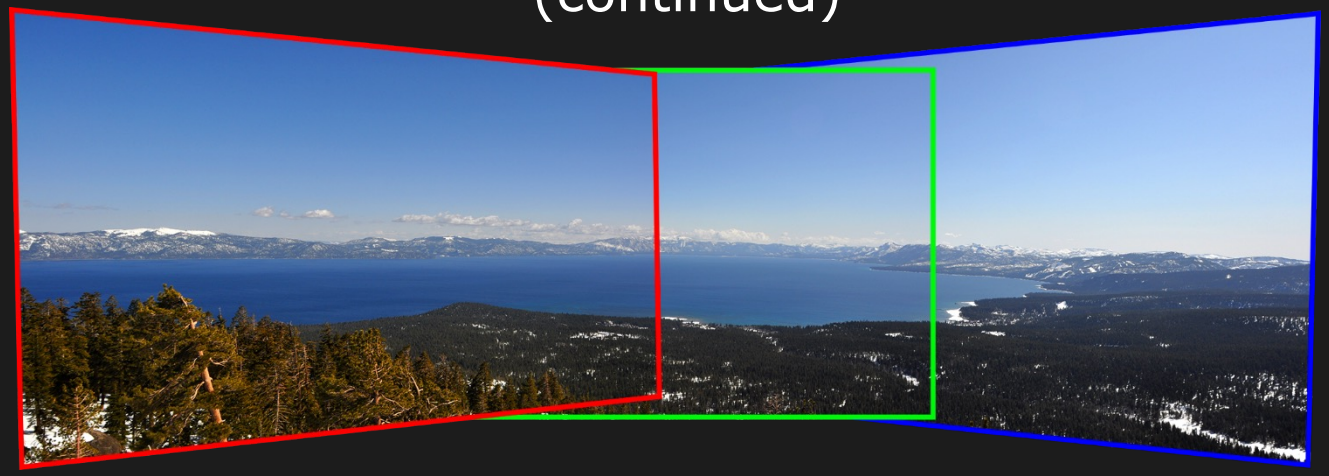
CS/ECE 766 : Computer Vision

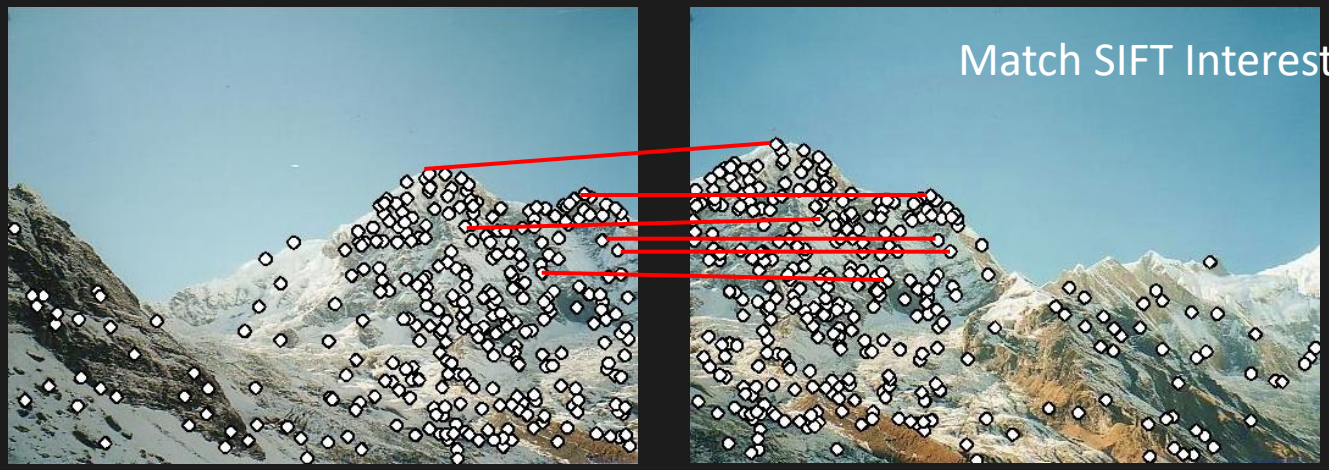
University of Wisconsin-Madison

Image Alignment and Stitching

(continued)

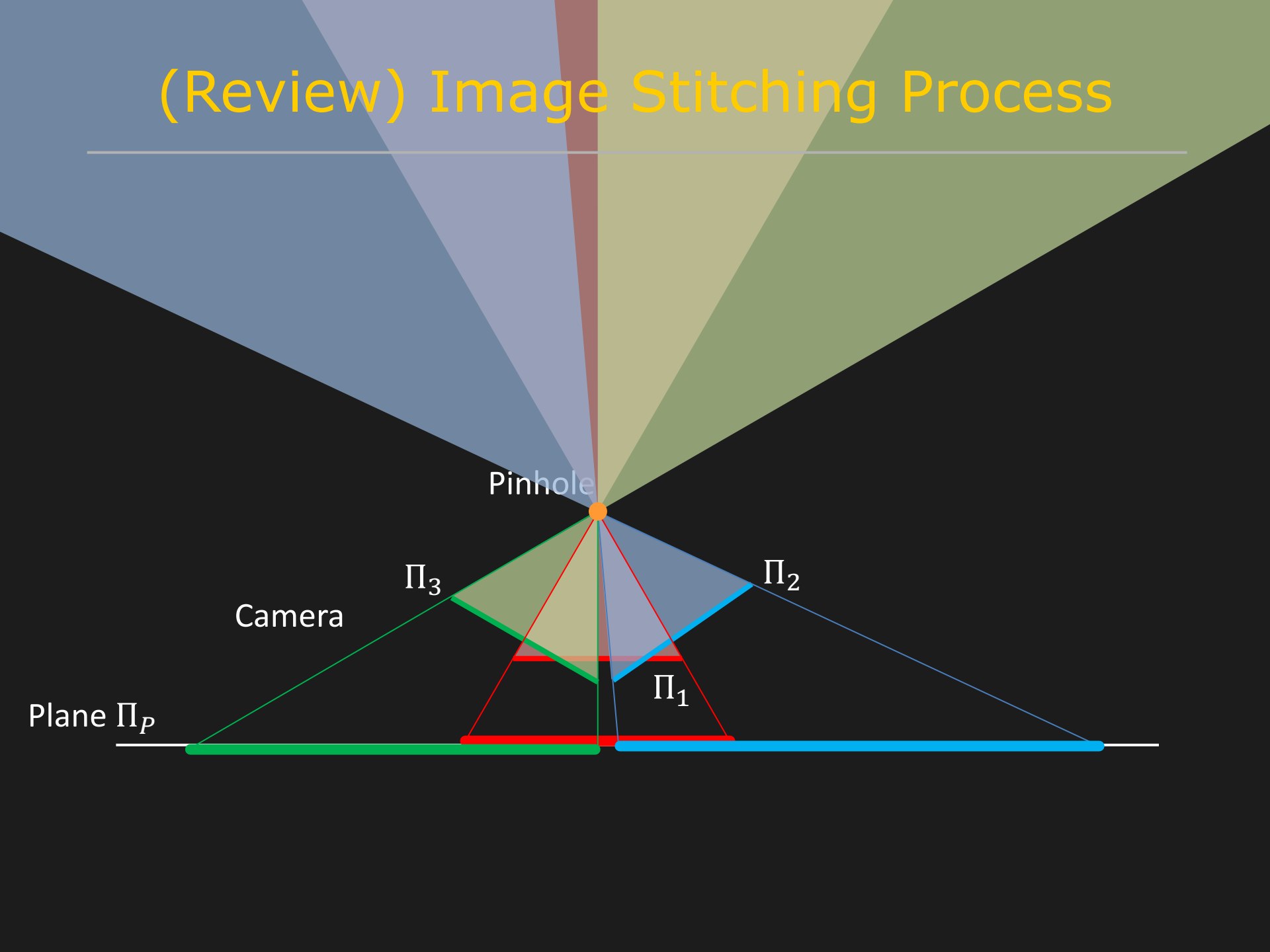


(Review) Panorama Stitching

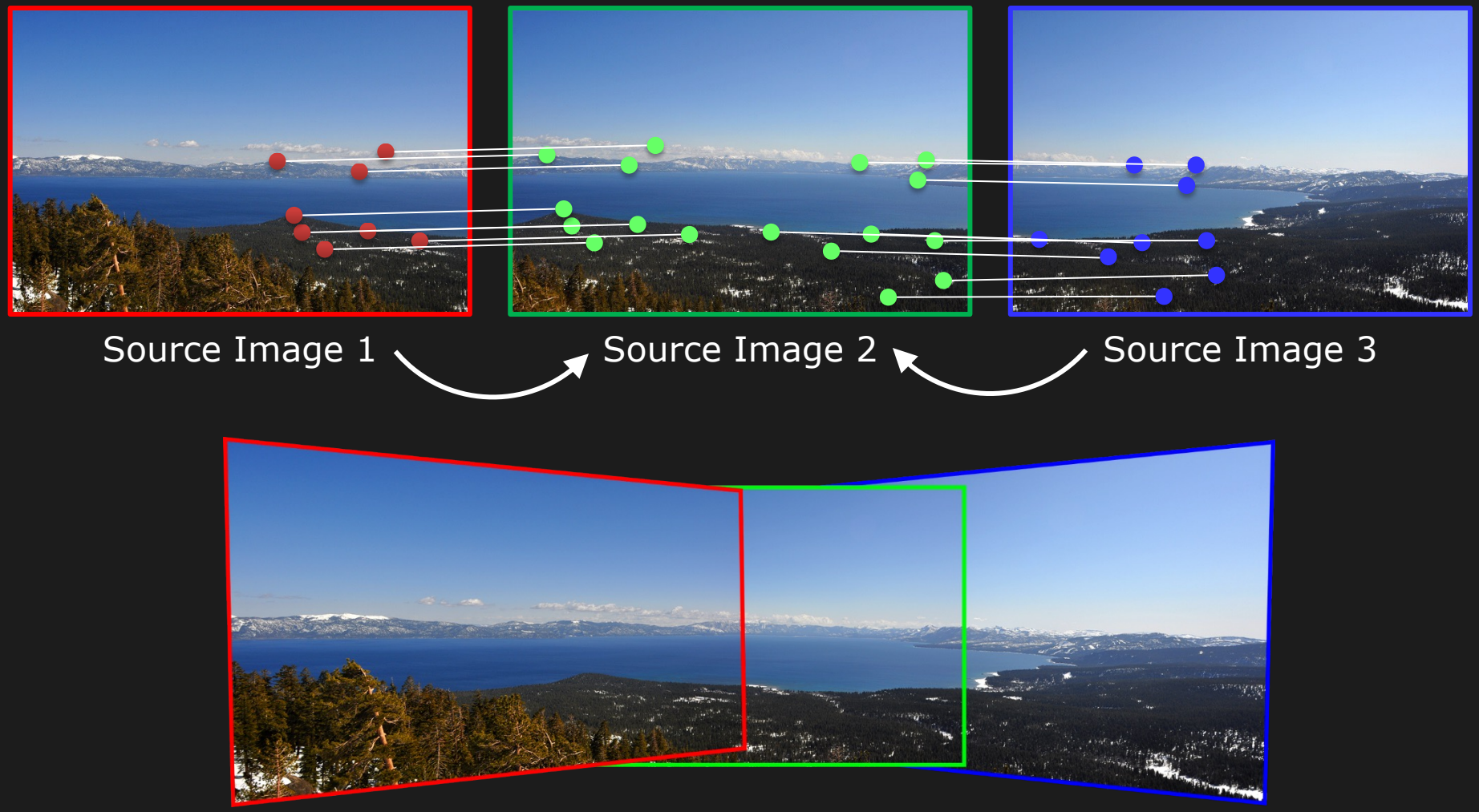


[Autostitch]

(Review) Image Stitching Process



(Review) Image Stitching Process



Warp images so that corresponding points align.

(Review) Global Warping/Transformation

Translation

Rotation

Scaling and Aspect

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

Affine

Perspective

Transformation T is the same over entire domain.

Linear Transformation of Homogeneous Coordinates

(Review) Scaling, Rotation, Skew, Translation

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

Scaling

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

Horizontal Skew

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

Rotation

Translations cannot be represented!

(Review) 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

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

(Review) Homogenous Coordinates

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

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

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

Converting to homogeneous: $(x, y) = \left(\frac{\tilde{x}}{\tilde{z}}, \frac{\tilde{y}}{\tilde{z}}\right)$

Converting from homogeneous: $(\tilde{x}, \tilde{y}, \tilde{z}) = (x, y, 1)$

(Review) 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

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix}1 & m_x & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Horizontal 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

(Review) Projective Transformation

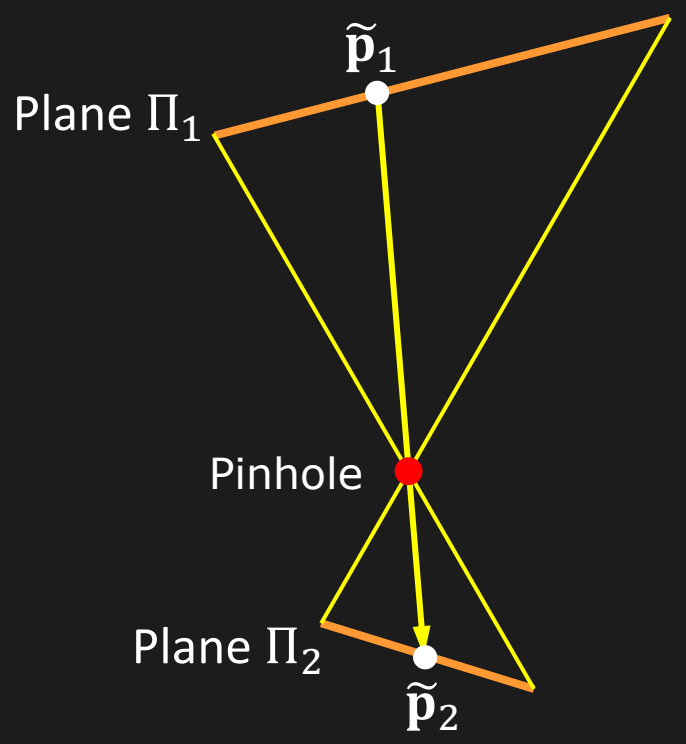
Any transformation of the form:

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

Also called Homography.

(Review) Projective Transformation

Mapping of one plane to another through a pinhole



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

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

Same as imaging a plane through a pinhole.

(Review) Projective Transformation

Homography can only be defined up to a scale.

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \equiv \underbrace{\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix}}_{\text{Because homogeneous coordinates are only defined up to a scale.}}$$

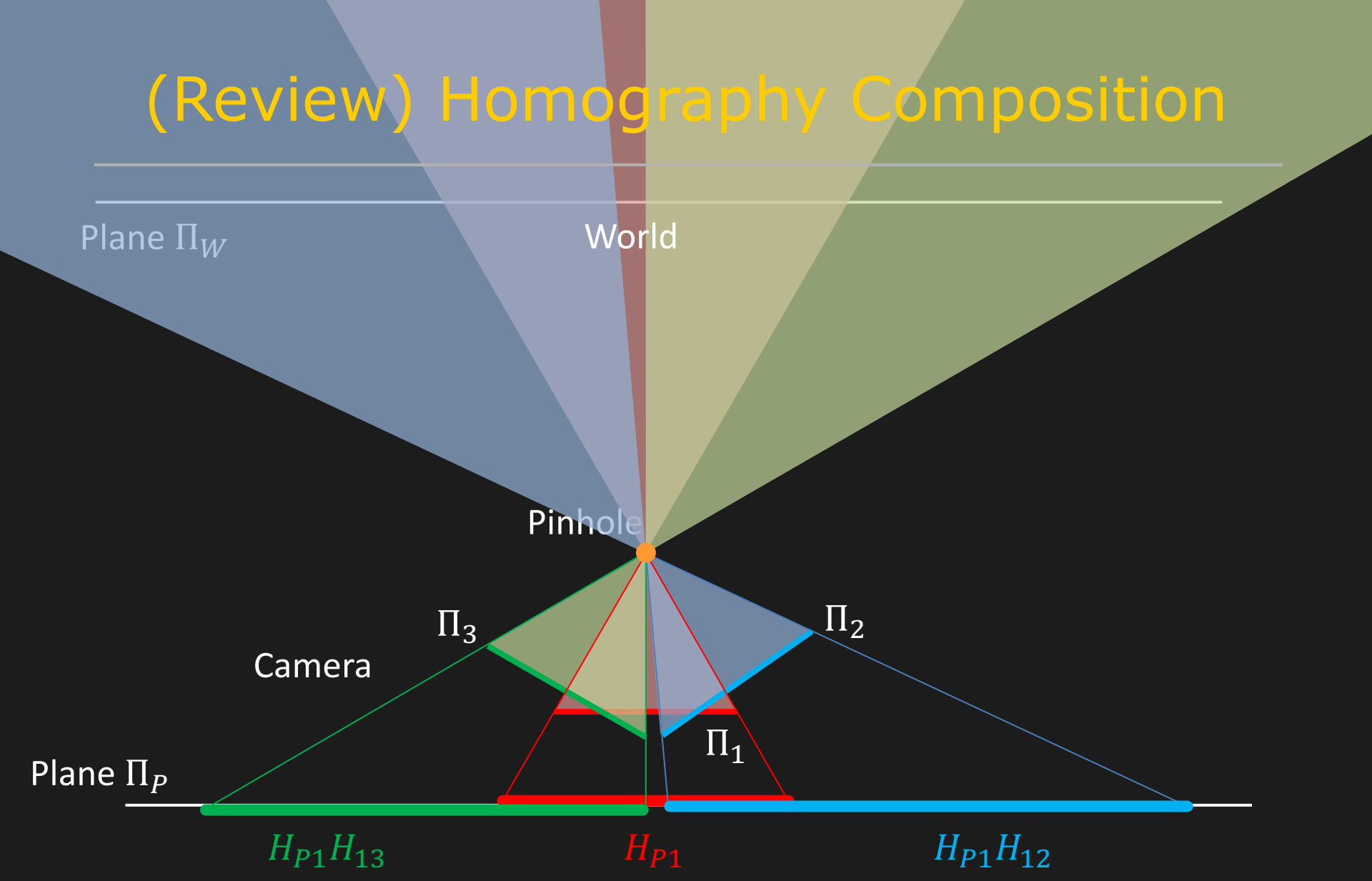
If we fix scale such that $\sqrt{\Sigma(h_{ij})^2} = 1$ then 8 free parameters

(Review) Projective Transformation

Transformation that represents mapping a plane into another through a pinhole.

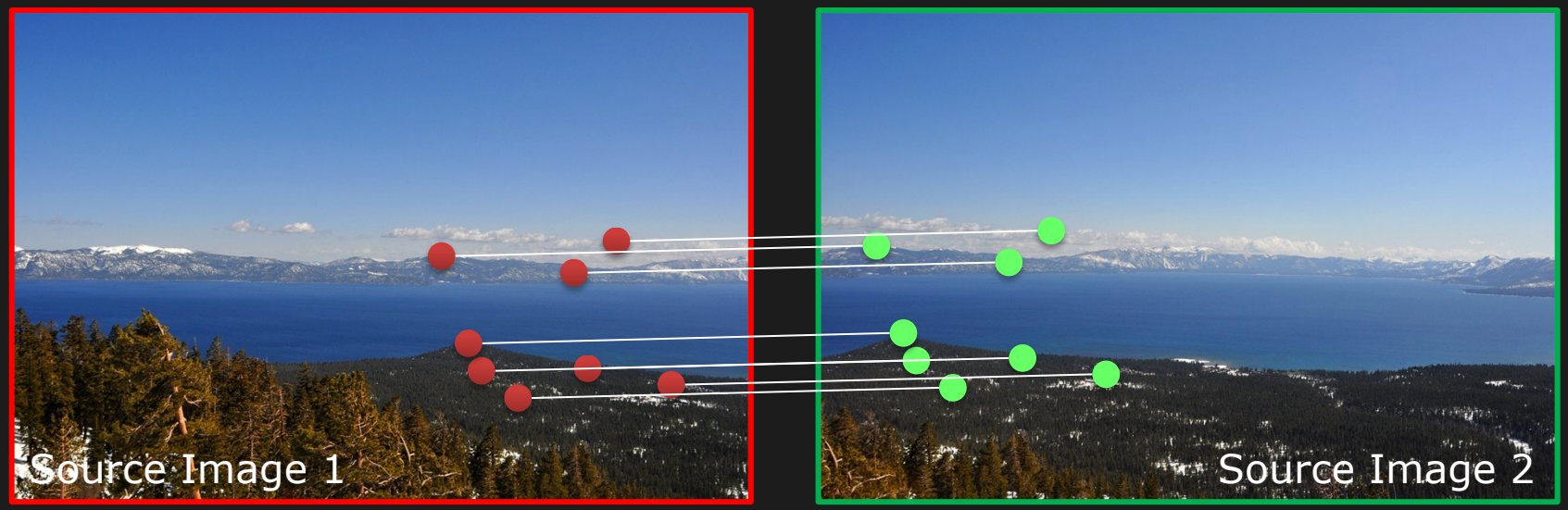
- Origin does not necessarily map to the origin
- Lines map to lines
- Parallel lines do not necessarily remain parallel
- Closed under composition

(Review) Homography Composition



Useful in stitching planar panoramas.

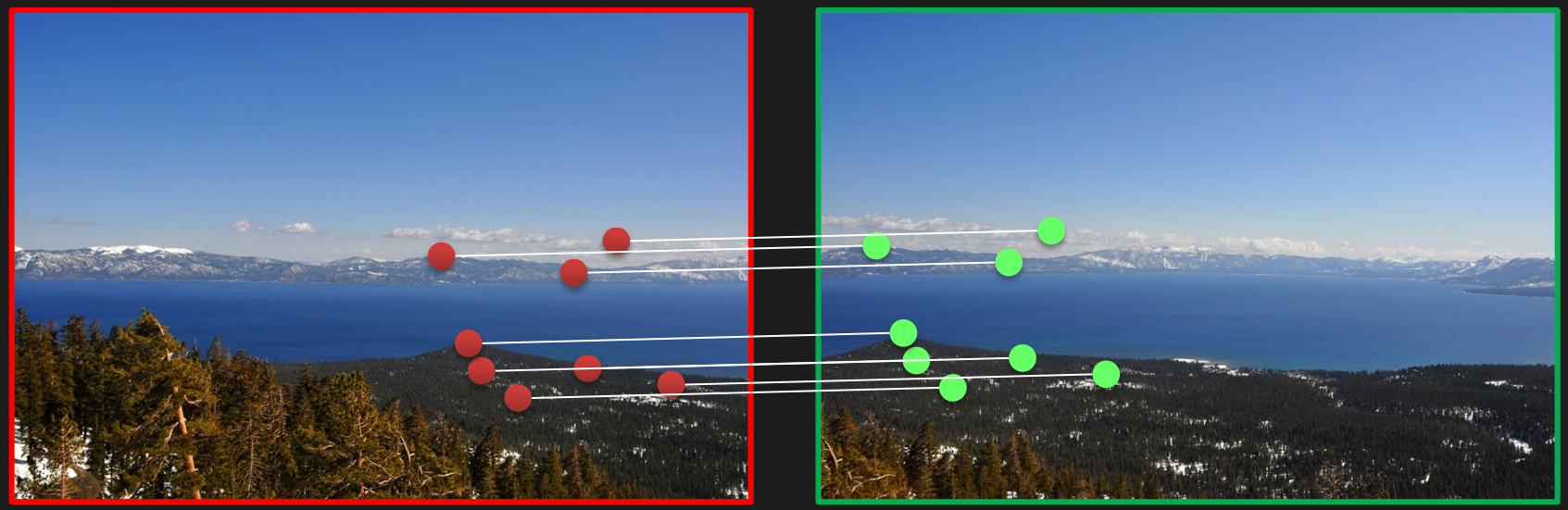
(Review) Computing Homography



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

The scene points should lie on a plane, or be distant (plane at infinity), or imaged from the same point.

(Review) Computing Homography



Source Image

Destination Image

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

How many pairs of points are needed to define the homography?

There are 9 unknowns, but only 8 degrees of freedom.

Each pair provides 2 constraints. So, 4 pairs are needed.

(Review) Computing Homography

For a given pair i of corresponding points (2 constraints):

$$x_d{}^{(i)} = \frac{\tilde{x}_d{}^{(i)}}{\tilde{z}_d{}^{(i)}} = \frac{h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}} \qquad y_d{}^{(i)} = \frac{\tilde{y}_d{}^{(i)}}{\tilde{z}_d{}^{(i)}} = \frac{h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}}$$

Rearranging the terms and writing as linear equation:

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

A (Known) $\qquad$ $\mathbf{h}$ (Unknown)

(Review) Constrained Least Squares

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

Define least squares problem:

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

Solve (unconstrained) Lagrangian function $L(\mathbf{h}, \lambda)$:

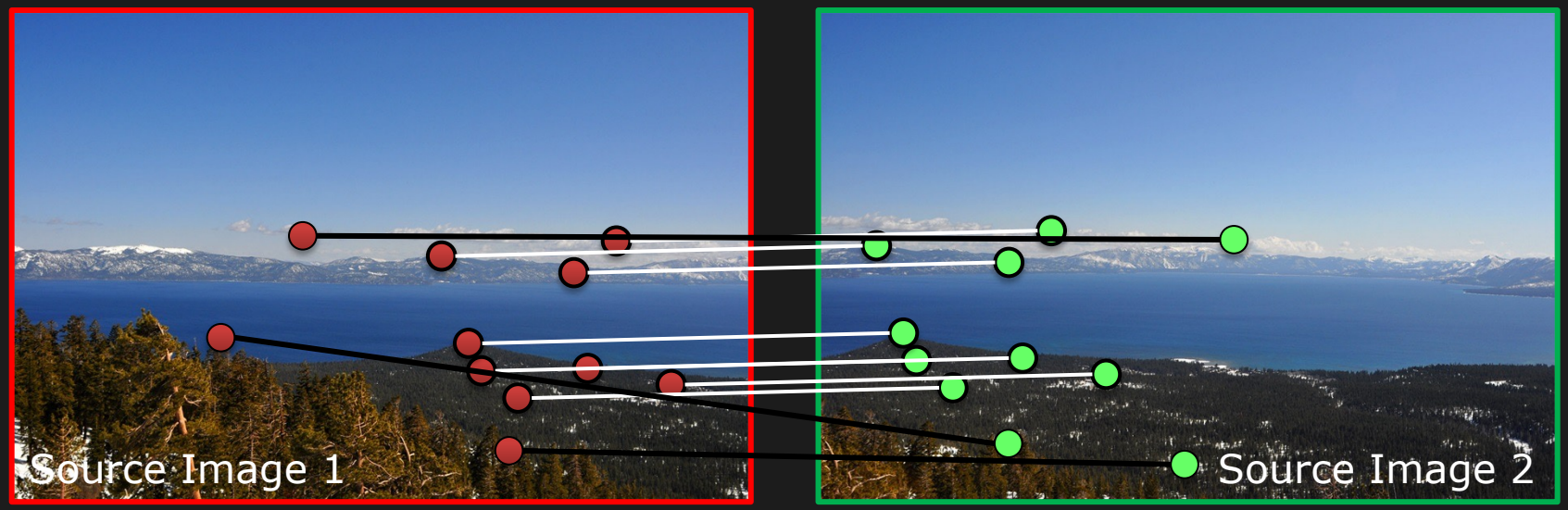
$$L(\mathbf{h}, \lambda) = \mathbf{h}^T A^T A\mathbf{h} - \lambda(\mathbf{h}^T\mathbf{h} - 1)$$

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

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

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

What Could Go Wrong?



Outliers!

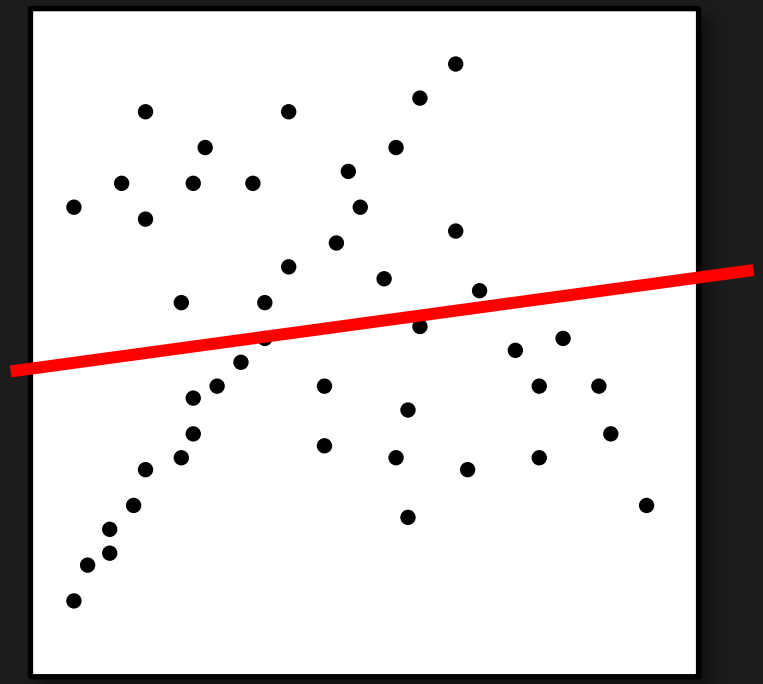
We need to robustly compute transformation in the presence of wrong matches.

How?

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

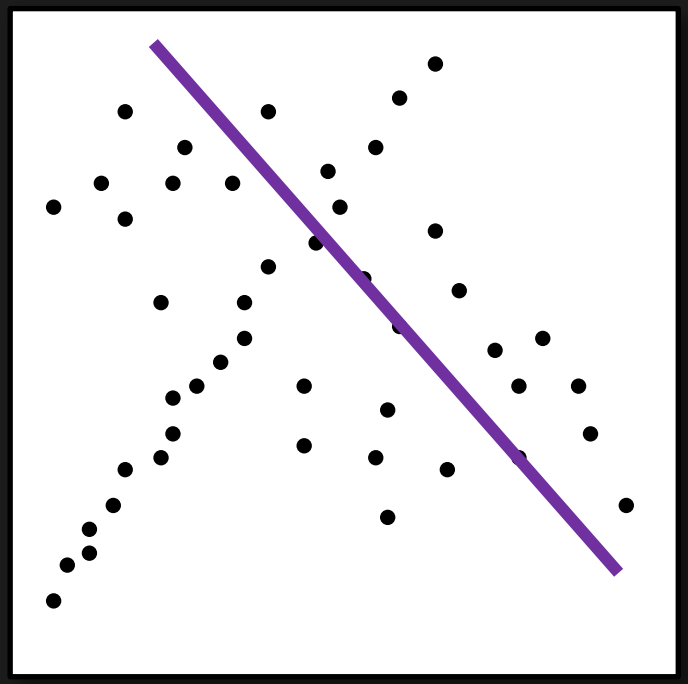
RANSAC Example: Line Fitting

Robust line fitting:



Least Squares Fitting

Inliers: 2

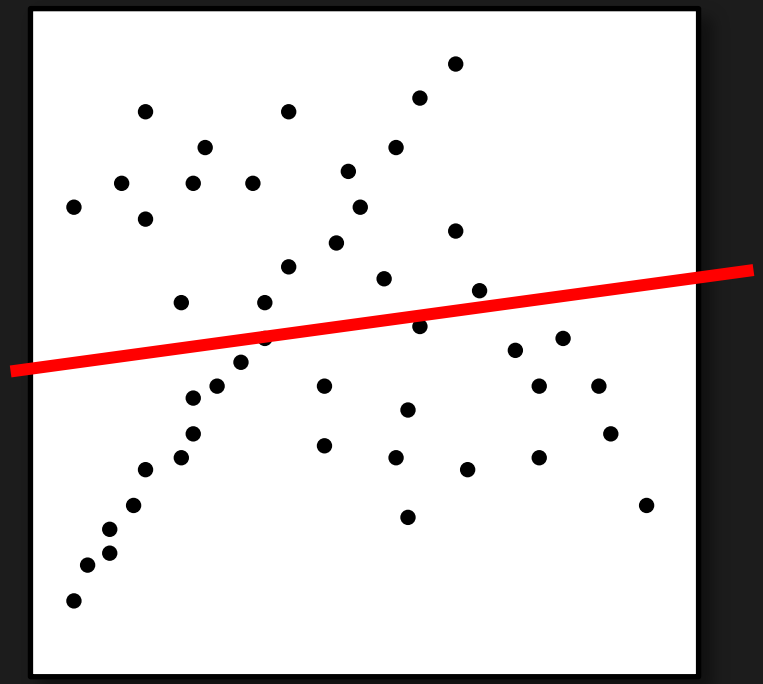


RANSAC Iteration 1

Inliers: 4

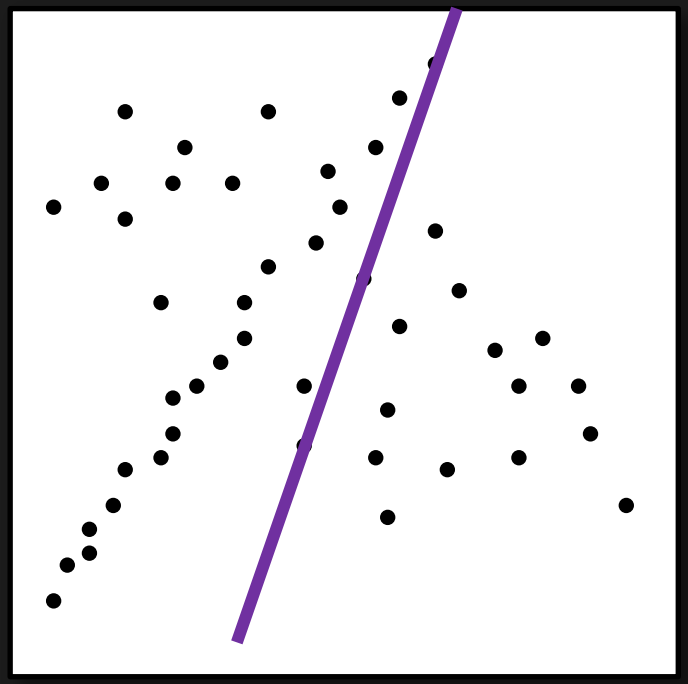
RANSAC Example: Line Fitting

Robust line fitting:



Least Squares Fitting

Inliers: 2

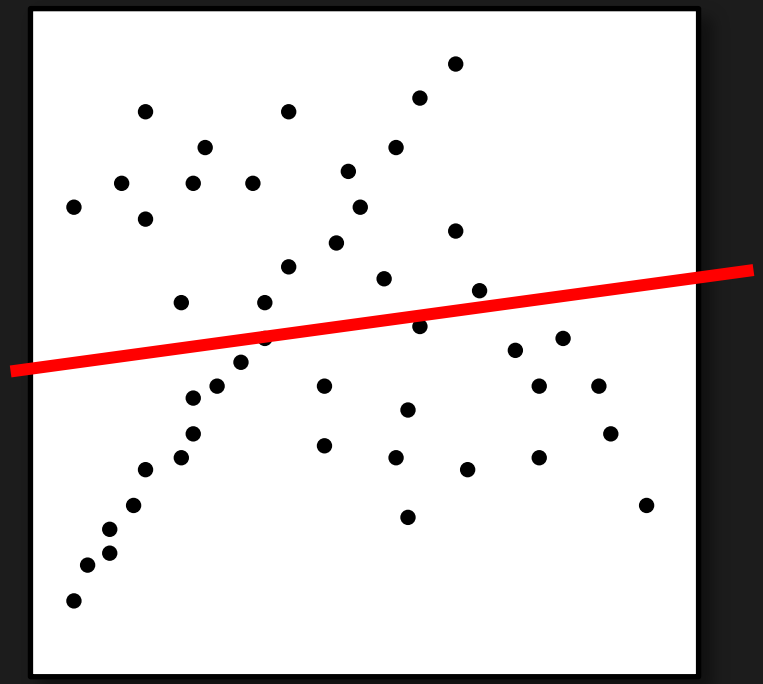


RANSAC Iteration 2

Inliers: 3

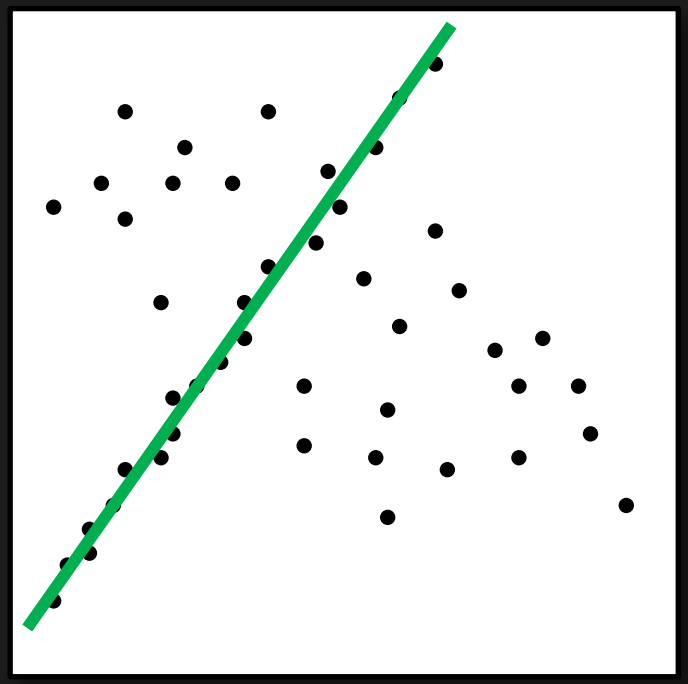
RANSAC Example: Line Fitting

Robust line fitting:



Least Squares Fitting

Inliers: 2



RANSAC Iteration i

Inliers: 20

RANdom SAmple Consensus

General RANSAC Algorithm:

1. Randomly choose s samples. Typically s is the minimum samples to fit a 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:

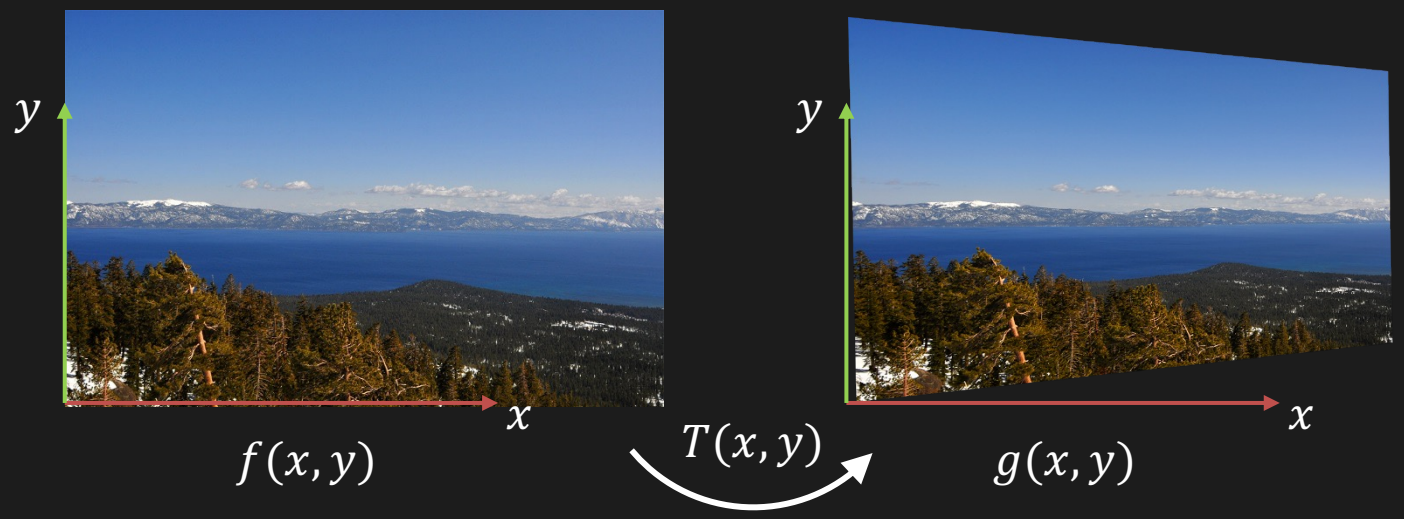
$s = 4$ points. ε is acceptable alignment error in pixels.

[Fischler 1981]

Back to Warping Images

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

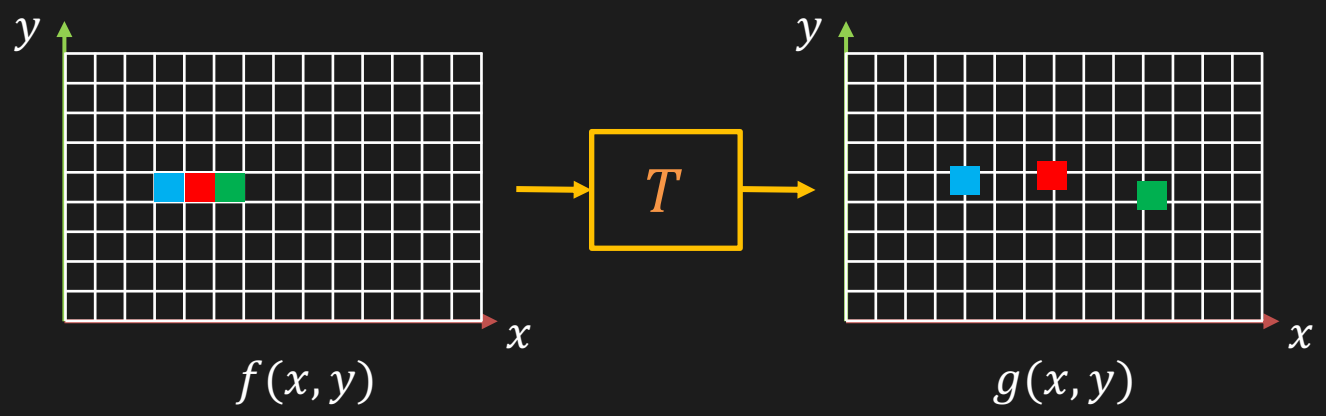
$$\boxed{g(x, y) = f\big(T(x, y)\big)}$$



Forward Warping

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

$$\boxed{g(x, y) = f\big(T(x, y)\big)}$$



What if pixel lands in between pixels?

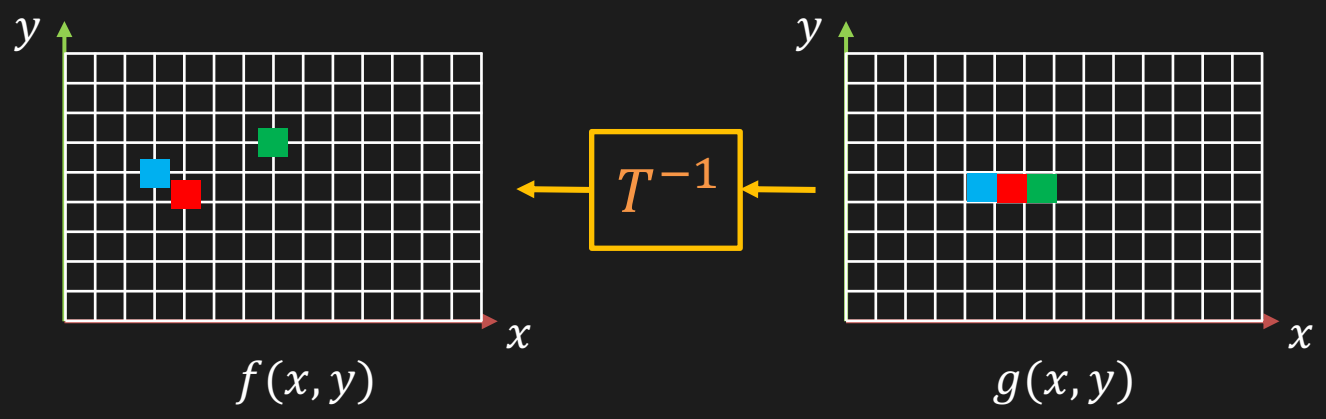
What if not all pixels in $g(x, y)$ are filled?

Can result in holes!

Backward Warping

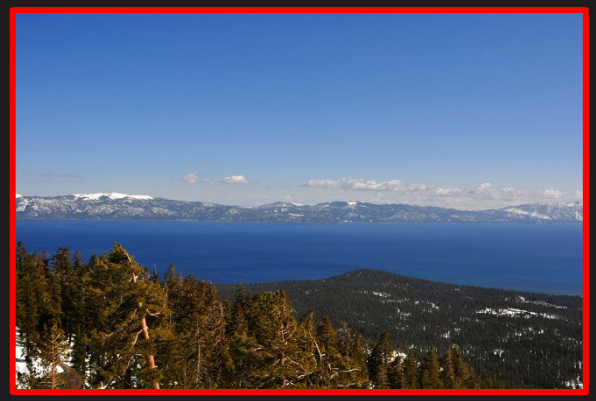
Get each pixel (x, y) in $g(x, y)$ from its corresponding location $T^{-1}(x, y)$ in $f(x, y)$

$$\boxed{g(x, y) = f\big(T(x, y)\big)}$$

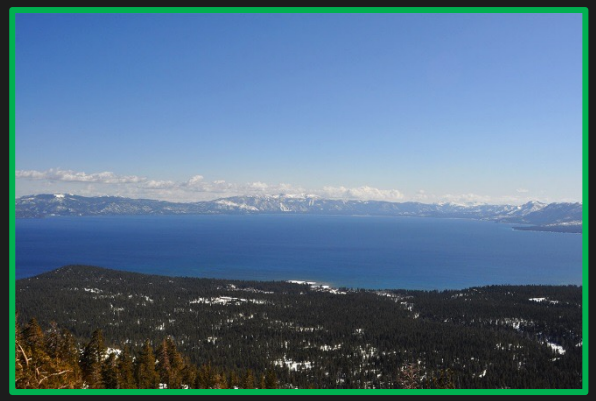


What if pixel lands between pixels?
No problem. Use Nearest Neighbor or Interpolate (Weighted Average of Neighbors)!

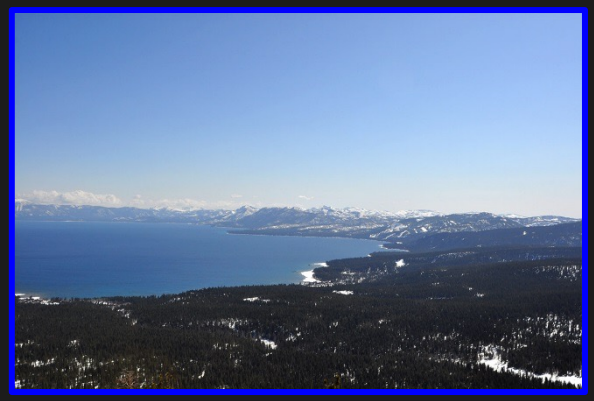
Image Alignment Process



Source Image 1



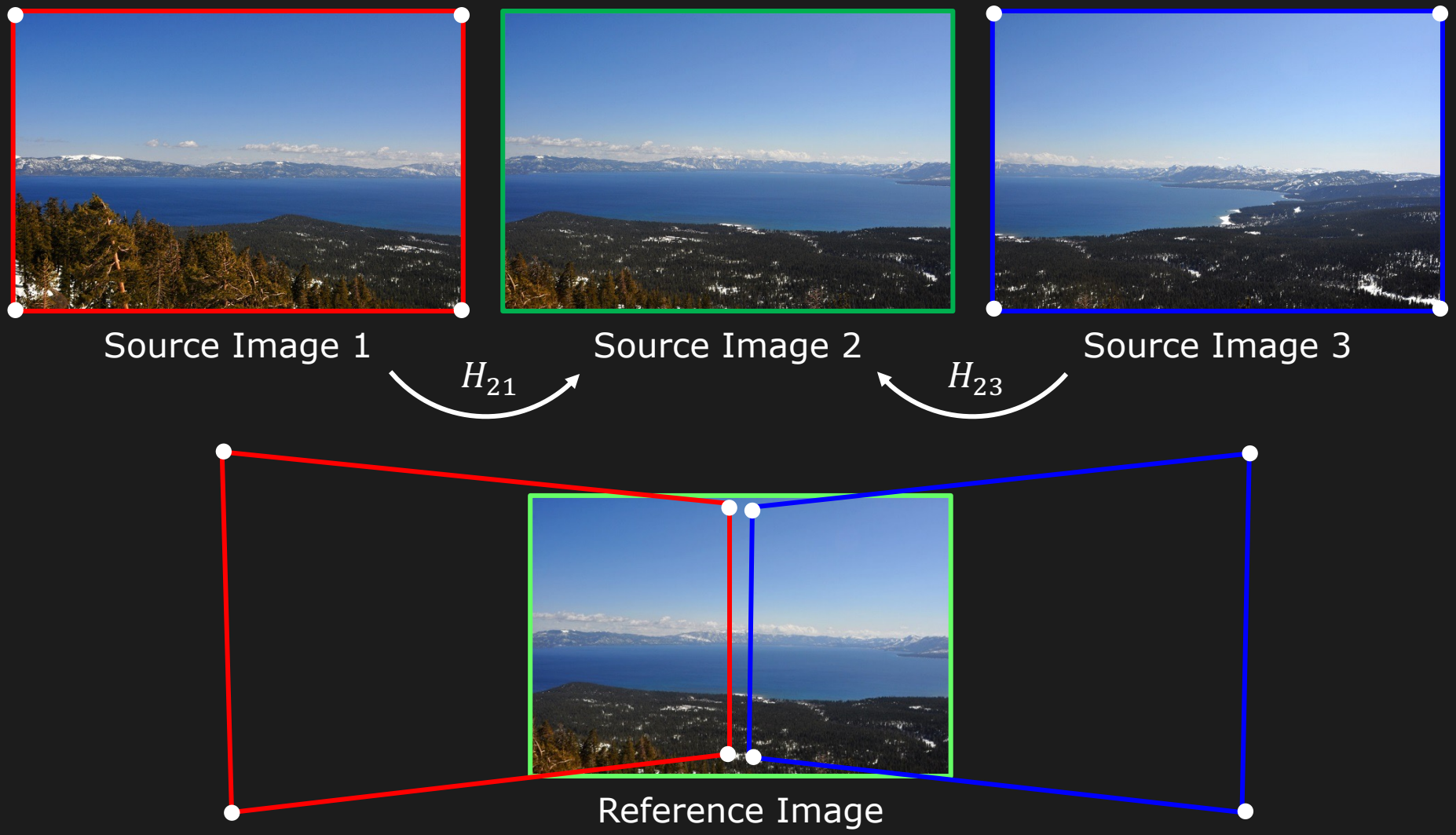
Source Image 2



Source Image 3

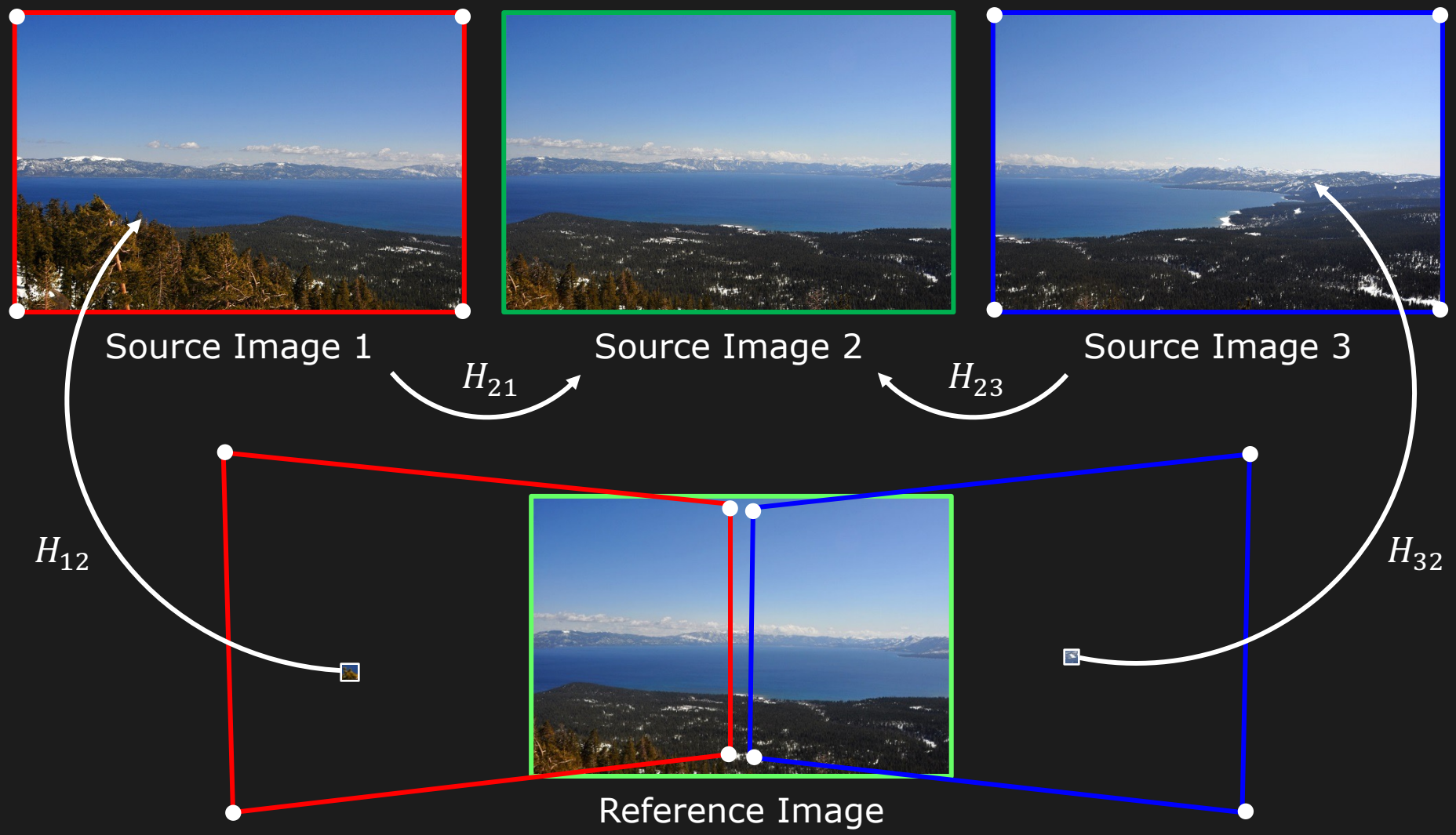
Reference Image
(Source Image 2)

Image Alignment Process



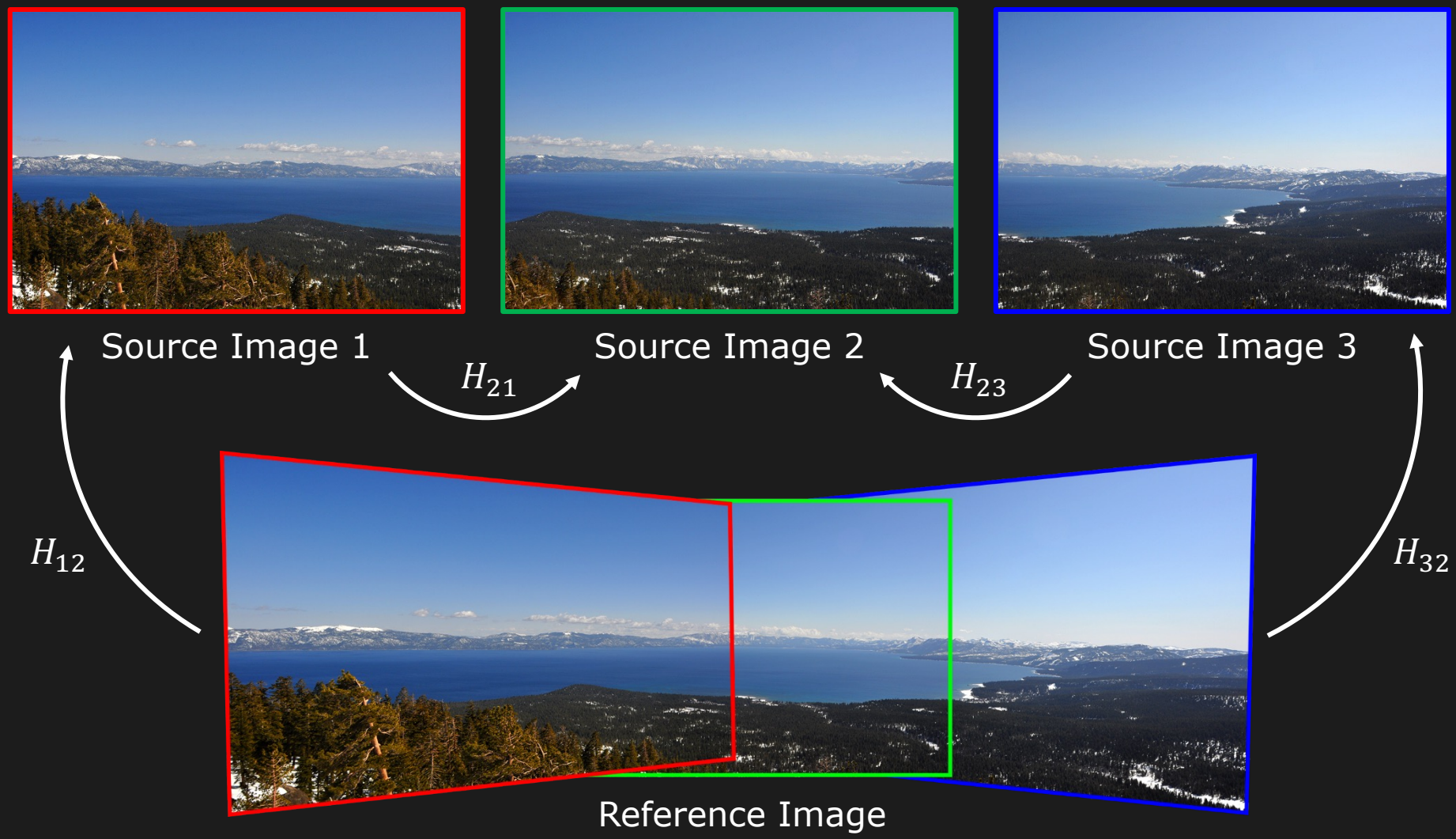
Compute the bounds of Image 1 and Image 3 in reference image space

Image Alignment Process



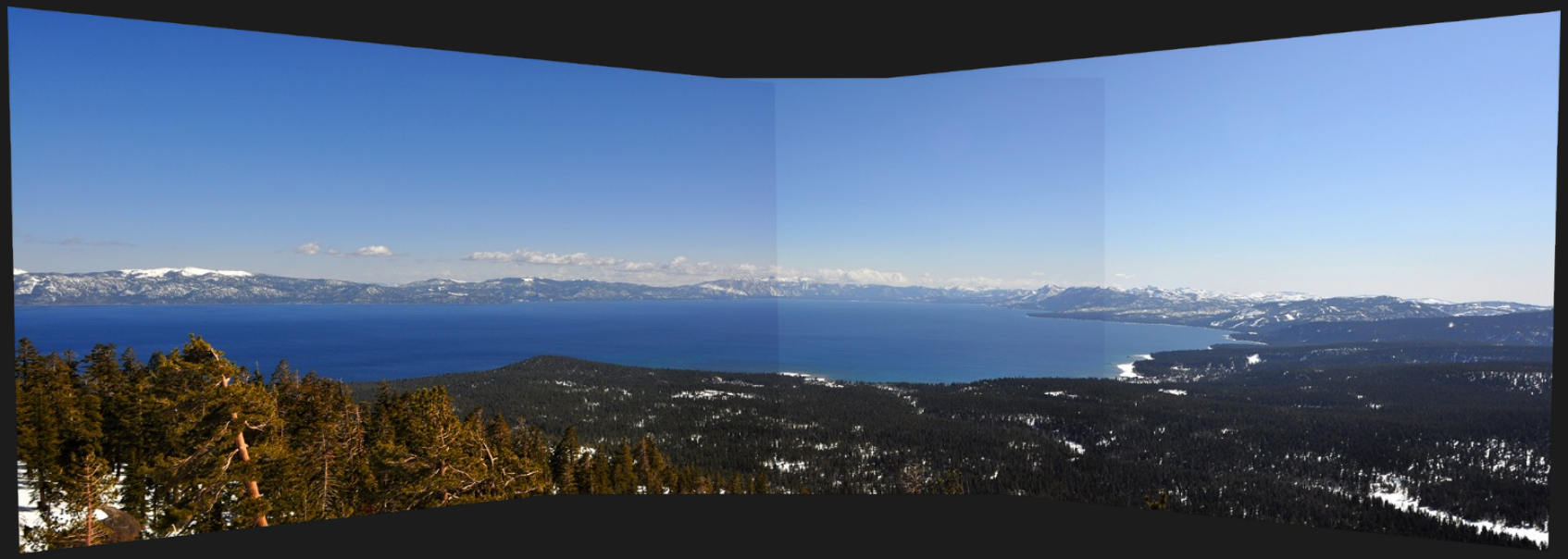
For each pixel within bounds, compute its location in source image

Image Alignment Process



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

Blending Images



Overlaid Aligned Images

Hard seams due to vignetting, exposure differences, etc.

Blending Images: Averaging

Averaged Images

Seams still visible.

Blending Images

Image I_1

+

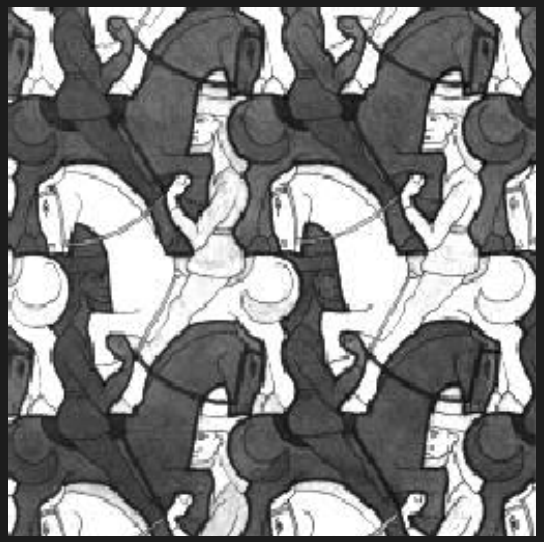
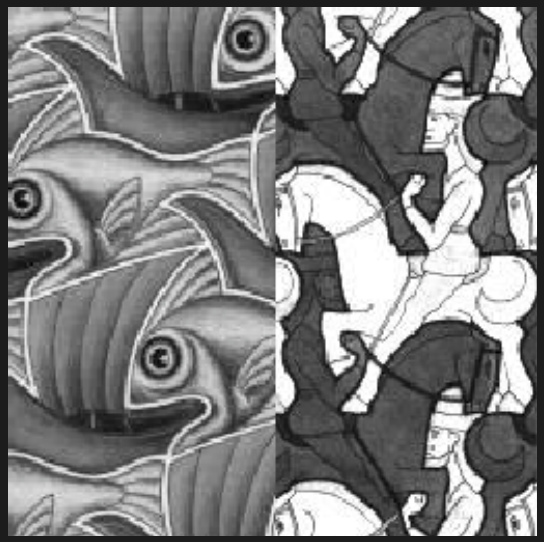


Image I_2

=



Hard overlay

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

Weighted Blending

Image I_1

+

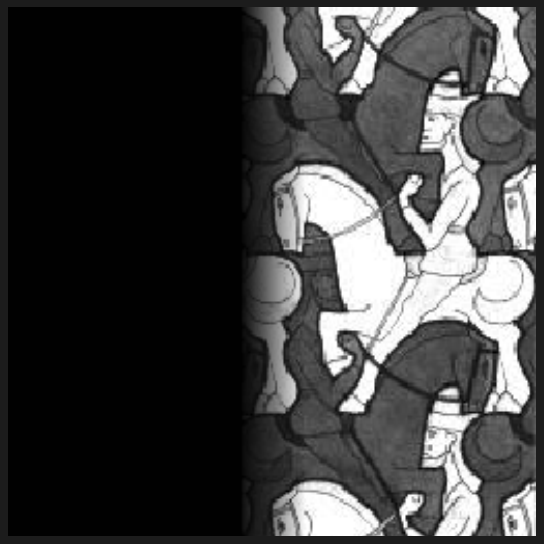


Image I_2

=

Blended Image I_{blend}

Weight w_1

Weight w_2

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

Weighted Blending

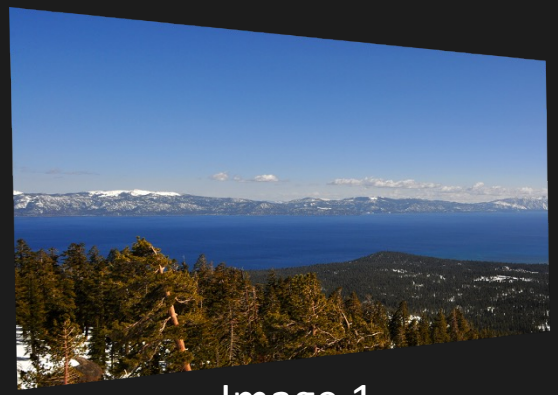


Image 1

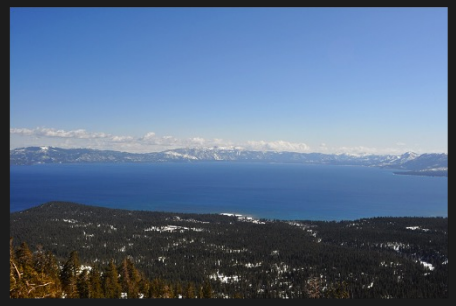


Image 2

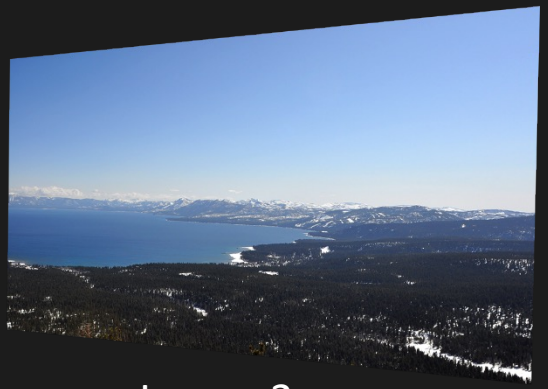
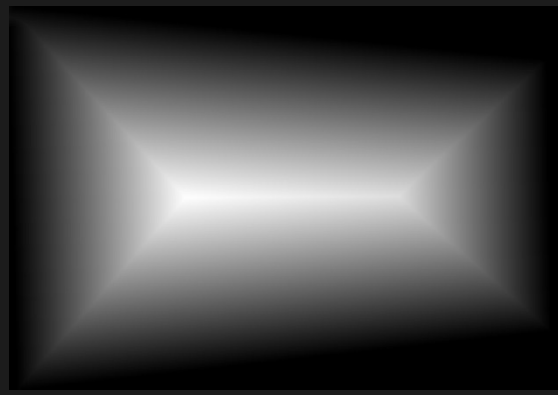
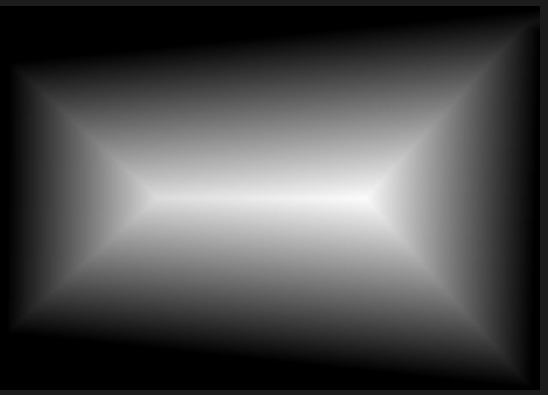


Image 3



Weight w_1

Weight w_2

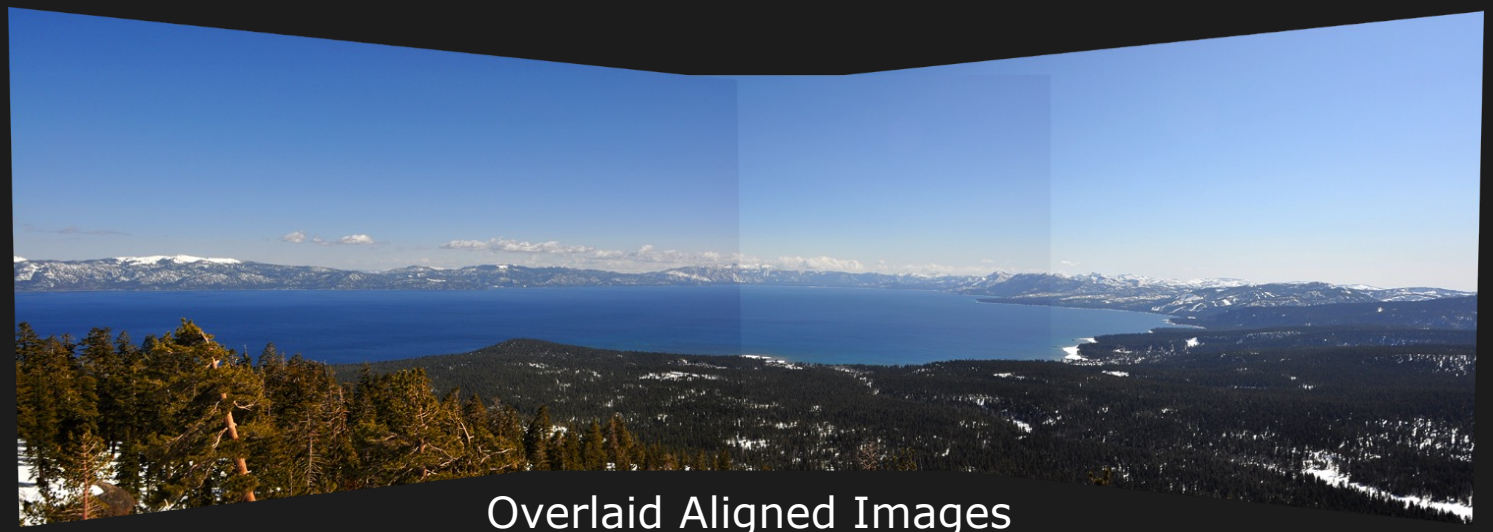


Weight w_3

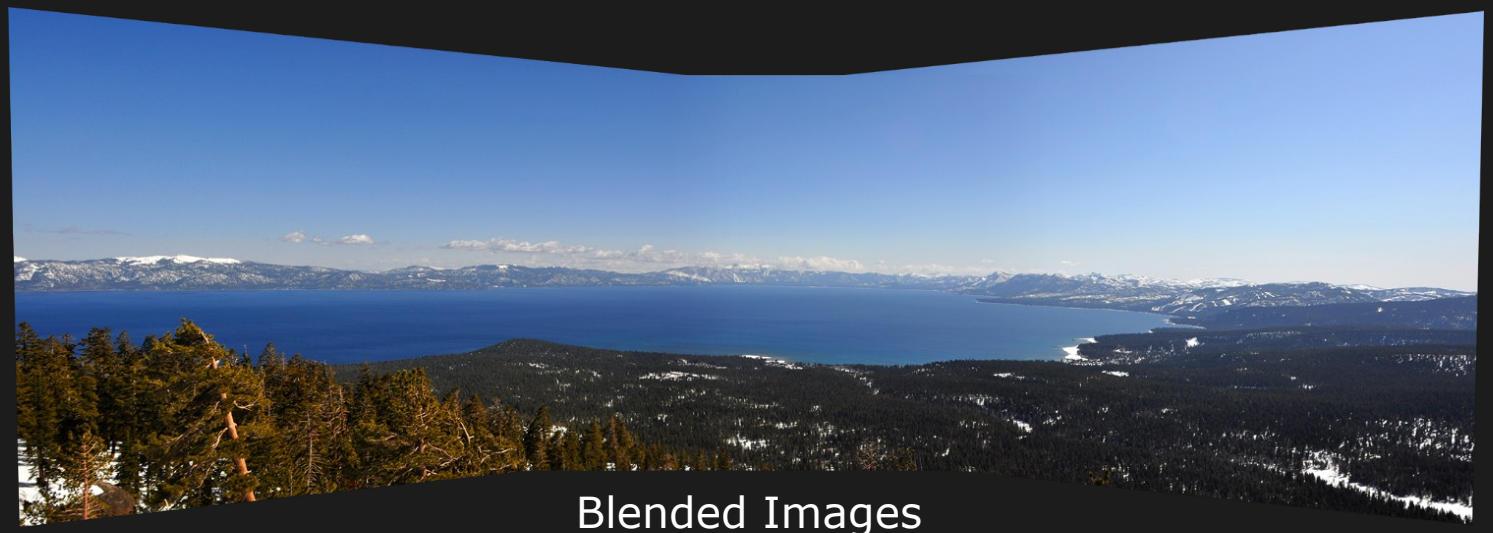
Pixels closer to the edge get a lower weight.

Python: `scipy.ndimage.distance_transform_edt`

Weighted Blending

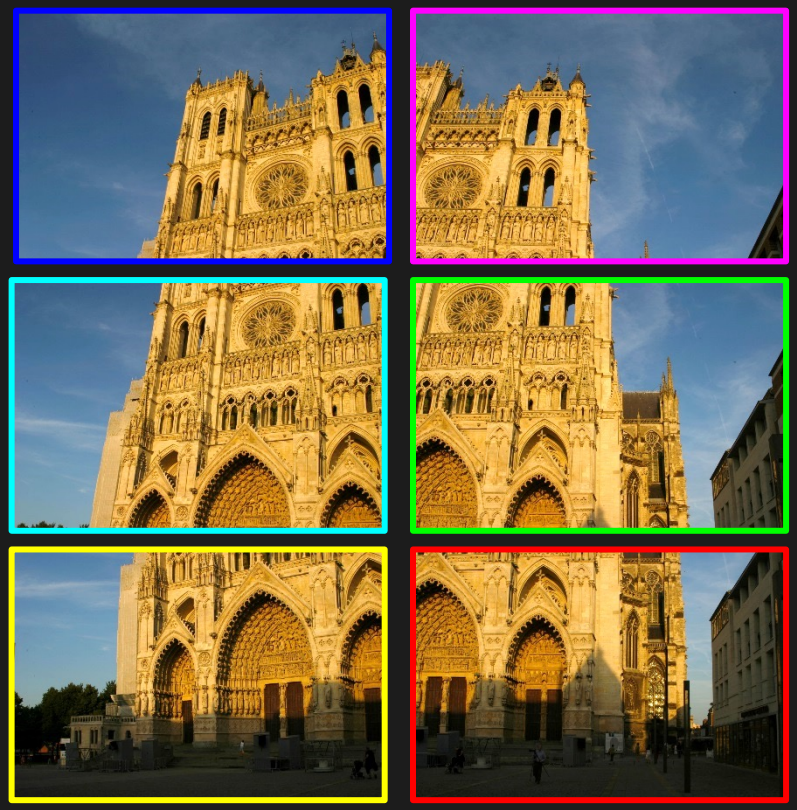


Overlaid Aligned Images

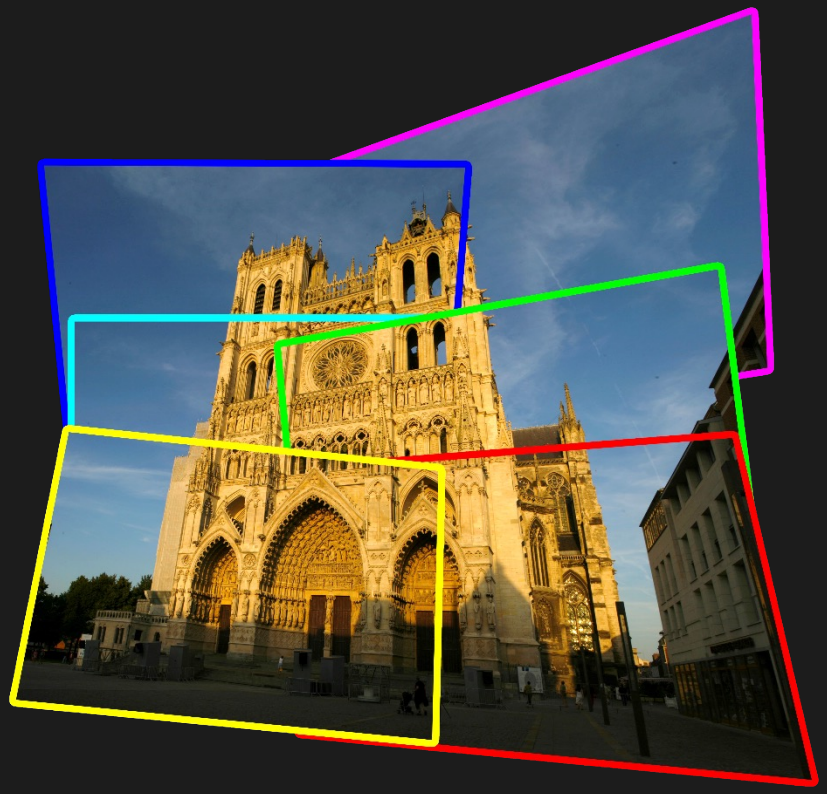


Blended Images

Image Stitching Example

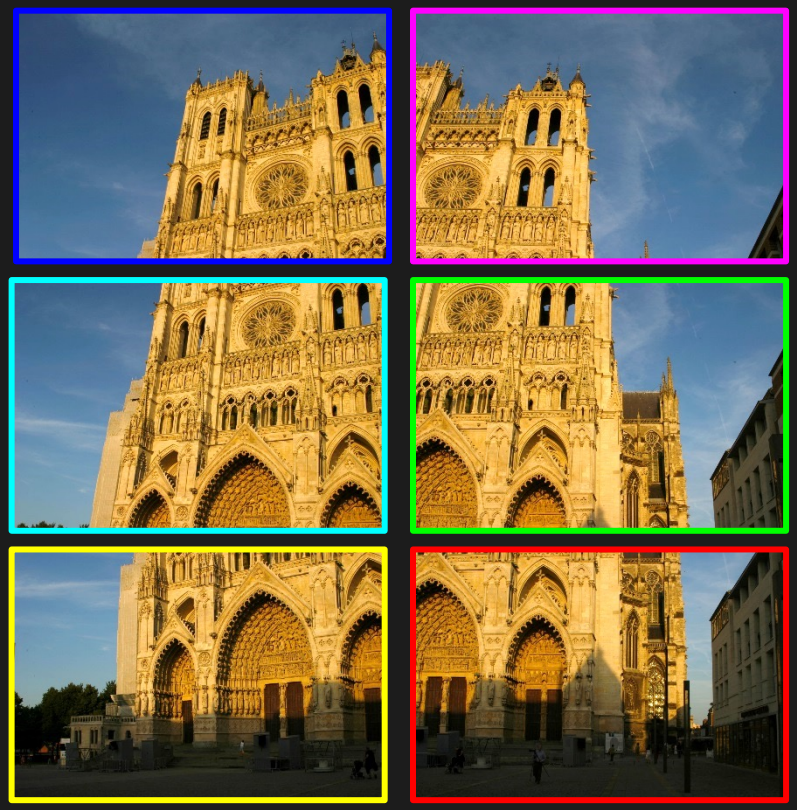


Source Images



Aligned Images

Image Stitching Example



Source Images

Blended Images

References: Textbooks

Computer Vision: Algorithms and Applications (Chapter 2, 9)
Szeliski, R., Springer

References: Papers

[Fischler 1981] Fischler M. A. and Bolles R. C. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, 1981.

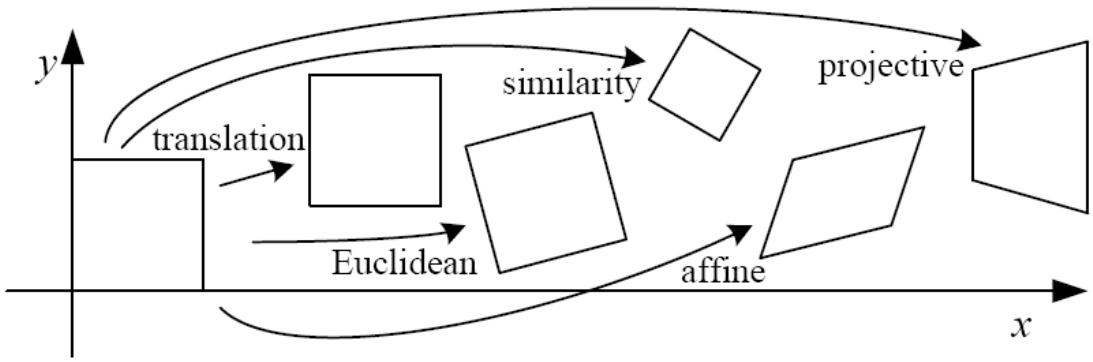
Image Credits

I.1 http://www.flickr.com/photos/byspice/4577634277

I.2 http://www.ptgui.com/examples/quicktour5/

I.3 Figure 2.4, Table 2.1, Computer Vision: Algorithms and Applications, Szeliski, R., Springer

Appendix A: Linear Transformations



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles $+\cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Appendix B: RANSAC

How many iterations?

Suppose,

- p: probability or confidence with which we need the correct fit.
- α: probability of choosing an inlier while selecting the s samples.

If the s points are selected independently. Then,

Minimum number of iterations required:

$$\boxed{N = \frac{\log(1-p)}{\log(1-\alpha^s)}}$$

[Fischler 1981]

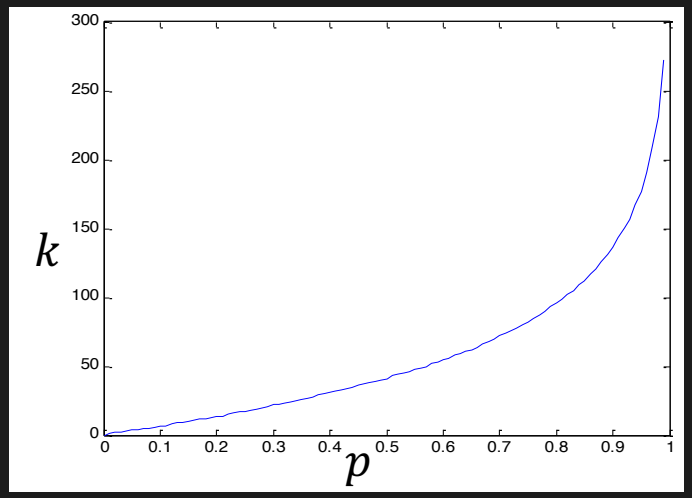
Appendix B: RANSAC

Minimum number of iterations required:

$$\boxed{k = \frac{\log(1-p)}{\log(1-\alpha^s)}}$$

	Proportion of inliers α						
s	95%	90%	80%	75%	70%	60%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Minimum number of iterations ($p = 0.99$)



$s = 8, \alpha = 60\%$

[Fischler 1981]