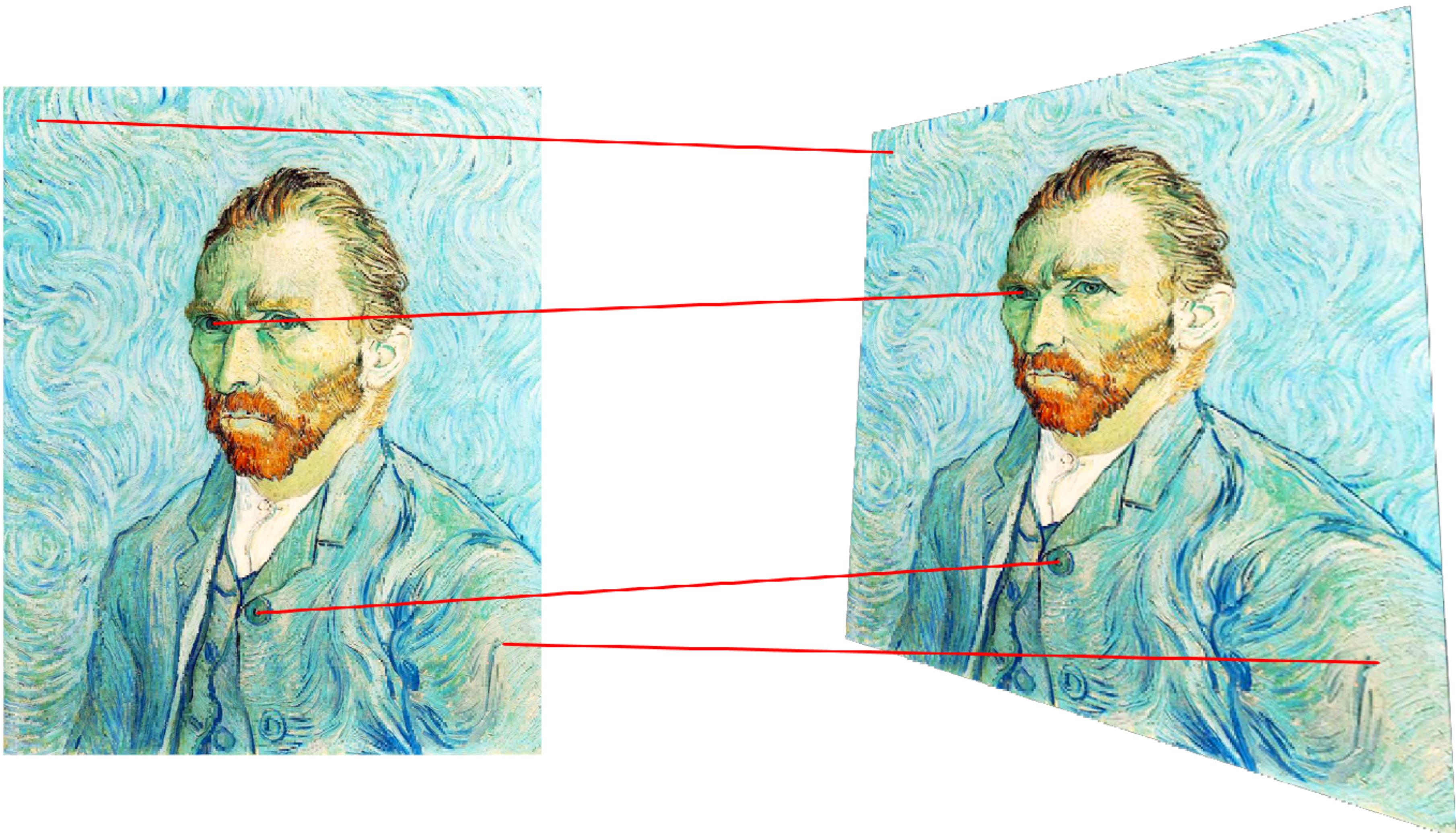




# Image Warping (cont'd)

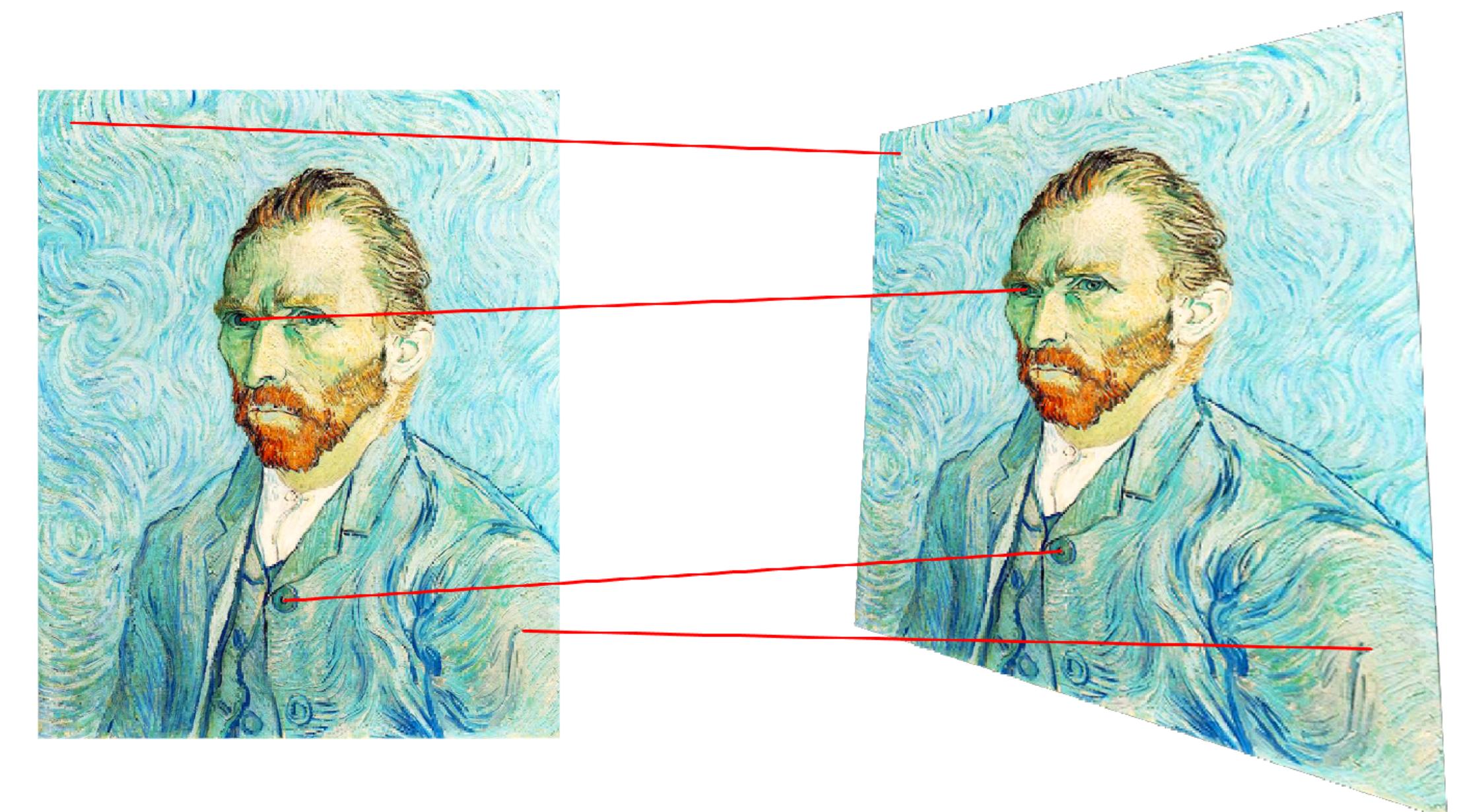
CS 355: Introduction to Graphics and Image Processing

# Perspective Warping



# Perspective Warping

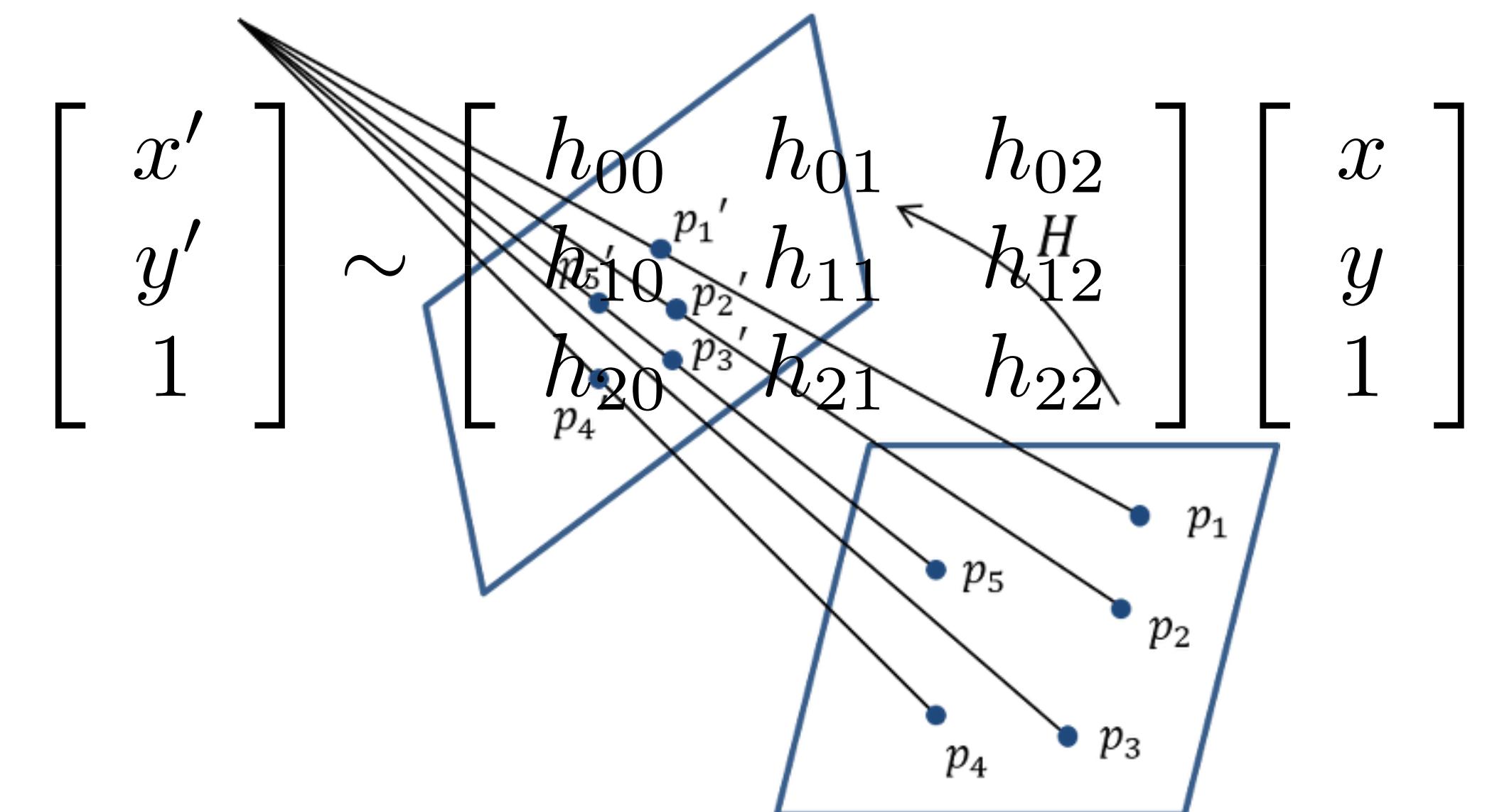
- You've already seen how to project 3D geometry to create an image
- Special case: if the points all lie on a plane in 3D, the transformation can be done with a single  $3 \times 3$  matrix
- Perspective projection of any 3D plane to any other 3D plane



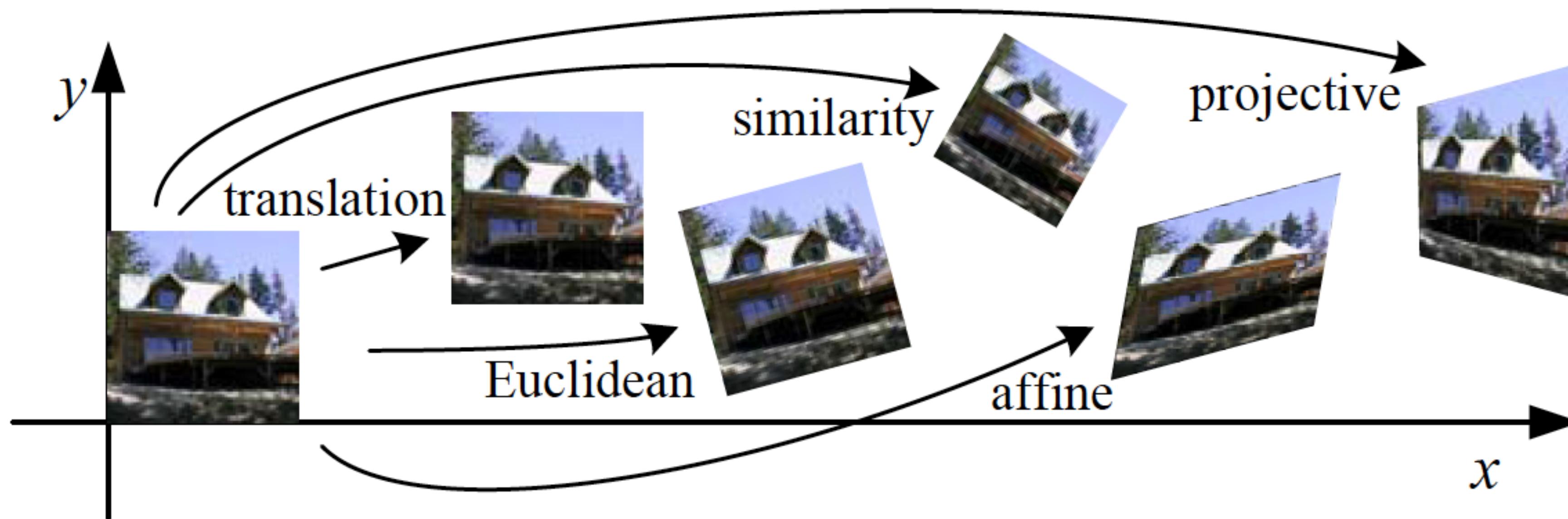
# Homographies

- Perspective projection from a plane to another plane is a homography
- Same  $3 \times 3$  matrix transformation you've seen before, but the bottom row doesn't have to be  $[0 \ 0 \ 1]$
- In practice, it's invertible!

$$\mathbf{p}' = \mathbf{H} \ \mathbf{p}$$



# Classes of Transformations



# Homographies

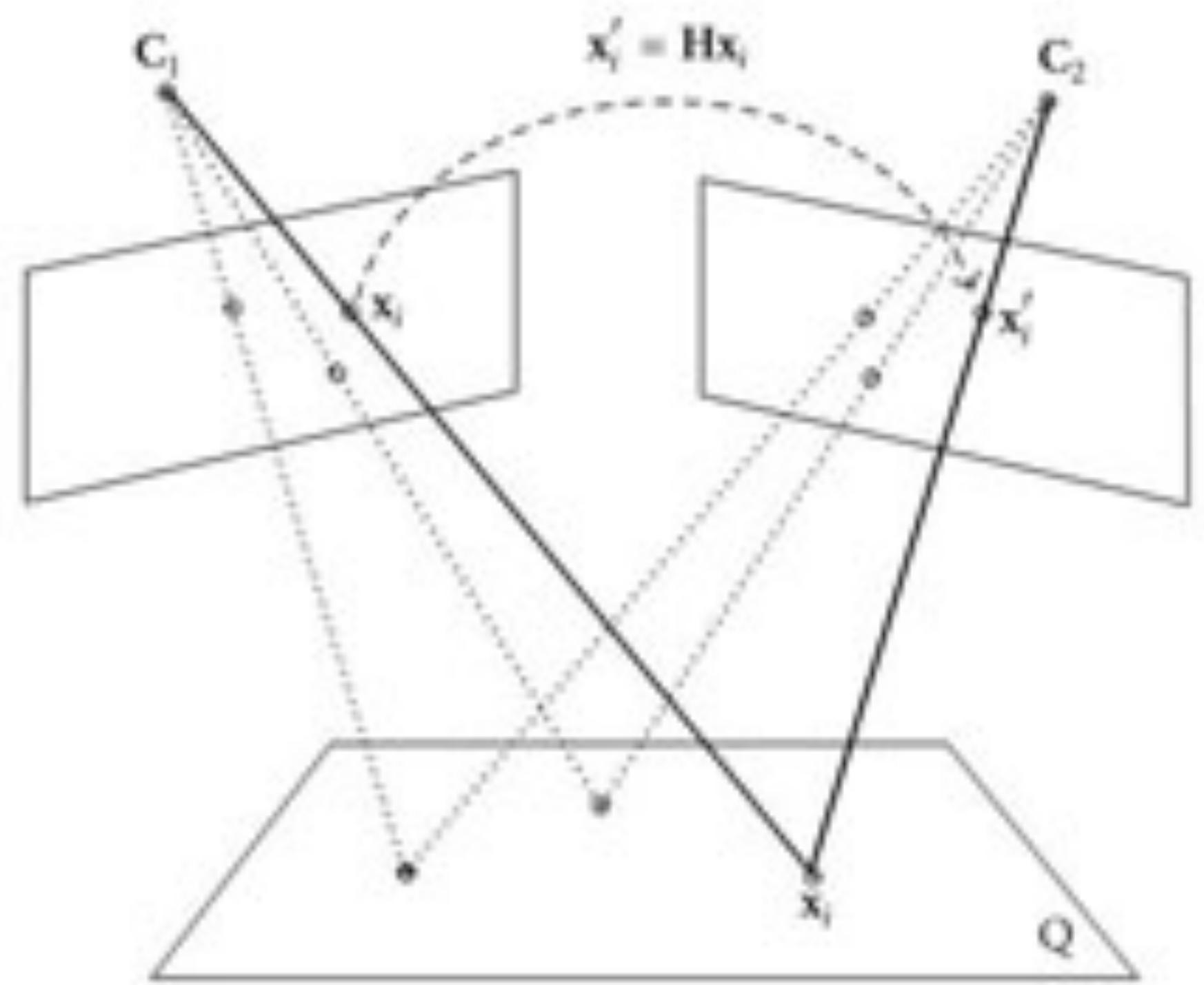
- Non-zero multiple of a homography does the same transformation
- That means there are really only 8 free parameters, not 9

$$\mathbf{p}' = \mathbf{H} \mathbf{p}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Multiple Pictures of the Same Plane

- If two cameras take pictures of a plane from different viewpoints,
  - The projection of the plane to each camera is a homography
  - The transformation of points from one picture to the other is also a homography!



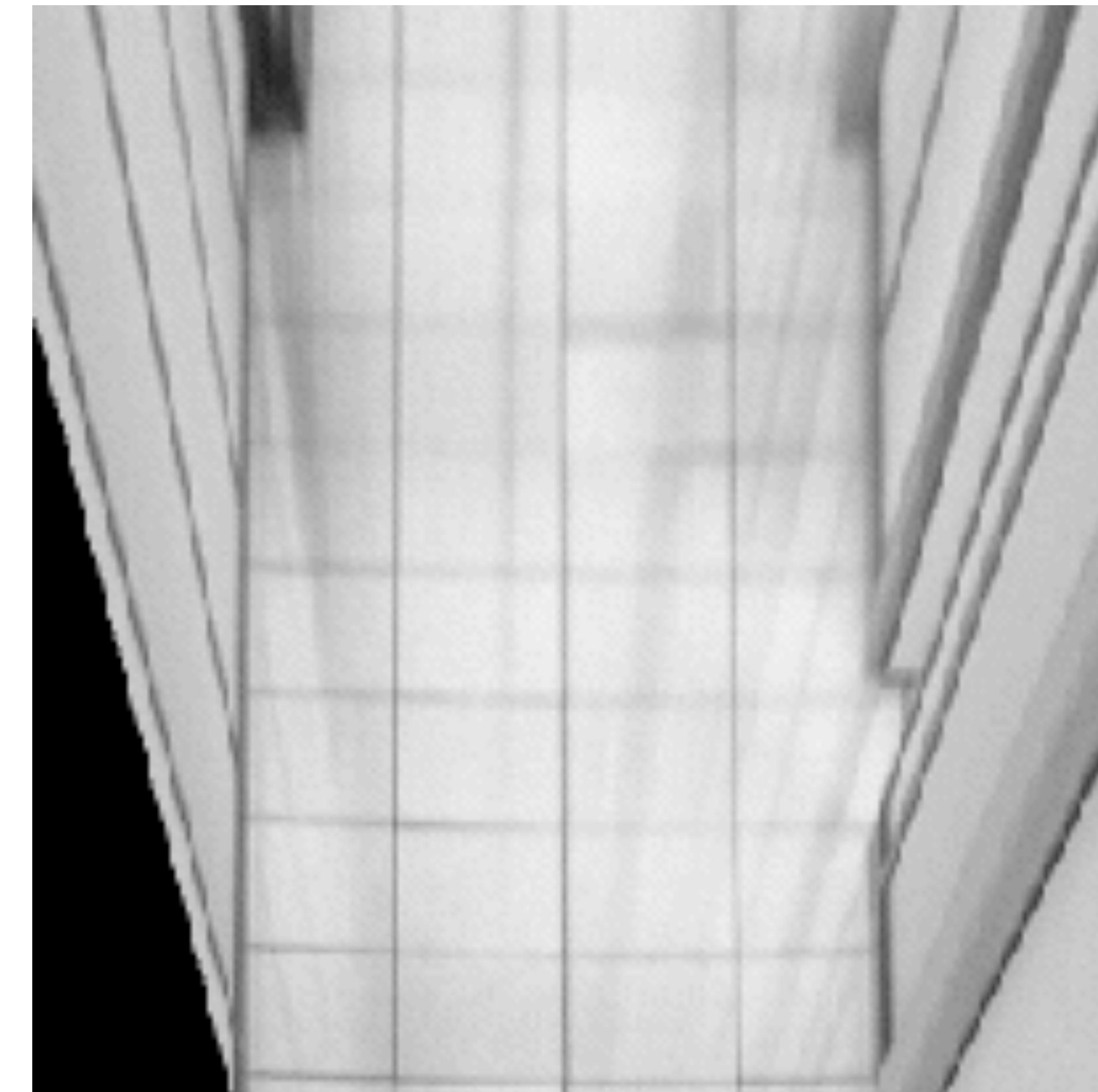
# Image Stitching



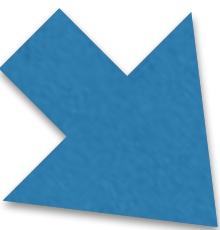
# “Straight On” View



# “Birds-Eye” View

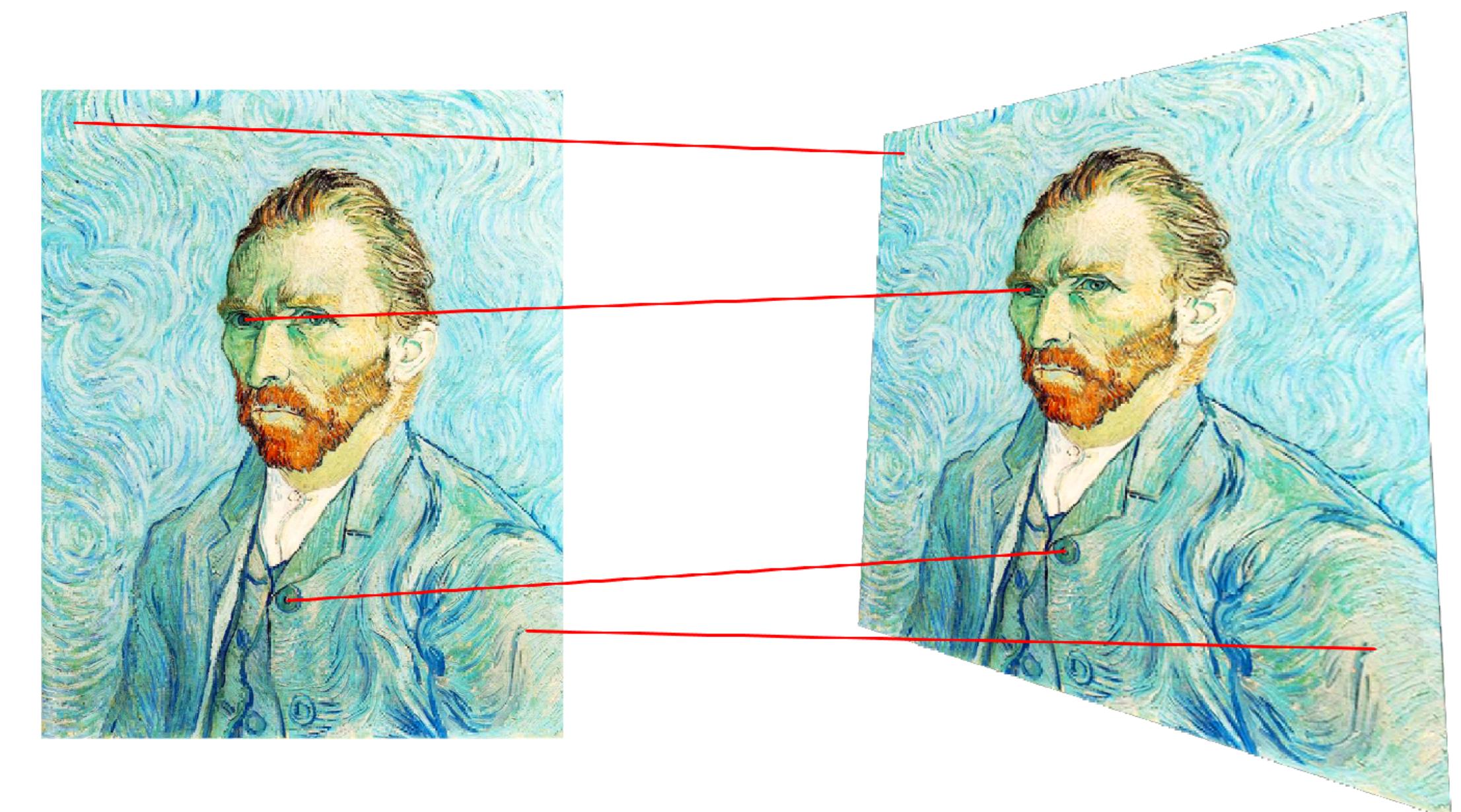


# Or Going The Other Way

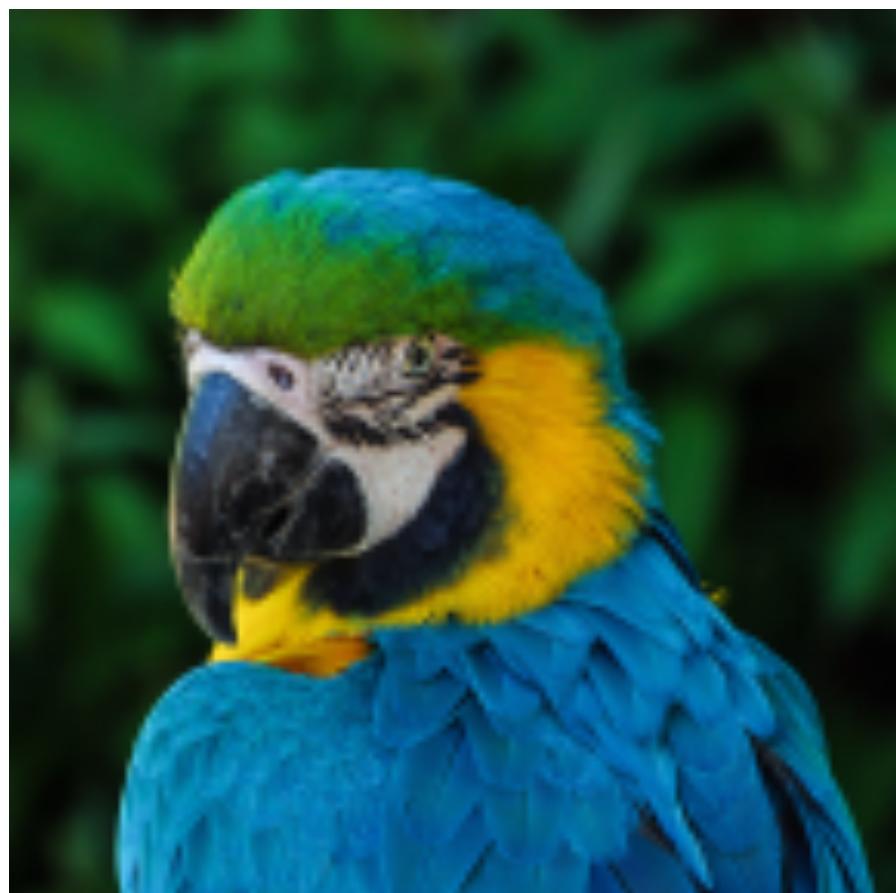


# Computing Homographies

- Eight free parameters
- A pair of matching points provides two constraints (one for x, one for y)
- Four points provides eight equations to solve for eight unknowns
- Call the *4-point algorithm*



# Lab 9



# Lab 9



# Lab 9

- We give you an implementation of the 4-point algorithm:
  - You give it matching points
  - It gives you back a homography
- You implement:
  - Backward warping
  - Bilinear interpolation
  - Point-in-frame geometric tests  
(more on this later)



# Coming up...

- Back to more geometry:
  - Representing lines, curves, surfaces
  - Geometric tests