Midterm Review - Fall 2017

CIS 581

3D Point Projection

Point projection in metric space

$$(X,Y,Z) \rightarrow (u_{ccd}, v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

2D projection onto CCD plane

Point projection in pixel space

$$(X,Y,Z) \rightarrow (u_{img}, v_{img}) = (f_m \frac{w_{img}}{w_{ccd}} \frac{X}{Z}, f_m \frac{h_{img}}{h_{ccd}} \frac{Y}{Z})$$

fm - focal length in m

$$(x,y) \rightarrow (x,y,1)$$

Homogenous coordinates

$$(x,y,1) = (f_m x, f_m y, f_m) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z}, f_m)$$

Camera Matrix

$$\lambda \begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Homogenous representation

Pixel space

Metric space

$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \mathrm{H} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

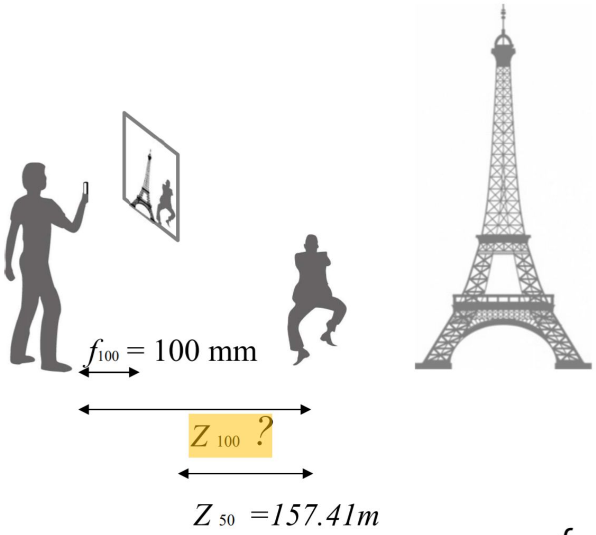
Perspective Transformation (Homography)

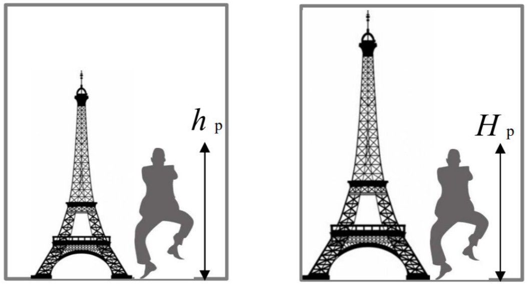
Pixel space

Pixel space

Dolly Zoom

Given focal length (f_m=100mm),
what Z_{100} to make the height of the person remain the same as f_m=50mm?





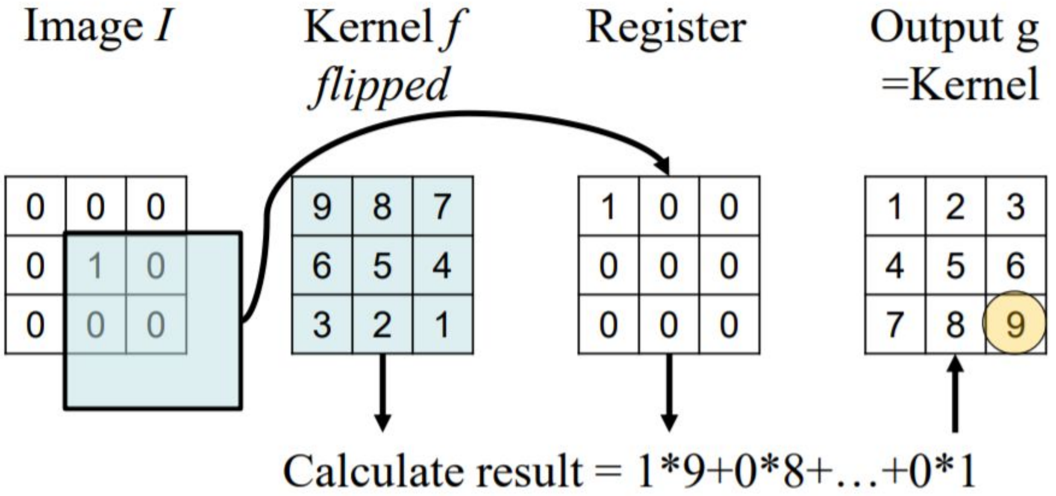
$$h_{50} = f_{50} \frac{Y}{Z_{50}} \qquad h_{100} = f_{100} \frac{Y}{Z_{100}}$$

s.t. $h_{100} = h_{50}$

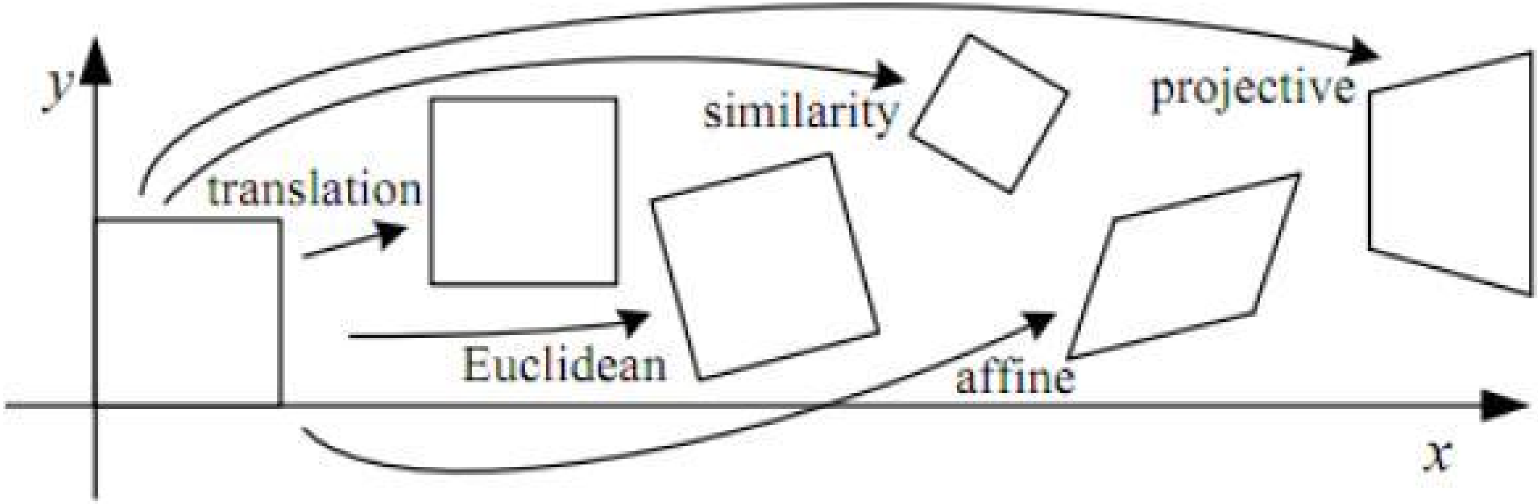
$$Z_{100} = \frac{f_{100}}{f_{50}} Z_{50} \qquad Z_{100} = \frac{100}{50} 157.41 = 314.8\text{m}$$

Convolution

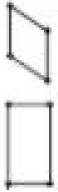
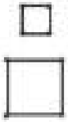
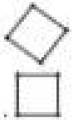
- Convolution is filtering with kernel flipped



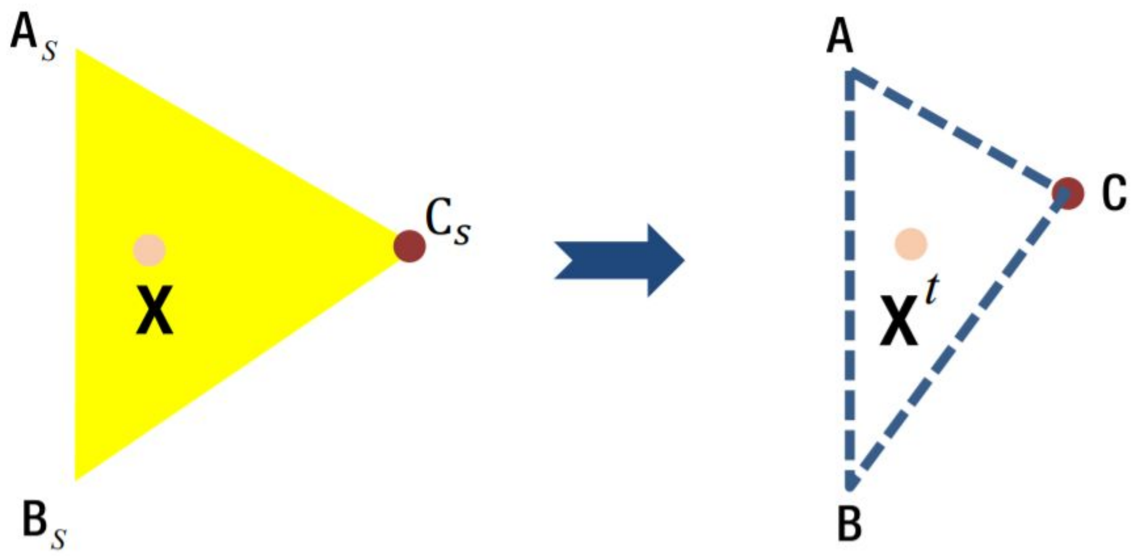
Geometric Transformations



Geometric Transformations

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, **I**, **J** (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

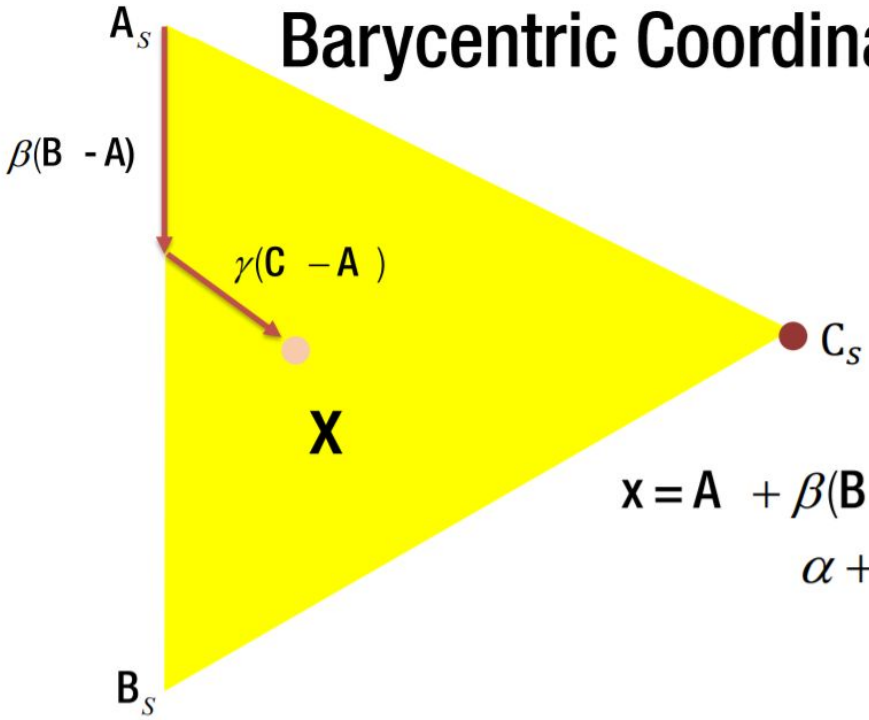
Triangle warping = Affine transform



Affine transform is a pixel transportation $X \rightarrow X^t$

It is controlled by the movement of the three vertices of the triangle

Barycentric Coordinates

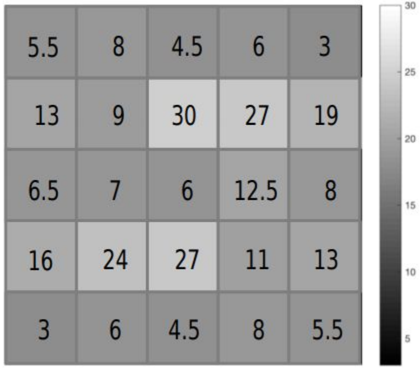


$$\mathbf{x} = \mathbf{A} + \beta(\mathbf{B} - \mathbf{A}) + \gamma(\mathbf{C} - \mathbf{A})$$

$$\alpha + \beta + \gamma = 1$$

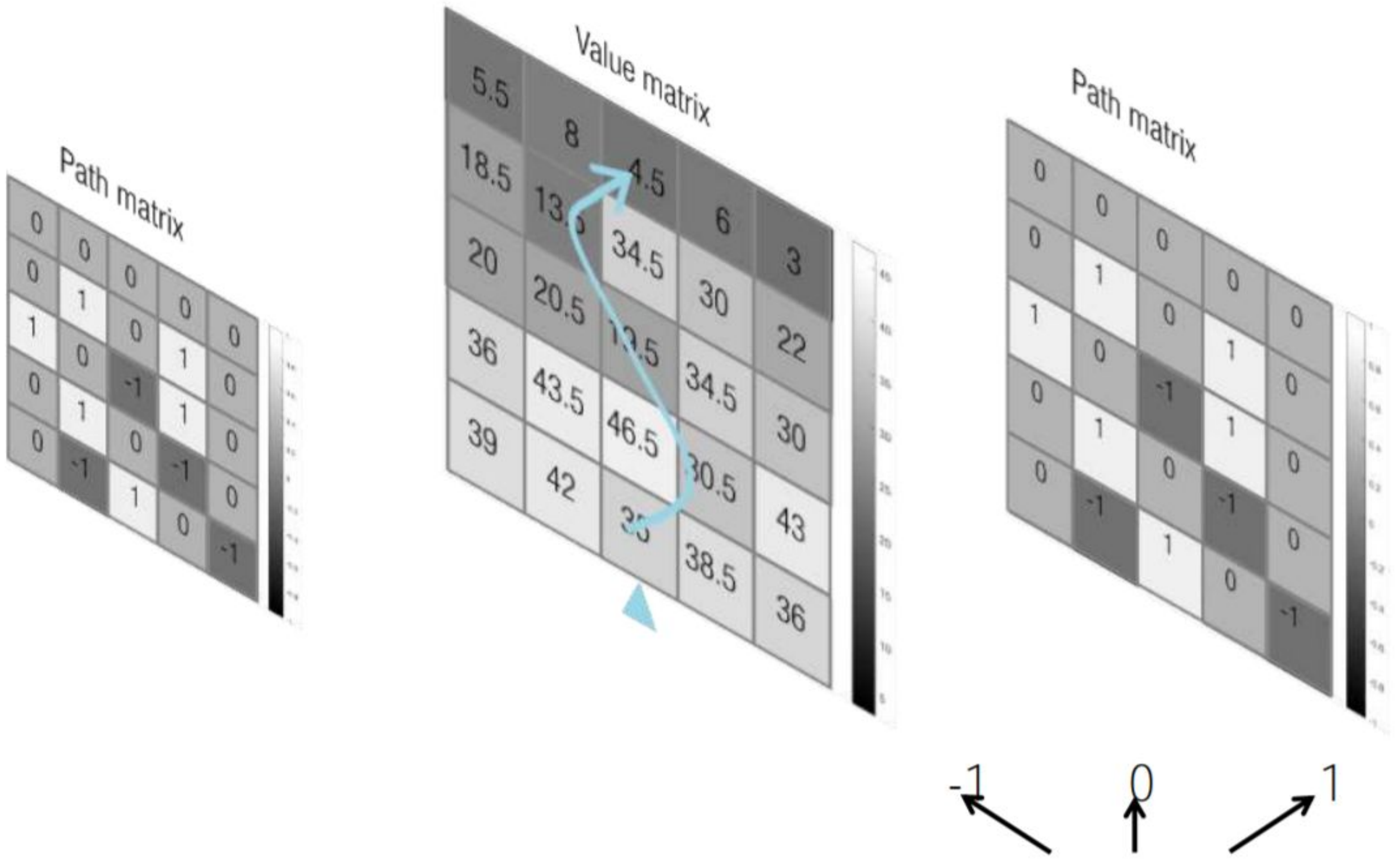
$$\begin{bmatrix} \mathbf{A}_x & \mathbf{B}_x & \mathbf{C}_x \\ \mathbf{A}_y & \mathbf{B}_y & \mathbf{C}_y \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \lambda \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$ linear equations in 3 unknowns

e: energy function



Energy function

- Energy function records the cost of a pixel.
- Typically it is the image gradient magnitude



Path matrix
Value matrix
Path matrix
-1
0
1

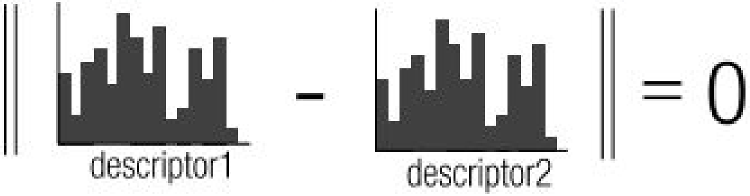
Local Scale Invariant Feature Transform (SIFT)

SIFT automatically finds the optimal scale of feature point and its orientation.

Desired properties:

- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.
- Orientation aware

Local Scale Invariant Feature Transform (SIFT)



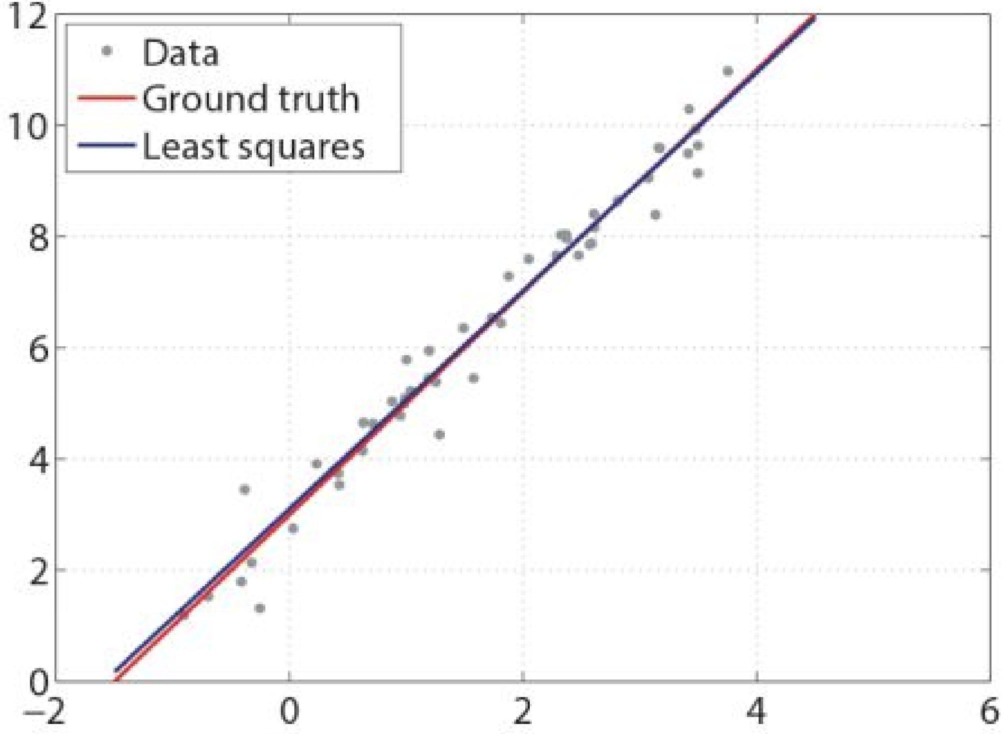
RANSAC: Random Sample Consensus: Linear Least Squares

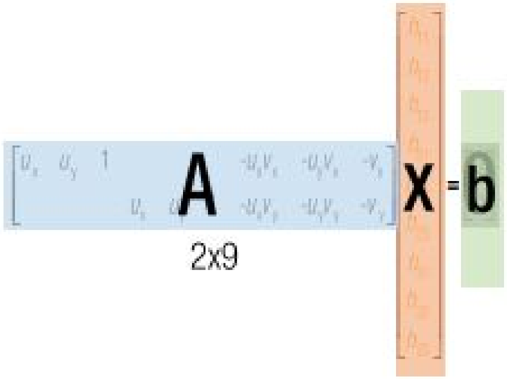
$$A\,x = 0$$

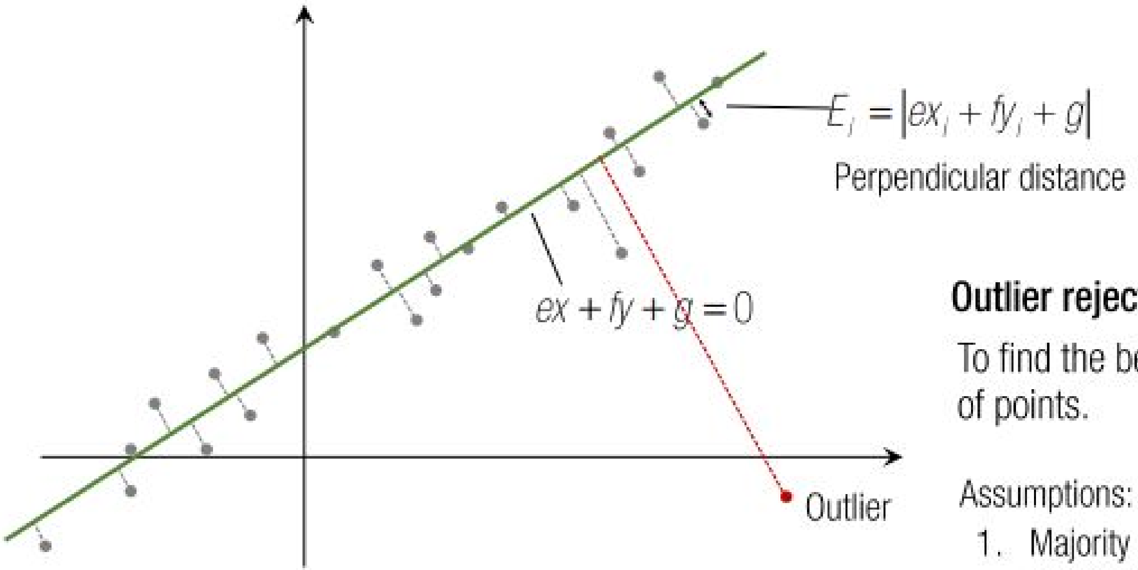
2x9

Outlier?

Recall: Line Fitting (Ax=b)







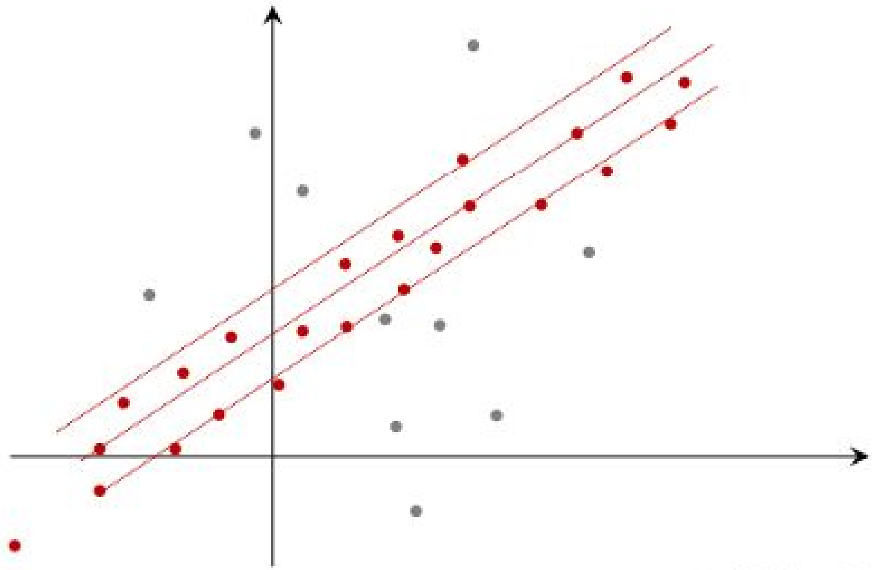
Outlier rejection strategy:

To find the best line that explanes the <u>maximum</u> number of points.

Assumptions:

1. Majority of good samples agree with the underlying model (good apples are same and simple.).

2. Bad samples does not consistently agree with a single model
   (all bad apples are different and complicated.).



1. Random sampling

2. Model building

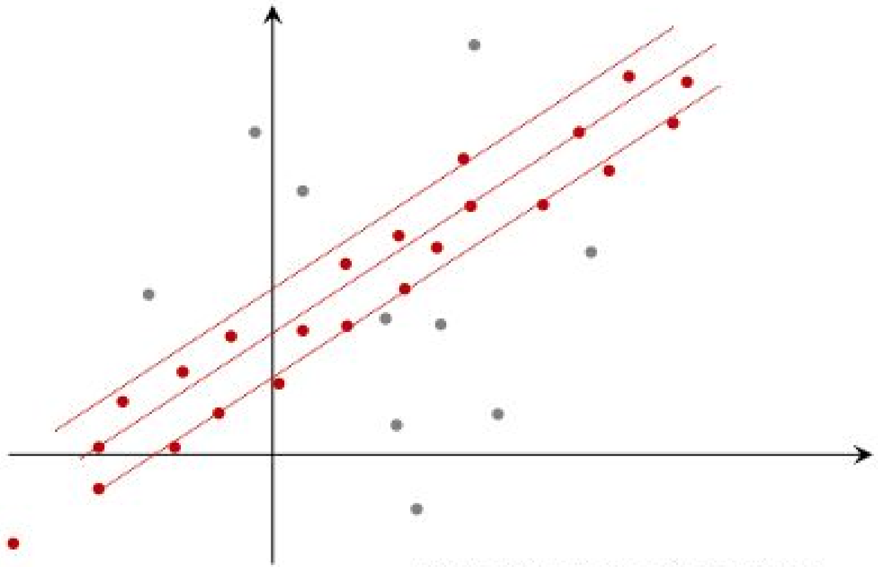
3. Thresholding

4. Inlier counting

# of inliers: 23

Maximum number of inliers

RANSAC: Random Sample Consensus



Required number of iterations with p success rate:

$$k = \frac{\log(1-p)}{\log(1-w^n)} \quad \text{where } w = \frac{\text{\# of inliers}}{\text{\# of samples}}$$

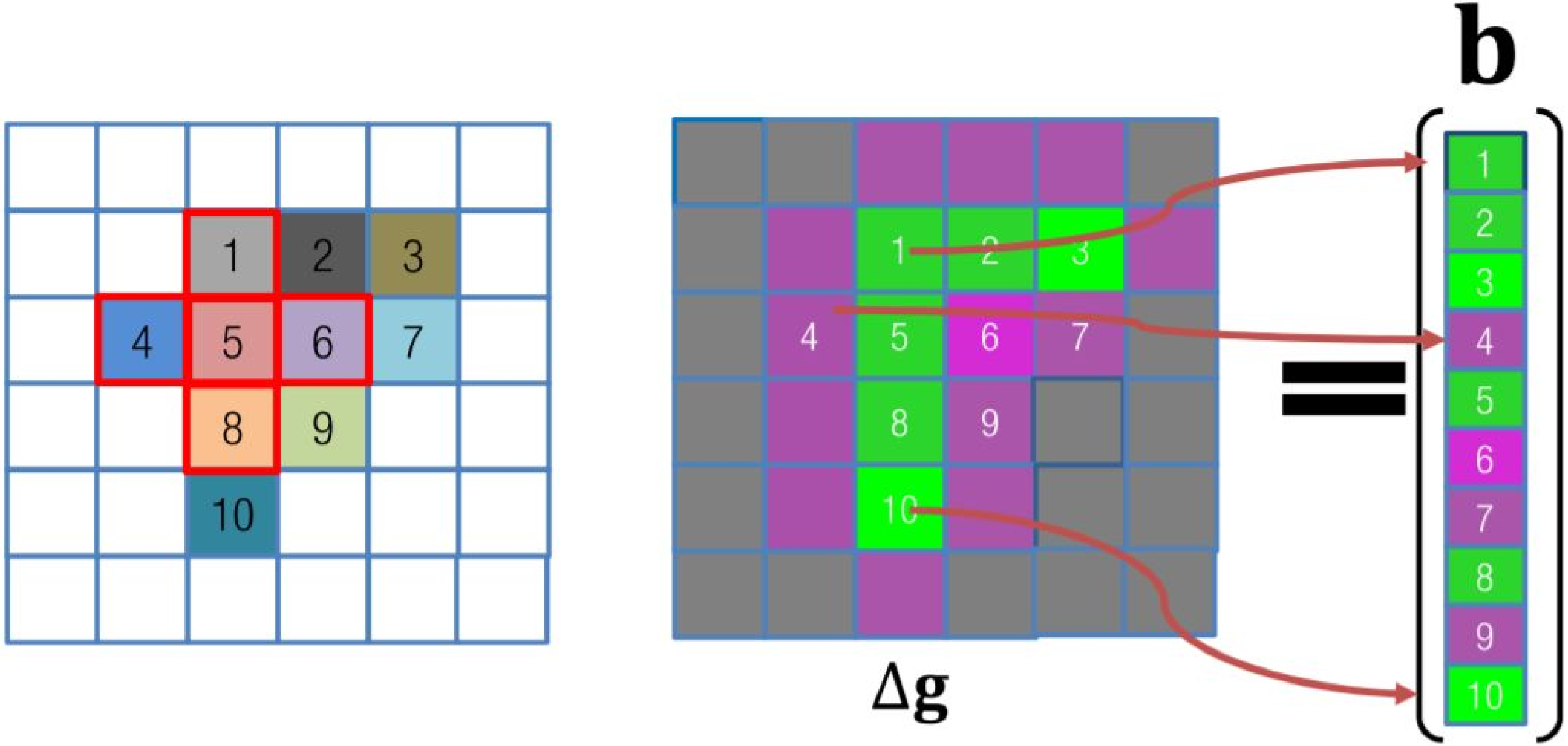
Probability of choosing an inlier: $w = \frac{\text{\# of inliers}}{\text{\# of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

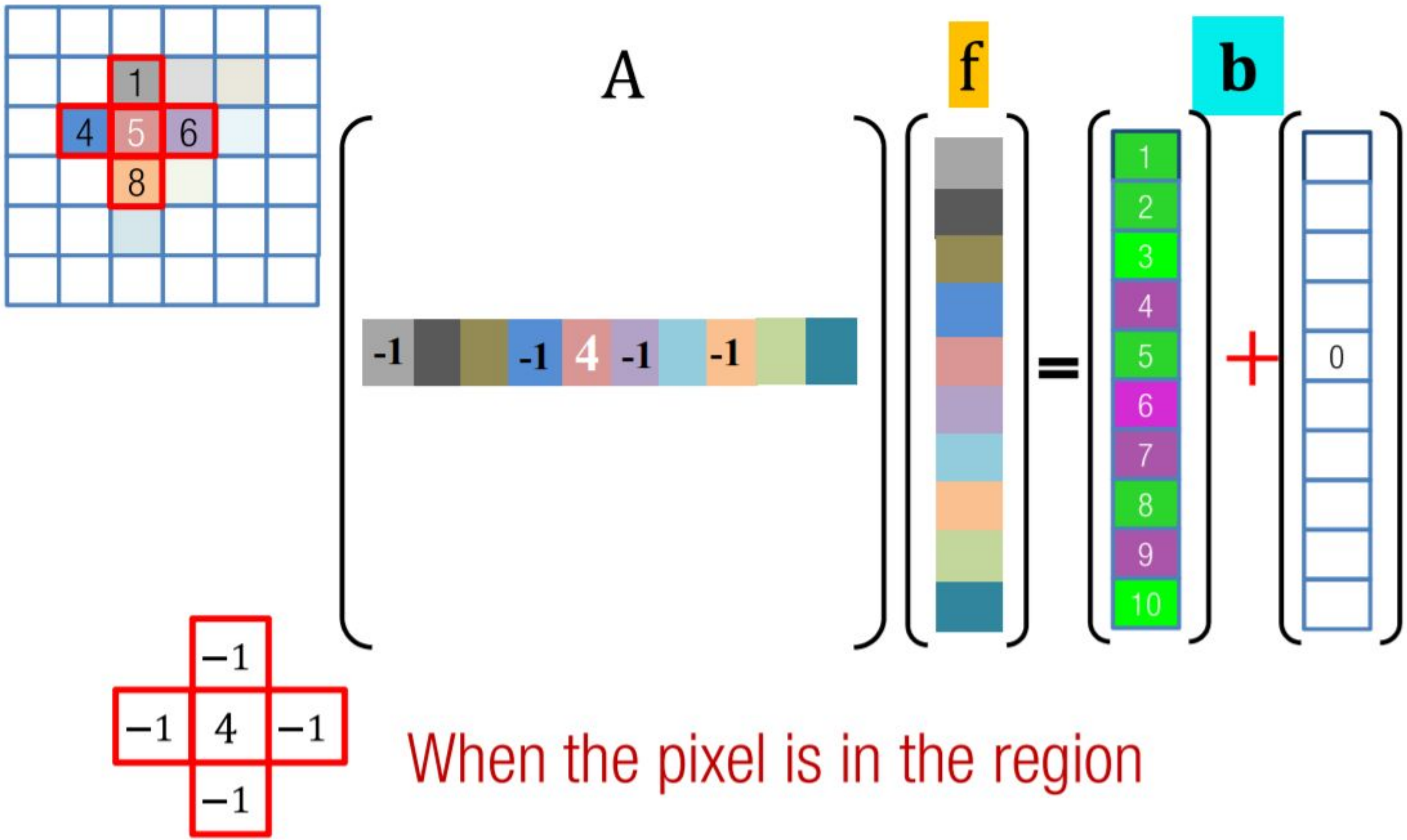
Probability of not building a correct model during k iterations: $(1-w^n)^k$

$(1-w^n)^k = 1-p$ where p is desired RANSAC success rate. $k = \frac{\log(1-p)}{\log(1-w^n)}$

Copying and *reshaping* the Laplacian of source

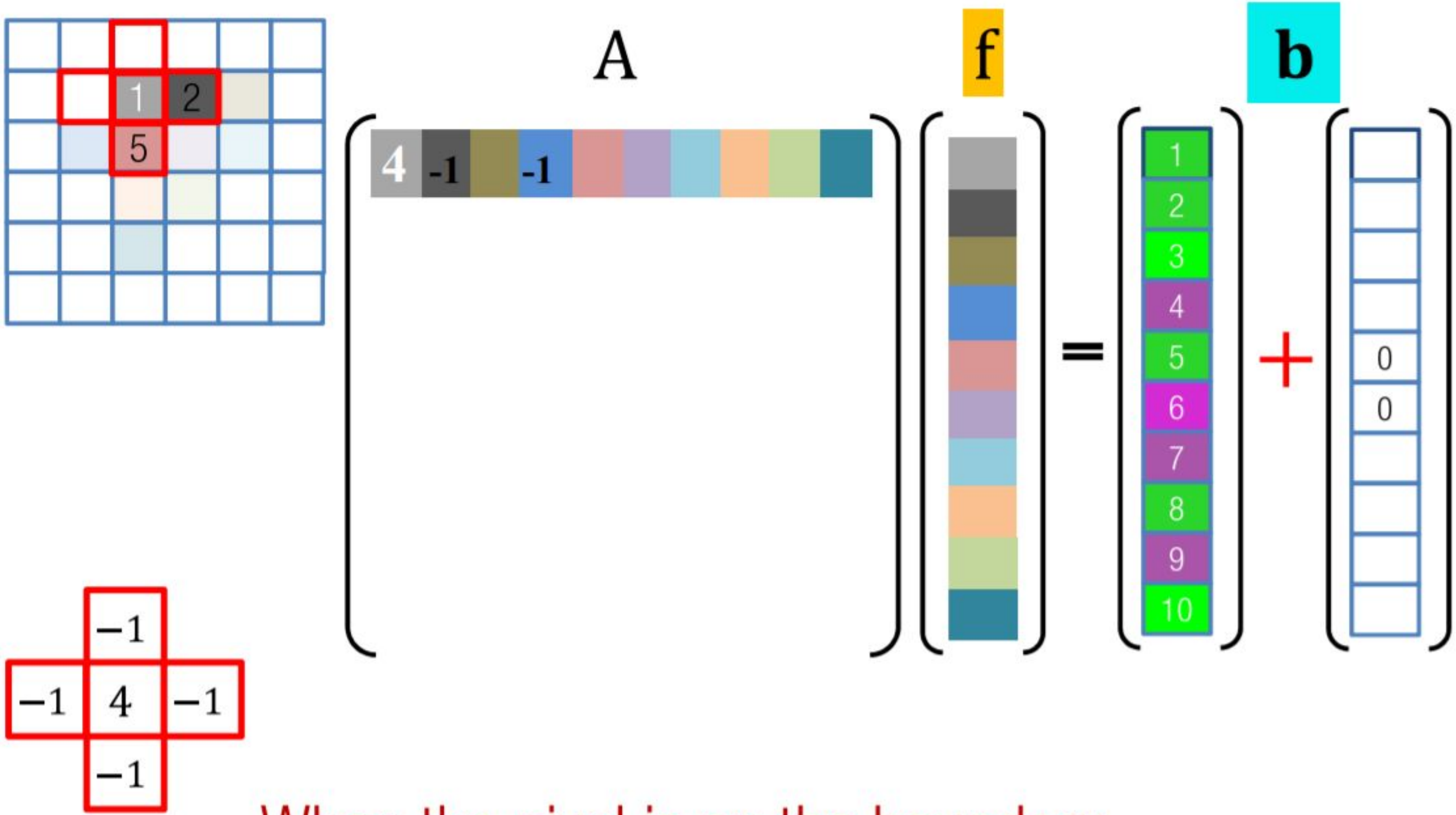


Matrix $\mathbf{A}$ encoding the Laplacian operator



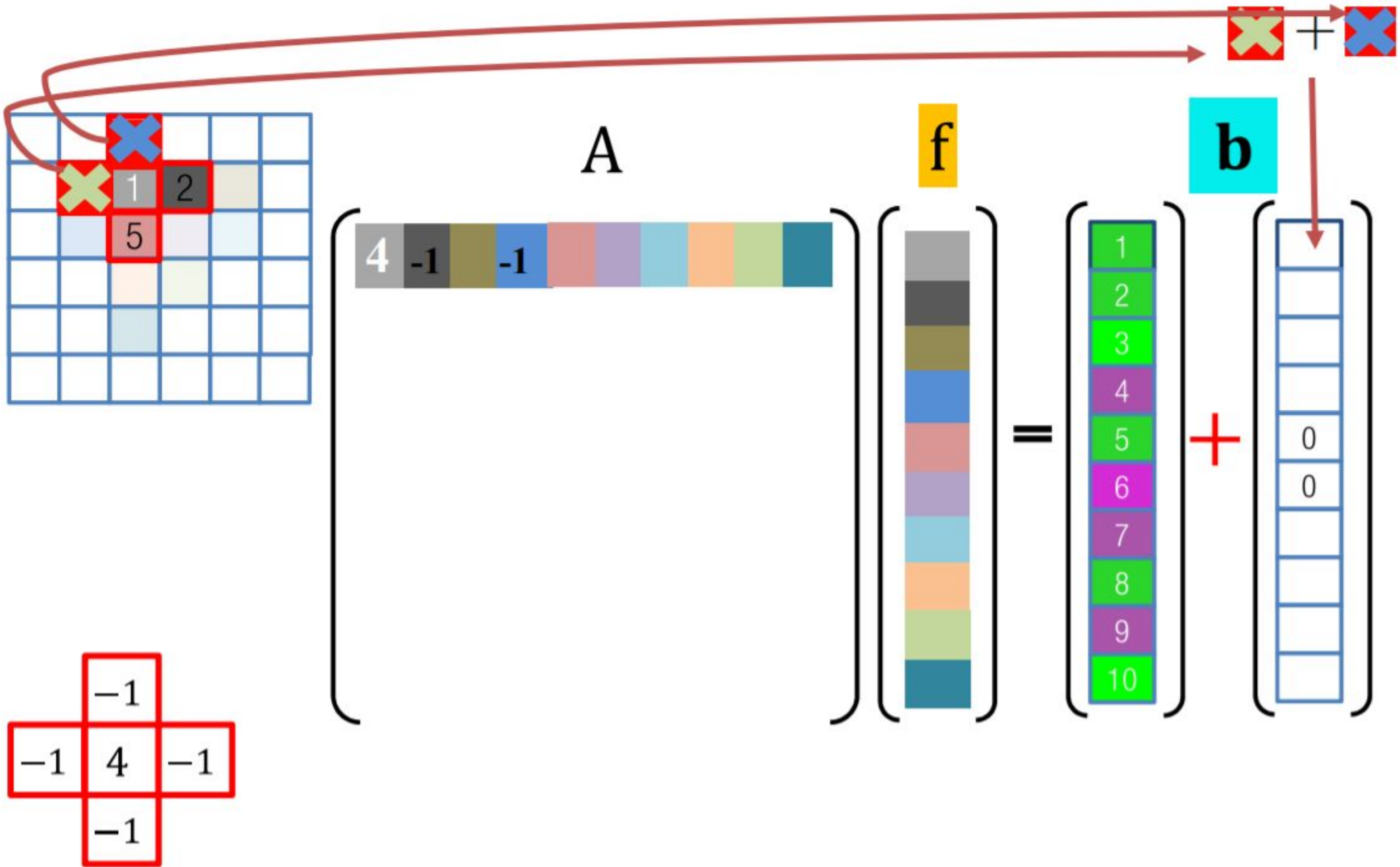
When the pixel is in the region

Matrix A : Laplacian operator



When the pixel is on the boundary

Move the boundary value of knowns to **b** side!



Optical Flow

Problem Definition

Given two consecutive image frames,
estimate the motion of each pixel

Assumptions

Brightness constancy $I(x(t), y(t), t) = C$

constant

Small motion

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

Brightness Constancy

flow velocities

$$I_x u + I_y v + I_t = 0$$

Image gradients
(at a point p)

temporal gradient

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

KLT Tracking

Summary of KLT tracking

- Find a good point to track (harris corner)
- Use intensity second moment matrix and difference across frames to find displacement
- Iterate and use coarse-to-fine search to deal with larger movements
- When creating long tracks, check appearance of registered patch against appearance of initial patch to find points that have drifted