CS 565 Computer Vision

Nazar Khan

Lectures 12, 13 and 14: Spatial Transformations

Transformations

- We will study 2D spatial transformations

$$T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

- Affine transformations include
  - Scaling
  - Rotation
  - Shear
  - Translation
- Can be carried out via matrix-vector multiplications.

Matrices as linear operators

- Every matrix is a linear operator.
- Every matrix-vector multiplication represents a linear operation.

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1x + a_2y \\ a_3x + a_4y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

- Alternatively, **x**’=**Mx**.

Transformations

- Scaling $$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$
- Rotation $$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$
- Shear $$\begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix}$$
- Translation ???

Translation is not linear

- Translation is not a linear operation.
  - Try finding a matrix that takes [x;y] to [x+10;y].
- No matrix in $\mathbb{R}^{2x2}$ corresponds to a translation.
- However, a 3x3 matrix can be used to perform 2D translation. $\begin{bmatrix} 1 & 0 & Tr_x \\ 0 & 1 & Tr_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- So if we move to a higher dimensional space, we can make translations linear.

Projective Space $\mathbb{P}^2$

- Appending 1 as a 3rd coordinate corresponds to homogenous coordinates.

$$\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

- $\mathbb{R}^2 \rightarrow \mathbb{P}^2$ where $\mathbb{P}^2$ is the so-called **projective space**.
- Dimensionality of $\mathbb{P}^2$ is 3.
- Dimensionality of $\mathbb{P}^n$ is n+1.

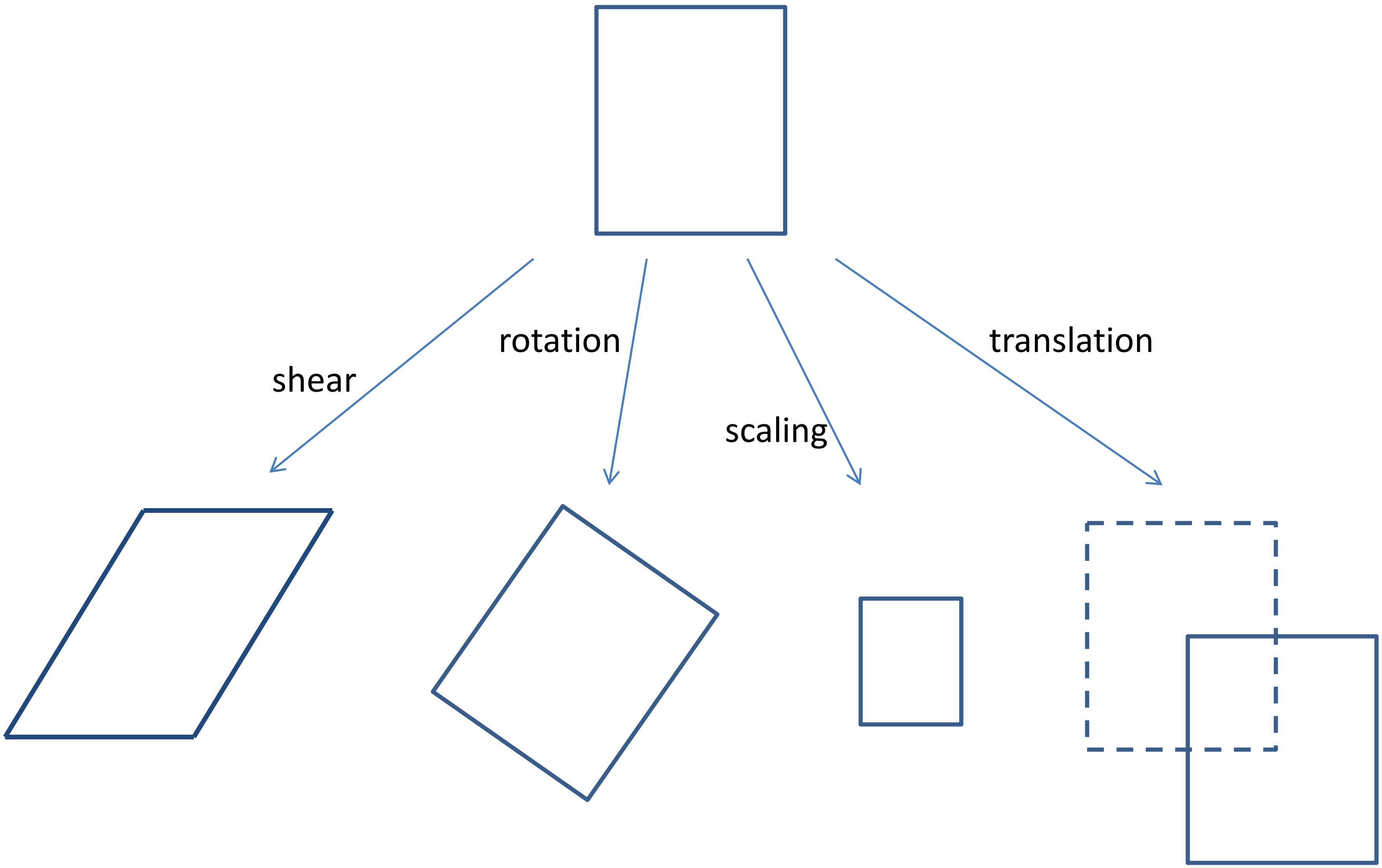
Projective Space $\mathbb{P}^2$

- $\mathbb{P}^2$ contains homogenised points from $\mathbb{R}^2$
- We go from $\mathbb{R}^2$ to $\mathbb{P}^2$ by appending a 3rd coordinate 1.
  - [x ; y] $\rightarrow$ [x ; y ; w] where w=1.
- We go back from $\mathbb{P}^2$ to $\mathbb{R}^2$ by dividing by 3rd coordinate and removing it.
  - [x ; y ; w] $\rightarrow$ [x/w ; y/w ; w/w] $\rightarrow$ [x/w ; y/w]

$\mathbb{P}^2$ vs. $\mathbb{R}^3$

- Both $\mathbb{P}^2$ and $\mathbb{R}^3$ are 3-dimensional.
- But $\mathbb{P}^2$ does not contain [0 ; 0 ; 0]. $\mathbb{P}^2 = \mathbb{R}^3 \setminus [0;0;0]$.
  - Because [0;0;0] → [0/0;0/0;0/0] → [NaN;NaN].
  - So [0;0;0] does not correspond to any point in $\mathbb{R}^2$.

2D Transformations



2D Transformations

- Basic operation of all 2D transformations is simple

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

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

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

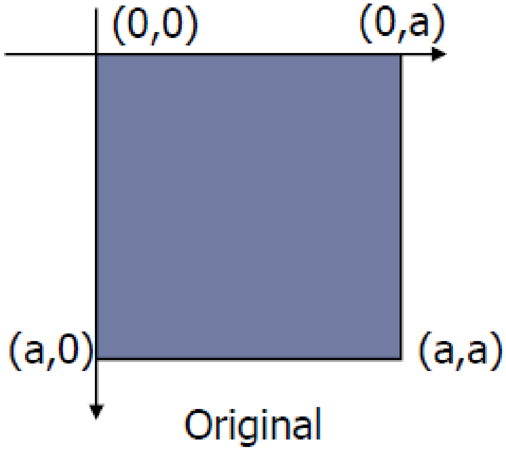
Transformation Matrix

Position before transformation

Position after transformation

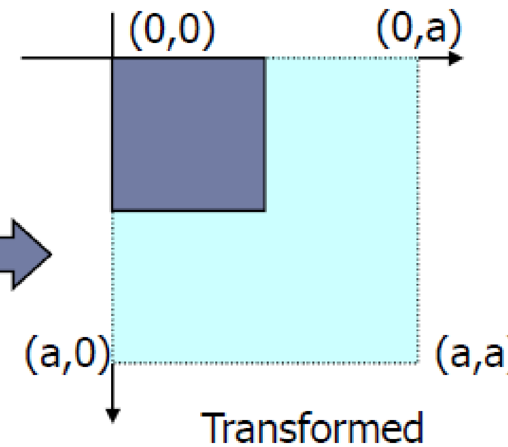
Courtesy: Sohaib Khan

Example



Apply

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



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

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

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

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

Courtesy: Sohaib Khan

2D Transformations

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

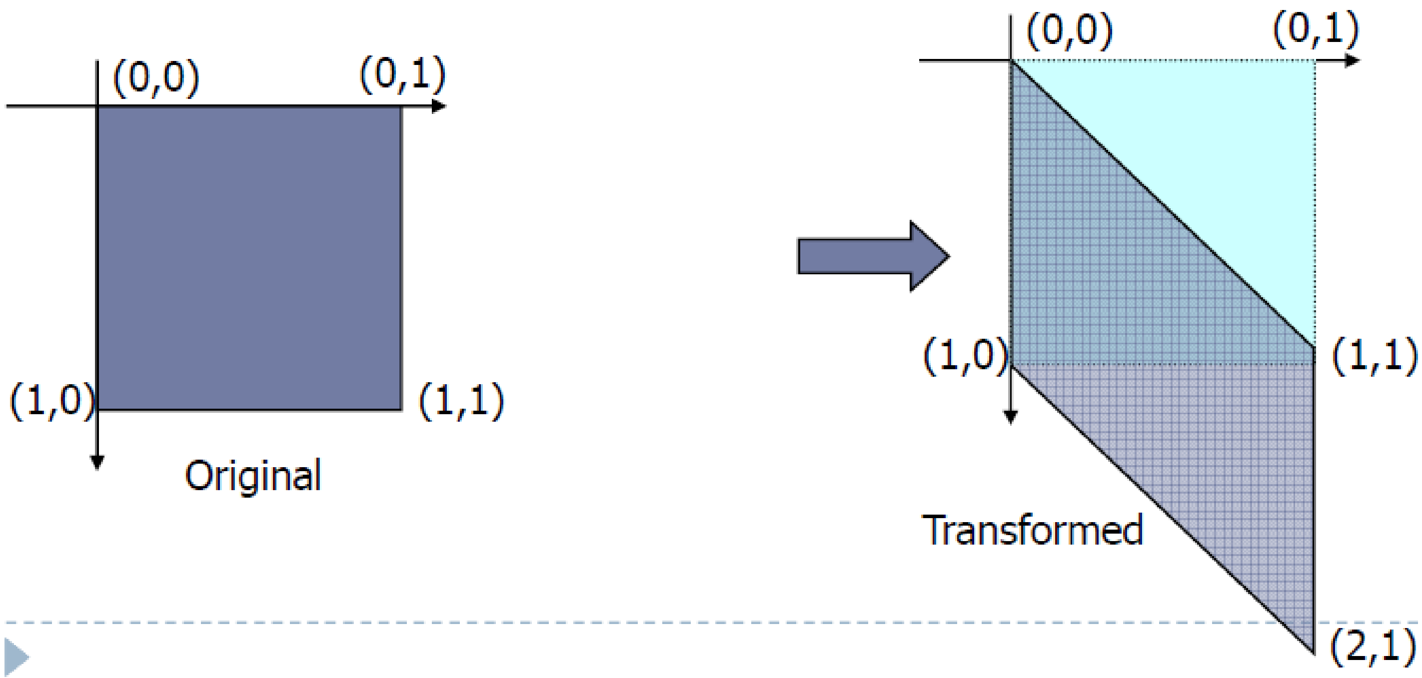
In general, scaling transformation is given by

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

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

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

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

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

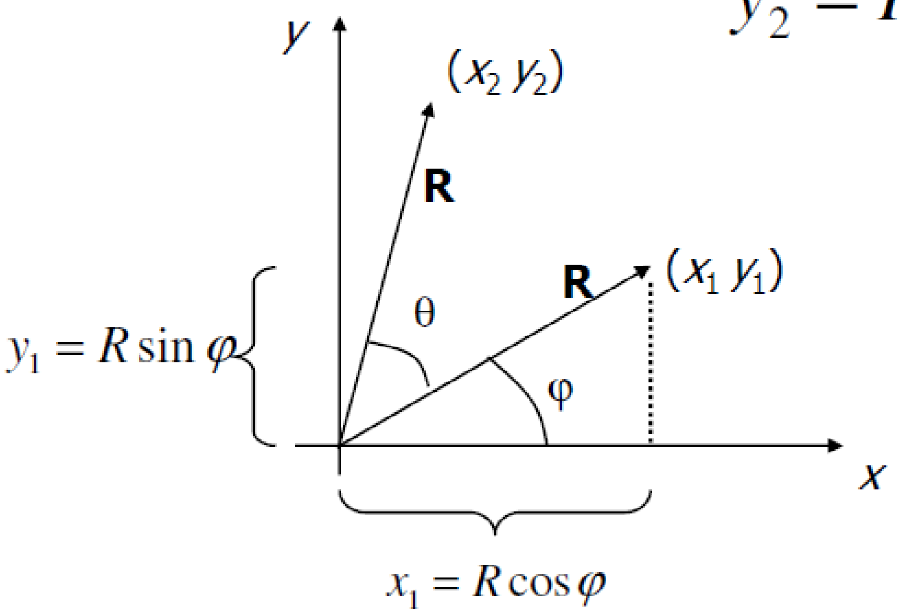
Reflection is negative scaling

Courtesy: Sohaib Khan

Rotation

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

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



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

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

$\boxed{\textbf{R} \text{ is rotation by } \theta \textbf{ counterclockwise about origin}}$

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

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

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

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

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?

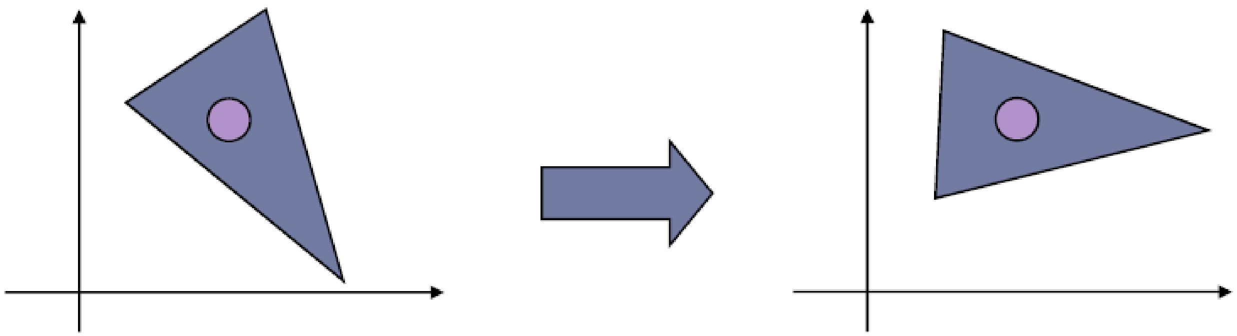
What is inverse of Translation?

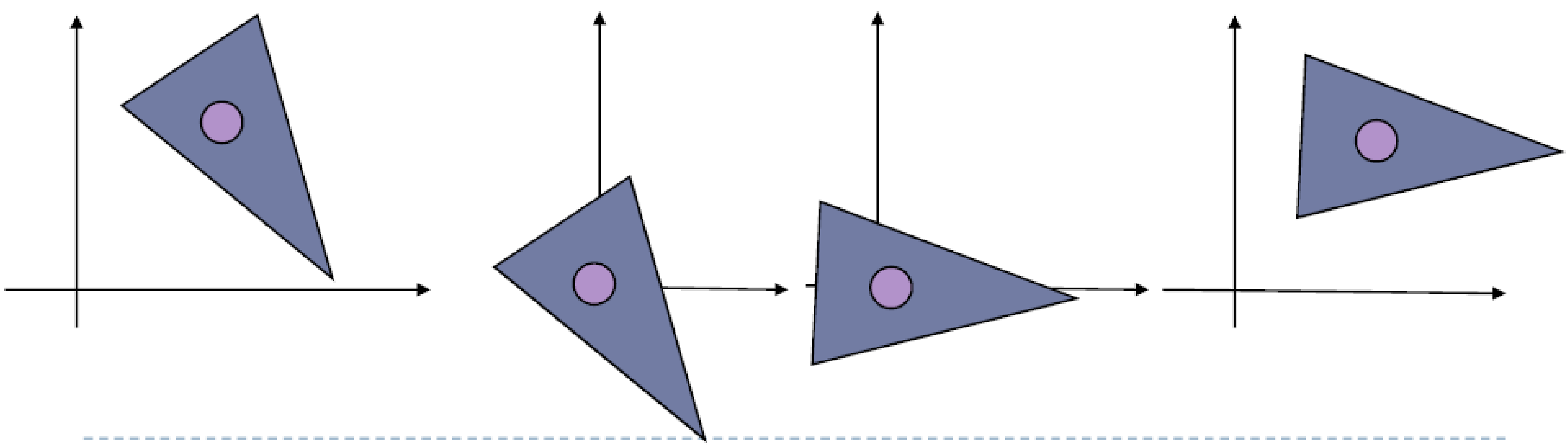
What is inverse of Shear in X-direction?

What is inverse of Shear in Y-direciton?

Courtesy: Sohaib Khan

Rotation about an Arbitrary Point





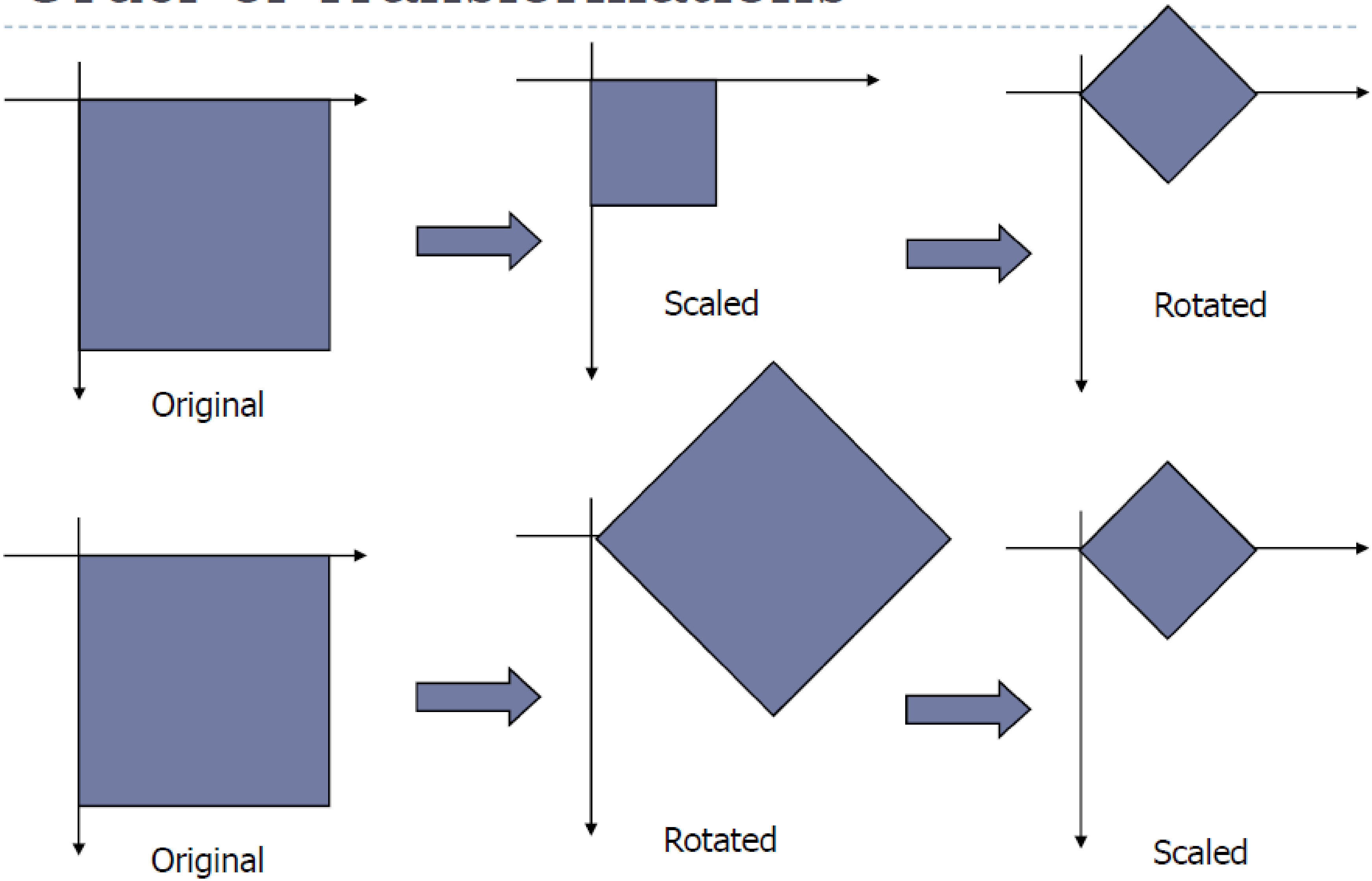
Courtesy: Sohaib Khan

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?

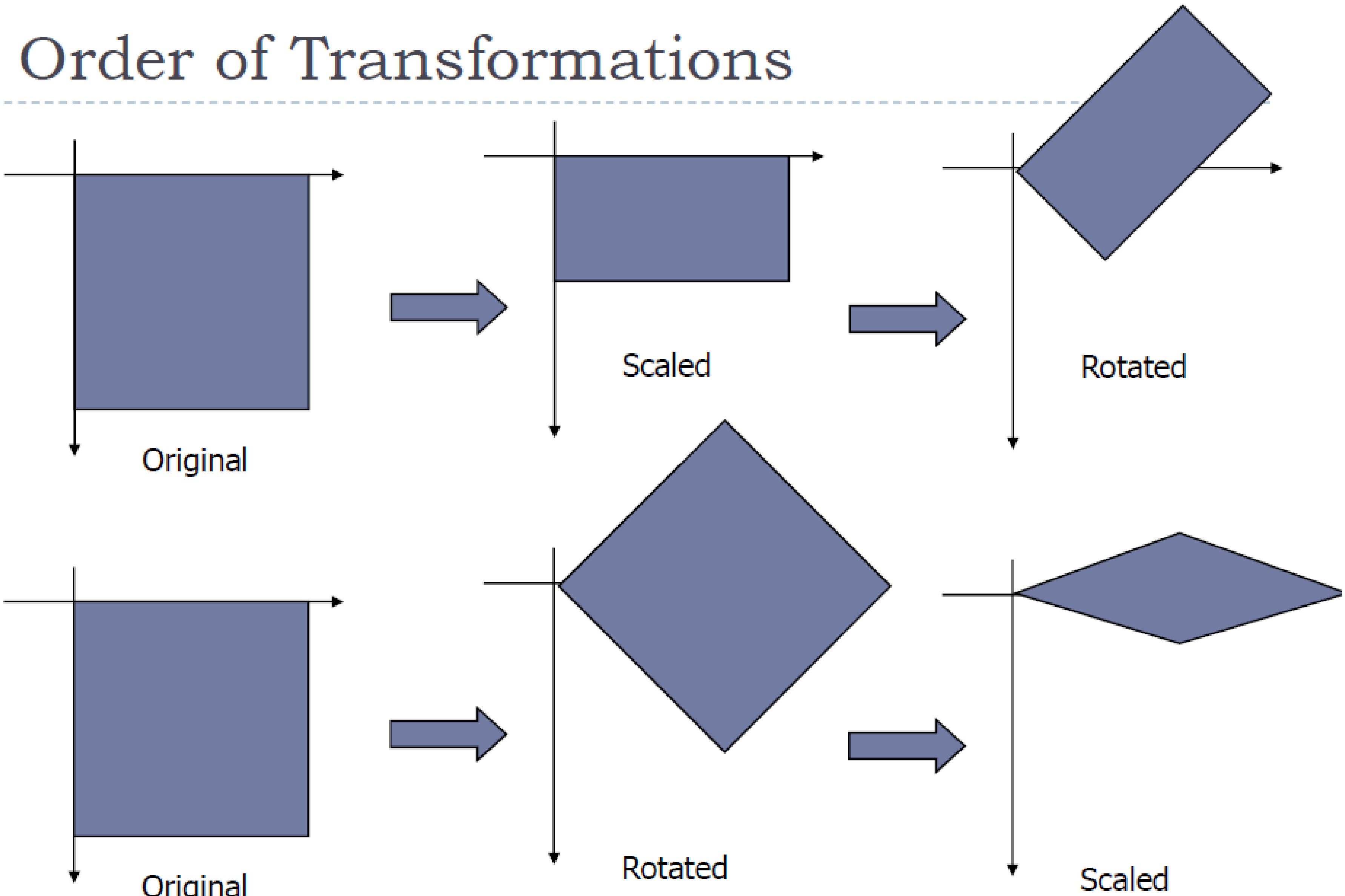
Courtesy: Sohaib Khan

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

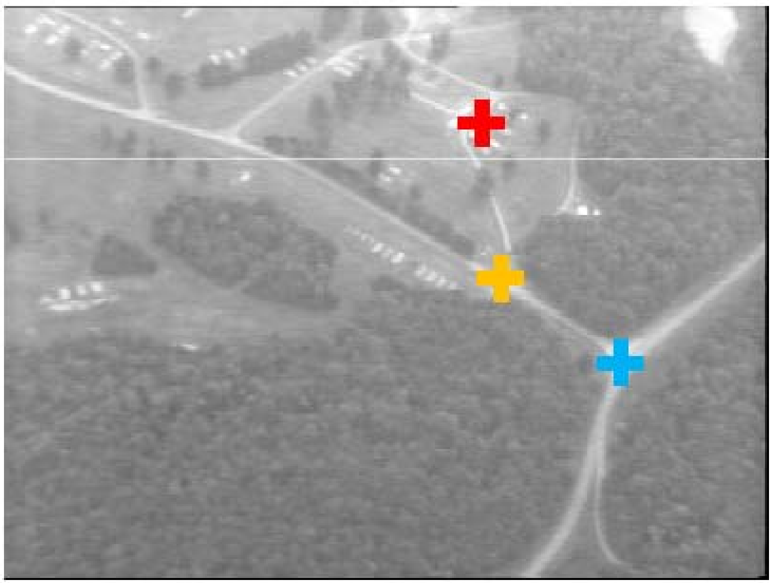
$$x_2 = a_1 x_1 + a_2 y_1 + b_1$$
$$y_2 = a_3 x_1 + a_4 y_1 + b_2$$

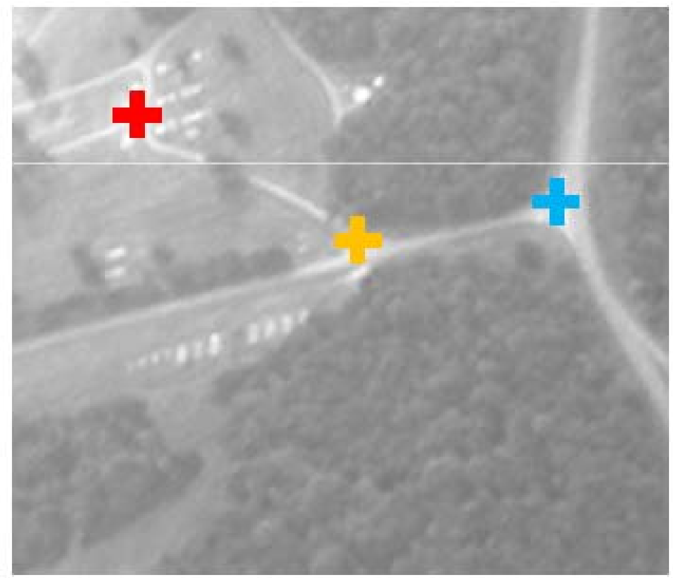
- 6 parameters
- Linear transformation
- Parallel lines are preserved [proof ?]

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

Recovering Best Affine Transformation

- Given 3 corresponding points $\mathbf{x}_1 \leftrightarrow \mathbf{x}_1'$, $\mathbf{x}_2 \leftrightarrow \mathbf{x}_2'$, $\mathbf{x}_3 \leftrightarrow \mathbf{x}_3'$ where $\mathbf{x}_i' = T\mathbf{x}_i$
- Find the 6 parameters [a_1;a_2;a_3;a_4;a_5;a_6] of the affine transformation T that maps **x** to **x'**.

$$x' = a_1 x + a_2 y + a_3$$
$$y' = a_4 x + a_5 y + a_6$$

can be written as

H.W. Verify.

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

- 1 correspondence yields 2 equations. So 3 correspondences will yield 6 equations which are enough to solve for 6 unknown parameters.

Recovering Best Affine Transformation

- The 3 correspondences can be written as

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

What are the sizes?

- So **v*** = A^{-1}**b**.
- But A**v**=**b** only for non-noisy measurements **x** and **x**'.
- Also, this works only when A is square and non-singular.

Recovering Best Affine Transformation

When

1. measurements are noisy, and/or
2. A is non-square (more than 3 correspondences)

we want to find $\mathbf{v}^*$ such that $A\mathbf{v}^*$ is as close as possible to $\mathbf{b}$. That is,

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} ||A\mathbf{v}-\mathbf{b}||^2$$

which is a least-squares problem.

Recovering Best Affine Transformation

At $\mathbf{v}^*$

$$\nabla_{\mathbf{v}} \{||A\mathbf{v}-\mathbf{b}||^2\} = \mathbf{0}$$

$$\Rightarrow \nabla_{\mathbf{v}} \{(A\mathbf{v}-\mathbf{b})^T(A\mathbf{v}-\mathbf{b})\} = \mathbf{0}$$

$$\Rightarrow 2A^T(A\mathbf{v}^*-\mathbf{b}) = \mathbf{0}$$ ← **Prove this. Not as simple as it looks.**

$$\Rightarrow A^T(A\mathbf{v}^*-\mathbf{b}) = \mathbf{0}$$

$$\Rightarrow A^TA\mathbf{v}^*-A^T\mathbf{b} = \mathbf{0}$$

$$\Rightarrow \mathbf{v}^* = (A^TA)^{-1}A^T\mathbf{b} = A^\dagger\mathbf{b}$$

The matrix $A^\dagger = (A^TA)^{-1}A^T$ is known as the **pseudo-inverse** of A.

Recovering Best Affine Transformation

Concise algorithm

Input: N point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$

1. Fill in the 2N x 6 matrix A using the $\mathbf{x}_i$
2. Fill in the 2N x 1 vector $\mathbf{b}$ using the $\mathbf{x}_i'$
3. Compute 6 x 6 pseudo-inverse $A^\dagger = (A^TA)^{-1}A^T$
4. Compute optimal affine transformation parameters as $\mathbf{v}^* = A^\dagger \mathbf{b}$

2D Displacement Models

- **Translation:**
$$x' = x + b_1$$
$$y' = y + b_2$$

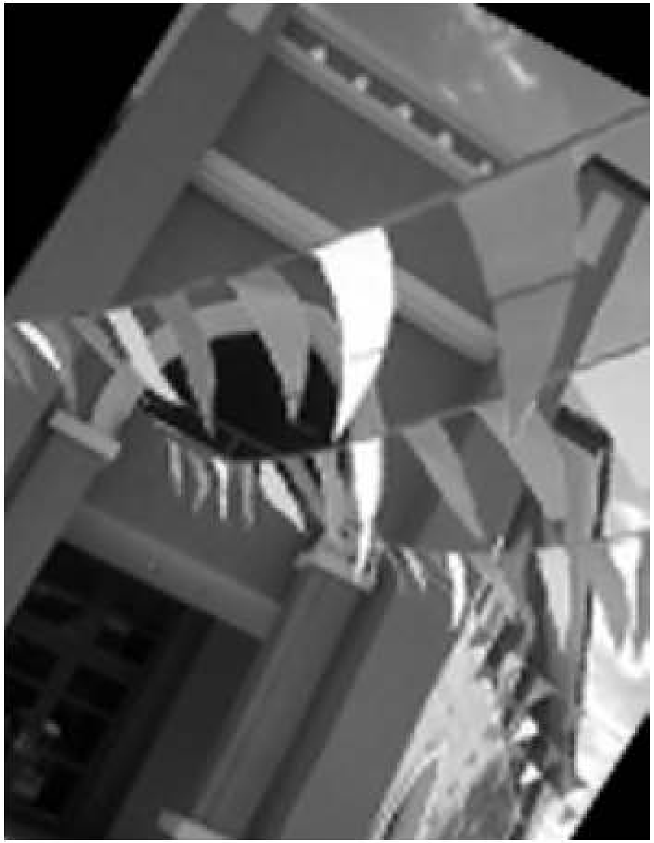
- **Rigid:**
$$x' = x\cos\theta - y\sin\theta + b_1$$
$$y' = x\sin\theta + y\cos\theta + b_2$$

- **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

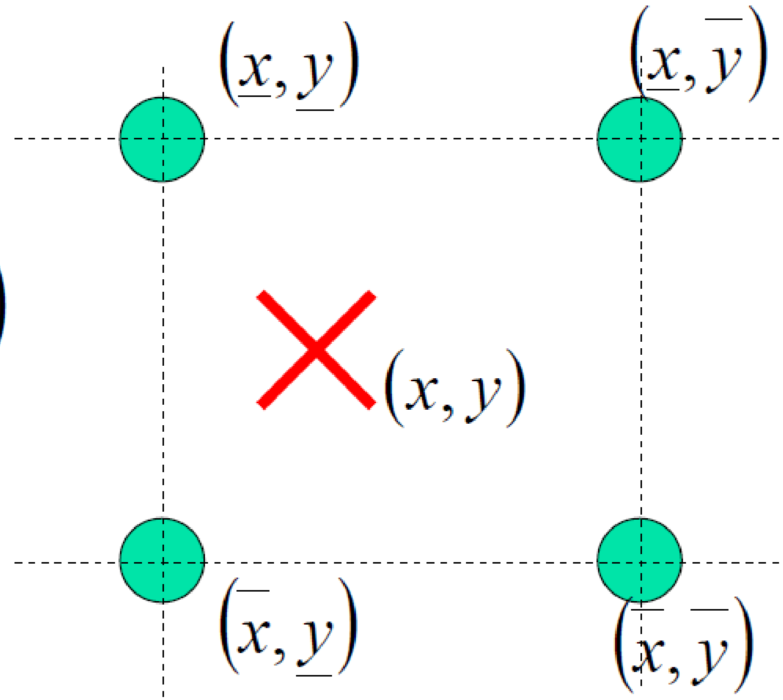
Courtesy: Sohaib Khan

2D Bilinear Interpolation

- Four nearest points of (x, y)

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

where
$$\underline{x} = \mathrm{int}(x)$$
$$\underline{y} = \mathrm{int}(y)$$
$$\overline{x} = \underline{x} + 1$$
$$\overline{y} = \underline{y} + 1$$



Courtesy: Sohaib Khan

Bilinear Interpolation

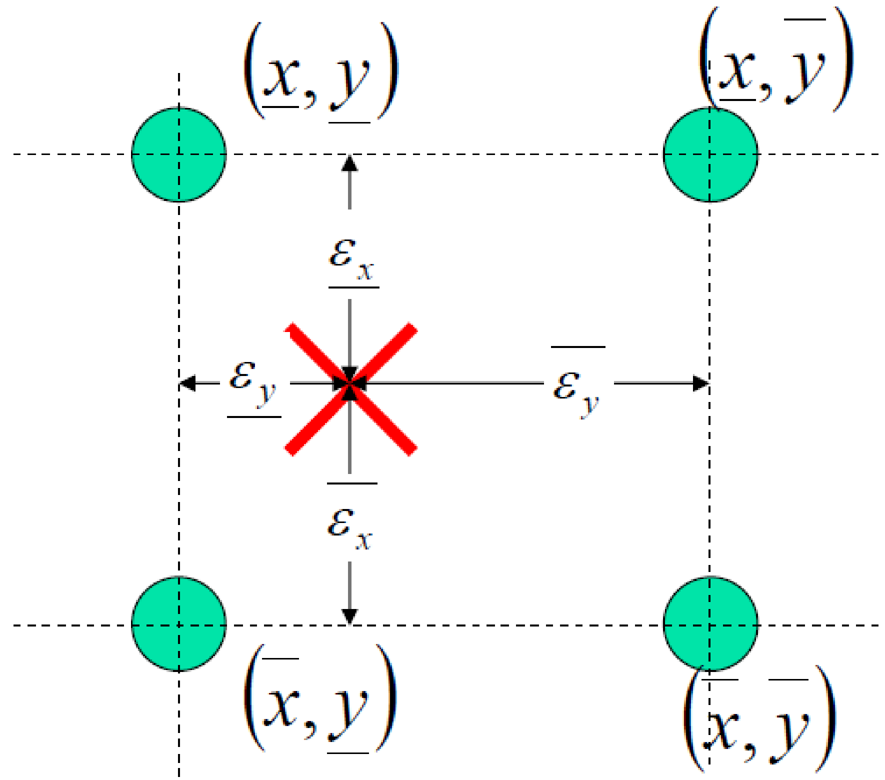
$$f'(x,y) = \overline{\varepsilon}_x \overline{\varepsilon}_y f(\underline{x},\underline{y}) + \underline{\varepsilon}_x \overline{\varepsilon}_y f(\overline{x},\underline{y}) + \overline{\varepsilon}_x \underline{\varepsilon}_y f(\underline{x},\overline{y}) + \underline{\varepsilon}_x \underline{\varepsilon}_y f(\overline{x},\overline{y})$$

$$\overline{\varepsilon}_x = \overline{x} - x$$

$$\overline{\varepsilon}_y = \overline{y} - y$$

$$\underline{\varepsilon}_x = x - \underline{x}$$

$$\underline{\varepsilon}_y = y - \underline{y}$$



Courtesy: Sohaib Khan

Beyond Affine – Projective Transformation

- Last row of affine transformation matrix is always [0 0 1].
- If this condition is relaxed we obtain the so-called **projective transformation**.
- Also called **homography** or **collineation**.
  - Lines are mapped to lines.
- 8 degrees of freedom. **Why?**
- Linear in $\mathbb{P}^2$.
- Non-linear in $\mathbb{R}^2$ because 3rd coordinate of **x**’ is not guaranteed to be 1.

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

$$x' = \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + h_9}$$

$$y' = \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + h_9}$$

Projective Transformation

- If $\mathbf{a} \in \mathbb{P}^2$ and $\mathbf{b} \in \mathbb{P}^2$ correspond to the same point in Cartesian space, then we say that **a** and **b** are **projectively equivalent**.
  - We write this as $\mathbf{a} \equiv \mathbf{b}$.
- In projective space, $\mathbf{v} \equiv k(\mathbf{v})$ for all $k \neq 0$ because both correspond to the same point in Cartesian space.
- So $k(H\mathbf{v}) \equiv H\mathbf{v} \Rightarrow kH\mathbf{v} \equiv H\mathbf{v} \Rightarrow kH \equiv H$.
- Let H'=H/H(3,3). Clearly, H'(3,3)=1 and therefore H' has 8 free parameters.
- But since $H' \equiv H$, <u>H must also have 8 free parameters</u>.

Recovering Best Projective Transform

- Given N corresponding points $\mathbf{x}_1 \leftrightarrow \mathbf{x}_1'$, $\mathbf{x}_2 \leftrightarrow \mathbf{x}_2'$, …, $\mathbf{x}_N \leftrightarrow \mathbf{x}_N'$ where $\mathbf{x}_i' \equiv H\mathbf{x}_i$ Why projectively equivalent?
- Find the 8 parameters $[h_1;h_2;h_3;h_4;h_5;h_6;h_7;h_8]$ of the projective transformation H that maps **x** to **x'**.
- 8 unknown parameters will require 8 equations.

Recovering Best Projective Transform

- $\mathbf{x}_i' \equiv H\mathbf{x}_i \Rightarrow$ both vectors point in the same direction.
- So cross-product $\mathbf{x}_i' \times H\mathbf{x}_i = \mathbf{0}_{3x1}$.
- Recall that

$$\mathbf{a}\times\mathbf{b} = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}}_{[\mathbf{a}]_\times} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = [\mathbf{a}]_\times \mathbf{b}$$

Recovering Best Projective Transform

- Let $\mathbf{h}^i$ denote the ith row of H. By default it is a column vector of size 3 x 1.
- $\mathbf{h}^{iT}$ denotes the ith row written as a 1 x 3 row vector.
- Let $\mathbf{x}_i' = [x_i' ; y_i' ; w_i']$
- Then $\mathbf{H}\mathbf{x}_i = \begin{bmatrix} \mathbf{h}^{1T}\mathbf{x}_i \\ \mathbf{h}^{2T}\mathbf{x}_i \\ \mathbf{h}^{3T}\mathbf{x}_i \end{bmatrix}$ and $\mathbf{x}_i' \times \mathbf{H}\mathbf{x}_i = \begin{bmatrix} y_i'\mathbf{h}^{3T}\mathbf{x}_i - w_i'\mathbf{h}^{2T}\mathbf{x}_i \\ w_i'\mathbf{h}^{1T}\mathbf{x}_i - x_i'\mathbf{h}^{3T}\mathbf{x}_i \\ x_i'\mathbf{h}^{2T}\mathbf{x}_i - y_i'\mathbf{h}^{1T}\mathbf{x}_i \end{bmatrix}$.

Recovering Best Projective Transform

$$\mathbf{x}_i' \times \mathbf{H}\mathbf{x}_i = \begin{bmatrix} y_i'\mathbf{h}^{3T}\mathbf{x}_i - w_i'\mathbf{h}^{2T}\mathbf{x}_i \\ w_i'\mathbf{h}^{1T}\mathbf{x}_i - x_i'\mathbf{h}^{3T}\mathbf{x}_i \\ x_i'\mathbf{h}^{2T}\mathbf{x}_i - y_i'\mathbf{h}^{1T}\mathbf{x}_i \end{bmatrix}_{3\times 1} = \mathbf{0}$$

$$\Rightarrow \begin{bmatrix} y_i'\mathbf{x}_i^T\mathbf{h}^3 - w_i'\mathbf{x}_i^T\mathbf{h}^2 \\ w_i'\mathbf{x}_i^T\mathbf{h}^1 - x_i'\mathbf{x}_i^T\mathbf{h}^3 \\ x_i'\mathbf{x}_i^T\mathbf{h}^2 - y_i'\mathbf{x}_i^T\mathbf{h}^1 \end{bmatrix}_{3\times 1} = \mathbf{0} \quad \text{since } \mathbf{a}^T\mathbf{b} = \mathbf{b}^T\mathbf{a}$$

$$\Rightarrow \begin{bmatrix} \mathbf{0}^T & -w_i'\mathbf{x}_i^T & y_i'\mathbf{x}_i^T \\ w_i'\mathbf{x}_i^T & \mathbf{0}^T & -x_i'\mathbf{x}_i^T \\ -y_i'\mathbf{x}_i^T & x_i'\mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix}_{3\times 9} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix}_{9\times 1} = \mathbf{0}$$

$$\Rightarrow A_i\mathbf{h} = \mathbf{0}$$

Recovering Best Projective Transform

- Correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$ yields 3 equations $A_i\mathbf{h}=\mathbf{0}$.
  - However, it can be shown that matrix A_i has 2 linearly independent rows (since $x_i' A_i^1 + y_i' A_i^2 + w_i' A_i^3 = 0$)
  - So one row can be discarded.
  - Through an abuse of notation, let us denote the resulting 2 x 9 matrix as A_i also.
- So one correspondence yields 2 equations.
- Since 8 unknowns will require 8 equations, we will need $N \geq 4$ corresponding points.
  - The points must be non-collinear.

Recovering Best Projective Transform

- This will yield the system A**h**=**0** where size of A is 2N x 9.
  - It can be shown that rank(A)=8 and dim(A)=9.
  - So nullity of A is 1 and therefore **h** can be found as the null space of A.
  - However, when measurements contain noise or N>4, then A**h**≠**0** and it is better to find **h** by minimising ||A**h**||.
  - This can be done via singular value decomposition
    - [U,D,V]=svd(A)
    - **h** is the last column of matrix V.

Recovering Best Projective Transform

Concise algorithm

Input: N point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$

1. Fill in the 2N x 9 matrix A using the $\mathbf{x}_i$ and $\mathbf{x}_i'$
2. [U,D,V]=svd(A)
3. Optimal projective transformation parameters $\mathbf{h}^*$ lie in last column of matrix V.

This algorithm is known as the **Direct Linear Transform (DLT).**
For some practical tips, please refer to slides 14—17 from
http://www.ele.puc-rio.br/~visao/Topicos/Homographies.pdf

Projective Warping

- Same as affine warping.
- Just remember to move back from $\mathbb{P}^2$ to $\mathbb{R}^2$.