Geometric Transformations, RANSAC and Morphing

CS448V — Computational Video Manipulation

April 2019

In previous lecture

In previous lecture

Feature detection

In previous lecture

Feature detection

Feature description

In previous lecture

Feature detection

Feature description

Feature matching

In previous lecture

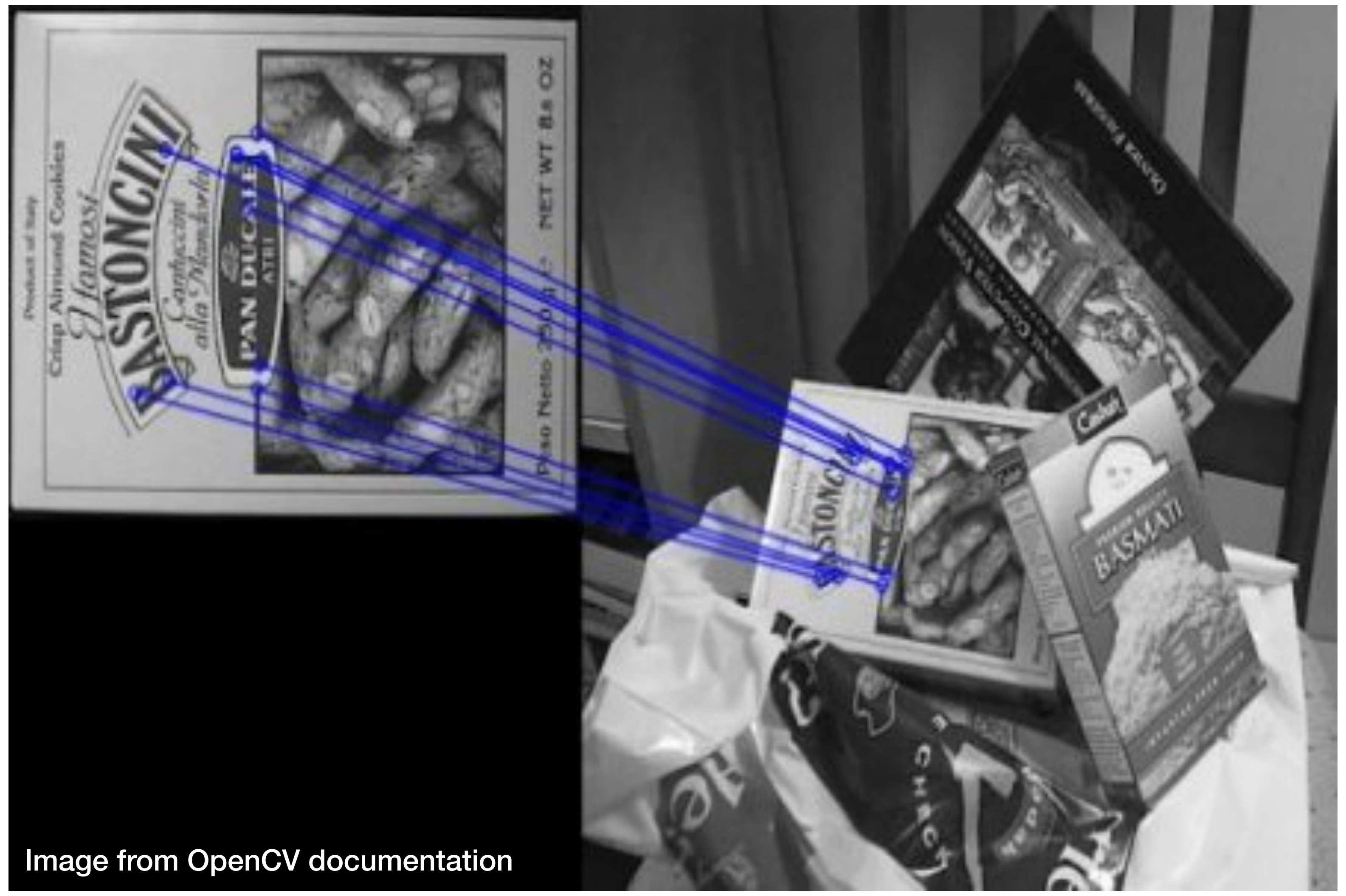
Feature detection

Feature description

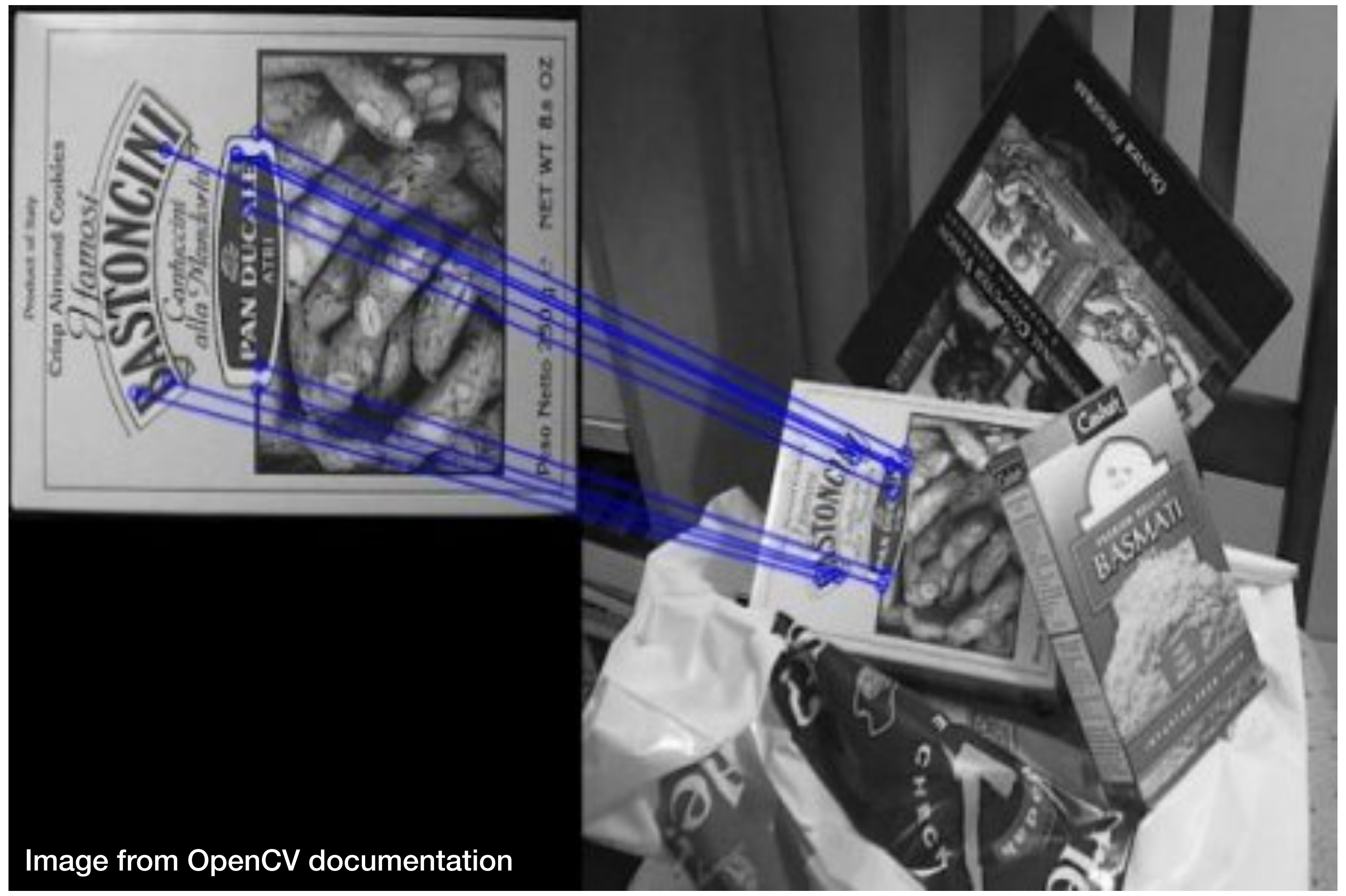
Feature matching

$$\text{ratio distance} = \frac{SSD(f_1, f_2)}{SSD(f_1, f_2')}$$

In previous lecture



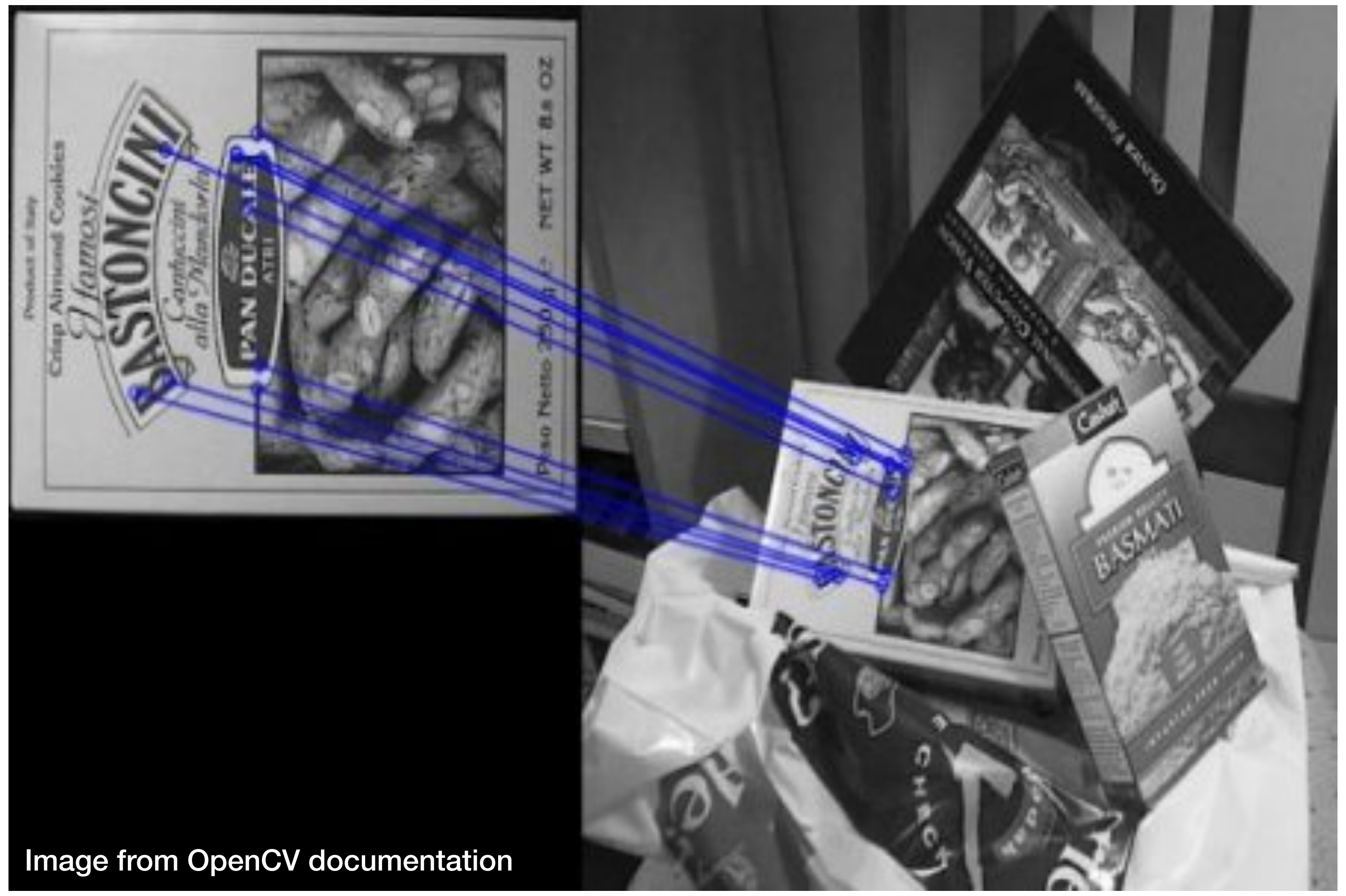
Relation between photos



Relation between photos

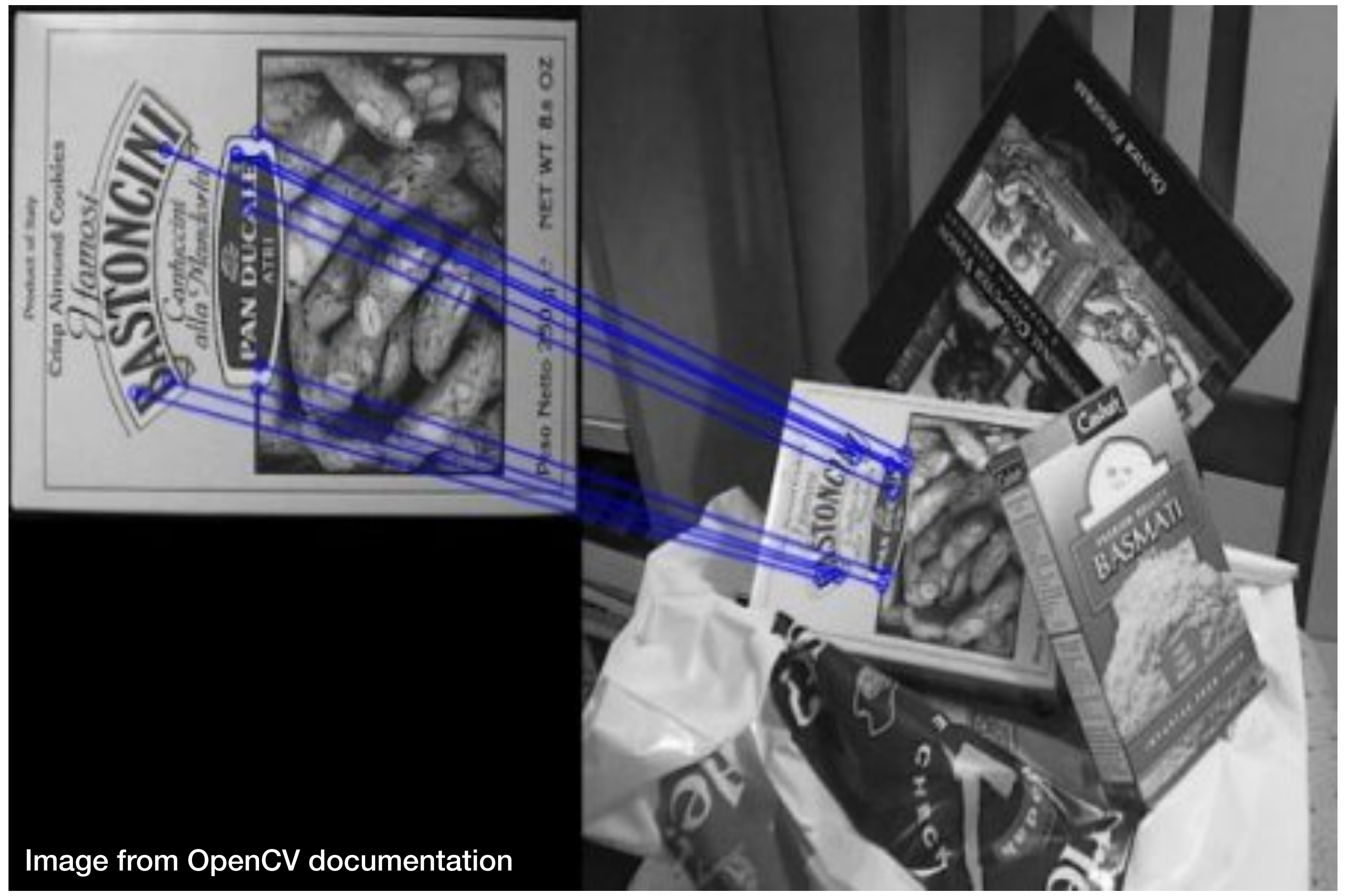
Warping

Image registration



Relation between photos

Image registration



Relation between photos

- Detect, describe and match features (last lecture)

Image registration

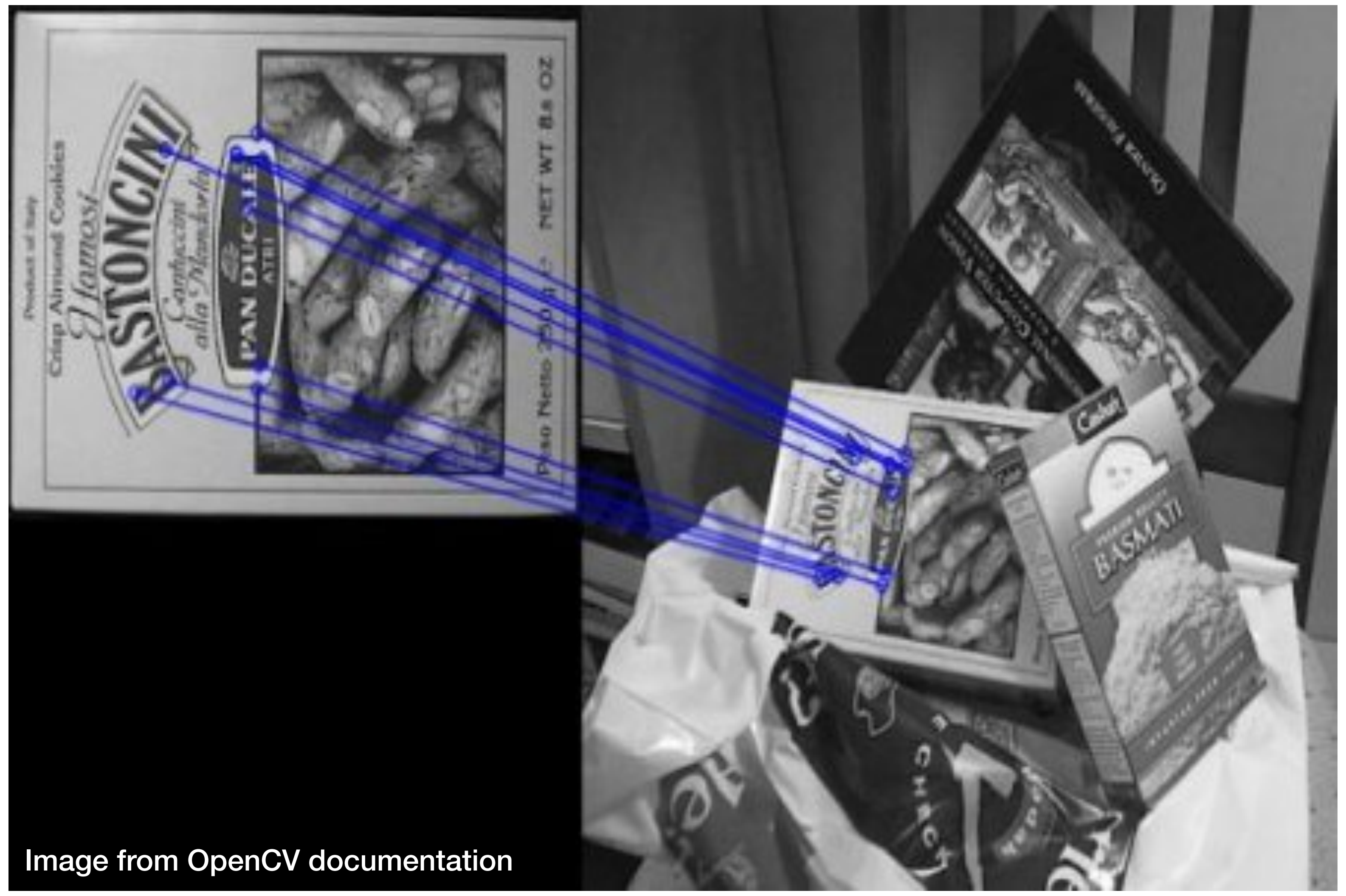
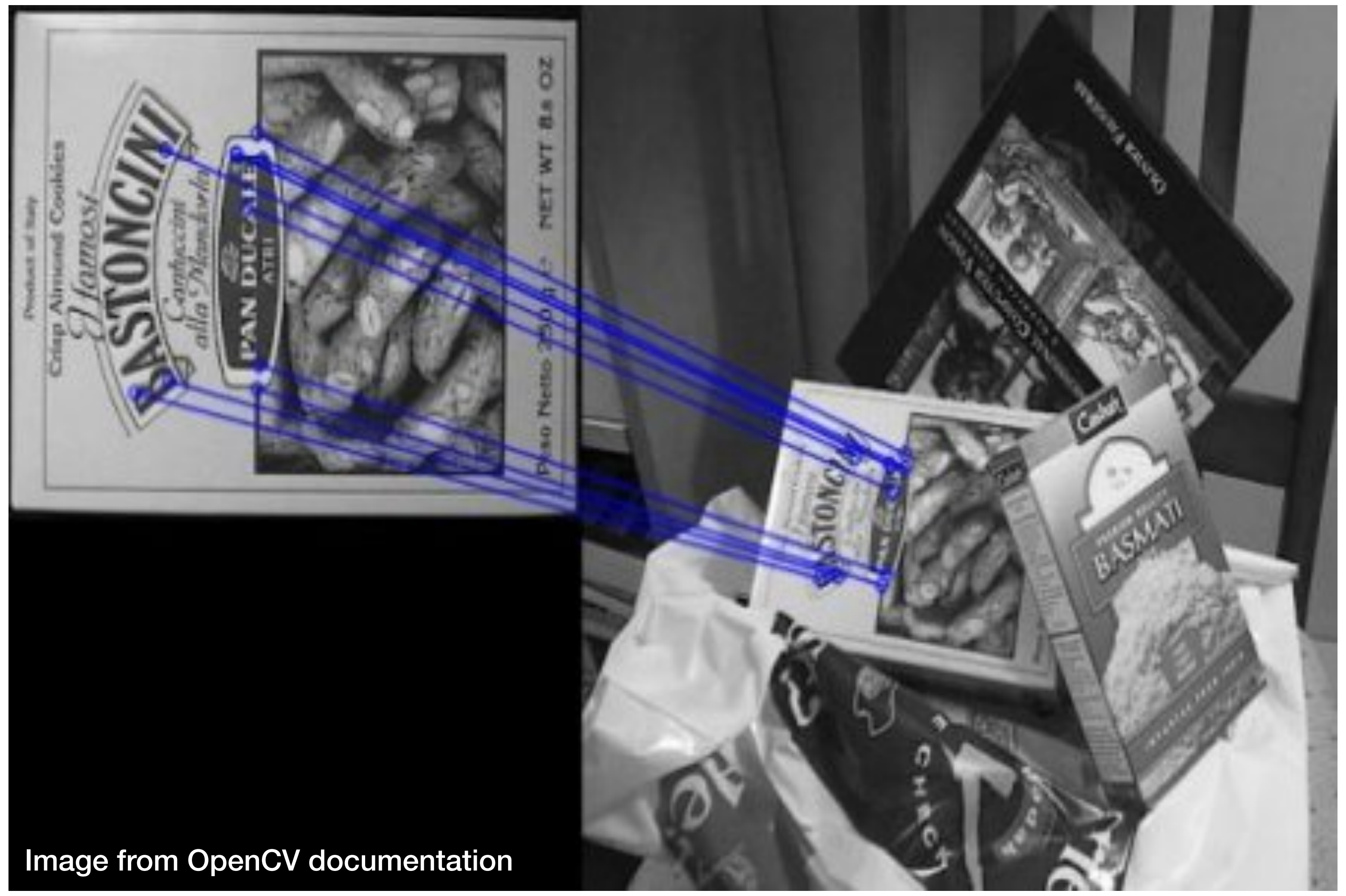


Image from OpenCV documentation

Relation between photos

- Detect, describe and match features (last lecture)
- Calculate transformation robustly

Image registration



Relation between photos

- Detect, describe and match features (last lecture)
- Calculate transformation robustly

RANSAC

Image registration

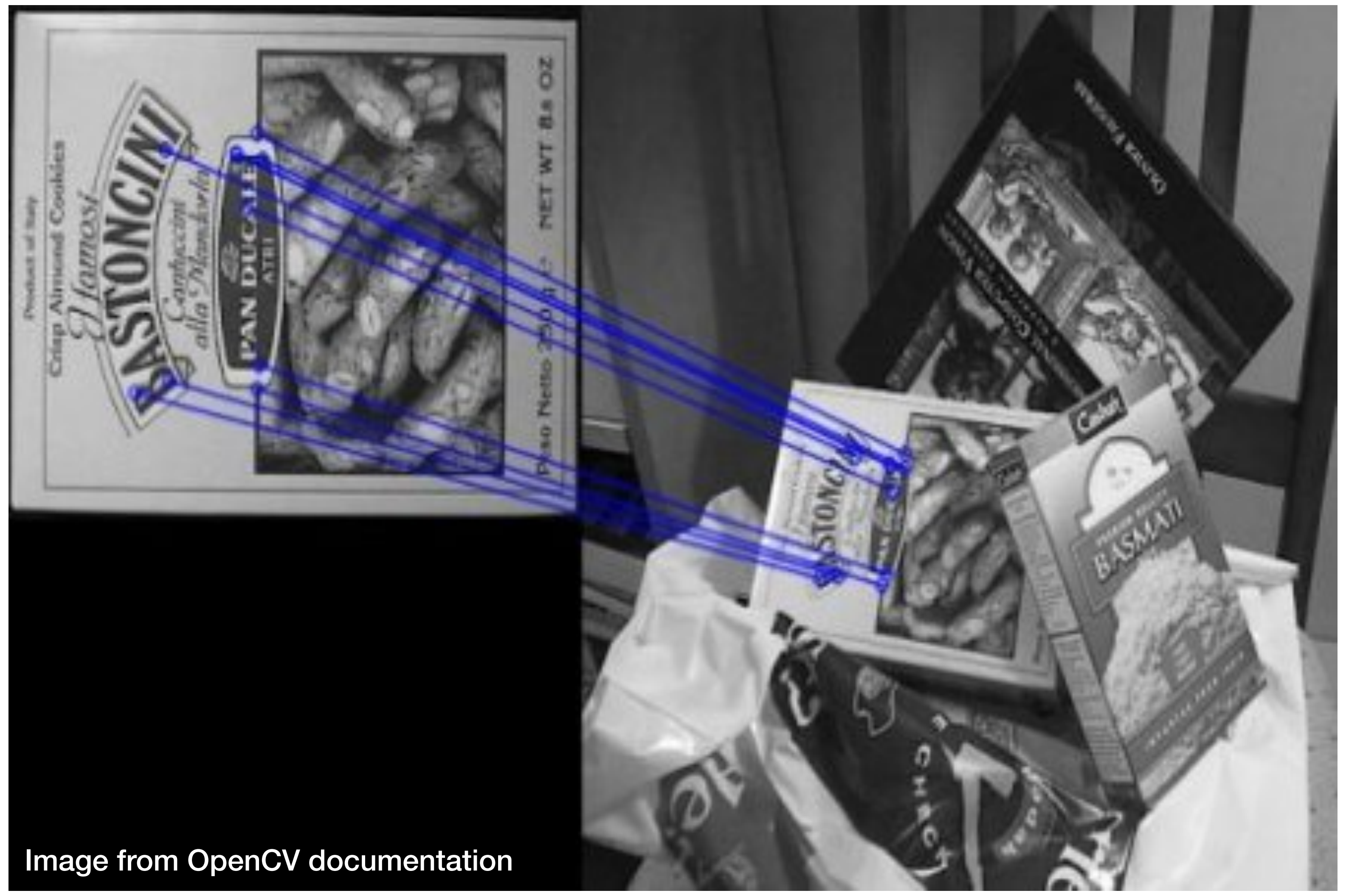


Image from OpenCV documentation

Relation between photos

- Detect, describe and match features (last lecture)
- Calculate transformation robustly

homography RANSAC

RANSAC

RANdom SAmple Consensus

RANSAC

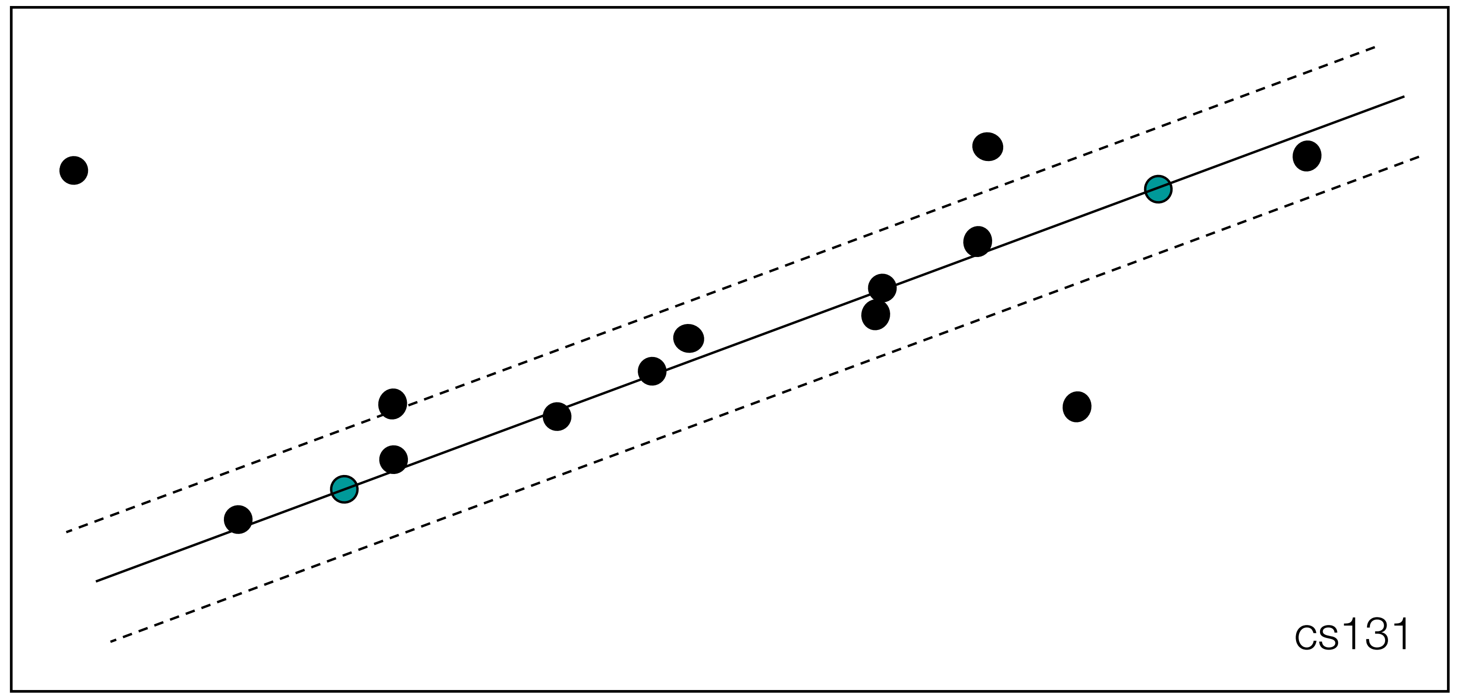
RANdom SAmple Consensus

Not just for feature matching!

RANSAC

RANdom SAmple Consensus

Not just for feature matching!



RANSAC

RANdom SAmple Consensus

- Start with a model

RANSAC

RANdom SAmple Consensus

- Start with a model
    - How many parameters?

RANSAC

RANdom SAmple Consensus

- Start with a model
  - How many parameters?
  - Minimal amount of data points n?

RANSAC

RANdom SAmple Consensus

- Start with a model
    - How many parameters?
    - Minimal amount of data points n?
- In each iteration

RANSAC

RANdom SAmple Consensus

- Start with a model
  - How many parameters?
  - Minimal amount of data points n?
- In each iteration
  - Sample n points

RANSAC

RANdom SAmple Consensus

- Start with a model
  - How many parameters?
  - Minimal amount of data points n?
- In each iteration
  - Sample n points
  - Fit model parameters

RANSAC

RANdom SAmple Consensus

- Start with a model
  - How many parameters?
  - Minimal amount of data points n?
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)

RANSAC

RANdom SAmple Consensus

- Start with a model
  - How many parameters?
  - Minimal amount of data points n?
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters

RANSAC

RANdom SAmple Consensus

- Start with a model
  - How many parameters?
  - Minimal amount of data points n?
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result

RANSAC

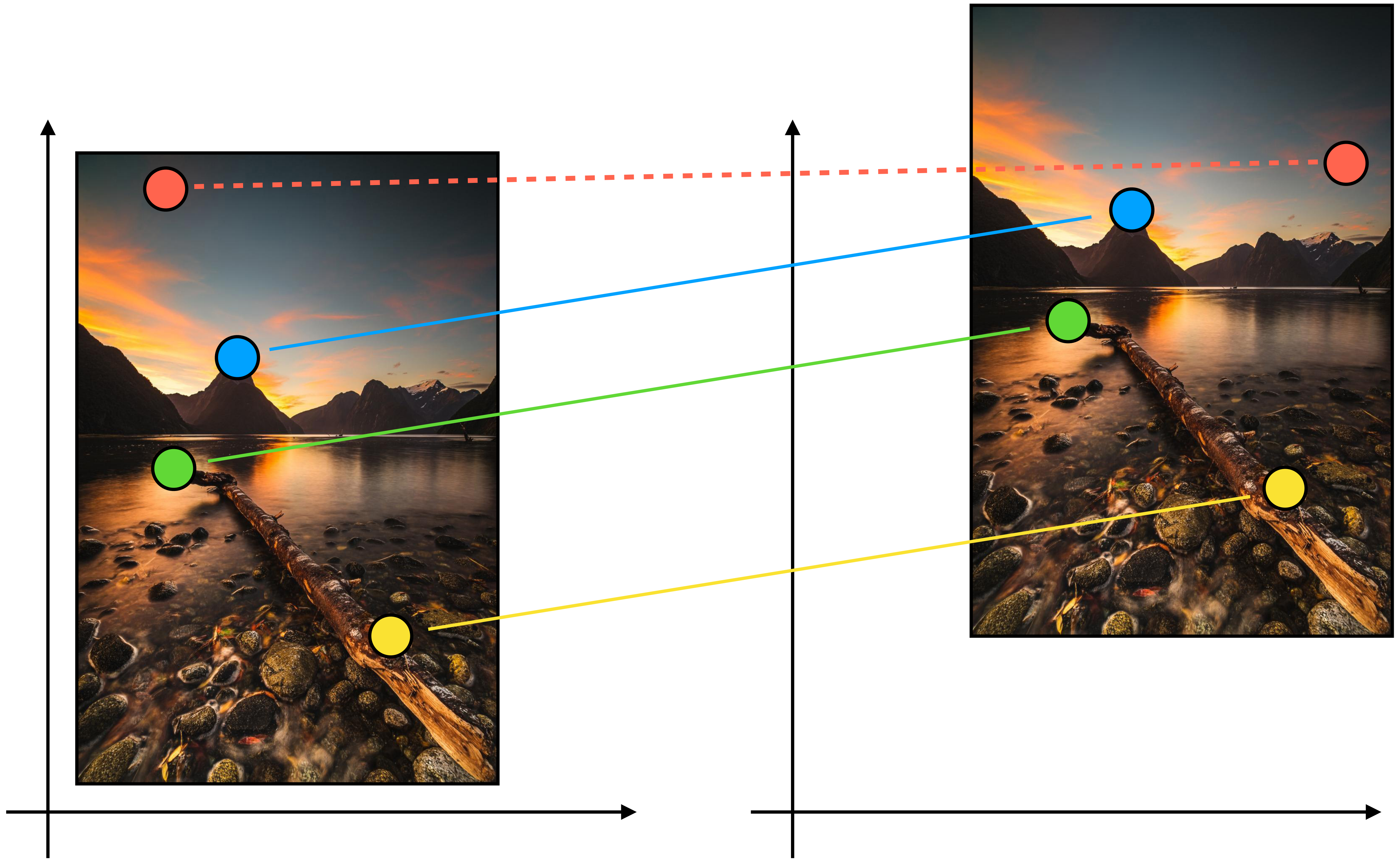
RANdom SAmple Consensus

- Start with a model
  - How many parameters?
  - Minimal amount of data points n?
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result

RANSAC

RANdom SAmple Consensus

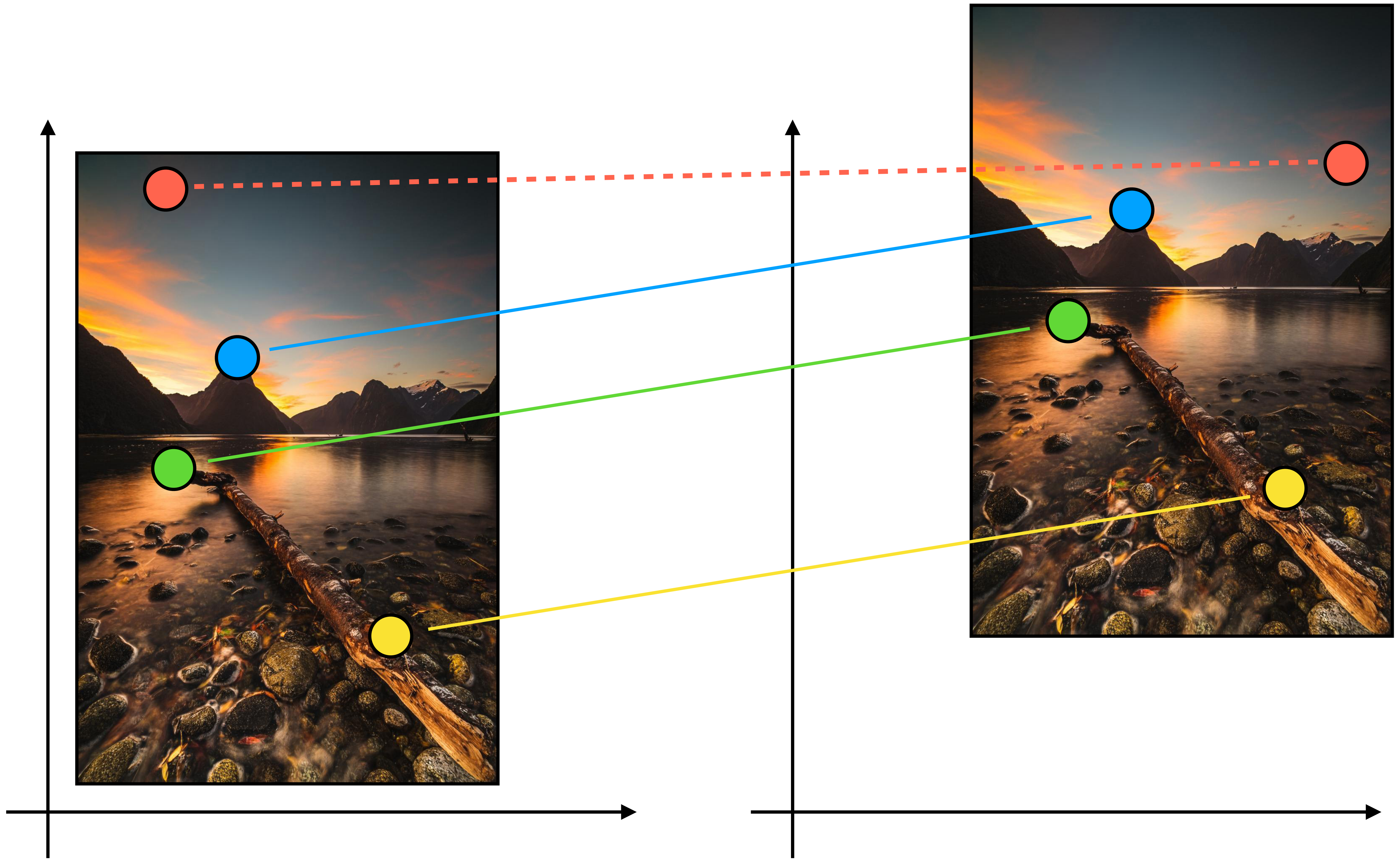
- Start with a model
  - How many parameters?
  - Minimal amount of data points n?
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

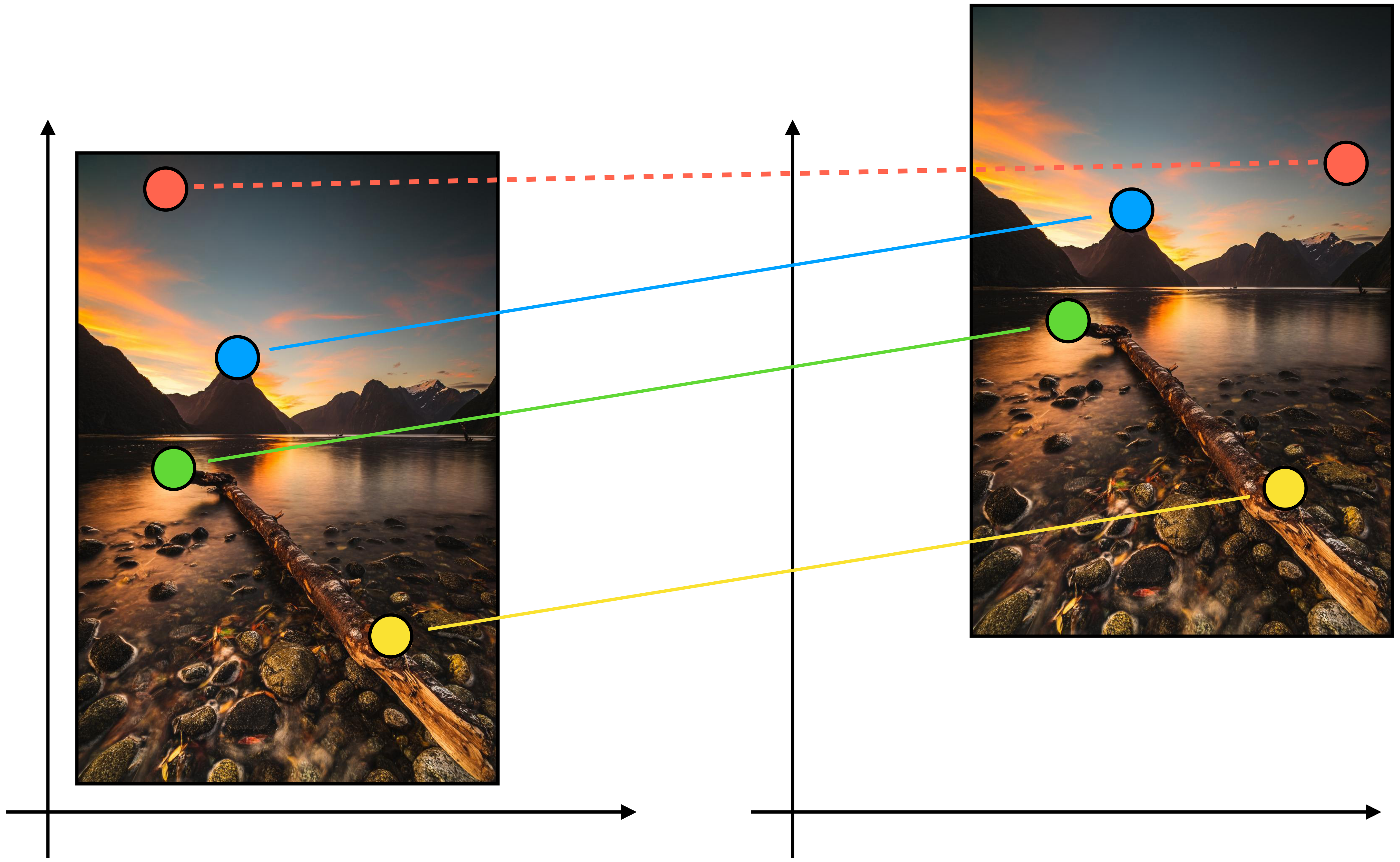
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n?
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

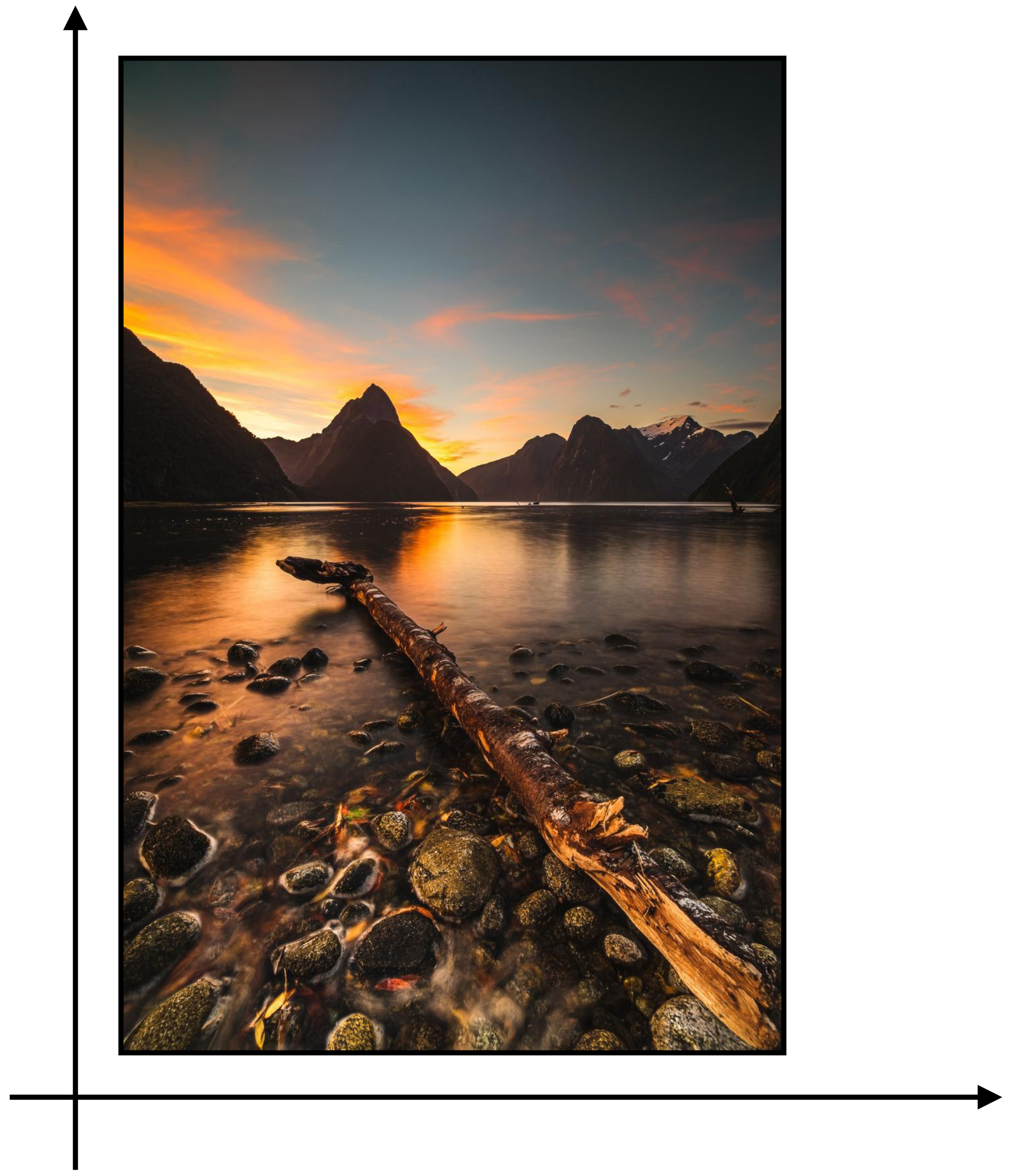
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

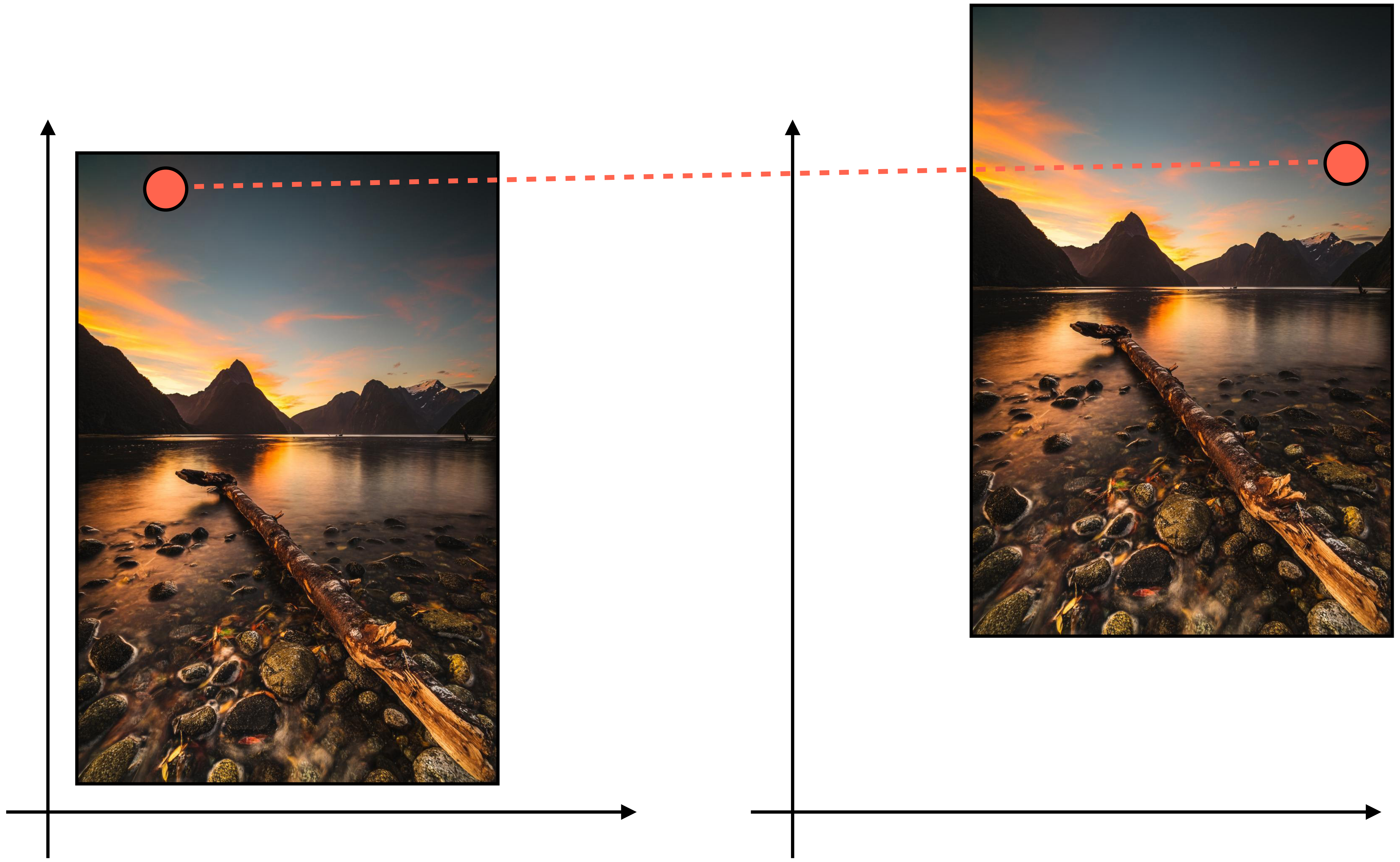
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

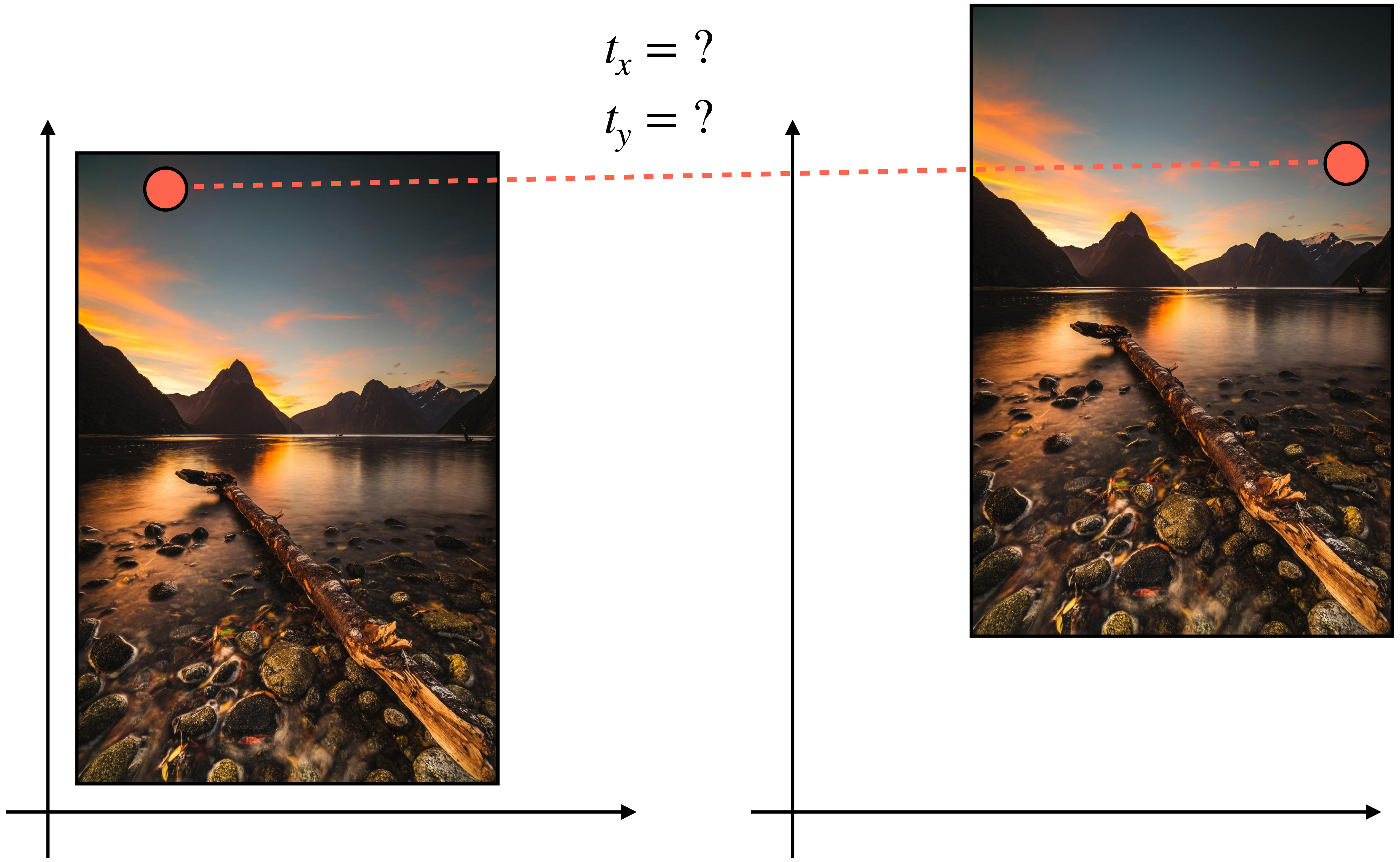
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - **Sample n points**
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

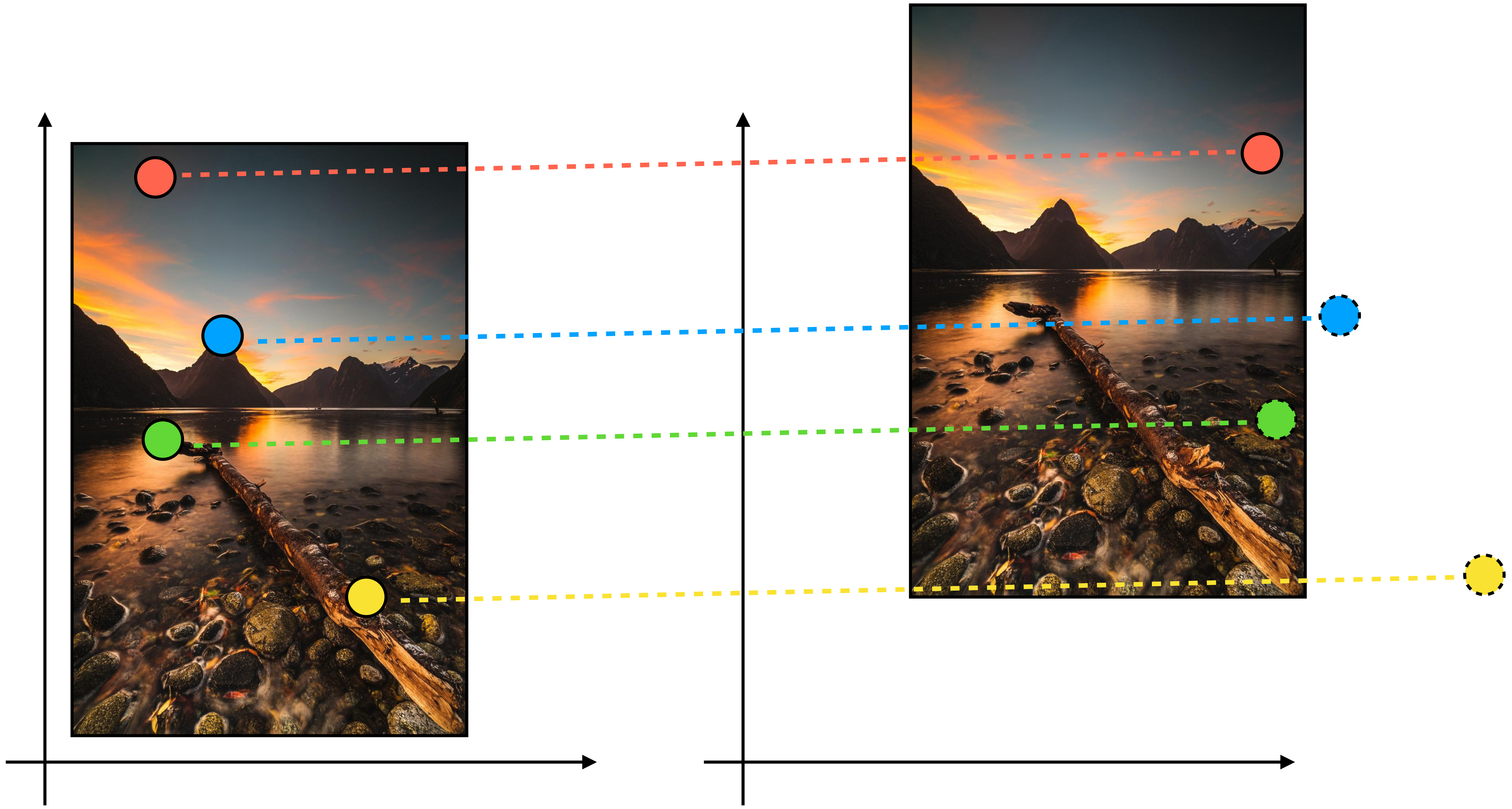
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - **Fit model parameters**
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

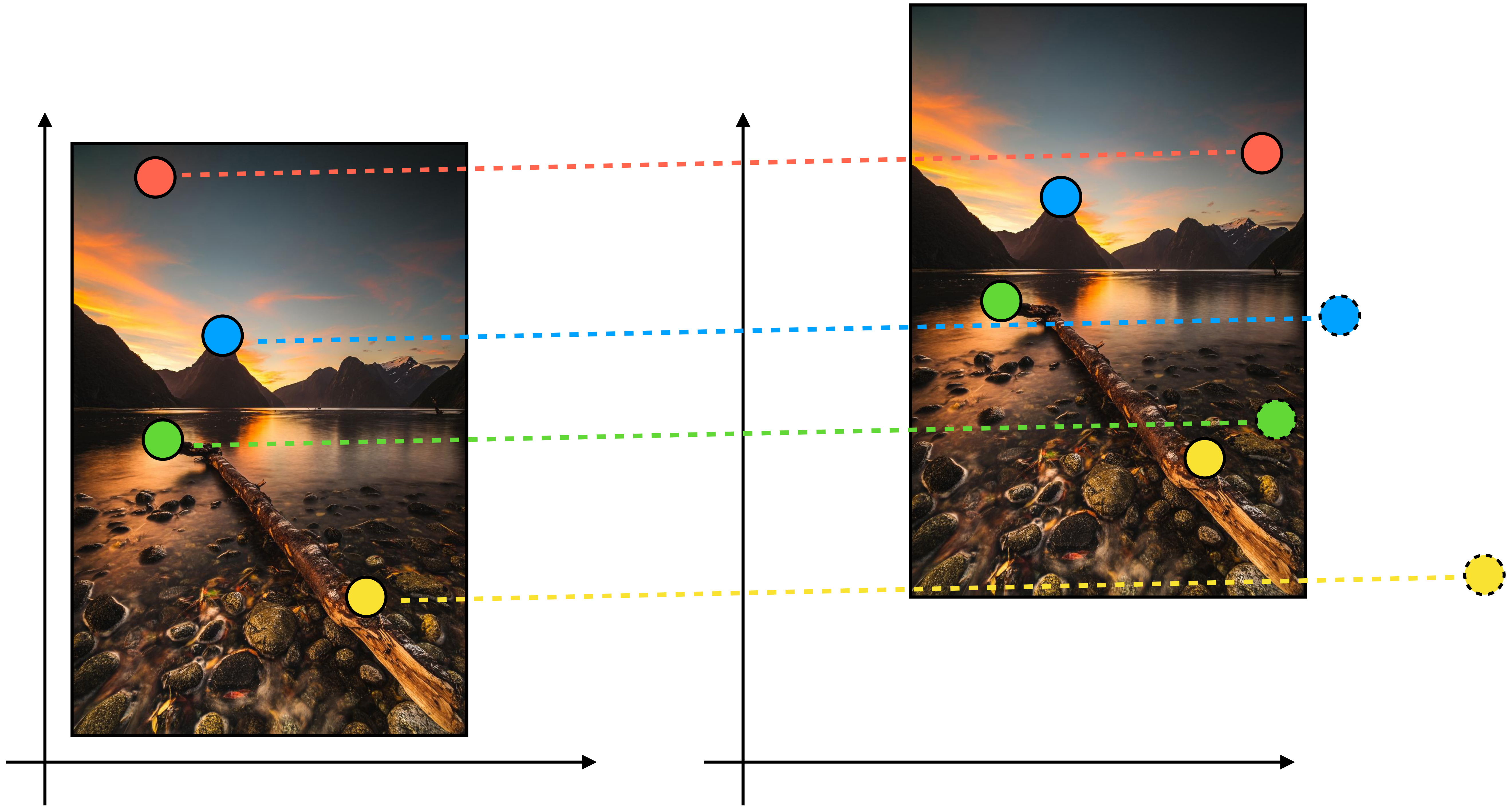
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

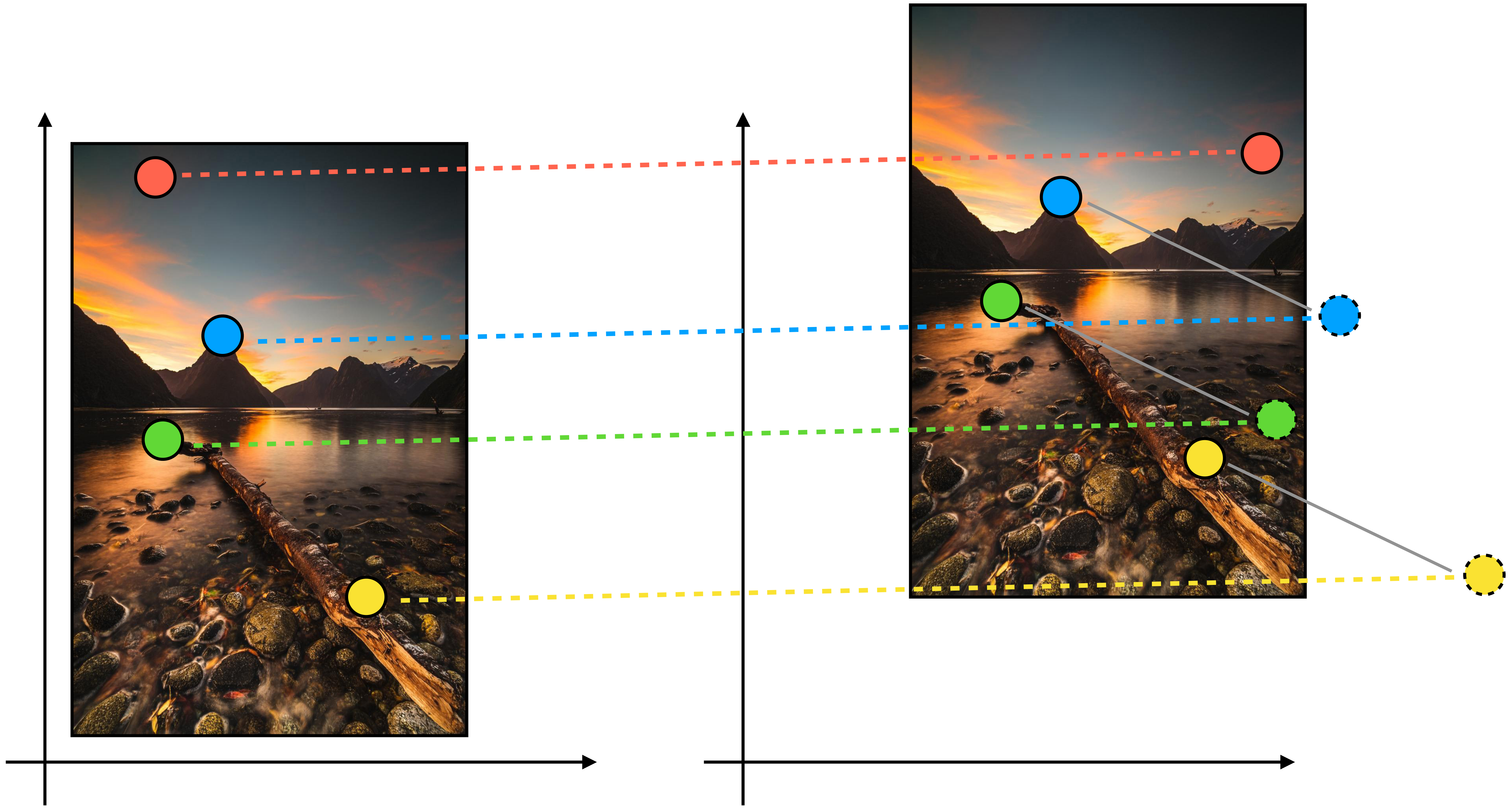
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

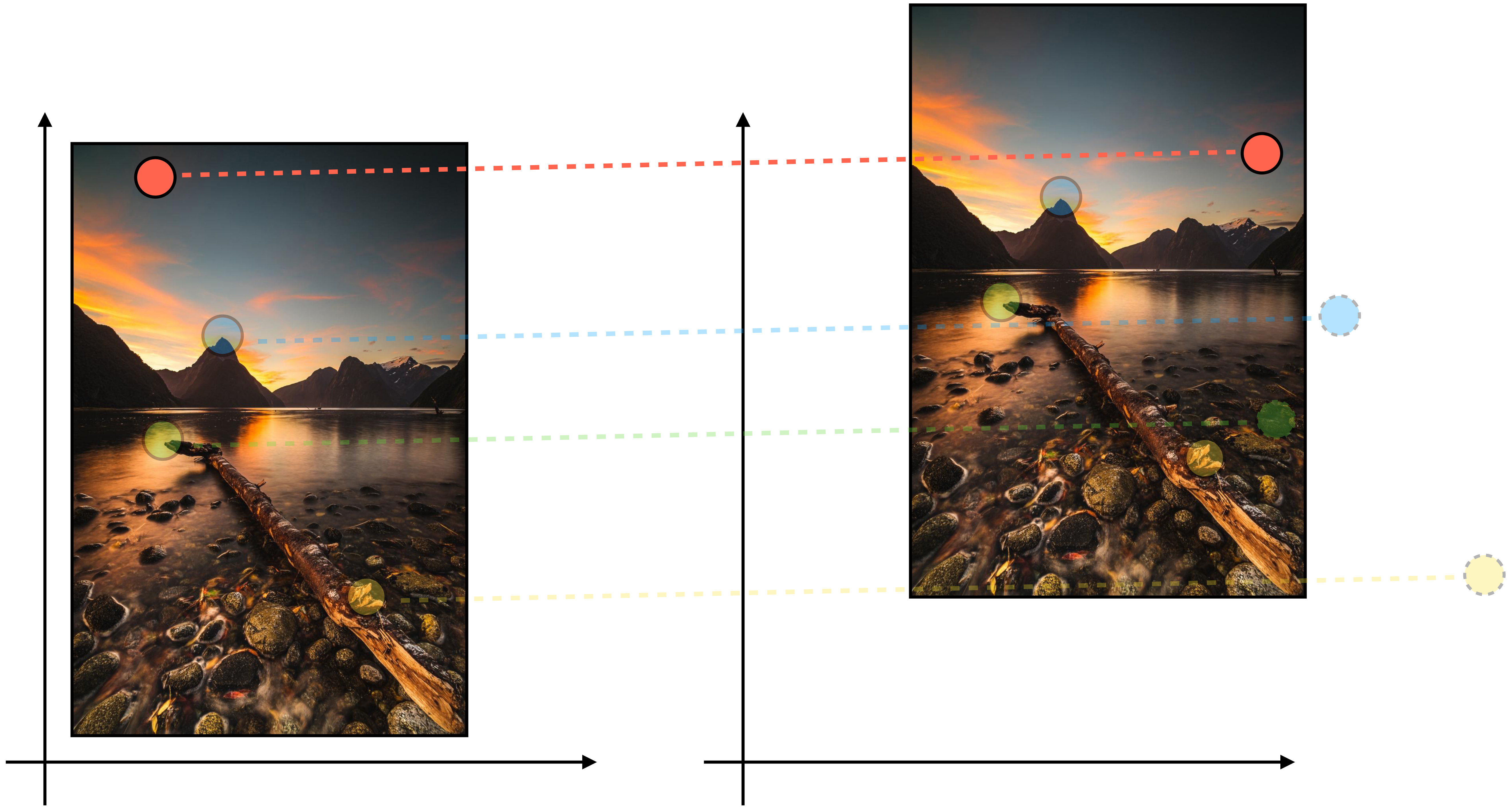
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

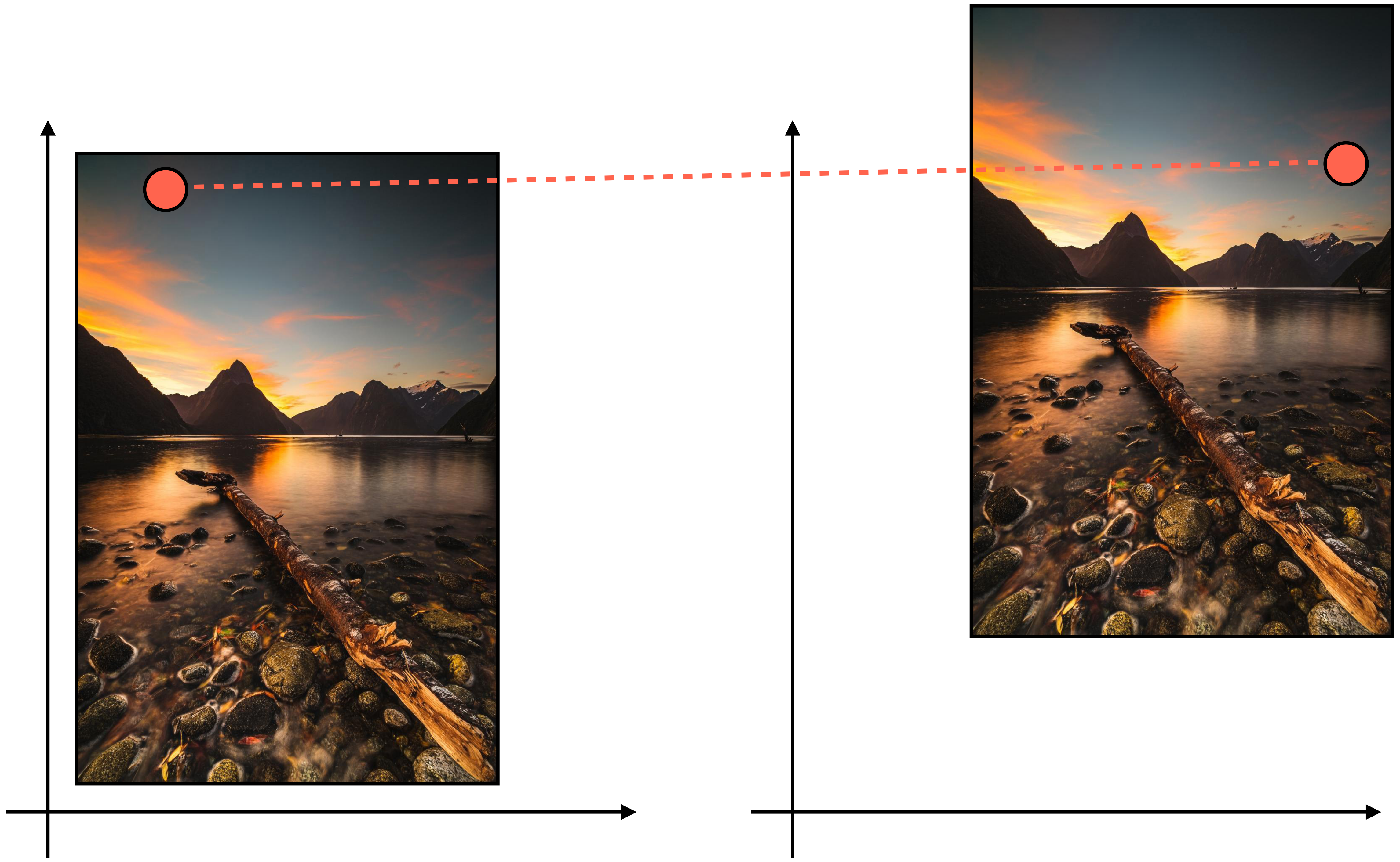
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

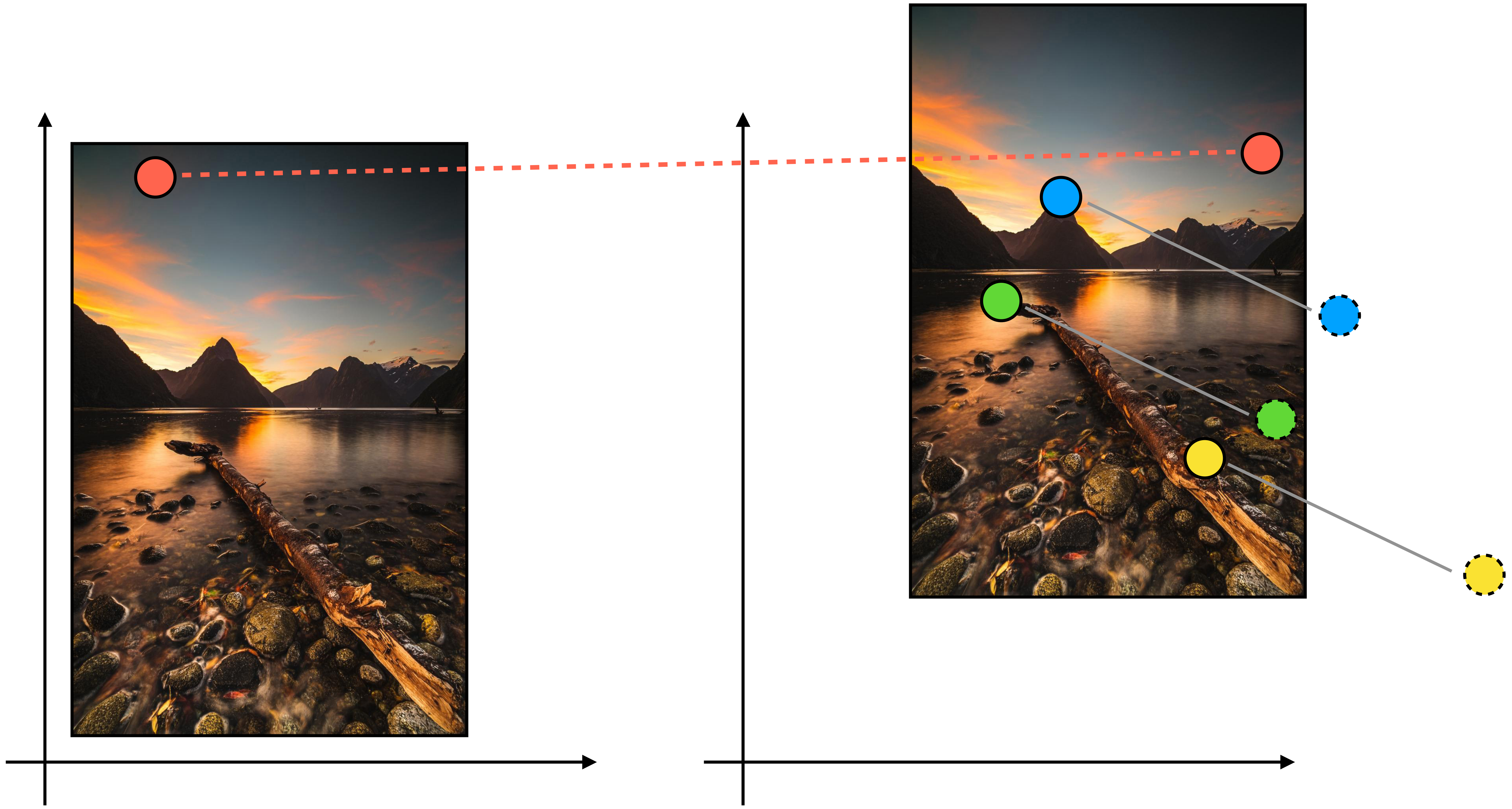
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - **Revise model parameters**
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

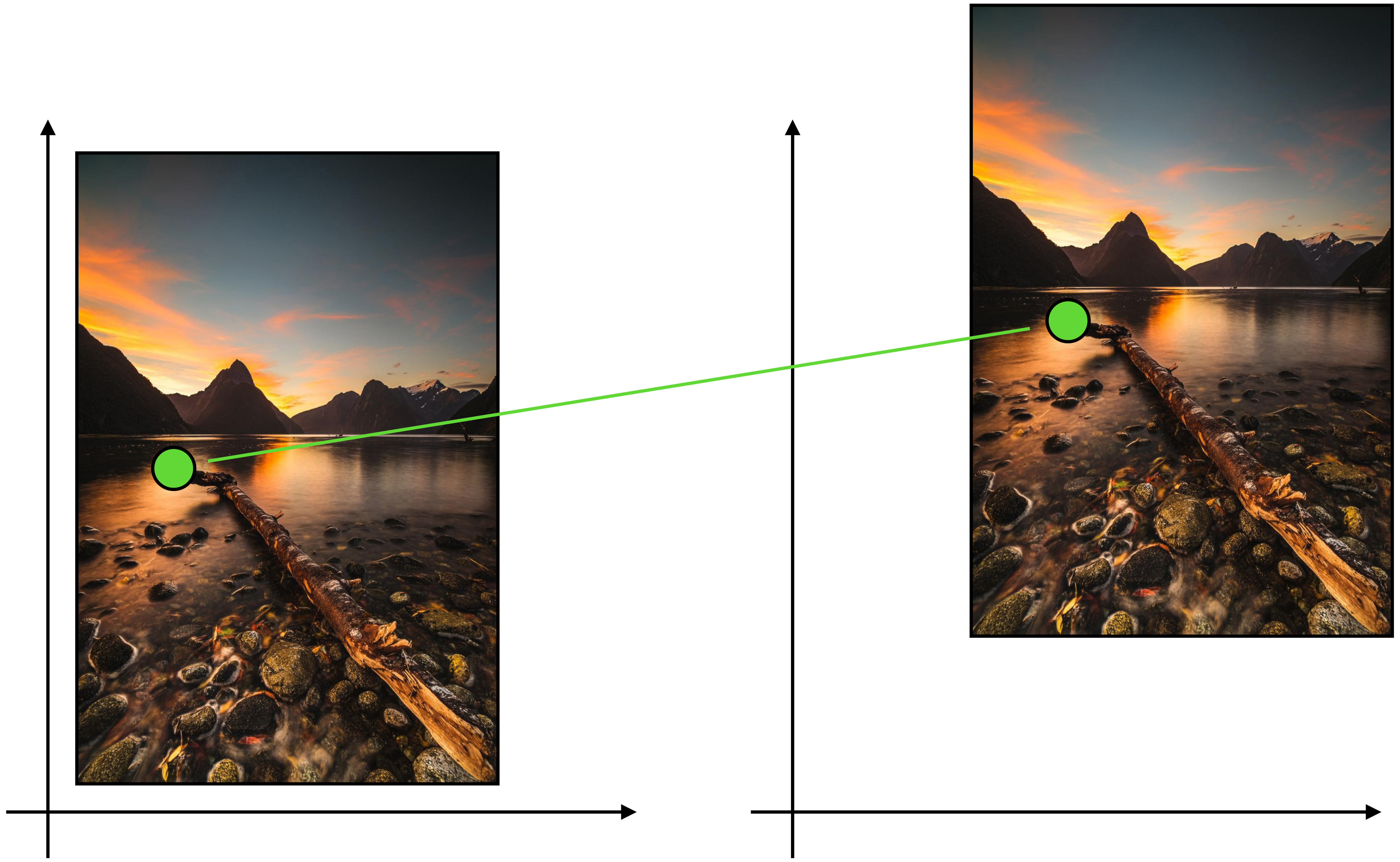
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - **Calculate error on all points, save best result**



RANSAC

RANdom SAmple Consensus

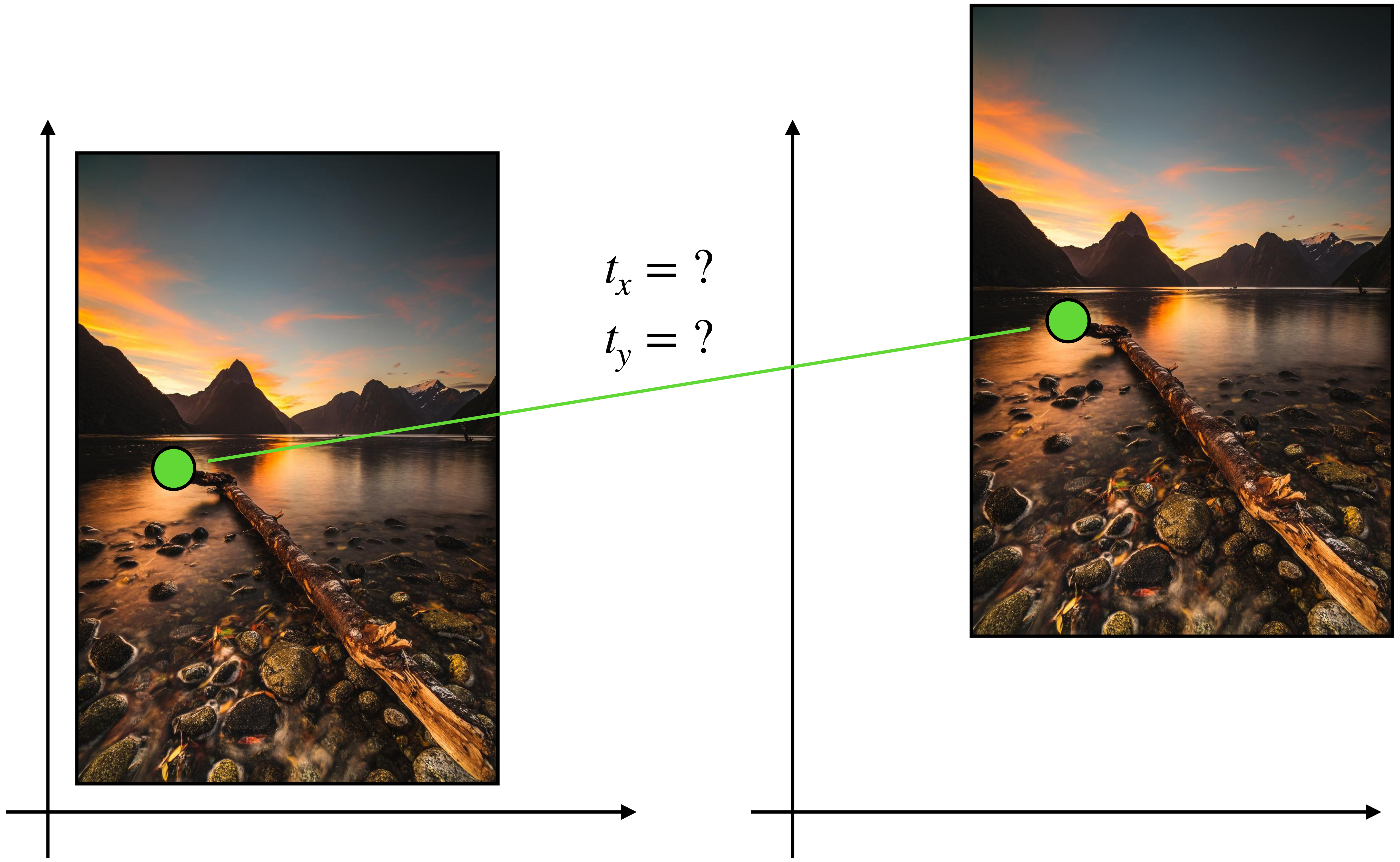
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - **Sample n points**
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

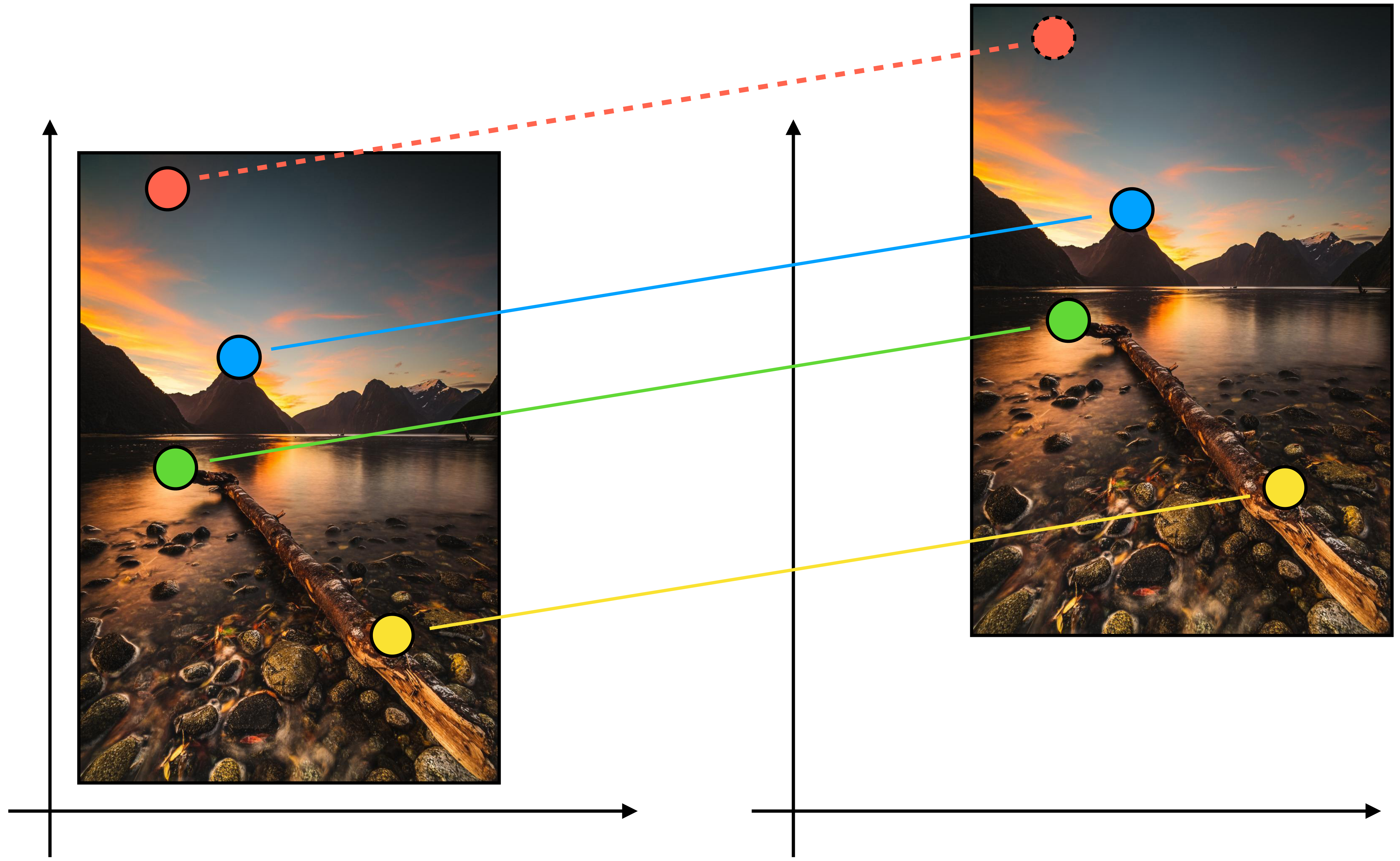
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - **Fit model parameters**
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

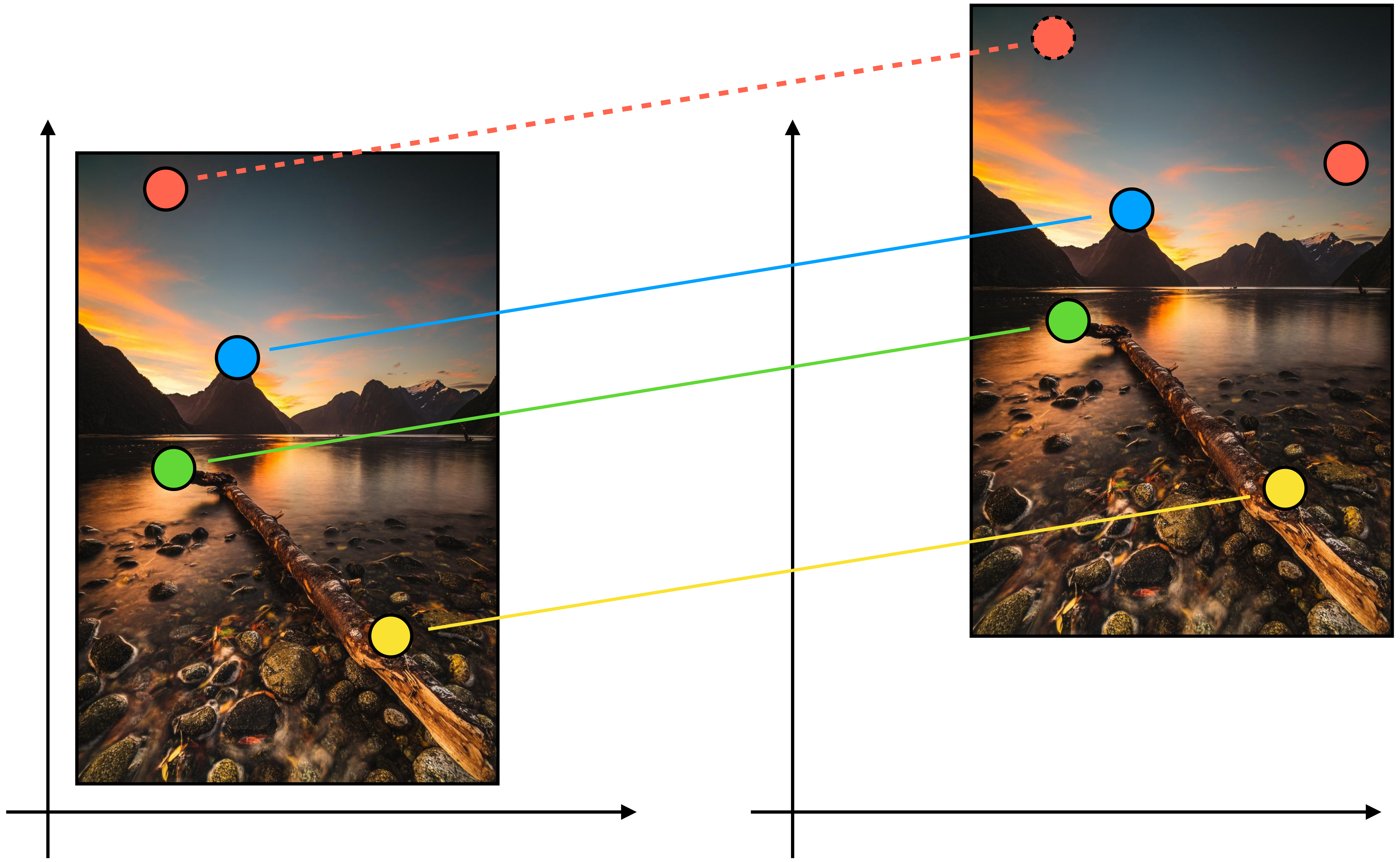
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

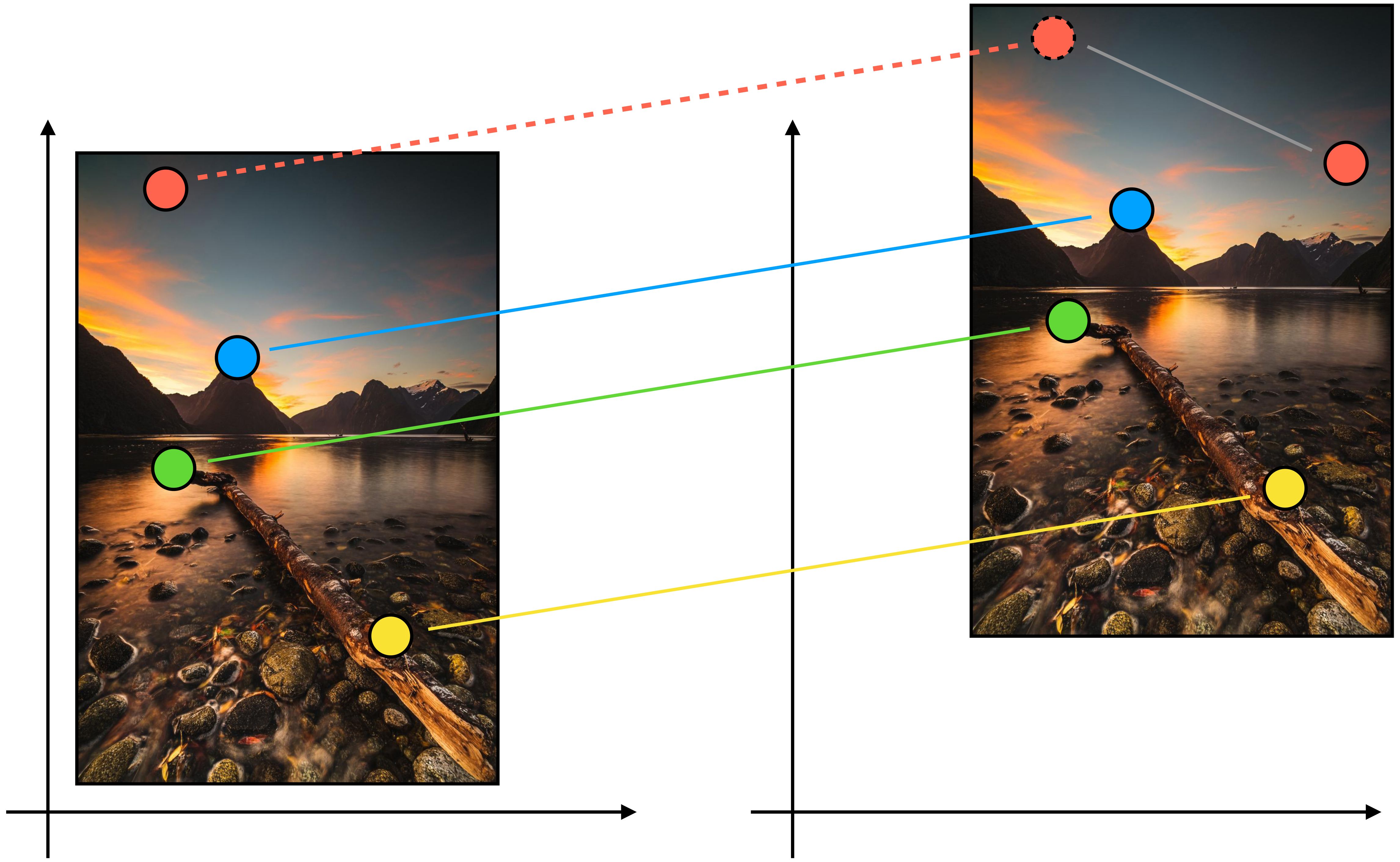
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

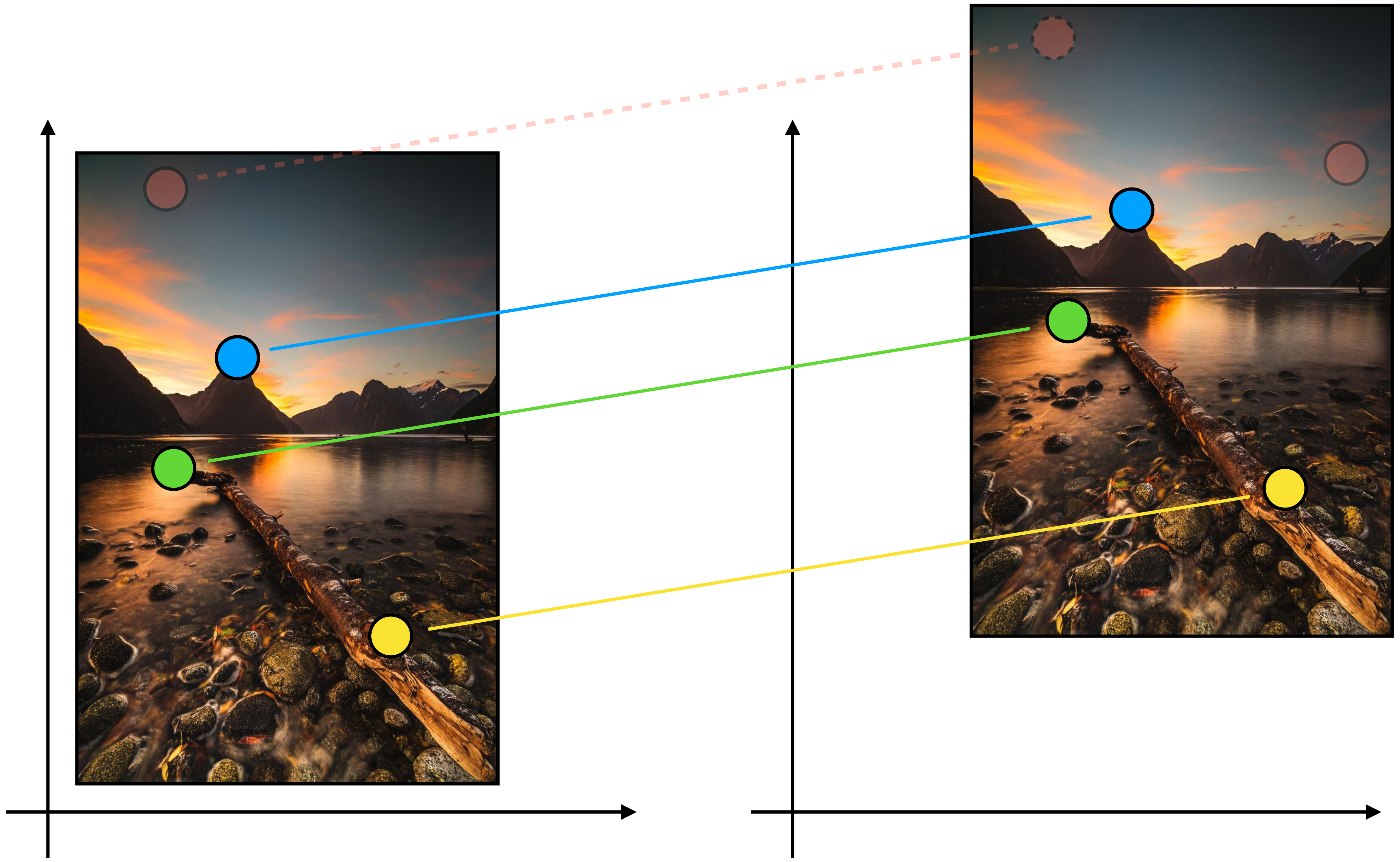
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

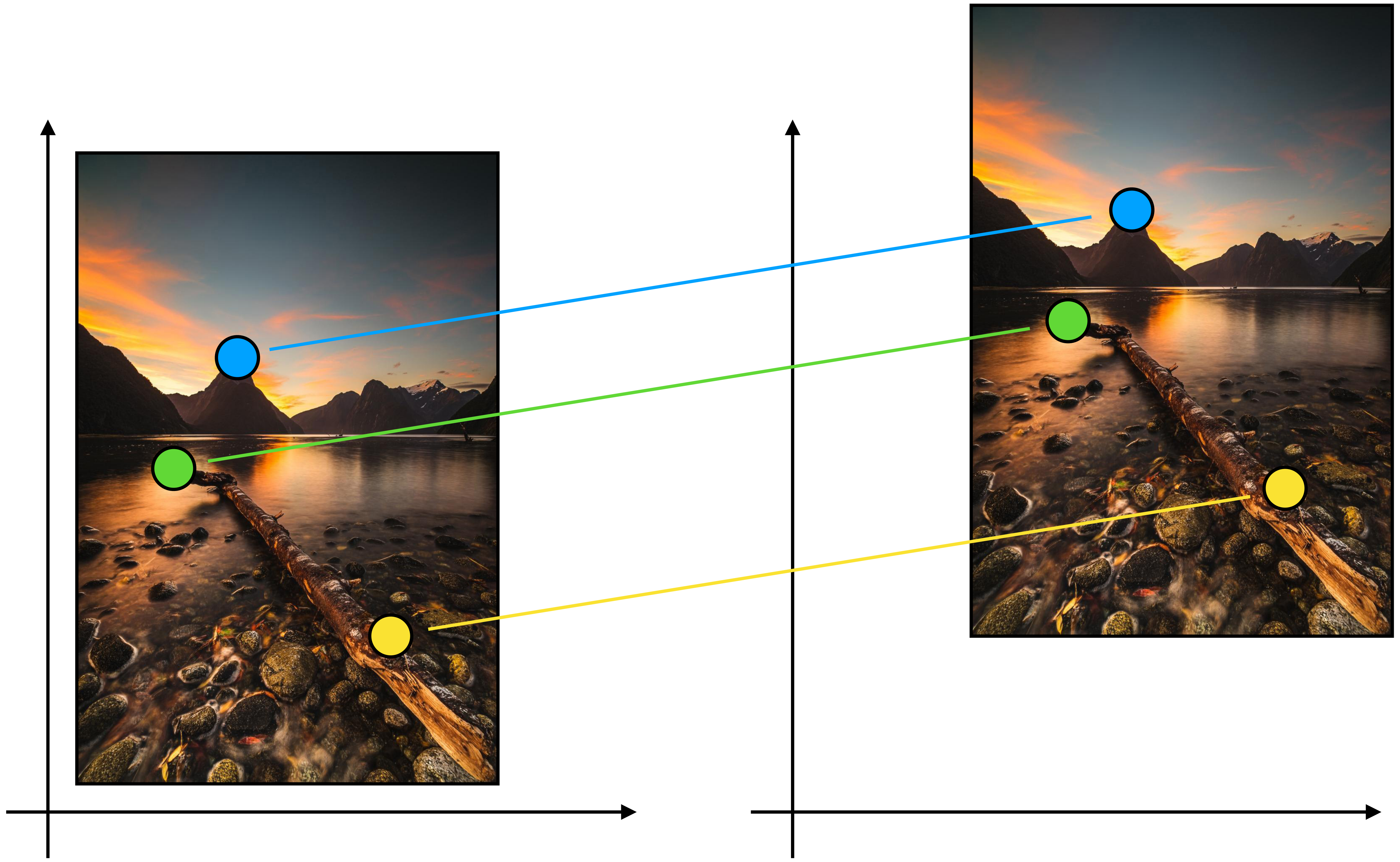
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - **Find inliers (below some threshold)**
  - Revise model parameters
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

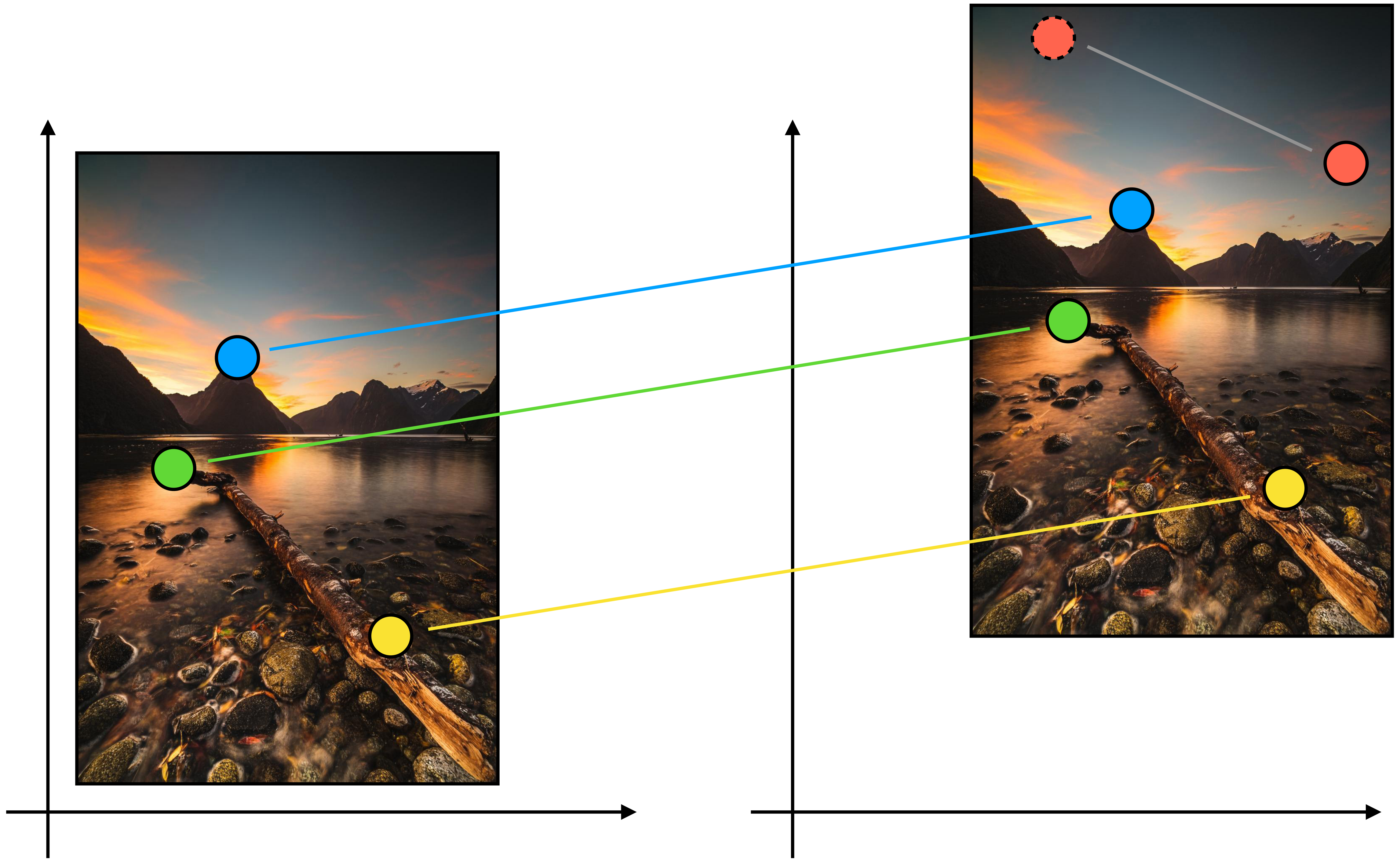
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - **Revise model parameters**
  - Calculate error on all points, save best result



RANSAC

RANdom SAmple Consensus

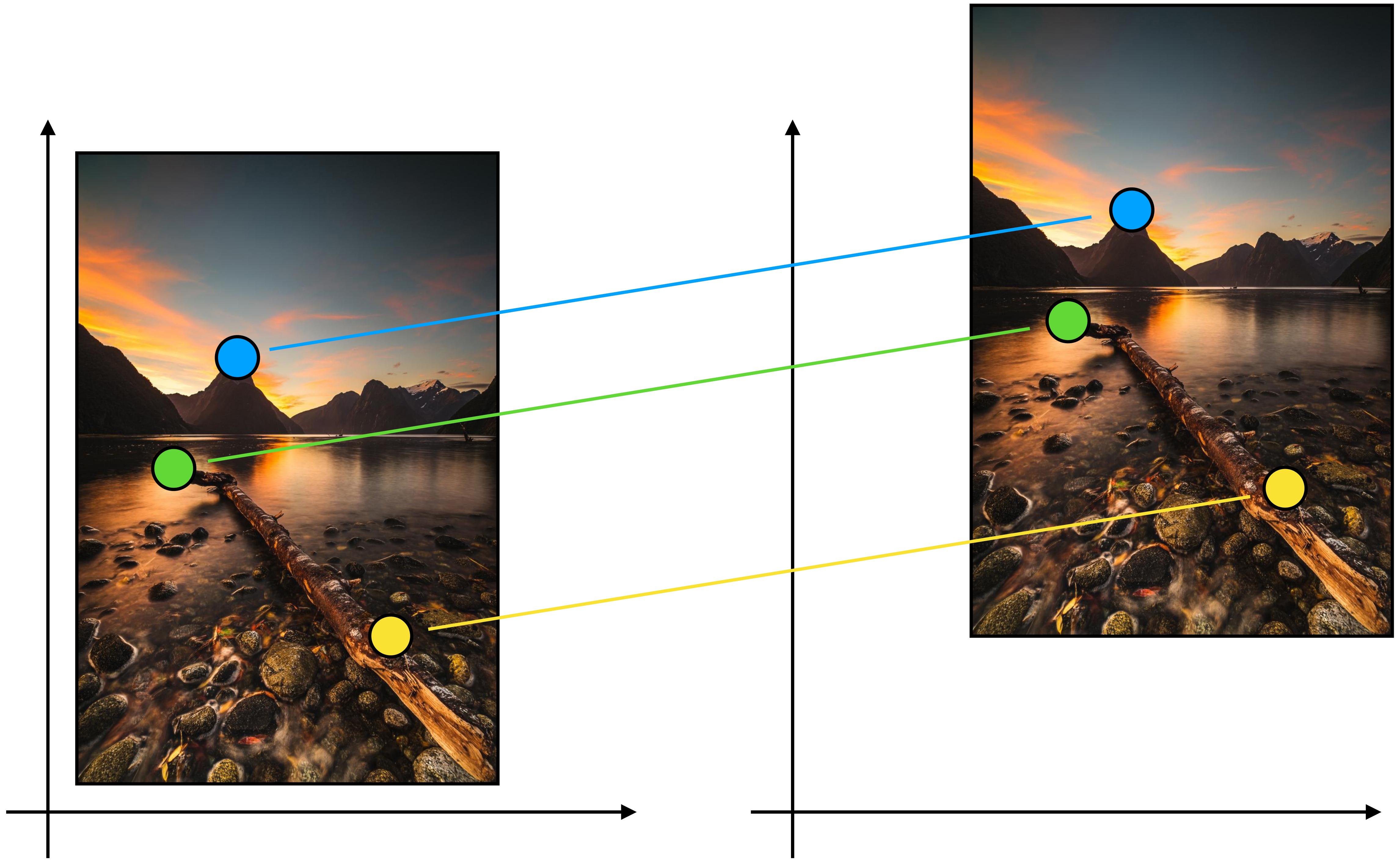
- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - **Calculate error on all points, save best result**



RANSAC

RANdom SAmple Consensus

- Start with a model
  - How many parameters? 2 (t_x, t_y)
  - Minimal amount of data points n? 1 pair
- In each iteration
  - Sample n points
  - Fit model parameters
  - Find inliers (below some threshold)
  - Revise model parameters
  - Calculate error on all points, save best result



Use best result

RANSAC

RANdom SAmple Consensus

How many iterations?

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers)

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers) p^n

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers) p^n

Probability of a failed iteration

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers) p^n

Probability of a failed iteration $1 - p^n$

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers) p^n

Probability of a failed iteration $1 - p^n$

Probability of all M iterations to fail

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers) p^n

Probability of a failed iteration $1 - p^n$

Probability of all M iterations to fail $(1 - p^n)^M$

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers) p^n

Probability of a failed iteration $1 - p^n$

Probability of all M iterations to fail $(1 - p^n)^M$

Probability of success after M iterations

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

Probability of a successful iteration (all points are inliers) p^n

Probability of a failed iteration $1 - p^n$

Probability of all M iterations to fail $(1 - p^n)^M$

Probability of success after M iterations $S = 1 - (1 - p^n)^M$

RANSAC

RANdom SAmple Consensus

How many iterations?

p	probability that a given data point is valid (an inlier)
n	amount of data points that define a transformation
M	number of iterations
S	probability of success after M iterations

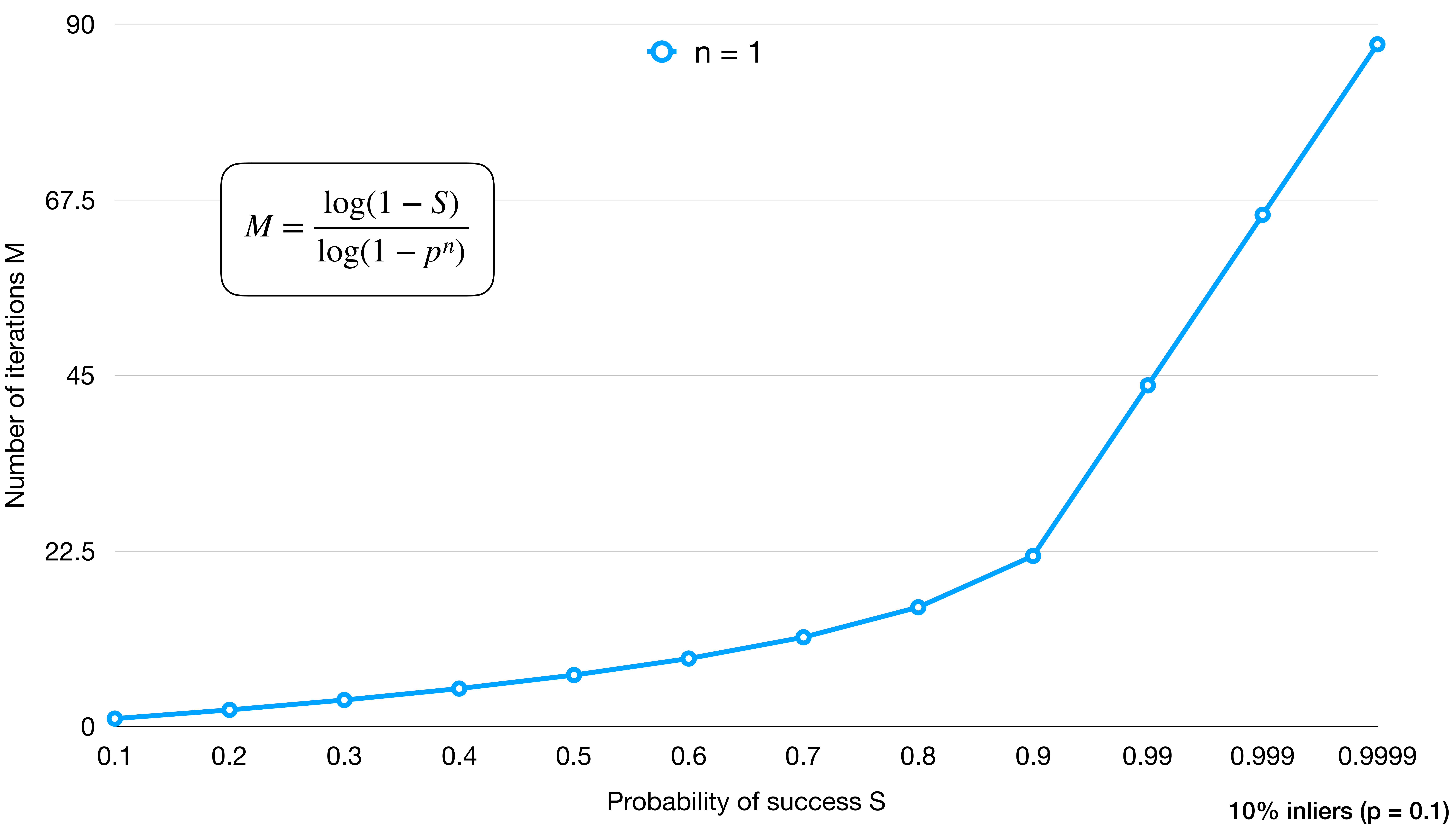
Probability of a successful iteration (all points are inliers) p^n

Probability of a failed iteration $1 - p^n$

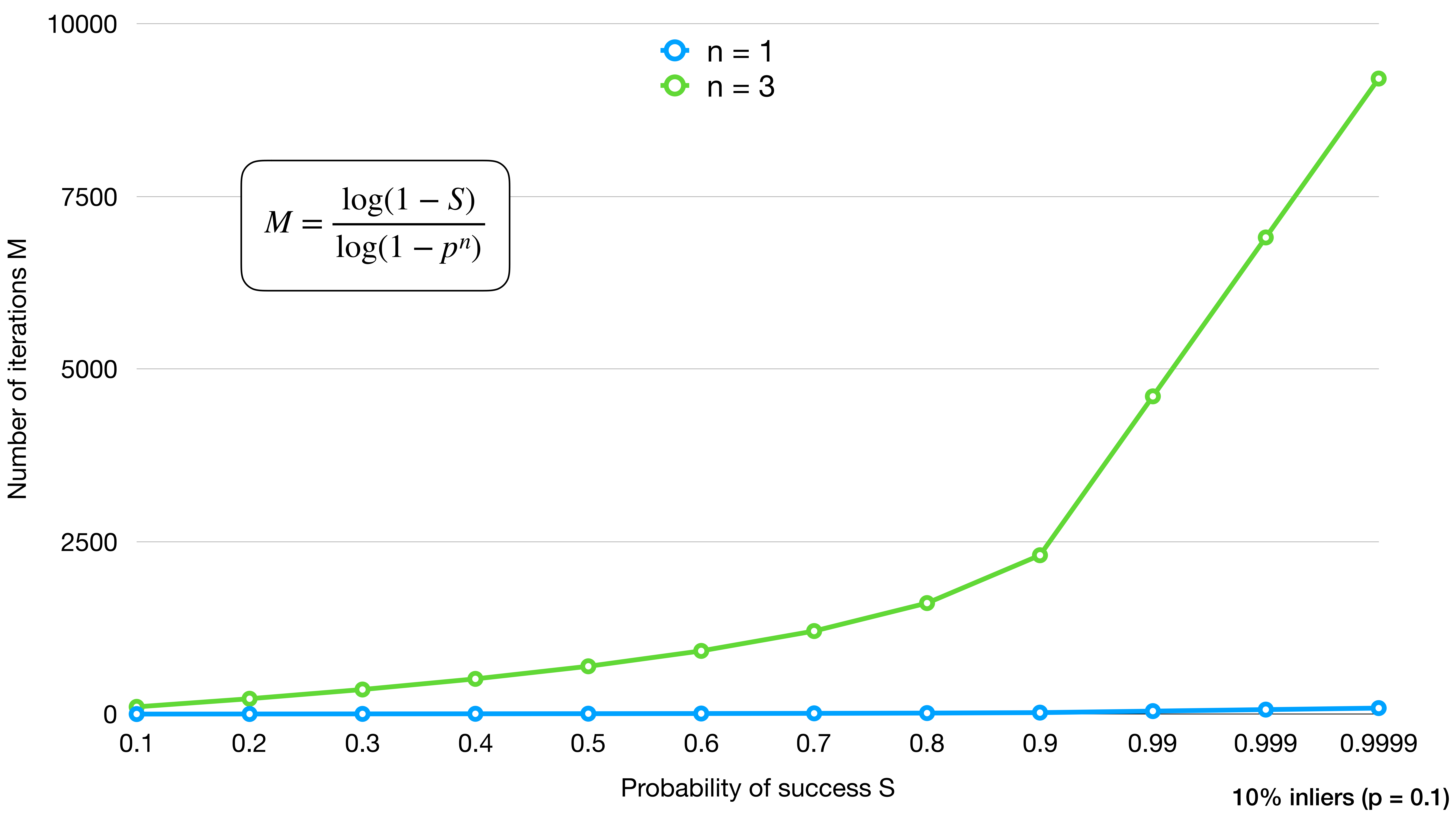
Probability of all M iterations to fail $(1 - p^n)^M$

Probability of success after M iterations $S = 1 - (1 - p^n)^M$

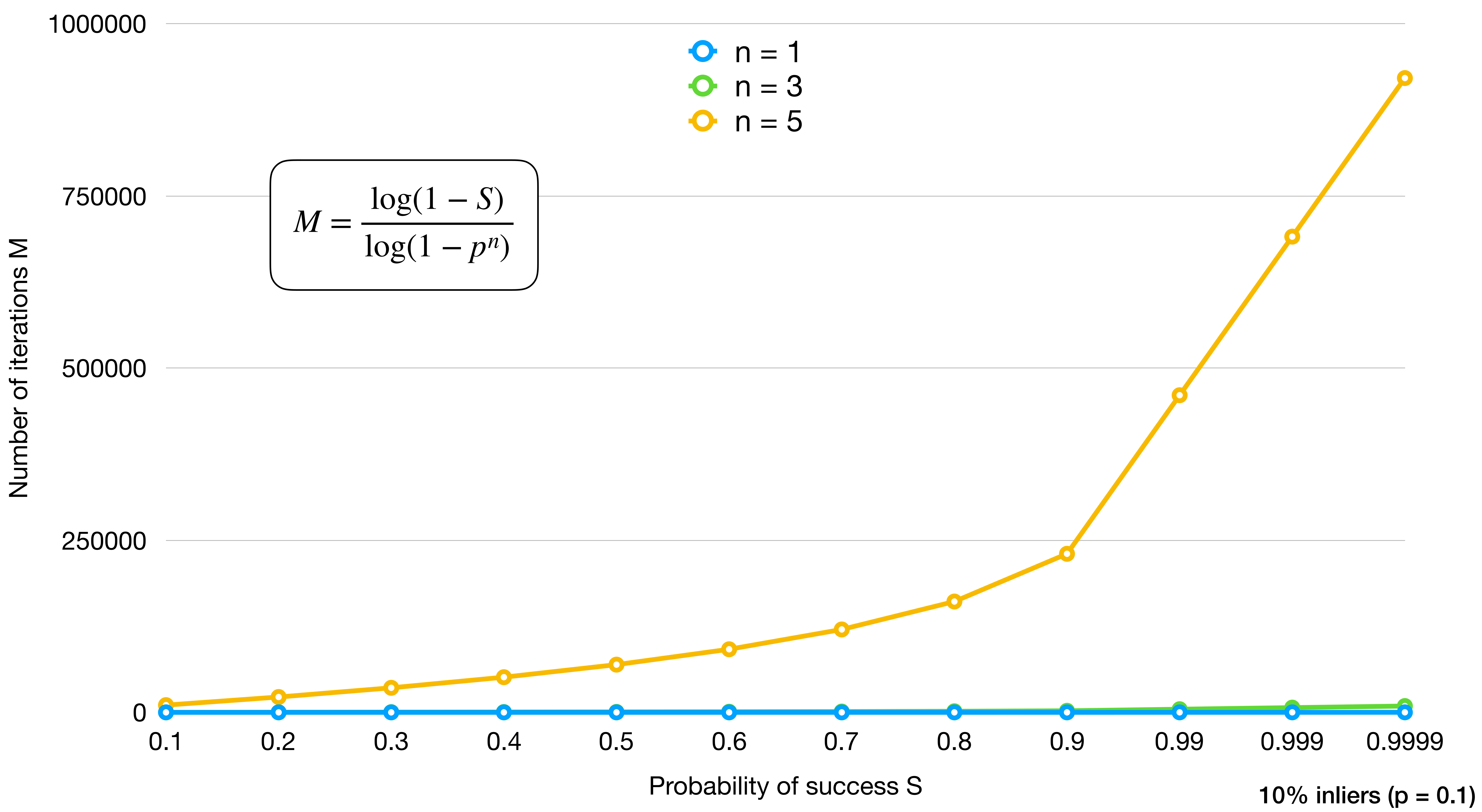
$$M = \frac{\log(1 - S)}{\log(1 - p^n)}$$

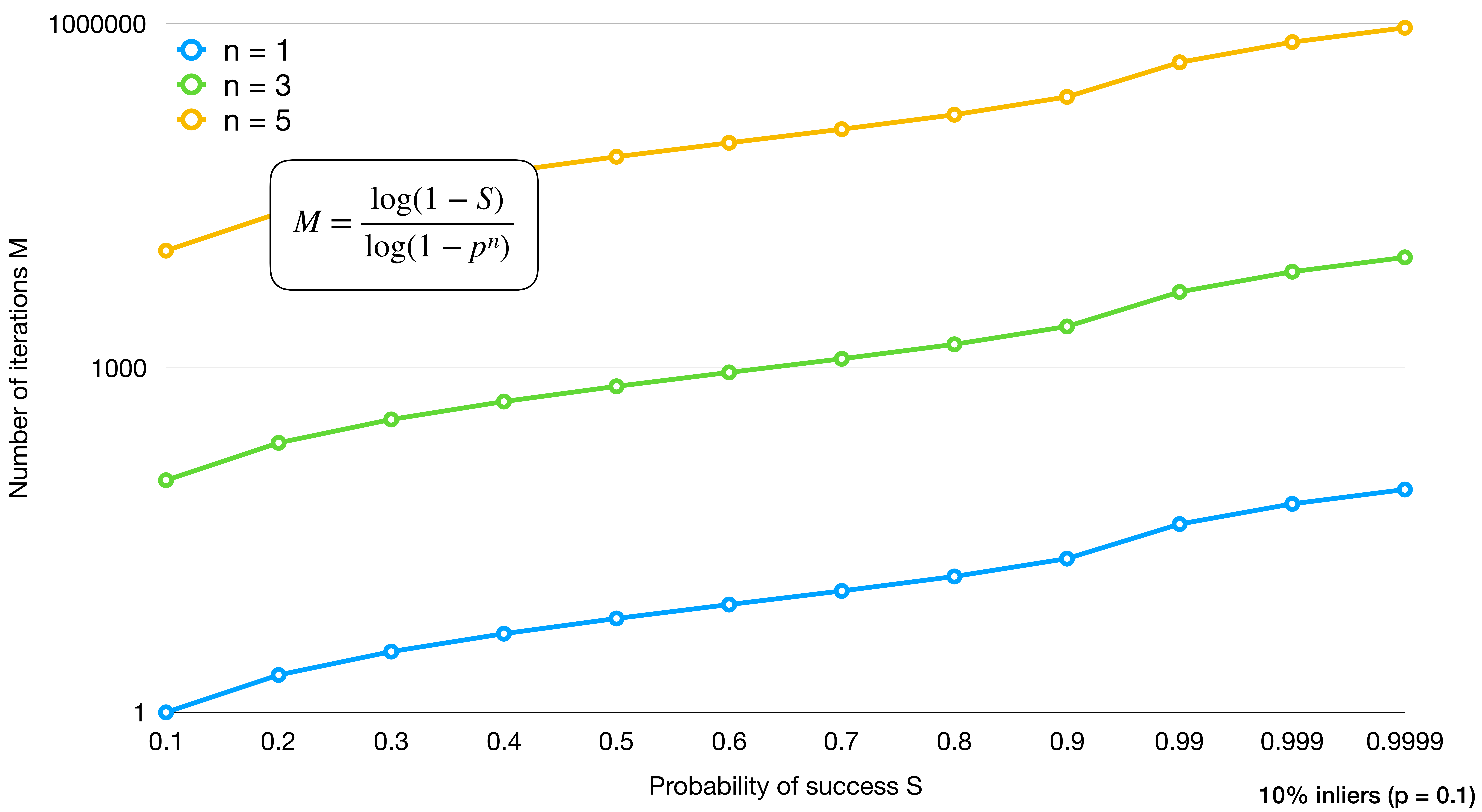


90
67.5
45
22.5
0
n = 1
$M = \frac{\log(1-S)}{\log(1-p^n)}$
Number of iterations M
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.99
0.999
0.9999
Probability of success S
10% inliers (p = 0.1)

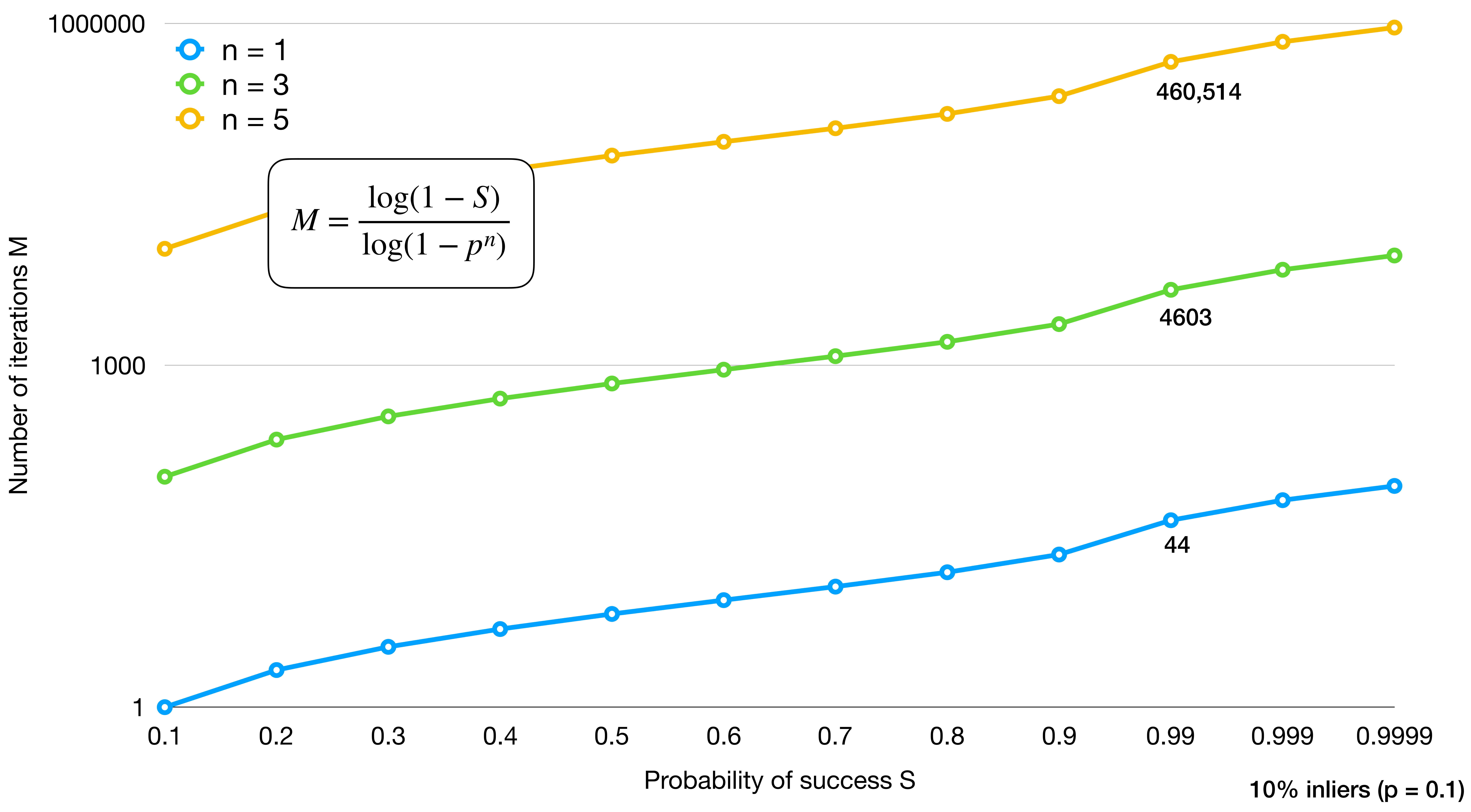


10000
7500
5000
2500
0
Number of iterations M
n = 1
n = 3
$M = \frac{\log(1 - S)}{\log(1 - p^n)}$
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.99
0.999
0.9999
Probability of success S
10% inliers (p = 0.1)


1000000
750000
500000
250000
0
Number of iterations M
n = 1
n = 3
n = 5
$M = \frac{\log(1 - S)}{\log(1 - p^n)}$
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.99
0.999
0.9999
Probability of success S
10% inliers (p = 0.1)



n = 1
n = 3
n = 5
$$M = \frac{\log(1 - S)}{\log(1 - p^n)}$$
1000000
1000
1
Number of iterations M
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.99
0.999
0.9999
Probability of success S
10% inliers (p = 0.1)



1000000
1000
1
n = 1
n = 3
n = 5
$M = \frac{\log(1 - S)}{\log(1 - p^n)}$
460,514
4603
44
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.99
0.999
0.9999
Number of iterations M
Probability of success S
10% inliers (p = 0.1)

Geometric Transformations

Review — homogeneous coordinates

$$[x, y, 1] \sim [\lambda x, \lambda y, \lambda]$$

Review — homogeneous coordinates

Cartesian coordinates

$$[\underbrace{x, y}, 1] \sim [\lambda x, \lambda y, \lambda]$$

Review — homogeneous coordinates

Cartesian coordinates

$$[\underbrace{x, y}, 1] \sim [\lambda x, \lambda y, \lambda]$$

Point at infinity $[x, y, 0]$

Review — homogeneous coordinates

Cartesian coordinates

$$[\underbrace{x, y}, 1] \sim [\lambda x, \lambda y, \lambda]$$

Point at infinity $[x, y, 0]$

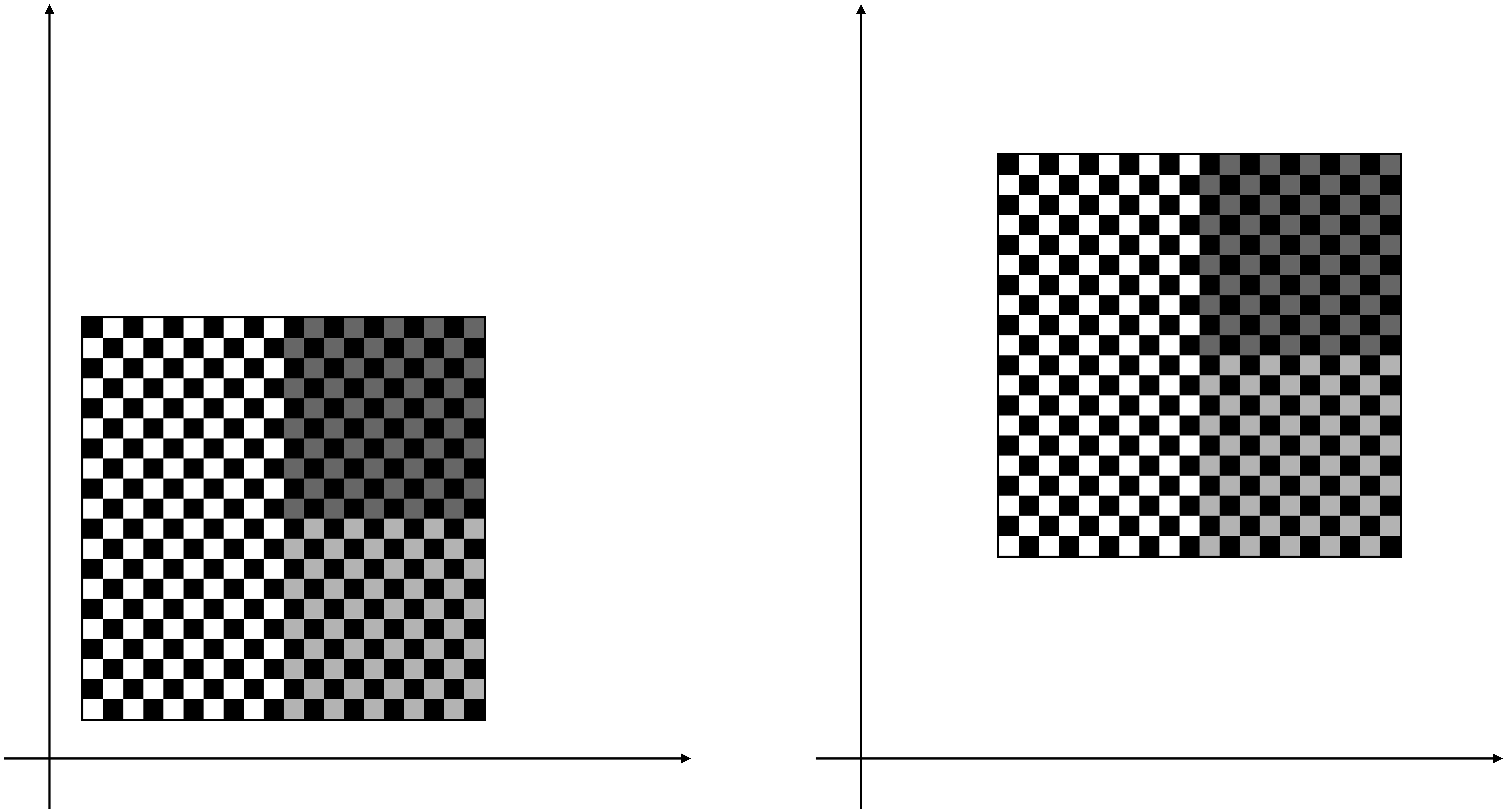
Omitted $[0, 0, 0]$

Review — homogeneous coordinates

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

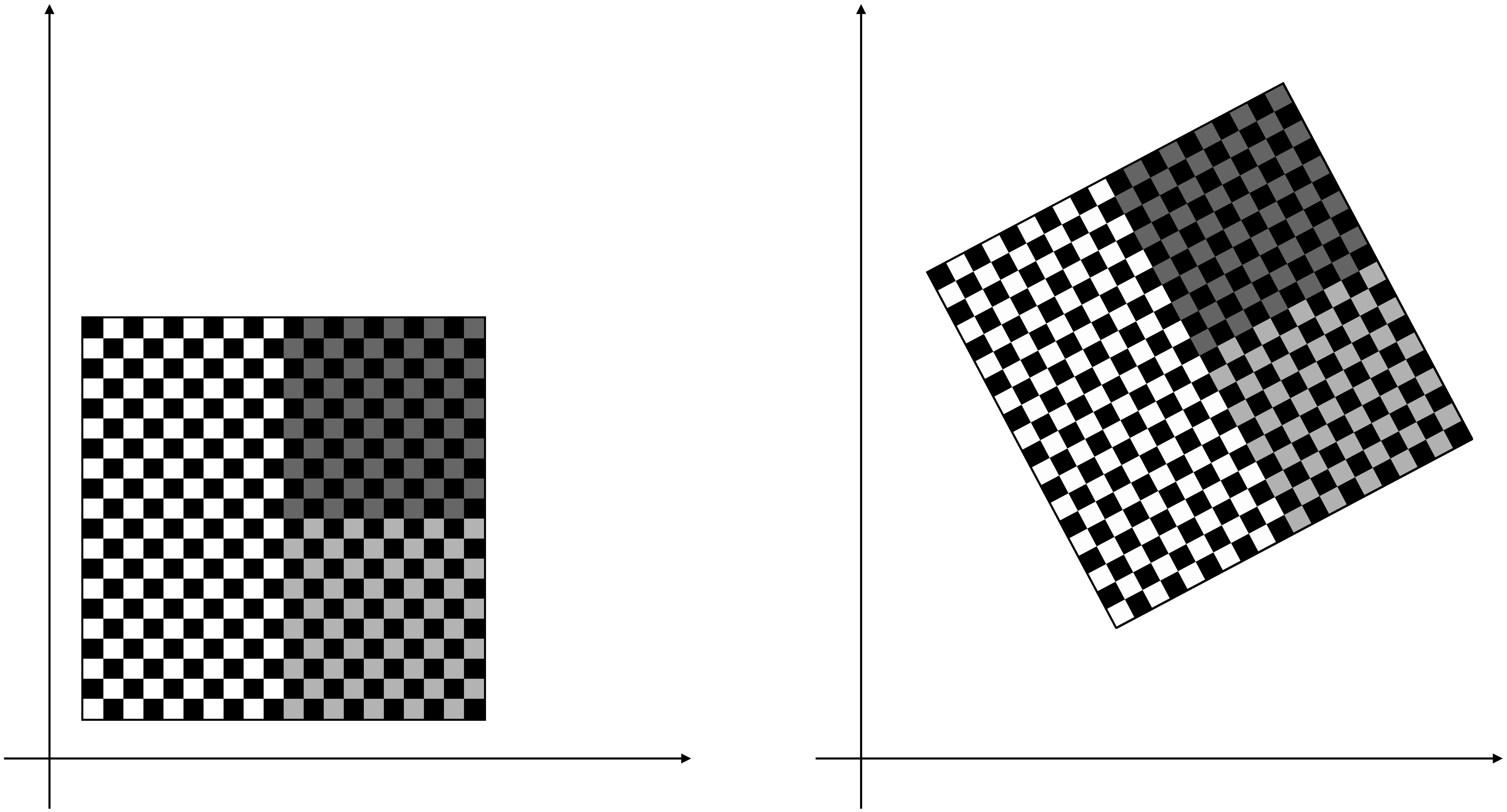
Translation as matrix multiplication

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



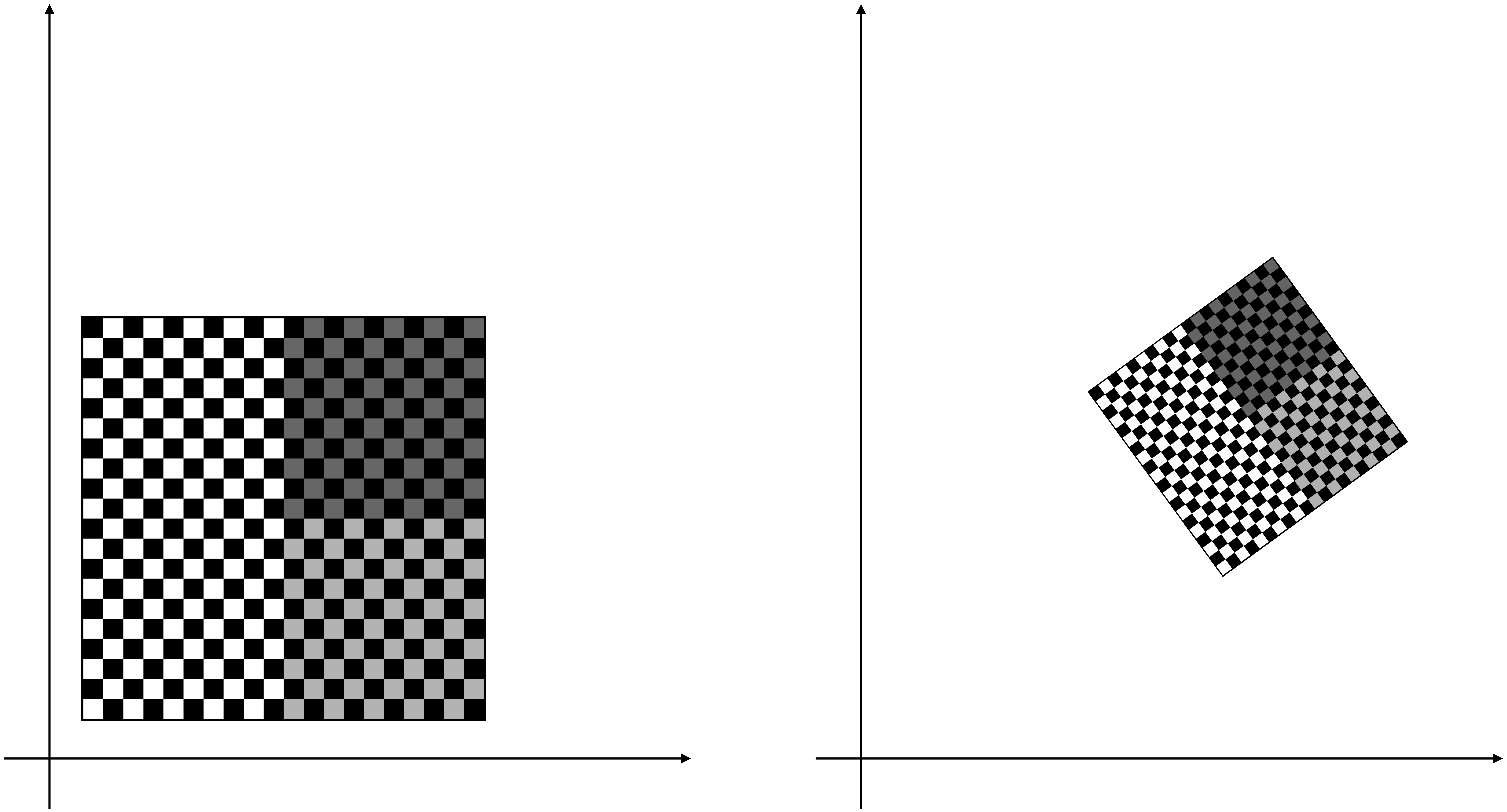
Translation

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



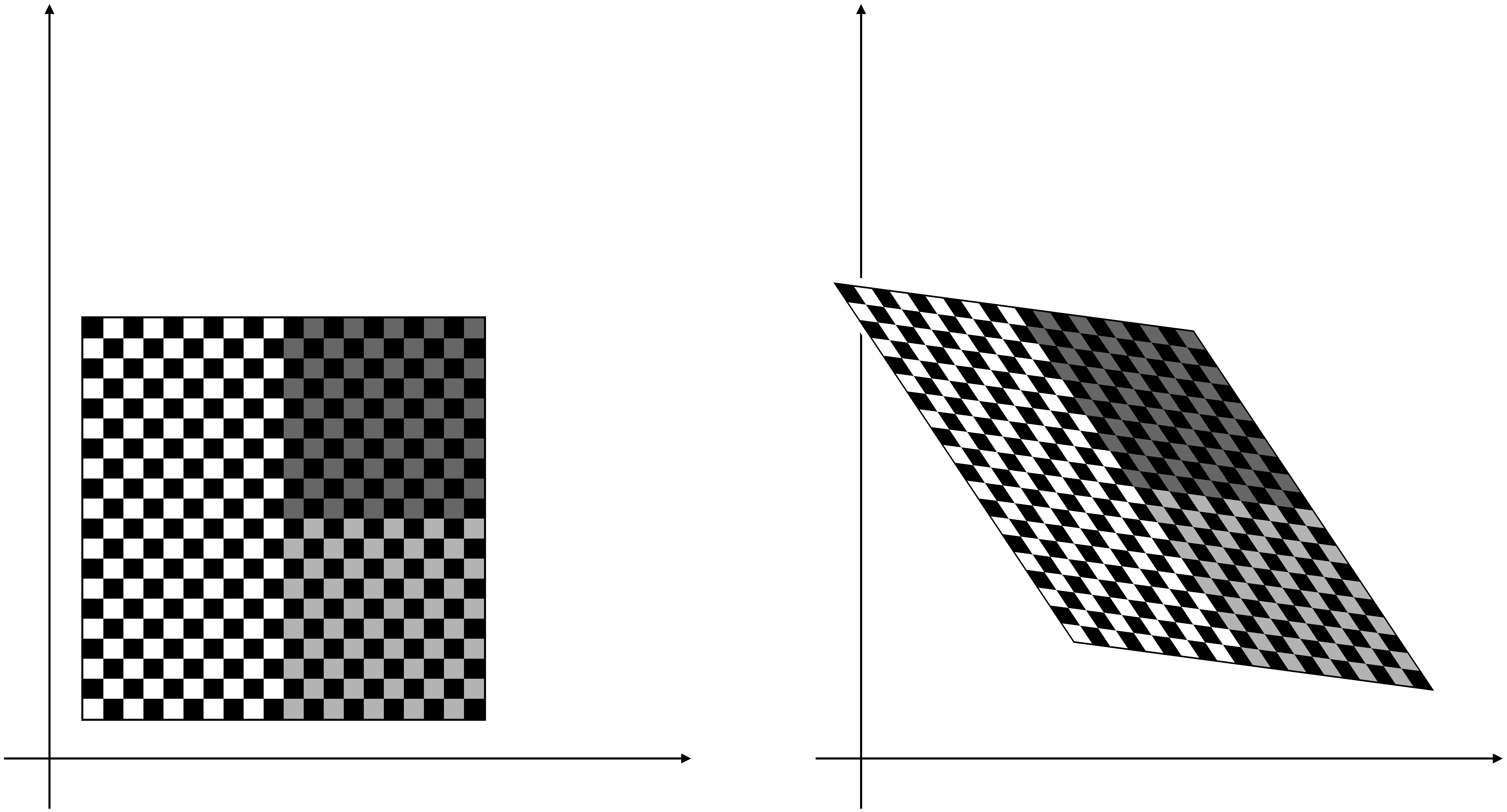
Translation + rotation = Rigid, Euclidean

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



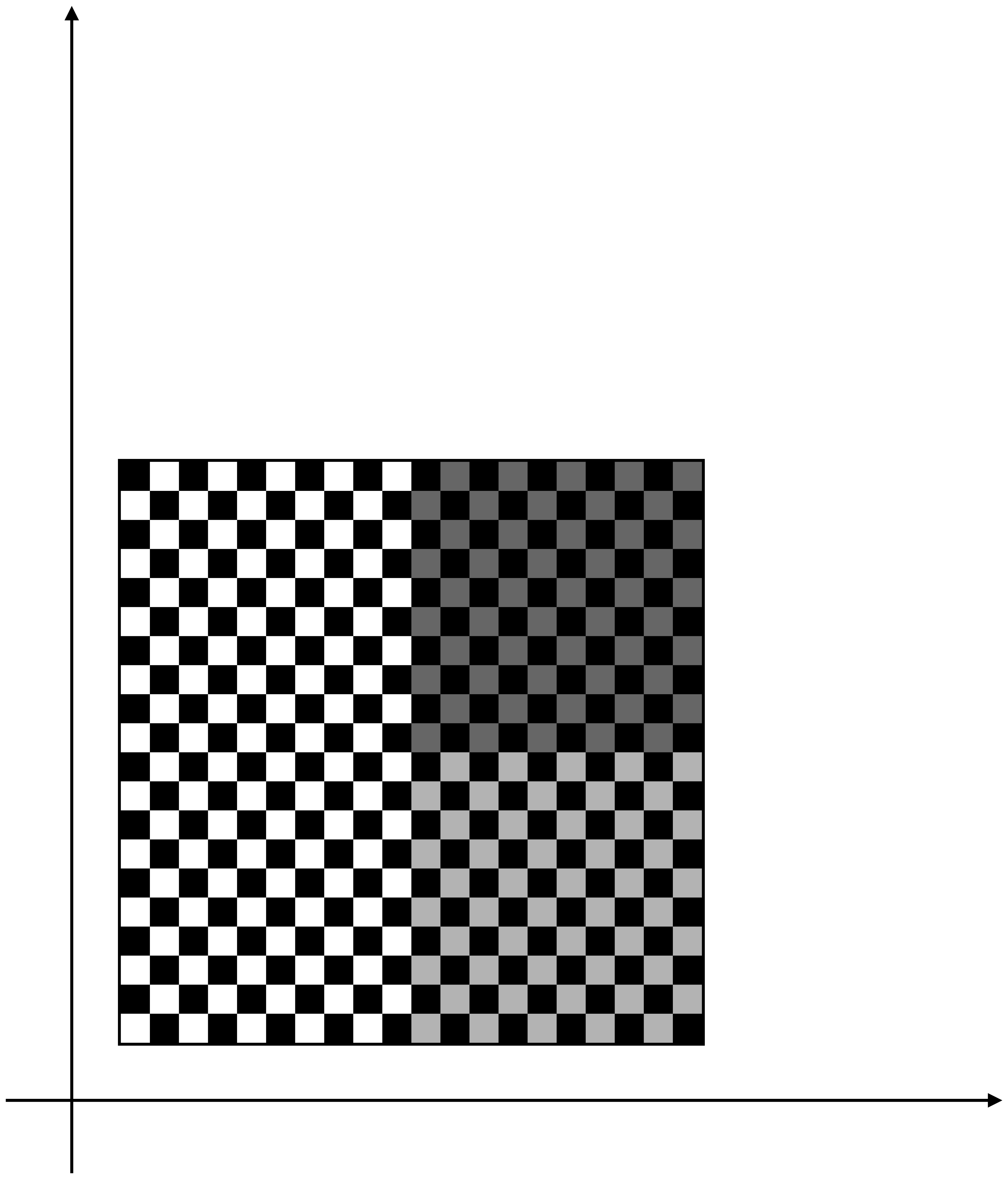
Translation + rotation + uniform scale = Similarity

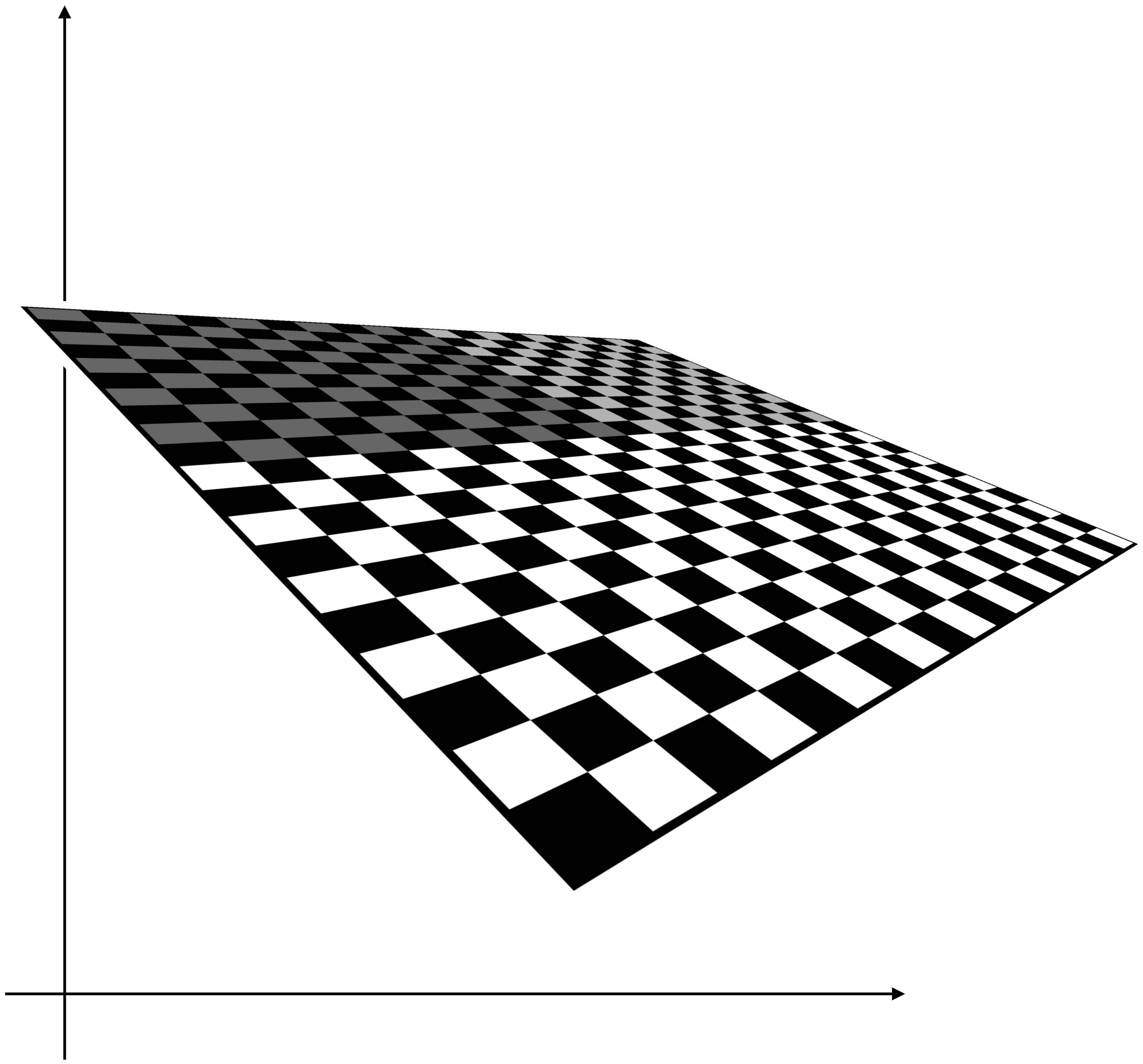
$$\begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



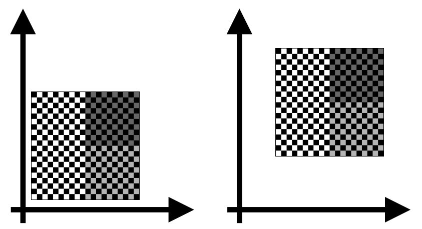
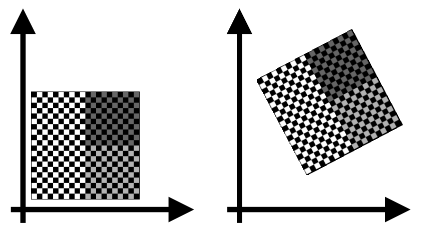
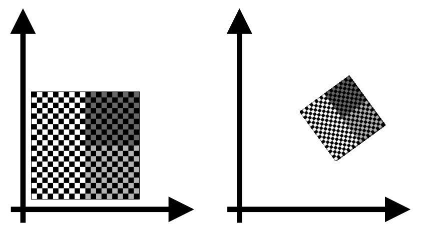
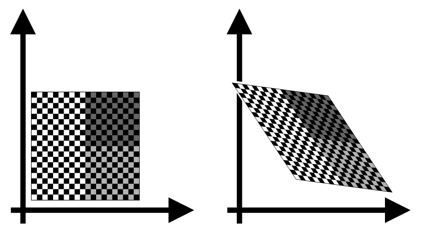
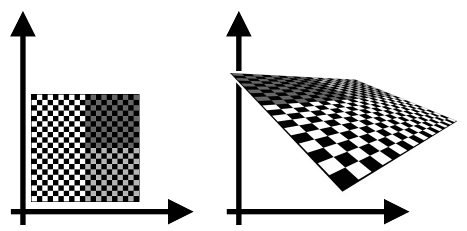
Translation + rotation + scale + shear = Affine

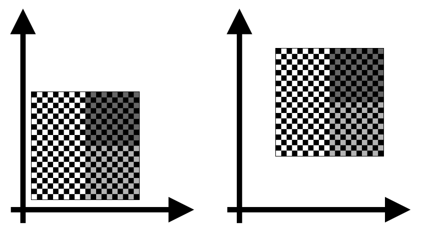
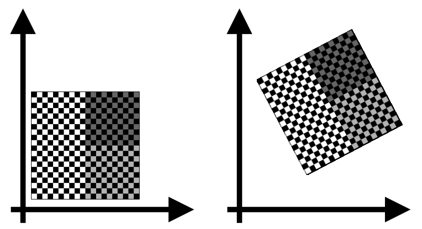
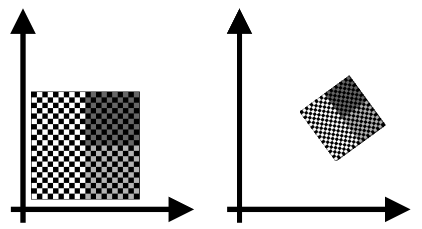
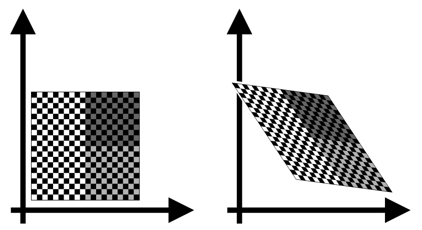
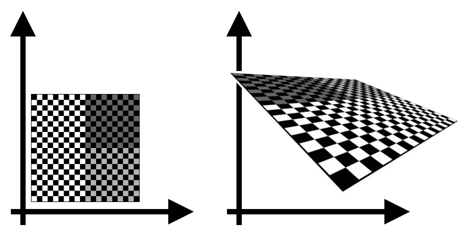
$$\begin{bmatrix} a & b & t_x \\ c & d & t_y \\ e & f & 1 \end{bmatrix}$$

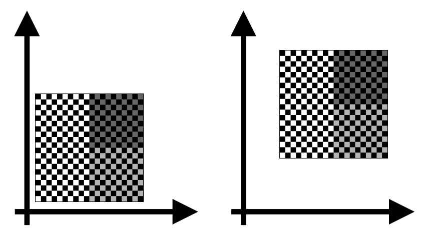
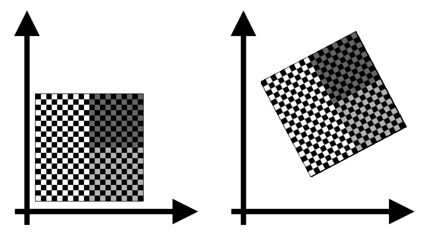
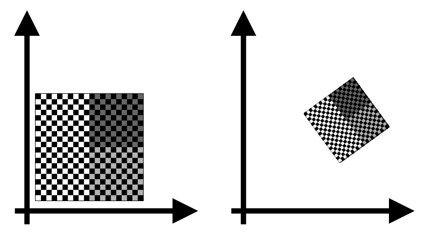
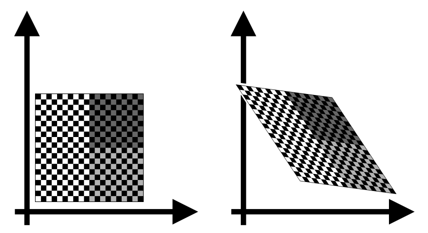
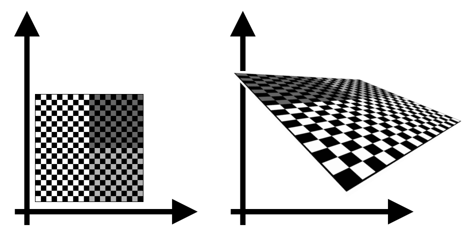


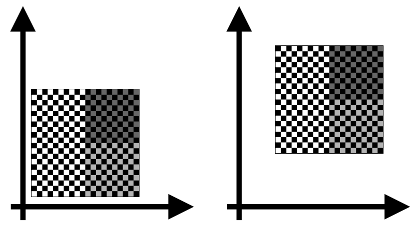
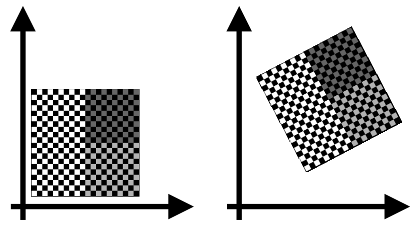
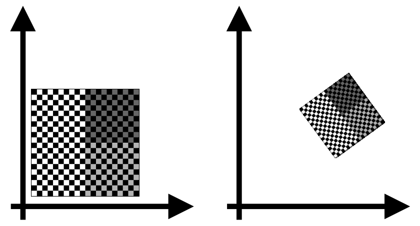
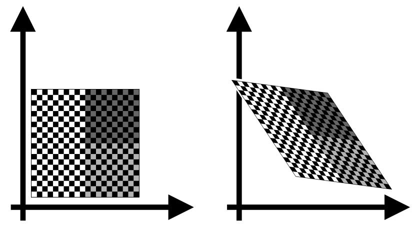
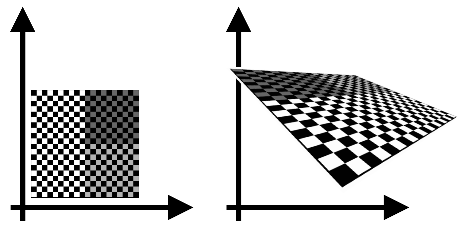


Projective, Perspective, Homography

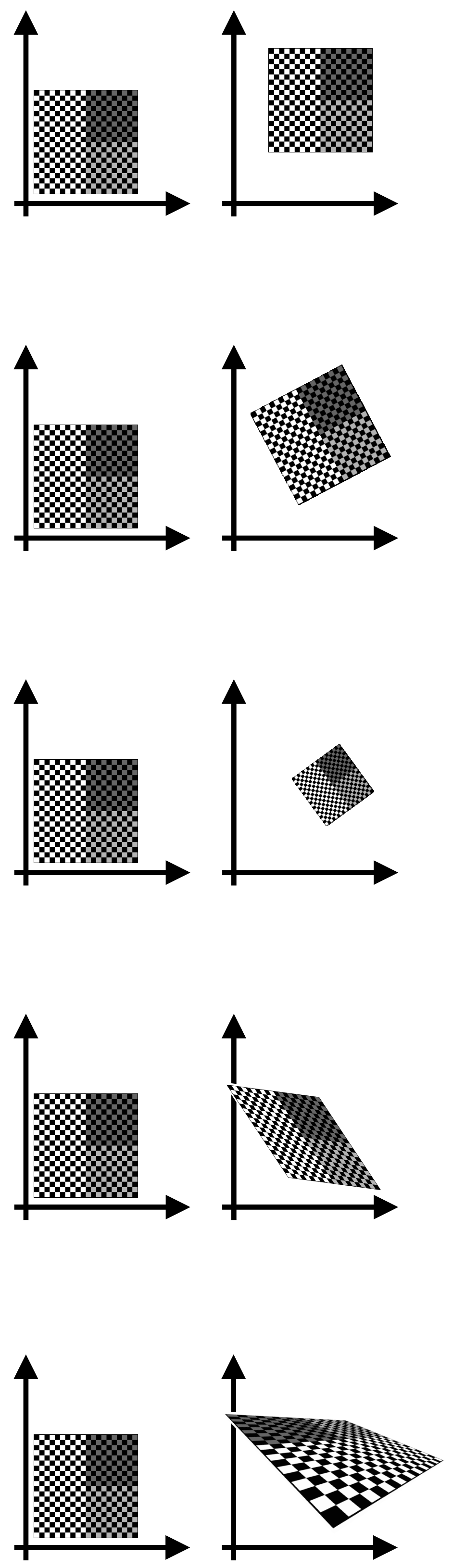
Transformation	Preserves	Degrees of freedom	
Translation			
Rigid			
Similarity			
Affine			
Projective			

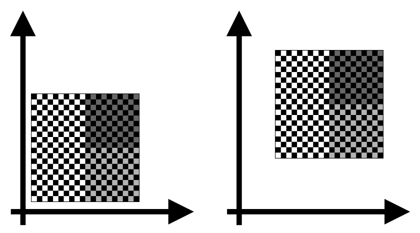
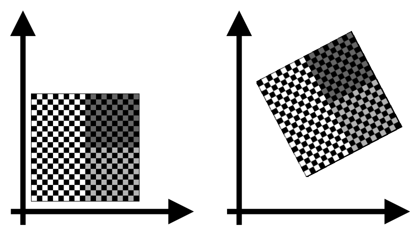
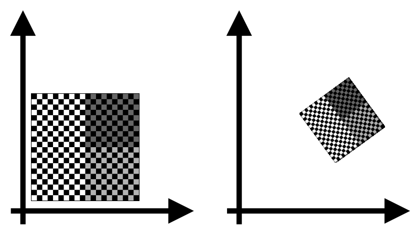
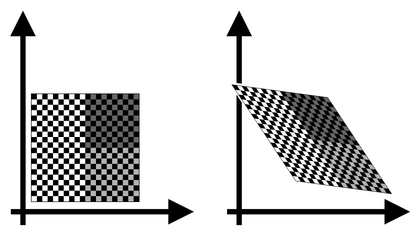
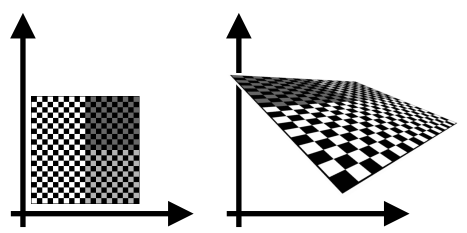
Transformation	Preserves	Degrees of freedom	
Translation			
Rigid			
Similarity			
Affine			
Projective	Straight lines		

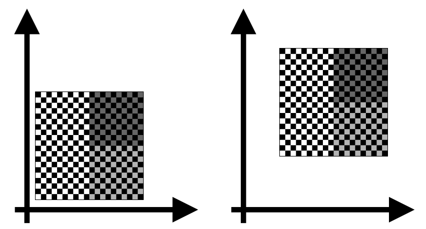
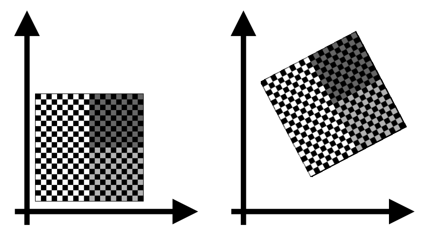
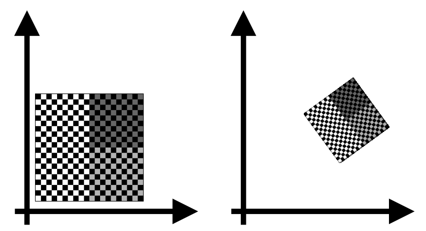
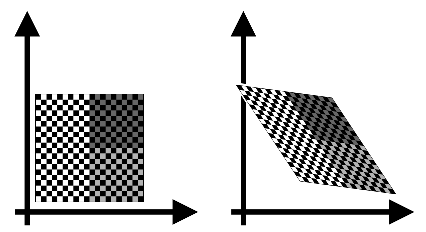
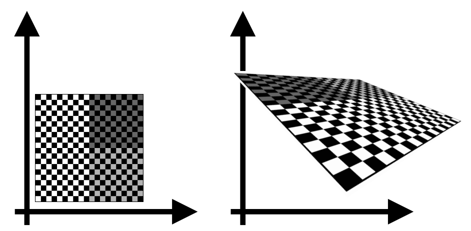
Transformation	Preserves	Degrees of freedom	
Translation			
Rigid			
Similarity			
Affine	Parallelism		
Projective	Straight lines		

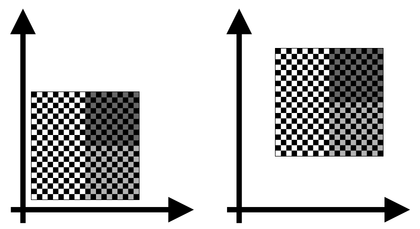
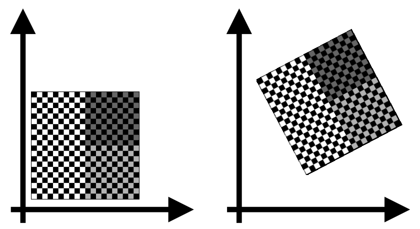
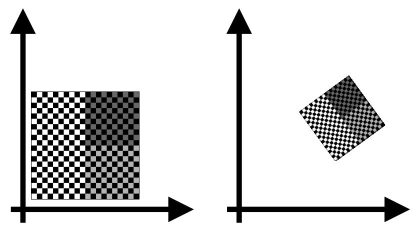
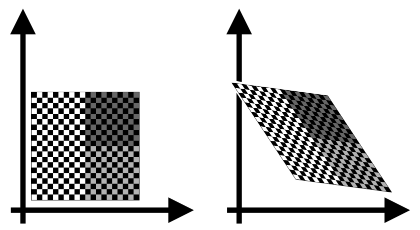
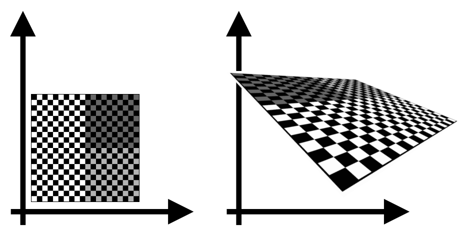
Transformation	Preserves	Degrees of freedom	
Translation			
Rigid			
Similarity	Angles		
Affine	Parallelism		
Projective	Straight lines		

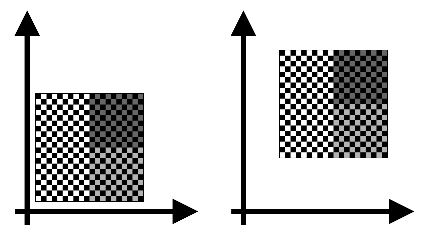
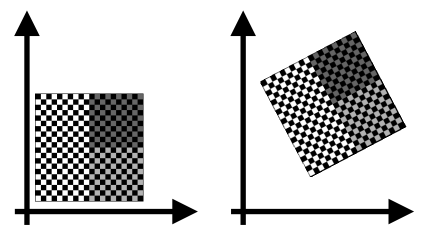
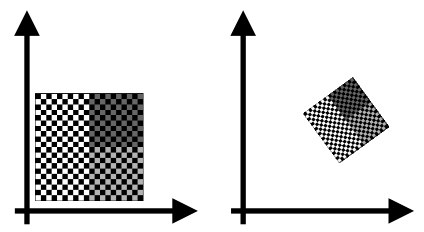
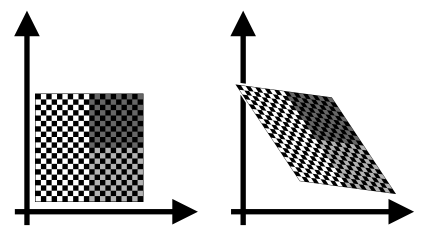
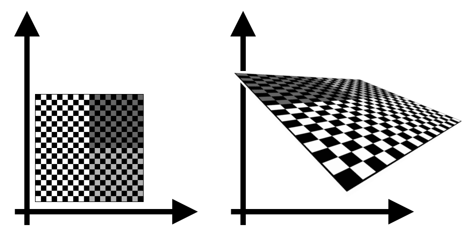
Transformation	Preserves	Degrees of freedom
Translation		
Rigid	Length	
Similarity	Angles	
Affine	Parallelism	
Projective	Straight lines	

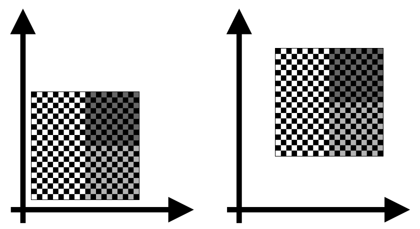
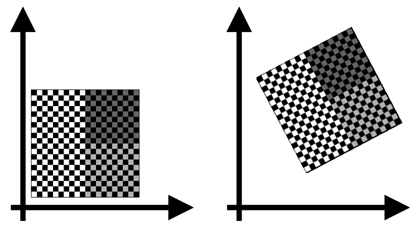
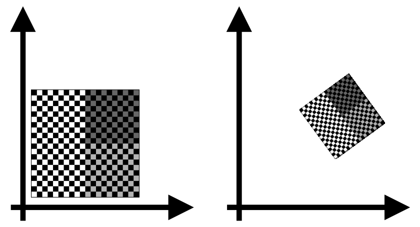
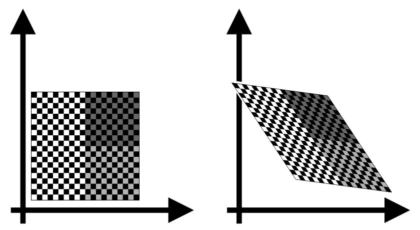
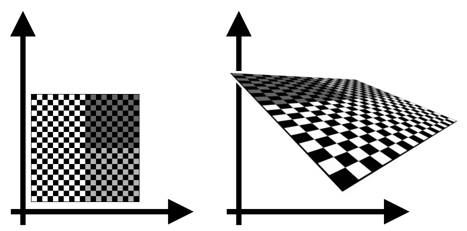


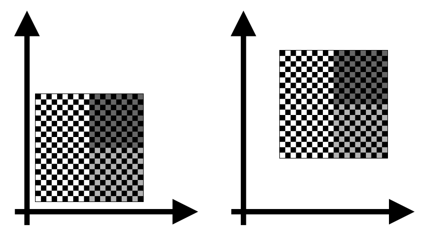
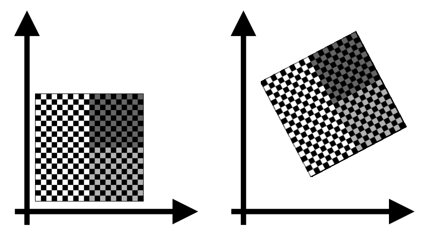
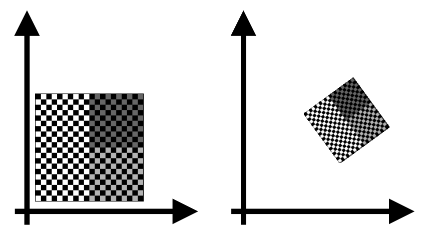
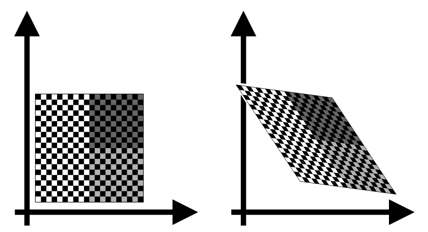
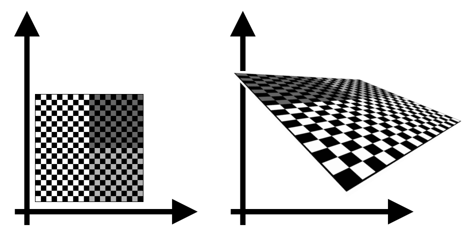
Transformation	Preserves	Degrees of freedom	
Translation	Orientation		
Rigid	Length		
Similarity	Angles		
Affine	Parallelism		
Projective	Straight lines		

Transformation	Preserves	Degrees of freedom	
Translation	Orientation	2	
Rigid	Length		
Similarity	Angles		
Affine	Parallelism		
Projective	Straight lines		

Transformation	Preserves	Degrees of freedom	
Translation	Orientation	2	
Rigid	Length	3	
Similarity	Angles		
Affine	Parallelism		
Projective	Straight lines		

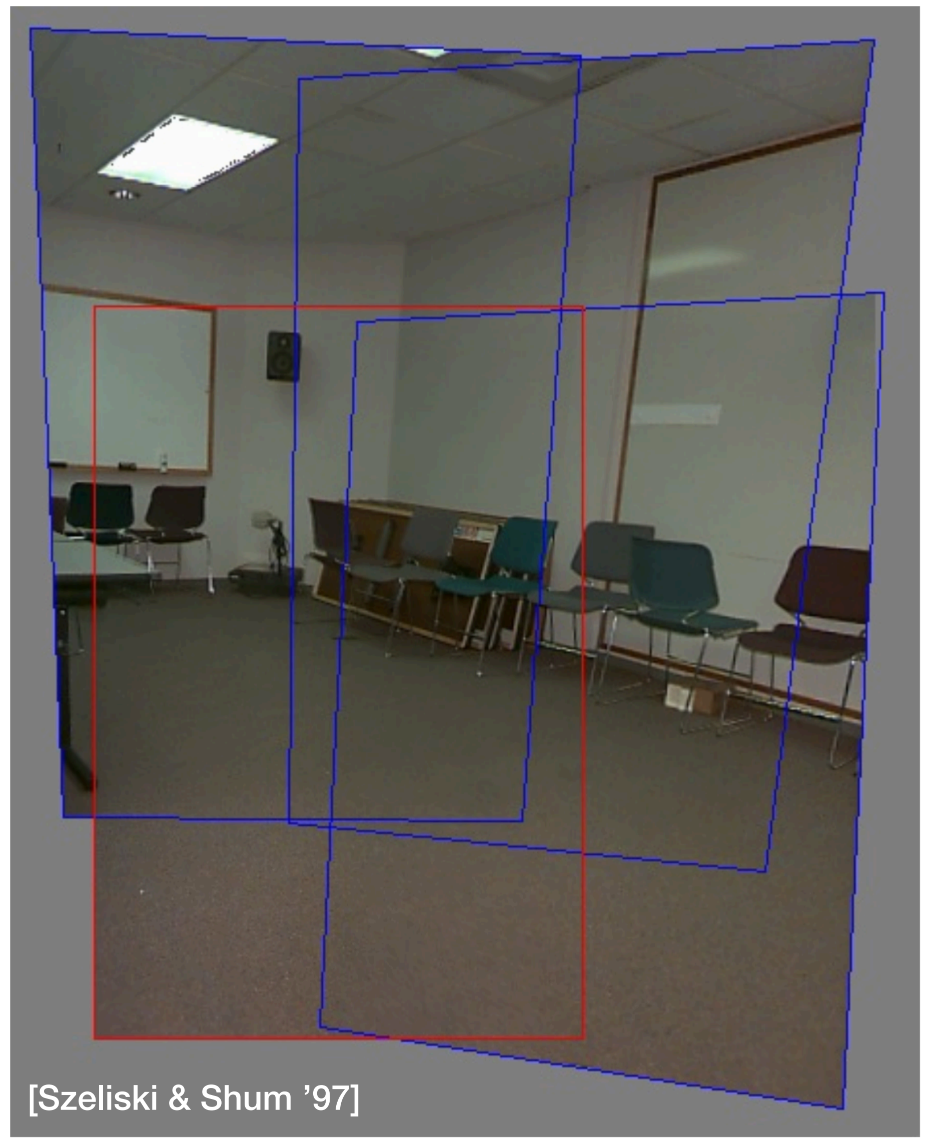
Transformation	Preserves	Degrees of freedom	
Translation	Orientation	2	
Rigid	Length	3	
Similarity	Angles	4	
Affine	Parallelism		
Projective	Straight lines		

Transformation	Preserves	Degrees of freedom	
Translation	Orientation	2	
Rigid	Length	3	
Similarity	Angles	4	
Affine	Parallelism	6	
Projective	Straight lines		

Transformation	Preserves	Degrees of freedom	
Translation	Orientation	2	
Rigid	Length	3	
Similarity	Angles	4	
Affine	Parallelism	6	
Projective	Straight lines	8	

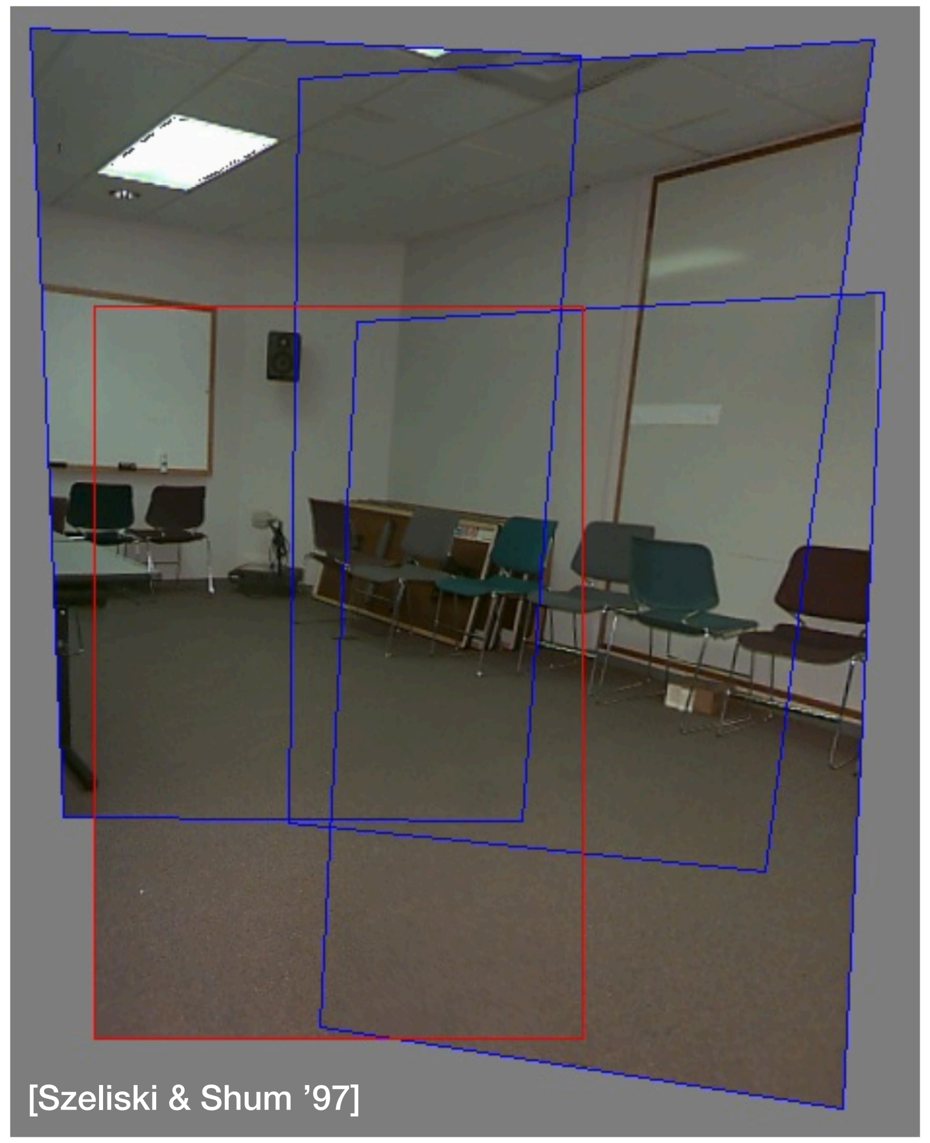
2D Homographies

2D Homographies

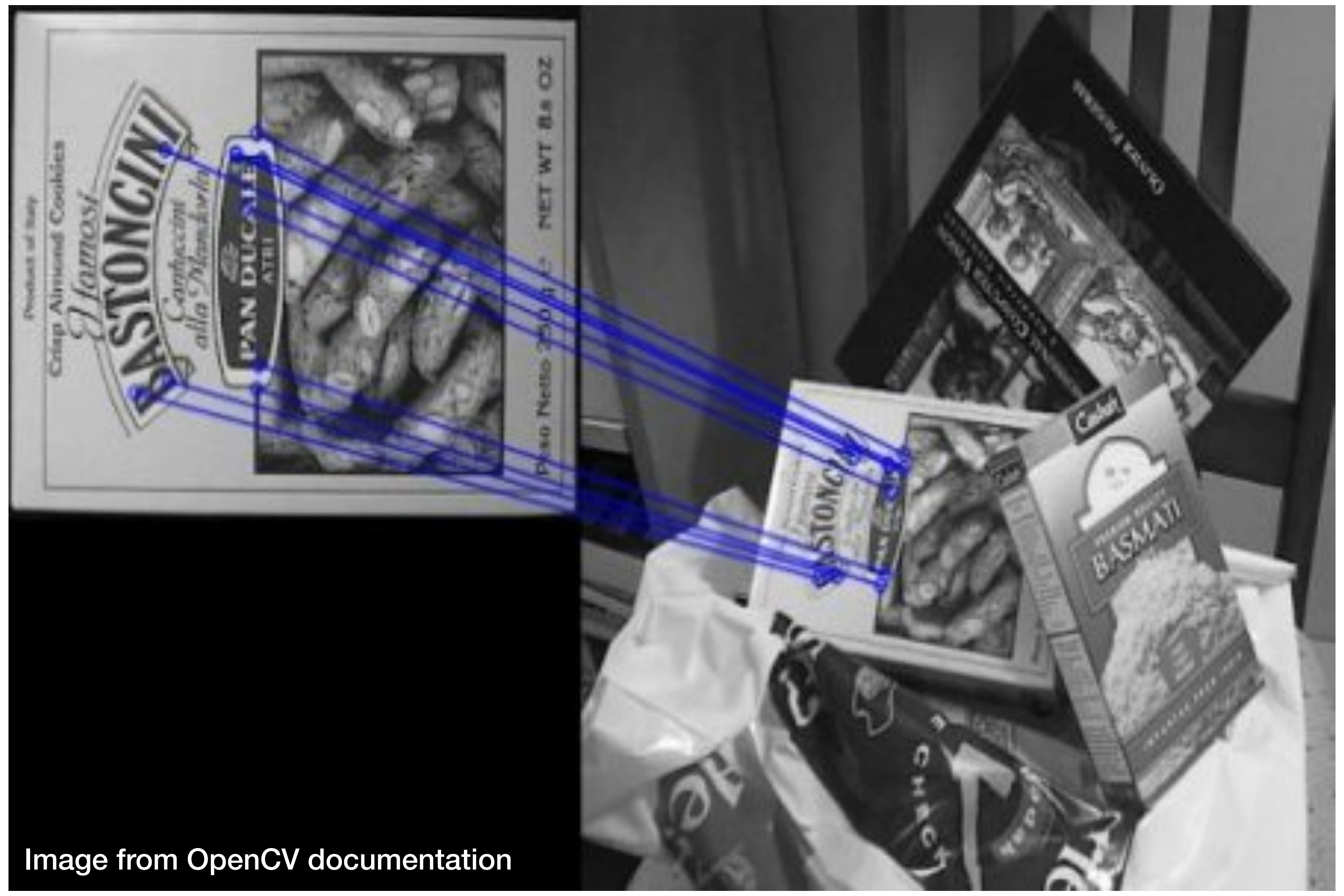


connect between 3D scenes
viewed by a rotating camera

2D Homographies

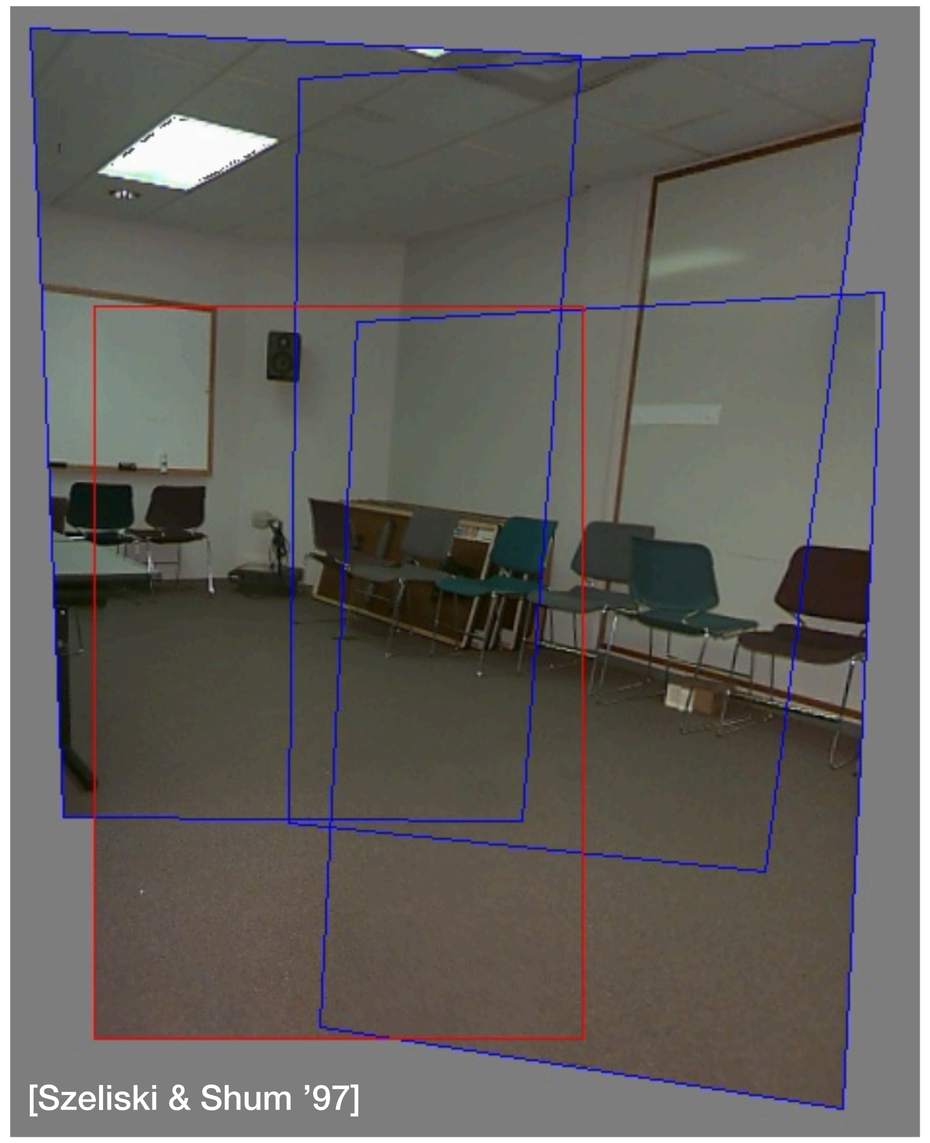


connect between 3D scenes
viewed by a rotating camera

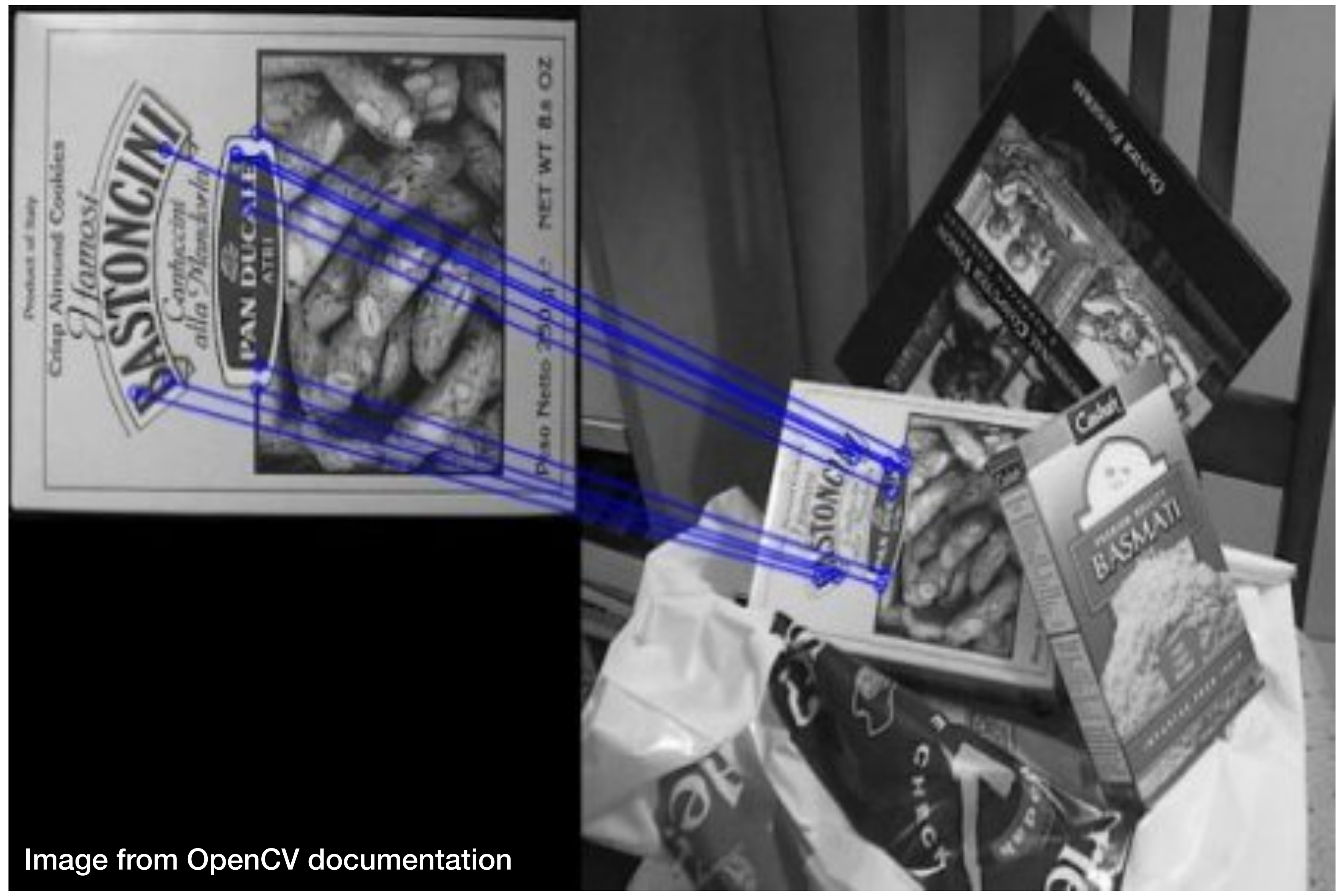


connect between planes seen by different cameras

2D Homographies



connect between 3D scenes viewed by a rotating camera



connect between planes seen by different cameras

We'll revisit these in a later class (structure from motion, scene building)

Non-linear

Non-linear

Quadratic

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix}$$

Or higher orders

Polynomial

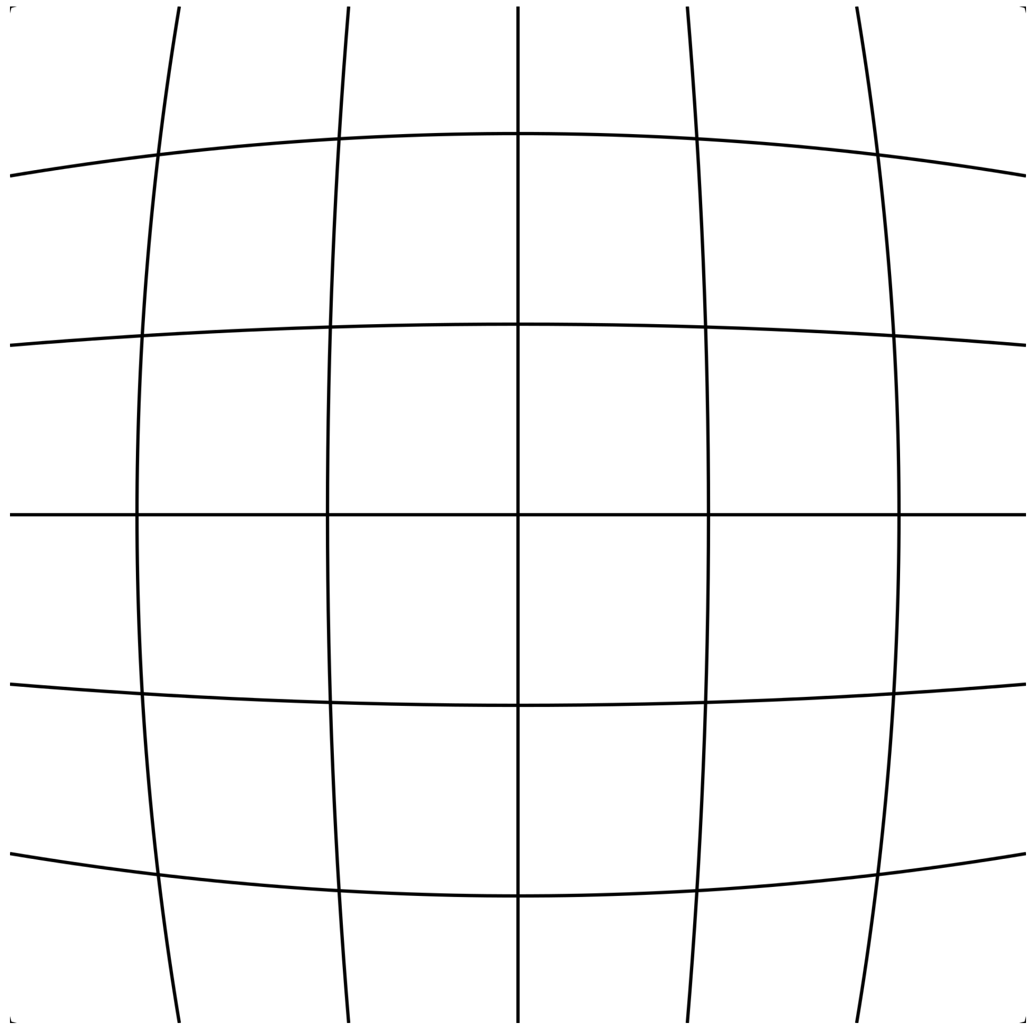
Non-linear

Quadratic

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \begin{bmatrix}a_0 & a_1 & a_2 & a_3 & a_4 & a_5\\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5\end{bmatrix}\begin{bmatrix}1\\ x\\ y\\ xy\\ x^2\\ y^2\end{bmatrix}$$

Or higher orders

Polynomial



Radial

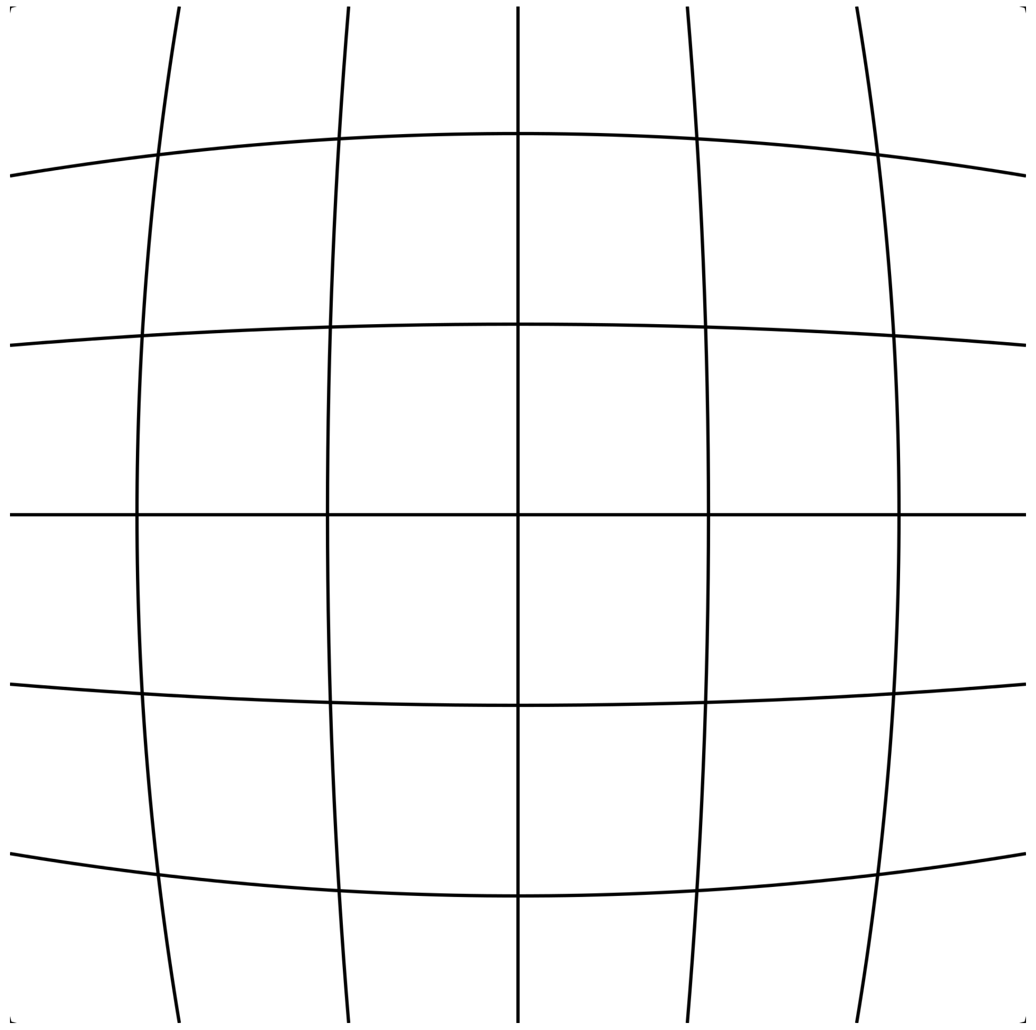
Non-linear

Quadratic

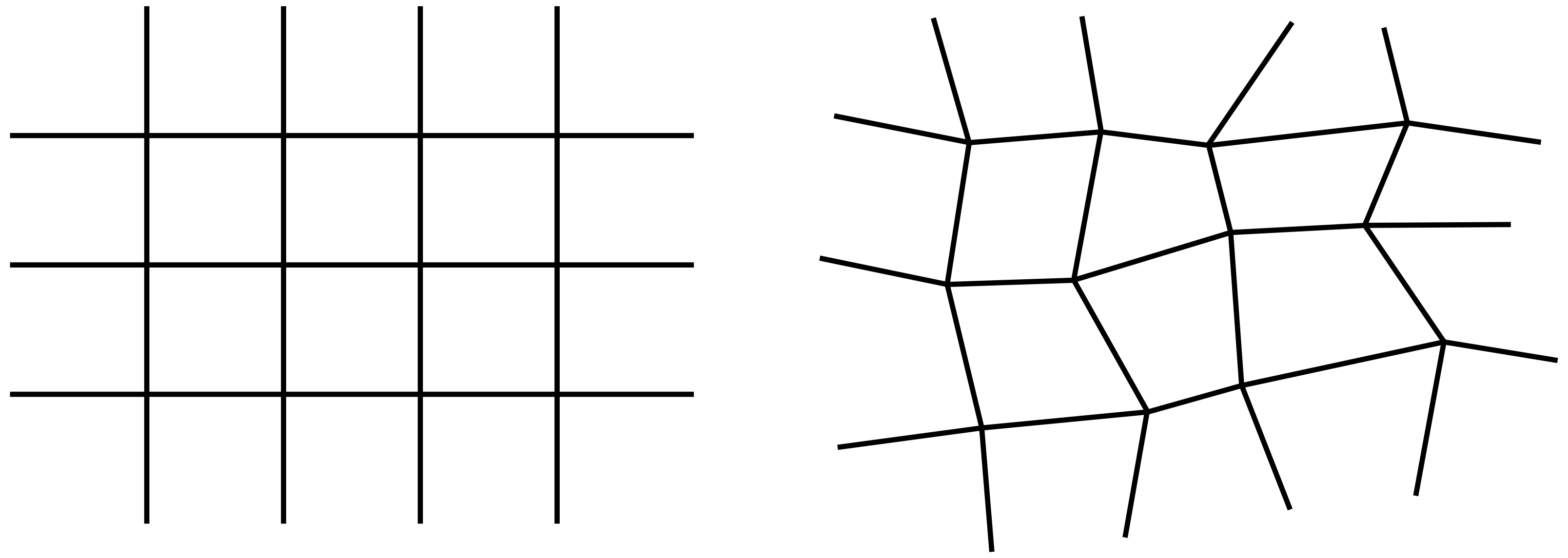
$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ xy \\ x^2 \\ y^2 \end{bmatrix}$$

Or higher orders

Polynomial

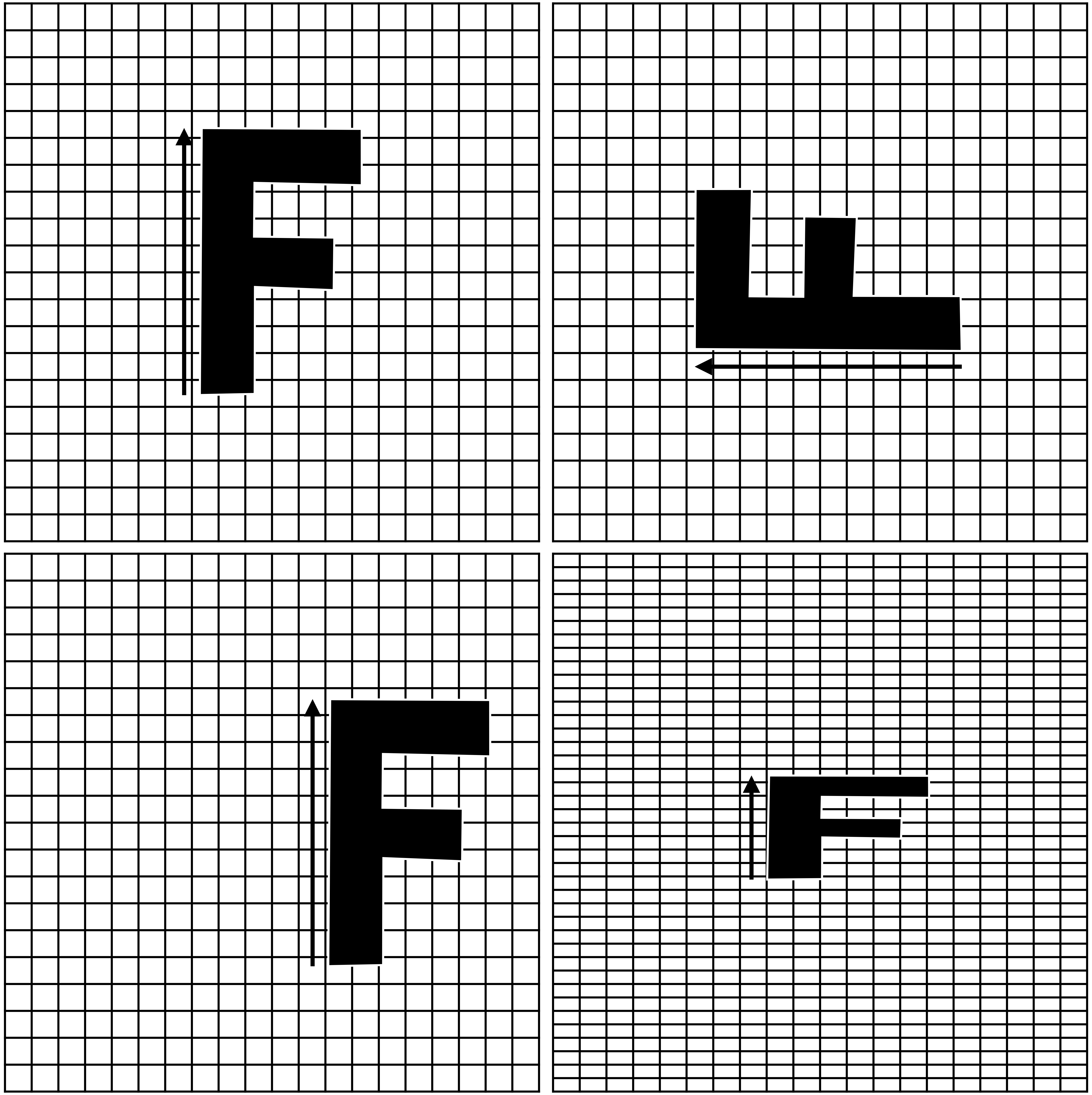


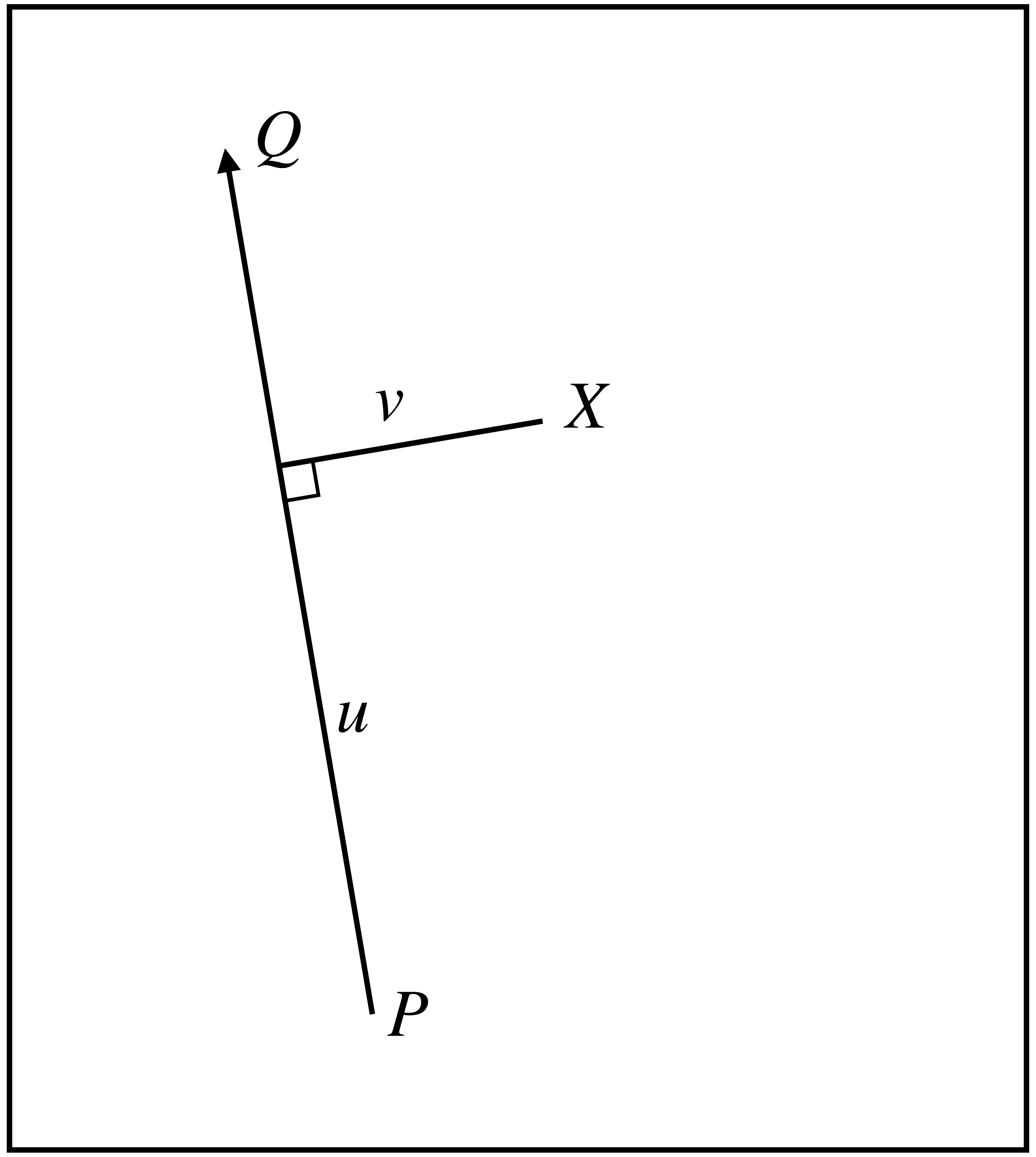
Radial



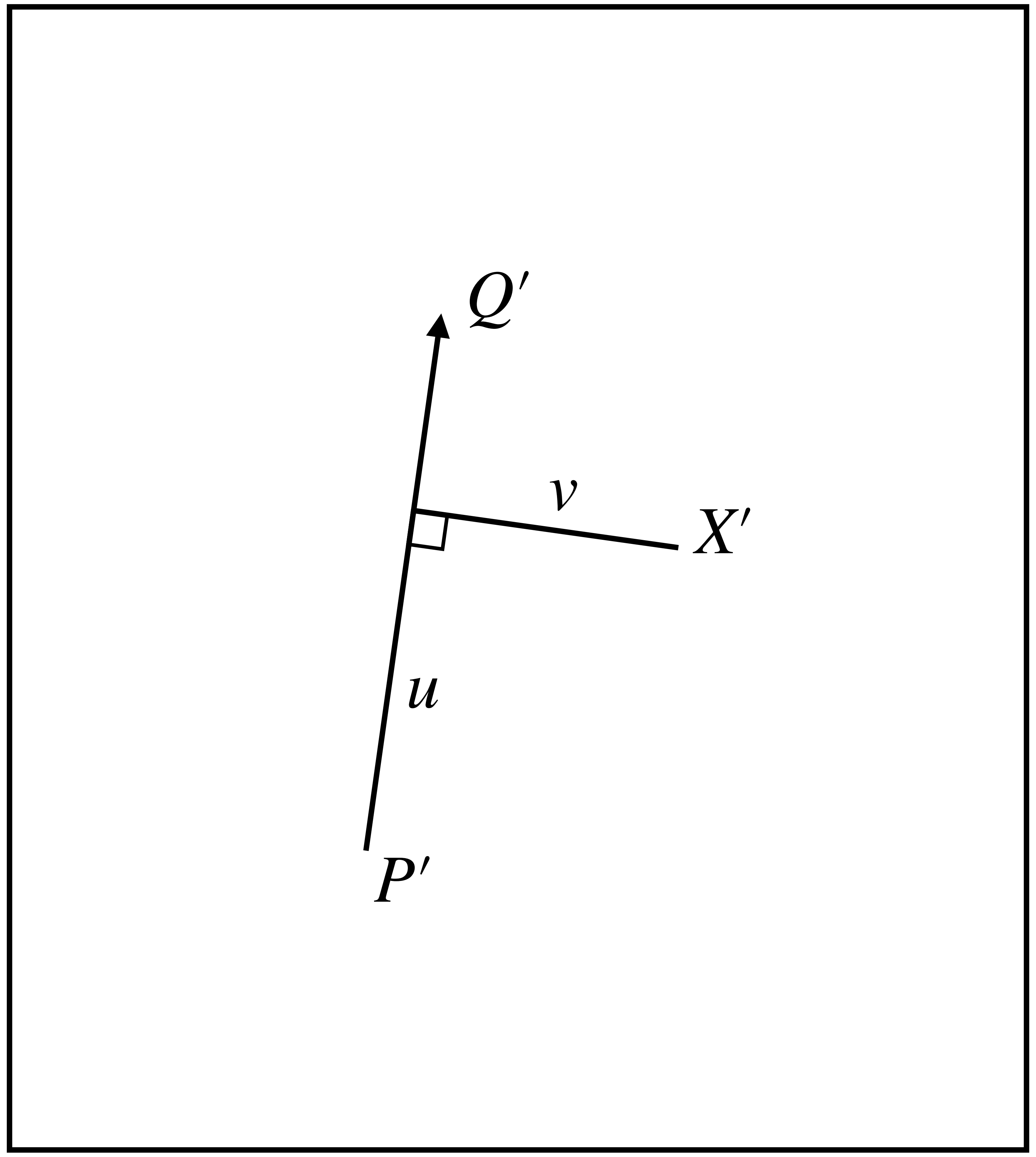
Deformation fields

Beier & Neely ‘92

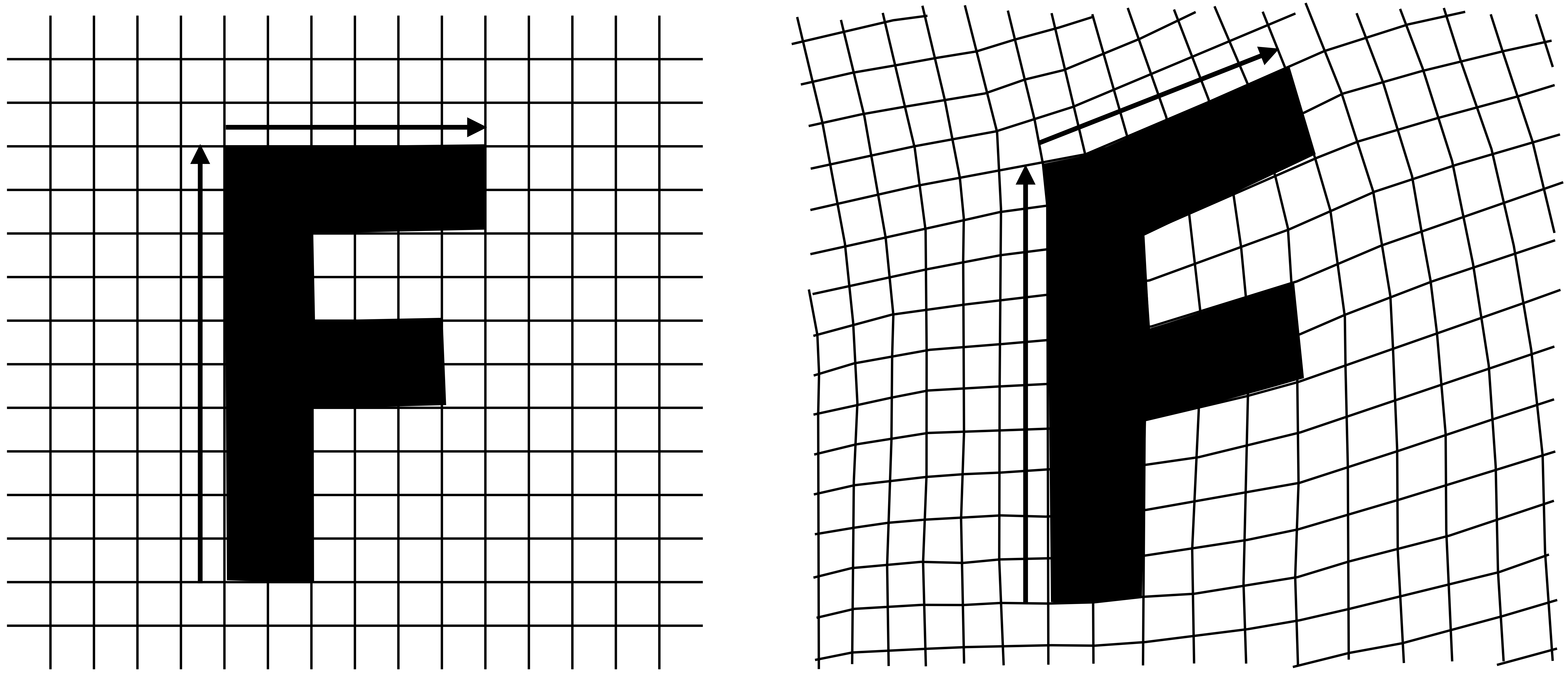




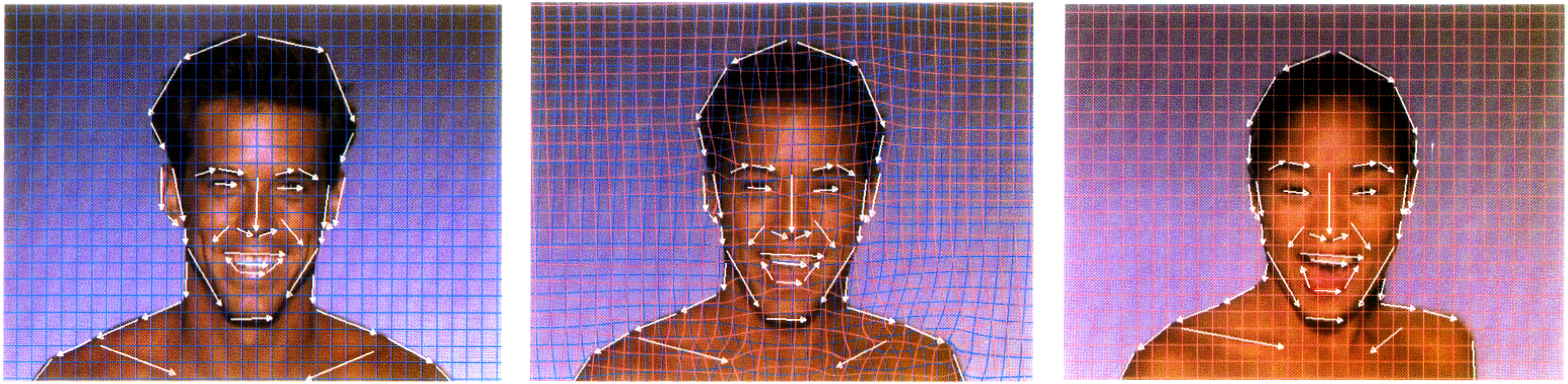
Destination Image

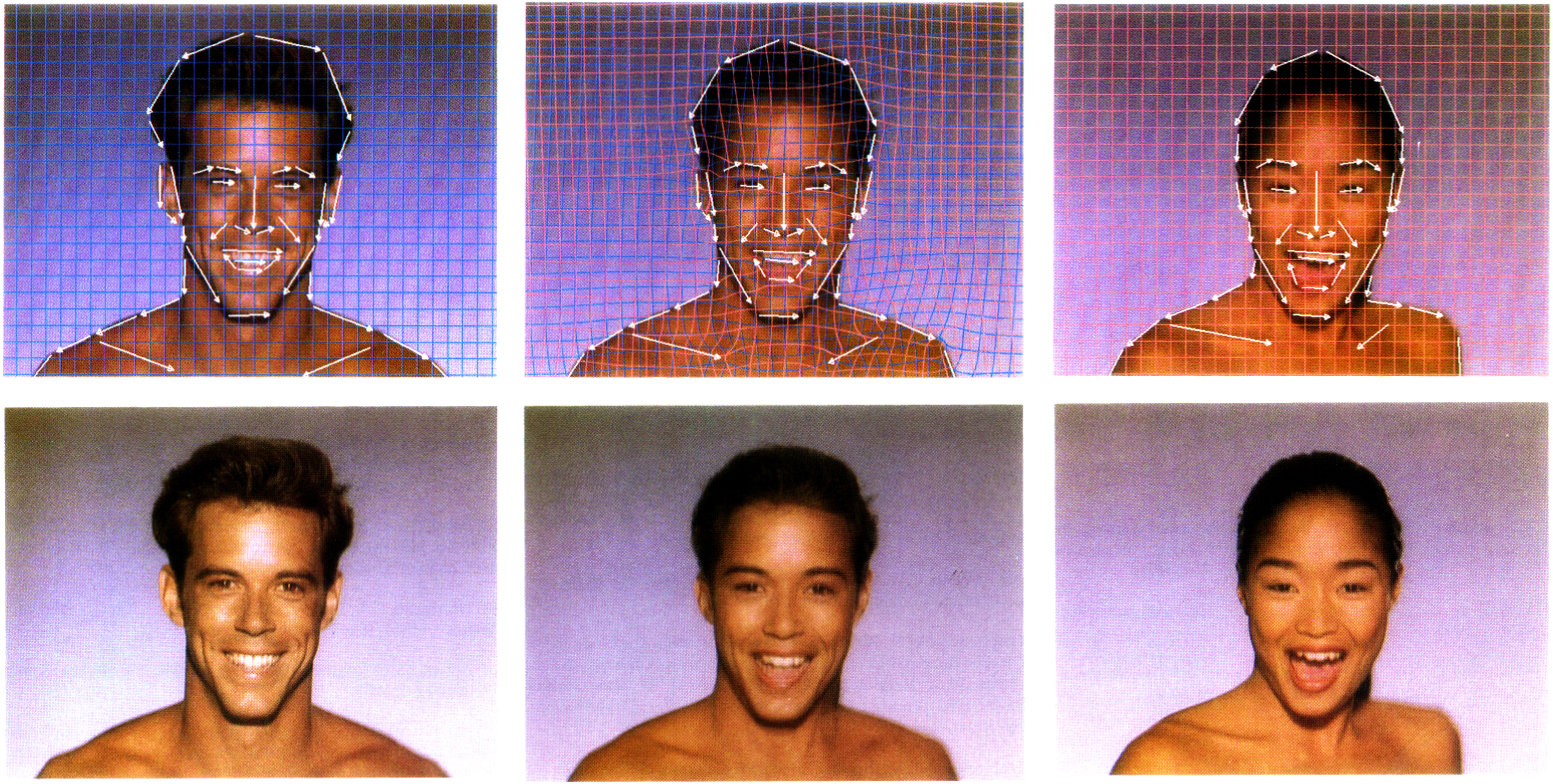


Source Image



$$\text{weight} = \left(\frac{\text{line\_len}^{p}}{a + \text{pixel\_dist}} \right)^{b}$$



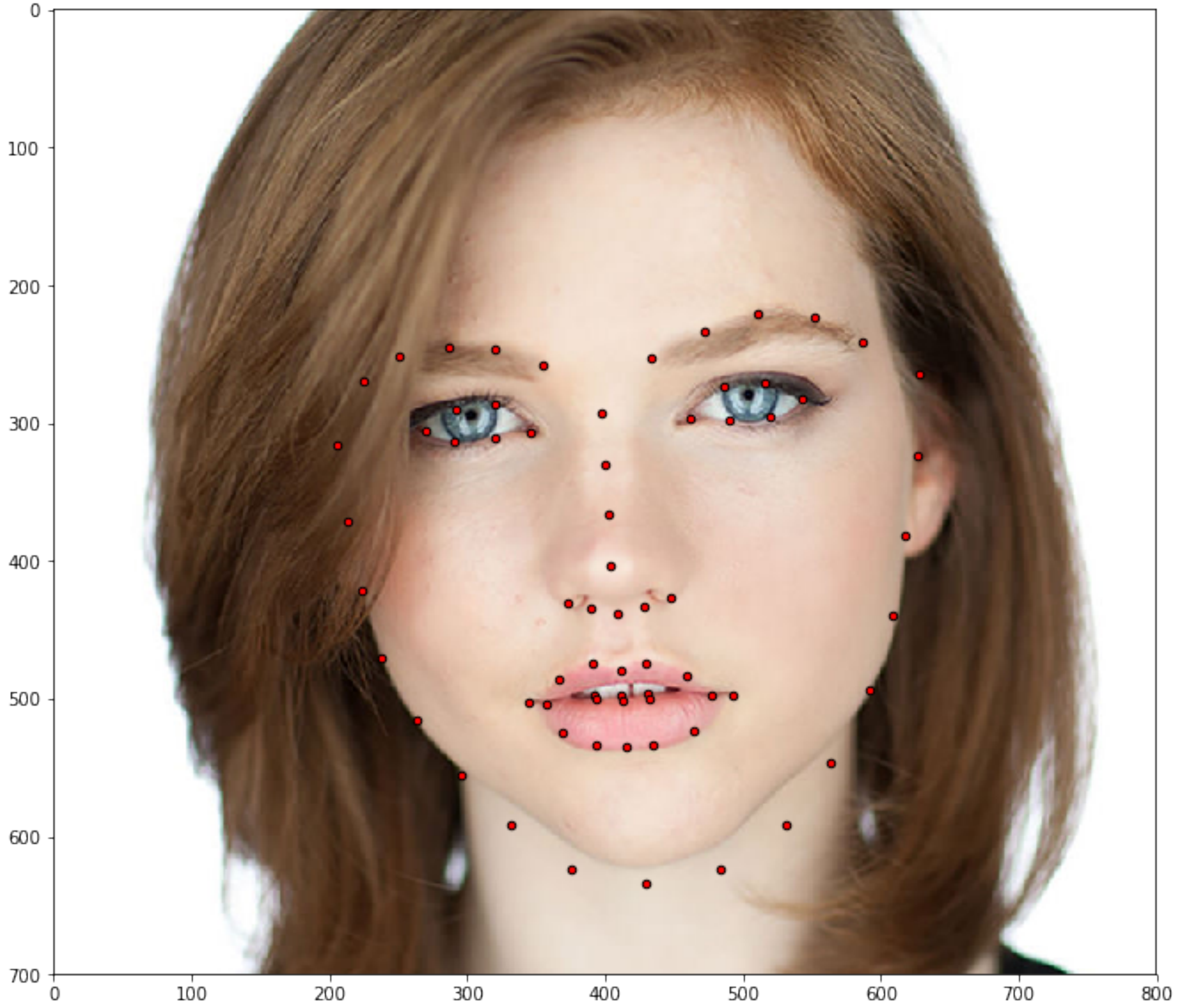


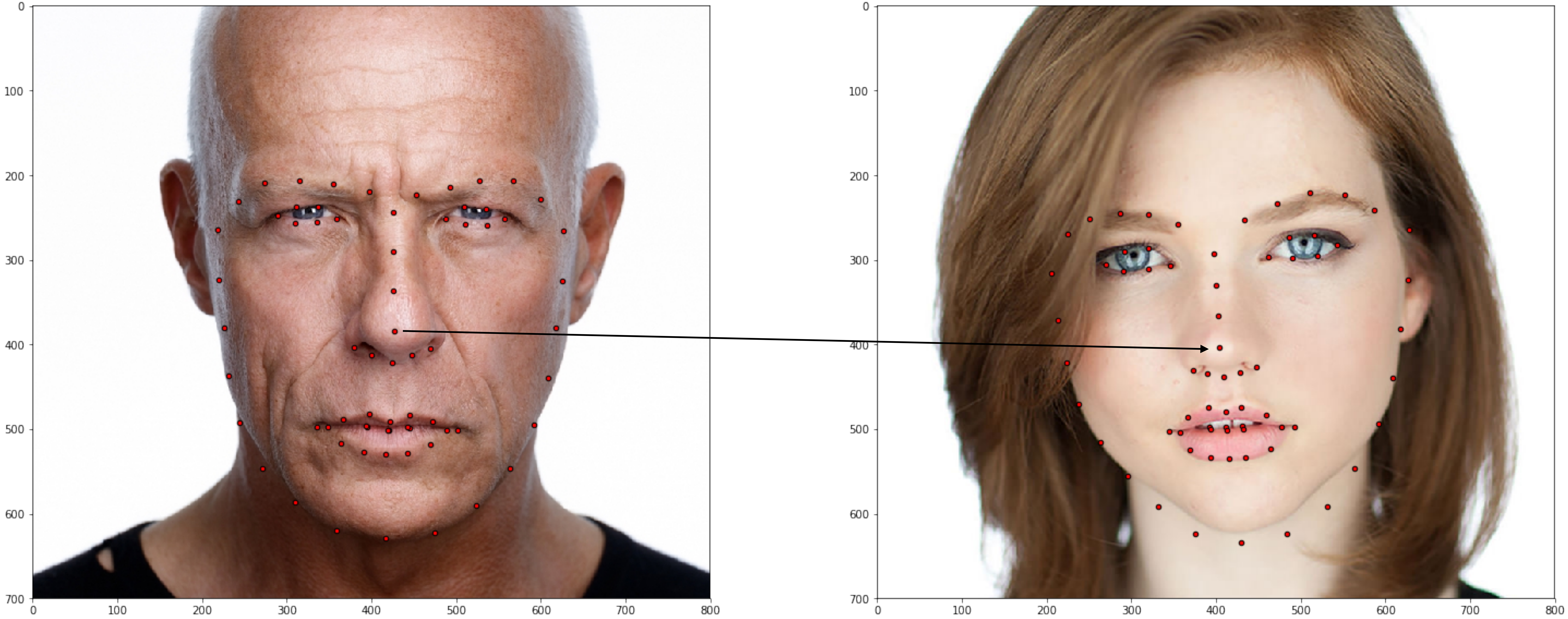
Describe other ways to specify
dense correspondences

Hypothesize regarding their
properties and differences

Assignment 2


0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700
800


0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700
800


0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700
800
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700
800

Forward sampling

“Where should this pixel go?”

Reverse sampling

“Where does this pixel come from?”

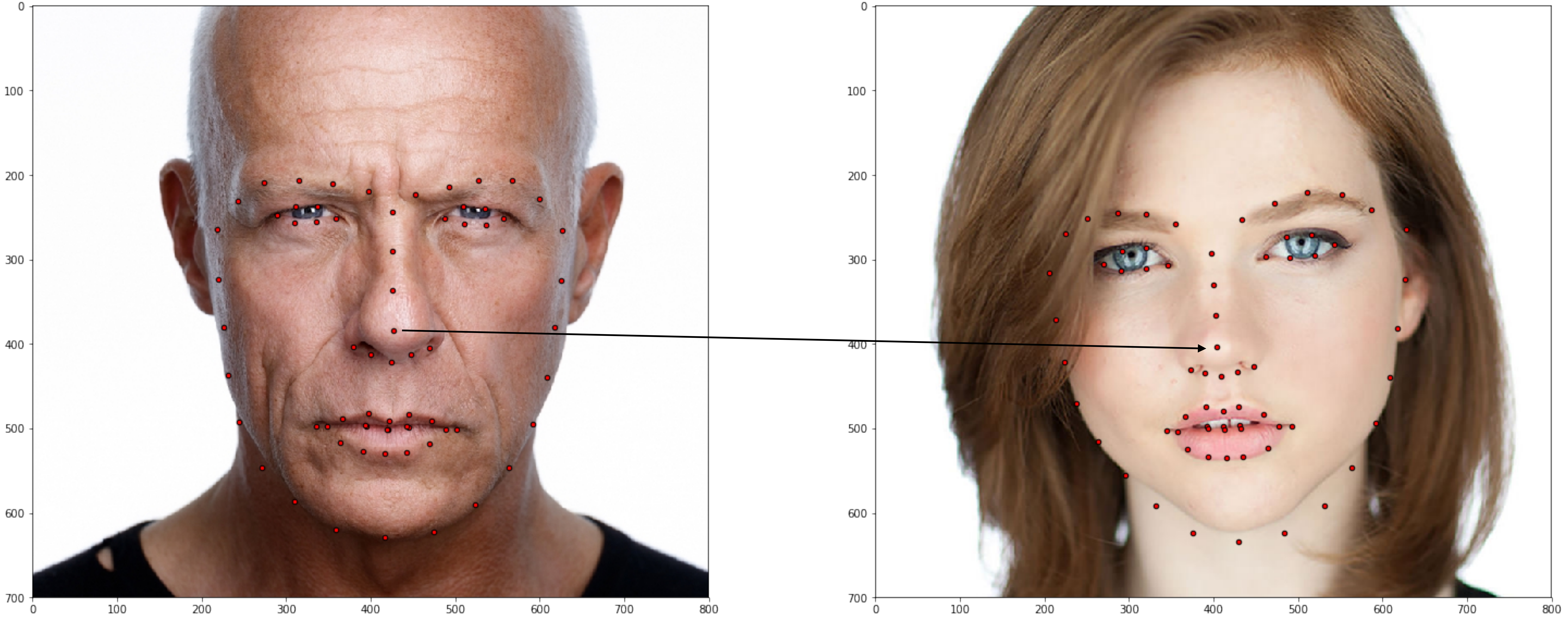
Forward sampling

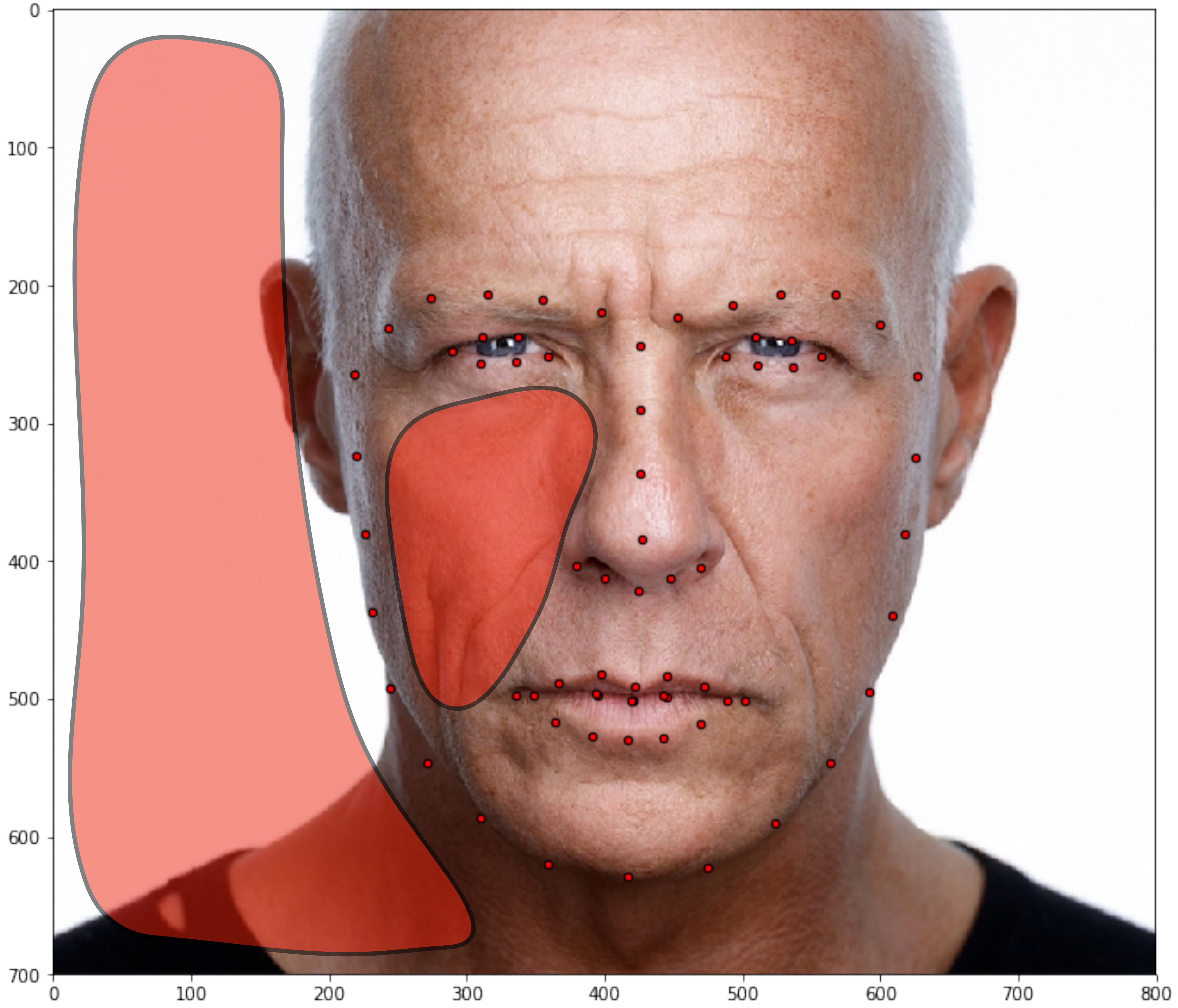
“Where should this pixel go?”

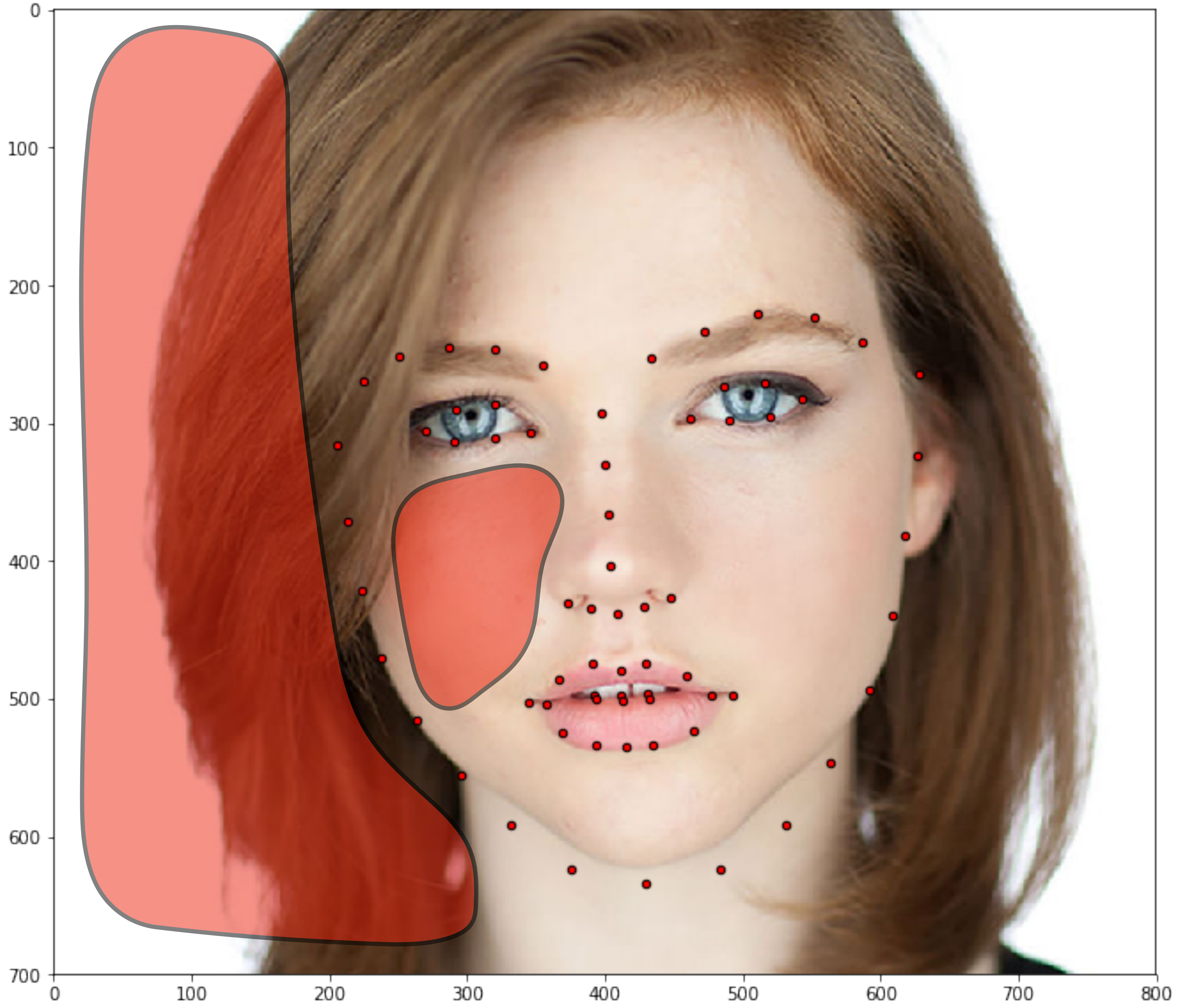
Reverse sampling

“Where does this pixel come from?”

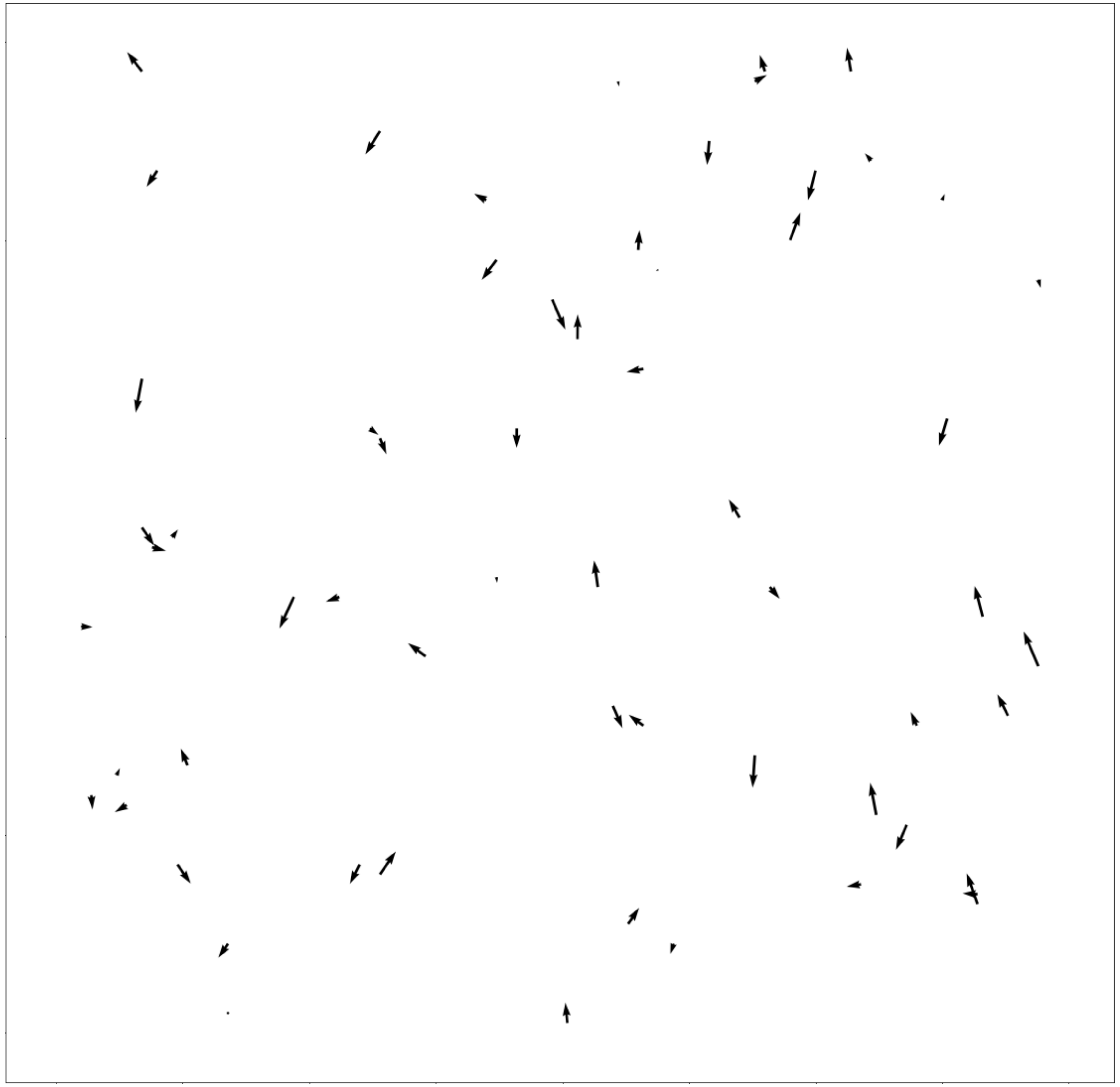
Advantage of reverse sampling?


0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700
800
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700
800

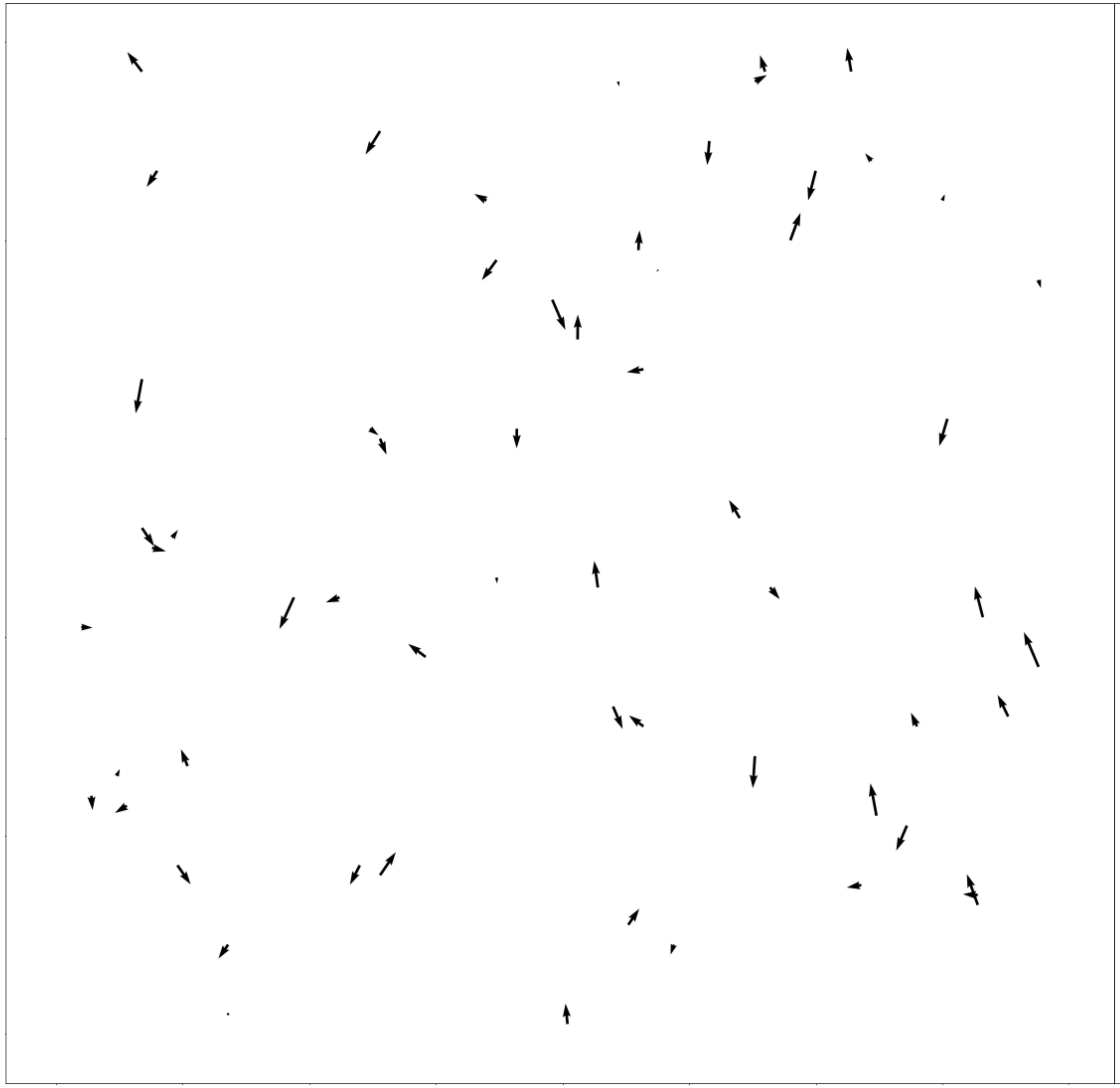




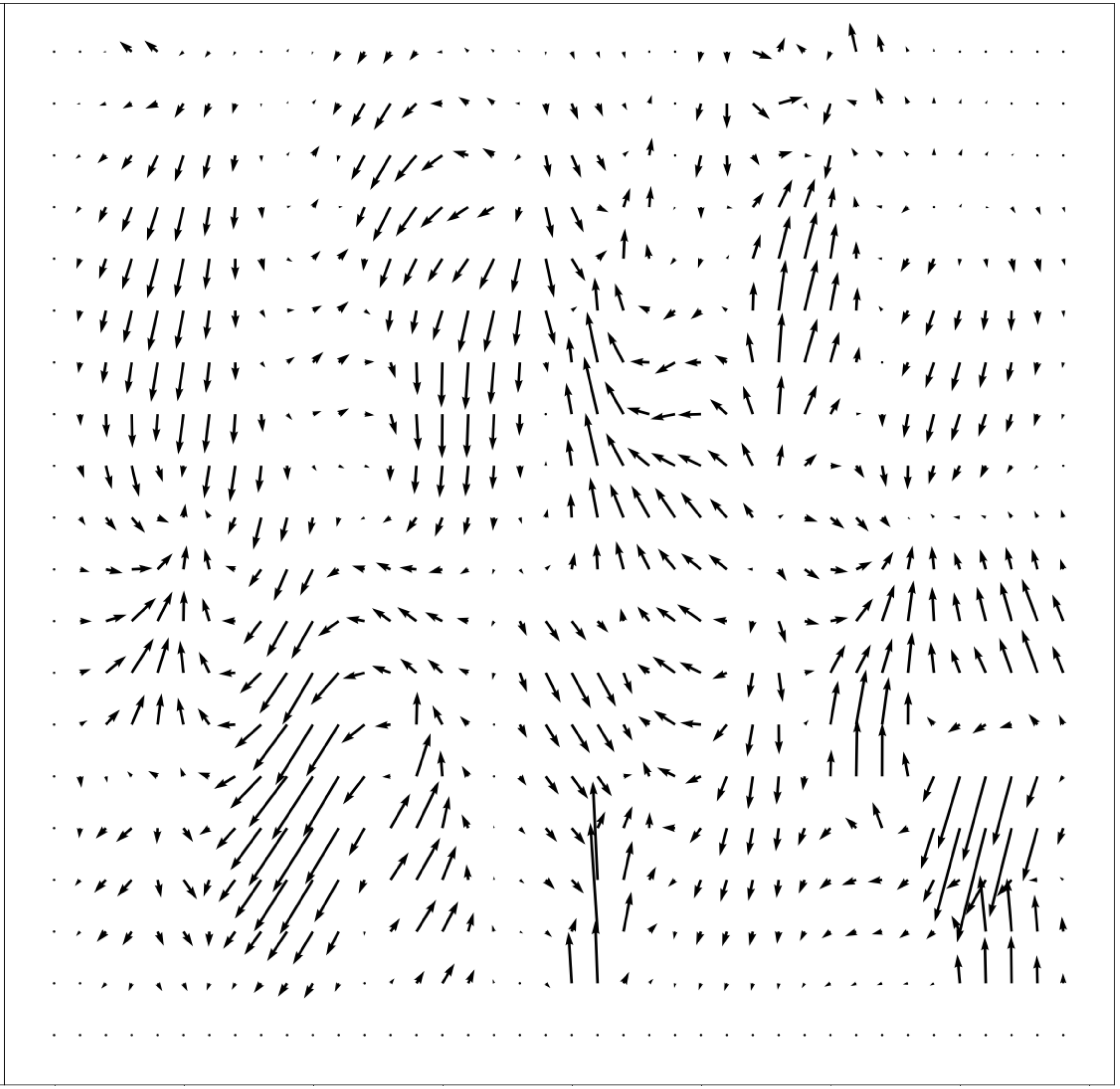
What about in-between landmarks?



Sparse vector field



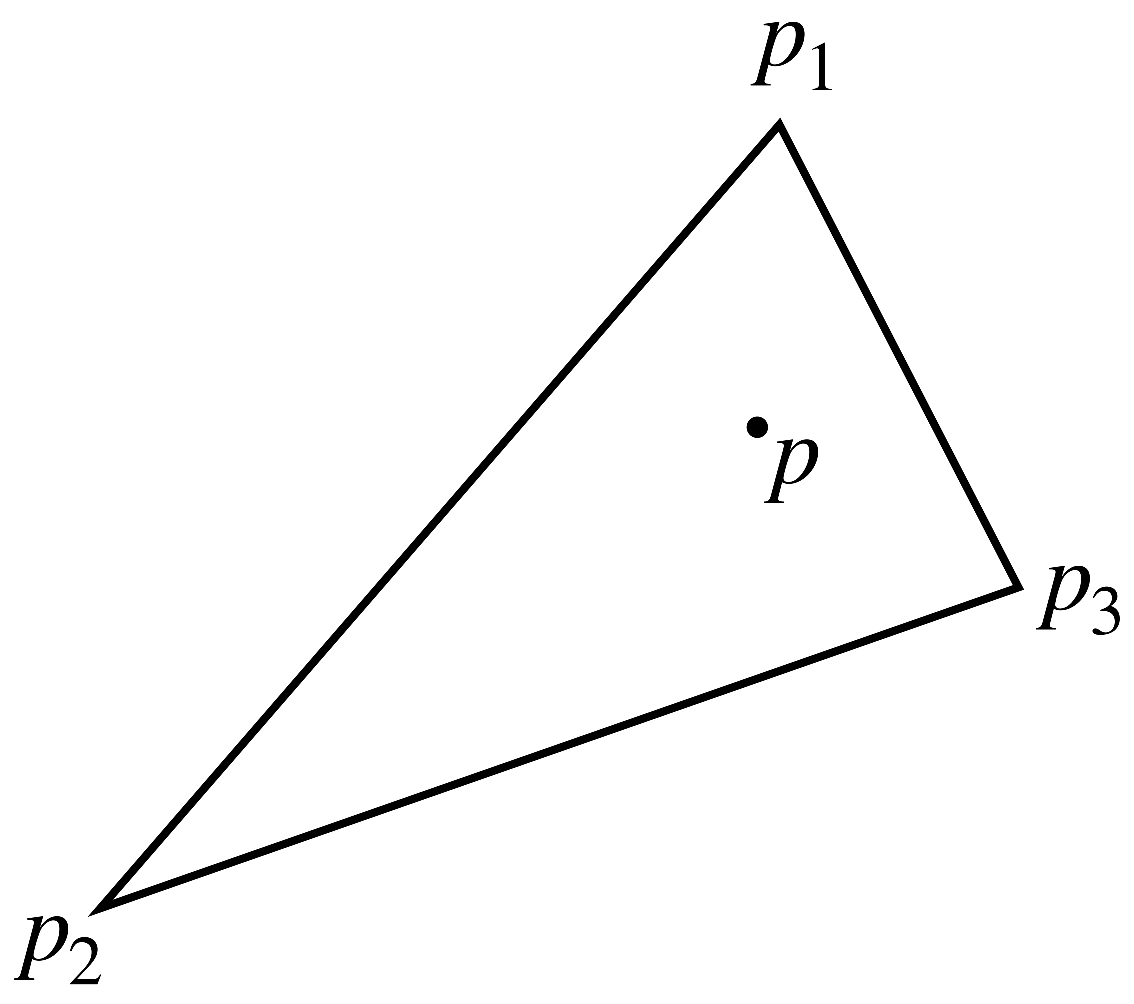
Sparse vector field



Dense vector field

scipy.interpolate.griddata

`scipy.interpolate.griddata`

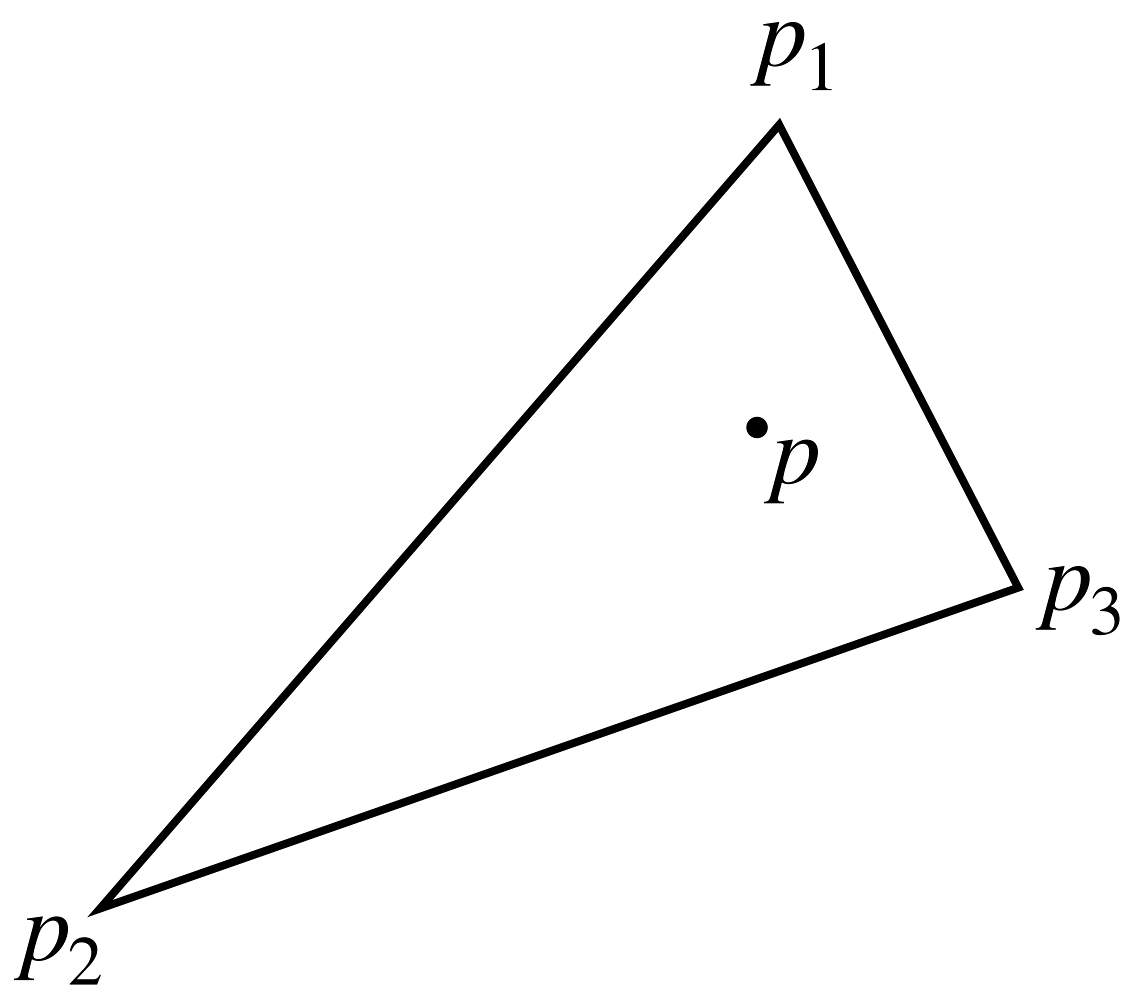


barycentric coordinates

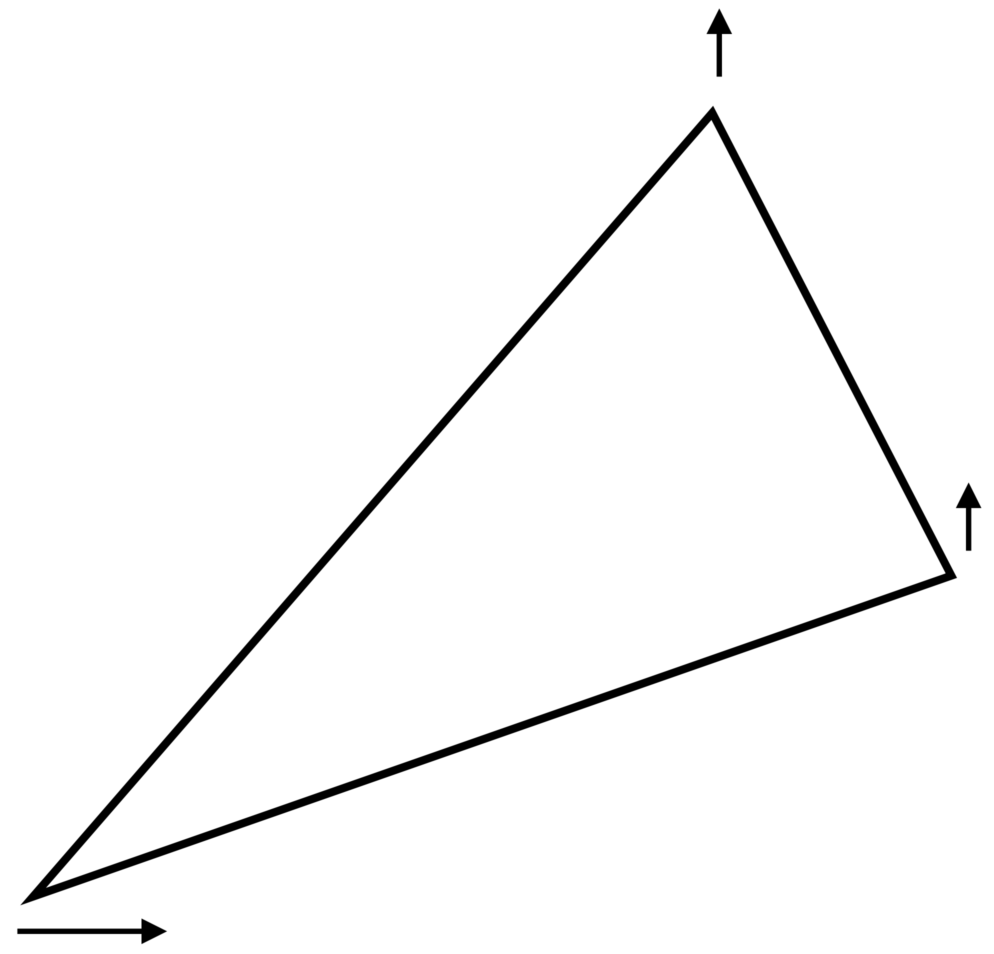
$$p = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$
$$\lambda_i \geq 0$$

`scipy.interpolate.griddata`



barycentric coordinates

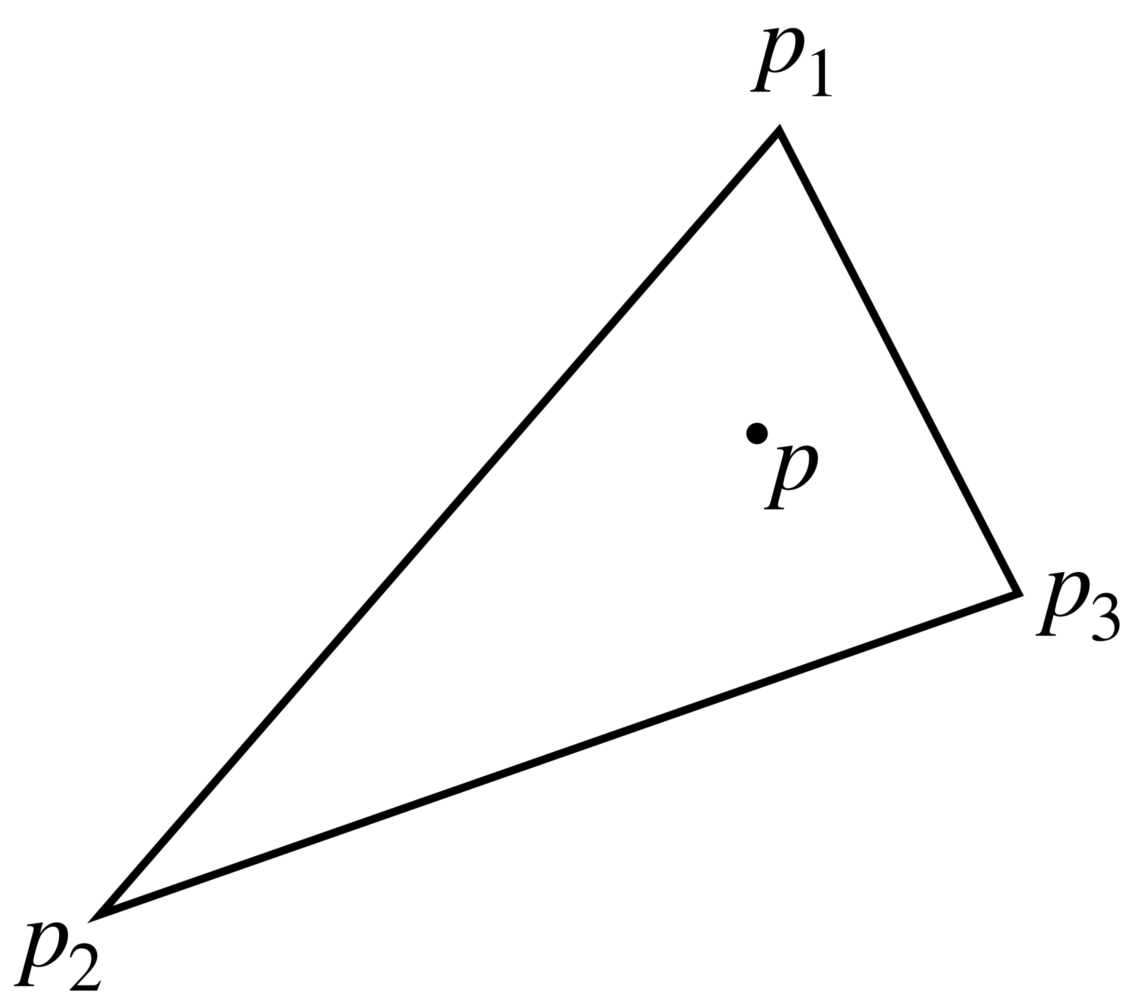


interpolate vectors

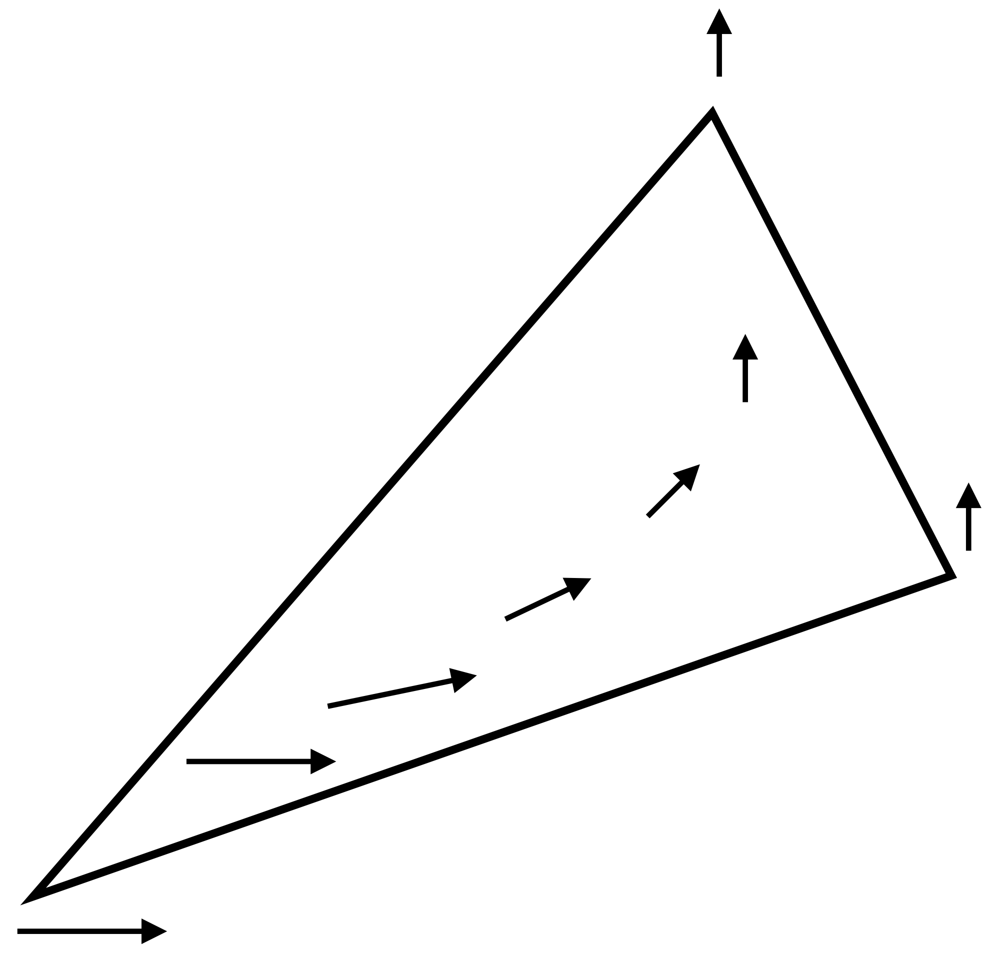
$$p = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$
$$\lambda_i \geq 0$$

`scipy.interpolate.griddata`



barycentric coordinates

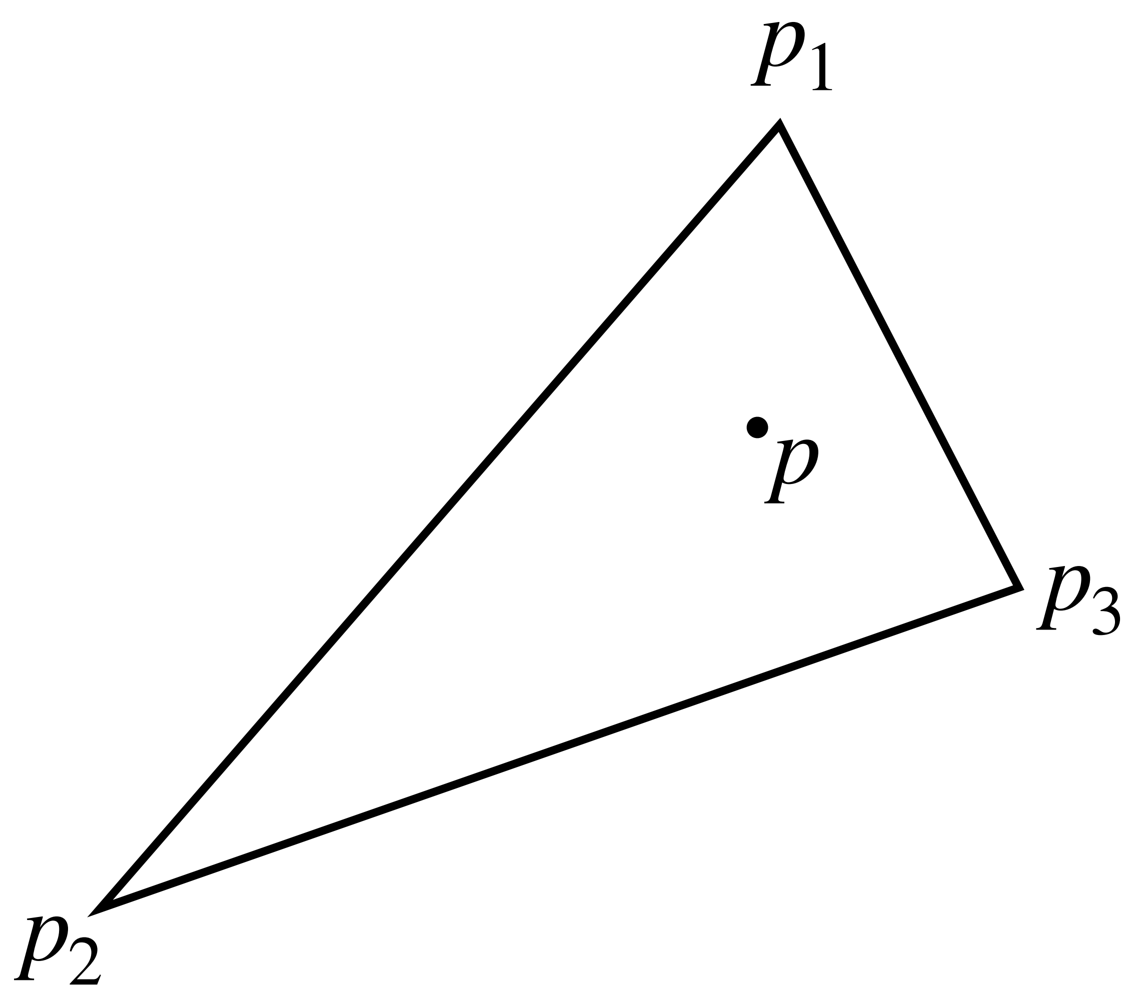


interpolate vectors

$$p = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$
$$\lambda_i \geq 0$$

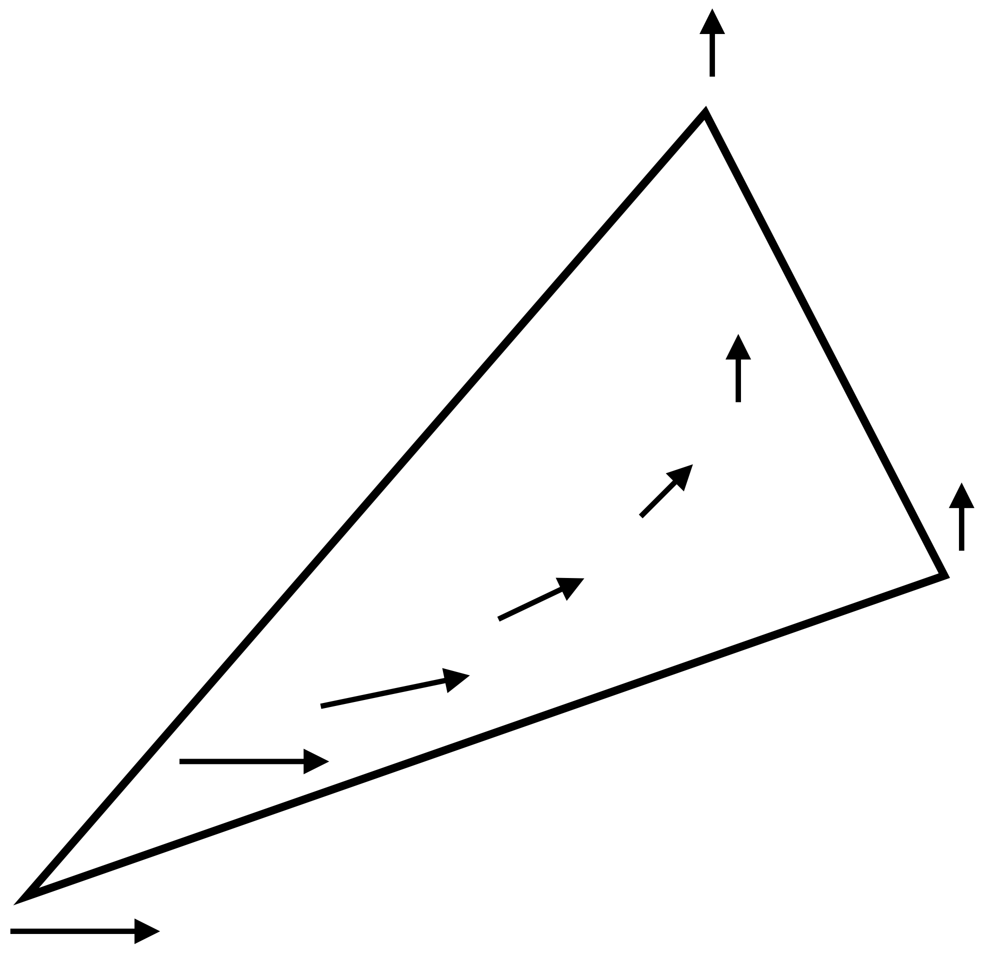
`scipy.interpolate.griddata`



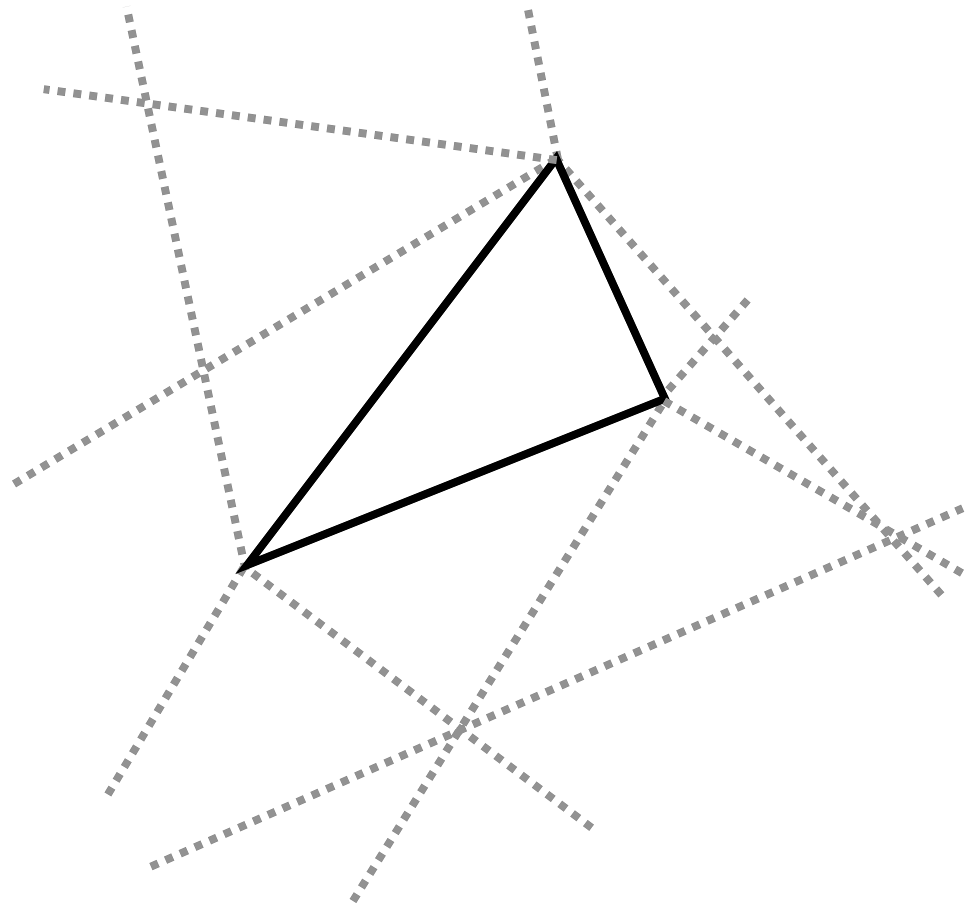
barycentric coordinates

$$p = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$$

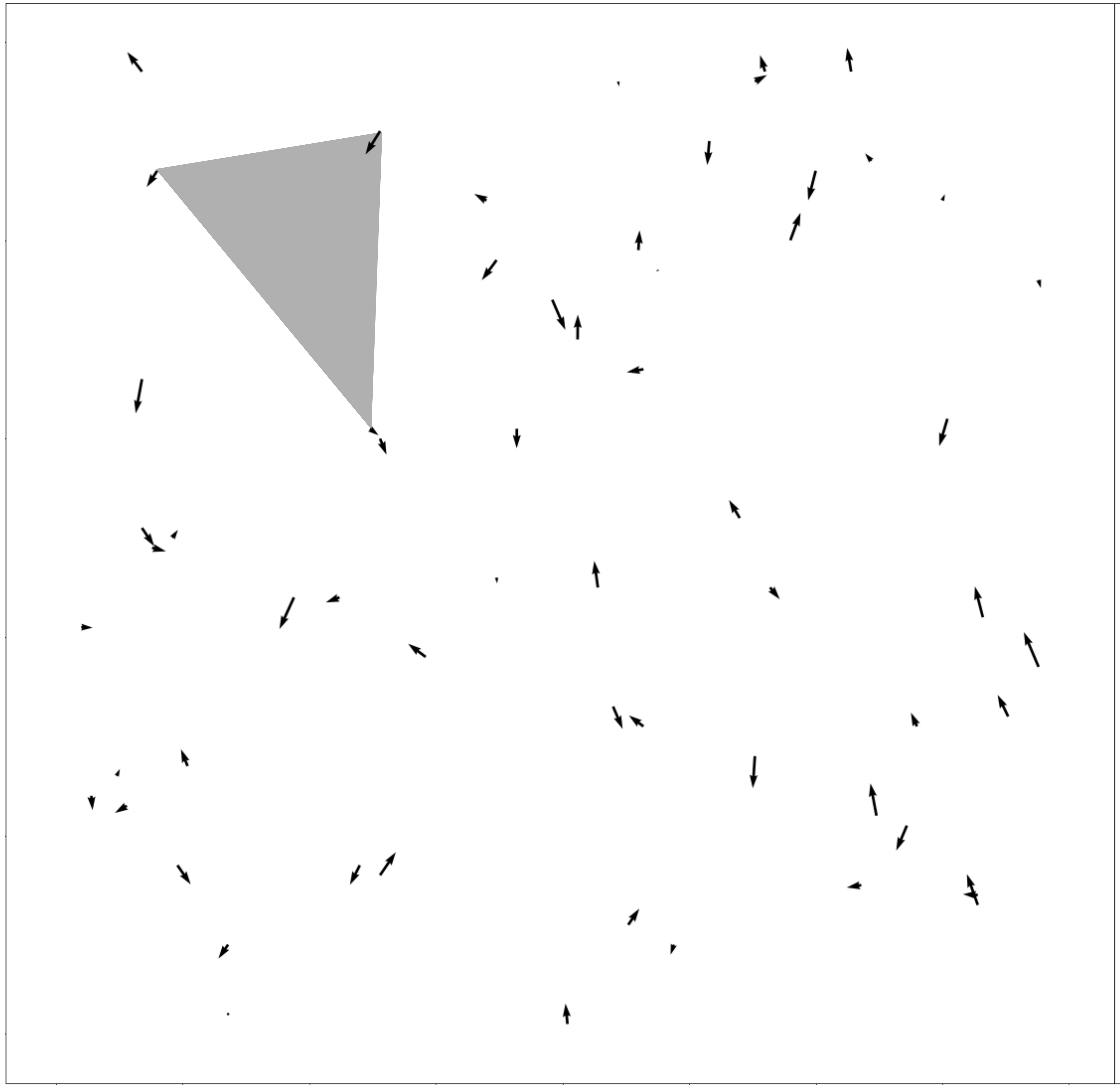
$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$
$$\lambda_i \geq 0$$



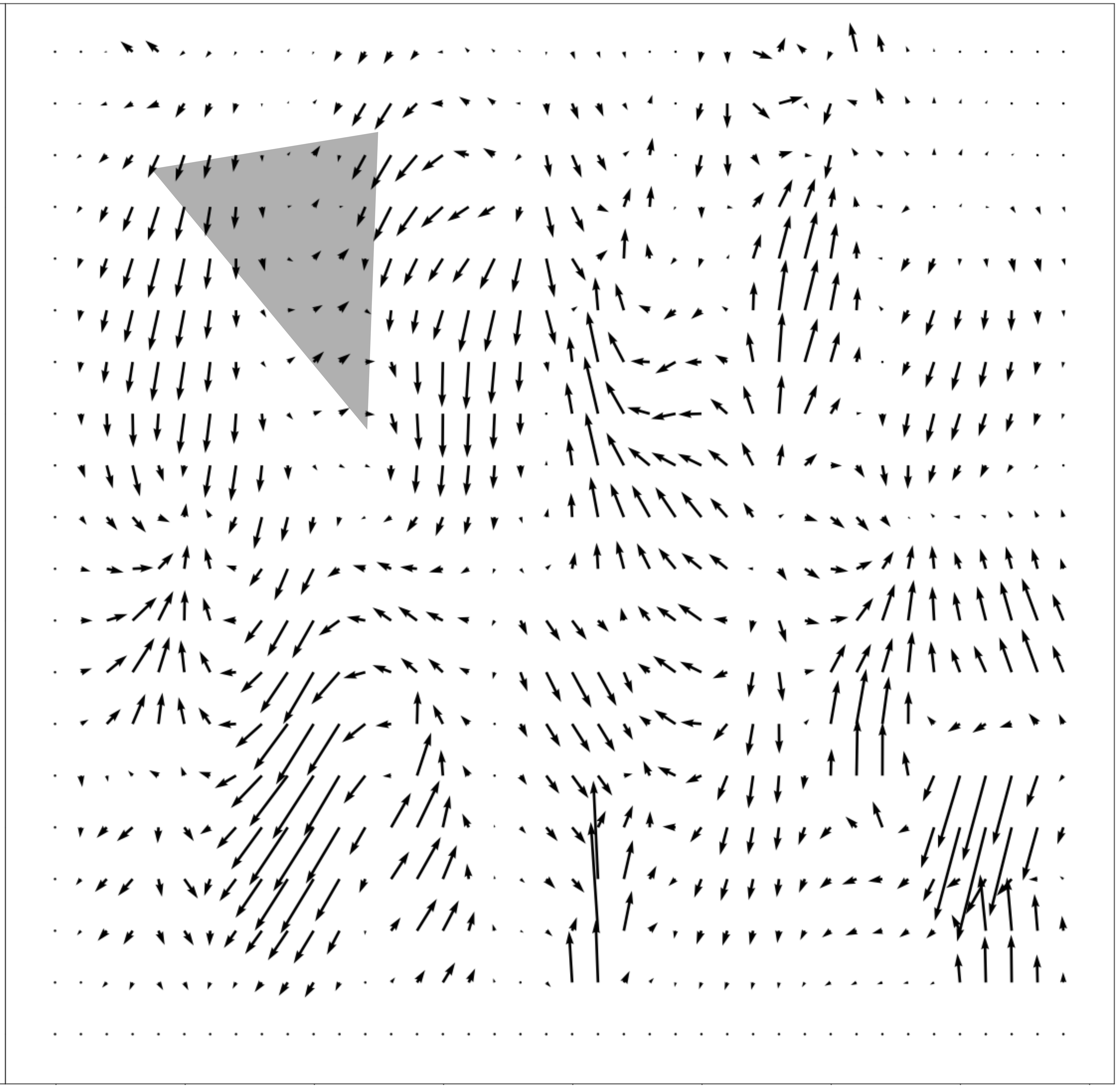
interpolate vectors



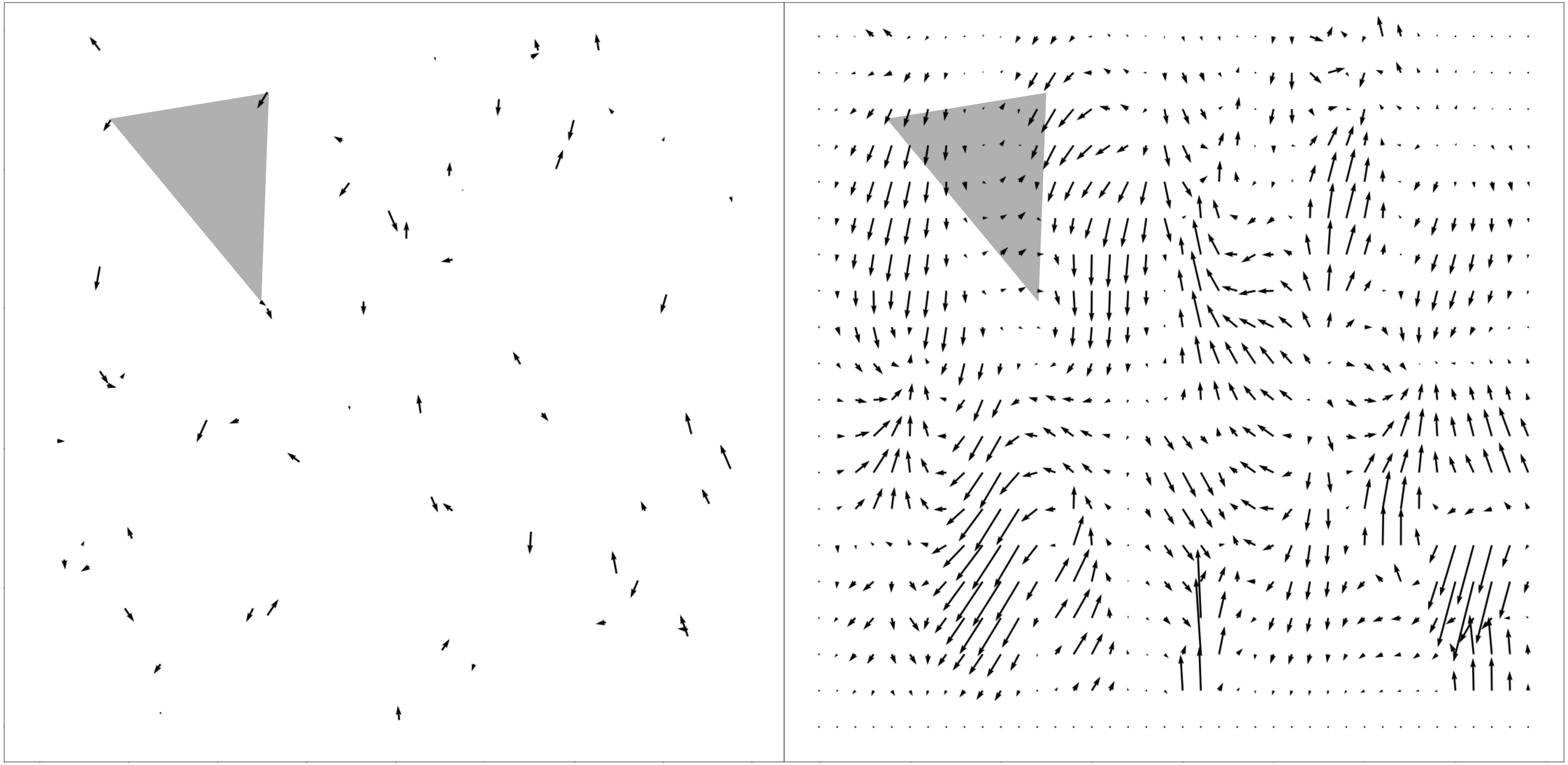
triangulate & interpolate



Sparse vector field



Dense vector field



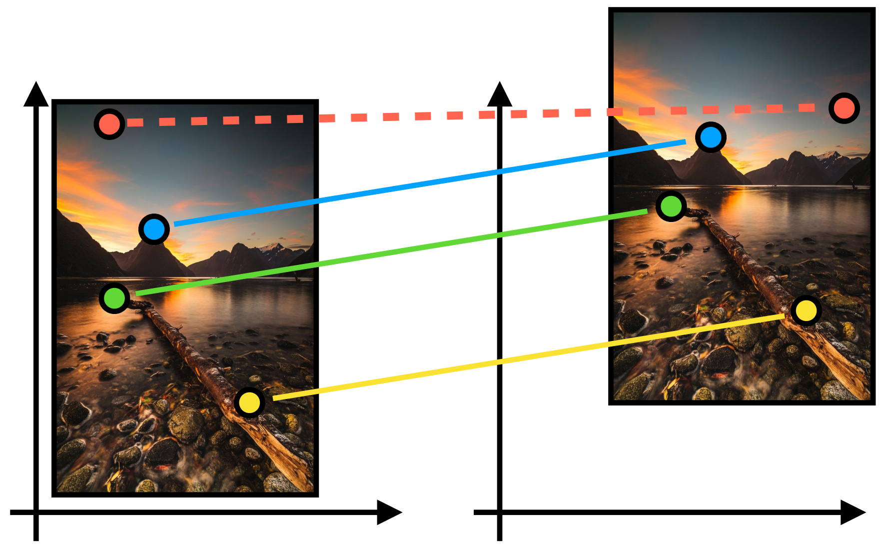
Sparse vector field

Dense vector field

Can be linear, cubic, …

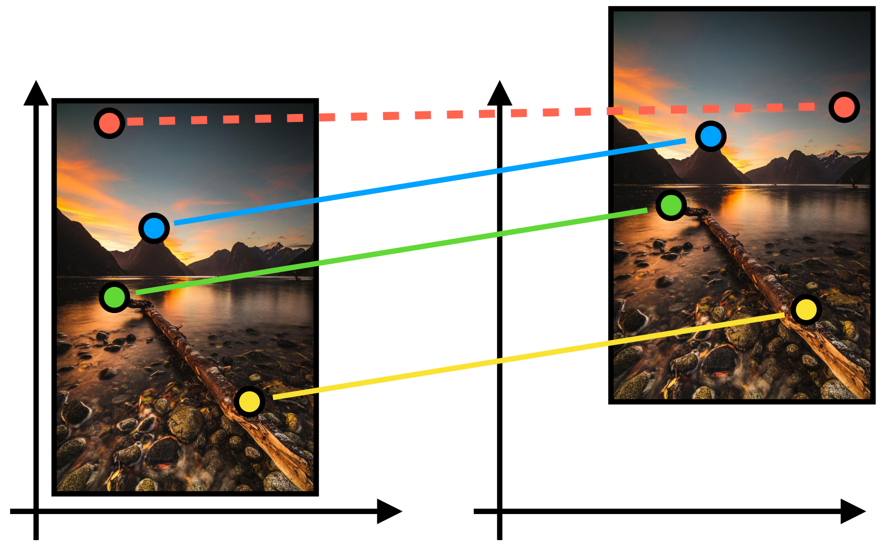
Recap

Recap

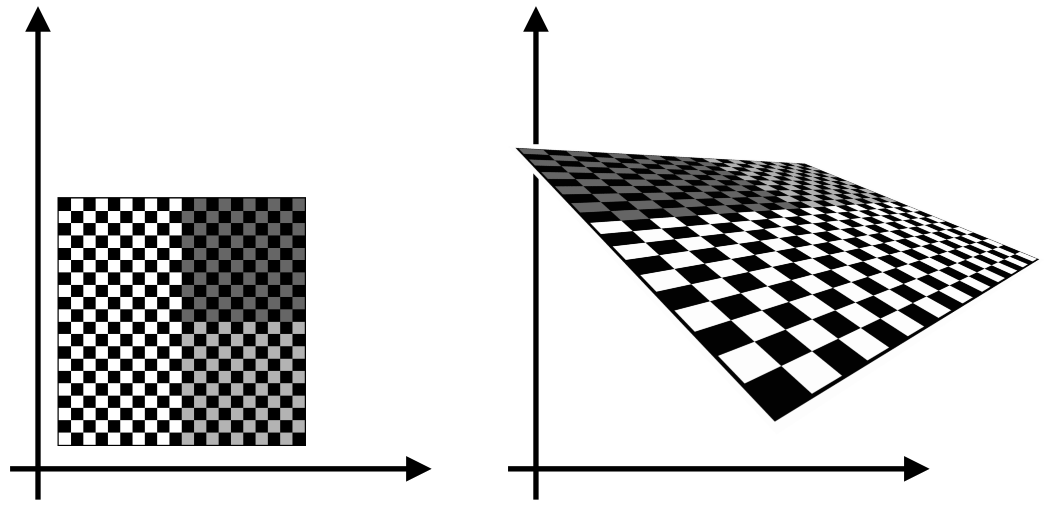


- RANSAC

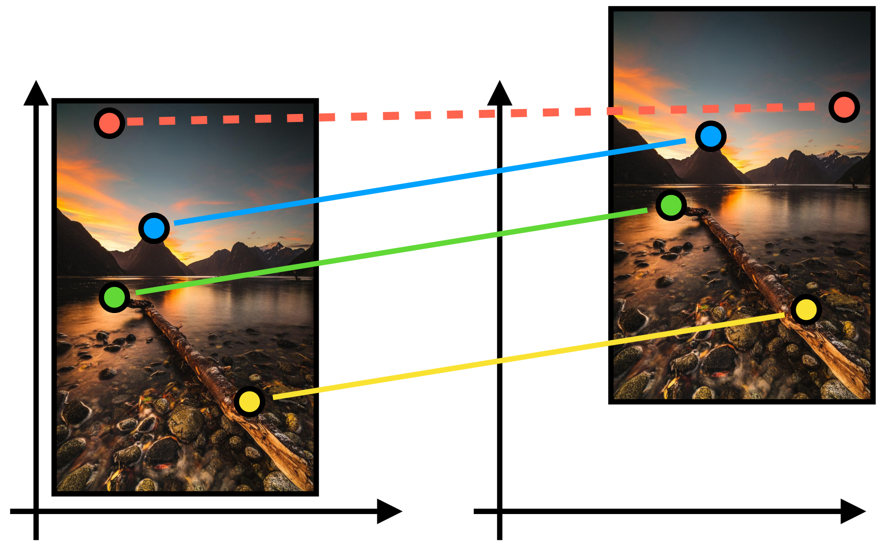
Recap



- RANSAC
- Geometric transformations

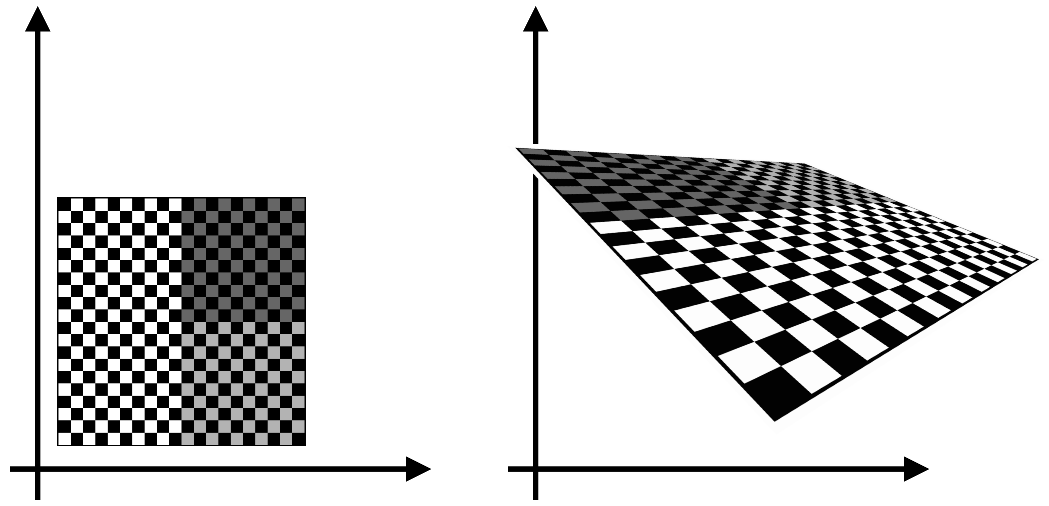


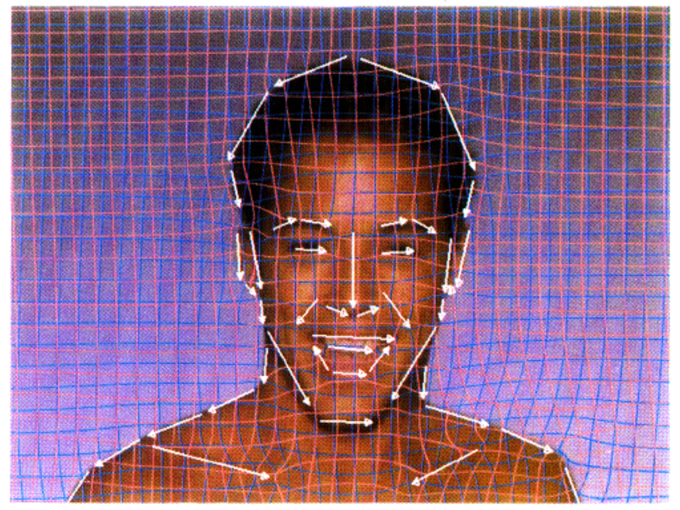
Recap



- RANSAC

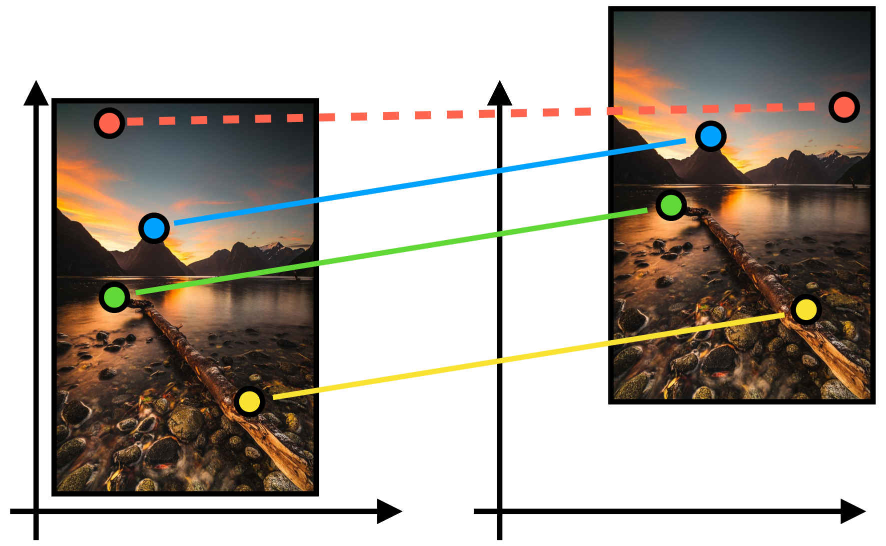
- Geometric transformations





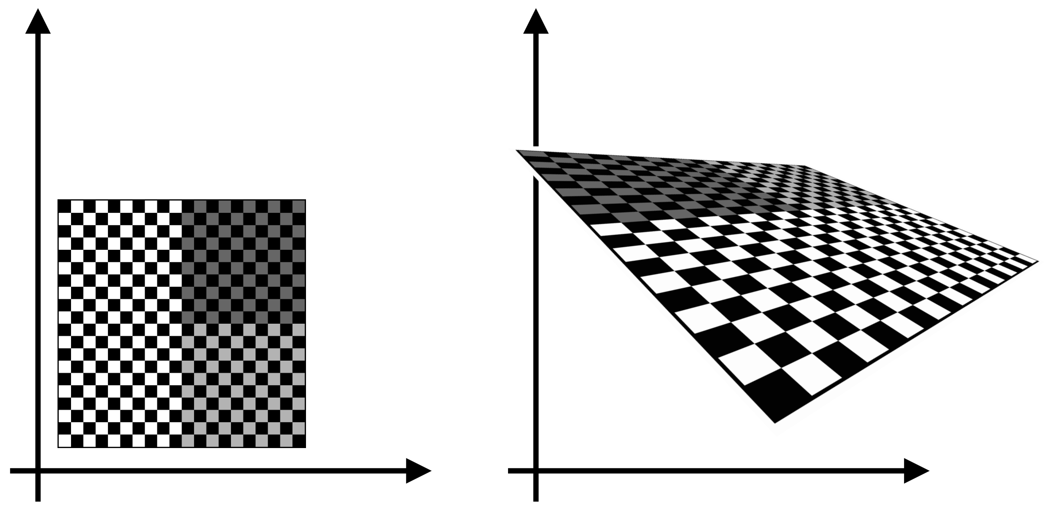
- Beier & Neely ’92

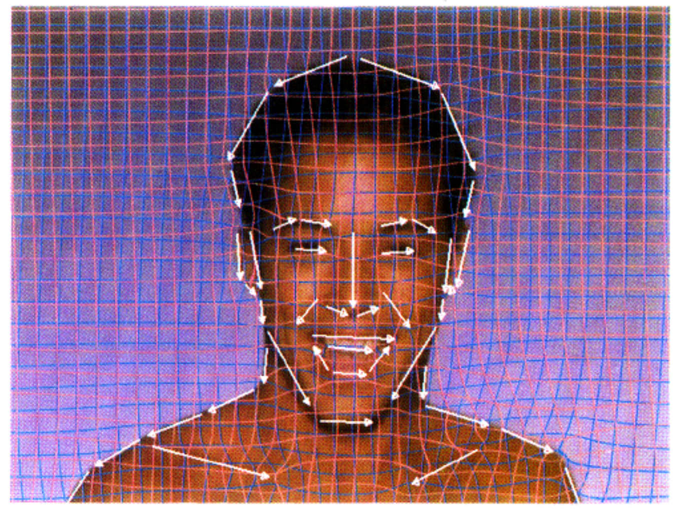
Recap



- RANSAC

- Geometric transformations





- Beier & Neely ’92

- Assignment 2