Medical Images Registration

Lecture 26-Nov

Waqas Sultani
MedAI Research Group
Information Technology University

Outline

- Motivation
  - Image registration is an alignment problem
- Registration basics
- Rigid registration
- Non-rigid registration
- Example Applications

Image Registration Taxonomy

- **Dimensionality**
  - 2D-2D, 3D-3D, 2D-3D
- **Nature of registration basis**
  - Image based
    - Extrinsic, Intrinsic
  - Non-image based
- **Nature of the transformation**
  - Rigid, Affine, Projective, Curved
- **Interaction**
  - Interactive, Semi-automatic, Automatic
- **Modalities involved**
  - Mono-modal, Multi-modal, Modality to model

- **Subject:**

  Intra-subject

  Inter-subject

  Atlas
- **Domain of transformation**
  - Local, global
- **Optimization procedure**
  - Gradient Descent, SGD, …
- **Object**
  - Whole body, organ, …

Open Source Implementation

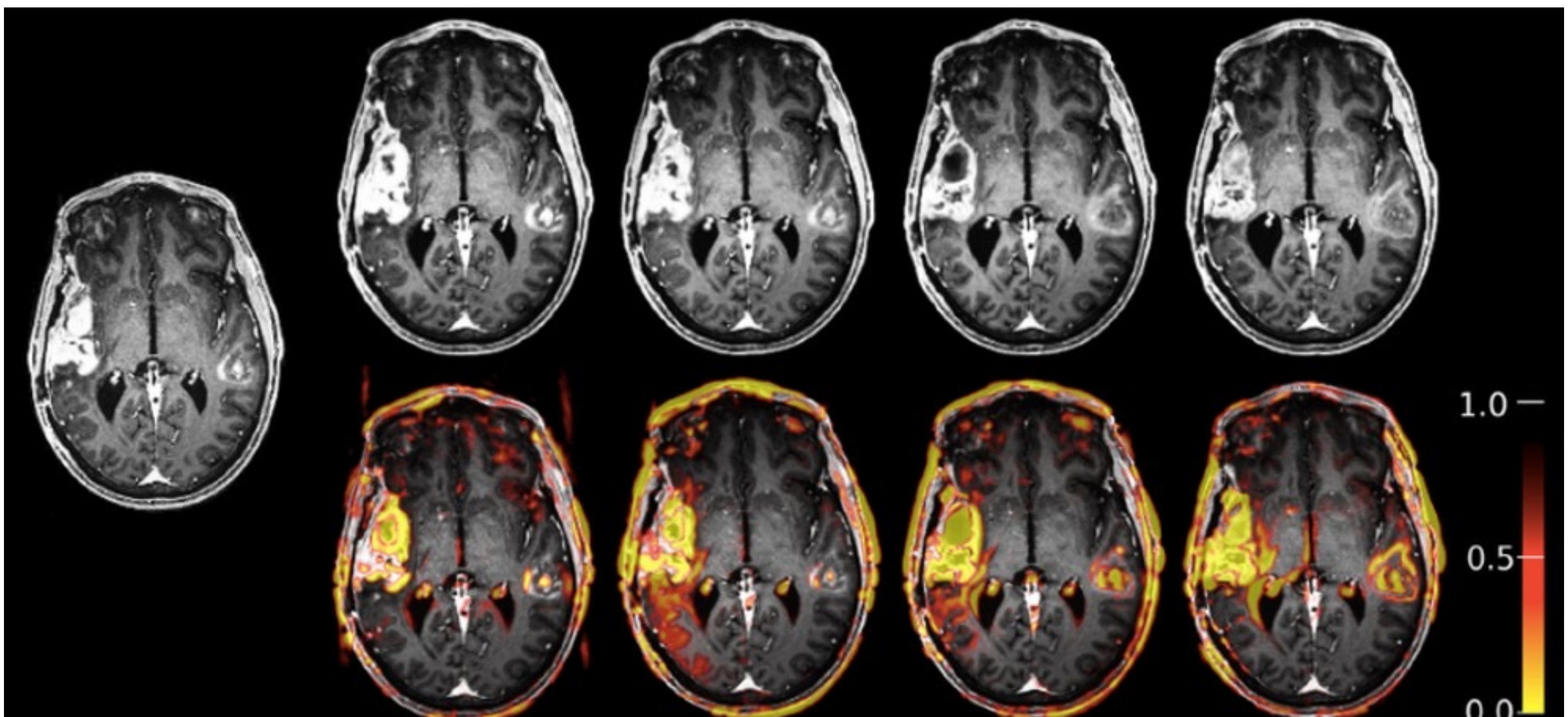
- ITK
- ANTS (advanced normalization tools) (PICSL of Upenn)
- CAVASS (MIPG of Upenn)
- Nifty Reg (UCL)
- Elastix (www.elastix.isi.uu.nl)
- FAIR (Modersitzki 2009), mostly matlab.
- 3D Slicer
- FSL
- …

Modalities in Medical Imaging

- Mono-modality:

Modalities in Medical Imaging

- Mono-modality:
  - ✓ A series of same modality images (CT/CT, MR/MR, Mammogram pairs,…).



Modalities in Medical Imaging

- Mono-modality:
  - ✓ A series of same modality images (CT/CT, MR/MR, Mammogram pairs,…).
  - ✓ Images may be acquired weeks or months apart; taken from different viewpoints.

Modalities in Medical Imaging

- Mono-modality:
  - ✓ A series of same modality images (CT/CT, MR/MR, Mammogram pairs,…).
  - ✓ Images may be acquired weeks or months apart; taken from different viewpoints.
  - ✓ Aligning images in order to detect subtle changes in intensity or shape

Modalities in Medical Imaging

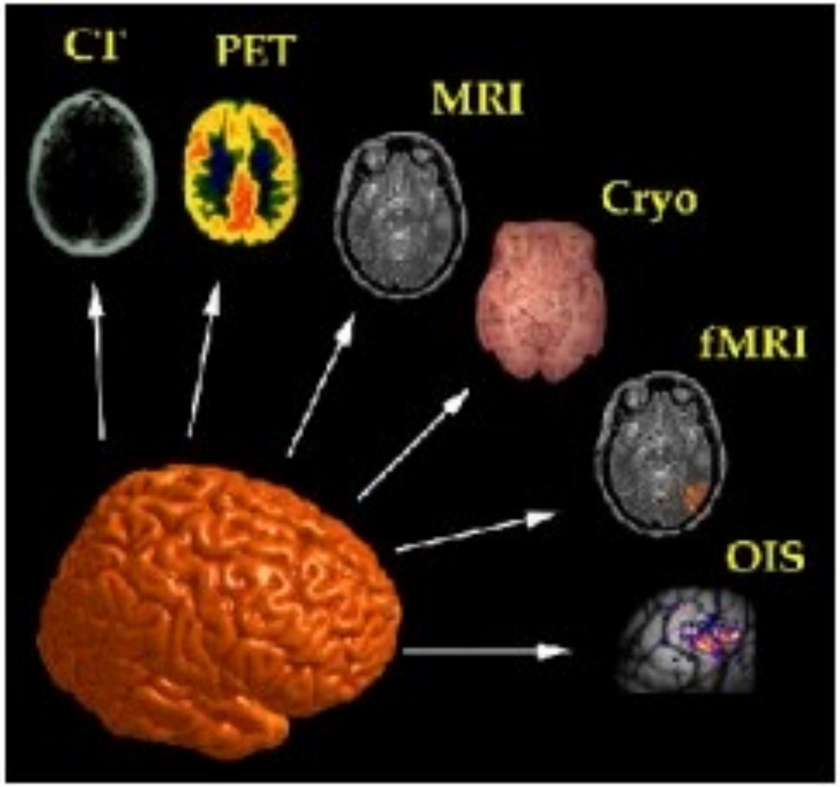
- Mono-modality:
  - ✓ A series of same modality images (CT/CT, MR/MR, Mammogram pairs,…).
  - ✓ Images may be acquired weeks or months apart; taken from different viewpoints.
  - ✓ Aligning images in order to detect subtle changes in intensity or shape
- Multi-modality:

Modalities in Medical Imaging

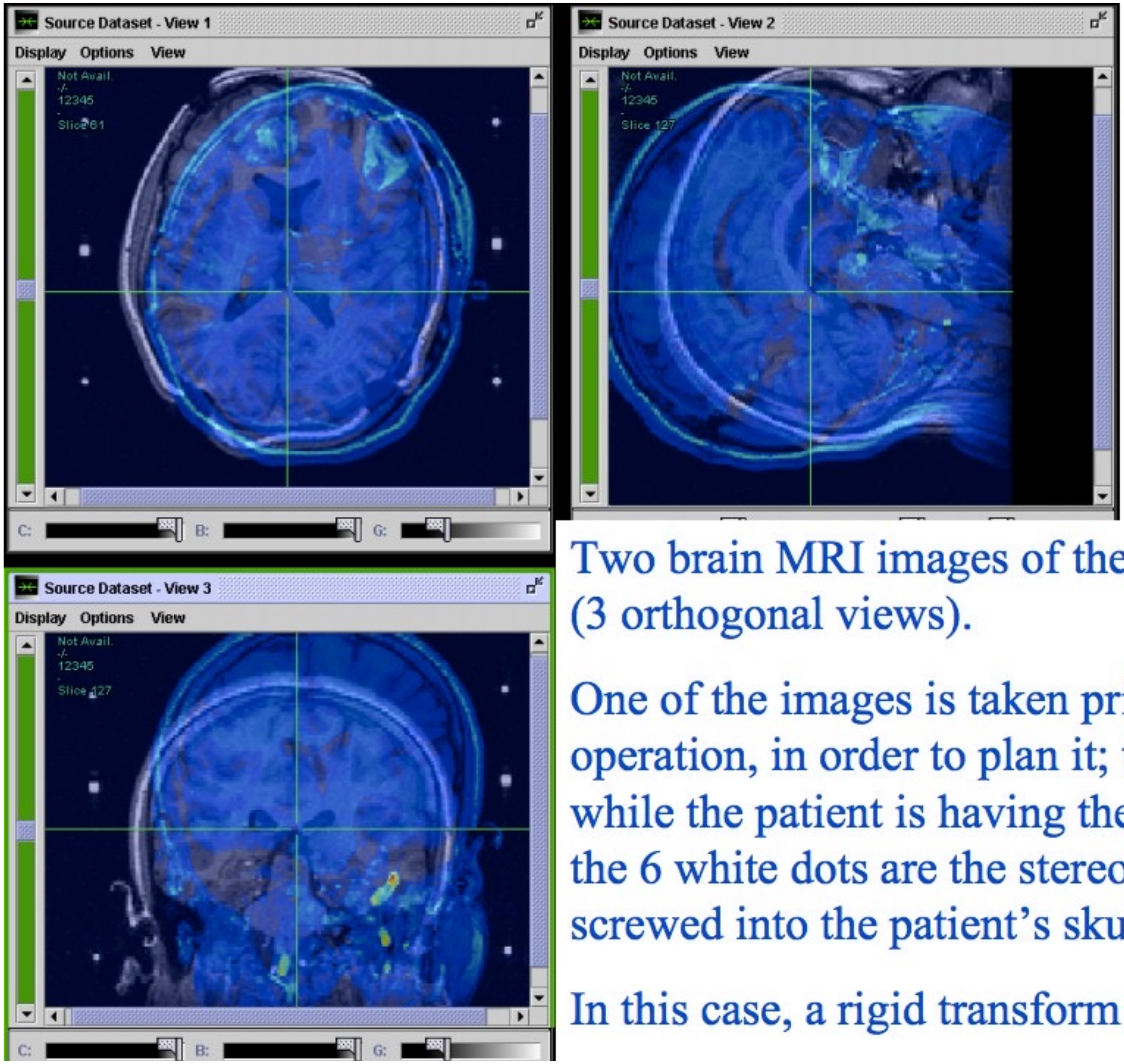
- Mono-modality:
  - ✓ A series of same modality images (CT/CT, MR/MR, Mammogram pairs,…).
  - ✓ Images may be acquired weeks or months apart; taken from different viewpoints.
  - ✓ Aligning images in order to detect subtle changes in intensity or shape
- Multi-modality:
  - ✓ Complementary anatomic and functional information from multiple modalities can be obtained for the precise diagnosis and treatment.

In other words,...

- Combining modalities (inter modality) gives extra information.
- Repeated imaging over time same modality, e.g. MRI, (intra modality) equally important.
- Have to spatially register the images.



Before Registration

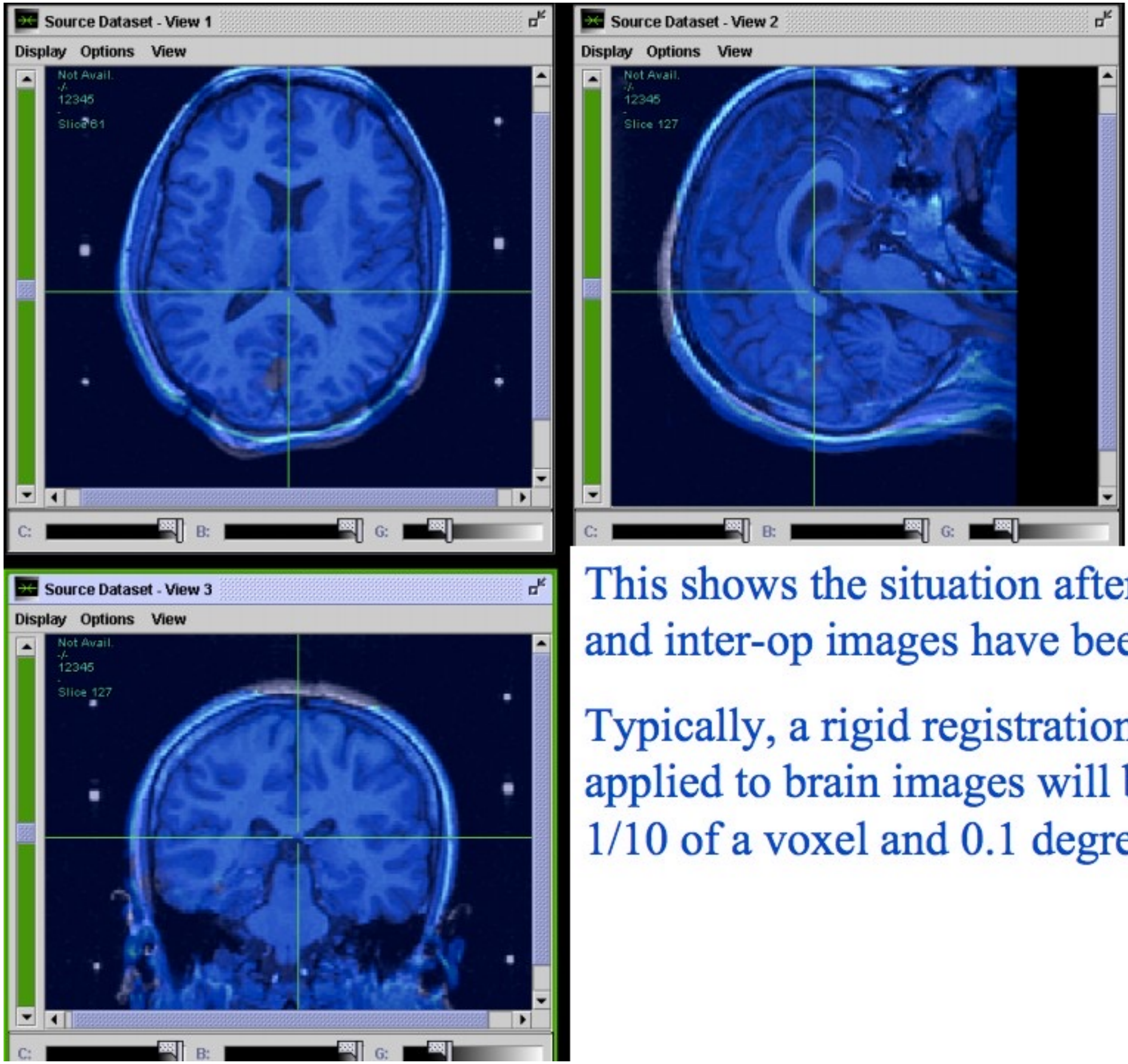


Two brain MRI images of the same patient (3 orthogonal views).

One of the images is taken prior to the operation, in order to plan it; the second while the patient is having the operation: the 6 white dots are the stereotactic frame screwed into the patient's skull.

In this case, a rigid transform suffices

After Registration

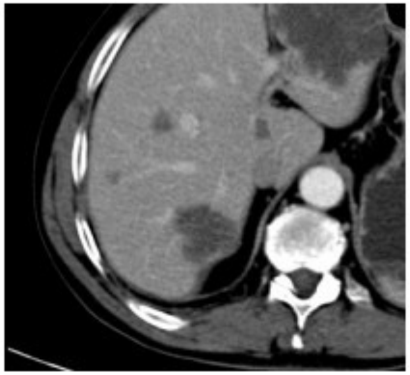


This shows the situation after the pre-op and inter-op images have been aligned.

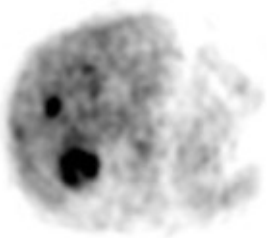
Typically, a rigid registration algorithm applied to brain images will be accurate to 1/10 of a voxel and 0.1 degrees of rotation

Fusion of information = registration plus combination in a single representation: PET/CT

CT



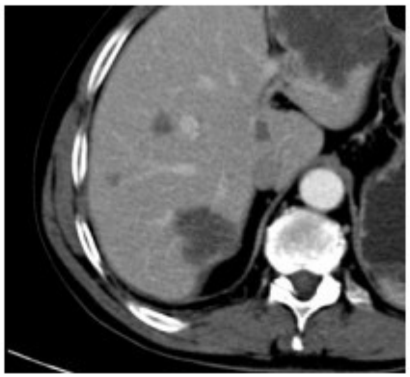
PET



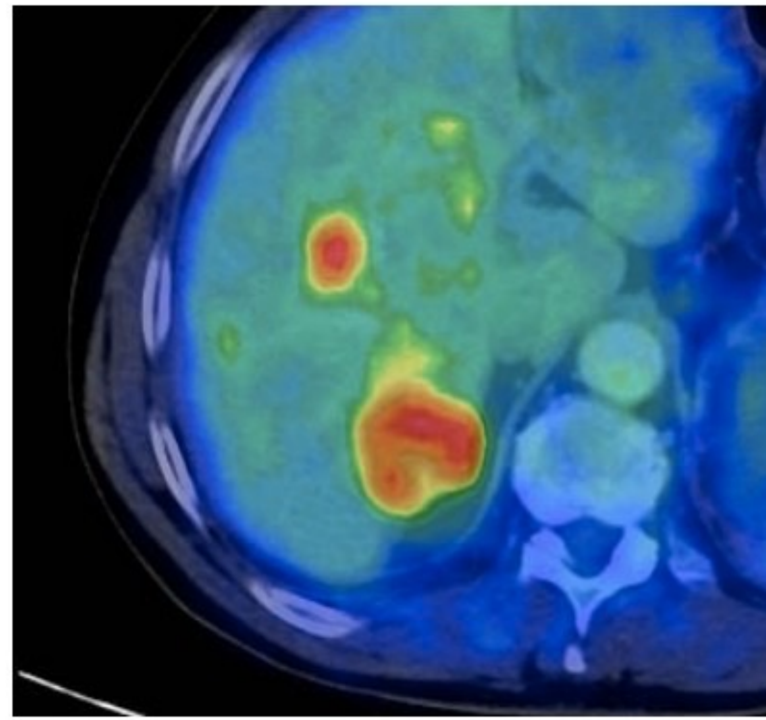
Deformable fusion- PET shows increased metabolism in lesions identified on CT, consistent with active tumour growth rather than necrosis post-radiotherapy

Fusion of information = registration plus combination in a single representation: PET/CT

CT



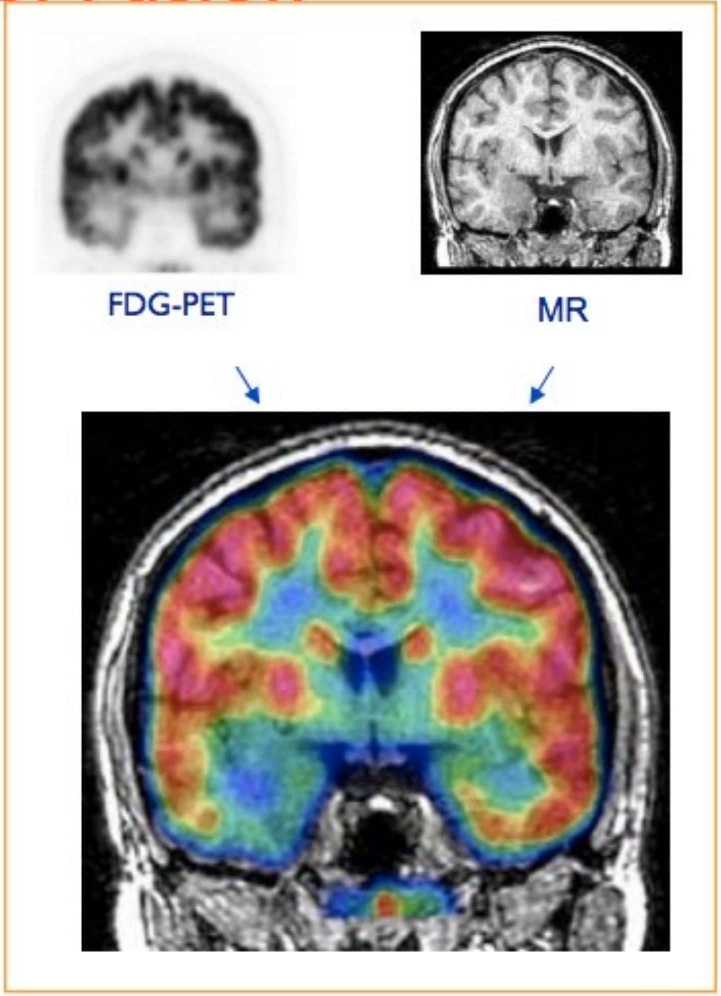
PET



Deformable fusion- PET shows increased metabolism in lesions identified on CT, consistent with active tumour growth rather than necrosis post-radiotherapy

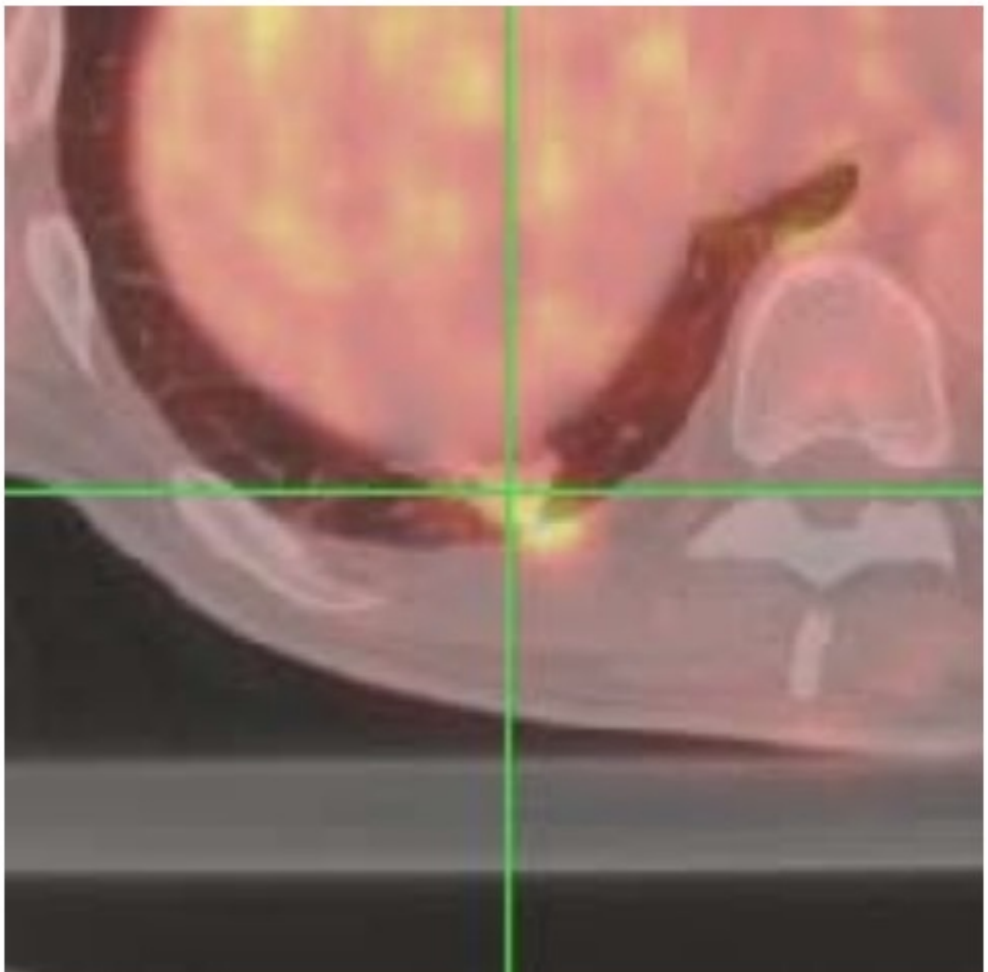
Many Clinical Applications of Fusion

- Cancer staging
- Biopsy planning
- Radiotherapy treatment planning
- Quantitative assessment of treatment response
- Pre-surgical assessment of other conditions e.g. epilepsy
- As an *effective* communication tool when reporting to clinical meetings, referring physicians or to patients
- Whenever multiple data sources may be better assessed together



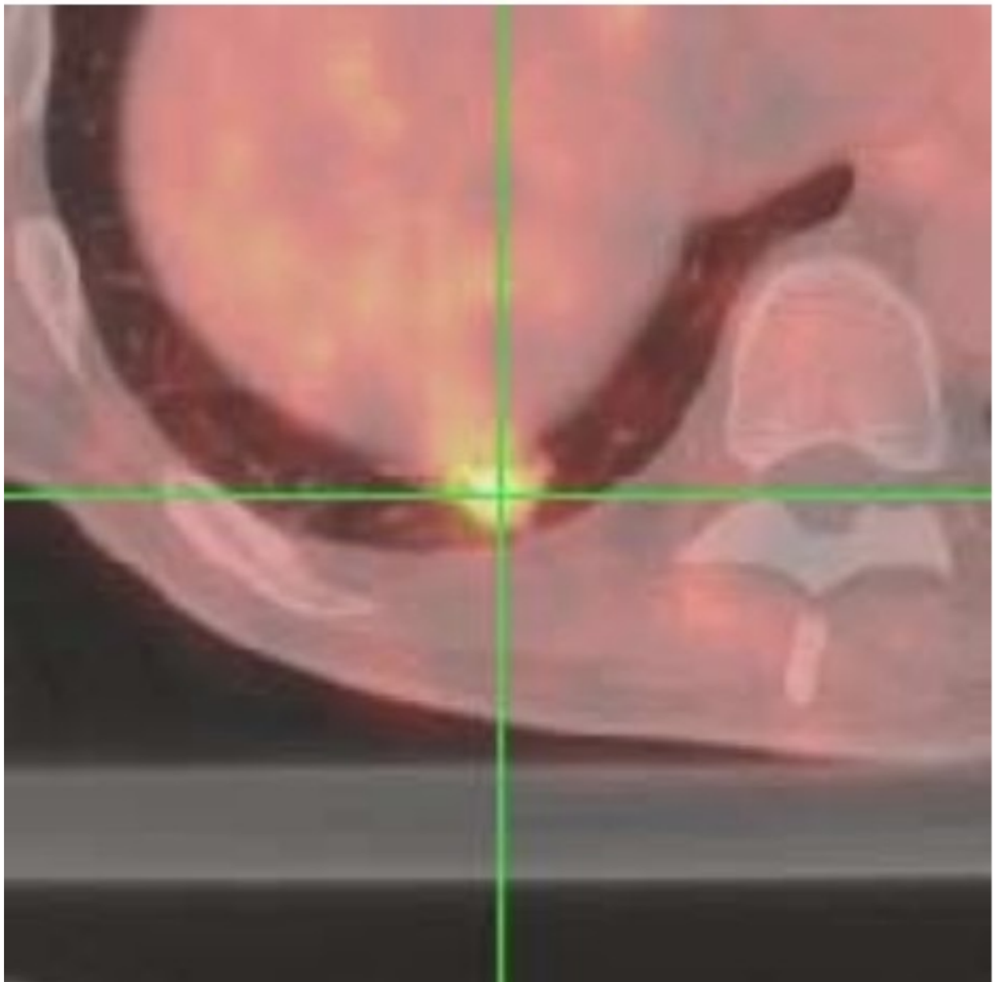
PET data identifies a region of hypometabolism due to epilepsy. Fusion with MR localises the damage to the anterior and medial areas of the right temporal gyrus

Rigid registration poor



Is the tumour in the lungs or the stomach?

Non-rigid registration



Looks plausible; but how could you be sure?

Are you prepared to risk your software against getting sued?

Summary of Mostly Used Applications

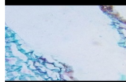
- **Diagnosis**
  - Combining information from multiple imaging modalities
- **Studying disease progression**
  - Monitoring changes in size, shape, position or image intensity over time
- **Image guided surgery or radiotherapy**
  - Relating pre-operative images and surgical plans to the physical reality of the patient
- **Patient comparison or atlas construction**
  - Relating one individual’s anatomy to a standardized atlas
  - This atlas provides **a standardized set of coordinates to determine specific sites within the region of interest**.


INFORMATION
TECHNOLOGY
UNIVERSITY
MedAI
INTELLIGENT
MACHINES
LAB
Stitched Image
Current Image

INTELLIGENT MACHINES LAB

ITU INFORMATION TECHNOLOGY UNIVERSITY

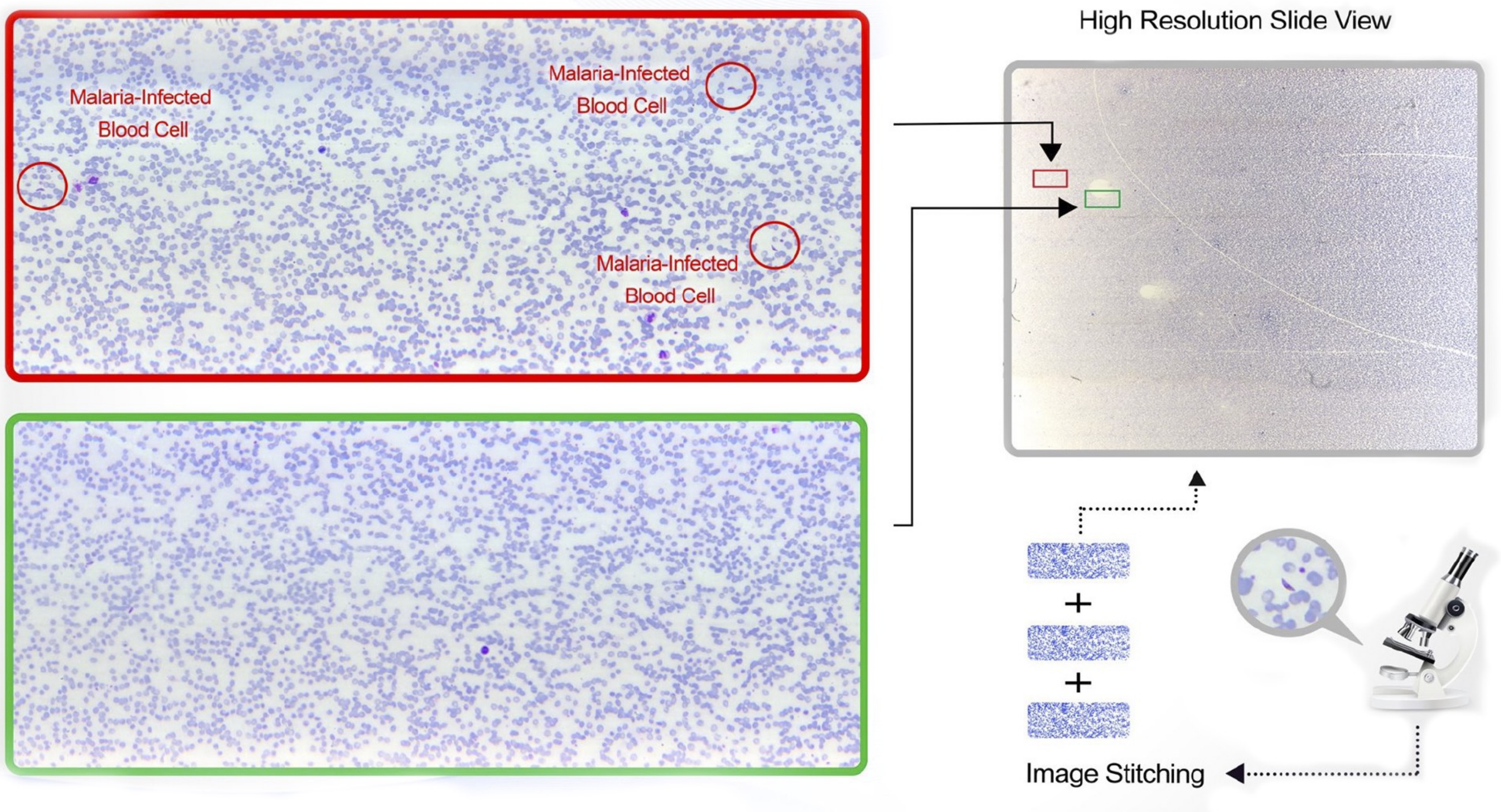
MedAI

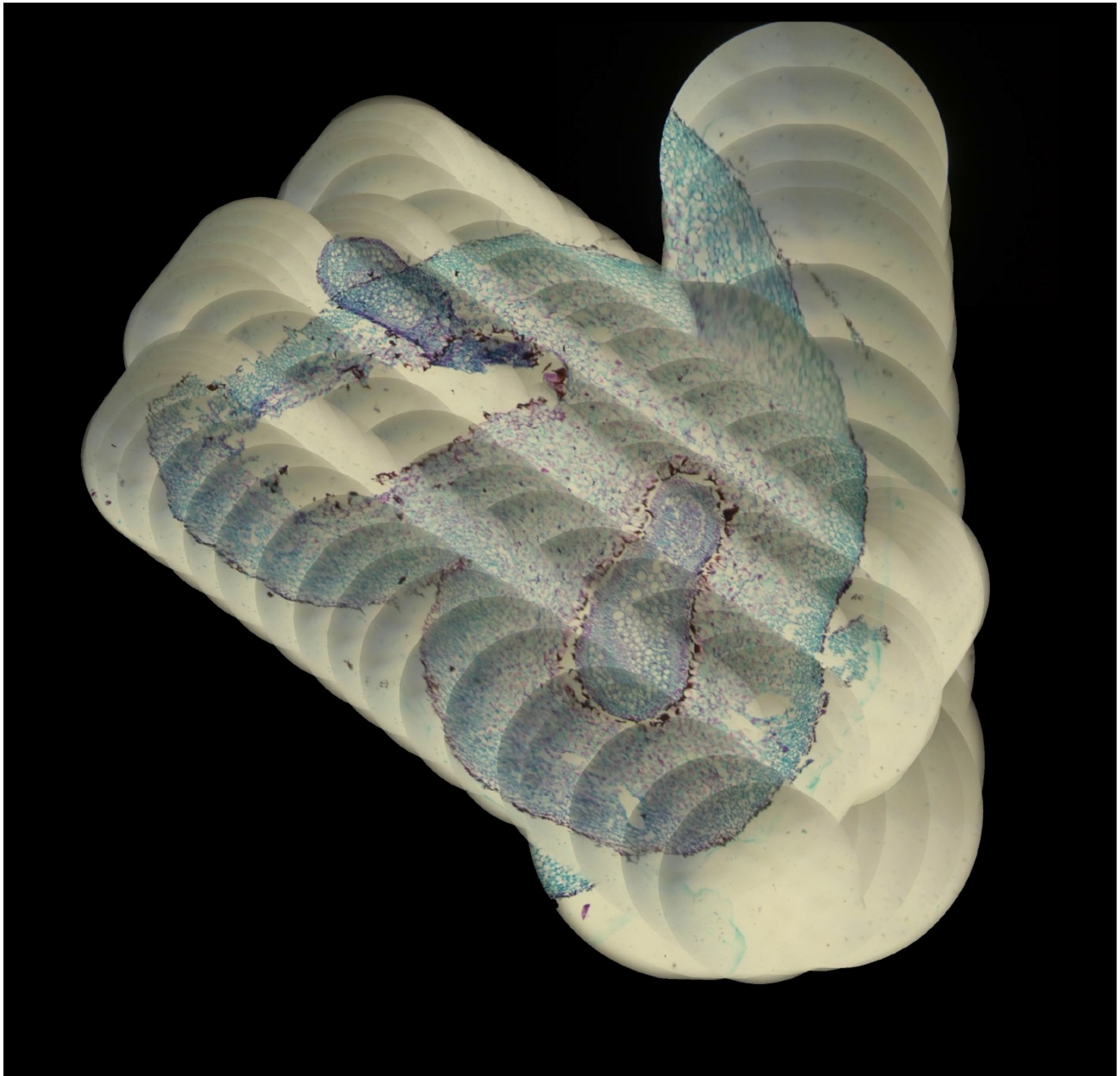


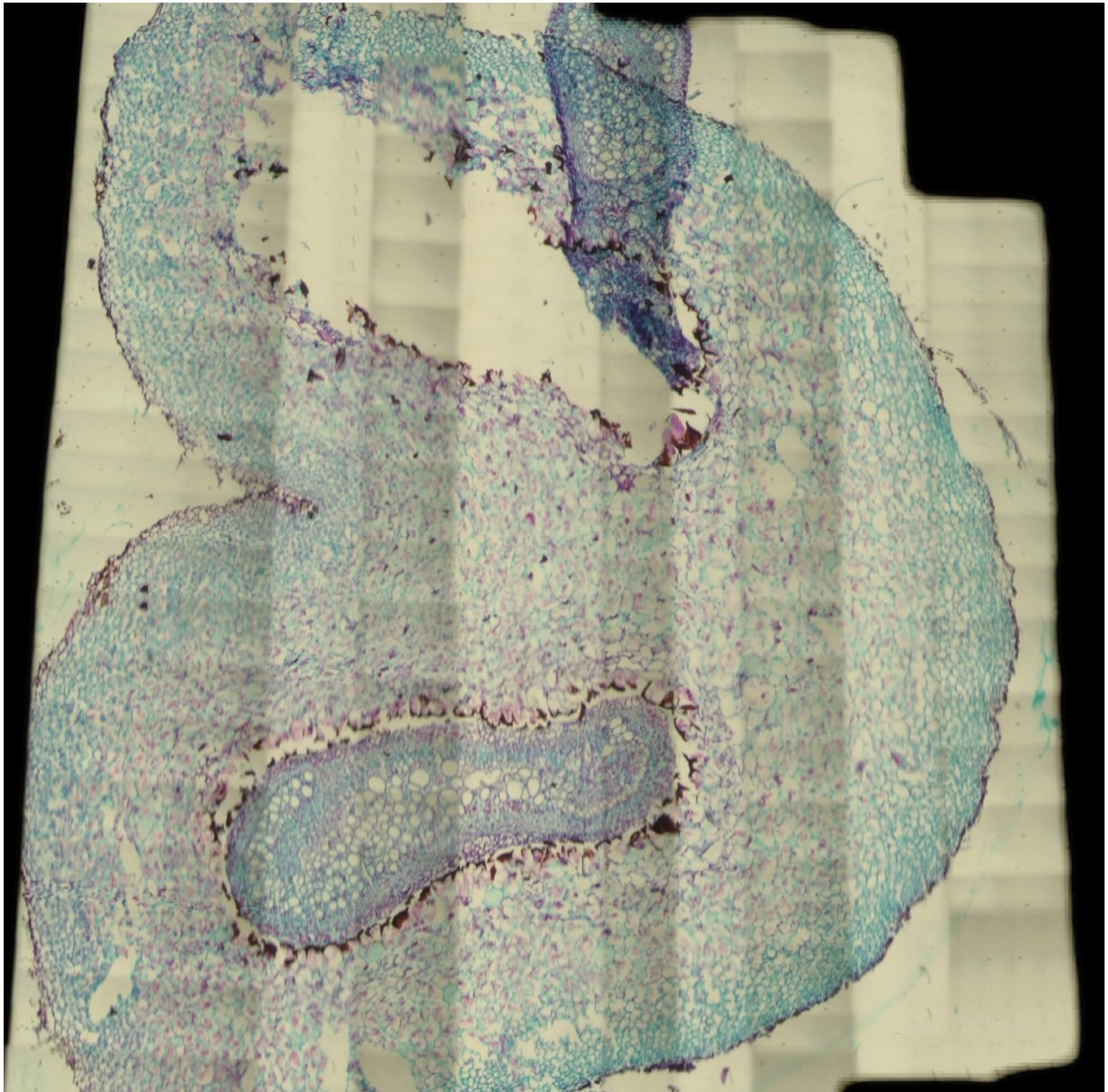
Current Image

Compound Microscope

Stitched Image

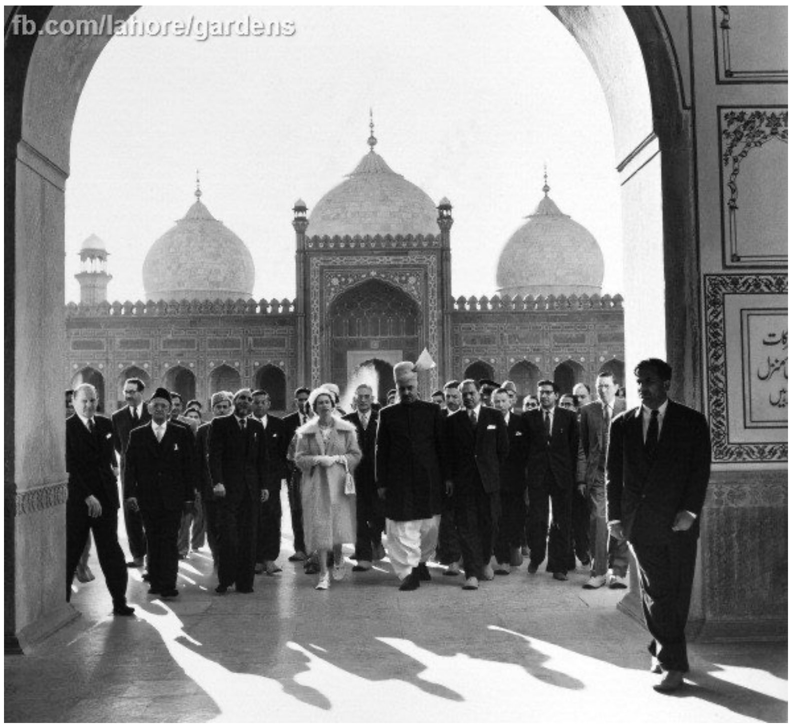

Malaria-Infected
Blood Cell
Malaria-Infected
Blood Cell
Malaria-Infected
Blood Cell
High Resolution Slide View
+
+
Image Stitching







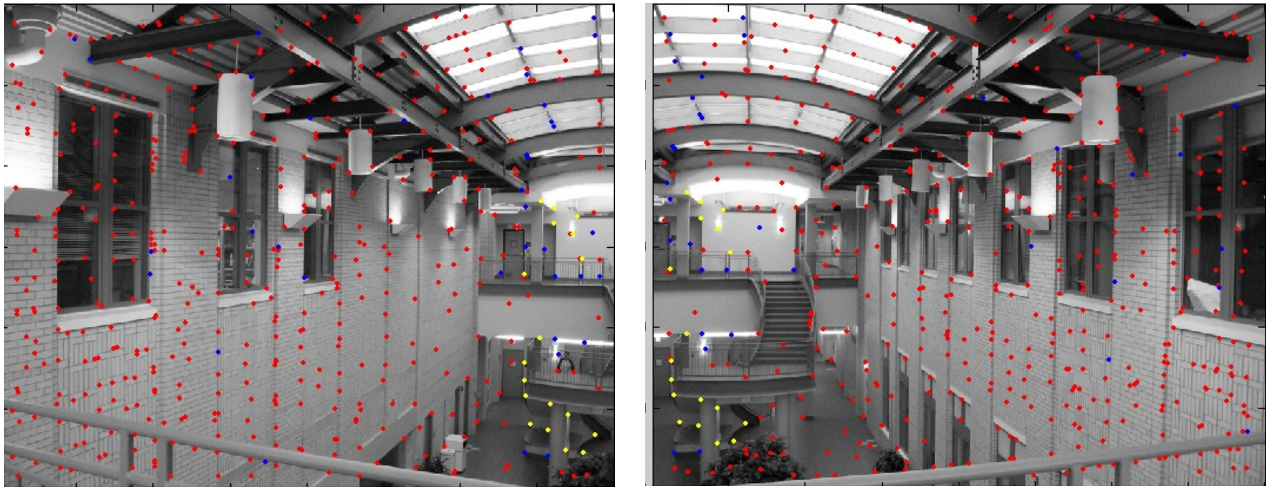
Old and New



Src: Dr. Mohsen Ali

Src: Dr. Mohsen Ali

Example: Stitching



Slide Credit: graphics.cs.cmu.edu/courses/15-463/2010_spring/.../feature-alignment.**ppt**

Applications of 2D Image Registration

Panoramas
Multiple Images Stitched Together

Source: Dr. Sohaib / Aamer Zaheer

Panoramas
Multiple Images Stitched Together

Applications of 2D Image Registration

Image by Sergey Semenov (http://www.sergesemenov.com/) - Winner of Epson International Photographic Pano Award 2013
http://www.dailymail.co.uk/sciencetech/article-2260276/New-York-youve-seen-Incredible-interactive-panorama-lets-zoom.html

Video Stablization

Example of Data Annotations GUI

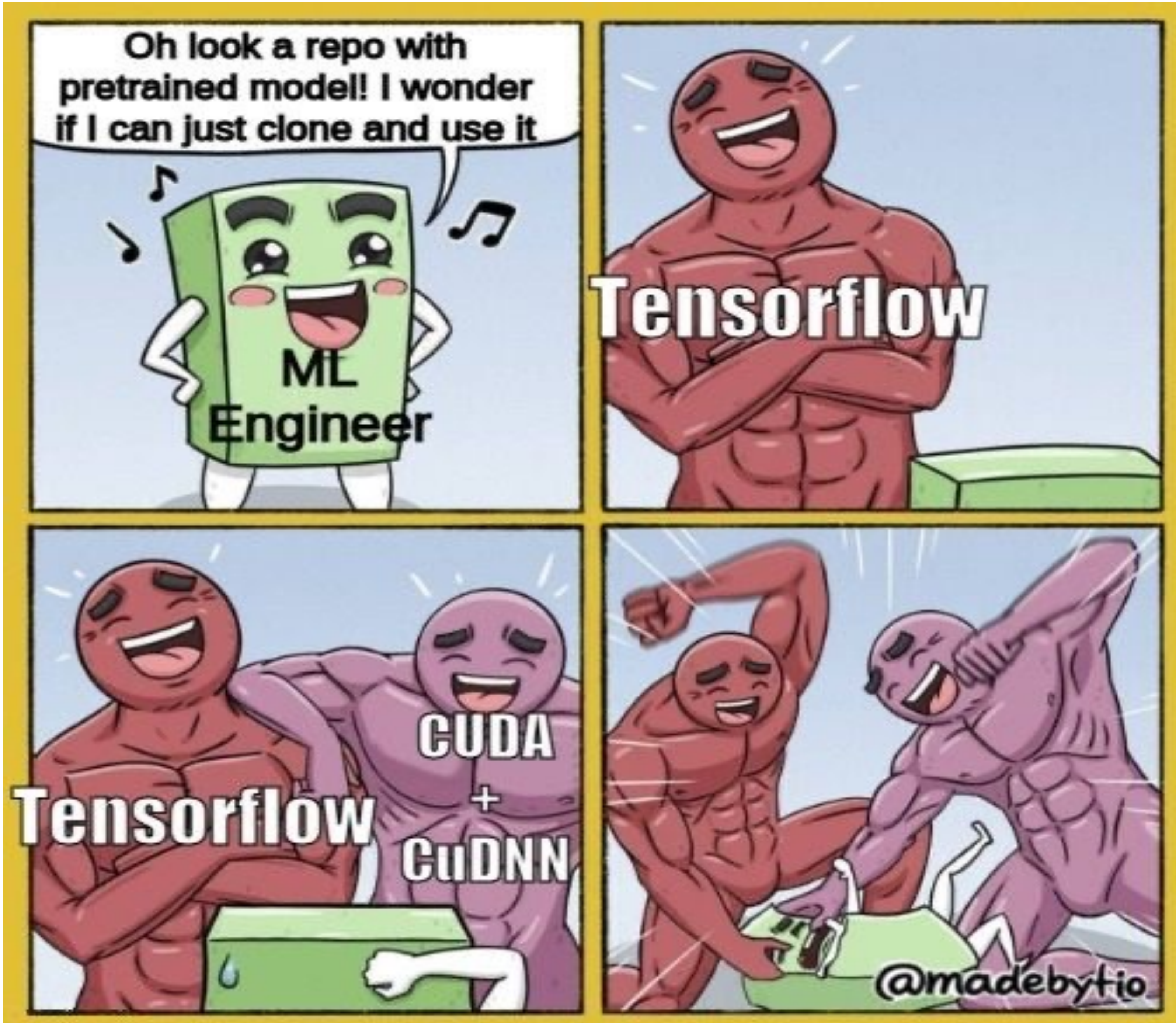
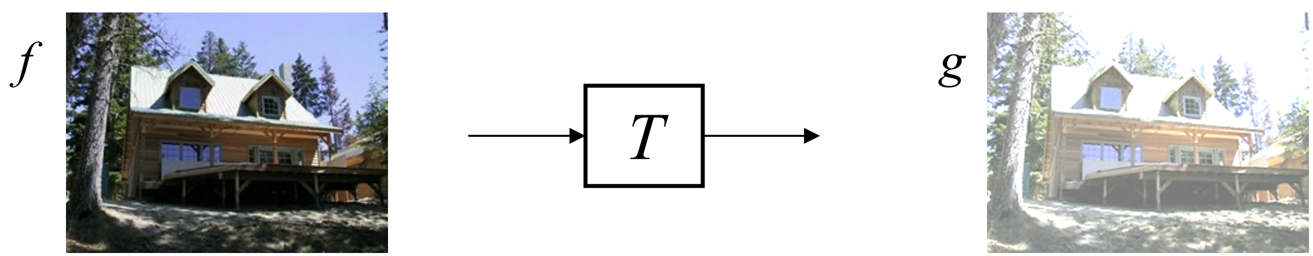

Oh look a repo with pretrained model! I wonder if I can just clone and use it
ML Engineer
Tensorflow
Tensorflow
CUDA + CuDNN
@madebytio

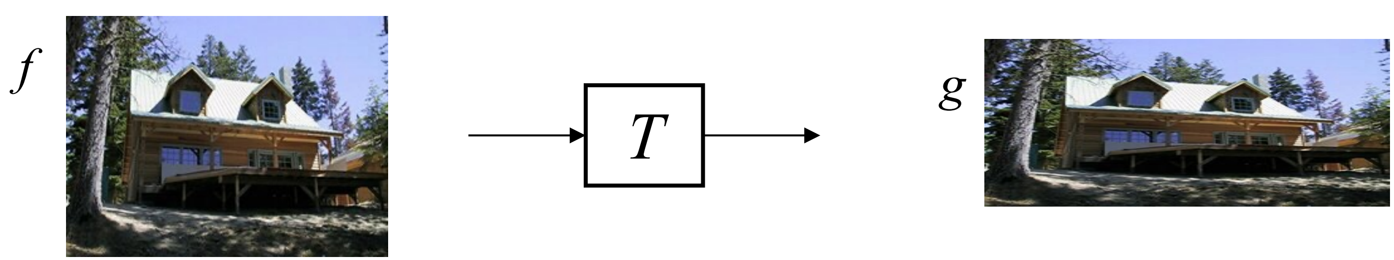
Image Transformations

Transformation: Transformation is a function.

A function that maps one set to another set after performing some operations.

Image Transformations





Source: Efros

Image Transformations

- image filtering: change ***range*** of image
    - $g(x) = T(f(x))$

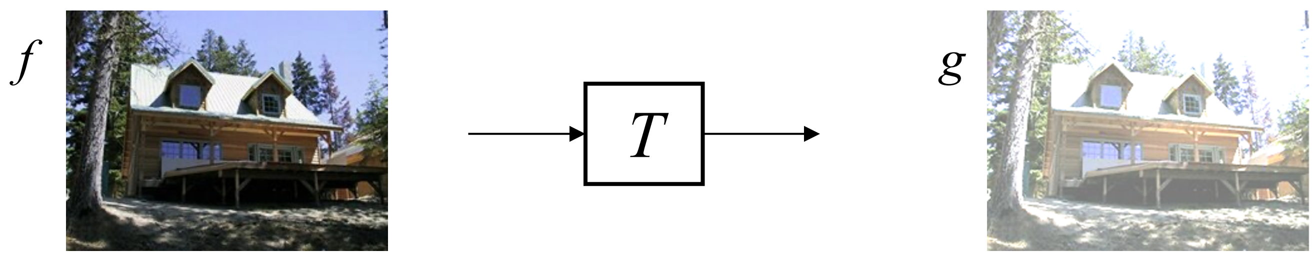
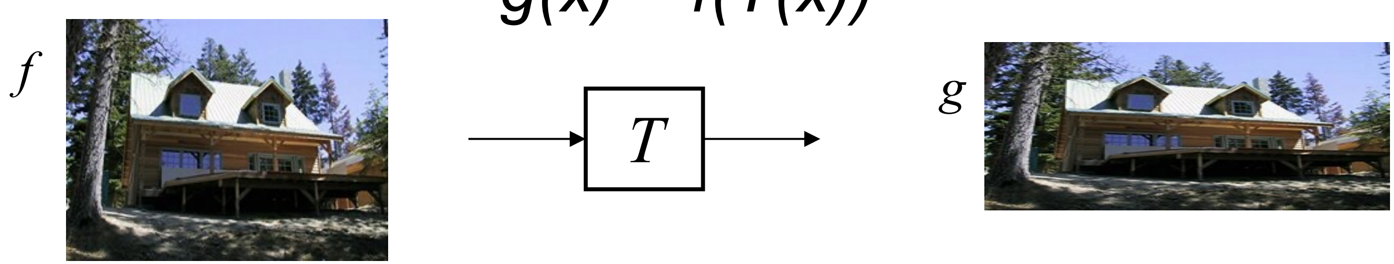


image warping: change ***domain*** of image

$$g(x) = f(T(x))$$



Source: Efros

Parametric (global) warping

translation

rotation

aspect

affine

perspective

cylindrical

Source: Efros

Parametric (global) warping

- Examples of parametric warps (**Image warping** is the process of digitally manipulating an image such that any shapes portrayed in the image have been significantly distorted):

translation

rotation

aspect

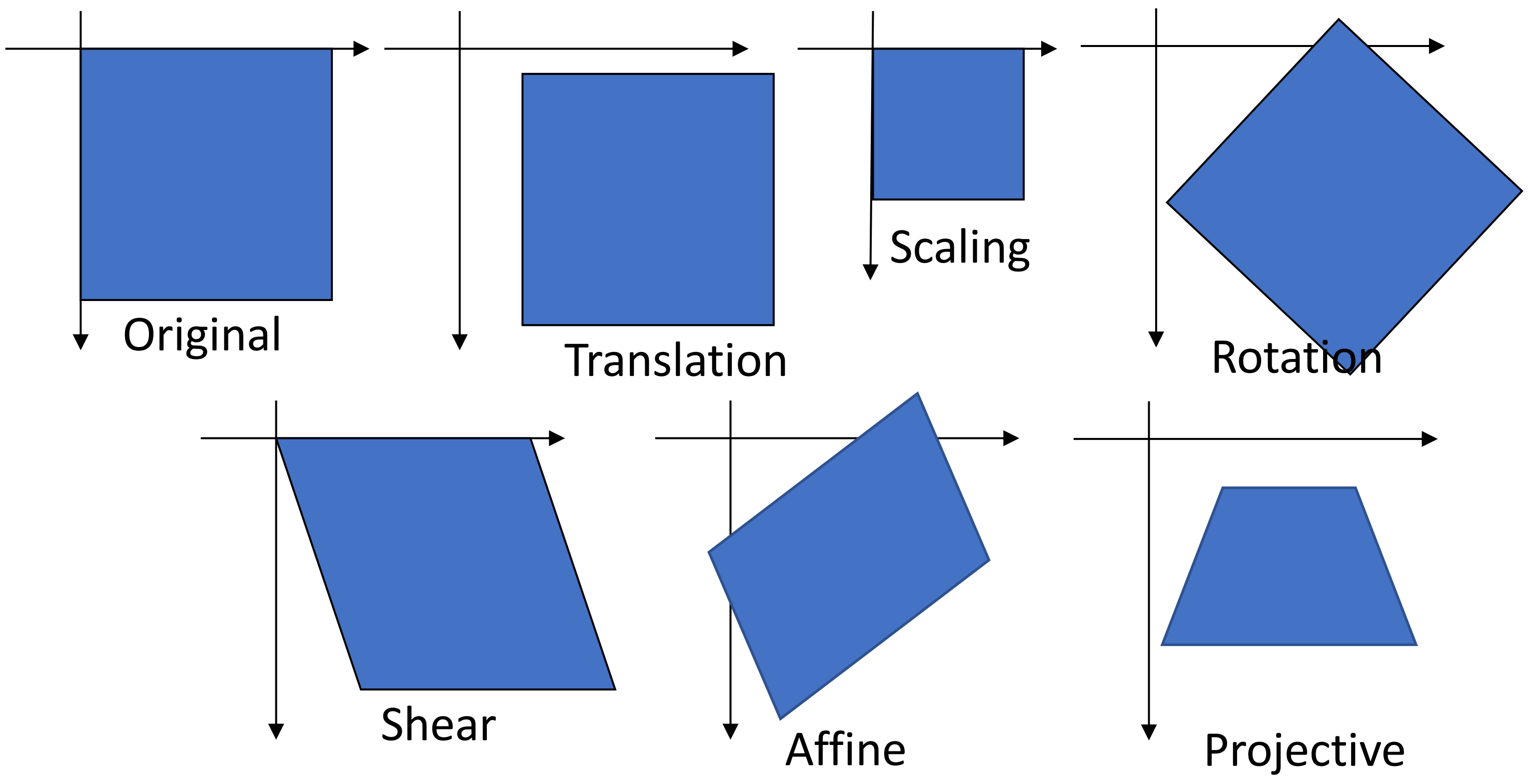
affine

perspective

cylindrical

Source: Efros

2-D Transformations



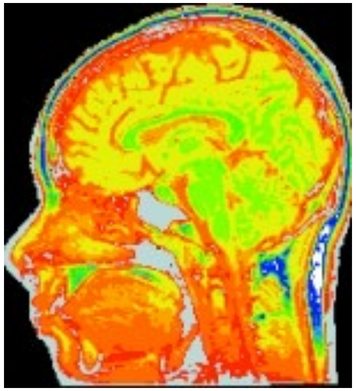
Then, What is Image Registration (formally)?

Image Registration is a

- **Spatial transform** that maps points from one image to corresponding points in another image

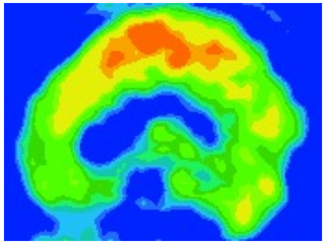
matching two images so that corresponding coordinate points in the two images <u>correspond to the same physical region of the scene</u> being imaged

Also referred to as image fusion, superimposition, matching or merge



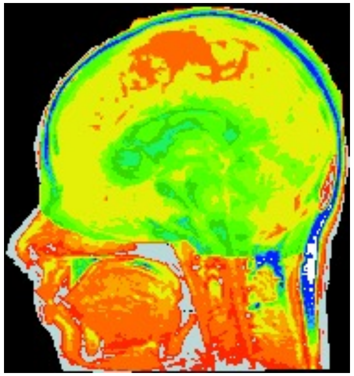
MR

+



SPECT

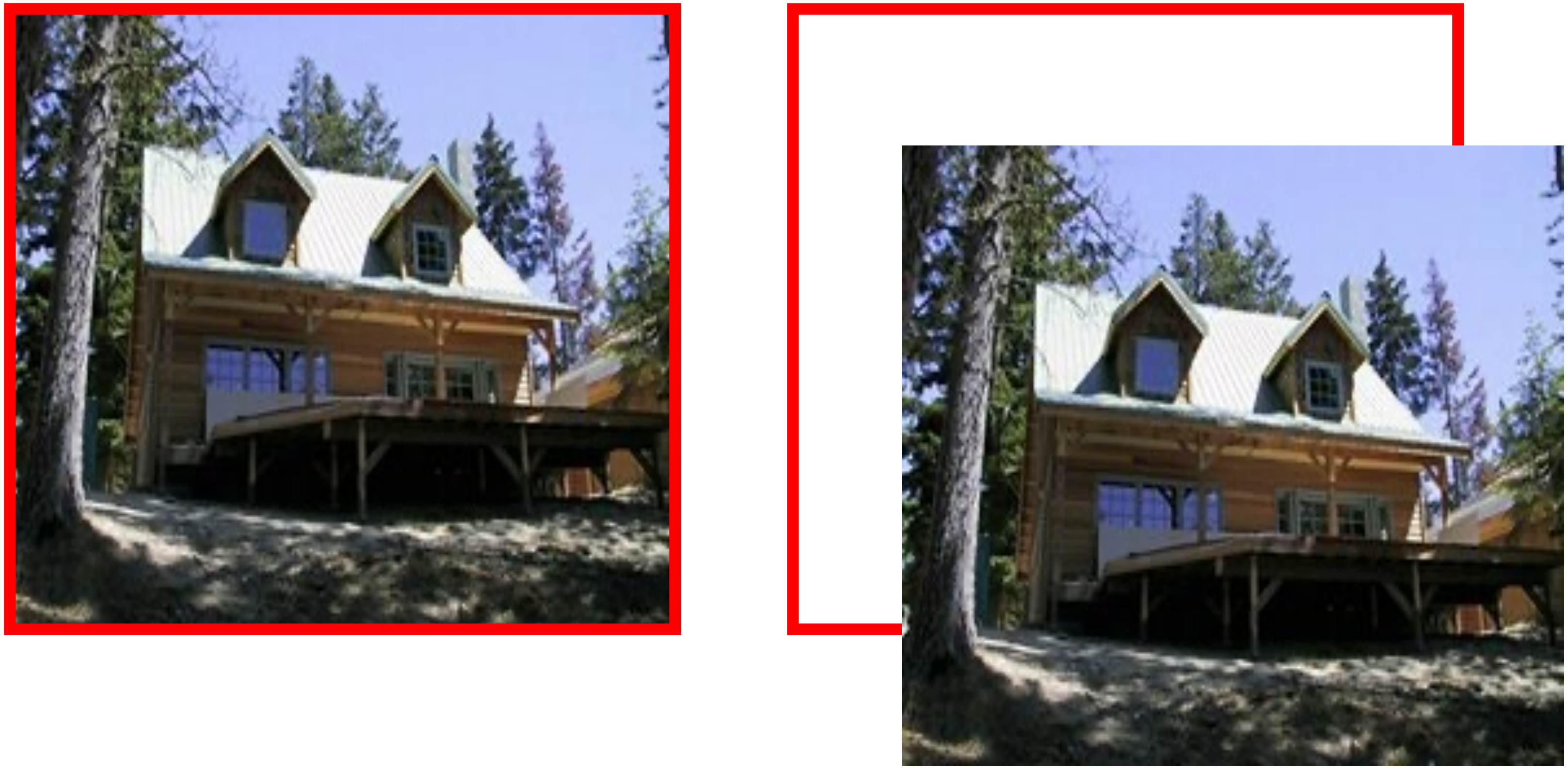
=



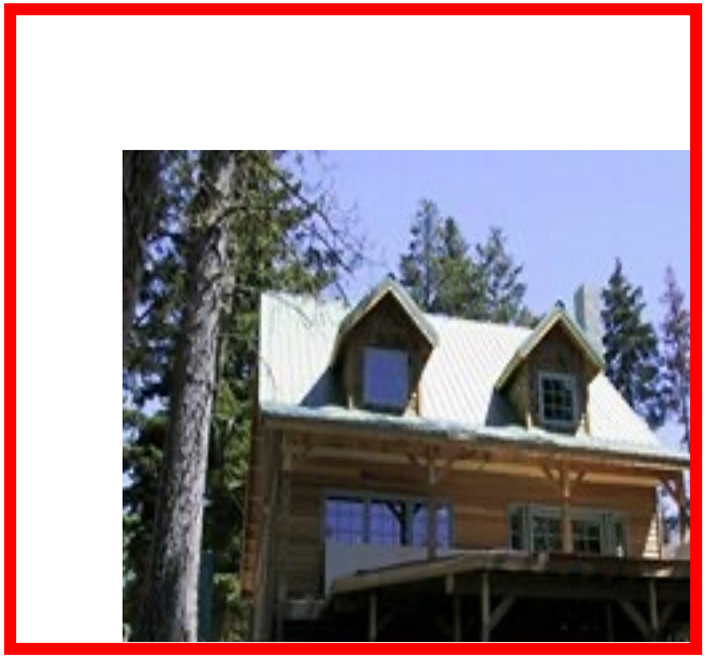
registered

2-D Transformations

2-D Transformations

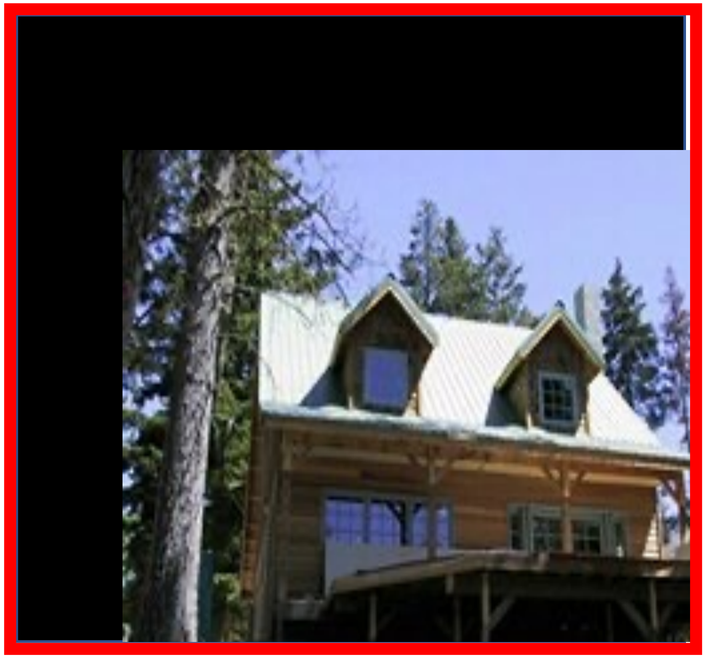


2-D Transformations



2-D Transformations

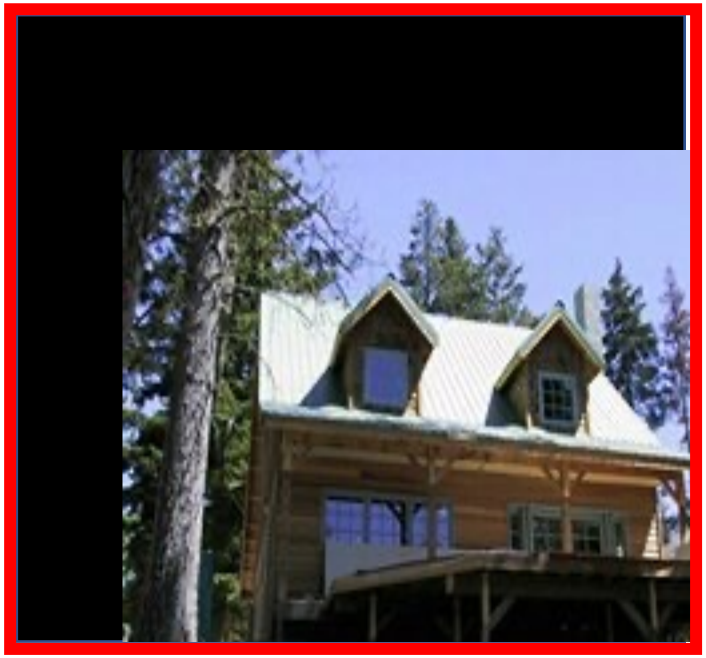
Original



Translation

2-D Transformations

Original

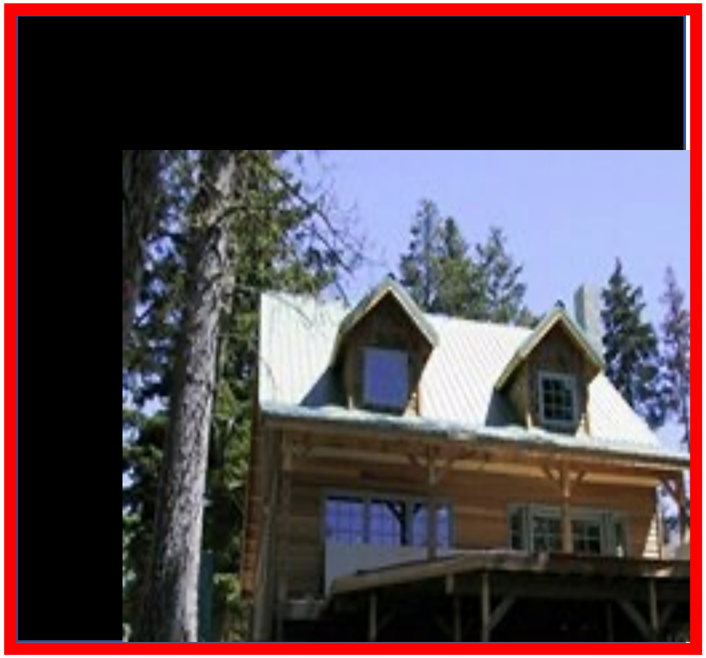


Translation

How much image translation has happened?

2-D Transformations

Original



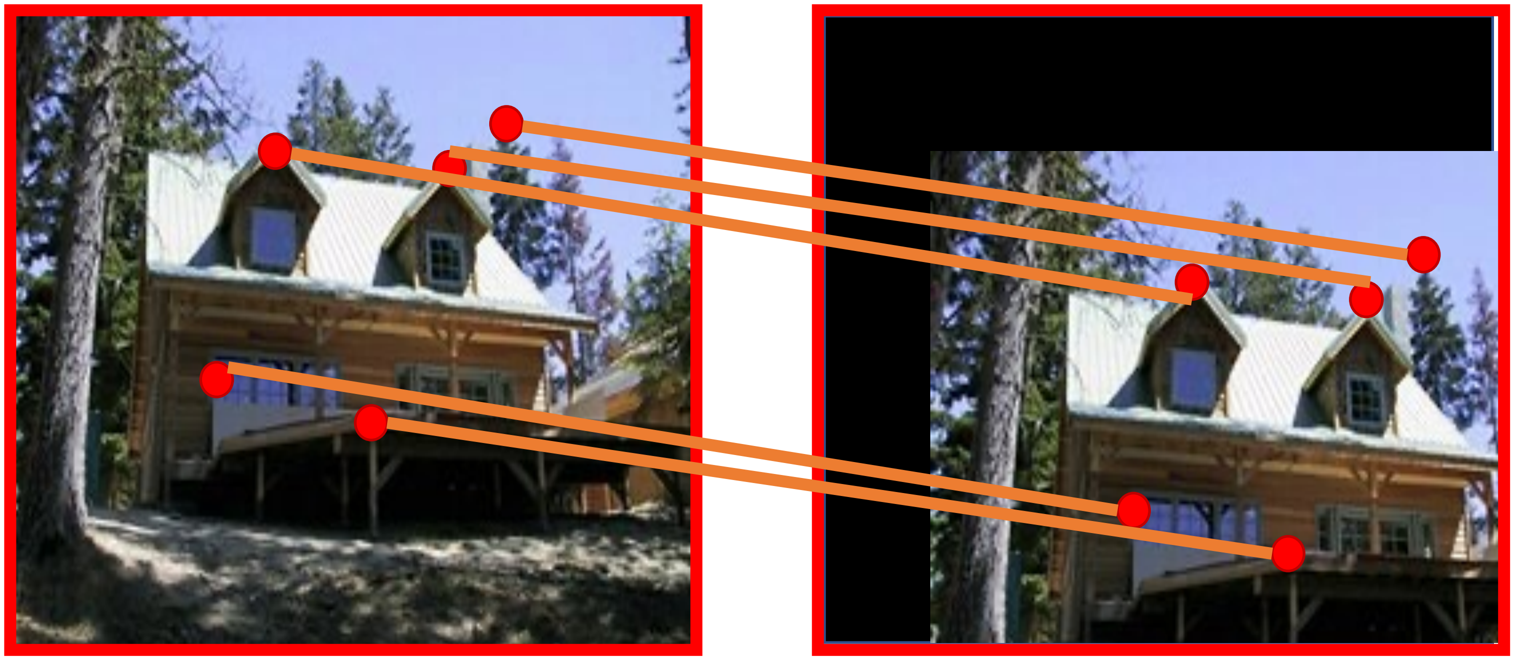
Translation

2-D Transformations

Original

Translation

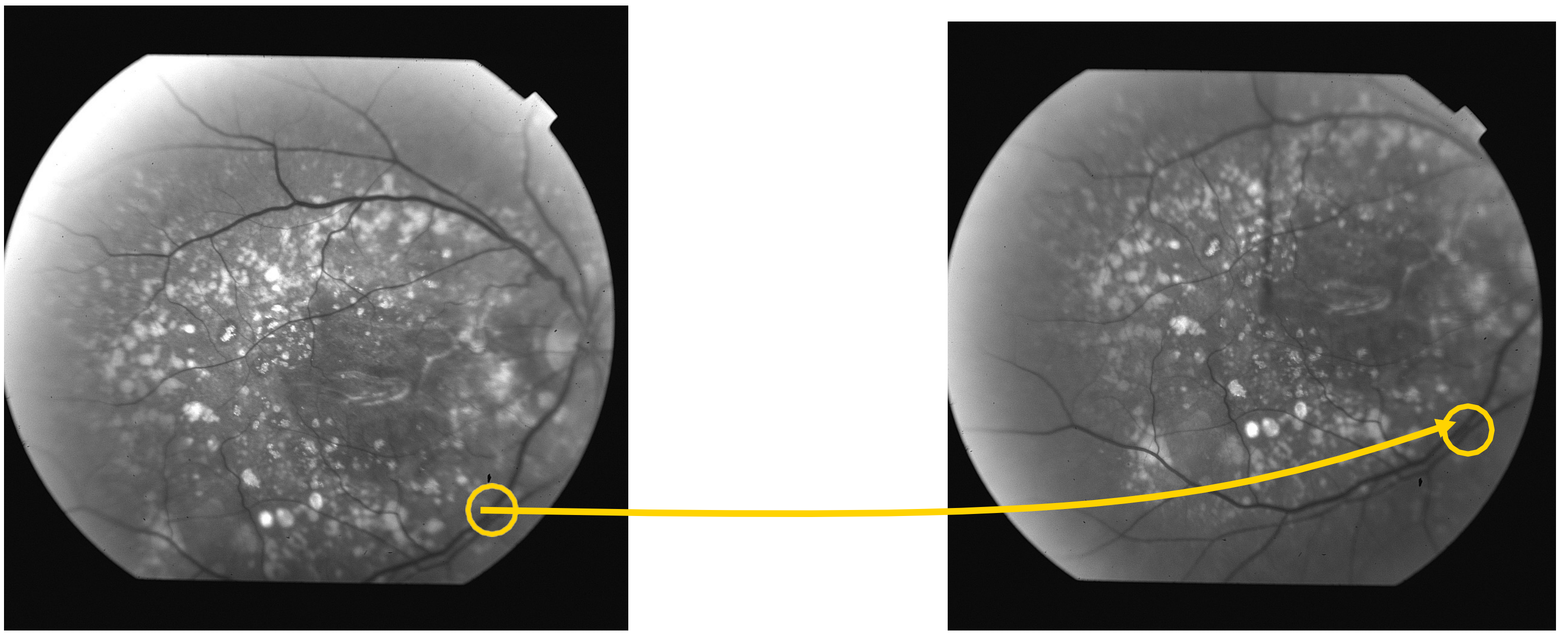
2-D Transformations



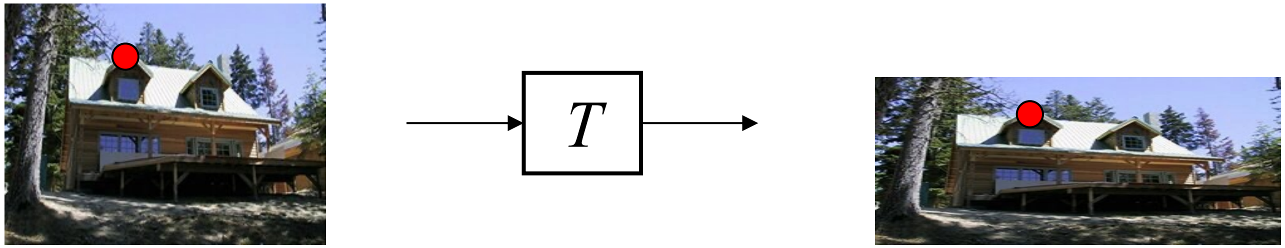
Original

Translation

Not an Easy Problem!!



Parametric (global) warping



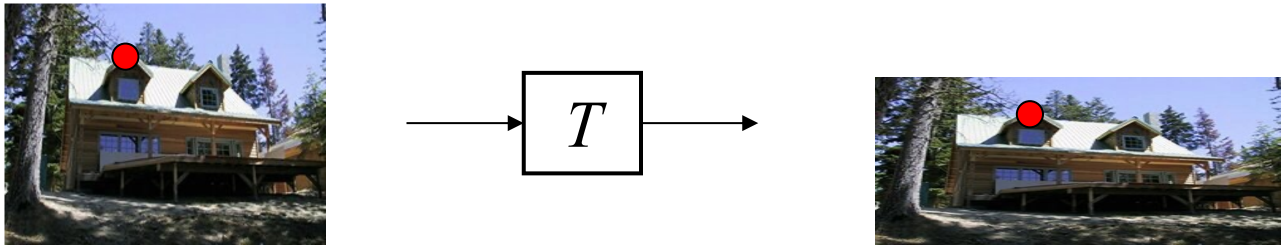
$\mathbf{p} = (x,y)$ $\qquad\qquad$ $\mathbf{p'} = (x',y')$

- Transformation T is a coordinate-changing machine:
- $p' = T(p)$
- What does it mean that T is global?
  - Is the same for any point p
  - can be described by just a few numbers (parameters)
- Let's represent a linear T as a matrix:
- $p' = \mathbf{M}p$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Efros

Parametric (global) warping



$\mathbf{p} = (x,y)$ $\mathbf{p'} = (x',y')$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

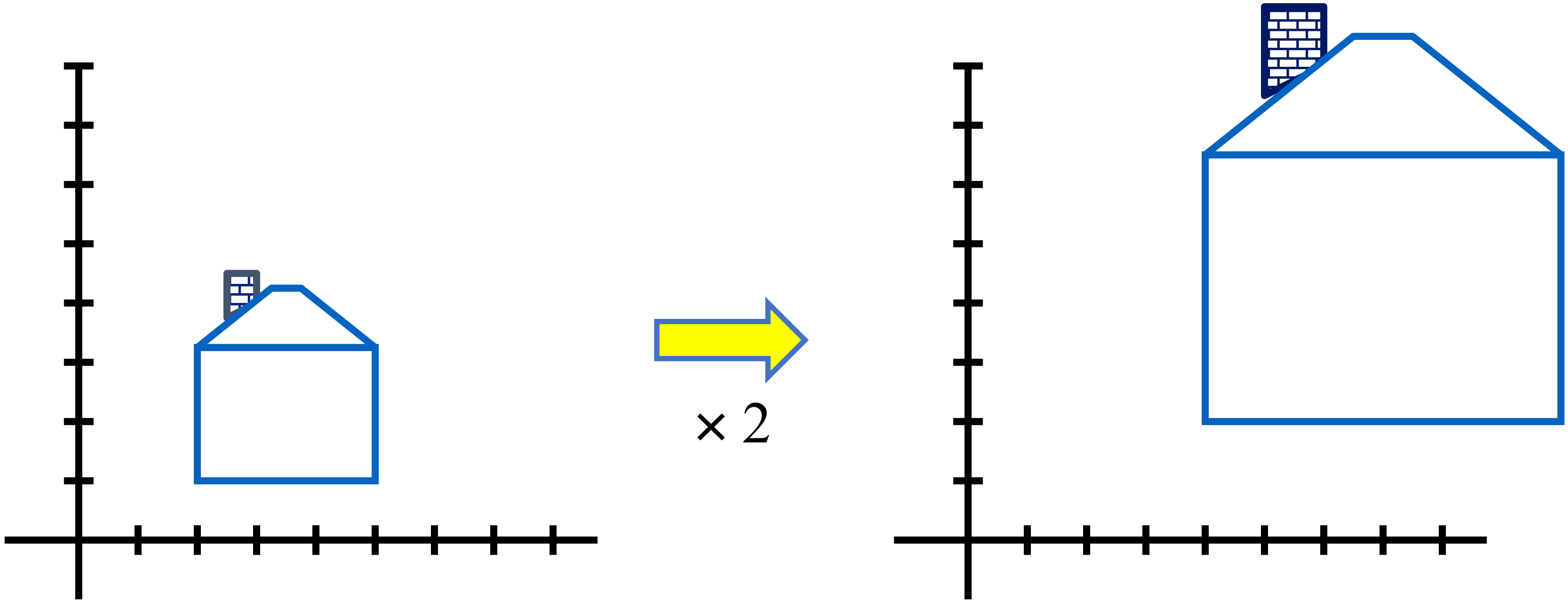
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformation Matrix

Source: Efros

Scaling

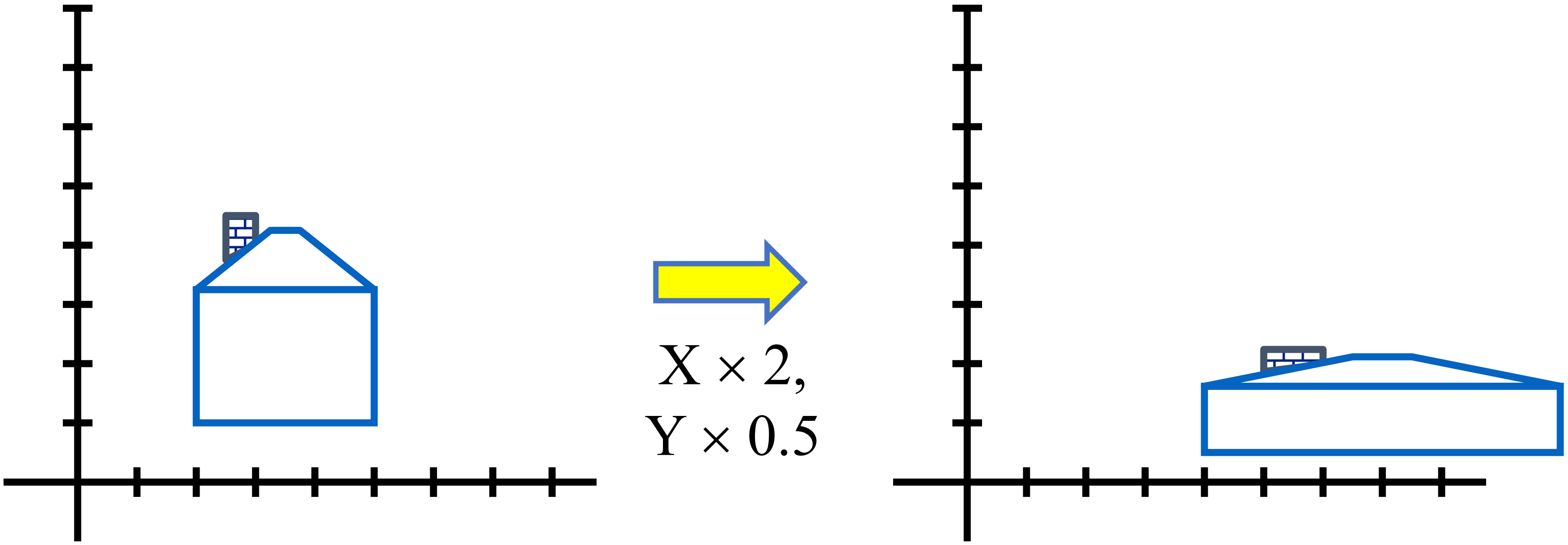
- *Scaling* a coordinate means multiplying each of its components by a scalar
- *Uniform scaling* means this scalar is the same for all components:



Source: Efros

Scaling

- *Non-uniform scaling*: different scalars per component:



Source: Efros

Scaling

- Scaling operation:

$$x' = ax$$

$$y' = by$$

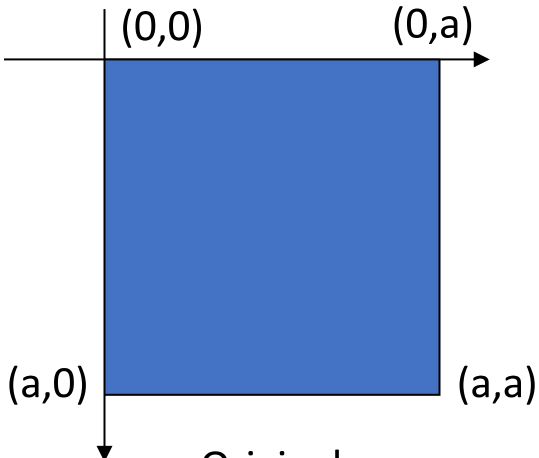
- Or, in matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\textit{scaling matrix } S} \begin{bmatrix} x \\ y \end{bmatrix}$$

What's inverse of S?

Source: Efros

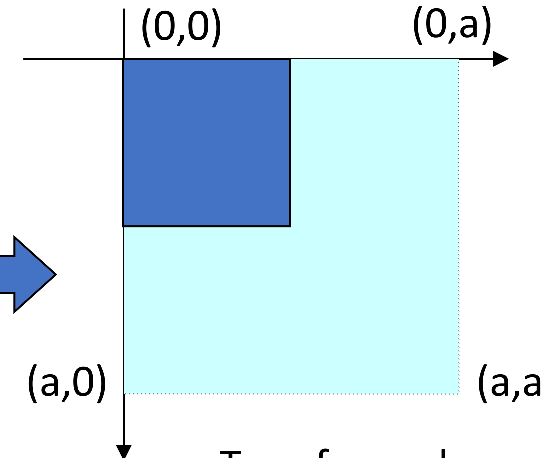
Example



Original

Apply

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



Transformed

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

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

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

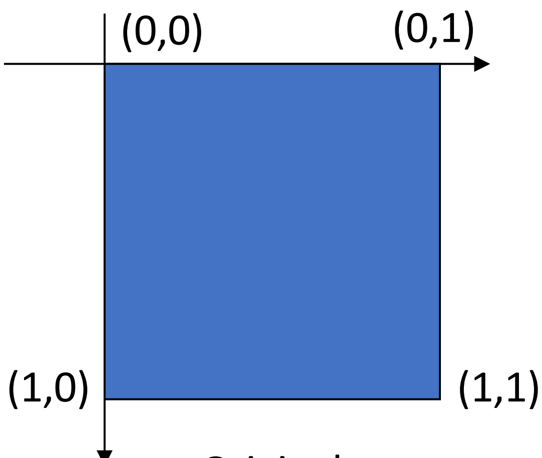
$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}\begin{bmatrix} 0.5a \\ 0.5a \end{bmatrix} = \begin{bmatrix} 0.25a \\ 0.25a \end{bmatrix}$$

2D Transformations

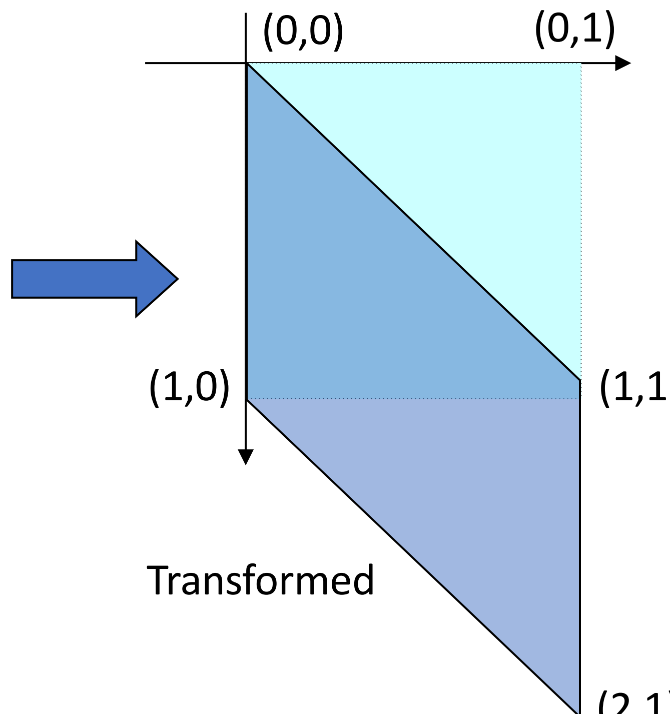
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformation Matrix

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



Original



Transformed

Shear Transformation

- A transformation that slants the shape of an object is called the shear transformation.

- There are two shear transformations **X-Shear** and **Y-Shear**.

- One shifts X coordinates values and other shifts Y coordinate values.

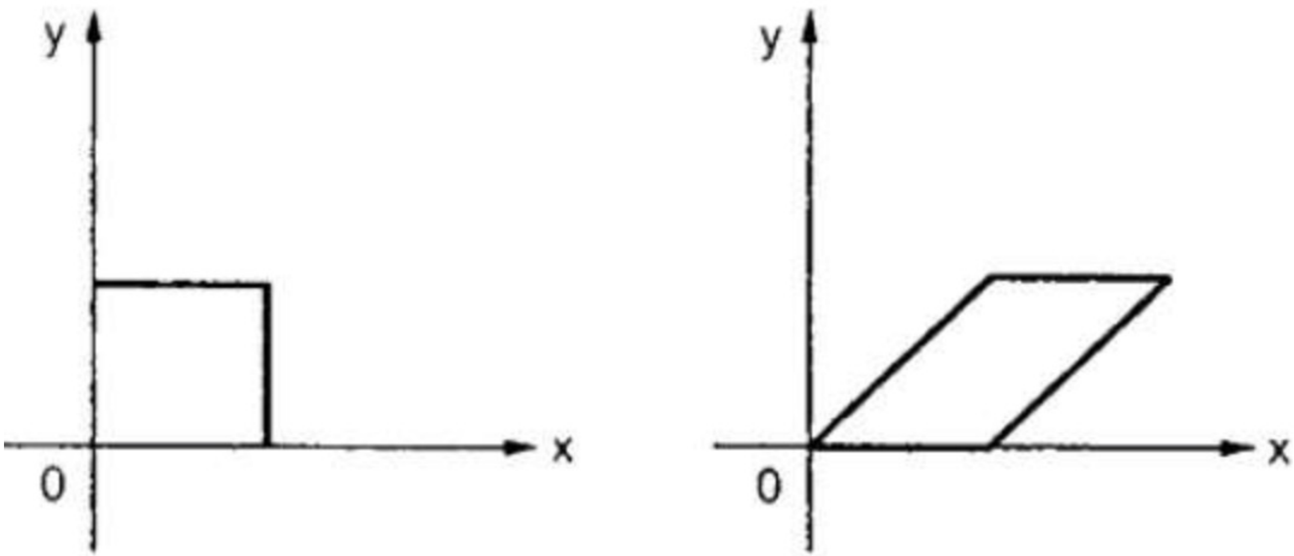
- However; in both the cases only one coordinate changes its coordinates and other preserves its values.

- Shearing is also termed as **Skewing**.

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

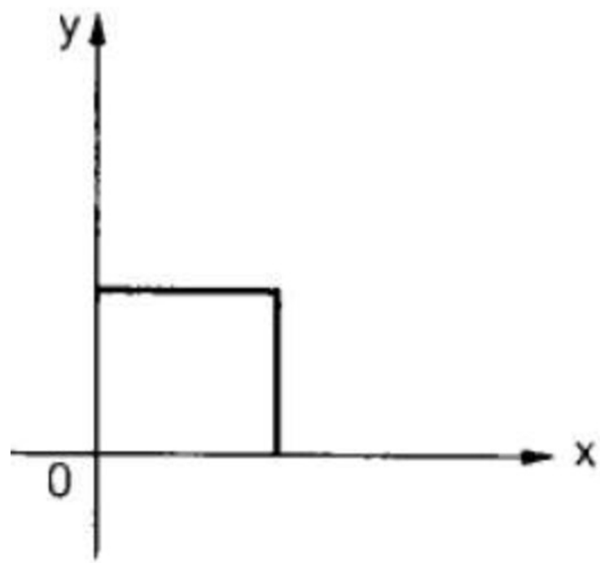


(a) Original object (b) Object after x shear

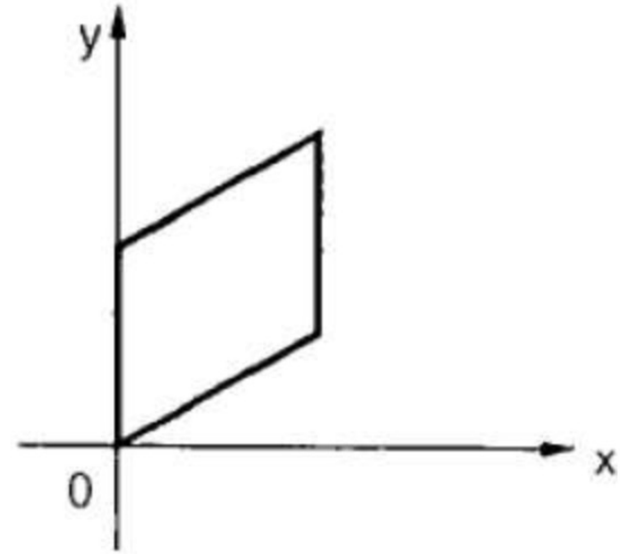
Shear in y-direction

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

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

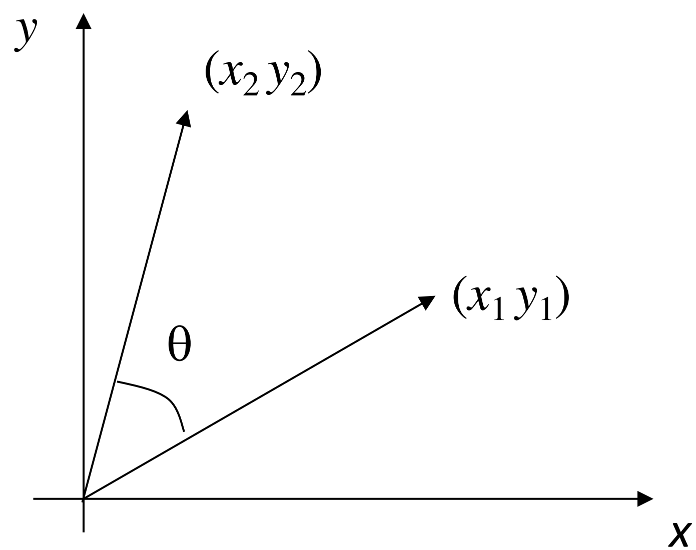


(a) Original object



(b) Object after y shear

Rotation

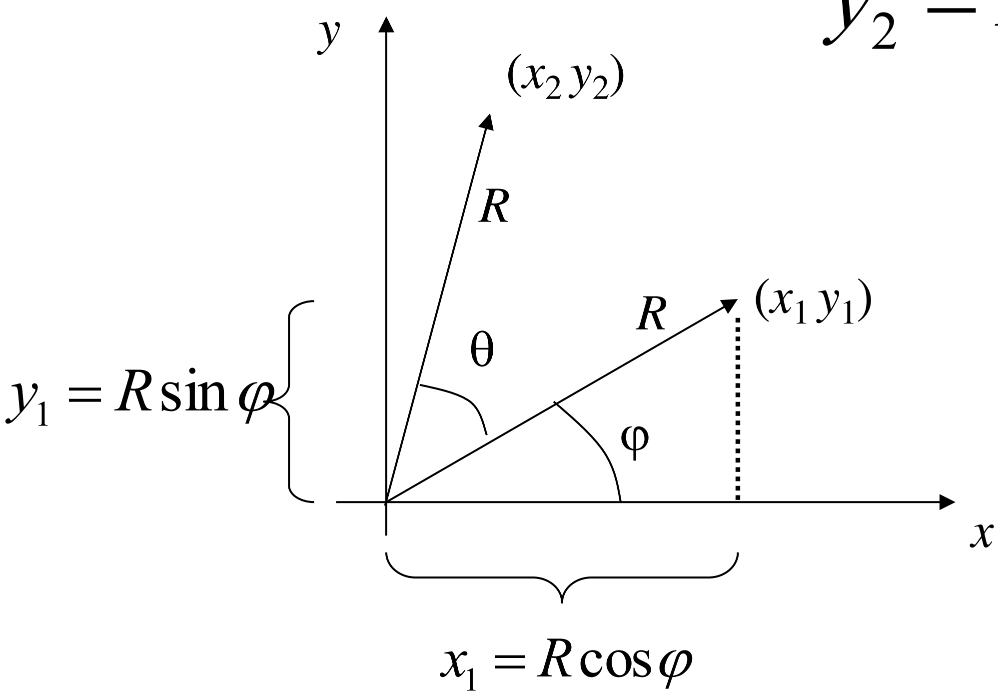


- Task: Relate (x_2, y_2) to (x_1, y_1)

Rotation

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

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



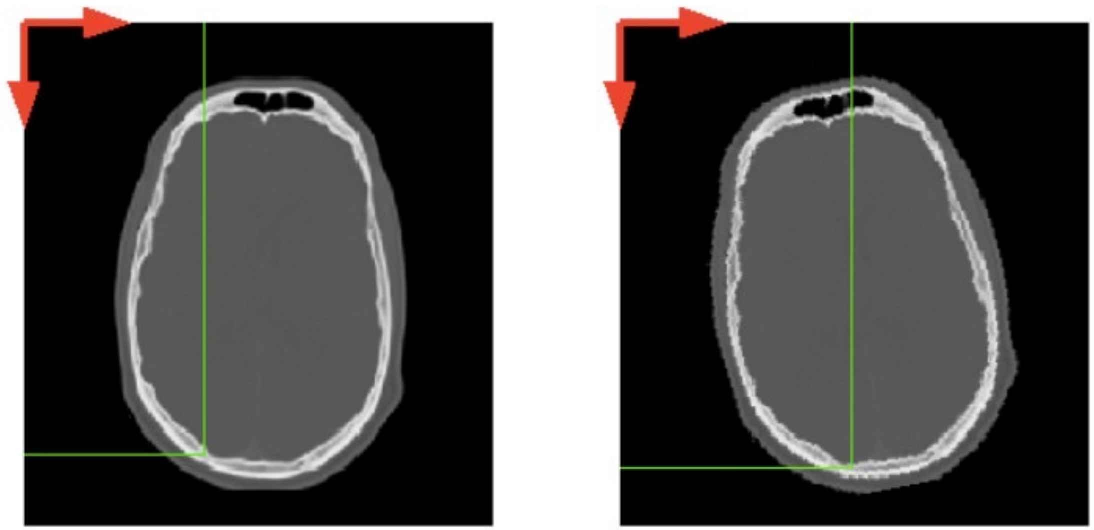
$$x_2 = x_1\cos\theta - y_1\sin\theta$$
$$y_2 = x_1\sin\theta + y_1\cos\theta$$

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

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

Source: Sohaib Khan, LUMS

Rigid Registration - Rotation



One parameter, the angle θ:

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

2-D Rotation

- This is easy to capture in matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x \\ y \end{bmatrix}$$

-

- What is the inverse transformation?
  - Rotation by $-\theta$
  - For rotation matrices [**prove it**]

$$\mathbf{R}^{-1} = \mathbf{R}^{T}$$

Source: Efros

2x2 Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D Identity?

$$x' = x$$
$$y' = y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Scale around (0,0)?

$$\boldsymbol{x}' = \boldsymbol{s}_x * \boldsymbol{x}$$
$$\boldsymbol{y}' = \boldsymbol{s}_y * \boldsymbol{y}$$

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 \\ 0 & \boldsymbol{s}_y \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}$$

Source: Efros

2x2 Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D Rotate around (0,0)?

$$x' = \cos\Theta * x - \sin\Theta * y$$
$$y' = \sin\Theta * x + \cos\Theta * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Shear?

$$x' = x + sh_x * y$$
$$y' = sh_y * x + y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Efros

2x2 Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D Mirror about Y axis?

$$x' = -x$$
$$y' = y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Mirror over (0,0)?

$$x' = -x$$
$$y' = -y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Efros

2x2 Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D Translation?

$$x' = x + t_x$$

$$y' = y + t_y$$

NO!

Only linear 2D transformations can be represented with a 2x2 matrix

Source: Efros

All 2D Linear Transformations

- Linear transformations are combinations of …
  - Scale,
  - Rotation,
  - Shear, and
  - Mirror
- Properties of linear transformations:
  - Origin maps to origin
  - Lines map to lines
  - Parallel lines remain parallel
  - Ratios are preserved
  - Closed under composition

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} i & j \\ k & l \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Efros

Homogeneous Coordinates

•***Homogeneous coordinates***

- represent coordinates in 2 dimensions with a 3-vector

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{\text{homogeneous coords}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Source: Efros

Homogeneous Coordinates

- **Q: How can we represent translation as a 3x3 matrix?**

$$x' = x + t_x$$

$$y' = y + t_y$$

- A: Using the rightmost column:

$$\mathbf{T}ranslation = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

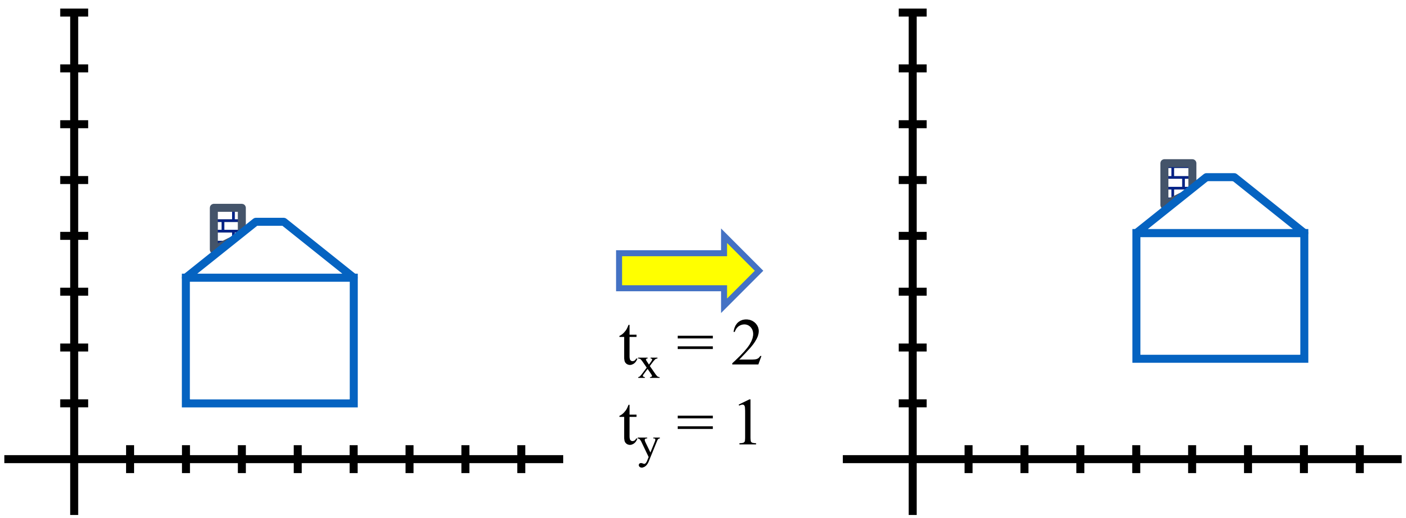
Source: Efros

Translation

- Example of translation

Homogeneous Coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$



Source: Efros

Basic 2D Transformations

- Basic 2D transformations as 3x3 matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s_x} & 0 & 0 \\ 0 & \boldsymbol{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{sh_x} & 0 \\ \boldsymbol{sh_y} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Shear

Source: Efros

Matrix Composition

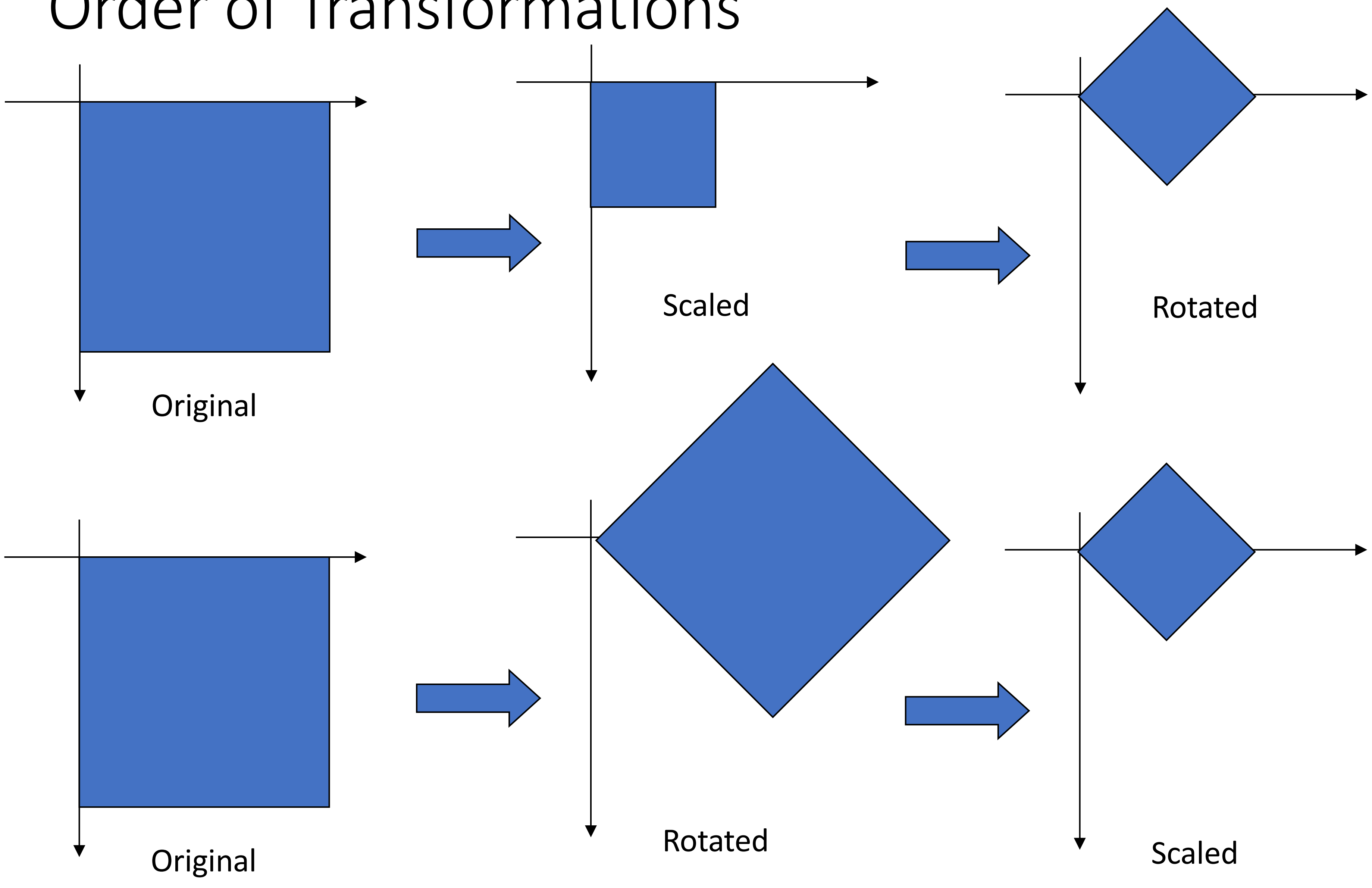
- Transformations can be combined by matrix multiplication

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

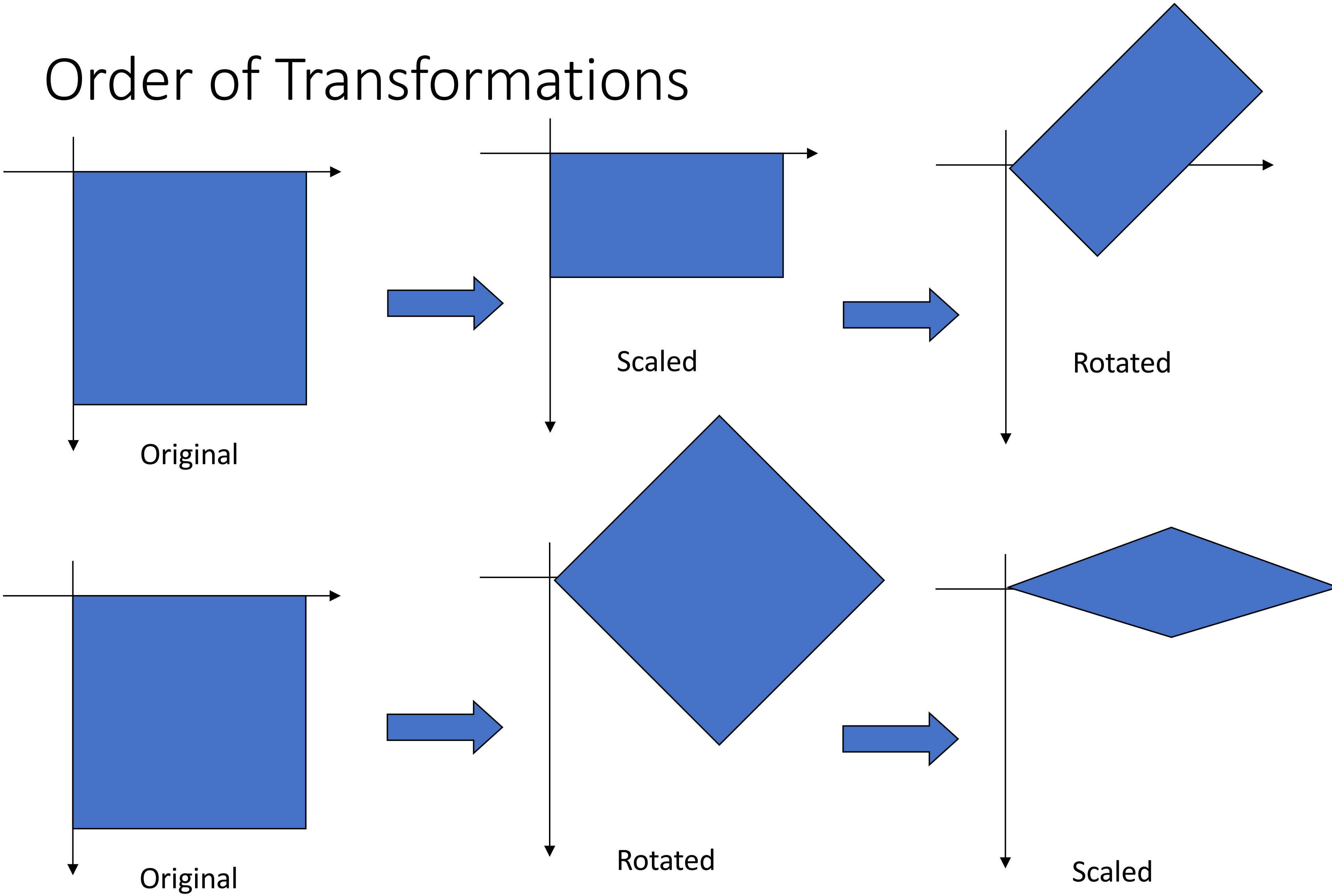
$$\mathbf{p}' \quad = \quad T(t_x, t_y) \qquad R(\Theta) \qquad S(s_x, s_y) \qquad \mathbf{p}$$

Source: Efros

Order of Transformations



Order of Transformations



Order of Transformations

- In general **AB ≠ BA**
- However, in specific cases, this might hold true
- In the previous example, if $s_x = s_y$, then order of transformations does not matter

Inverse Transformations

- Inverse transformation should 'undo' the effect of the original transformation
- Simply taking the matrix inverse will work $$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

- Inverse Transforms

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & -e_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Remember that when inverting concatenation of transforms, their order reverses

$$(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}$$

Hierarchy of Transformation Groups

- Translation

$$\mathbf{x}' = \left[\begin{array}{c|c} \mathbf{I}_{2\times 2} & \mathbf{t} \end{array} \right]_{2\times 3} \bar{\mathbf{x}}$$

$$\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \end{aligned}$$

- Rigid Body Transformation

$$\mathbf{x}' = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array} \right]_{2\times 3} \bar{\mathbf{x}}$$

$$\begin{aligned} x' &= x\cos\theta - y\sin\theta + t_x \\ y' &= x\sin\theta + y\cos\theta + t_y \end{aligned}$$

- Similarity

$$\mathbf{x}' = \left[\begin{array}{c|c} s\mathbf{R} & \mathbf{t} \end{array} \right]_{2\times 3} \bar{\mathbf{x}}$$

$$\begin{aligned} x' &= sx\cos\theta - sy\sin\theta + t_x \\ y' &= sx\sin\theta + sy\cos\theta + t_y \end{aligned}$$

Hierarchy of Transformation Groups

- Translation

$$\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \end{aligned}$$

- Rigid Body Transformation

$$\begin{aligned} x' &= x\cos\theta - y\sin\theta + t_x \\ y' &= x\sin\theta + y\cos\theta + t_y \end{aligned}$$

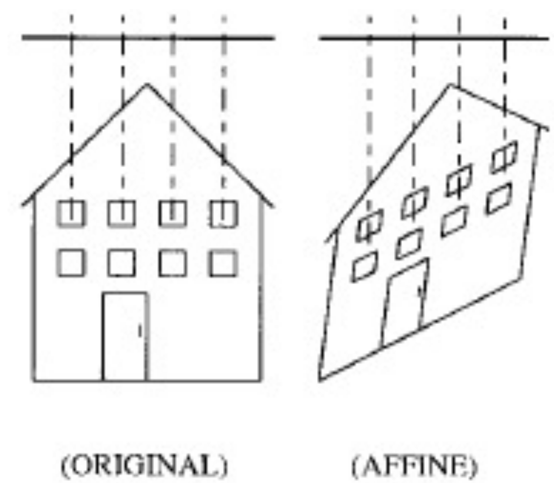
- Similarity

$$x' = sx\cos\theta - sy\sin\theta + t_x$$
$$y' = sx\sin\theta + sy\cos\theta + t_y$$

Affine Transformations

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- Affine transformations are combinations of …
  - Linear transformations rotation, scaling, shear, and
  - Translations
- Properties of affine transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines remain parallel
  - Ratios are preserved



Ref: Steve Mann & Rosalind W. Picard, "Video Orbits of the Projective Group: A simple approach to featureless estimation of parameters", IEEE Trans. on Image Processing, Vol. 6, No. 9, September 1997

Source: Efros / Sohaib Khan, LUMS

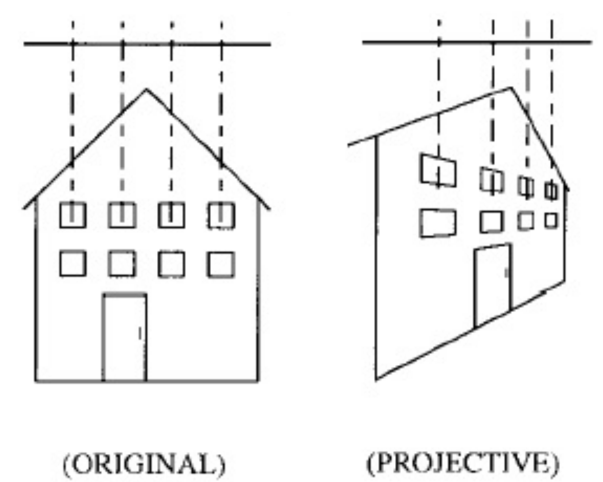
Projective Transformations

$$\mathbf{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \bar{\mathbf{x}}$$

- Projective transformations …
- Properties of projective transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines do not necessarily remain parallel
  - Ratios are not preserved
  - Closed under composition
  - Models change of basis
  - Simulates out of plane rotations

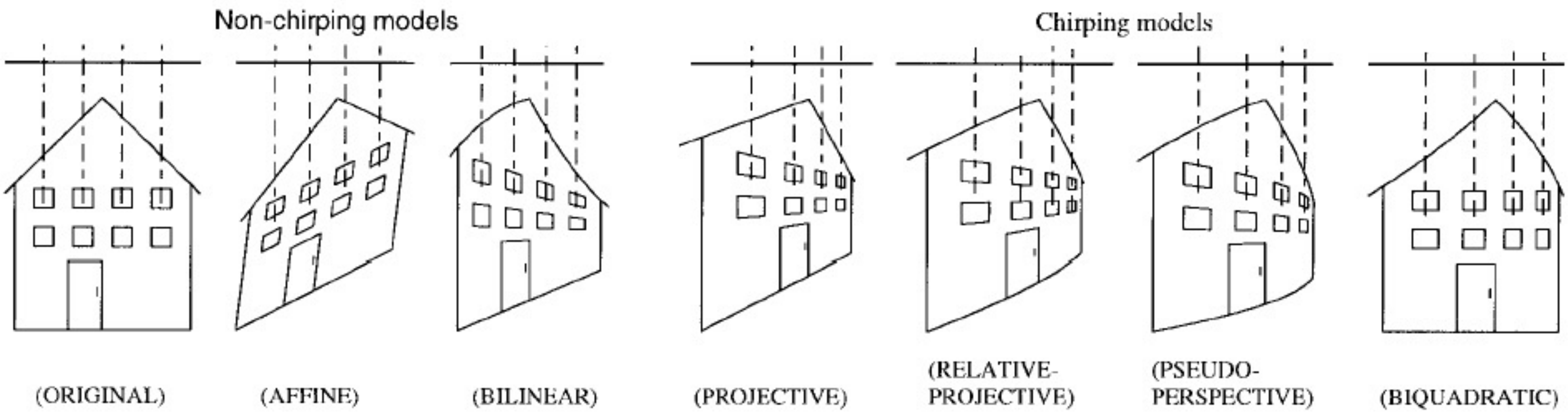
$$x' = \frac{a_{00}x + a_{01}y + a_{02}}{a_{20}x + a_{21}y + a_{22}}$$

$$y' = \frac{a_{10}x + a_{11}y + a_{12}}{a_{20}x + a_{21}y + a_{22}}$$



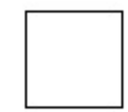
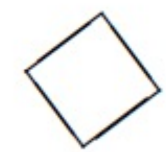
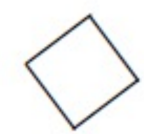
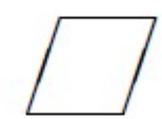
Source: Sohaib Khan, LUMS

Displacement Models



Ref: Steve Mann & Rosalind W. Picard, “Video Orbits of the Projective Group: A simple approach to featureless estimation of parameters”, IEEE Trans. on Image Processing, Vol. 6, No. 9, September 1997

Hierarchy of 2D Transformations

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Basic 2D transformations

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \alpha_x \\ \alpha_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shear

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotate

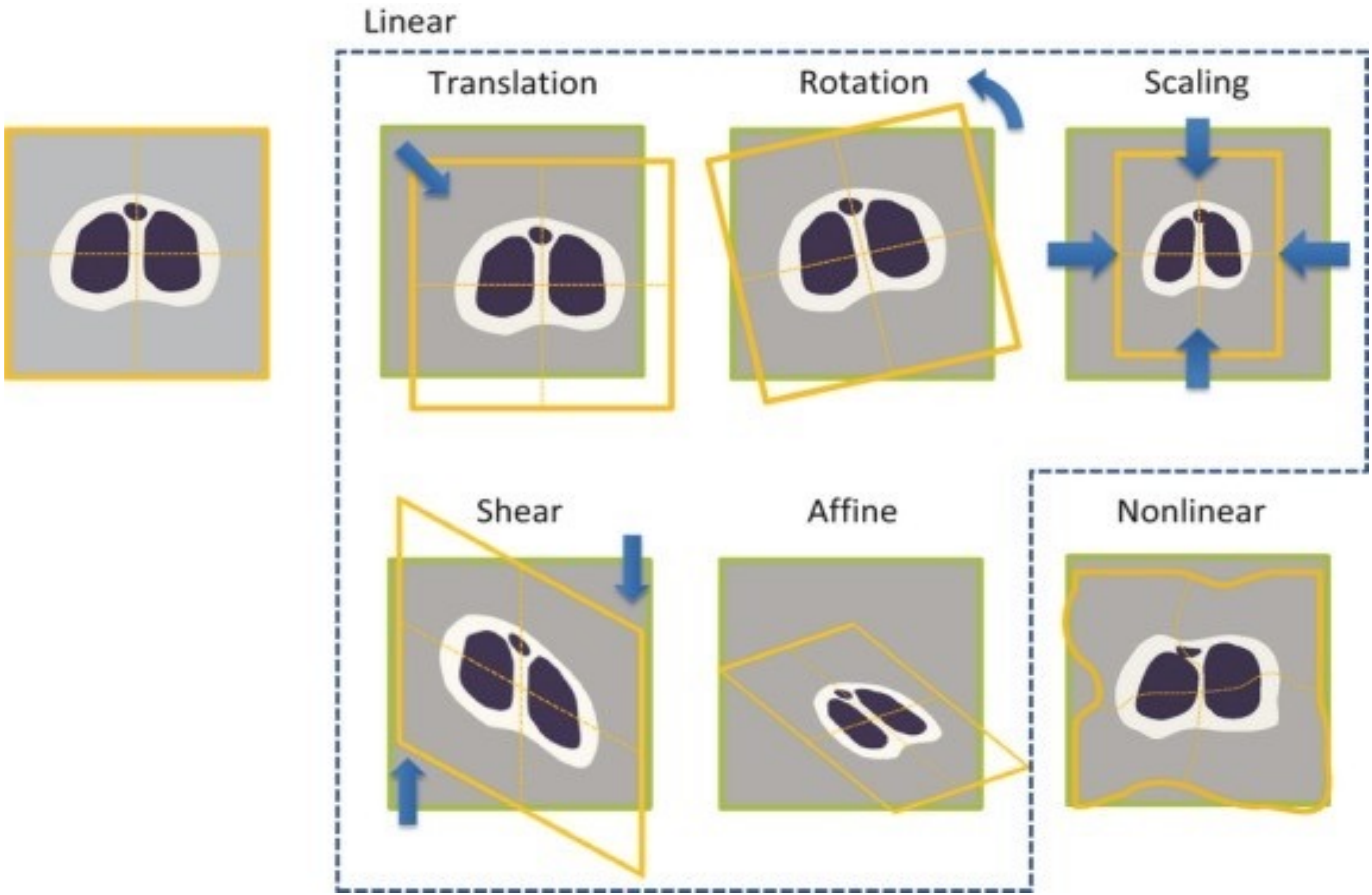
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

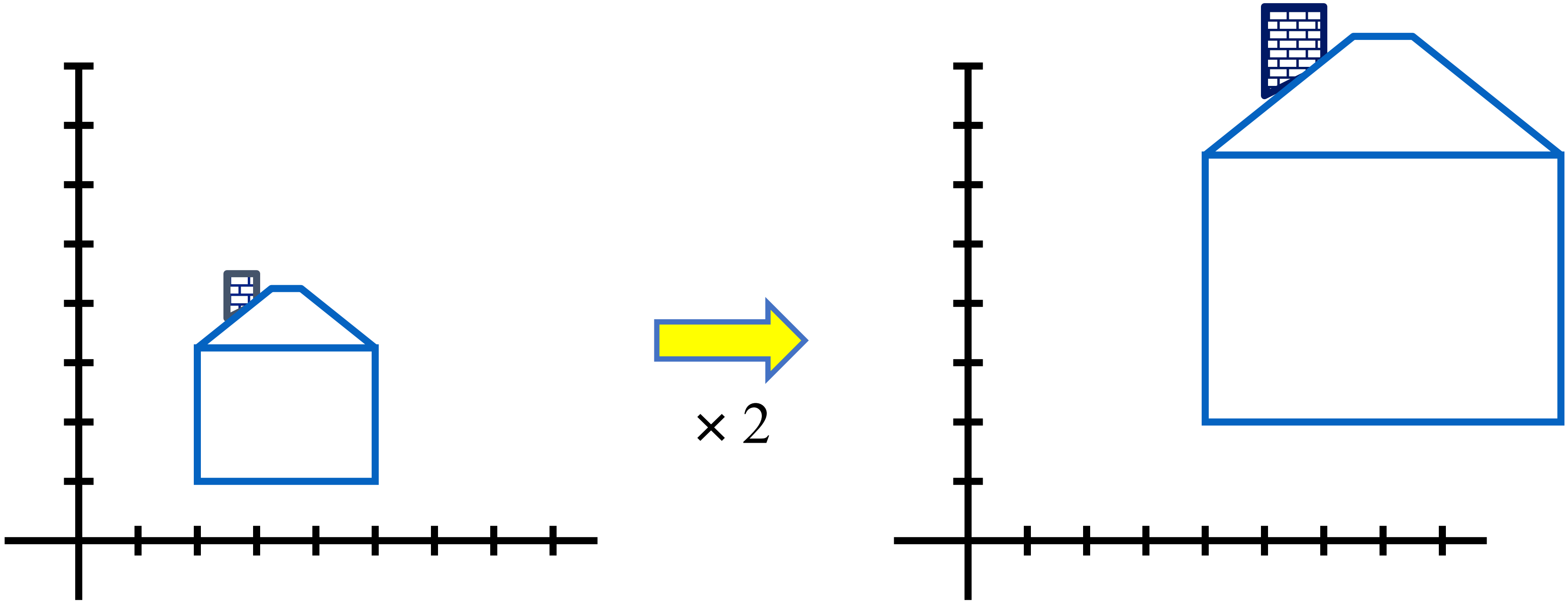
Affine

Affine is any combination of translation, scale, rotation, shear


Linear
Translation
Rotation
Scaling
Shear
Affine
Nonlinear

Scaling

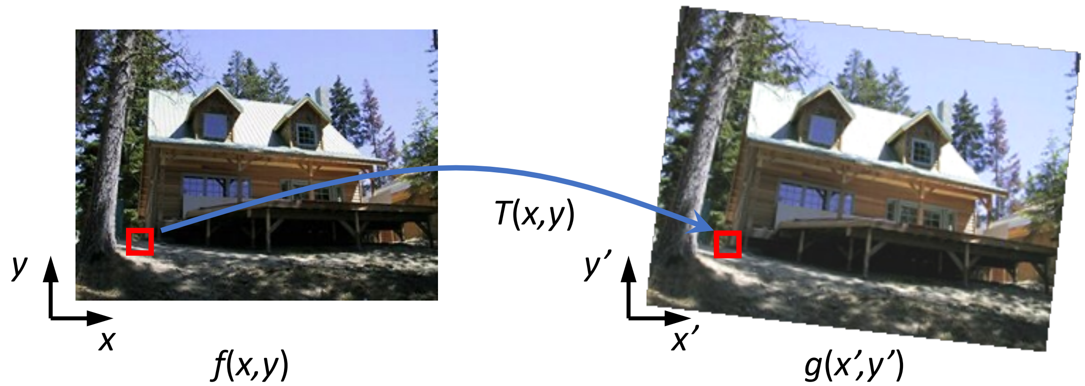
- *Scaling* a coordinate means multiplying each of its components by a scalar
- *Uniform scaling* means this scalar is the same for all components:



Source: Efros

Image warping

Image Warping is application of transformation such that it changes 'spatial' configuration of an image

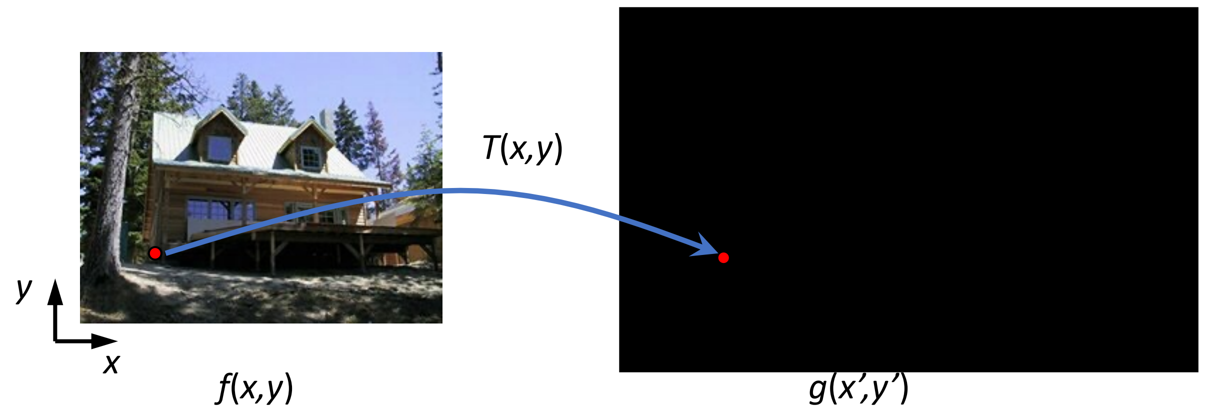


- Given a coordinate transform *(x',y')* = *h*(*x,y*) and a source image *f*(*x,y*), how do we compute a transformed image *g(x',y')* = *f*(*T*(*x,y*))?

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?

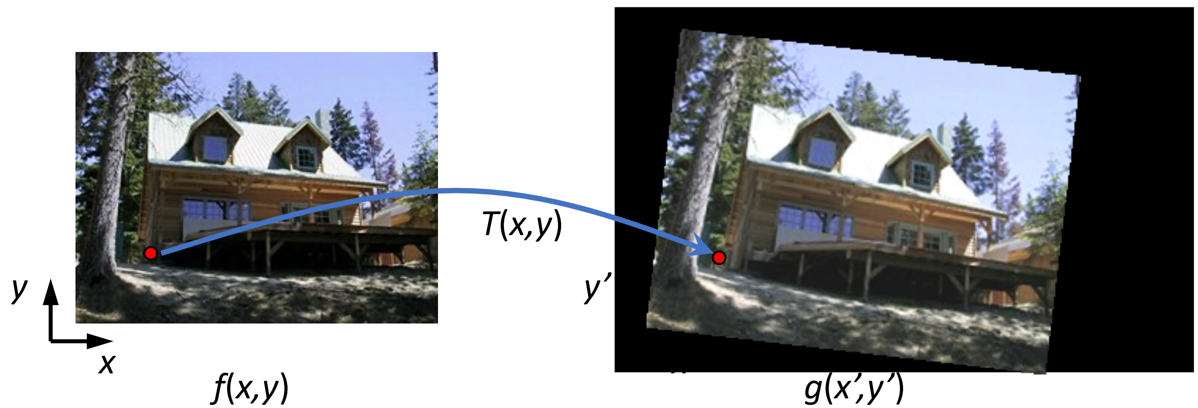
Forward warping



- Send each pixel $f(x,y)$ to its corresponding location
- $(x',y') = T(x,y)$ in the second image

Q: what if pixel lands "between" two pixels?

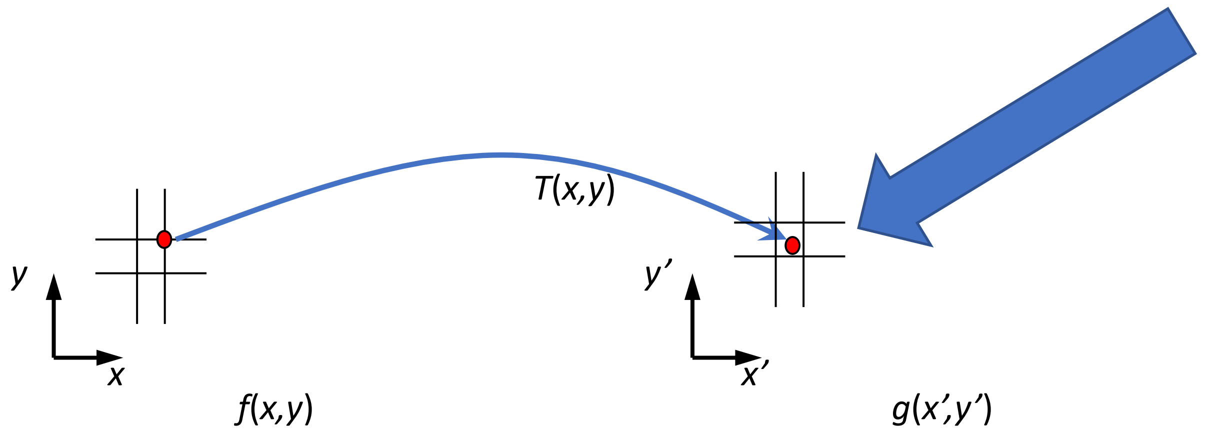
Forward warping



- Send each pixel $f(x,y)$ to its corresponding location
- $(x',y') = T(x,y)$ in the second image

Q: what if pixel lands “between” two pixels?

Forward warping



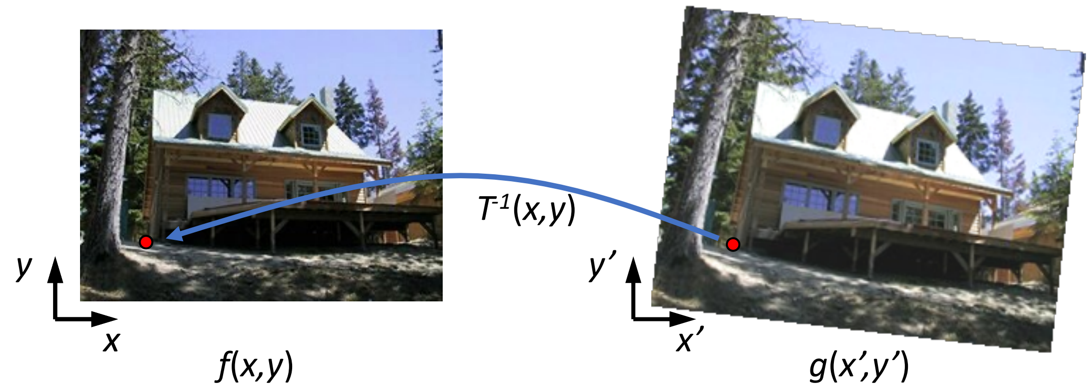
- Send each pixel $f(x,y)$ to its corresponding location
- $(x',y') = T(x,y)$ in the second image

Q: what if pixel lands “between” two pixels?

A: distribute color among neighboring pixels (x’,y’)

– Known as “splatting” NOT A SIMPLE TASK

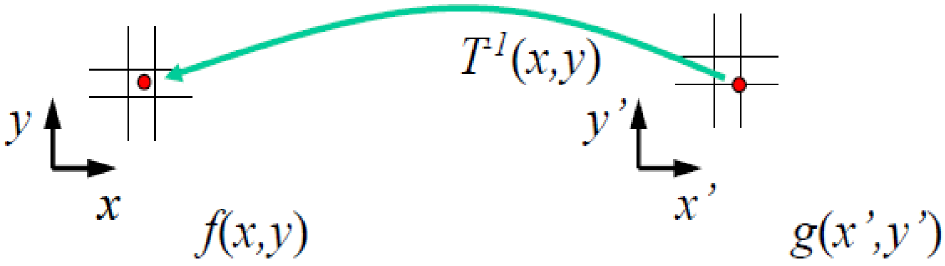
Inverse warping



- Get each pixel $g(x',y')$ from its corresponding location
- $(x,y) = T^{-1}(x',y')$ in the first image

Q: what if pixel comes from "between" two pixels?

Inverse Mapping



Get each pixel $g(x',y')$ from its corresponding location

$(x,y) = T^{-1}(x',y')$ in the first image

Q: what if pixel comes from "between" two pixels?

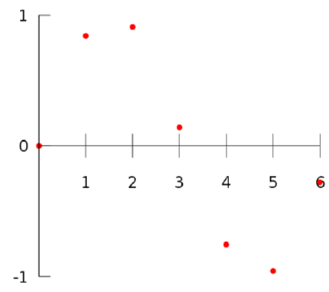
A: *Interpolate* color value from neighbors

– nearest neighbor, bilinear, Gaussian, bicubic

Interpolations

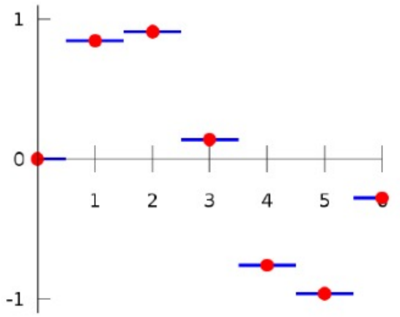
Interpolation

interpolation is a method of constructing new data points within the range of a discrete set of known data points.

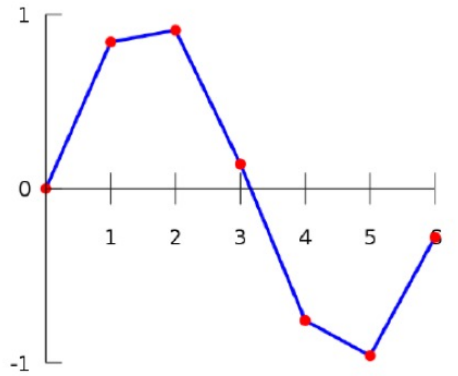


Interpolations

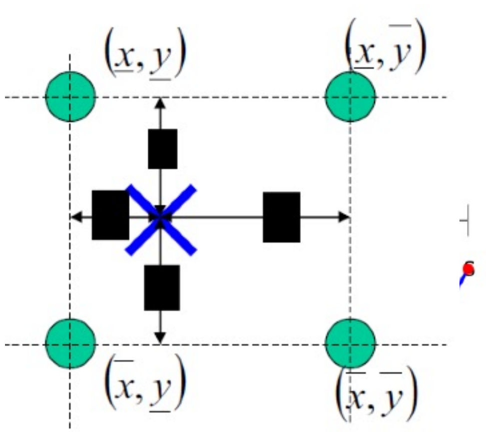
- Nearest Neighbor Interpolations



- Linear Interpolation

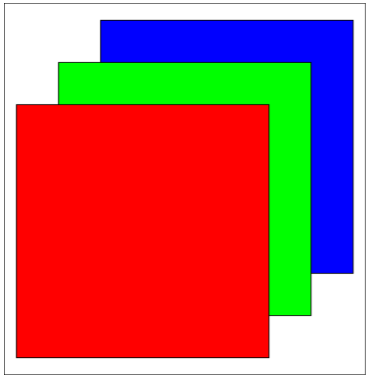


- 2D Bilnear Interpolation



Warping Color Images

- Same as applying to Gray Scale Images
- Since Color images consists of
  - More than one 2D matrix
  - In case or RGB image
    - Apply transformation to each layer separately
  - Bilinear Interpolation
    - Also applied separately



2D Transformation & Image Registration

- **Definition**: A mapping from one 2D coordinate system to another
- Also called
  - spatial transformation,
  - geometric transformation,
  - warp
- Image Registration: Process of transforming two images so that same features overlap

Recovering the best transformation

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Source: Sohaib Khan, LUMS

Recovering Best Affine Transformation

- Given two images with unknown transformation between them…

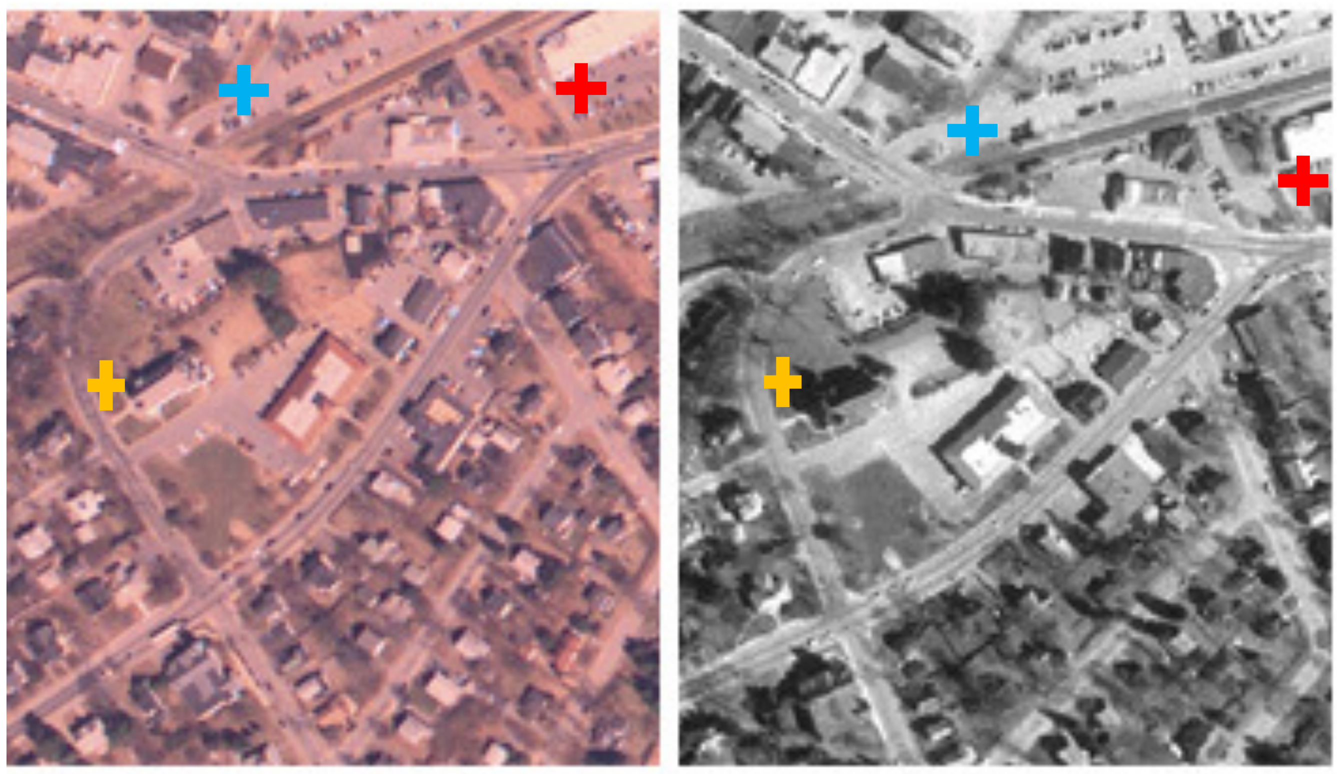
- Compute the values for [a_1 … a_6]

Source: updated from Sohaib Khan, LUMS

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

Source: updated from Sohaib Khan, LUMS

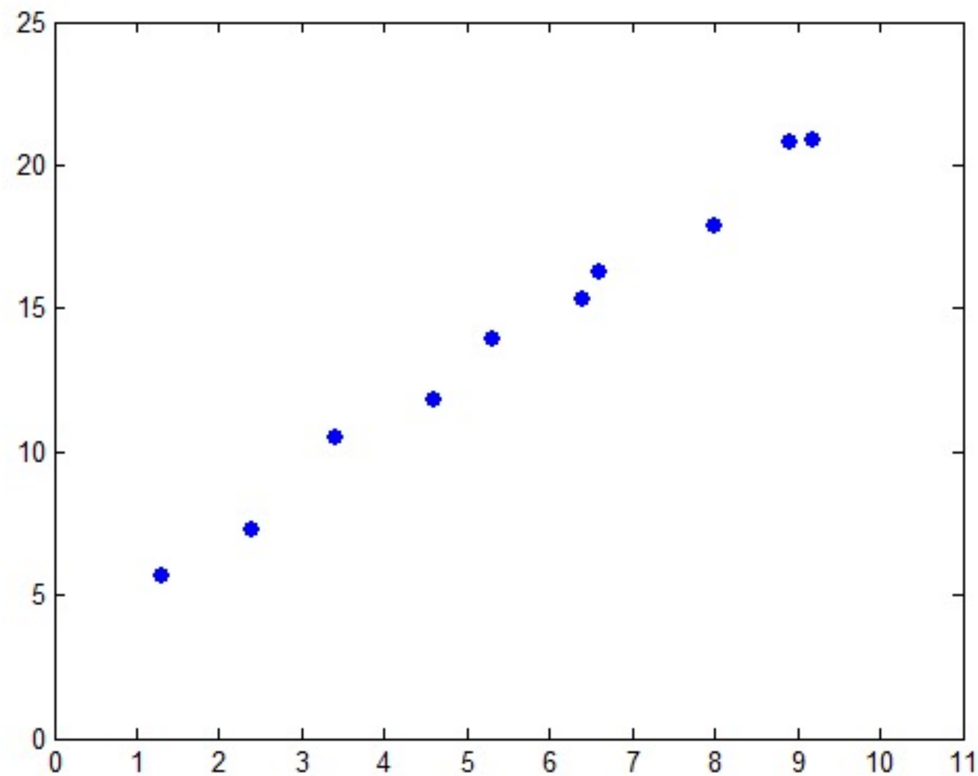
HM (a):

Why we need 3 points for six variables?

Parameter Optimization:
Least Squared Error Solutions

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 ?

Source: Sohaib Khan, LUMS

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

- 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

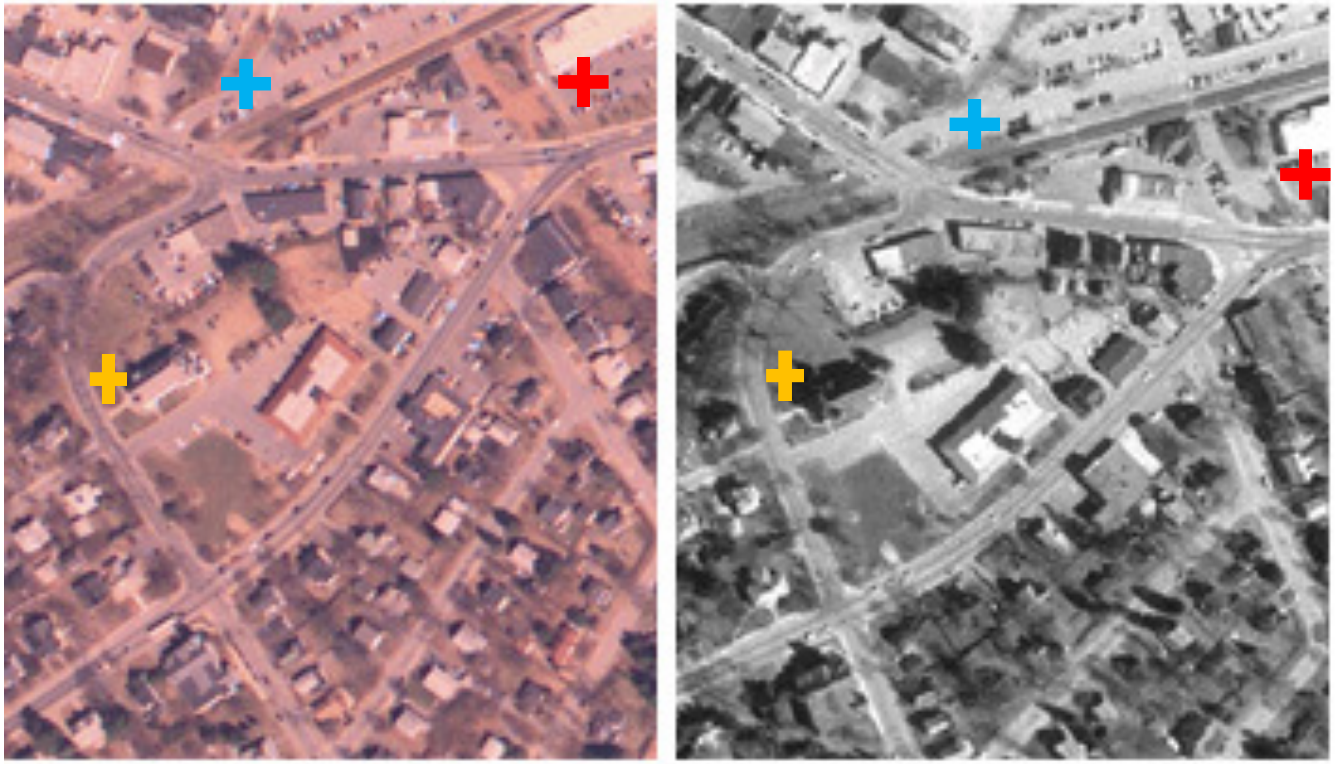
HM (b):
Differentiate Error w.r.t. parameters, put equal to zero and solve for minimum point

Source: Sohaib Khan, LUMS

Recovering Best Affine Transformation

Input: Set of correspondences

- Image 1: (x_i , y_i) Image 2: (x_i', y_i')



Source: updated from Sohaib Khan, LUMS

Recovering Best Affine Transformation

Least Squares Error Solution

- Is the solution (i.e. set of parameters a_1 … a_6) such that the sum of the square of error in each corresponding point is as minimum as possible
- No other set of parameters exists that may have a lower error (in the squared error sense)

Source: Sohaib Khan, LUMS

Recovering Best Affine Transformation

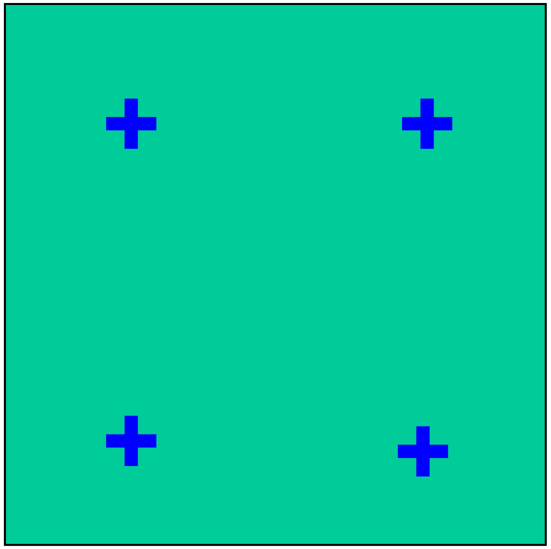


Image 1

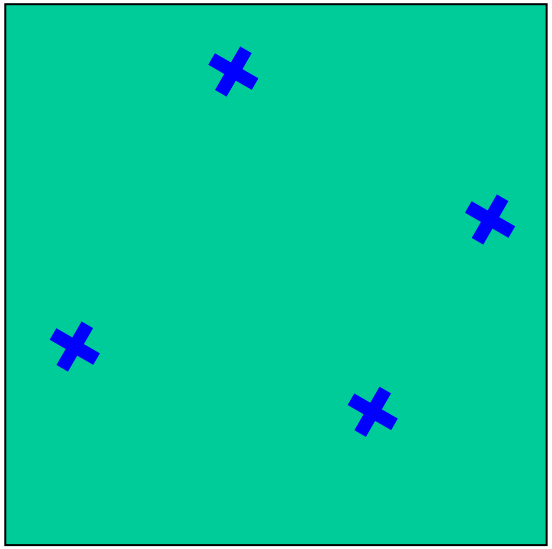
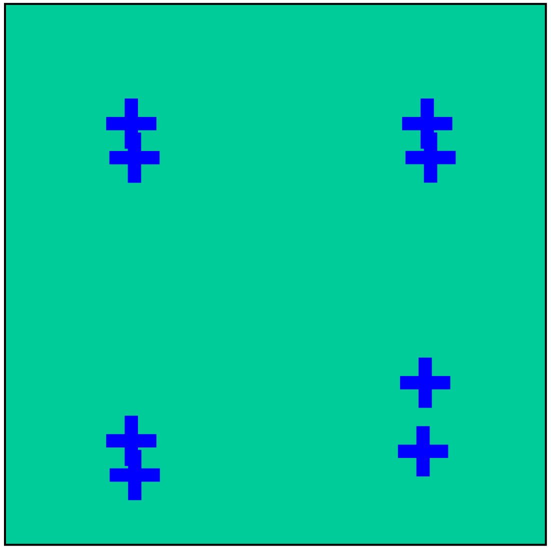


Image 2

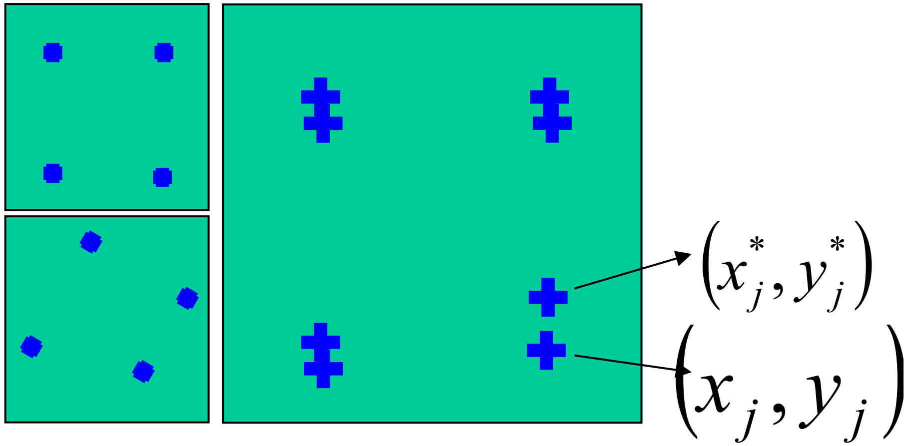


Overlap of points after recovering the transformation

We can try to find the set of parameters in which the **error** is minimum

Least Squares Error Solution

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



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

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

Source: Sohaib Khan, LUMS

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. a

Compute $\partial E / \partial a_i$, put equal to zero, solve simultaneously

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

Source: Sohaib Khan, LUMS

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. **a**

Compute $\partial E / \partial a_i$, put equal to zero, solve simultaneously

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

HM (c): How?

Recovering Best Affine Transformation (alternate way)

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

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

Source: Sohaib Khan, LUMS

Recovering Best Affine Transformation

Given three pairs of corresponding points, we get 6 equations

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

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

Source: Sohaib Khan, LUMS

Recovering Best Affine Transformation

What if we knew four corresponding points?

We should be able to utilize the additional information

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \\ x_4 & y_4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix}$$

Source: Sohaib Khan, LUMS

Recovering Best Affine Transformation

Ax = B

Cannot take inverse directly

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \\ x_4 & y_4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix}$$

Source: Sohaib Khan, LUMS

Pseudo inverse

For an over-constrained linear system

$Ax = B$

A has more rows than columns

Multiply by A^T on both sides

$A^T A x = A^T B$

$A^T A$ is a square matrix of as many rows as x

We can take its inverse

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

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

Source: Sohaib Khan, LUMS

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

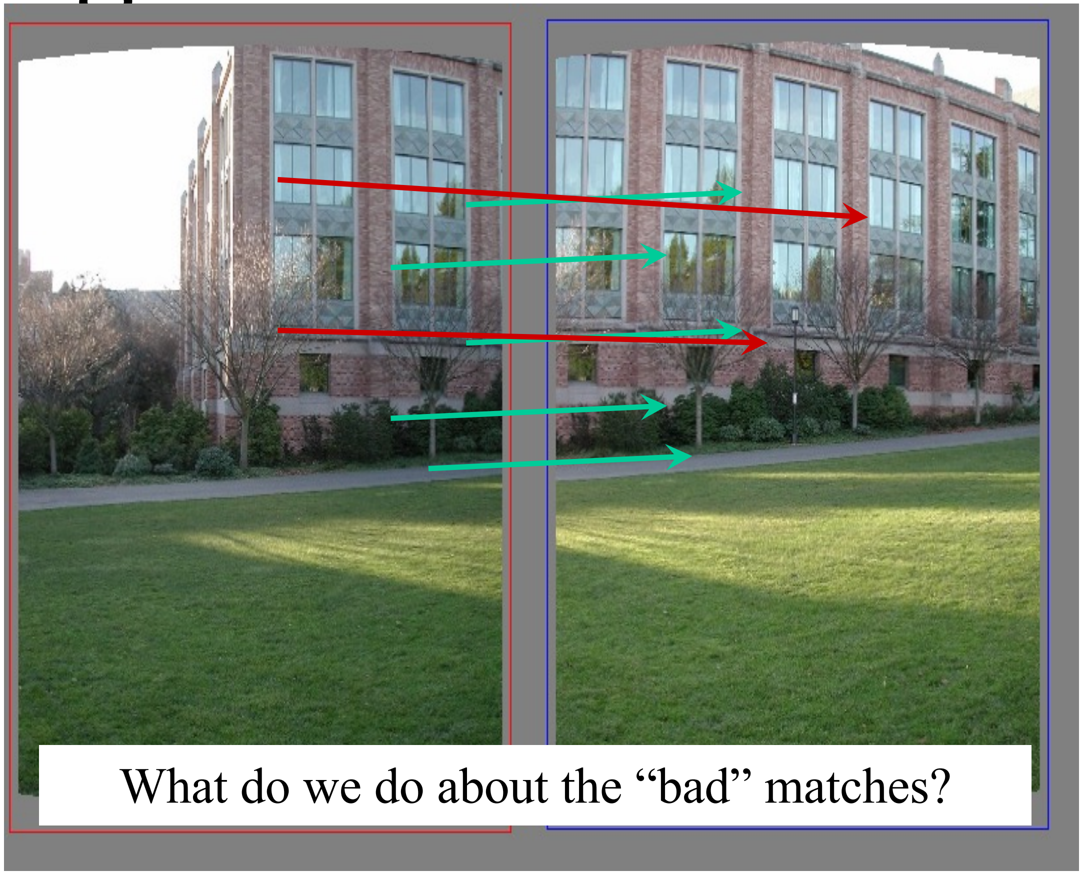
Source: Sohaib Khan, LUMS

RANSAC Algorithm

- "Random Sample Consensus"
  - Can be used to find inliers for any type of model fitting
- Widely used in computer vision
- Requires two parameters:
  - The number of trials N (how many do we need?)
  - The agreement threshold (how close does an inlier have to be?)

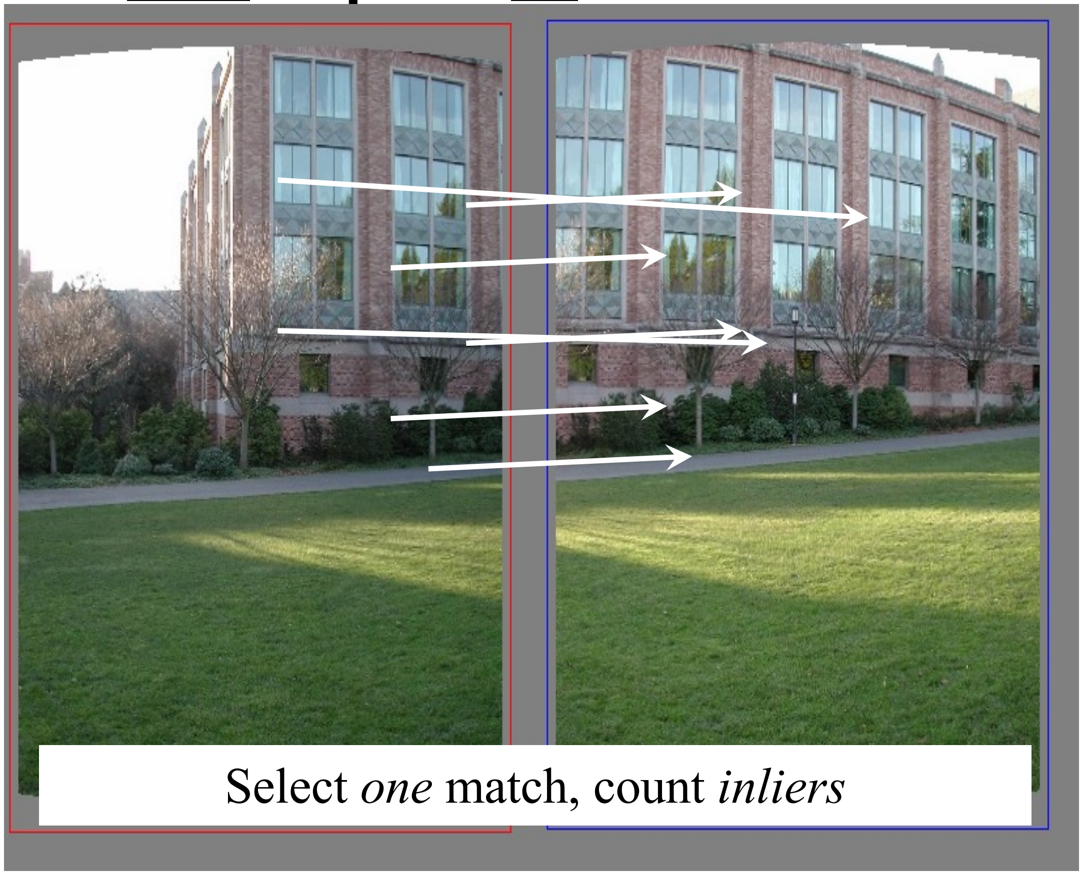
Fischler, M. A., & Bolles, R. C. 1981. Random sampling consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the Associations for Computing Machinery, 24(26), 381–395.

Matching features

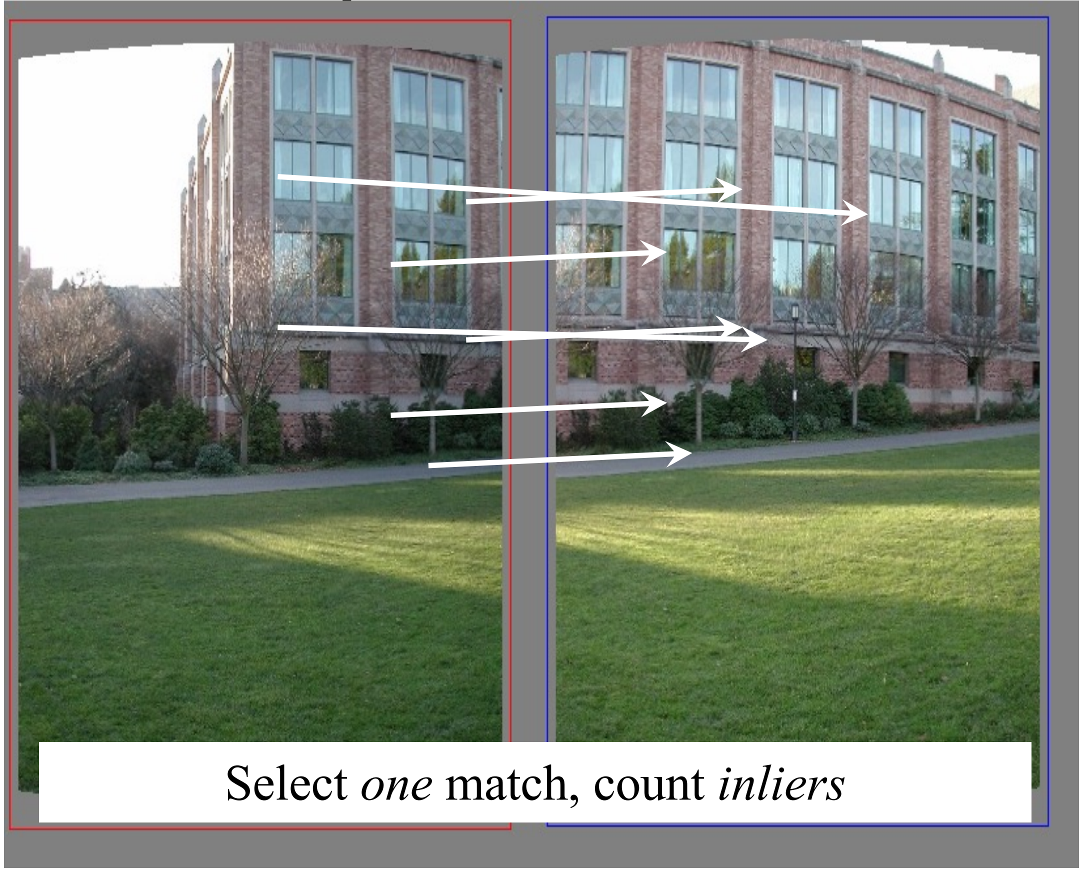


What do we do about the “bad” matches?

RAndom SAmple Consensus

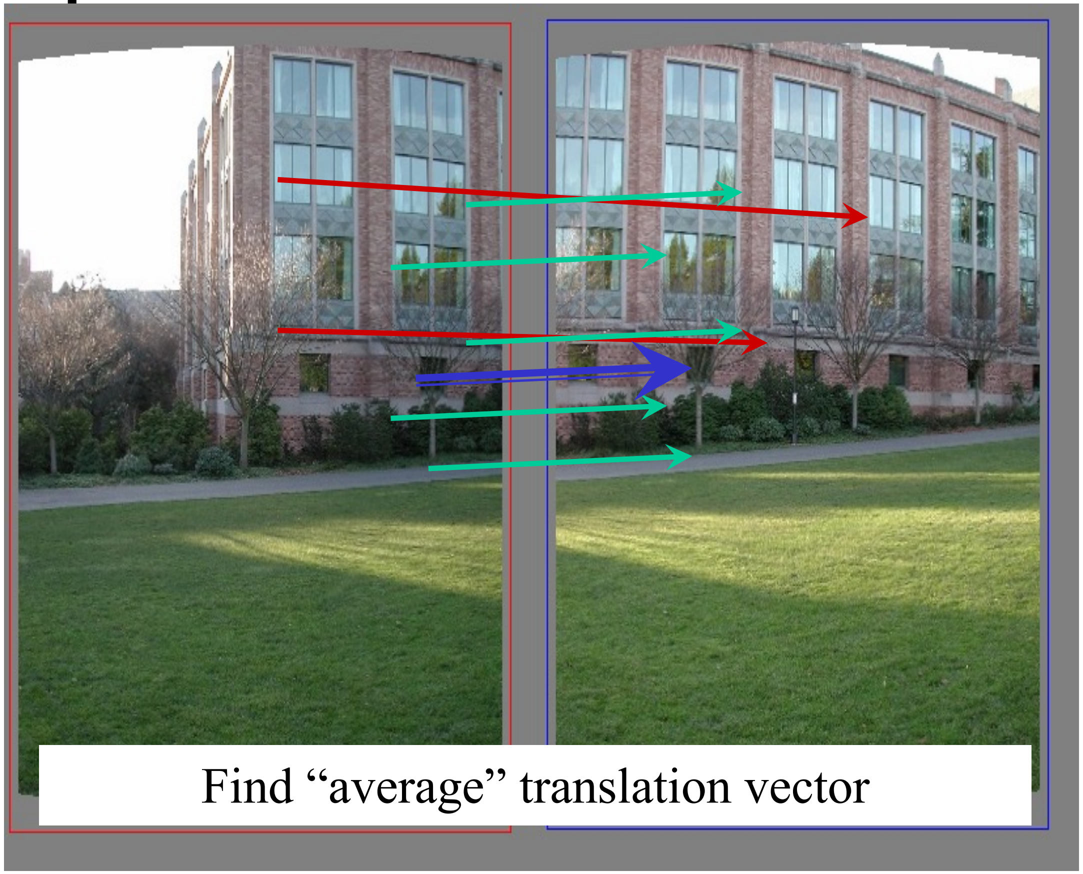


RAndom SAmple Consensus



Select *one* match, count *inliers*

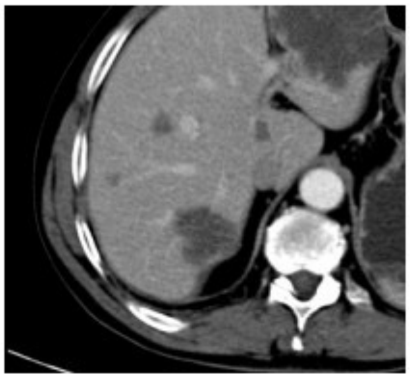
Least squares fit



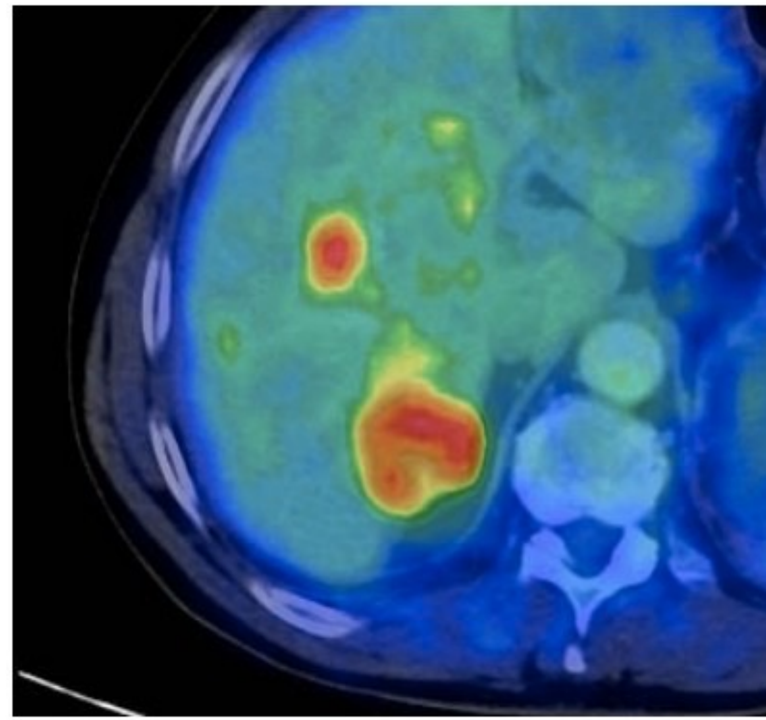
Find “average” translation vector

Fusion of information = registration plus combination in a single representation: PET/CT

CT



PET



Deformable fusion- PET shows increased metabolism in lesions identified on CT, consistent with active tumour growth rather than necrosis post-radiotherapy

Slide Credits

- Jayaram K. Udupa, MIPG of University of Pennsylvania, PA.
- P. Suetens, Fundamentals of Medical Imaging, Cambridge Univ. Press.
- N. Bryan, Intro. to the science of medical imaging, Cambridge Univ. Press.
- CAP 5415 Computer Vision (Fall 2016) Lecture Presentations
- Computer Vision (Lecture Presentations) by Dr. Mohsen Ali