中国科学技术大学
University of Science and Technology of China

Image Stitching

张举勇
中国科学技术大学

Overview

- Image stitching is to combine multiple photos to create a larger photo.
- This technology is now widely available. It's on pretty much all smart phones that are in market today.
- It's also used in other domains, such as medical imaging, and remote sensing.

Image Stitching

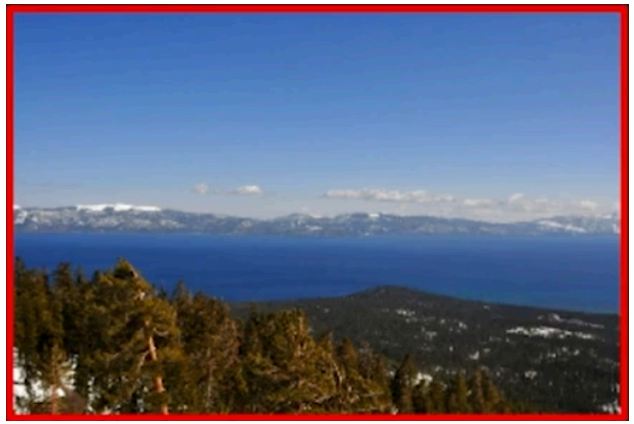


Image 1

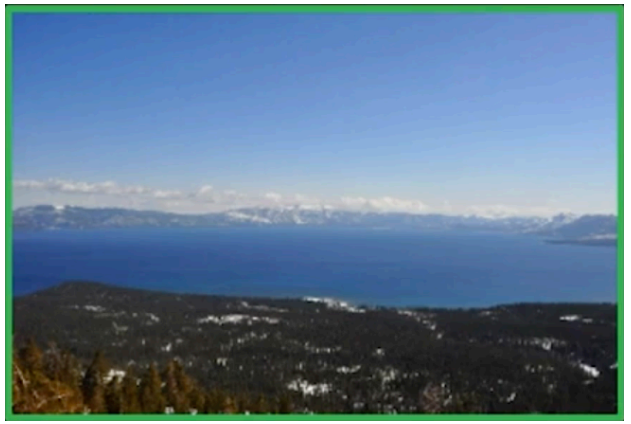


Image 2

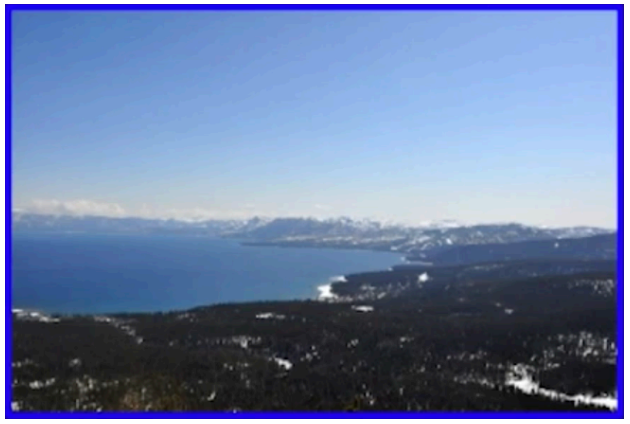


Image 3

How would you align these images?

Image Stitching

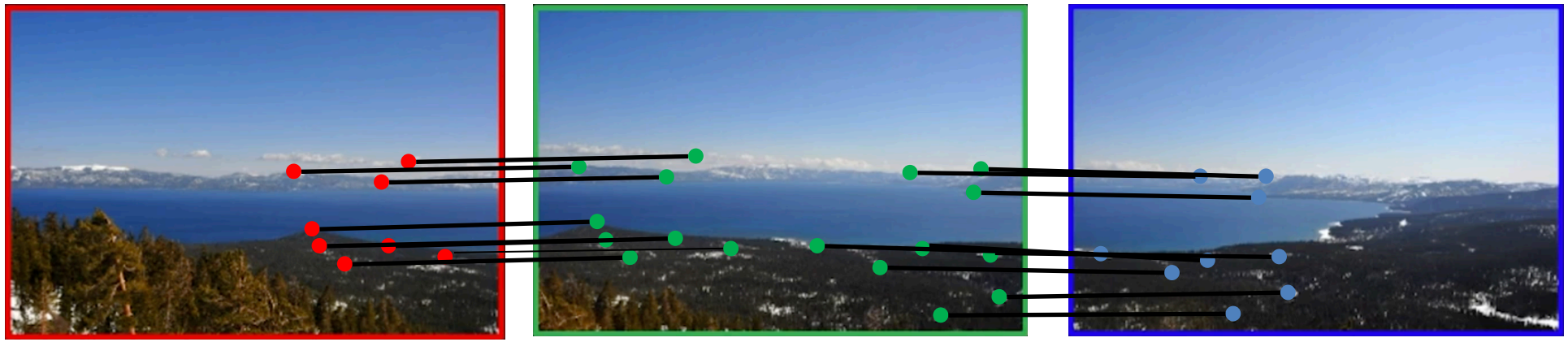
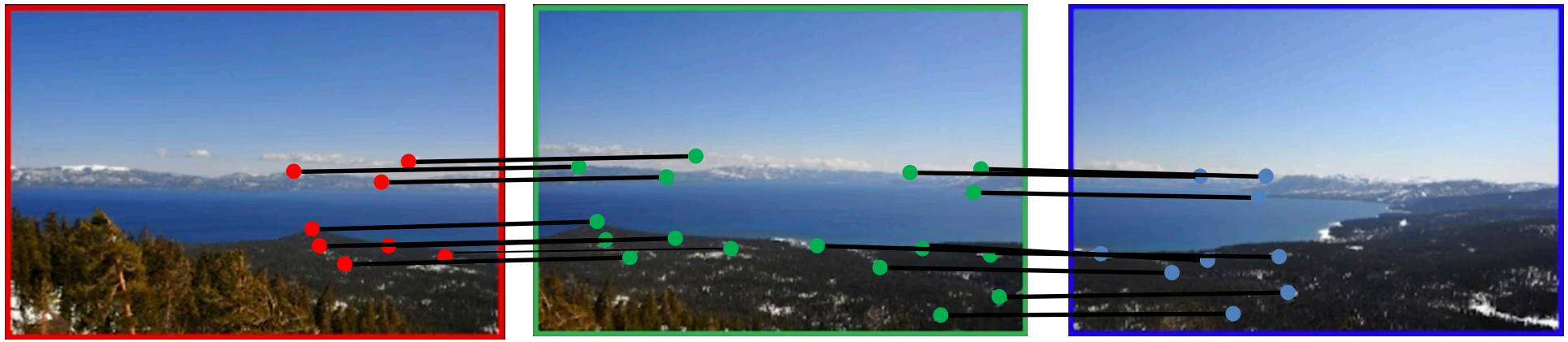


Image 1 Image 2 Image 3

Find corresponding points
(using feature detectors like SIFT)

Image Stitching



Find geometric relationship between the images

Image Stitching

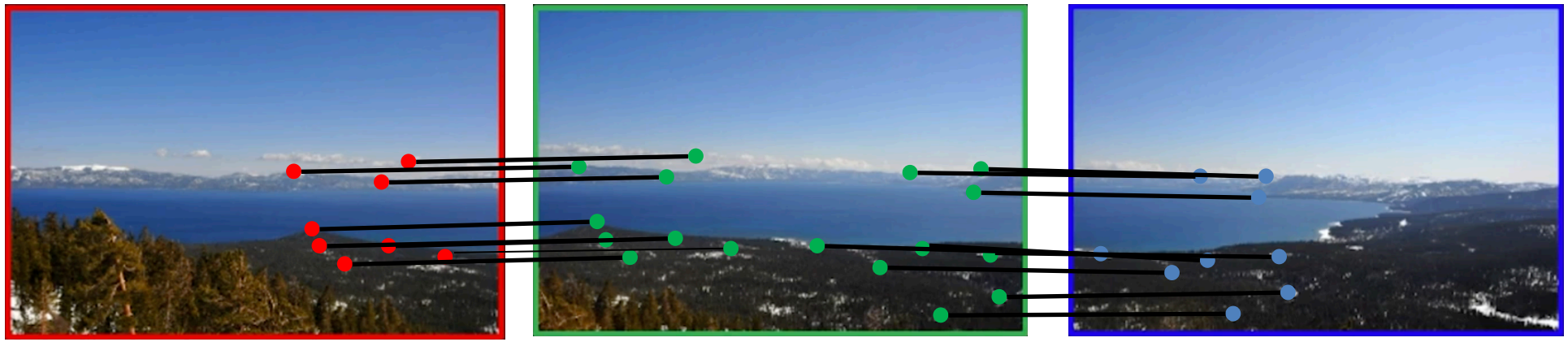
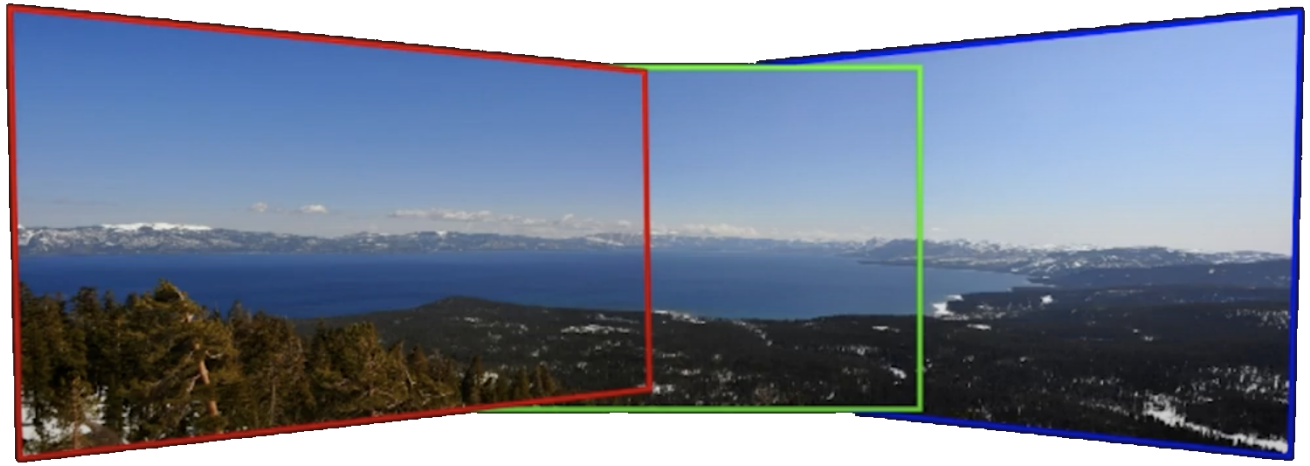
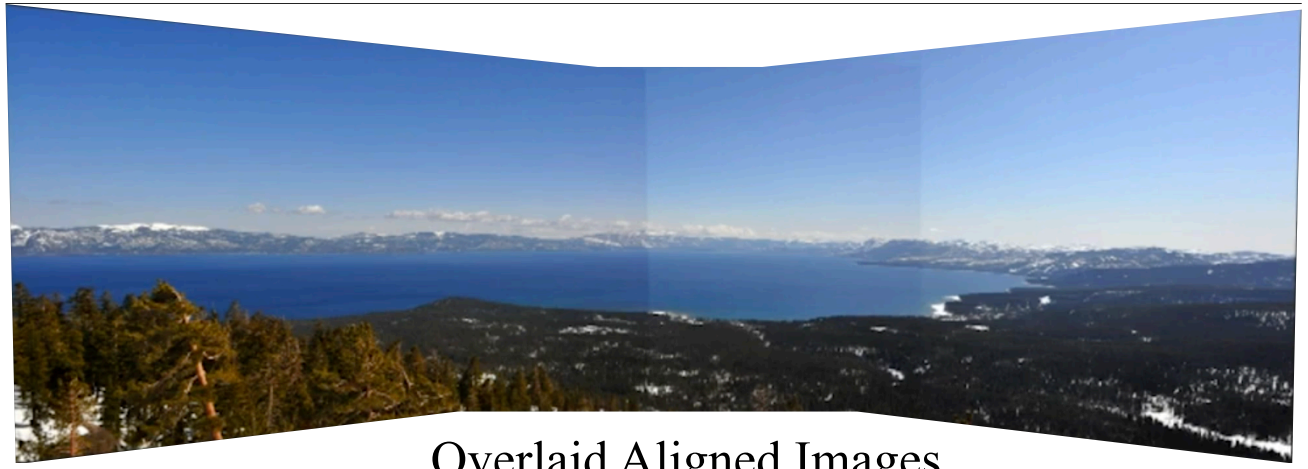


Image 1 Image 2 Image 3

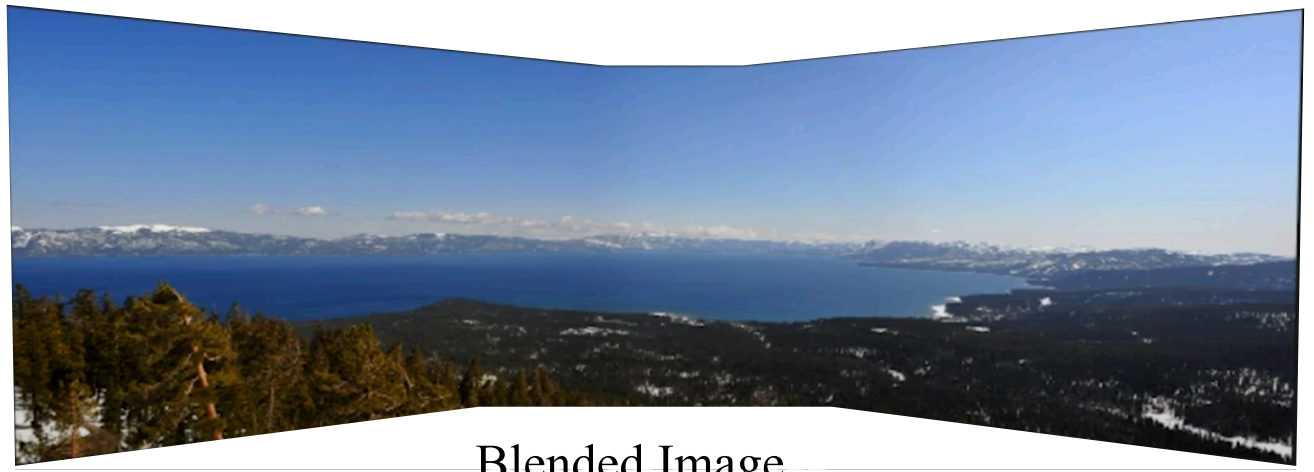


Warp images so that corresponding points align

Image Stitching



Overlaid Aligned Images



Blended Image

Blend images to remove hard seams

Image Stitching

Topics:

- 2x2 Image Transformations
- 3x3 Image Transformations
- Computing Homography
- Dealing with Outliers:RANSAC
- Warping and Blending Images

2x2 Image Transformations

Image Filtering: Change range (brightness)

$$g(x, y) = T_r(f(x, y))$$

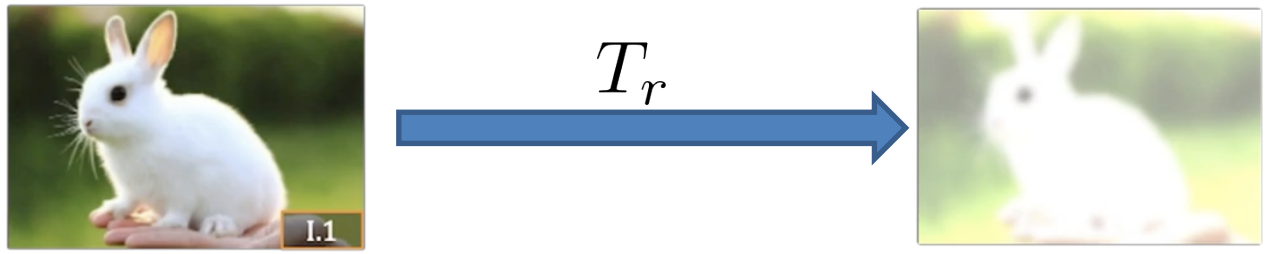
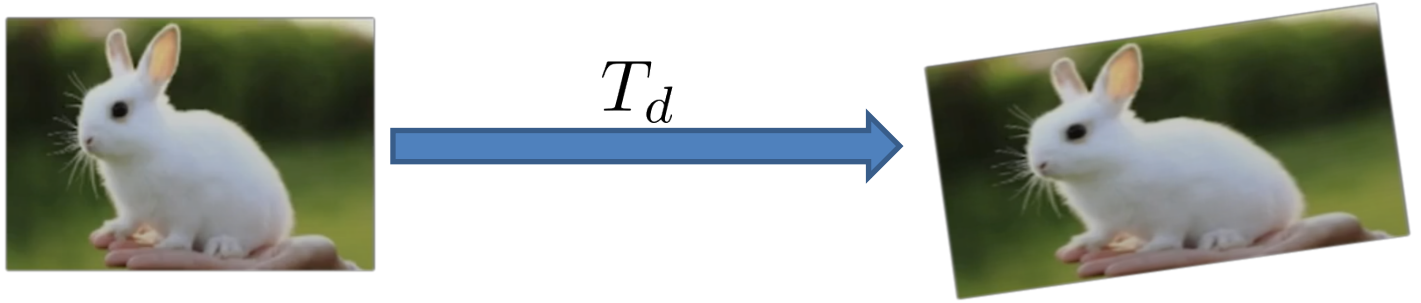


Image Warping: Change domain (location)

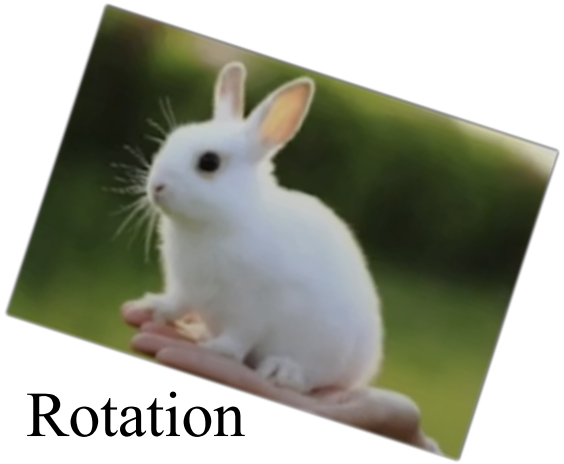
$$g(x, y) = f(T_d(x, y))$$

Transformation T_d is a coordinate changing operator



Global Warping/Transformation

Translation



Rotation

Scaling and Aspect

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

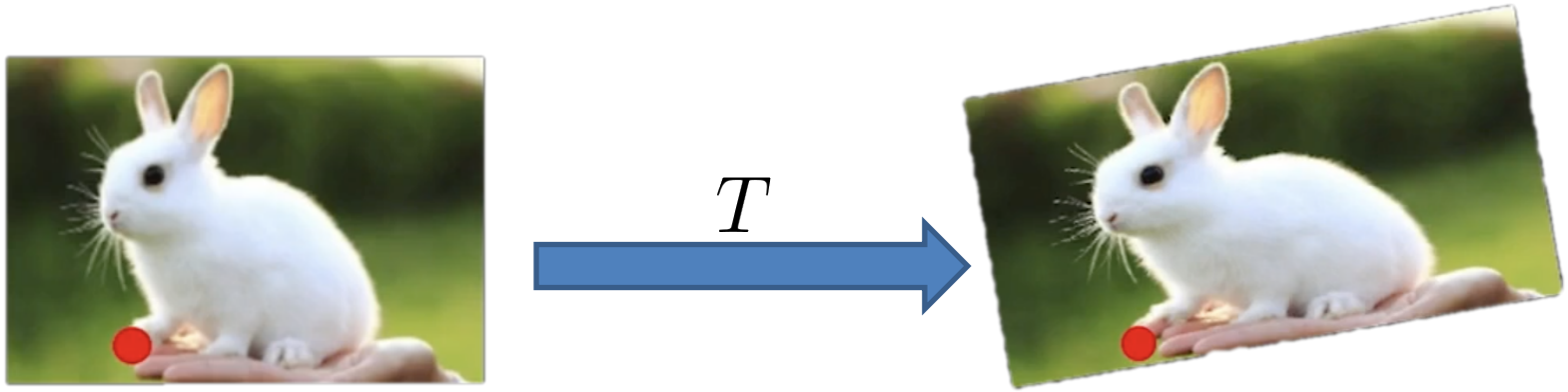
Affine

Projective

Barrel

Transformation T is the same over entire domain
often can be described by just a few parameters

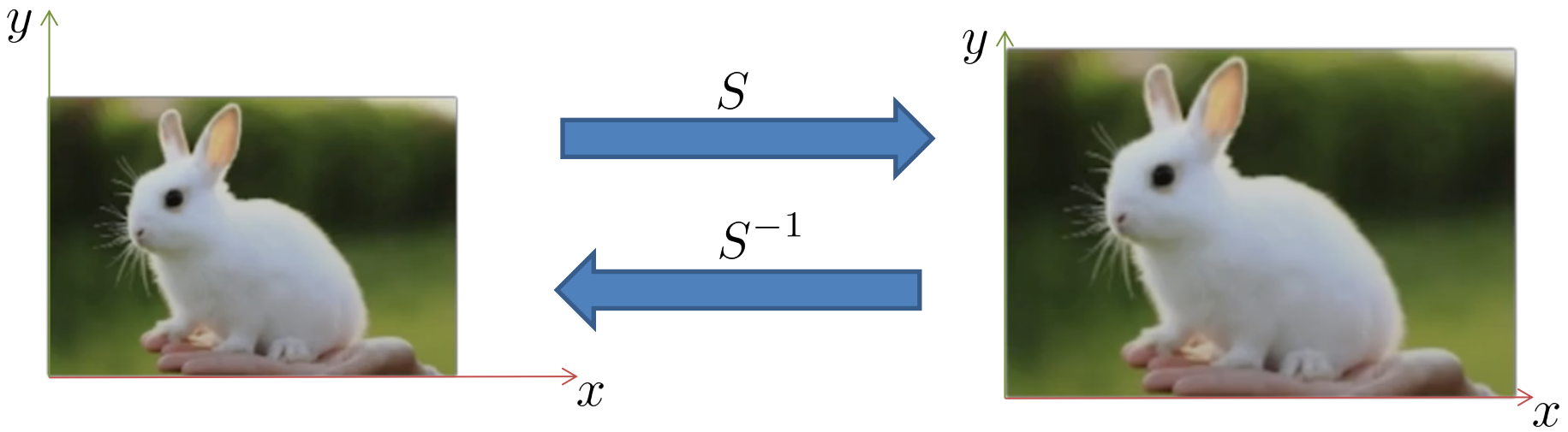
2x2 Linear Transformations



T can be represented by a matrix.

$$\mathbf{p}_2 = T\mathbf{p}_1 \qquad \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = T \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \qquad \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Scaling(Stretching and Squishing)



Forward:

$$x_2 = ax_1 \qquad y_2 = by_1$$

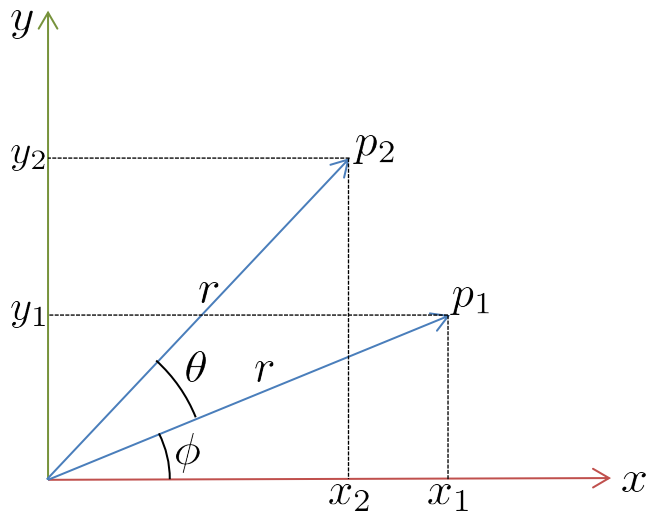
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Inverse:

$$x_1 = \frac{1}{a}x_2 \qquad y_1 = \frac{1}{b}y_2$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = S^{-1} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1/a & 0 \\ 0 & 1/b \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

2D Rotation



$$x_1 = r\cos(\phi)$$

$$y_1 = r\sin(\phi)$$

$$x_2 = r\cos(\phi+\theta)$$

$$x_2 = r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta)$$

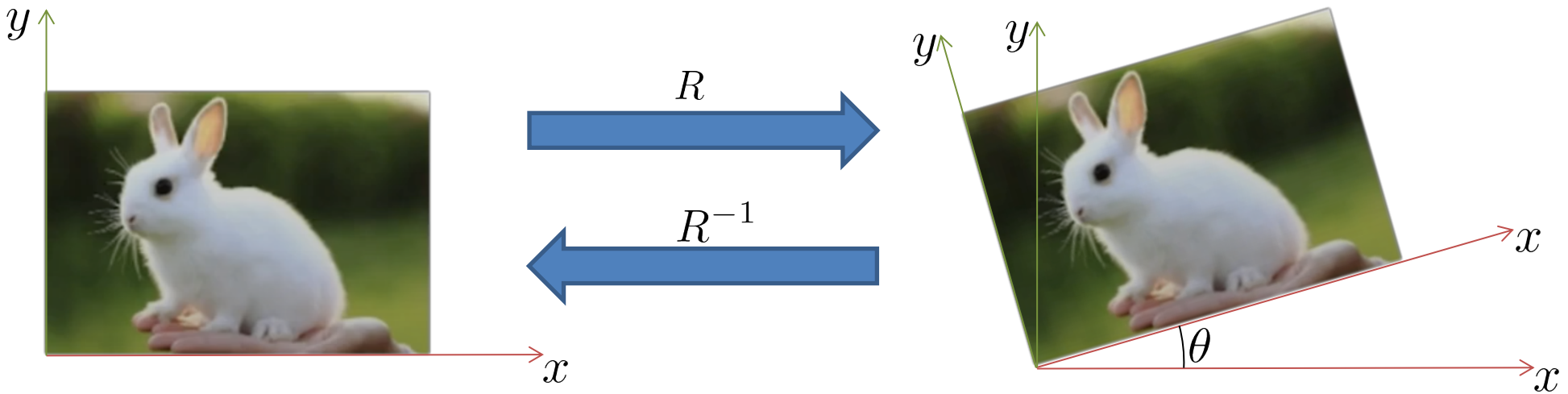
$$x_2 = x_1\cos(\theta) - y_1\sin(\theta)$$

$$y_2 = r\sin(\phi+\theta)$$

$$y_2 = r\cos(\phi)\sin(\theta) + r\sin(\phi)\cos(\theta)$$

$$y_2 = x_1\sin(\theta) + y_1\cos(\theta)$$

Rotation



Forward:

$$x_2 = x_1 \cos(\theta) - y_1 \sin(\theta)$$

$$y_2 = x_1 \sin(\theta) + y_1 \cos(\theta)$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = R \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

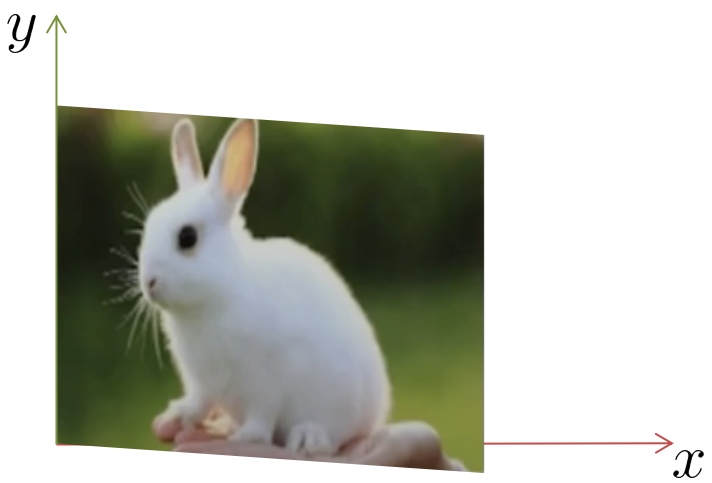
Inverse:

$$x_1 = x_2 \cos(\theta) + y_2 \sin(\theta)$$

$$y_1 = -x_2 \sin(\theta) + y_2 \cos(\theta)$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = R^{-1} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

Skew



Horizontal Skew:

$$x_2 = x_1 + m_x y_1$$

$$y_2 = y_1$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S_x \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & m_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

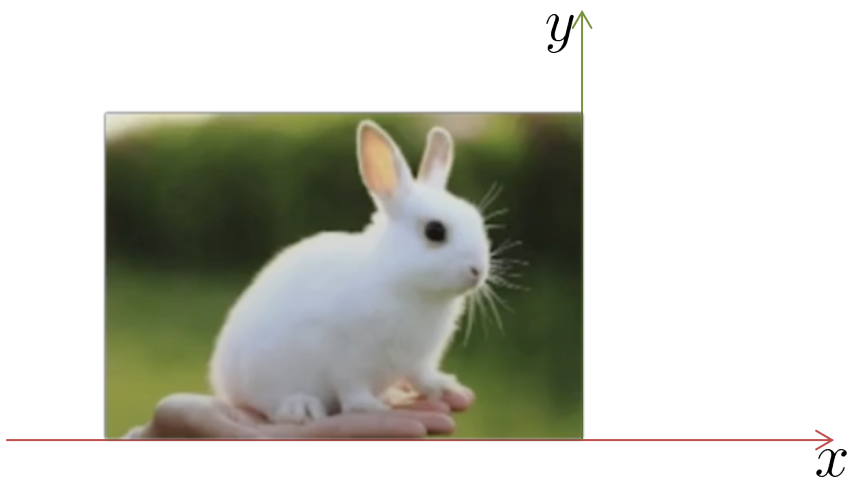
Vertical Skew:

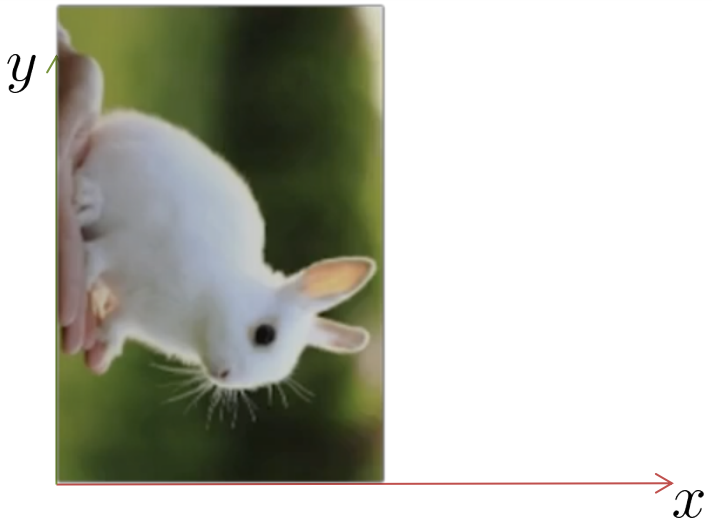
$$x_2 = x_1$$

$$y_2 = m_y x_1 + y_1$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S_y \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ m_y & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Mirrow





Mirrow about Y-axis:

$$x_2 = -x_1$$

$$y_2 = y_1$$

$$M_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Mirrow about line $y = x$:

$$x_2 = y_1$$

$$y_2 = x_1$$

$$M_{xy} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

2x2 Matrix Transformations

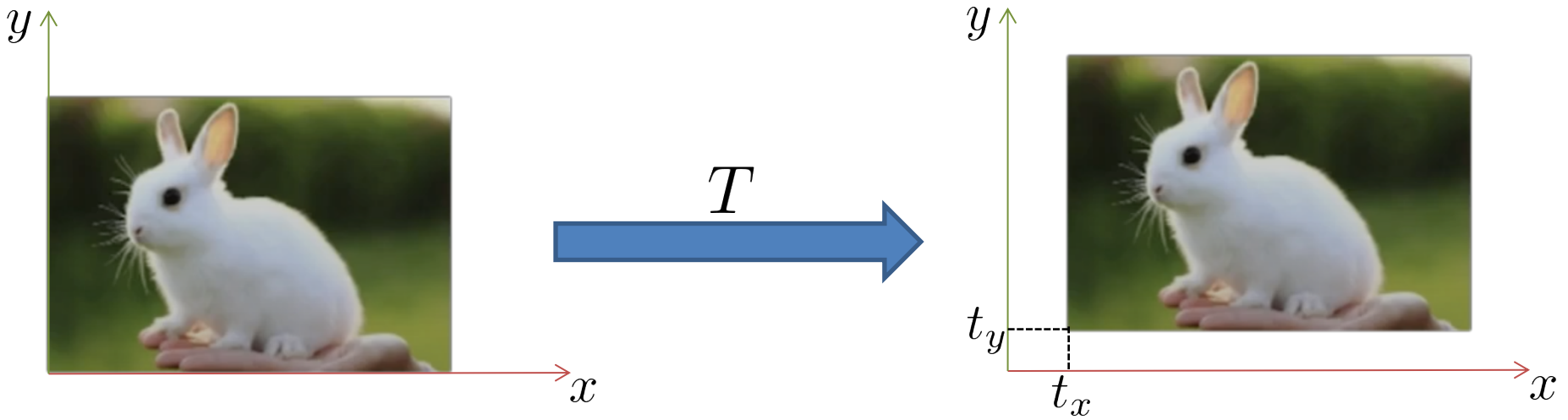
Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

- Origin maps to the origin
- Lines map to lines
- Parallel lines remain parallel
- Closed under composition

$$\left.\begin{aligned} \mathbf{p}_2 &= T_{21}\mathbf{p}_1 \\ \mathbf{p}_3 &= T_{32}\mathbf{p}_2 \\ \mathbf{p}_3 &= T_{31}\mathbf{p}_1 \end{aligned}\right\} \mathbf{p}_3 = T_{32}\mathbf{p}_2 = T_{32}T_{21}\mathbf{p}_1 \Rightarrow T_{31} = T_{32}T_{21}$$

Translation



$$x_2 = x_1 + t_x \qquad y_2 = y_1 + t_y$$

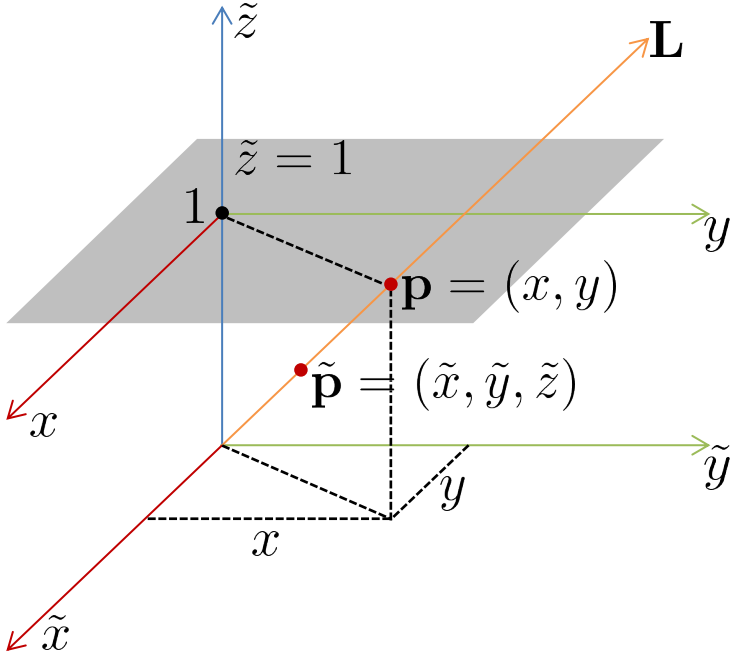
Can translation be expressed as a 2x2 matrix? No

Homogenous Coordinates

The homogenous representation of a 2D point $\mathbf{p} = (\boldsymbol{x}, \boldsymbol{y})$ is a 3D point $\tilde{\mathbf{p}} = (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})$. The third coordinate $\tilde{\boldsymbol{z}} \neq 0$ is fictitious such that:

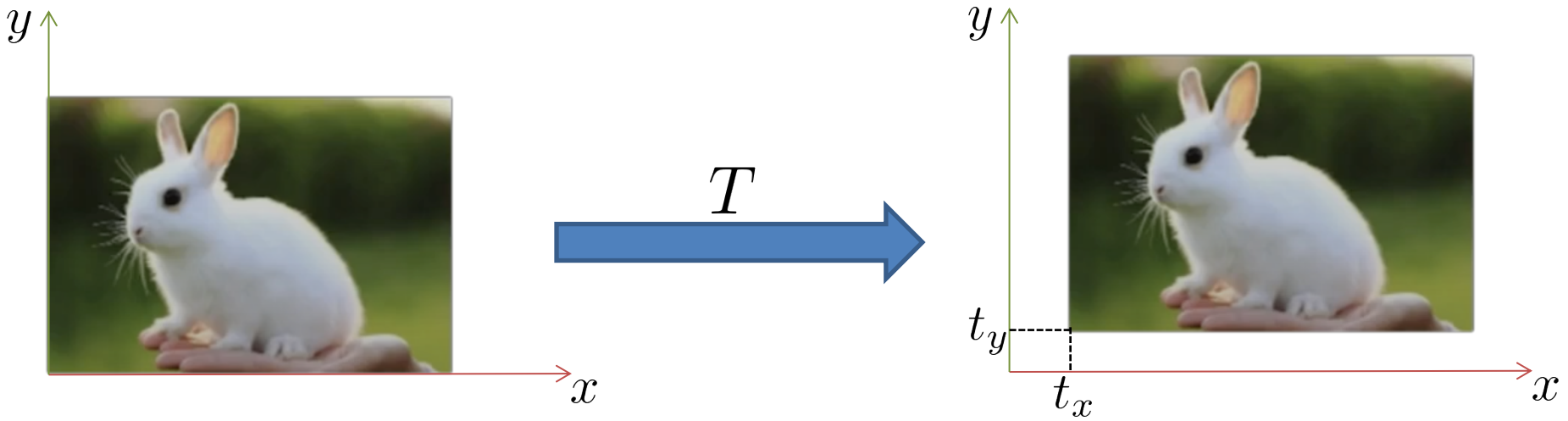
$$x = \frac{\tilde{x}}{\tilde{z}} \qquad y = \frac{\tilde{y}}{\tilde{z}}$$

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{z}x \\ \tilde{z}y \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \tilde{\mathbf{p}}$$



Every point on line **L**(except origin) represent the homogenous coordinate of $\mathbf{p}(\boldsymbol{x}, \boldsymbol{y})$

Translation



$$x_2 = x_1 + t_x \qquad y_2 = y_1 + t_y$$

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Scaling, Rotation, Skew, Translation

$$\begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Scaling

$$\begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} 1 & m_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Skew

$$\begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

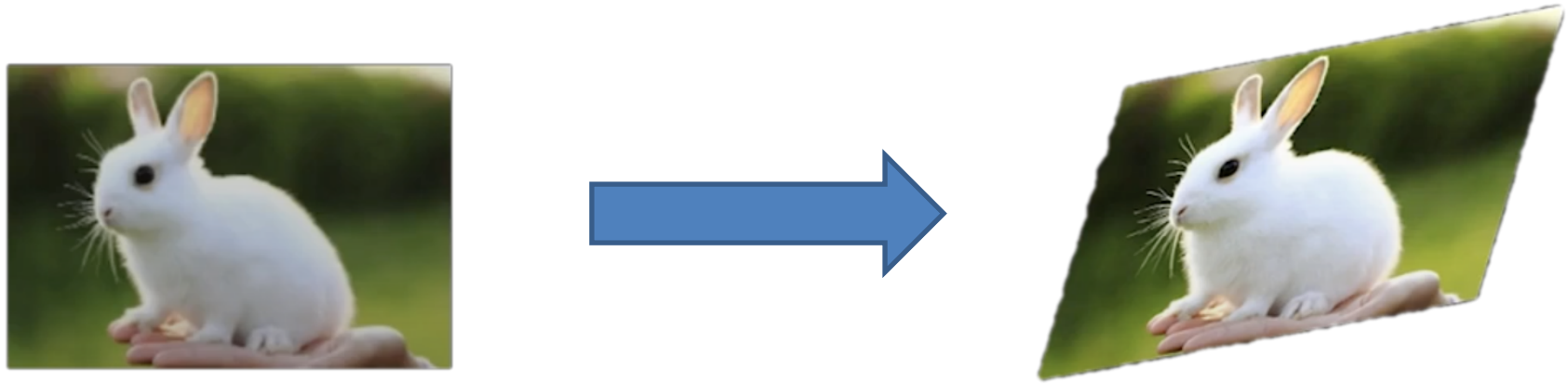
Rotation

Composition of these transformations?

Affine Transformation

Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x_1} \\ \tilde{y_1} \\ \tilde{z_1} \end{bmatrix}$$



Affine Transformation

Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x_1} \\ \tilde{y_1} \\ \tilde{z_1} \end{bmatrix}$$

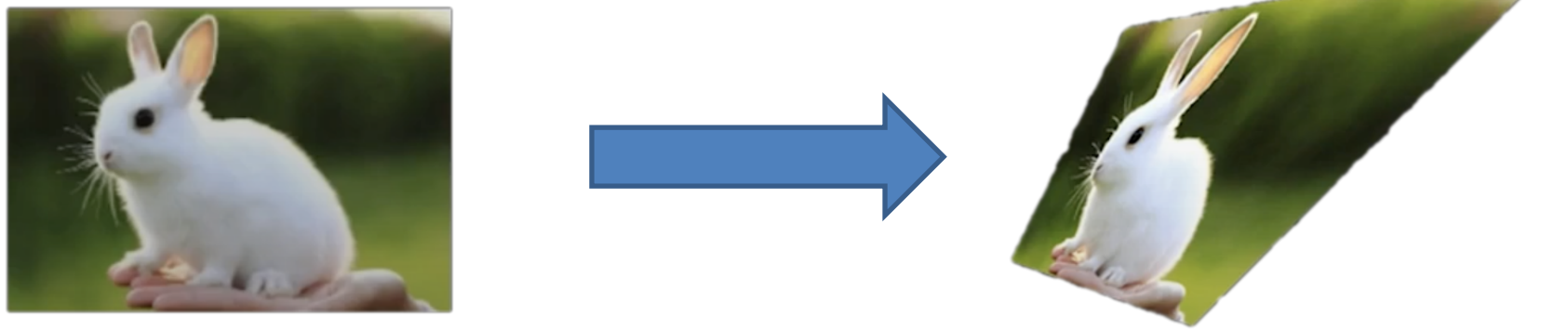
- Origin does not necessarily map to the origin
- Lines map to lines
- Parallel lines remain parallel
- Closed under composition

Projective Transformation

Any transformation of the form:

$$\begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x_1} \\ \tilde{y_1} \\ \tilde{z_1} \end{bmatrix}$$

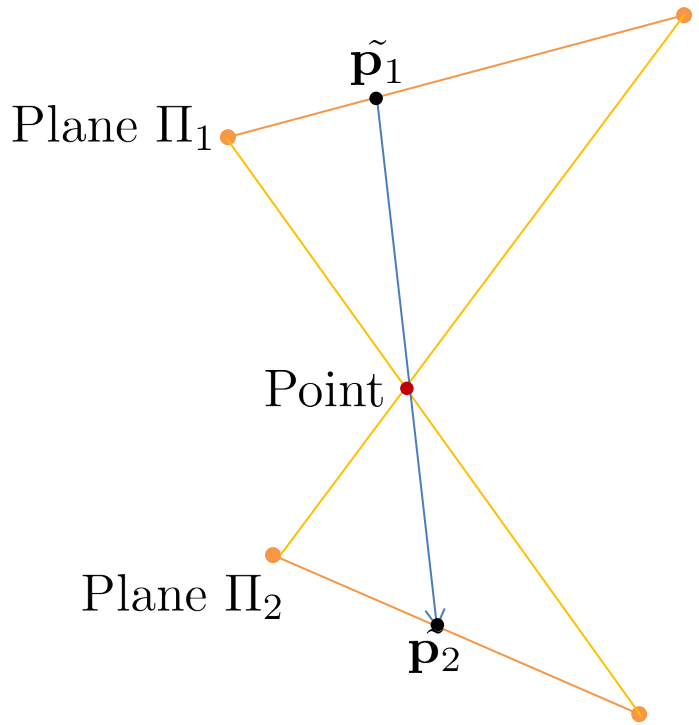
$$\tilde{\mathbf{p}_2} = H\tilde{\mathbf{p}_1}$$



Also called Homography

Projective Transformation

Mapping of one plane to another through a point



$$\tilde{\mathbf{p}}_2 = H\tilde{\mathbf{p}}_1$$

$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix}$$

Same as imaging a plane through a pinhole

Projective Transformation

Homography can only be defined up to a scale.

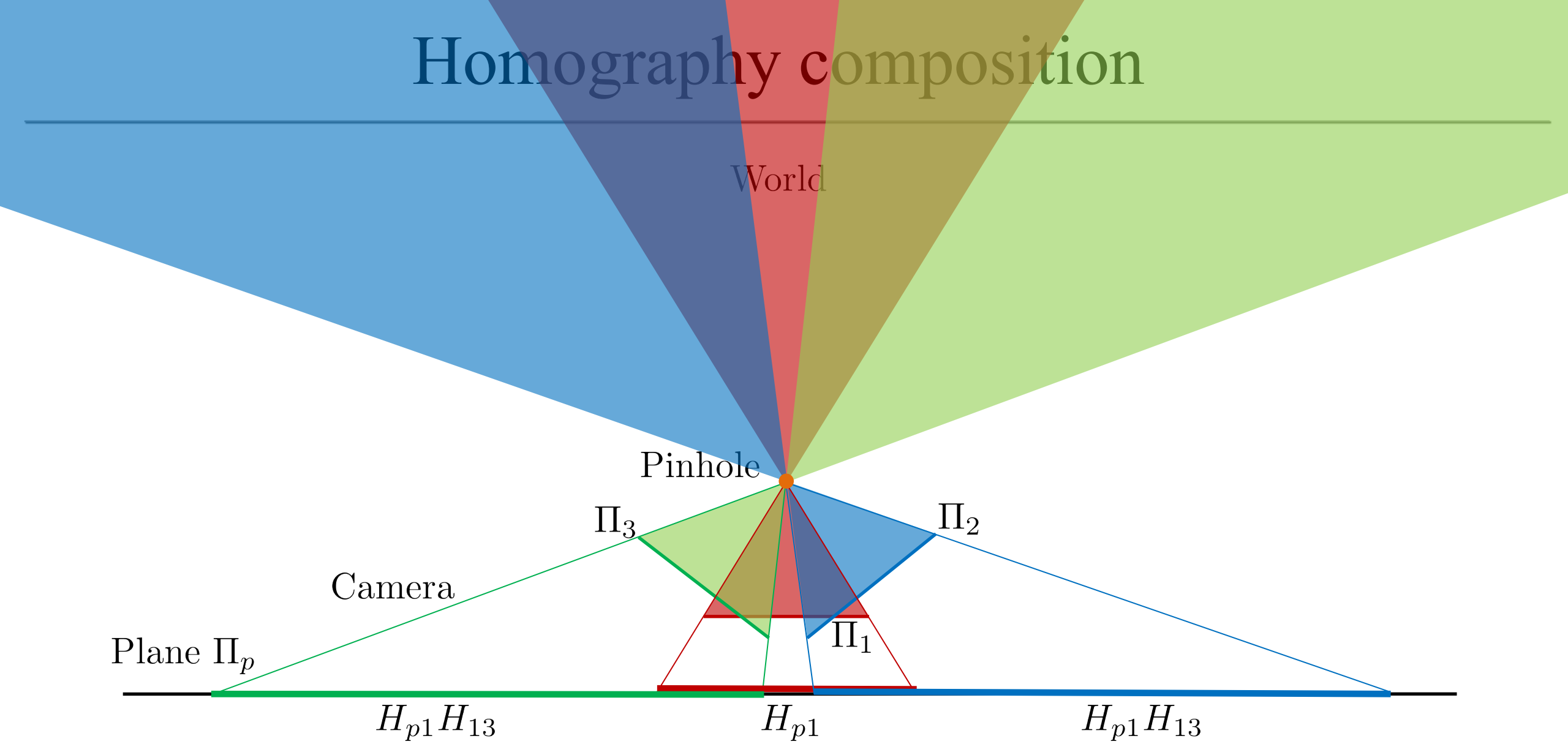
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x_1} \\ \tilde{y_1} \\ \tilde{z_1} \end{bmatrix} = \begin{bmatrix} \tilde{x_2} \\ \tilde{y_2} \\ \tilde{z_2} \end{bmatrix} = k \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x_1} \\ \tilde{y_1} \\ \tilde{z_1} \end{bmatrix}$$

If we fix scale such that $\sqrt{\sum (h_{ij})^2}$ then 8 free parameters

- Origin does not necessarily map to the origin
- Lines map to lines
- Parallel lines do not necessarily remain parallel
- Closed under composition

Remember Vanishing Points?

Homography composition



Useful in stitching planar panoramas

Computing Homography

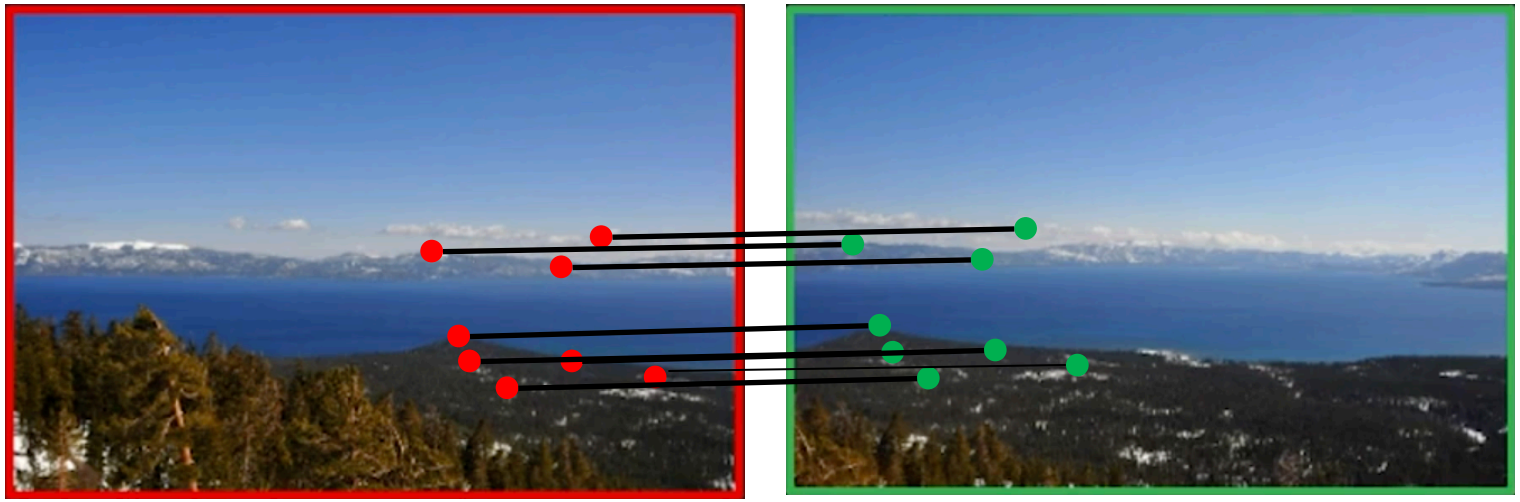
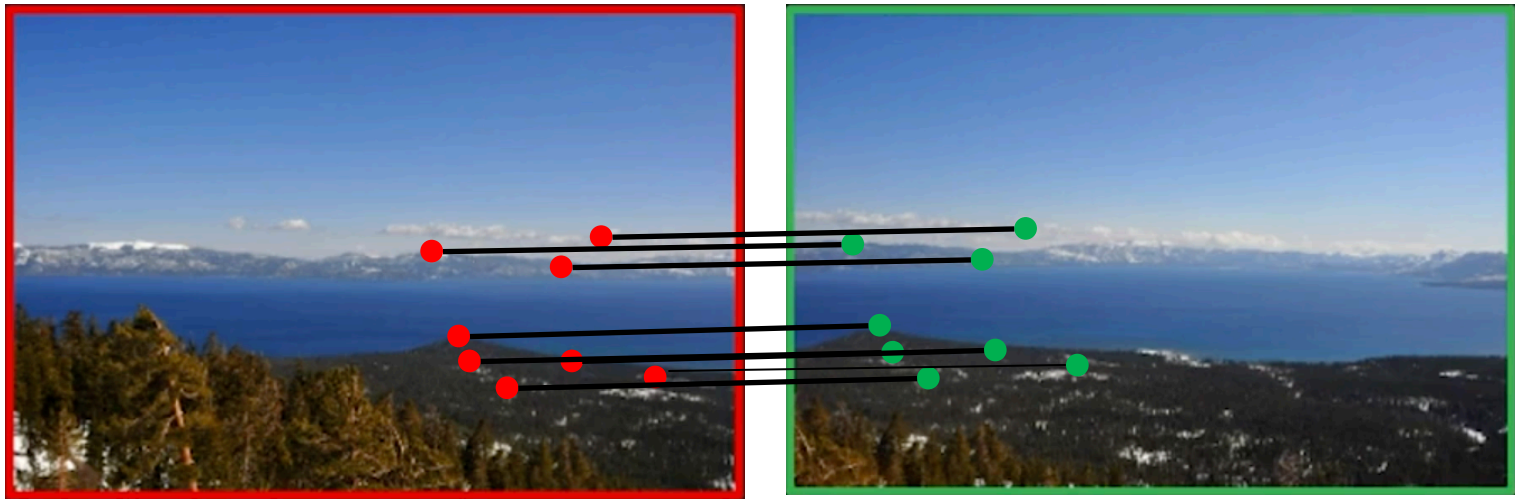


Image 1 Image 2

Given a set of matching features/points between image images 1 and 2, find the homography H that best "agrees" with the matches.

Computing Homography



Source Image

Destination Image

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{x_d} \\ \tilde{y_d} \\ \tilde{z_d} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix}$$

How many unknows? 9 ...But 8 degrees of freedom

How many minimum pairs of matching points? 4

Computing Homography

For a given pair *i* of corresponding points:

$$x_d^{(i)} = \frac{\tilde{x}_d^{(i)}}{\tilde{z}_d^{(i)}} = \frac{h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}}$$

$$y_d^{(i)} = \frac{\tilde{y}_d^{(i)}}{\tilde{z}_d^{(i)}} = \frac{h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}}$$

Rearranging the terms:

$$x_d^{(i)}(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}) = h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13}$$

$$y_d^{(i)}(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}) = h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23}$$

Computing Homography

$$x_d^{(i)}(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}) = h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13}$$
$$y_d^{(i)}(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}) = h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23}$$

Rearranging the terms and writing as linear equation:

$$\begin{bmatrix} x_s^{(i)} & y_s^{(i)} & 1 & 0 & 0 & 0 & -x_d^{(i)}x_s^{(i)} & -x_d^{(i)}y_s^{(i)} & -x_d^{(i)} \\ 0 & 0 & 0 & x_s^{(i)} & y_s^{(i)} & 1 & -y_d^{(i)}x_s^{(i)} & -y_d^{(i)}y_s^{(i)} & -y_d^{(i)} \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

(Known)

h

(Unknown)

Computing Homography

Combining the equation for all corresponding points:

$$\underbrace{\begin{bmatrix} x_s^{(1)} & y_s^{(1)} & 1 & 0 & 0 & 0 & -x_d^{(1)}x_s^{(1)} & -x_d^{(1)}y_s^{(1)} & -x_d^{(1)} \\ 0 & 0 & 0 & x_s^{(1)} & y_s^{(1)} & 1 & -y_d^{(1)}x_s^{(1)} & -y_d^{(1)}y_s^{(1)} & -y_d^{(1)} \\ & & & & & \vdots & & & \\ x_s^{(i)} & y_s^{(i)} & 1 & 0 & 0 & 0 & -x_d^{(i)}x_s^{(i)} & -x_d^{(i)}y_s^{(i)} & -x_d^{(i)} \\ 0 & 0 & 0 & x_s^{(i)} & y_s^{(i)} & 1 & -y_d^{(i)}x_s^{(i)} & -y_d^{(i)}y_s^{(i)} & -y_d^{(i)} \\ & & & & & \vdots & & & \\ x_s^{(n)} & y_s^{(n)} & 1 & 0 & 0 & 0 & -x_d^{(n)}x_s^{(n)} & -x_d^{(n)}y_s^{(n)} & -x_d^{(n)} \\ 0 & 0 & 0 & x_s^{(n)} & y_s^{(n)} & 1 & -y_d^{(n)}x_s^{(n)} & -y_d^{(n)}y_s^{(n)} & -y_d^{(n)} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}}_{\mathbf{h}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$\boldsymbol{A}$ (Known)

$\mathbf{h}$ (Unknown)

Solve for $\mathbf{h}$: $\boxed{\boldsymbol{A}\mathbf{h} = 0}$ such that $\|\mathbf{h}\|^2 = 1$

Constrained Least Squares

Solve for $\mathbf{h}$: $\boxed{\boldsymbol{A}\mathbf{h} = 0}$ such that $\|\mathbf{h}\|^2 = 1$

Define least squares problem:

$$\min_{\mathbf{h}} \|\boldsymbol{A}\mathbf{h}\|^2 \text{ such that } \|\mathbf{h}\|^2 = 1$$

We know that:

$$\|\boldsymbol{A}\mathbf{h}\|^2 = (\boldsymbol{A}\mathbf{h})^T \boldsymbol{A}\mathbf{h} = \mathbf{h}^T \boldsymbol{A}^T \boldsymbol{A}\mathbf{h} \quad \text{and} \quad \|\mathbf{h}\|^2 = \mathbf{h}^T\mathbf{h} = 1$$

$$\boxed{\min_{\mathbf{h}} (\mathbf{h}^T \boldsymbol{A}^T \boldsymbol{A}\mathbf{h}) \text{ such that } \|\mathbf{h}\|^2 = 1}$$

Constrained Least Squares

$$\min_{\mathbf{h}}(\mathbf{h}^T \boldsymbol{A}^T \boldsymbol{A}\mathbf{h}) \text{ such that } \|\mathbf{h}\|^2 = 1$$

Define Loss function $L(\mathbf{h}, \lambda)$:

$$L(\mathbf{h}, \lambda) = \mathbf{h}^T \boldsymbol{A}^T \boldsymbol{A}\mathbf{h} - \lambda(\mathbf{h}^T\mathbf{h} - 1)$$

Taking derivatives of $L(\mathbf{h}, \lambda)$ w.r.t $\mathbf{h}$: $2\boldsymbol{A}^T\boldsymbol{A}\mathbf{h} - 2\lambda\mathbf{h} = 0$

$$\boxed{\boldsymbol{A}^T\boldsymbol{A}\mathbf{h} = \lambda\mathbf{h}}$$ Eigenvalue Problem

Eigenvector $\mathbf{h}$ with smallest eigenvalue λ of matrix $\boldsymbol{A}^T\boldsymbol{A}$ minimizes the loss function $L(\mathbf{h})$.

Matlab: eig(A'*A) returns eigenvalues and vectors of $\boldsymbol{A}^T\boldsymbol{A}$

What could go wrong?

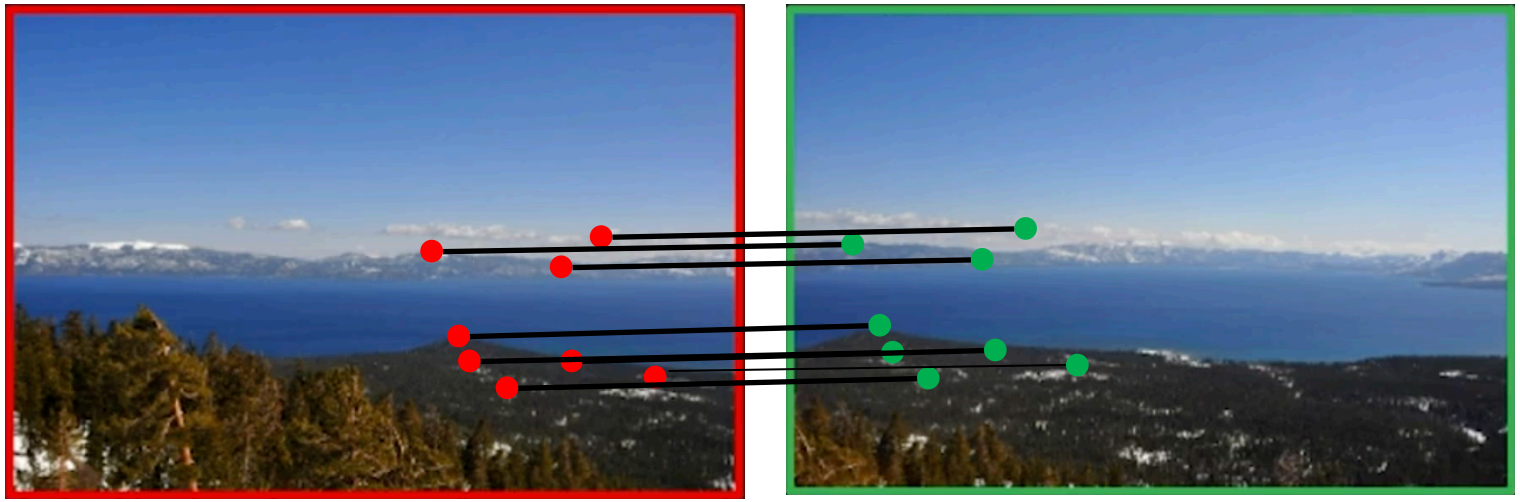


Image 1 Image 2

What could go wrong?

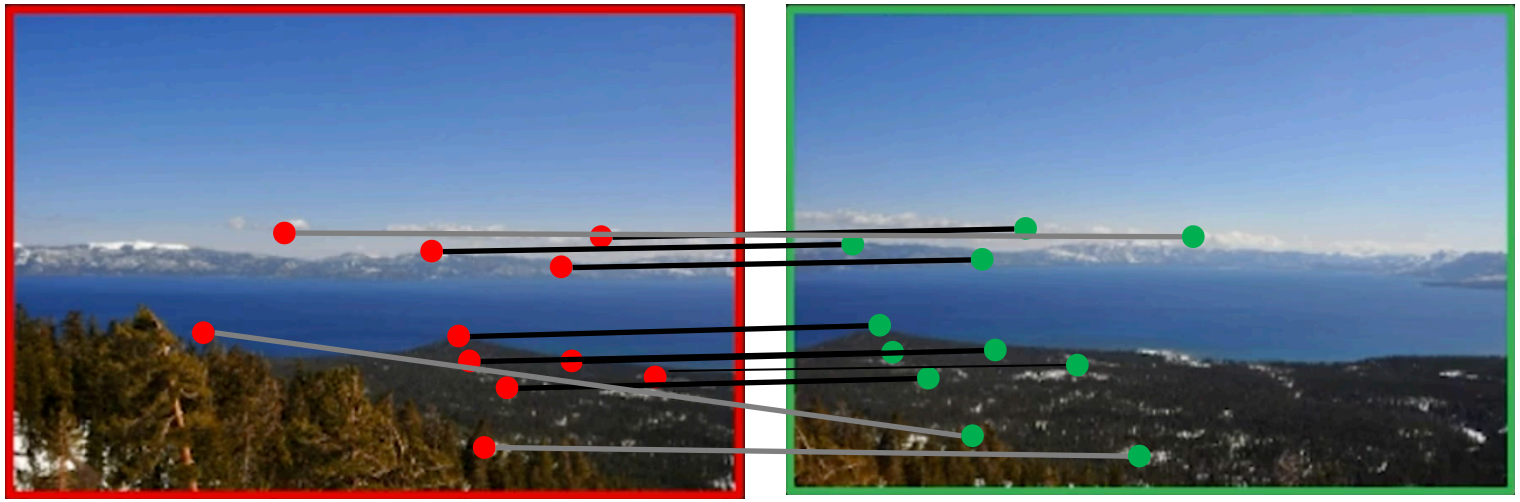


Image 1 Image 2

Outliers!

We need to robustly compute transformation in the presence of wrong matches.

If number of outliers < 50%, then RANSAC to the rescue!

RANdom SAmple Consensus

General RANSAC algorithm:

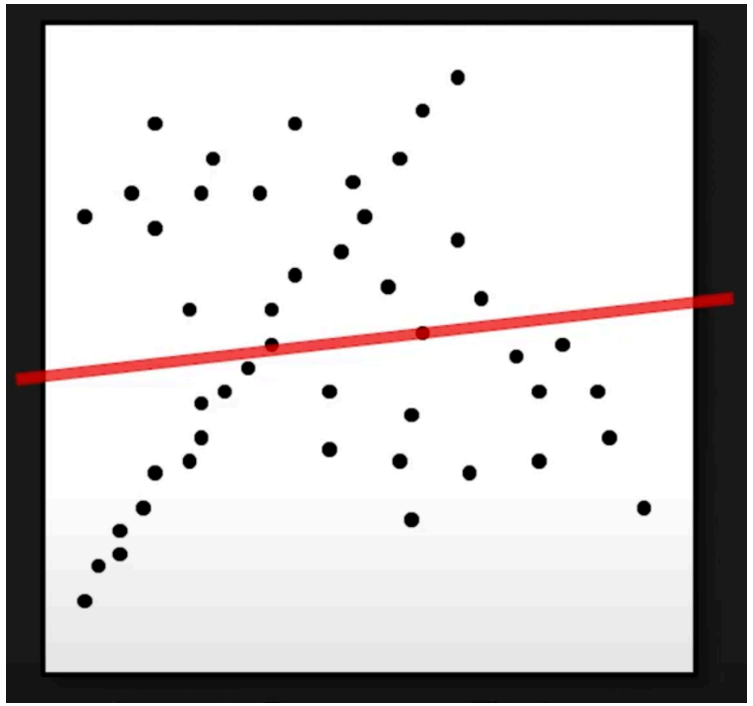
1. Randomly choose s samples. Typically s is the minimum samples to fit the model.
2. Fit the model to the randomly chosen samples.
3. Count the number M of data points(inliers) that fit the model within a measure of error ε.
4. Repeat Steps 1-3 N times
5. Choose the model that has the largest number M of inliers.

For homography:

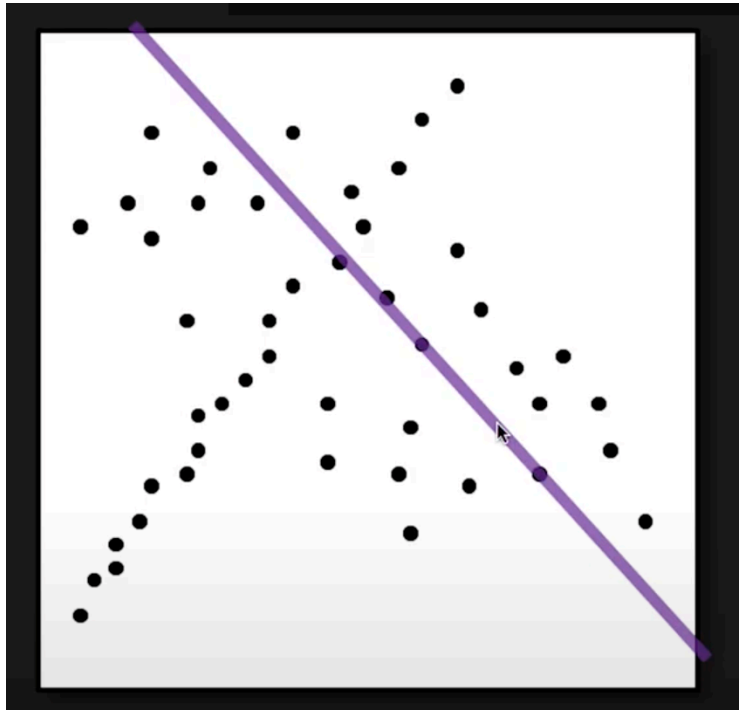
s = 4 points. ε is acceptable alignment error in pixels.

RANSAC Example: Line Fitting

Robust line fitting:



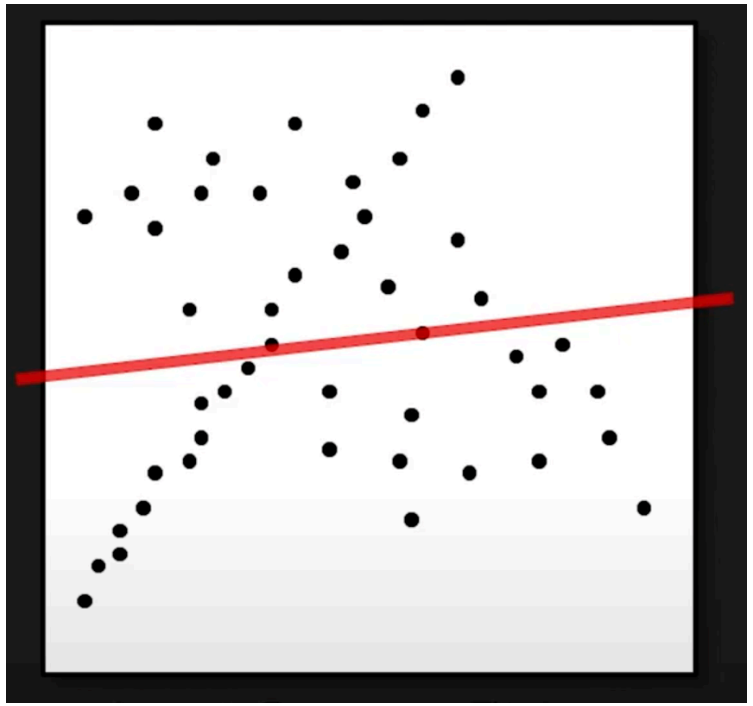
Least Squares Fitting
Inliers:2



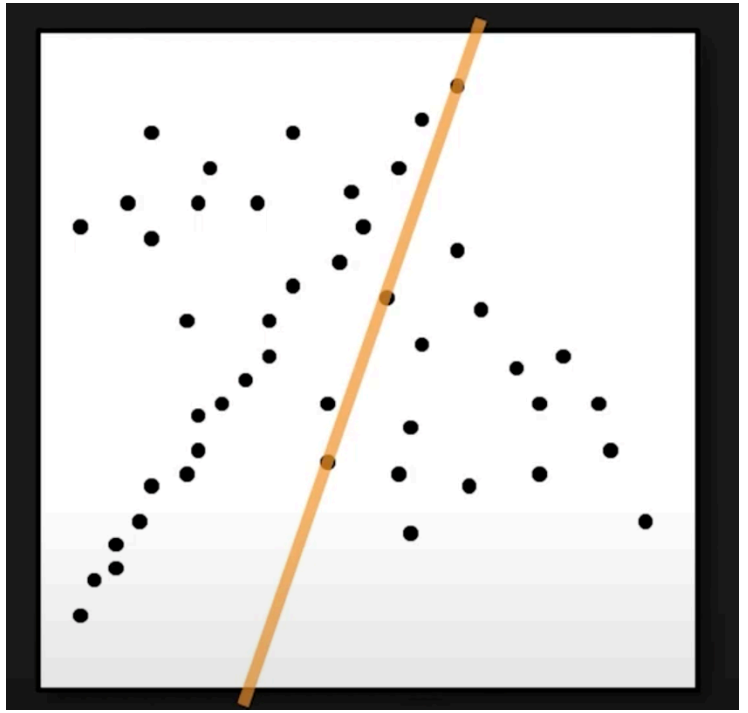
RANSAC Interation 1
Inliers:4

RANSAC Example: Line Fitting

Robust line fitting:



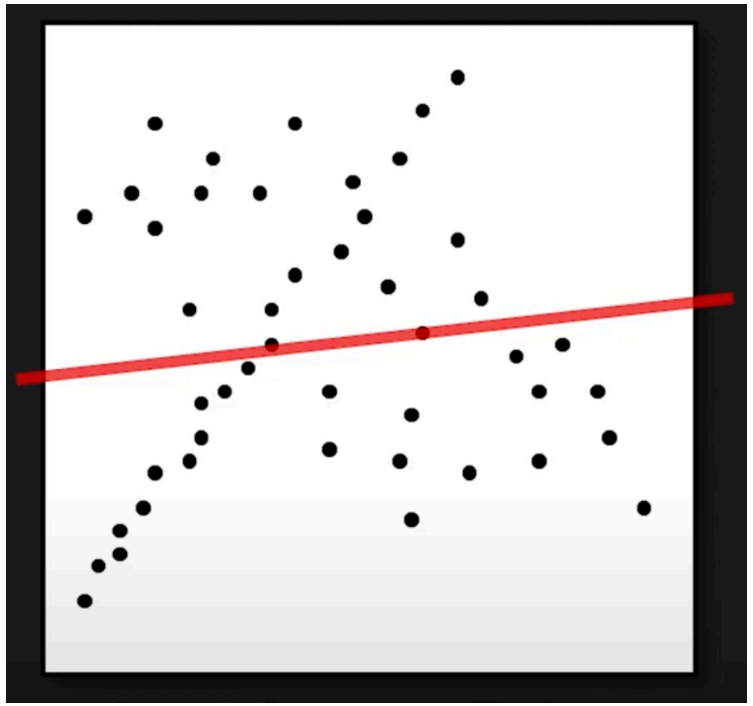
Least Squares Fitting
Inliers:2



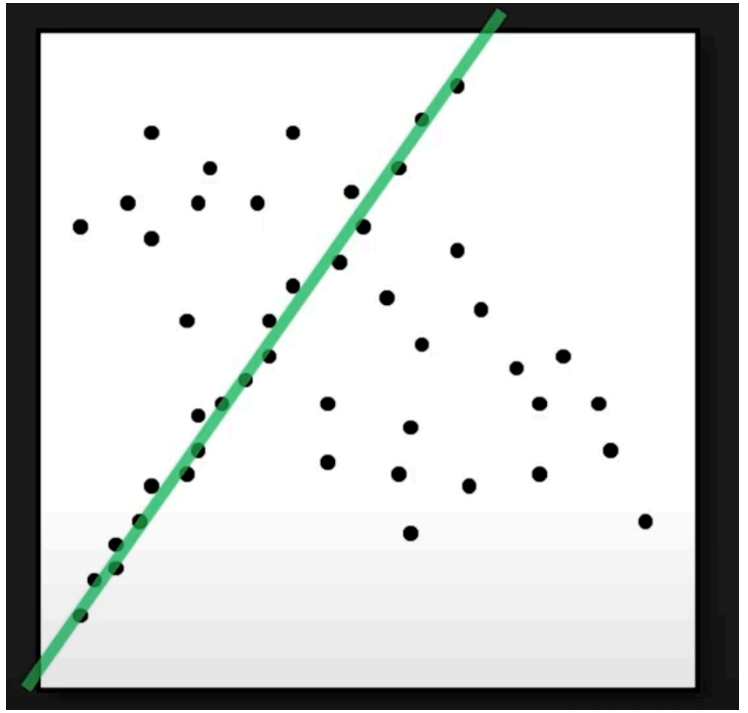
RANSAC Interation 2
Inliers:3

RANSAC Example: Line Fitting

Robust line fitting:



Least Squares Fitting
Inliers:2

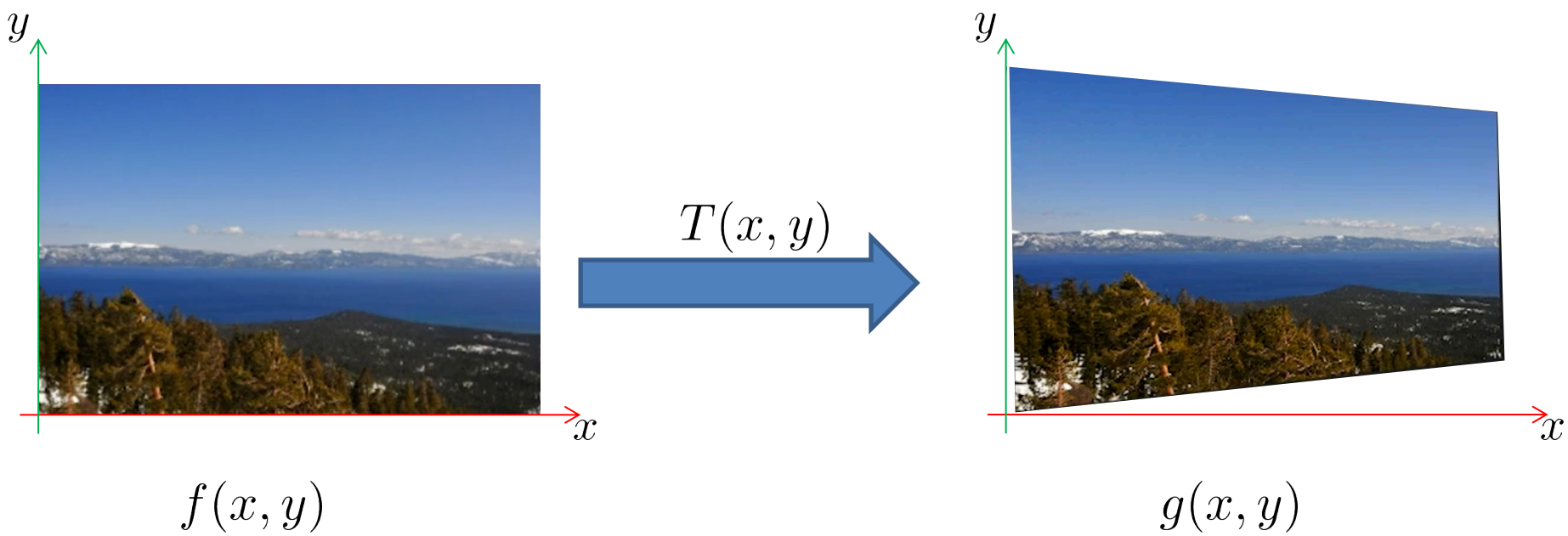


RANSAC Interation i
Inliers:20

Warping Images

Given a transformation T and a image $f(x,y)$, compute the transformed image $g(x,y)$

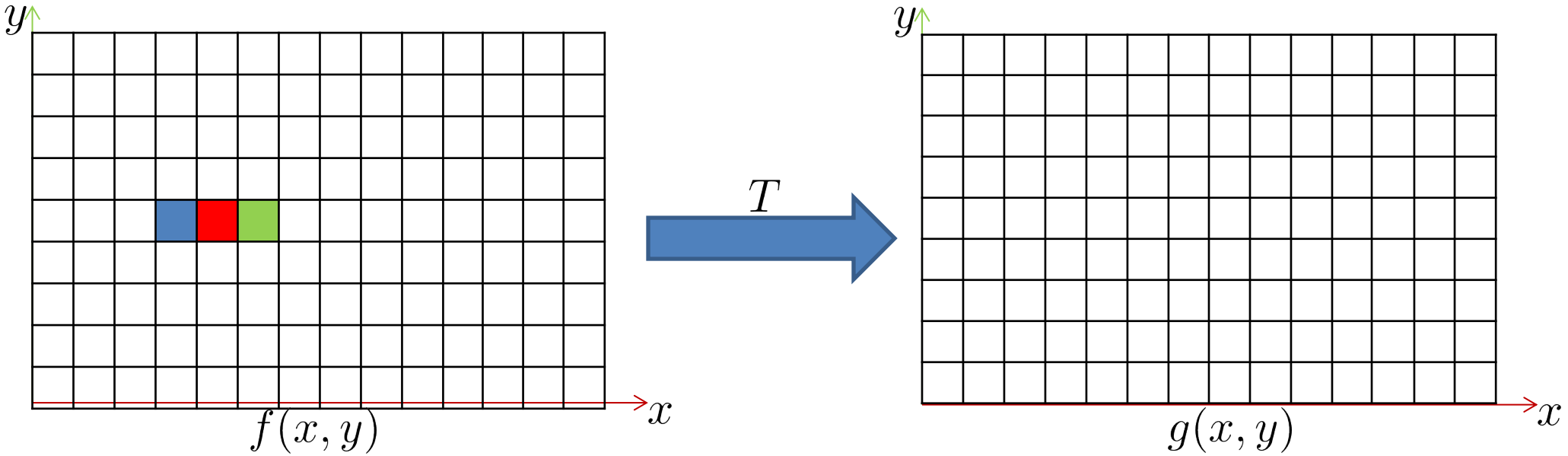
$$g(x, y) = f(T(x, y))$$



Forward Warping

Send each pixel (x,y) in $f(x,y)$ to its corresponding location $T(x,y)$ in $g(x,y)$

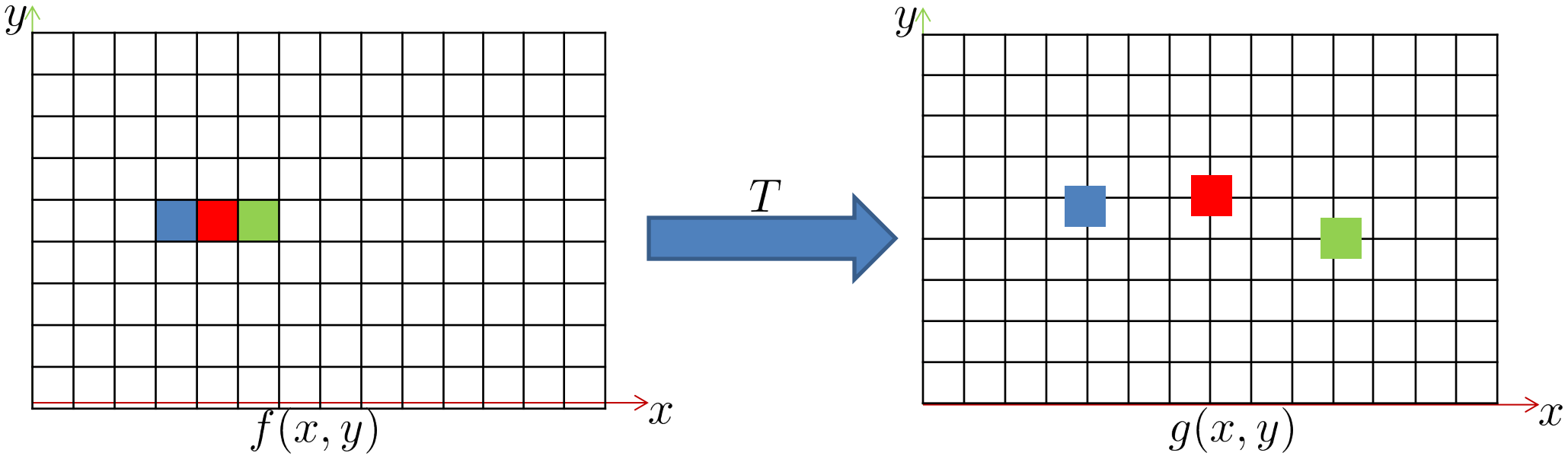
$$g(x,y) = f(T(x,y))$$



Forward Warping

Send each pixel (x,y) in $f(x,y)$ to its corresponding location $T(x,y)$ in $g(x,y)$

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



What if pixel lands in between pixels?

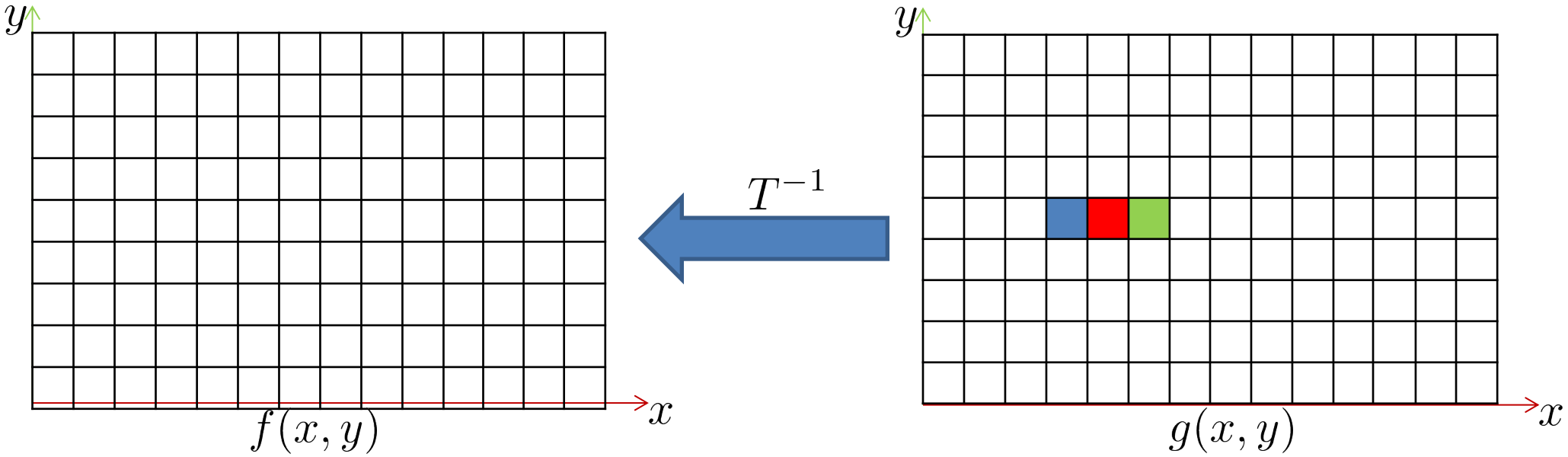
What if not all pixels in $g(x,y)$ are filled?

Can result in holes!

Backward Warping

Get each pixel (x,y) in $g(x,y)$ from its corresponding location $T^{-1}(x,y)$ in $f(x,y)$

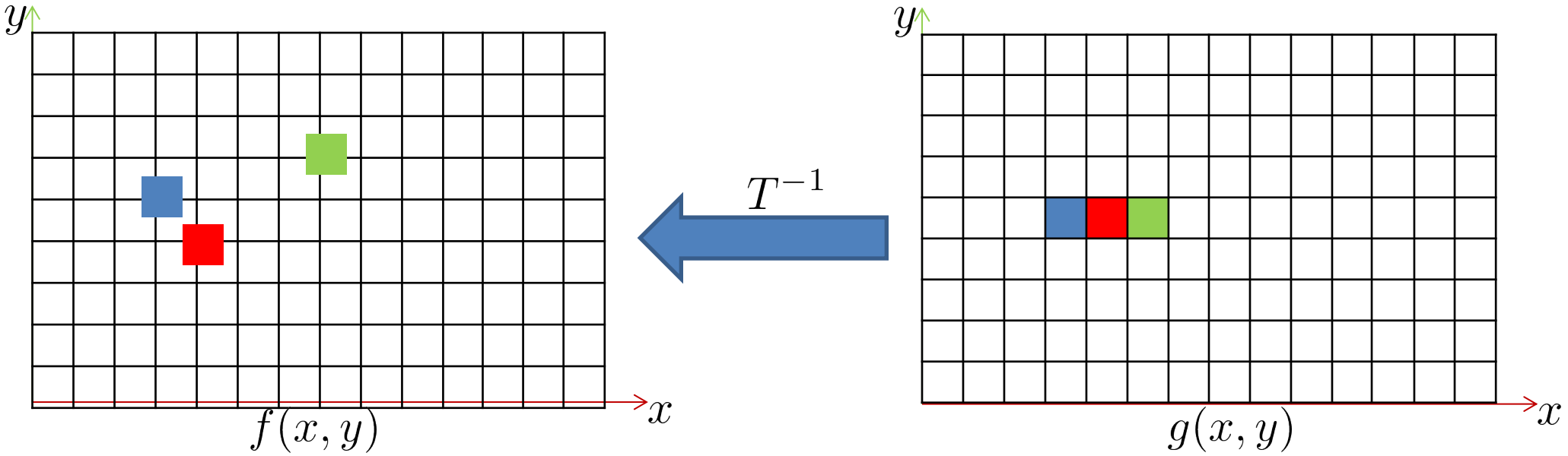
$$g(x,y) = f(T(x,y))$$



Backward Warping

Get each pixel (x,y) in $g(x,y)$ from its corresponding location $T^{-1}(x,y)$ in $f(x,y)$

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



What if pixel lands between pixels?

Use Nearest Neighbor or Interpolate

Image Alignment Process

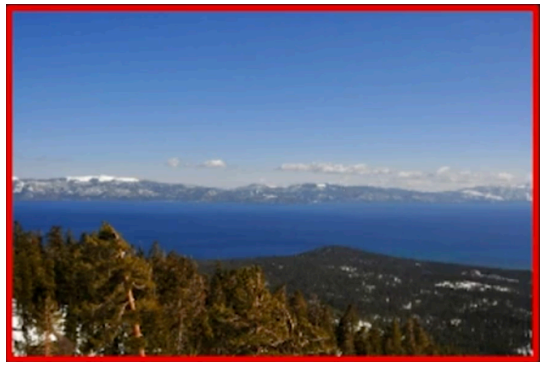


Image 1

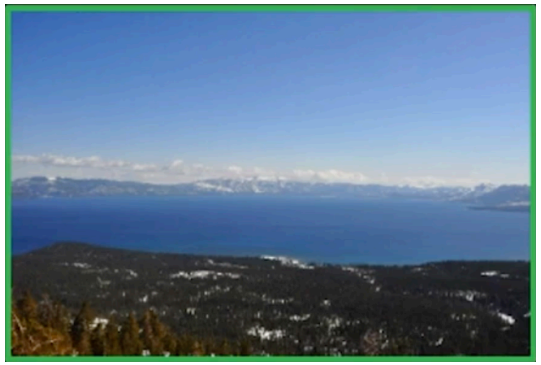


Image 2

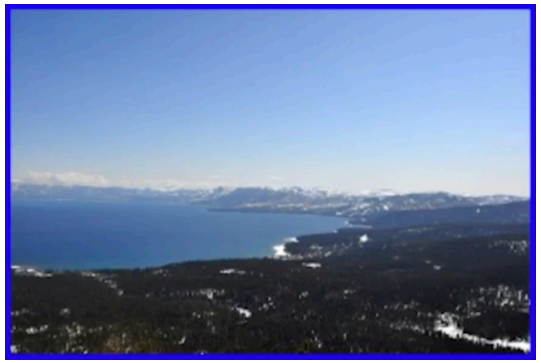
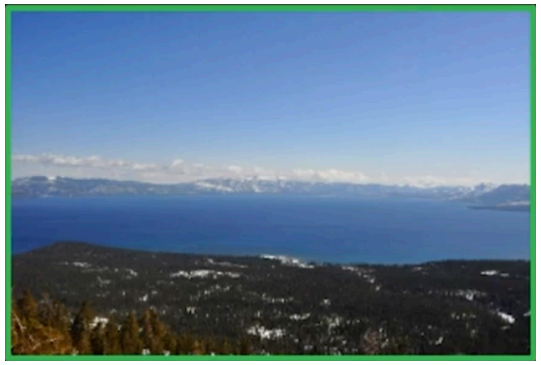
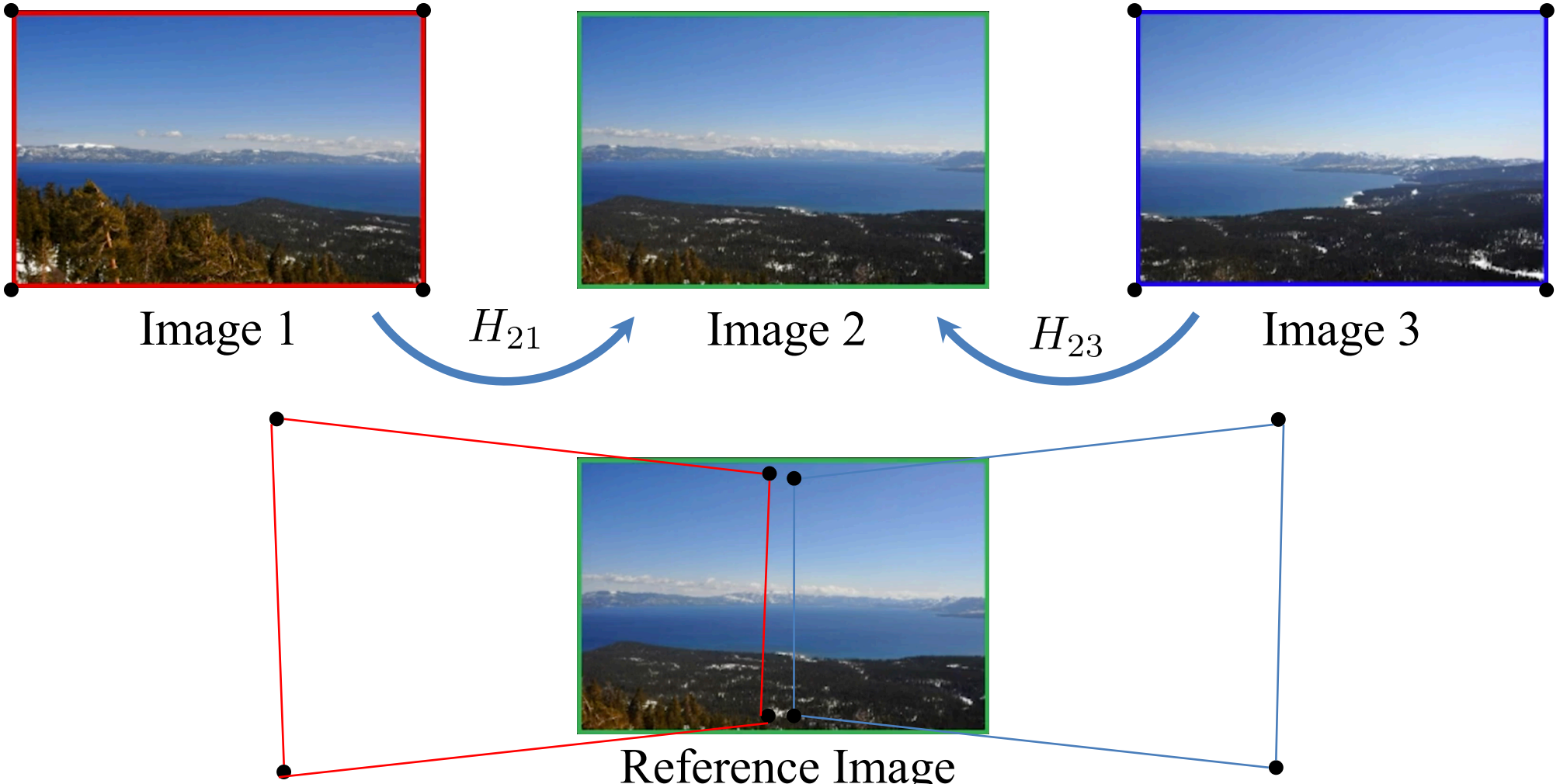


Image 3



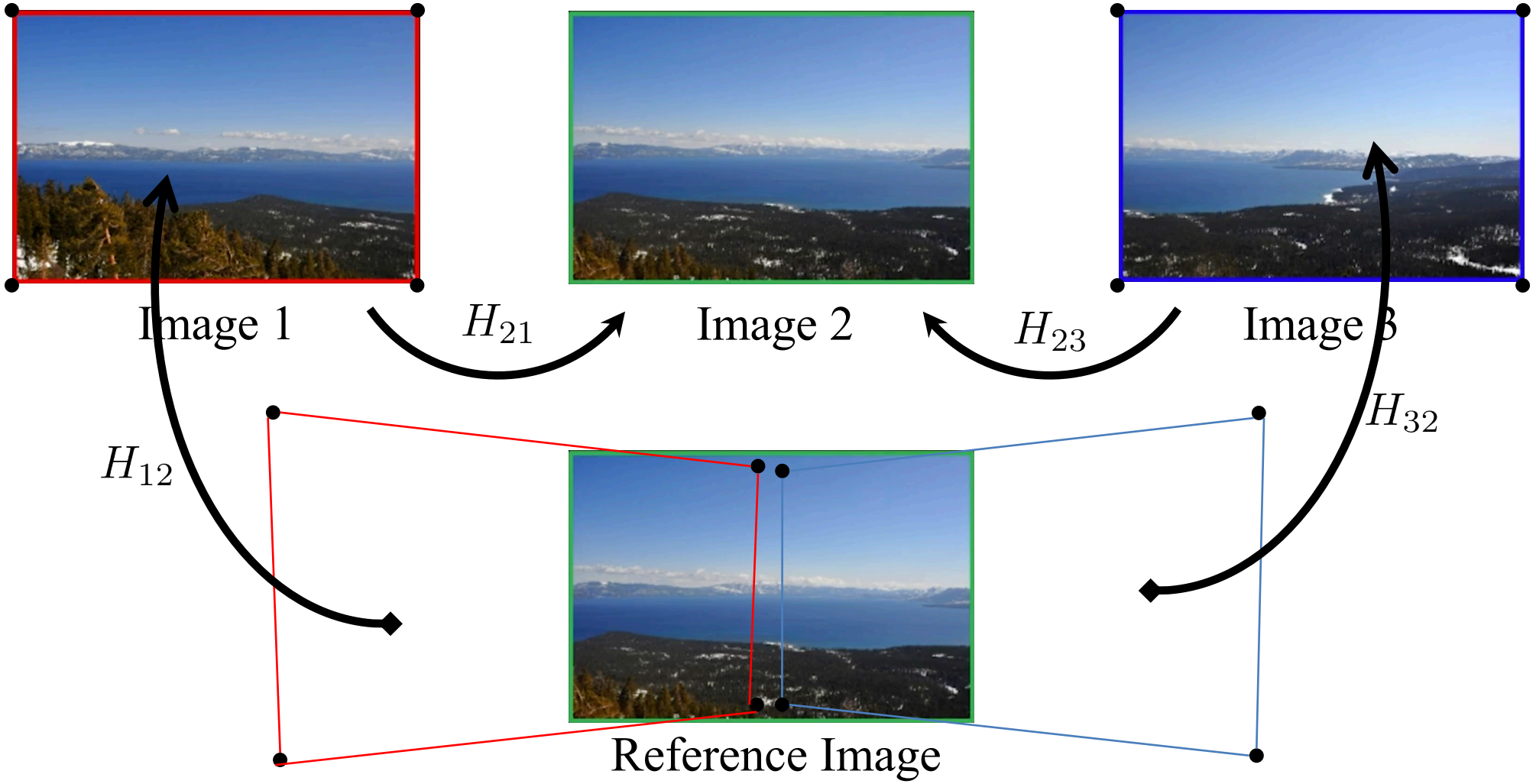
Reference Image
(Image 2)

Image Alignment Process



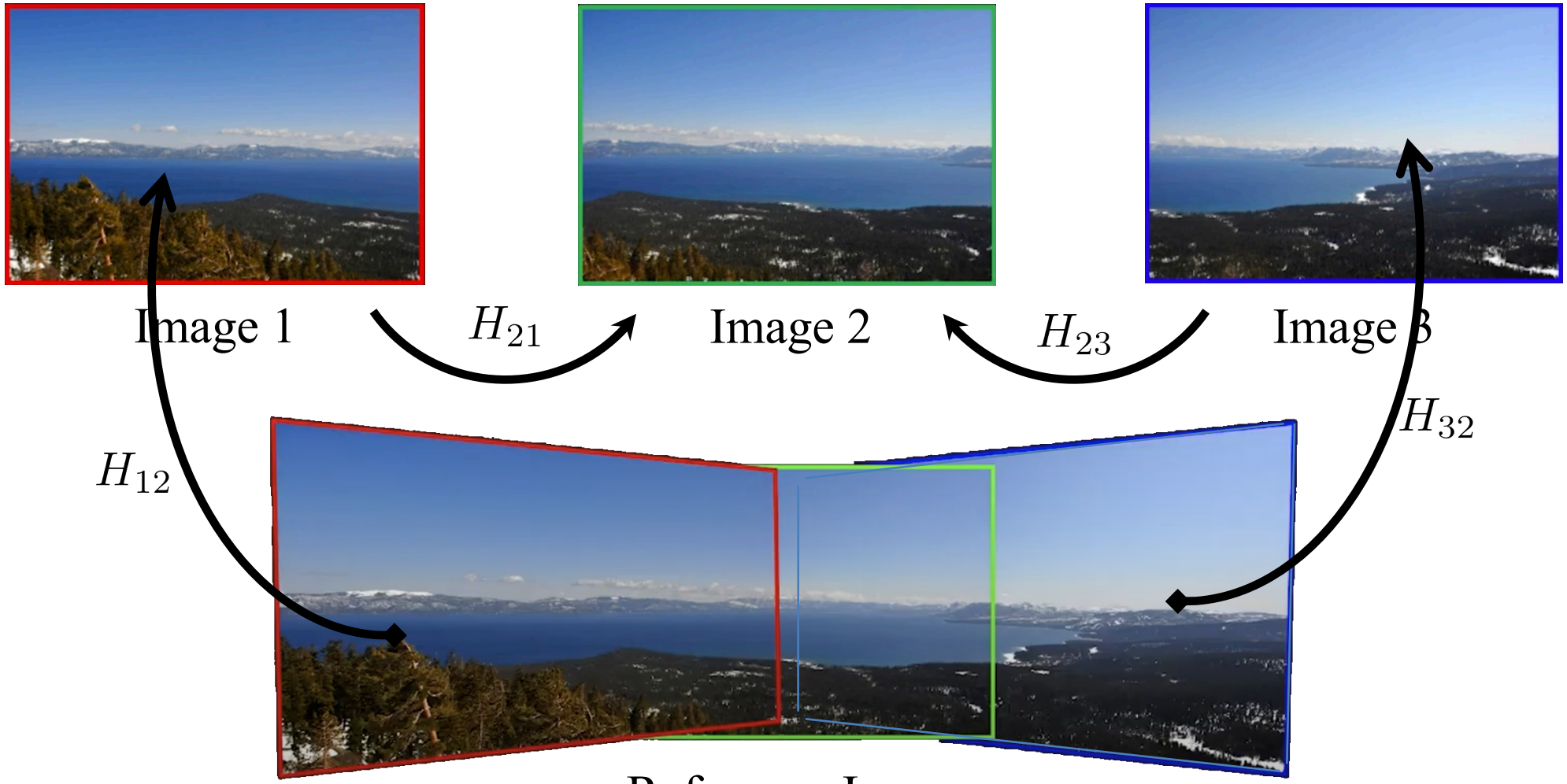
Compute the bounds of Image 1 and Image 3 in reference image space

Image Alignment Process



For each pixel within bounds, compute its location in captured image

Image Alignment Process



For each pixel within bounds, compute its location in captured image

Blending Images

Overlaid Aligned Images

Hard seams due to vignetting, exposure differences, etc.

Blending Images: Averaging

Averaged Images

Seams still visible

Blending Images

Say we want to blend images I_1 and I_2 at the center

Image I_1

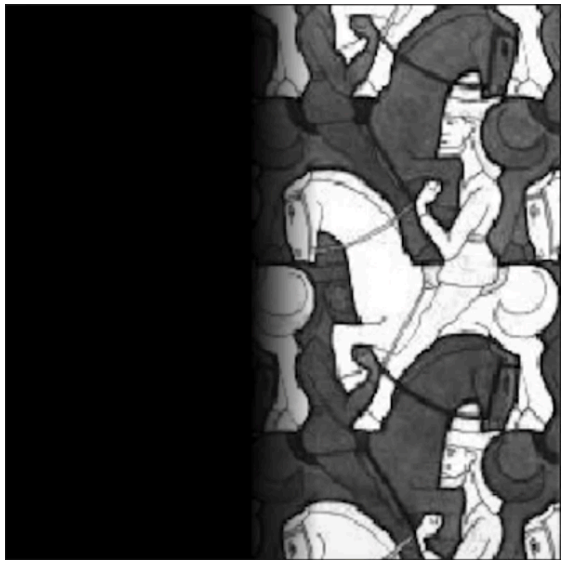


Image I_2

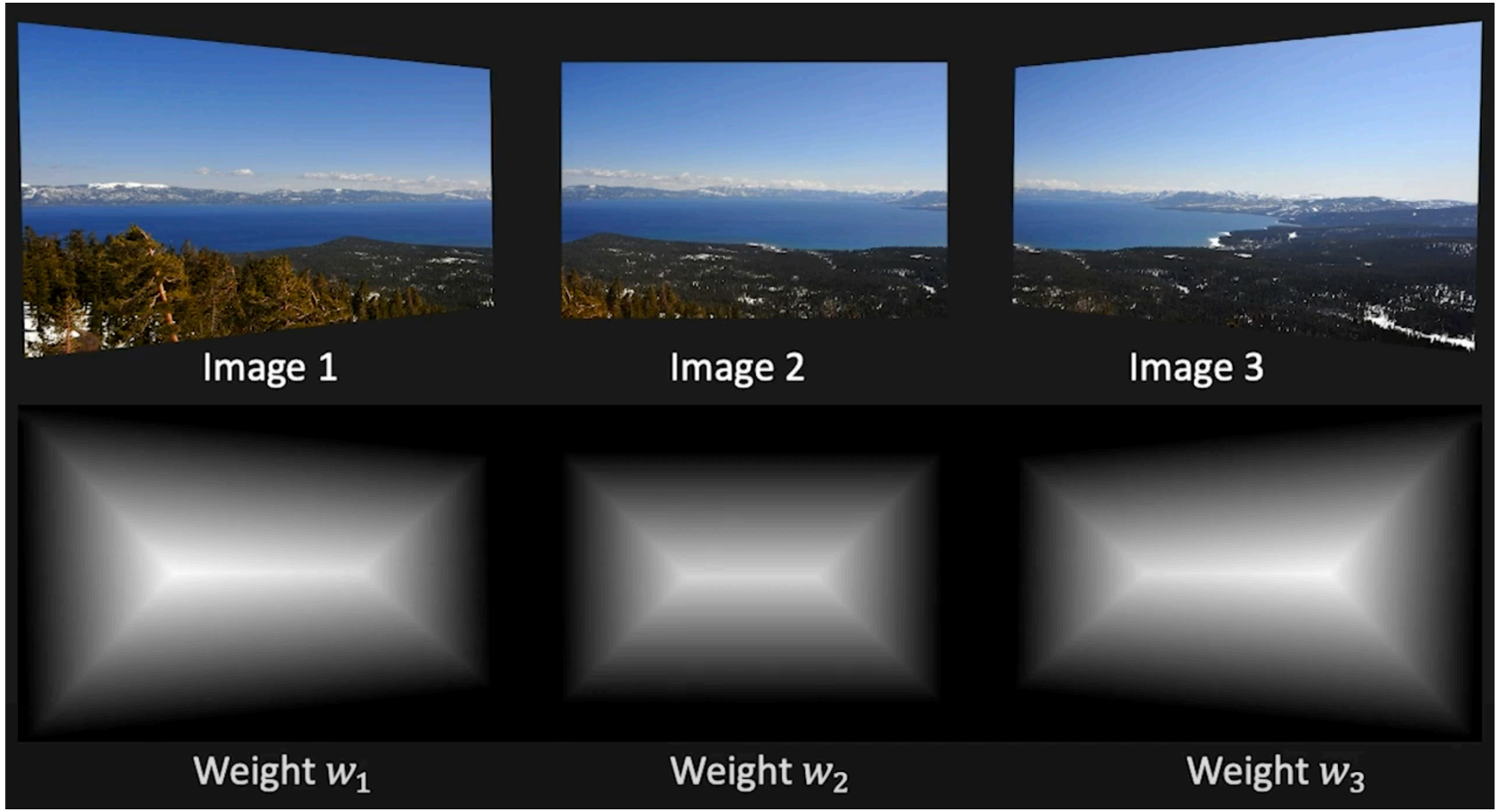
Blended Image I_{blend}

Weight w_1

Weight w_2

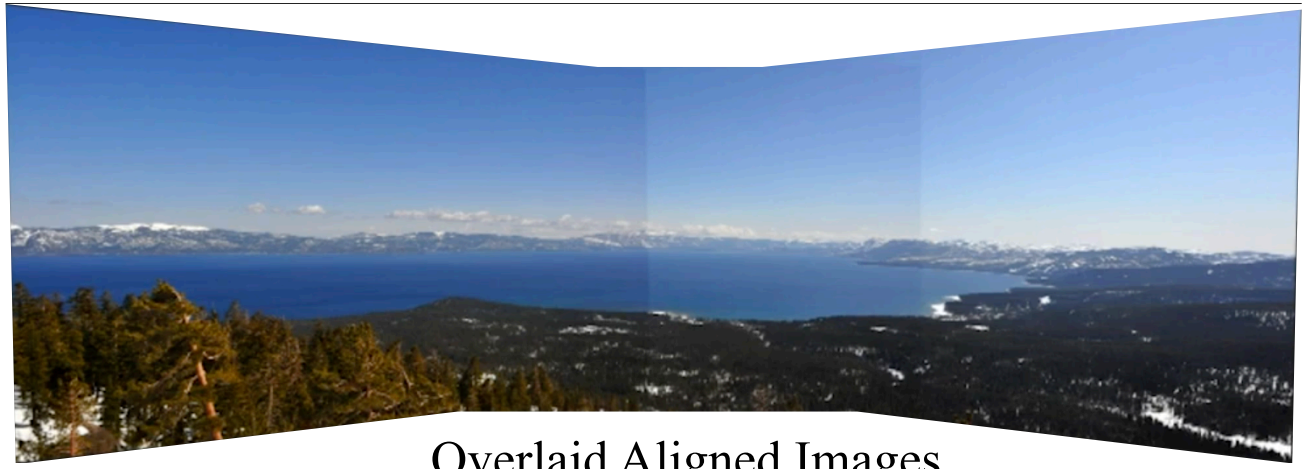
$$I_{blend} = \frac{w_1 I_1 + w_2 I_2}{w_1 + w_2}$$

Computing Weighting Functions

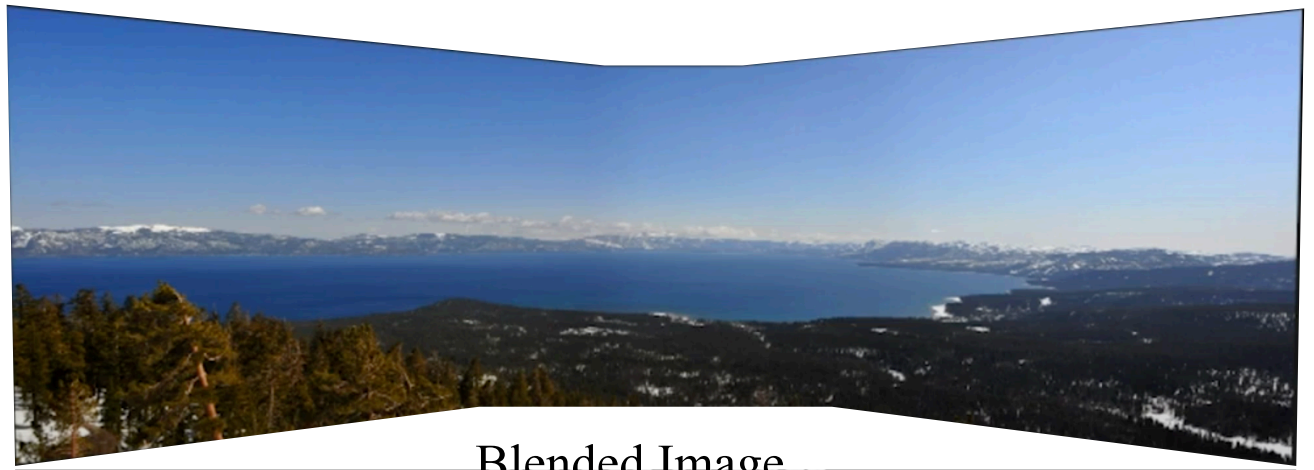


Pixels closer to the edge get a lower weight.
Ex: Distance Transform (bwdist in MATLAB)

Weighted Blending

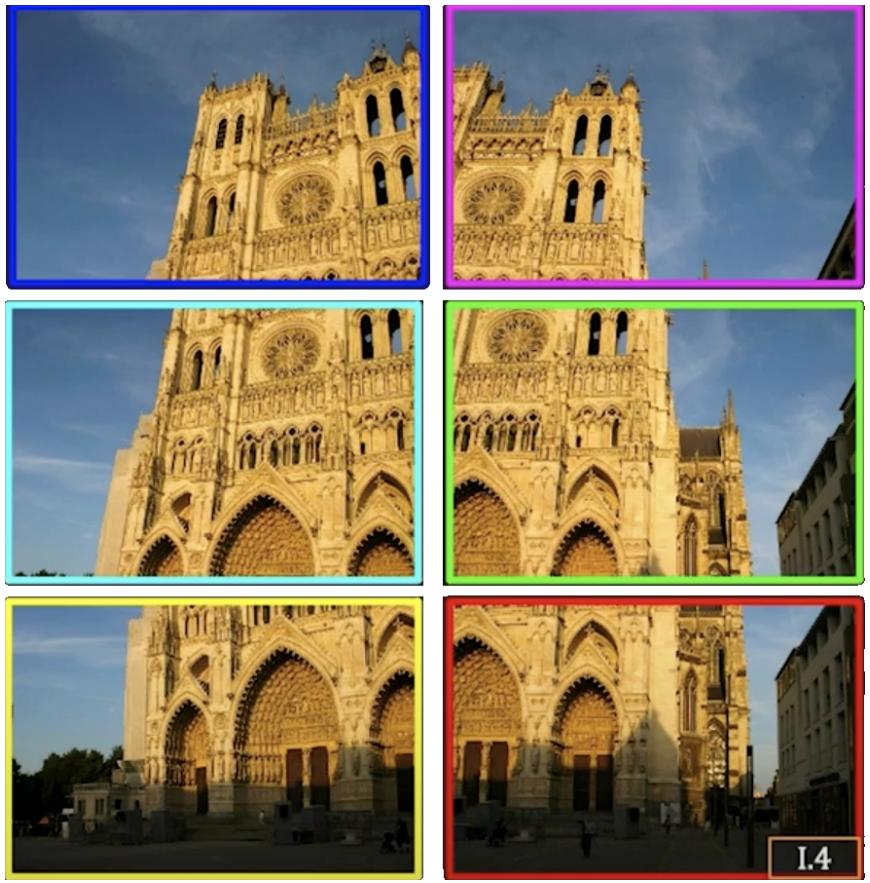


Overlaid Aligned Images

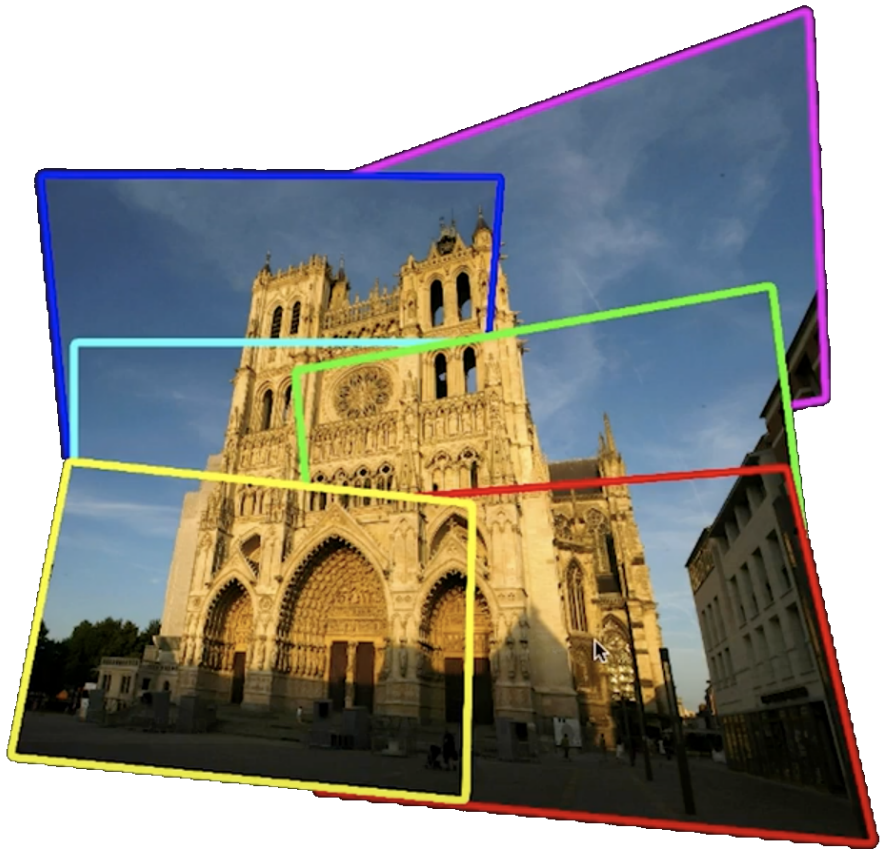


Blended Image

Image Stitching Example

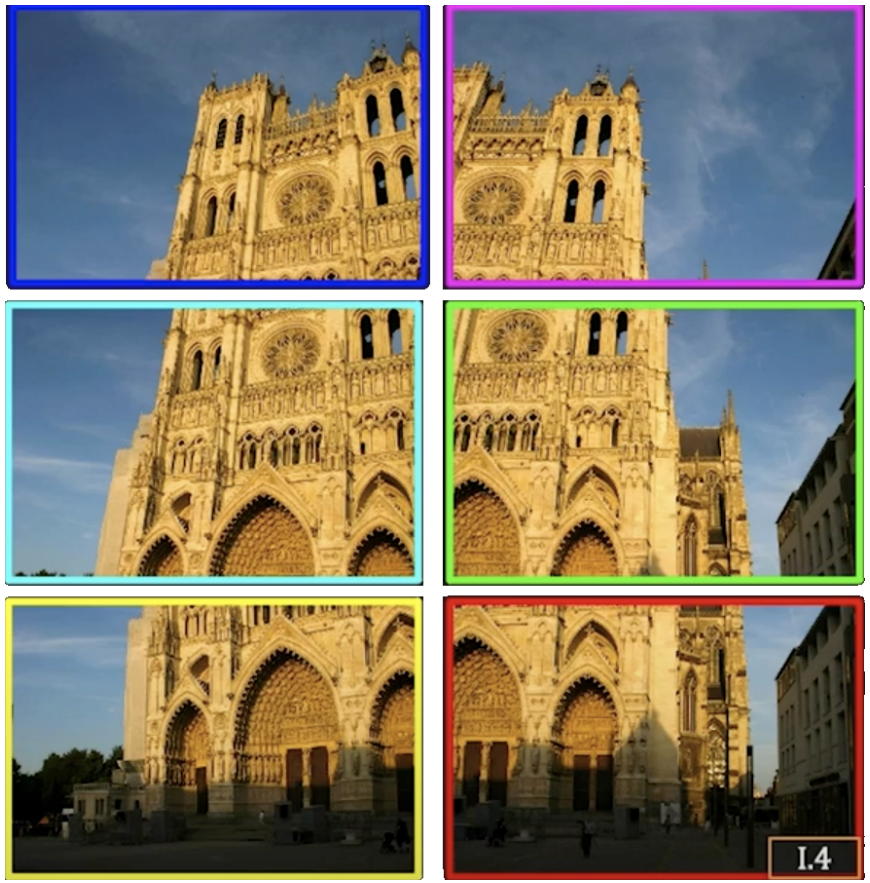


Source Images

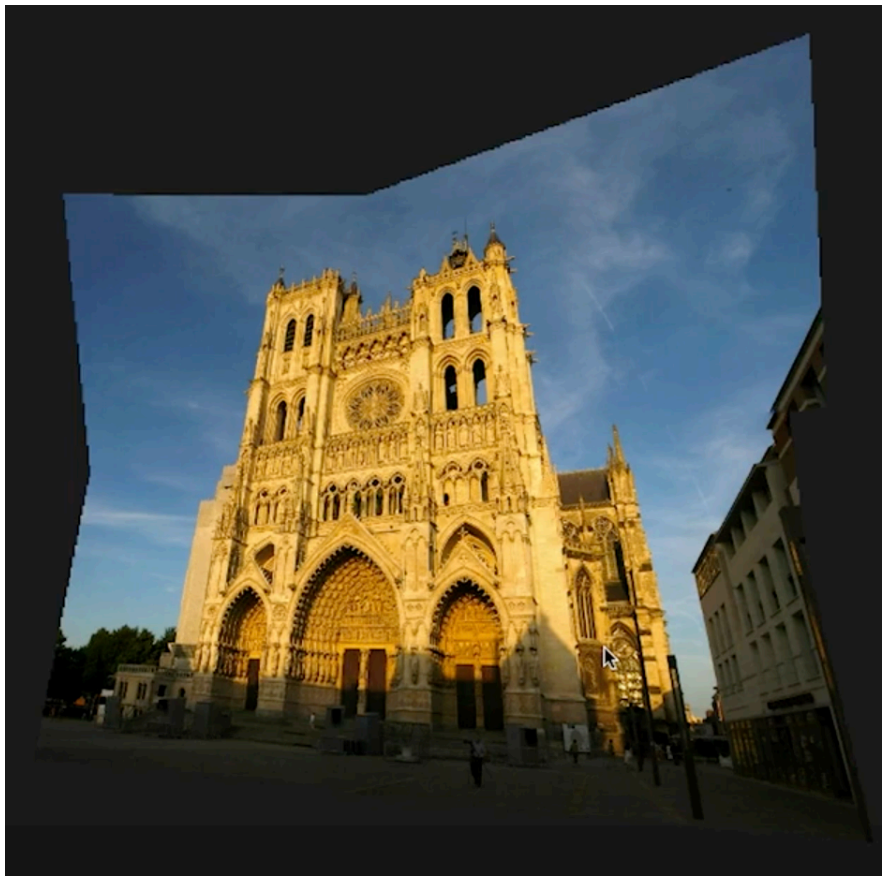


Aligned Images

Image Stitching Example



Source Images



Blended Images