2.4 Some Basic Relationship Between Pixels

2.4.1 Neighbors of a pixel

- 4-neighbors of pixel p at (x, y):

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

  - denoted by $N_4(P)$
  - unit distance from p

- 4 diagonal neighbors of p:

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

  - denoted by $N_D(P)$
  - $N_4(P) + N_D(P) = N_8(P)$ 8-neighbor of P

2.4.2 Connectivity

- connectivity:

- being adjacent between two pixel and having similar gray level

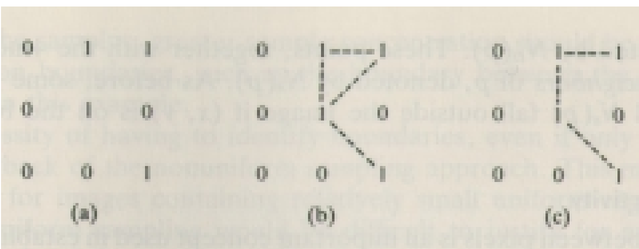
- 3 types of *connectivity:*

1) 4-connectivity ; 4-connected

2) 8-connectivity ; 8-connected

3) m(mixed)-connectivity ; p and q are m-connected if

   i. q is in $N_4(P)$ or
   ii. q is $N_D(P)$ and the set $N_4(P) \cap N_8(P)$ is empty
      - ✓ modification of $N_8(P)$
      - ✓ elimination of the multi path connections



- *adjacent* if pixels are connected.

- Path:

$$\underset{(x,y)}{pixel} \quad p \xrightarrow[(x_0,y_0)(x_1,y_1)\Lambda\ (x_n,y_n)]{path} \quad \underset{(s,t)}{pixel} \quad q$$

where $(x_0, y_0) = (x, y)$

$(x_n, y_n) = (s, t)$

(x_i, y_i) : adjacent to (x_{i-1}, y_{i-1}) $1 \le i \le n$

n : length of path

- p is connected to q in image subset S
: if there is a path from p to q consisting entirely of pixels in S

- connected components
: the set of pixels in S that are connected to p

2.4.3 Labeling of Connected Components

2.4.4 Relations, Equivalence and Transitive Closure

- Binary relation R on a set of A:

- a set of pairs of elements from A
- a R b : a is related to b
- ex. p_1 p_2
  p_3
  p_4
- 4-connected relation : $R = \{(p_1, p_2), (p_2, p_1), (p_1, p_3), (p_3, p_1)\}$
  - ✓ if pair (a,b) is in R
    1) a R a : reflexive
    2) $aRb \xrightarrow{imply} bRa$: symmetric
    3) $aRb, bRc \longrightarrow aRc$: transitive (a, b, c 가 서로 연속되어 놓여 있음)
  - ✓ relation satisfying these three properties : equivalence relation

- Binary matrix:

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

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

- reflexive relation

  : all the main diagonal terms : 1
  i.e. $R = \{(a,a),(b,b),\Lambda\ (e,e)\}$ by definition

- symmetric relation

  : ***B*** – symmetric matrix

- Transitivity:

  - implies that if aRb and bRc, then aRc
    (즉, 임의의 a, b, c 에 대해서 aRb와 bRc가 R 에 존재할 때 aRc도 존재하면 이를 transitive relation.이라 한다.

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

    if $bRd\ \&\ dRb \rightarrow \underline{bRb}$ if $dRb\ \&\ bRd \rightarrow \underline{dRd}$

  - transitive closure of R

    ✓ the set containing these “implied” relation
    ✓ denoted by R^+

2.4.5 Distance Measures

- Distance function or metric:

  For pixels p, q and z with coordinates (x,y), (s,t) and (u,v)

  A. $D(p,q) \geq 0 \quad (D(p,q) = 0 \quad its \quad p = q)$
  B. $D(p,q) = D(q,p)$
  C. $D(p,z) \leq p(p,q) + D(q,z)$

- Euclidean distance:

$$D_e(p,q) = [(x-s)^2 + (y-t)^2]^{1/2}$$

- D_4 distance (city-block distance) :

  - $D_4(p,q) = |x-s| + |y-t|$
  - pixels having a D_4 distance from (x,y)
    : diamond form

- D_8 distance (chessboard distance) between p, q:

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

- D_m connectivity and distance:

  - Dependant on the connectivity of pixels.

① form :

$$\begin{matrix} & p_3 & p_4 \\ p_1 & p_2 & \\ p & & \end{matrix}$$

where ——— : connection

②

$$\begin{matrix} & 0 & 1 \\ 0 & 1 & \\ 1 & & \end{matrix}$$

distance : 2

③

$$\begin{matrix} & 0 & 1 \\ 1 & 1 & \\ 1 & & \end{matrix}$$

distance : 3

④

$$\begin{matrix} & 1 & 1 \\ 1 & 1 & \\ 1 & & \end{matrix}$$

distance : 4

2.4 6 Arithmetic/Logic Operations (reading assignment)

2.5 Imaging Geometry

2.5.1 Some Basic Transformations

- 3-D Cartesian coordinate system : (X,Y, Z)
- 2-D Cartesian coordinate system : (x, y)

1) Translation:

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

(New Coor: Z^*; Old Coor: Z; Displacement: Z_0)

- in matrix form

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

or using the square matrix (very simple)

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

$$\mathbf{v}^* = \mathbf{A}\mathbf{v}$$

where **A** : 4×4 Transformation matrix
v : column vector containing the original coordinate
$\mathbf{v}^*$: column vector containing the transformed coordinate

- the matrix used for translation

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

2) Scaling:

- Scaling by factors S_x, S_y and S_z along the X,Y and Z

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3) Rotation:

- rotation of a point about the Z-axis by angle θ

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

- rotation of a point about the X-axis by angle α

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- rotation of a point about the Y-axis by angle α

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4) Concatenation and inverse transformations:

- translation, scaling and rotation about the Z-axis of a point **V**

$$\begin{aligned} \mathbf{v}^* &= \mathbf{R}_\theta(\mathbf{S}(\mathbf{Tv})) \\ &= \mathbf{Av} \end{aligned}$$

where $\mathbf{A} = \mathbf{R}_\theta \mathbf{ST}$ 4×4 matrix

- inverse transformation

  ✓ inverse translation matrix

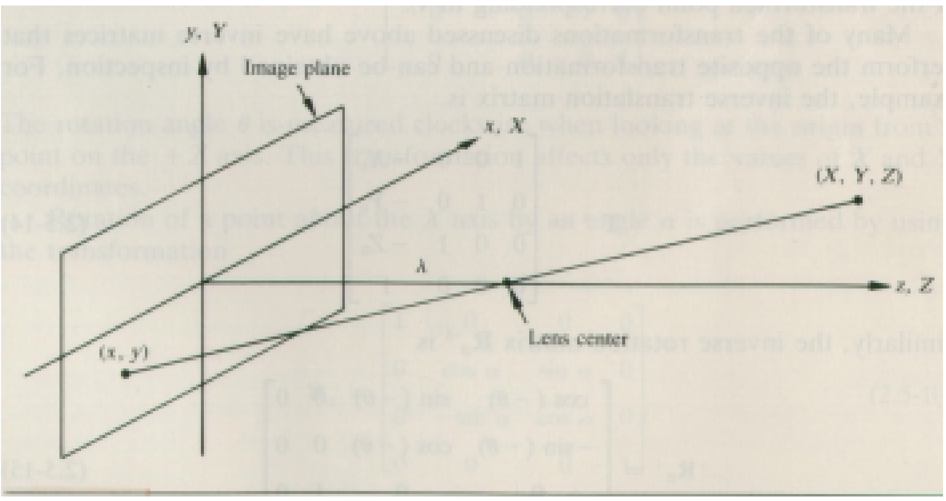
$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  ✓ inverse rotation matrix

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

2.5.2 Perspective Transformations (Imaging Transformations)

- projects 3-D points onto a plane (2-D)



- camera coordinate system : (x, y, z):

  image plane (film) : (x, y)
  lens center : $(0,0,\lambda)$
  world coordinate system (object) : (X, Y, Z) (where $Z \rangle \lambda$)
  assumption : the camera coord. system is aligned with world coord. system.

- relationship that gives the coordinates (x, y) of the projection of the point (X, Y, Z) onto the image plane:

$$\frac{x}{\lambda} = -\frac{X}{Z-\lambda} = \frac{X}{\lambda - Z}$$
$$\frac{y}{\lambda} = -\frac{Y}{Z-\lambda} = \frac{Y}{\lambda - Z}$$

$$\Rightarrow x = \frac{\lambda X}{\lambda - Z},\ y = \frac{\lambda Y}{\lambda - Z} \qquad \text{: nonlinear}$$

- linear matrix form:

  - the homogeneous coordinates of a point with Cartesian coordinates (X, Y, Z).
  : defined as (kX, kY, kZ, k), where k: arbitrary, nonzero constant.

  - homogenous coordi. $\xrightarrow{conversion}$ Cartesian coordi.
  : diving the first three homo. coordi. by the fourth.

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \mathbf{w}_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix}$$

  - the perspective transformation matrix

$$\mathbf{P} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix}$$

$$\mathbf{c}_h = \mathbf{P}\mathbf{w}_h = \begin{bmatrix} kX \\ kY \\ kZ \\ -\frac{kZ}{\lambda} + k \end{bmatrix}$$

✓ the elements of $\mathbf{c}_h$: the camera coordinates in homogeneous form

  - the camera coordinates in Cartesian form (conversion)

$$\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix}$$ 3-D object 를 2-D 로 복사

where (x,y) : coordi. of image plane and z : no interest

- inverse perspective transformation

: maps an image point back into 3-D

$$\mathbf{w}_h = \mathbf{P}^{-1}\mathbf{c}_h$$

where $$\mathbf{P}^{-1} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix}$$

Cartesian coordi. of image points : $(x_0, y_0, 0)$

: homogenous vector form

$$\mathbf{c}_h = \begin{bmatrix} kX_0 \\ kY_0 \\ 0 \\ k \end{bmatrix}$$

- the homogeneous world coordi. Vector
-

✓ $$\mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix}$$

✓ in Cartesian coordi.

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix}$$ $X = x_0, Y = y_0, Z = 0$ 가 되기 때문에 3-D 연산이 확립되지 않는다.
그에 대한 정보가 손실되었으므로

- Z=0 for any 3-D point

- Mapping a 3-D scene onto the image plane

  - many-to-one transformation
  - lose "depth" information

● z : free variable

if $\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix}$

$$\rightarrow \mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ \frac{kz}{\lambda} + k \end{bmatrix}$$

- Cartesian coord.

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{\lambda x_0}{\lambda + z} \\ \frac{\lambda y_0}{\lambda + z} \\ \frac{\lambda Z}{\lambda + z} \end{bmatrix}$$

solving for z in terms of Z

$$\rightarrow X = \frac{x_0}{\lambda}(\lambda - Z)$$

$$Y = \frac{y_0}{\lambda}(\lambda - Z)$$

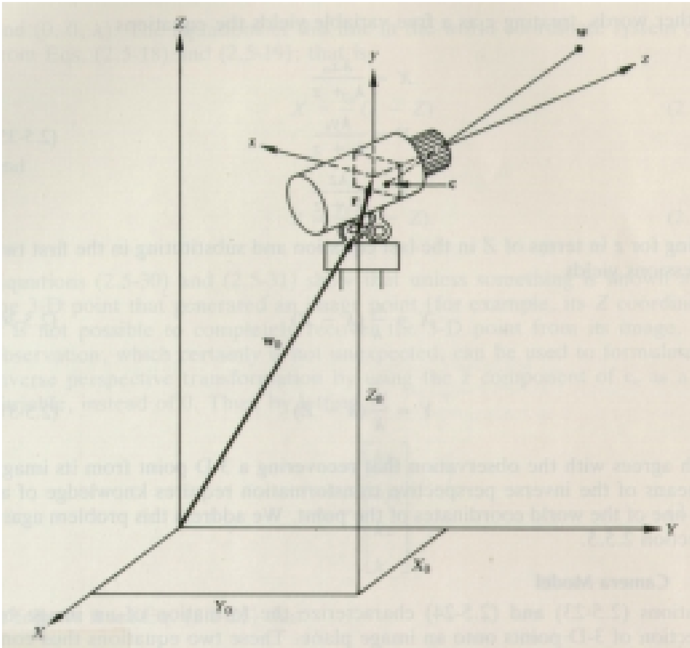
3-D image 에 대한 한 축 (Z)의 정보가 있어야 X,Y 를 복원

∴ it is not possible to completely recover the 3-D point from its image

→ "stereo imaging" needed

2.5.3 Camera Model

- The two coordinates. : separate (more general case)



- (X, Y, Z) : world coord. (position of camera and 3-D point)
  (x, y, z) : camera coord. and image point (C)
- pan (θ) : angle between x and X
  tilt (α) : angle between z and Z
- camera : mounted on the gimbal
- $\mathbf{w}_0$: offset of center of gimbal from origin of world coordi.
- **r** : offset of center of image plane from gimbal center

- Problem:

- geometric arrangement + perspective transformation

- Geometric arrangement:

- Suppose that, initially, the camera was in normal position
- Geometric arrangement

1) displacement of the gimbal center from the origin

2) pan of x-axis
3) tilt of z-axis
4) displacement of image plane from the gimbal center

- Translation of origin of world coordi. system to location of gimbal center

  - transformation matrix

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

$$\mathbf{w}_h^* = \mathbf{G}\mathbf{w}_h$$

- Pan:

  - use of transformation matrix $\mathbf{R}_\theta$ (including $\mathbf{G}\mathbf{w}_h$)

    ✓ counterclockwise : positive angle

- Tilt:

  - use of transformation matrix $\mathbf{R}_\alpha$ (including $\mathbf{R}_\theta\,\mathbf{G}\mathbf{w}_h$)

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

- displacement of origin of image plane by vector **r**

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- a series of transformations $\mathbf{CRGw}_h$

  - world and camera coordi. systems : coincidence

- perspective transformation involving two coordi. System

  $\mathbf{C_h}$=$\mathbf{PCRGw_h}$

  : homogenous representation in camera coordi. system

- Cartesian coordi. (x,y):

  eq. 2.5-42, 43 (p. 64)

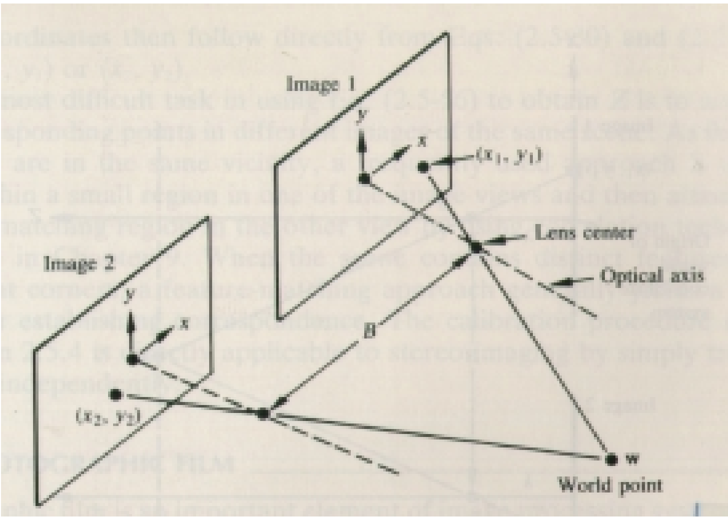
2.5.5 Stereo Imaging

- 3-D scene →2-D mapping:

  - many-to-one mapping
  - mapping depth information
  : can be obtained by using stereoscopic imaging tech.

- stereo imaging:

  - an object →two separate image views



  - base line **B**: distance between the centers of the two lenses

  - objectives: to find the coordi. (X,Y,Z) of the point w from imaging points (x_1, y_1) and (x_2, y_2)

  - assumptions
    - ✓ cameras : identical
    - ✓ coordi. systems of both cameras : perfectly aligned (일직선으로 맞추다)

- coincidence:

  - coincidence of the first camera with the world coordi. system
    - ✓ the first camera

      $$X_1 = \frac{x_1}{\lambda}(\lambda - Z_1)$$

    - ✓ the second camera

      $$X_2 = \frac{x_2}{\lambda}(\lambda - Z_2)$$

  - $X_2 + (-X_1) = B \quad \rightarrow \quad X_2 = X_1 + B$

    $Z_2 = Z_1 = Z$

    $$\Rightarrow Z = \lambda - \frac{\lambda B}{x_2 - x_1}$$

- the most difficult task to obtain Z

  : to find two corresponding points in different images of the same scene

  - frequently used approach (correlation tech.)
  - find the best matching region in the other view by correlation tech.