SE 461 Computer Vision

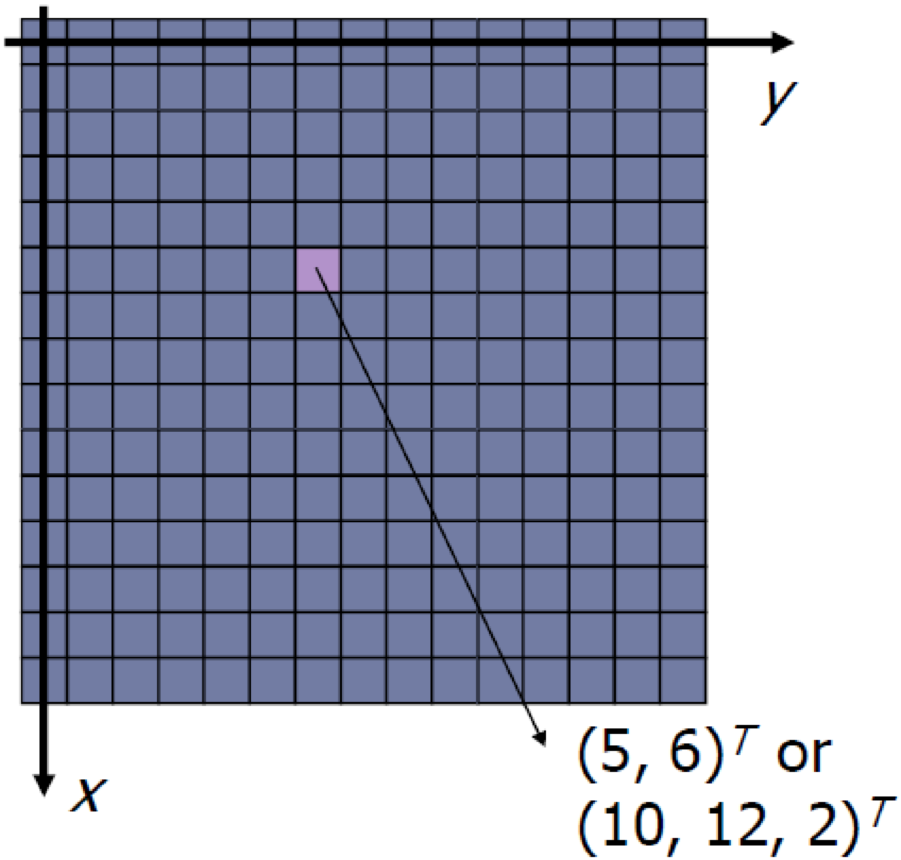
Nazar Khan
PUCIT
Lectures 15, 16 and 17

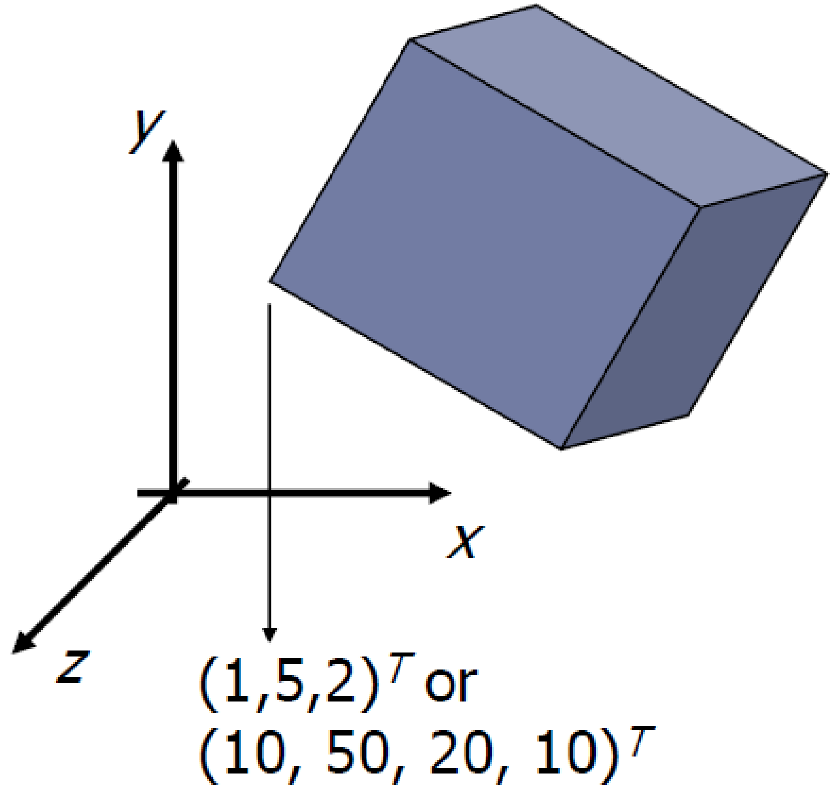
Transformations

- 2D-to-2D (image-to-image)
- 3D-to-3D (world-to-world)
- 3D-to-2D (camera model)
- 2D-to-3D (shape from X)
  - Shape from Stereo
  - Shape from Shading
  - Shape from Texture
  - Structure from Motion

Courtesy: Sohaib Khan

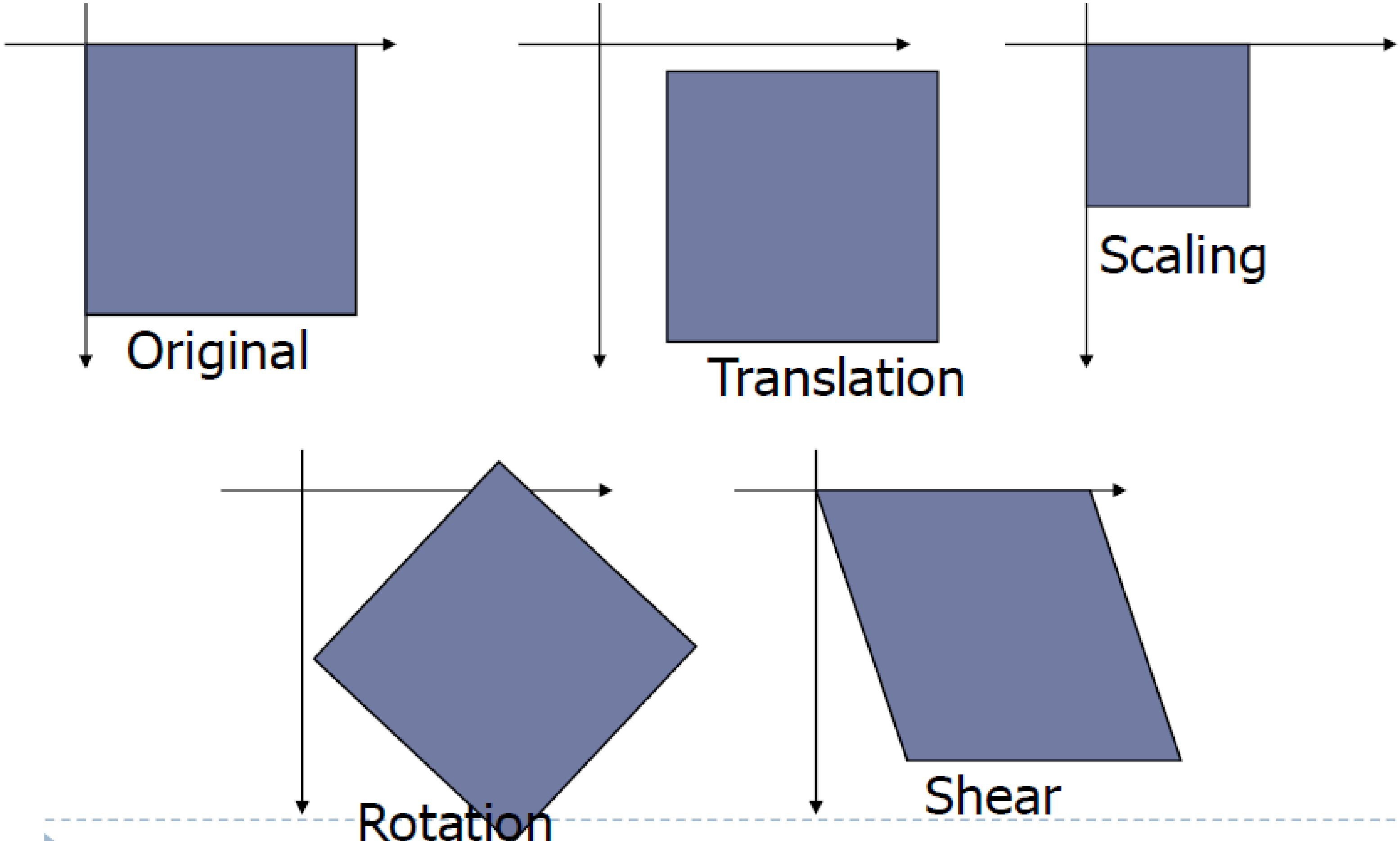
Points





Courtesy: Sohaib Khan

2-D Transformations



Courtesy: Sohaib Khan

2D Transformations

- Basic operation of all 2D transformations is simple

Point to be transformed: [*x*, *y*]

Point after transformation: [*x'*, *y'*]

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 x & a_2 y \\ a_3 x & a_4 y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

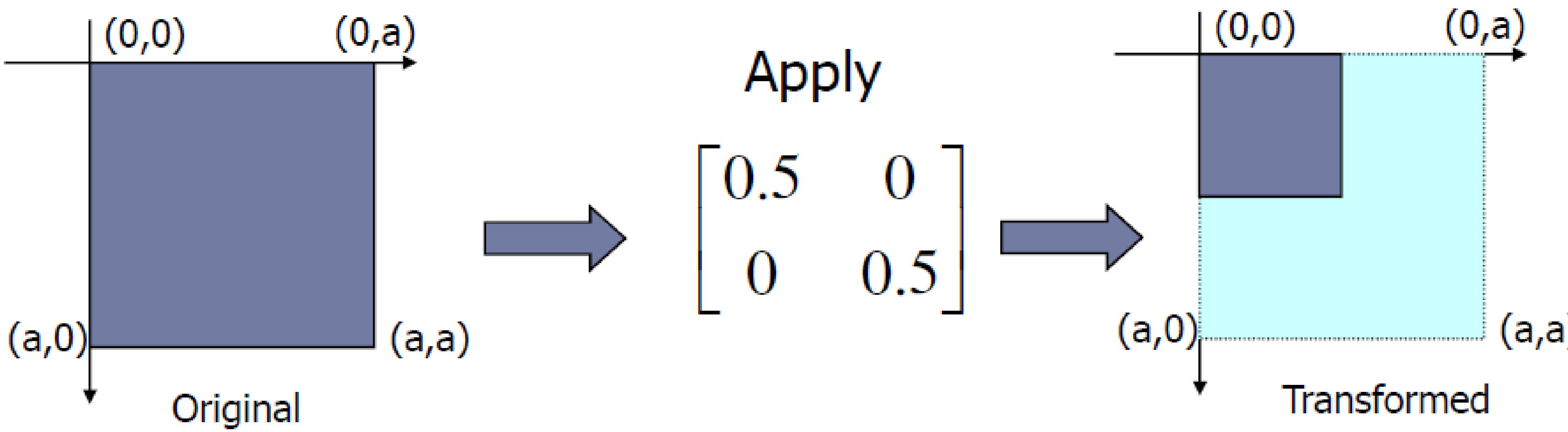
Transformation Matrix

Position before transformation

Position after transformation

Courtesy: Sohaib Khan

Example



$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} = \begin{bmatrix} 0.5a \\ 0.5a \end{bmatrix}$$

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} a \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5a \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 0.5a \\ 0.5a \end{bmatrix} = \begin{bmatrix} 0.25a \\ 0.25a \end{bmatrix}$$

Courtesy: Sohaib Khan

2D Transformations

$$\begin{bmatrix} 1.5 & 0 \\ 0 & 0.5 \end{bmatrix} = ?$$

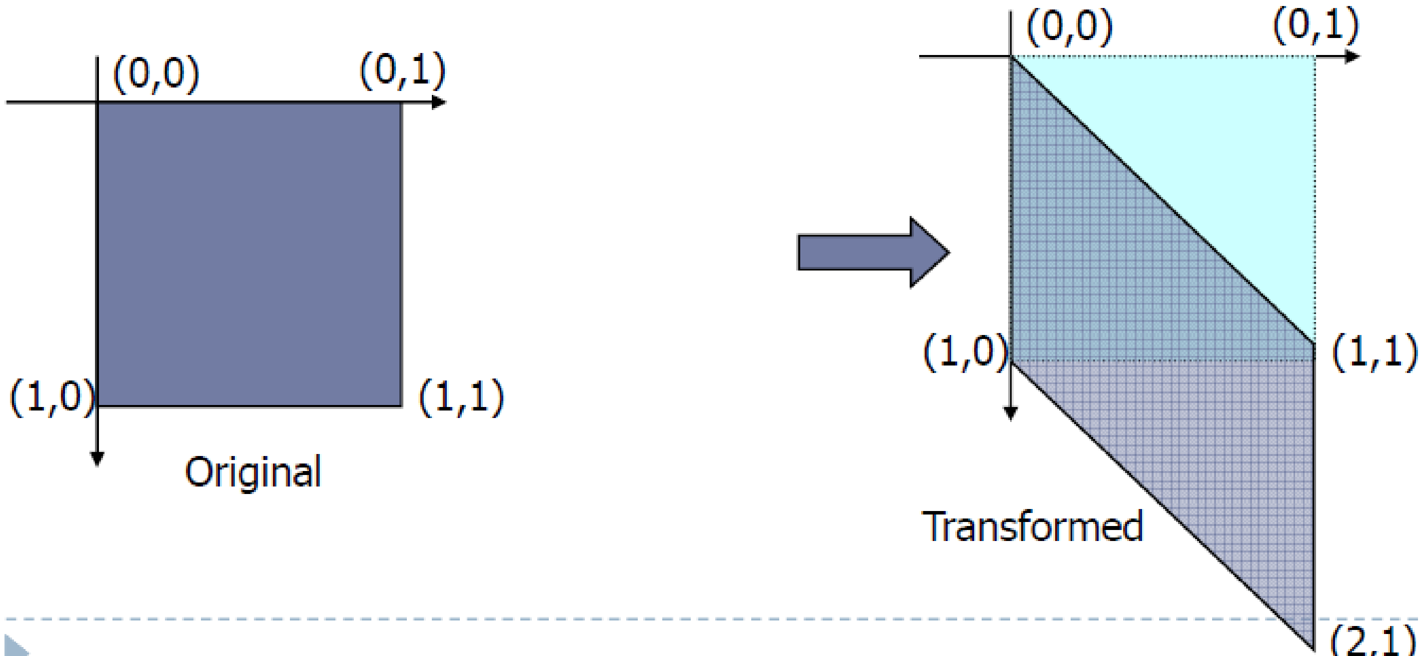
In general, scaling transformation is given by

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

Courtesy: Sohaib Khan

2D Transformations

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



Courtesy: Sohaib Khan

Shear in x-direction

$$\begin{bmatrix} 1 & e \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + ey \\ y \end{bmatrix}$$

- x-coordinate moves with an amount proportional to the y-coordinate

Shear in y-direction

$$\begin{bmatrix} 1 & 0 \\ e & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ ex + y \end{bmatrix}$$

- y-coordinate moves with an amount proportional to the x-coordinate

Courtesy: Sohaib Khan

2D Transformations

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

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

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

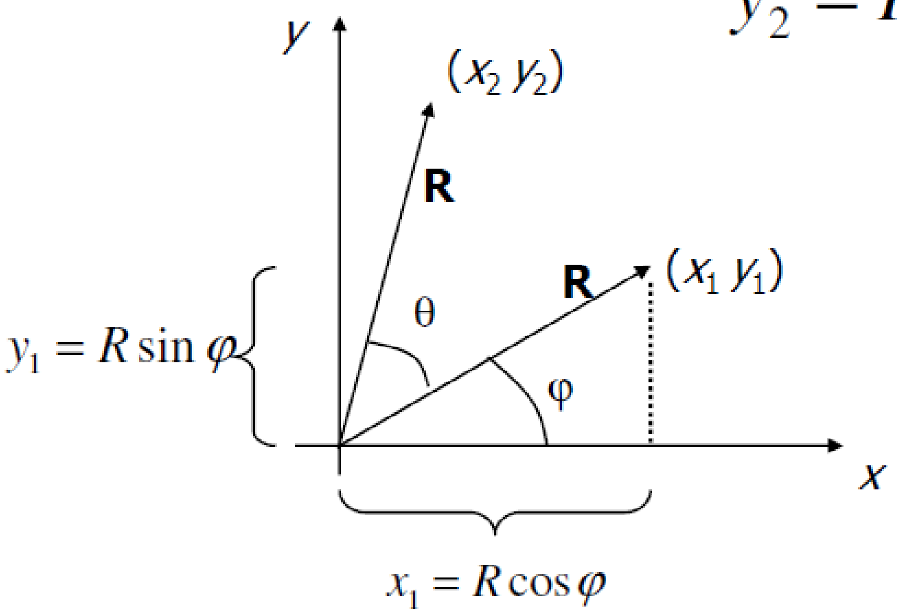
Reflection is negative scaling

Courtesy: Sohaib Khan

Rotation

$$x_2 = R\cos(\theta + \varphi)$$
$$y_2 = R\sin(\theta + \varphi)$$

$$x_2 = R\cos\theta\cos\varphi - R\sin\theta\sin\varphi$$
$$y_2 = R\sin\theta\cos\varphi + R\cos\theta\sin\varphi$$



$$x_2 = x_1\cos\theta - y_1\sin\theta$$
$$y_2 = x_1\sin\theta + y_1\cos\theta$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

R is rotation by θ **counterclockwise about origin**

Courtesy: Sohaib Khan

Rotation

- Rotation Matrix has some special properties
  - Each row/column has norm of 1 [prove]
  - Each row/column is orthogonal to the other [prove]
  - So Rotation matrix is an **orthonormal** matrix

Courtesy: Sohaib Khan

2D Translation

- Point in 2D given by $(x_1\ y_1)$
- Translated by $(d_x\ d_y)$

$$x_2 = x_1 + d_x$$

$$y_2 = y_1 + d_y$$

Courtesy: Sohaib Khan

Translation

- In matrix form

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- We could not have written **T** multiplicatively without using homogeneous coordinates

Courtesy: Sohaib Khan

Basic 2D Transformations

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

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

$$\begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & e_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ e_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Courtesy: Sohaib Khan

Inverse Transforms

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

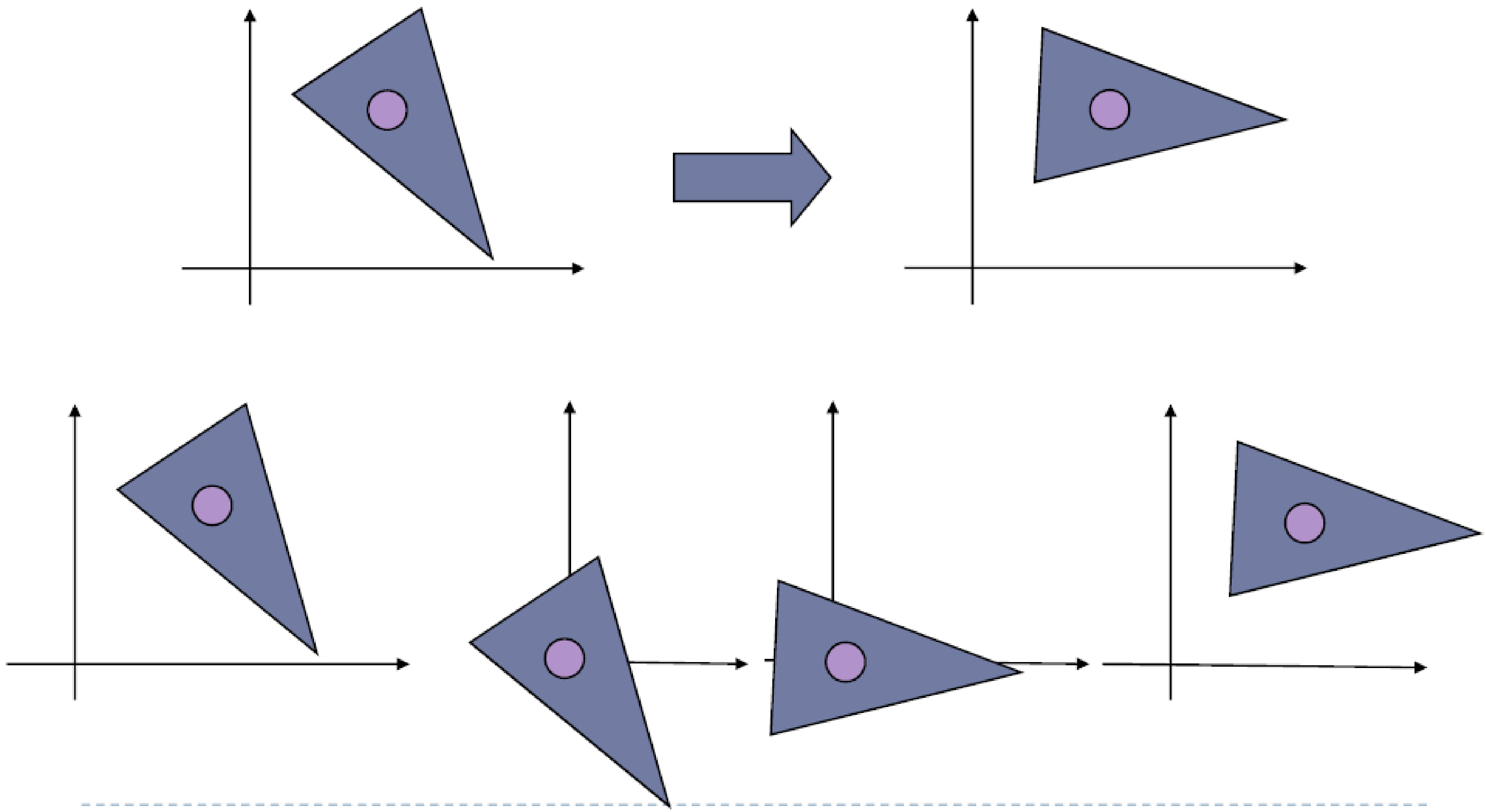
$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}\,\mathbf{S}^{-1} = \boldsymbol{I}$$

What is Inverse of Rotation?
What is inverse of Translation?
What is inverse of Shear in X-direction?
What is inverse of Shear in Y-direction?

Courtesy: Sohaib Khan

Rotation about an Arbitrary Point



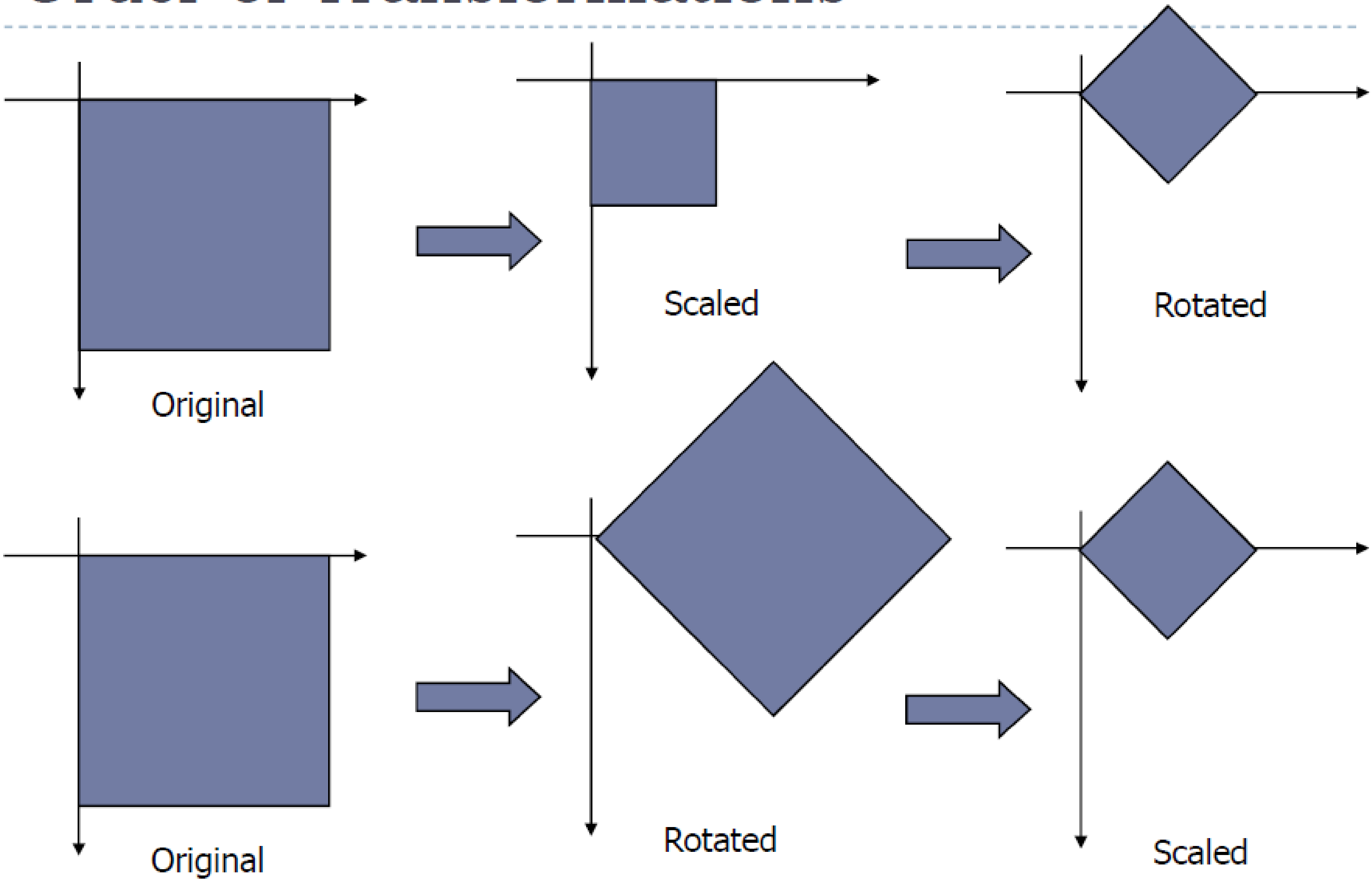
Courtesy: Sohaib Khan

Concatenation or Composition of Transformations

- We can concatenate a large number of transformations into a single transformation
- $\mathbf{p}_2 = \mathbf{T}_{[dx\ dy]}\ \mathbf{S}_{[s\ s]}\mathbf{R}_{\theta}\,\mathbf{p}_1$
- Rules of matrix multiplication apply
- If we do not use homogeneous coordinates, what might be the problem here?

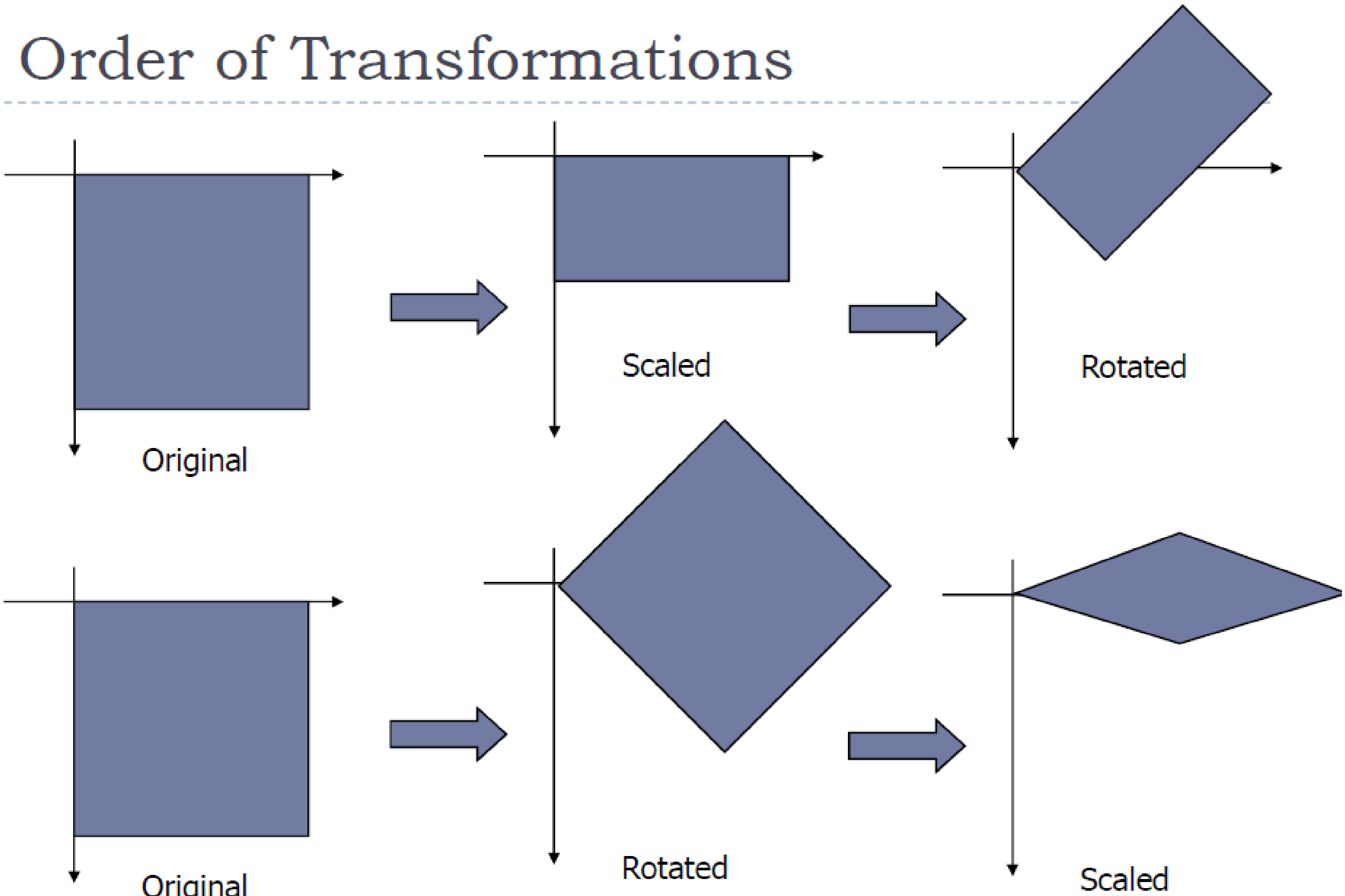
Courtesy: Sohaib Khan

Order of Transformations



Courtesy: Sohaib Khan

Order of Transformations



Courtesy: Sohaib Khan

Order of Transformations

- Rotation/Scaling/Shear, followed by Translation

$$\begin{bmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

- Translation, followed by Rotation/Scaling/Shear

$$\begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_1 b_1 + a_2 b_2 \\ a_3 & a_4 & a_3 b_1 + a_4 b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Courtesy: Sohaib Khan

Affine Transformation

- Encodes rotation, scaling, translation and shear

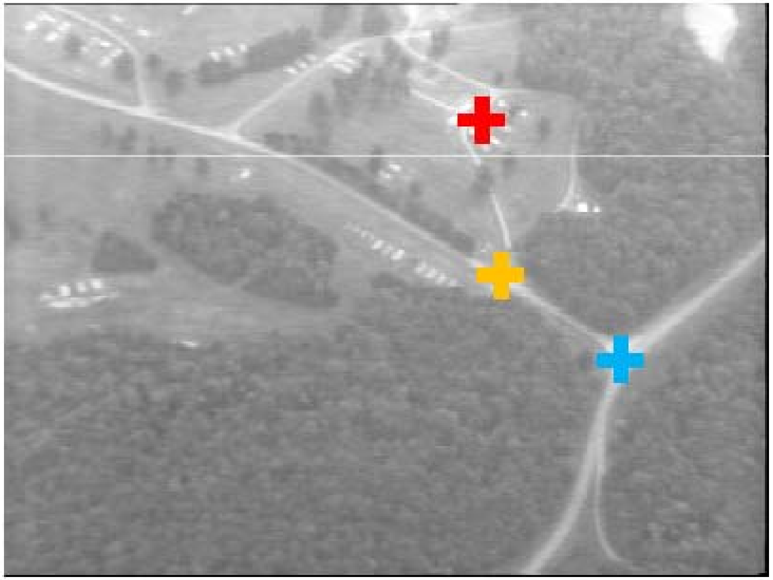
$$x_2 = a_1 x_1 + a_2 y_1 + b_1$$
$$y_2 = a_3 x_1 + a_4 y_1 + b_2$$

- 6 parameters
- Linear transformation
- Parallel lines are preserved [proof ?]

Courtesy: Sohaib Khan

Recovering Best Affine Transformation

- Input: we are given some correspondences
- Output: Compute $a_1 - a_6$ which relate the images



- This is an optimization problem… Find the 'best' set of parameters, given the input data

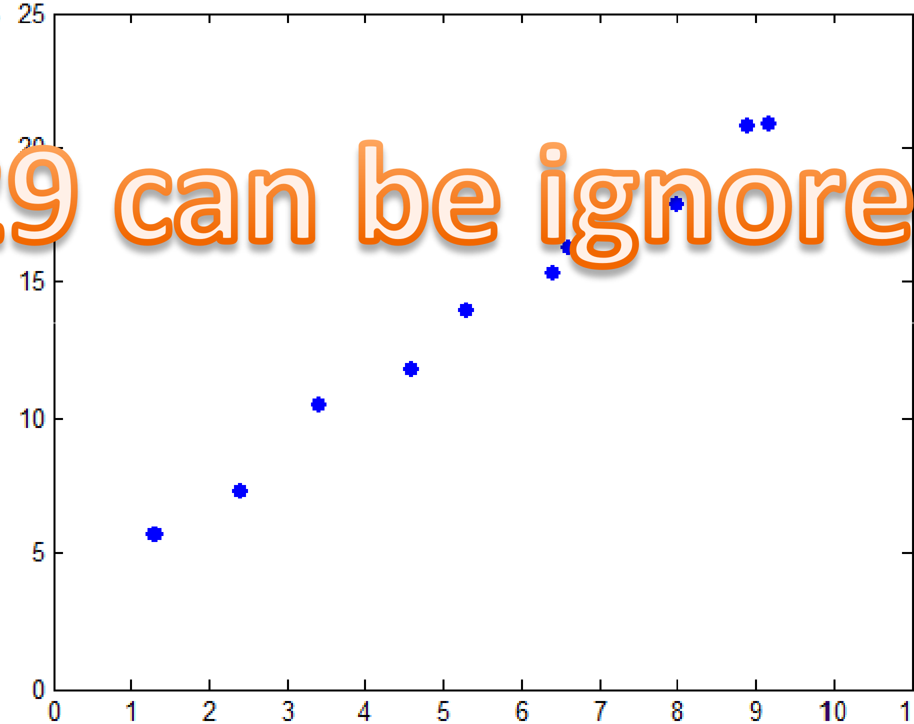
Courtesy: Sohaib Khan

Parameter Optimization: Least Squared Error Solutions

Slides 24 to 29 can be ignored

- Let us first consider the 'simpler' problem of fitting a line to a set of data points…

x	y
1.3	5.7
2.4	7.3
3.4	10.5
4.6	11.8
5.3	13.9
6.6	16.3
6.4	15.3
8.0	17.9
8.9	20.8
9.2	20.9



- Equation of best fit line ?

Courtesy: Sohaib Khan

Line Fitting: Least Squared Error Solution

- Step 1: Identify the model
  - Equation of line: $y = mx + c$
- Step 2: Set up an error term which will give the goodness of every point with respect to the (unknown) model

Slides 24 to 29 can be ignored

  - Error induced by i^{th} point:
  - $$e_i = mx_i + c - y_i$$
  - Error for whole data: $E = \Sigma_i e_i^2$
  - $$E = \Sigma_i (mx_i + c - y_i)^2$$
- Step 3: Differentiate Error w.r.t. parameters, put equal to zero and solve for minimum point

Courtesy: Sohaib Khan

Line Fitting: Least Squared Error Solution

$$E = \sum_i (mx_i + c - y_i)^2$$

$$\frac{\partial E}{\partial m} = \sum_i (mx_i + c - y_i)x_i = 0$$

Slides 24 to 29 can be ignored

$$\frac{\partial E}{\partial c} = \sum_i (mx_i + c - y_i) = 0$$

$$\begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & \sum_i 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{bmatrix}$$

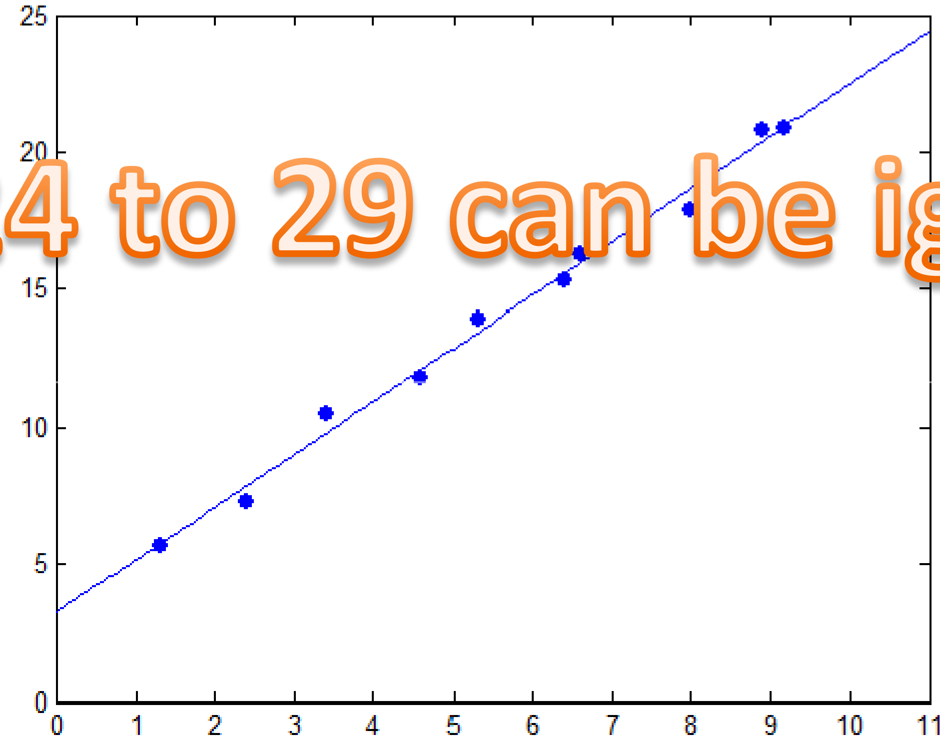
x	y
1.3	5.7
2.4	7.3
3.4	10.5
4.6	11.8
5.3	13.0
[illegible].6	[illegible]6.[illegible]
6.4	15.3
8.0	17.9
8.9	20.8
9.2	20.9

$$\begin{pmatrix} 380.63 & 56.1 \\ 56.1 & 10 \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} 914.68 \\ 140.4 \end{pmatrix}$$

Solution: $m = 1.9274$ $c = 3.227$

Courtesy: Sohaib Khan

Line Fitting: Least Squared Error Solution

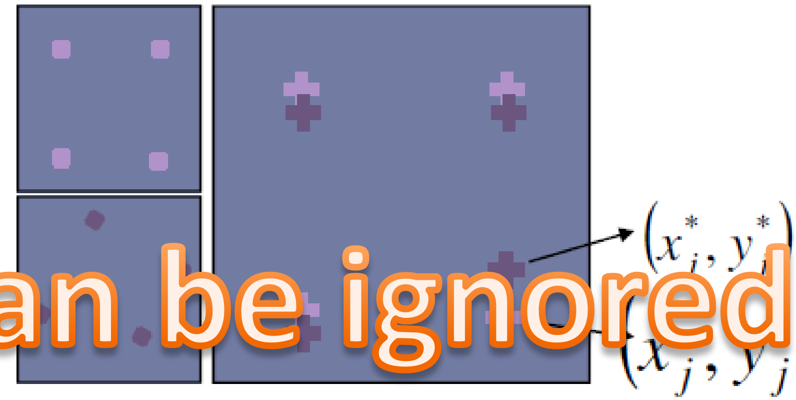


Slides 24 to 29 can be ignored

Courtesy: Sohaib Khan

Least Squares Error Solution

$$\begin{bmatrix} x_j^* \\ y_j^* \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \\ 1 \end{bmatrix}$$



Slides 24 to 29 can be ignored

$$E(a_1, a_2, a_3, a_4, a_5, a_6) = \sum_{j=1}^{n} (x_j^* - x_j)^2 + (y_j^* - y_j)^2$$

$$E(\mathbf{a}) = \sum_{j=1}^{n} \left((a_1 x'_j + a_2 y'_j + a_3 - x_j)^2 + (a_4 x'_j + a_5 y'_j + a_6 - y_j)^2 \right)$$

Courtesy: Sohaib Khan

Least Squares Error Solution

$$E(\mathbf{a}) = \sum_{j=1}^{n} \left((a_1 x_j + a_2 y_j + a_3 - x_j')^2 + (a_4 x_j + a_5 y_j + a_6 - y_j')^2 \right)$$

- Minimize E w.r.t. $\mathbf{a}$
- Compute $\partial E / \partial a$, put equal to zero, solve simultaneously

Slides 24 to 29 can be ignored

$$\begin{bmatrix} \sum_j x_j^2 & \sum_j x_j y_j & \sum_j x_j & 0 & 0 & 0 \\ \sum_j x_j y_j & \sum_j y_j^2 & \sum_j y_j & 0 & 0 & 0 \\ \sum_j x_j & \sum_j y_j & \sum_j 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum_j x_j^2 & \sum_j x_j y_j & \sum_j x_j \\ 0 & 0 & 0 & \sum_j x_j y_j & \sum_j y_j^2 & \sum_j y_j \\ 0 & 0 & 0 & \sum_j x_j & \sum_j y_j & \sum_j 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sum_j x_j x_j' \\ \sum_j y_j x_j' \\ \sum_j x_j' \\ \sum_j x_j y_j' \\ \sum_j y_j y_j' \\ \sum_j y_j' \end{bmatrix}$$

Courtesy: Sohaib Khan

Recovering Best Affine Transformation

- Given three pairs of corresponding points, we get 6 equations

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \end{bmatrix}$$

$$\mathbf{Ax=B} \qquad \mathbf{x=A^{-1}B}$$

Courtesy: Sohaib Khan

Pseudo inverse

For an over-constrained linear system

$$\mathbf{Ax = B}$$

A has more rows than columns

Multiply by $\mathbf{A^T}$ on both sides

$$\mathbf{A^T A x = A^T B}$$

$\mathbf{A^T A}$ is a square matrix of as many rows as **x**

We can take its inverse

$$\mathbf{x = (A^T A)^{-1} A^T B}$$

Pseudo-inverse gives the least squares error solution! [Proof?]

Courtesy: Sohaib Khan

Recovering Best Affine Transformation

- In general, we may be given *n* correspondences
- Concatenate *n* correspondences in **A** and **B**
- **A** is 2*n* x 6
- **B** is 2*n* x 1
- Solve using Least Squares
- $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}$

Courtesy: Sohaib Khan

2D Displacement Models

- Translation:

$$x' = x + b_1$$
$$y' = y + b_2$$

- Rigid:

$$x' = x\cos\theta - y\sin\theta + b_1$$
$$y' = x\sin\theta + y\cos\theta + b_2$$

- Affine:

$$x' = a_1 x + a_2 y + b_1$$
$$y' = a_3 x + a_4 y + b_2$$

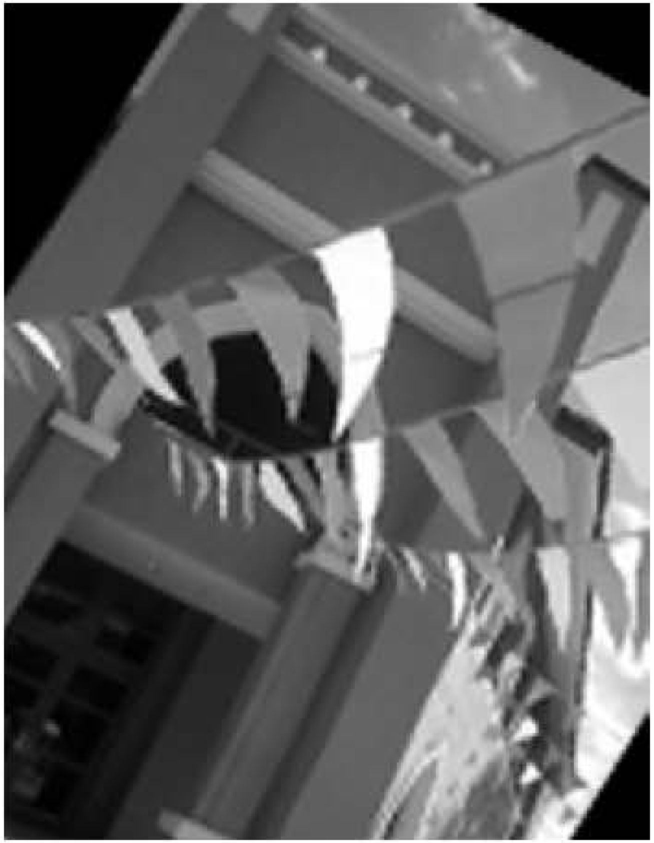
- Projective:

$$x' = \frac{a_1 x + a_2 y + b_1}{c_1 x + c_2 y + 1}$$

$$y' = \frac{a_3 x + a_4 y + b_2}{c_1 x + c_2 y + 1}$$

Courtesy: Sohaib Khan

2D Affine Warping



Courtesy: Sohaib Khan

Warping

- Inputs:
  - Image X
  - Affine Transformation A = $[a_1\ a_2\ b_1\ a_3\ a_4\ b_2]^T$
- Output:
  - Generate X′ such that X′ = AX
- Obvious Process:
  - For each pixel in X
  - Apply transformation
  - At that location in X′, put the same color as at the original location in X
- Problems?

Courtesy: Sohaib Khan

Warping

- This will leave holes…
  - Because every pixel does not map to an integer location!
- Reverse Transformation
- For each integer location in X'
- Apply inverse mapping
  - Problem?
- Will not result in answers at integer locations, in general
- Bilinearly interpolate from 4 neighbors

Courtesy: Sohaib Khan

2D Bilinear Interpolation

- Four nearest points of (x, y)

$$(\underline{x}, \underline{y}), (\underline{x}, \overline{y}), (\overline{x}, \underline{y}), (\overline{x}, \overline{y})$$

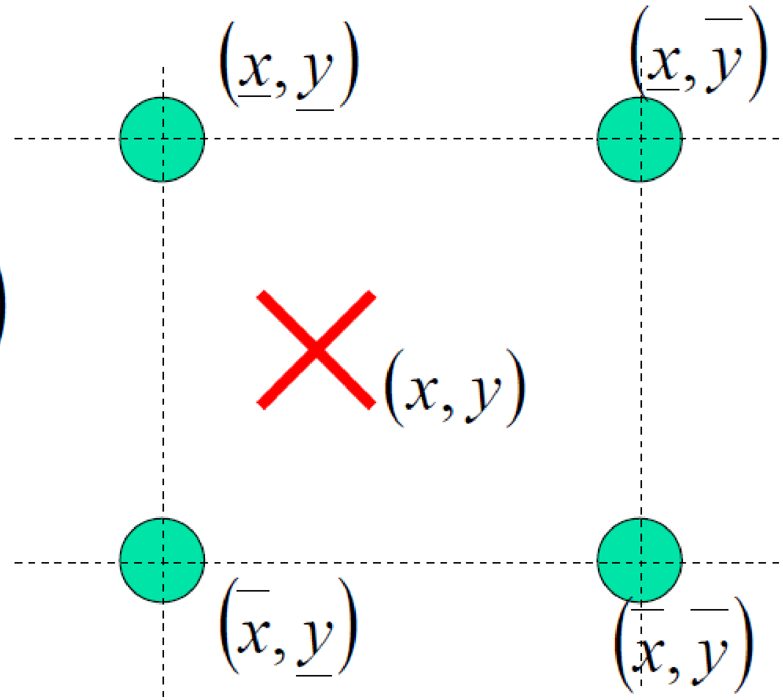
where

$$\underline{x} = \text{int}(x)$$

$$\underline{y} = \text{int}(y)$$

$$\overline{x} = \underline{x} + 1$$

$$\overline{y} = \underline{y} + 1$$



Courtesy: Sohaib Khan

Bilinear Interpolation

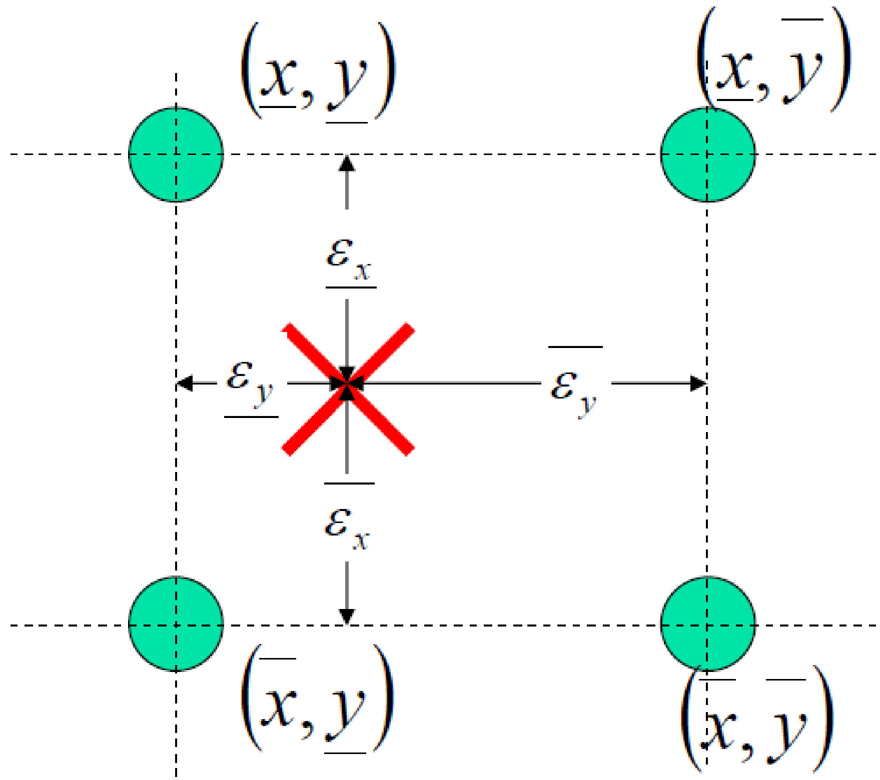
$$f'(x,y) = \overline{\varepsilon_x}\,\overline{\varepsilon_y}\,f(\underline{x},\underline{y}) + \underline{\varepsilon_x}\,\overline{\varepsilon_y}\,f(\overline{x},\underline{y}) + \overline{\varepsilon_x}\,\underline{\varepsilon_y}\,f(\underline{x},\overline{y}) + \underline{\varepsilon_x}\,\underline{\varepsilon_y}\,f(\overline{x},\overline{y})$$

$$\overline{\varepsilon_x} = \overline{x} - x$$

$$\overline{\varepsilon_y} = \overline{y} - y$$

$$\underline{\varepsilon_x} = x - \underline{x}$$

$$\underline{\varepsilon_y} = y - \underline{y}$$



Courtesy: Sohaib Khan

2-D Projective Transform

- Also called 2-D Homography or Colineation.
- 8 degrees of freedom because k*H represents the same homography as H.
- Invariants? Lines map to lines, but parallel lines can become non-parallel.
- Linear Transform?
  - In projective space, yes it's linear.
  - But it represents a non-linear operation in R^2.

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

Estimation of Homography: DLT

Direct Linear Transform (4.1 MVG Hartly & Zisserman)

$$\mathbf{x}'_i = \mathrm{H}\mathbf{x}_i$$

Projectively Equivalent
(Different scale, same direction)

$$\mathbf{x}'_i \times \mathrm{H}\mathbf{x}_i = \mathbf{0}$$

Cross product of vectors in same direction is zero

If the j-th row of the matrix H is denoted by $\mathbf{h}^{j\mathsf{T}}$, then we may write

$$\mathrm{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\mathsf{T}}\mathbf{x}_i \\ \mathbf{h}^{2\mathsf{T}}\mathbf{x}_i \\ \mathbf{h}^{3\mathsf{T}}\mathbf{x}_i \end{pmatrix}.$$

Estimation of Homography: DLT

Direct Linear Transform (4.1 MVG Hartly & Zisserman)

If the j-th row of the matrix H is denoted by $\mathbf{h}^{j\top}$, then we may write

$$\mathrm{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top}\mathbf{x}_i \\ \mathbf{h}^{2\top}\mathbf{x}_i \\ \mathbf{h}^{3\top}\mathbf{x}_i \end{pmatrix}.$$

$$\mathbf{a} = (a_1, a_2, a_3)^\top$$

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \left(\mathbf{a}^\top [\mathbf{b}]_\times\right)^\top.$$

Estimation of Homography: DLT

Direct Linear Transform (4.1 MVG Hartly & Zisserman)

If the j-th row of the matrix H is denoted by $\mathbf{h}^{j\mathsf{T}}$, then we may write

$$\mathrm{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\mathsf{T}}\mathbf{x}_i \\ \mathbf{h}^{2\mathsf{T}}\mathbf{x}_i \\ \mathbf{h}^{3\mathsf{T}}\mathbf{x}_i \end{pmatrix}.$$

Writing $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\mathsf{T}$, the cross product may then be given explicitly as

$$\mathbf{x}'_i \times \mathrm{H}\mathbf{x}_i = \begin{pmatrix} y'_i\mathbf{h}^{3\mathsf{T}}\mathbf{x}_i - w'_i\mathbf{h}^{2\mathsf{T}}\mathbf{x}_i \\ w'_i\mathbf{h}^{1\mathsf{T}}\mathbf{x}_i - x'_i\mathbf{h}^{3\mathsf{T}}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\mathsf{T}}\mathbf{x}_i - y'_i\mathbf{h}^{1\mathsf{T}}\mathbf{x}_i \end{pmatrix}.$$

Estimation of Homography: DLT

Direct Linear Transform (4.1 MVG Hartly & Zisserman)

Writing $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top$, the cross product may then be given explicitly as

$$\mathbf{x}'_i \times \mathrm{H}\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}^{3\top}\mathbf{x}_i - w'_i \mathbf{h}^{2\top}\mathbf{x}_i \\ w'_i \mathbf{h}^{1\top}\mathbf{x}_i - x'_i \mathbf{h}^{3\top}\mathbf{x}_i \\ x'_i \mathbf{h}^{2\top}\mathbf{x}_i - y'_i \mathbf{h}^{1\top}\mathbf{x}_i \end{pmatrix}.$$

$$\mathbf{h}^{j\top}\mathbf{x}_i = \mathbf{x}_i^\top \mathbf{h}^j$$

Dot product of two vectors a and b can be written as $a^T b$ or $b^T a$.

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{x}_i^\top \\ -y'_i \mathbf{x}_i^\top & x'_i \mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}.$$

Estimation of Homography: DLT

Direct Linear Transform (4.1 MVG Hartly & Zisserman)

3rd row is linear combination of first two

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \\ -y'_i\mathbf{x}_i^\top & x'_i\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}.$$

$$\mathbf{A}_i \mathbf{h} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}.$$

Estimation of Homography: DLT

- Direct Linear Transform (4.1 MVG Hartly & Zisserman)

$$\mathbf{A}_i \mathbf{h} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{x}_i^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}.$$

- How many correspondences?
- A is rank deficient!
- Null Space of A
- Invertibilty of H?
- Over determined system in the existence of noisy markings?

Projective Warping

- Same as affine.
- But now you MUST convert back to non-homogenous coordinates after applying the transformation.
  - Because 3^{rd} component will not necessarily be 1.