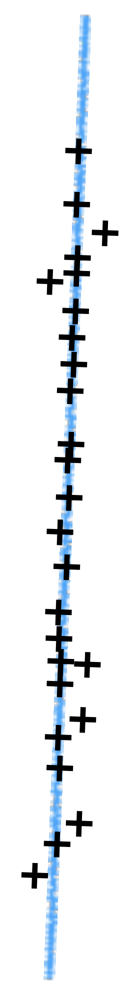
SSY097 - Image Analysis

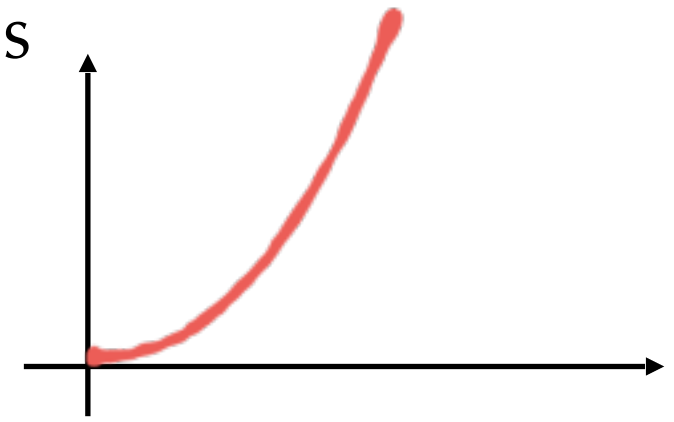
Lecture 8 - Image Registration

Torsten Sattler
(slides adapted from Olof Enqvist)

Last Lecture



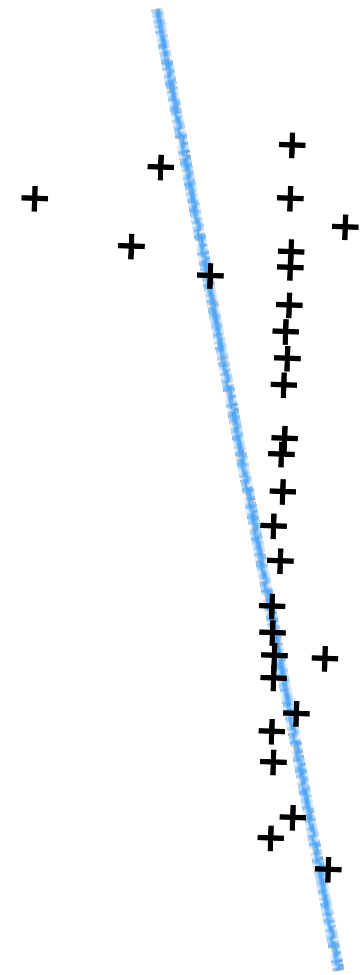
ℓ^2-solution



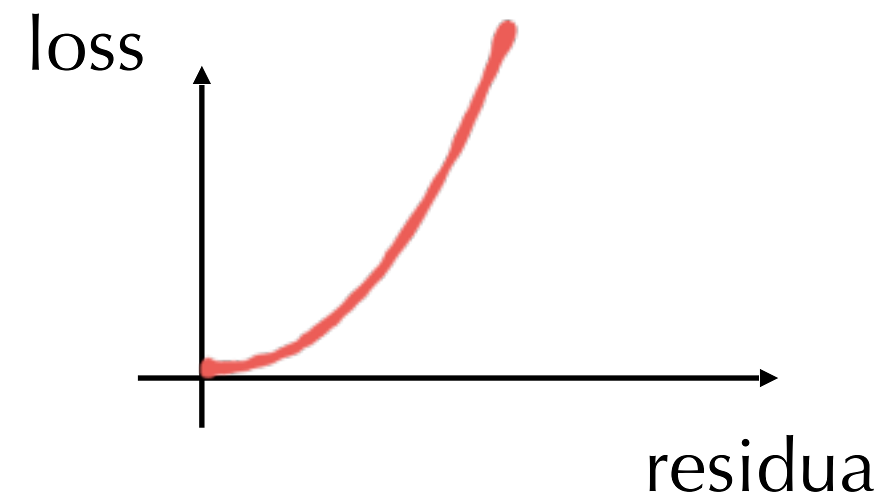
$$\min_{\theta} \sum r_i(\theta)^2$$

Robust Model Fitting

Last Lecture



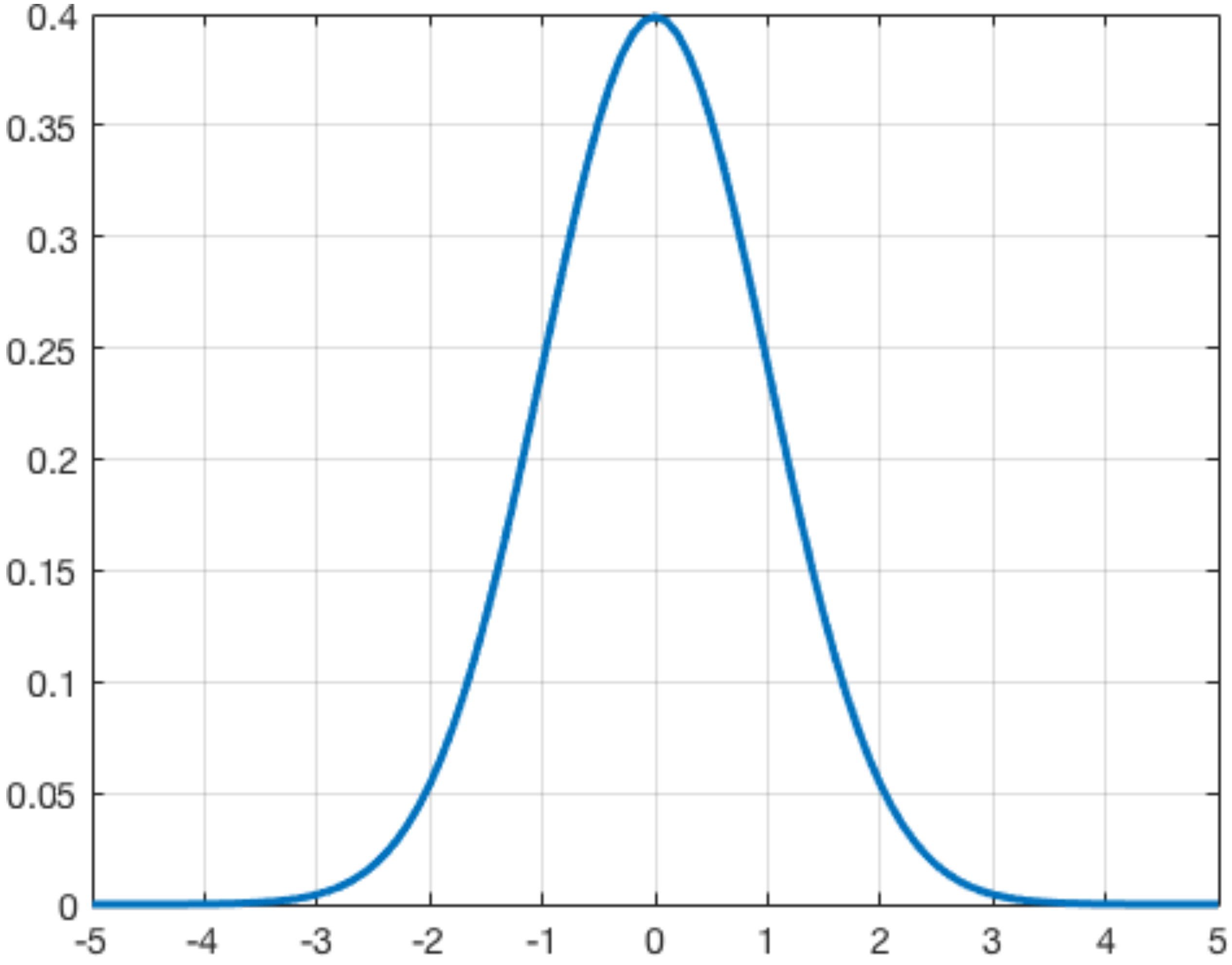
ℓ^2-solution

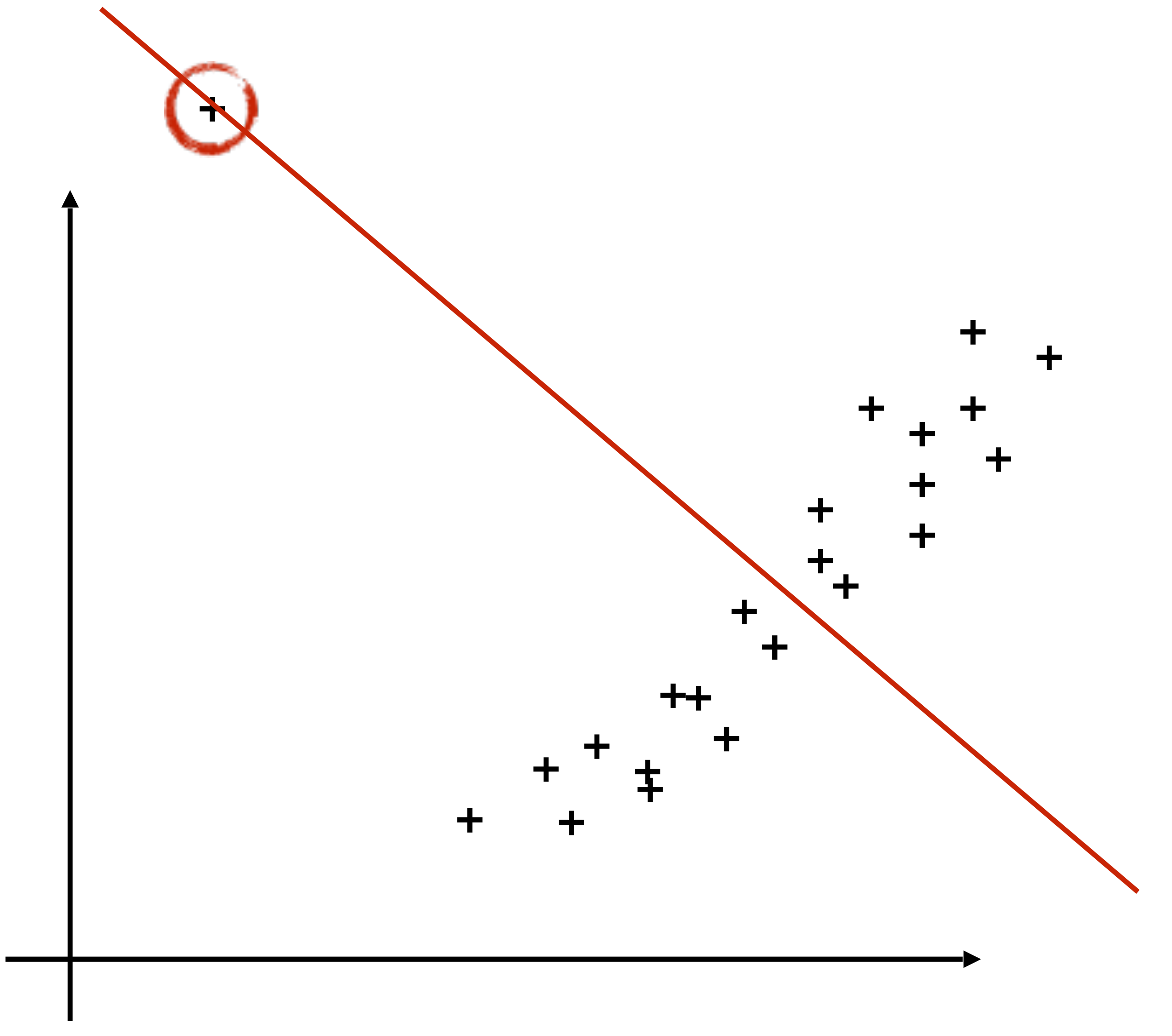


$$\min_\theta \sum r_i(\theta)^2$$

Robust Model Fitting

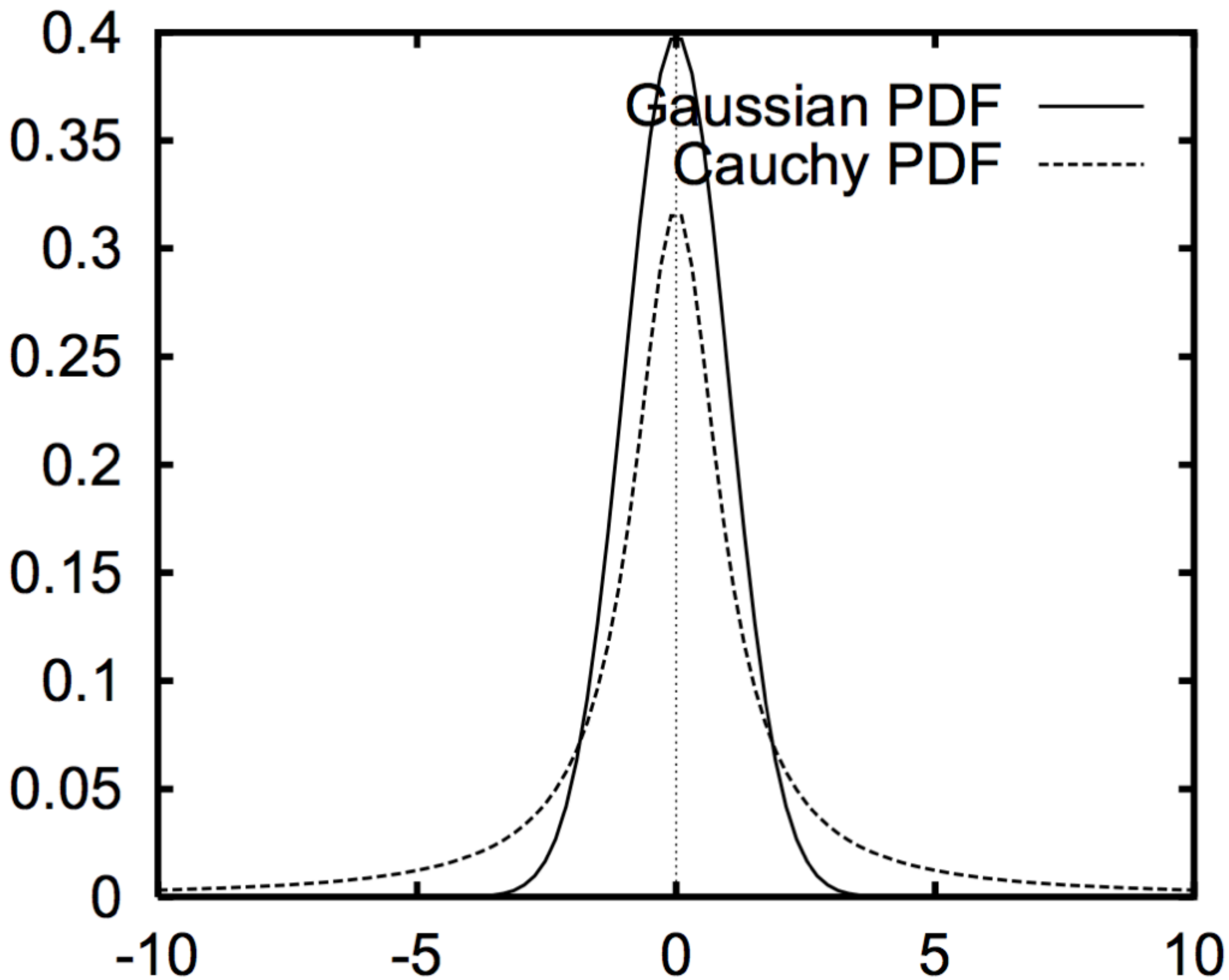
Last Lecture





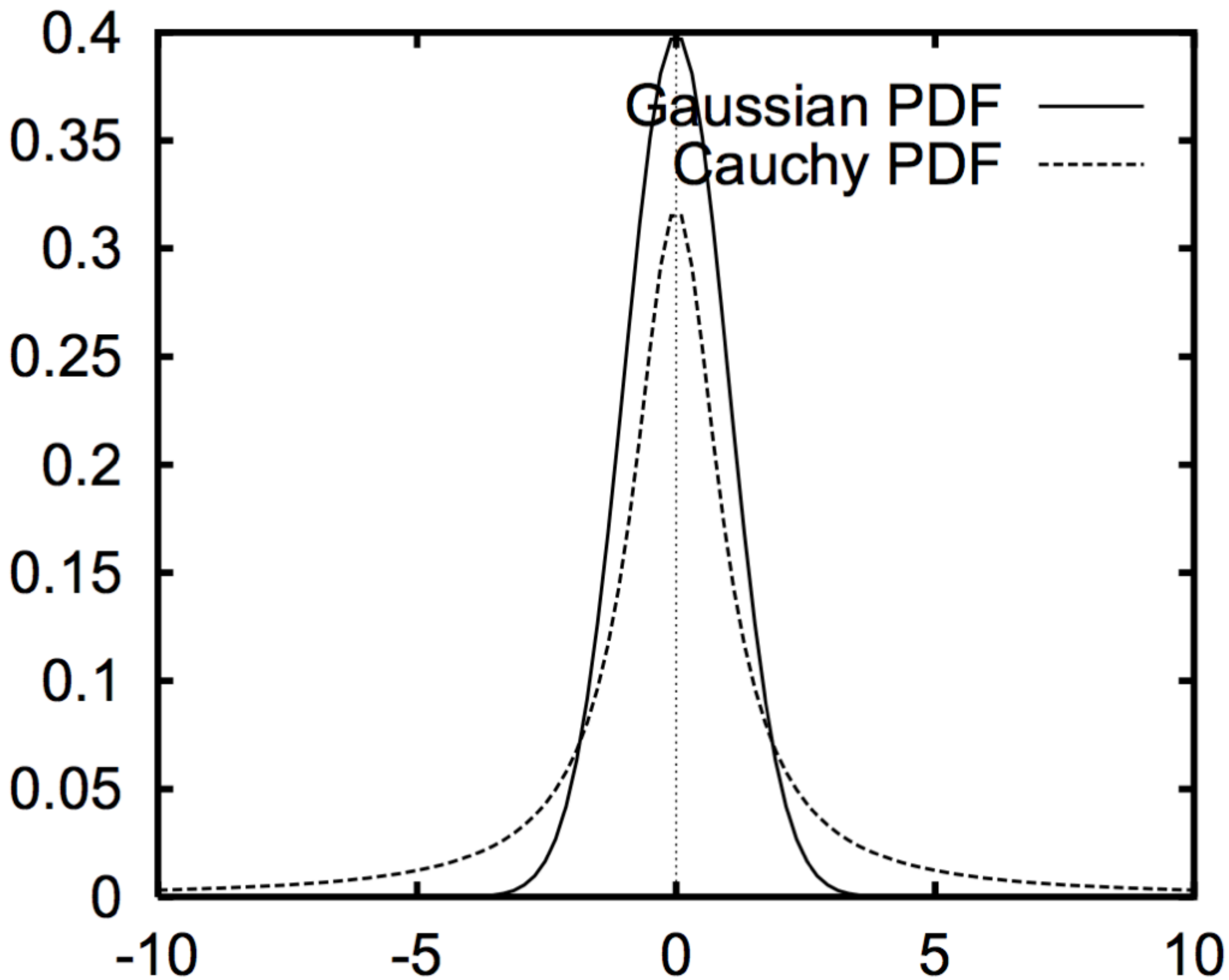
Outliers unlikely under Gaussian noise assumption

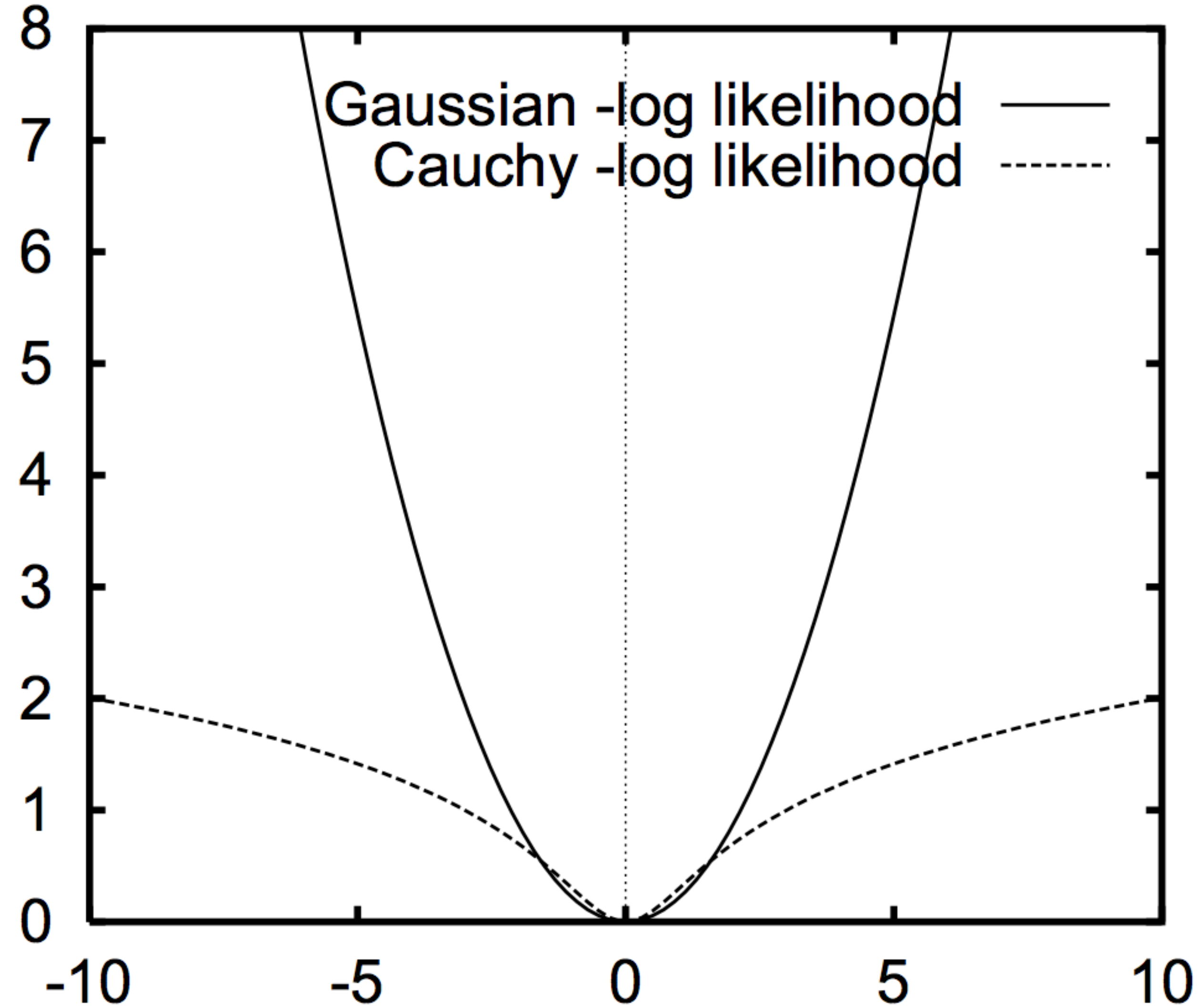
Last Lecture



Robust Loss Functions

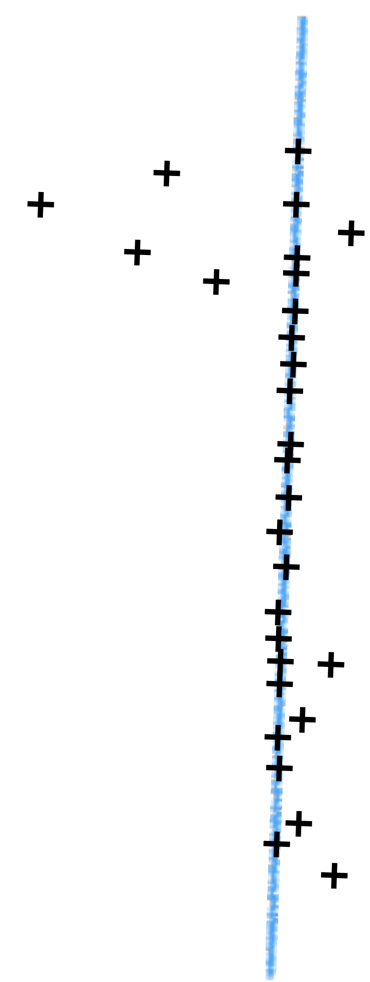
Last Lecture



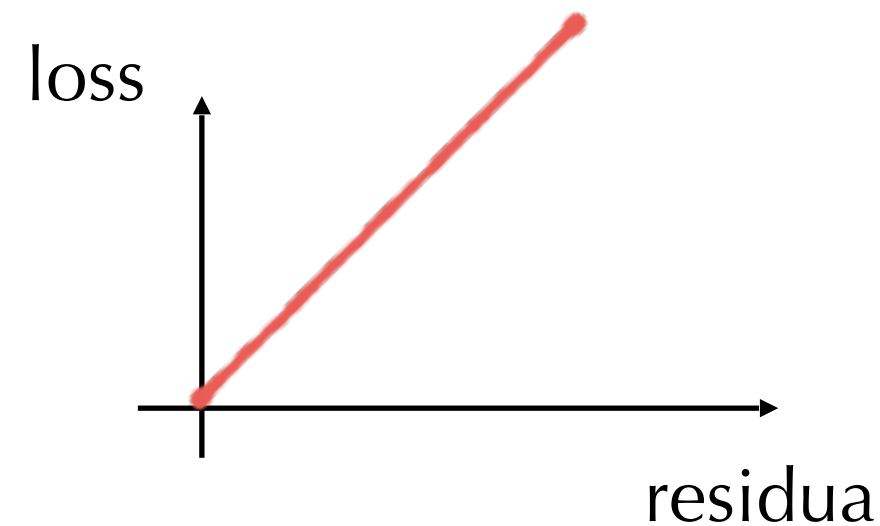


Robust Loss Functions

Last Lecture



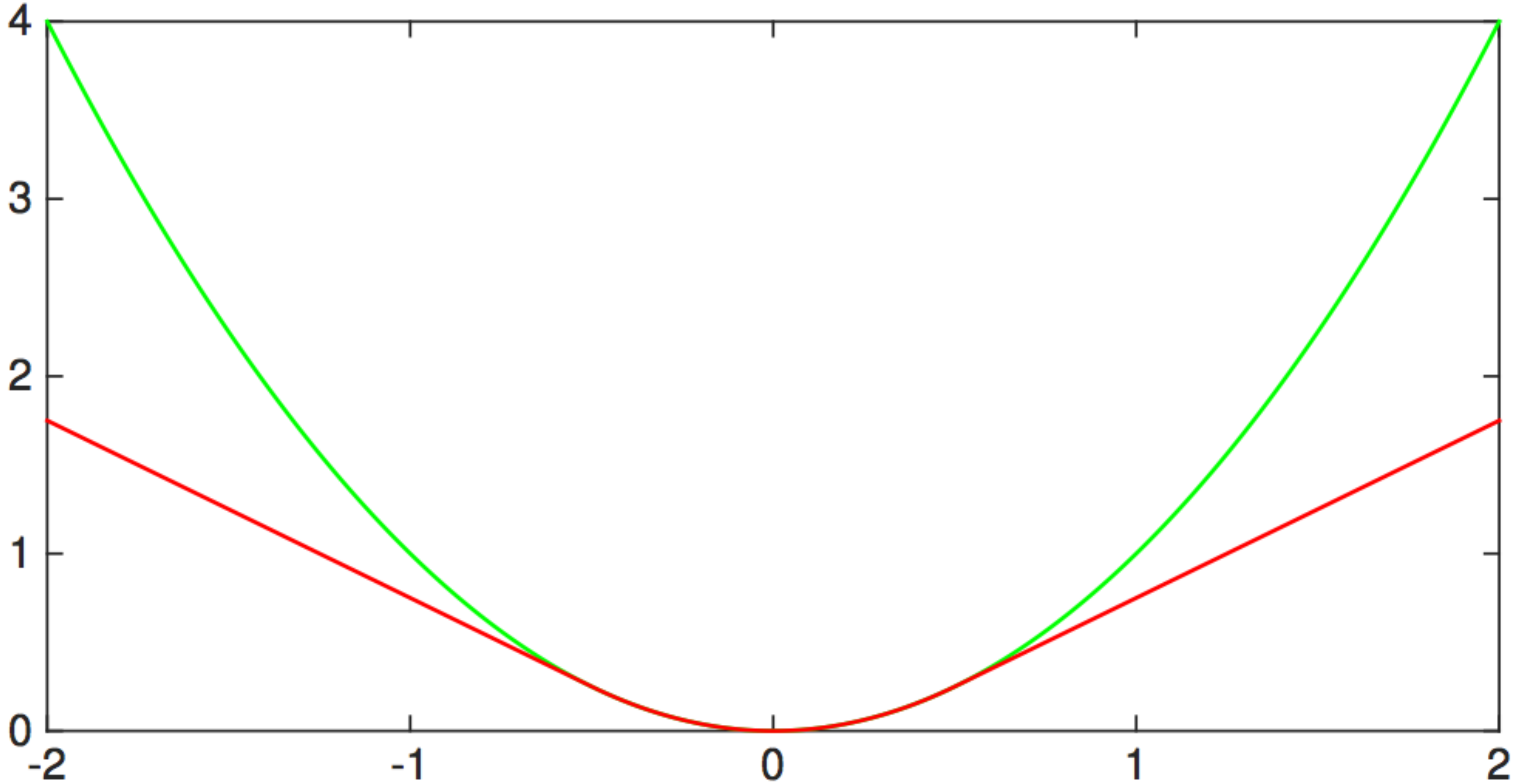
ℓ^1-solution



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

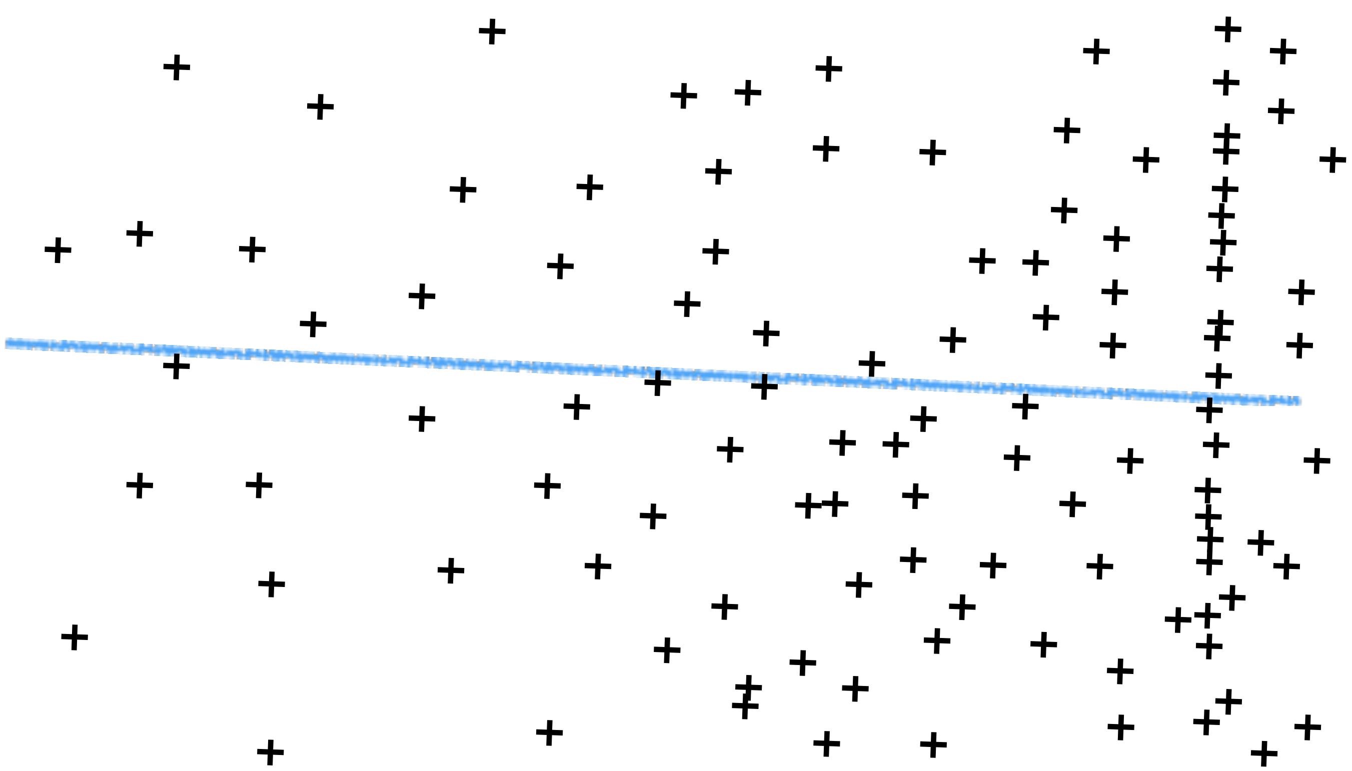
Robust Loss Functions

Last Lecture

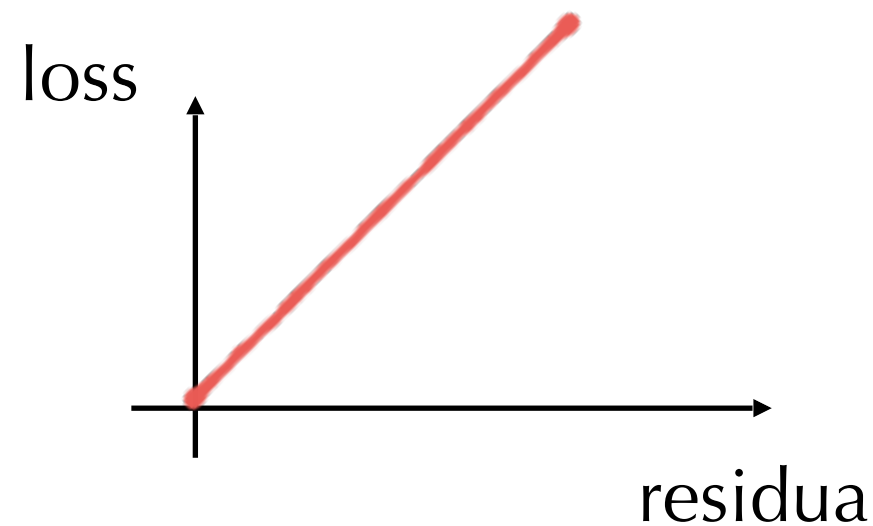


Robust Loss Functions

Last Lecture



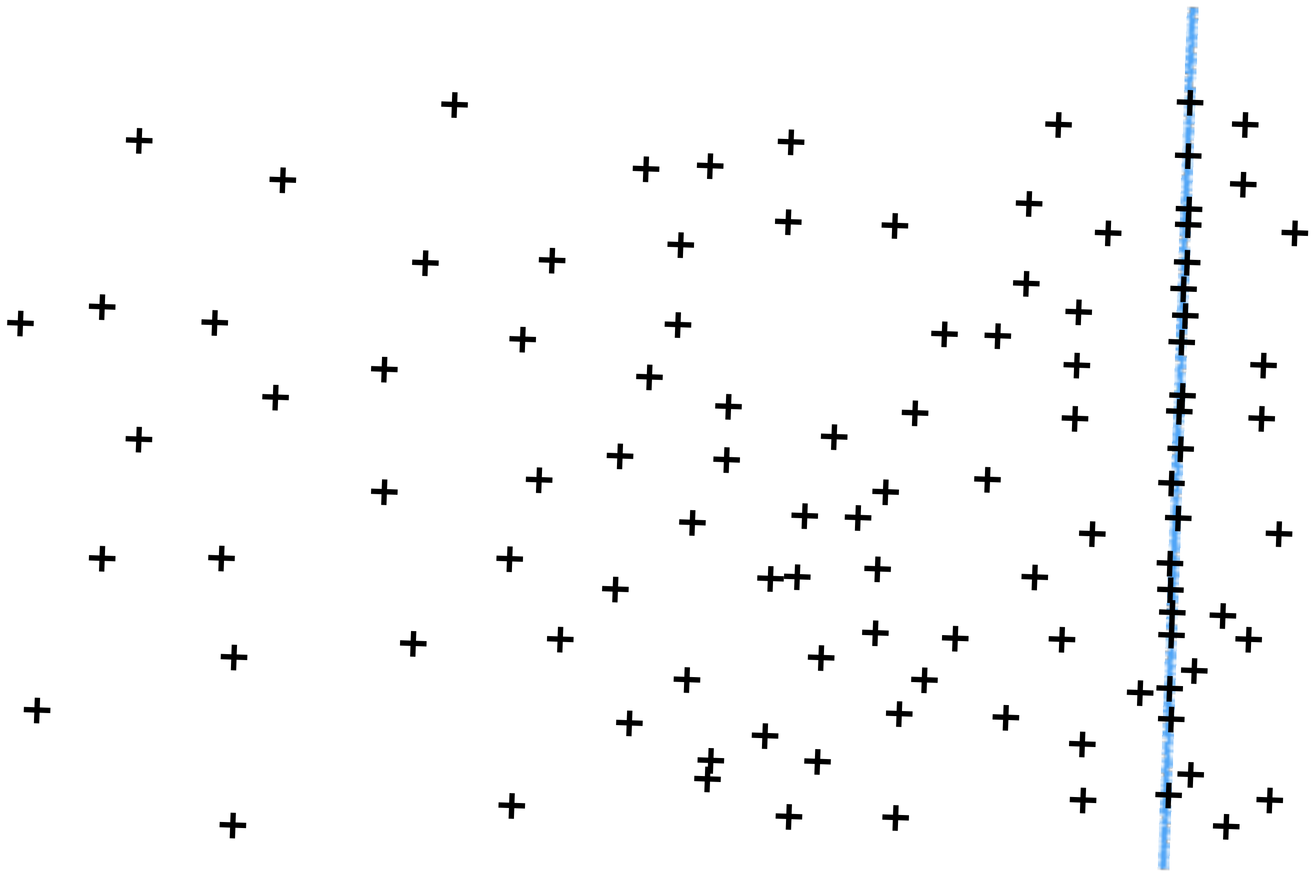
ℓ^1-solution

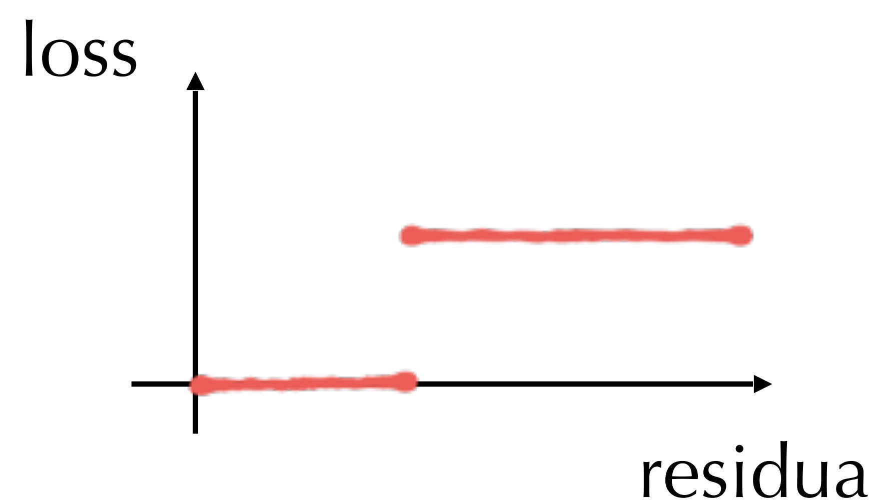


$$\min_\theta \sum |r_i(\theta)|$$

Robust Loss Functions

Last Lecture





Outlier count

Minimizing #outliers / maximizing #inliers

Last Lecture

While probability of missing correct model >η

Estimate model from *n* random data points

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

If more inliers than previous best model

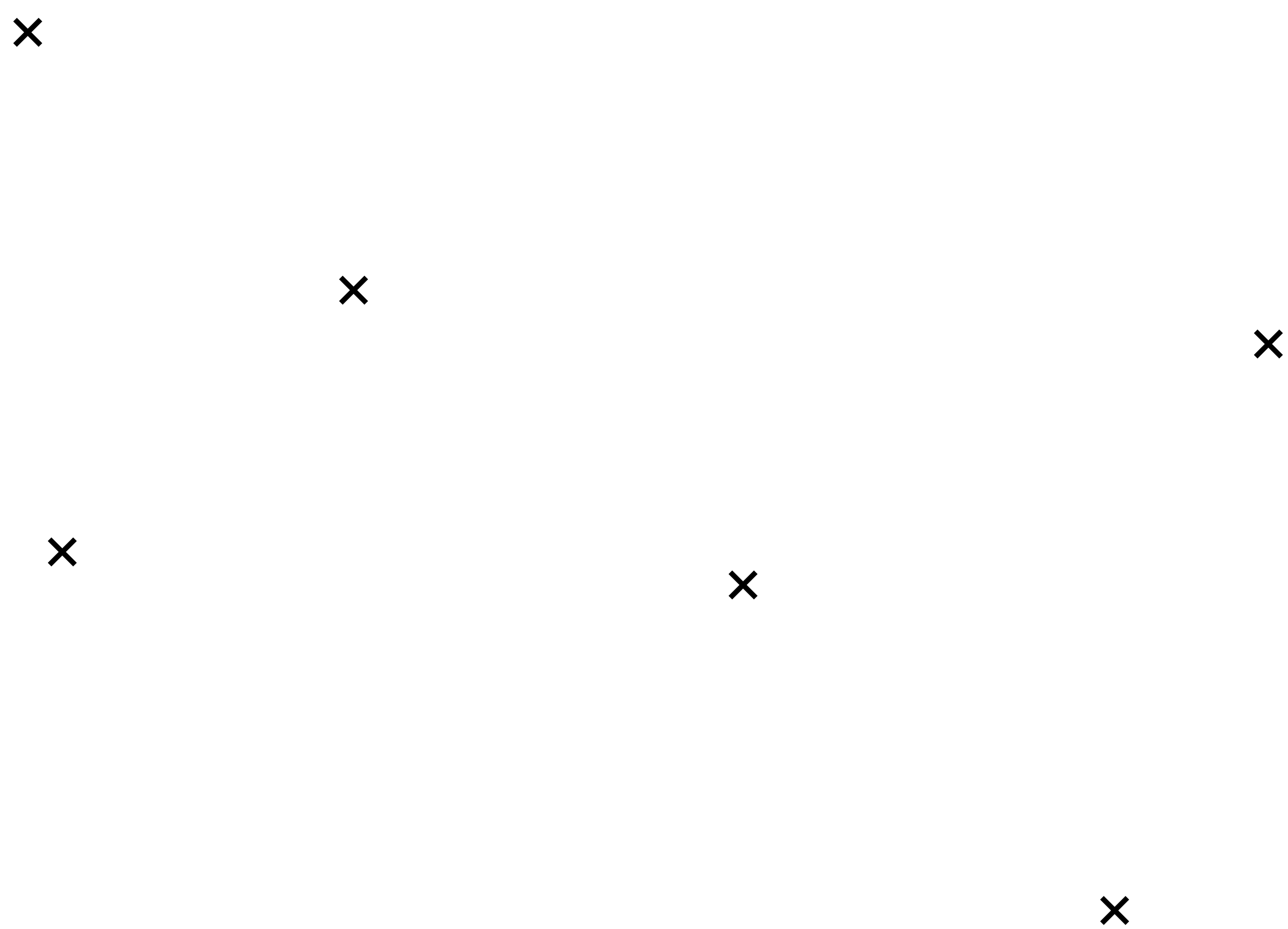
update best model

Return: Model with most inliers

RANSAC

Last Lecture

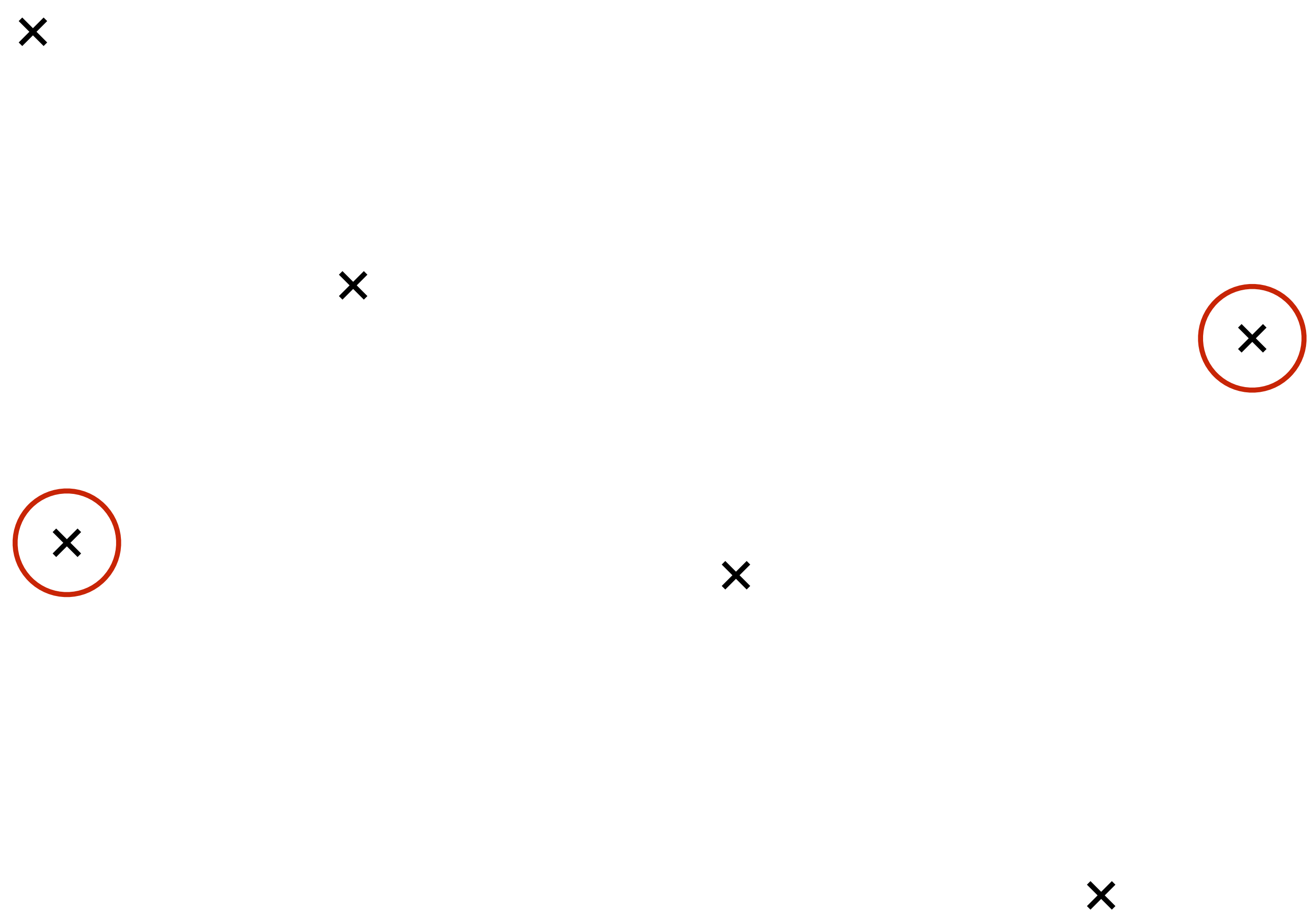
Line fitting example



RANSAC

Last Lecture

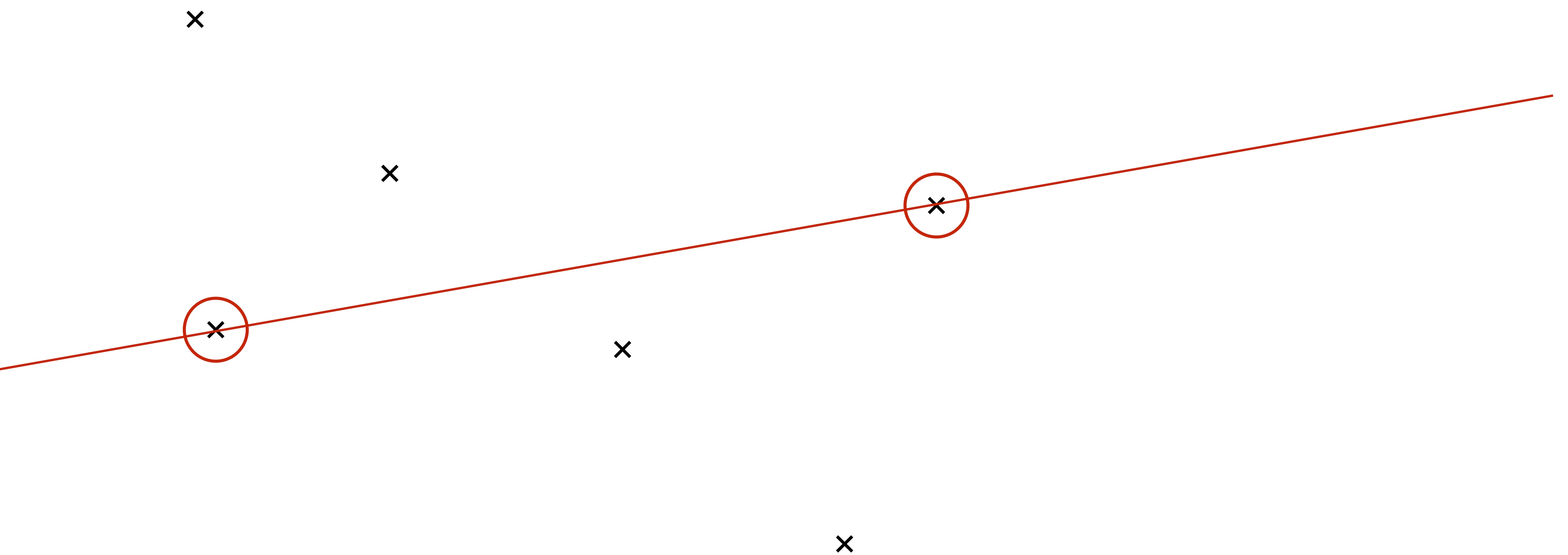
Line fitting example



RANSAC

Last Lecture

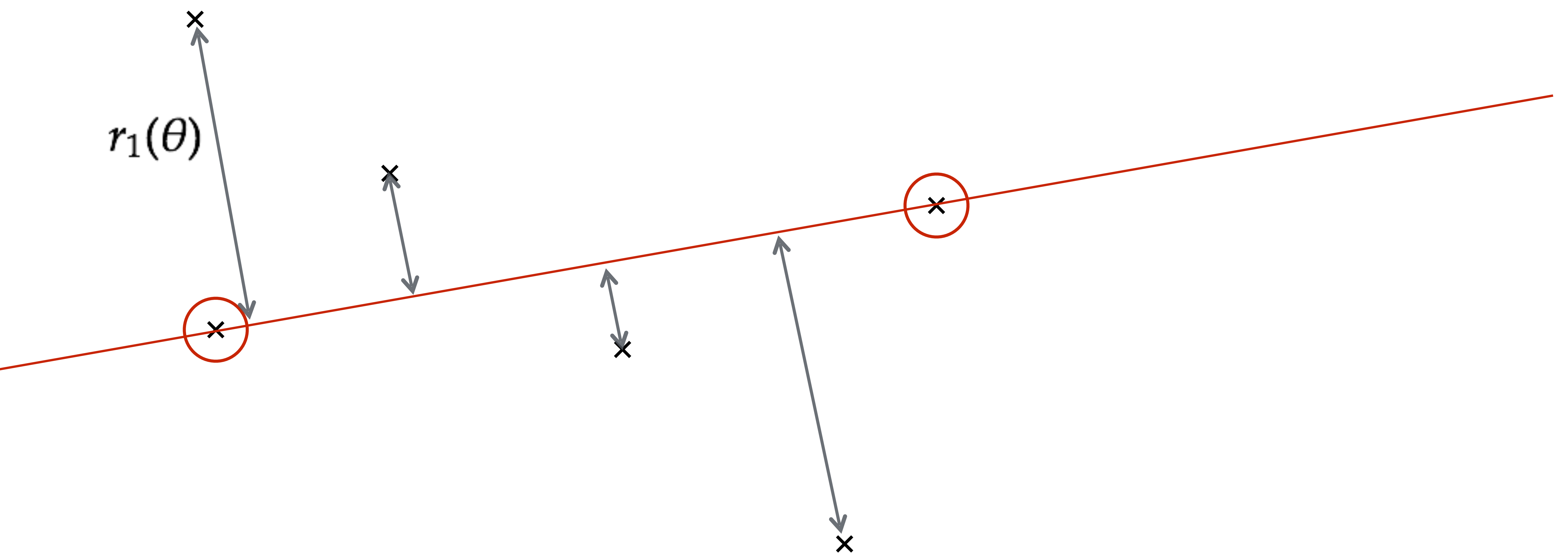
Line fitting example



RANSAC

Last Lecture

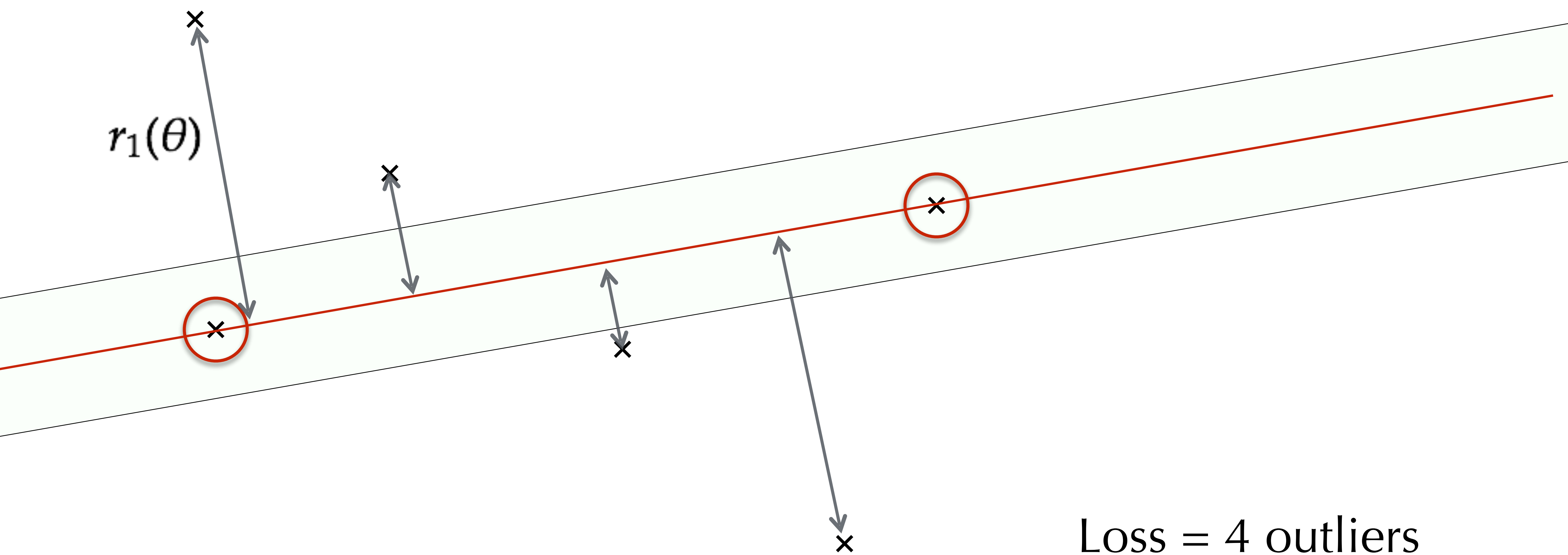
Line fitting example



RANSAC

Last Lecture

Line fitting example



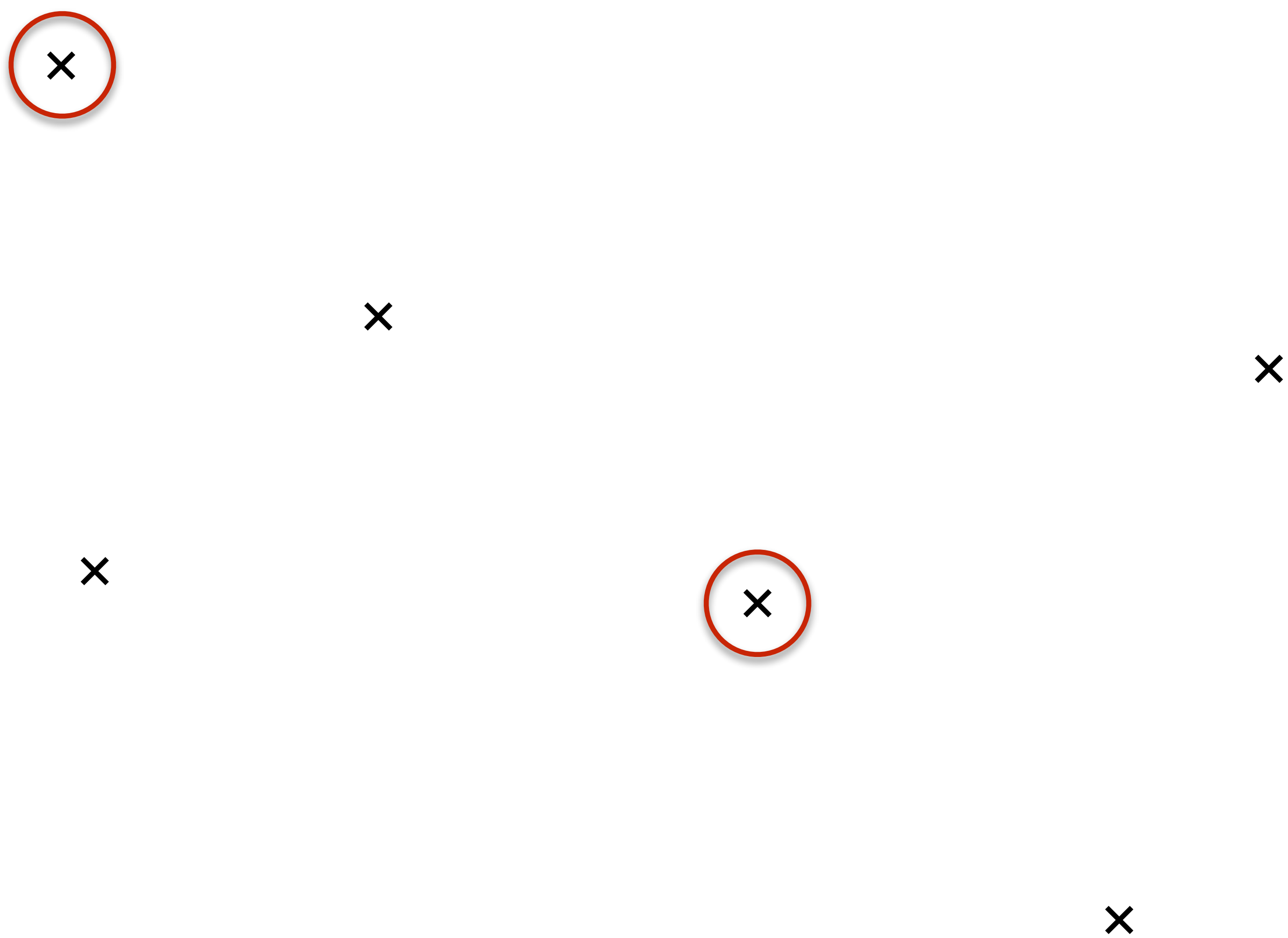
Loss = 4 outliers

RANSAC

Last Lecture

Line fitting example

Best loss so far = 4 outliers

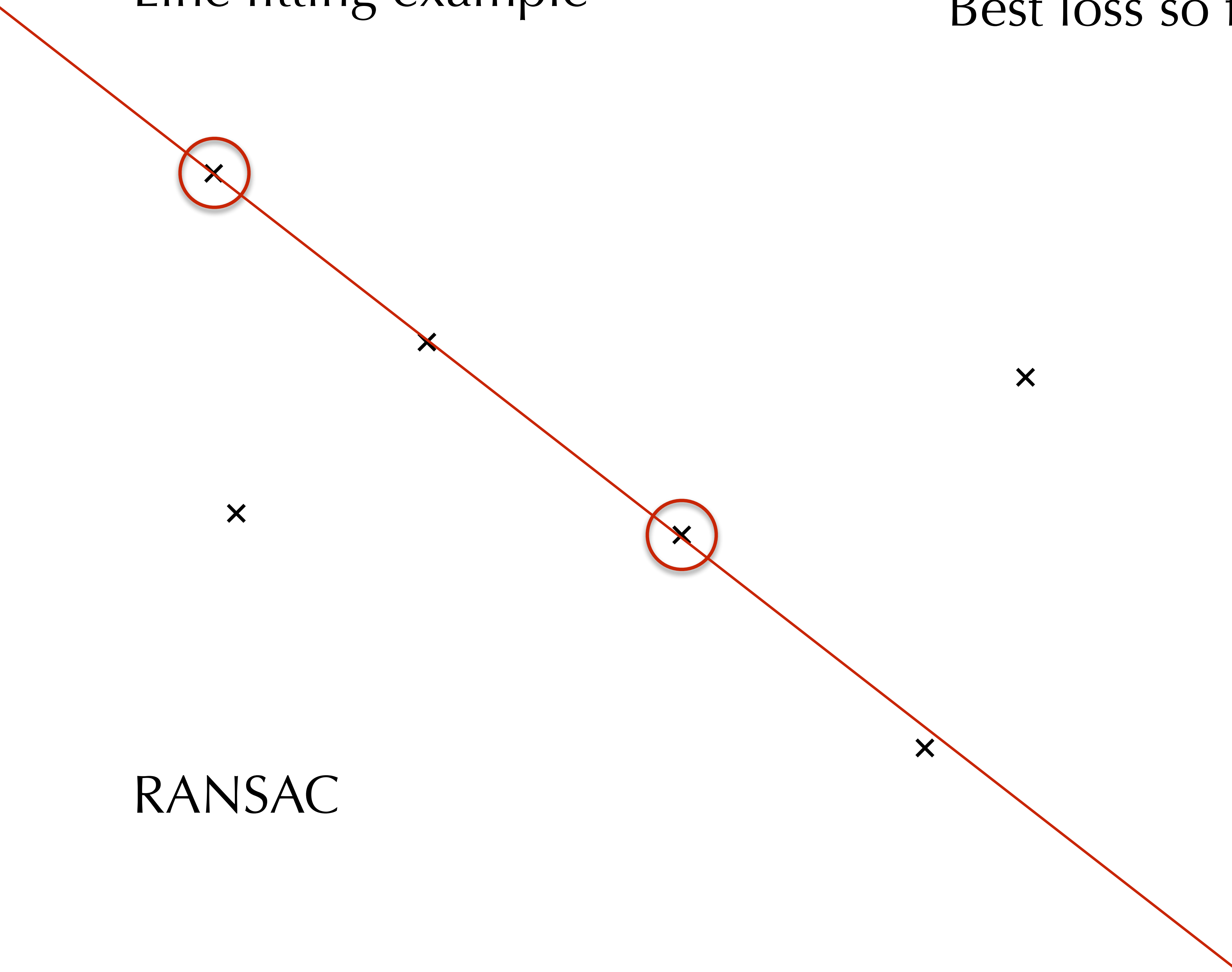


RANSAC

Last Lecture

Line fitting example

Best loss so far = 4 outliers

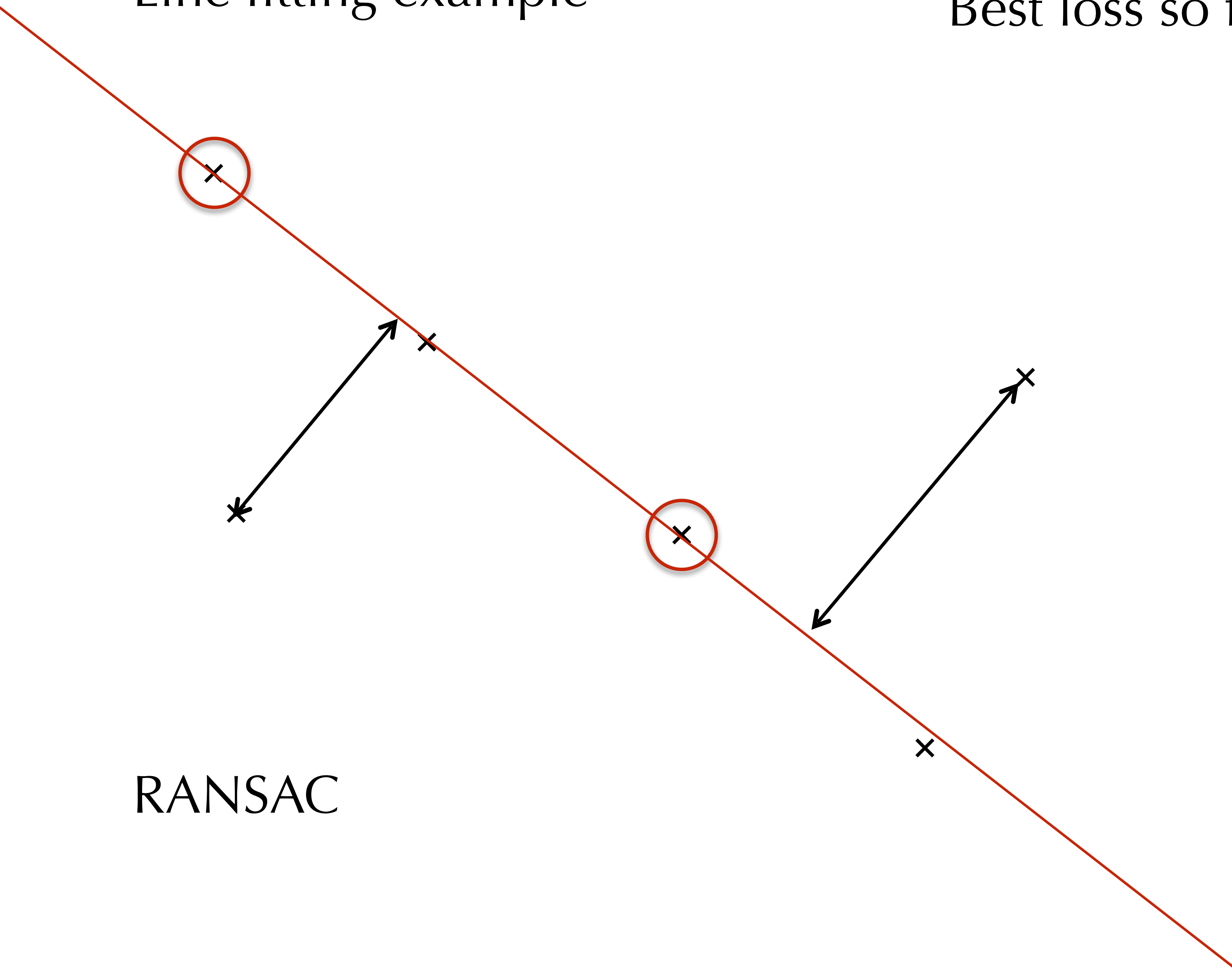


RANSAC

Last Lecture

Line fitting example

Best loss so far = 4 outliers

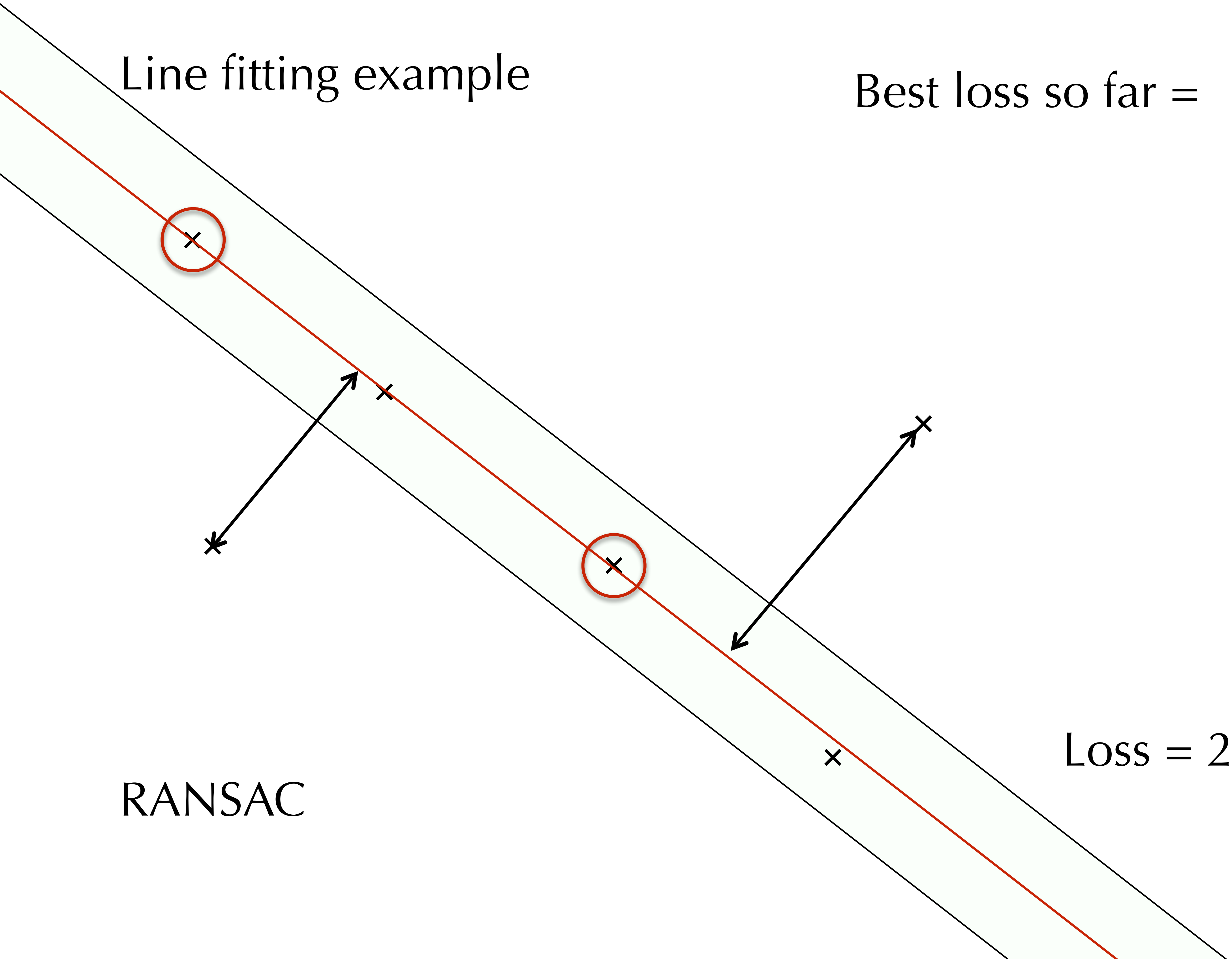


RANSAC

Last Lecture

Line fitting example

Best loss so far = 4 outliers



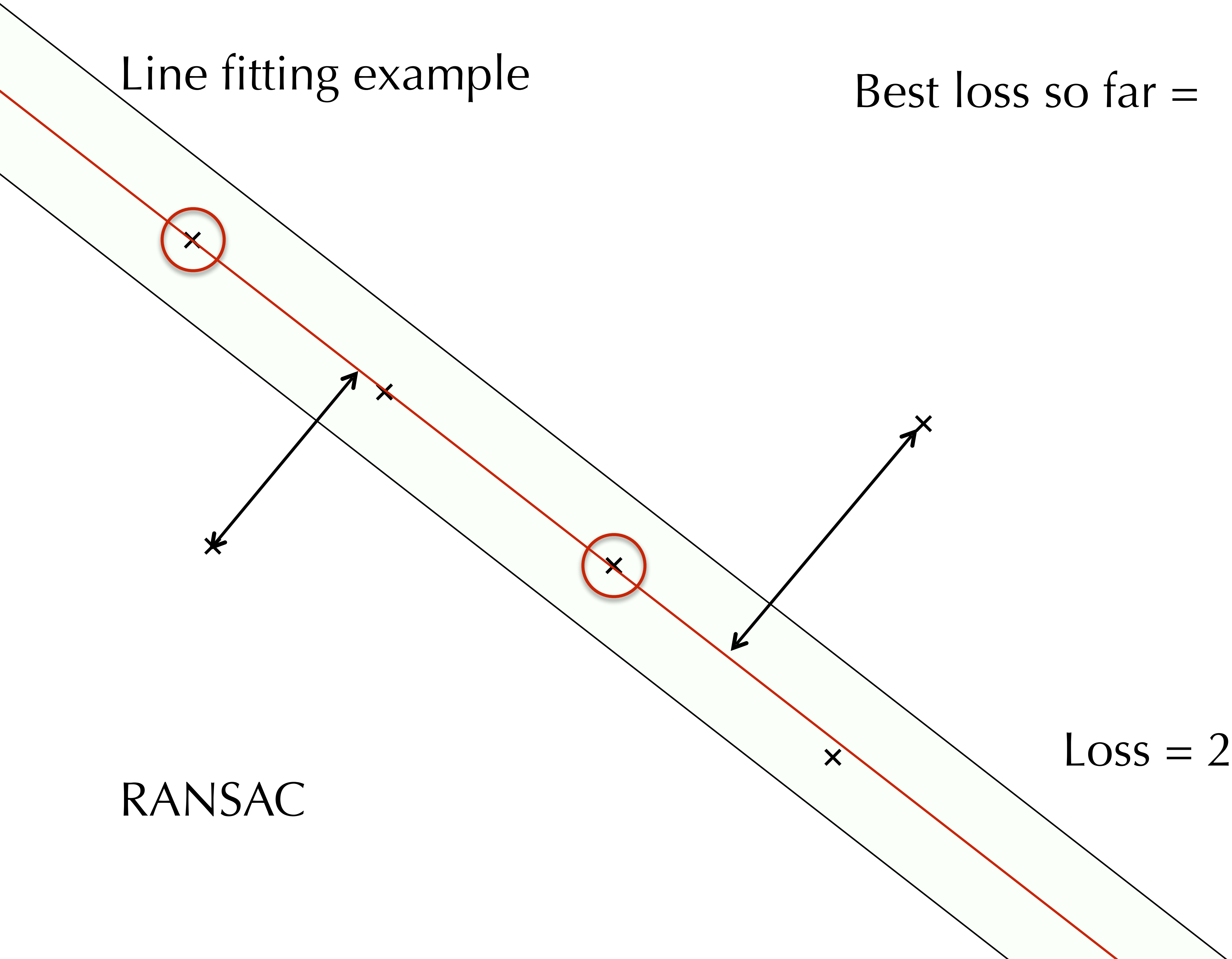
Loss = 2 outliers

RANSAC

Last Lecture

Line fitting example

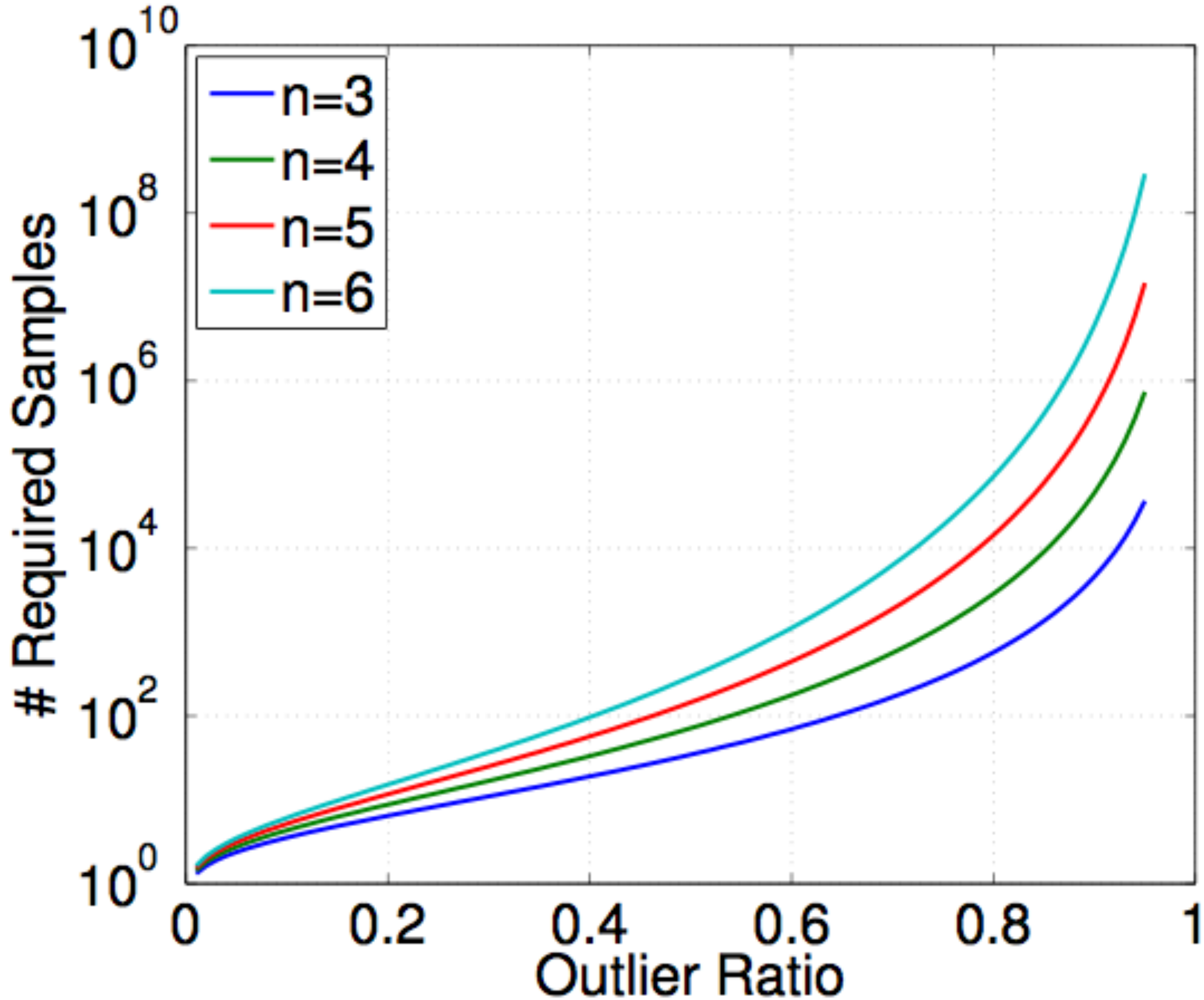
Best loss so far = 2 outliers



Loss = 2 outliers

RANSAC

Last Lecture



Minimal solvers

Today

Image Registration

SIFT Features

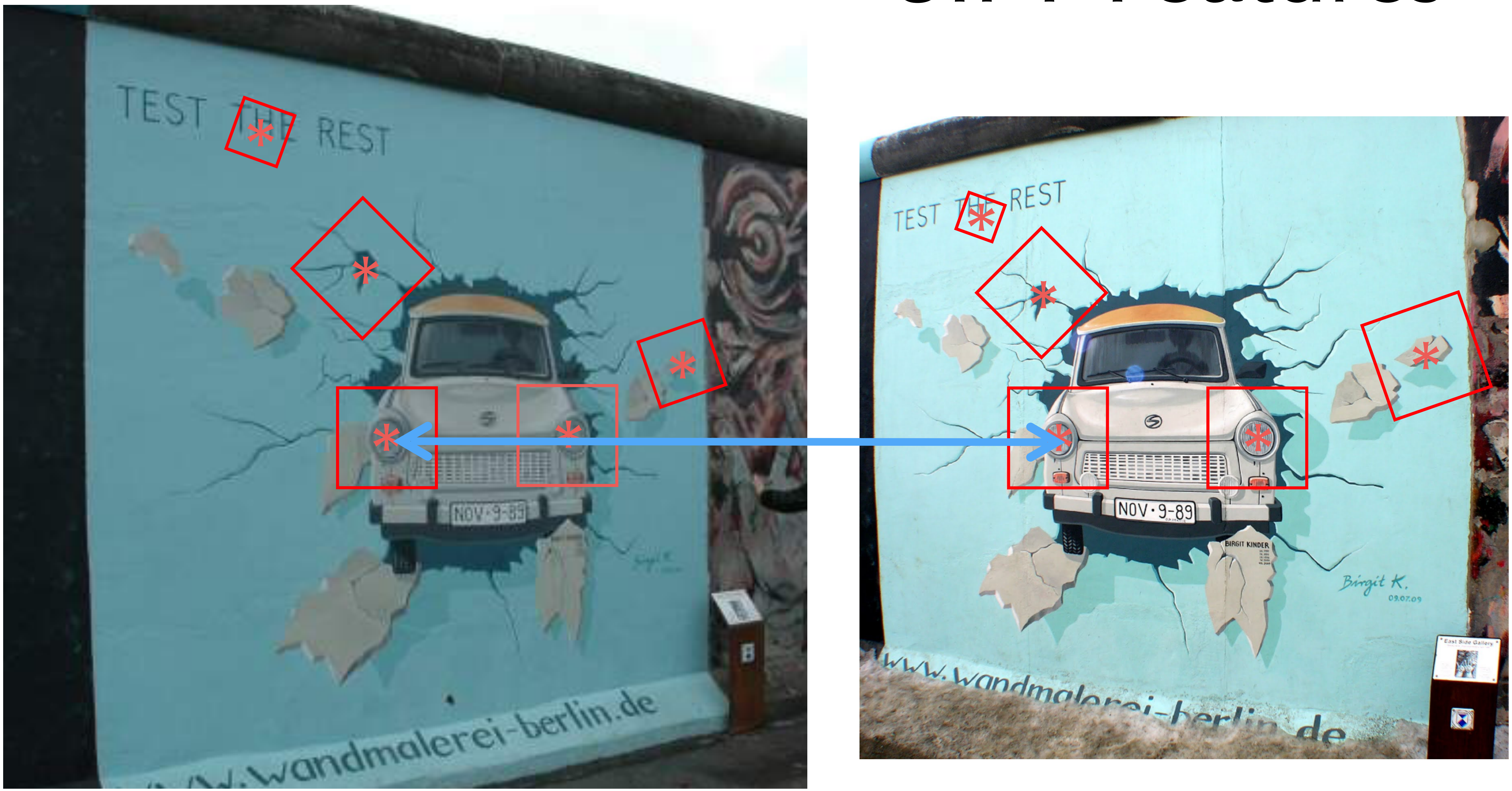


Image Stitching

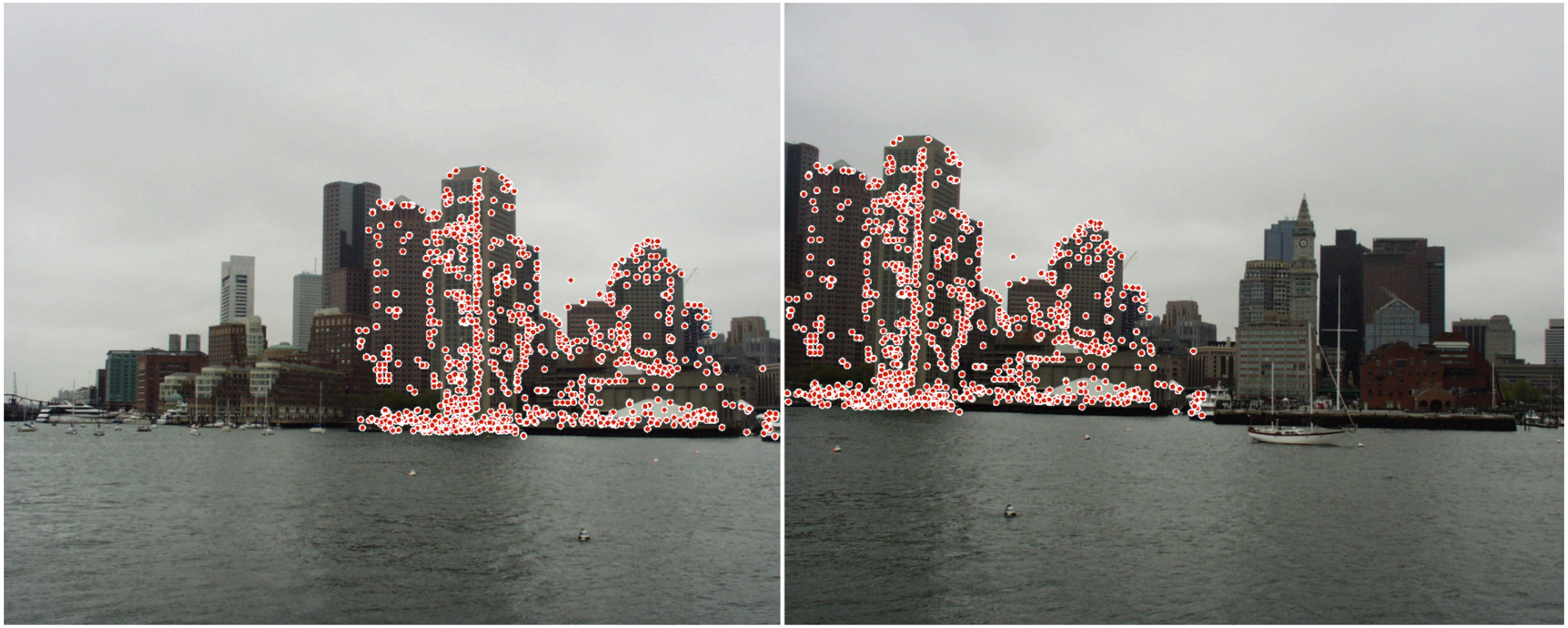
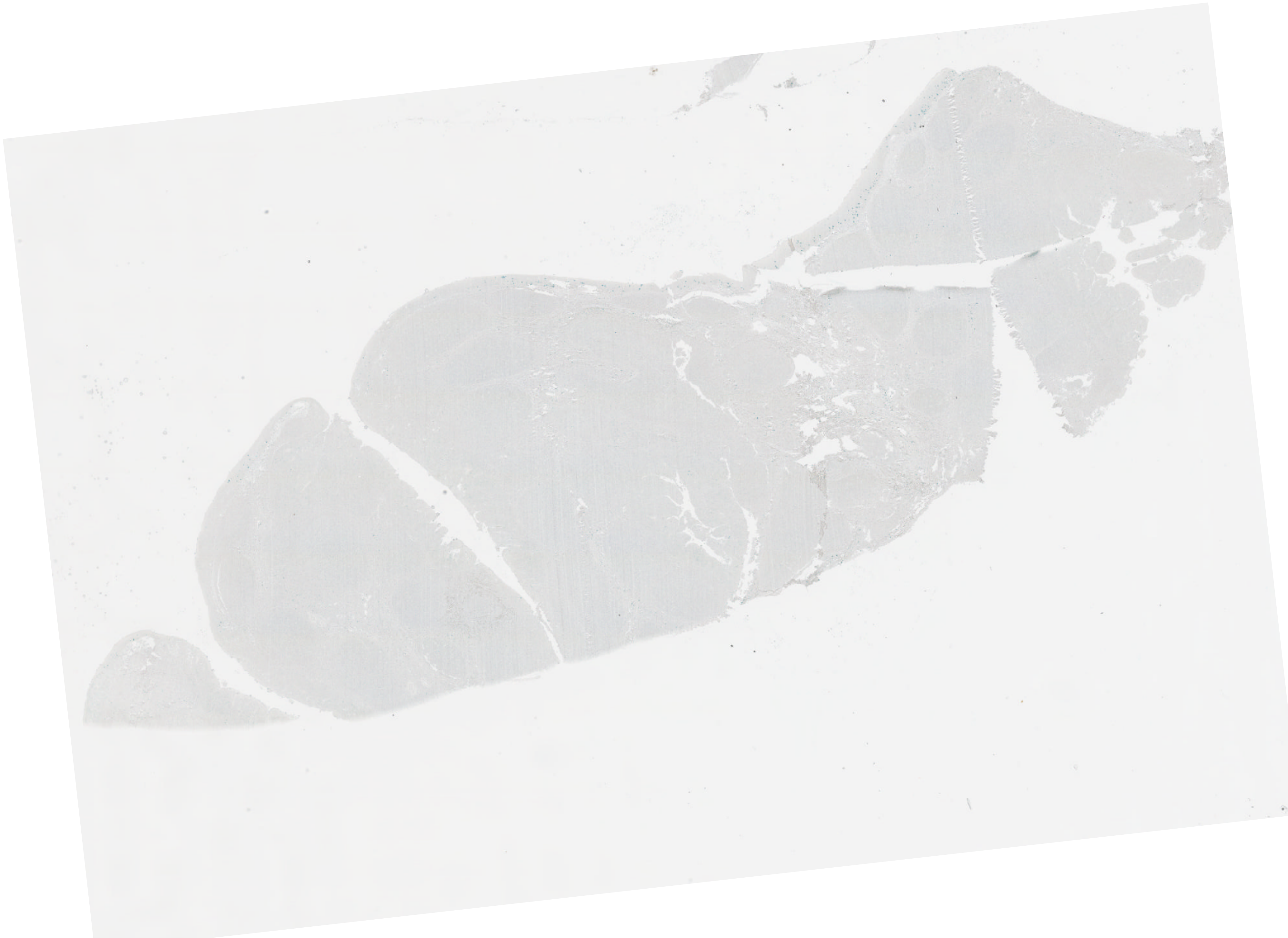
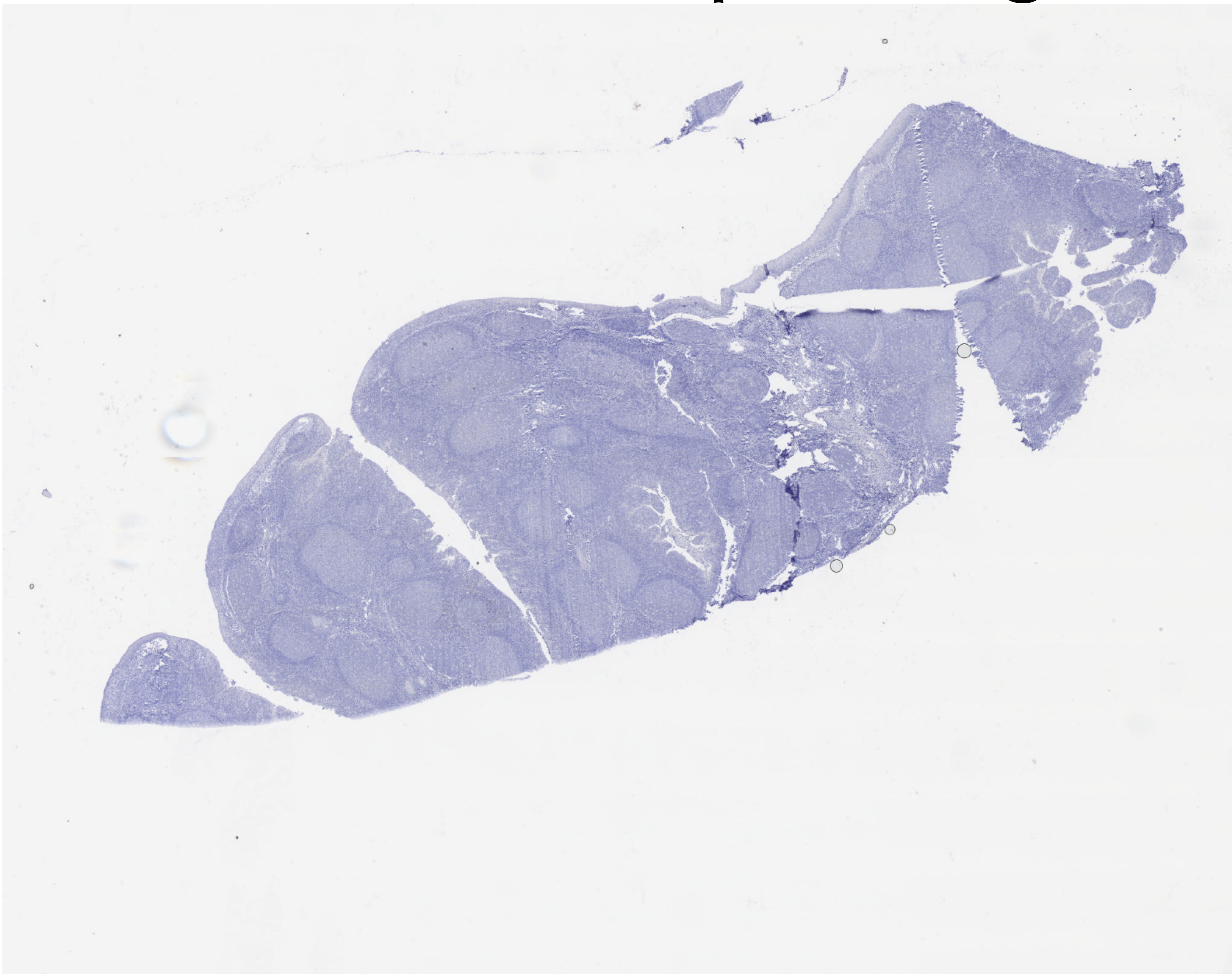


Image Stitching

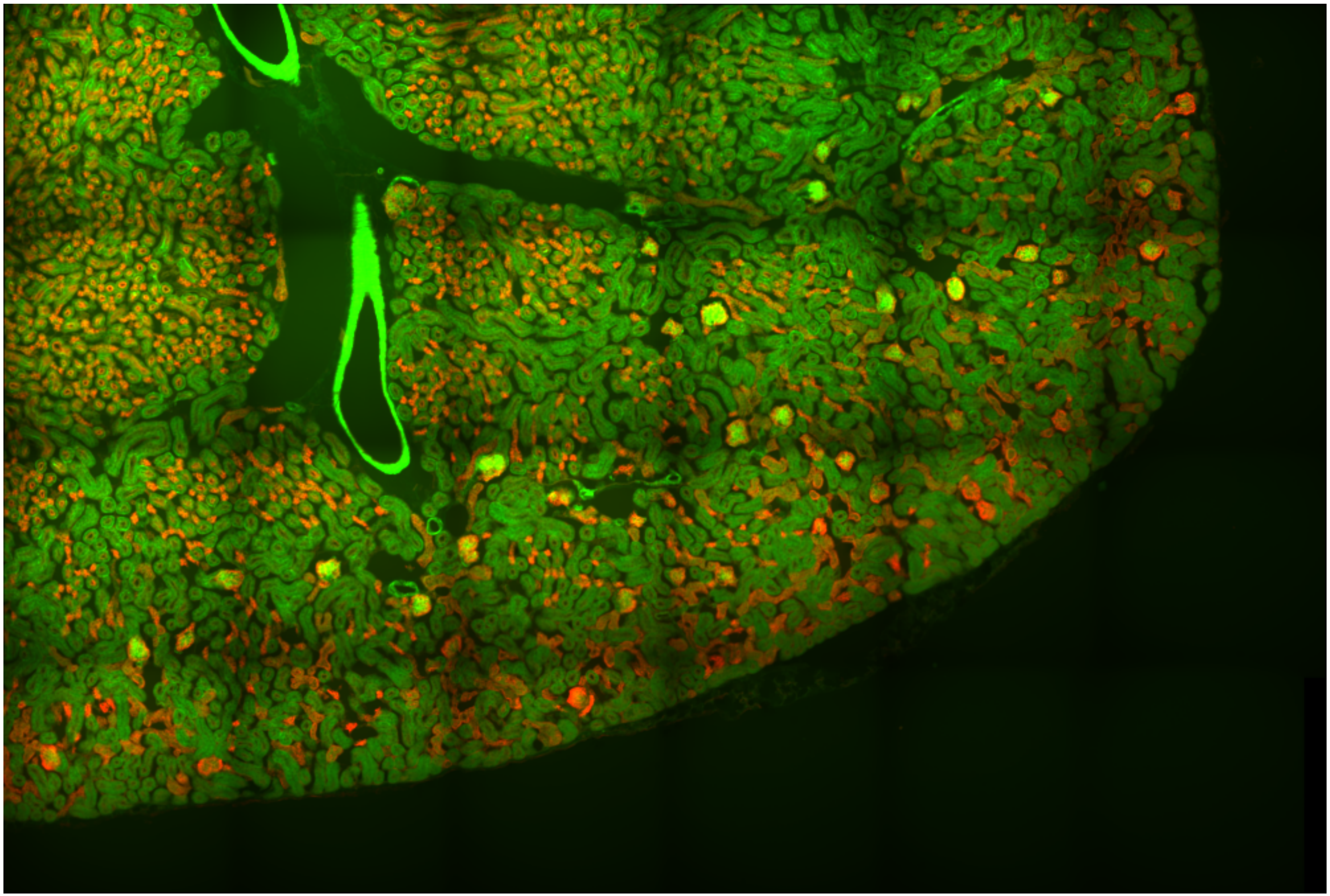
Tissue Sample Alignment



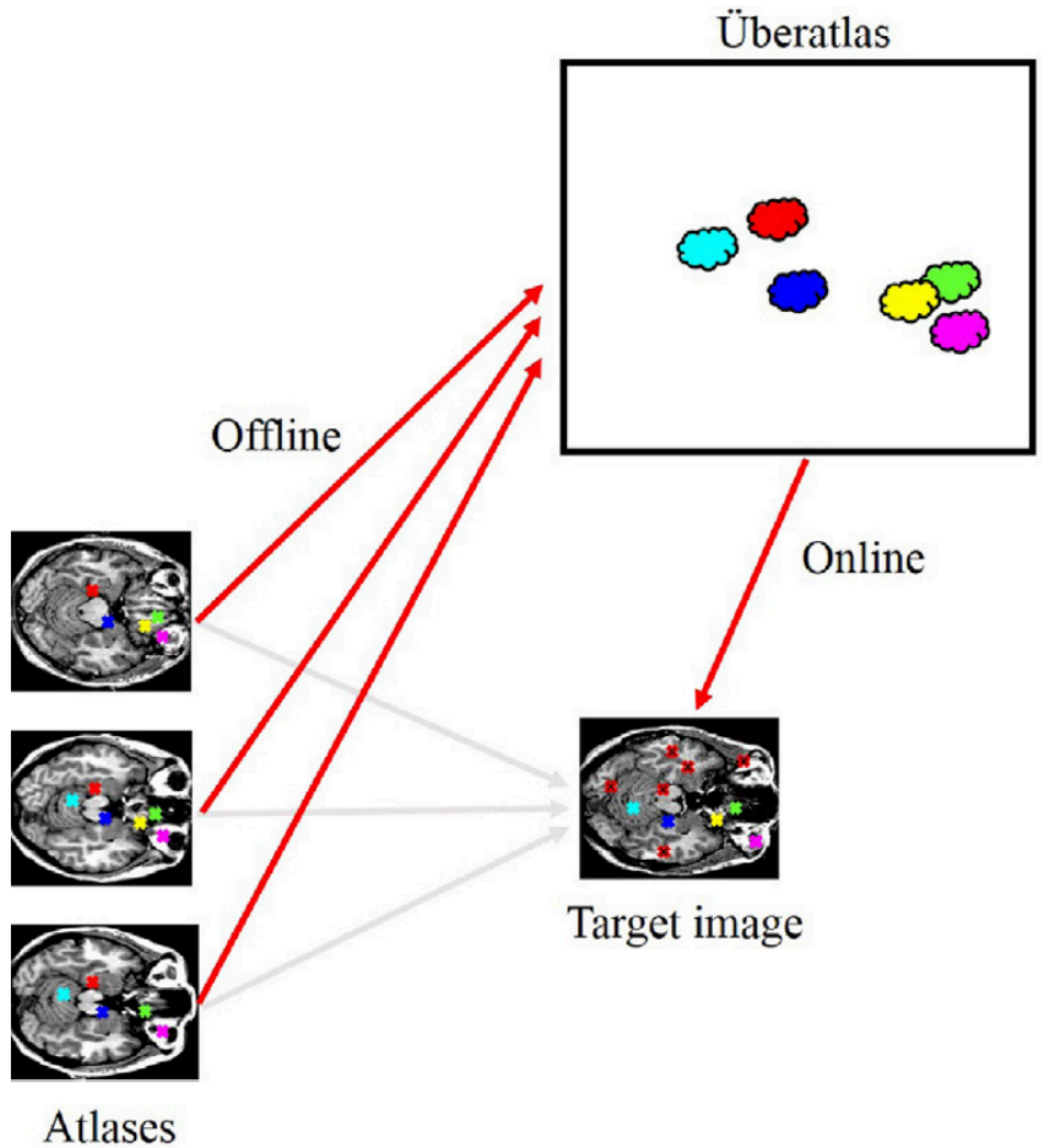
Tissue Sample Alignment



Stitching Microscopy Images



Atlas-Based Segmentation

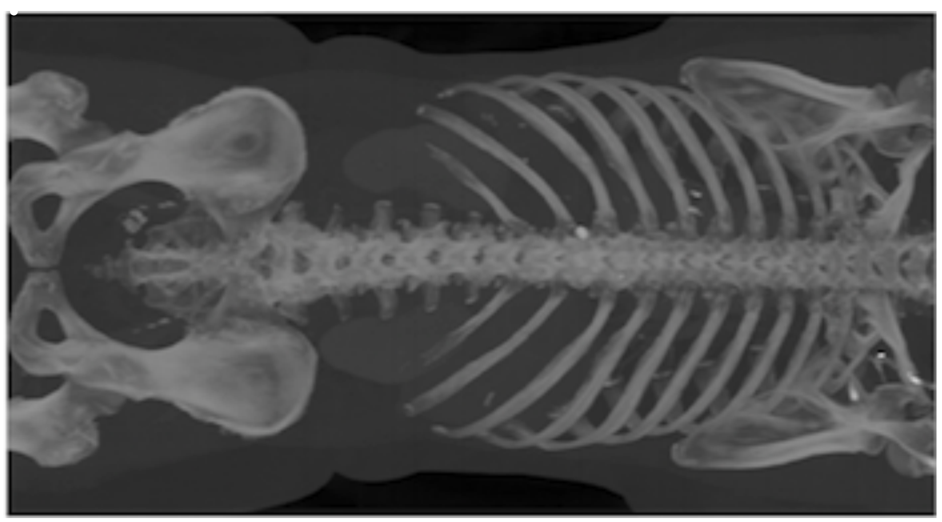


[Jennifer Alvén, Alexander Norlén, Olof Enqvist, Fredrik Kahl, Überatlas: Fast and robust registration for multi-atlas segmentation, Pattern Recognition Letters 2016]

Image Registration

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

Image Registration



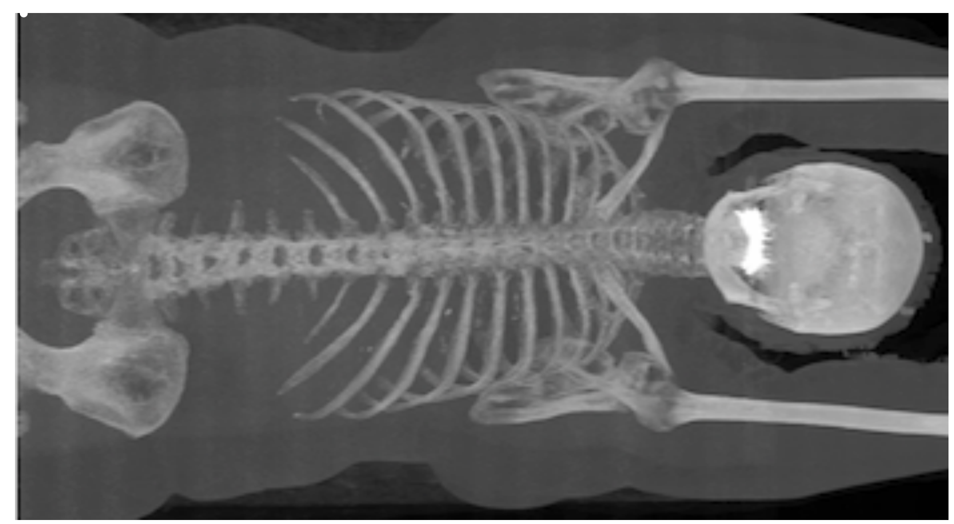
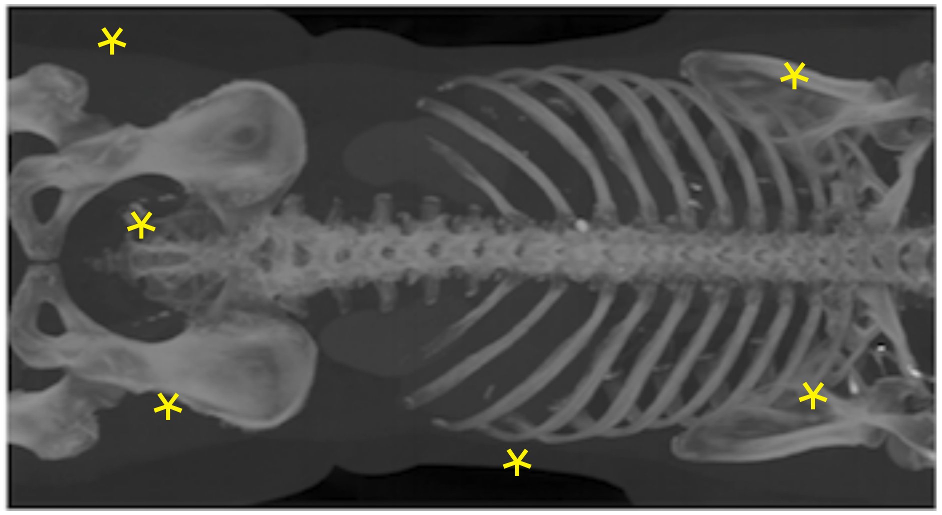
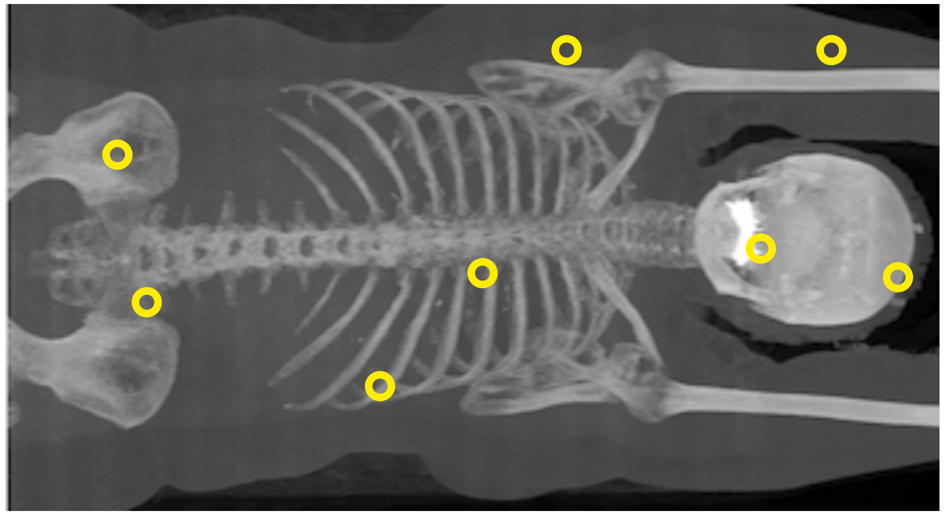


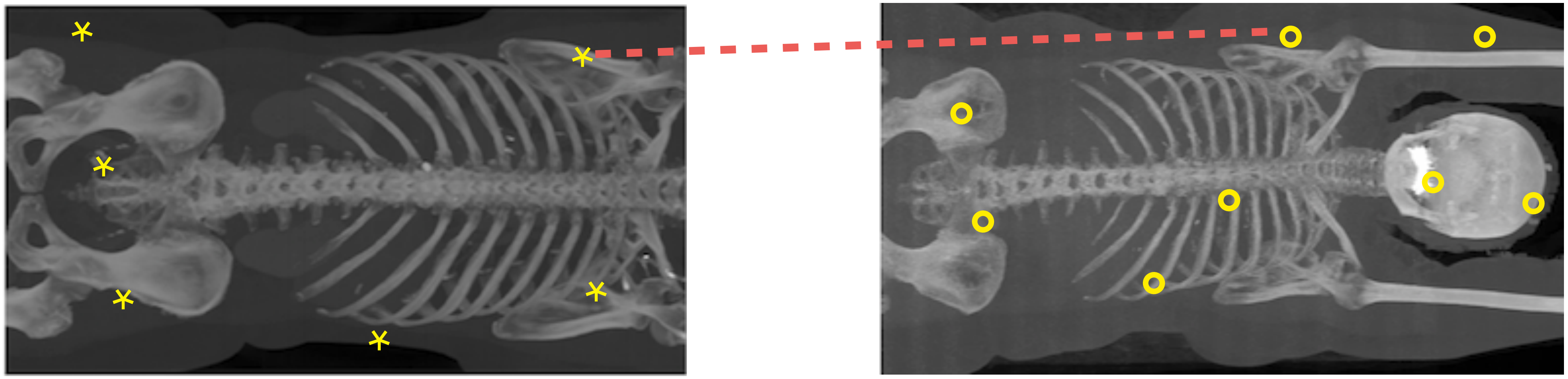
Image Registration





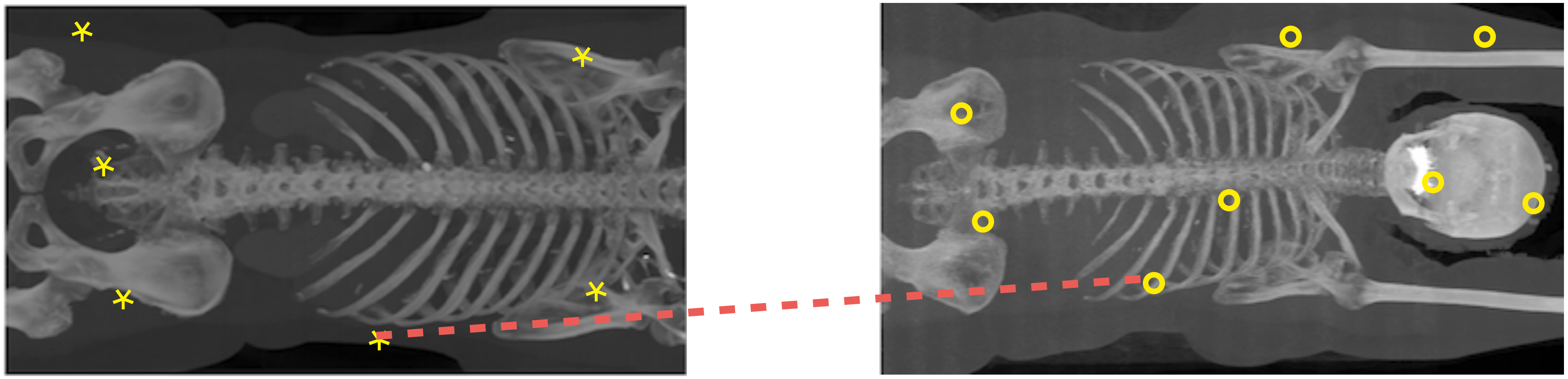
Extract features

Image Registration



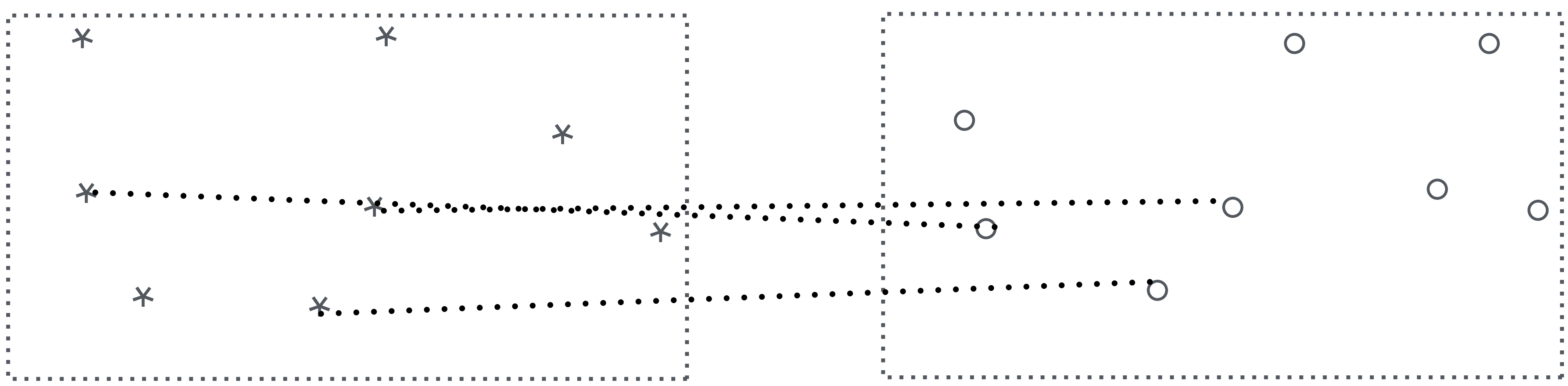
Find correspondences

Image Registration



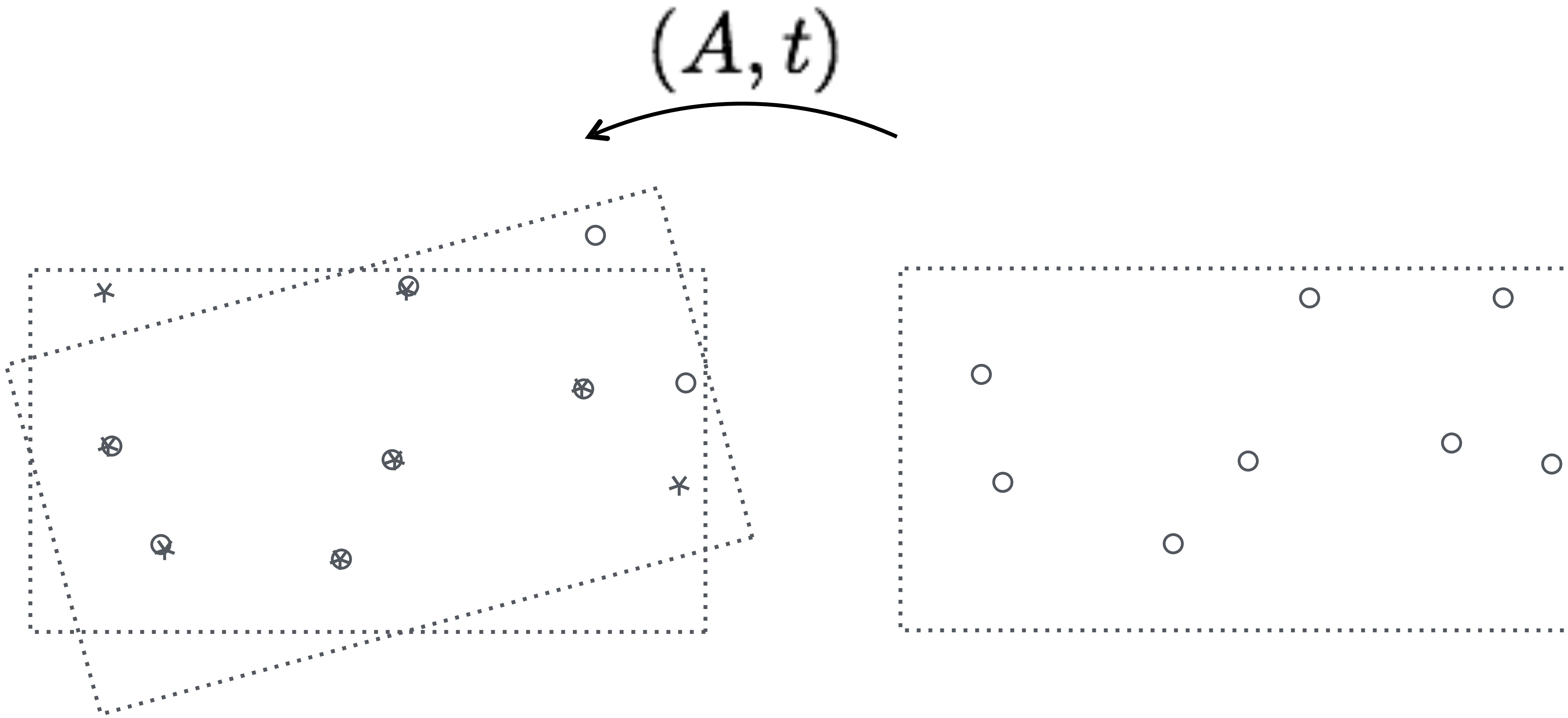
Find correspondences

Point Set Registration



Point Set Registration

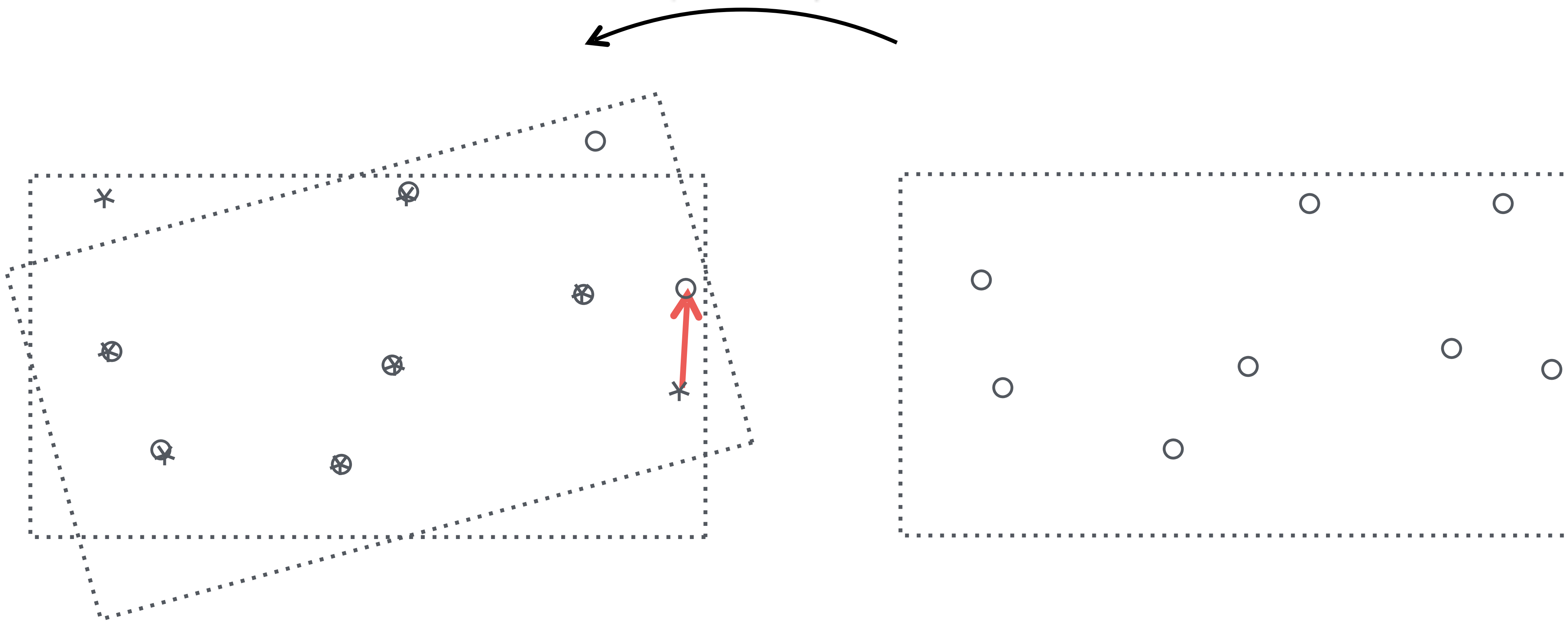
Find a consistent transformation.



Point Set Registration

Find a consistent transformation.

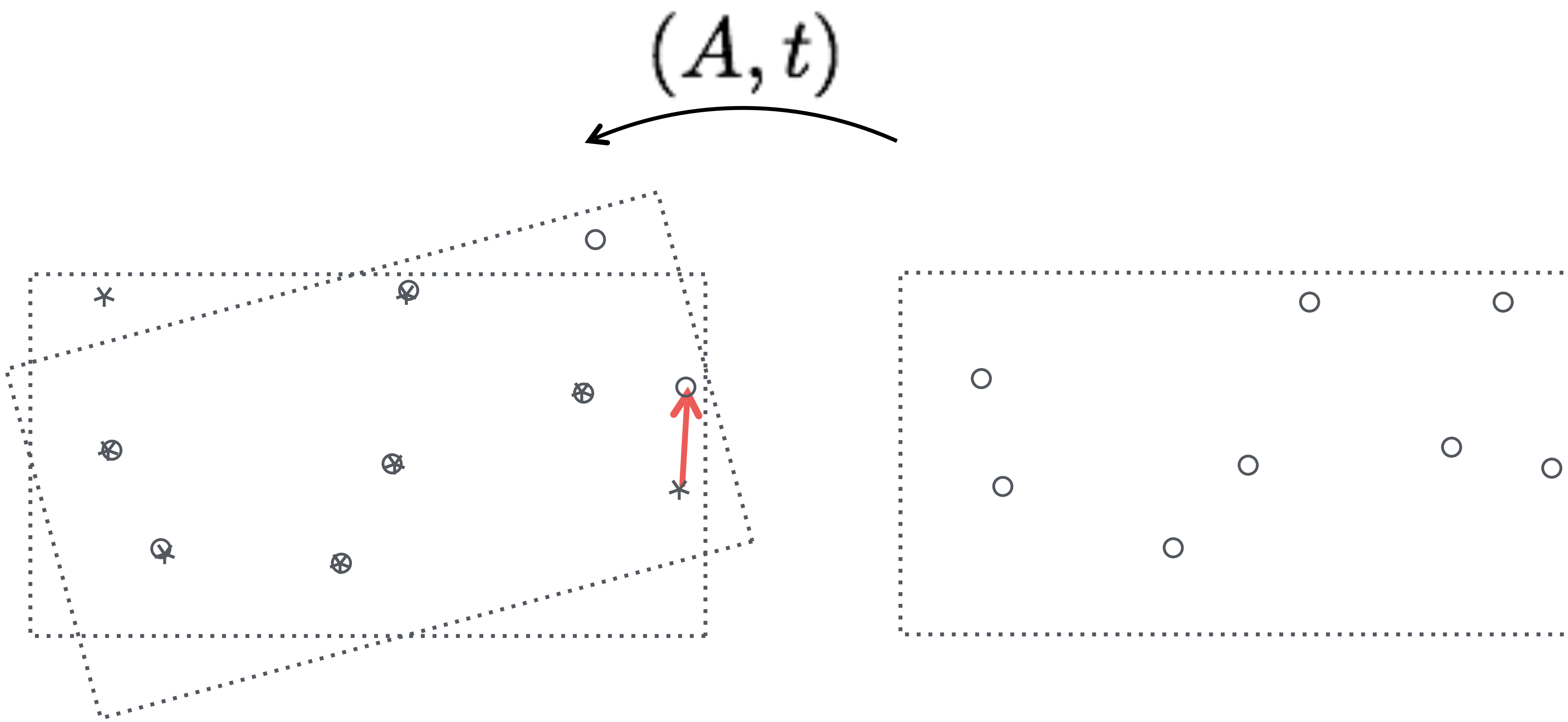
$$(A, t)$$



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

Point Set Registration

Find a consistent transformation.



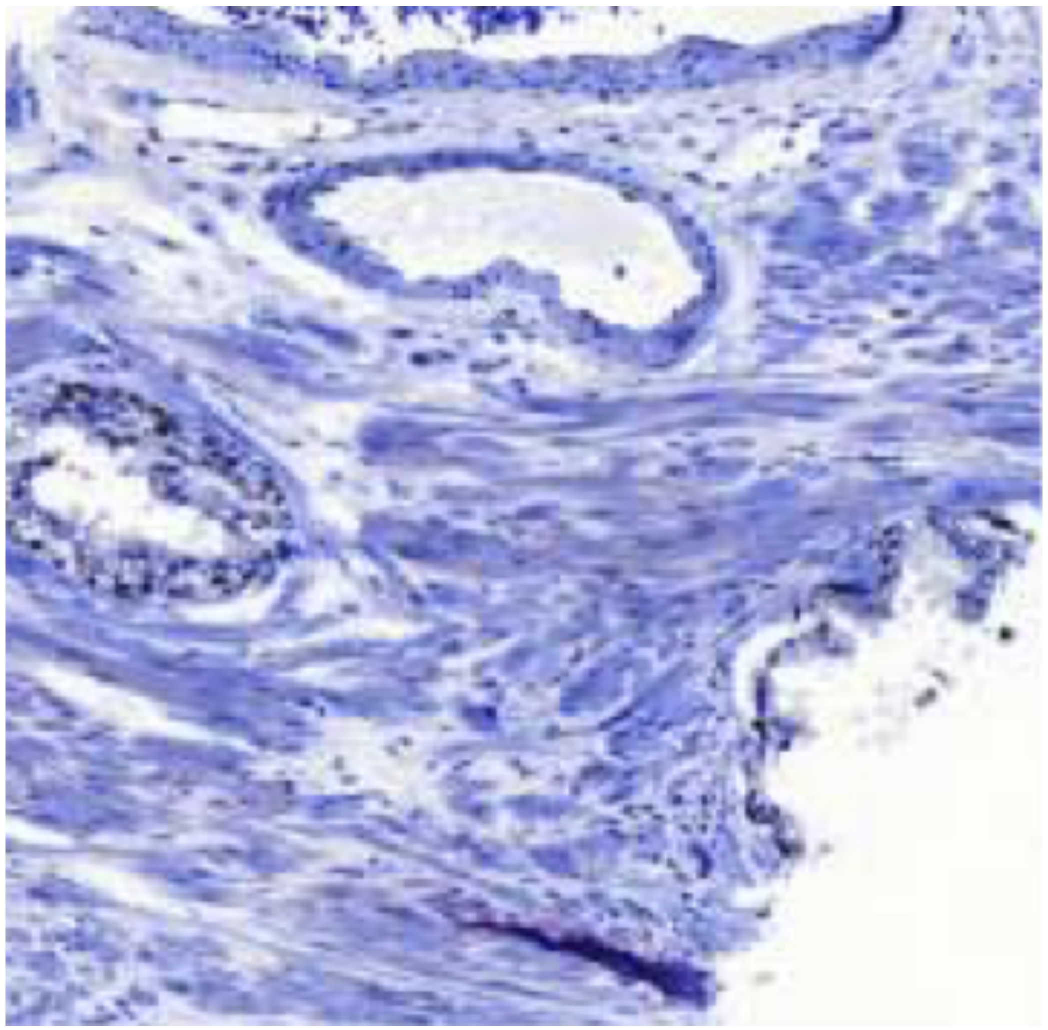
$$r_i(\theta) = A \begin{pmatrix} x_j \\ y_j \end{pmatrix} + t - \begin{pmatrix} \hat{x}_j \\ \hat{y}_j \end{pmatrix}$$

inlier test in RANSAC:

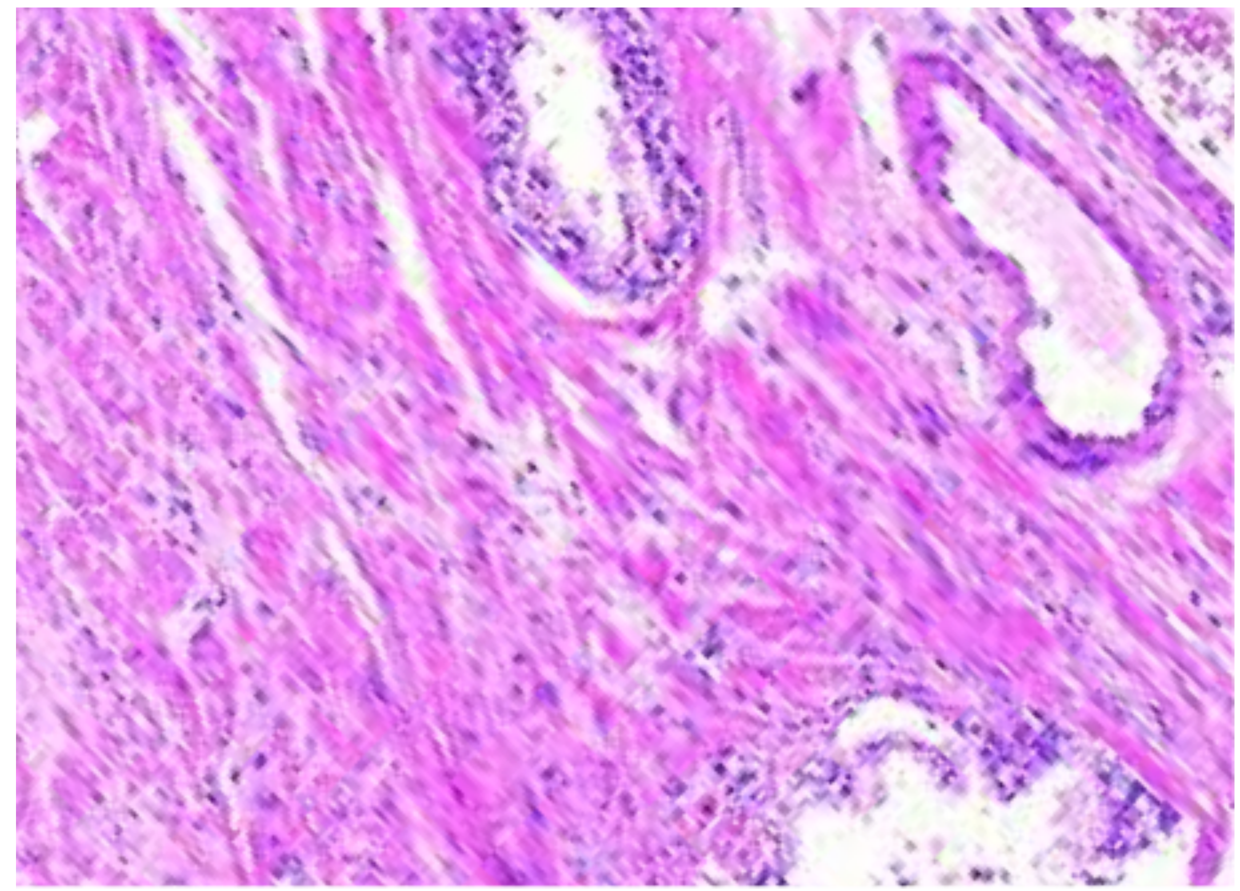
$$||r_i(\theta)|| < \tau$$

Image Warping

transformation



source image



target image

Image Warping

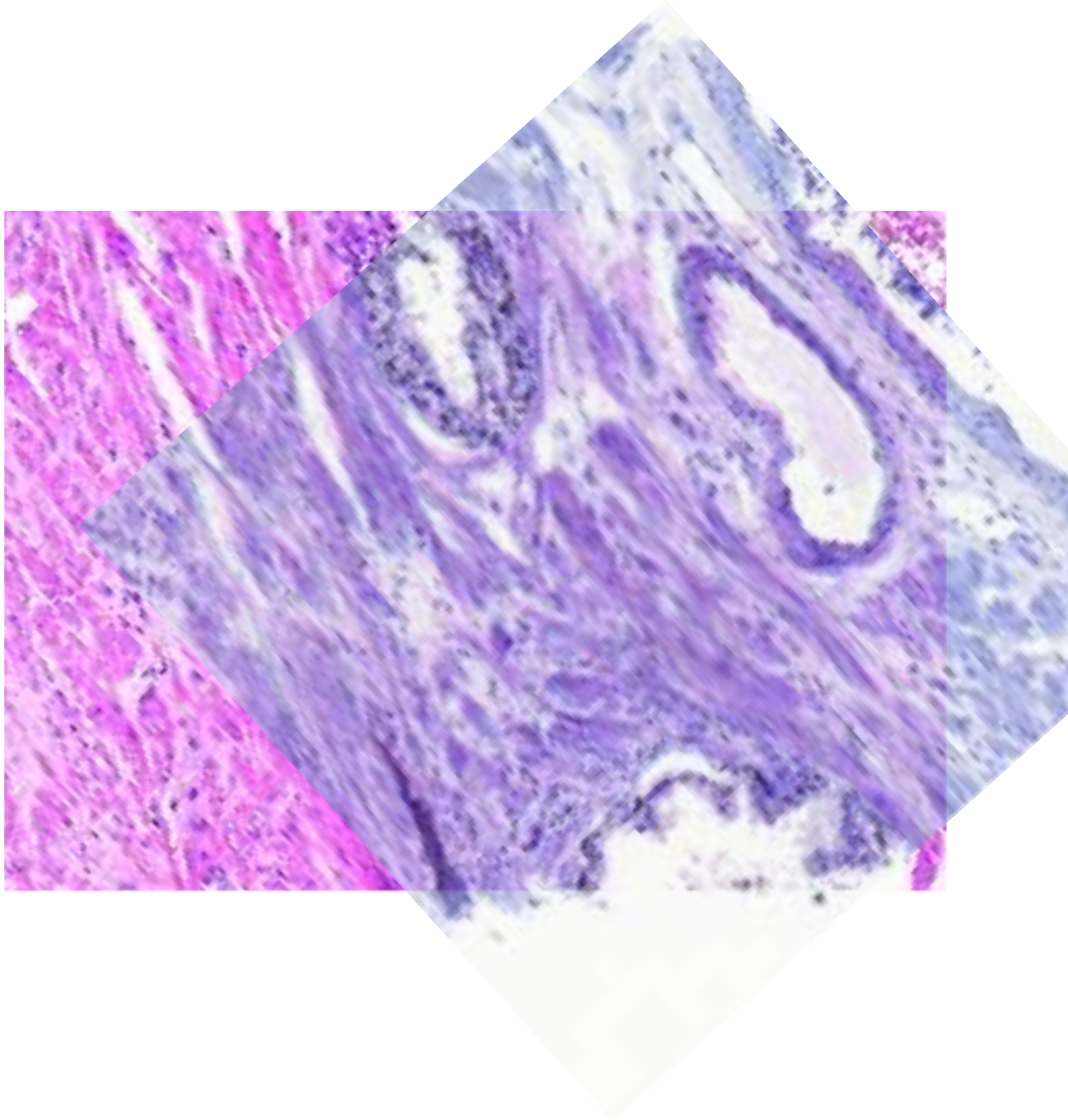


Image Warping

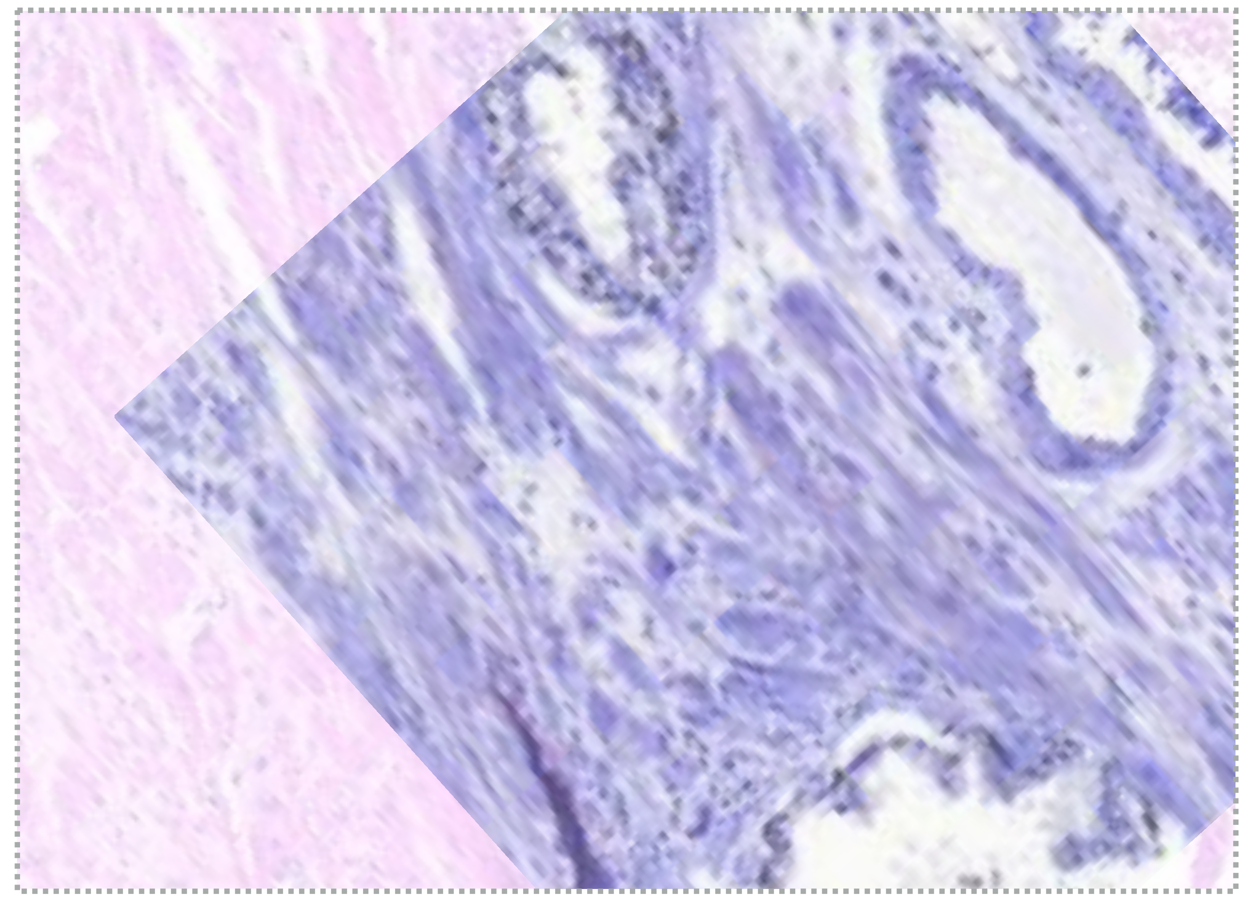
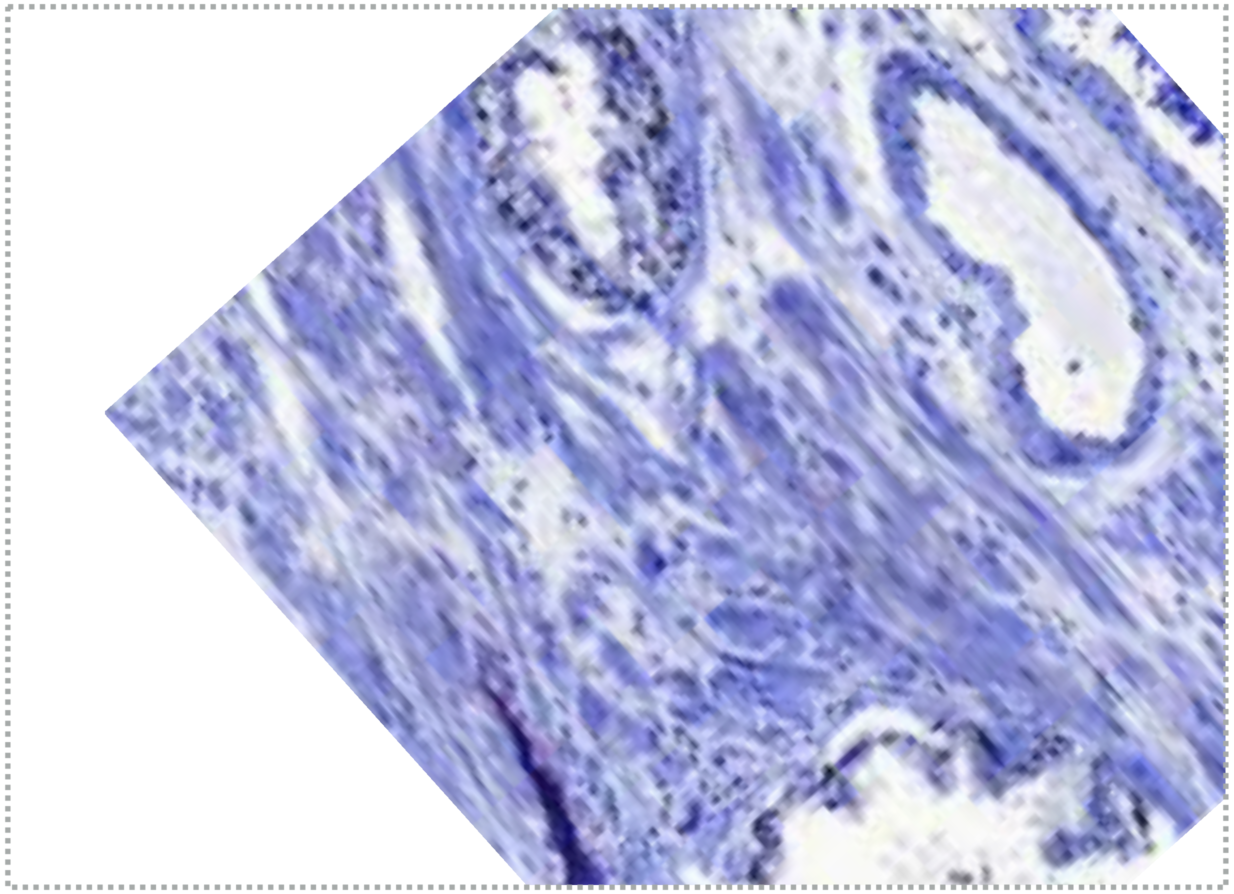
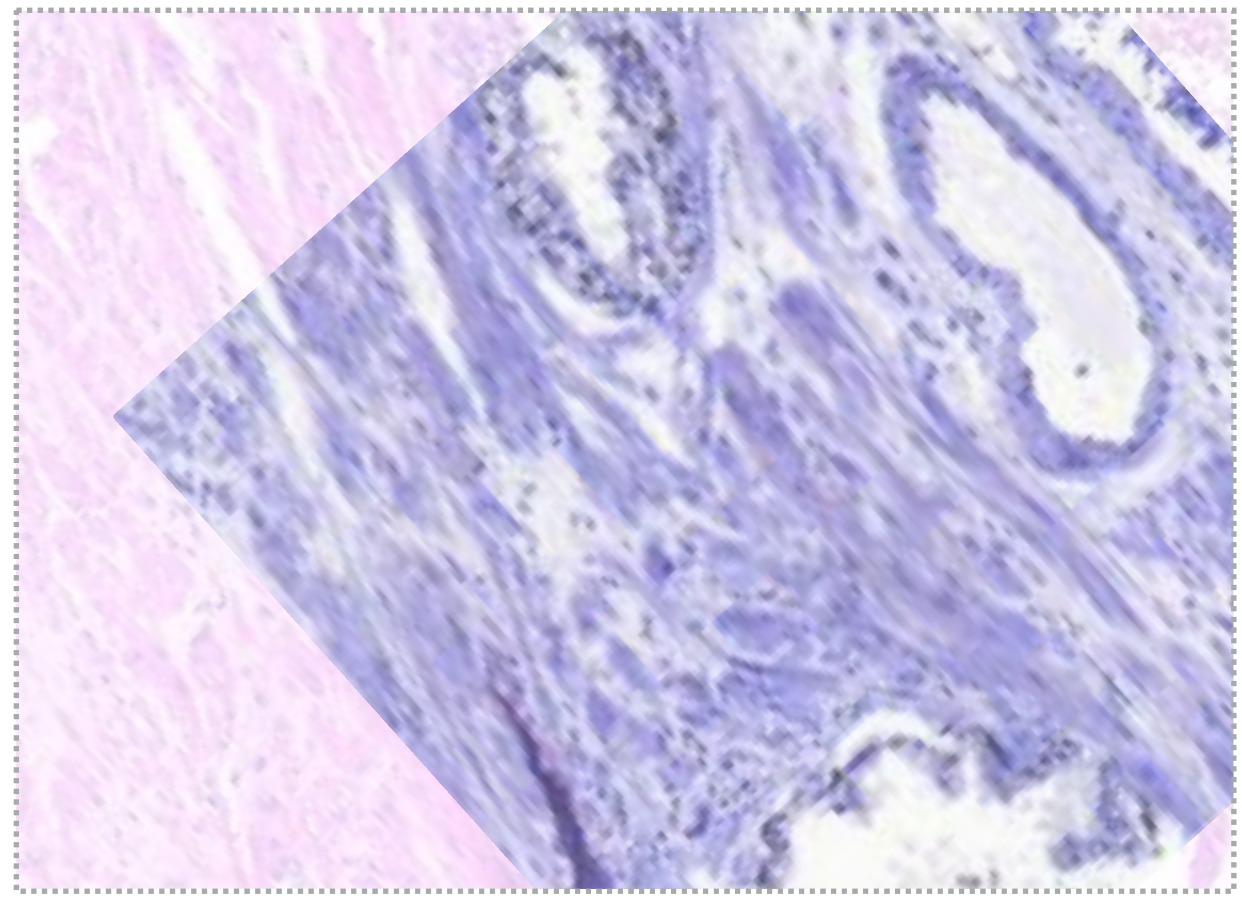


Image Warping



warped image

Image Analysis

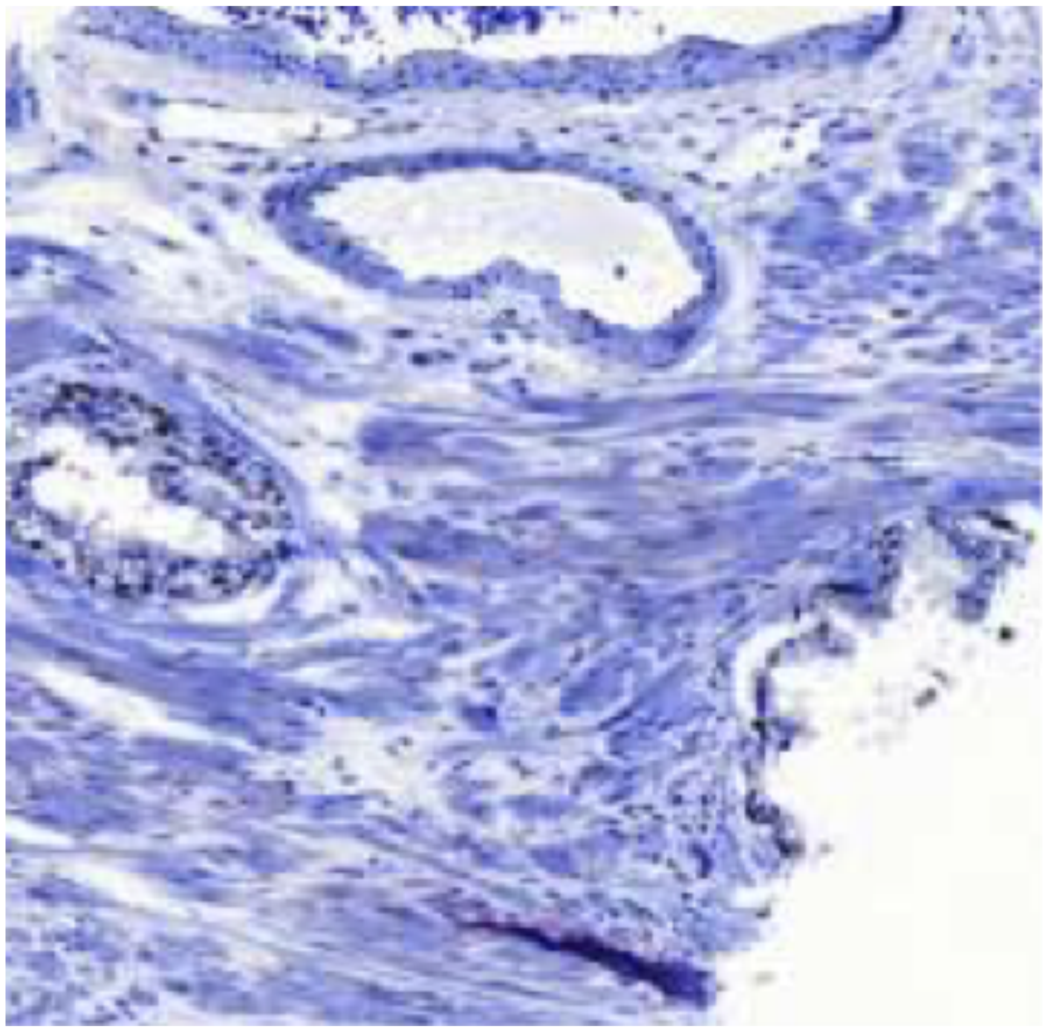


Today

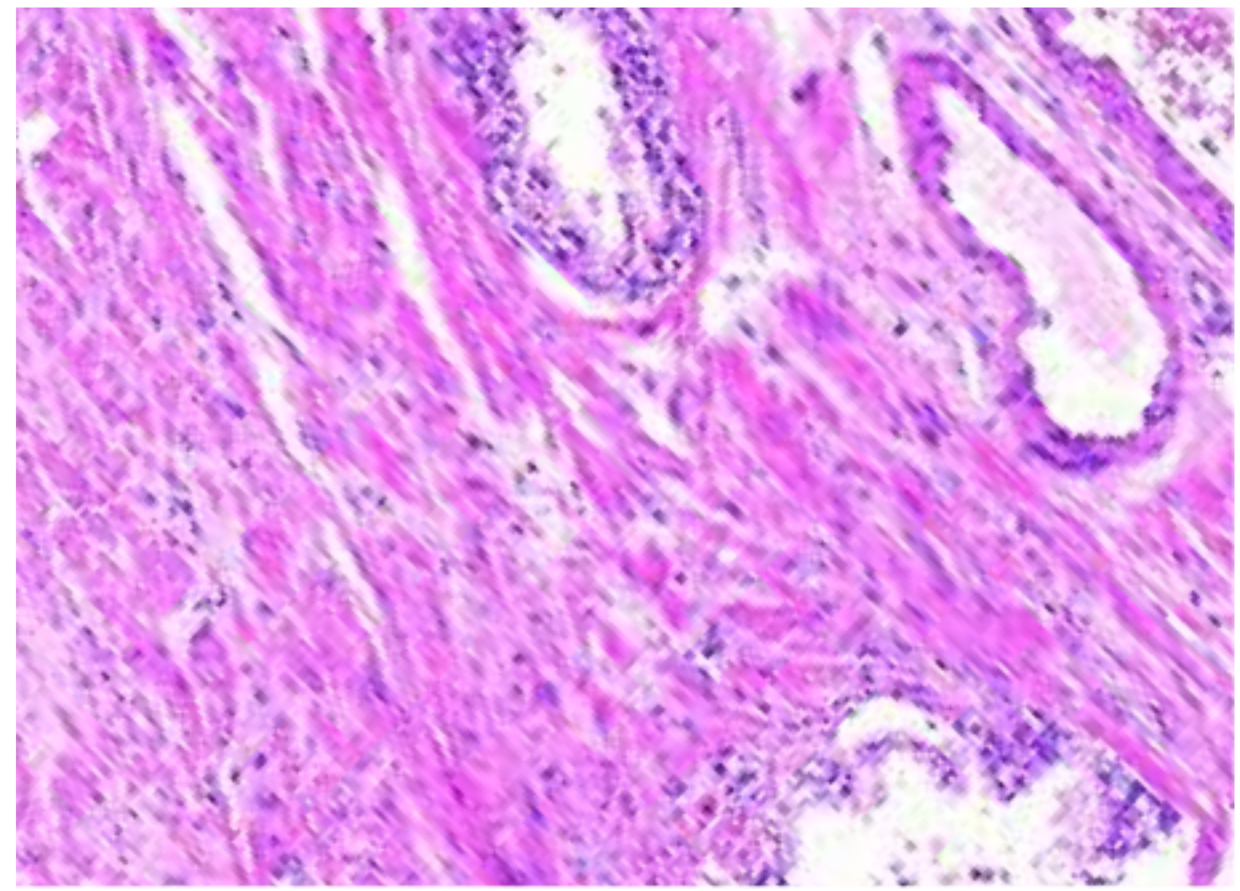
- Image Warping
- Transformation Estimation

Image Warping

transformation



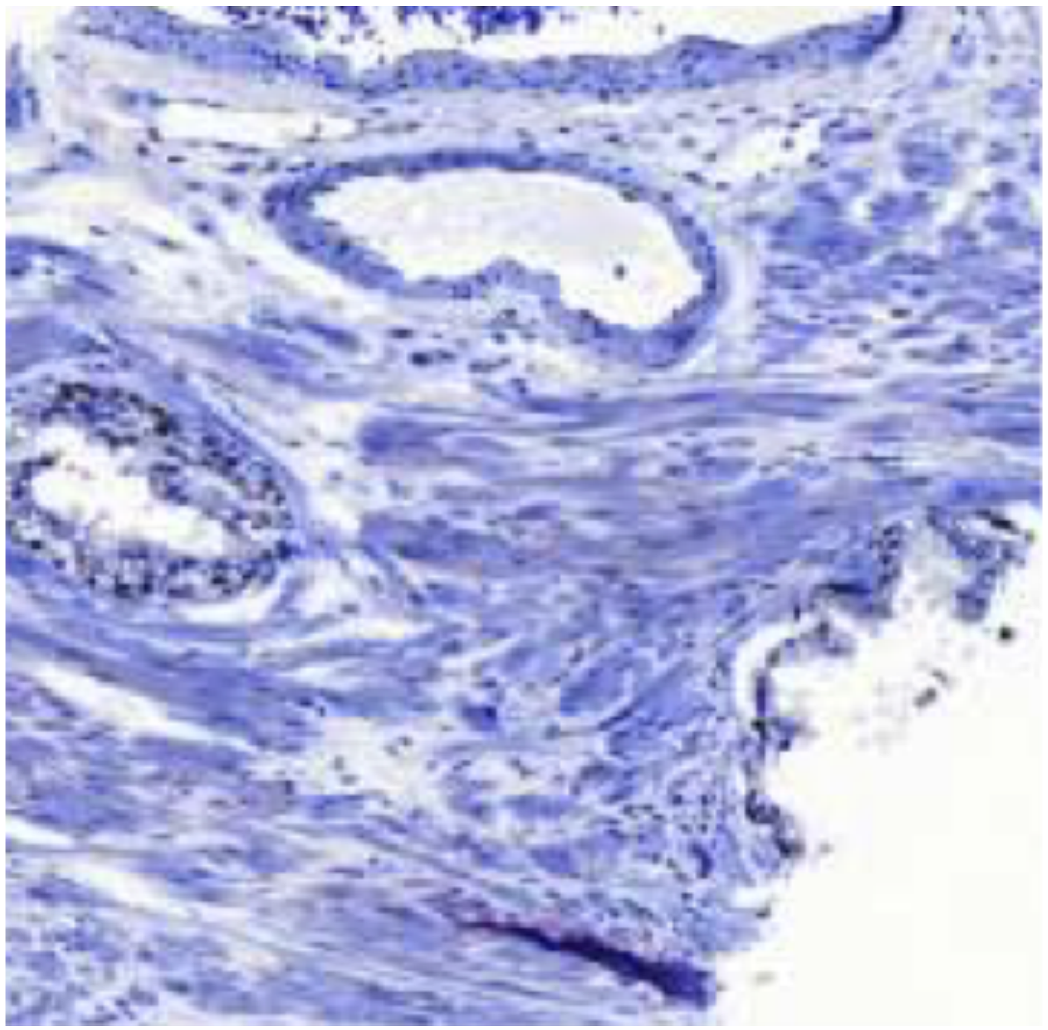
source image



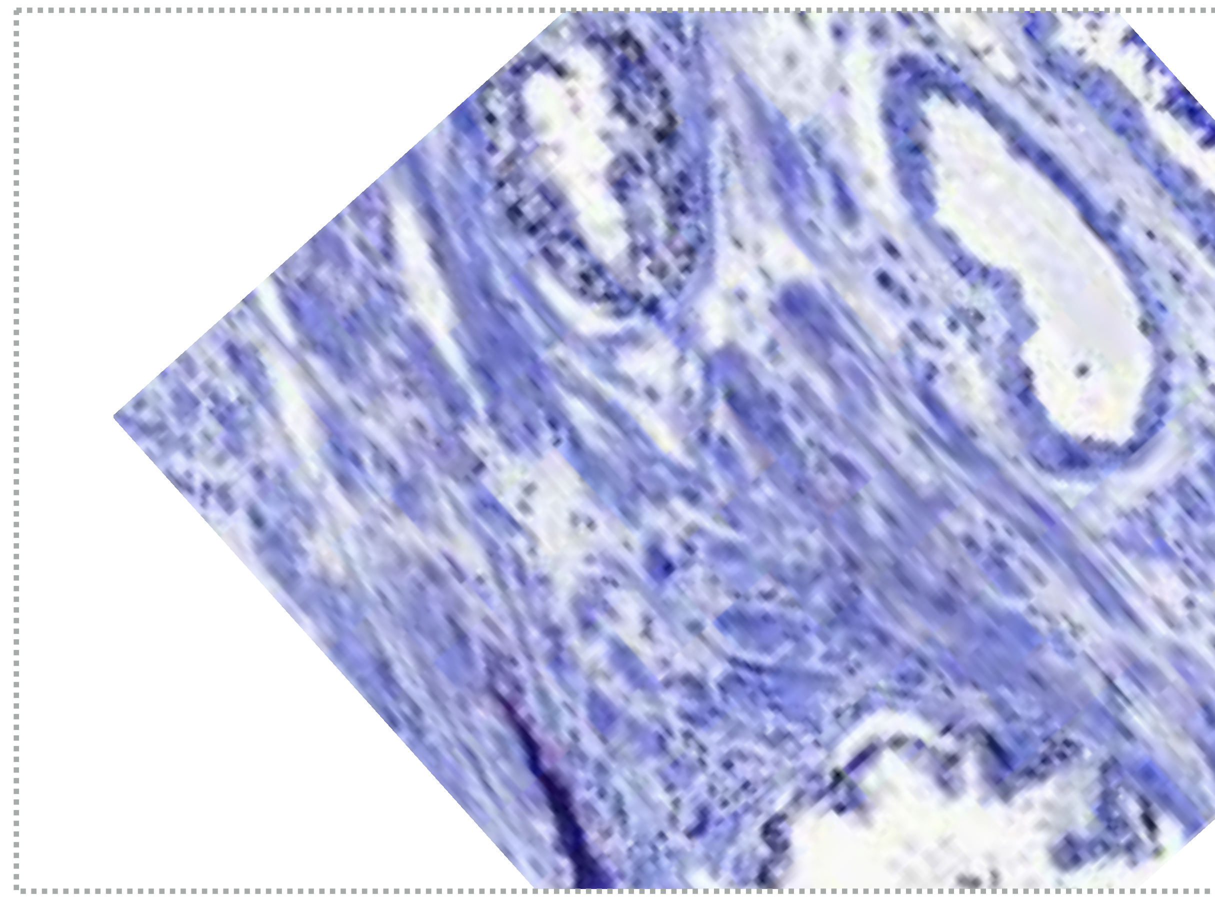
target image

Image Warping

transformation

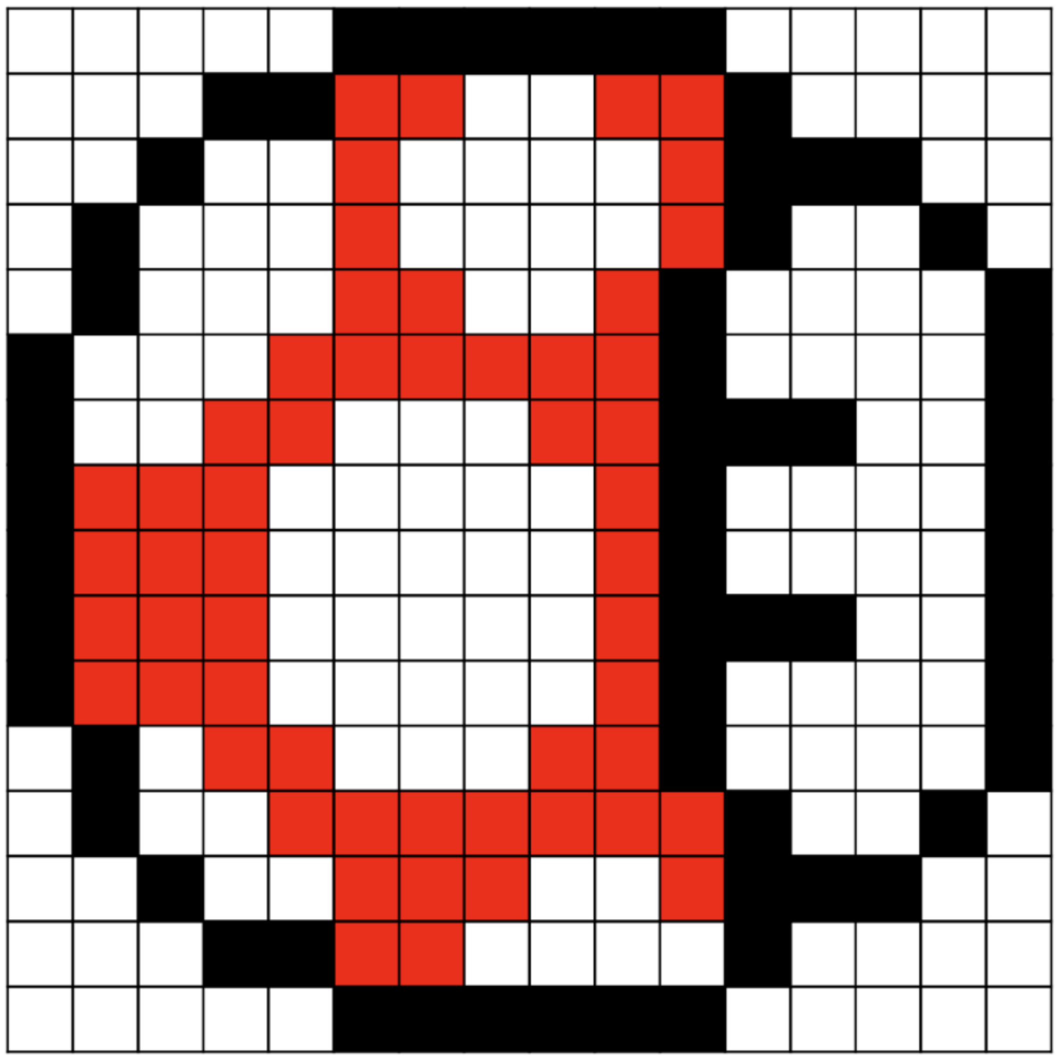


source image

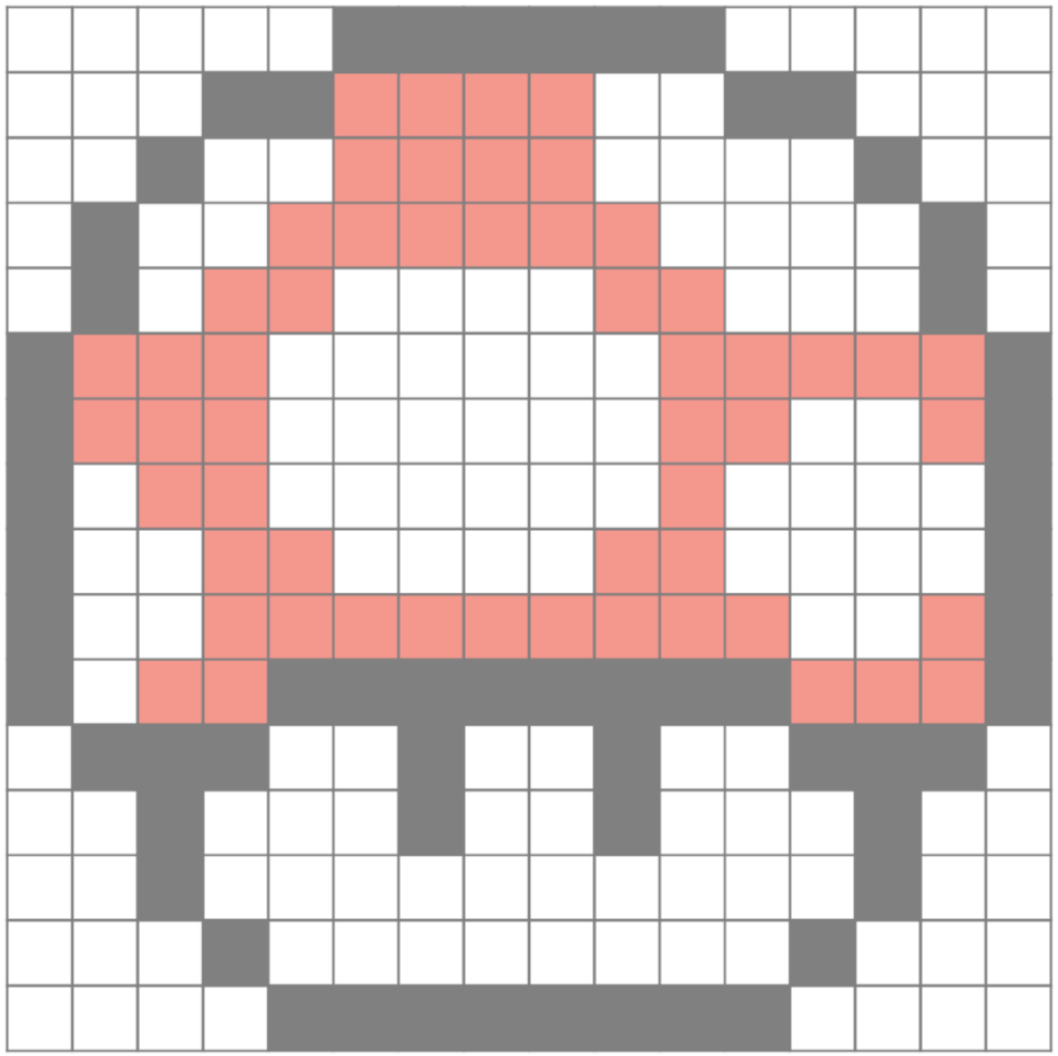


warped image

A Simple Example

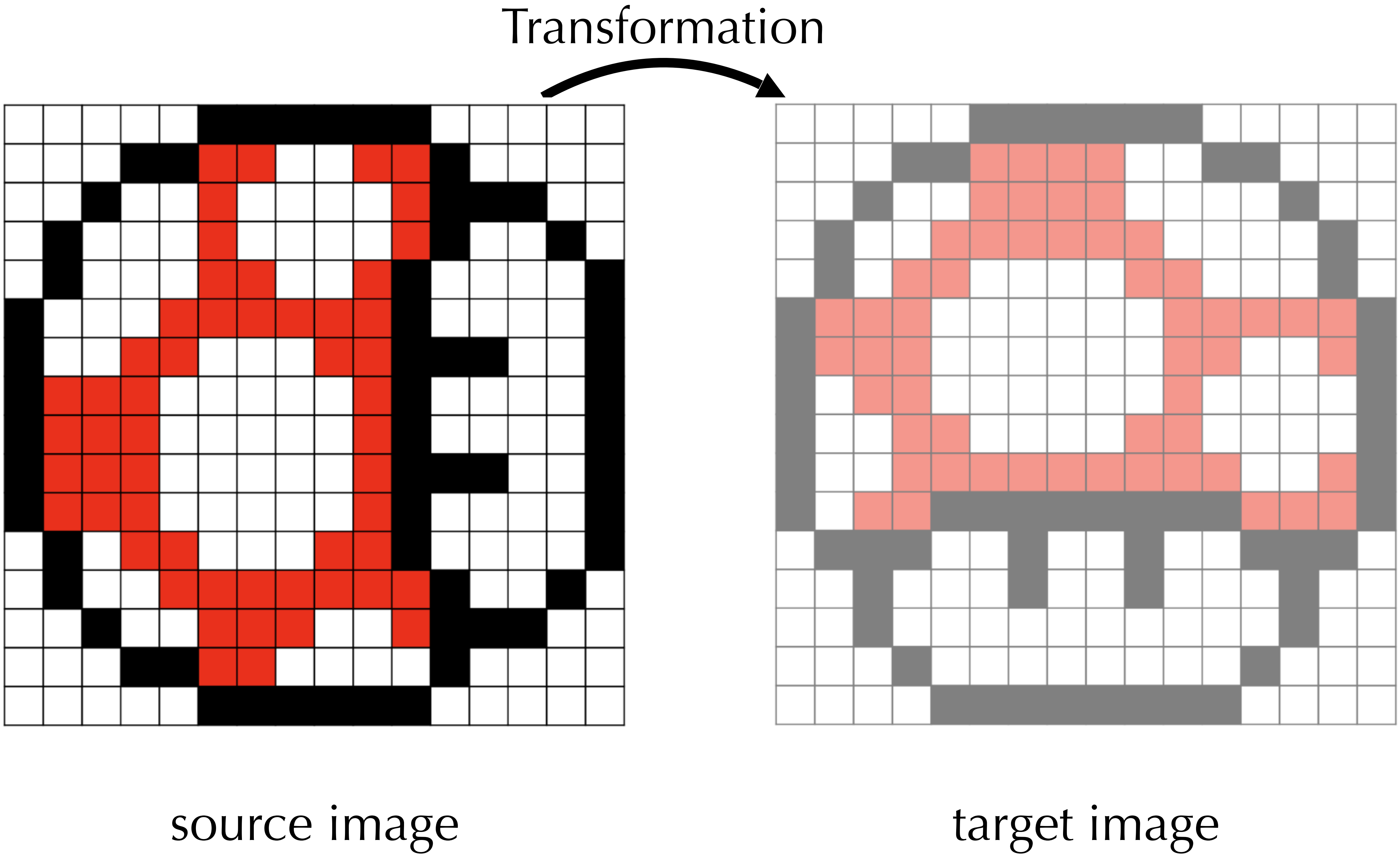


source image

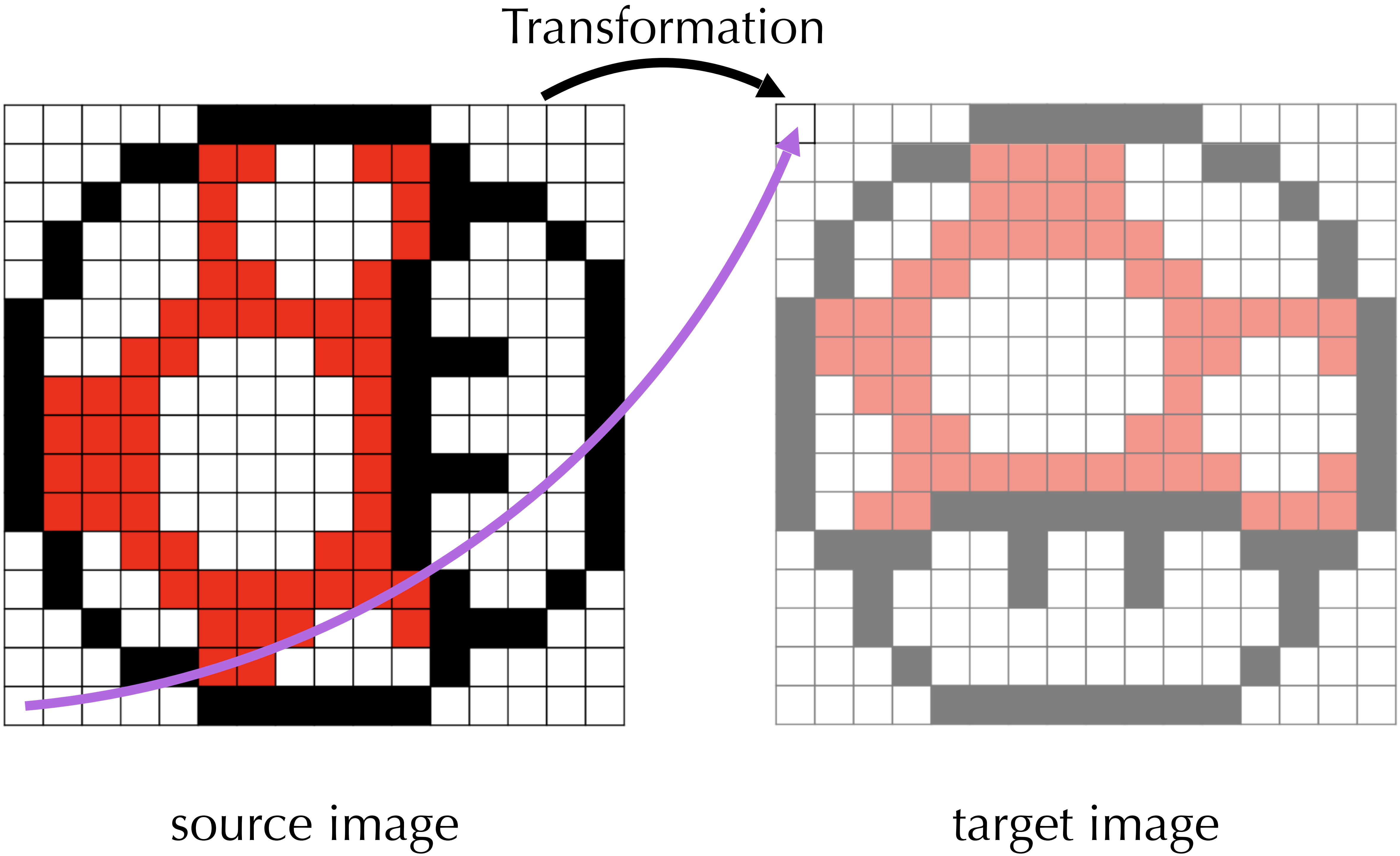


target image

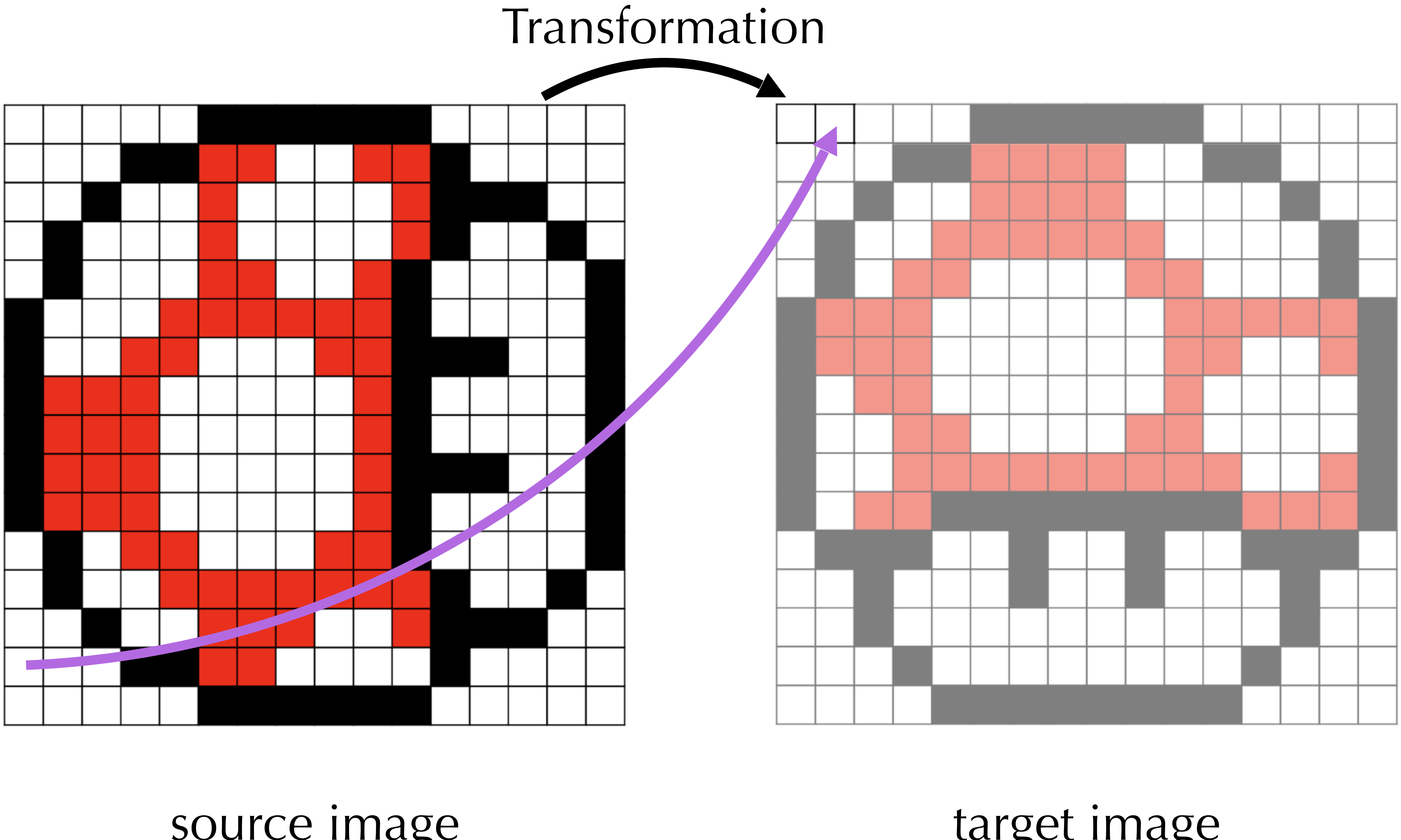
A Simple Example



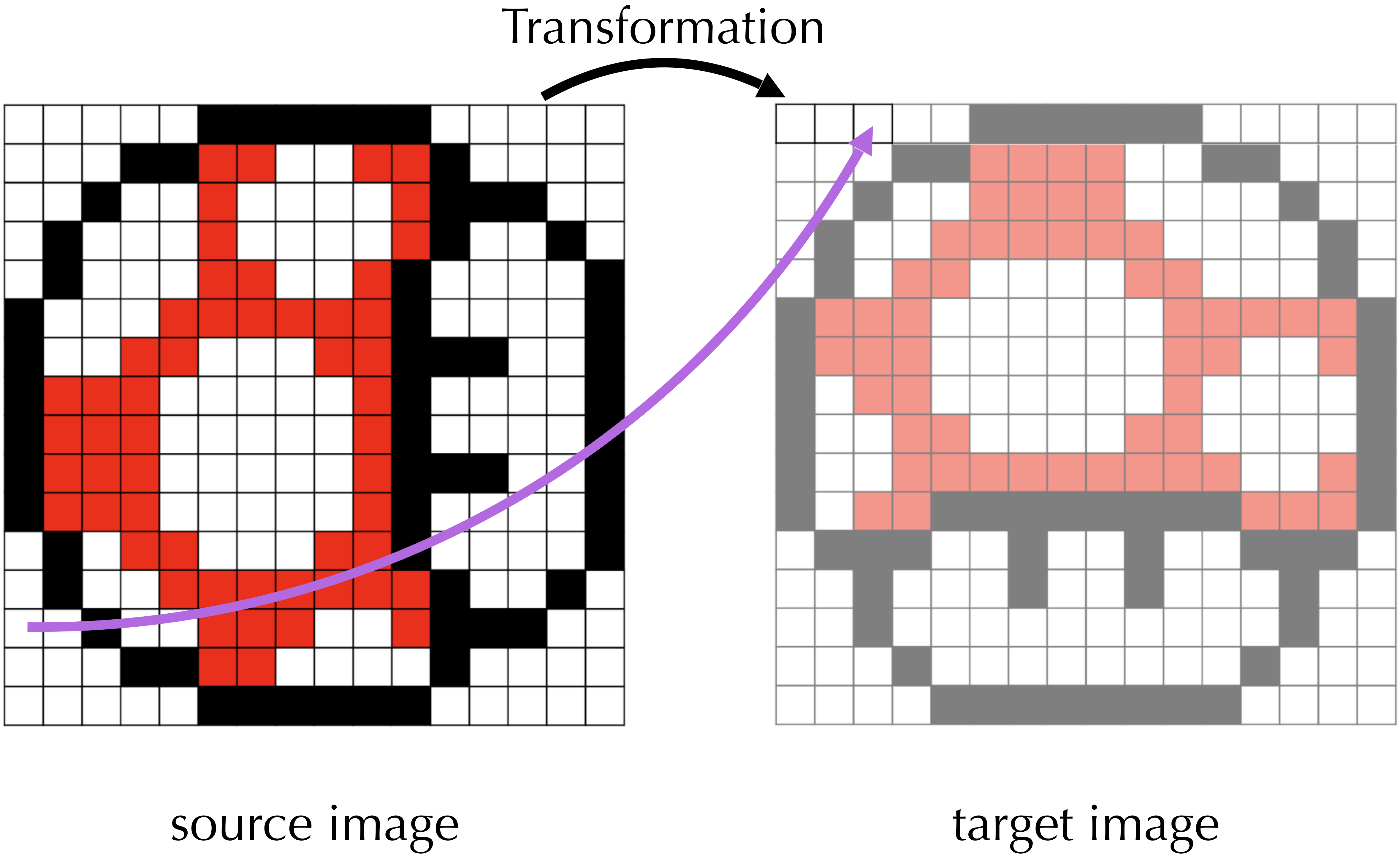
A Simple Example



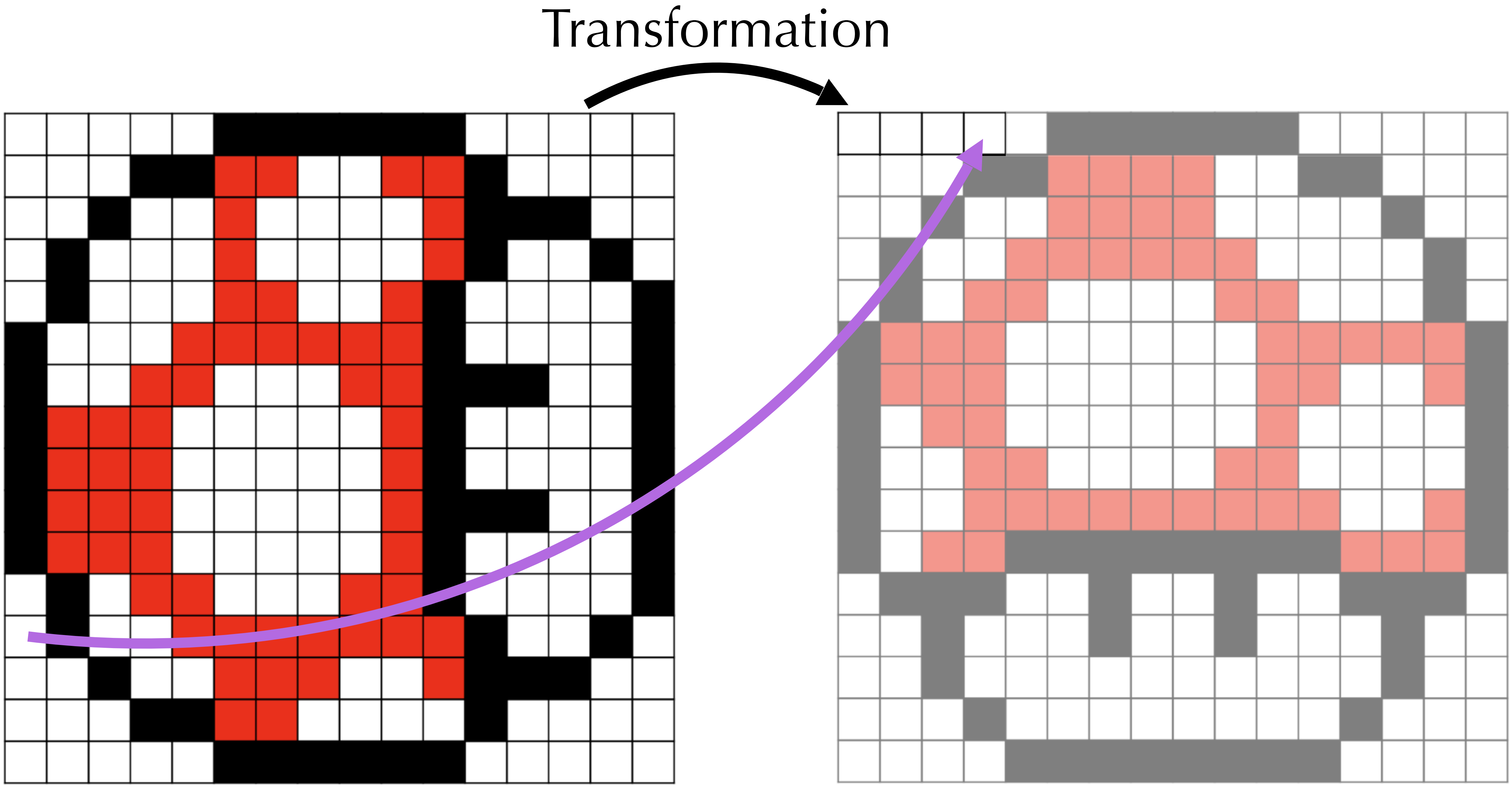
A Simple Example



A Simple Example

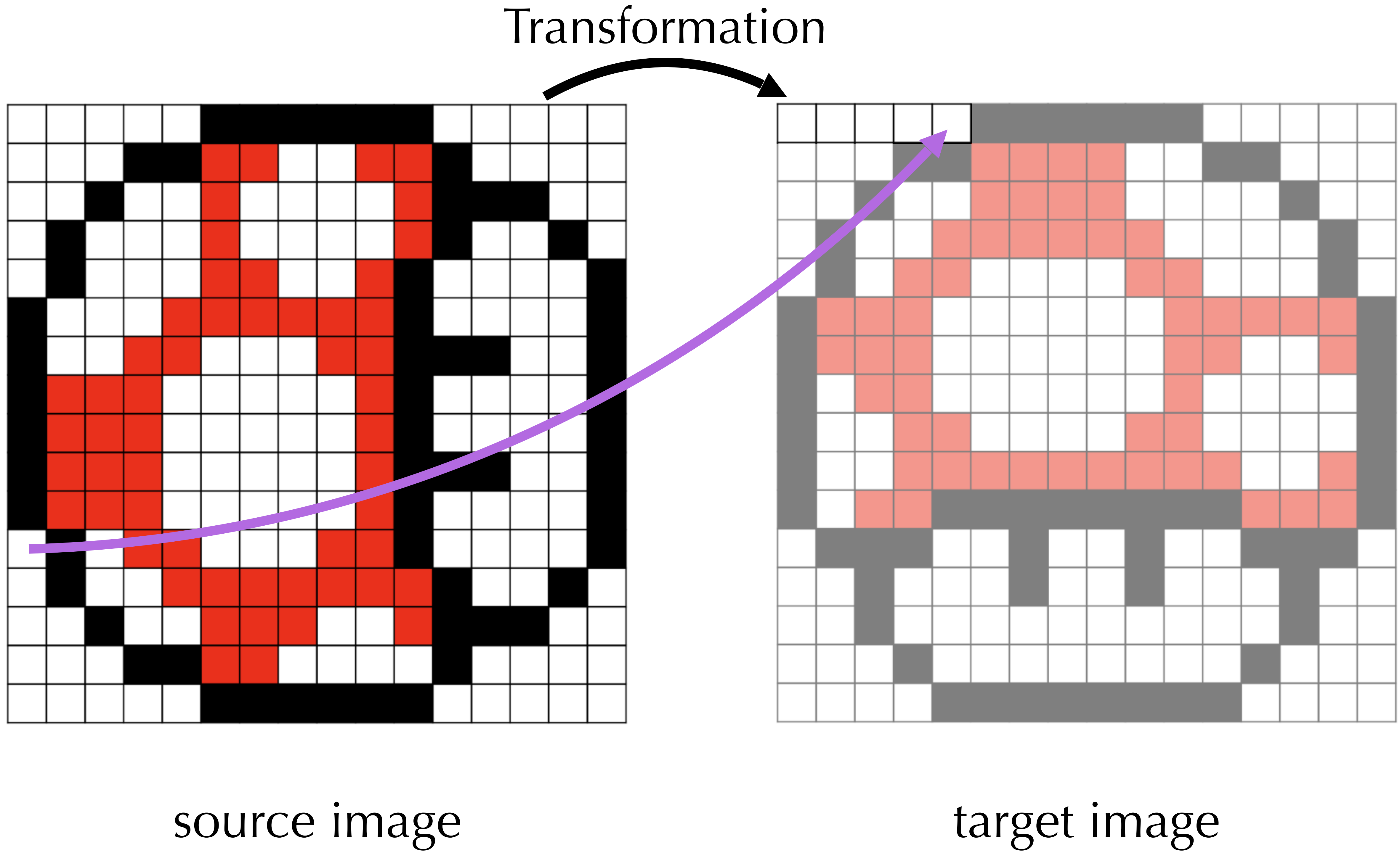


A Simple Example

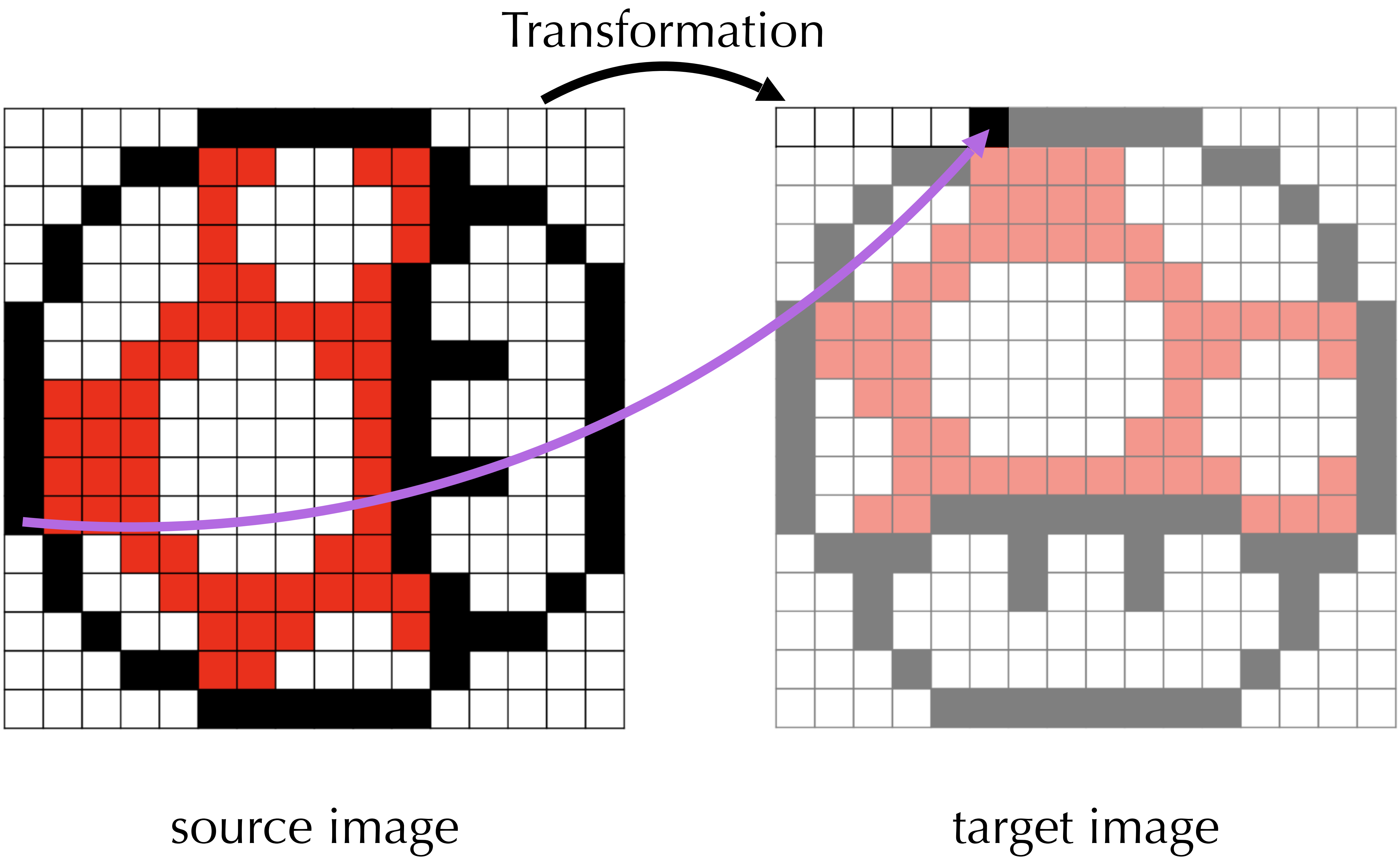


source image   target image

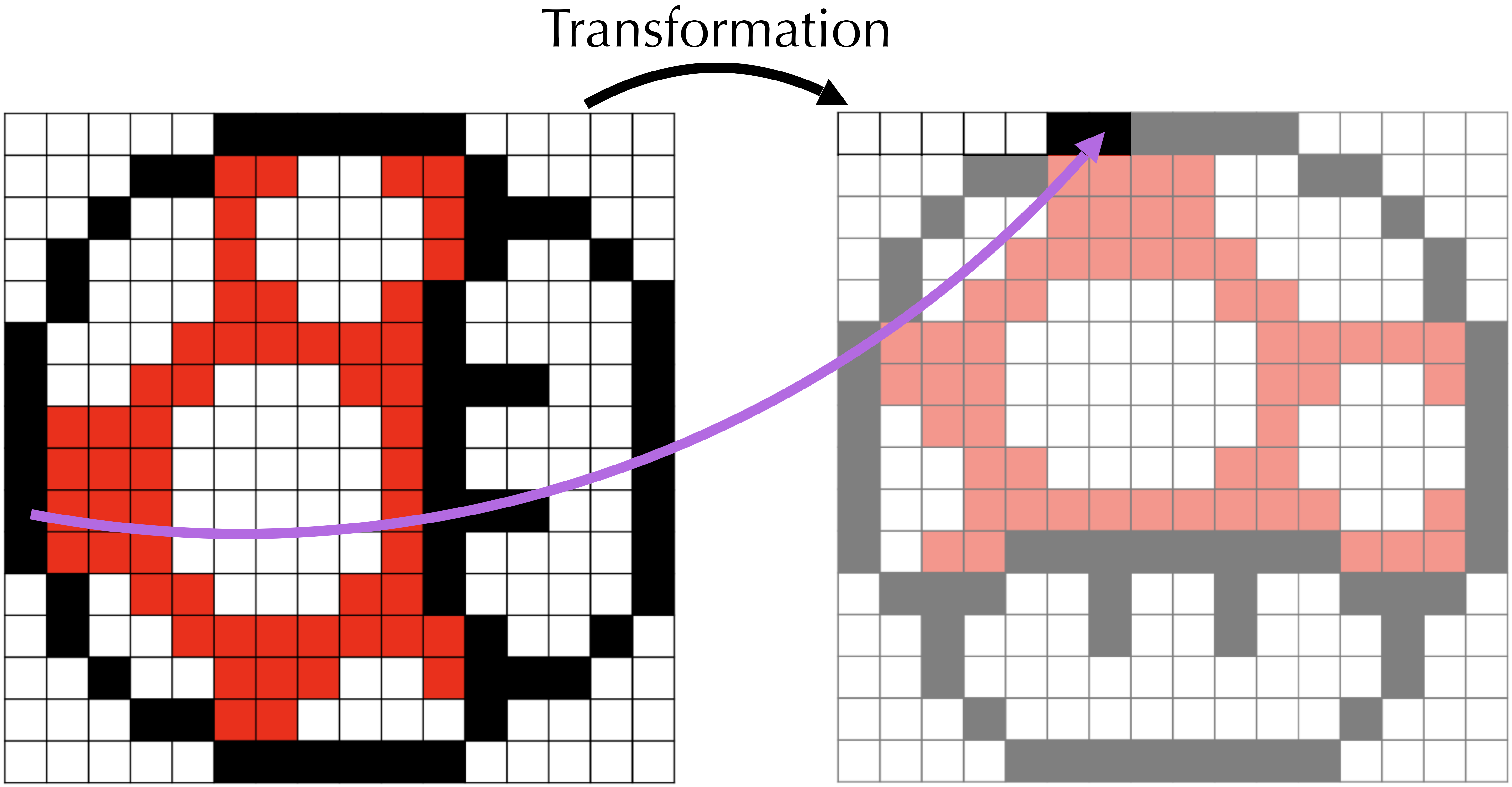
A Simple Example



A Simple Example



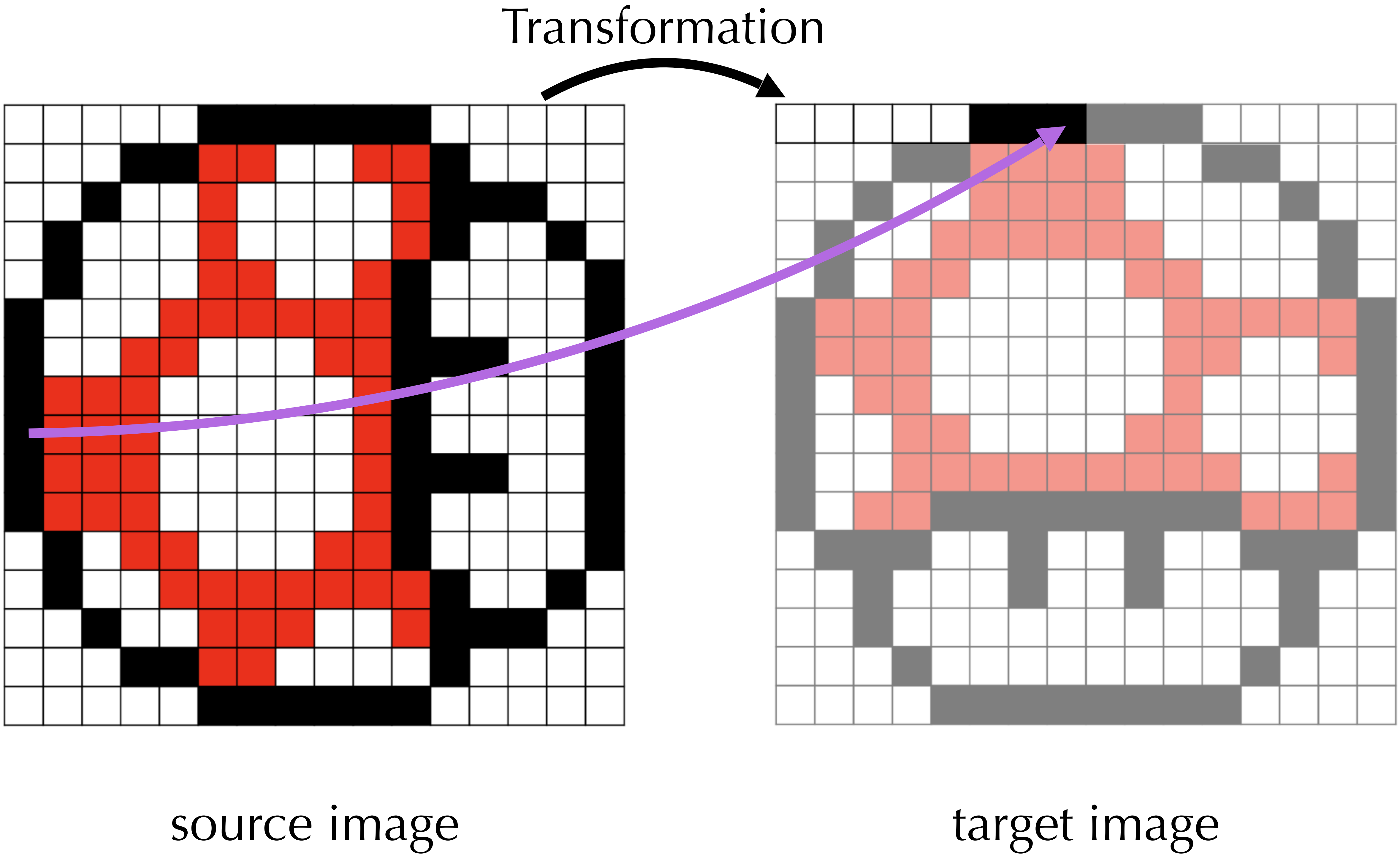
A Simple Example



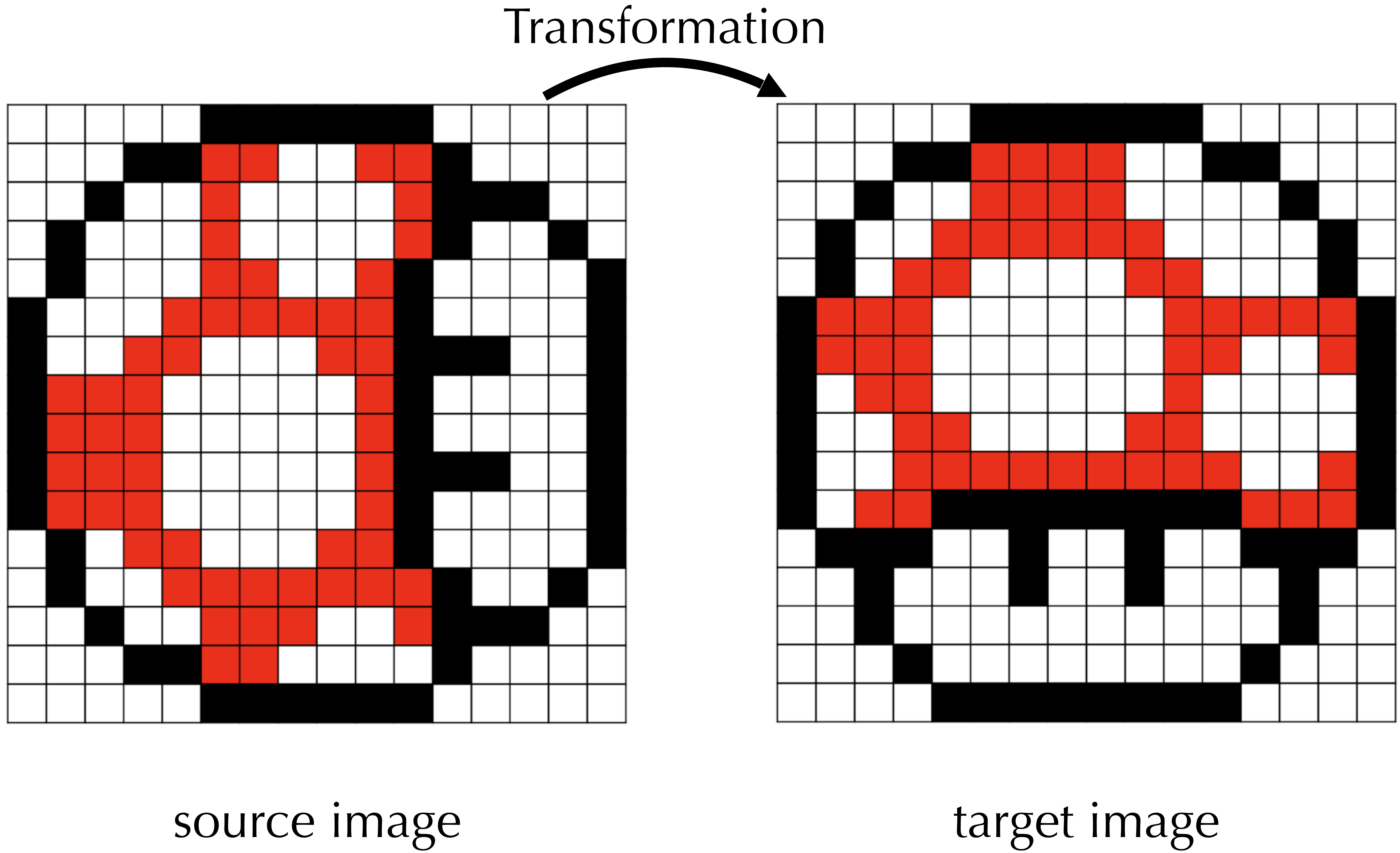
source image

target image

A Simple Example

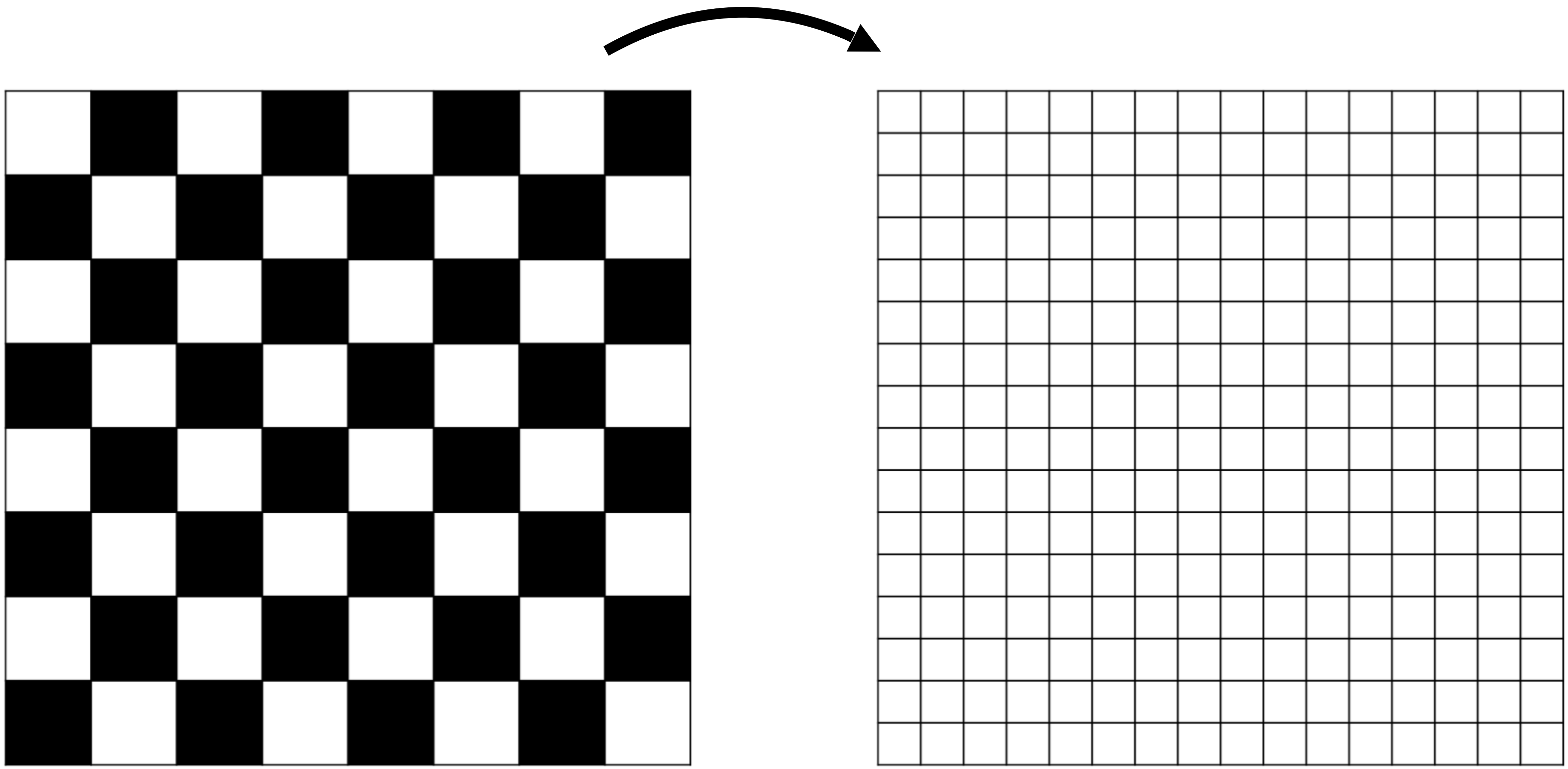


A Simple Example



Another Simple Example

Transformation

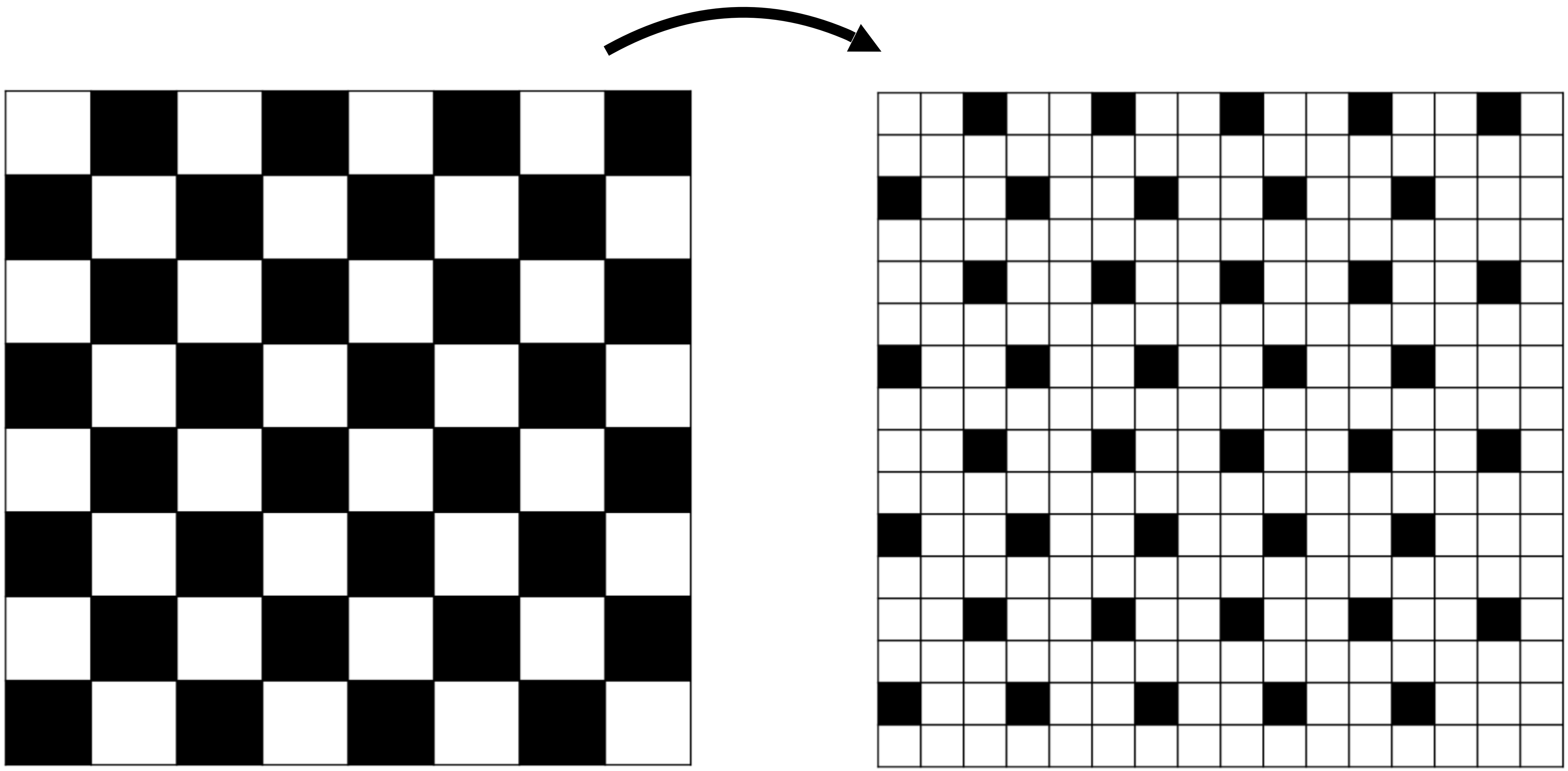


source image

target image

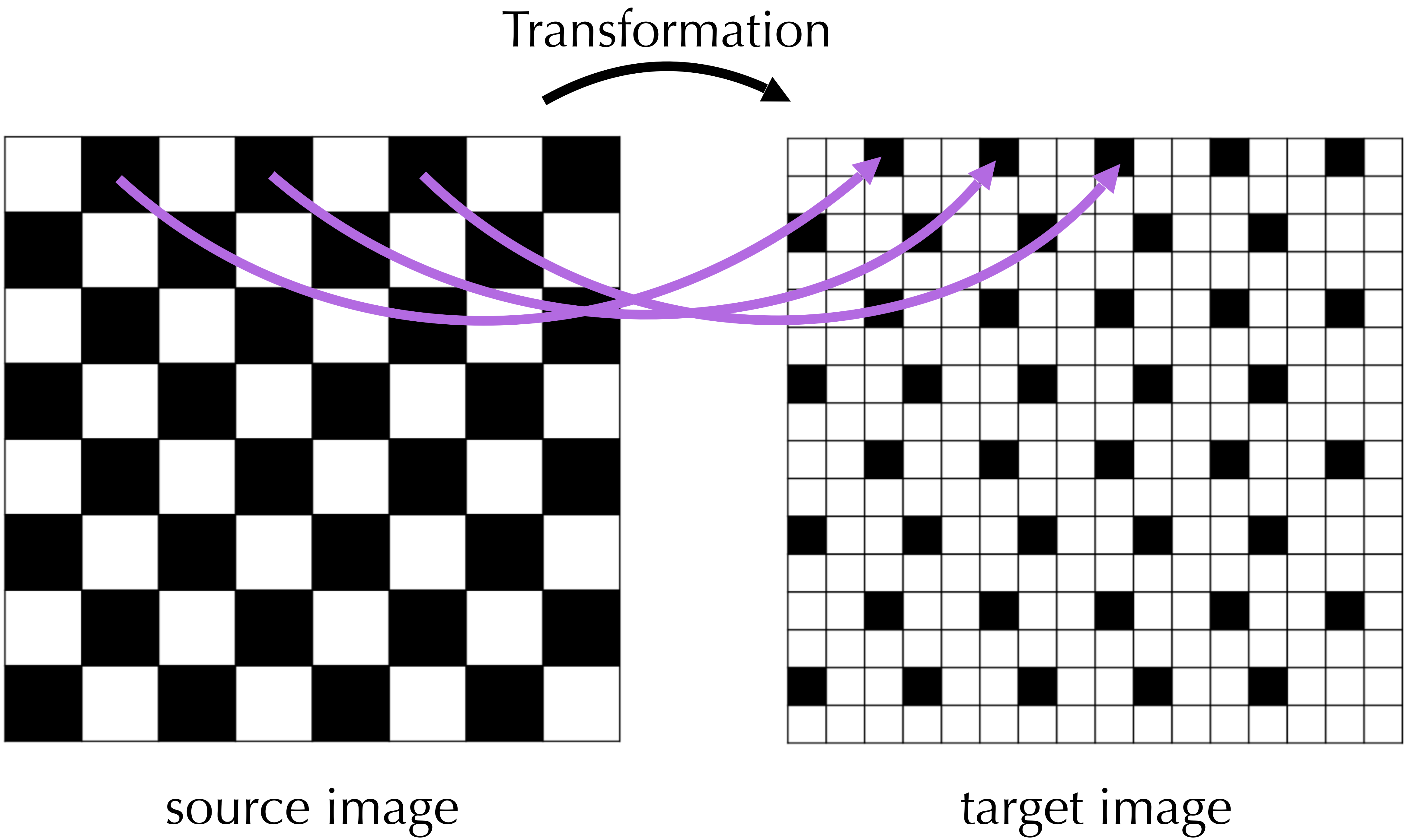
Another Simple Example

Transformation

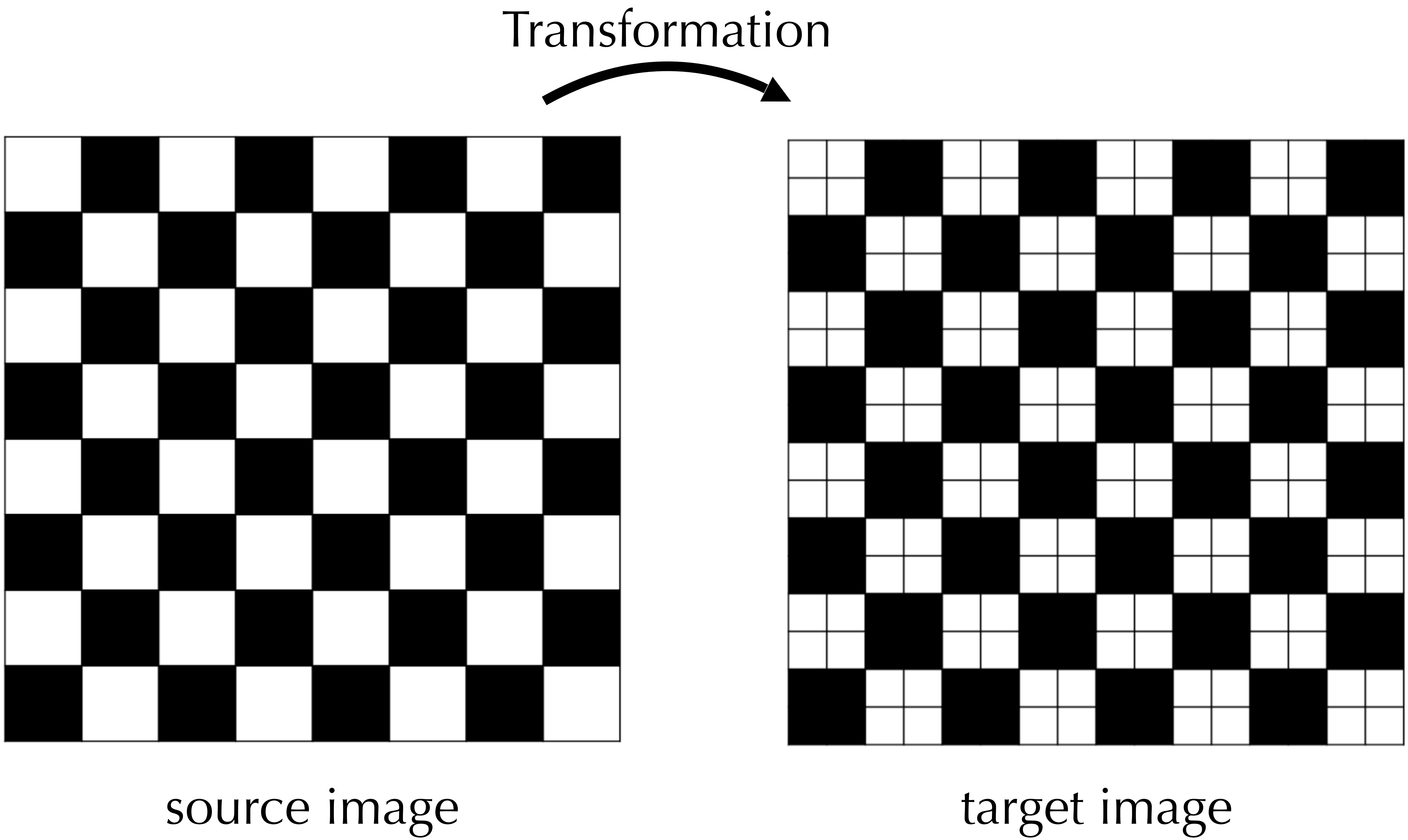


source image target image

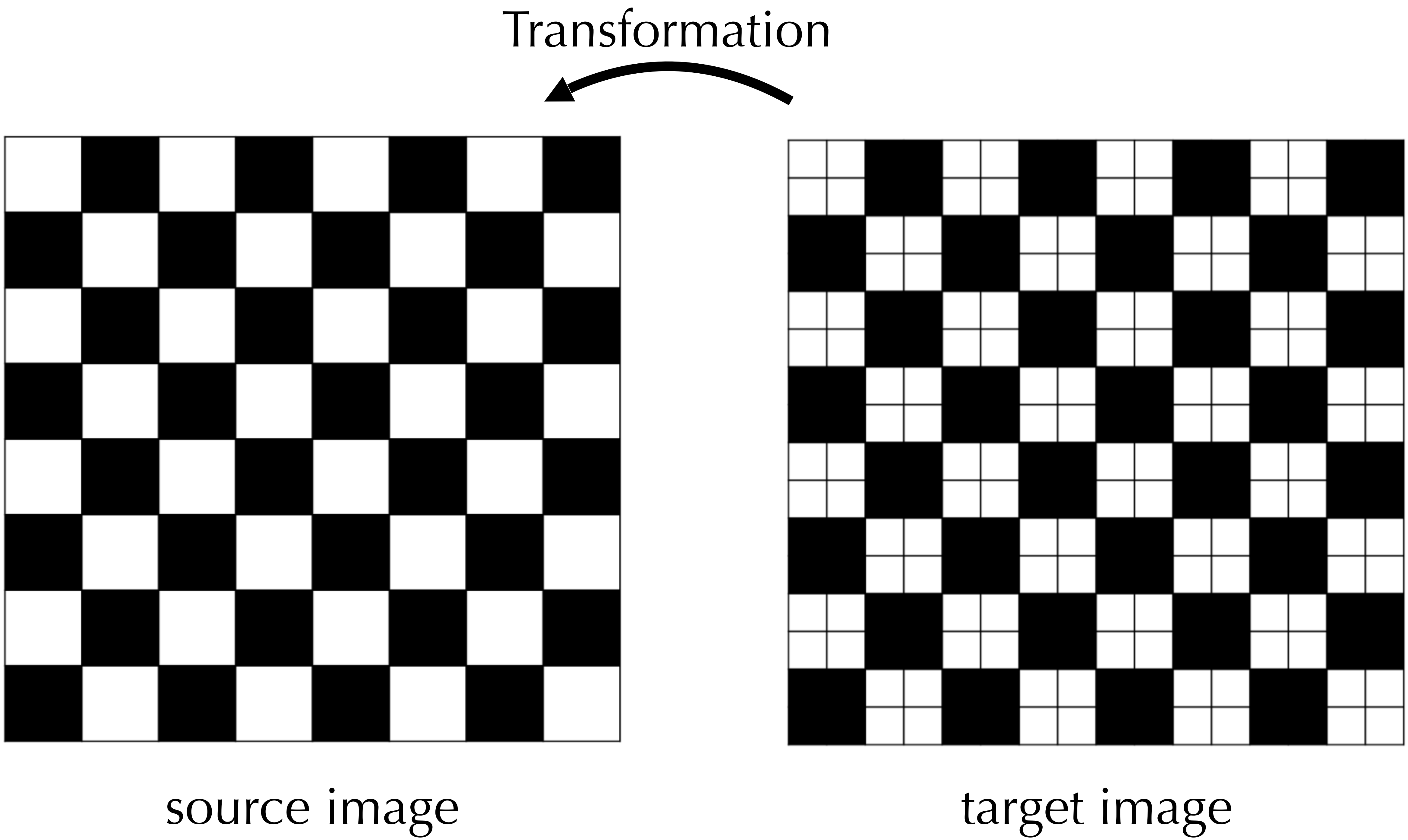
Another Simple Example



Another Simple Example



Another Simple Example



Another Simple Example

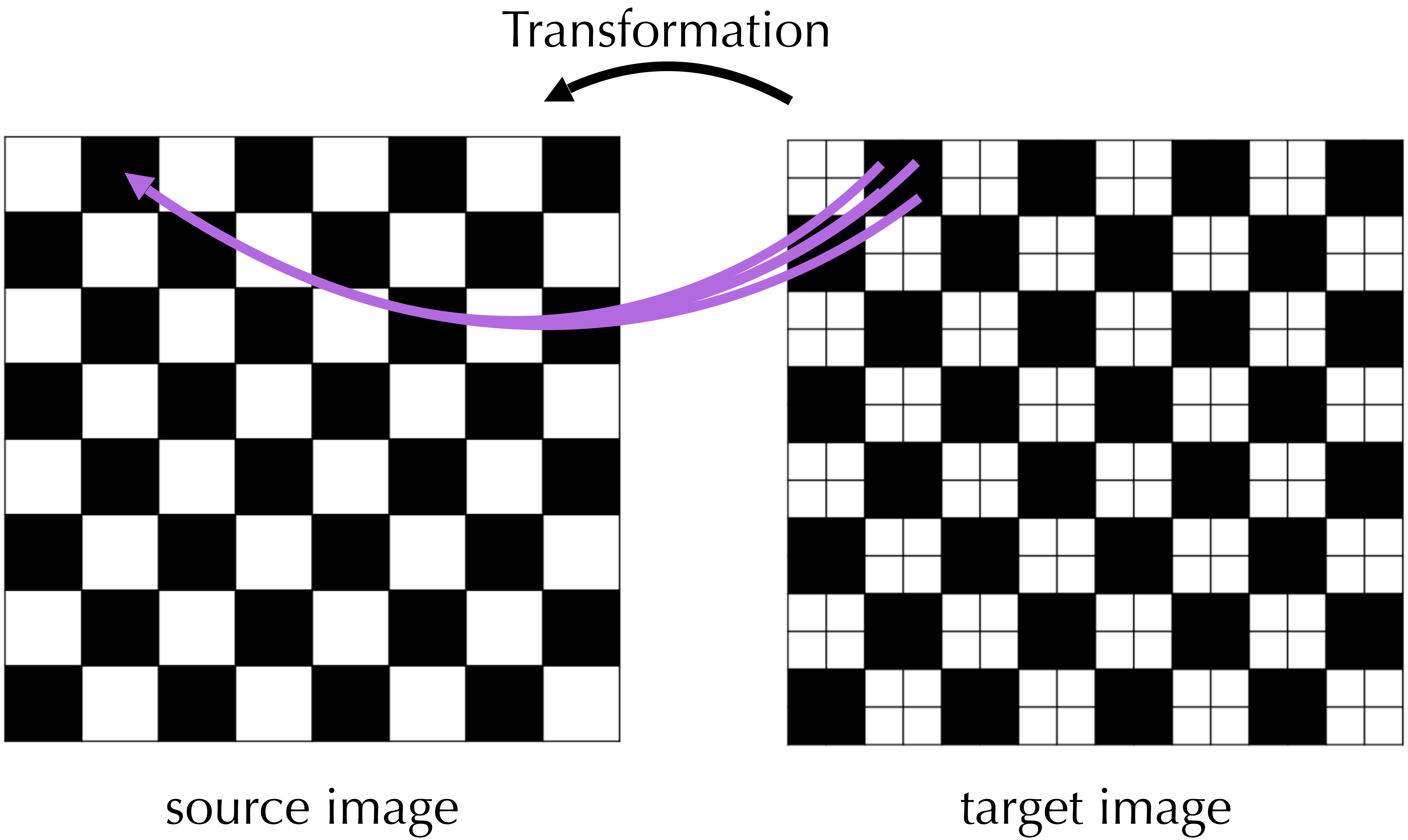


Image Warping

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

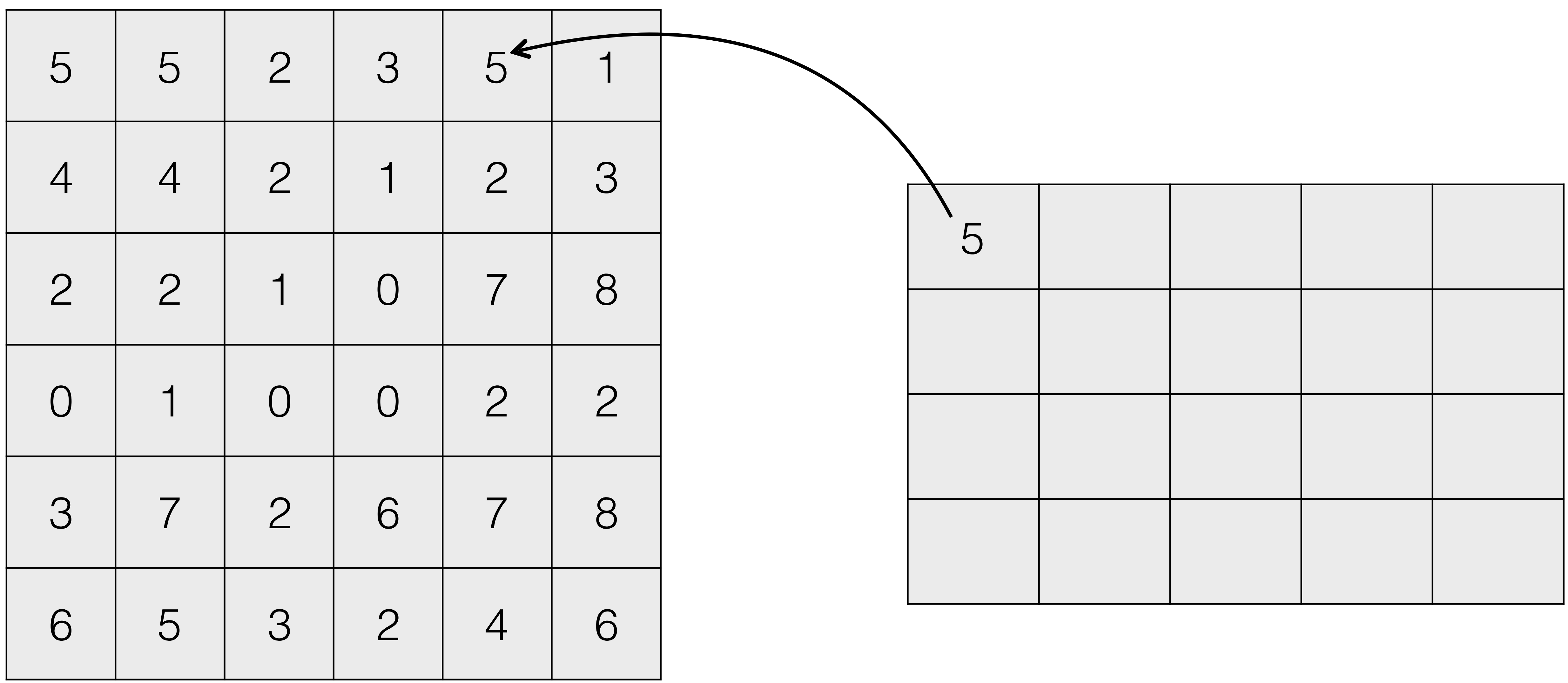
Image Warping

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

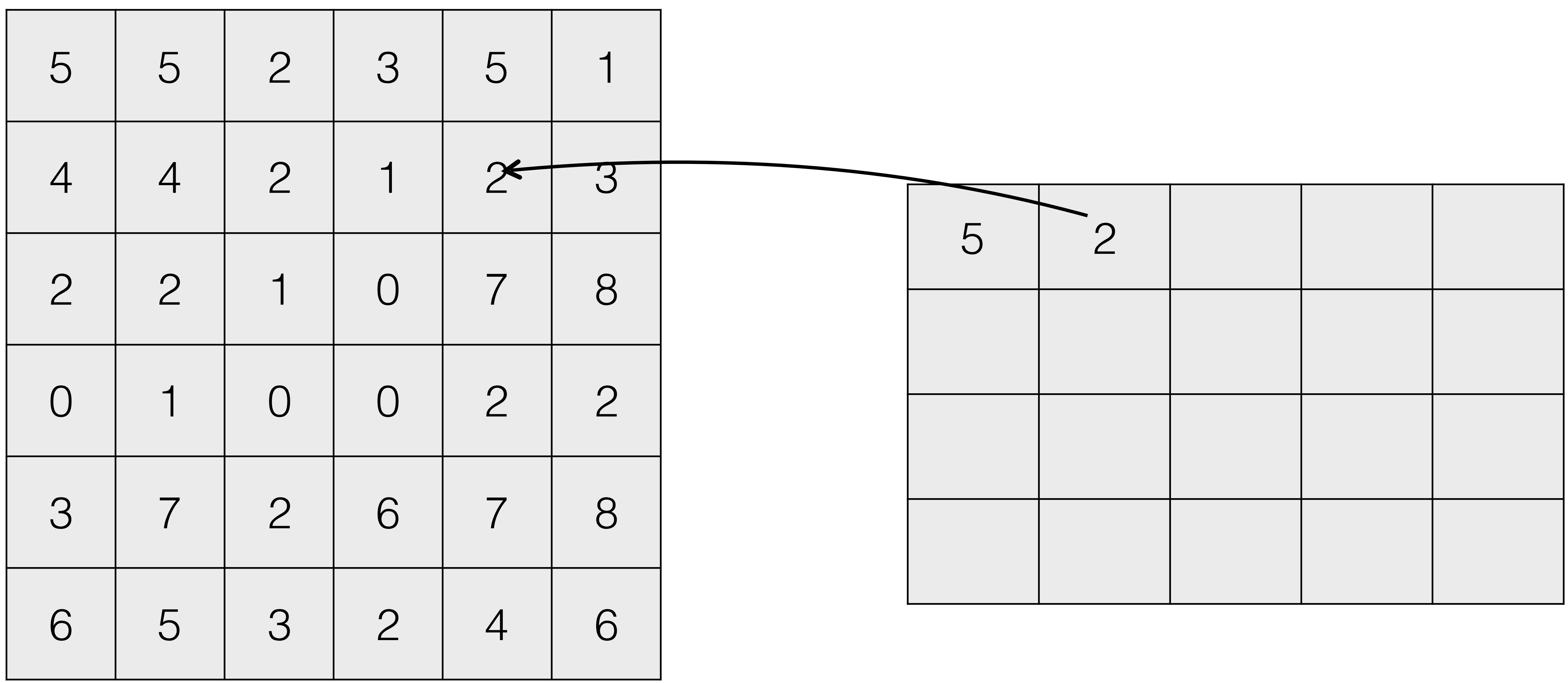
Image Warping



source image

warped image

Image Warping



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	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

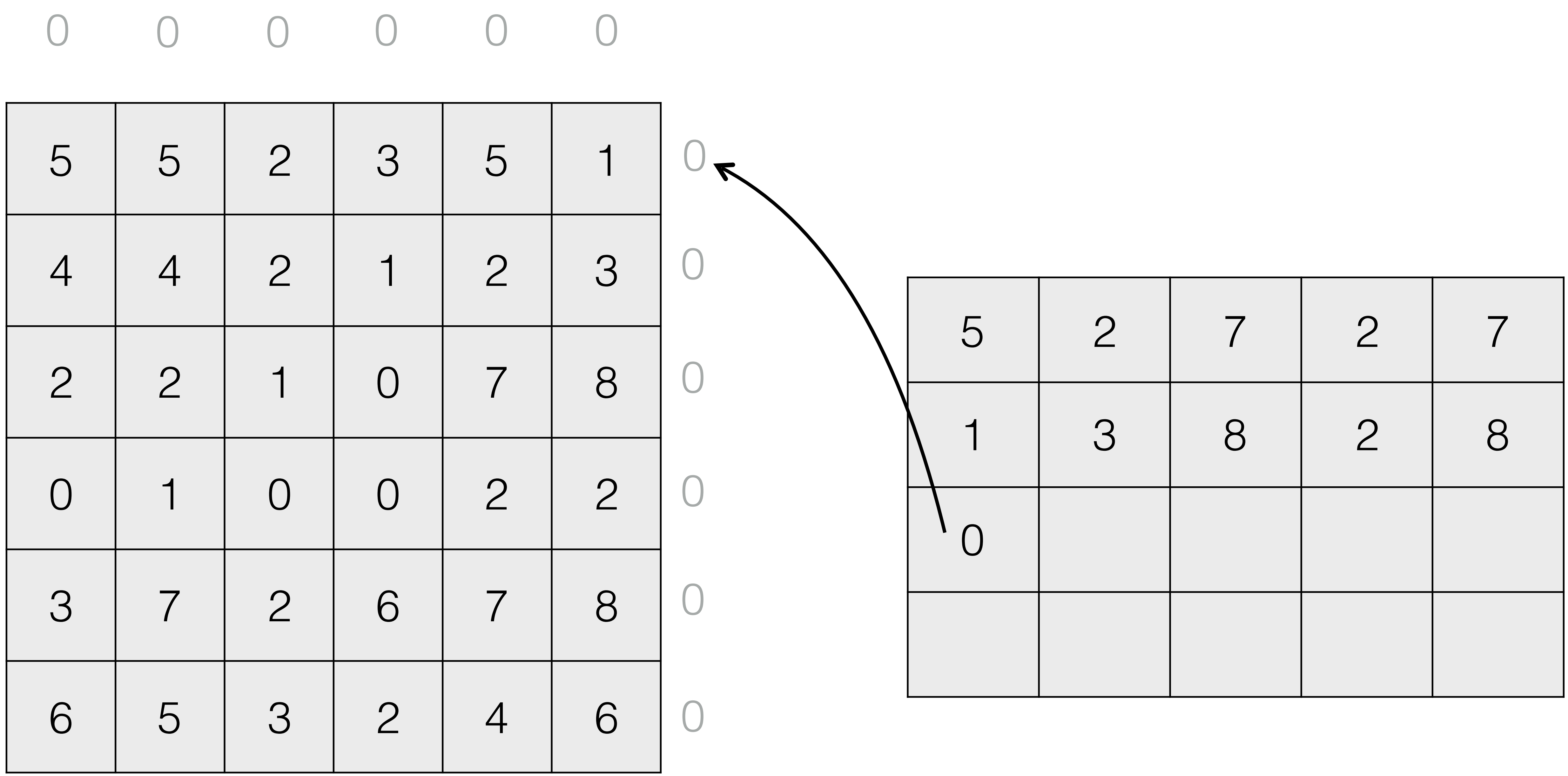
5	5	2	3	5	1
4	4	2	1	2	3
2	2	1	0	7	8
0	1	0	0	2	2
3	7	2	6	7	8
6	5	3	2	4	6

source image

5	2	7	2	7

warped image

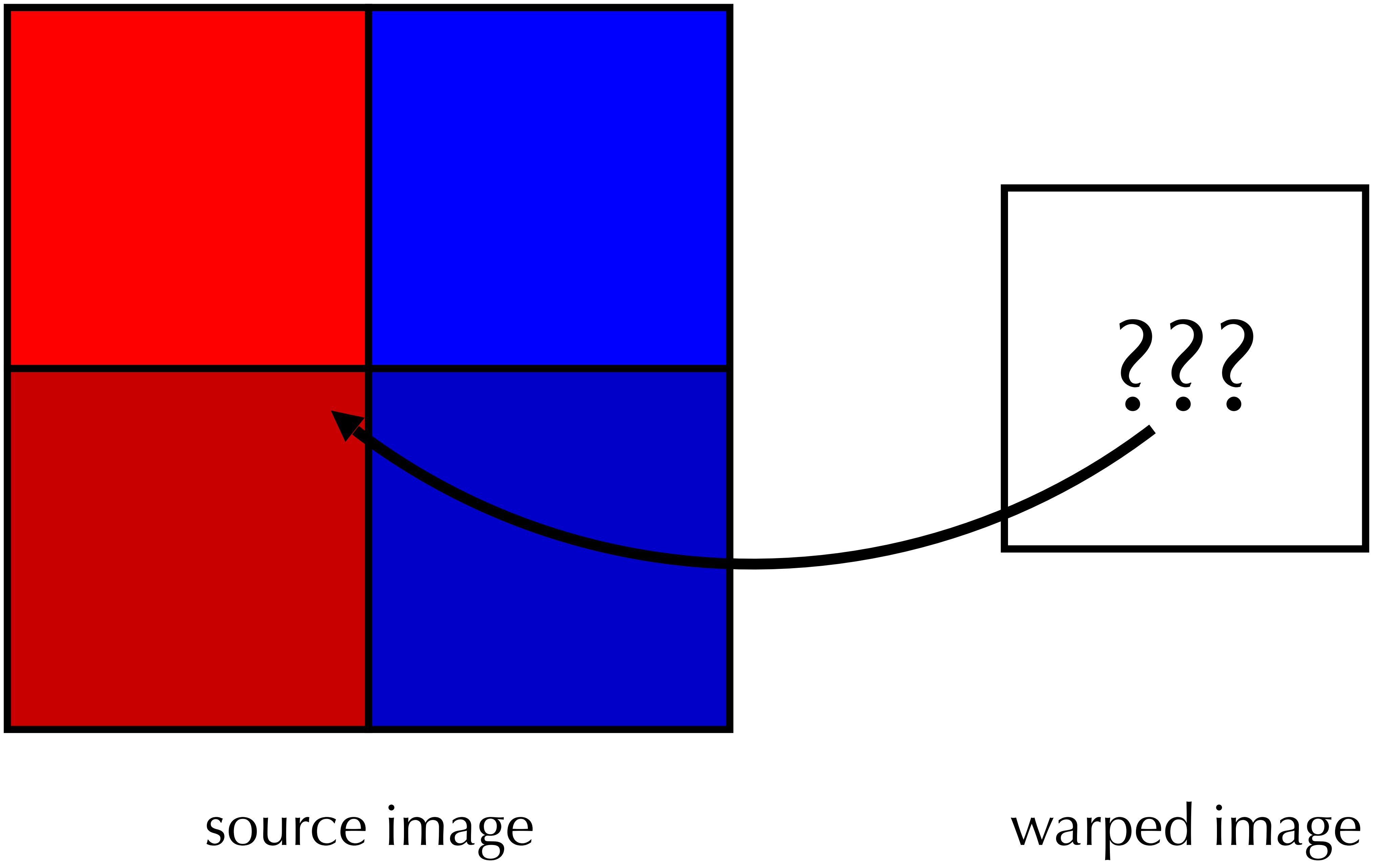
Image Warping



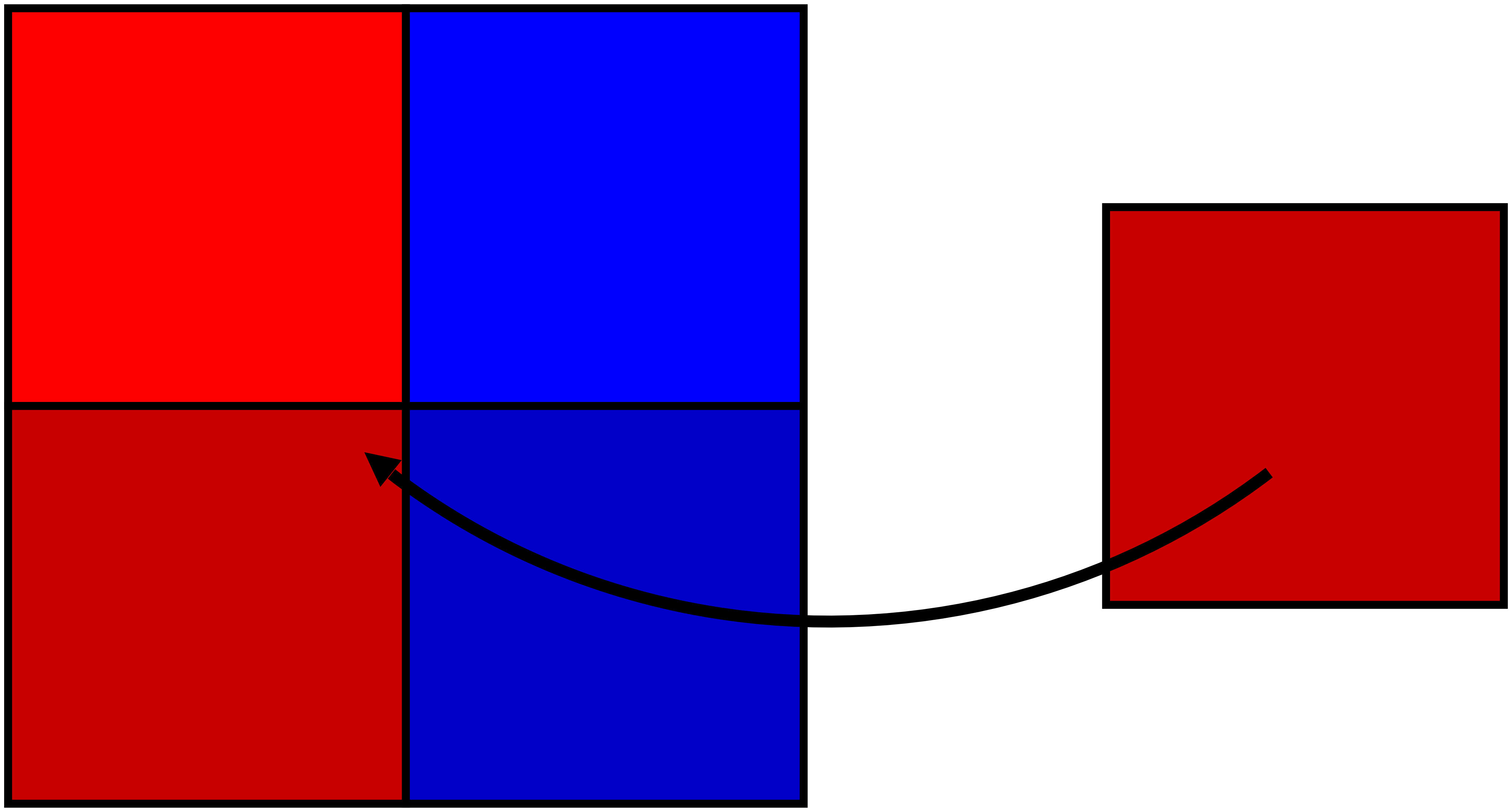
source image

warped image

Pixel Interpolation



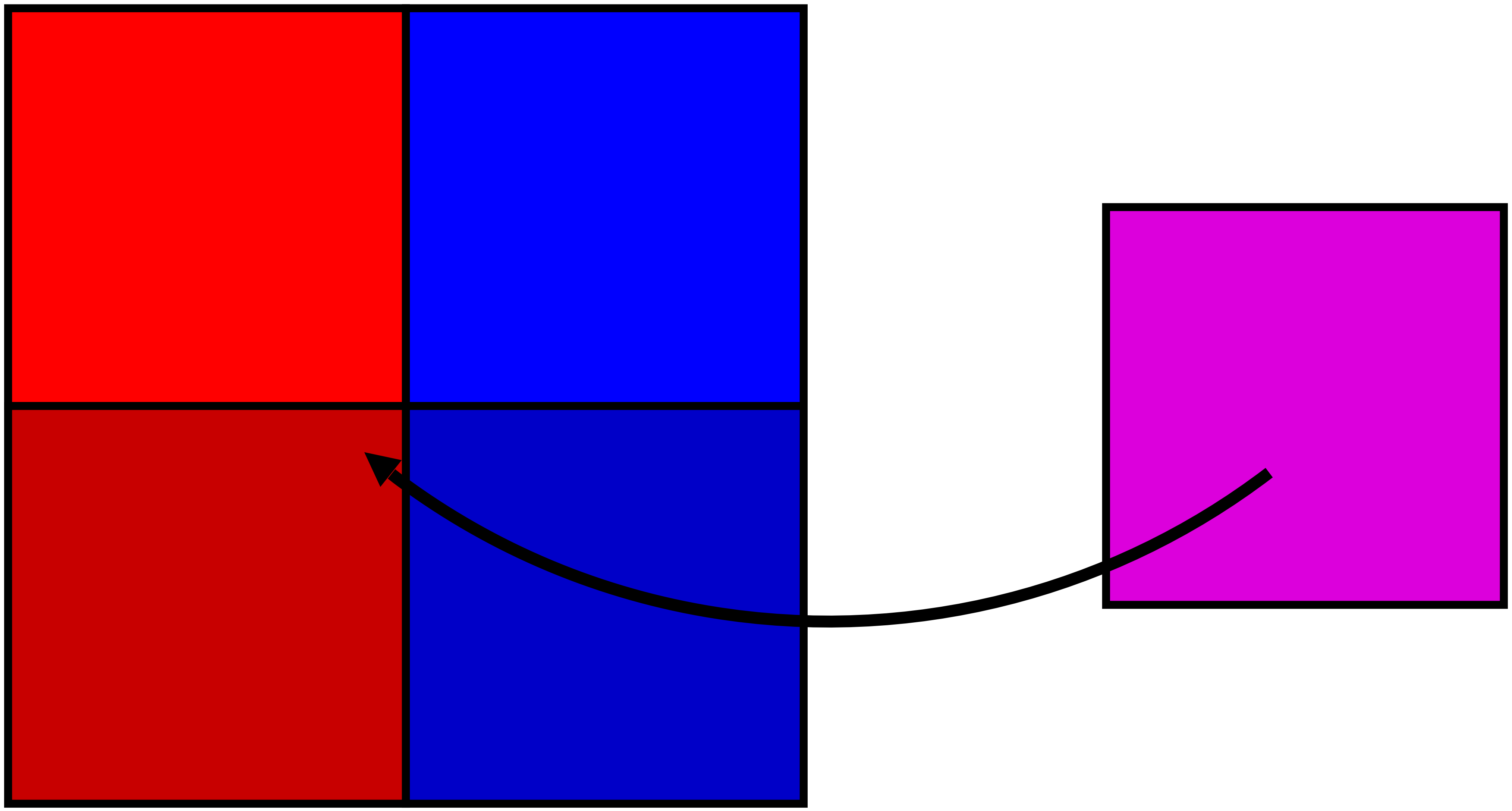
Nearest Neighbor



source image

warped image

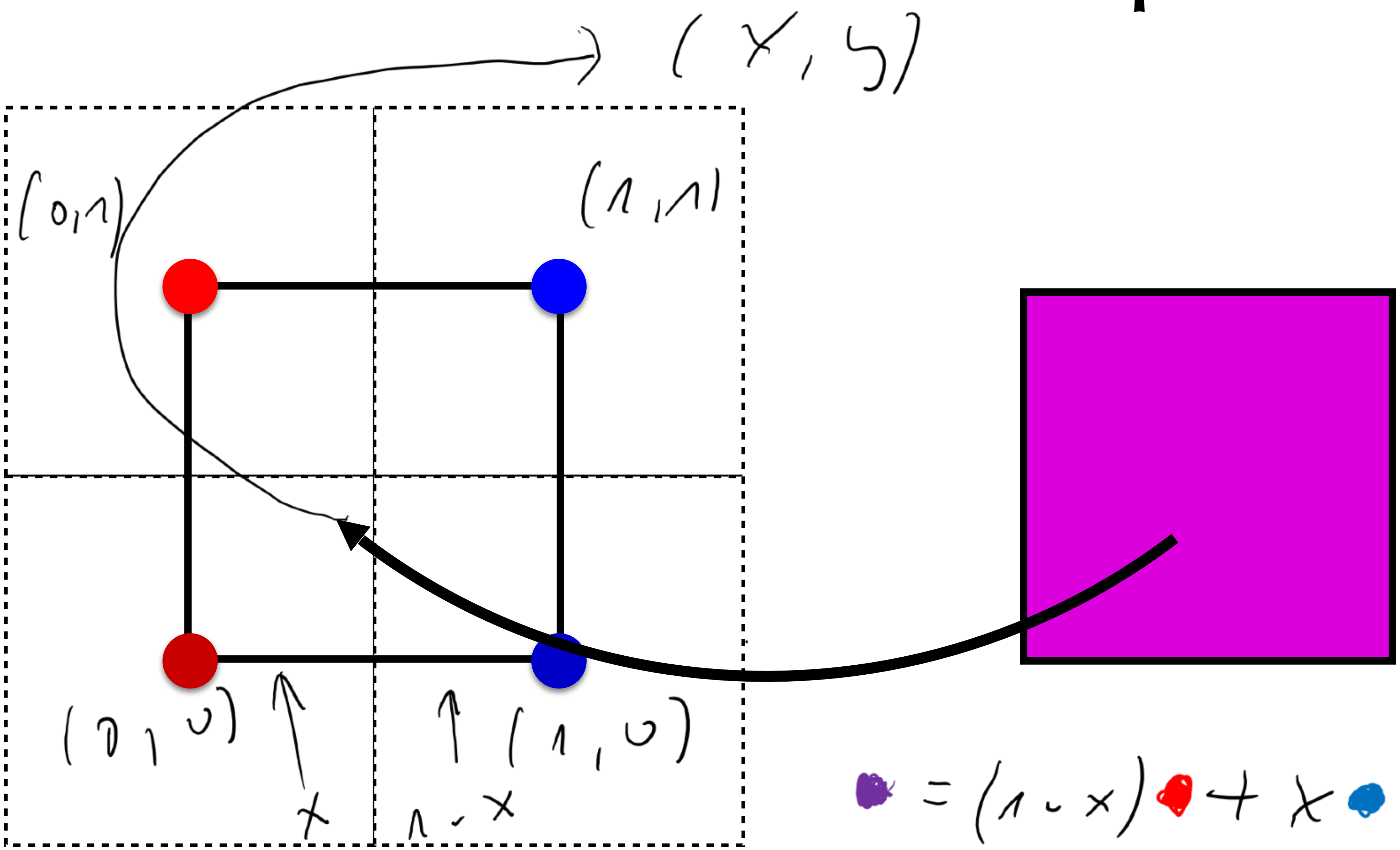
Bilinear Interpolation



source image

warped image

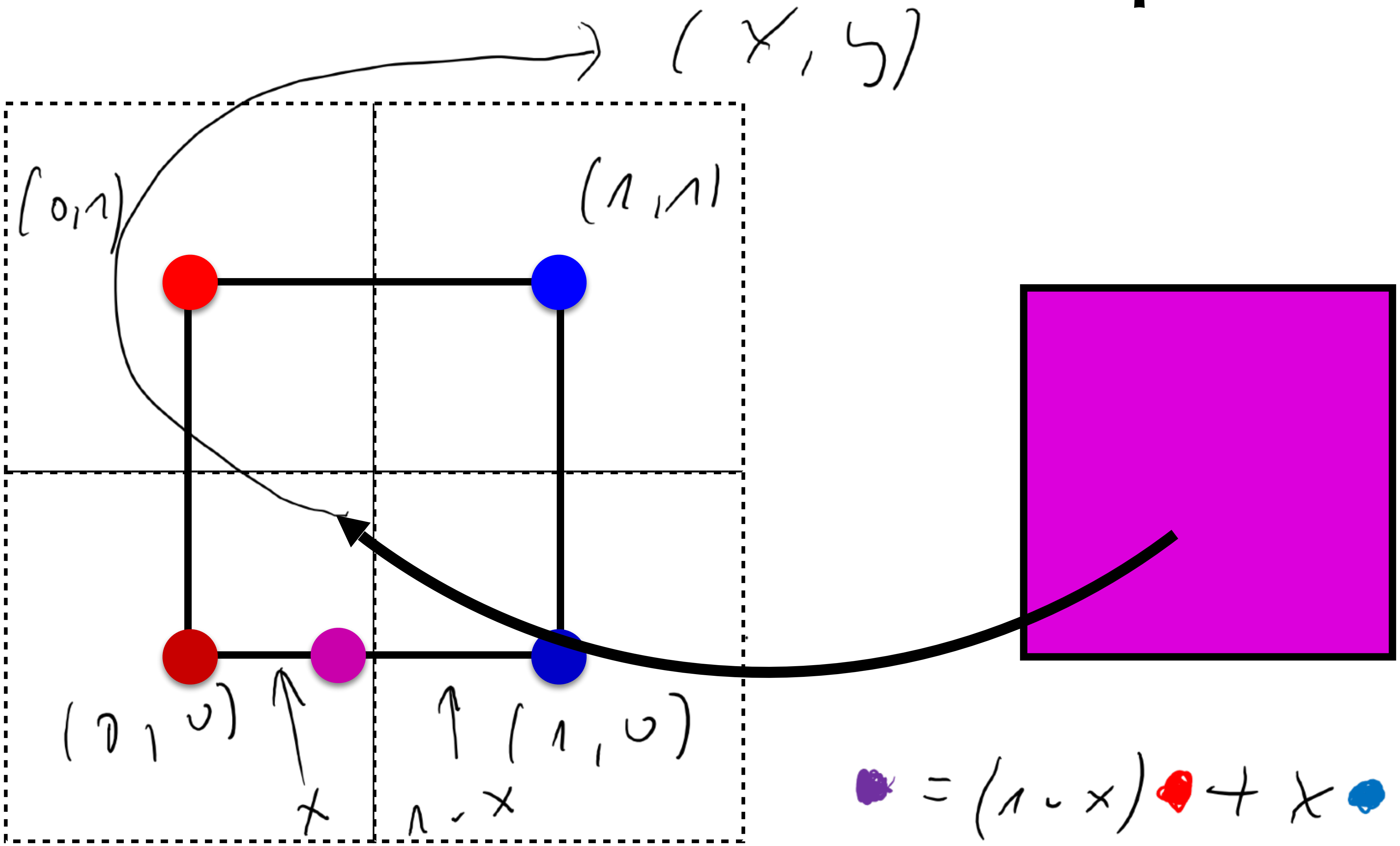
Bilinear Interpolation



source image

warped image

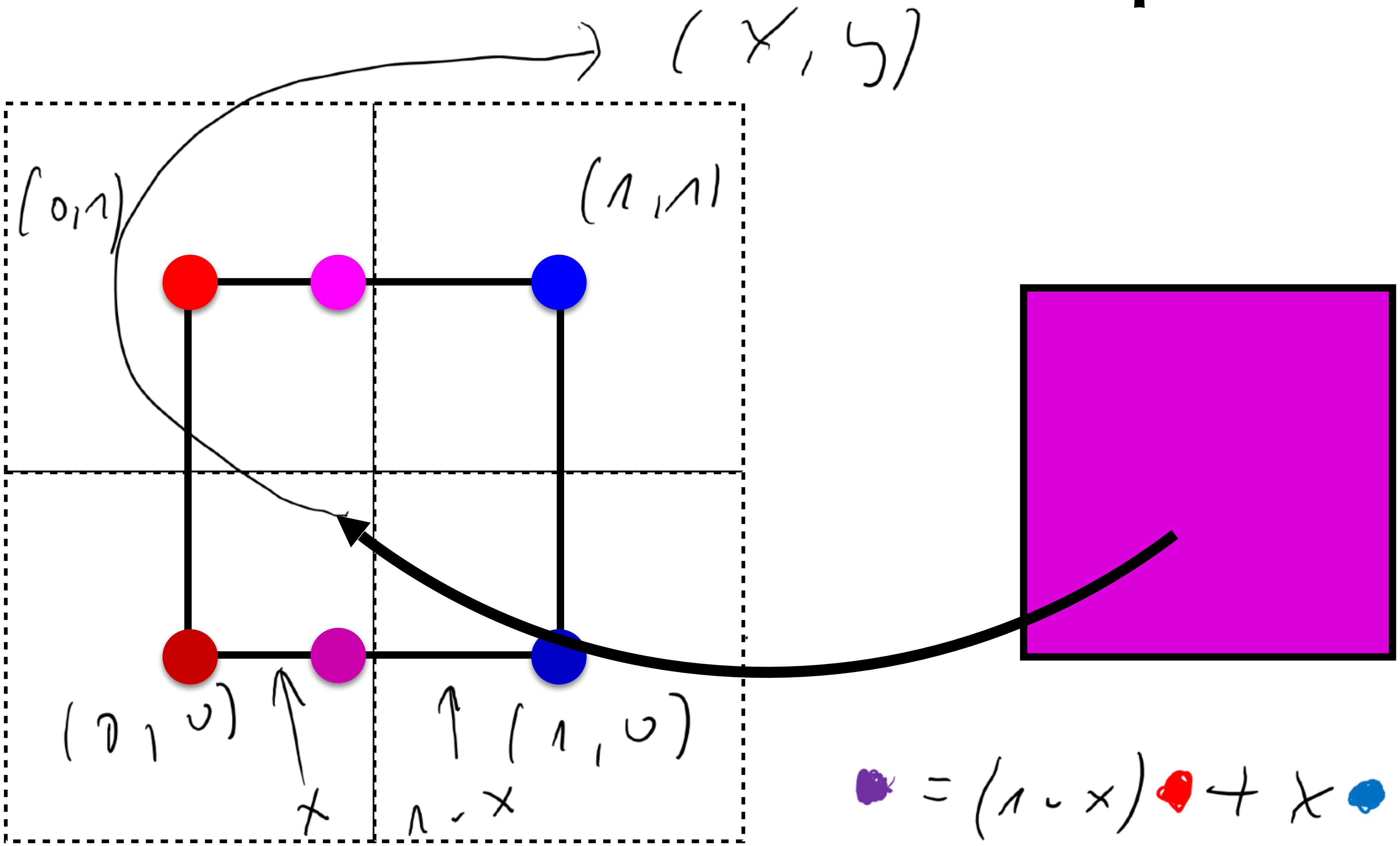
Bilinear Interpolation



source image

warped image

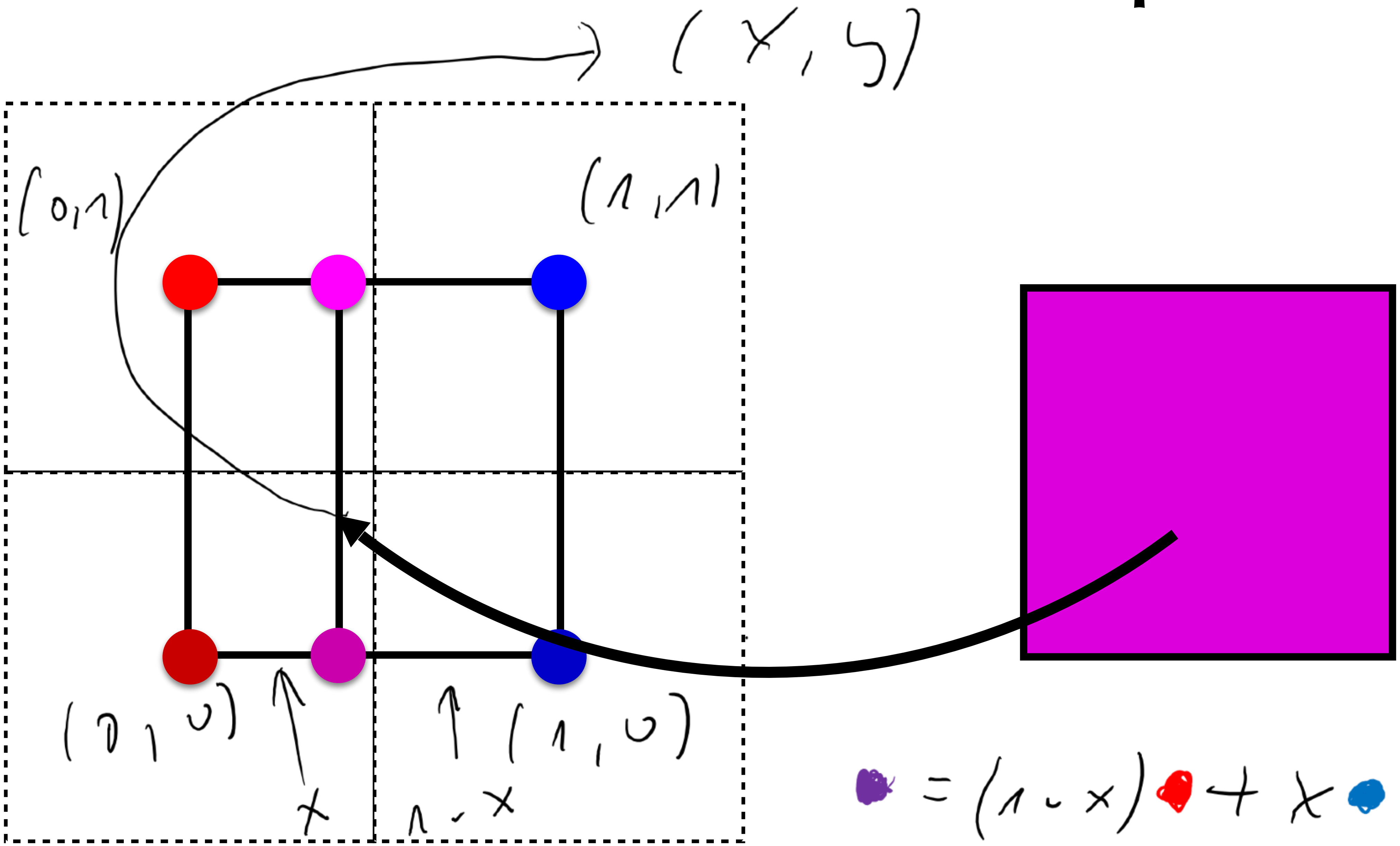
Bilinear Interpolation



source image

warped image

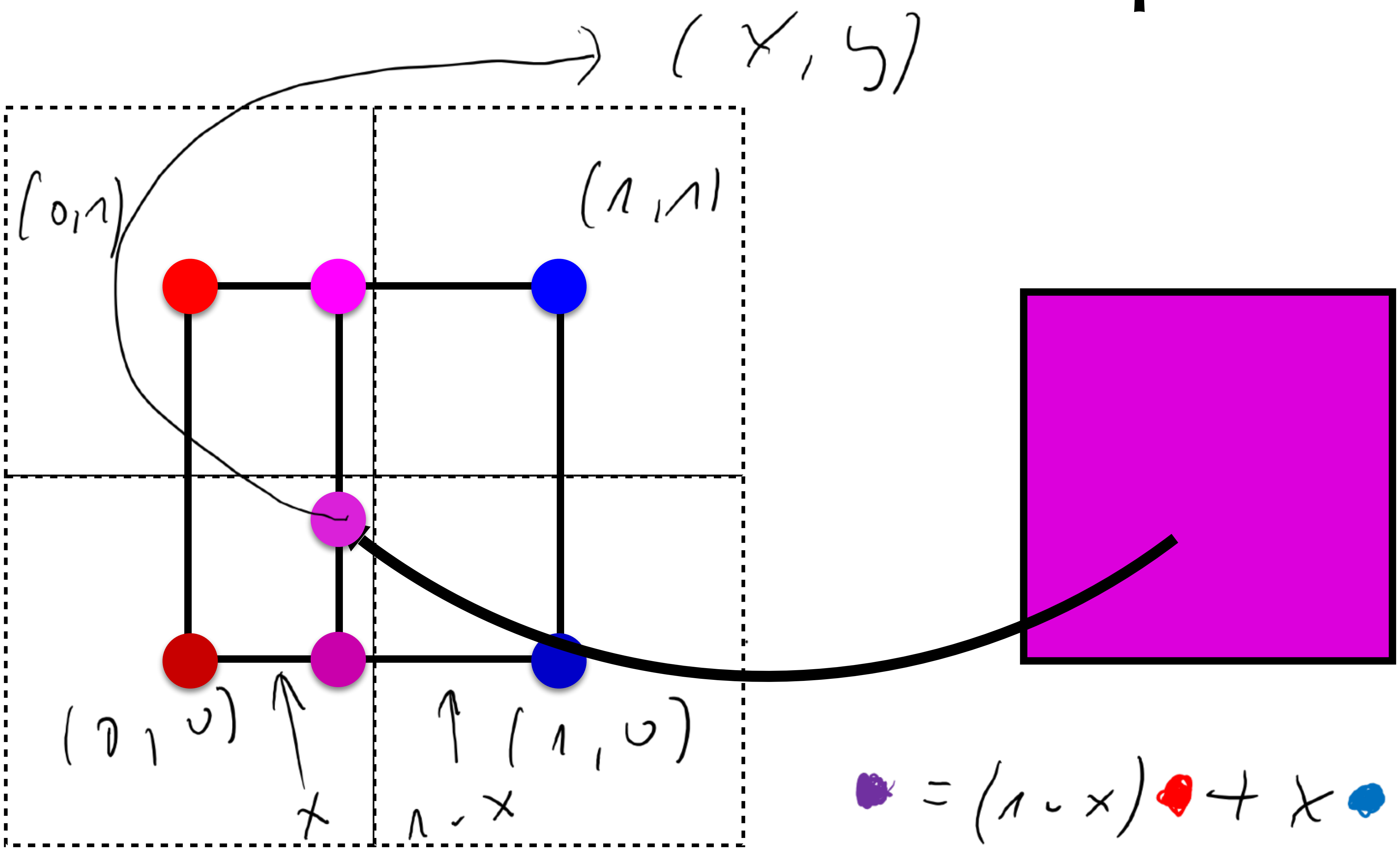
Bilinear Interpolation



source image

warped image

Bilinear Interpolation



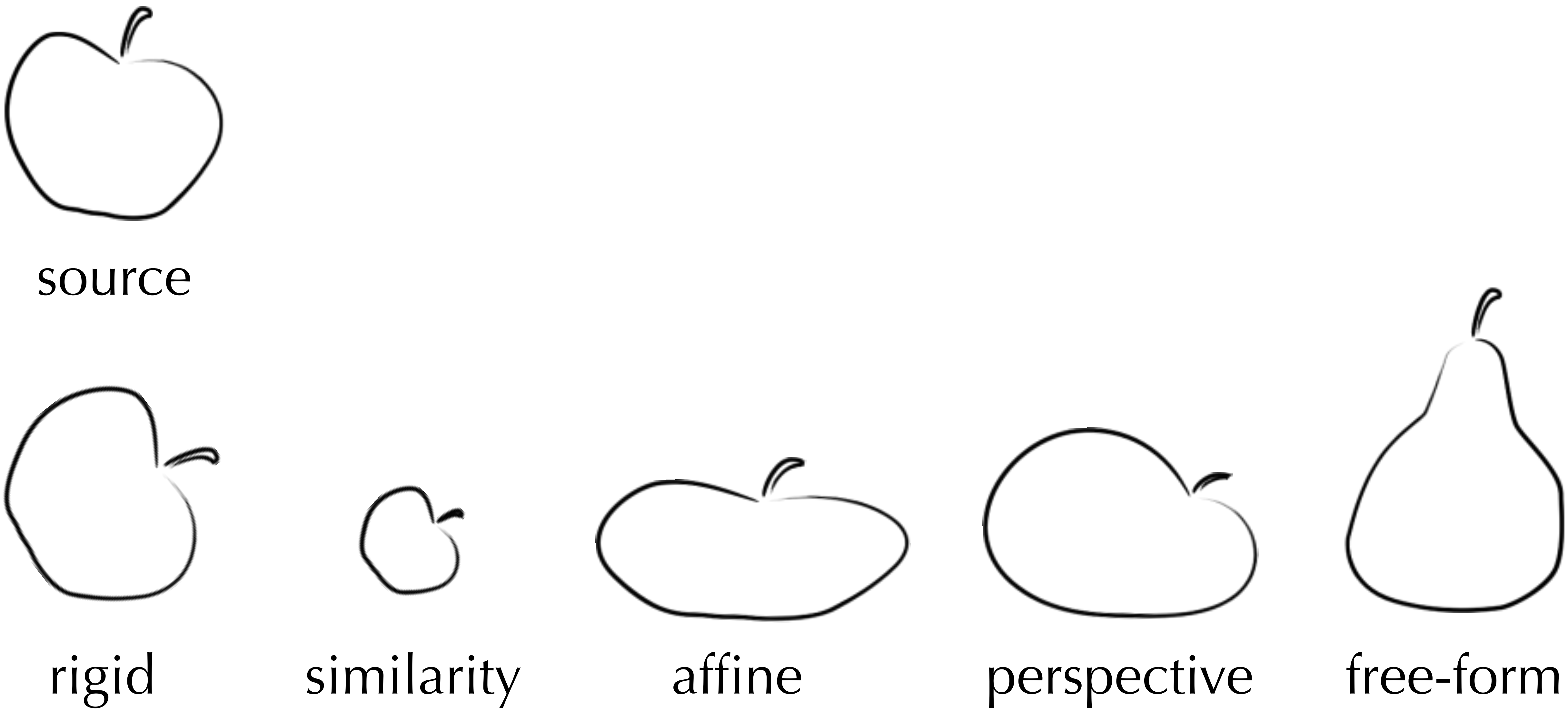
source image

warped image

Today

- Image Warping
- Transformation Estimation

Different Transformation Types



Different Transformation Types

source

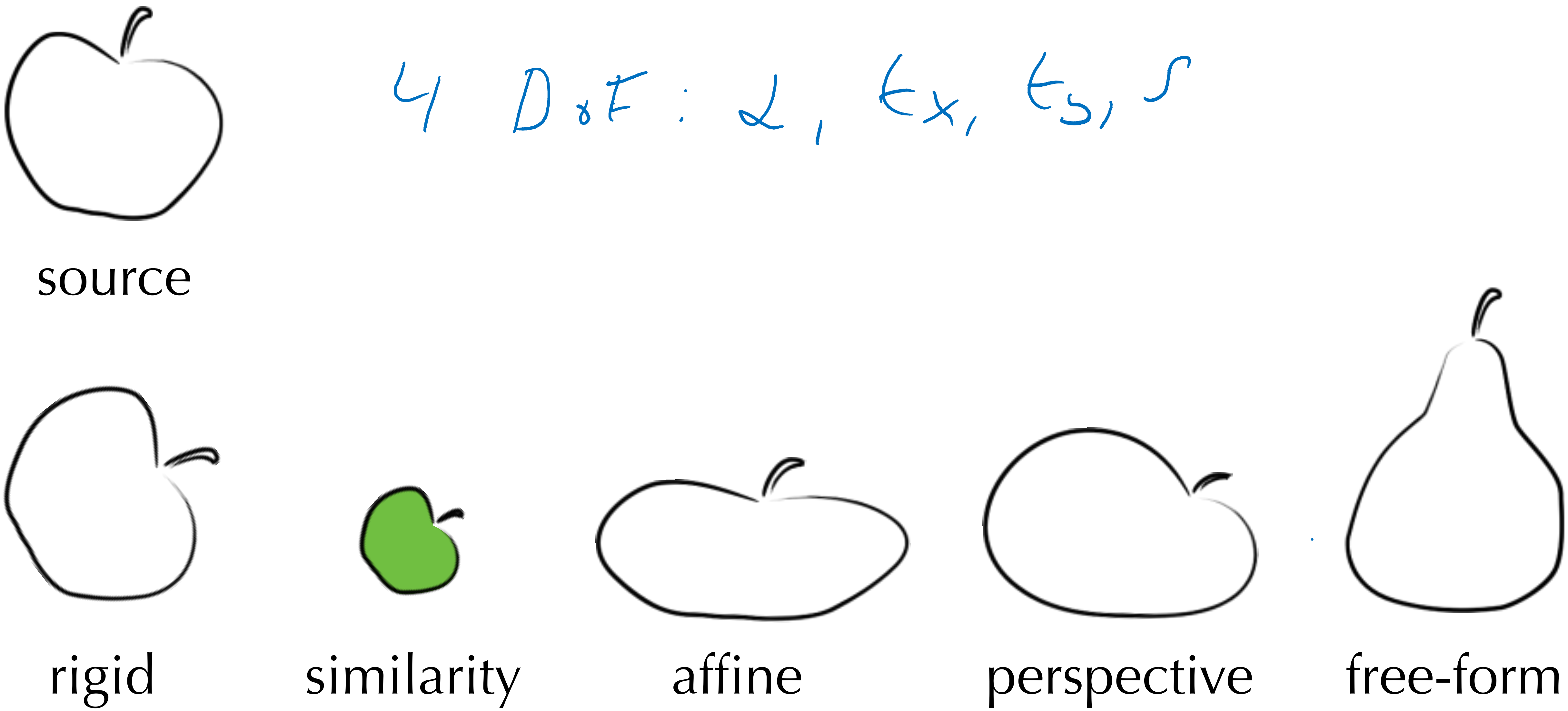
3 degrees of freedom
(3 DoF): α, t_x, t_y

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

4 DoF: α, t_x, t_y, s

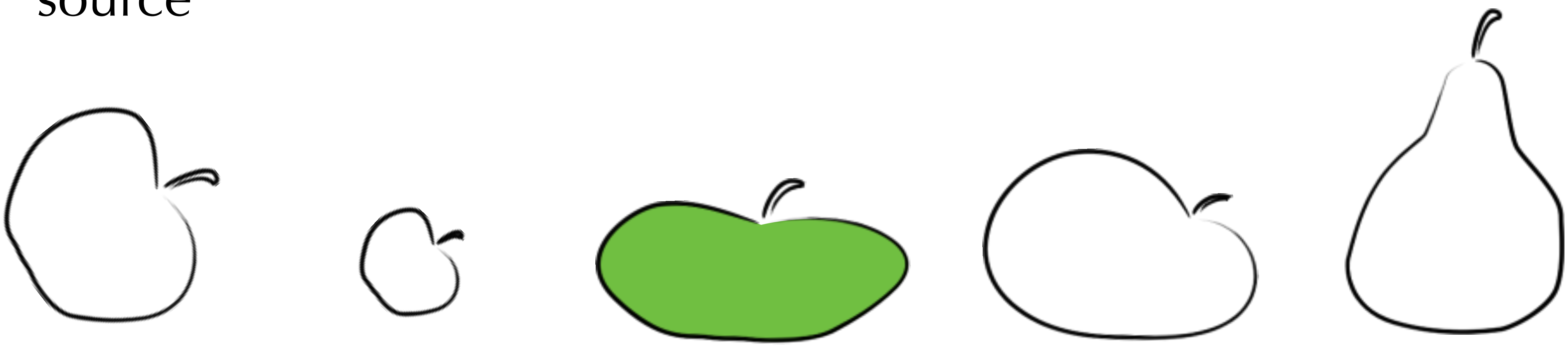


$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Different Transformation Types

source

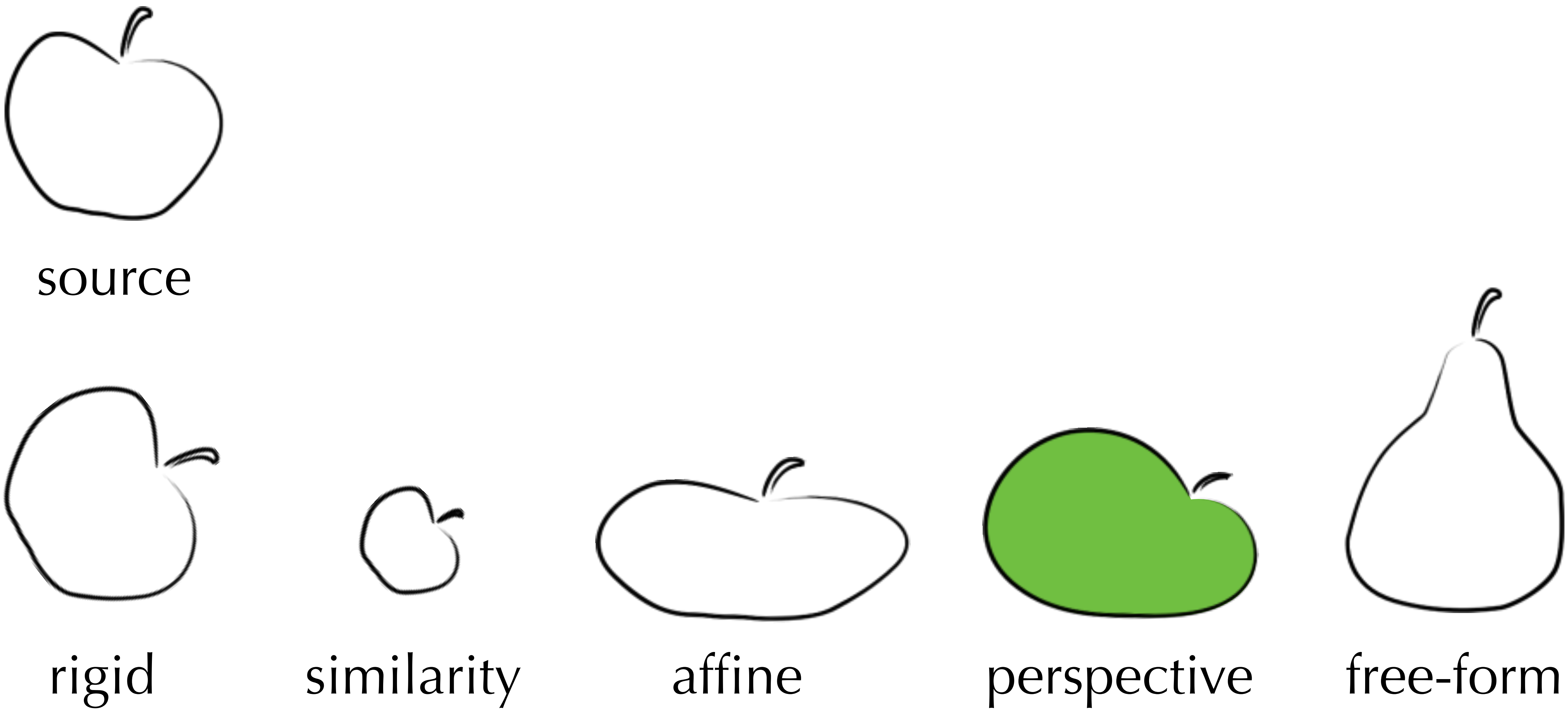
6 DoF:
a, b, c, d, t_x, t_y



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



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

$\hat{w} \neq 0$

Different Transformation Types

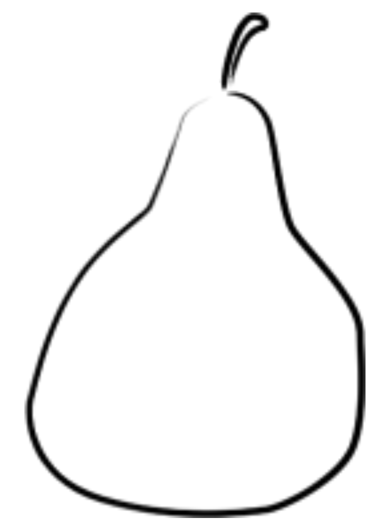
source

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:

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

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

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

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:

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

parameters to be estimated

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

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

$$\_\ \_ = A^{-1} b$$

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

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

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

$$= A^{-1} b$$

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

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

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

Solve in Matlab as `x=A\b` $= A^{-1} b$

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

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

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

$\mathtt{A}$ $\qquad\qquad$ $\mathtt{x}$ $\qquad\qquad$ $\mathtt{b}$

Solve in Matlab as `x=A\b` $= A^{-1} b$

Reshape parameters into matrix and translation $\begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$

Fitting A Similarity Transformation

Minimal solver from 2 correspondences

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

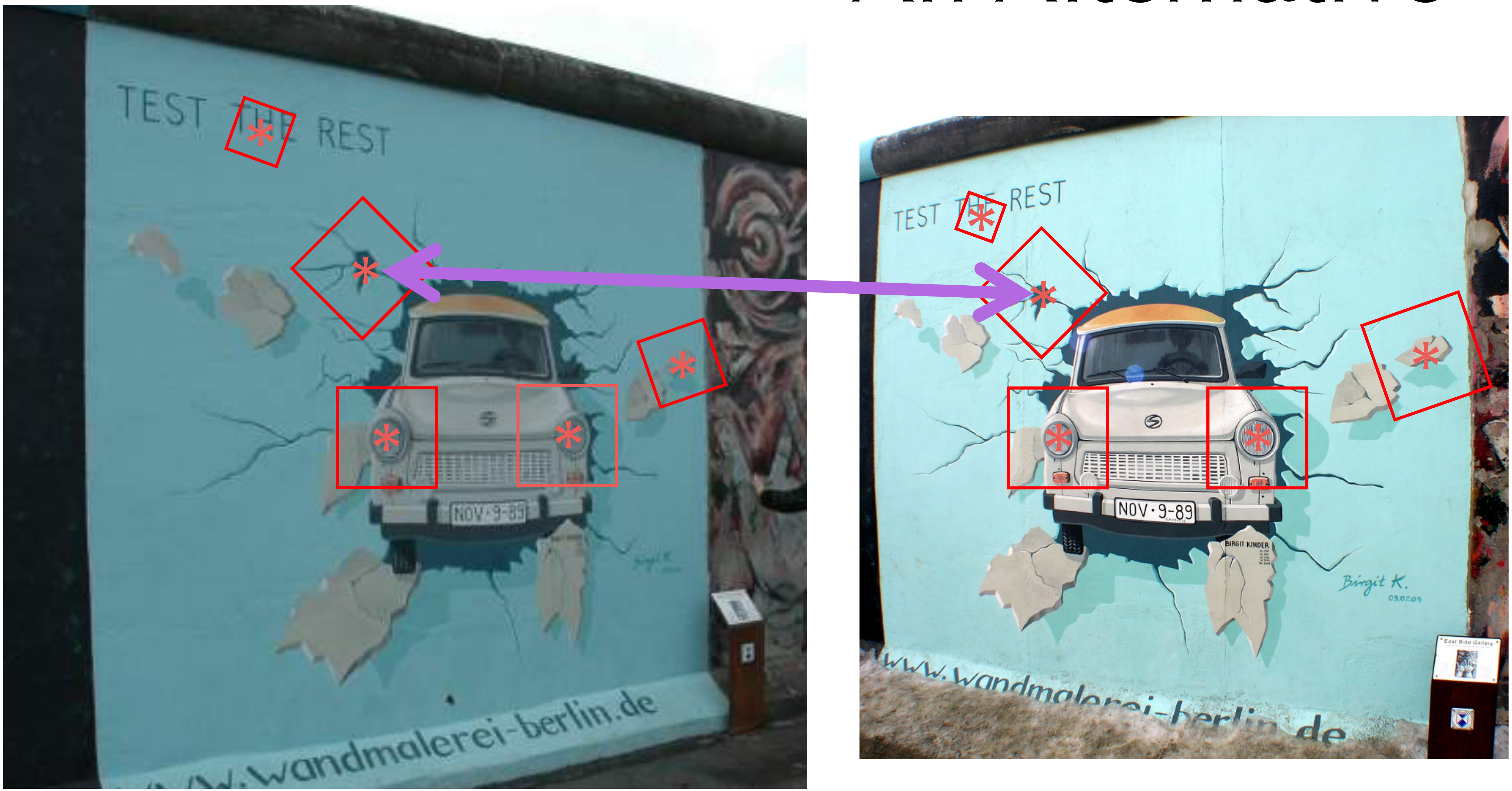
`A` `x` `b`

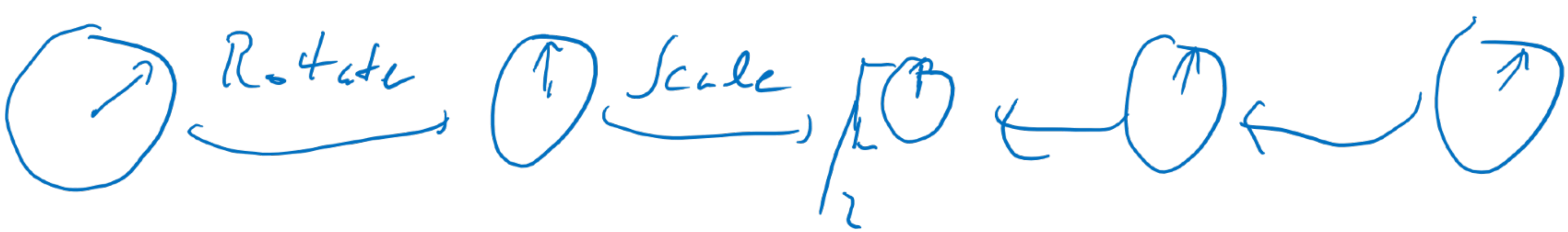
Solve in Matlab as `x=A\b` $= A^{-1} b$

Reshape parameters into matrix and translation $\begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$

Use solver inside RANSAC

An Alternative





Fitting A Similarity Transformation

Least squares solution:

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

Fitting A Similarity Transformation

Least squares solution:

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

Fitting A Similarity Transformation

Least squares solution:

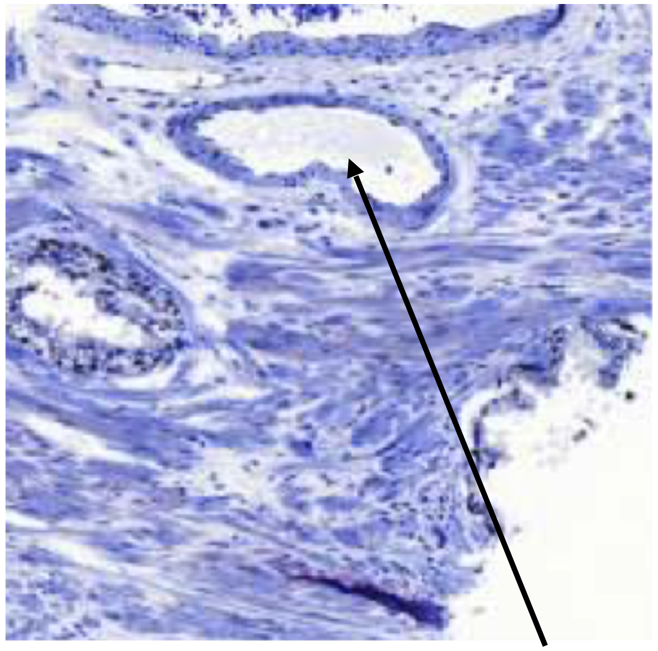
$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_2 \\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

`A` `x` `b`

Solve in Matlab as `x=A\b`

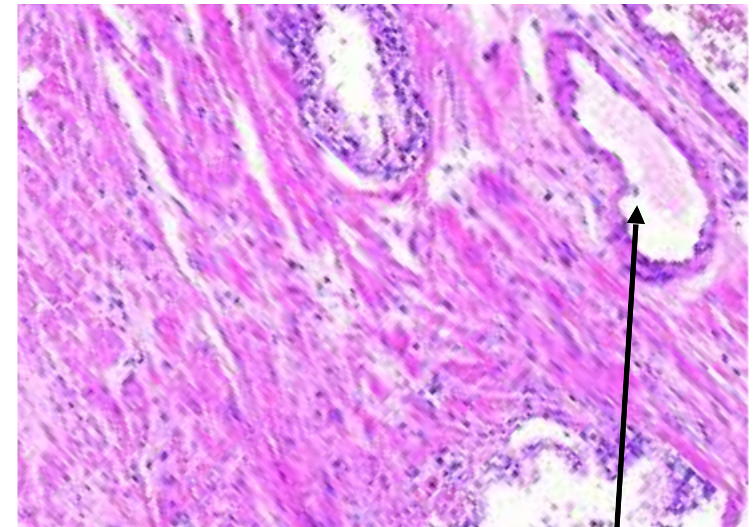
BUG ALERT

source image



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

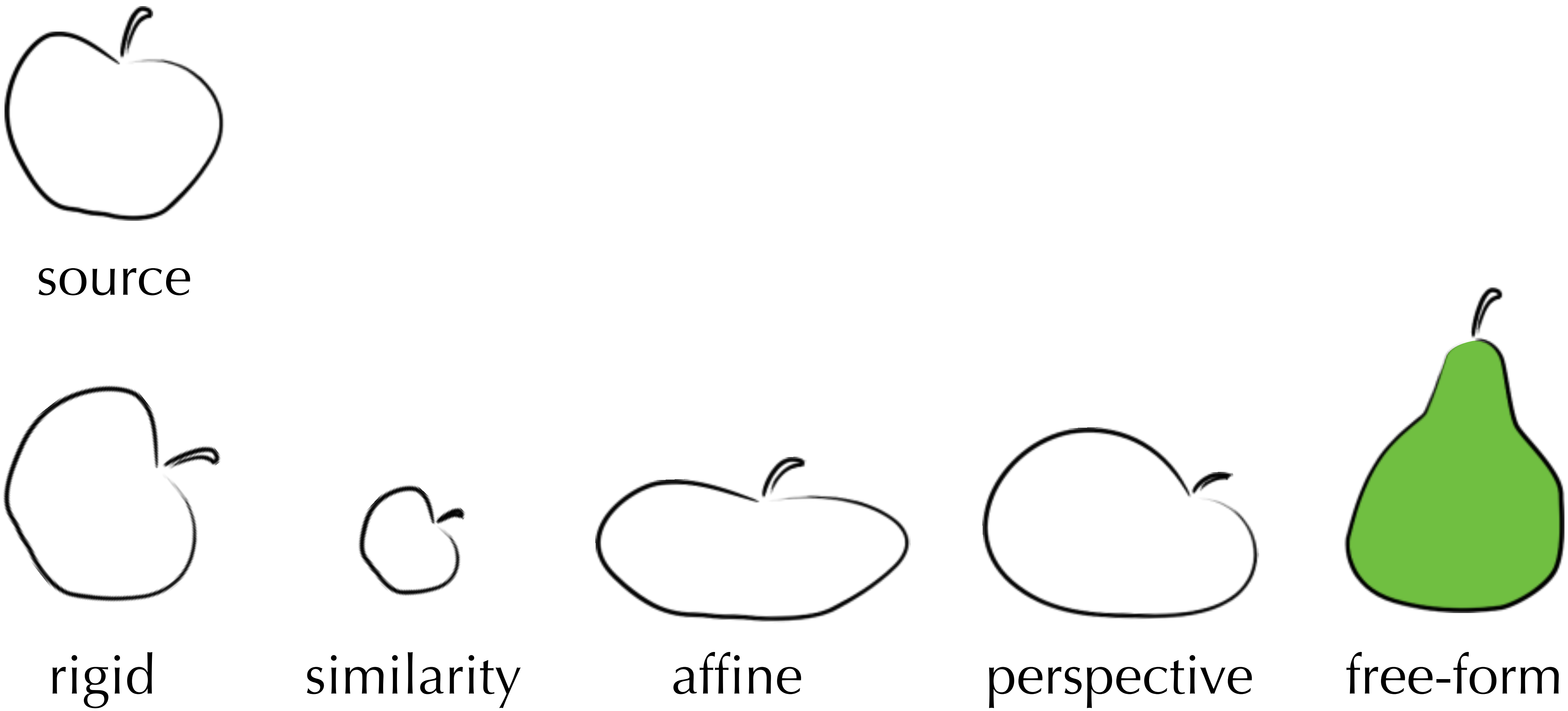
target image



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

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

Different Transformation Types

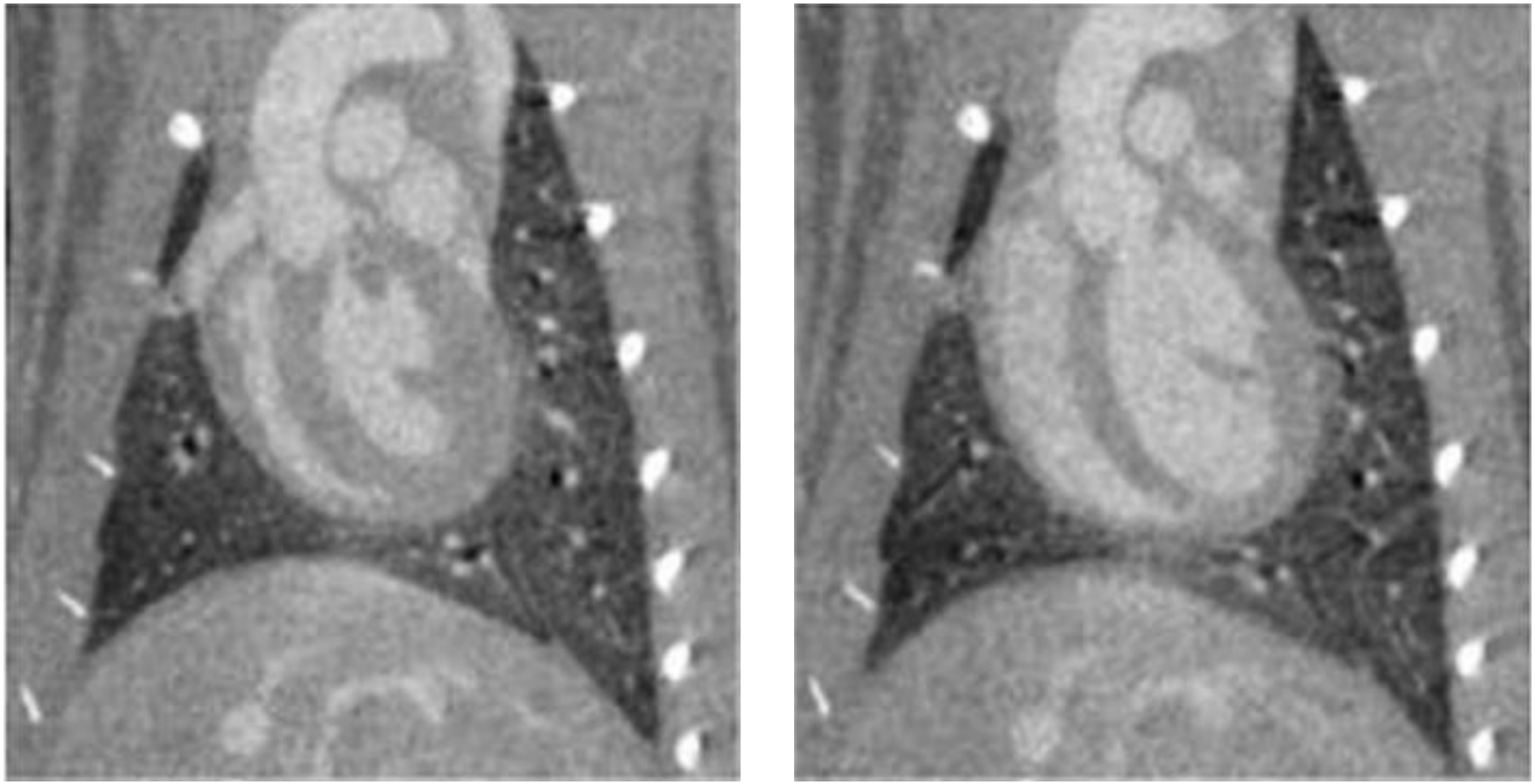


Free-Form Registration

source (systole)

target (diastole)

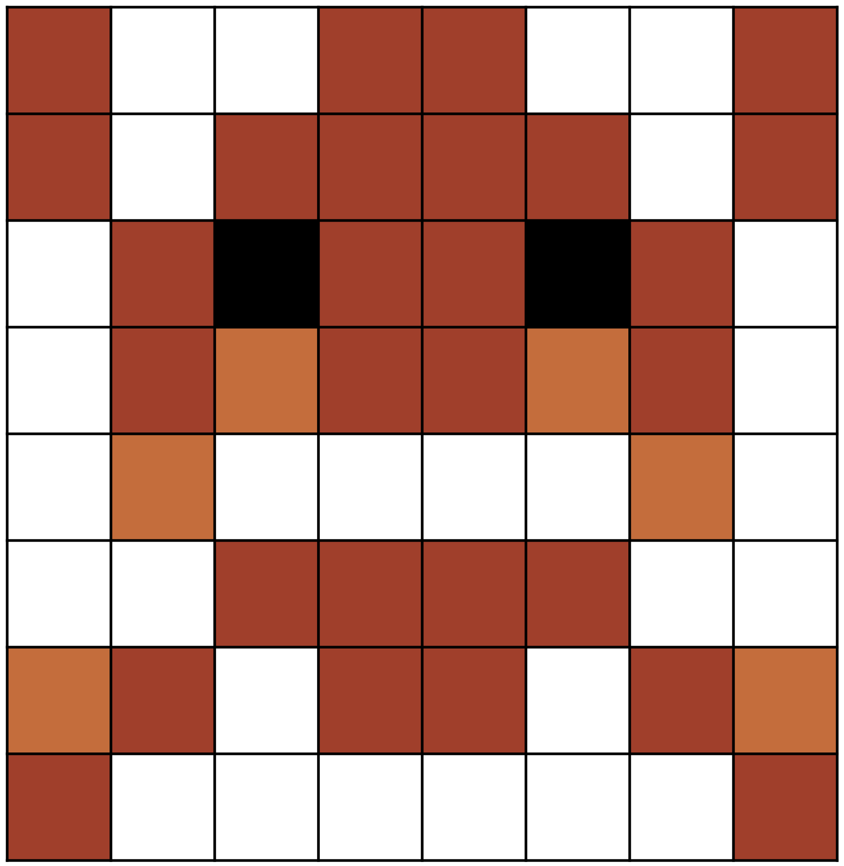
Parameterization



Specify the motion of every pixel?

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

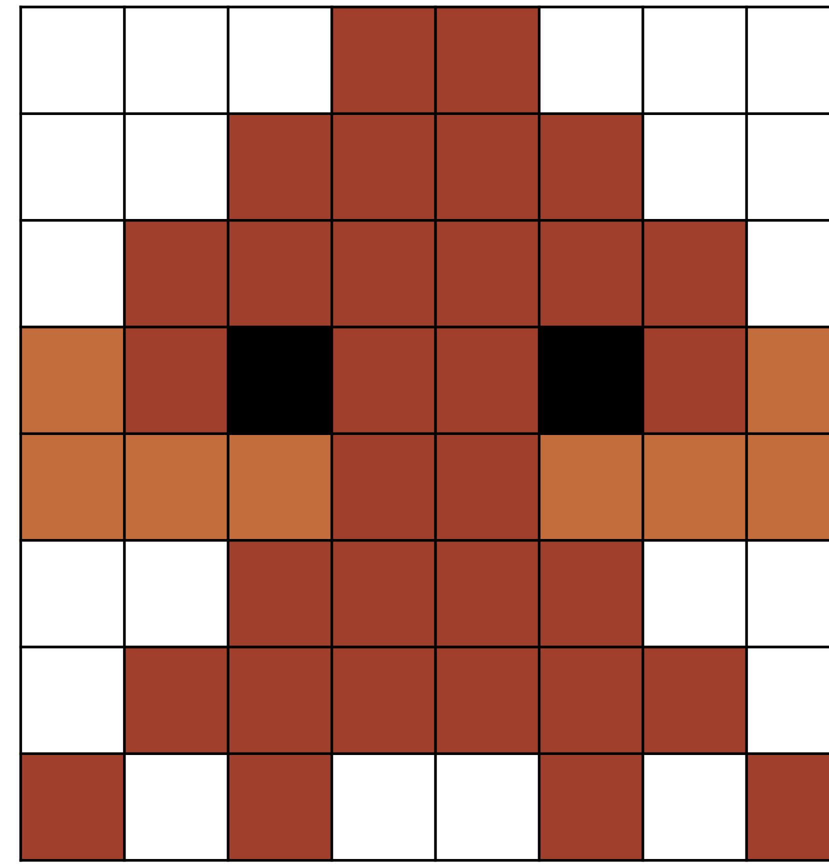
Example



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		

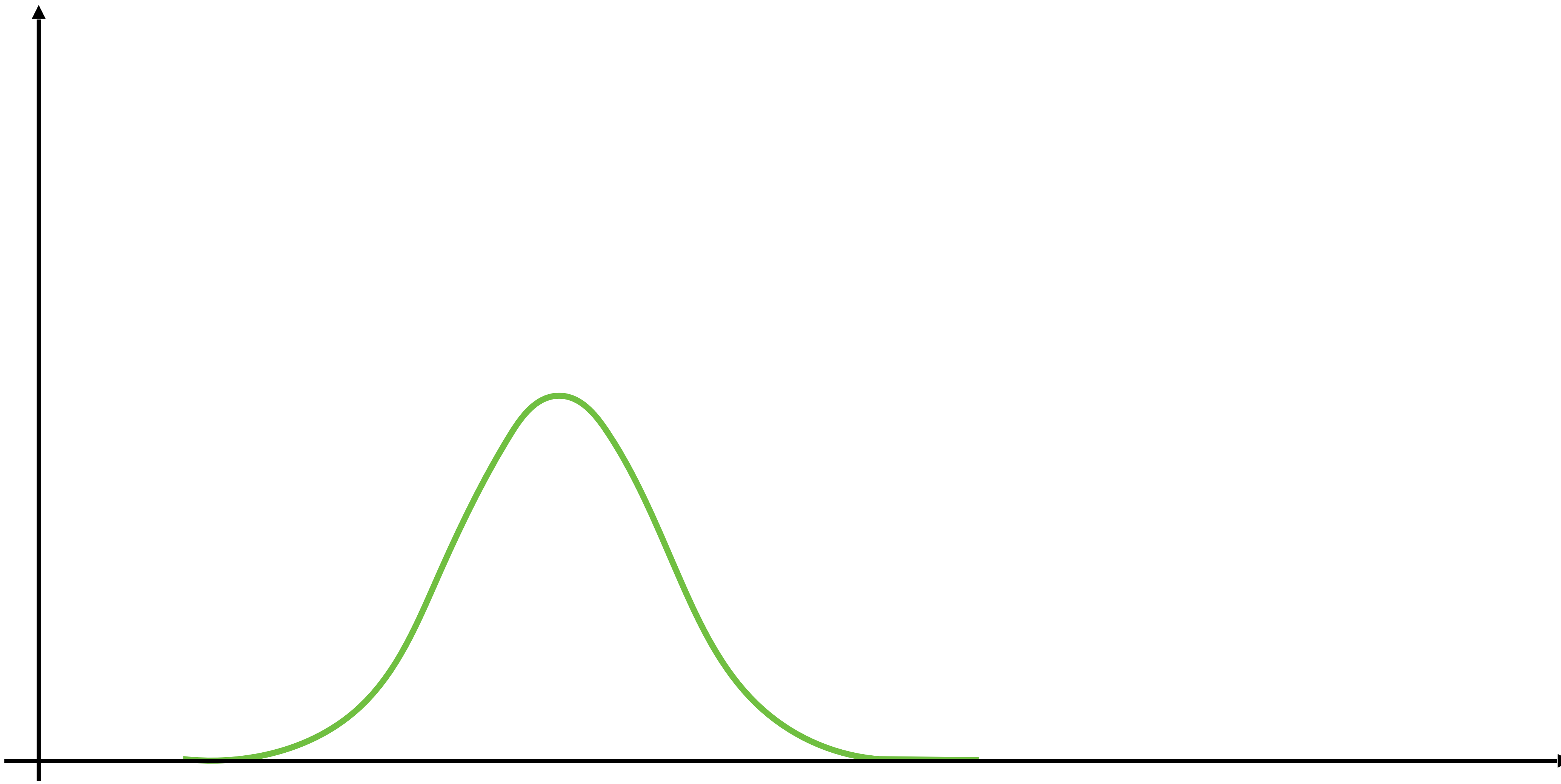
Δ_y



warped image

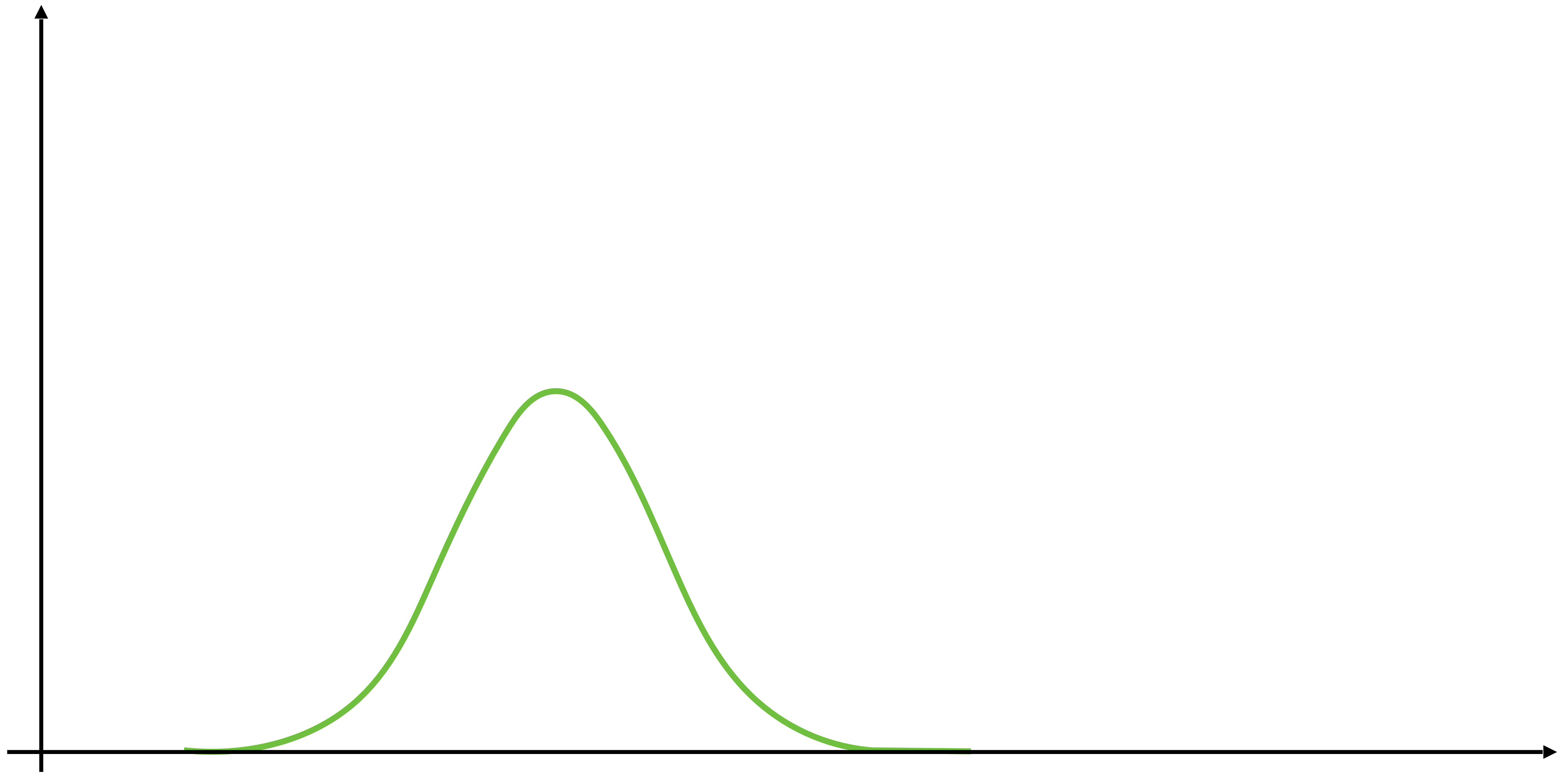
$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \Delta(x, y)$$

Use Basis Functions



Cubic B-spline

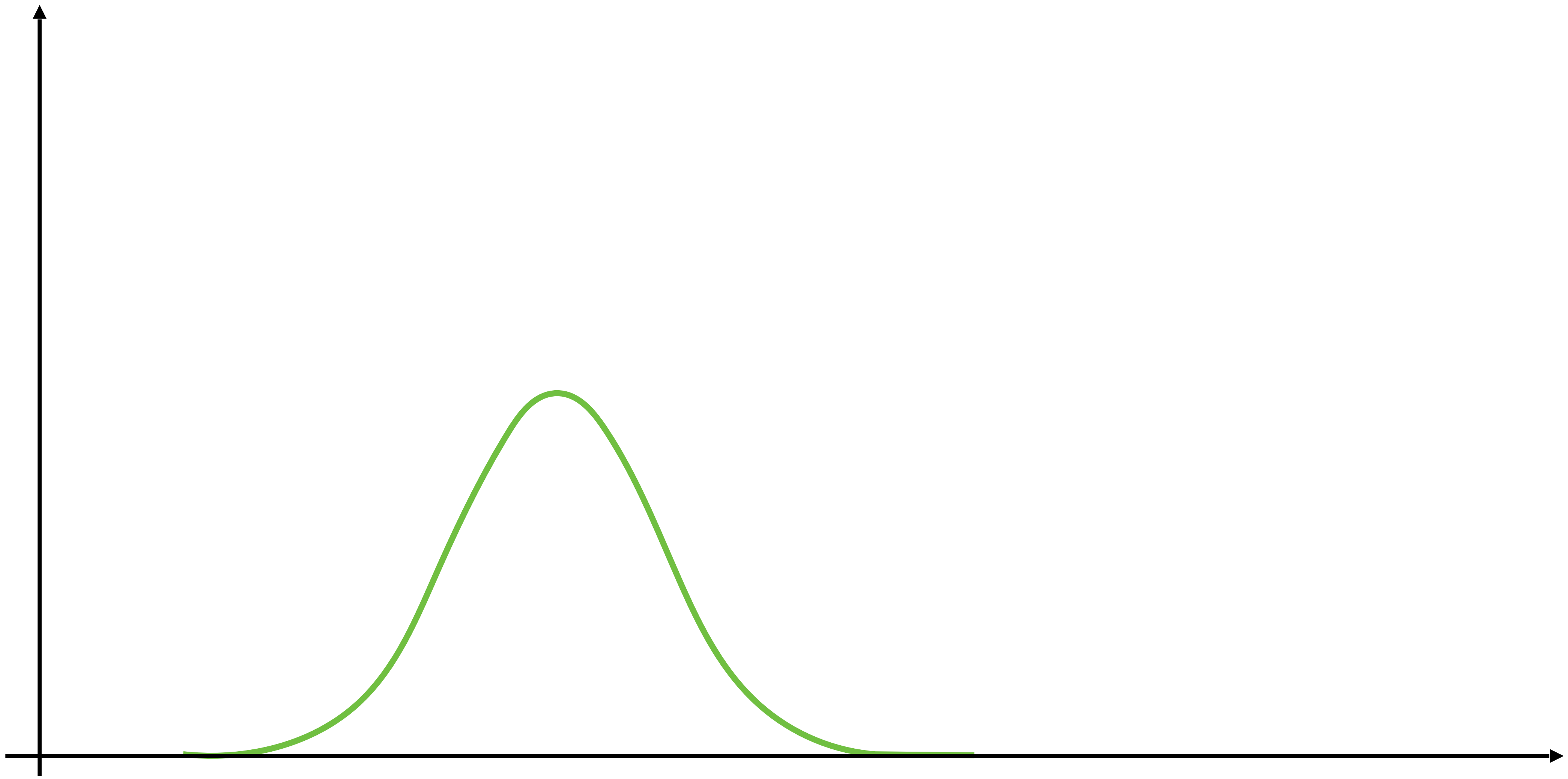
Use Basis Functions



Cubic B-spline

✓ Local

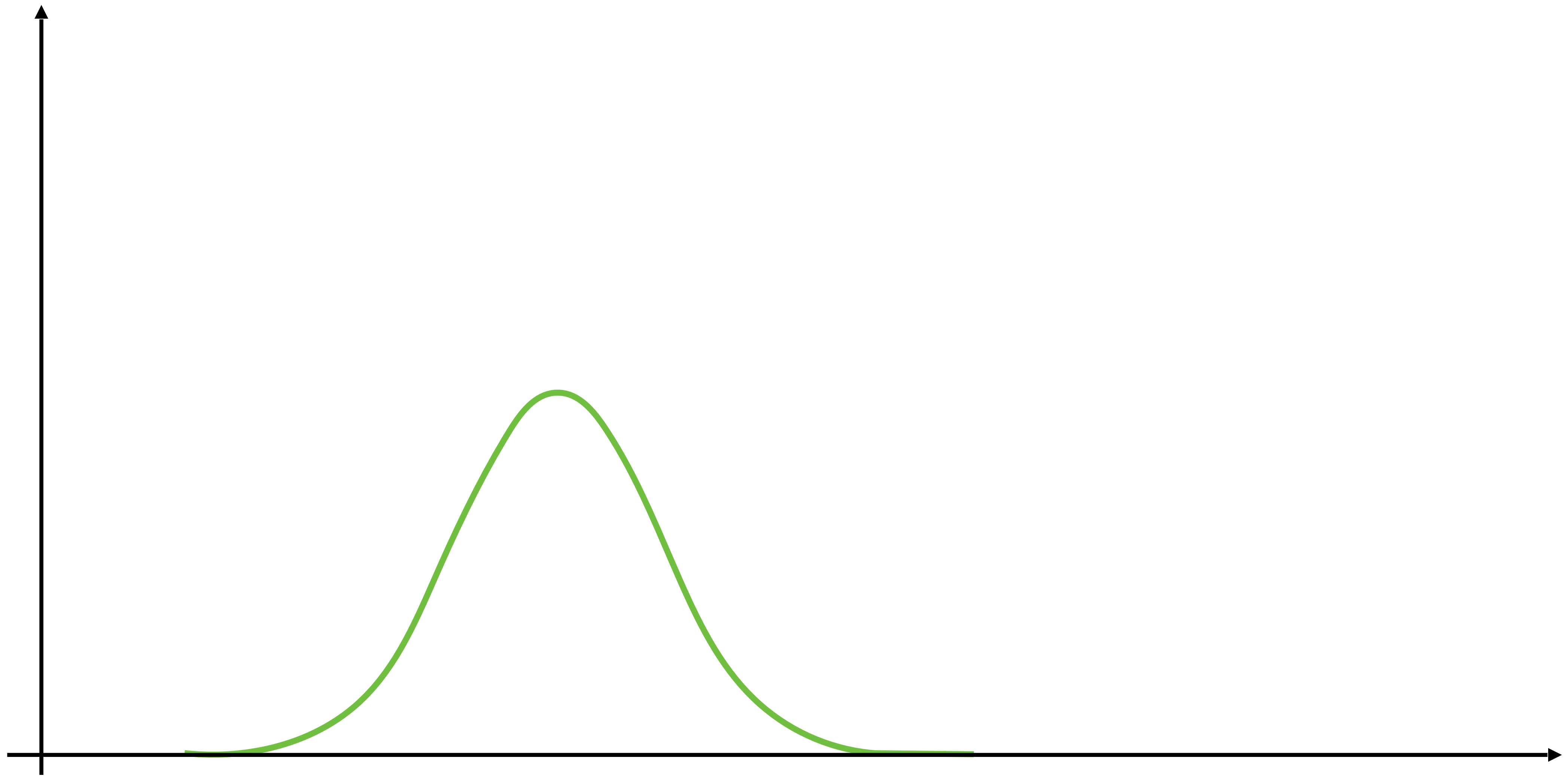
Use Basis Functions



Cubic B-spline

✓ Local ✓ Smooth

Use Basis Functions

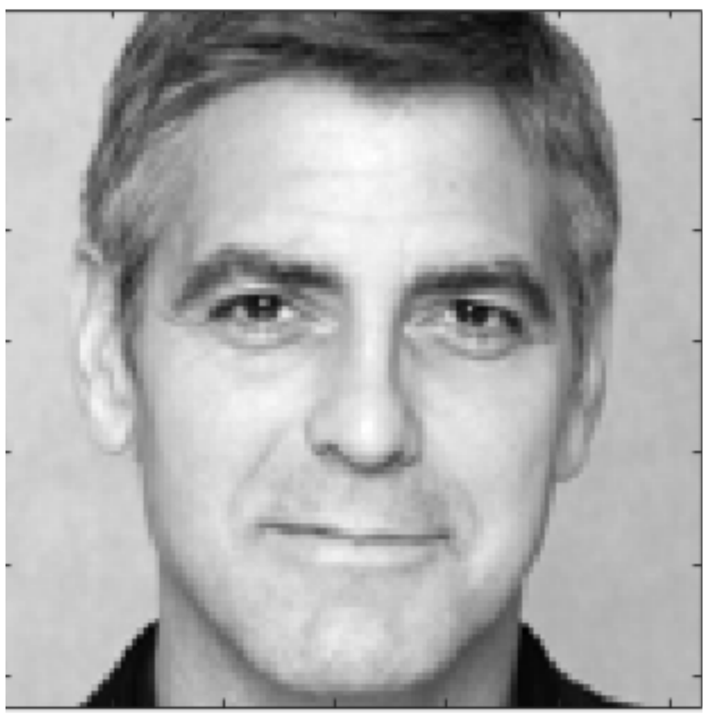


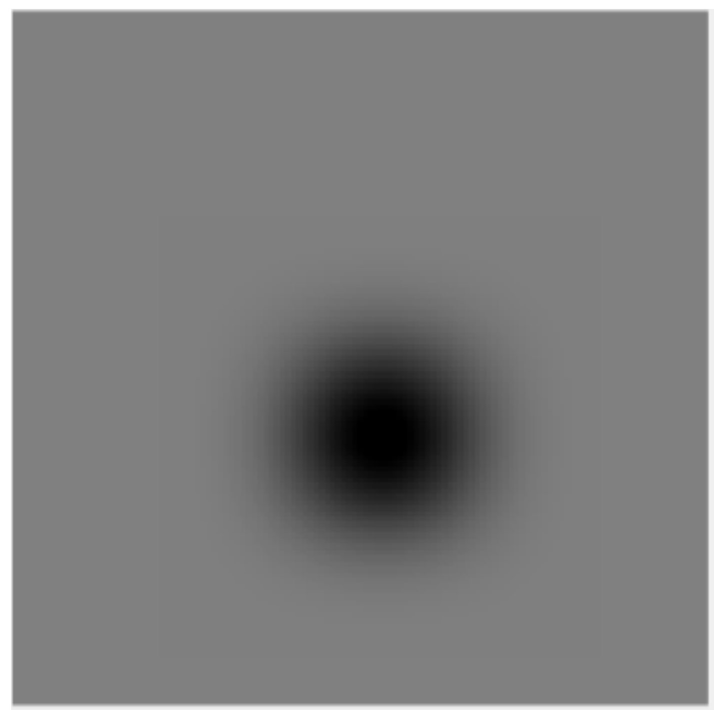
Cubic B-spline

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

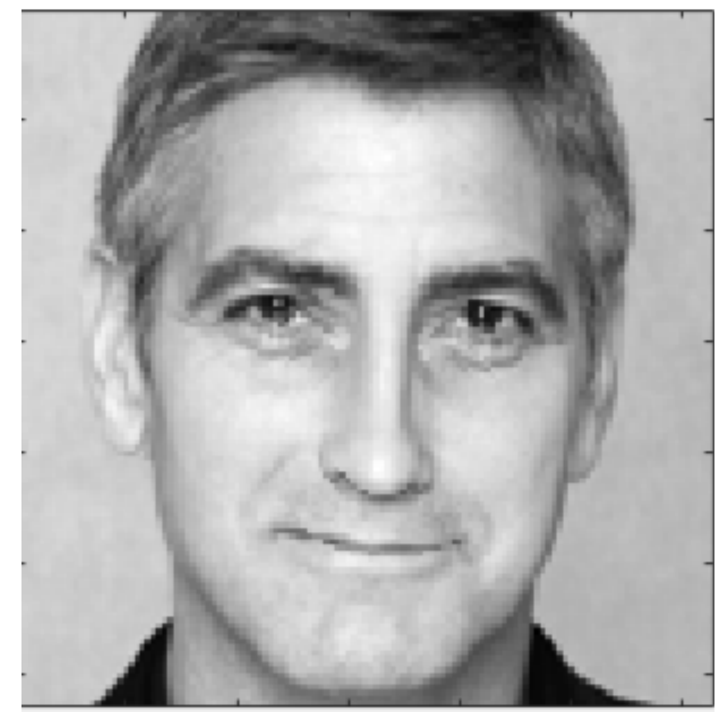
In 2D

Cubic B-Splines

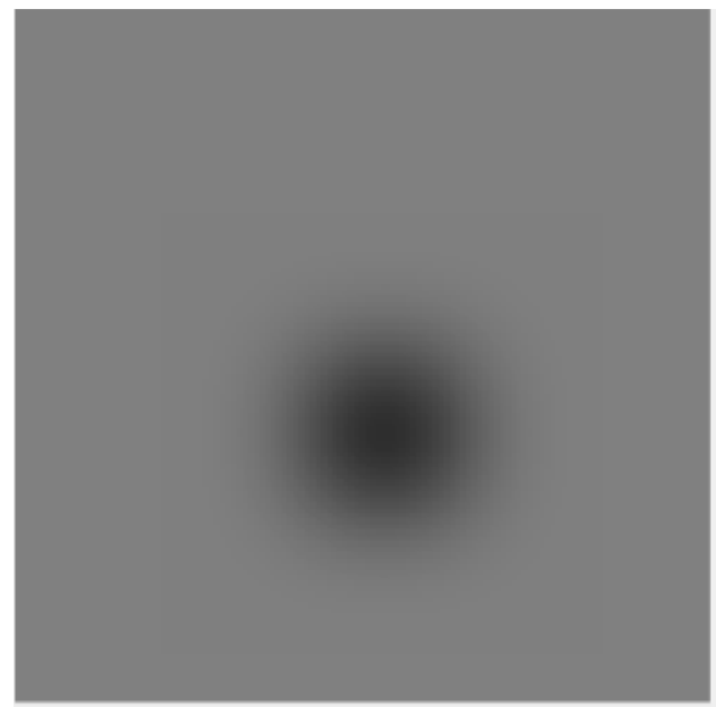




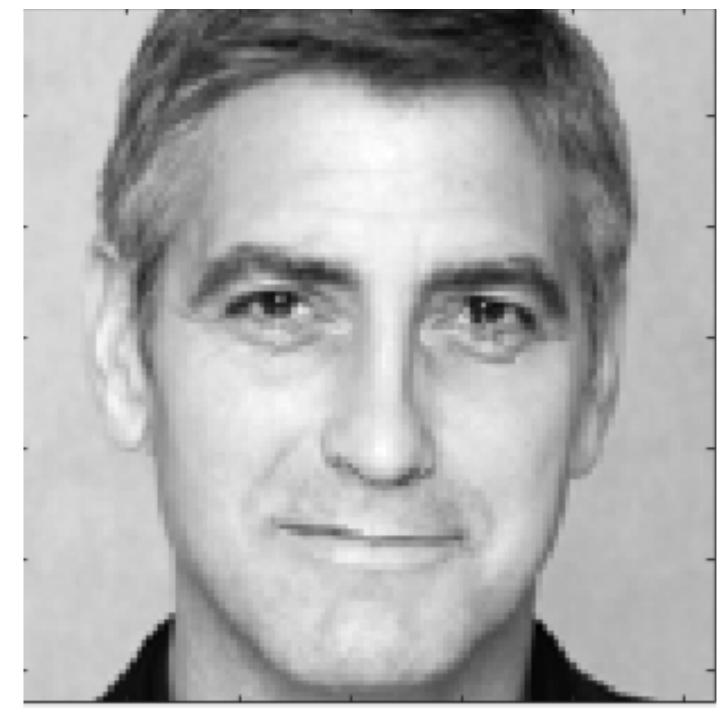
Δ_y



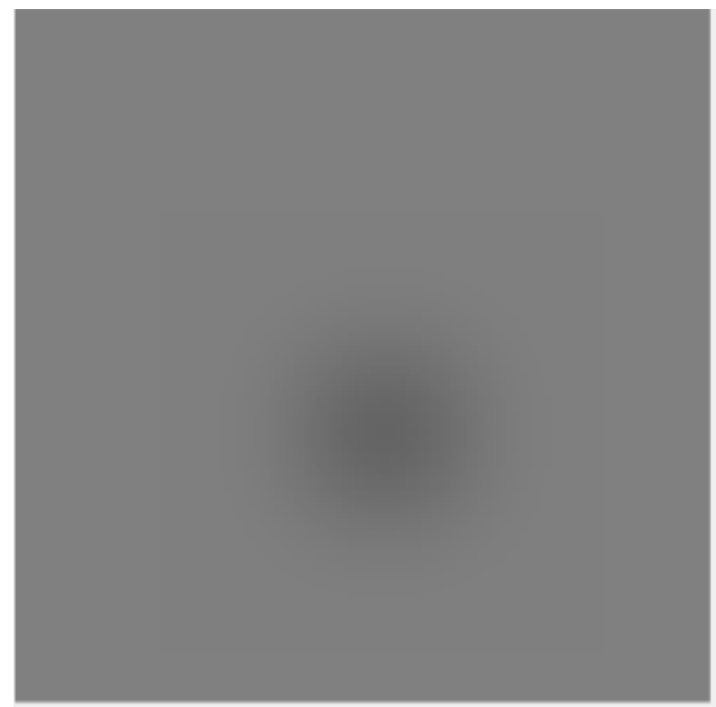
Cubic B-Splines



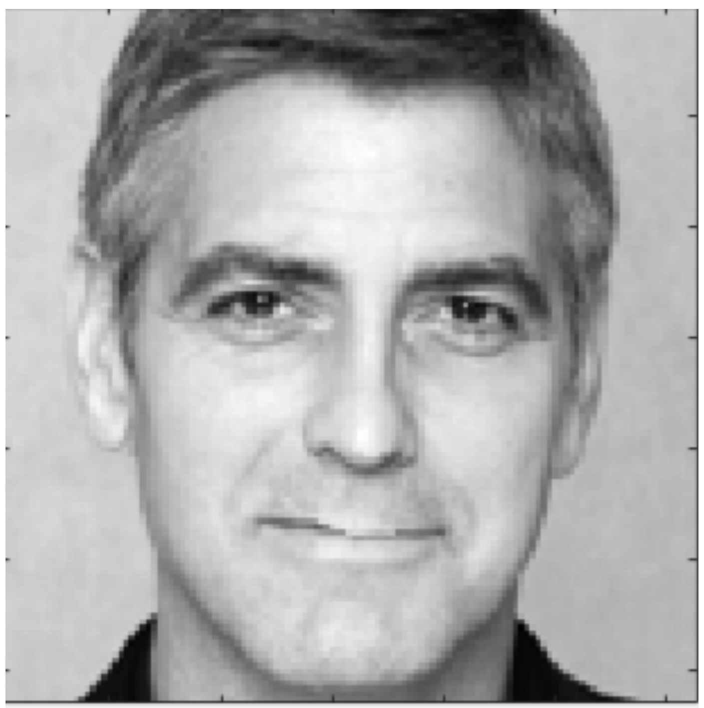
Δ_y



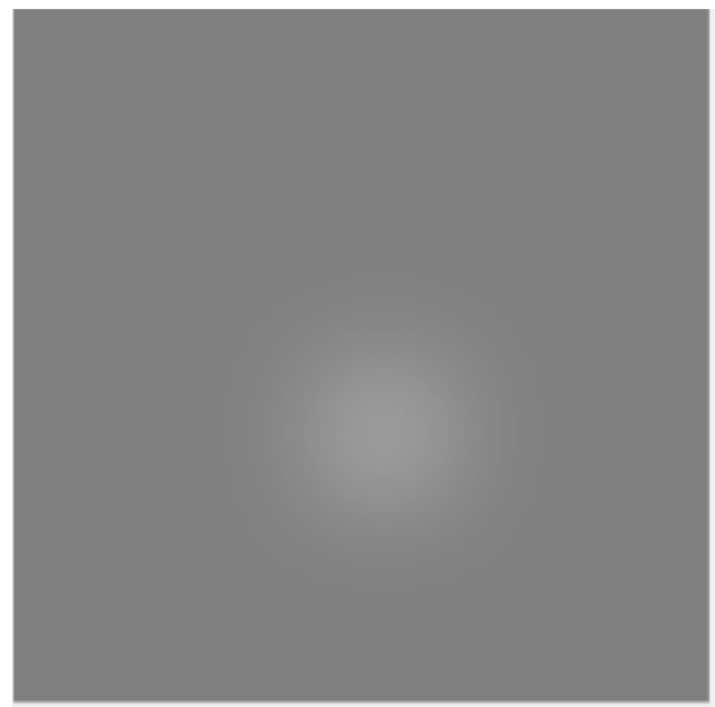
Cubic B-Splines



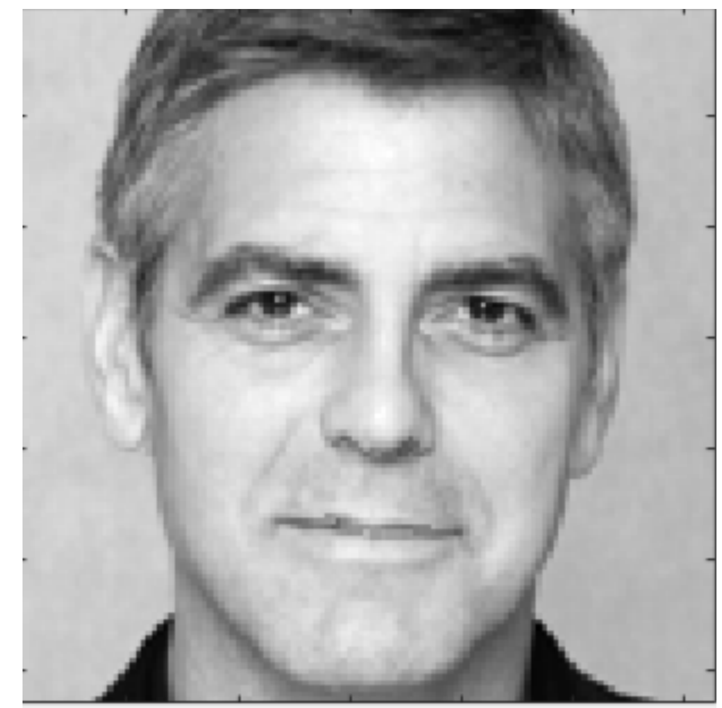
Δ_y



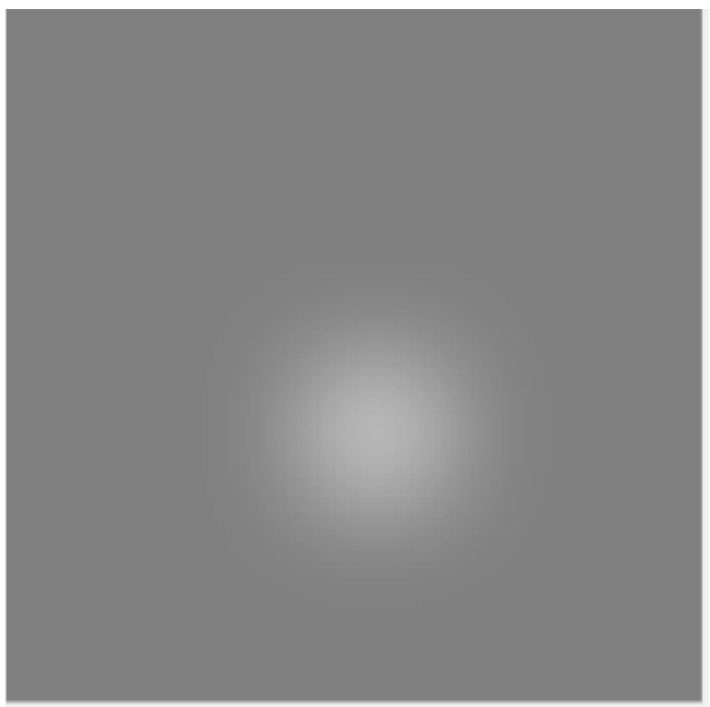
Cubic B-Splines



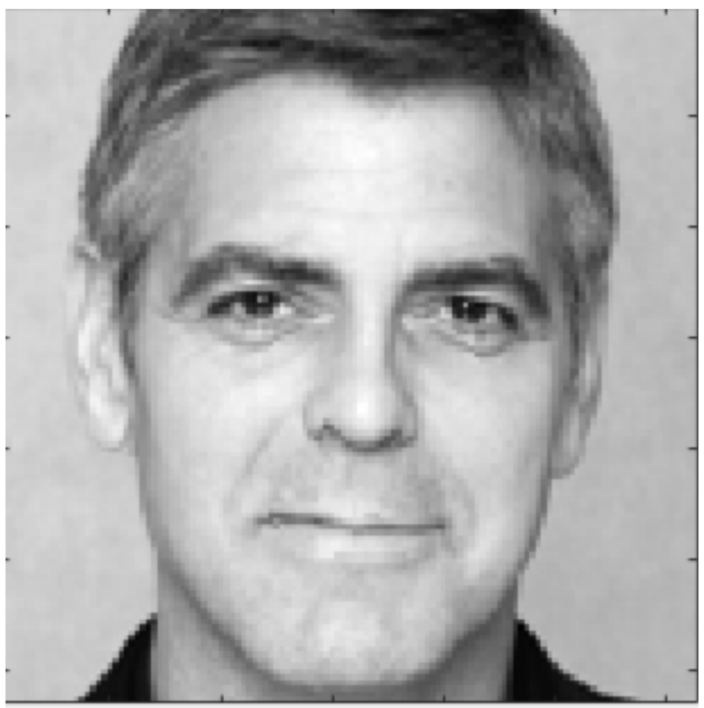
Δ_y



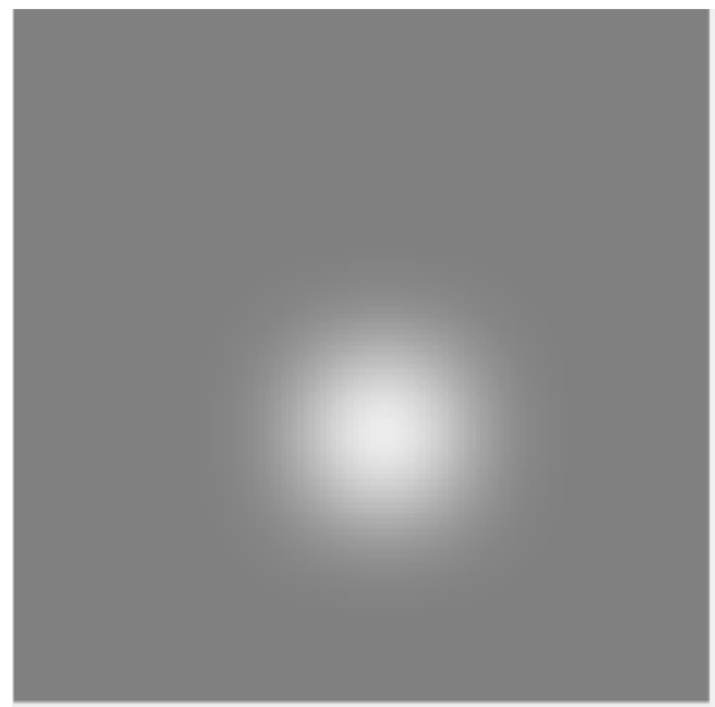
Cubic B-Splines



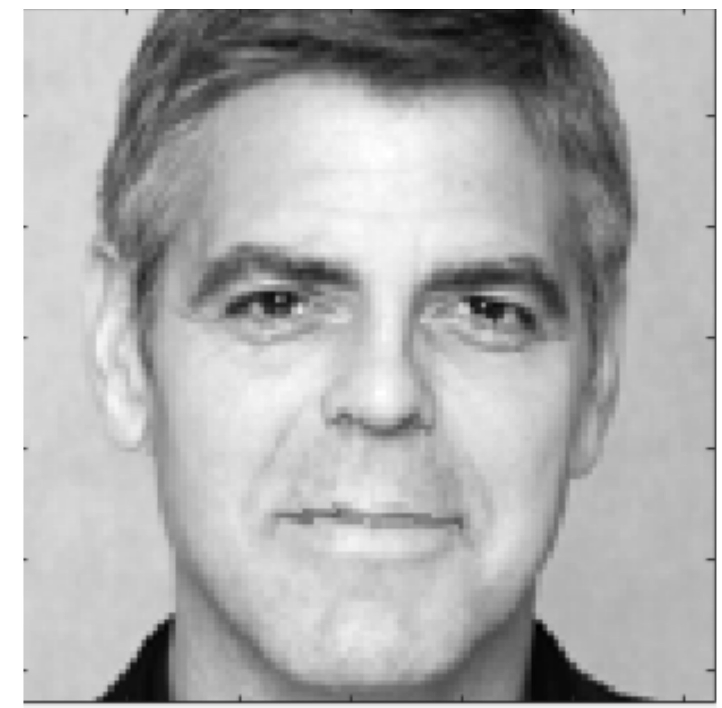
Δ_y



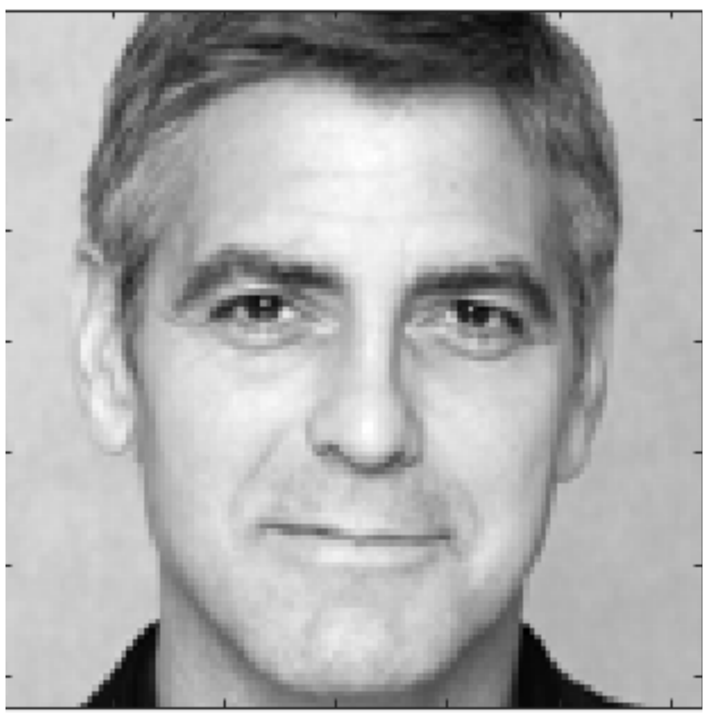
Cubic B-Splines

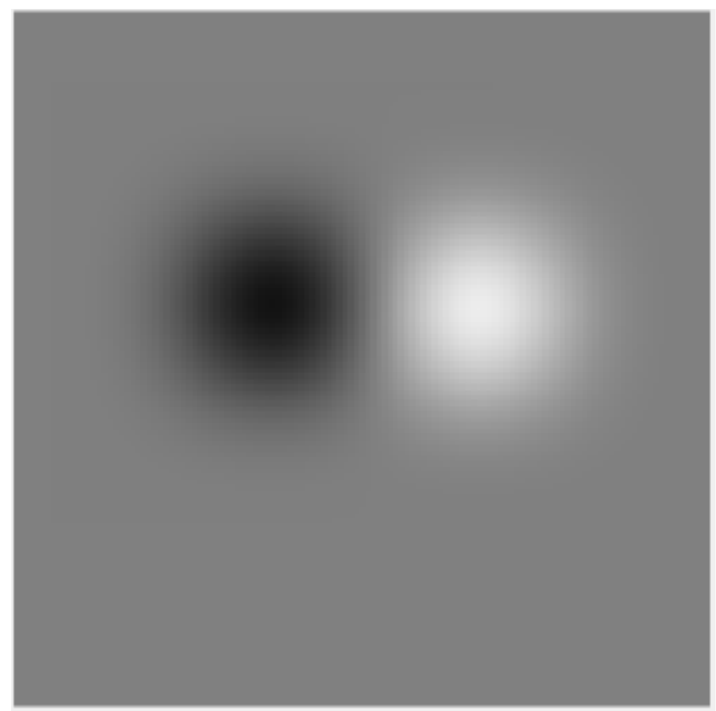


Δ_y

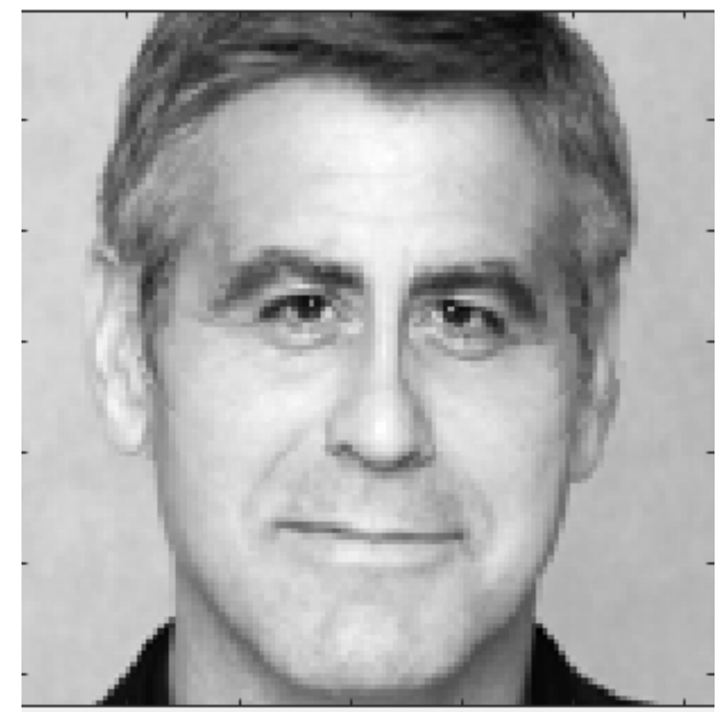


Cubic B-Splines

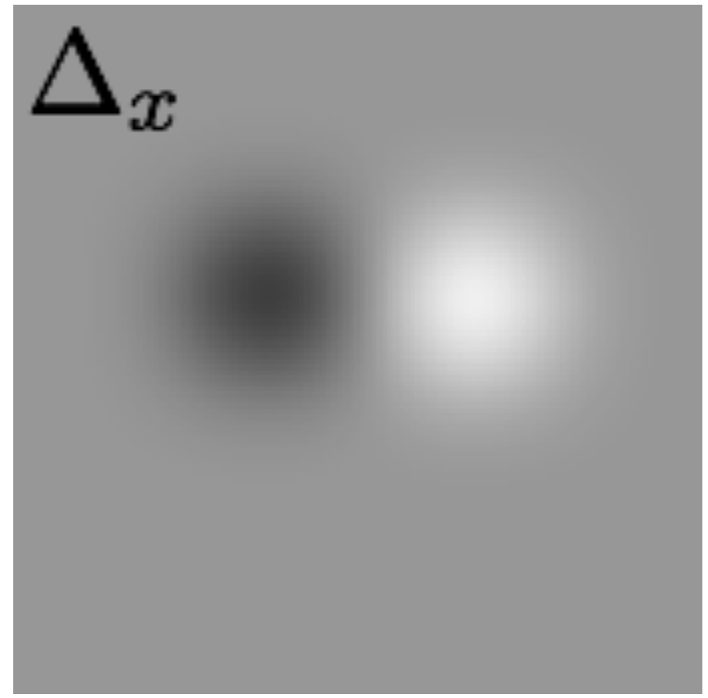


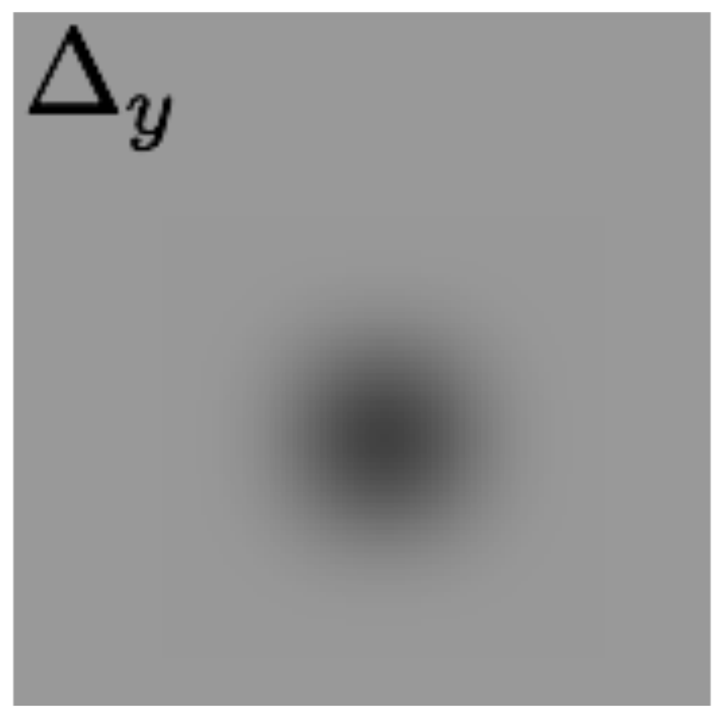


Δ_x

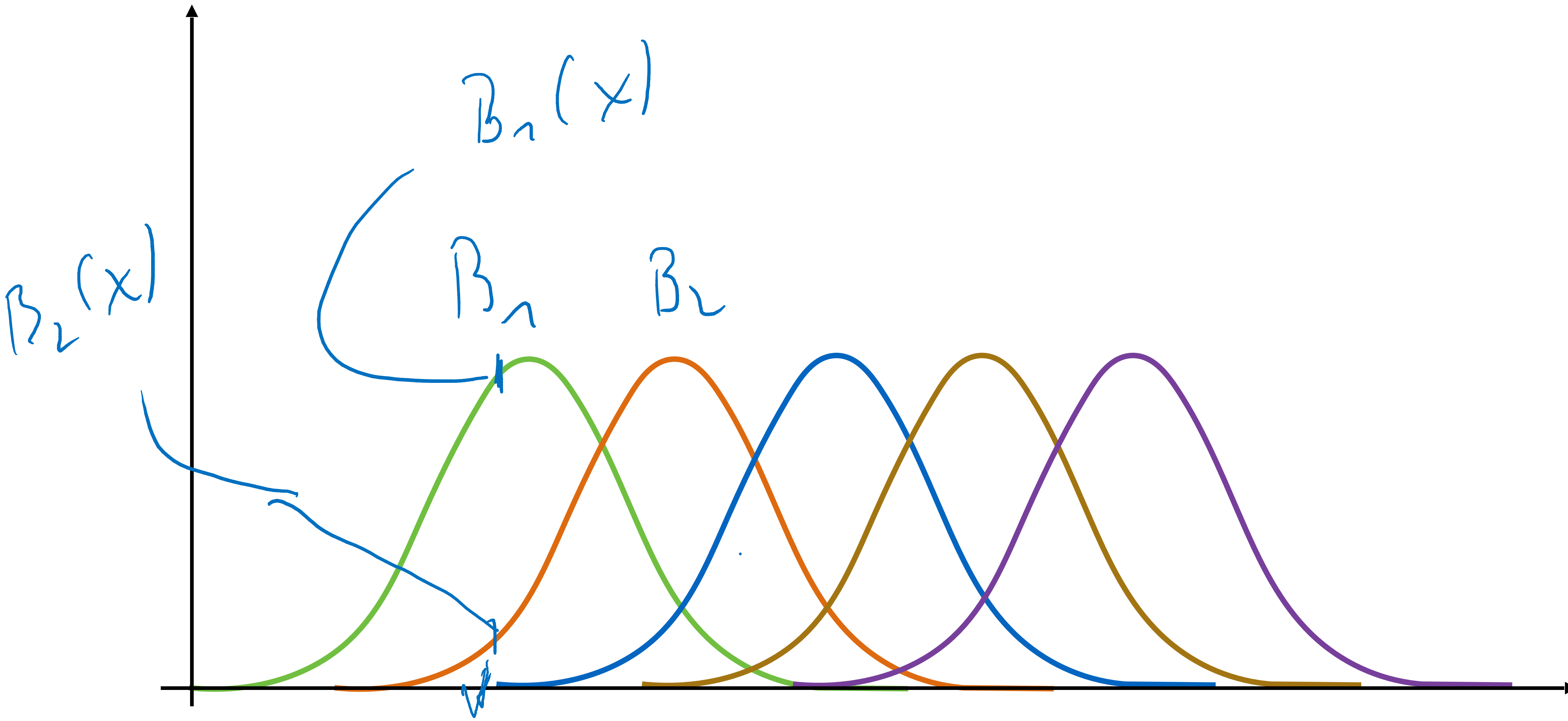


Cubic B-Splines



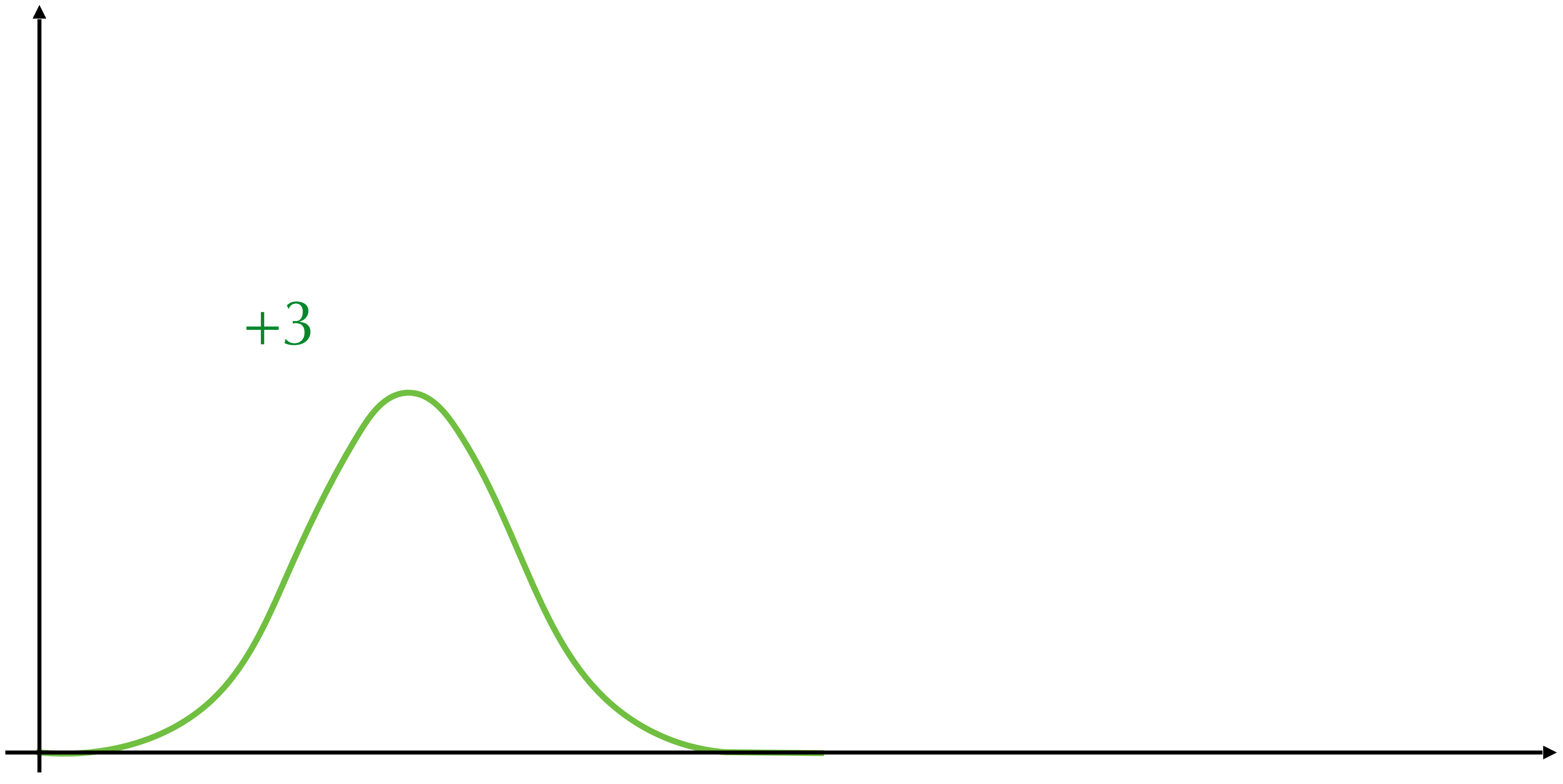


Use Basis Functions



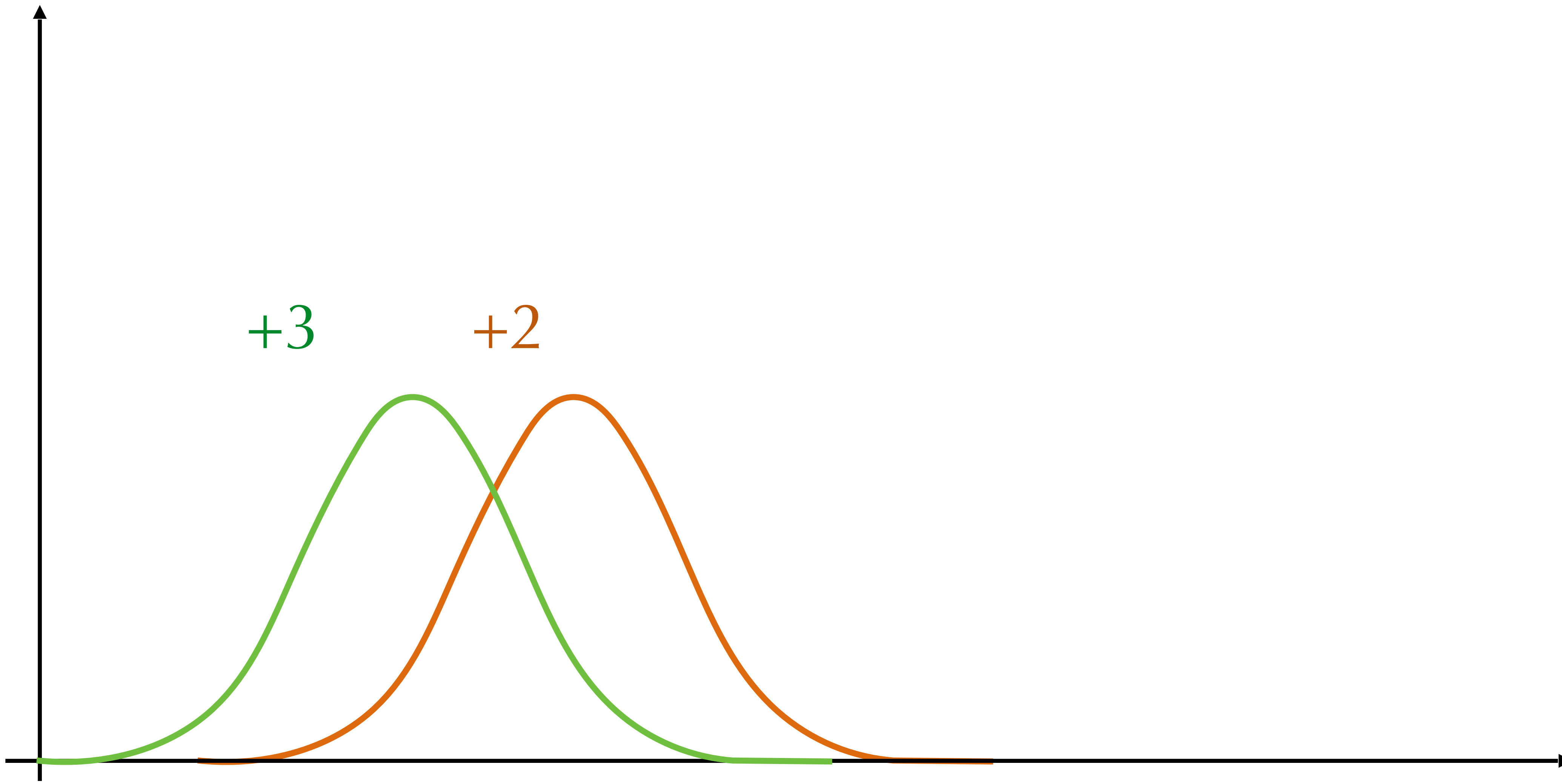
$$\Delta_x(x,y) = \sum_{k=1}^{n} \alpha_k B_k(x,y) \qquad \Delta_y(x,y) = \sum_{k=1}^{n} \beta_k B_k(x,y)$$

Use Basis Functions



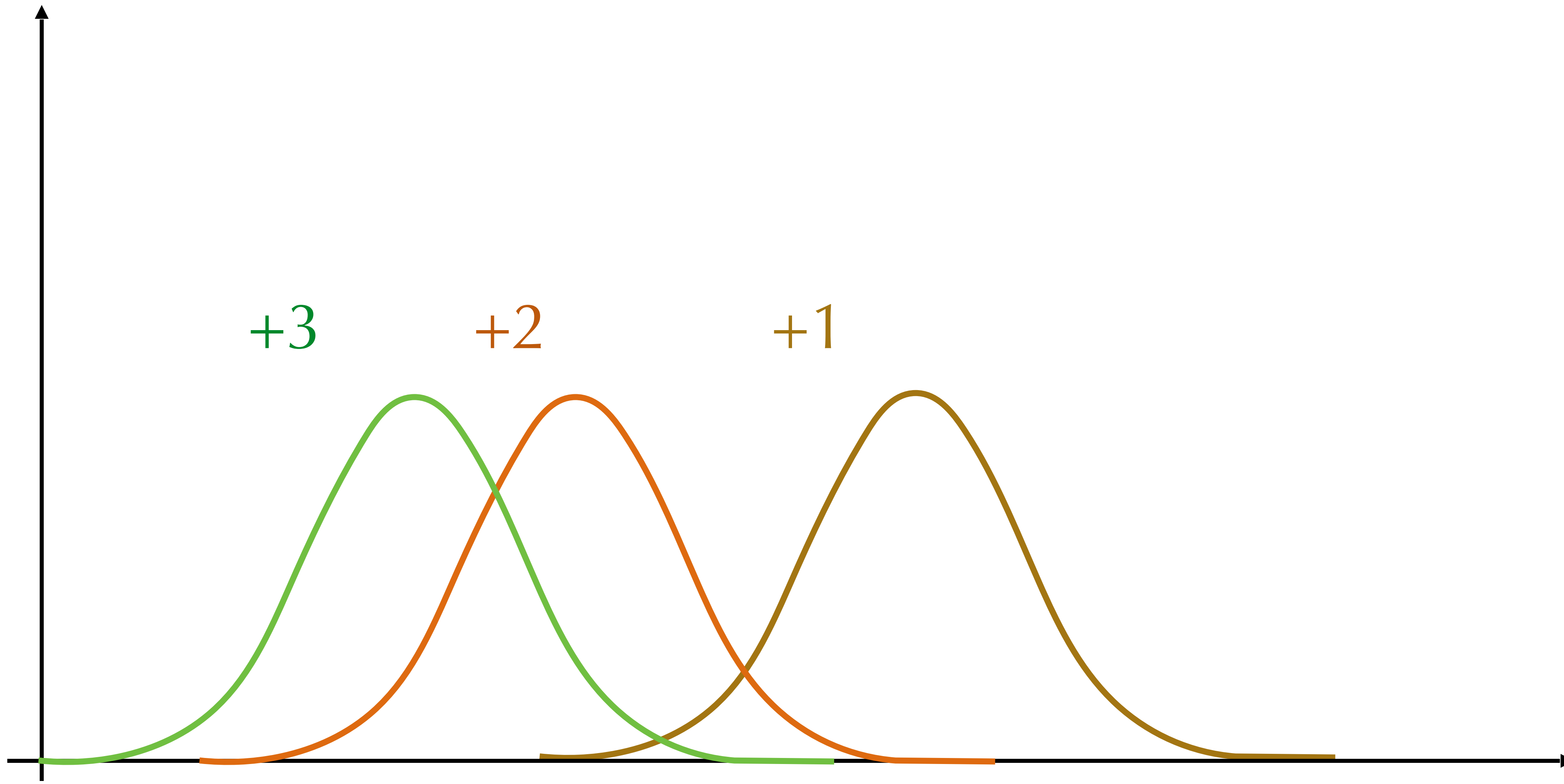
$$\Delta_x(x, y) = \sum_{k=1}^{n} \alpha_k B_k(x, y)$$

Use Basis Functions



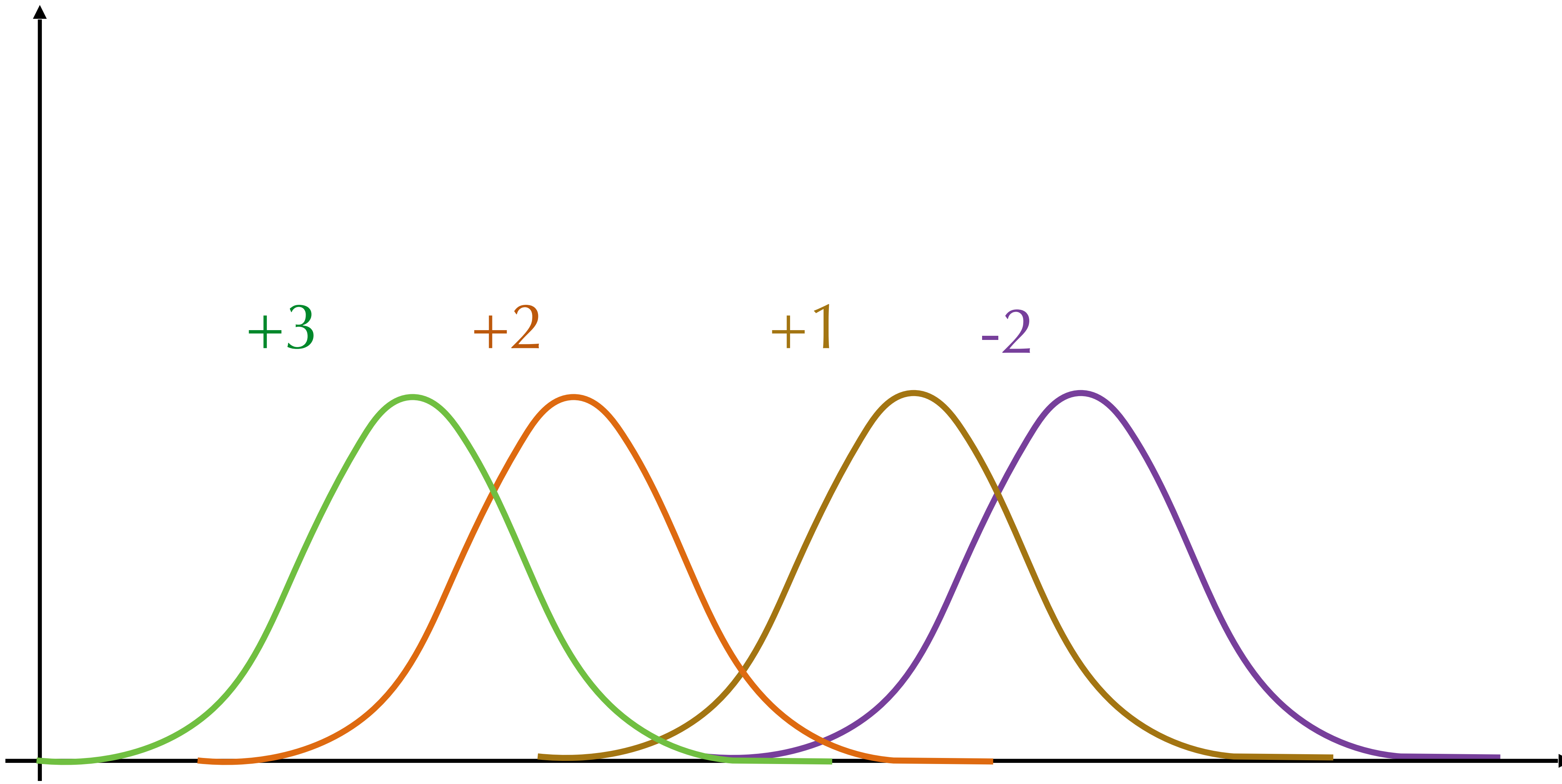
$$\Delta_x(x, y) = \sum_{k=1}^{n} \alpha_k B_k(x, y)$$

Use Basis Functions



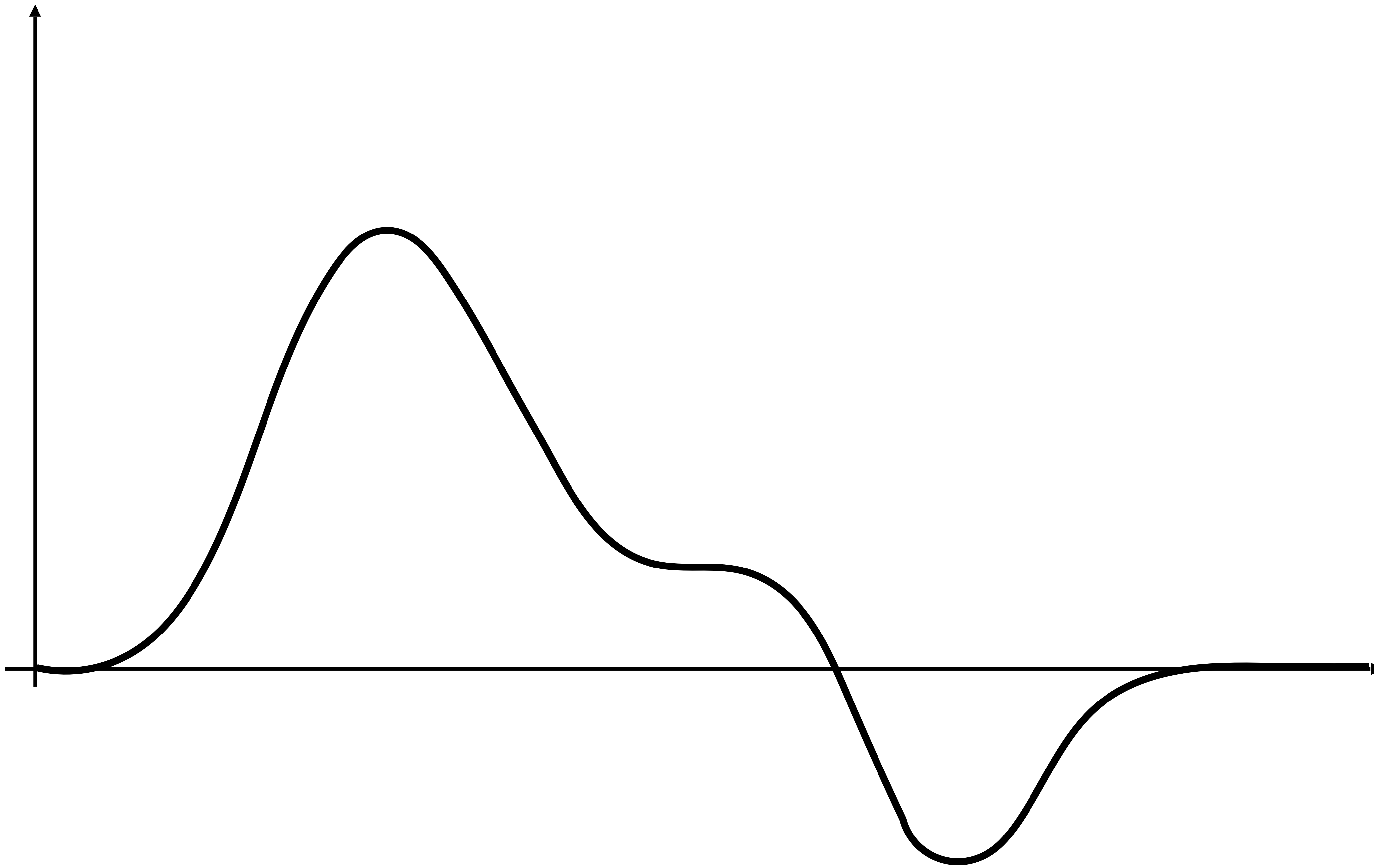
$$\Delta_x(x, y) = \sum_{k=1}^{n} \alpha_k B_k(x, y)$$

Use Basis Functions



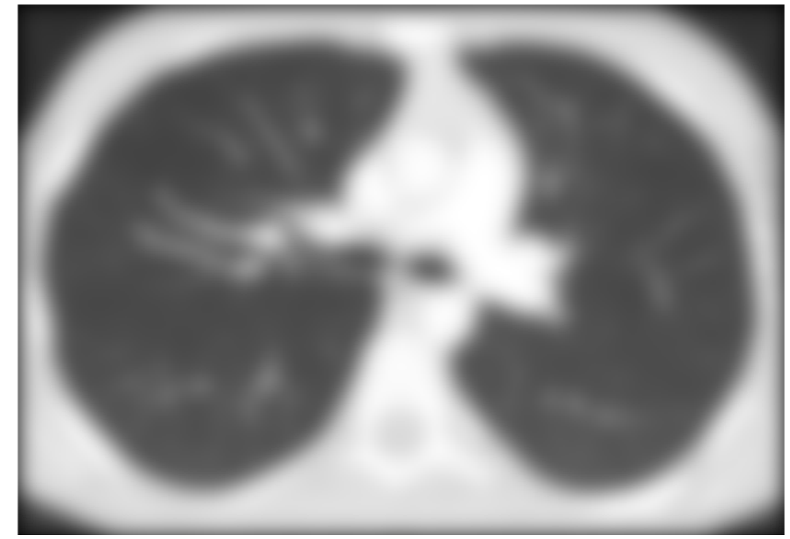
$$\Delta_x(x, y) = \sum_{k=1}^{n} \alpha_k B_k(x, y)$$

Use Basis Functions



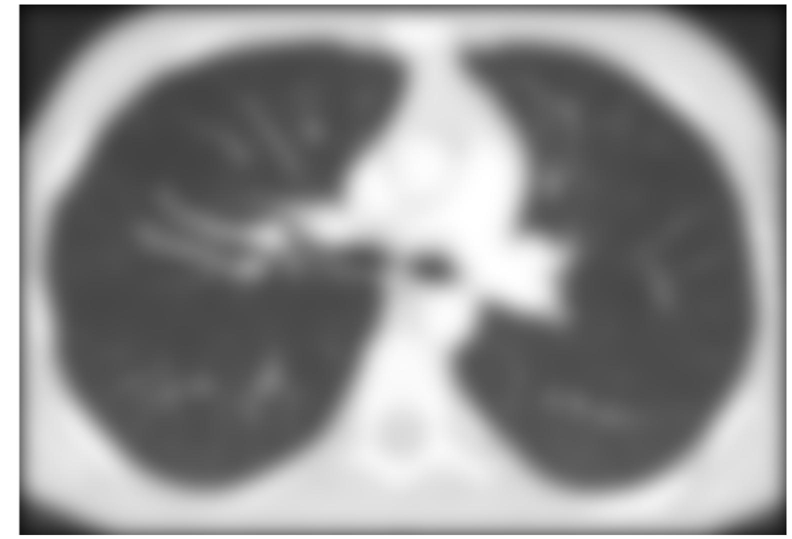
$$\Delta_x(x, y) = \sum_{k=1}^{n} \alpha_k B_k(x, y)$$

Coarse-to-Fine Alignment

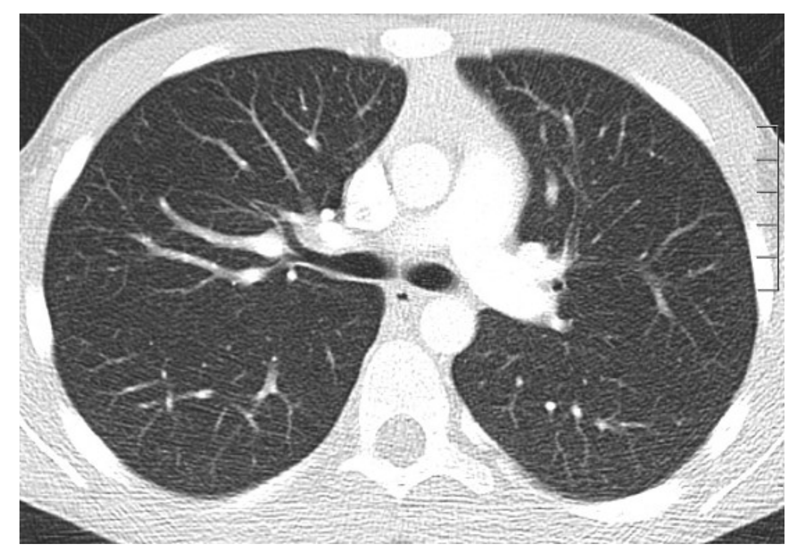


- Few basis function at coarse scale

Coarse-to-Fine Alignment

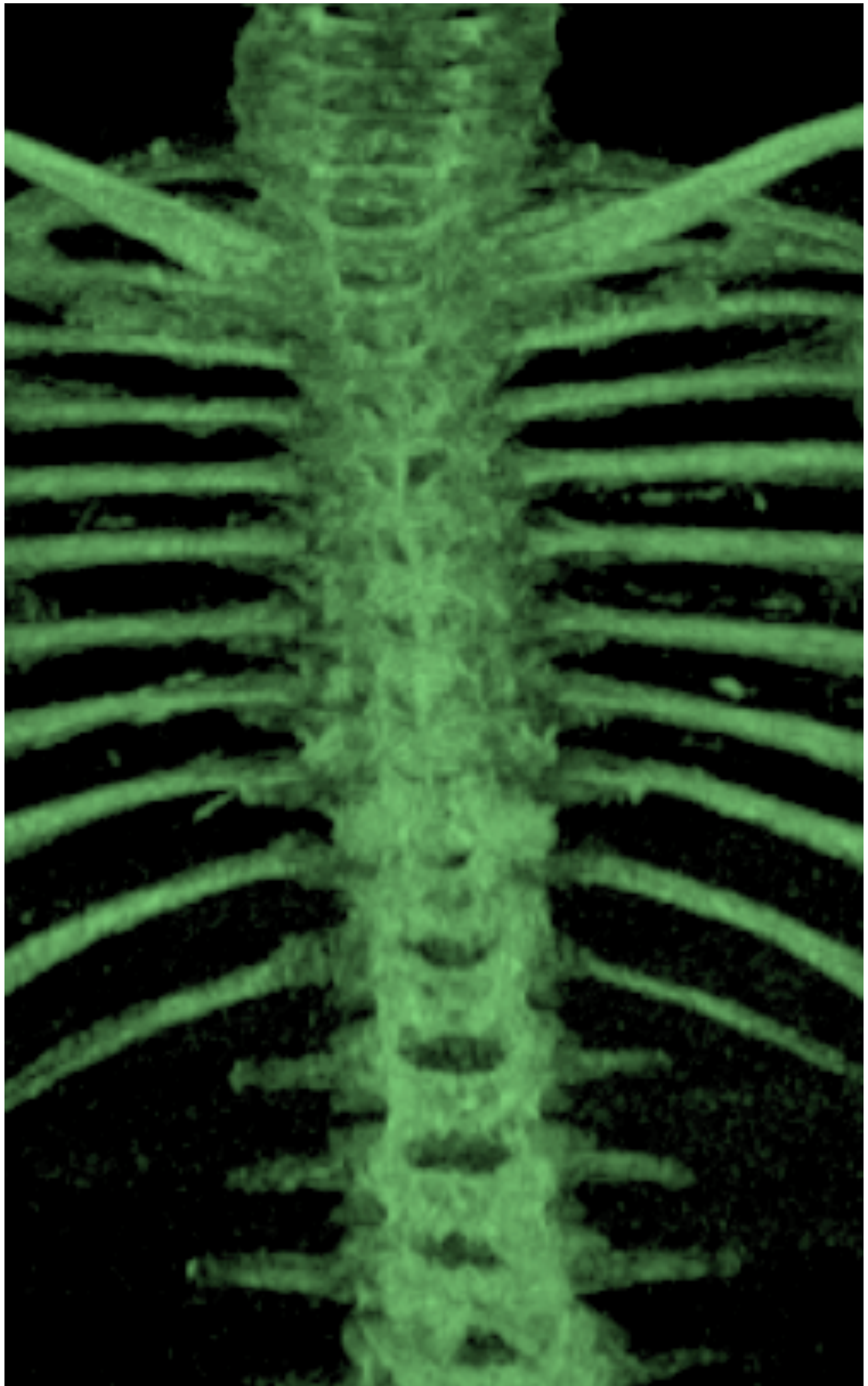


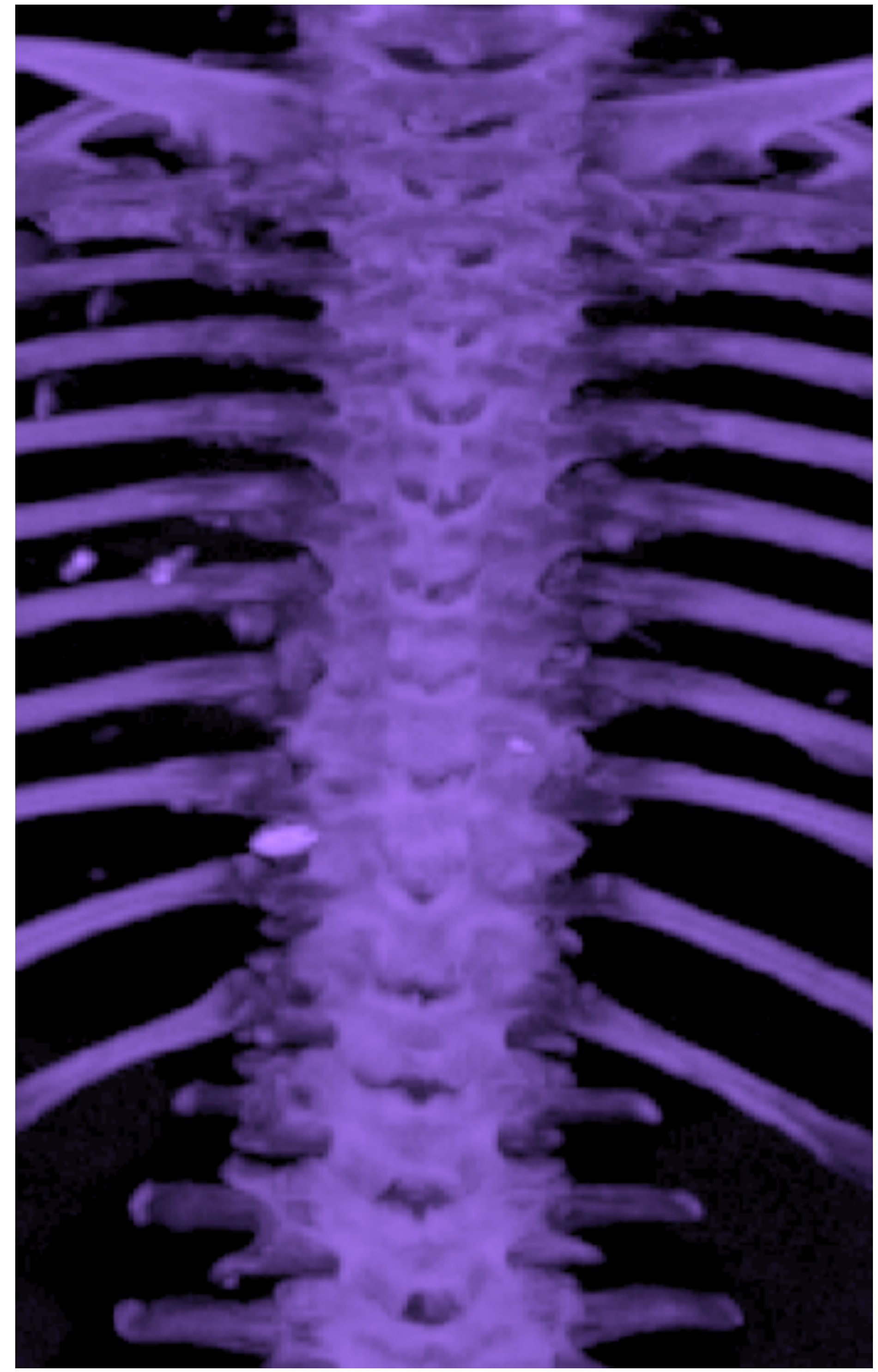
- Few basis function at coarse scale



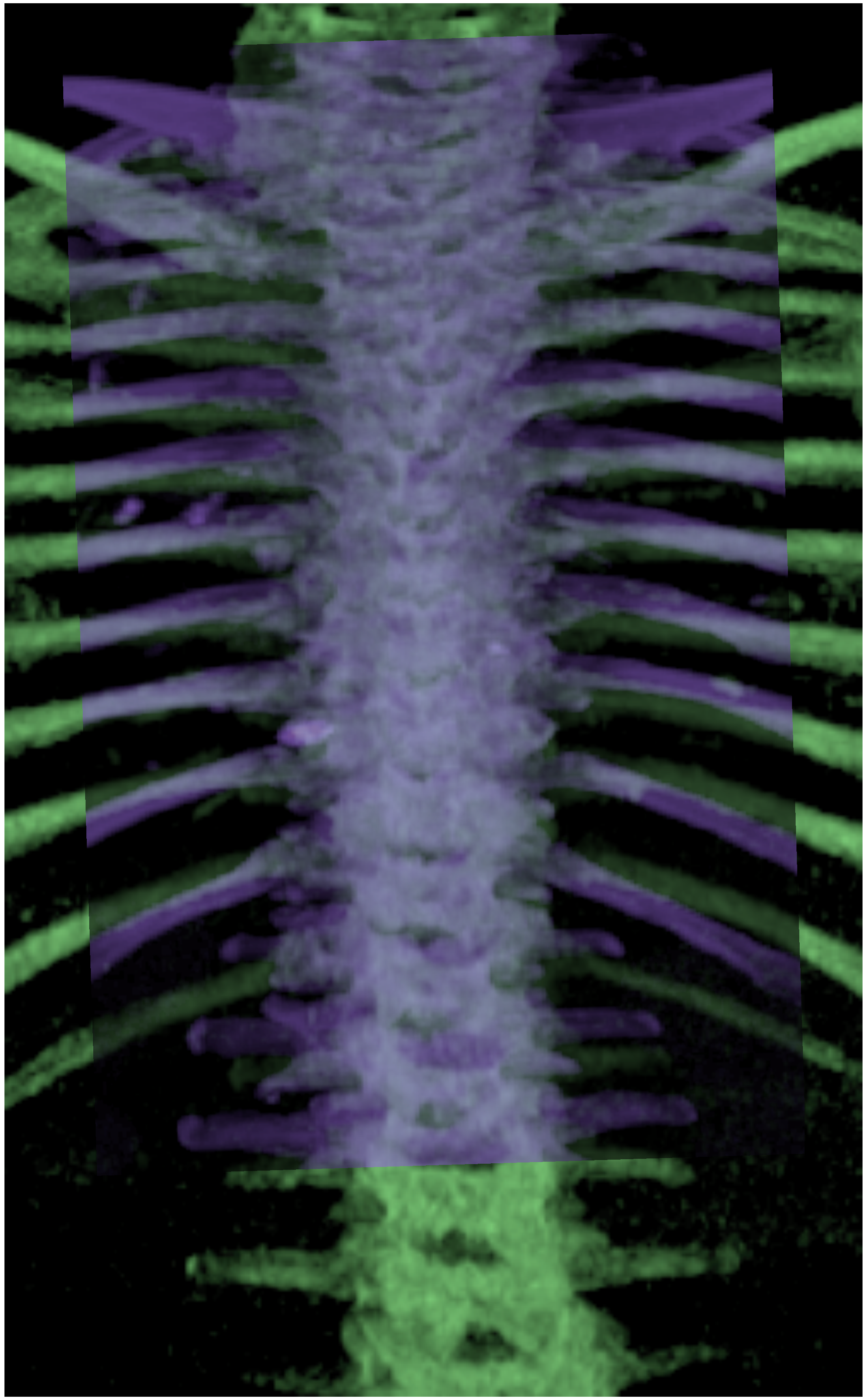
- Many basis functions at fine scale

Local Minima

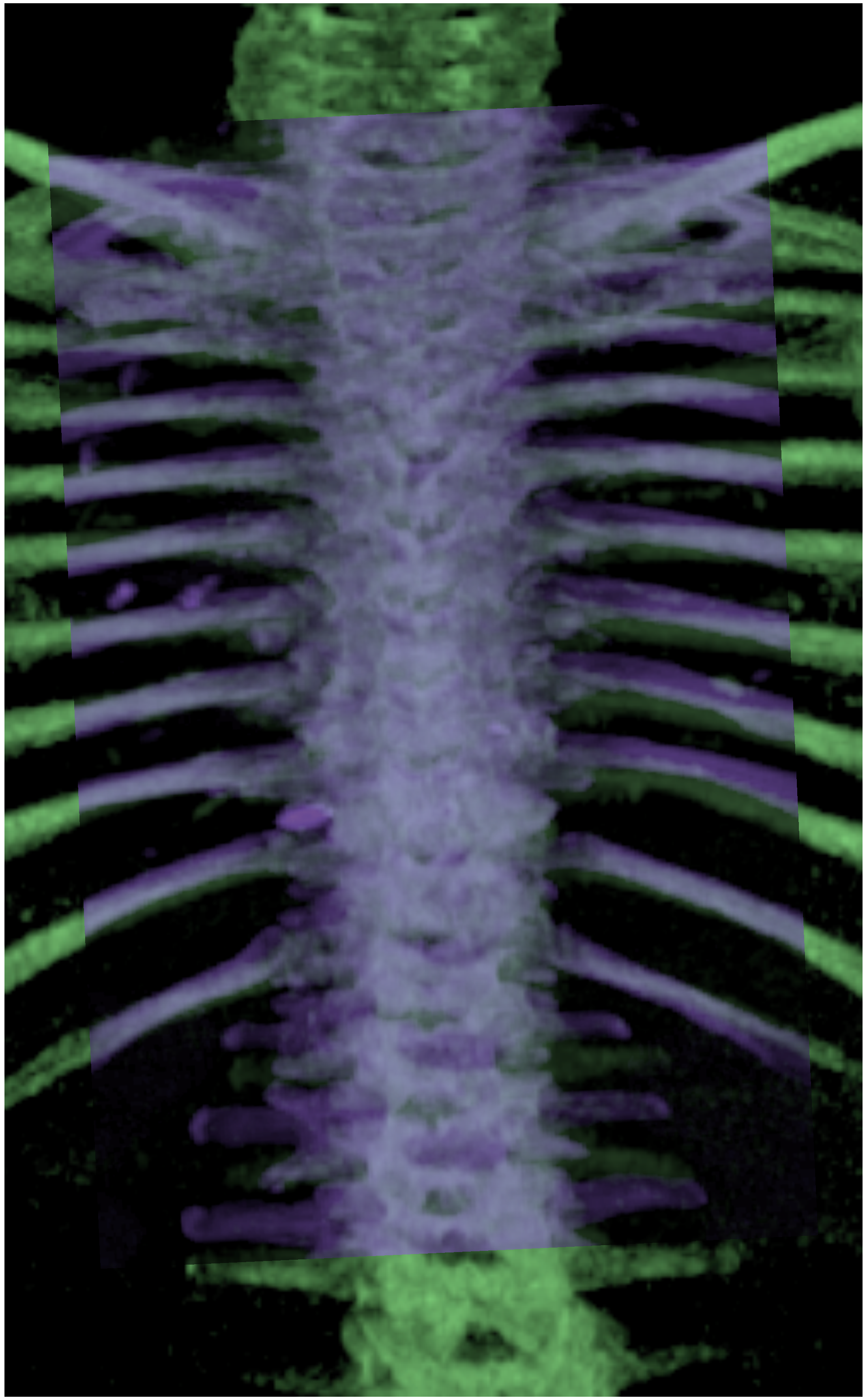




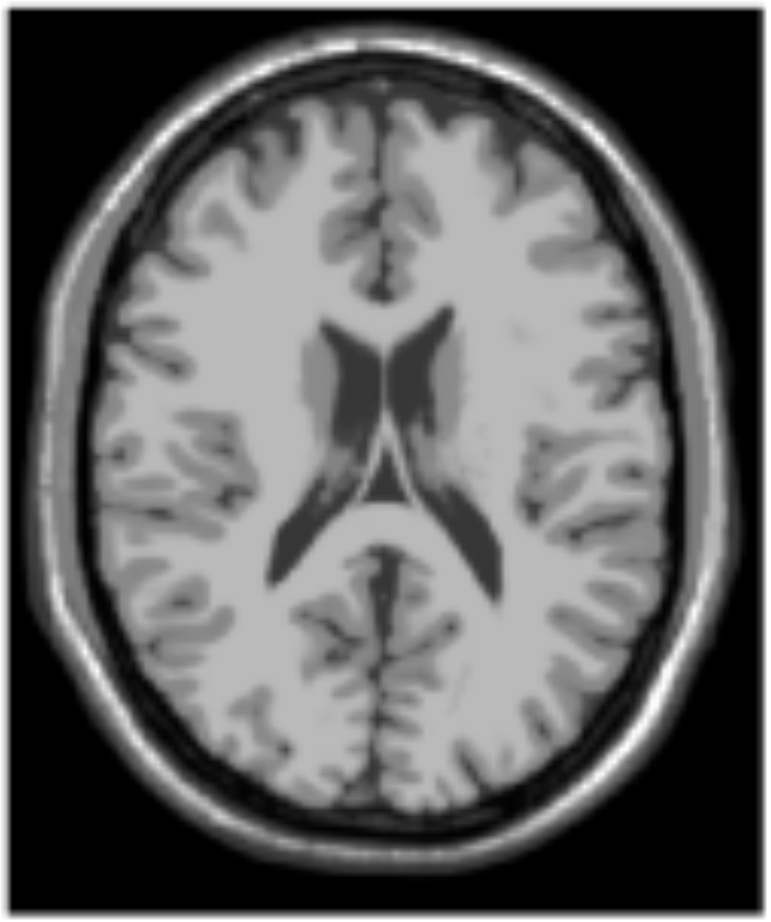
Local Minima



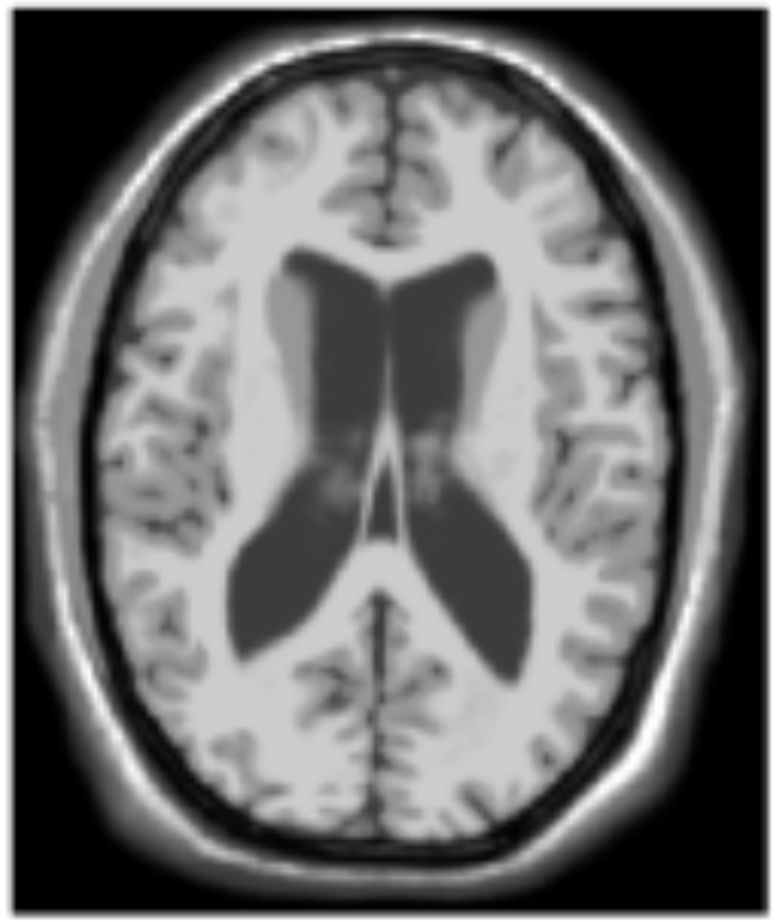
Local minima



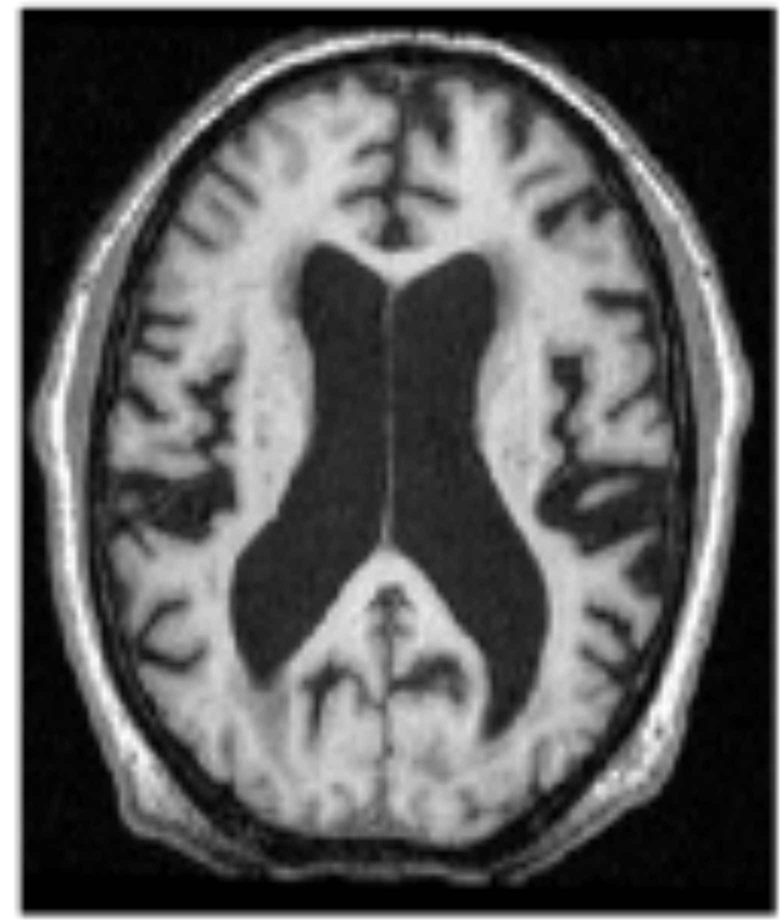
Example



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

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