

SSY098 - Image Analysis

Lecture 8 - Image Registration

*Torsten Sattler
(slides adapted from Olof Enqvist)*

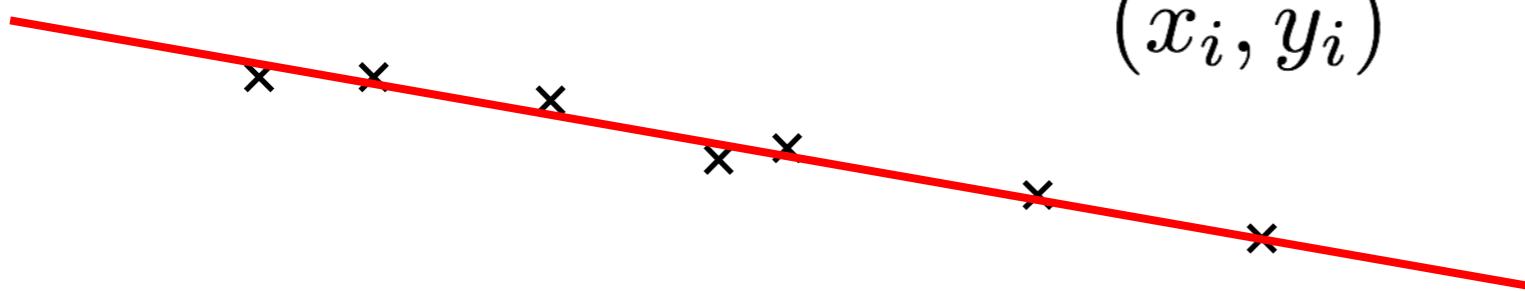
Last Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	
Feb. 3	Convolutional neural networks	Lab 2
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	
Feb. 13	Image registration	Lab 3
Feb. 17	Camera Geometry	
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	

Last Lecture

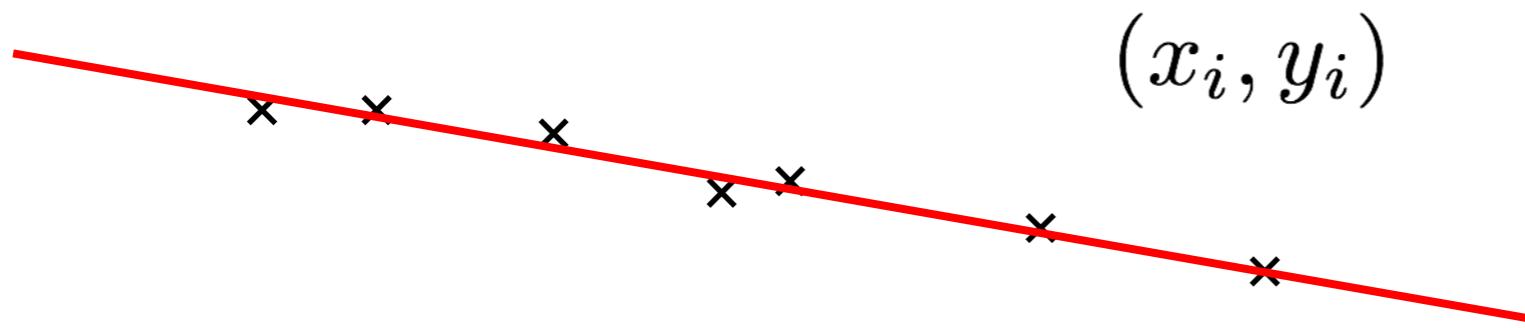
Measurements

$$(x_i, y_i)$$



Last Lecture

Measurements

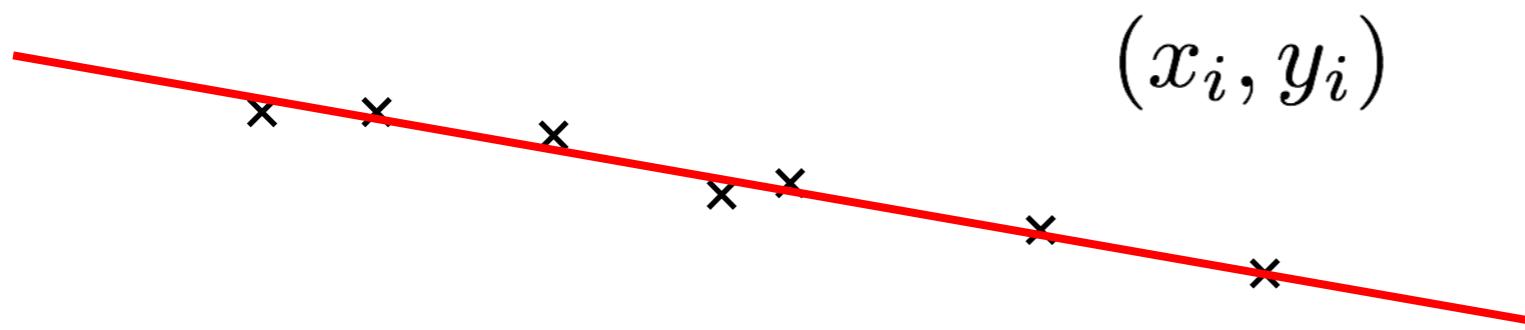


(x_i, y_i)

Good formulation: $ax_i + by_i + c = 0$ (implicit line representation)

Last Lecture

Measurements



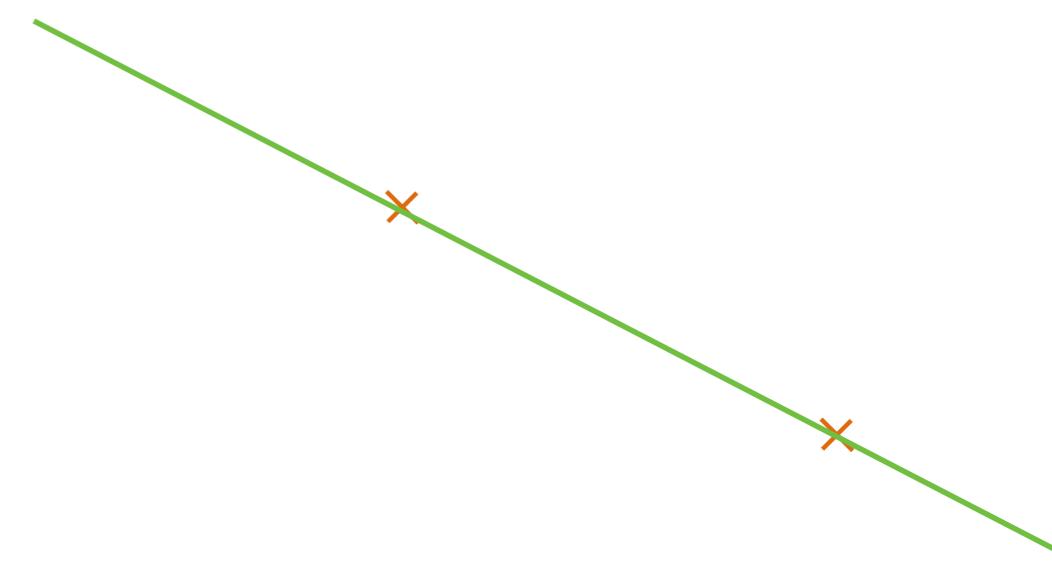
Good formulation: $ax_i + by_i + c = 0$ (implicit line representation)

Model fitting : Estimate a, b and c from measurements

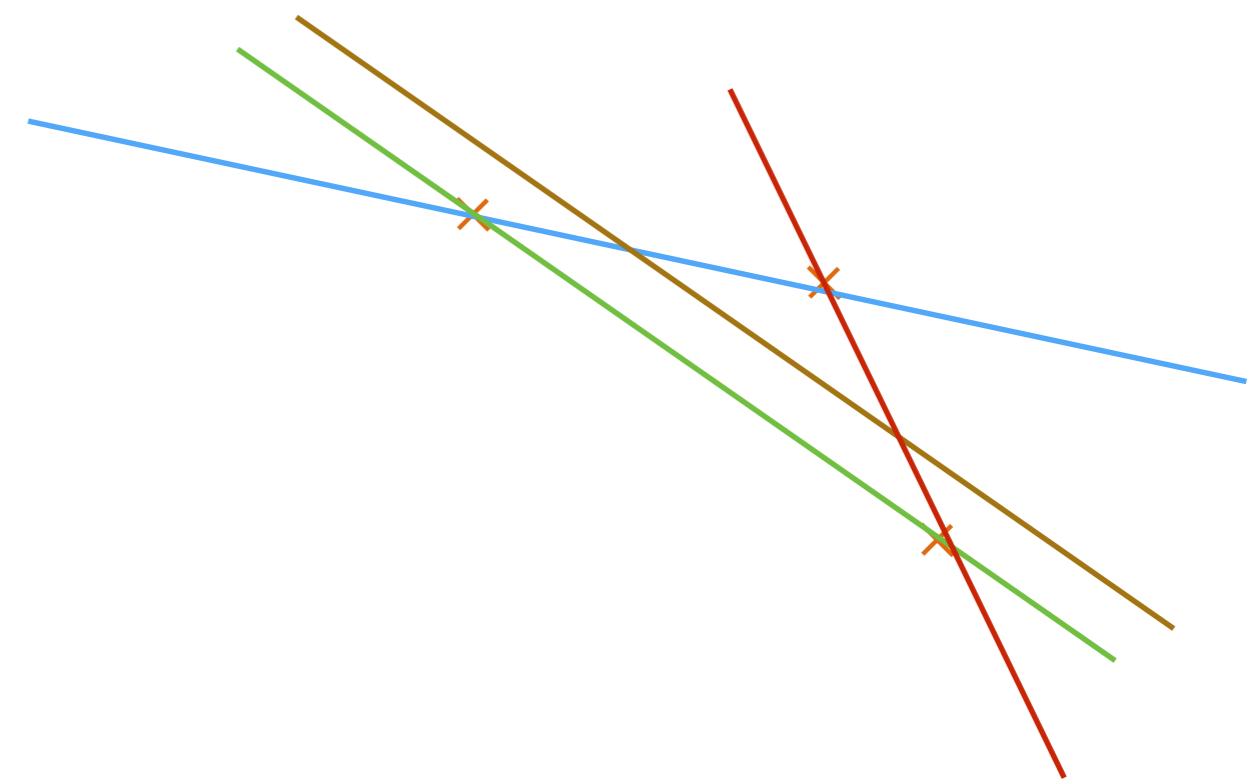
$$\theta = (a, b, c)$$

Last Lecture

Two measurements - exact solution

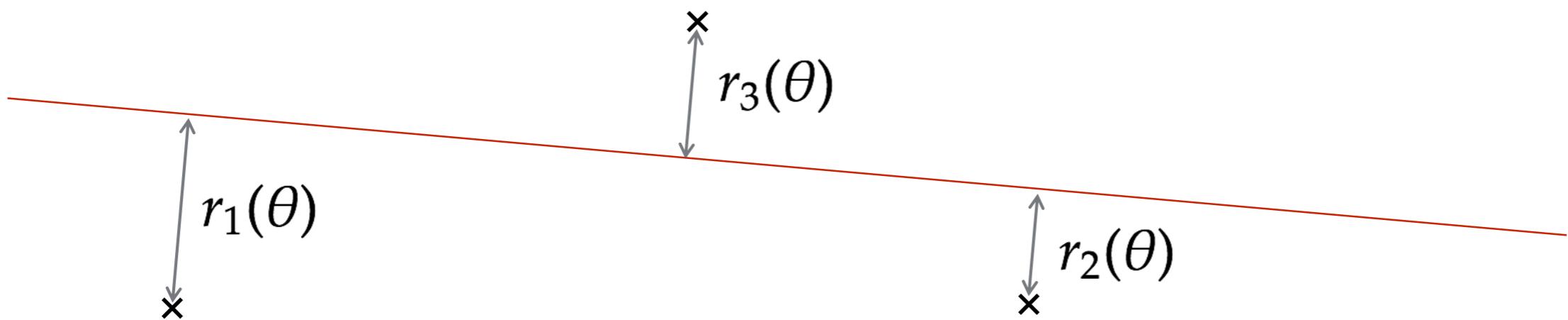


More than two?



Last Lecture

Residuals

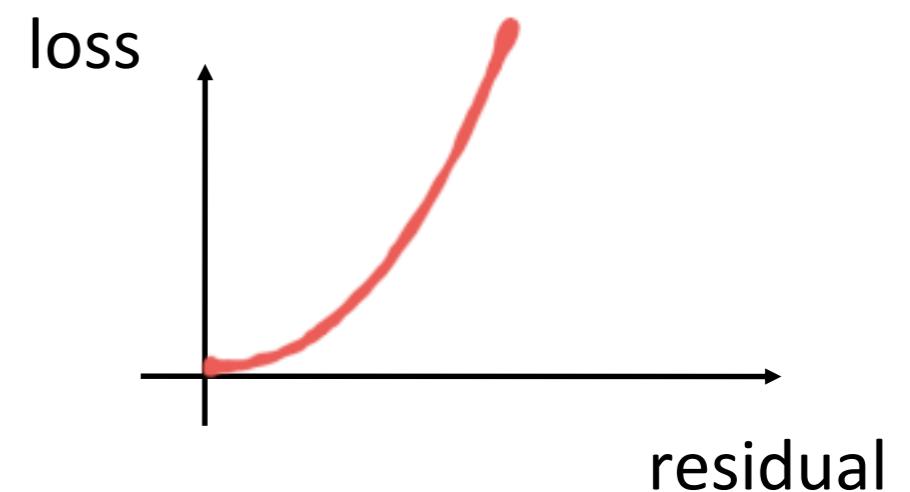


$$r_i(\theta) = |ax + by + c| \text{ such that } a^2 + b^2 = 1$$

Last Lecture



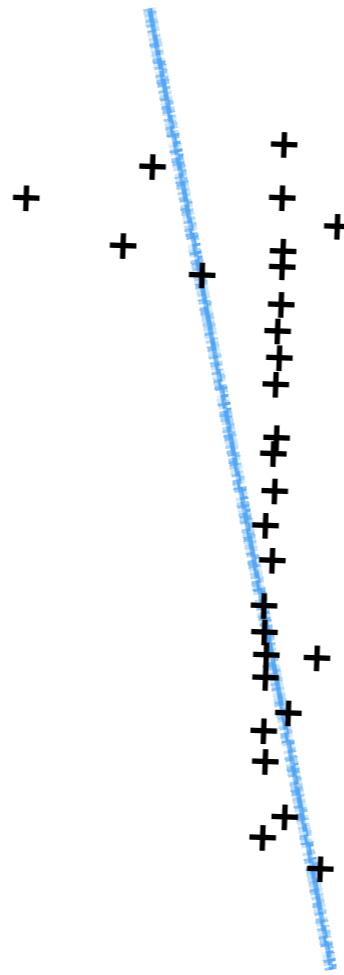
ℓ^2 -solution



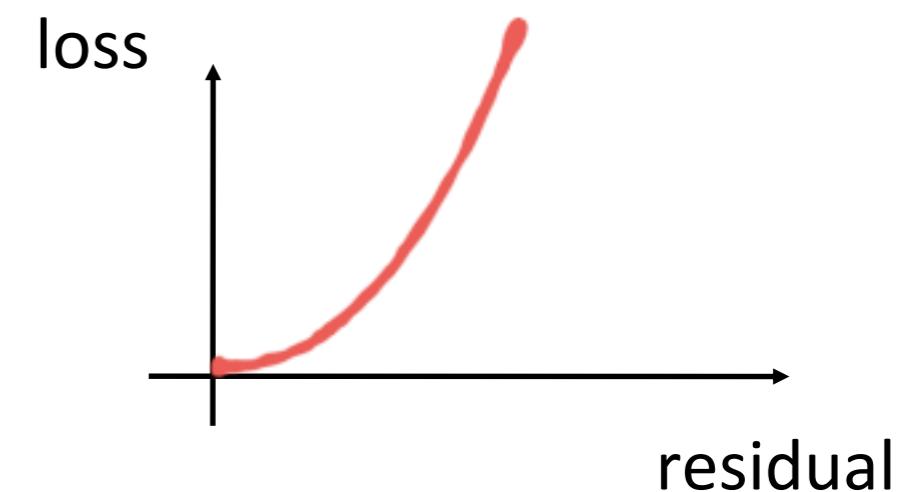
$$\min_{\theta} \sum r_i(\theta)^2$$

Robust Model Fitting

Last Lecture



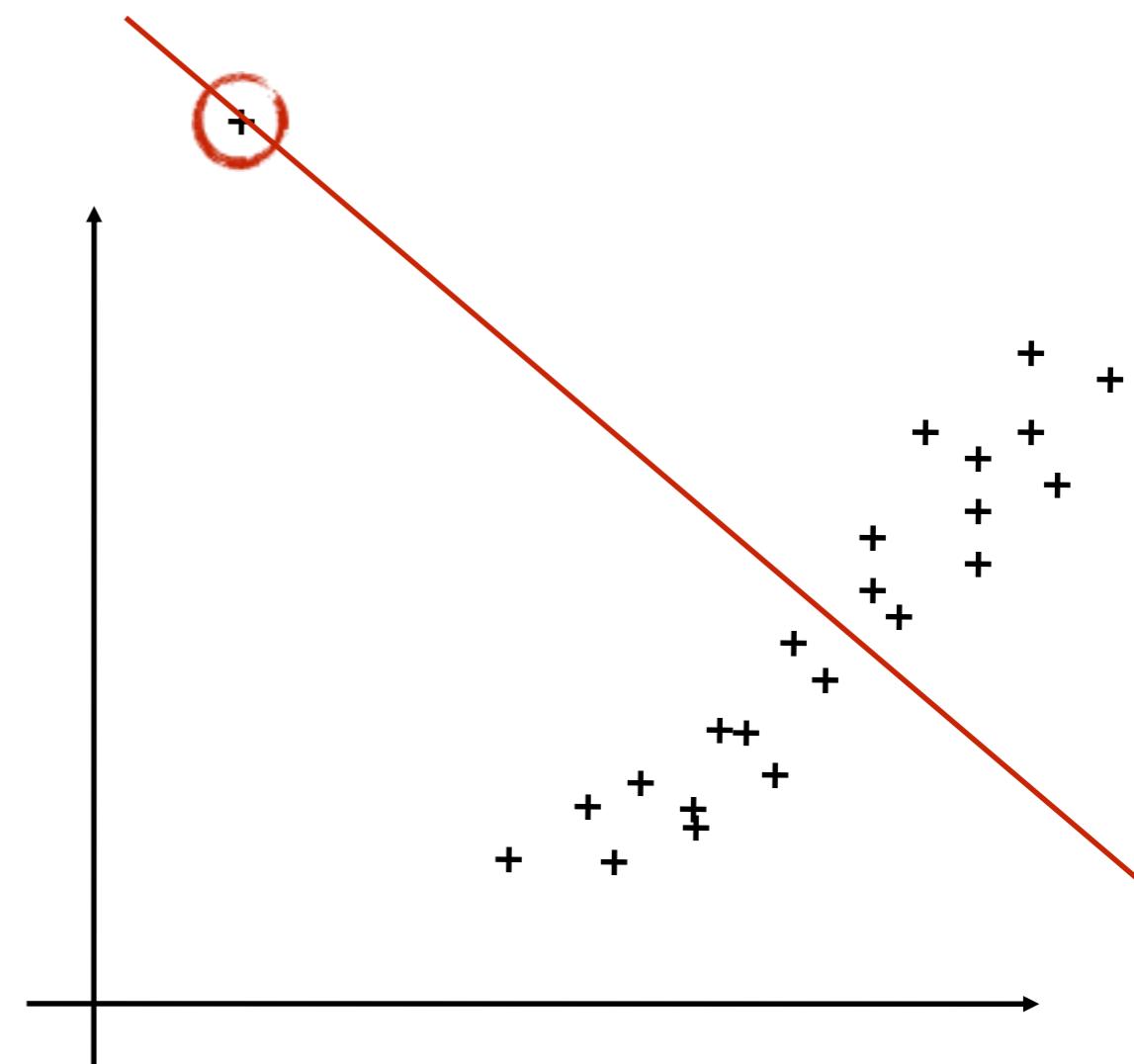
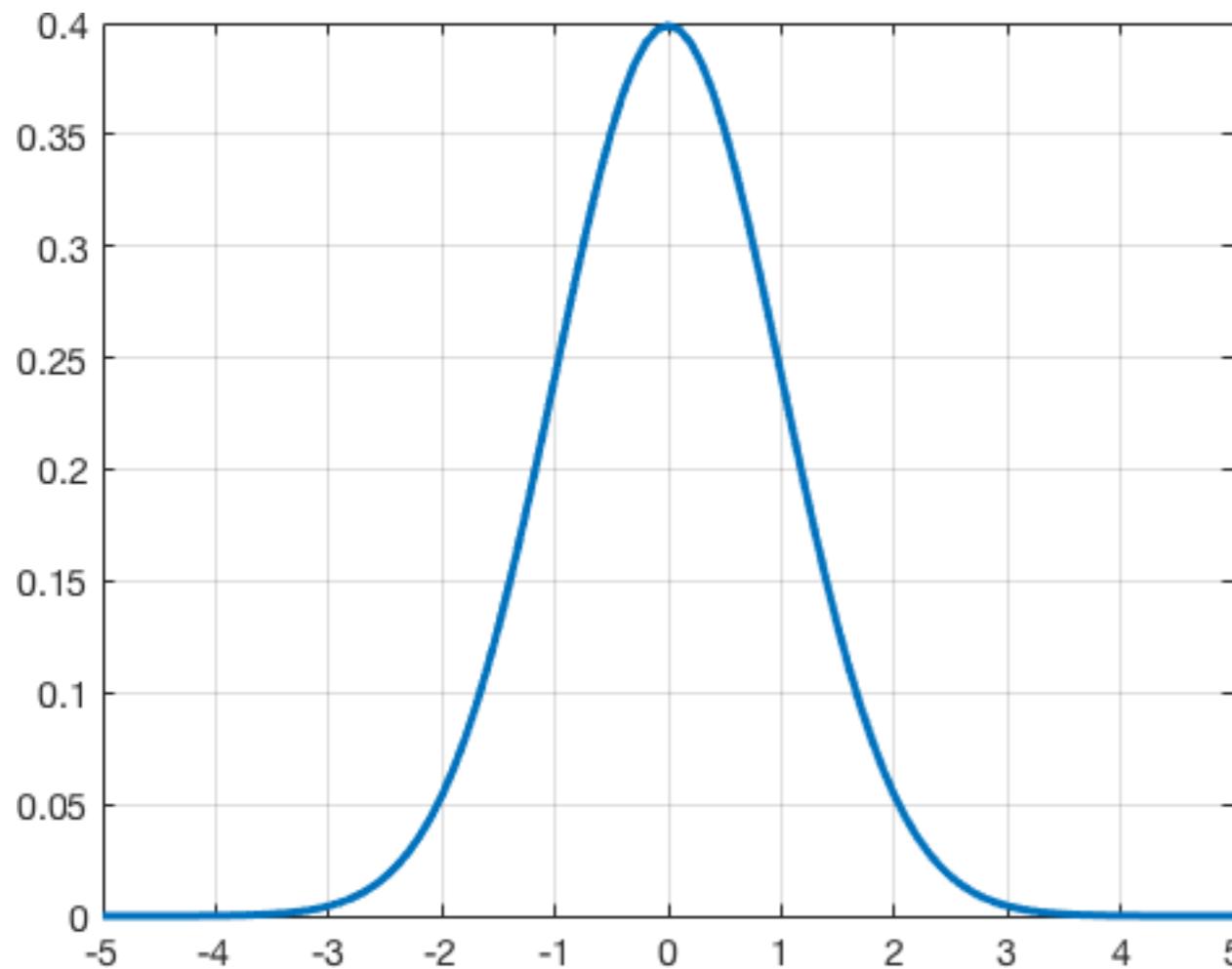
ℓ^2 -solution



$$\min_{\theta} \sum r_i(\theta)^2$$

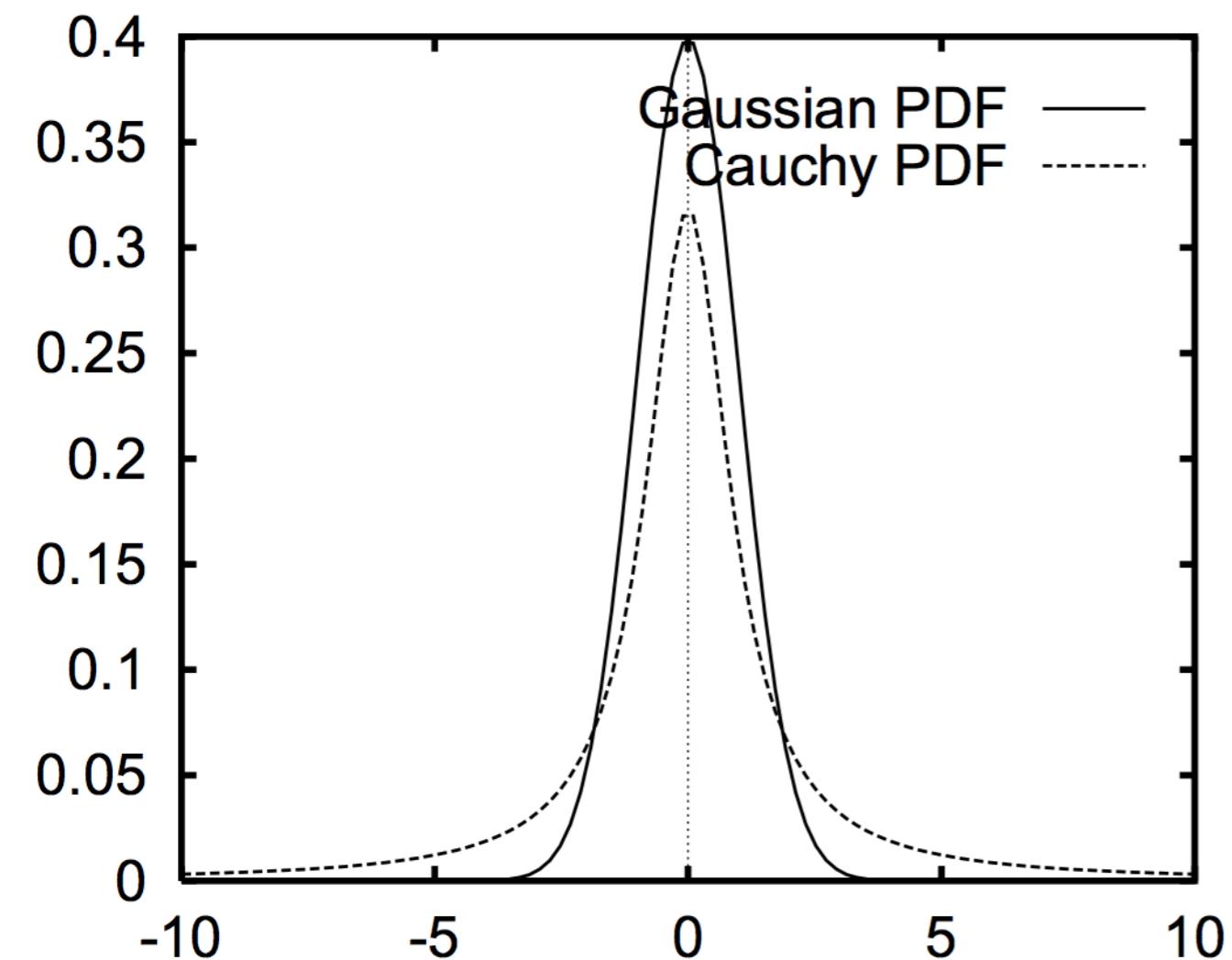
Robust Model Fitting

Last Lecture



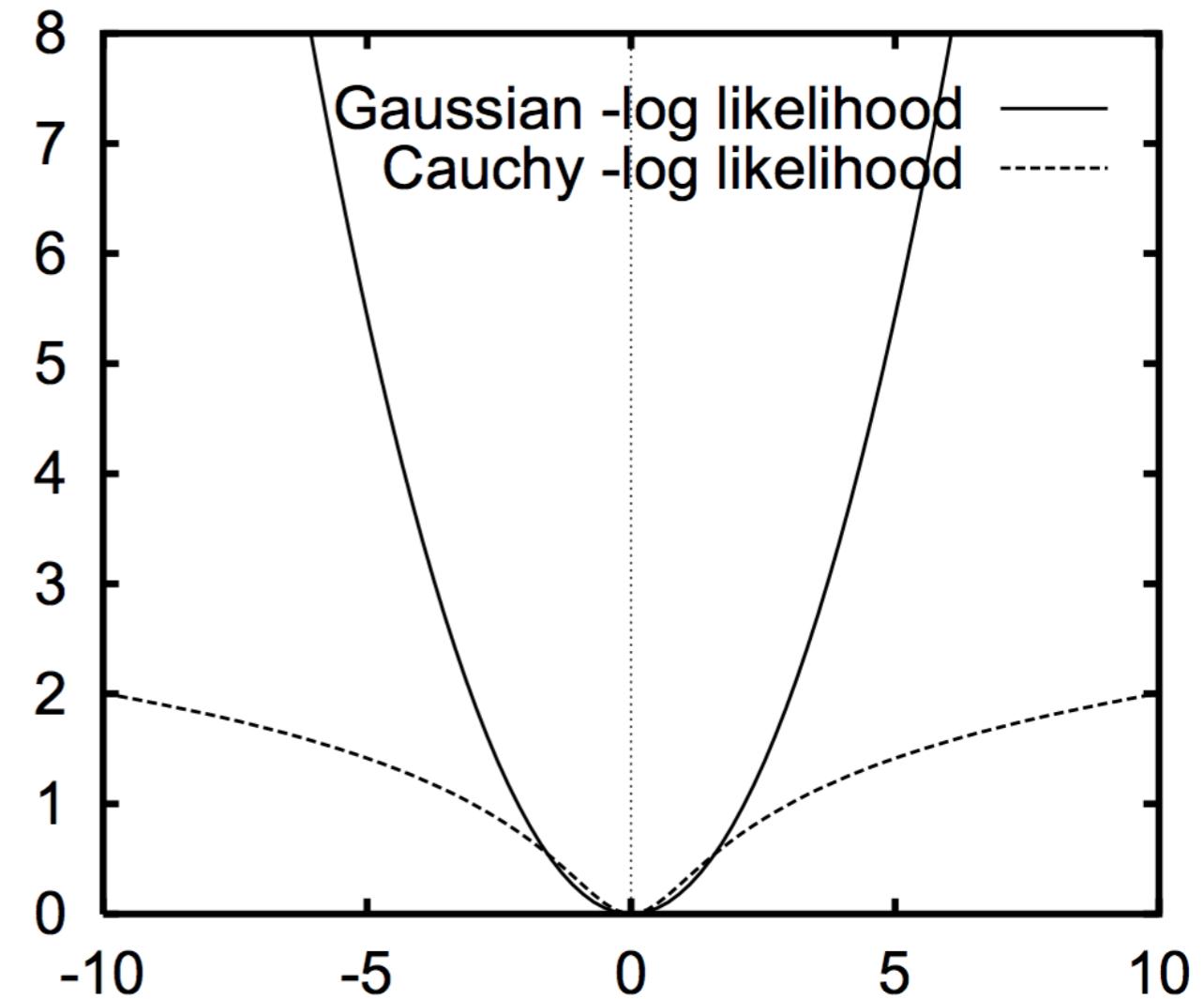
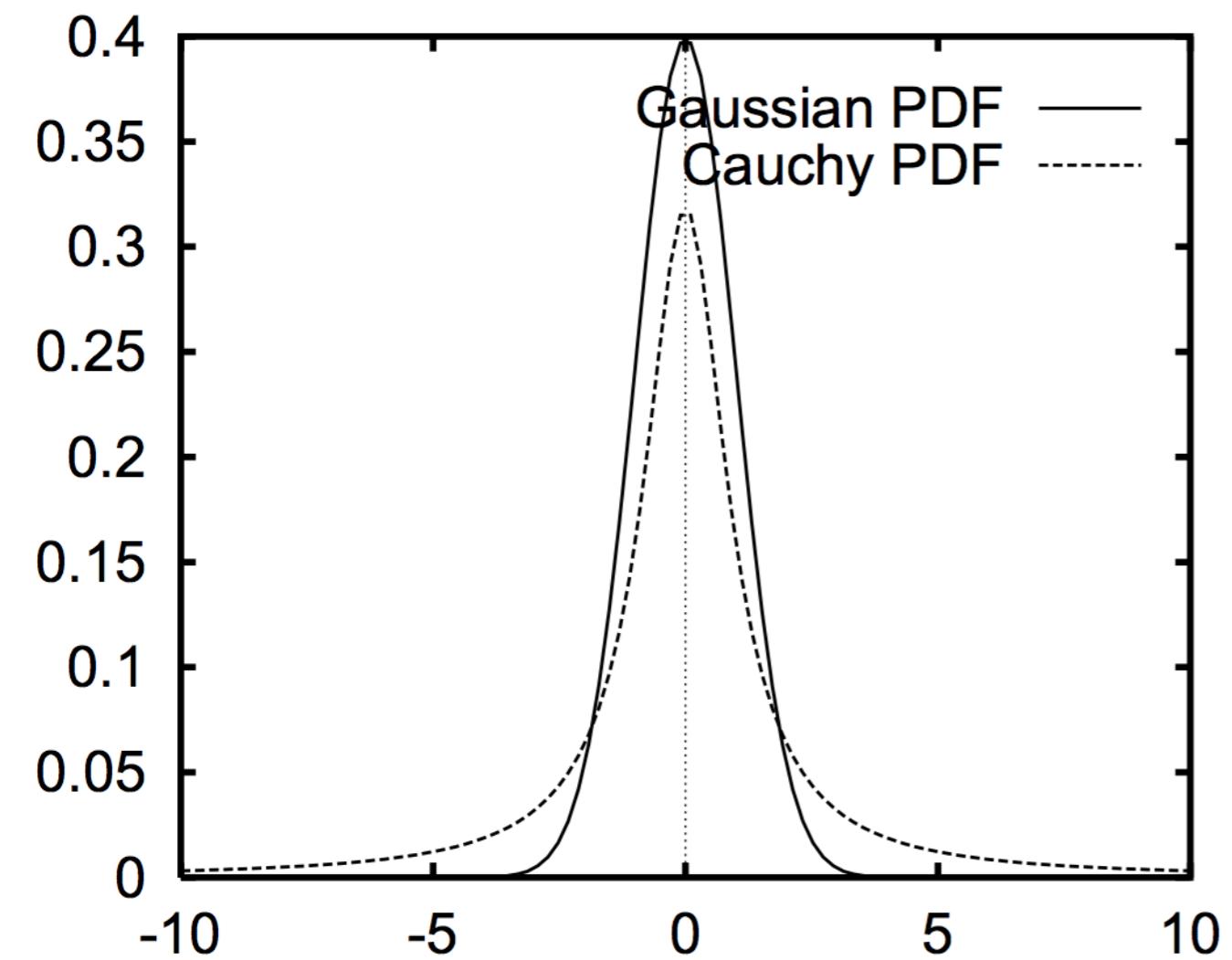
Outliers unlikely under Gaussian noise assumption

Last Lecture



Robust Loss Functions

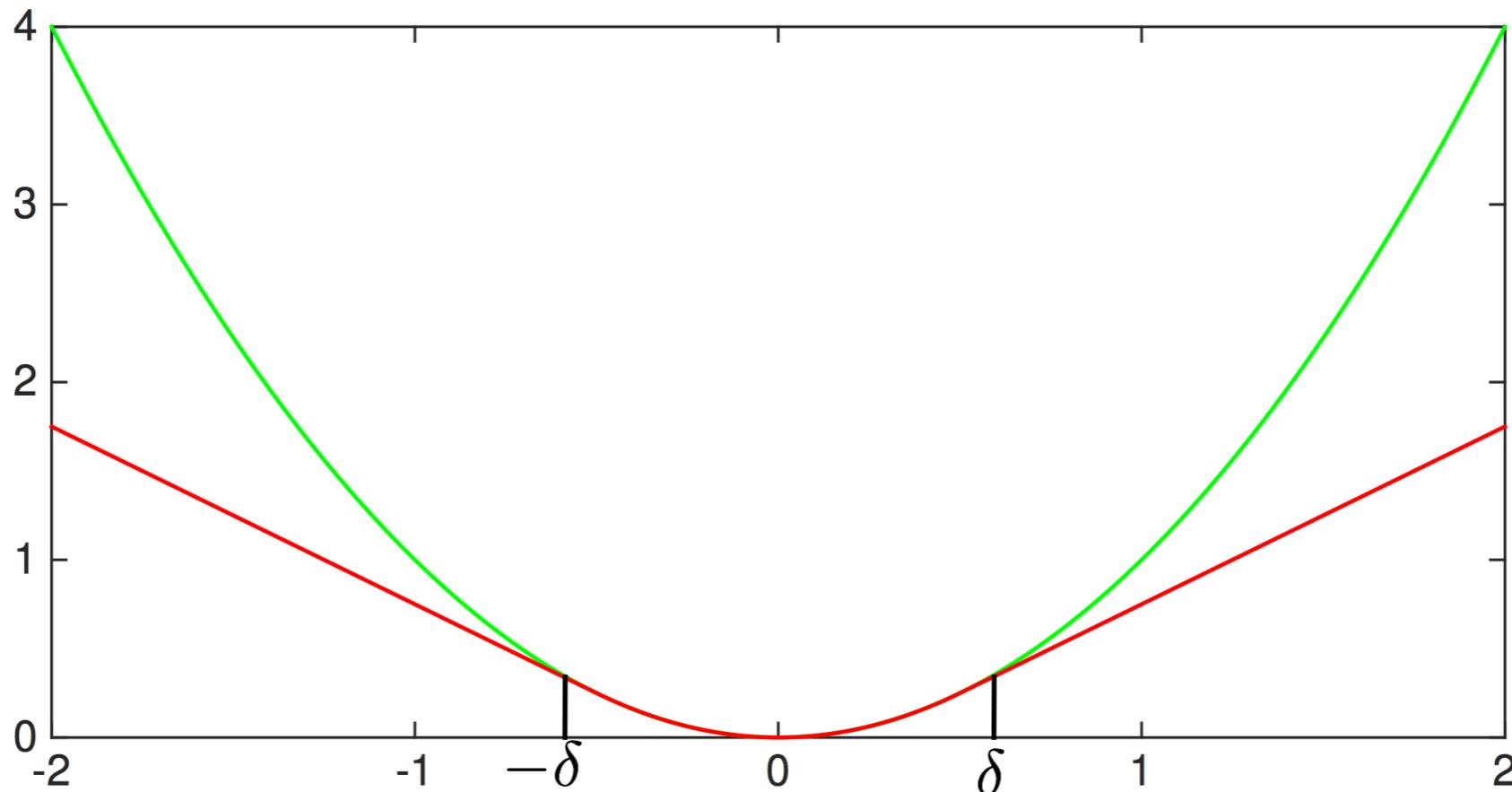
Last Lecture



Robust Loss Functions

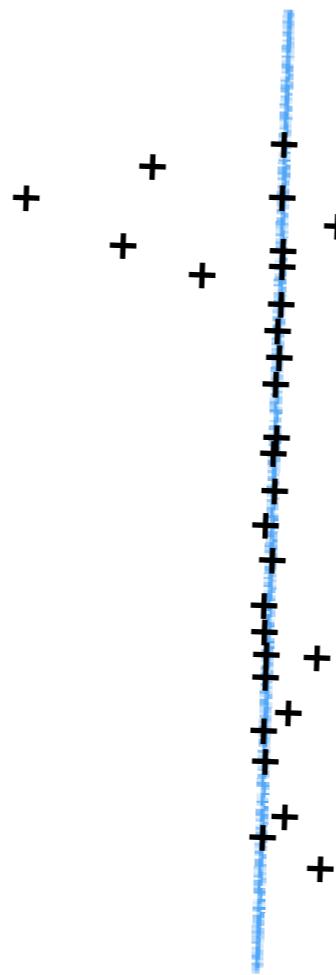
Last Lecture

Robust Loss Functions: Huber Loss

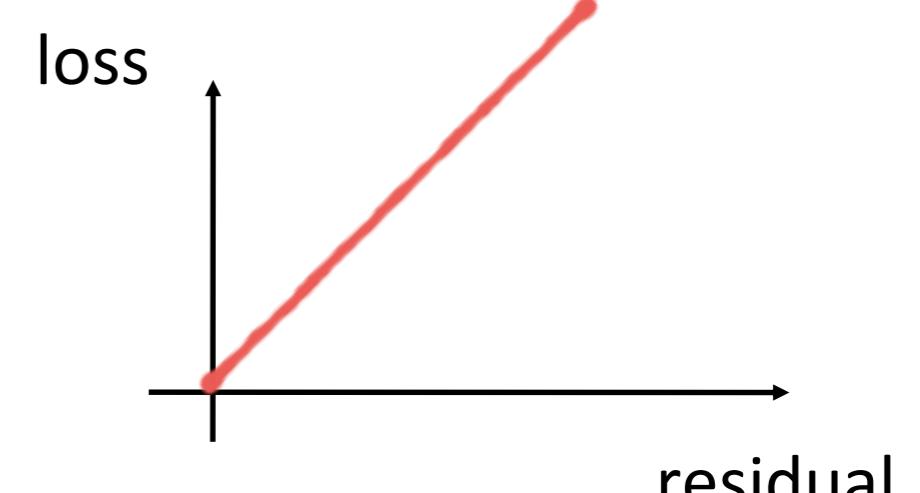


$$\min_{\theta} \sum_i h(r_i(\theta)^2) \text{ with } h(x) = \begin{cases} x & \text{if } x < \delta \\ 2\delta\sqrt{x} - \delta^2 & \text{if } x \geq \delta \end{cases}$$

Last Lecture



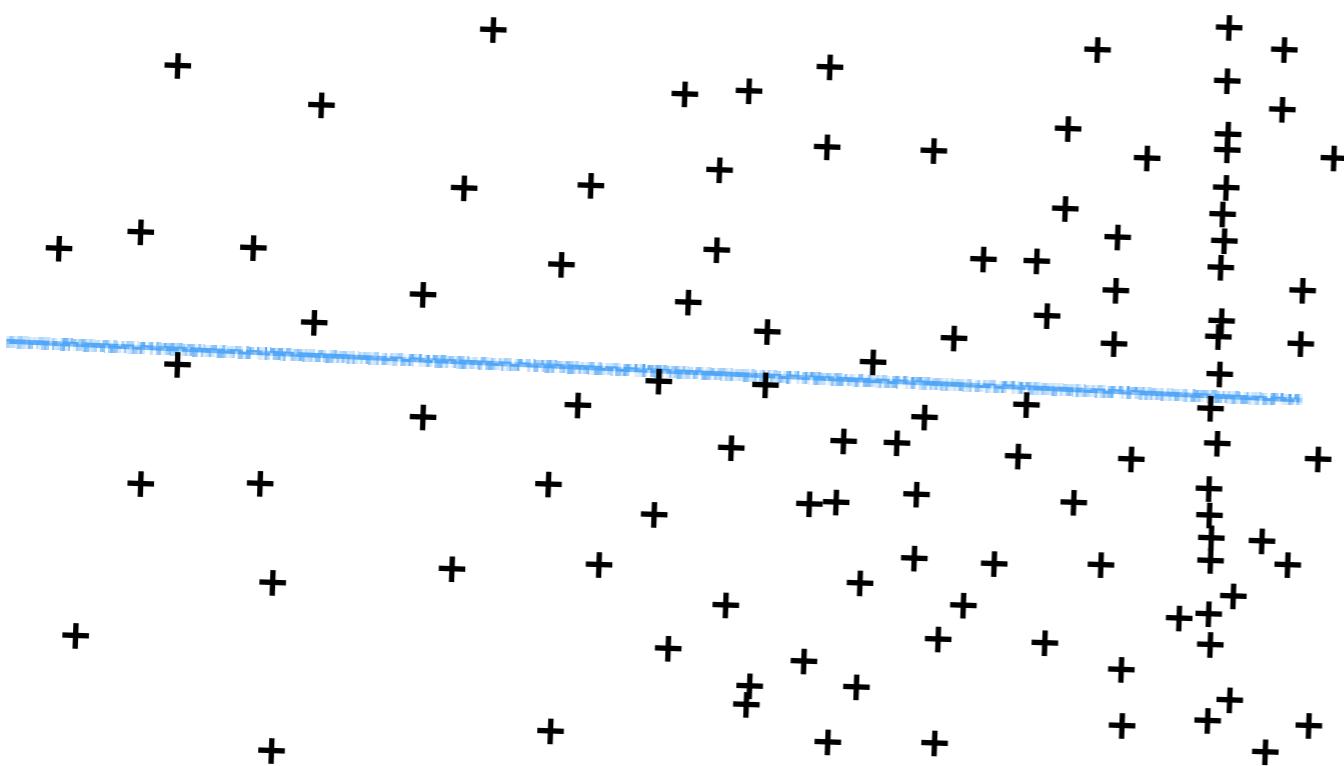
ℓ^1 -solution



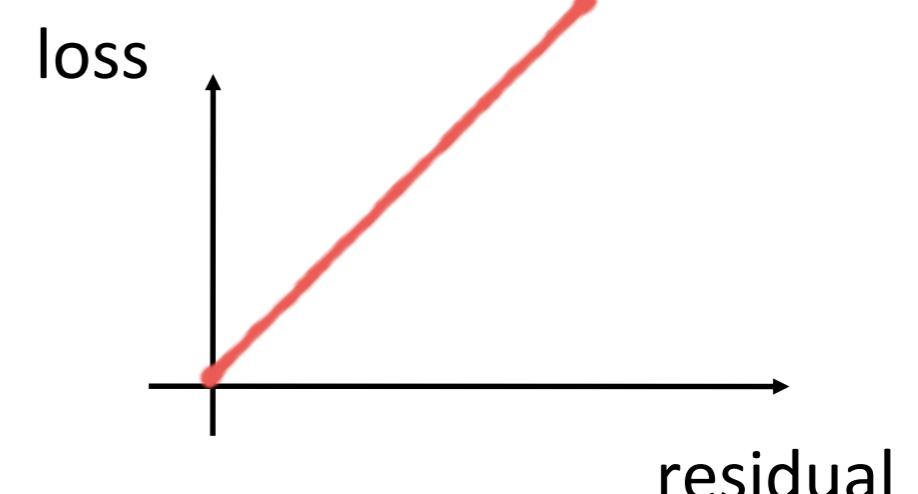
$$\min_{\theta} \sum |r_i(\theta)|$$

Robust Loss Functions

Last Lecture



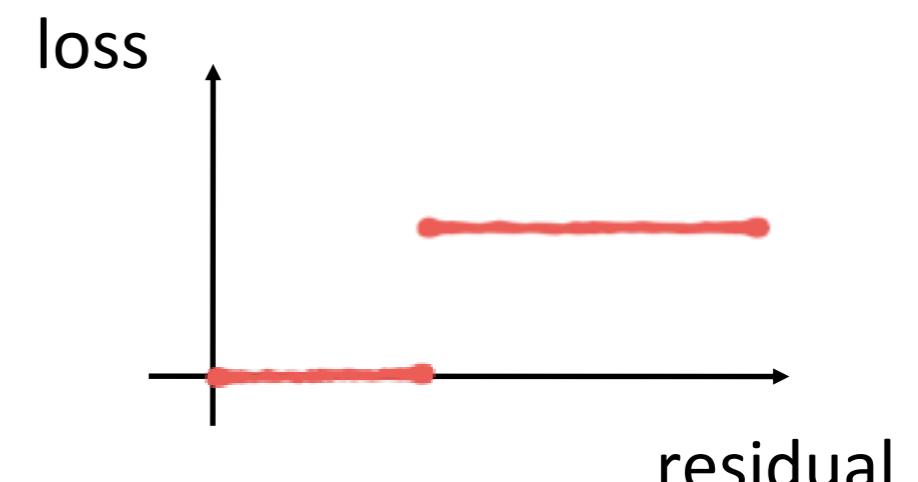
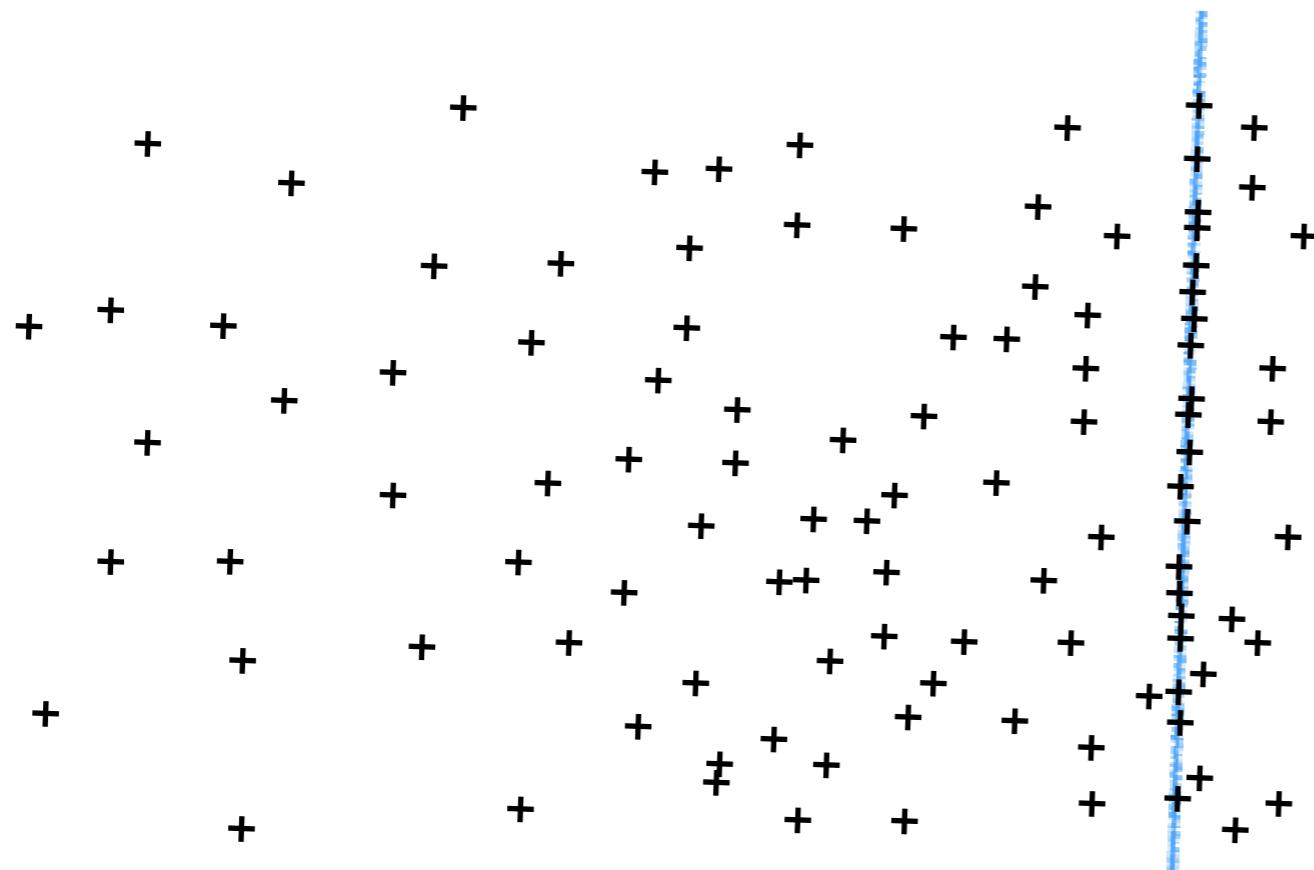
ℓ^1 -solution



$$\min_{\theta} \sum |r_i(\theta)|$$

Robust Loss Functions

Last Lecture



Minimizing #outliers / maximizing #inliers

Last Lecture

While probability of missing correct model $> \eta$

 Estimate model from n random data points

 Estimate support (= #**inliers**) of model

 If more inliers than previous best model

 update best model

Return: Model with most inliers

RANSAC

Last Lecture

Line fitting example

x

x

x

x

x

x

RANSAC

Last Lecture

Line fitting example

x

x

x

x

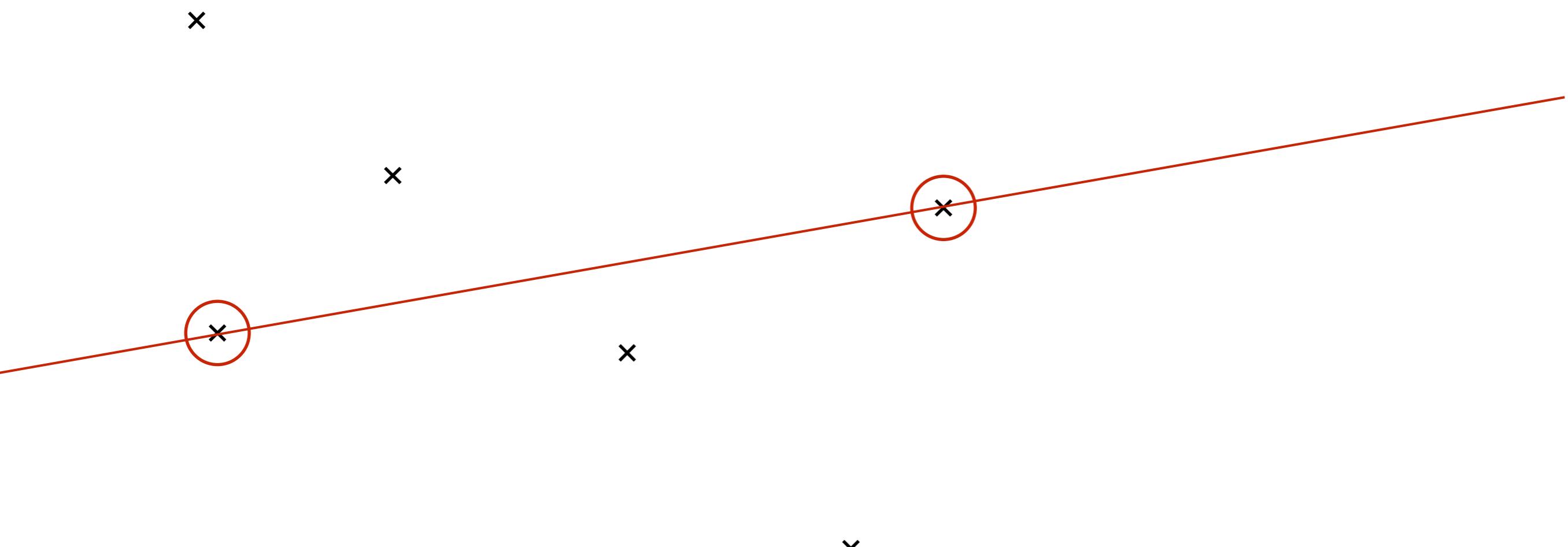
x

x

RANSAC

Last Lecture

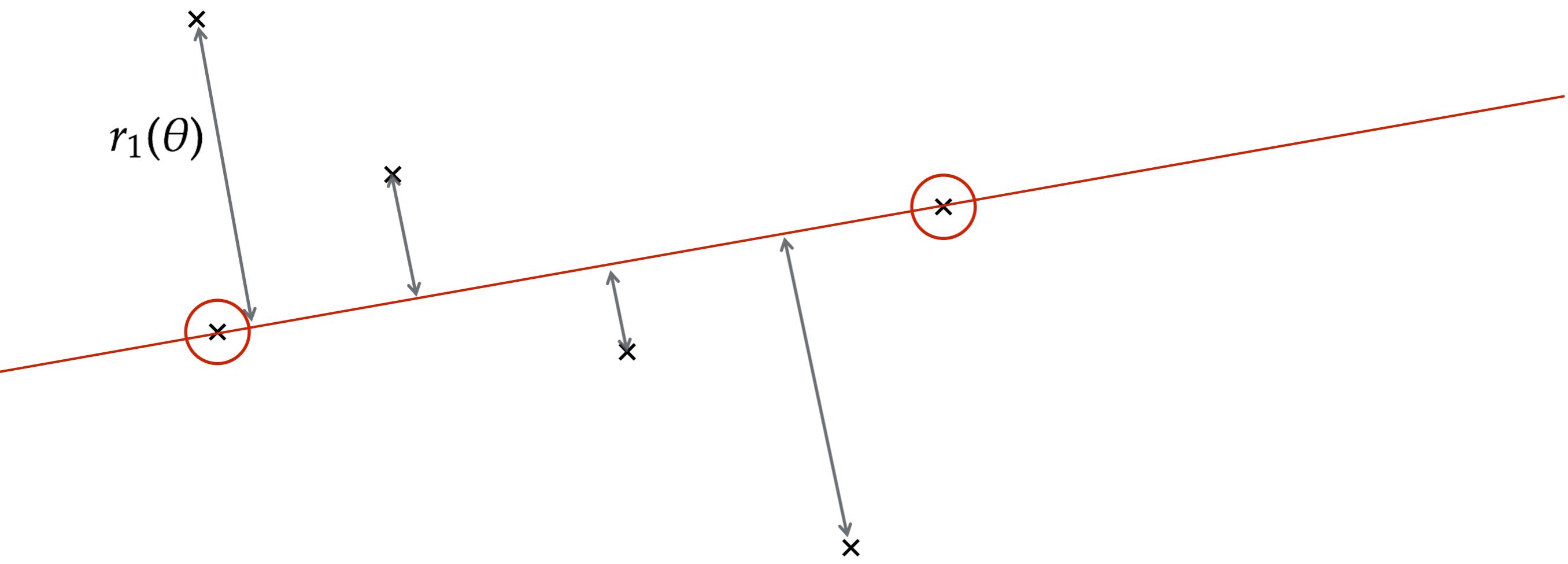
Line fitting example



RANSAC

Last Lecture

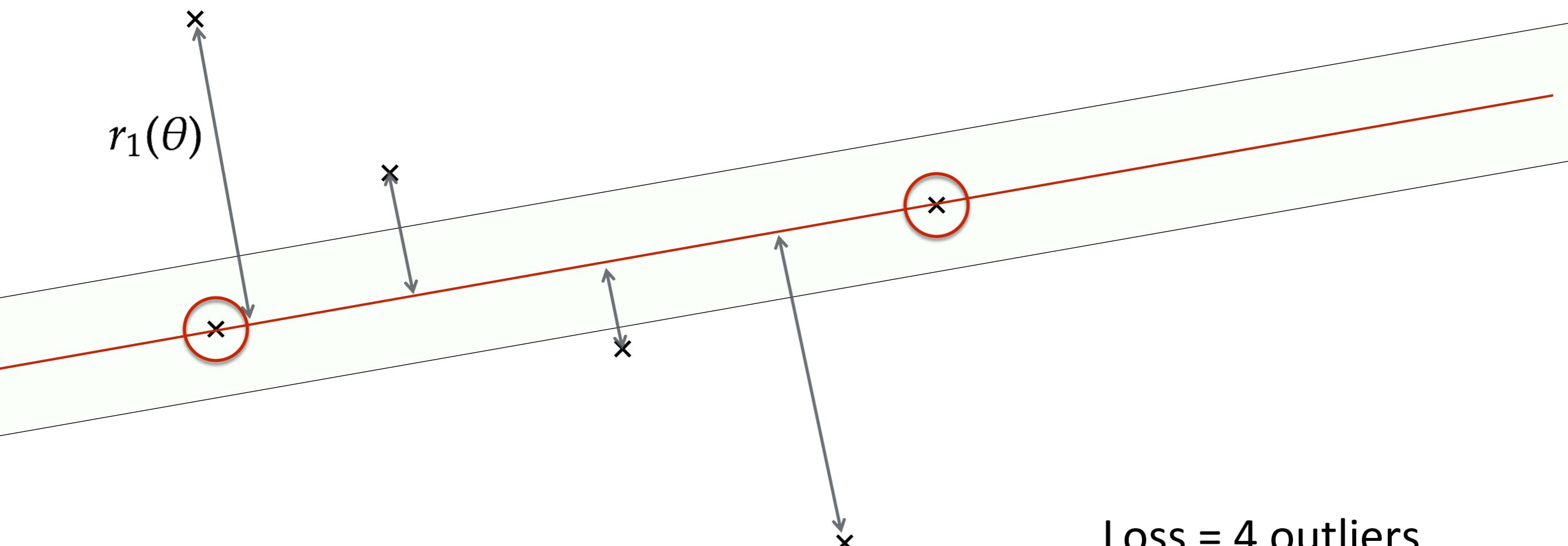
Line fitting example



RANSAC

Last Lecture

Line fitting example



RANSAC

Loss = 4 outliers

Last Lecture

Line fitting example

Best loss so far = 4 outliers



x

x

x



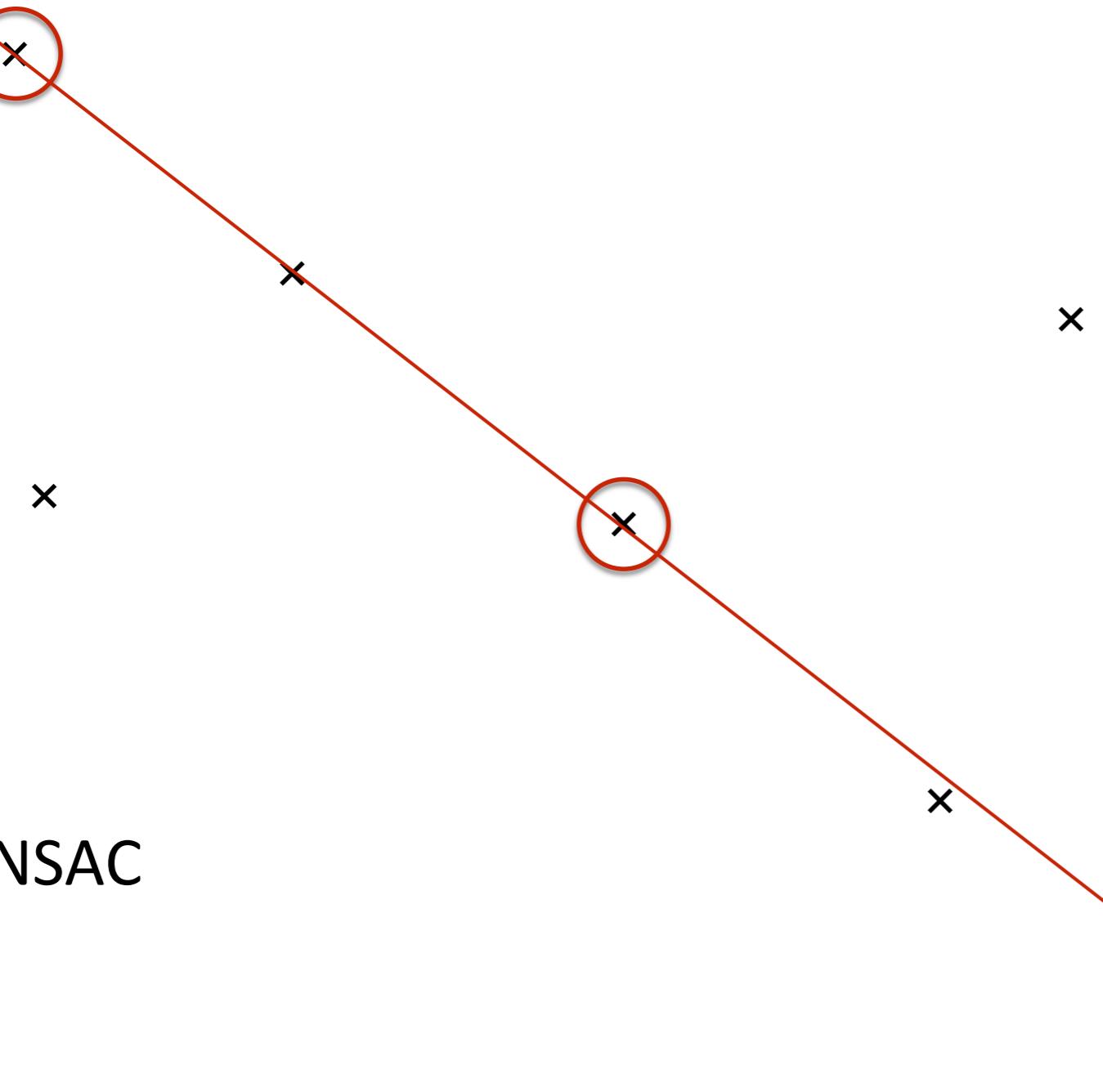
x

RANSAC

Last Lecture

Line fitting example

Best loss so far = 4 outliers

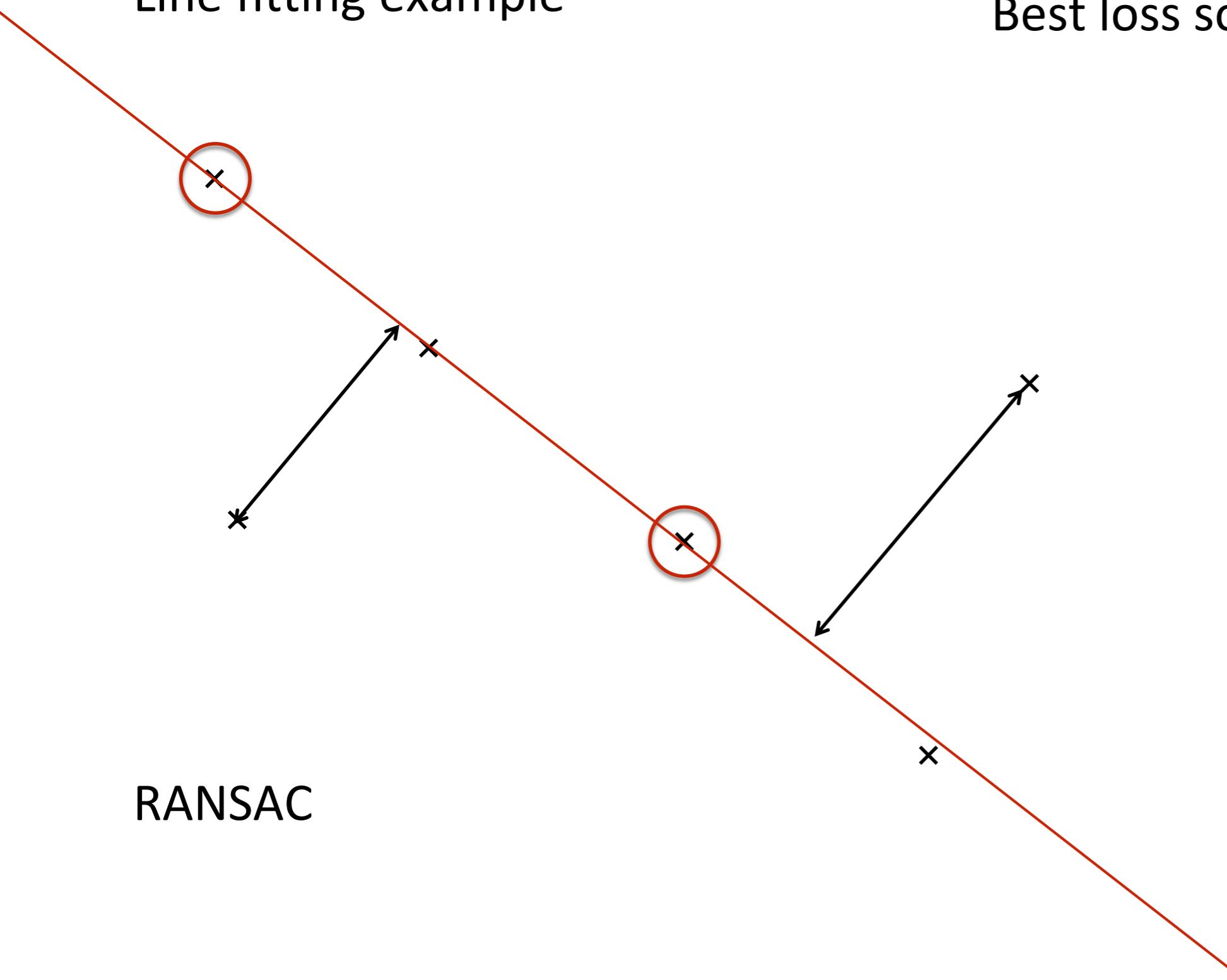


RANSAC

Last Lecture

Line fitting example

Best loss so far = 4 outliers



RANSAC

Last Lecture

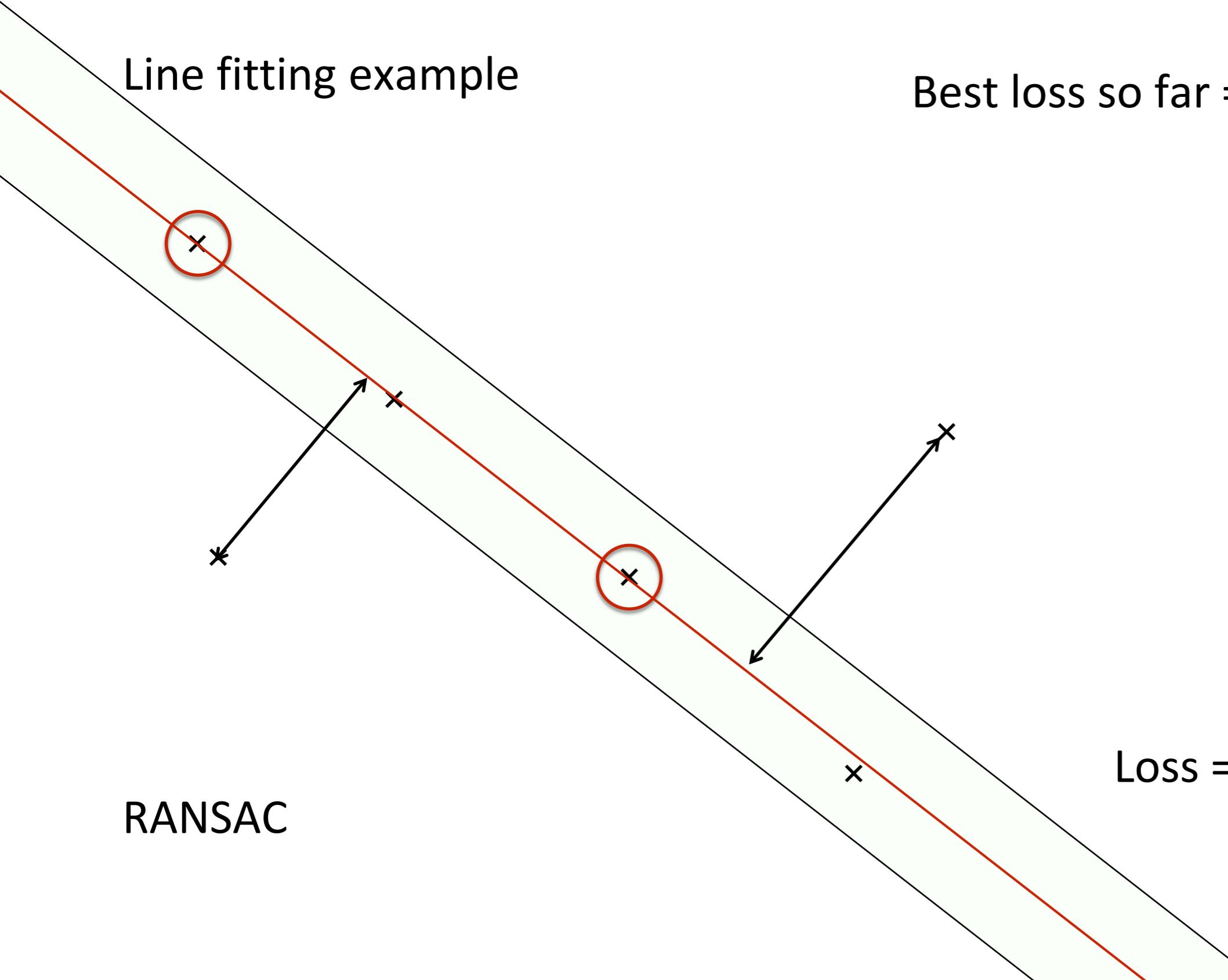
Line fitting example

Best loss so far = 4 outliers

RANSAC

x

Loss = 2 outliers



Last Lecture

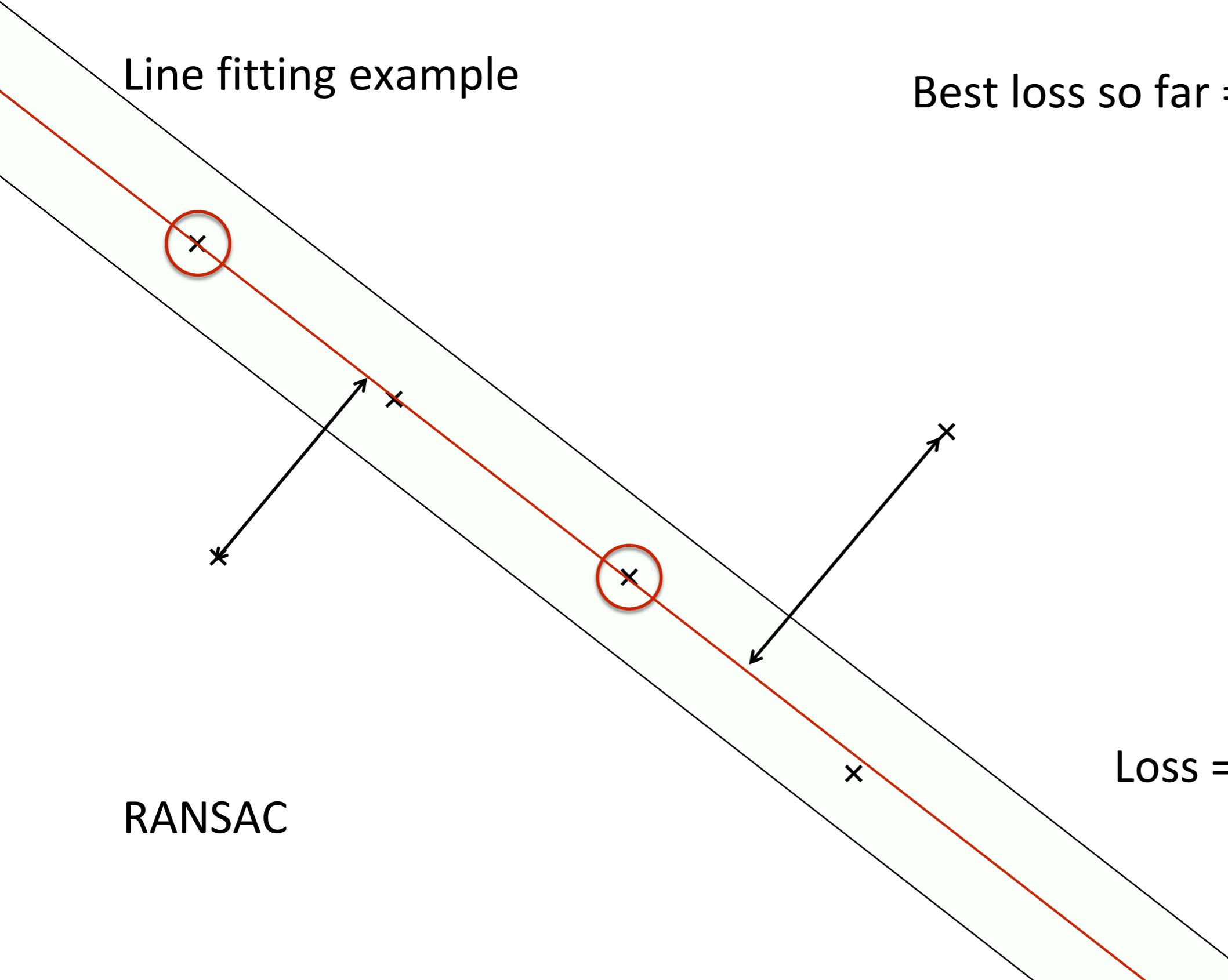
Line fitting example

Best loss so far = 2 outliers

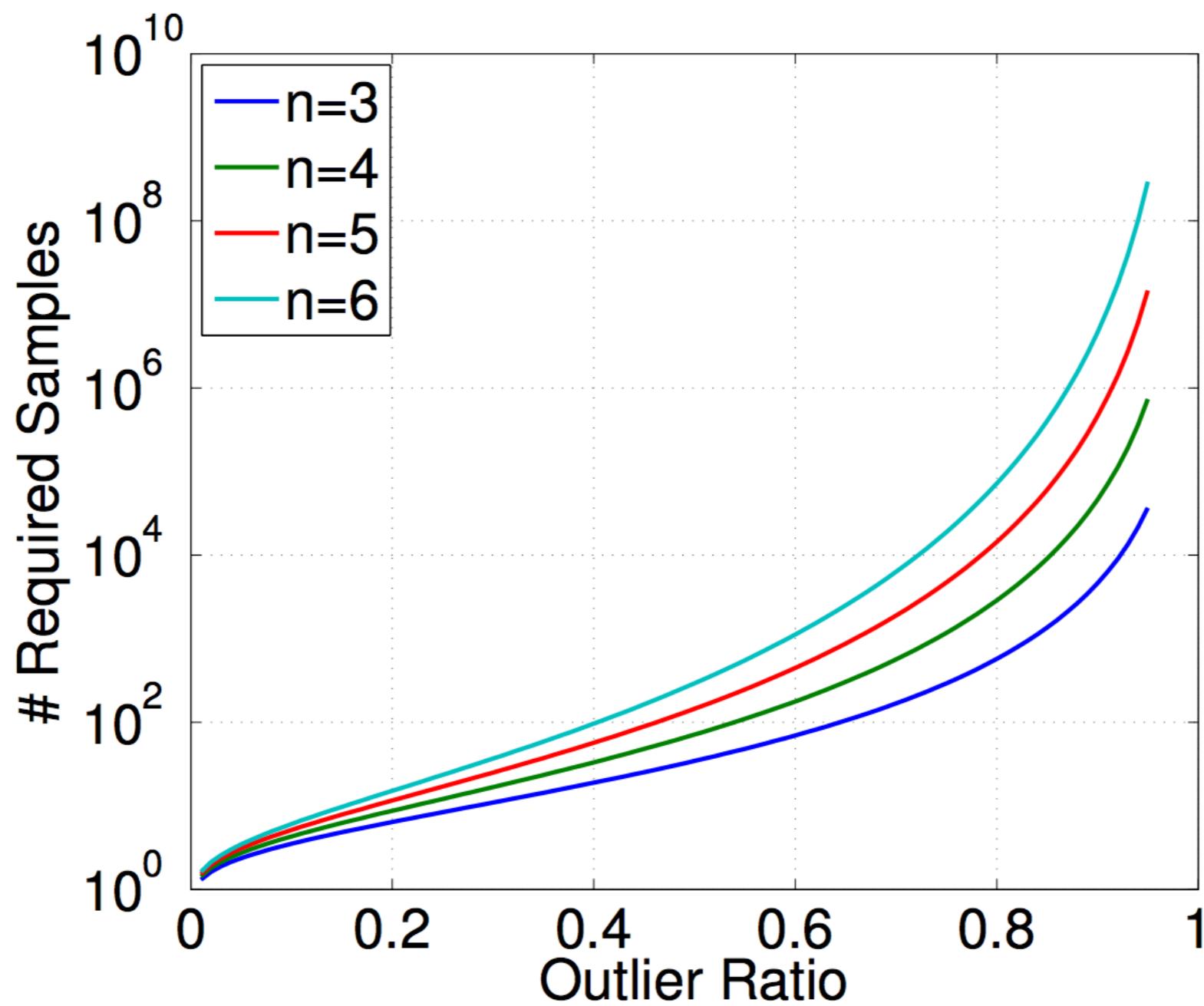
RANSAC

x

Loss = 2 outliers



Last Lecture



Minimal solvers

Today

Image Registration

SIFT Features

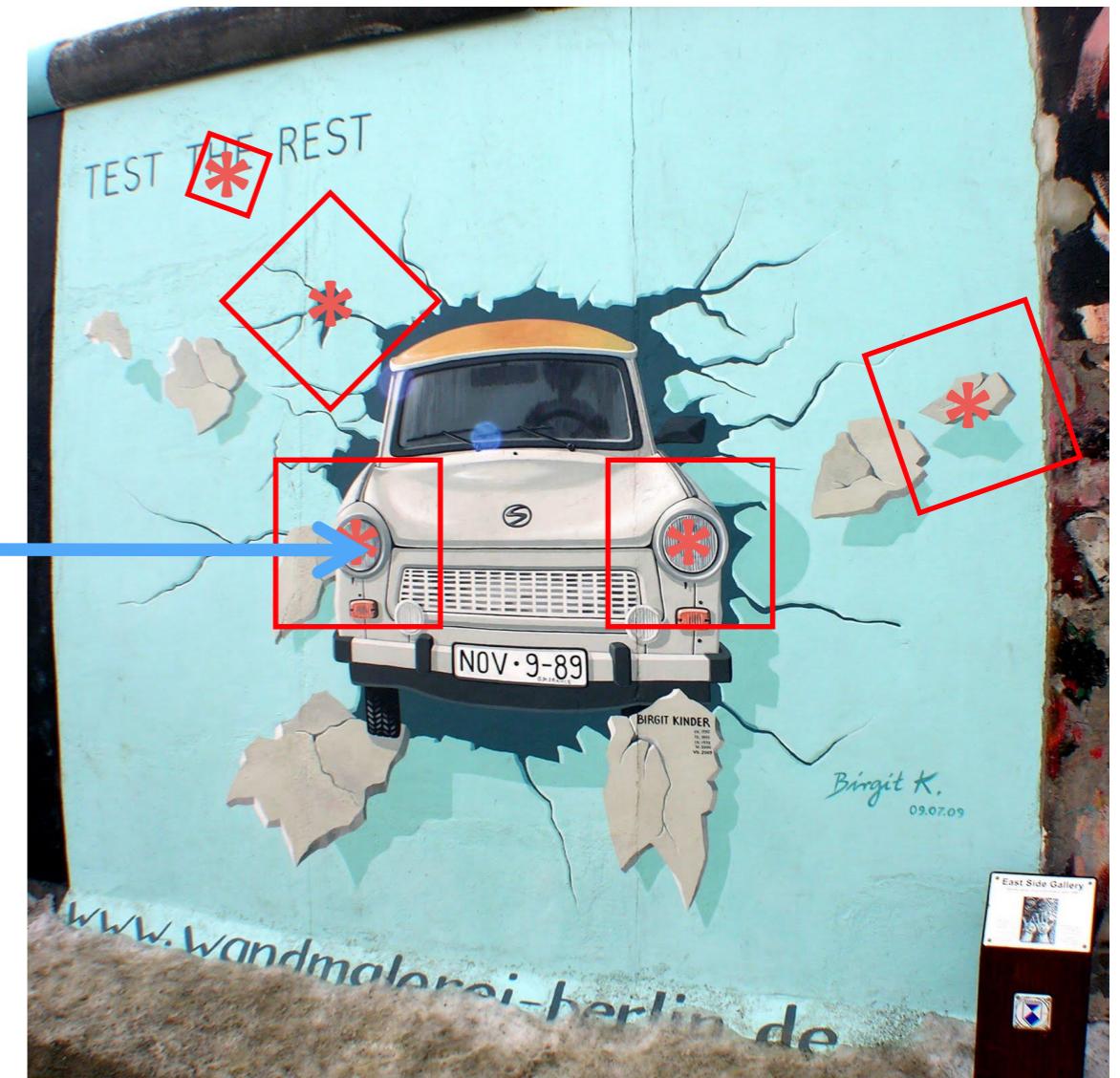


Image Stitching

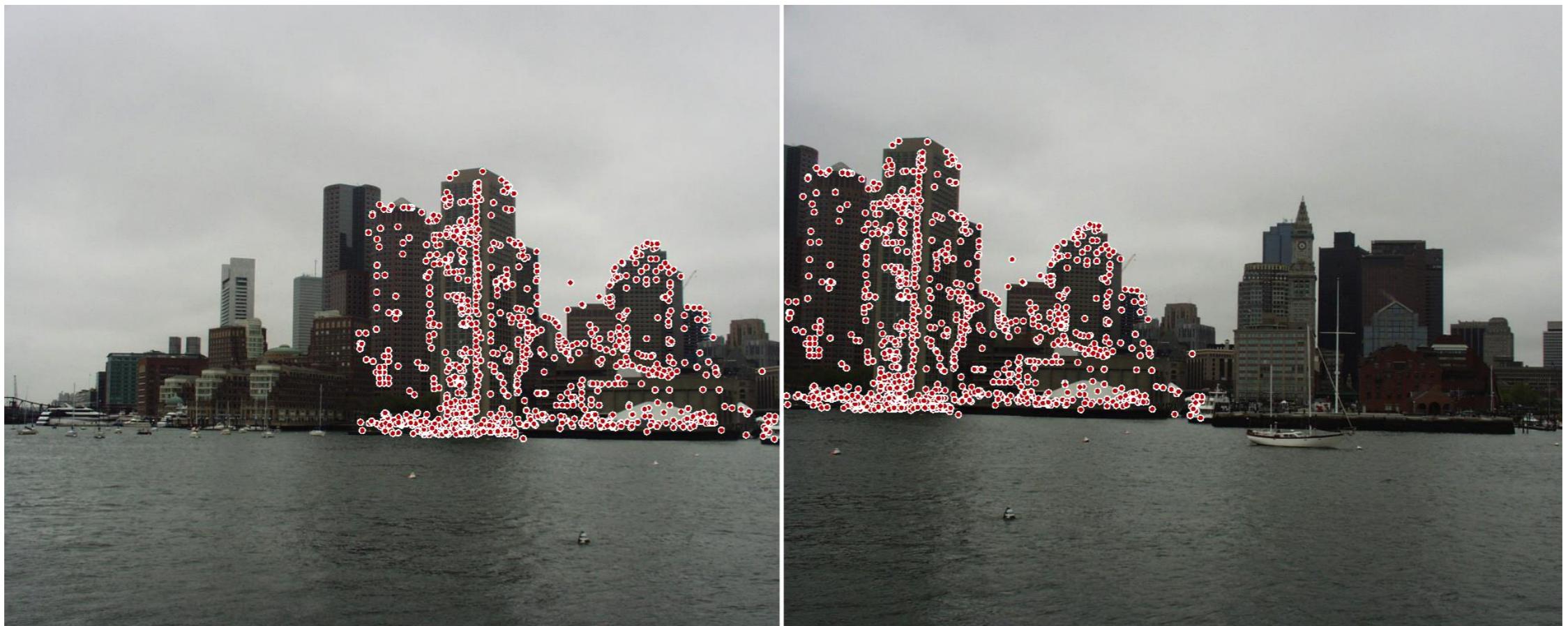


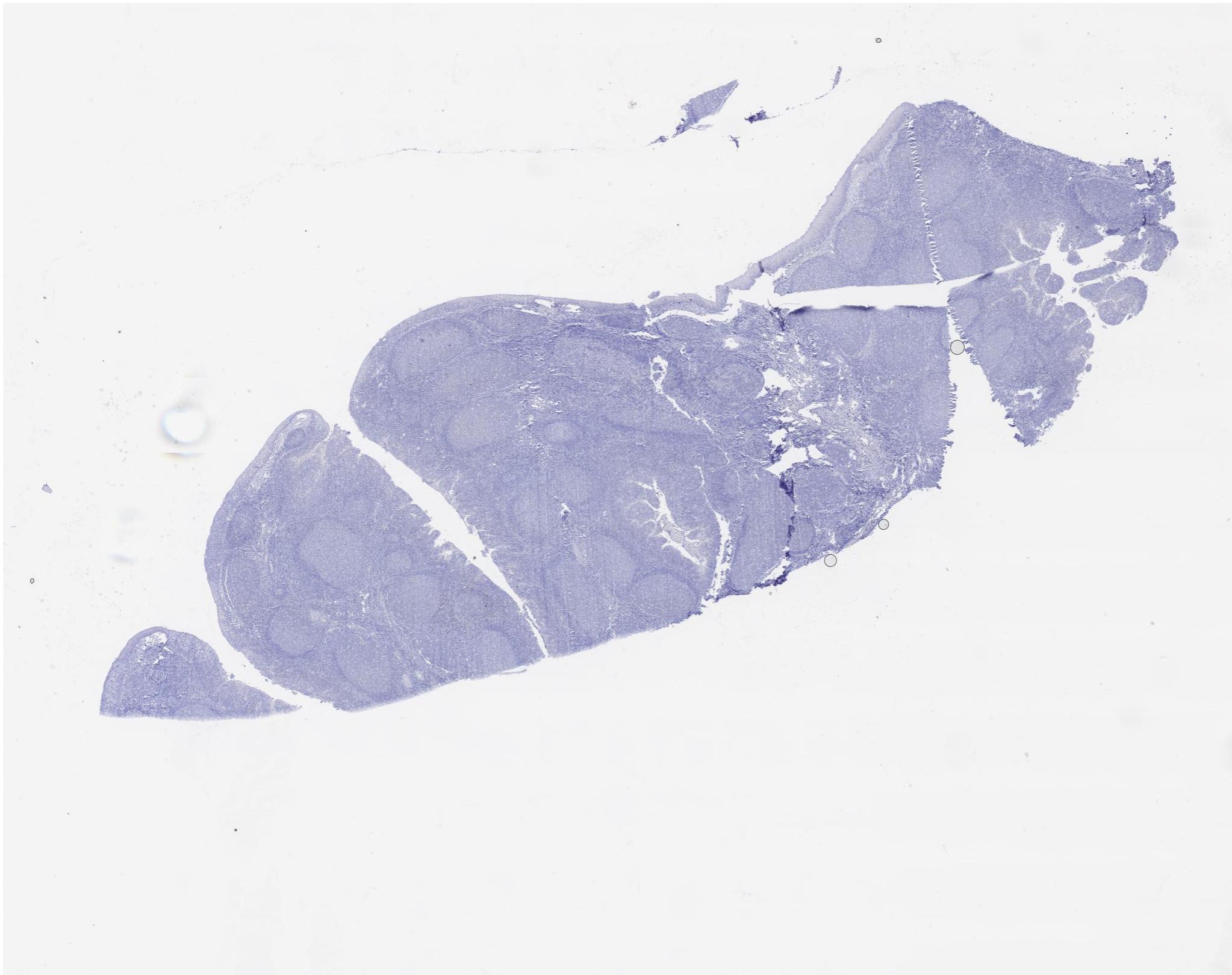
Image Stitching



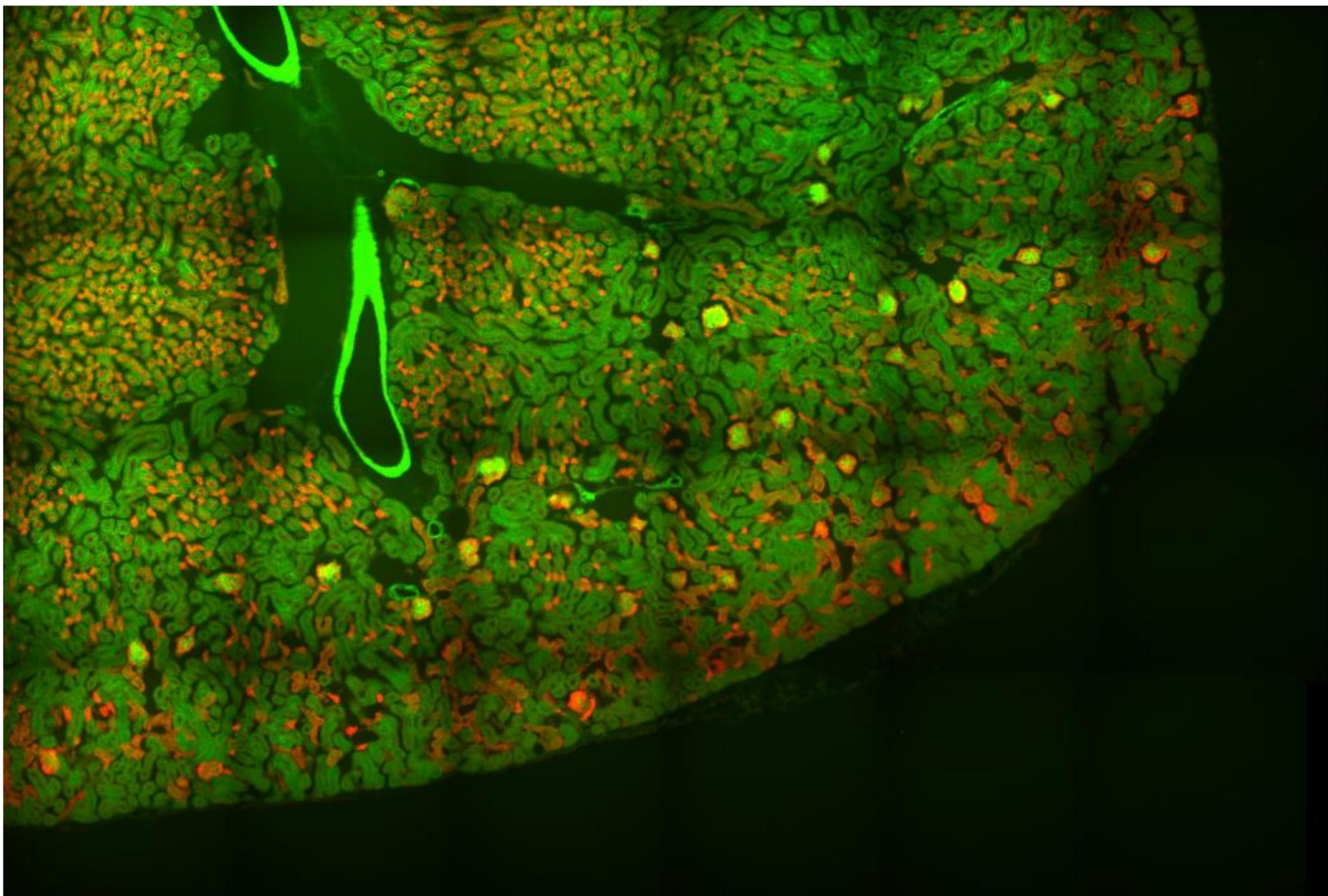
Tissue Sample Alignment



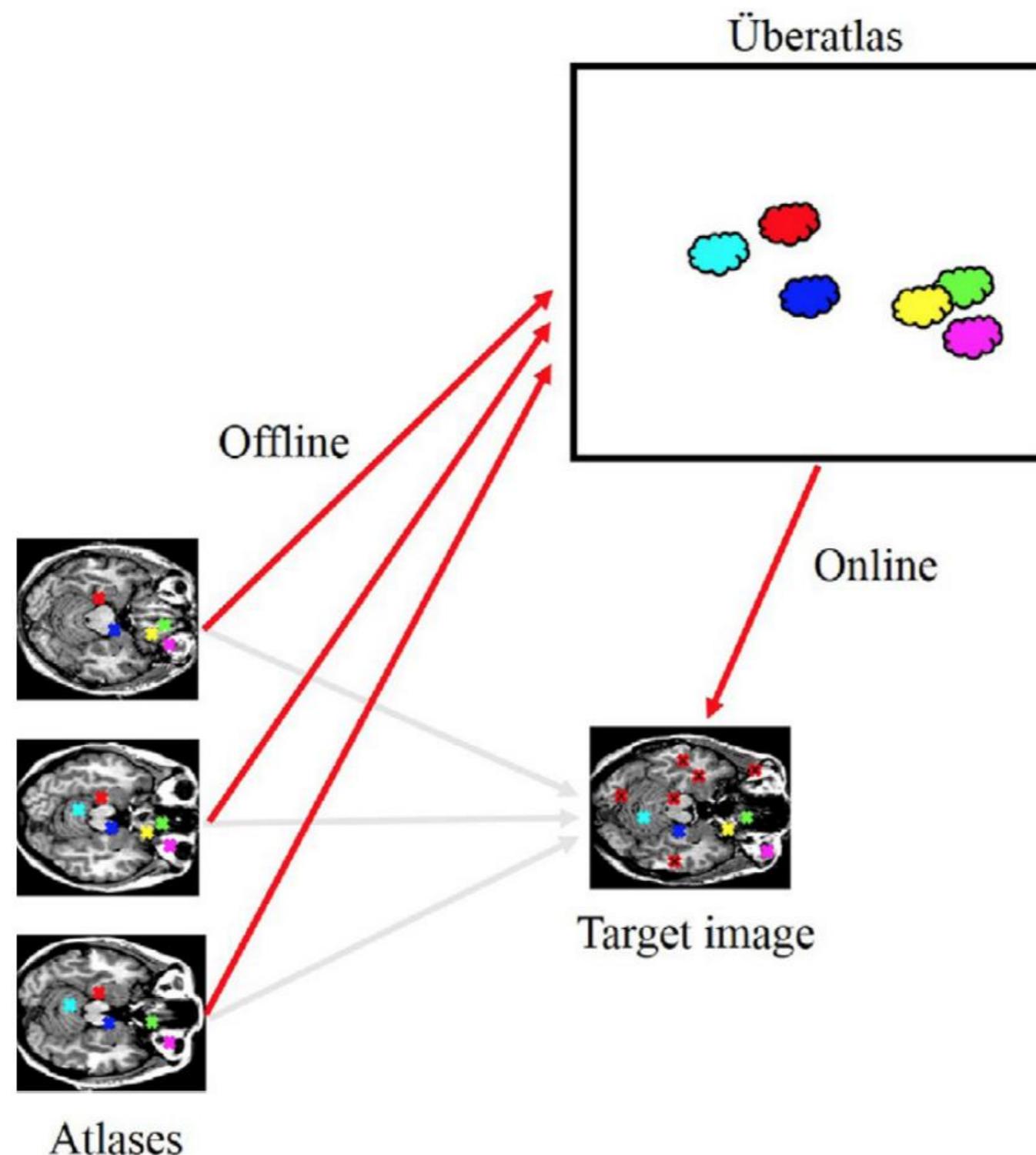
Tissue Sample Alignment



Stitching Microscopy Images



Atlas-Based Segmentation



[Jennifer Alvén, Alexander Norlén, Olof Enqvist, Fredrik Kahl, Überatlas: Fast and robust registration for multi-atlas segmentation, Pattern Recognition Letters 2016]

Image Registration

1. Extract & match features
2. Estimate transformation
3. Warp target to source image

Image Registration

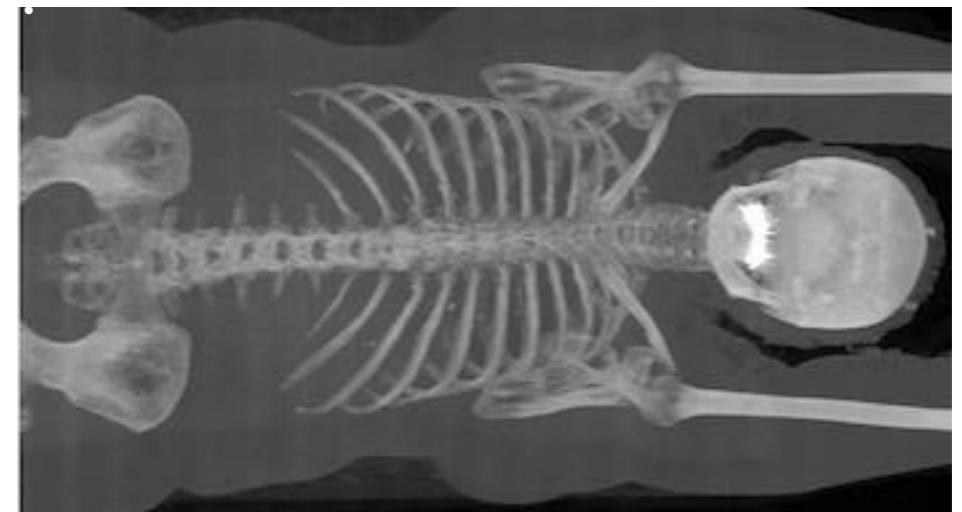
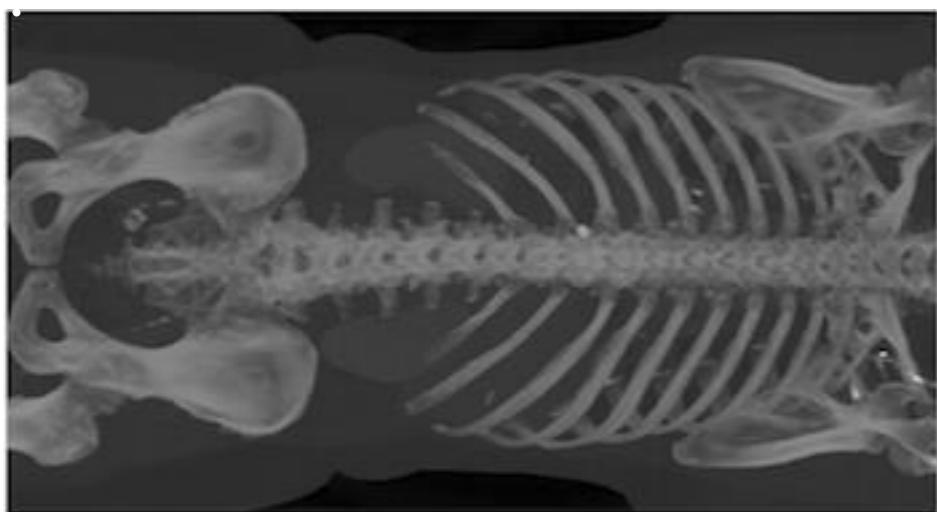
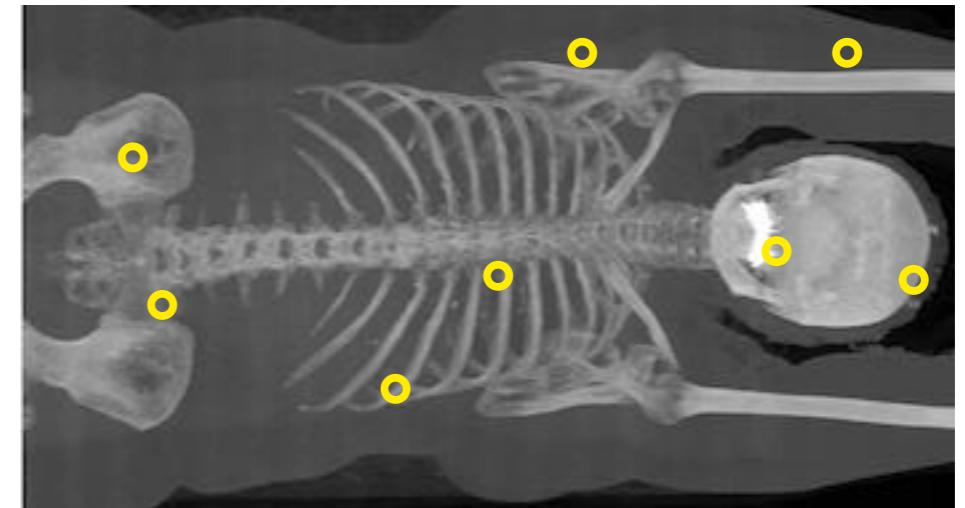
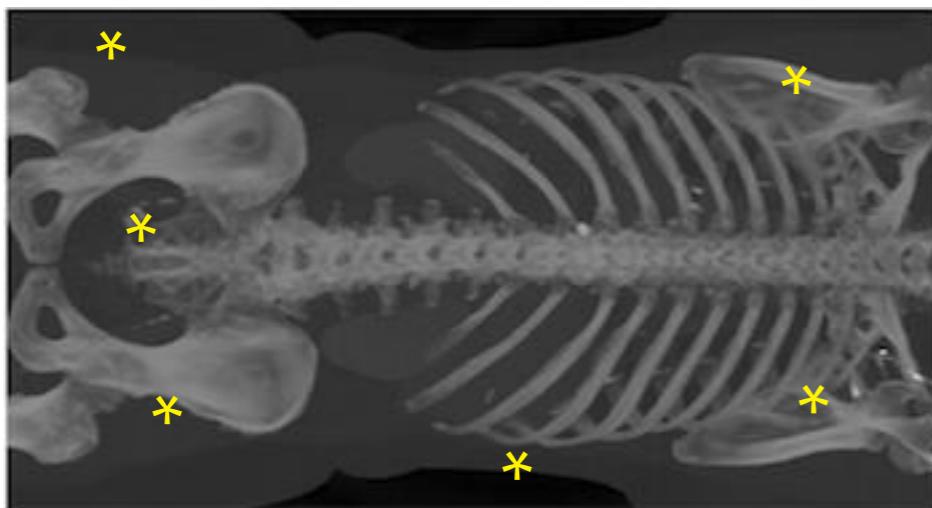
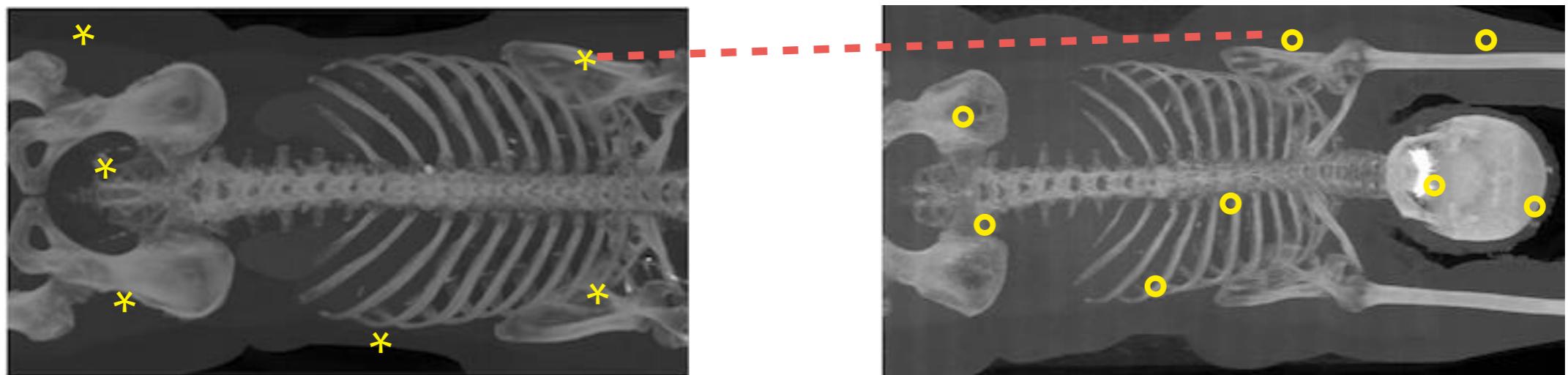


Image Registration



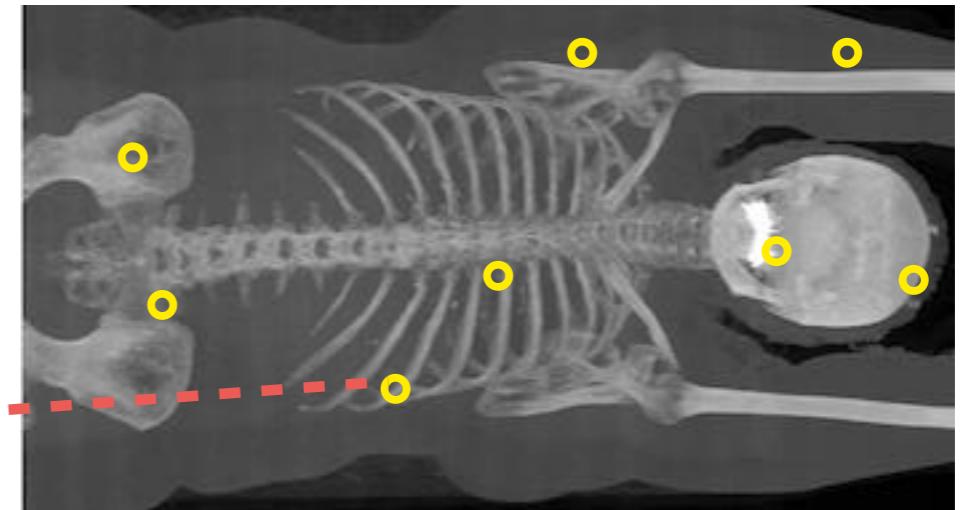
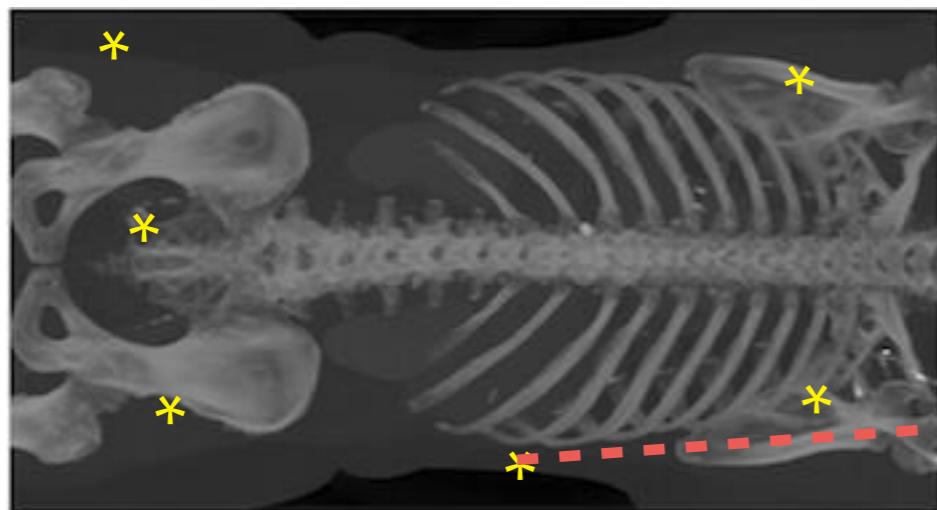
Extract features

Image Registration



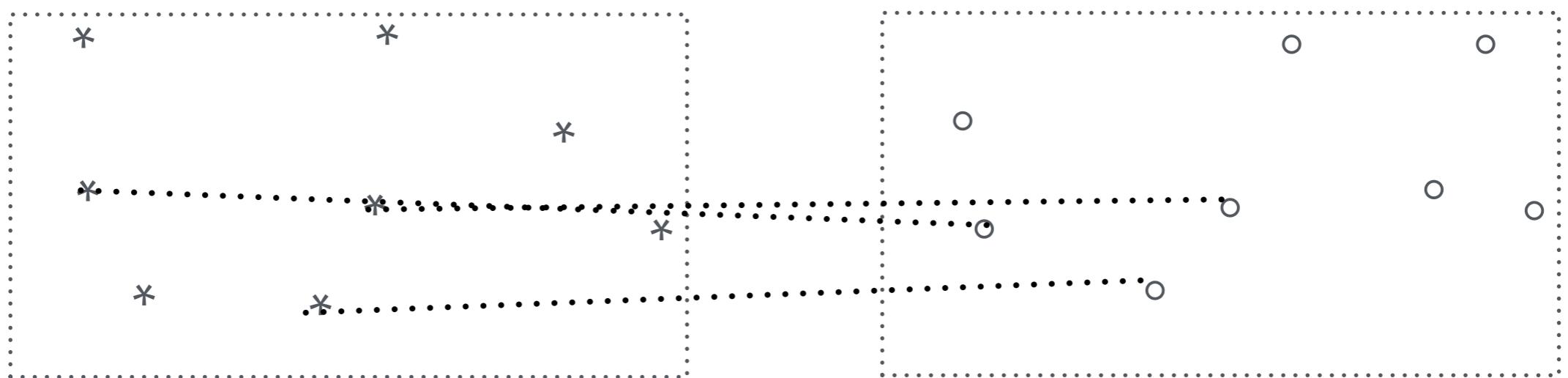
Find correspondences

Image Registration



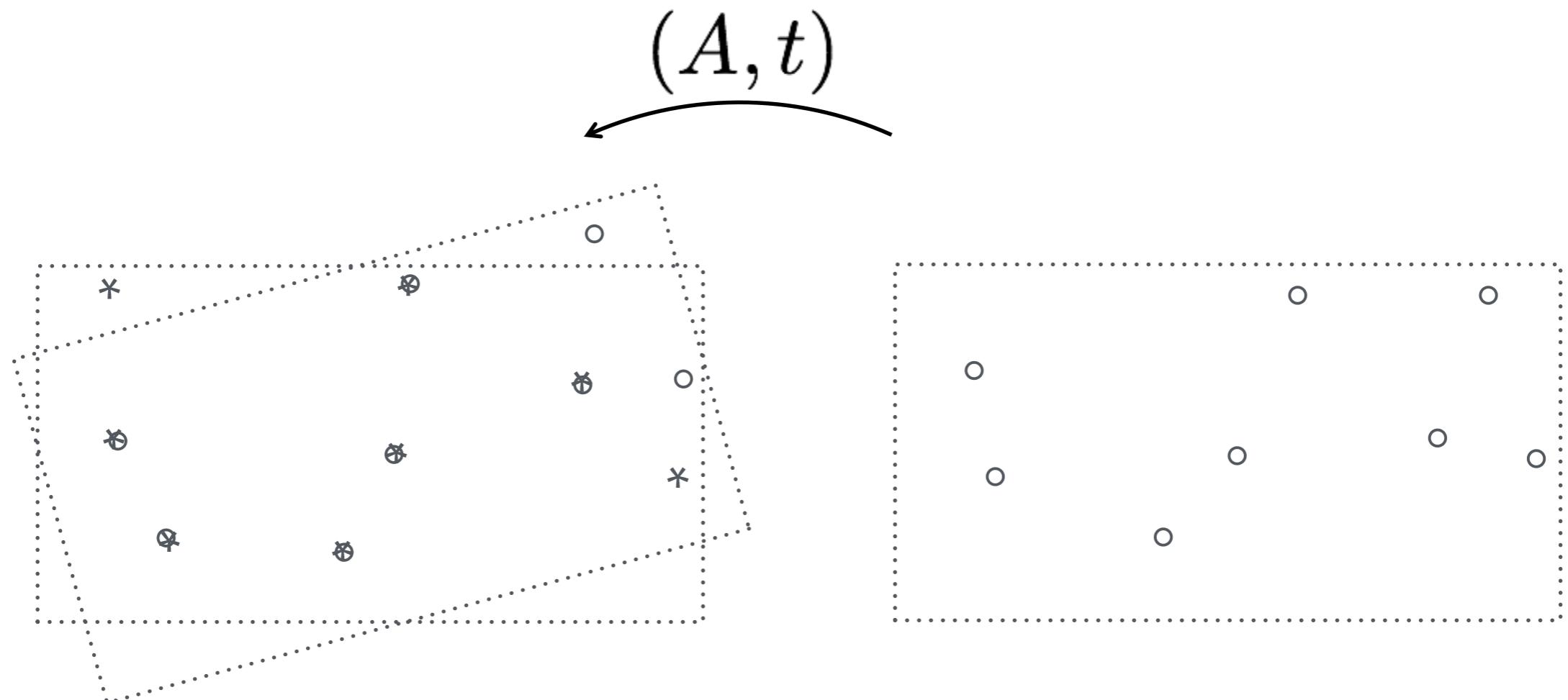
Find correspondences

Point Set Registration



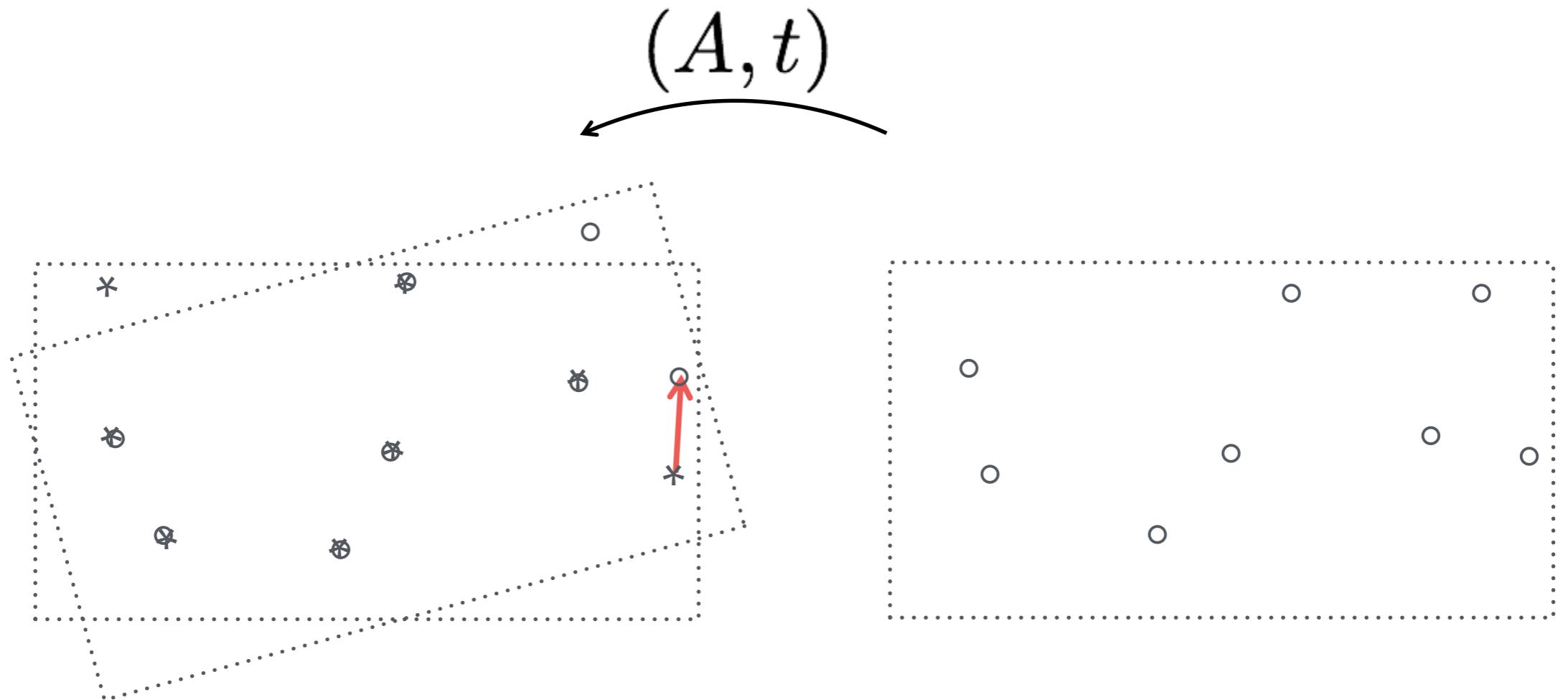
Point Set Registration

Find a consistent transformation (RANSAC+ minimal solver).



Point Set Registration

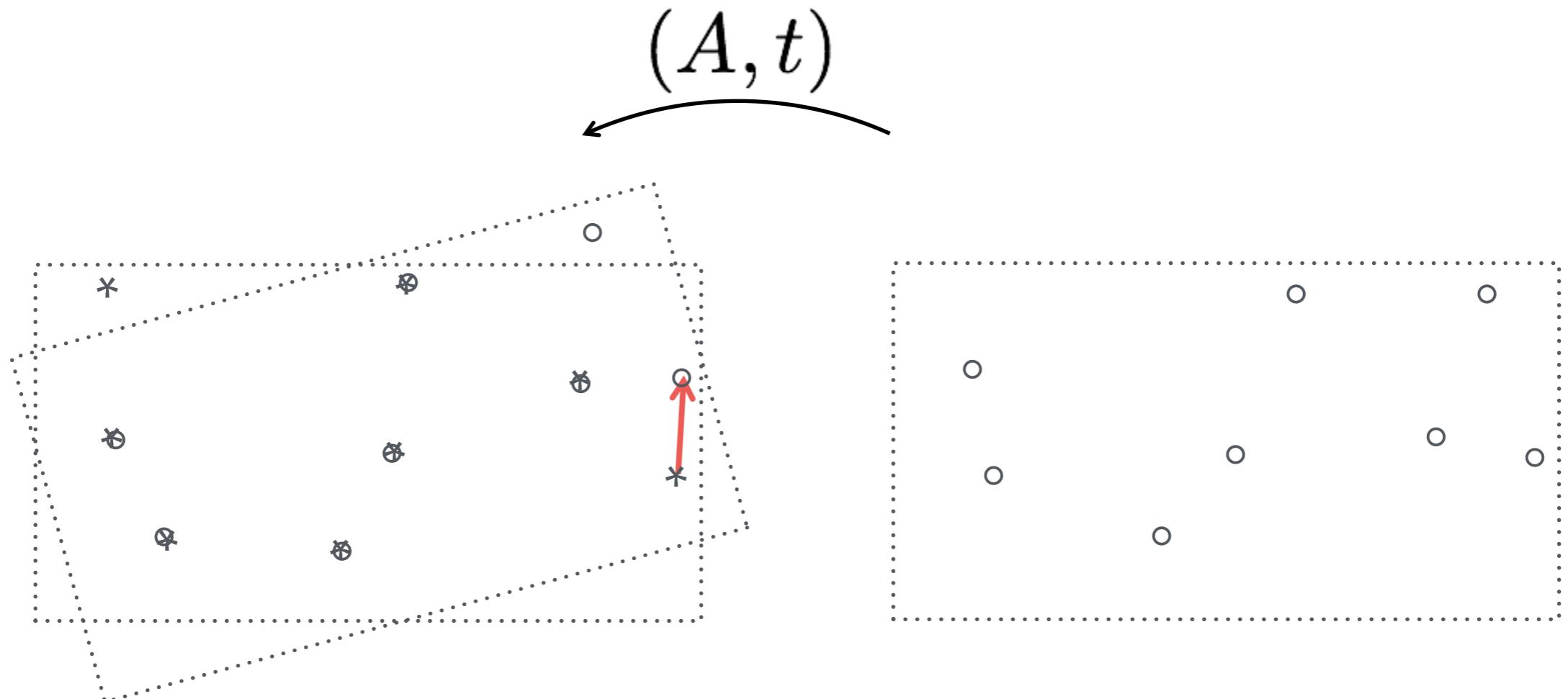
Find a consistent transformation (RANSAC+ minimal solver).



$$r_i(\theta) = A \begin{pmatrix} x_j \\ y_j \end{pmatrix} + t - \begin{pmatrix} \hat{x}_j \\ \hat{y}_j \end{pmatrix}$$

Point Set Registration

Find a consistent transformation (RANSAC+ minimal solver).

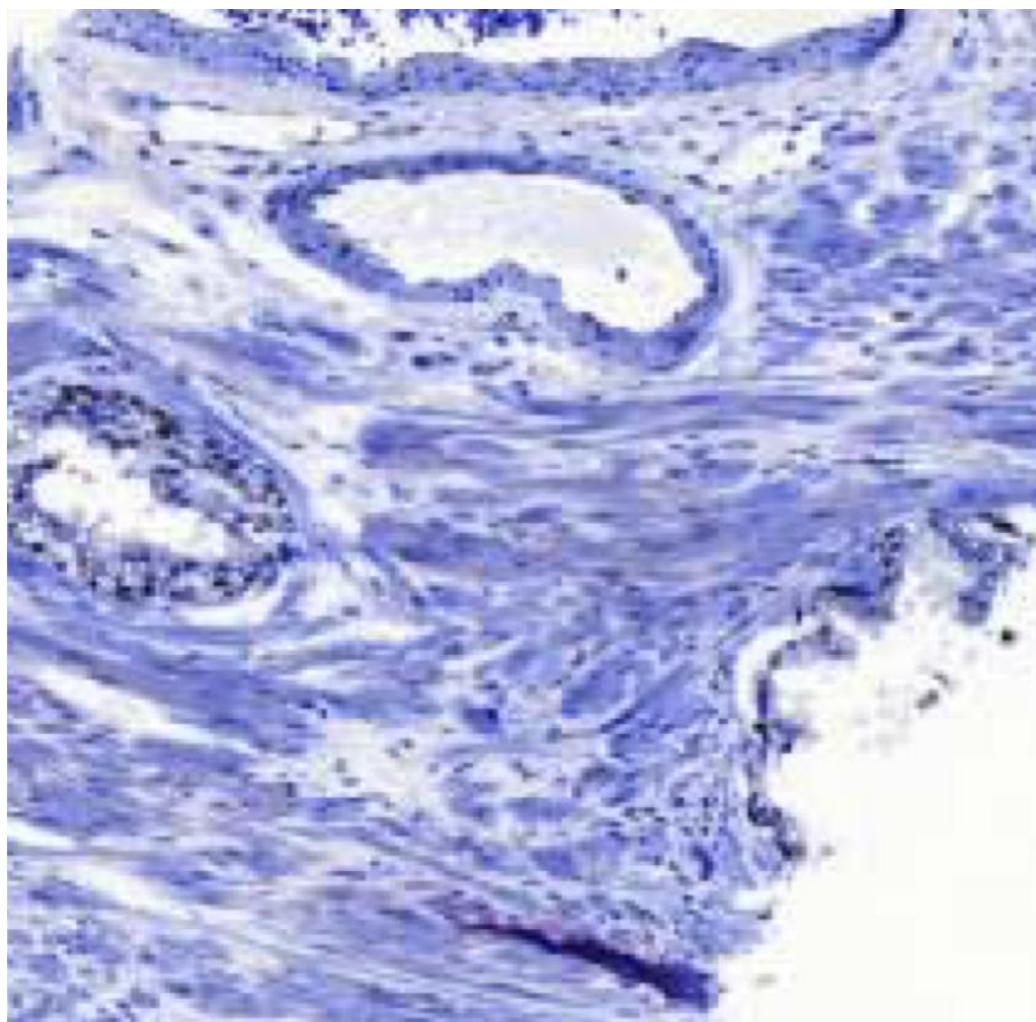


$$r_i(\theta) = A \begin{pmatrix} x_j \\ y_j \end{pmatrix} + t - \begin{pmatrix} \hat{x}_j \\ \hat{y}_j \end{pmatrix}$$

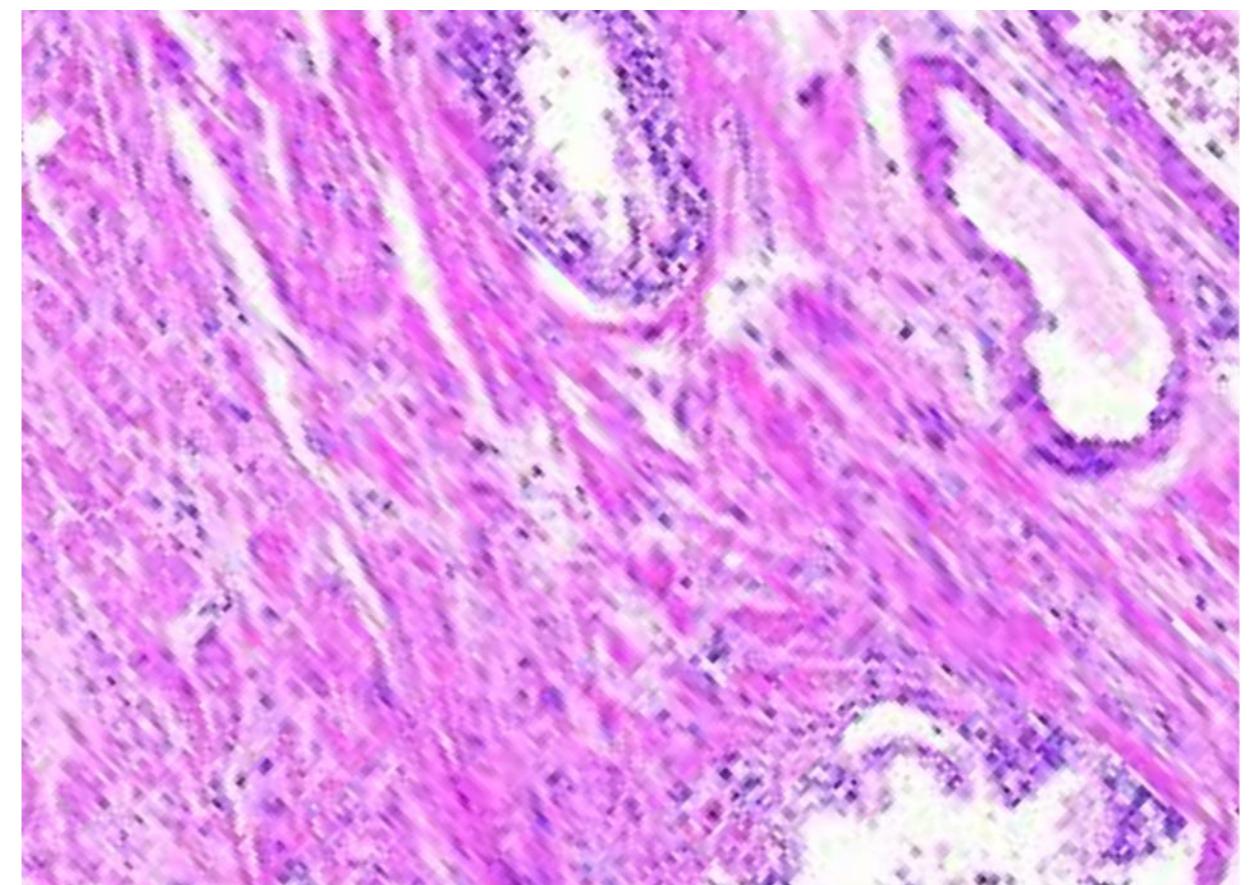
inlier test in RANSAC:
 $\|r_i(\theta)\| < \tau$

Image Warping

transformation



source image



target image

Image Warping

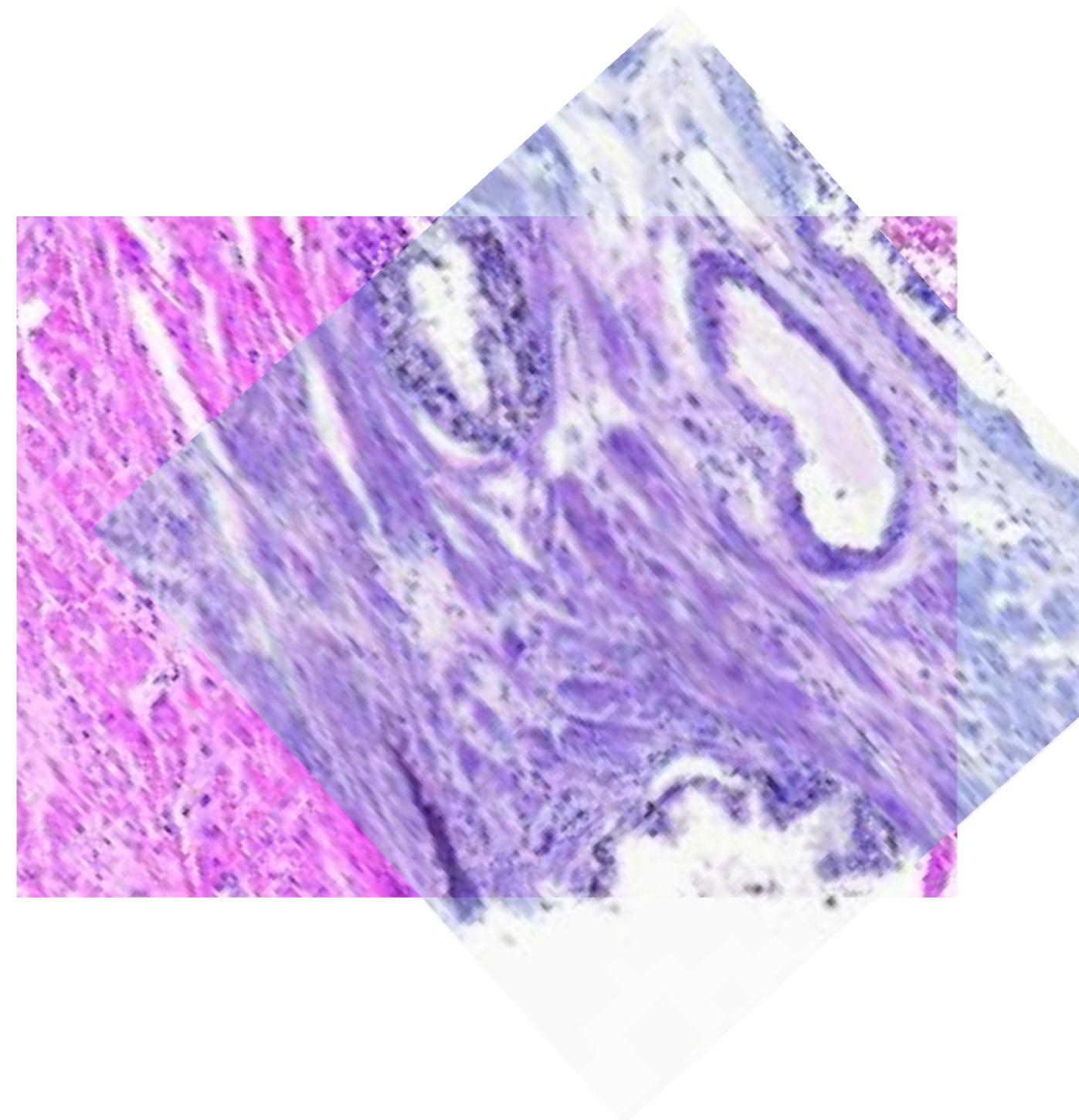


Image Warping

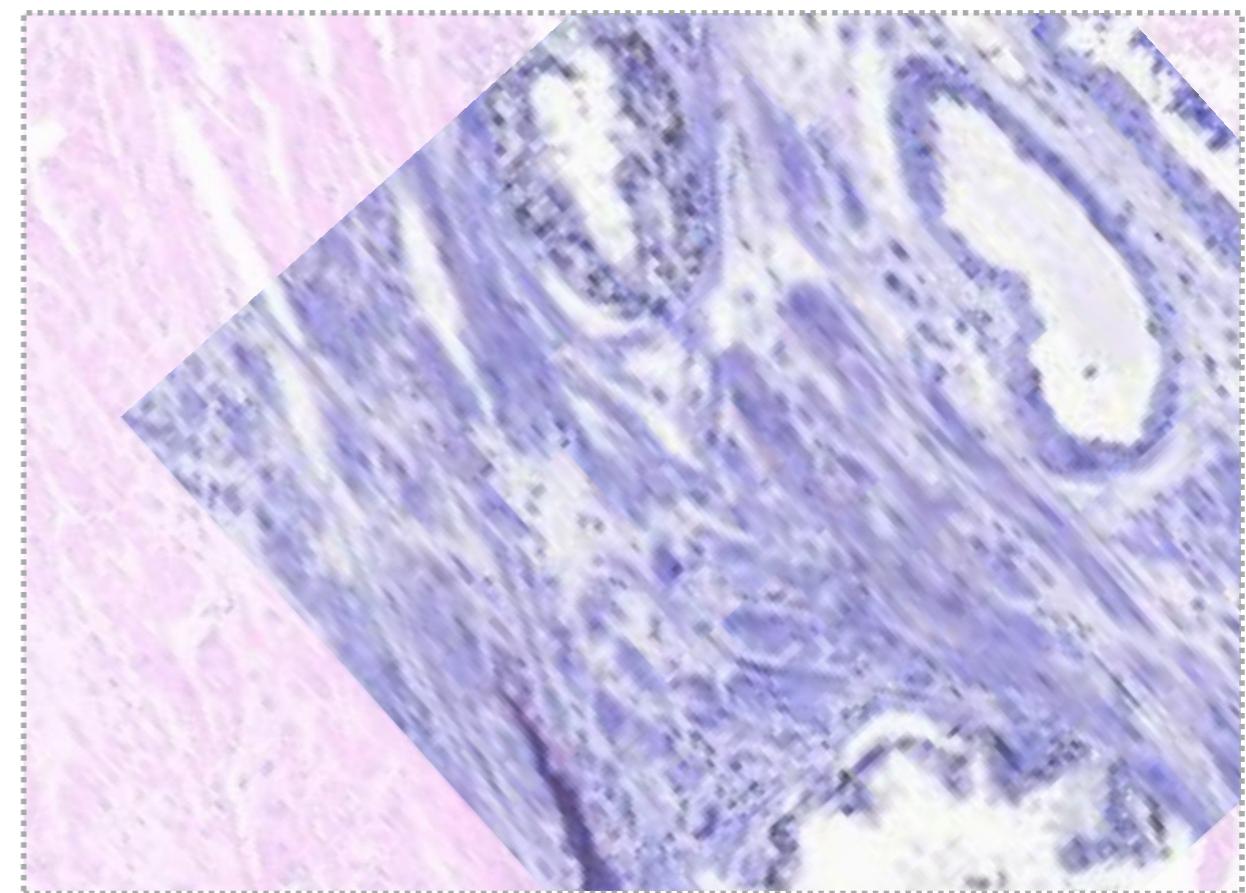
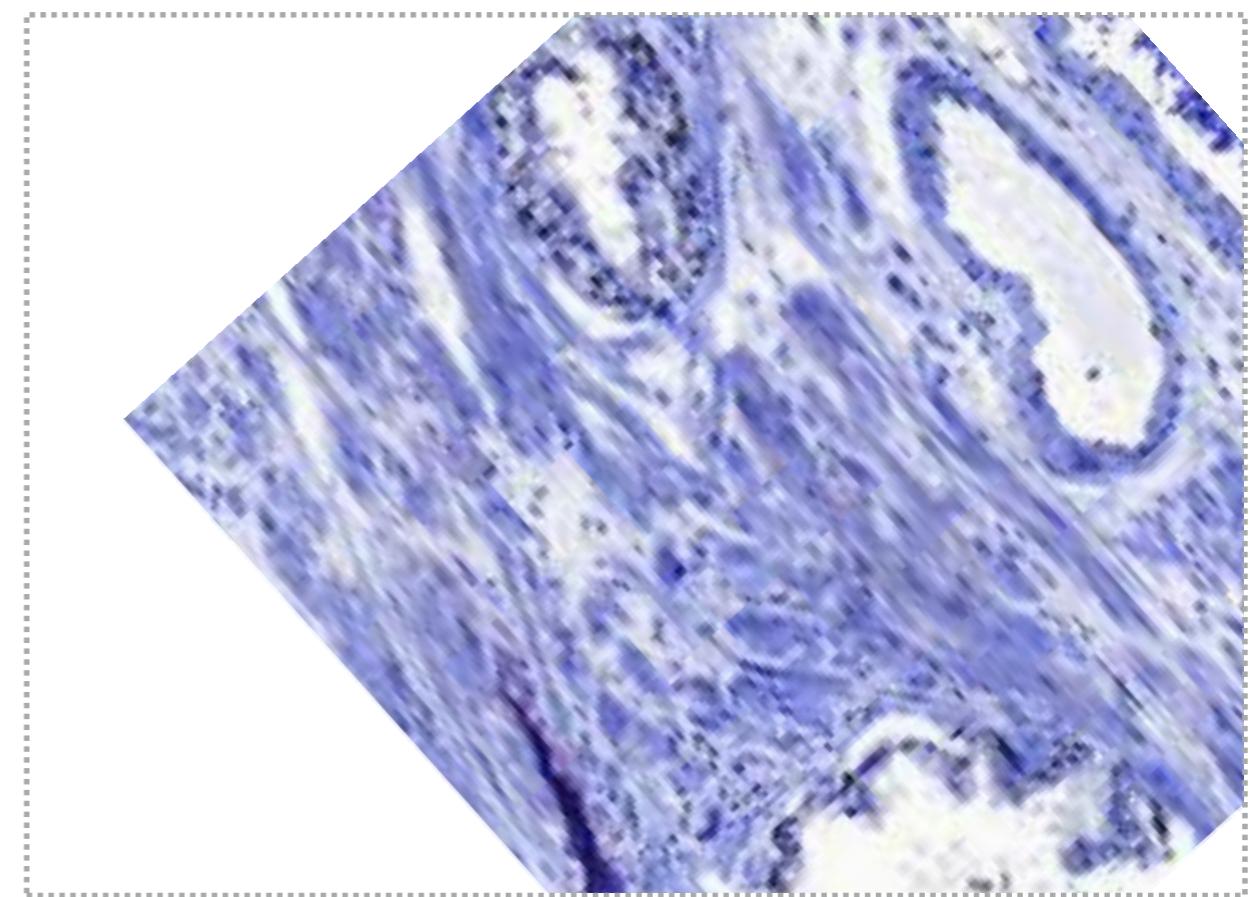
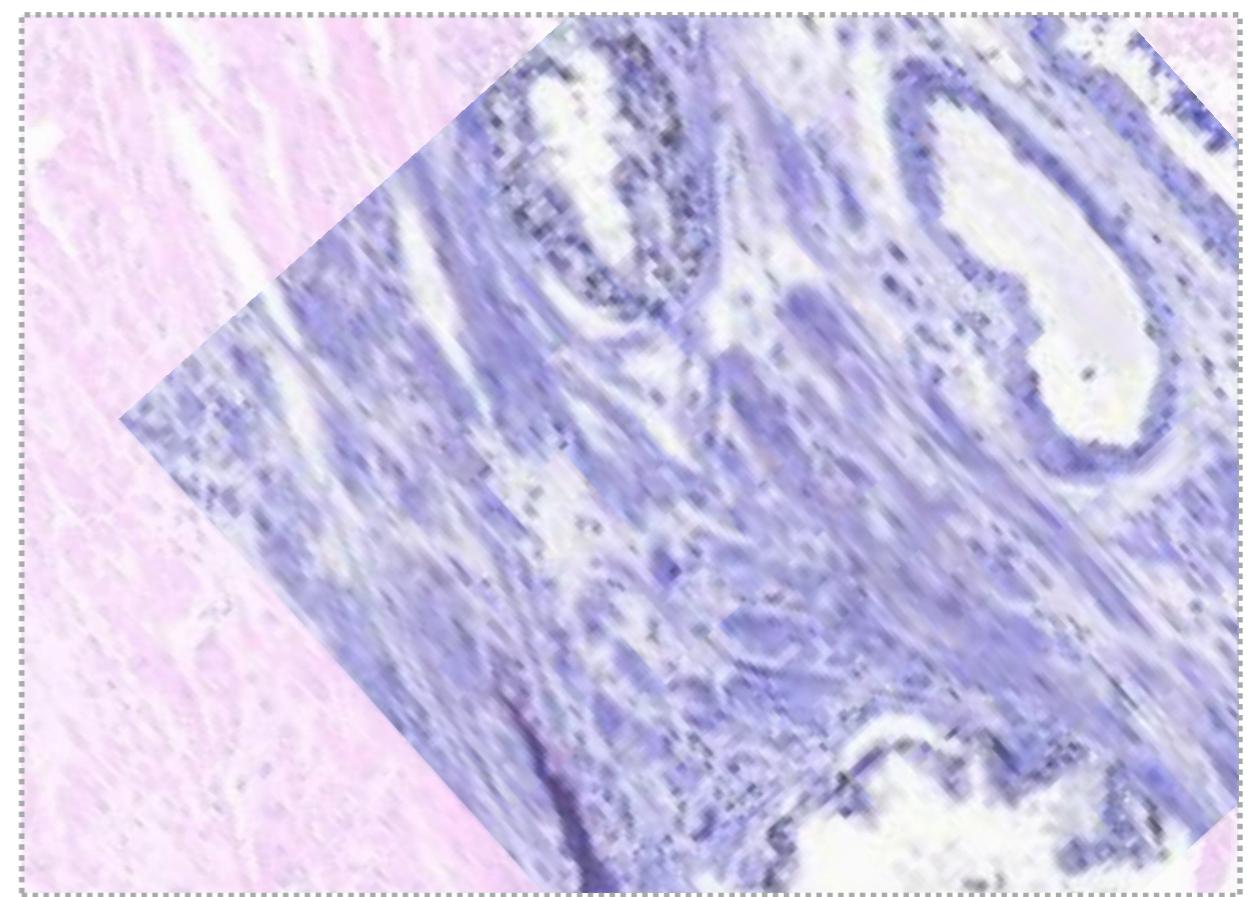


Image Warping



warped image

Image Analysis

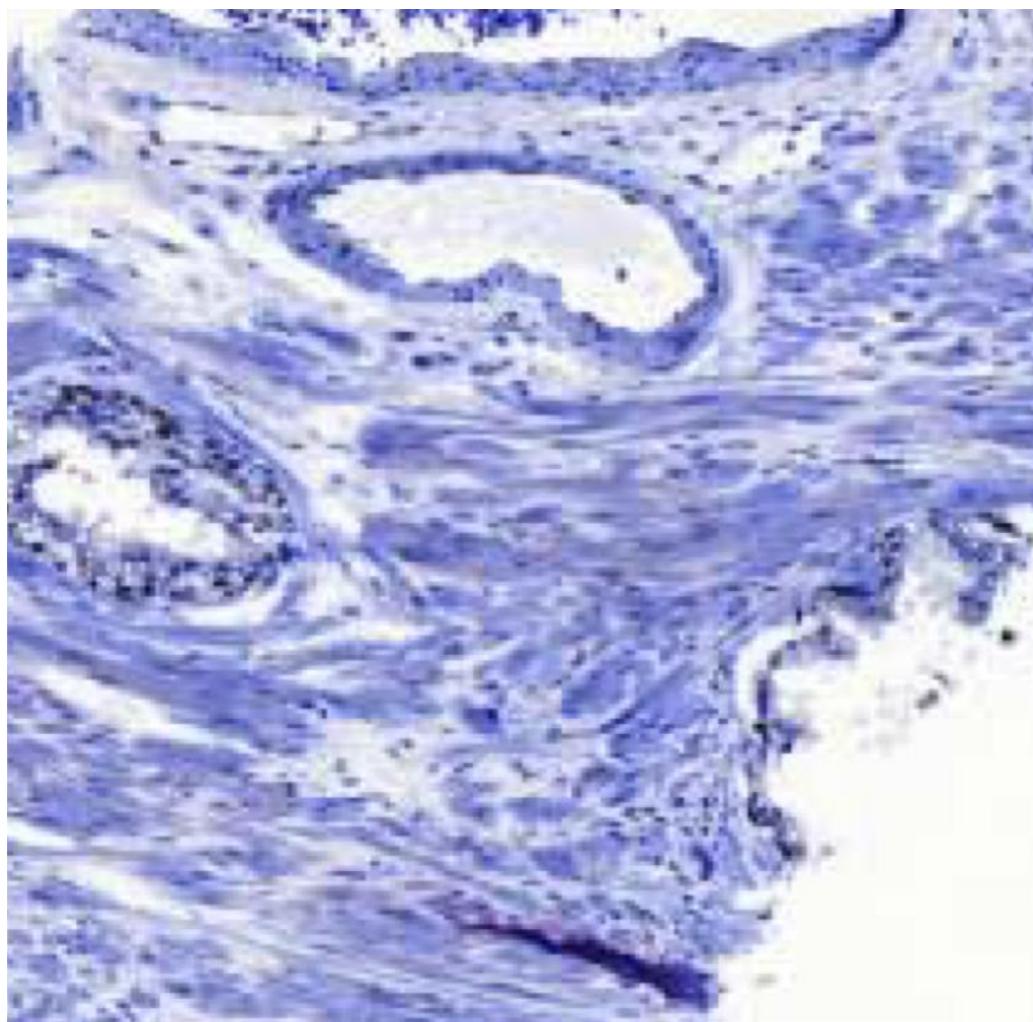


Today

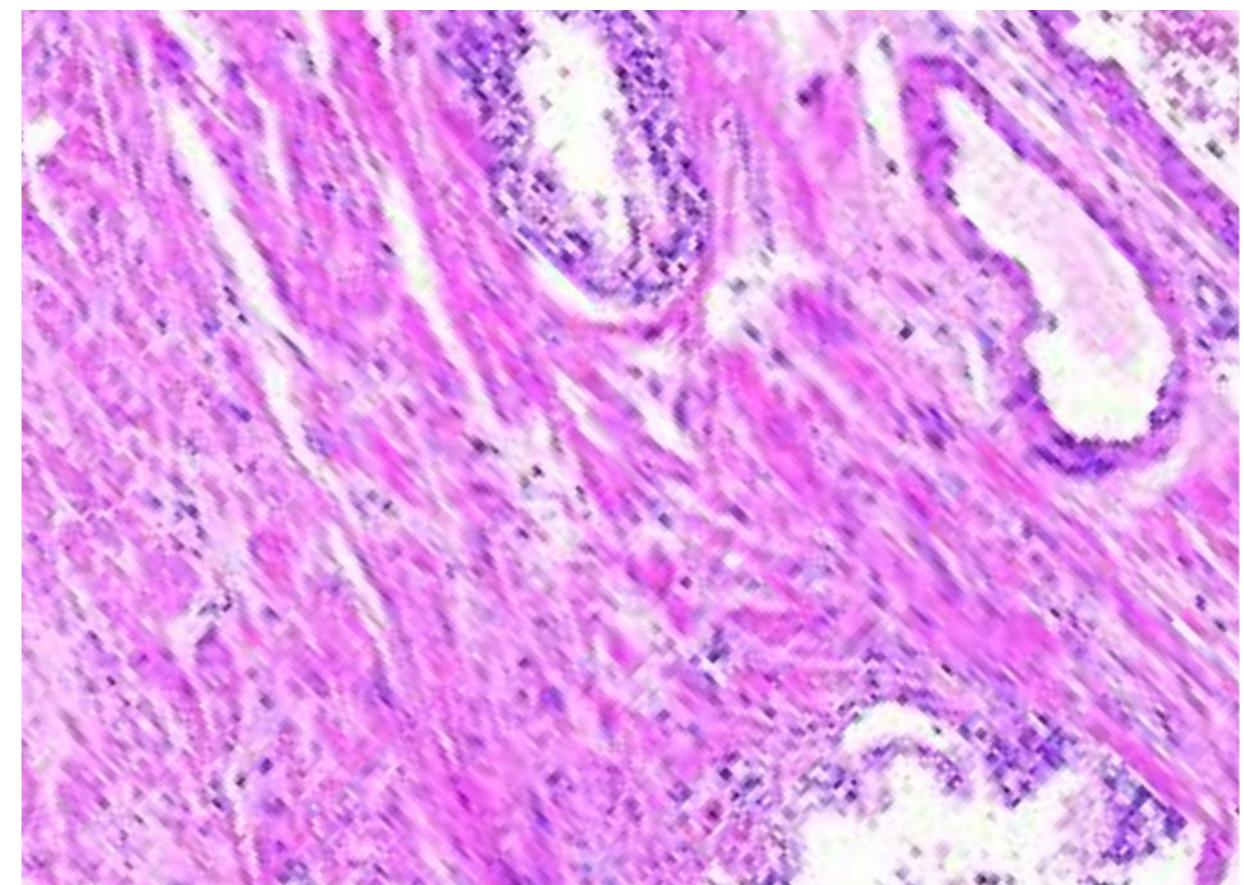
- Image Warping
- Transformation Estimation

Image Warping

transformation



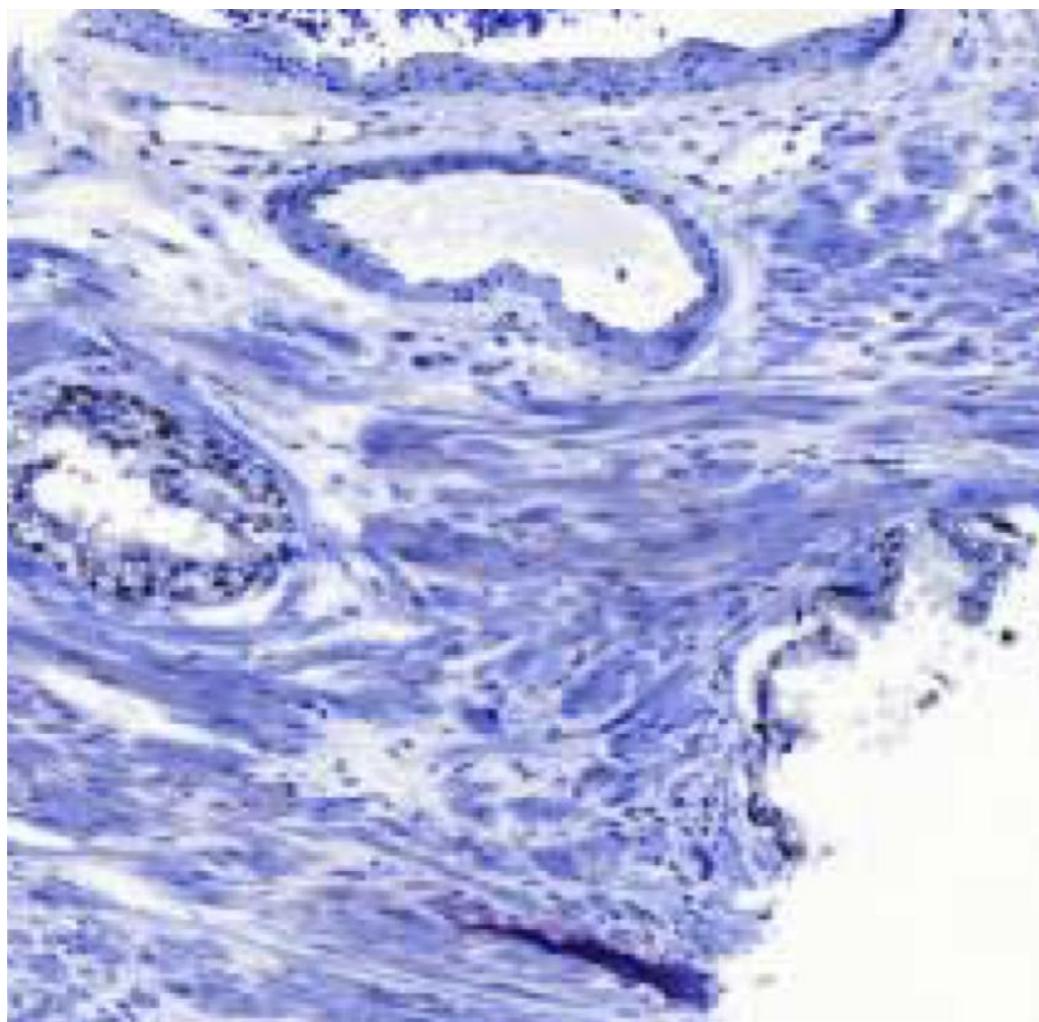
source image



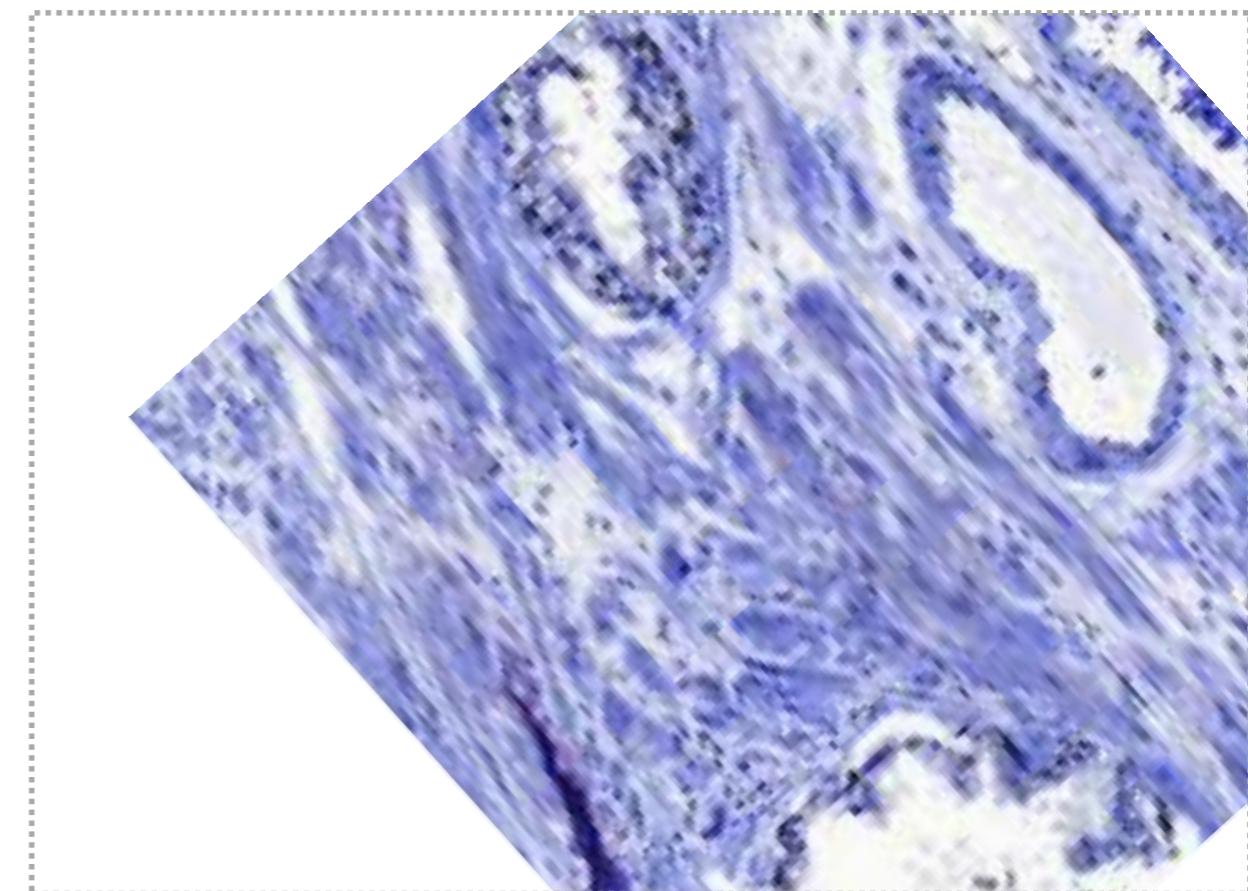
target image

Image Warping

transformation

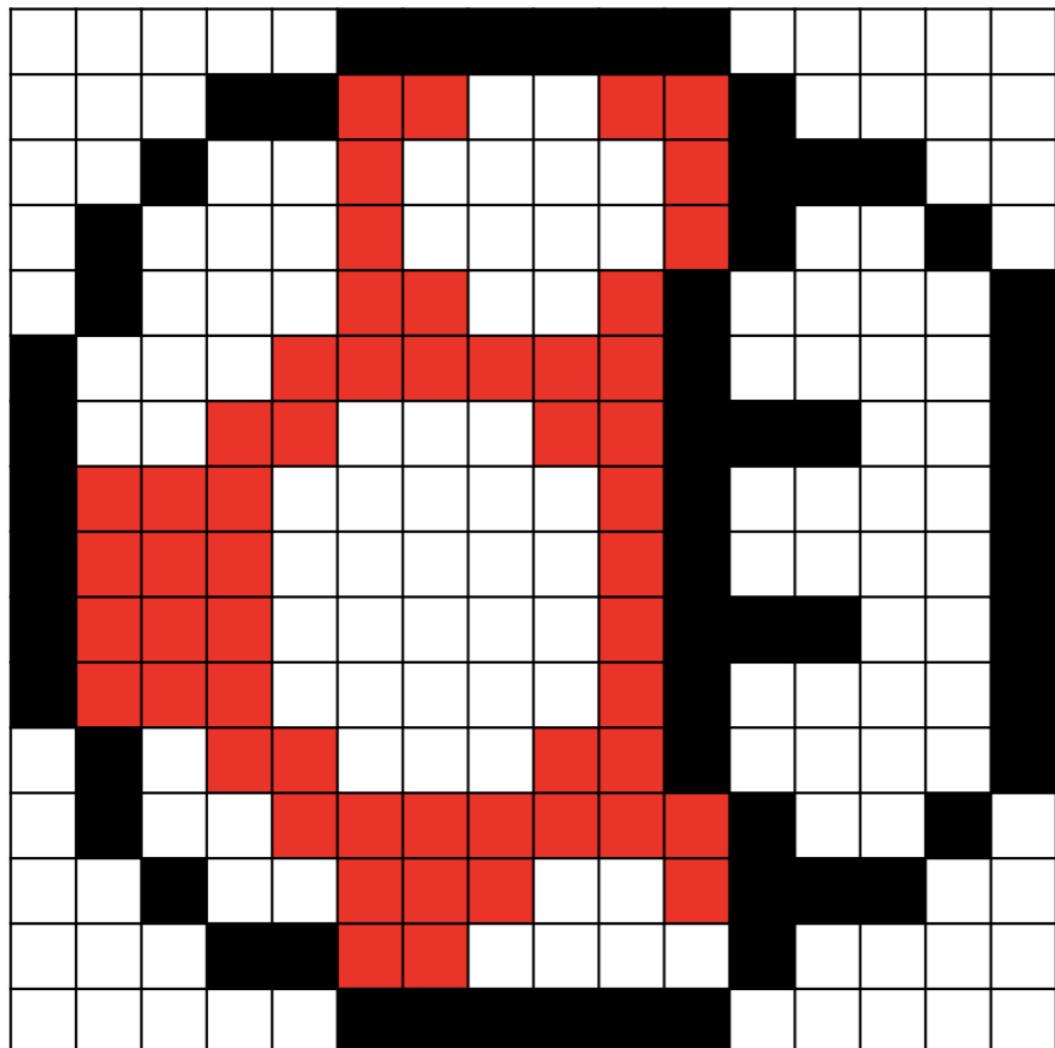


source image

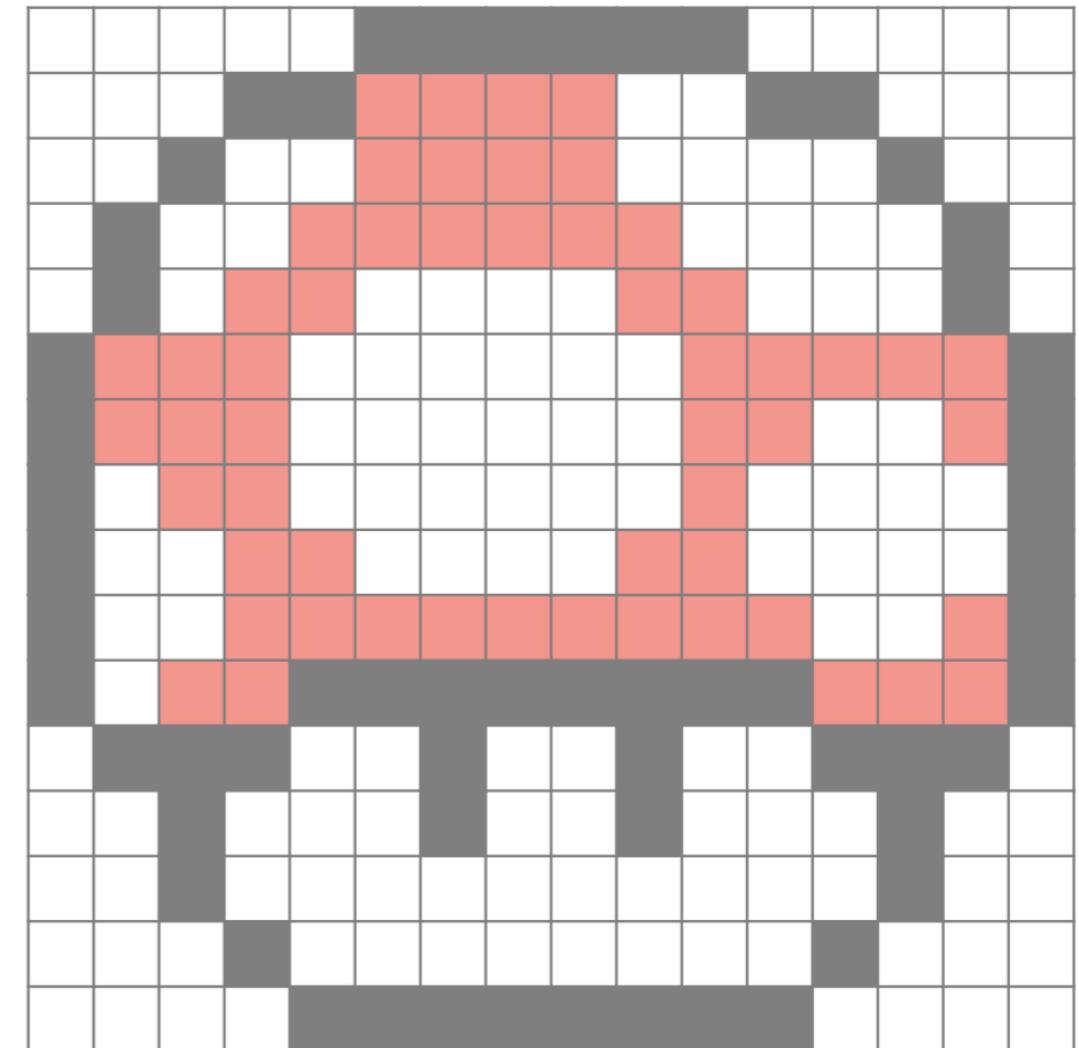


warped image

A Simple Example



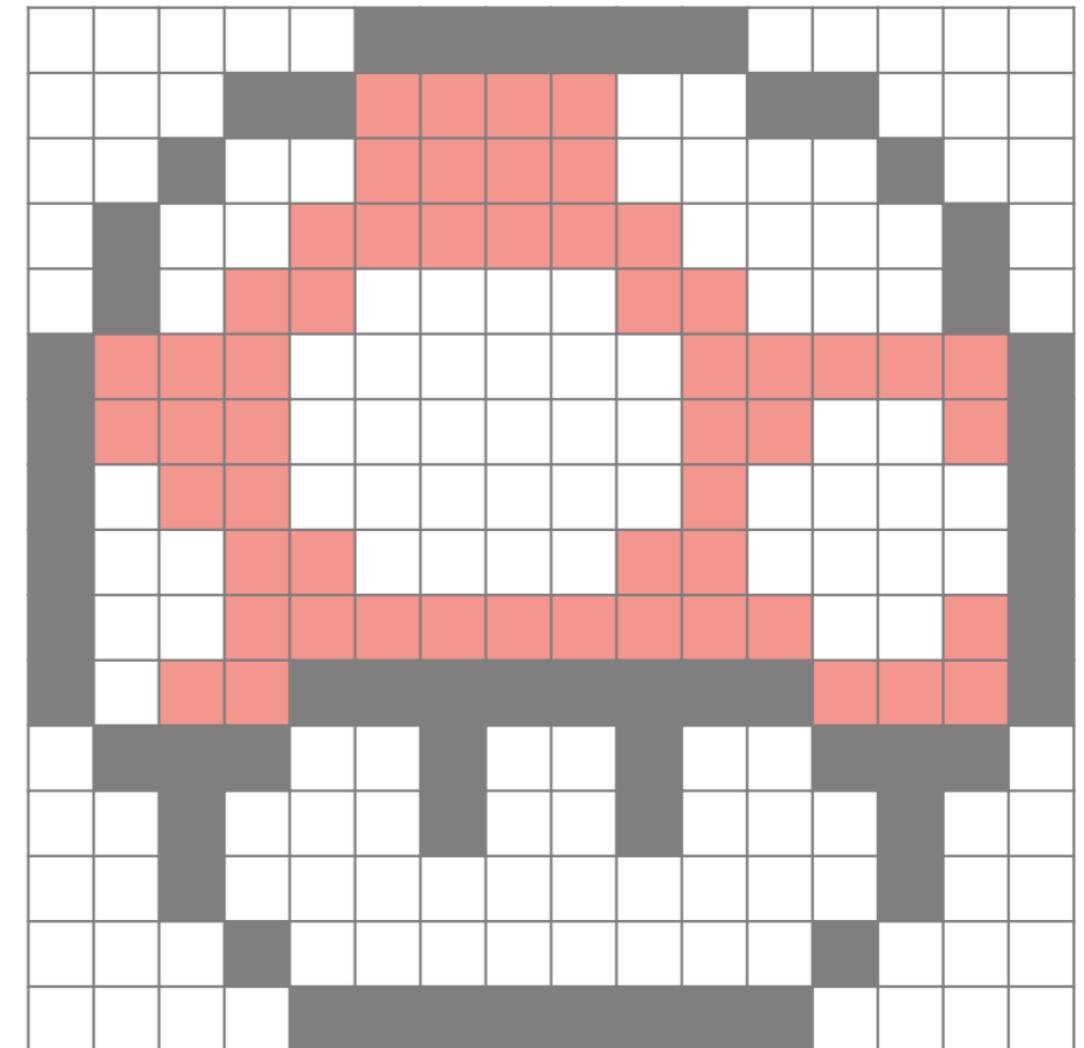
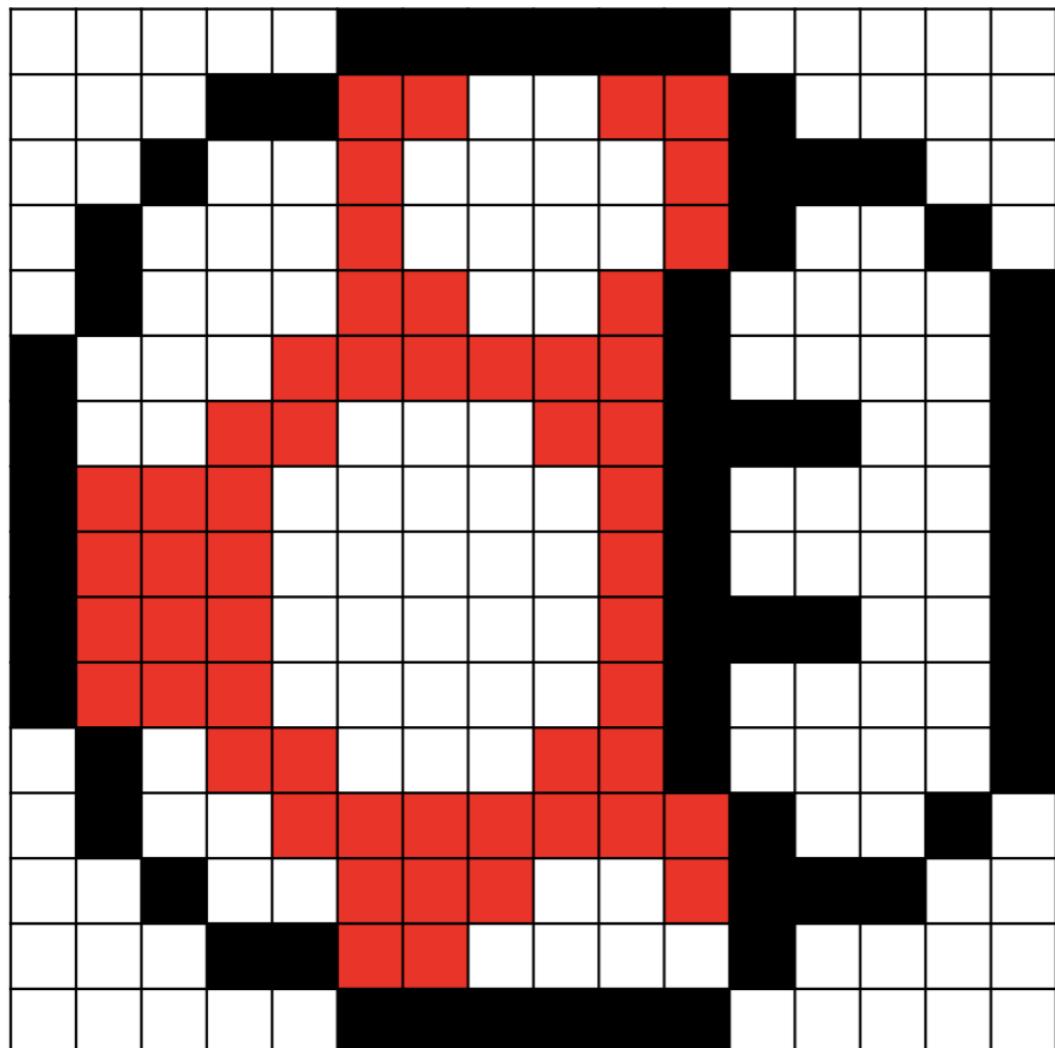
source image



target image

A Simple Example

Transformation (rotation)

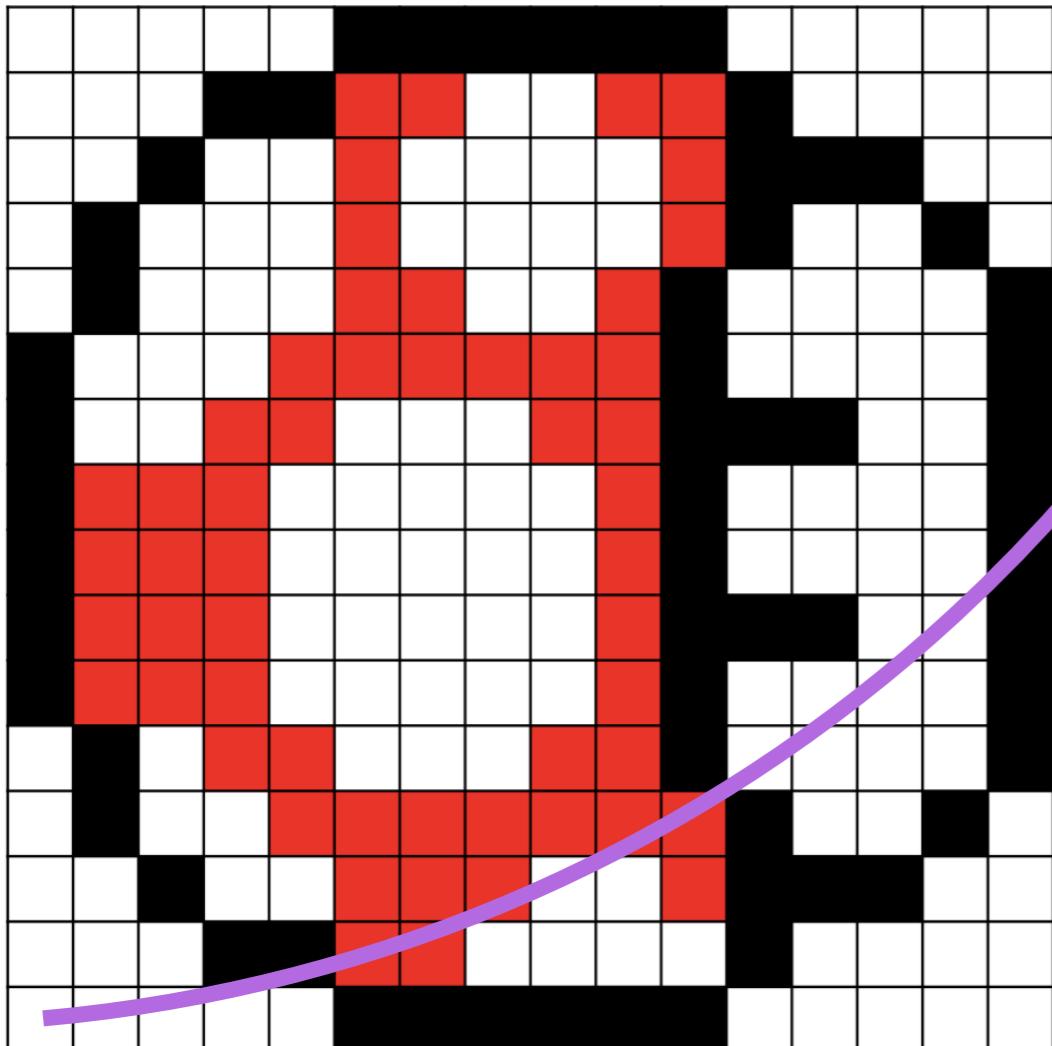


source image

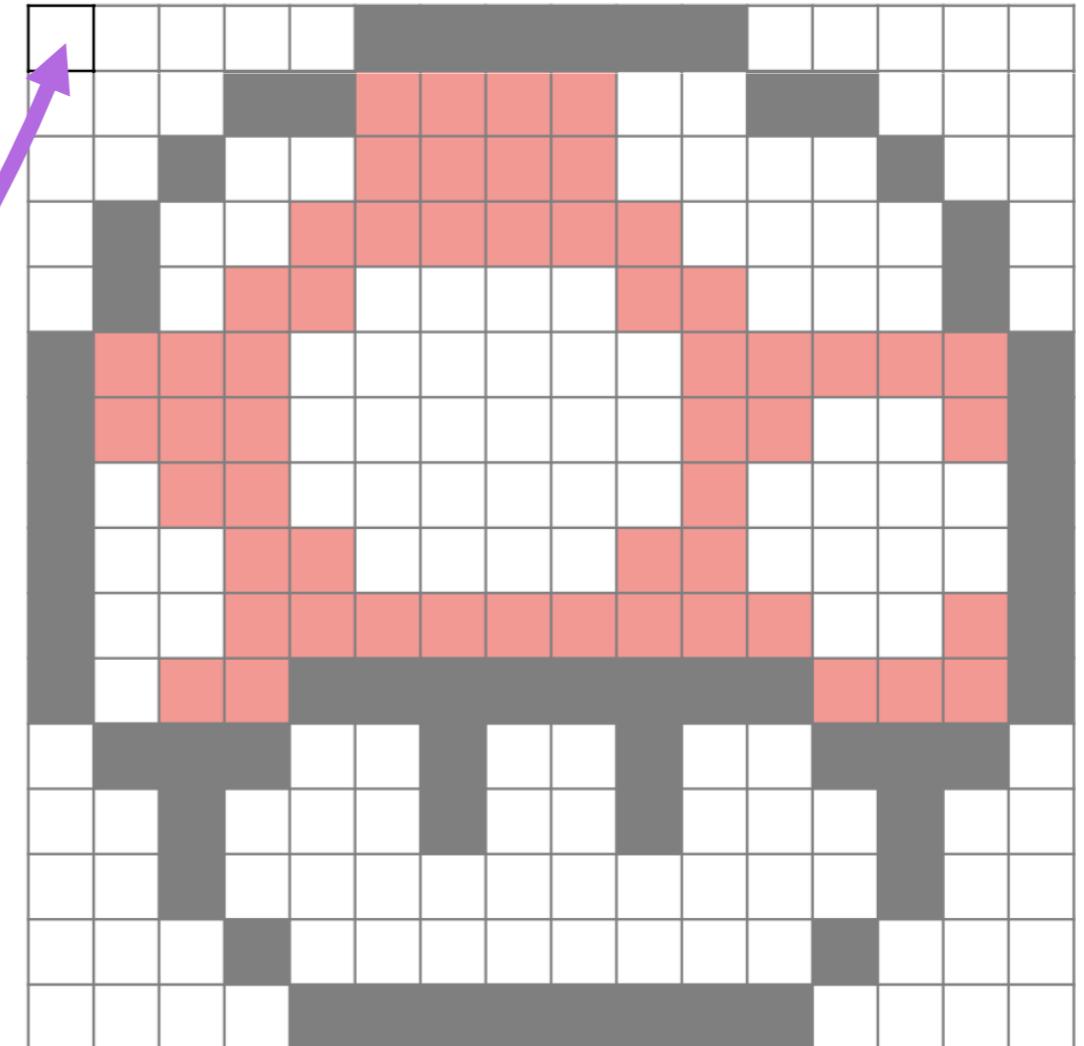
target image

A Simple Example

Transformation (rotation)



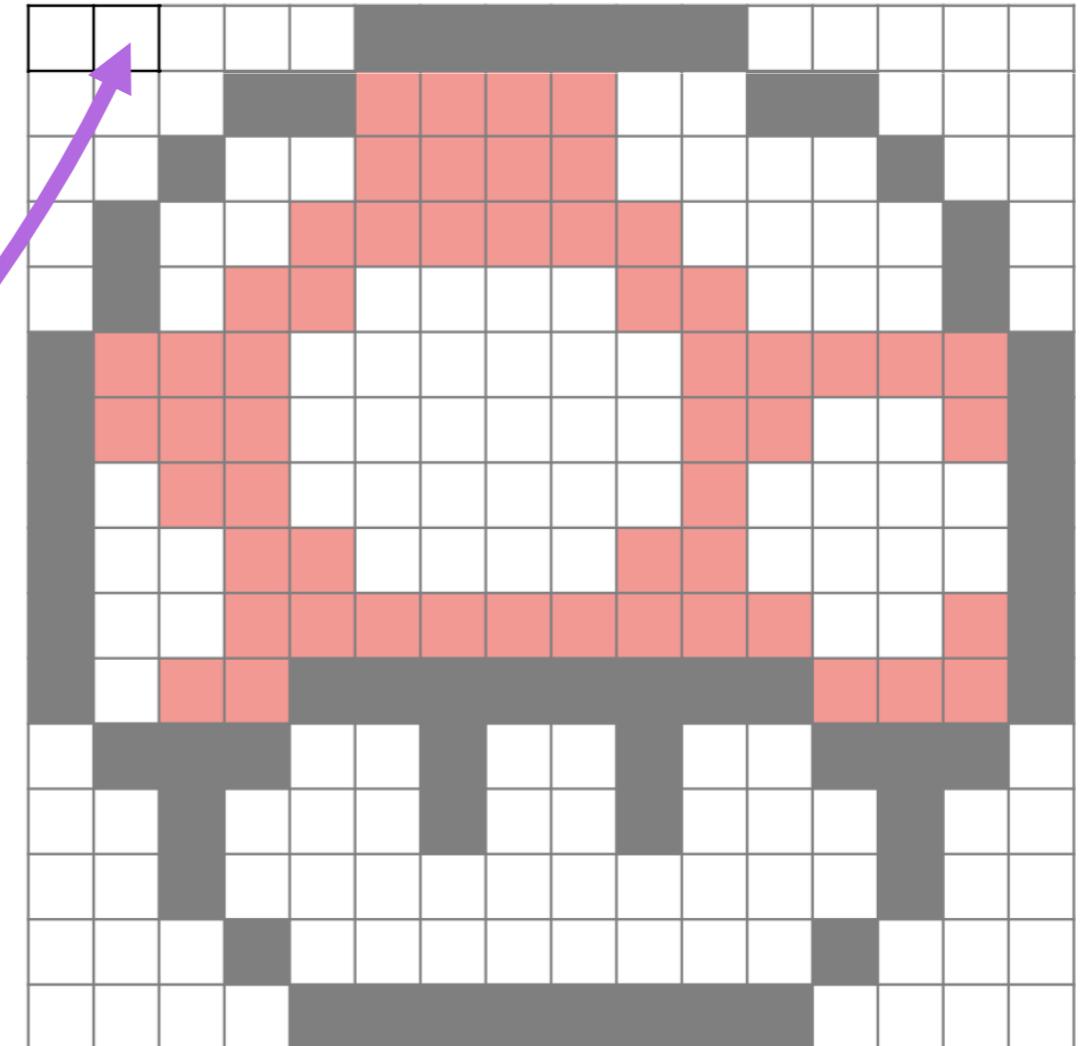
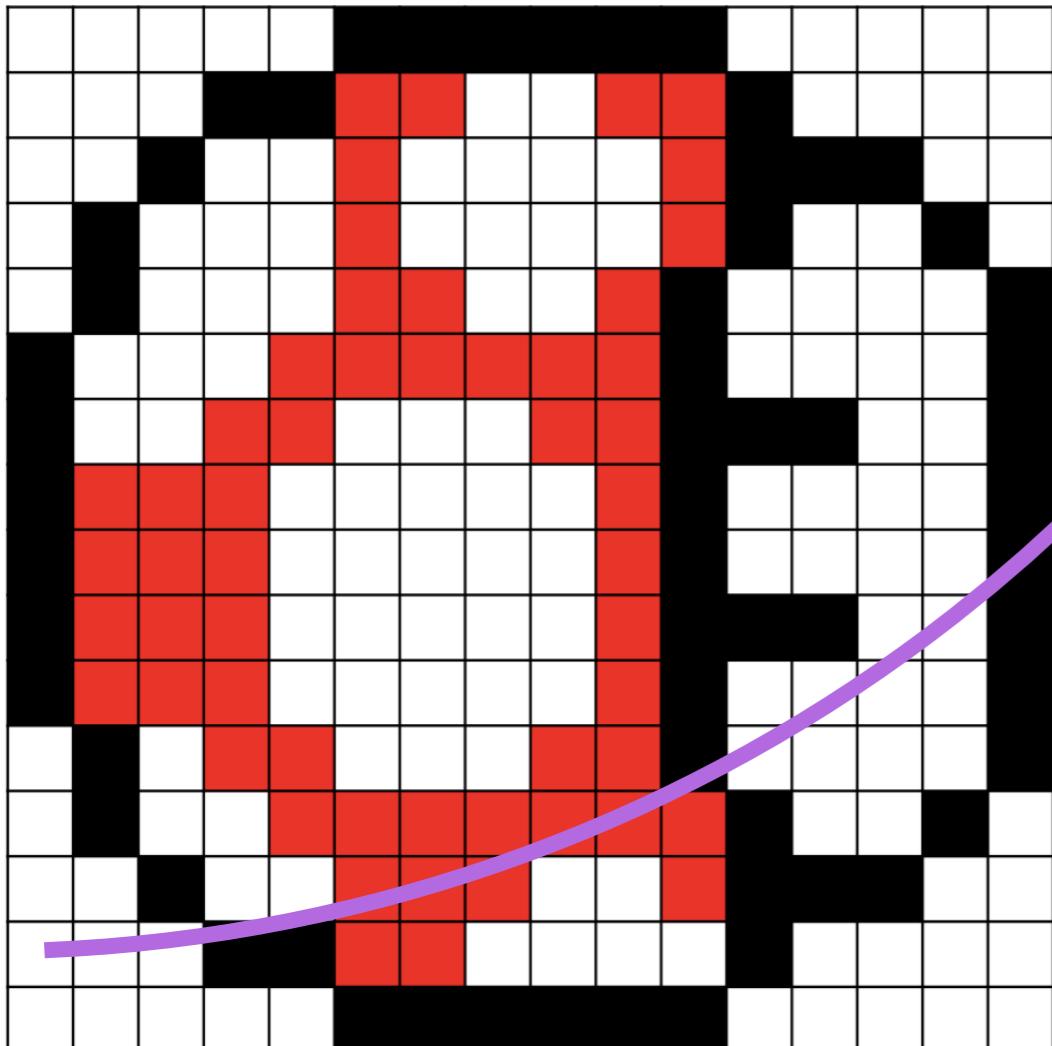
source image



target image

A Simple Example

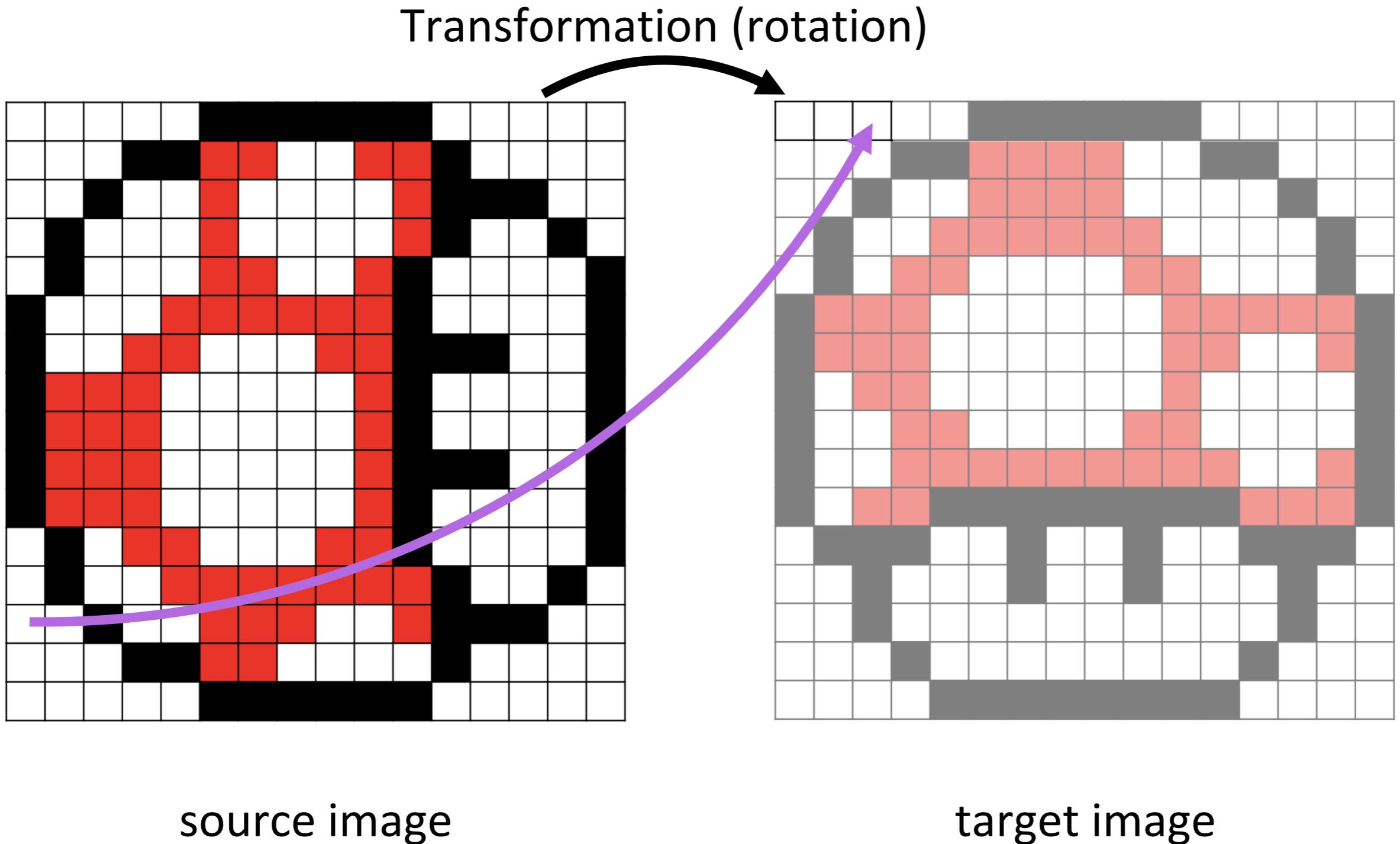
Transformation (rotation)



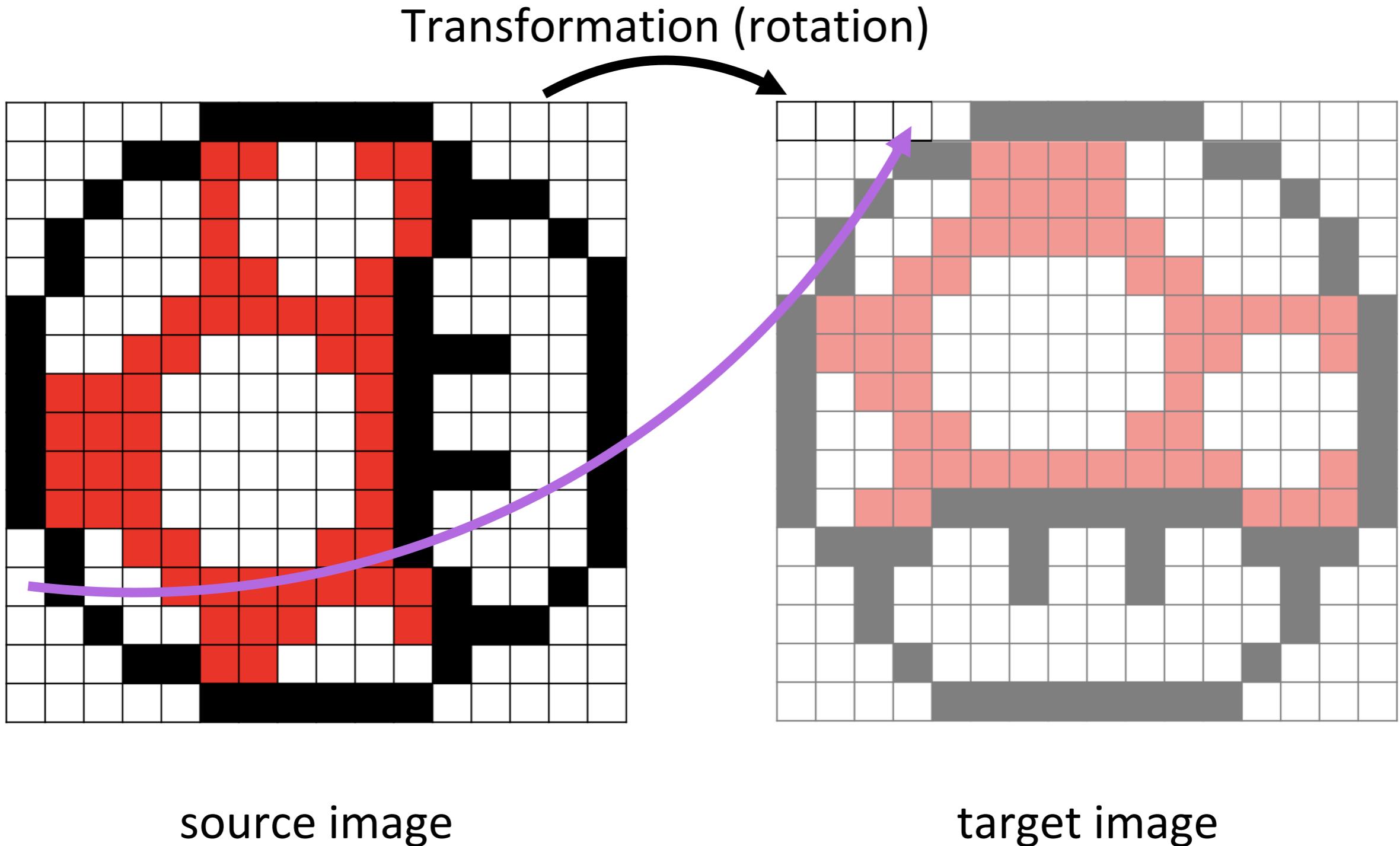
source image

target image

A Simple Example

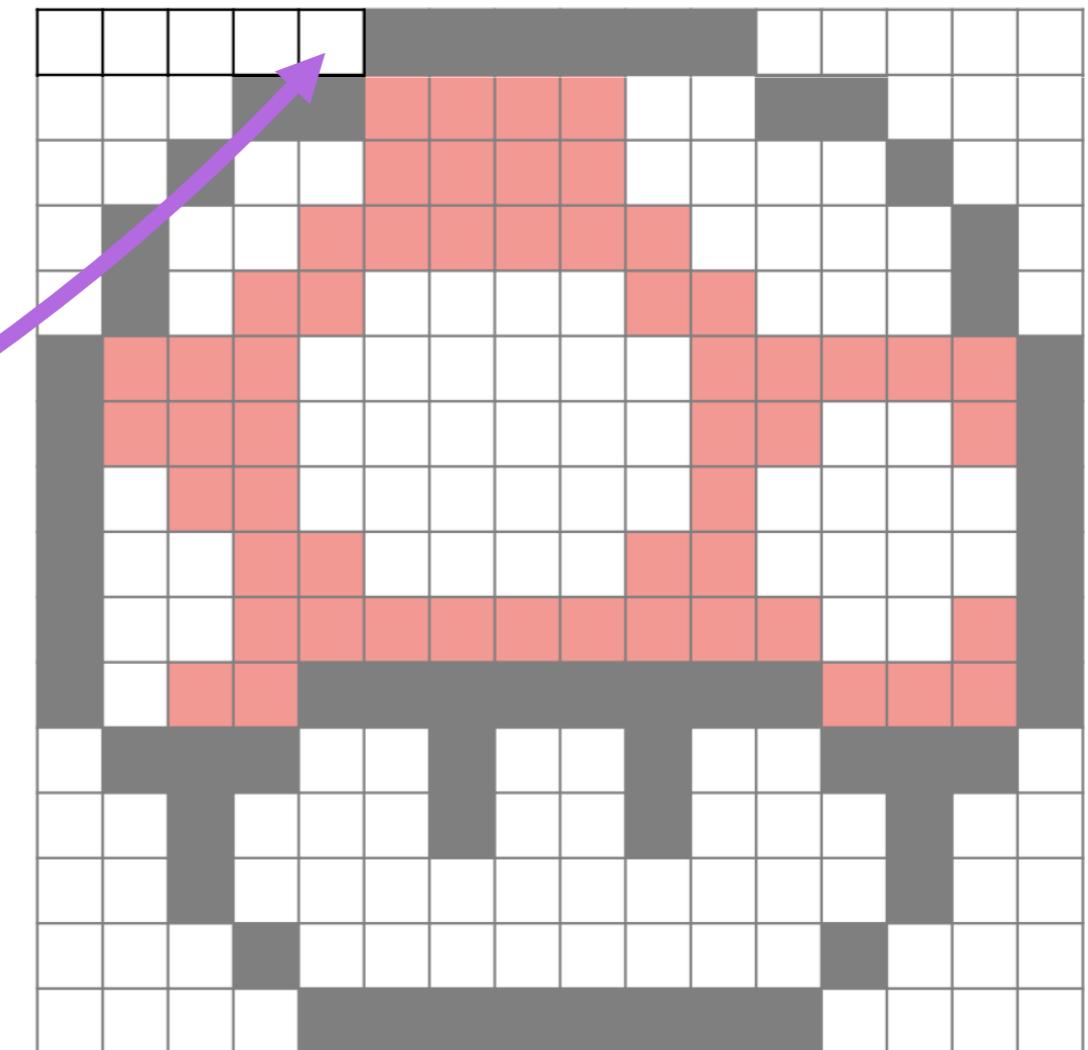
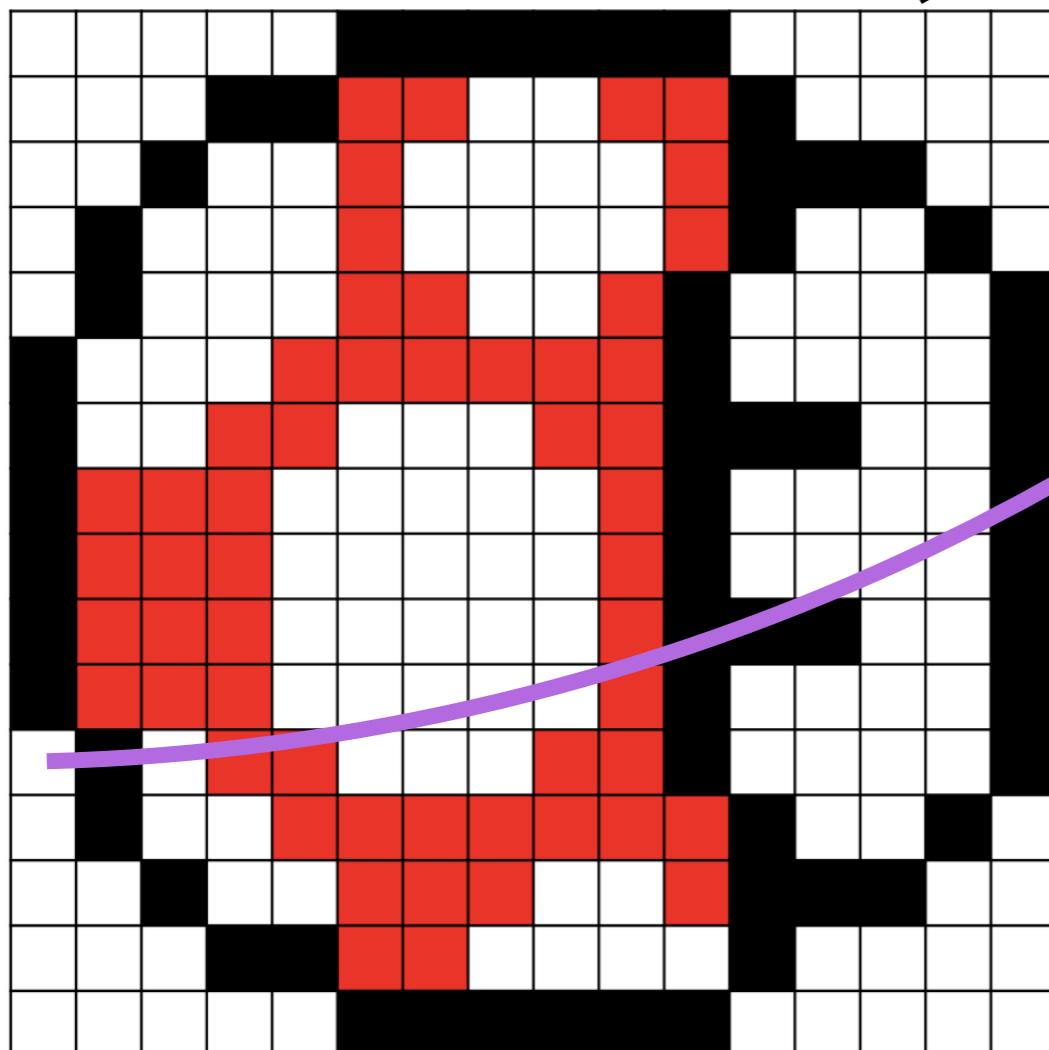


A Simple Example



A Simple Example

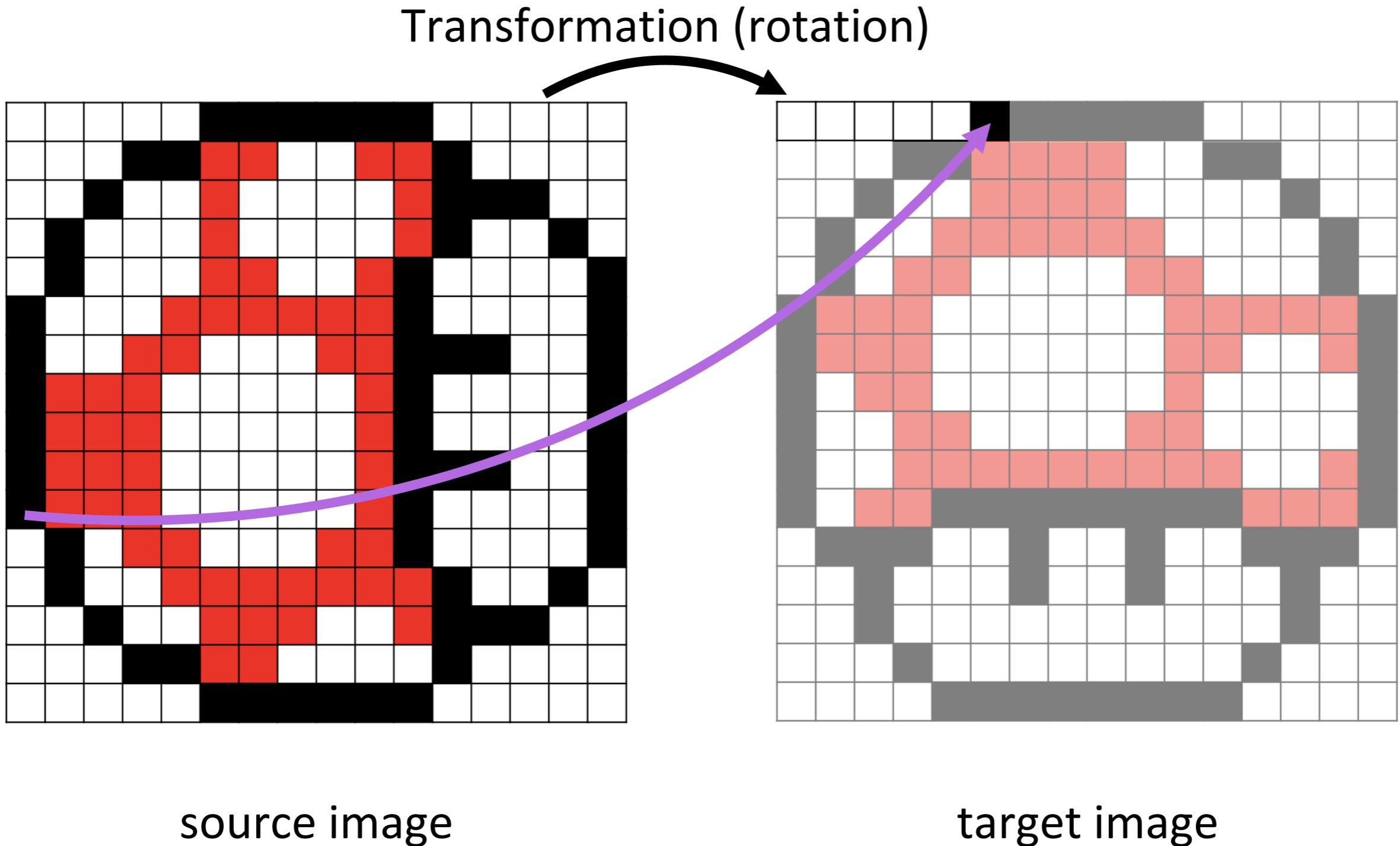
Transformation (rotation)



source image

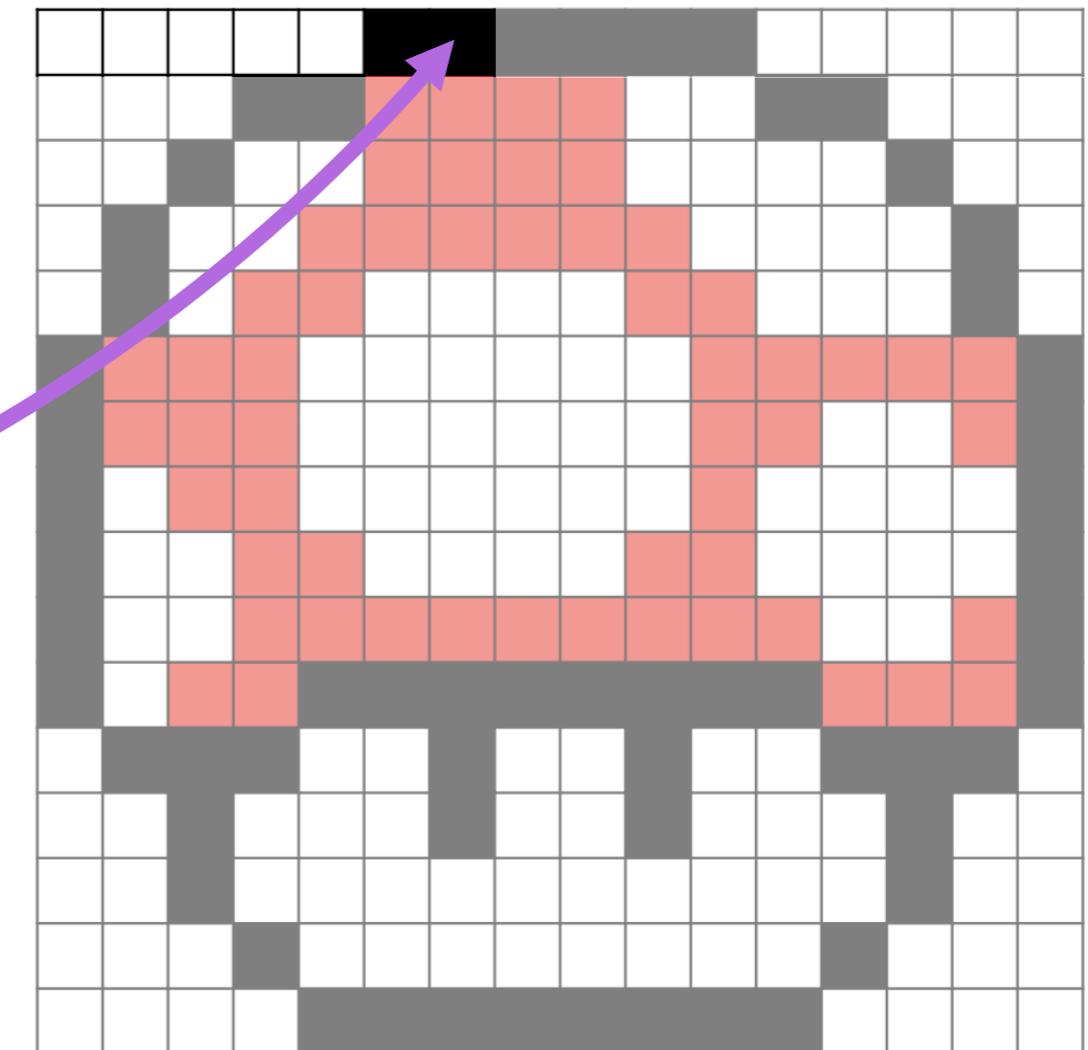
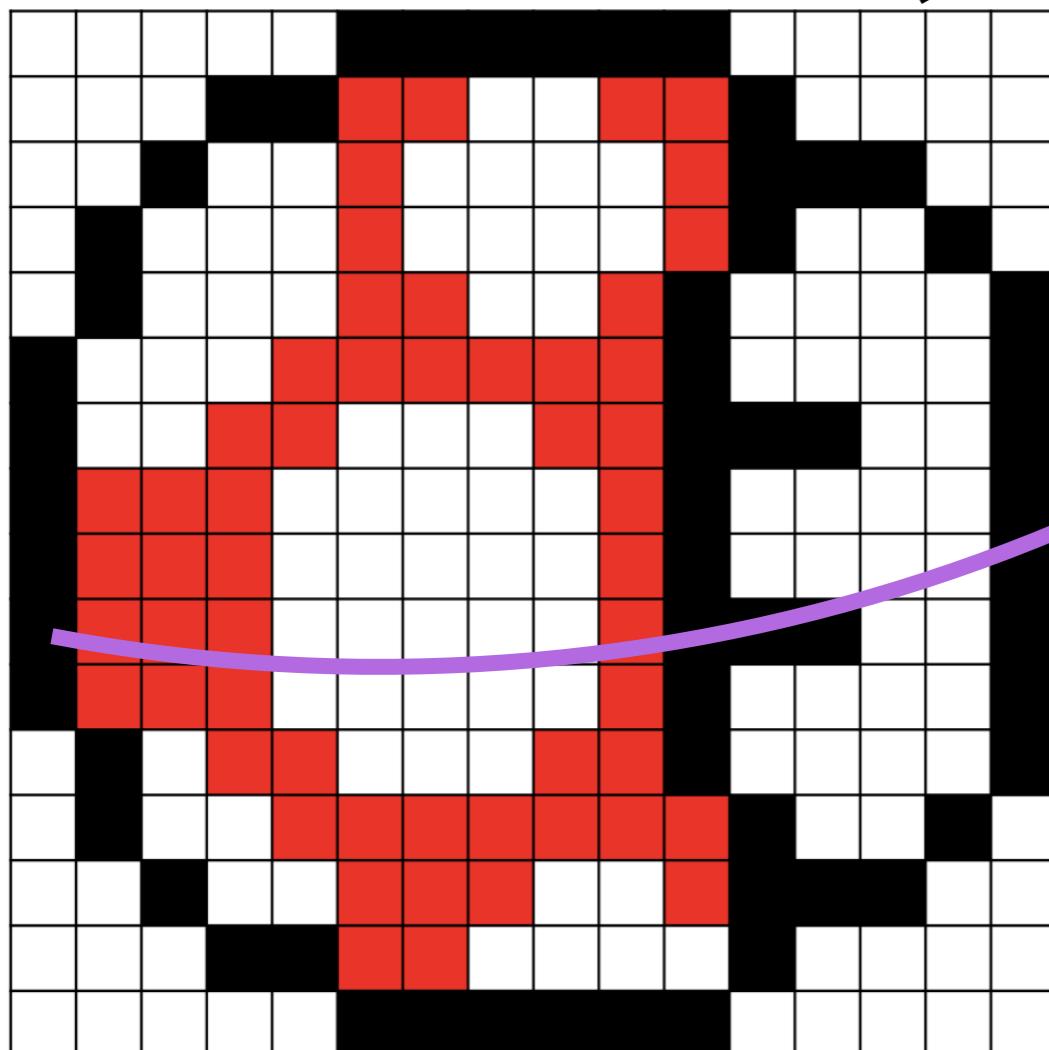
target image

A Simple Example



A Simple Example

Transformation (rotation)

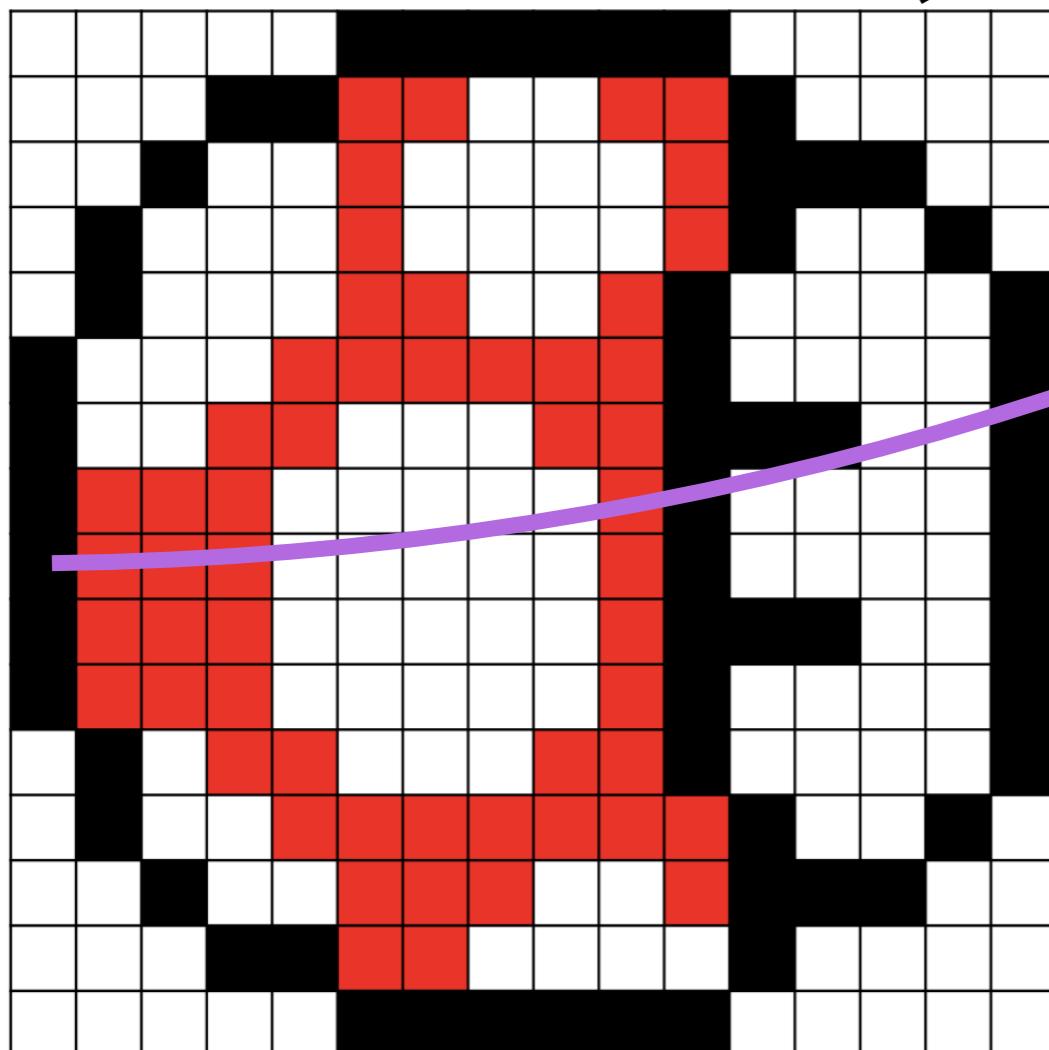


source image

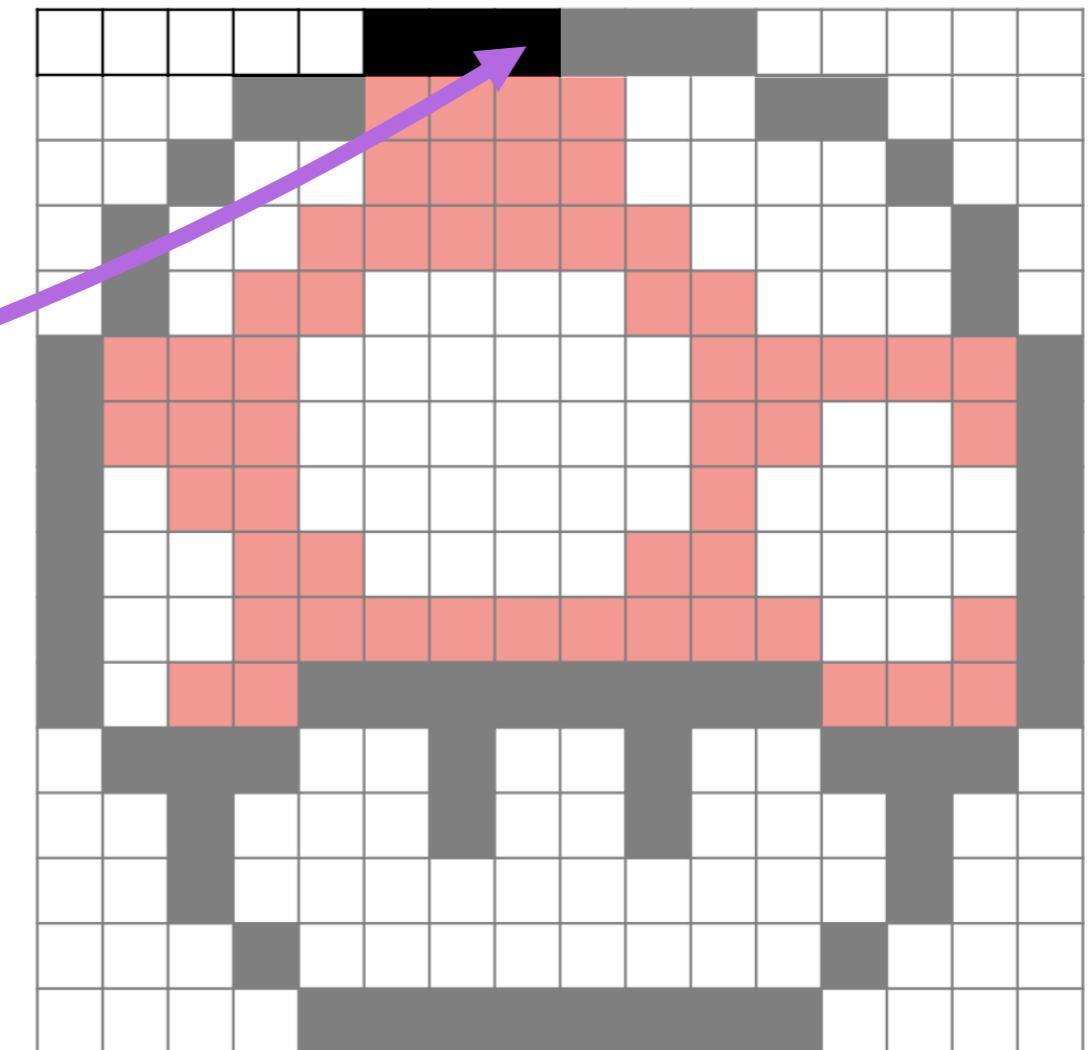
target image

A Simple Example

Transformation (rotation)



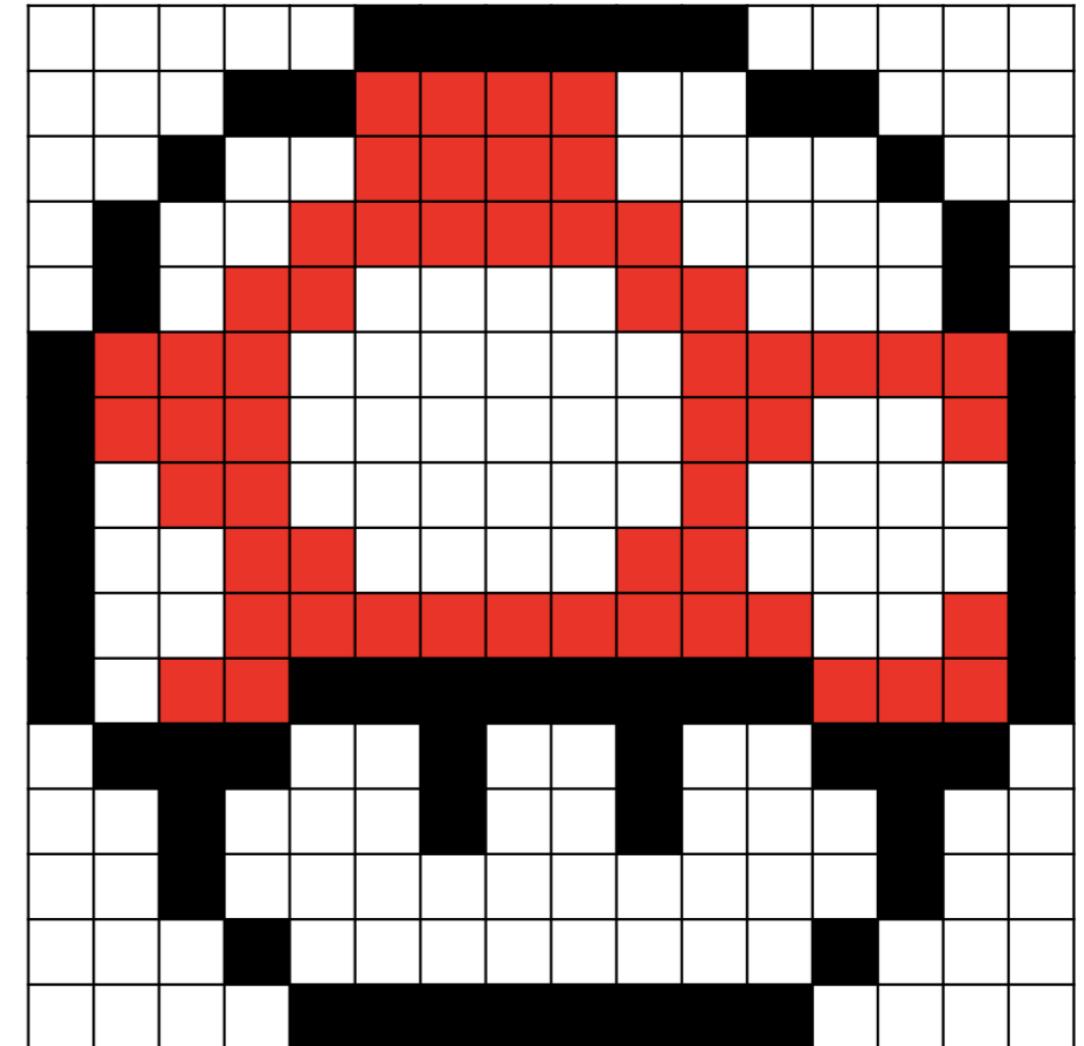
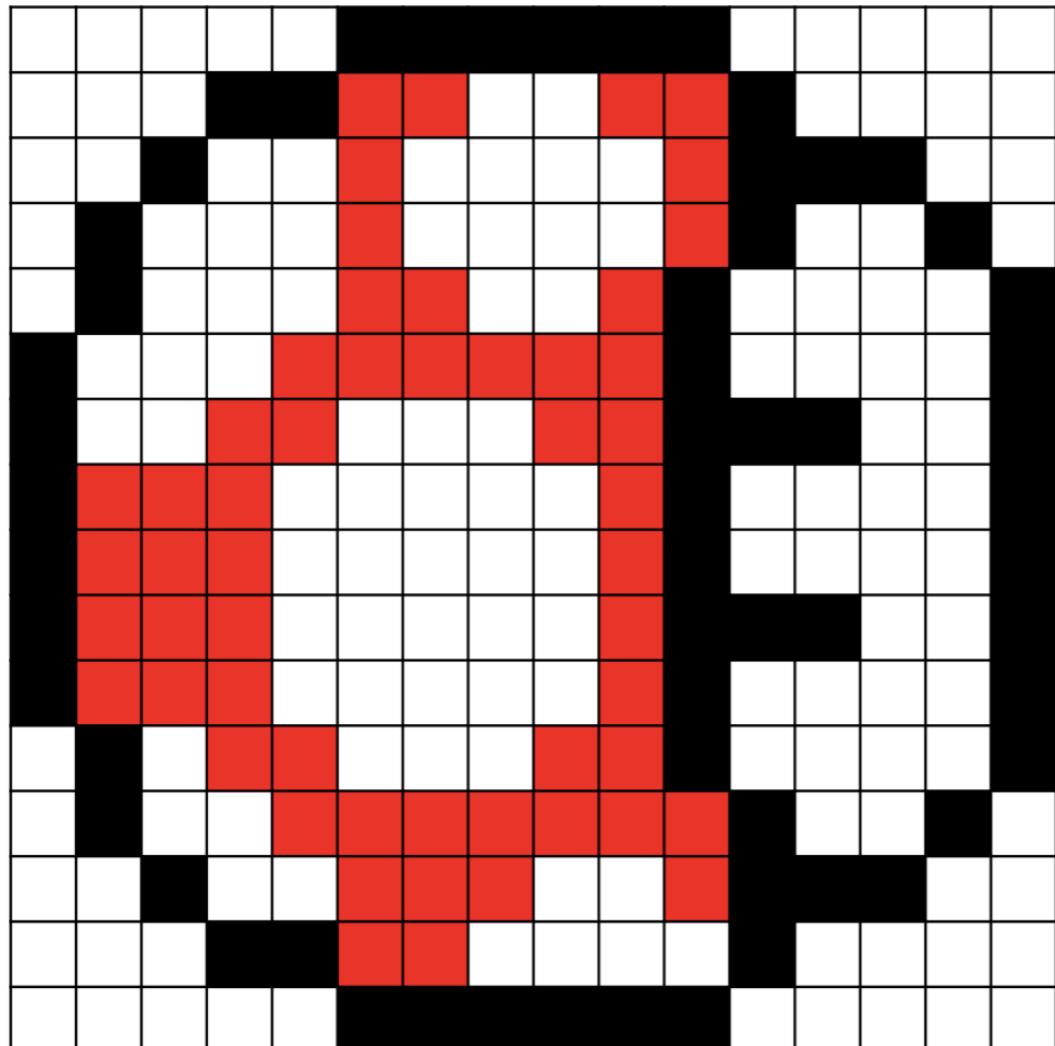
source image



target image

A Simple Example

Transformation (rotation)

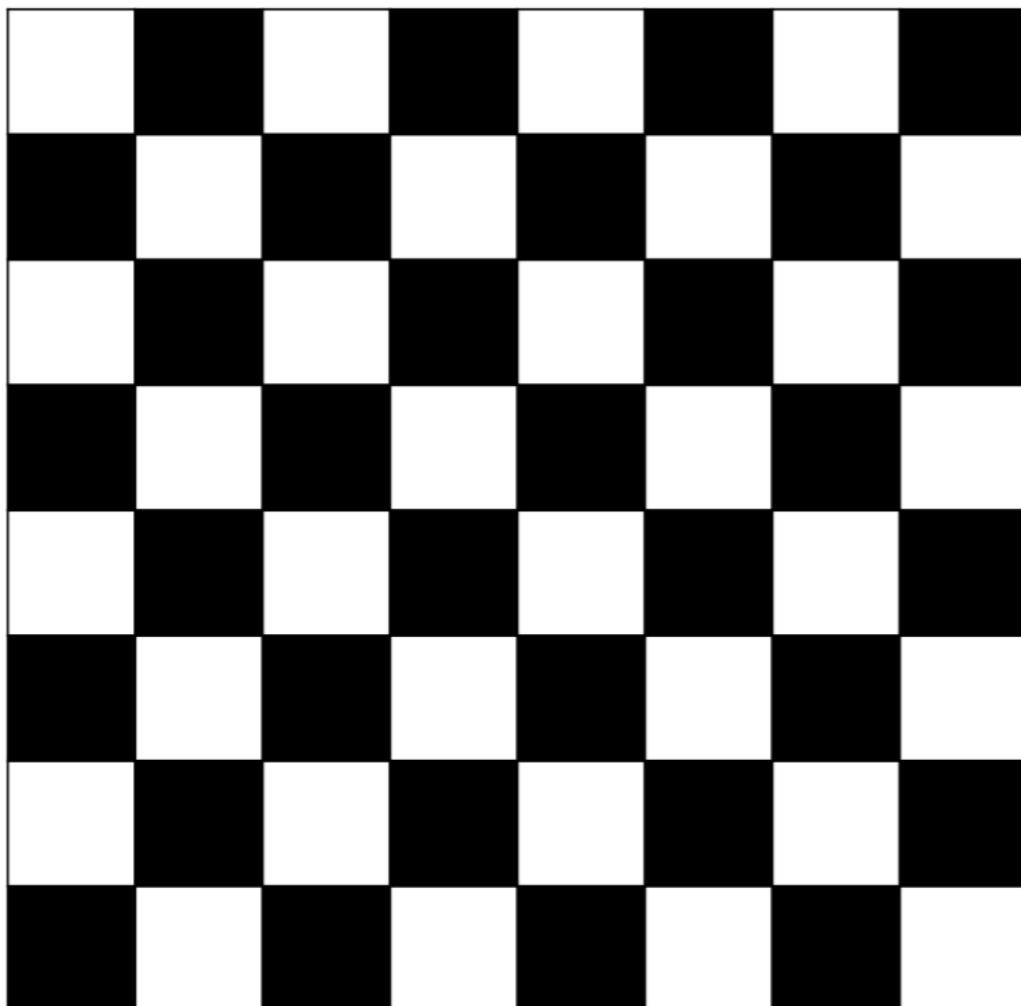


source image

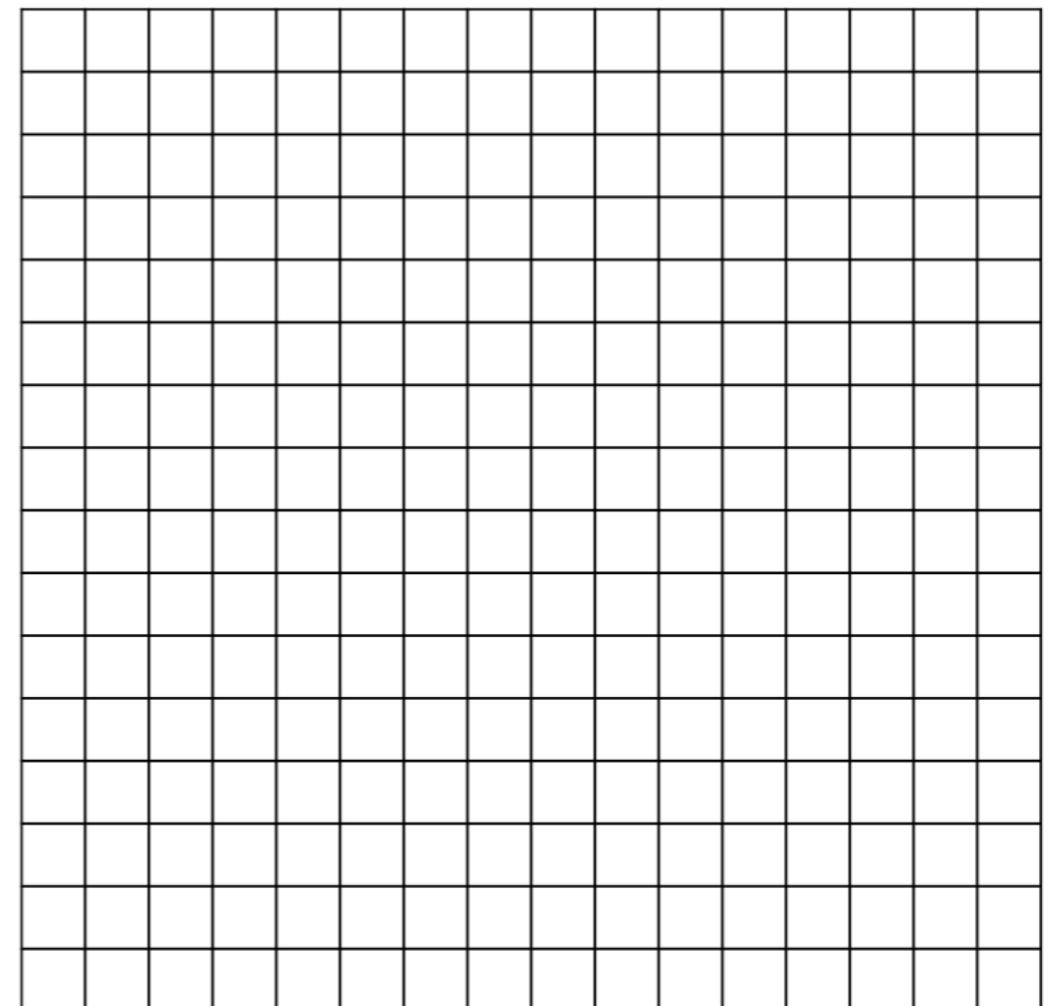
target image

Another Simple Example

Transformation (zoom)



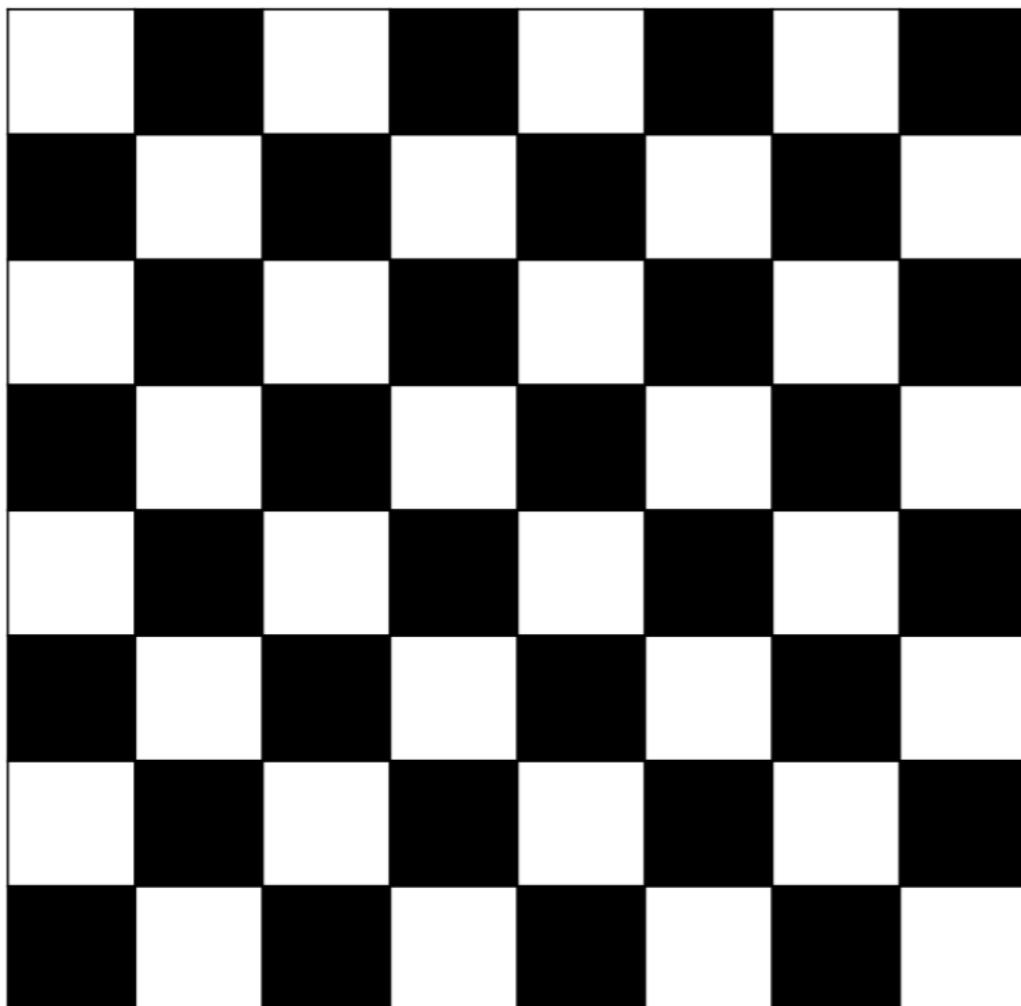
source image



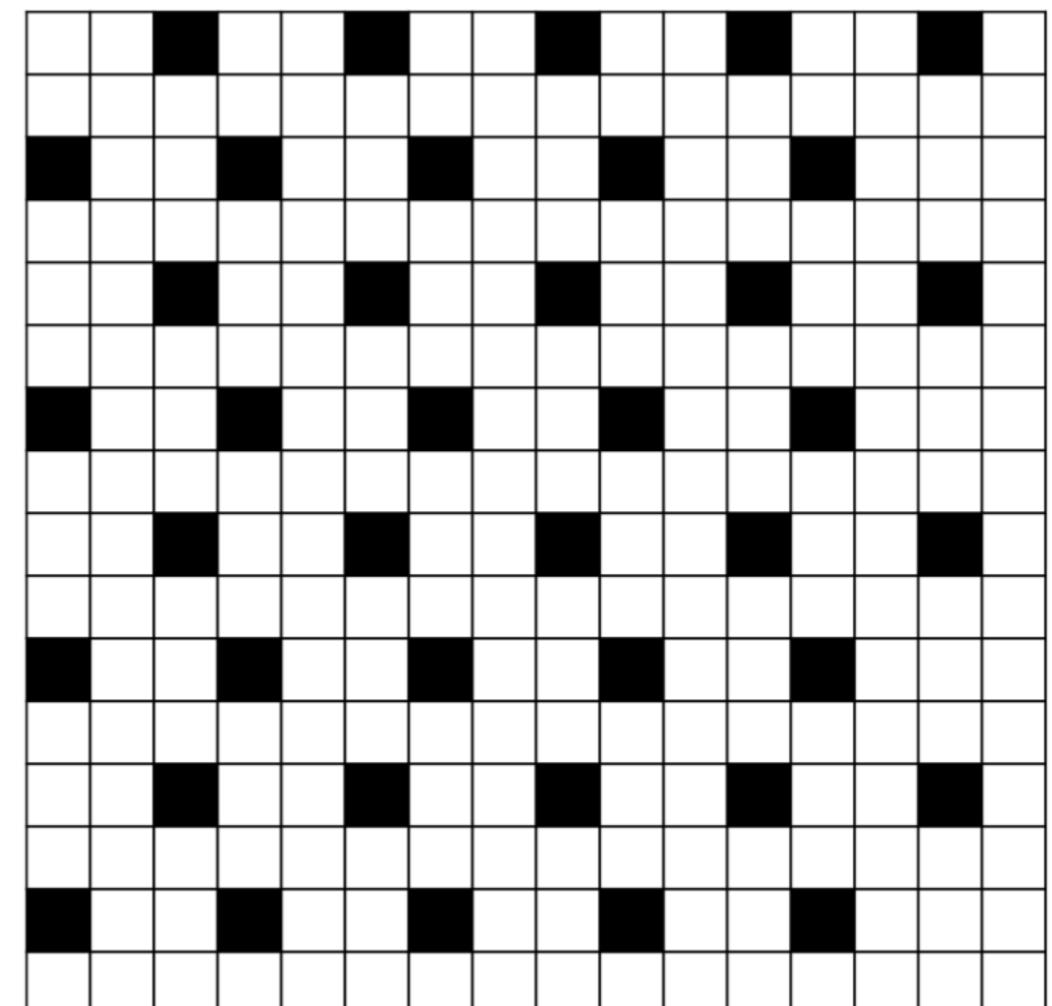
target image

Another Simple Example

Transformation (zoom)

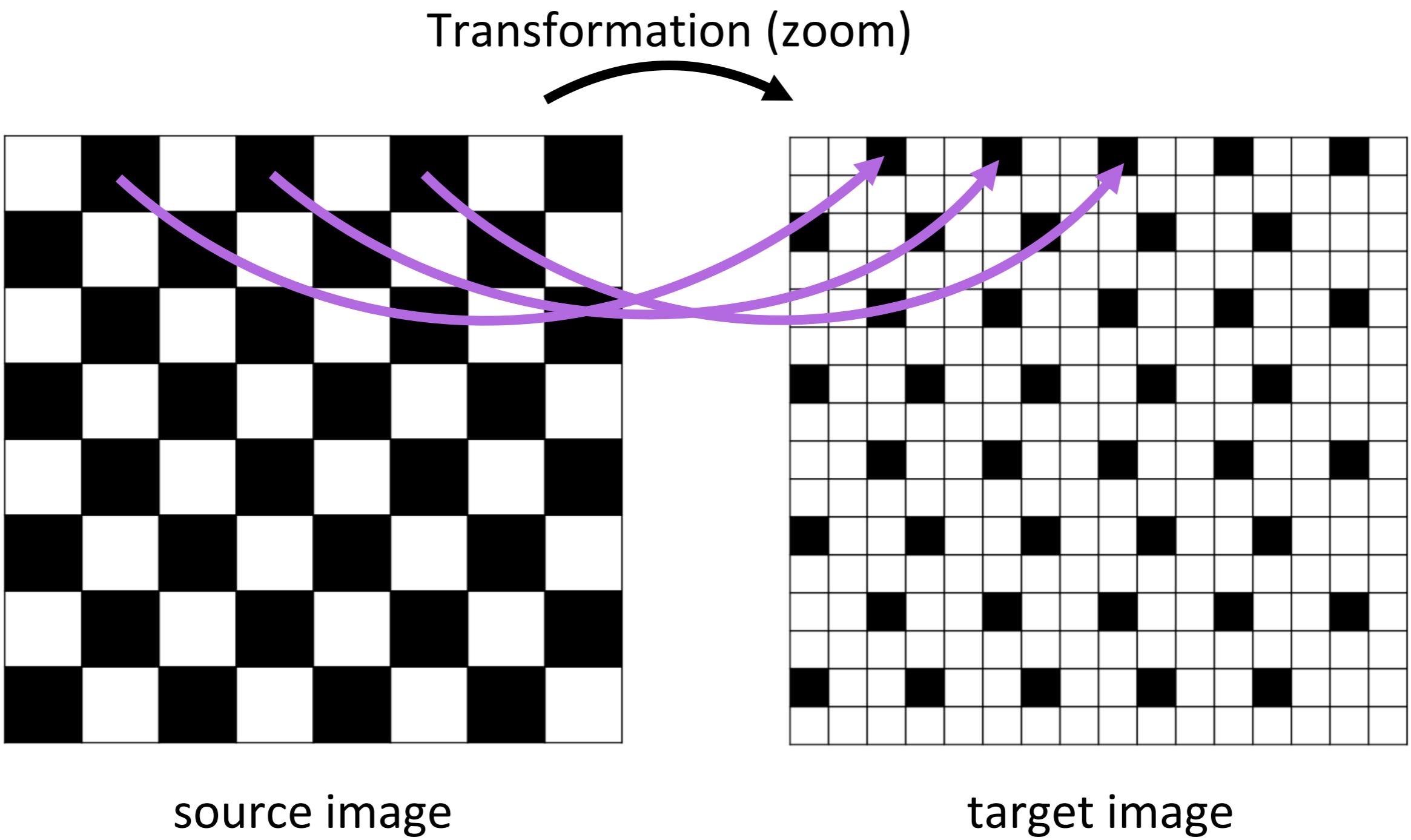


source image



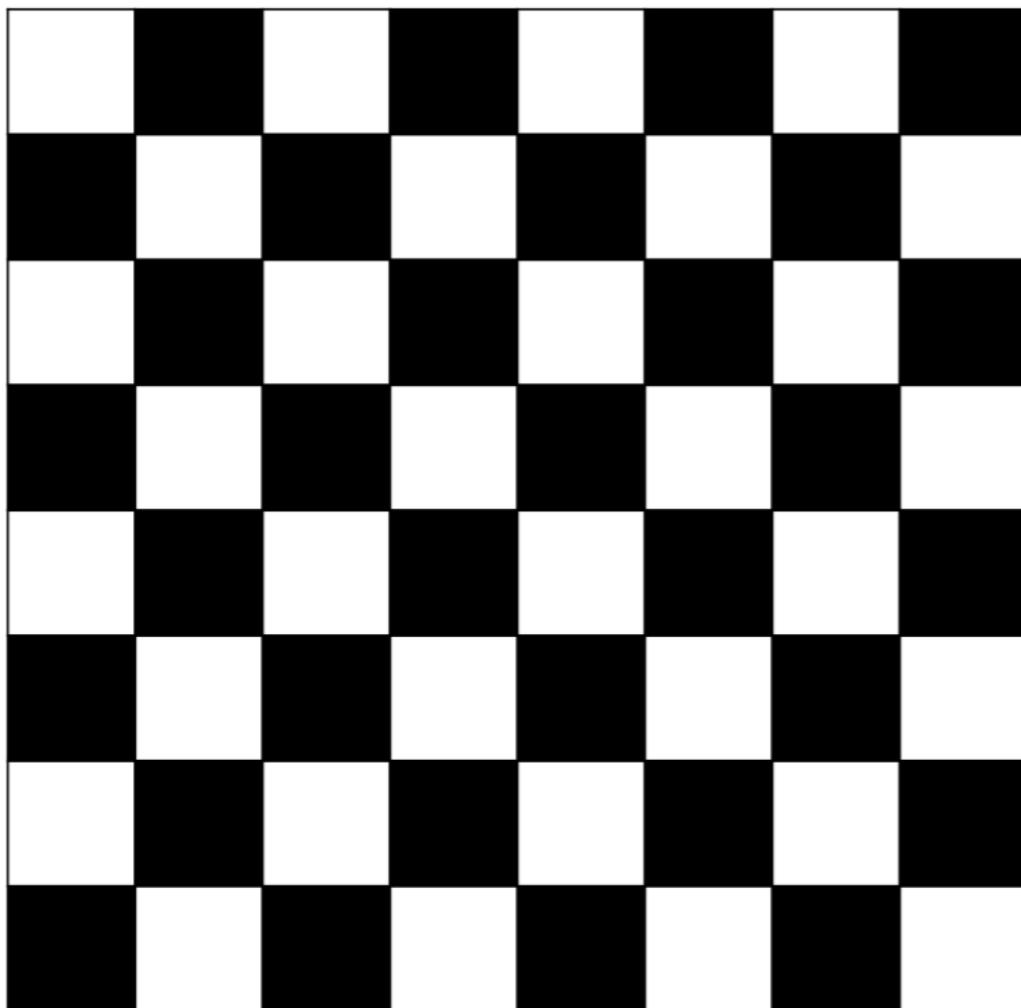
target image

Another Simple Example

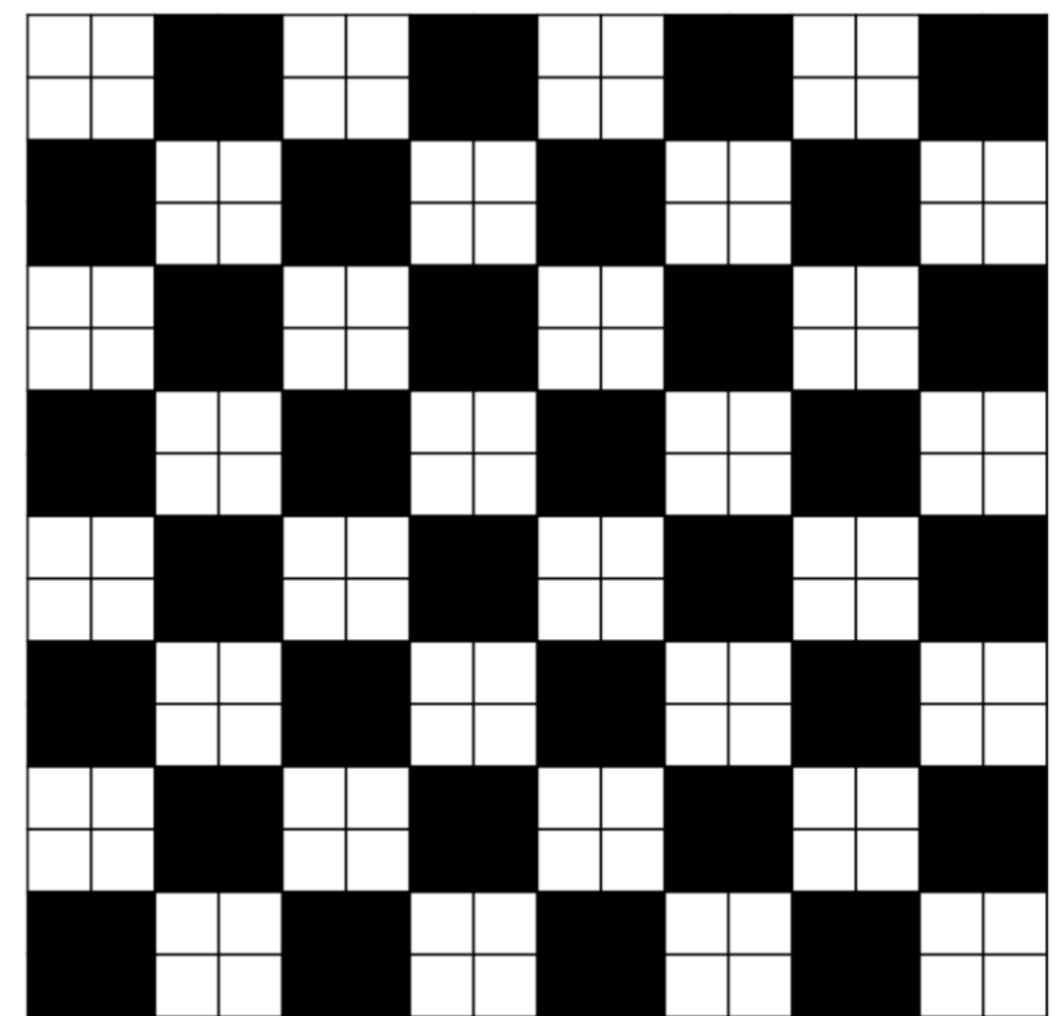


Another Simple Example

Transformation (zoom)



source image

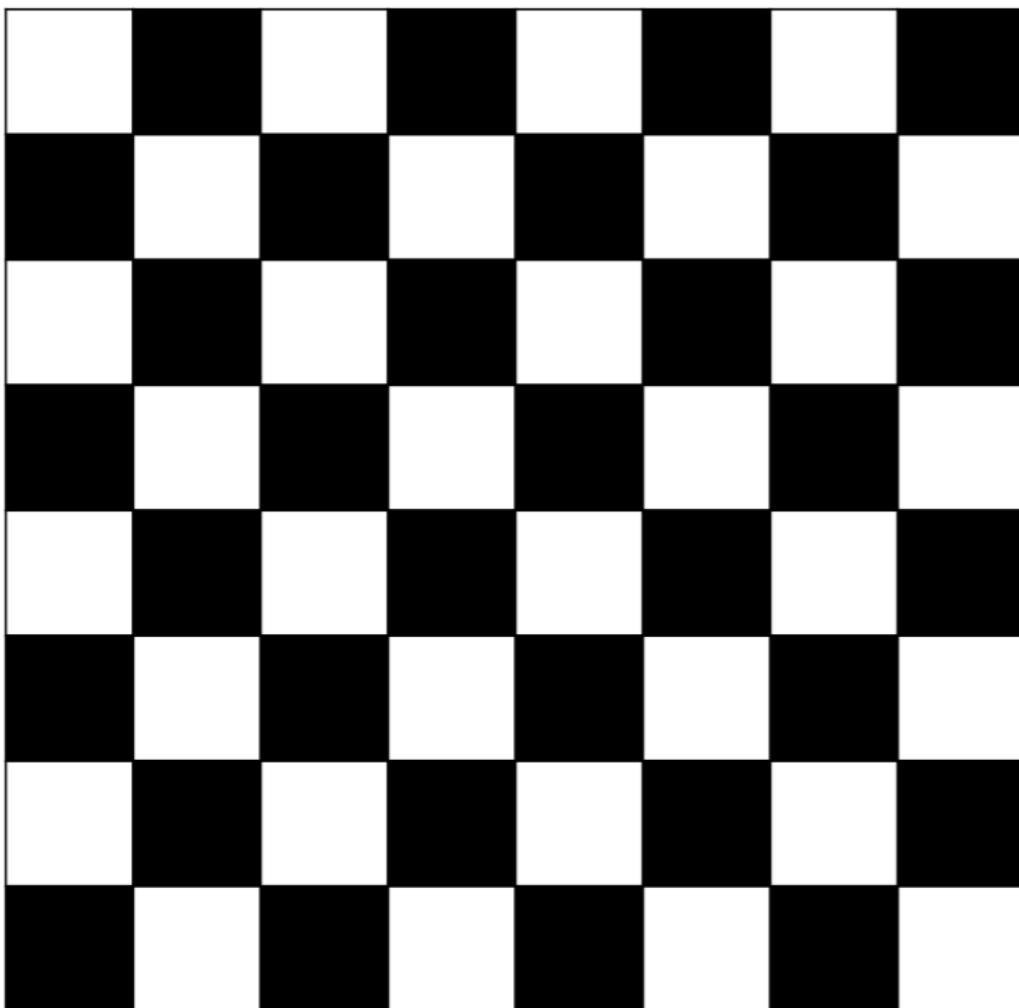


target image

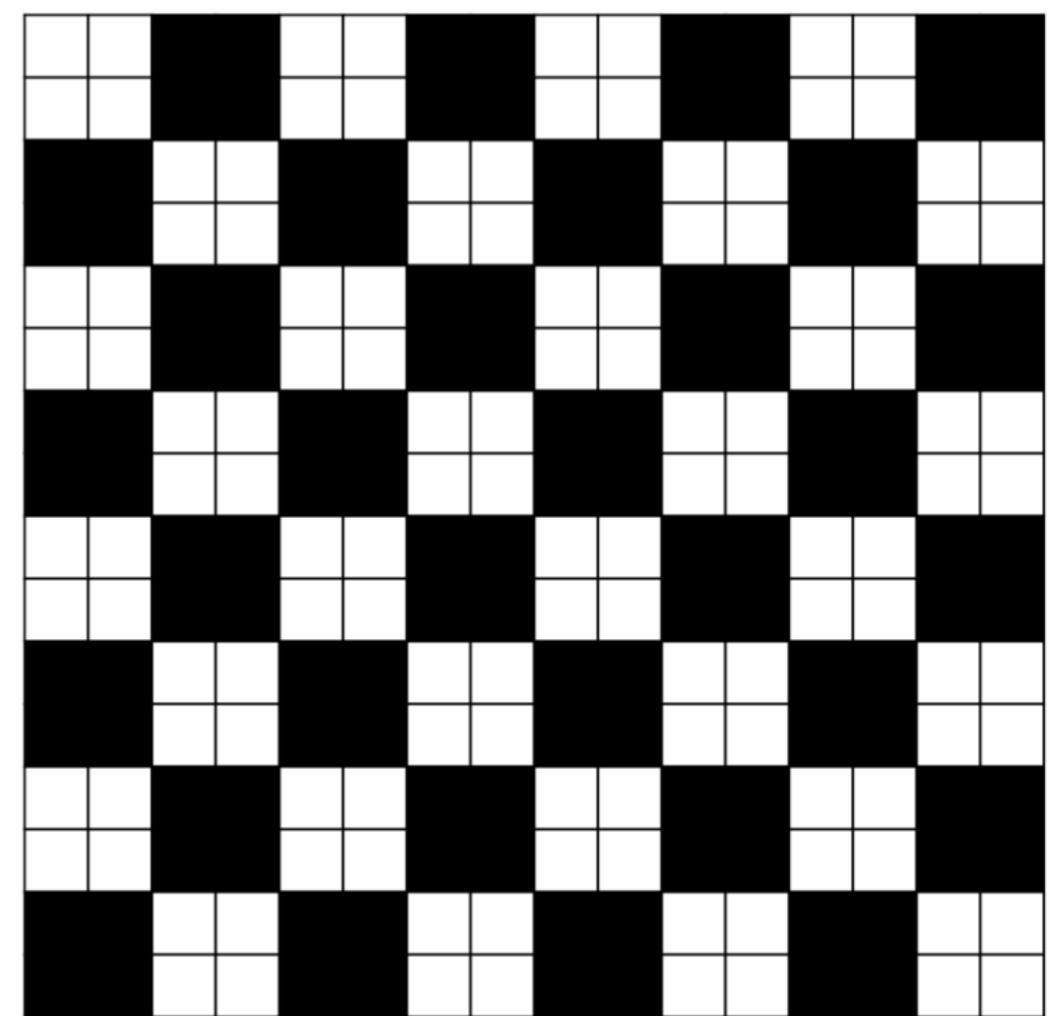
Problem: Multiple pixels in target correspond to one pixel in source

Another Simple Example

Transformation



source image



target image

Another Simple Example

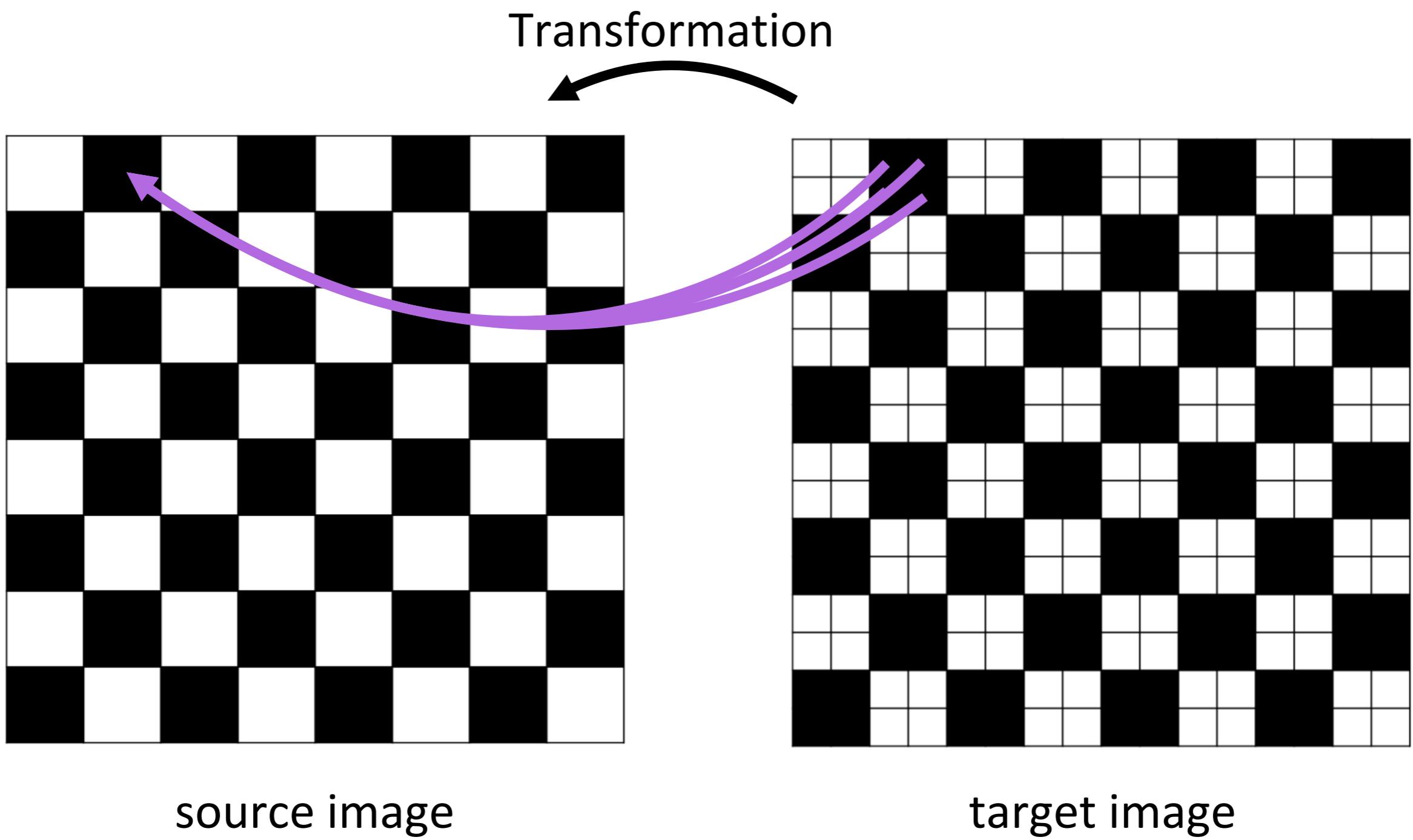


Image Warping

5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

1	3	4	1	2
3	4	5	5	5
2	3	3	2	1
1	2	4	5	5

target image

Image Warping

5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

warped image

Image Warping

5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

5				

warped image

Image Warping

5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

5	2			

warped image

Image Warping

5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

5	2	7		

warped image

Image Warping

5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

5	2	7	2	

warped image

Image Warping

5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

5	2	7	2	7

warped image

7 ← 8

Image Warping

0	0	0	0	0	0
5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

5	2	7	2	7
0				

warped image

Image Warping

0 0 0 0 0 0

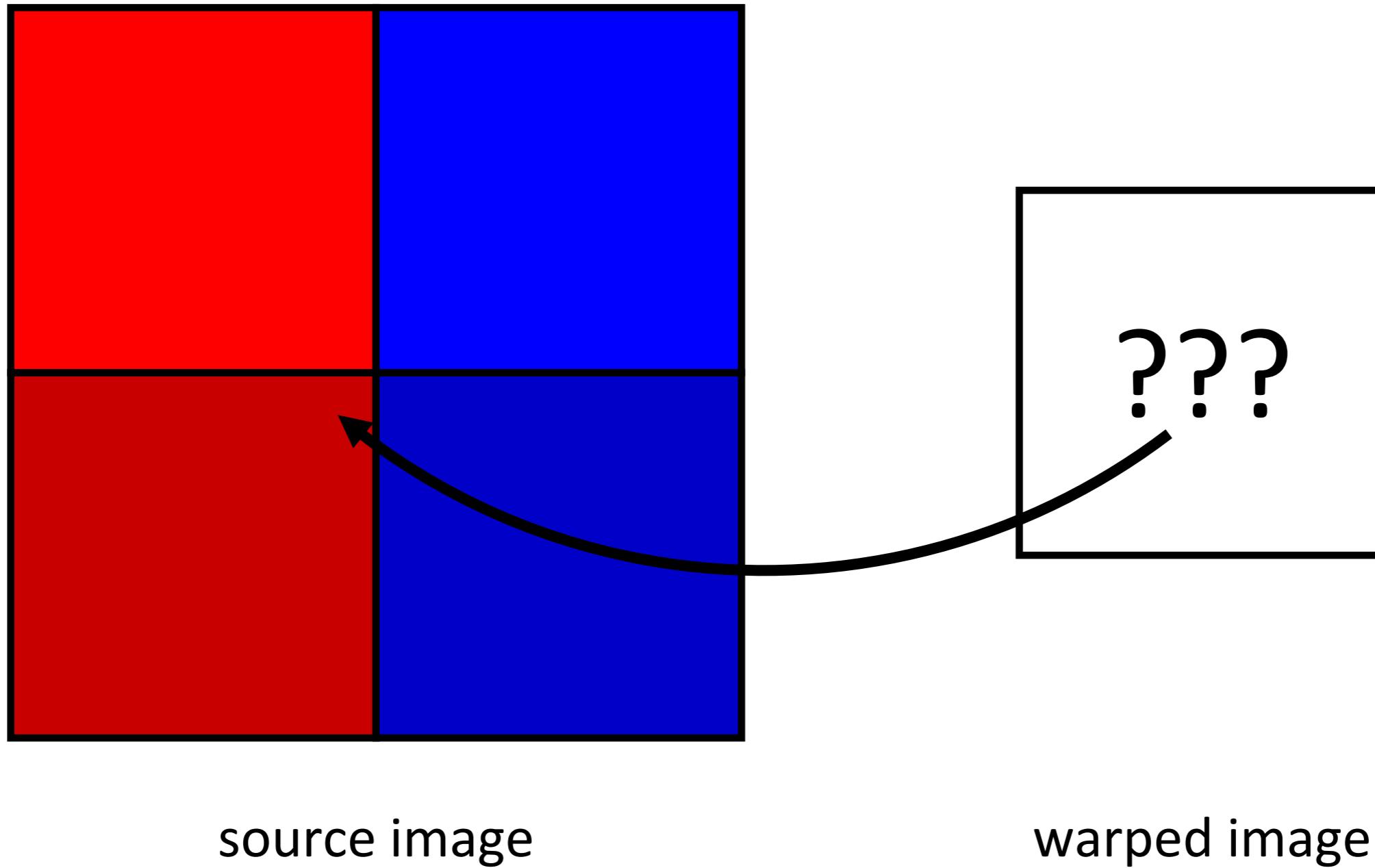
5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

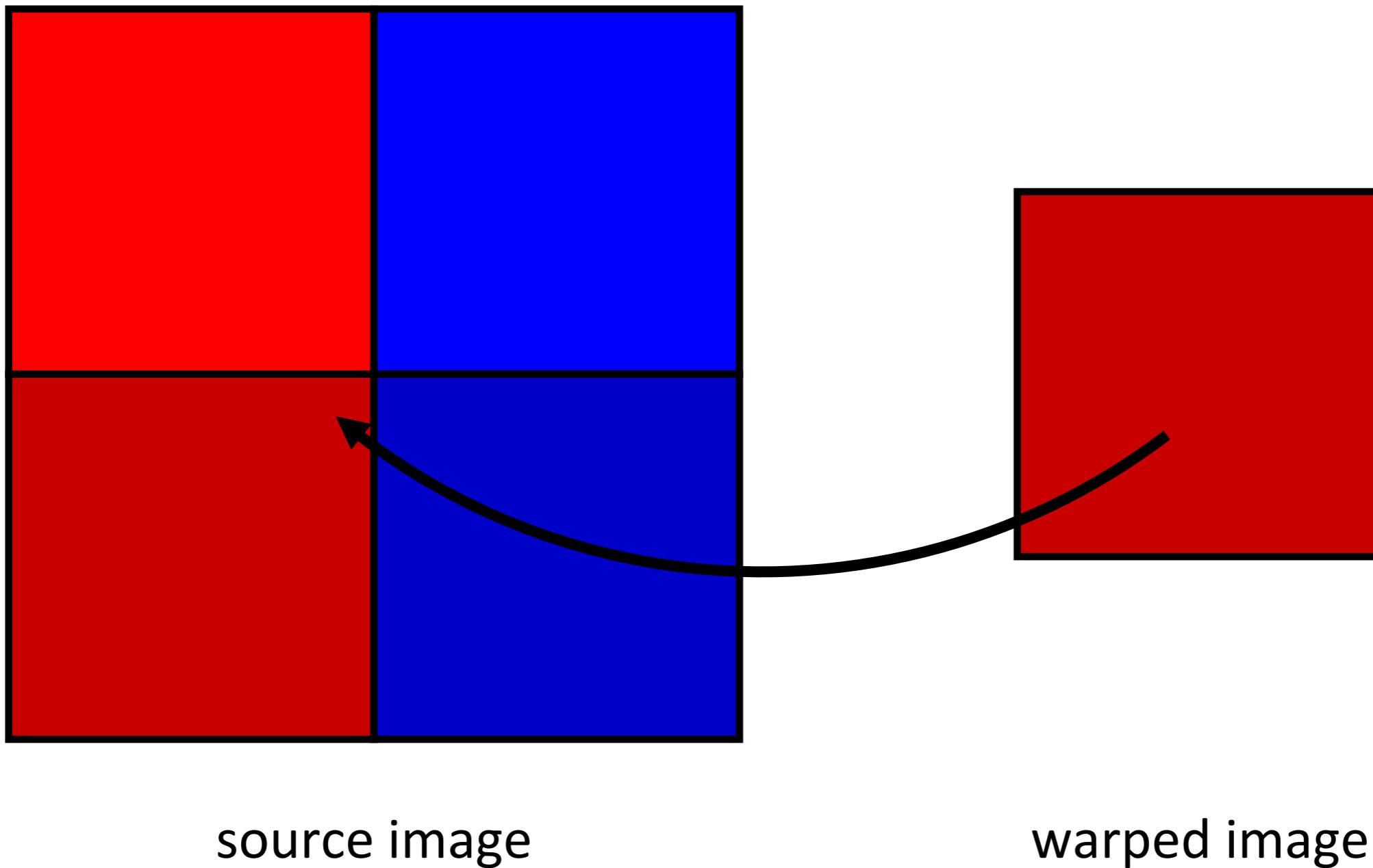
5	2	7	2	7
0	3			

warped image

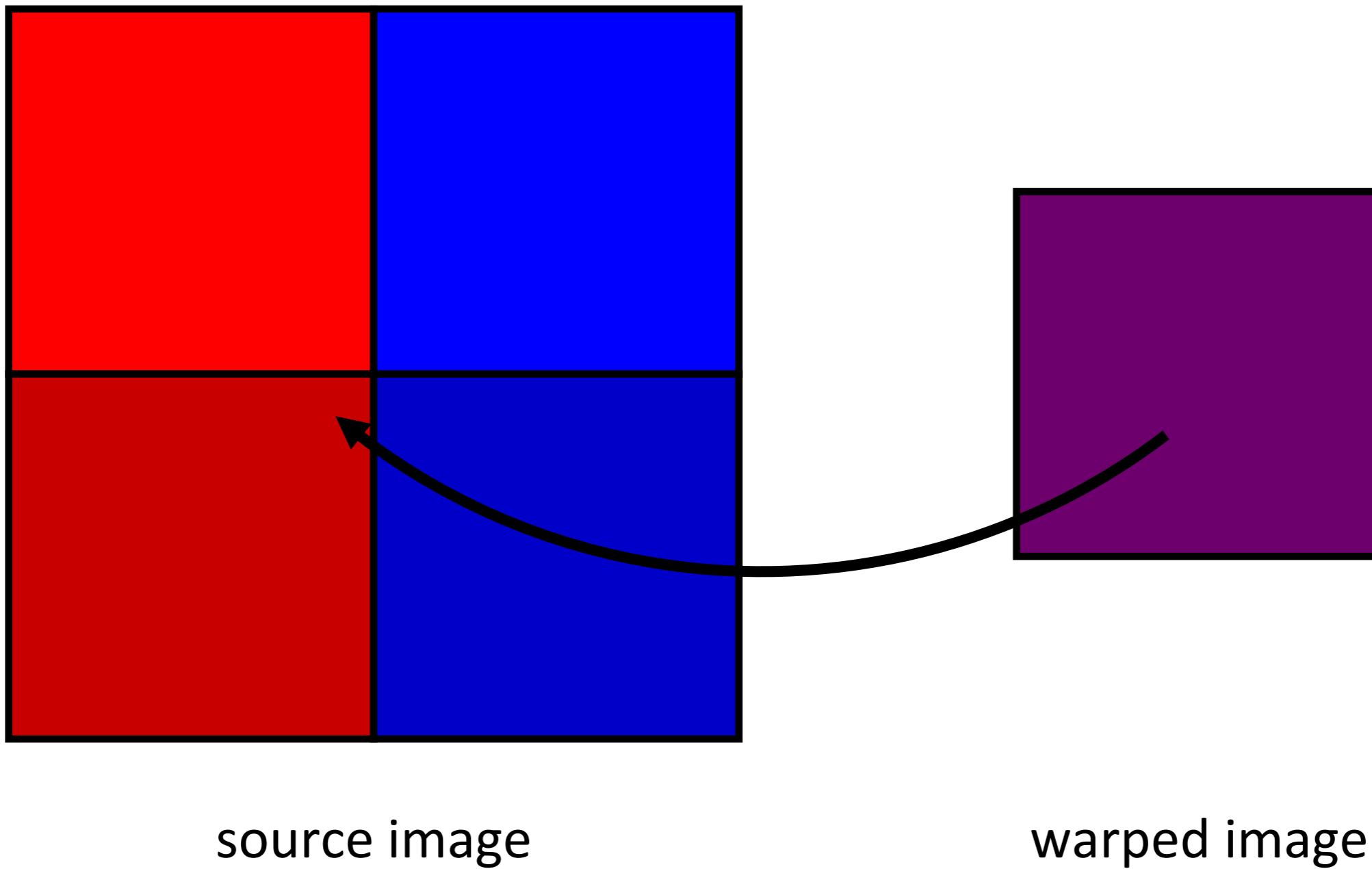
Pixel Interpolation



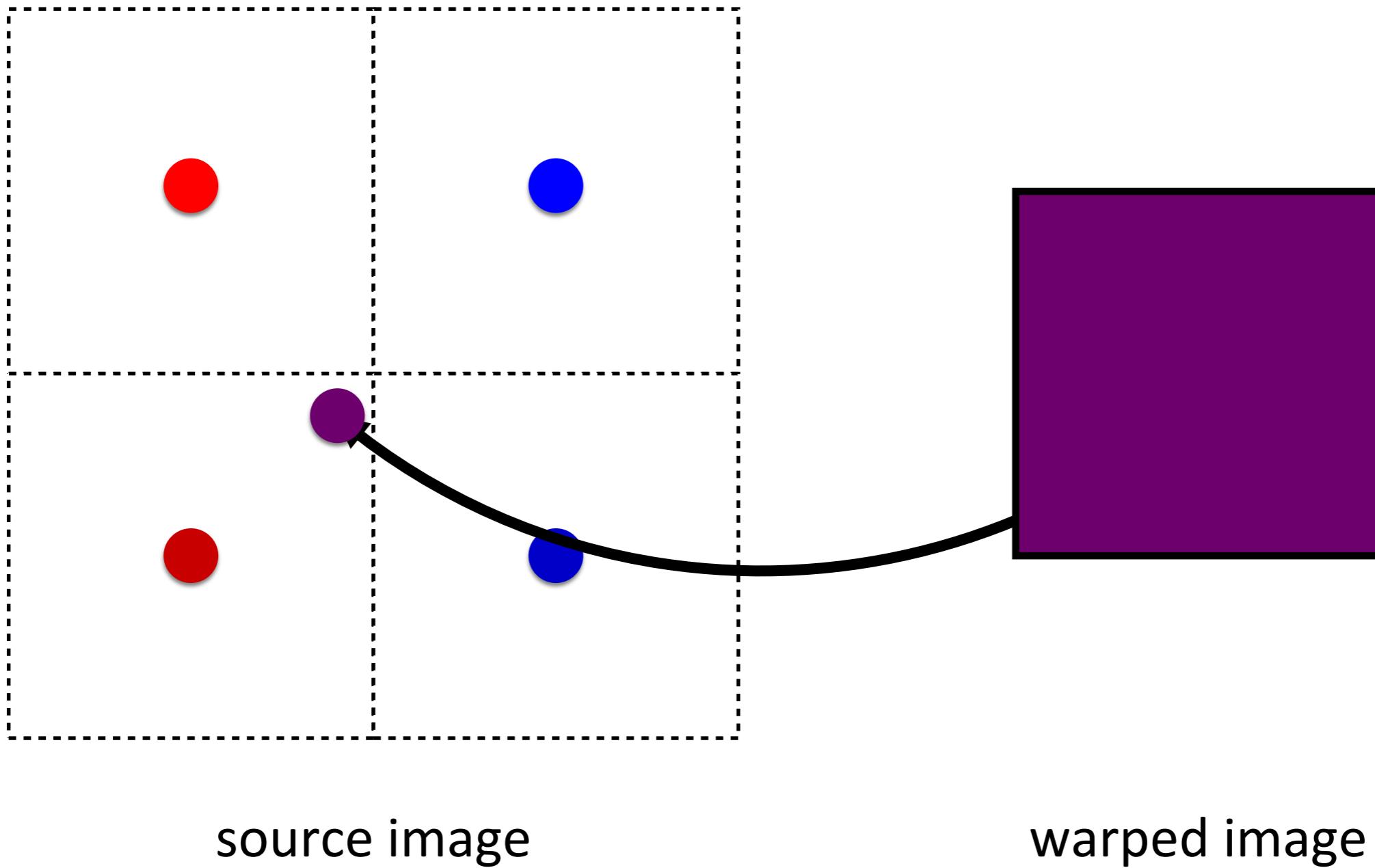
Nearest Neighbor Interpolation



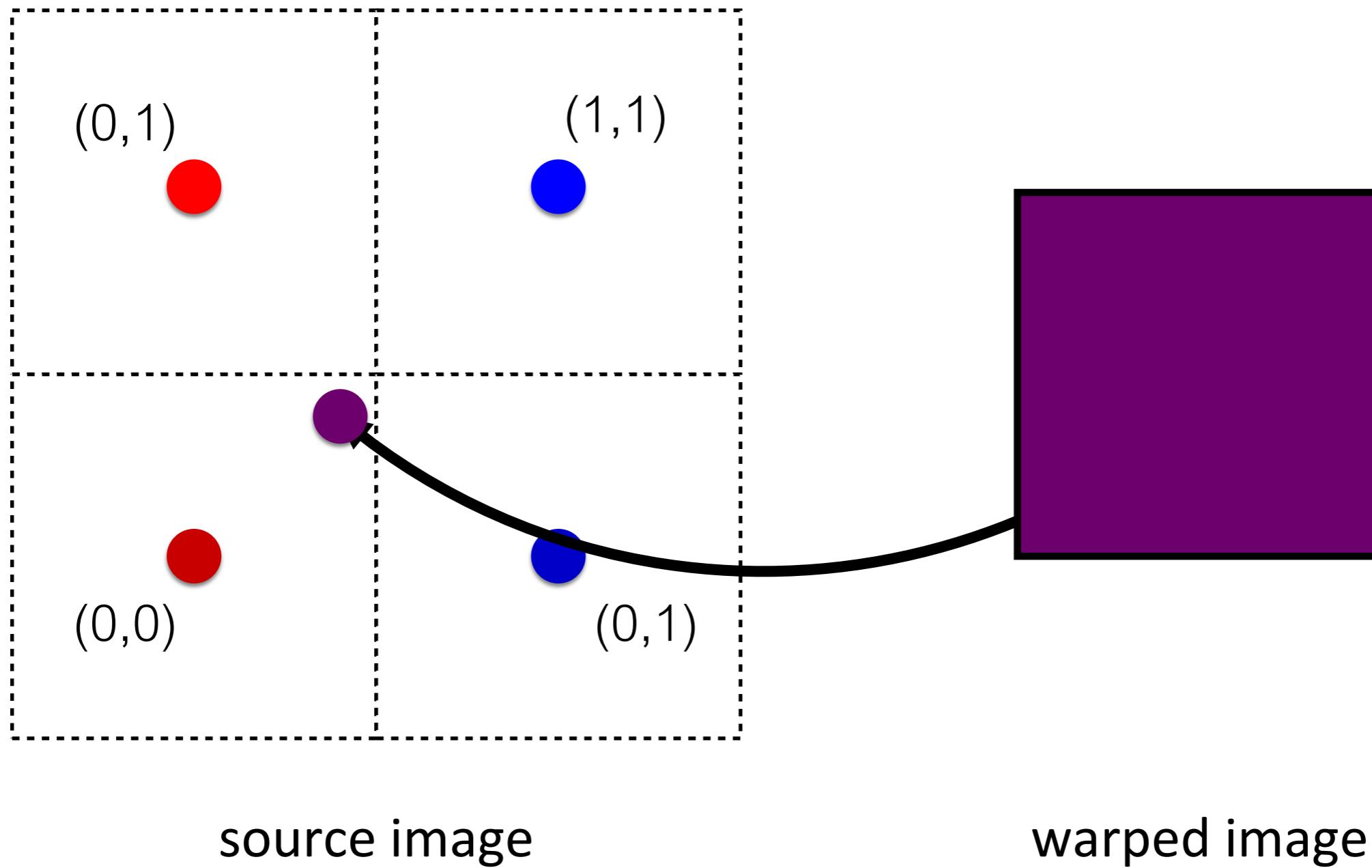
Bilinear Interpolation



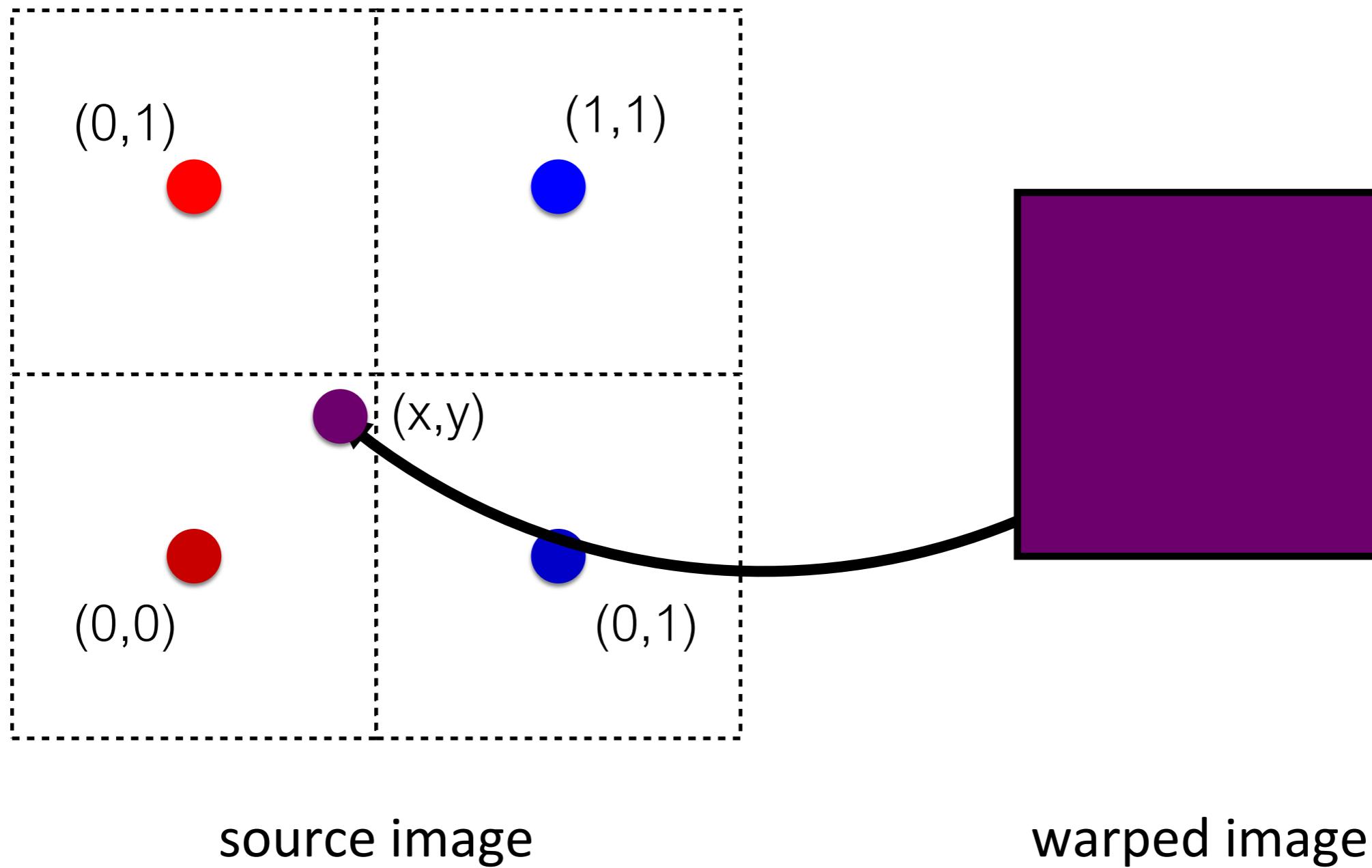
Bilinear Interpolation



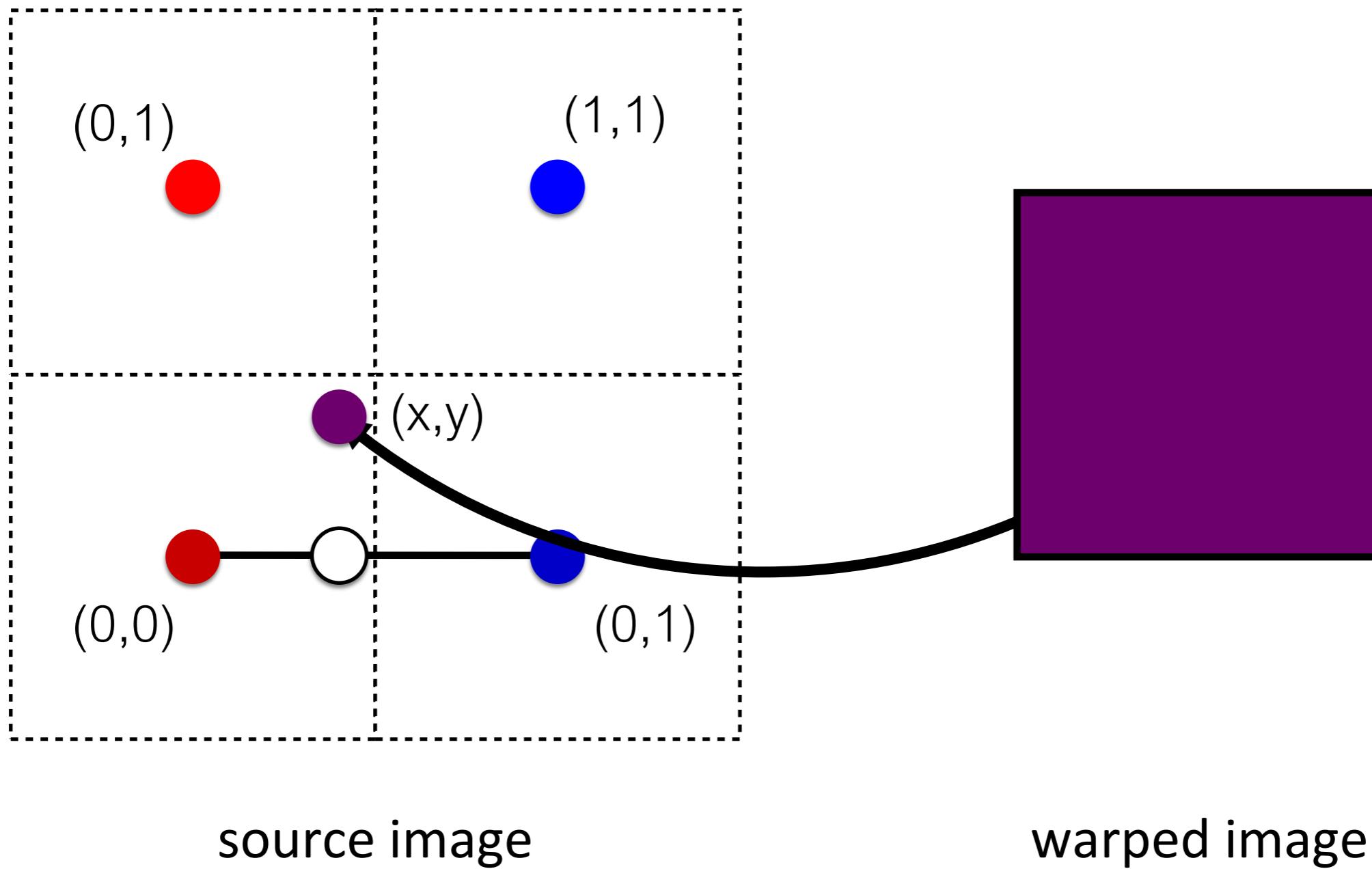
Bilinear Interpolation



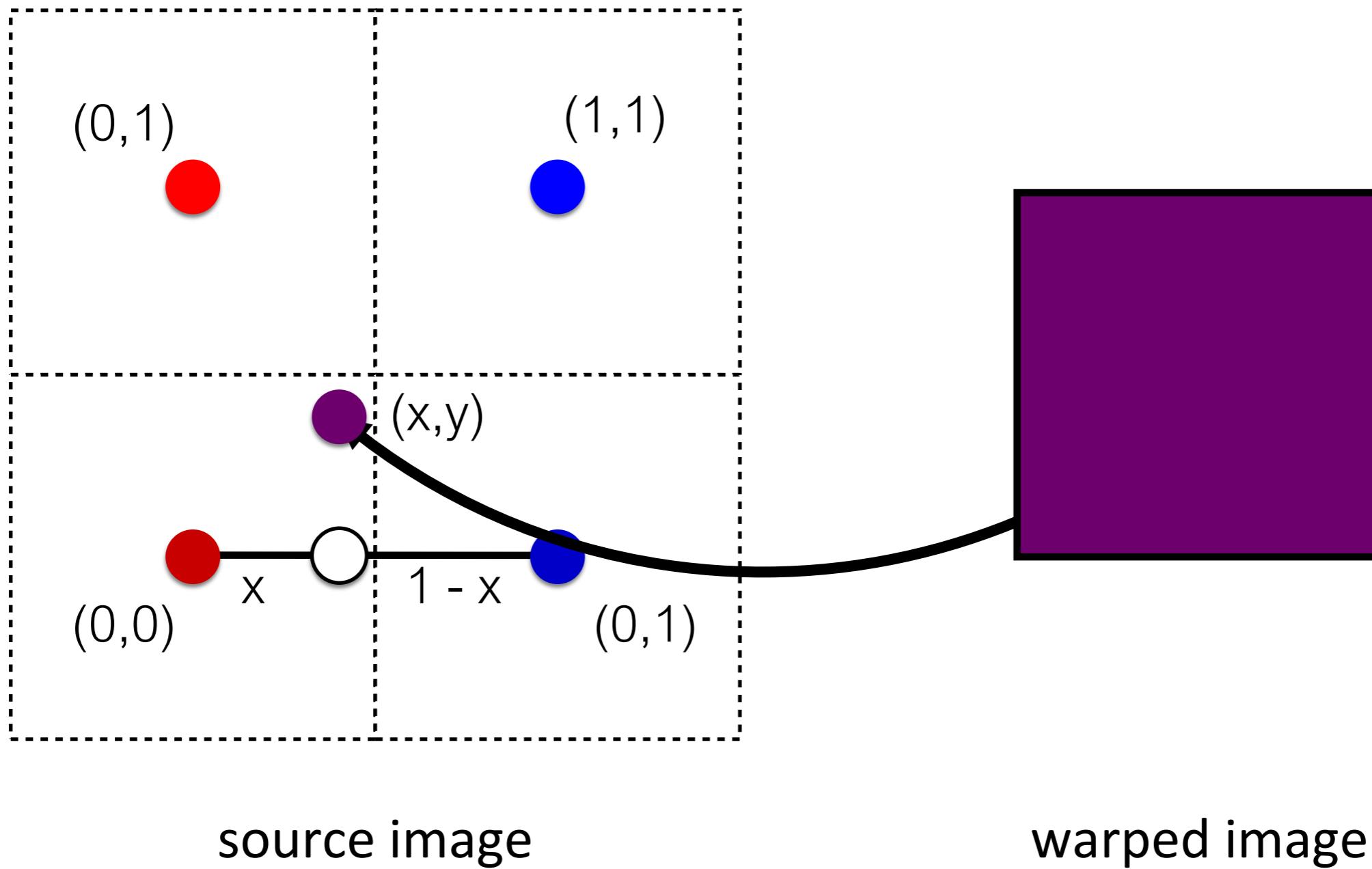
Bilinear Interpolation



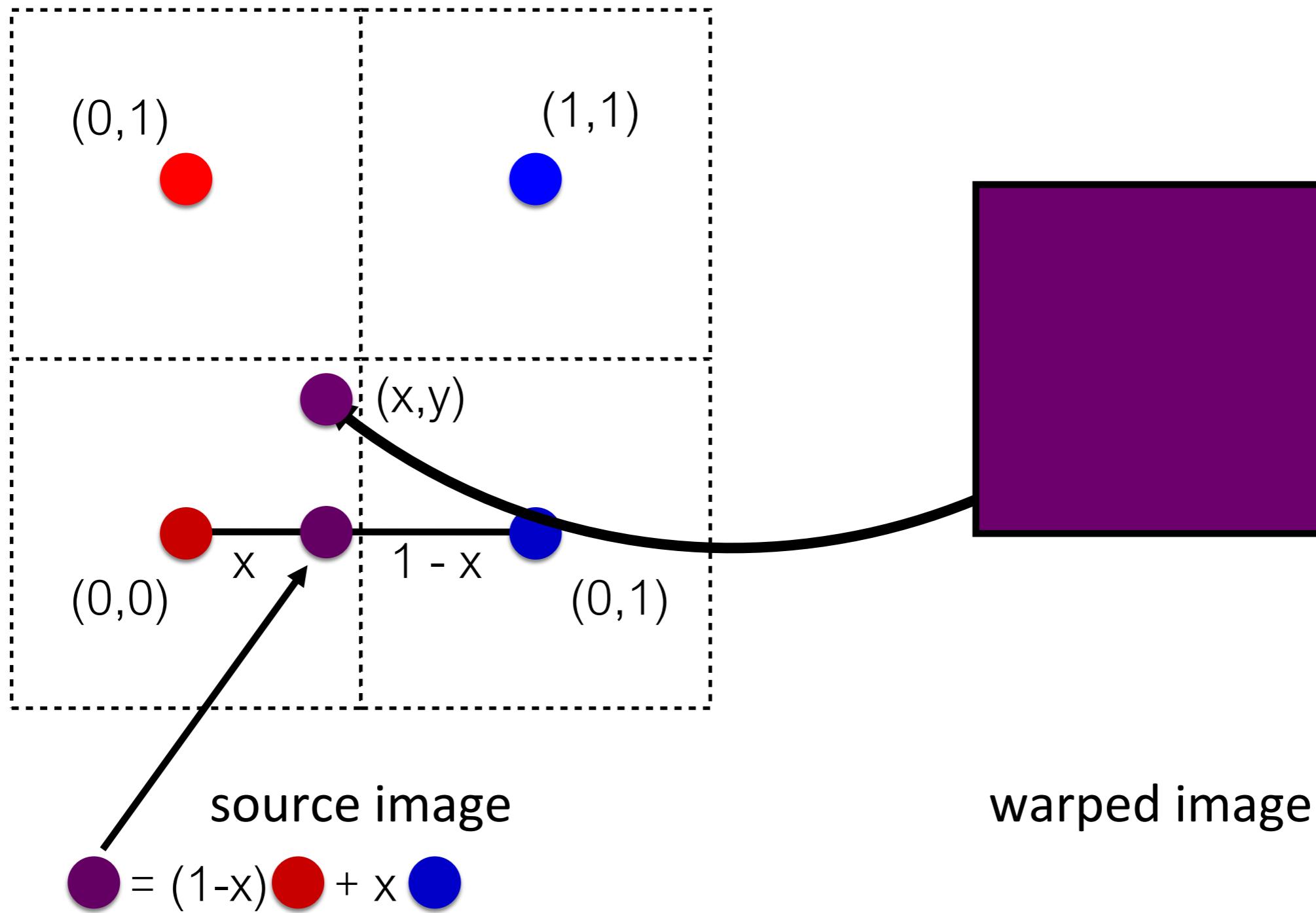
Bilinear Interpolation



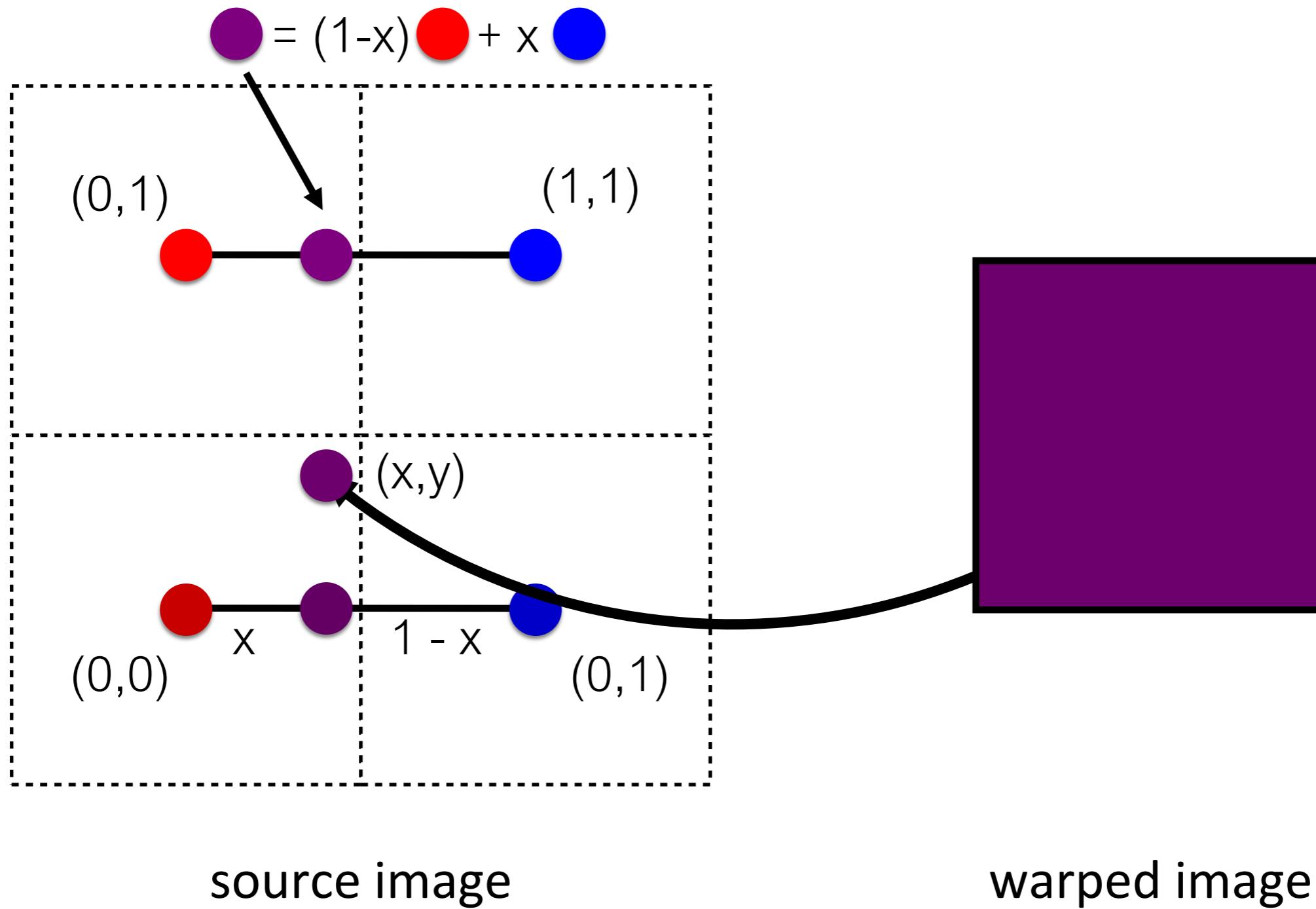
Bilinear Interpolation



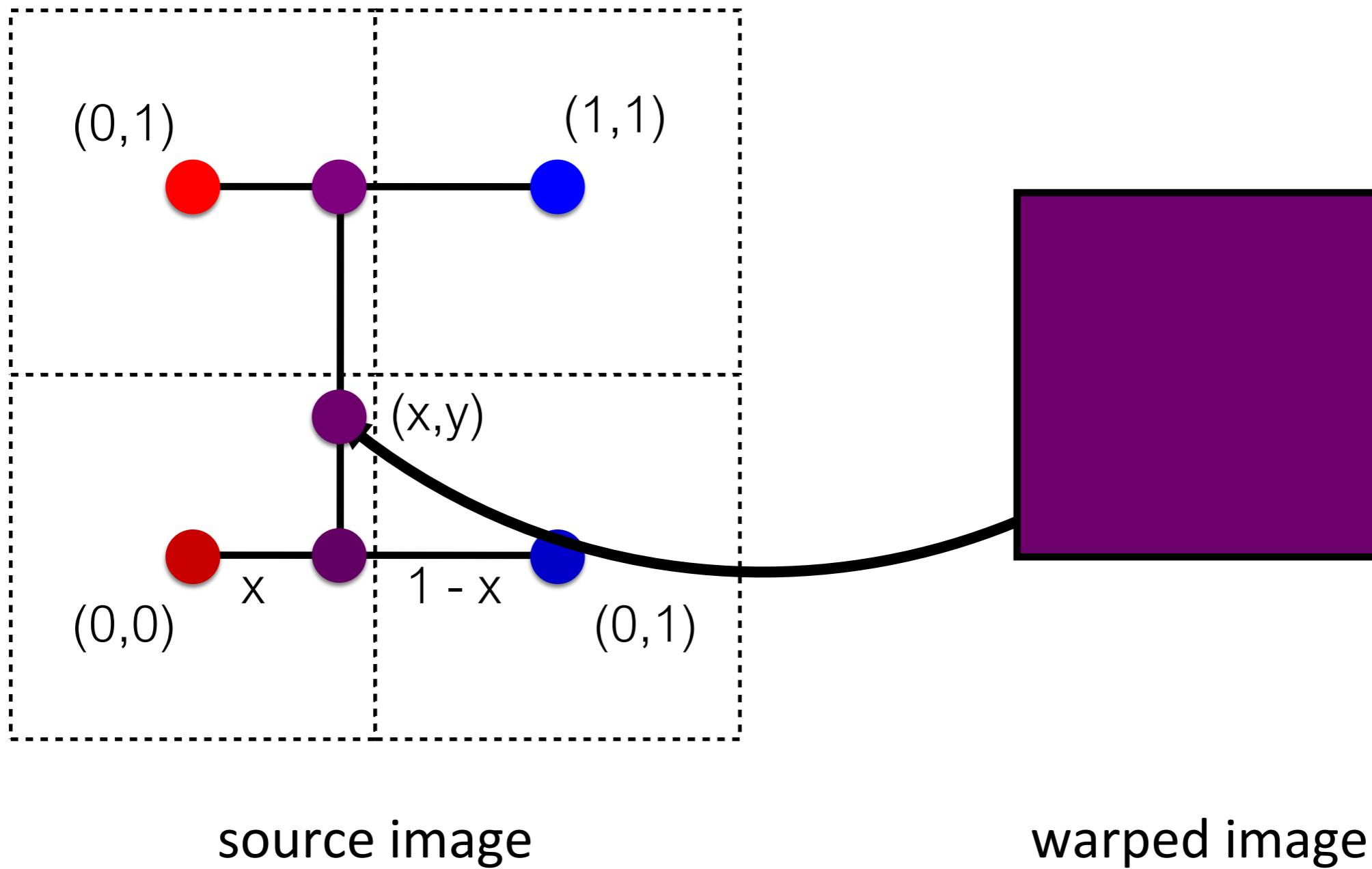
Bilinear Interpolation



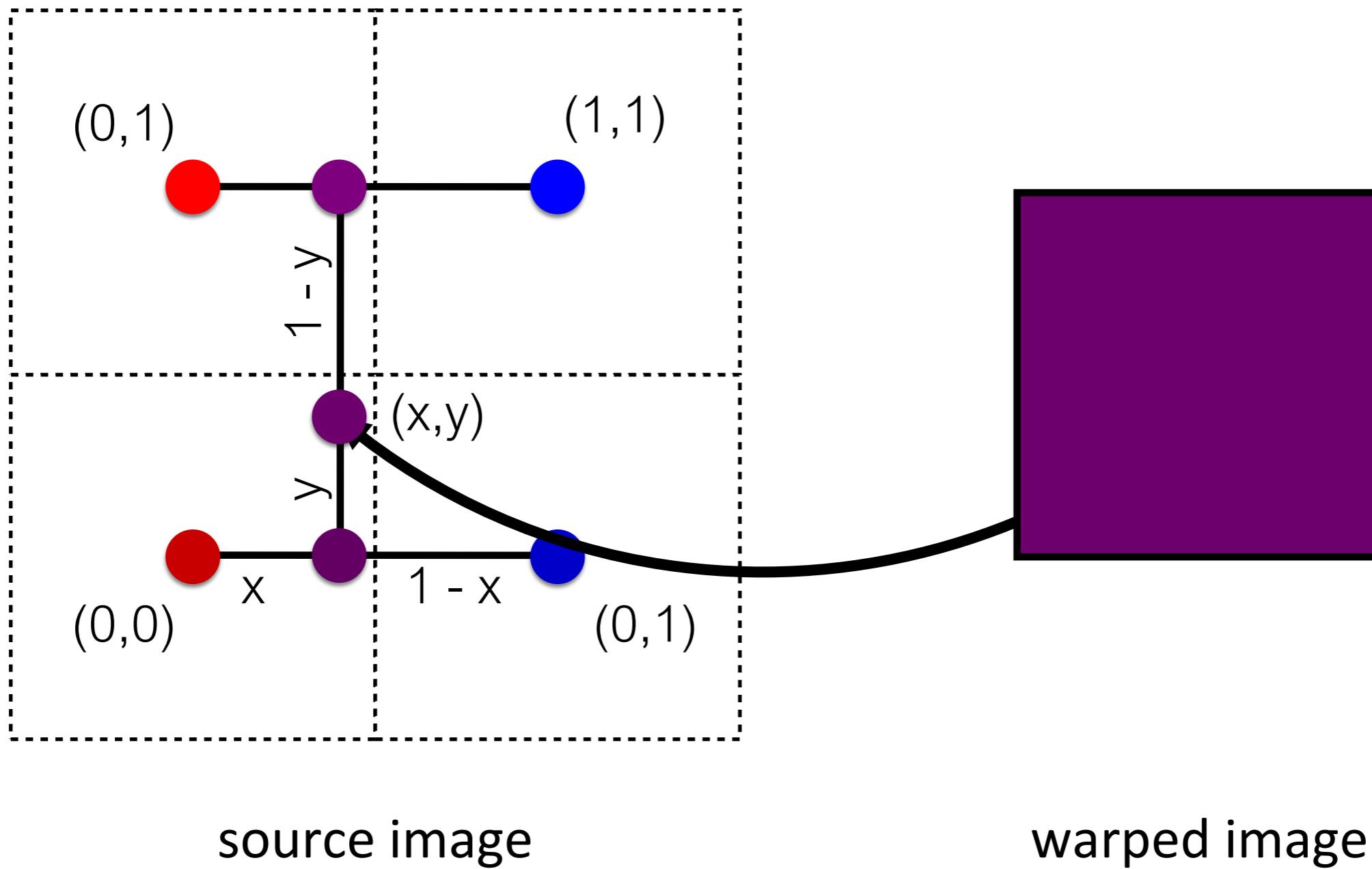
Bilinear Interpolation



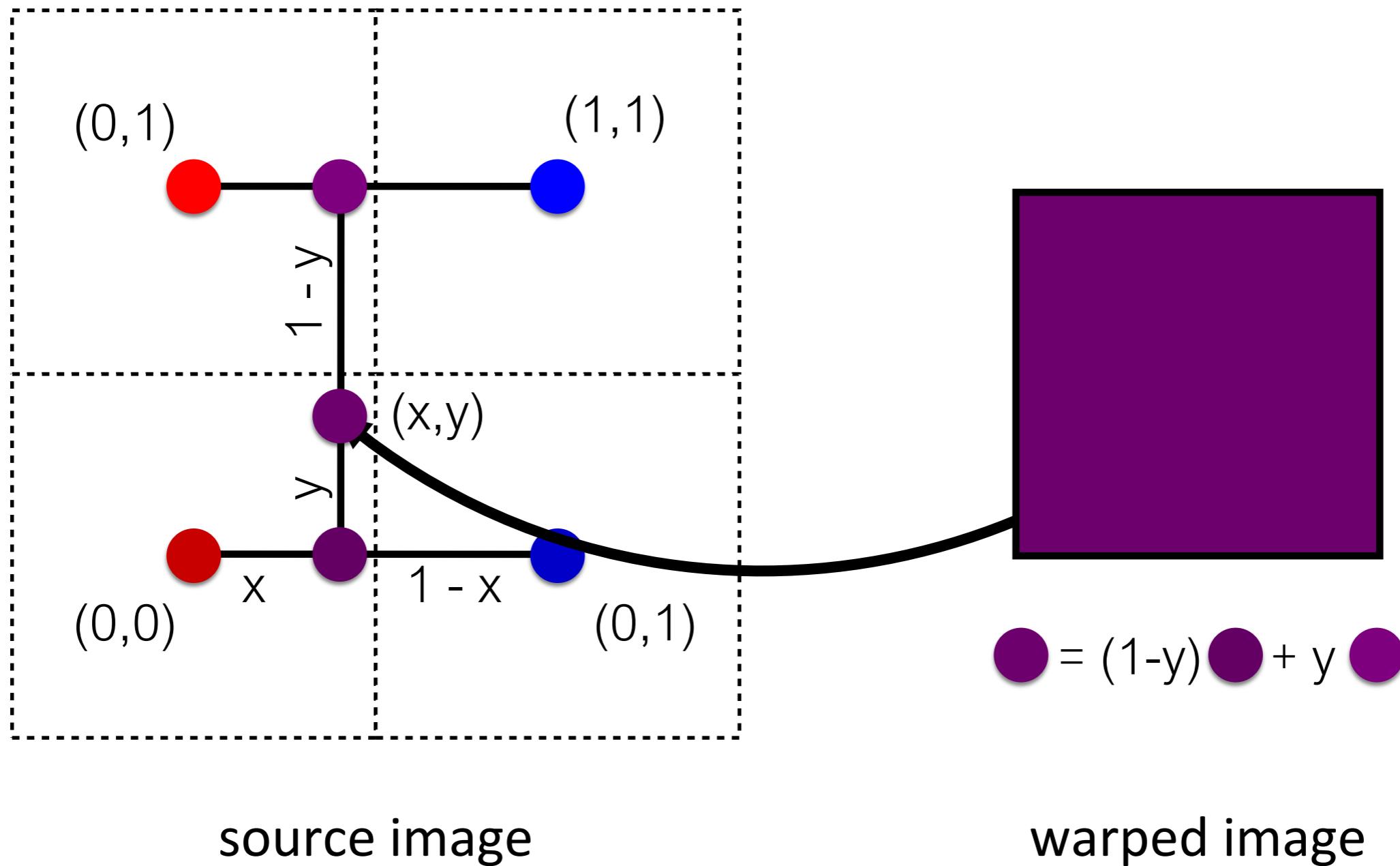
Bilinear Interpolation



Bilinear Interpolation



Bilinear Interpolation



Today

- Image Warping
- Transformation Estimation

Different Transformation Types



source



rigid



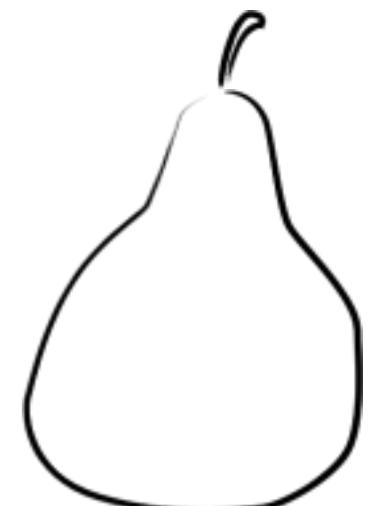
similarity



affine



perspective

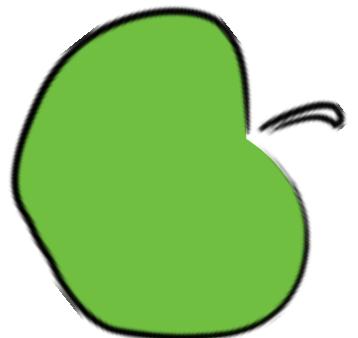


free-form

Different Transformation Types



source



rigid



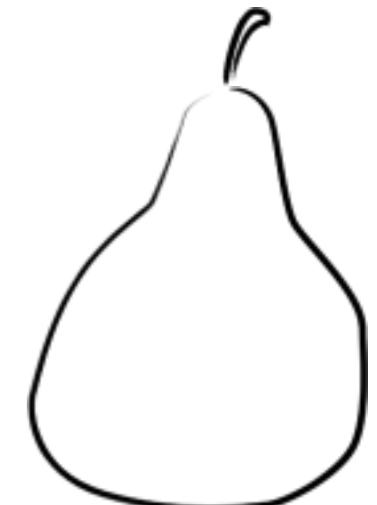
similarity



affine



perspective



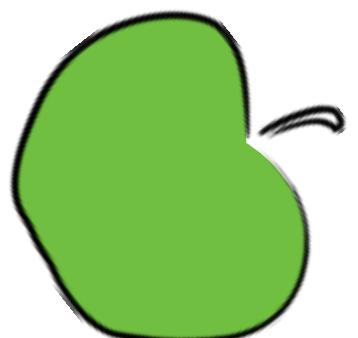
free-form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Different Transformation Types



source



rigid



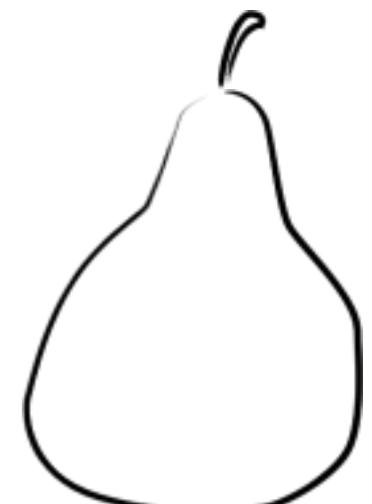
similarity



affine



perspective



free-form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Different Transformation Types



source



rigid



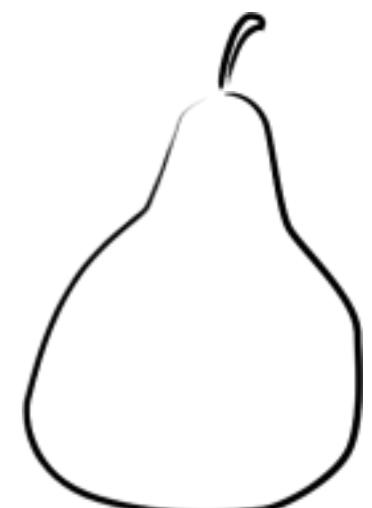
similarity



affine



perspective



free-form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

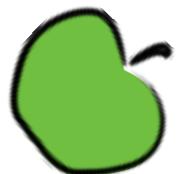
Different Transformation Types



source



rigid



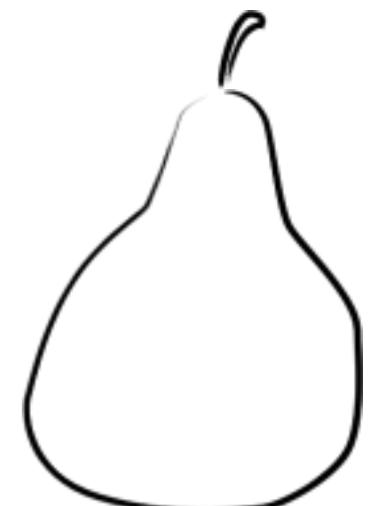
similarity



affine



perspective



free-form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Different Transformation Types



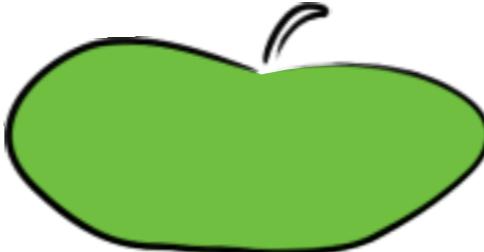
source



rigid



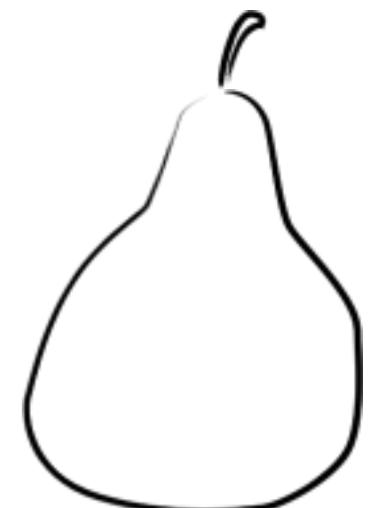
similarity



affine



perspective



free-form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Different Transformation Types



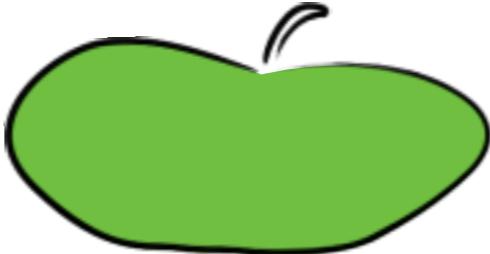
source



rigid



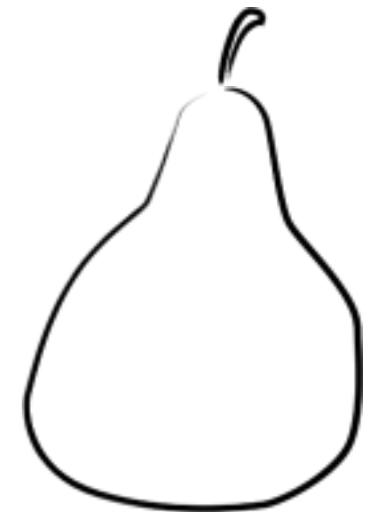
similarity



affine



perspective



free-form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Different Transformation Types



source



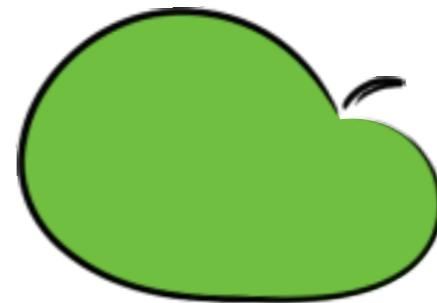
rigid



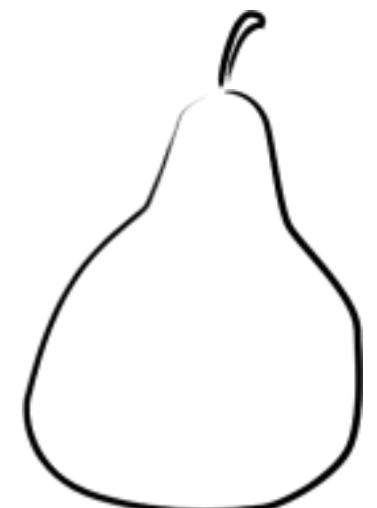
similarity



affine



perspective



free-form

$$\hat{w} \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \hat{w} \neq 0$$

Different Transformation Types



source

8 DoF (up to an arbitrary scaling)
(more in the next lecture)



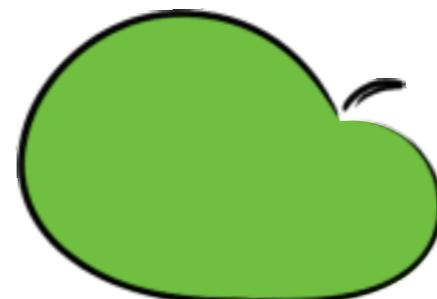
rigid



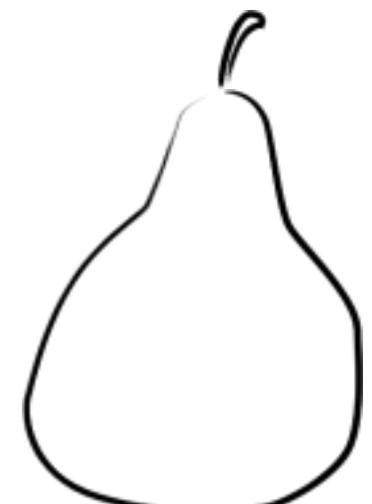
similarity



affine



perspective



free-form

$$\hat{w} \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \hat{w} \neq 0$$

Different Transformation Types



source



rigid



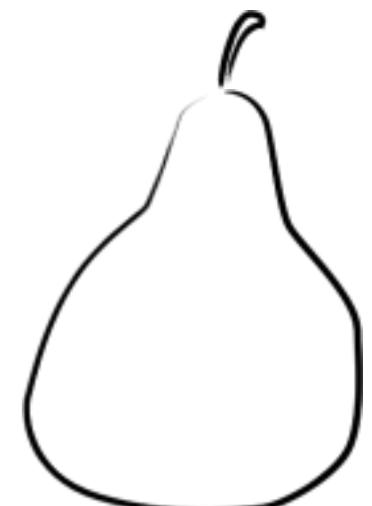
similarity



affine



perspective



free-form

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Fitting A Similarity Transformation

Similarity transformation:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Fitting A Similarity Transformation

Similarity transformation:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Simplified:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \boxed{\begin{pmatrix} a & -b \\ b & a \end{pmatrix}} \begin{pmatrix} x \\ y \end{pmatrix} + \boxed{\begin{pmatrix} t_x \\ t_y \end{pmatrix}}$$


parameters to be estimated

Fitting A Similarity Transformation

Rewrite

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

as linear system in parameters:

$$\begin{pmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}$$

Fitting A Similarity Transformation

Rewrite

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

as linear system in parameters:

$$\begin{pmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}$$

parameters to be estimated

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\ \hat{y}_2 \end{pmatrix}$$

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\ \hat{y}_2 \end{pmatrix}$$

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\ \hat{y}_2 \end{pmatrix}$$

A x b

Solve in Matlab as $x=A\b$

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\ \hat{y}_2 \end{pmatrix}$$

A x b

Solve in Matlab as $x=A\backslash b$

Reshape parameters into matrix and translation

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\ \hat{y}_2 \end{pmatrix}$$

A x b

Solve in Matlab as $x=A\backslash b$

Reshape parameters into matrix and translation

Use solver inside RANSAC

Fitting A Similarity Transformation

Least squares solution: Stack equations from > 2 correspondences

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\ \hat{y}_2 \\ \vdots \\ \hat{x}_n \\ \hat{y}_n \end{pmatrix}$$

Fitting A Similarity Transformation

Least squares solution: Stack equations from > 2 correspondences

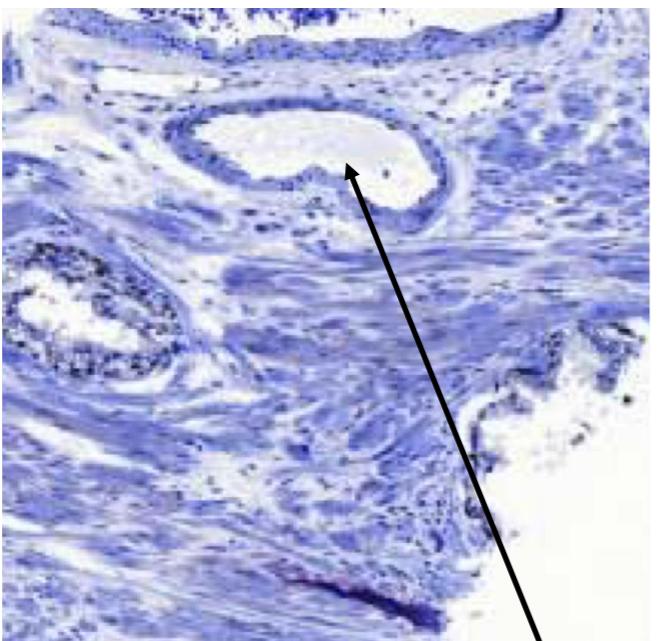
$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\ \hat{y}_2 \\ \vdots \\ \hat{x}_n \\ \hat{y}_n \end{pmatrix}$$

A x b

Solve in Matlab as $x=A\backslash b$

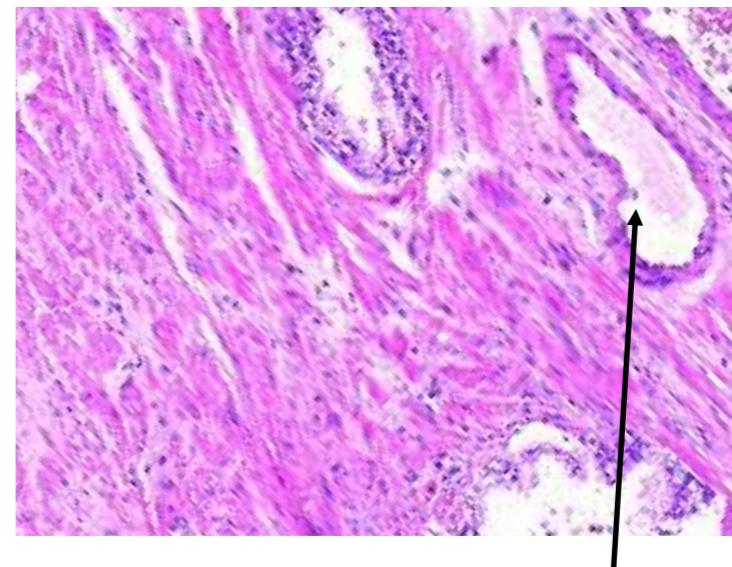
BUG ALERT

source
image



$$\begin{pmatrix} \tilde{x}_j \\ \tilde{y}_j \end{pmatrix}$$

target
image



$$\begin{pmatrix} x_j \\ y_j \end{pmatrix}$$

Find $\begin{pmatrix} \tilde{x}_j \\ \tilde{y}_j \end{pmatrix} = A \begin{pmatrix} x_j \\ y_j \end{pmatrix} + t$

Different Transformation Types



source



rigid



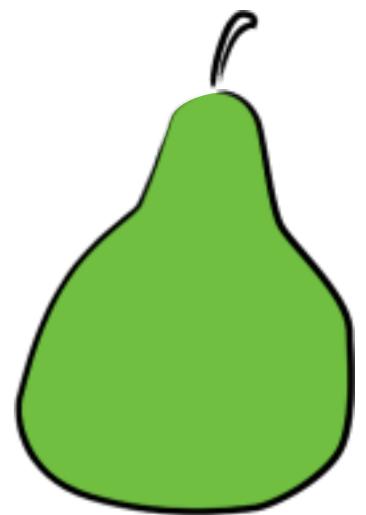
similarity



affine

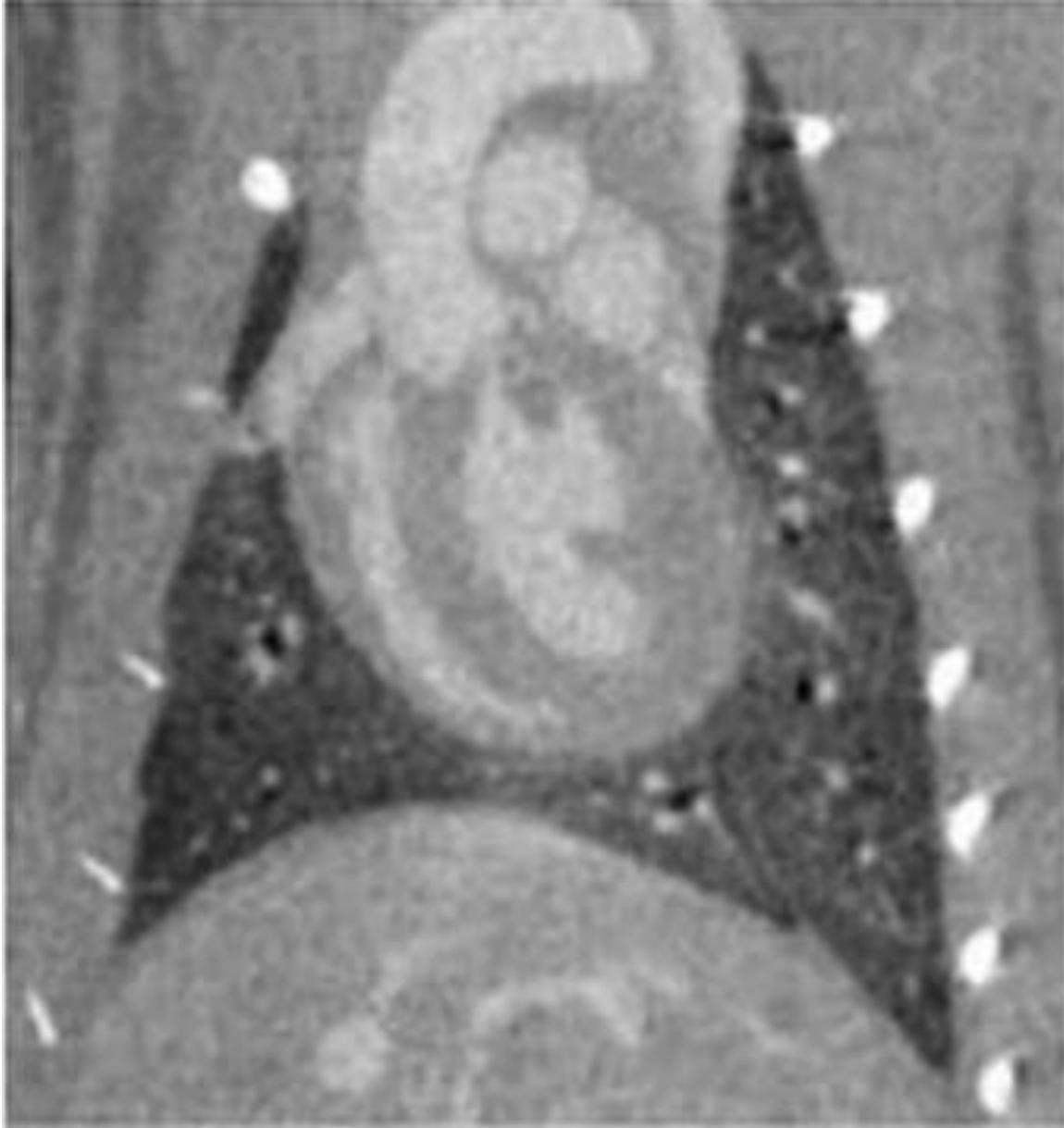


perspective



free-form

Free-Form Registration

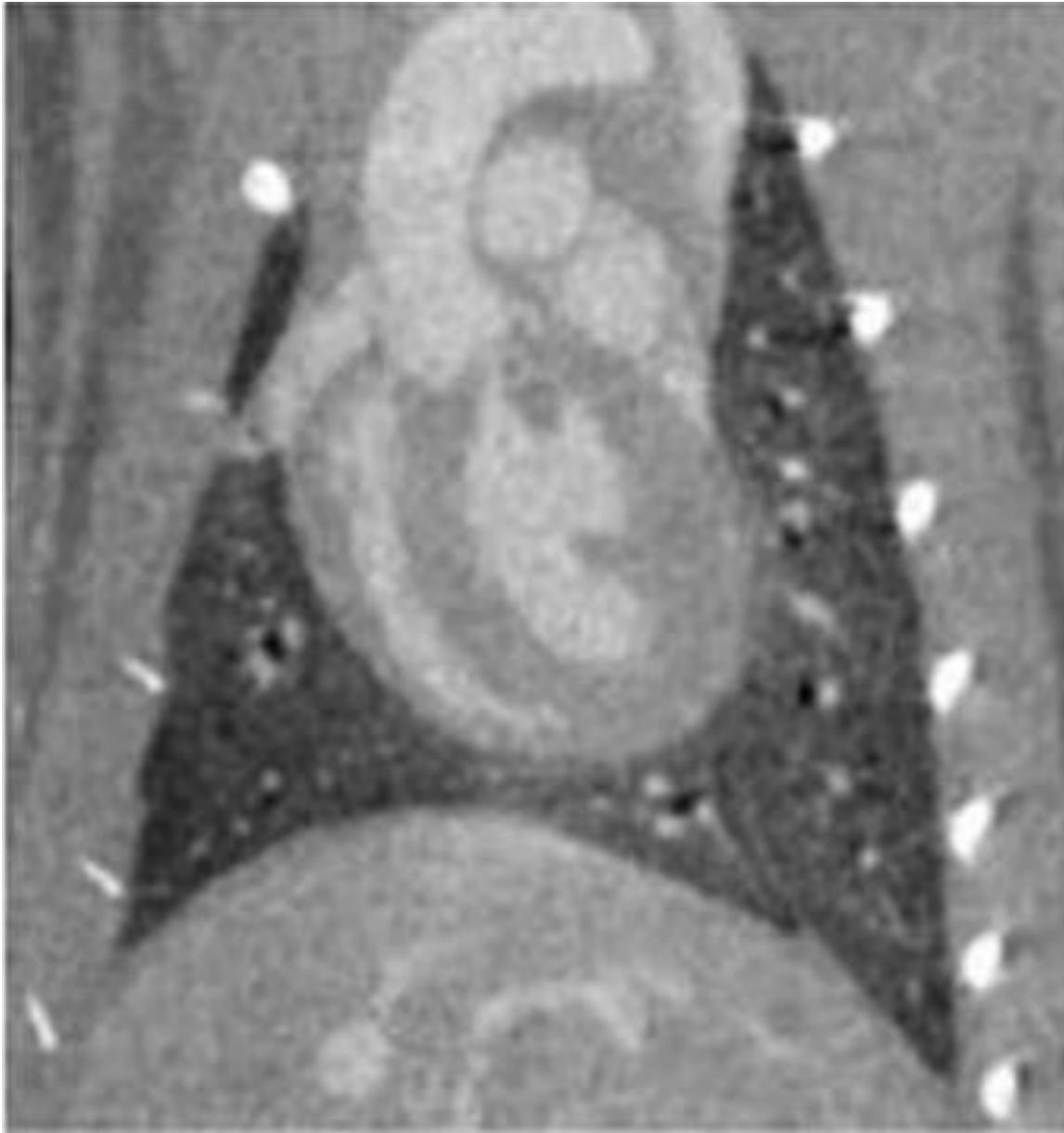


source (systole)



target (diastole)

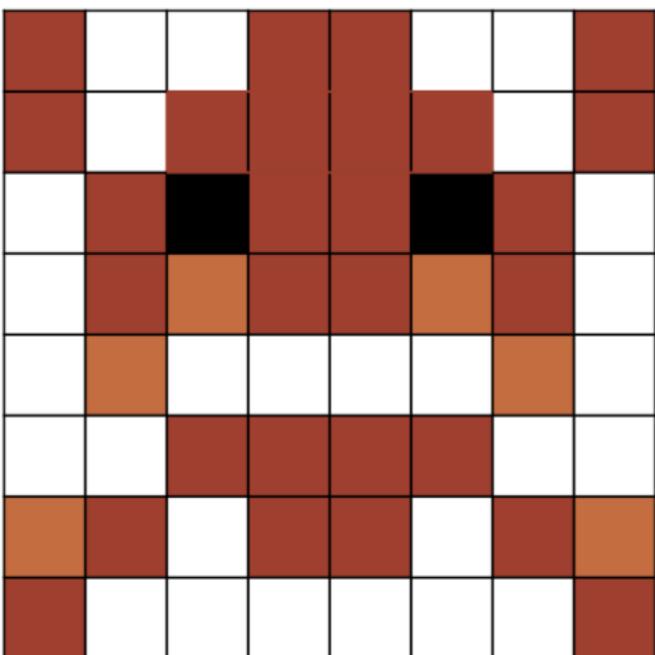
Parameterization



Specify the motion of every pixel?

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \Delta(x, y)$$

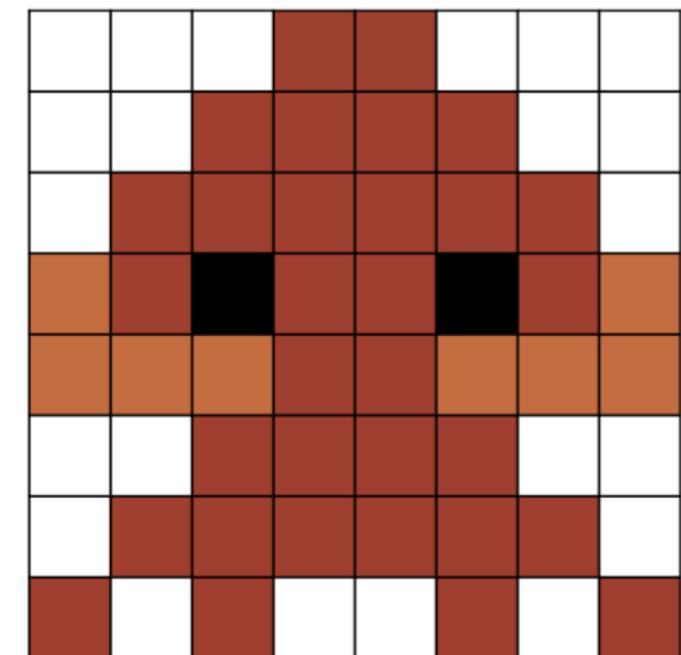
Example



source image

2							2
2							2
2		-1	-1	-1	-1		2
3		-1	-1	-1	-1		3
2		-1	-1	-1	-1		2
-1		-1	-1	-1	-1		-1
		-2			-2		

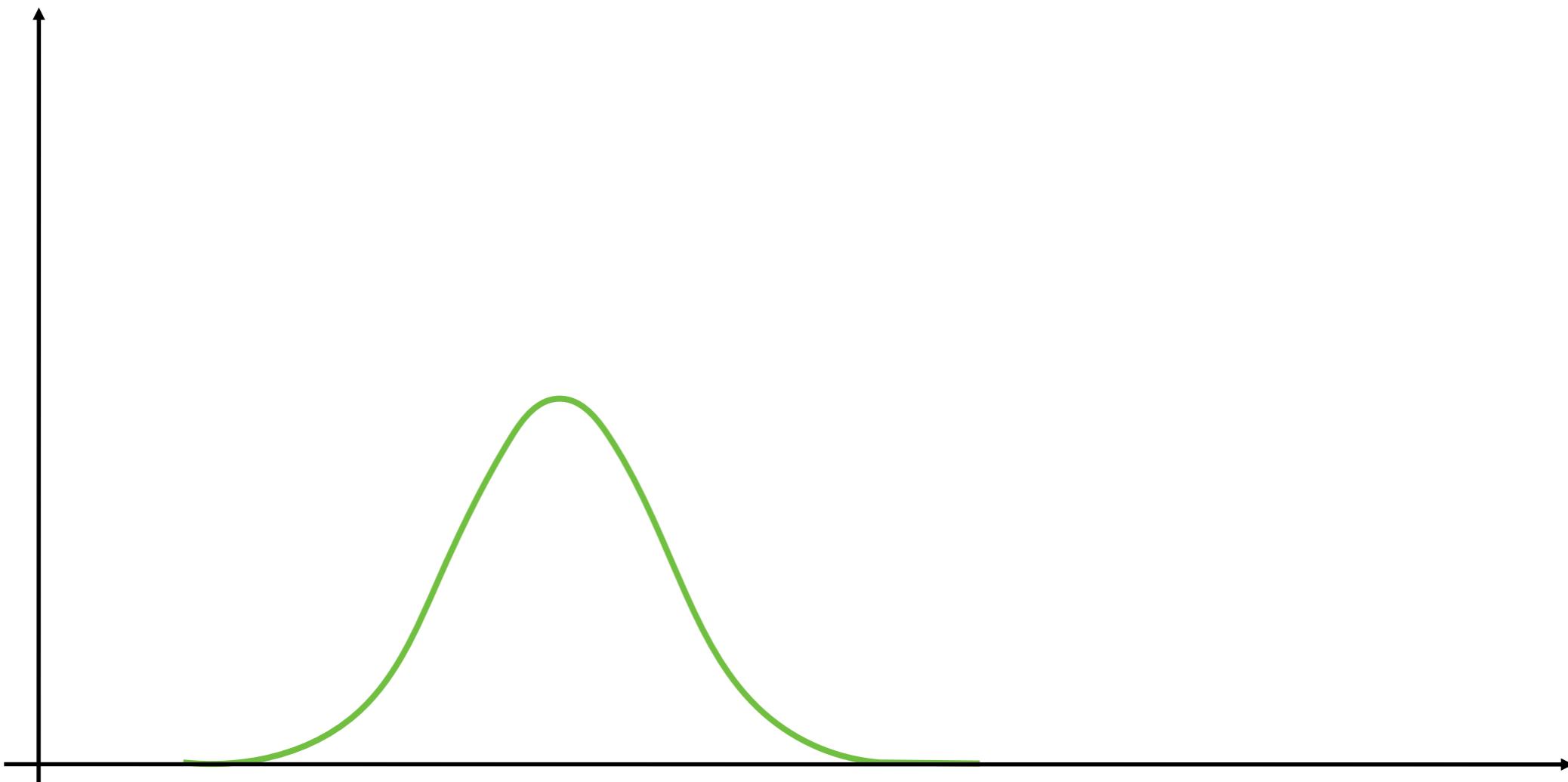
$$\Delta_y$$



warped image

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \Delta(x, y)$$

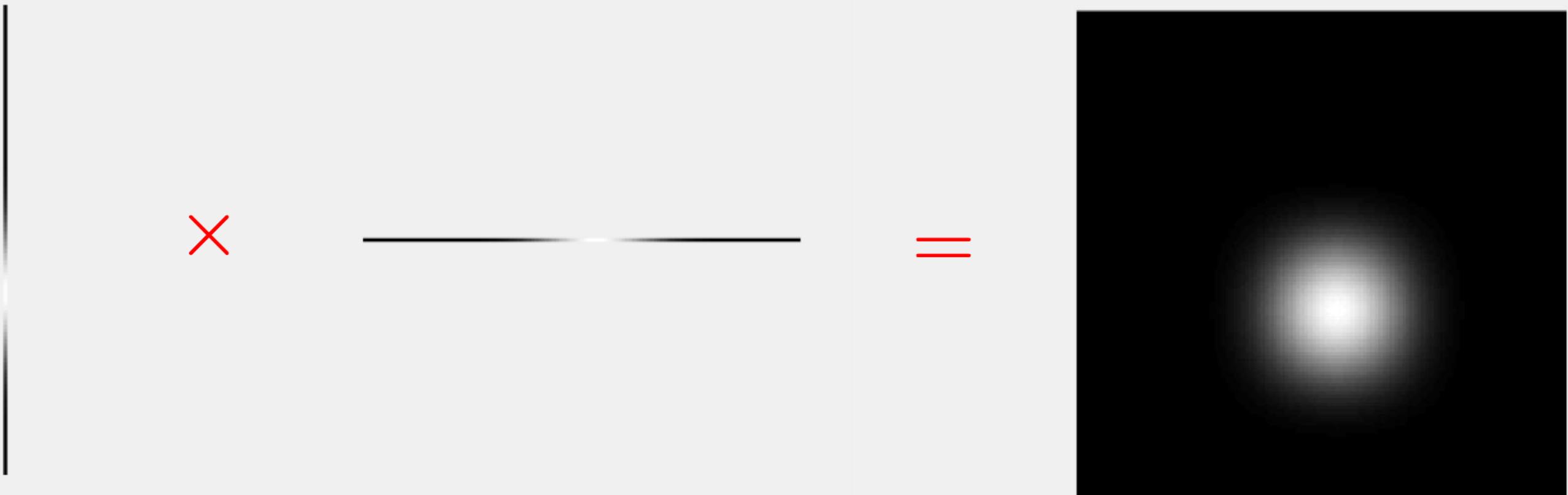
Use Basis Functions



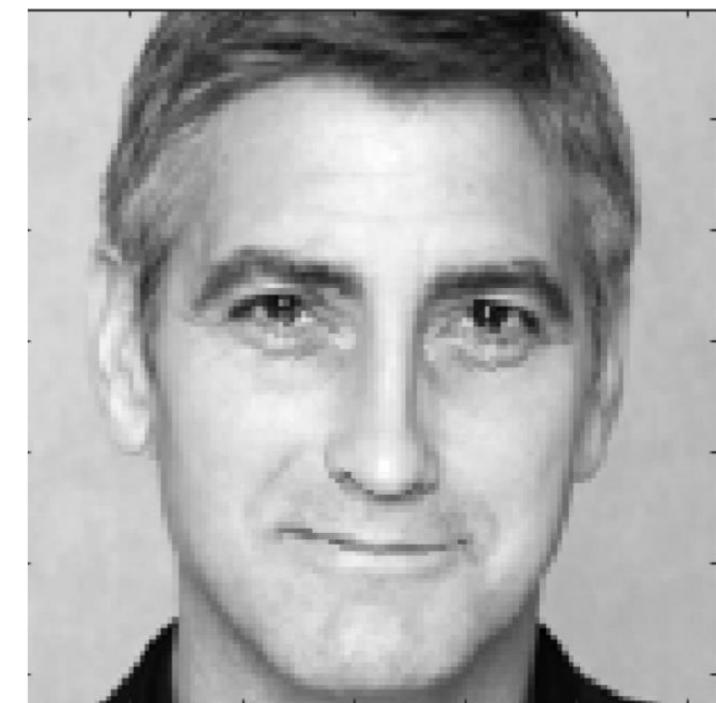
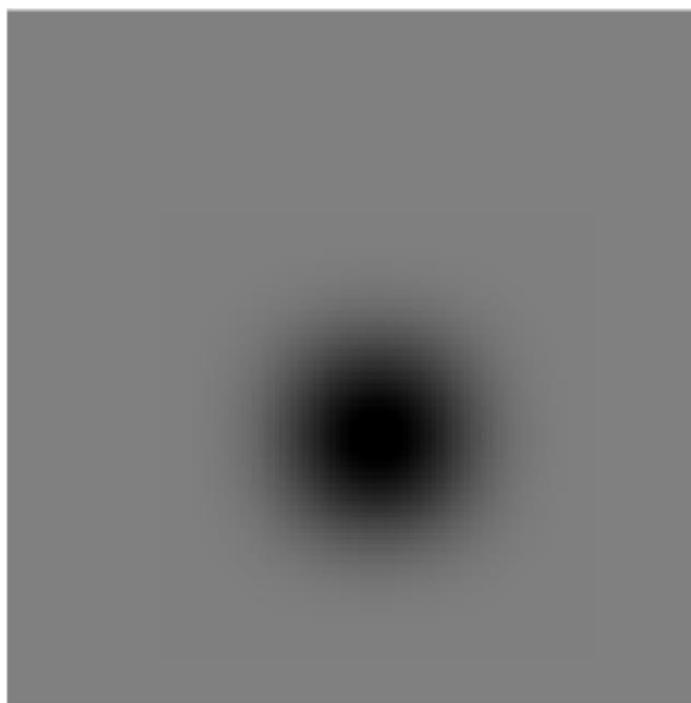
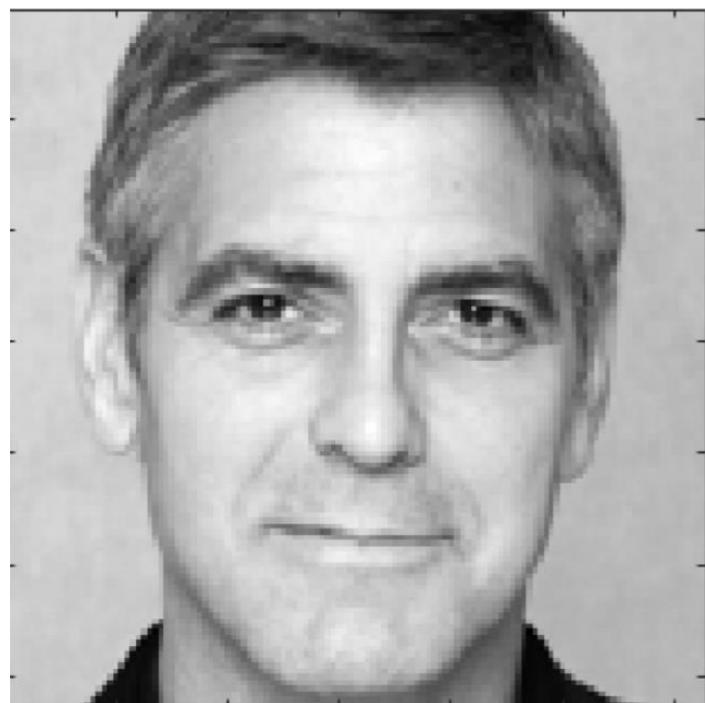
Cubic B-spline

- ✓ Local
- ✓ Smooth
- ✓ Simple equation (efficient to optimize)

In 2D

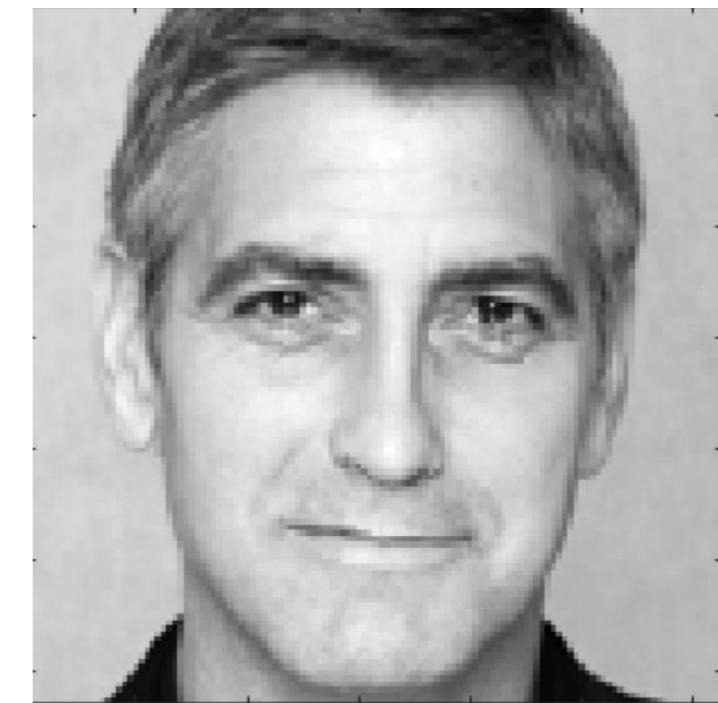
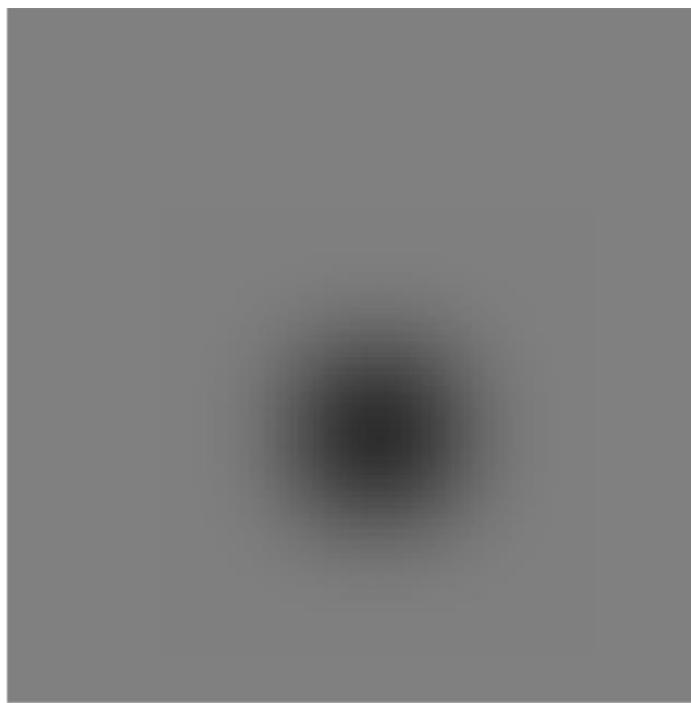
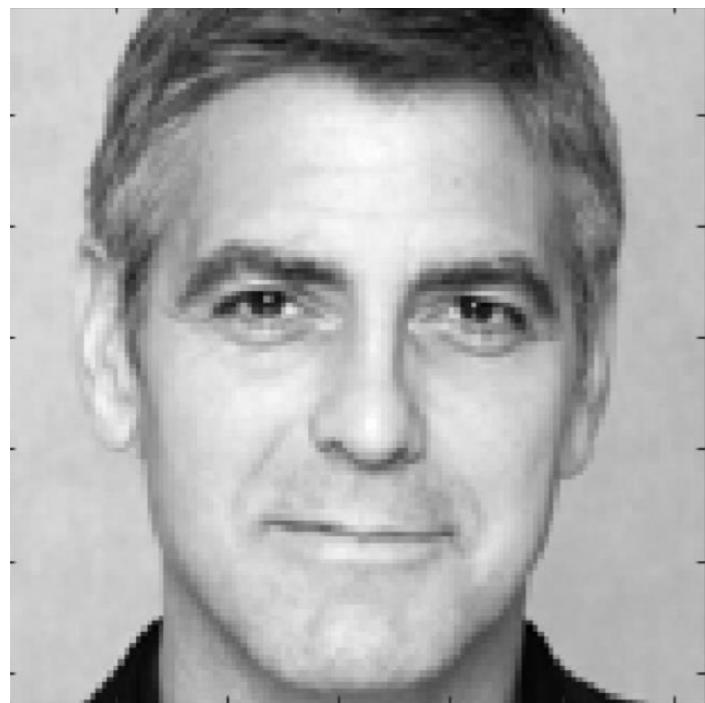


Cubic B-Splines



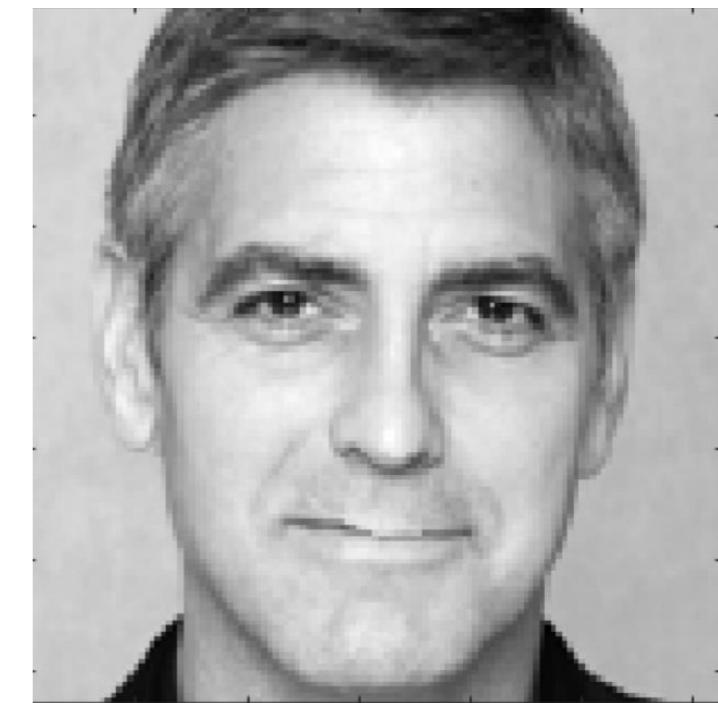
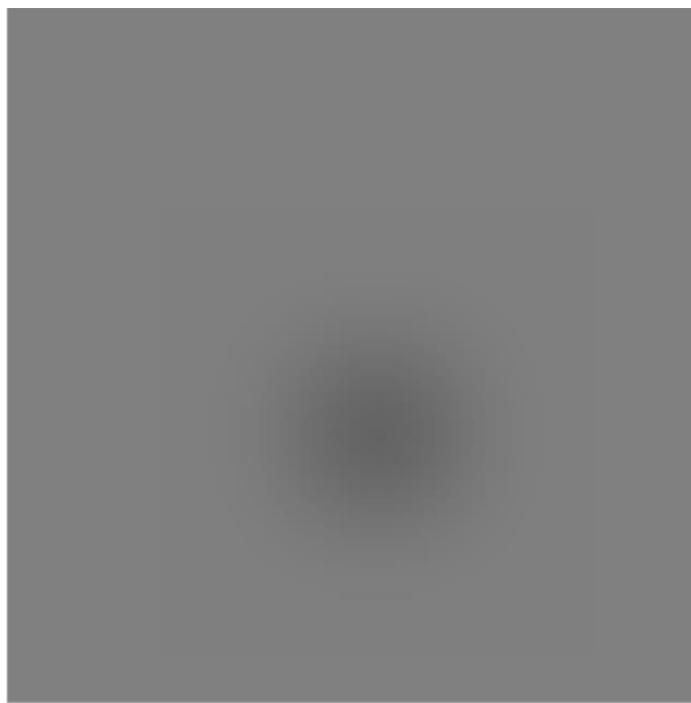
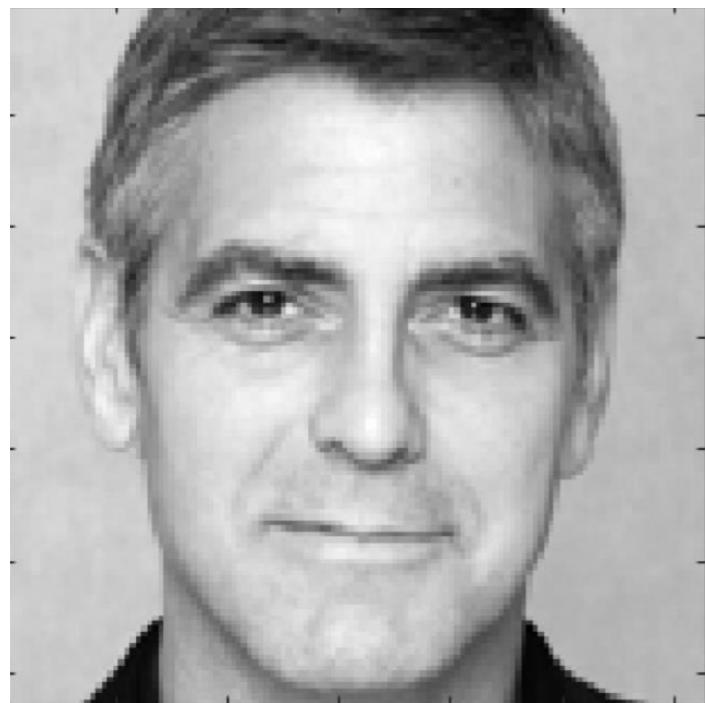
$$\Delta_y$$

Cubic B-Splines



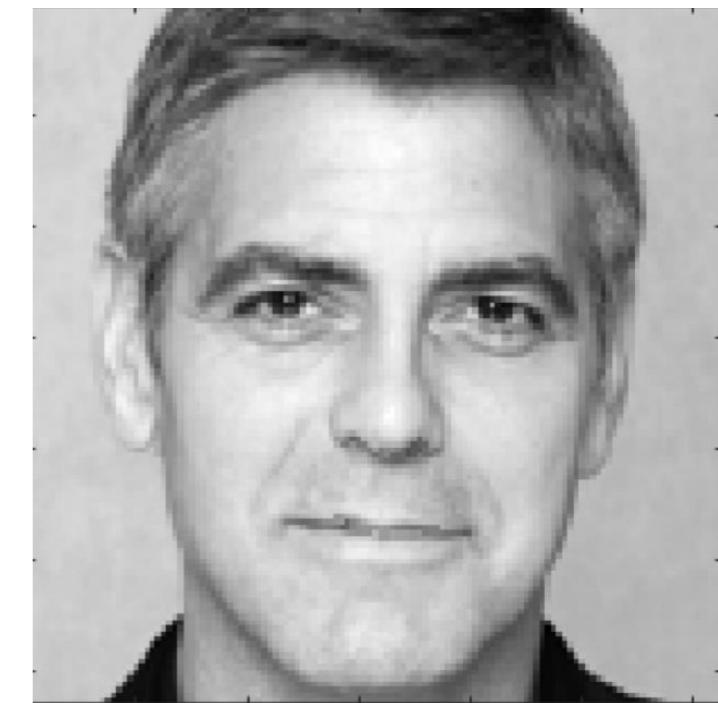
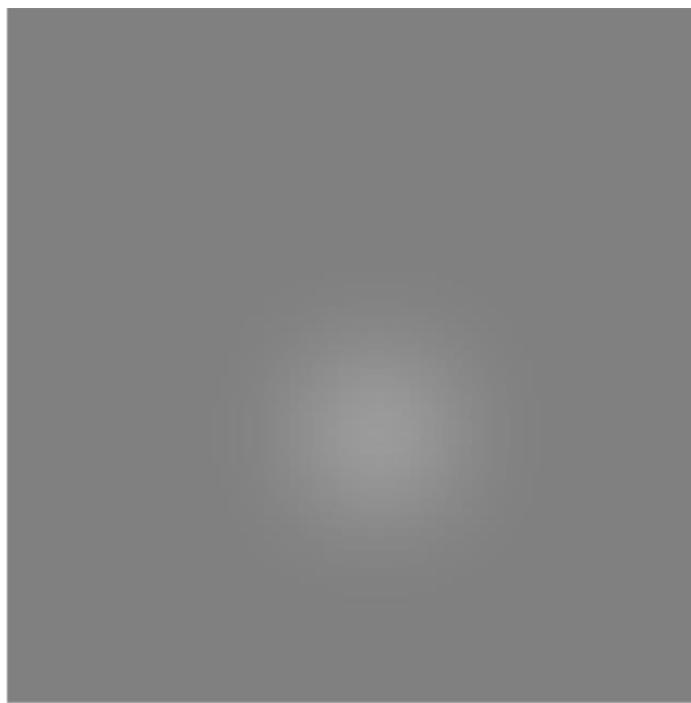
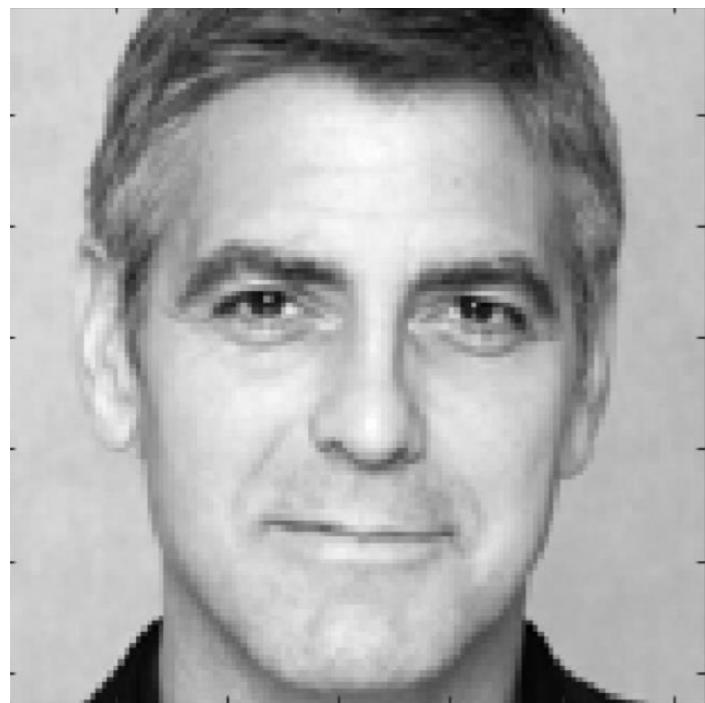
$$\Delta_y$$

Cubic B-Splines



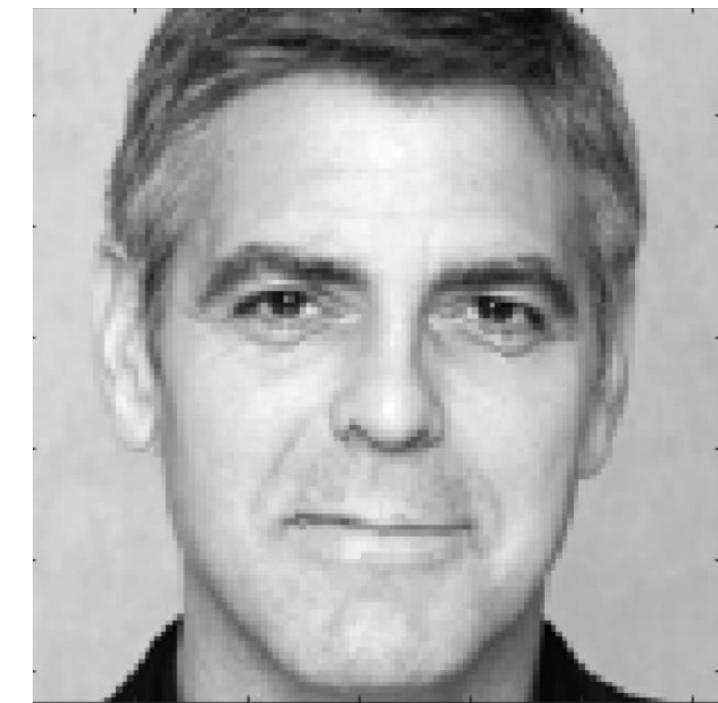
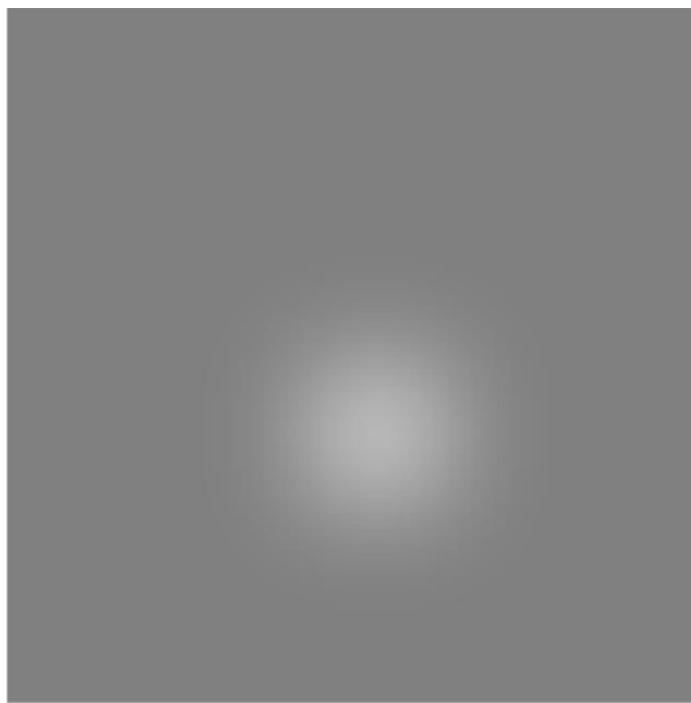
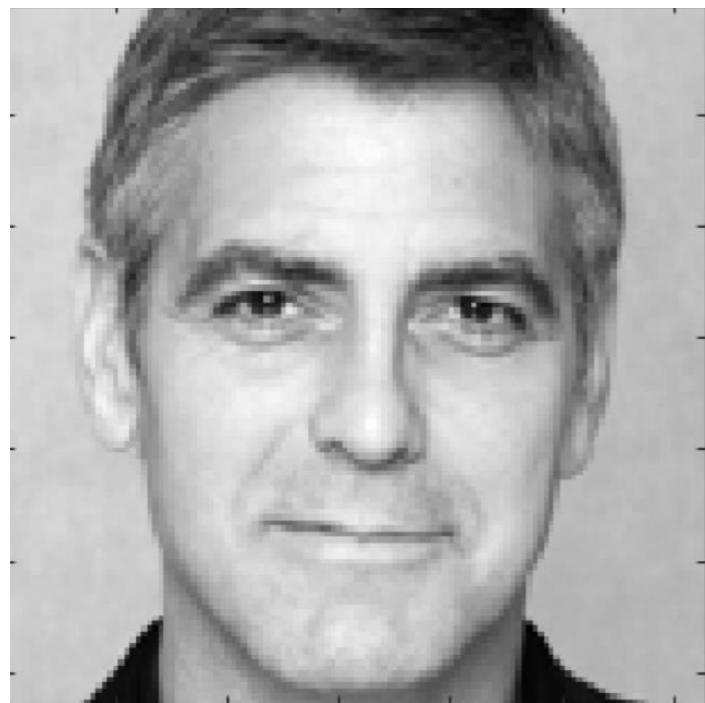
$$\Delta_y$$

Cubic B-Splines



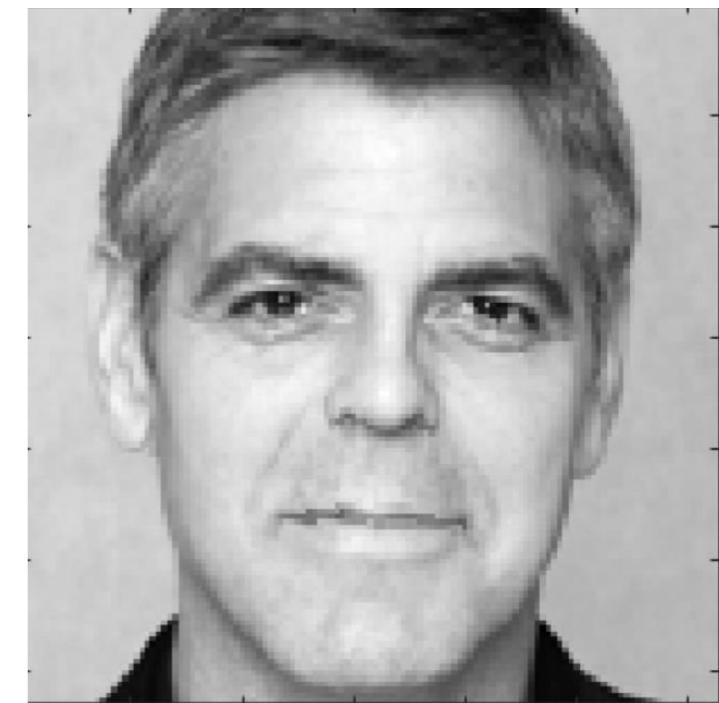
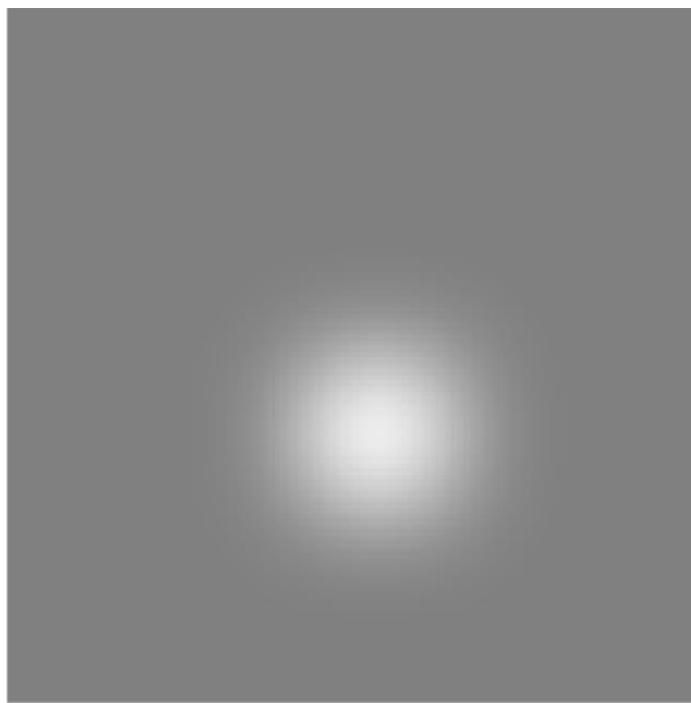
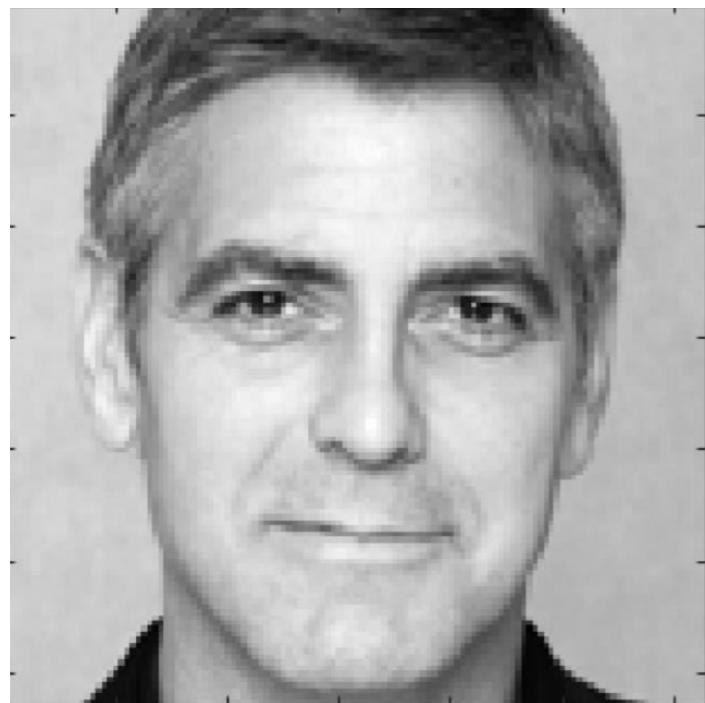
$$\Delta_y$$

Cubic B-Splines



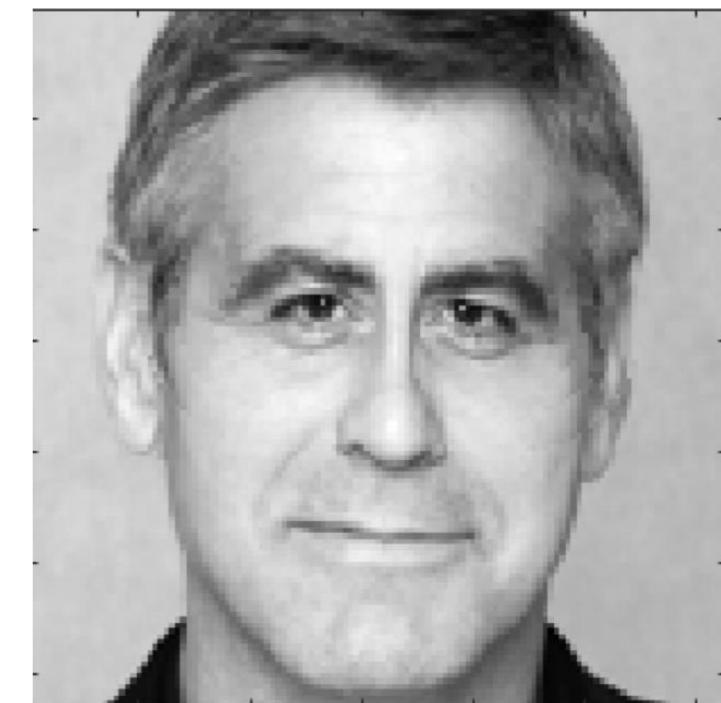
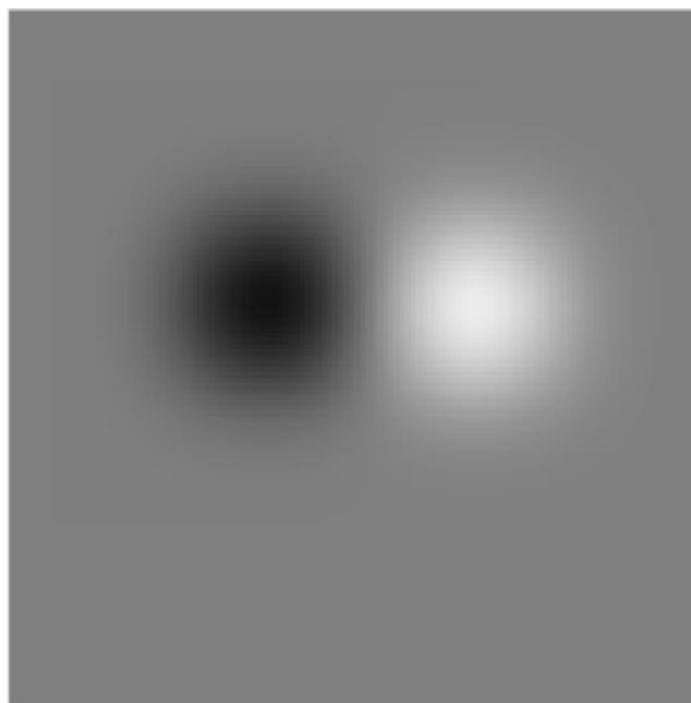
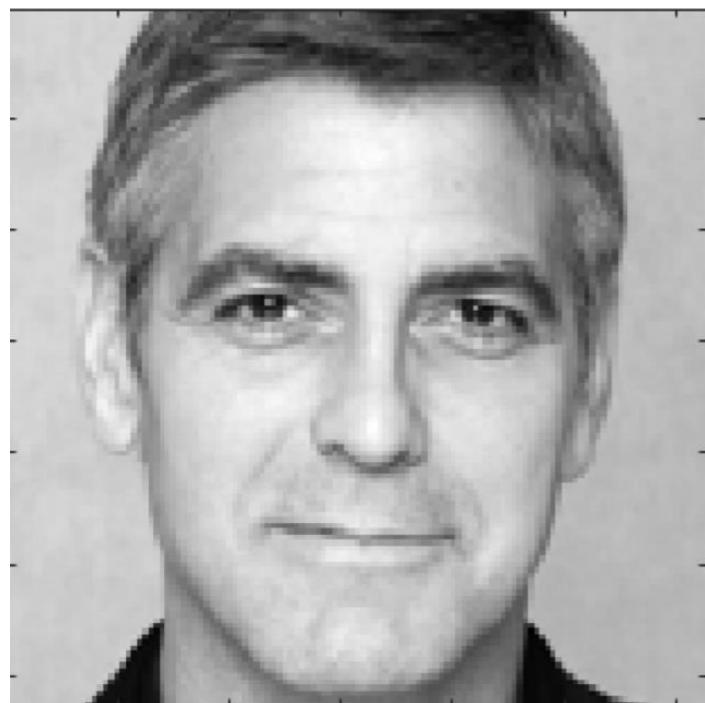
$$\Delta_y$$

Cubic B-Splines



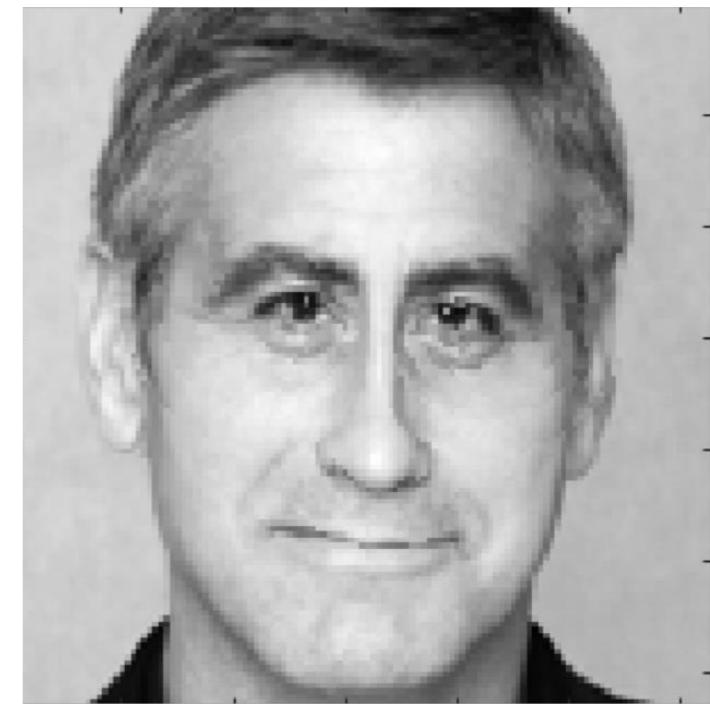
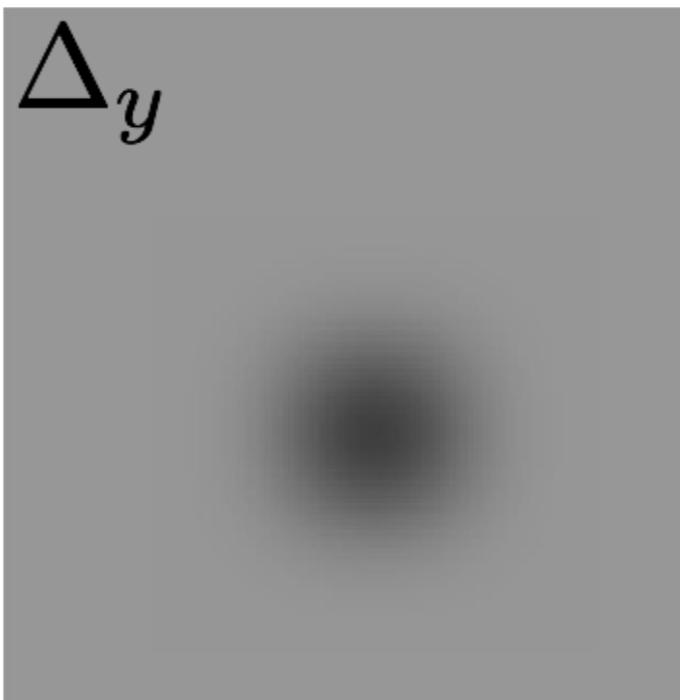
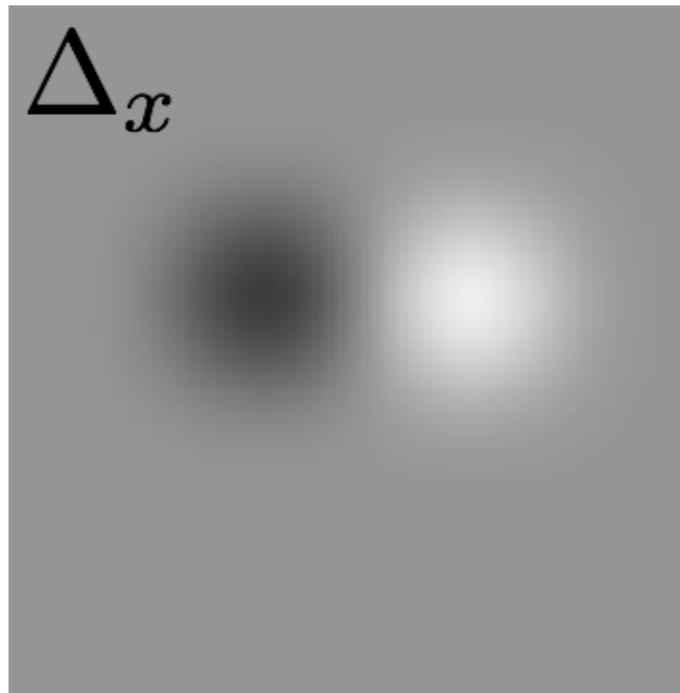
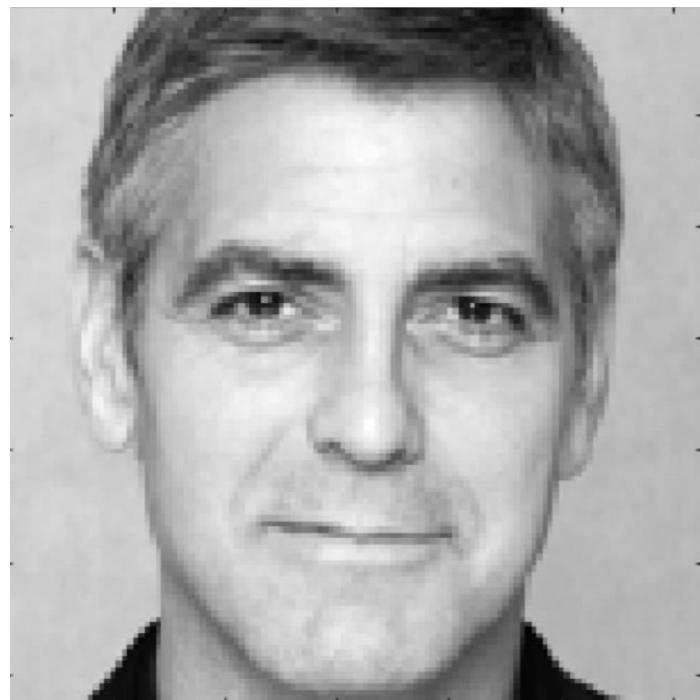
$$\Delta_y$$

Cubic B-Splines

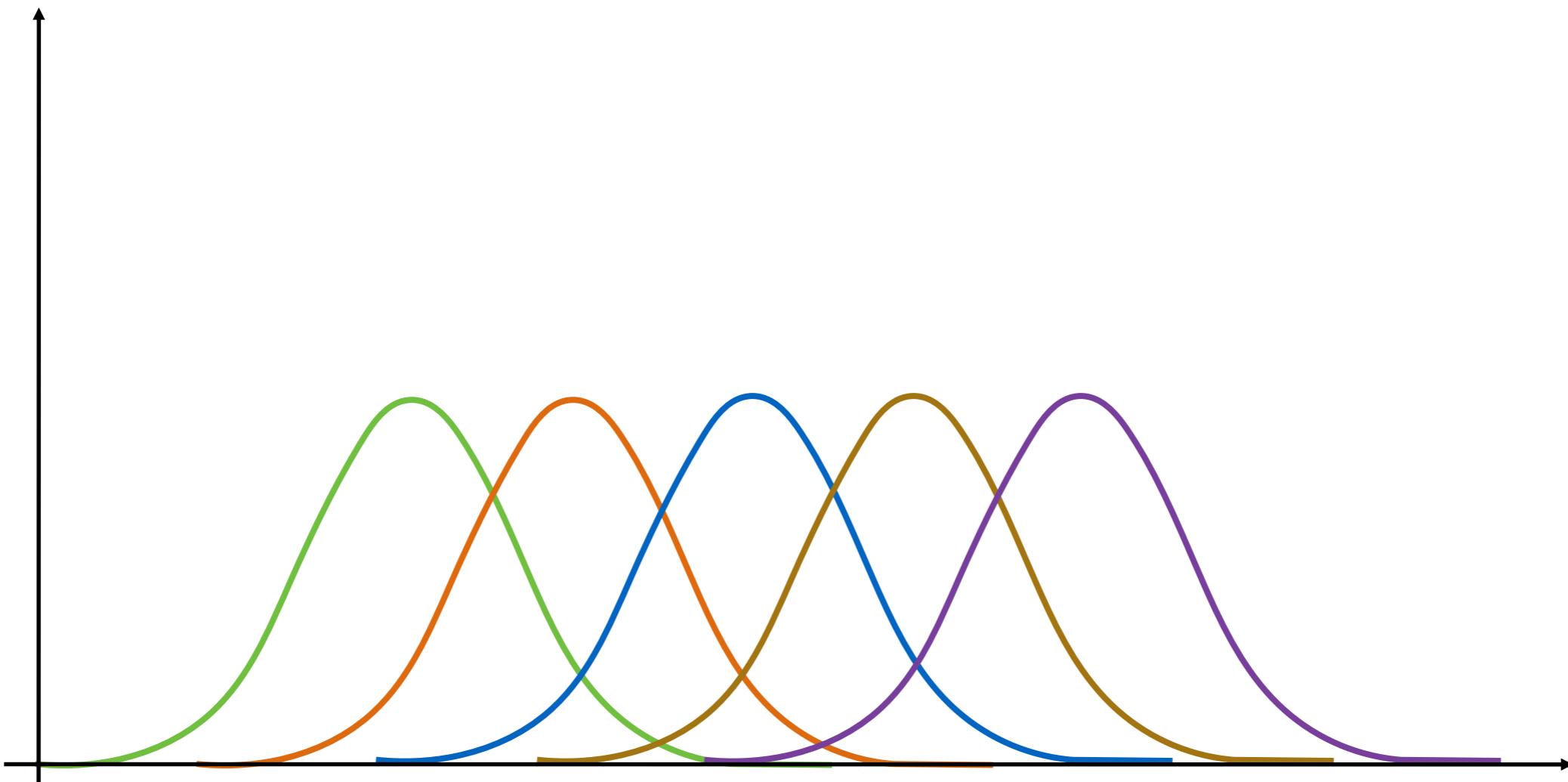


$$\Delta_x$$

Cubic B-Splines



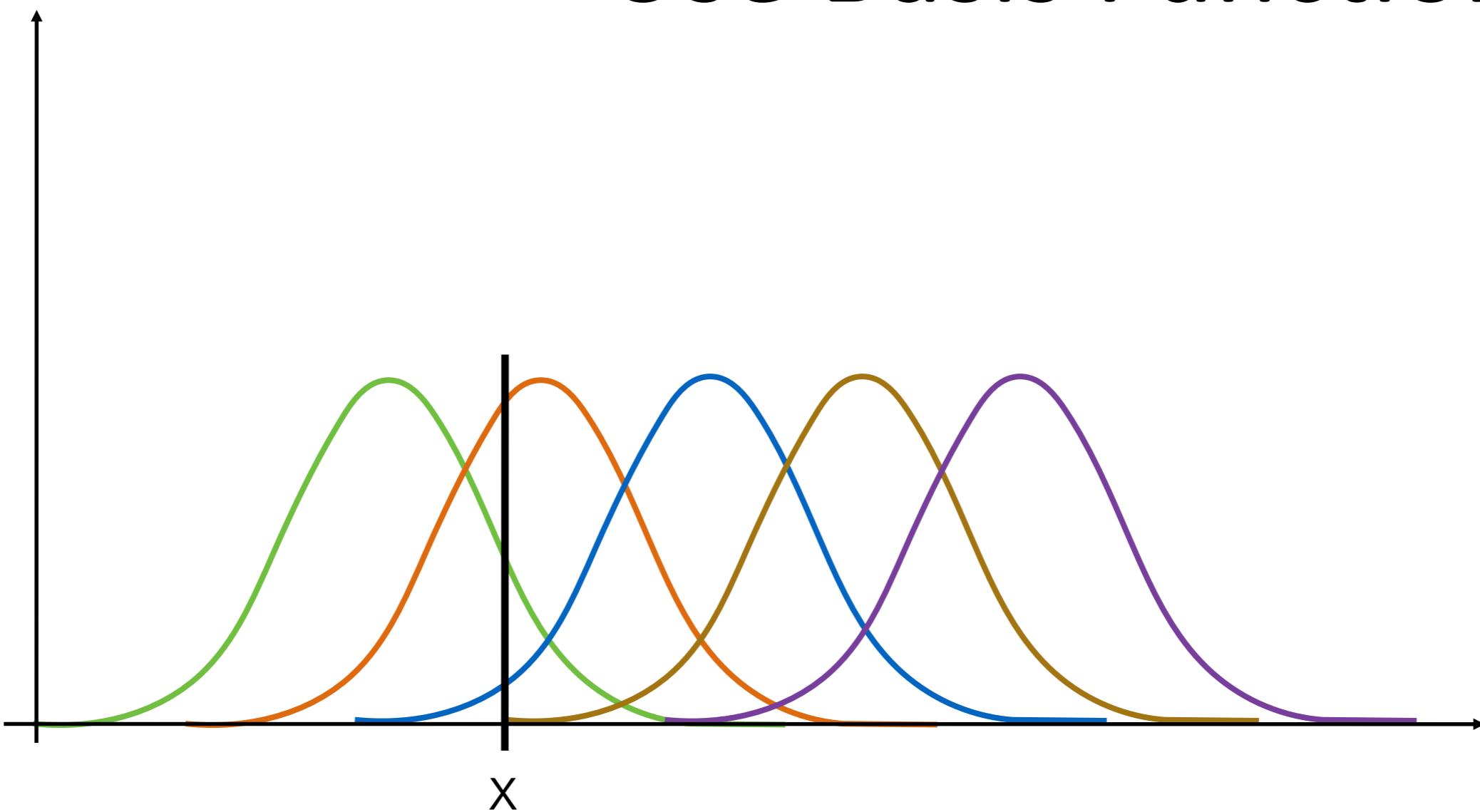
Use Basis Functions



$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

$$\Delta_y(x, y) = \sum_{k=1}^n \beta_k B_k(x, y)$$

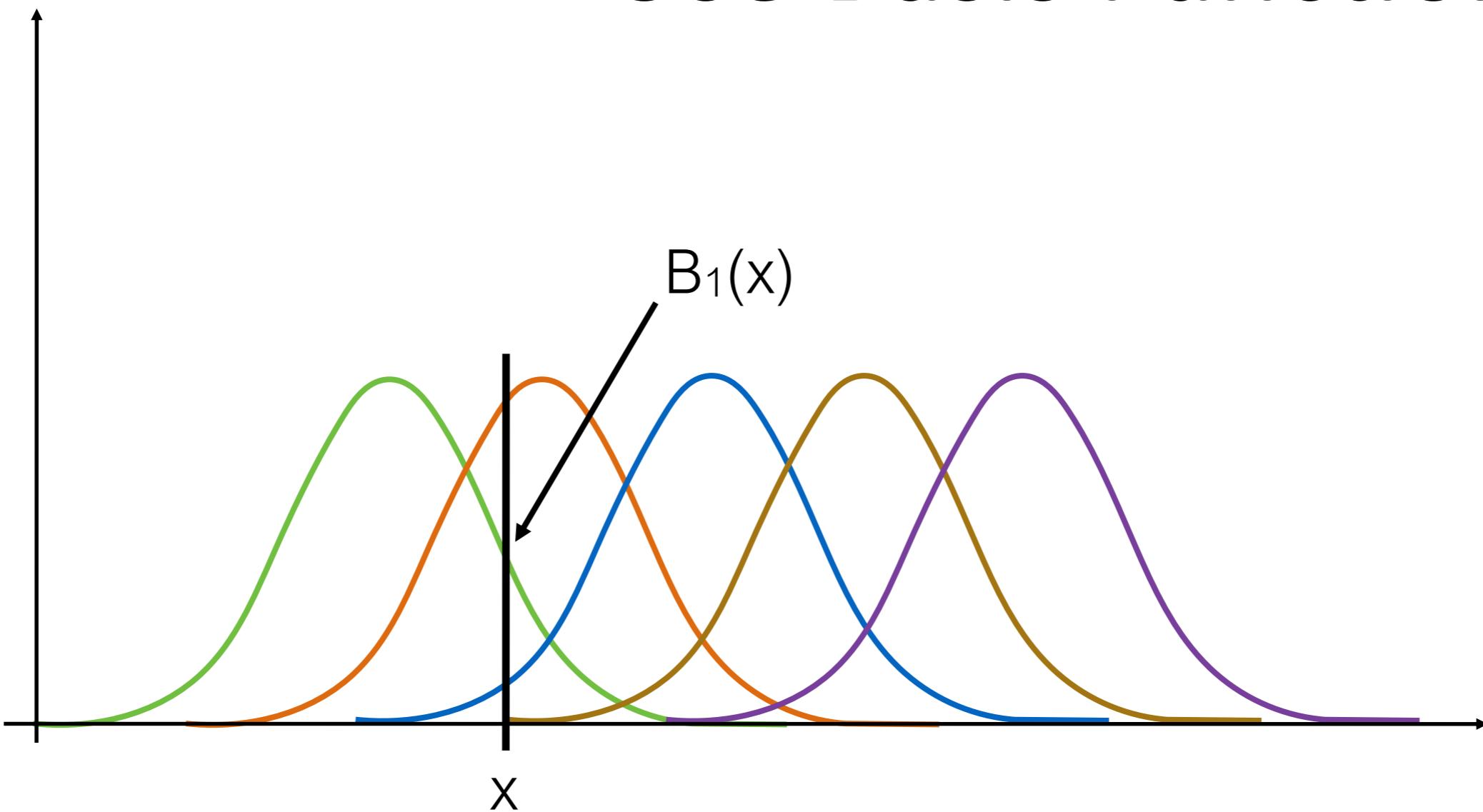
Use Basis Functions



$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

$$\Delta_y(x, y) = \sum_{k=1}^n \beta_k B_k(x, y)$$

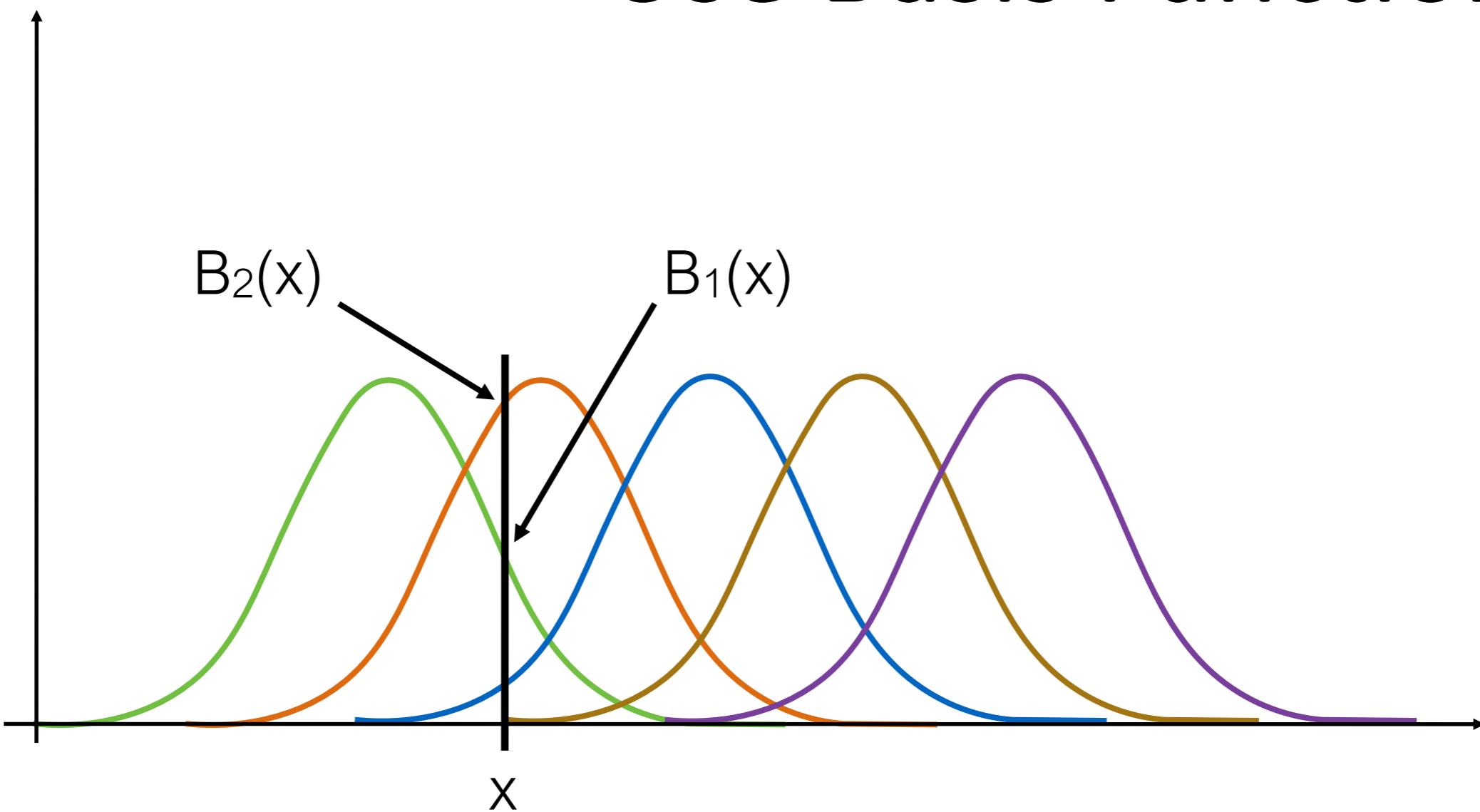
Use Basis Functions



$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

$$\Delta_y(x, y) = \sum_{k=1}^n \beta_k B_k(x, y)$$

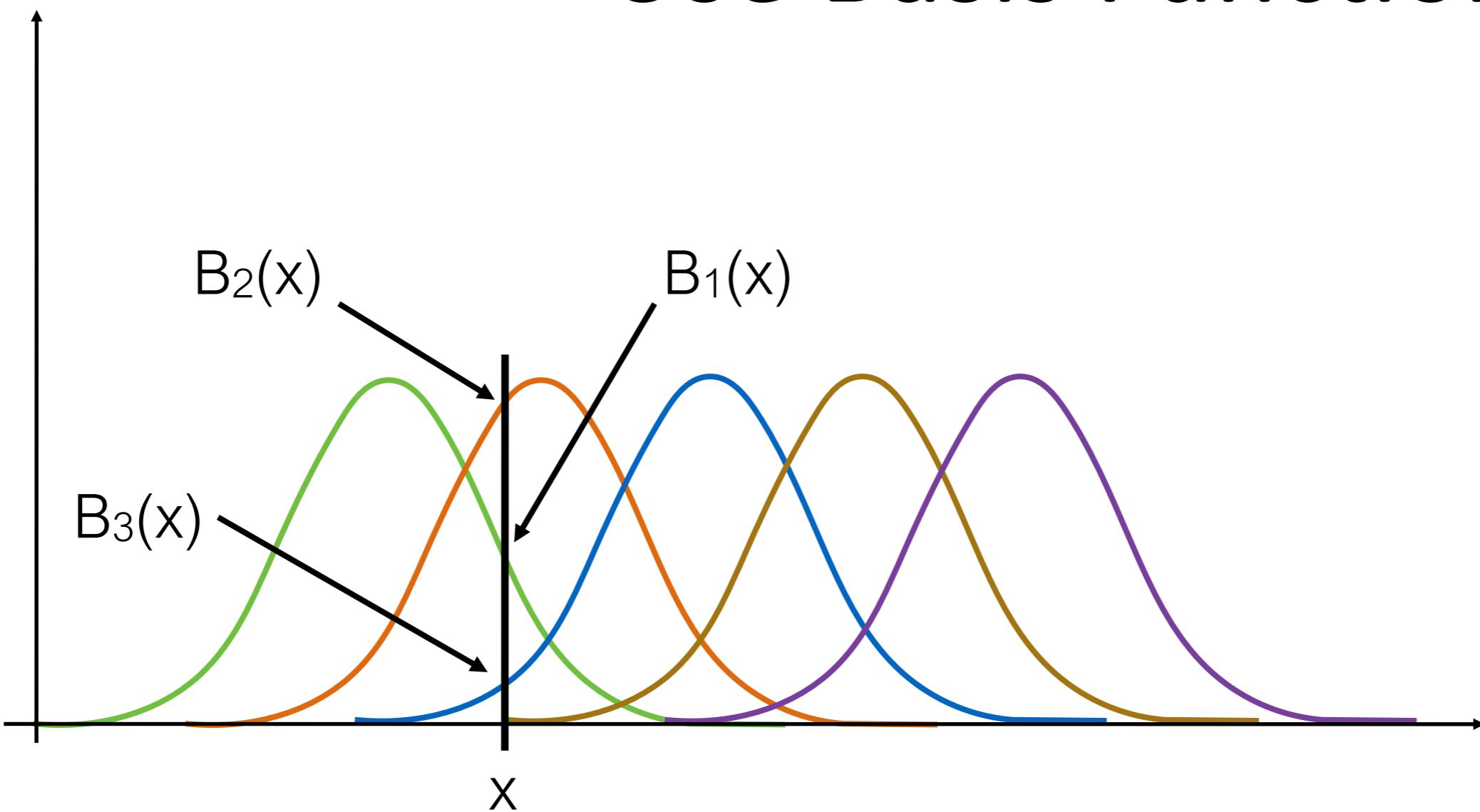
Use Basis Functions



$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

$$\Delta_y(x, y) = \sum_{k=1}^n \beta_k B_k(x, y)$$

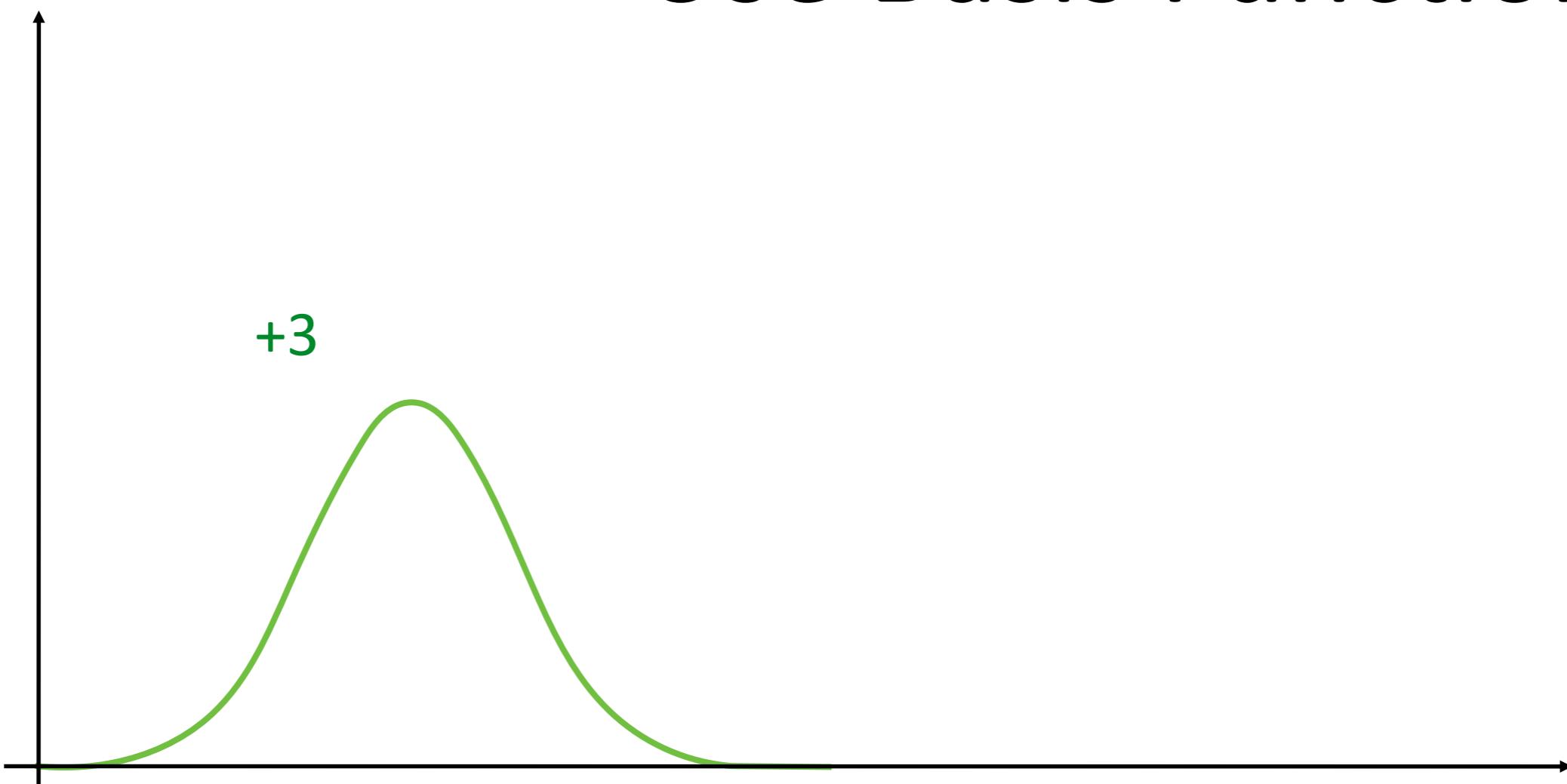
Use Basis Functions



$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

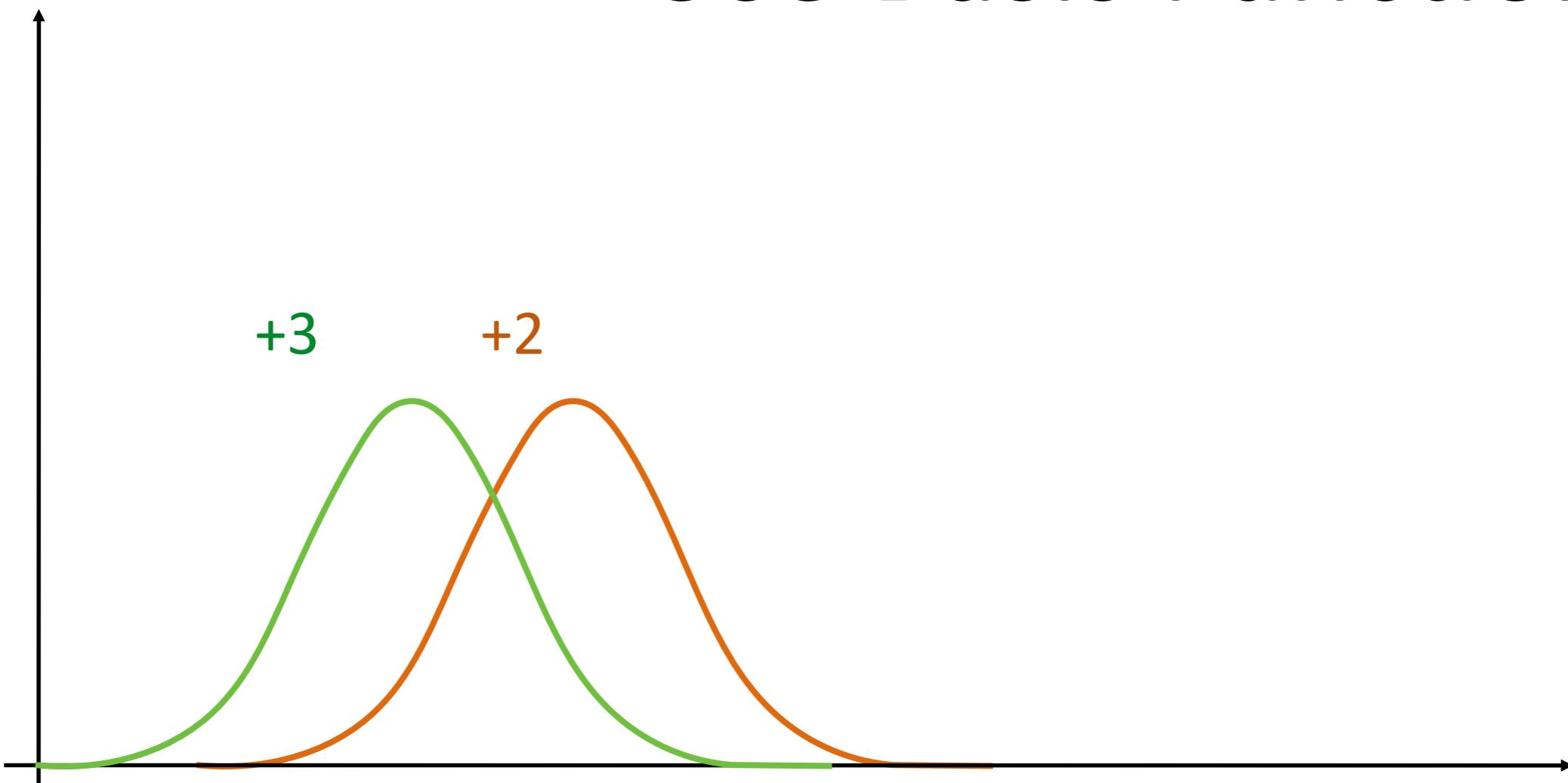
$$\Delta_y(x, y) = \sum_{k=1}^n \beta_k B_k(x, y)$$

Use Basis Functions



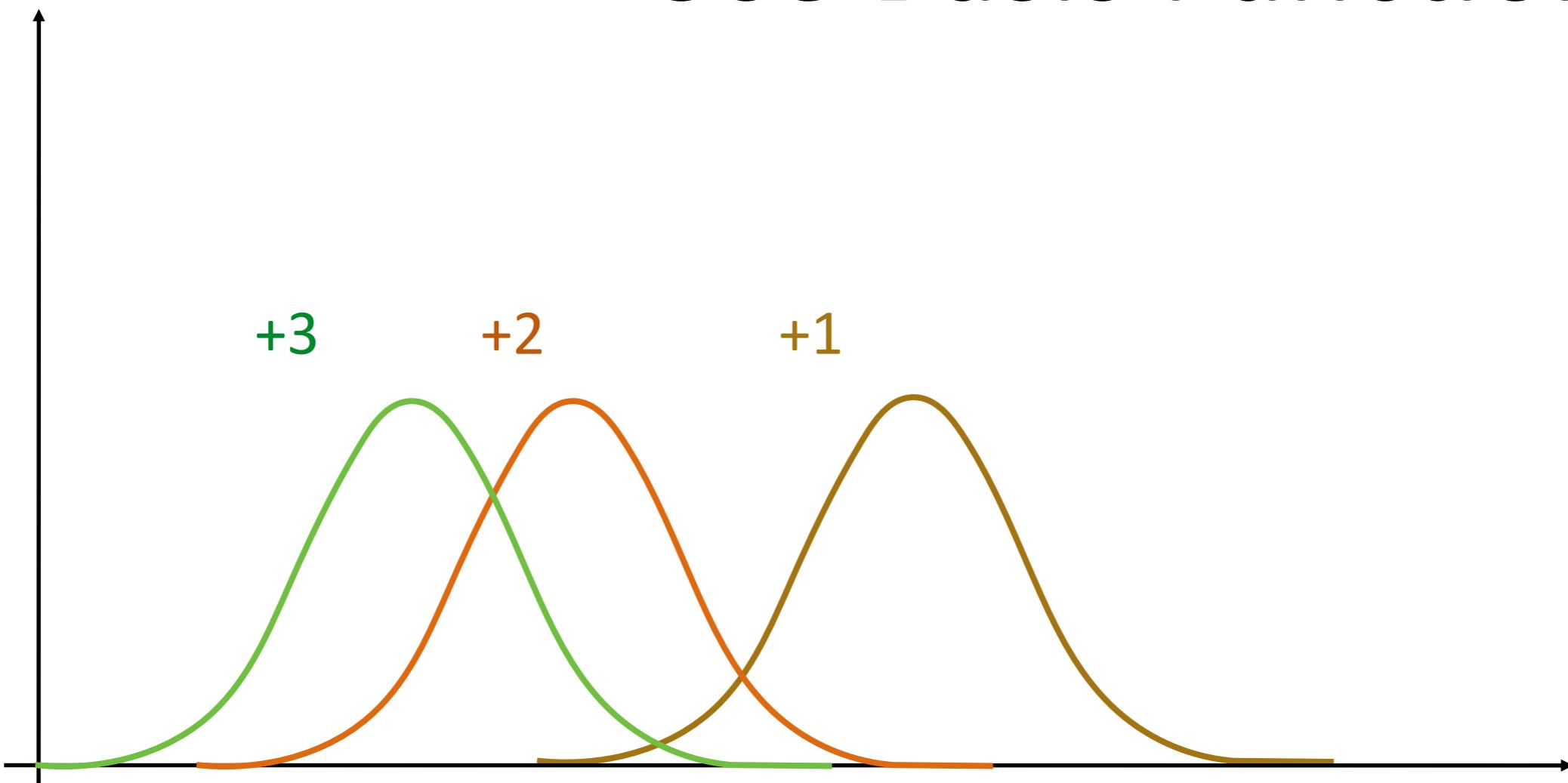
$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

Use Basis Functions



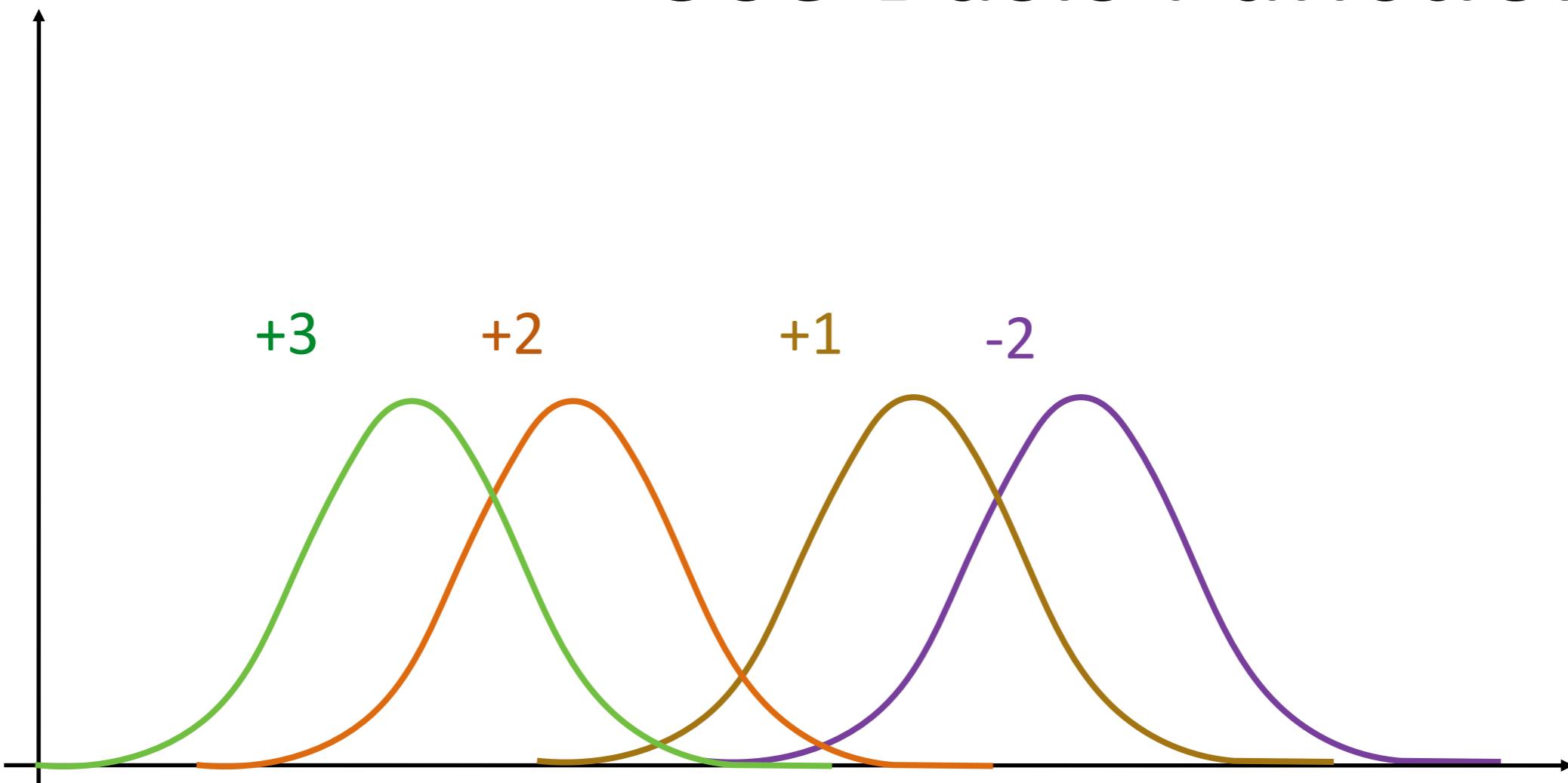
$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

Use Basis Functions



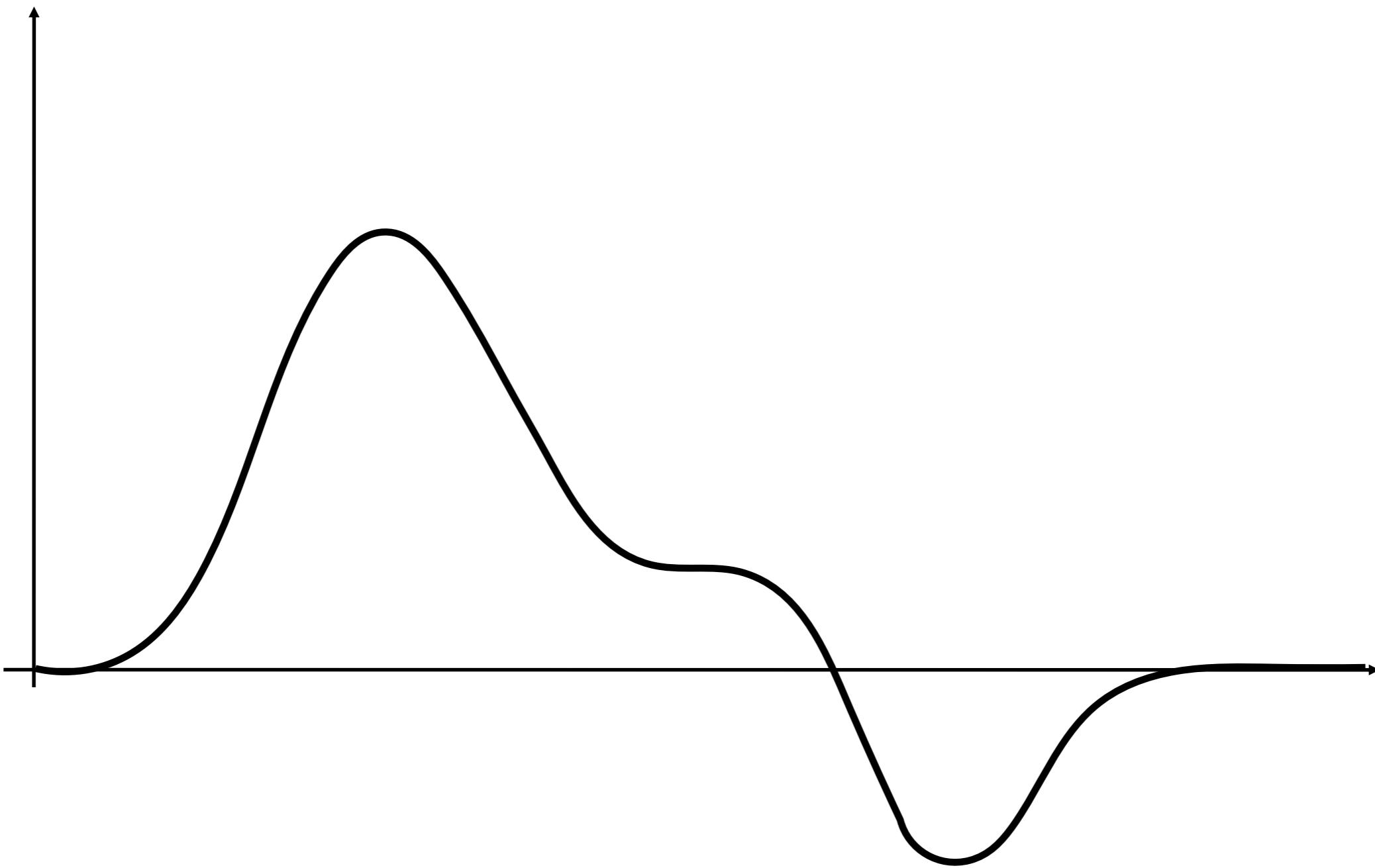
$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

Use Basis Functions



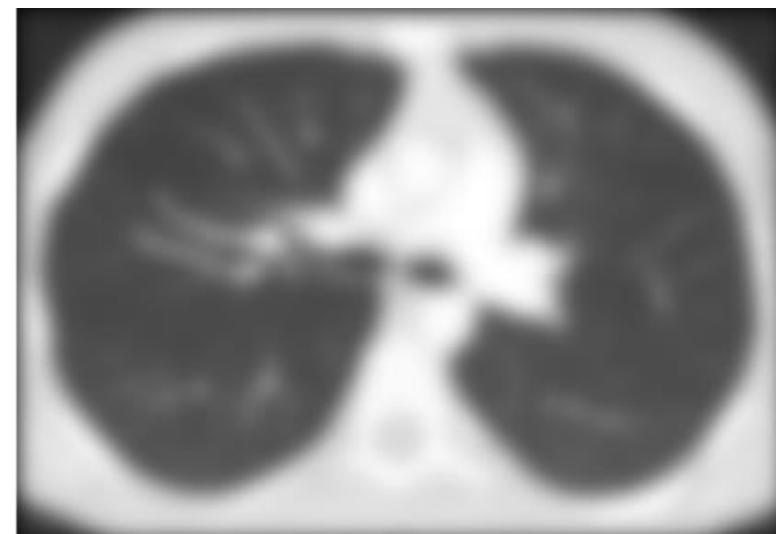
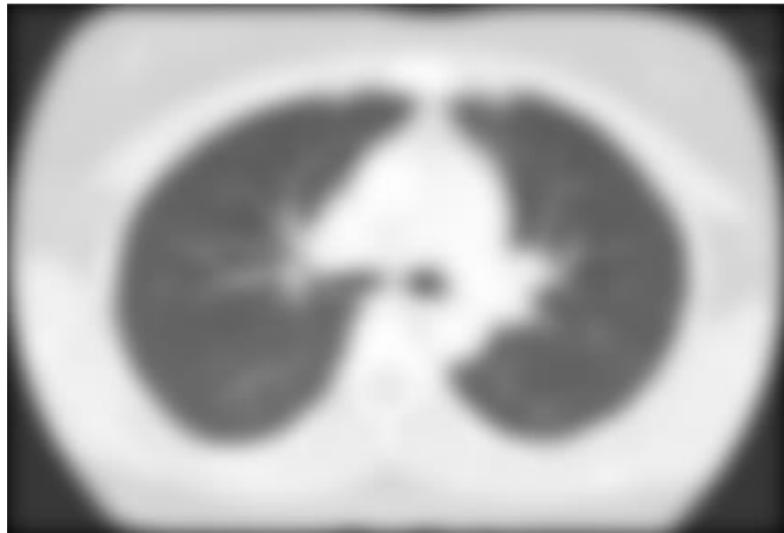
$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

Use Basis Functions



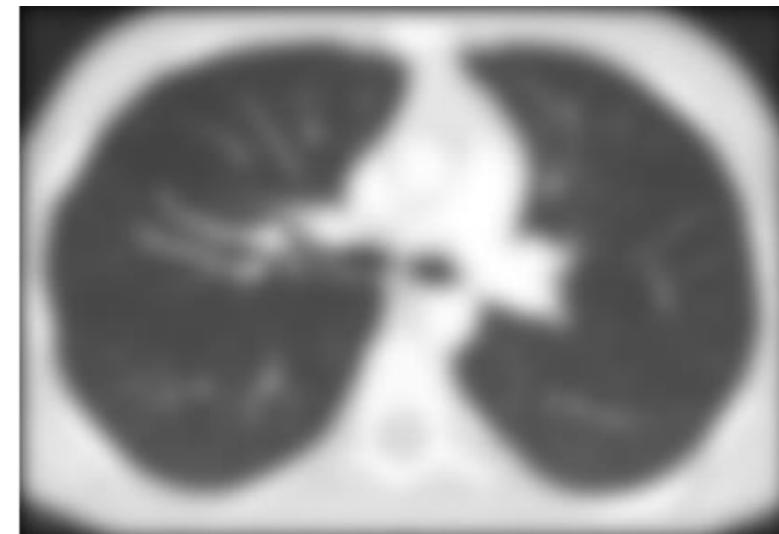
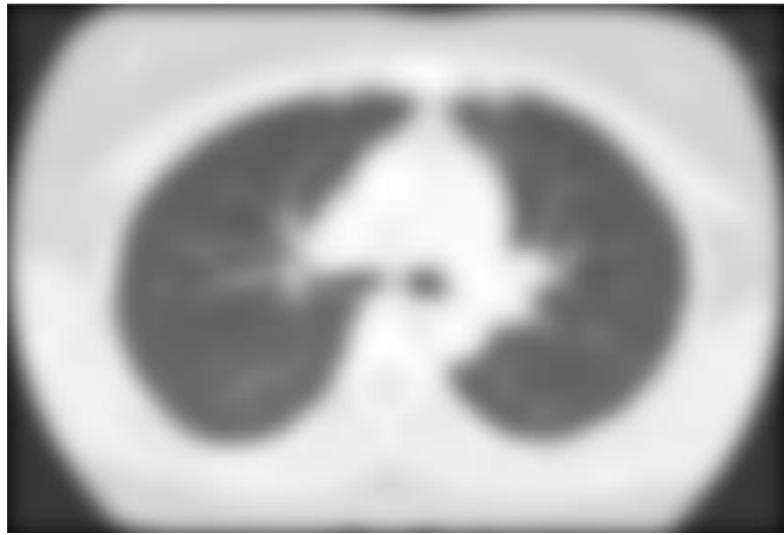
$$\Delta_x(x, y) = \sum_{k=1}^n \alpha_k B_k(x, y)$$

Coarse-to-Fine Alignment

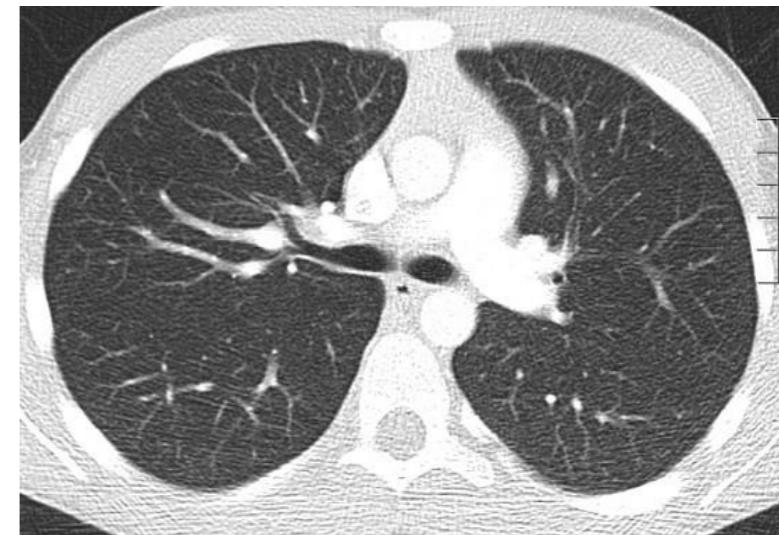


- Few basis function at coarse scale

Coarse-to-Fine Alignment

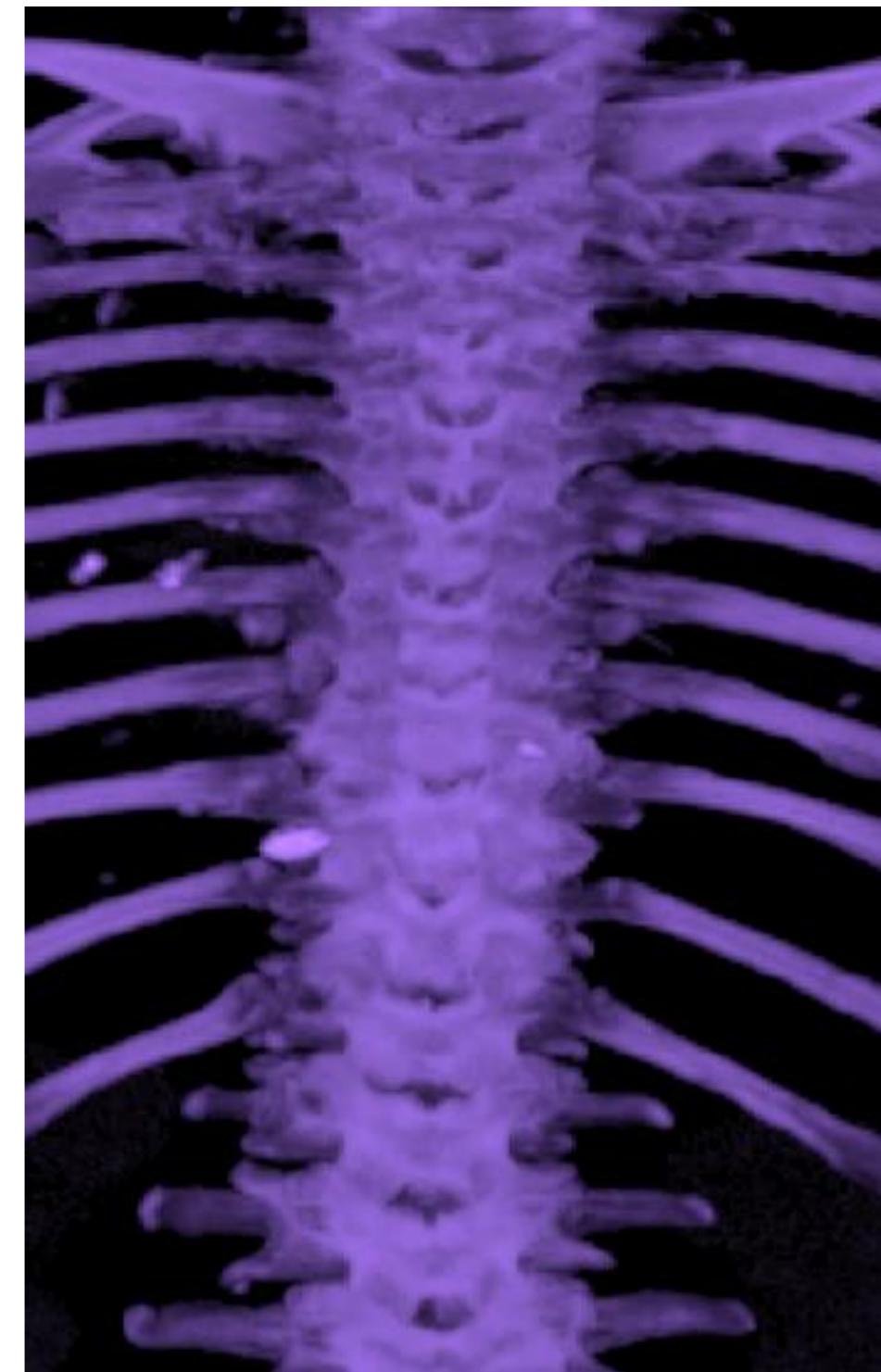
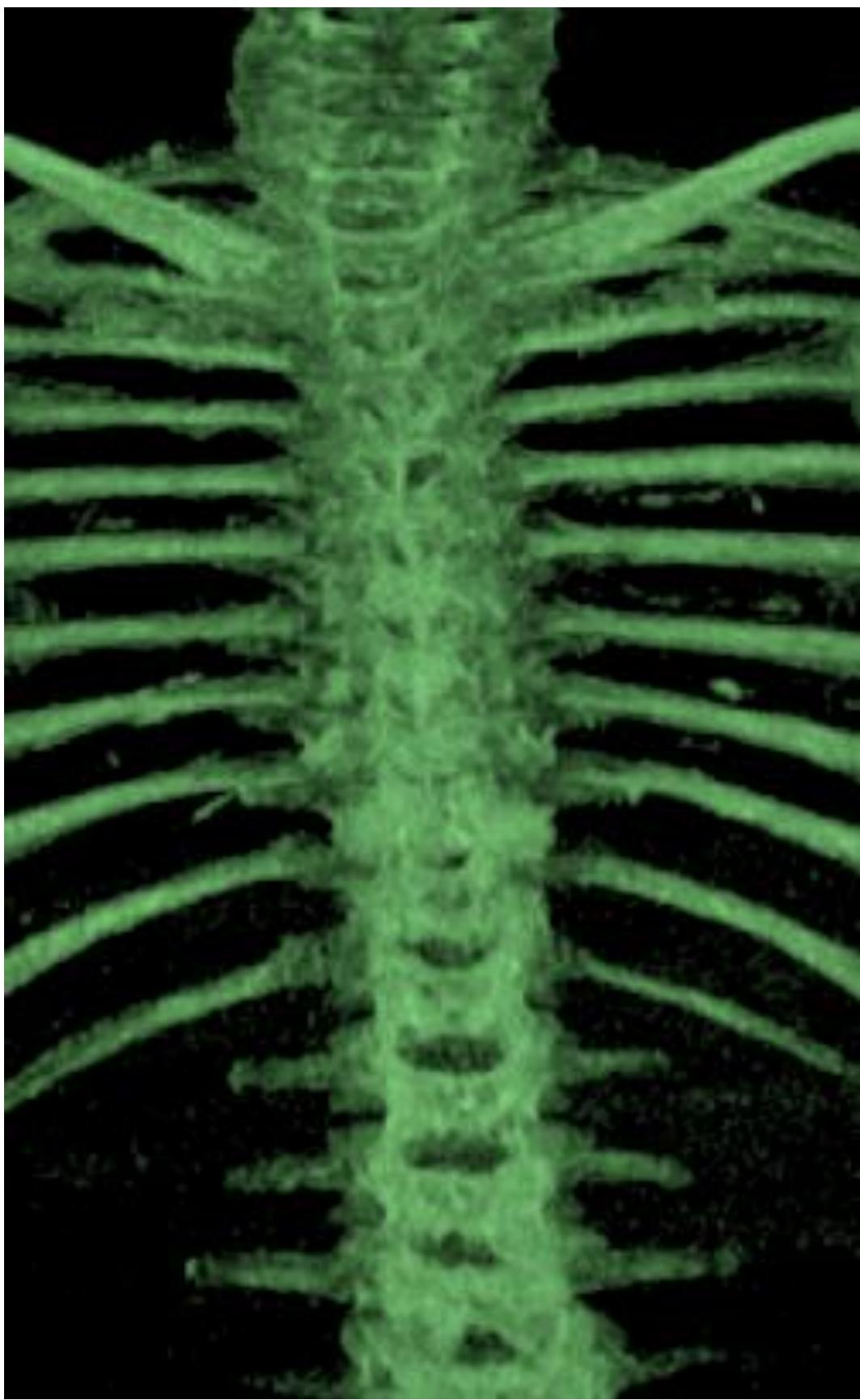


- Few basis function at coarse scale

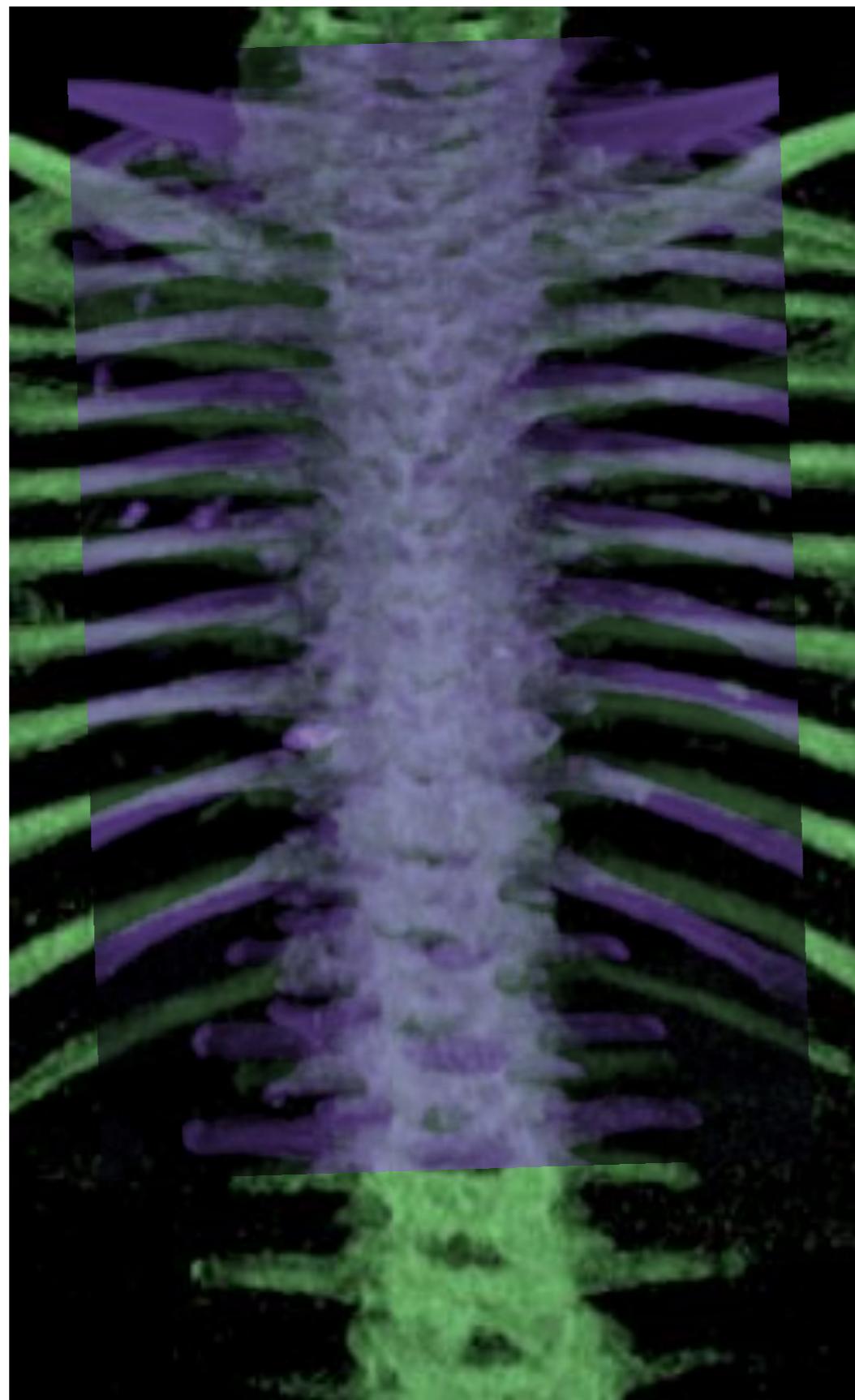


- Many basis functions at fine scale

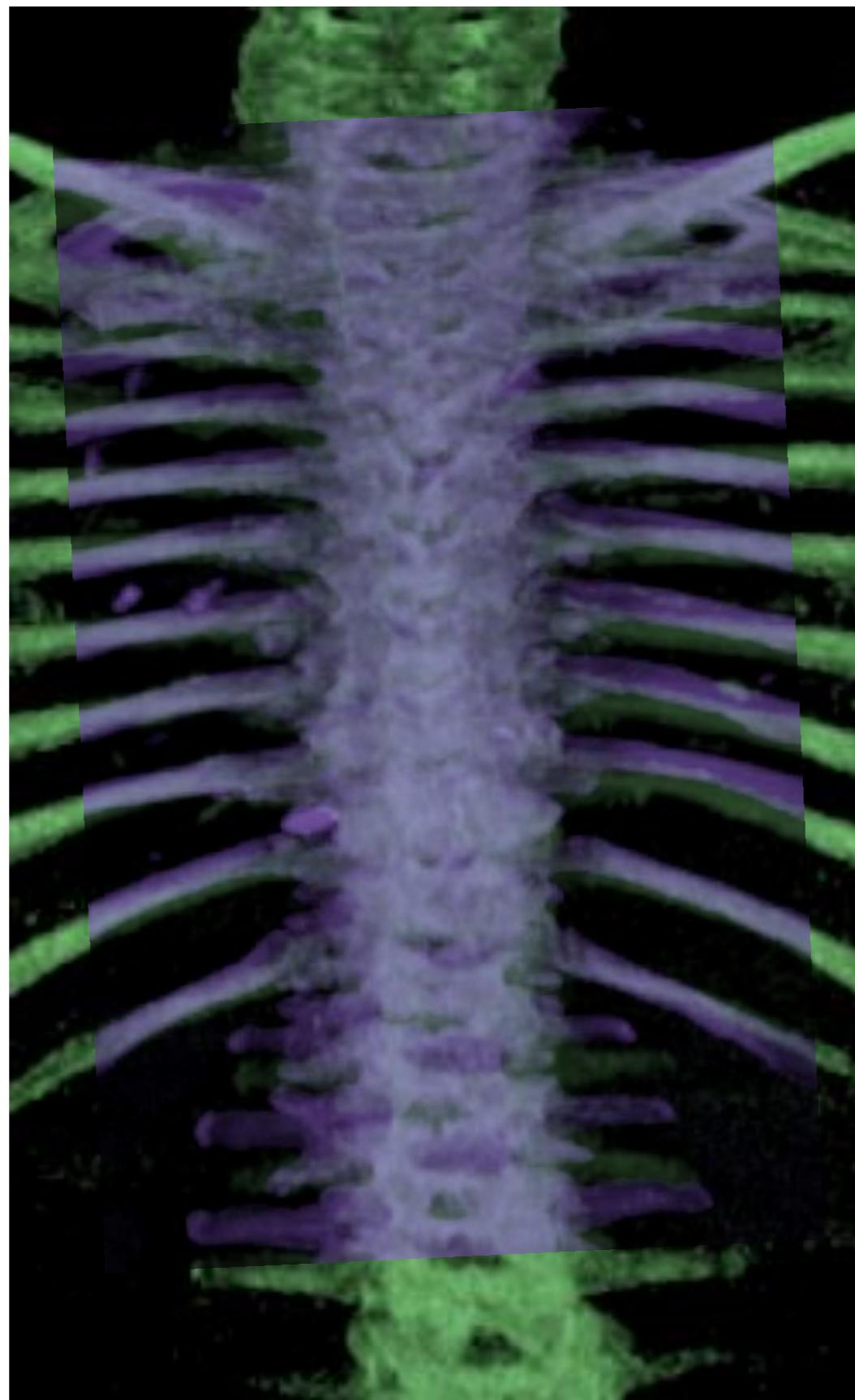
Local Minima



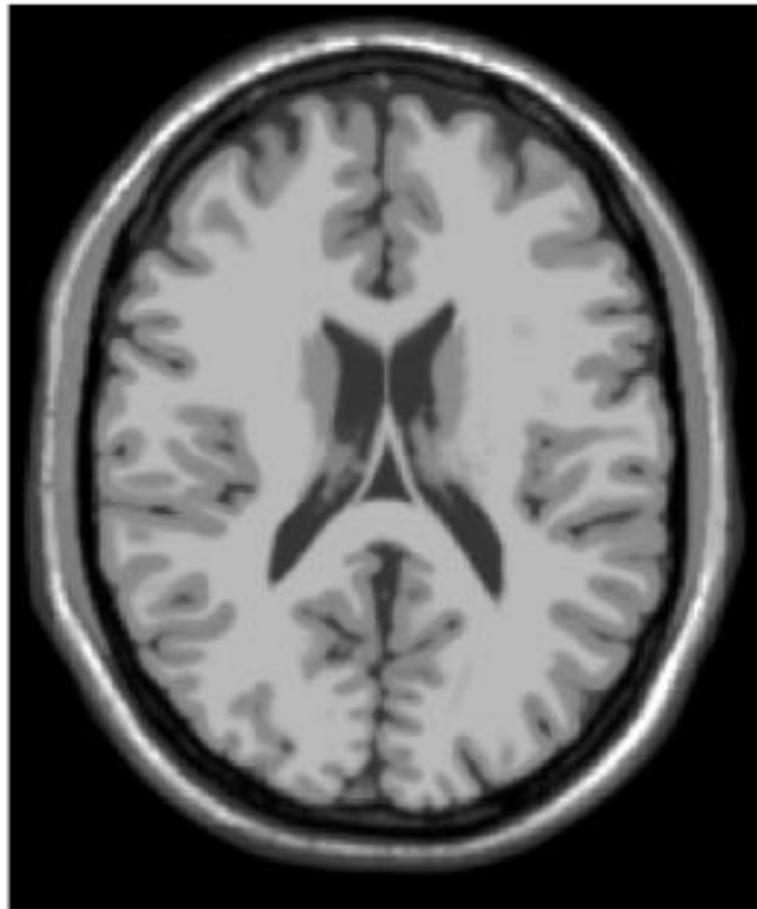
Local Minima



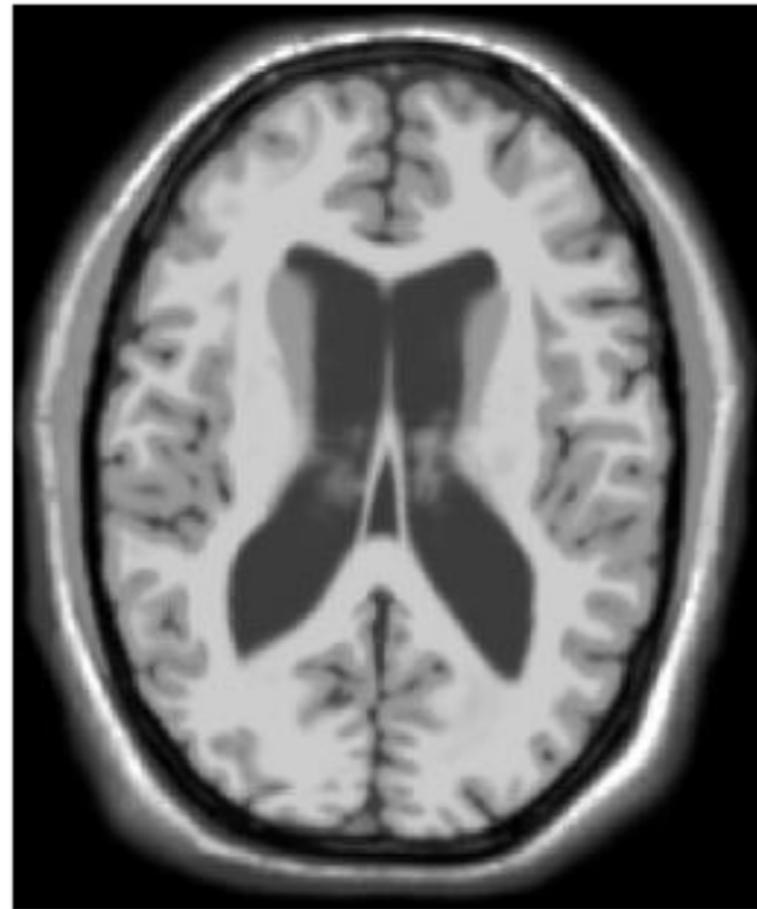
Local minima



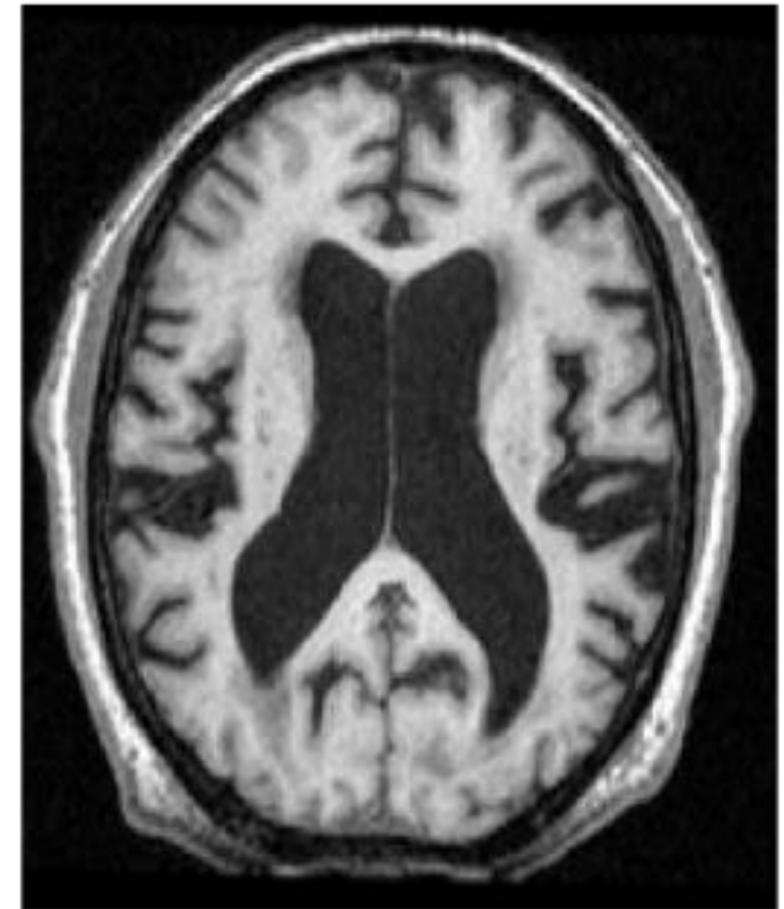
Example



source



warped



target

Lessons Learned

- Main lessons from this lecture
 - Image alignment via transformation estimation
 - Different types of transformation
 - Minimal and least squares solvers for similarity transformations
 - Image warping from target to source
- Next lecture: Camera Geometry

Next Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	
Feb. 3	Convolutional neural networks	Lab 2
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	
Feb. 13	Image registration	Lab 3
Feb. 17	Camera Geometry	
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	Lab 4
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	