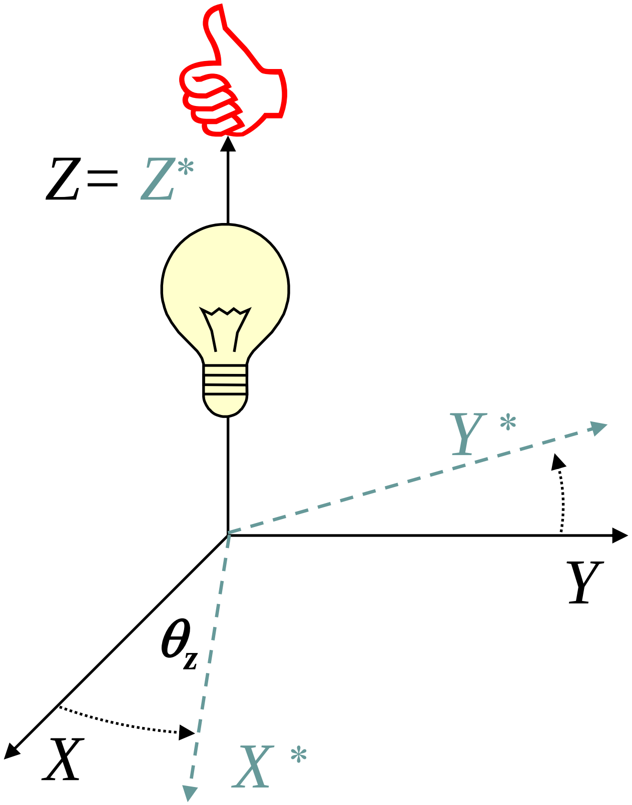


$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Note : In this case the object is moved. Only x and y are changed while z stills the same.

Image Geometry: Rotating a frame about Z-axis

Rotate the frame about *Z*-axis by θ_z in a counterclockwise direction.



$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 & 0 \\ -\sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Note : In this case the object is not moved. The frame is rotated instead.

Concatenation and inverse transformations

$$\mathbf{v}^* = \mathbf{R}_\theta(\mathbf{S}(\mathbf{T}\mathbf{v}))$$
$$= \mathbf{A}\mathbf{v}$$

$$\mathbf{V}^* = \mathbf{A}\mathbf{V}.$$

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Similarly, the inverse rotation matrix $\mathbf{R}_\theta^{-1}$ is

$$\mathbf{R}_\theta^{-1} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) & 0 & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$