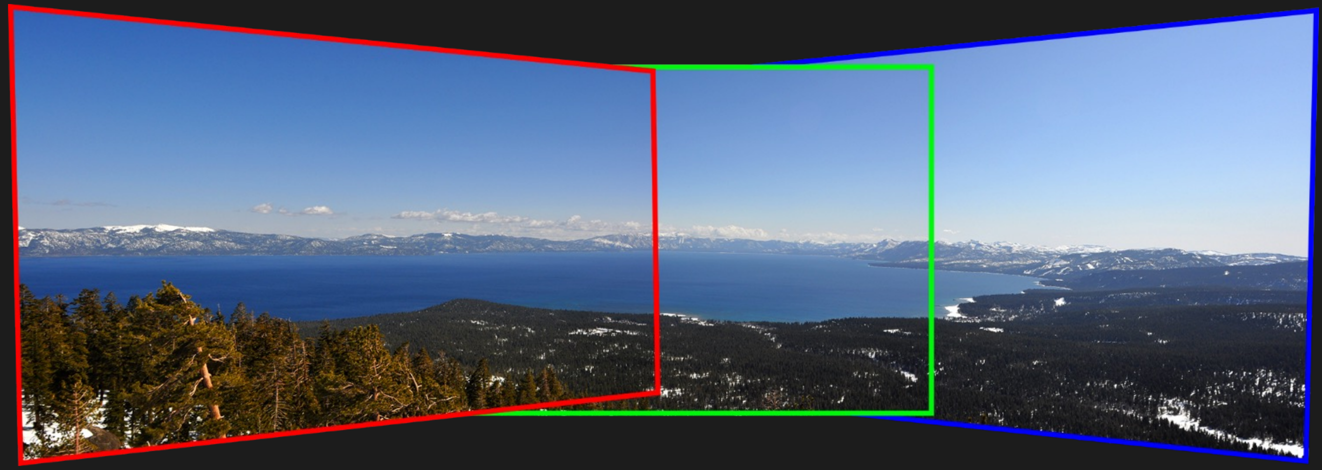
CS/ECE 766 : Computer Vision

University of Wisconsin-Madison

Image Alignment and Stitching



Review: Keypoint Matching w/ SIFT

Two key components:

1. Find location and scale invariant Interest Points.

- Blobs

2. Describe each interest point through a vector representation that is invariant to:

- Scale
- Orientation
- Brightness

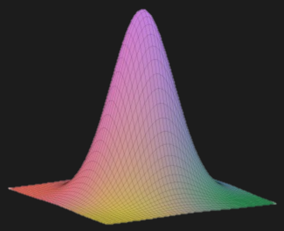
Review: 2D Blob Detector

Normalized Laplacian of Gaussian (NLoG) is used as the 2D equivalent for Blob Detection.

Laplacian

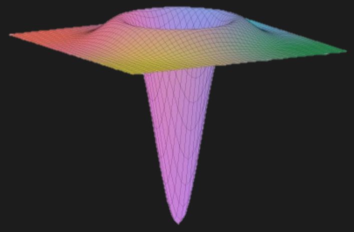
$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Gaussian



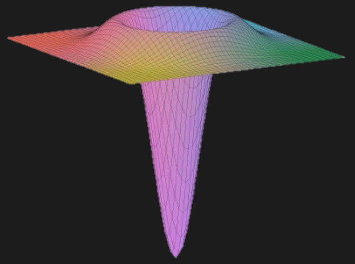
n_σ

LoG



$\nabla^2 n_\sigma$

NLoG



$\sigma^2 \nabla^2 n_\sigma$

Location of Blobs given by Local Extrema after applying Normalized Laplacian of Gaussian at many scales.

Review: 2D Blob Detection Summary

Given an image $I(x, y)$.

Convolve the image using NLoG at many scales σ.

Find:

$$\boxed{(x^*, y^*, \sigma^*) = \underset{(x,y,\sigma)}{\arg\max} \left|\sigma^2 \nabla^2 n_\sigma * I(x, y)\right|}$$

or

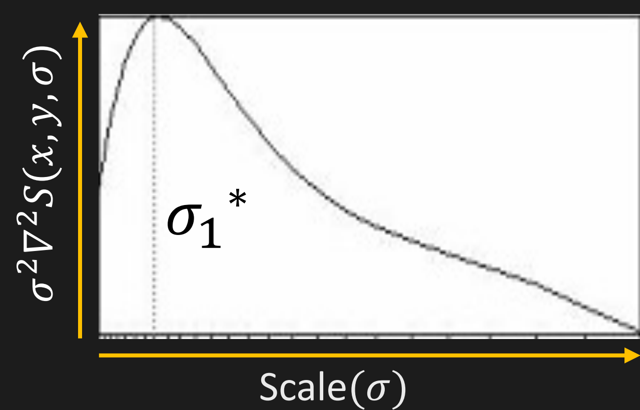
$$\boxed{(x^*, y^*, \sigma^*) = \underset{(x,y,\sigma)}{\arg\max} \left|\sigma^2 \nabla^2 S(x, y, \sigma)\right|}$$

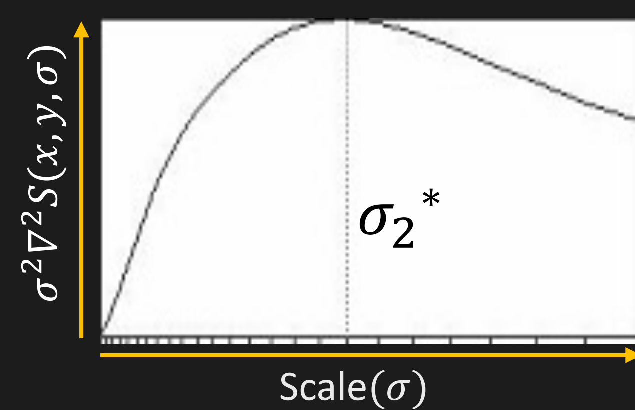
(x^*, y^*): Position of the blob

σ^*: Size of the blob

Review: SIFT Detection Examples

Review: Scale Invariance by Resizing





$\dfrac{\sigma_1{}^*}{\sigma_2{}^*}$:Ratio of Blob Sizes

[Mikolajczyk 2002]

Review: Computing the Principal Orientation

Use the histogram of gradient directions

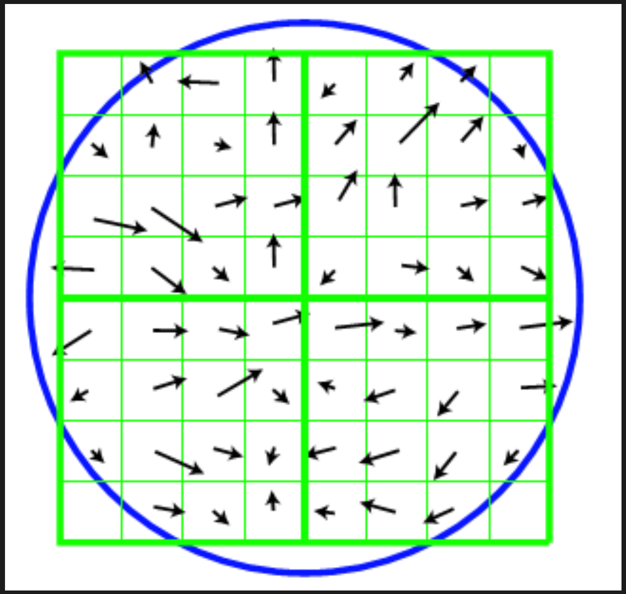
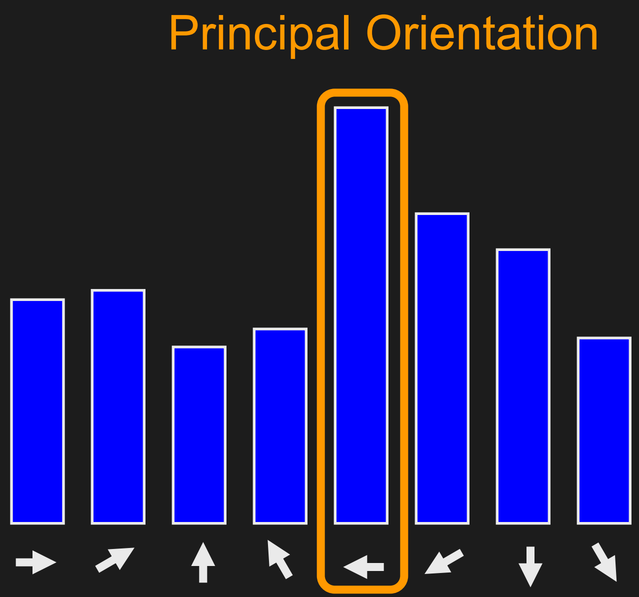


Image gradient directions

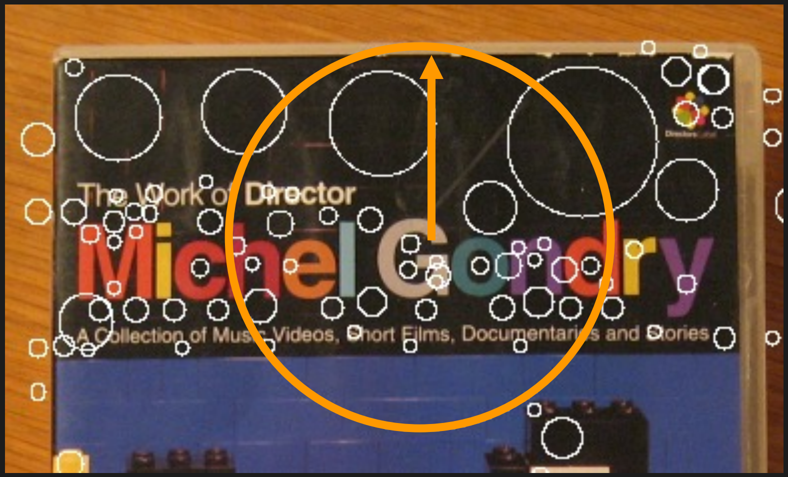
$$\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$$

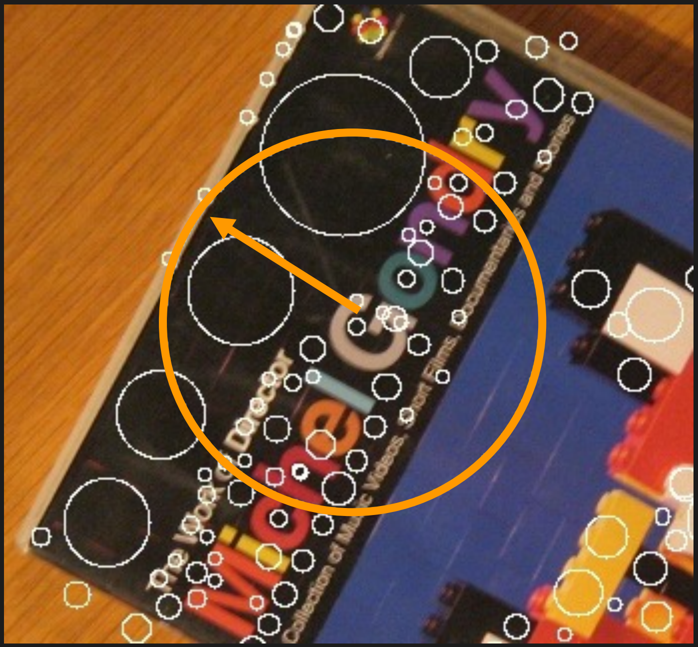


Choose the most prominent gradient direction

Review: Rotation Invariance by Orientation Alignment

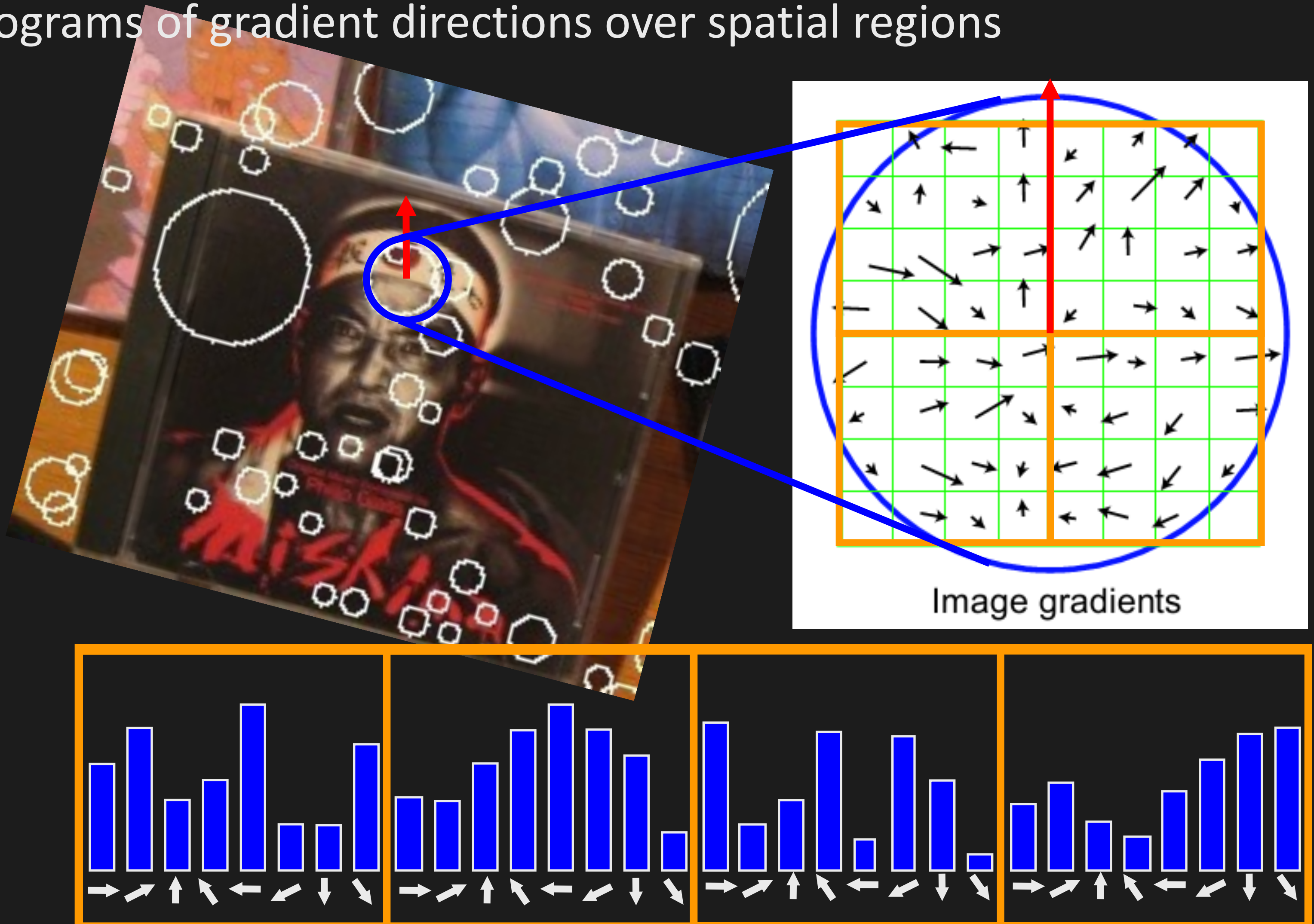
Use the principal orientations to match rotation





The SIFT Descriptor

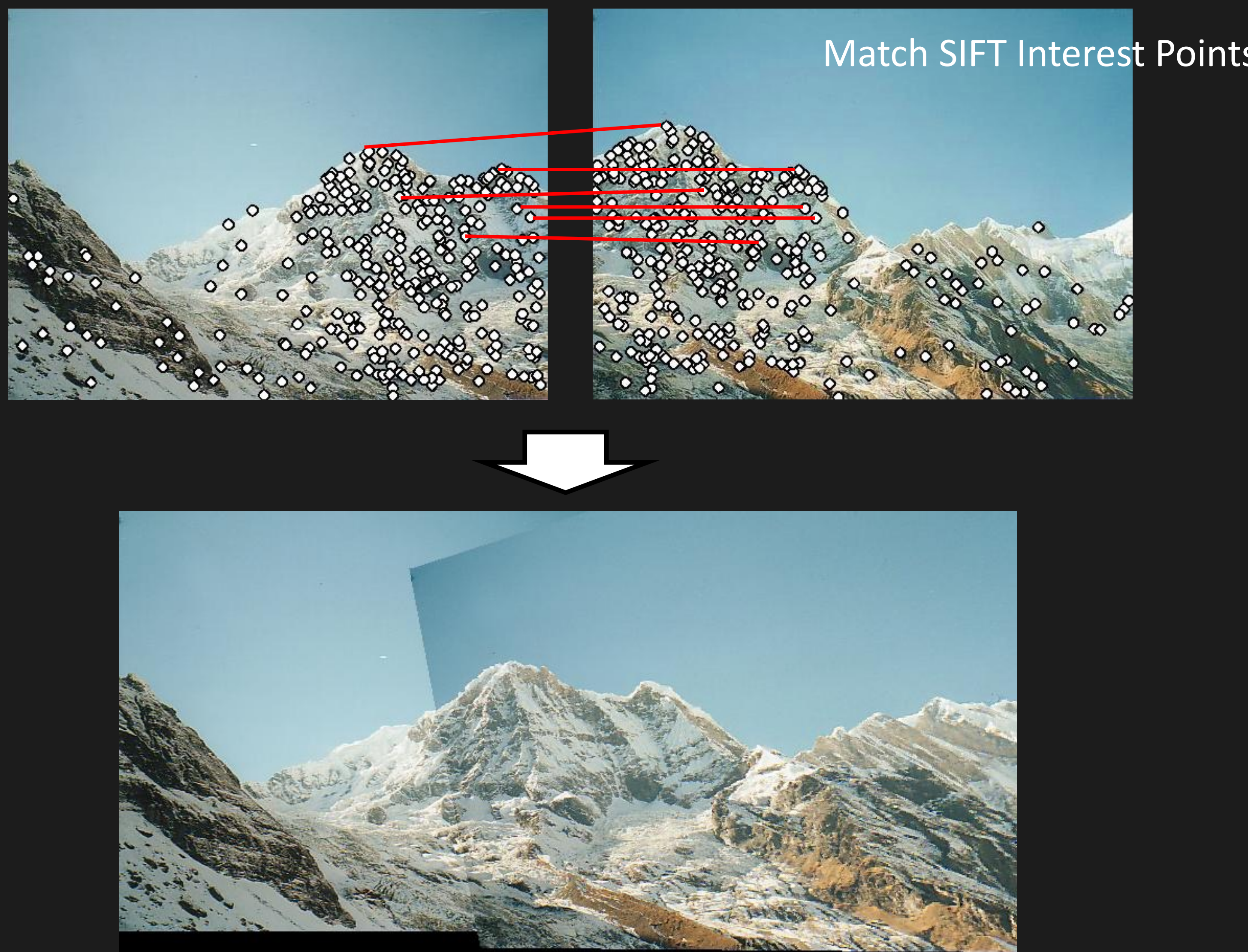
Histograms of gradient directions over spatial regions



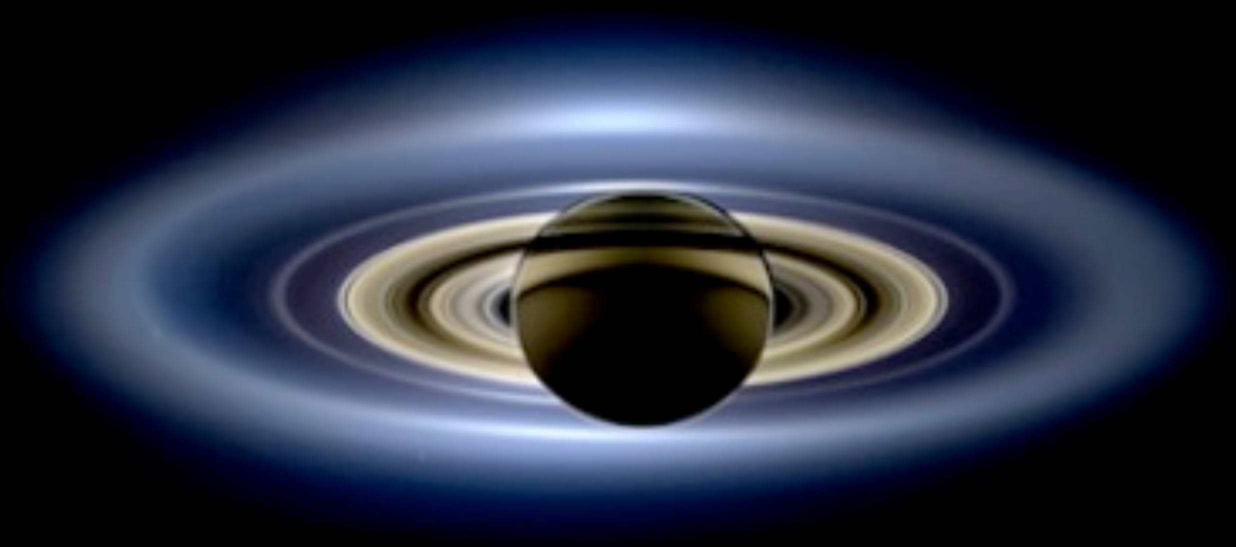
Normalized Histogram: Invariant to Rotation, Scale, Brightness

[Lowe 2004]

Today: Panorama Stitching using SIFT



[Autostitch]

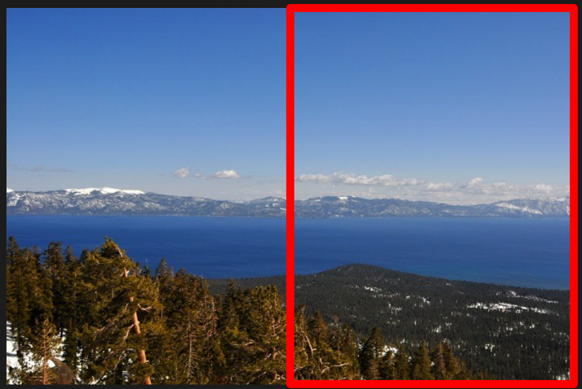


With the Cassini satellite's wide-angle camera aimed at Saturn, Cassini was able to capture 323 images in just over four hours in 2013. This final panorama used 141 of those images taken using red, green and blue spectral filters.

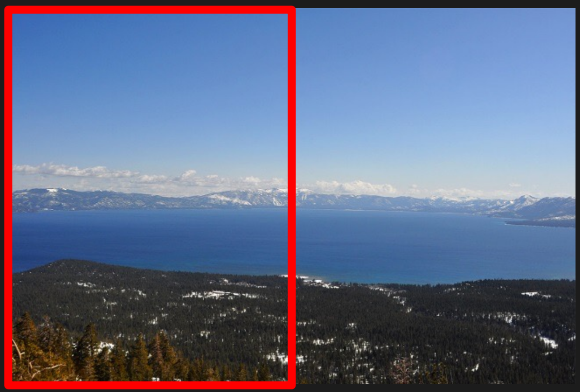
http://www.jpl.nasa.gov/spaceimages/details.php?id=pia17172

www.gigapan.com

Image Stitching Process



Source Image 1

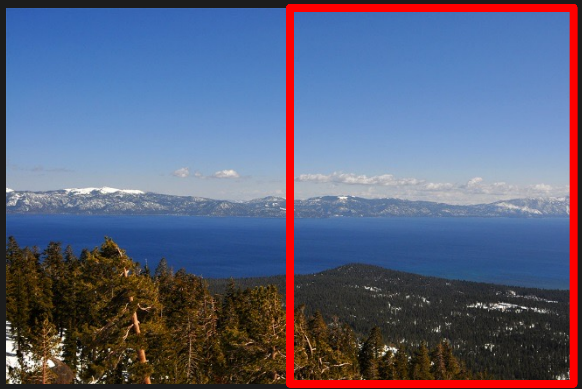


Source Image 2

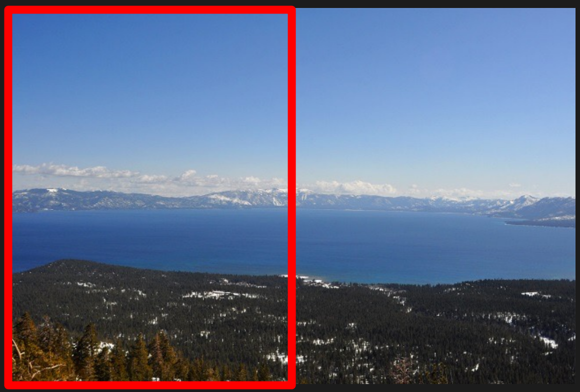
Source Image 3

How would you align these images?

Image Stitching Process



Source Image 1



Source Image 2

Source Image 3

How would you align these images?

Image Stitching Process

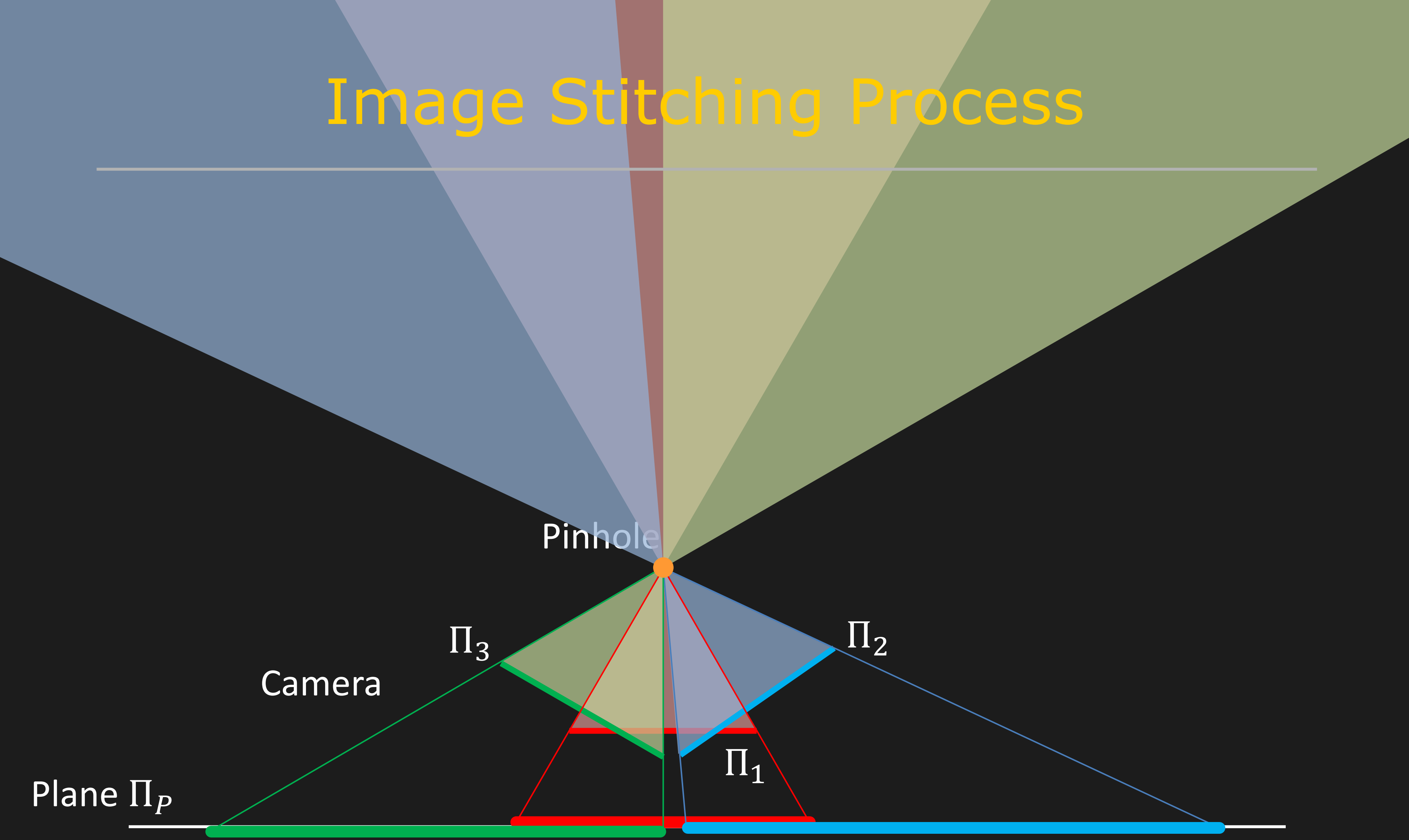


Image Stitching Process

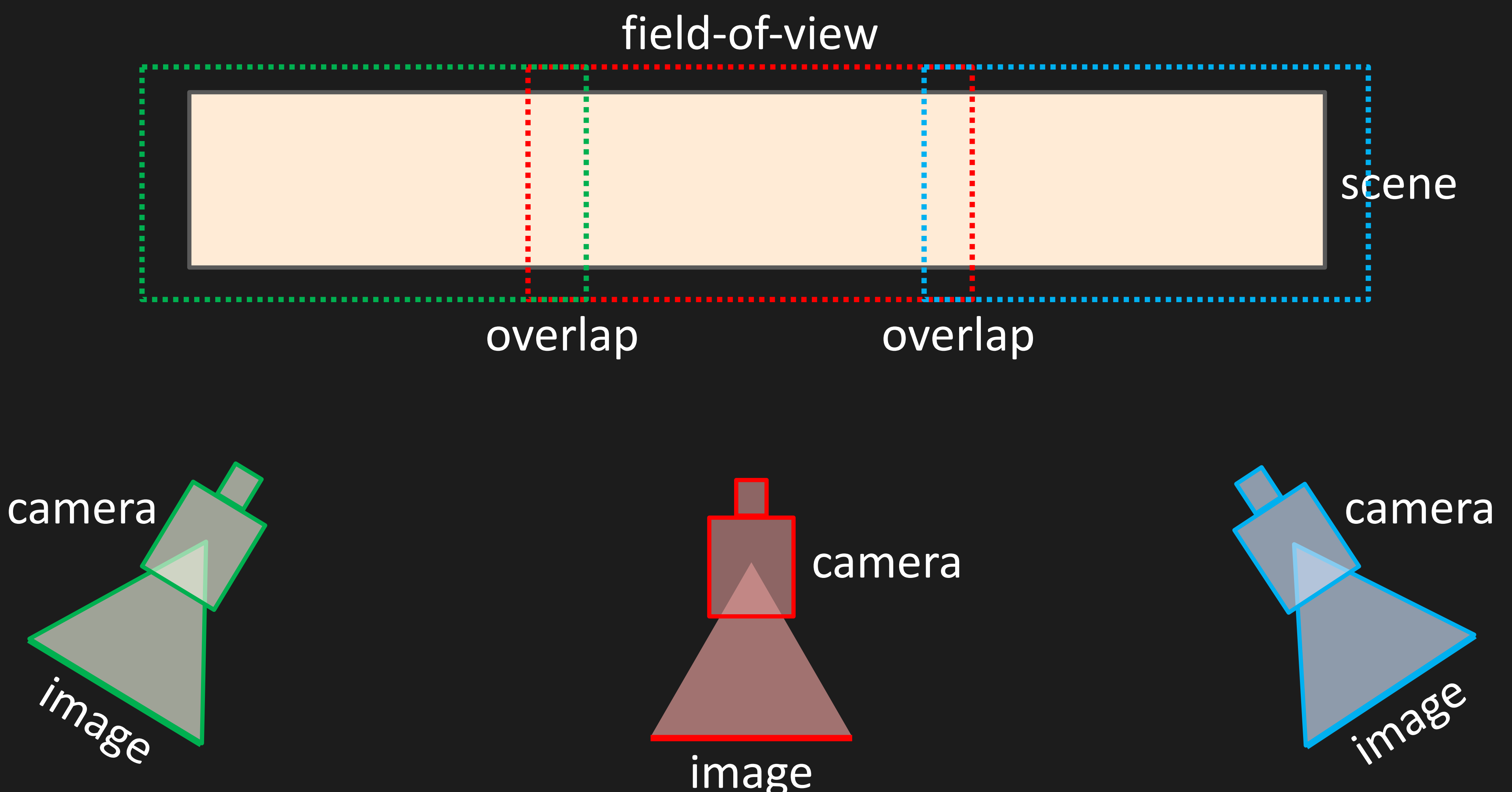
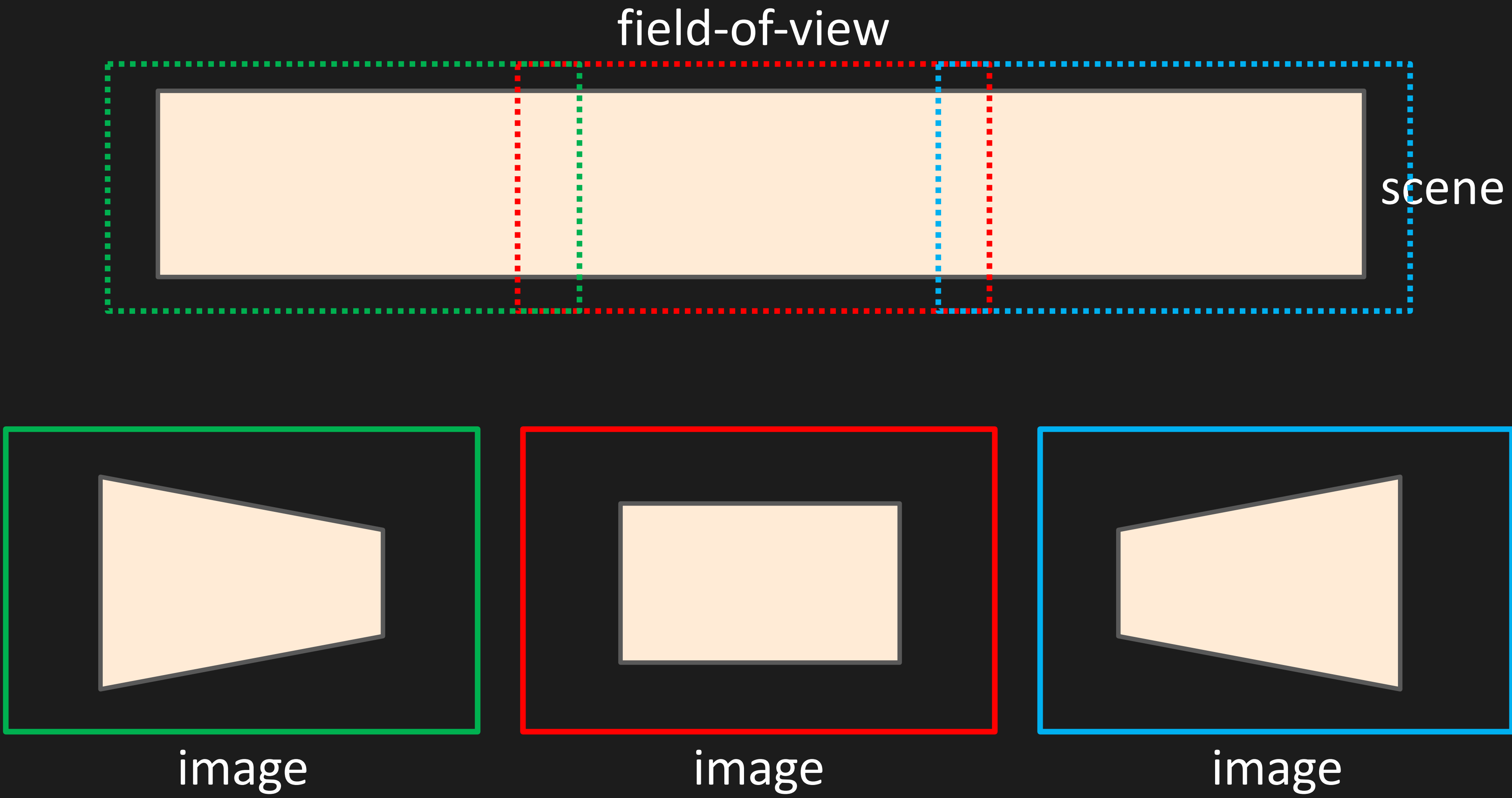


Image Stitching Process

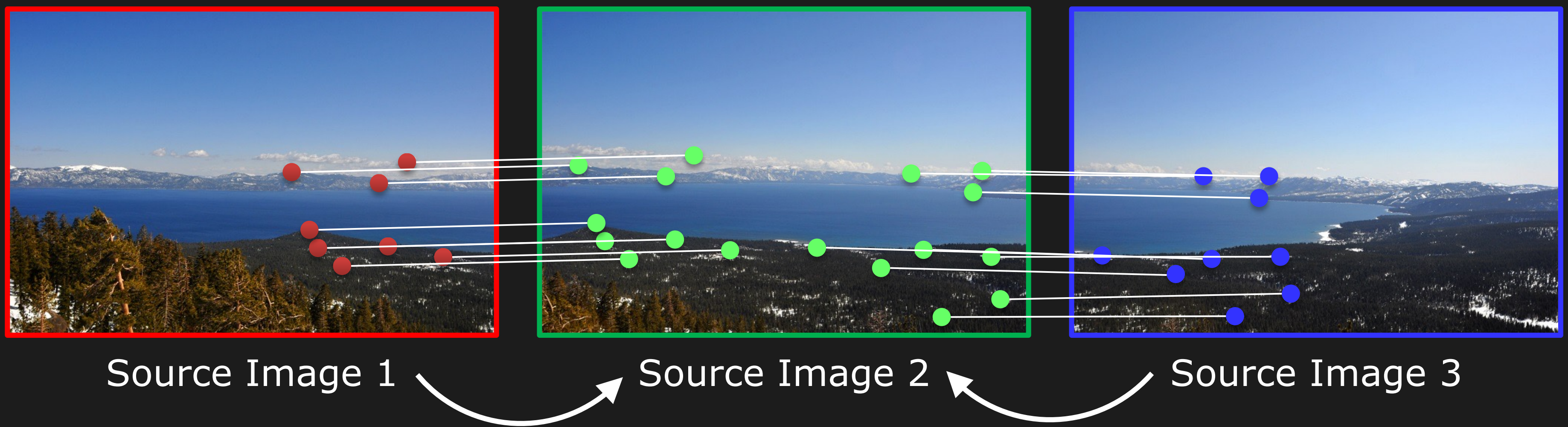


Overlap, but not Aligned

Image Stitching Process

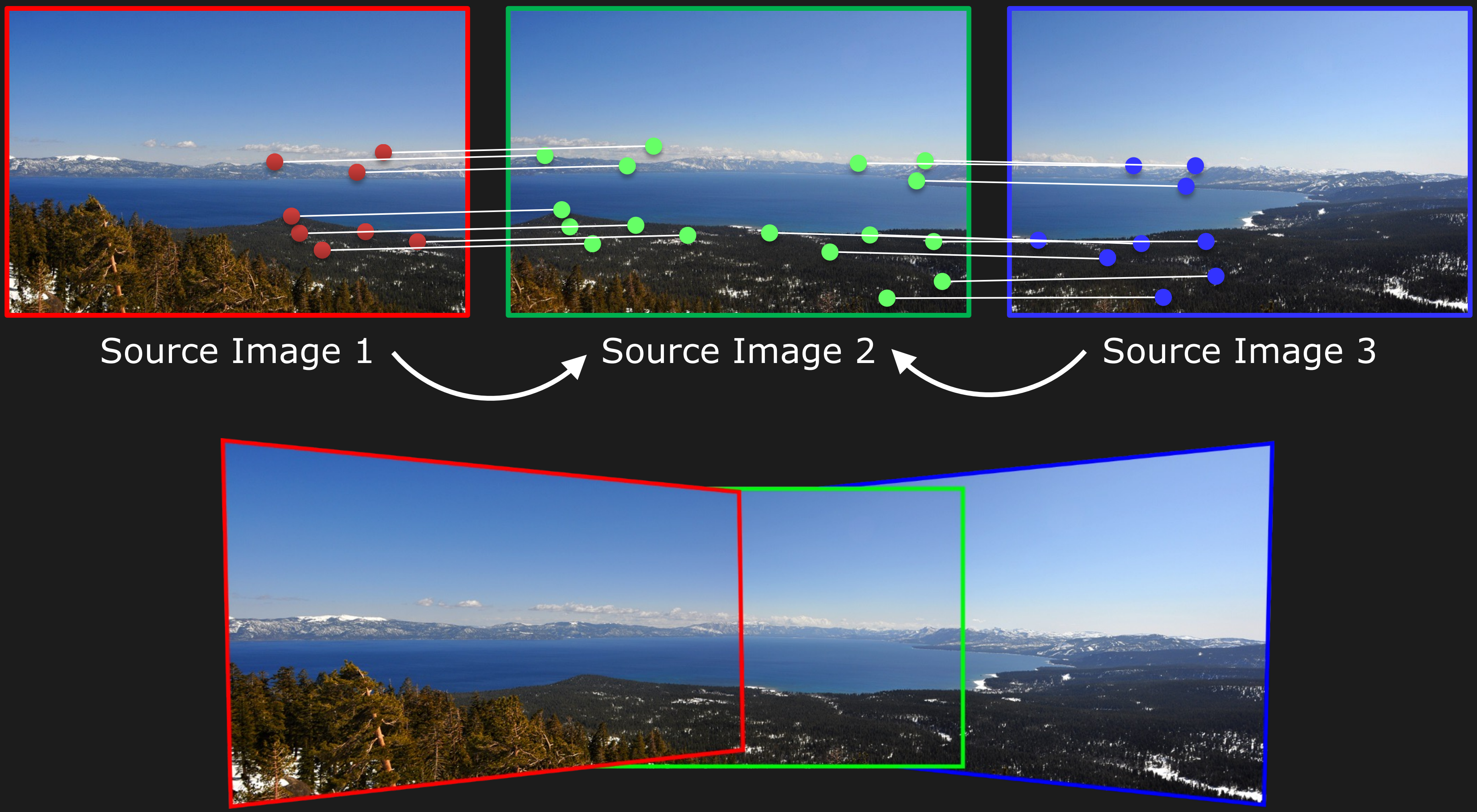
Find corresponding points
(using feature detectors like SIFT)

Image Stitching Process



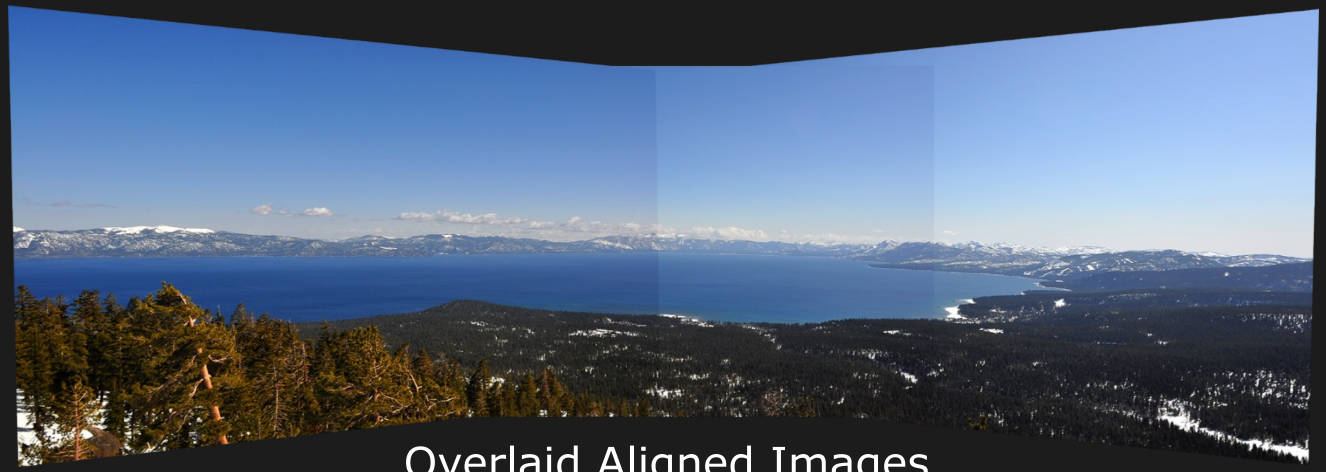
Find geometric relationship between the corresponding points and therefore the images.

Image Stitching Process

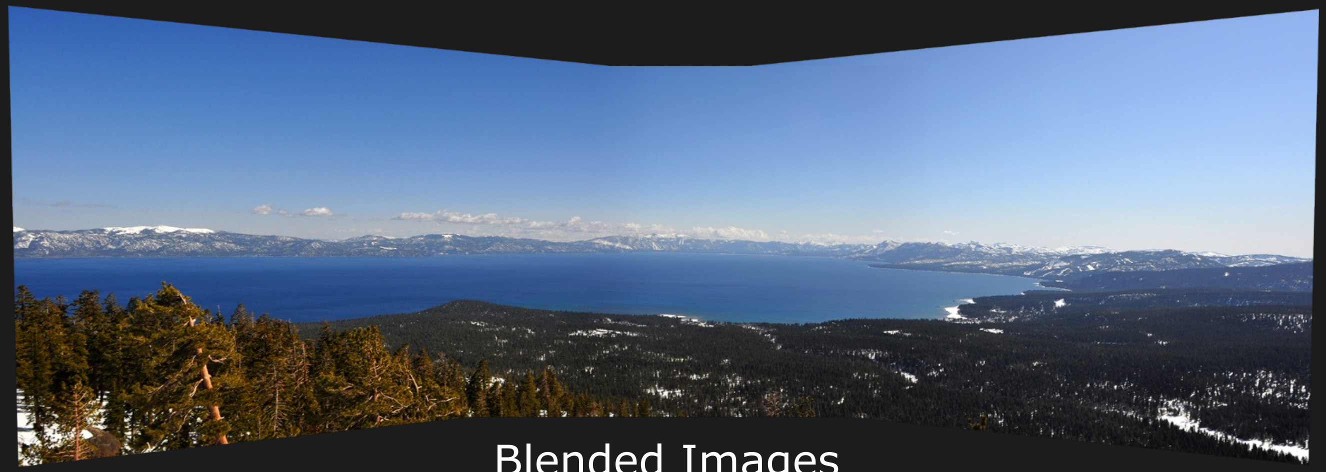


Warp images so that corresponding points align.

Image Stitching Process



Overlaid Aligned Images



Blended Images

Blend images to remove hard seams

Image Alignment and Stitching

Combine multiple photos to create a larger photo

Topics:

(1) Image Transformations

(2) Computing Transformations

(3) Warping Images

(4) Blending Images

Image Manipulation

Image Filtering: Change range (brightness) of image

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

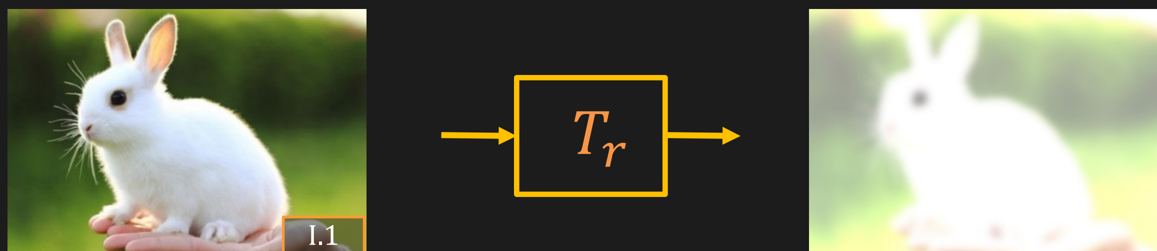
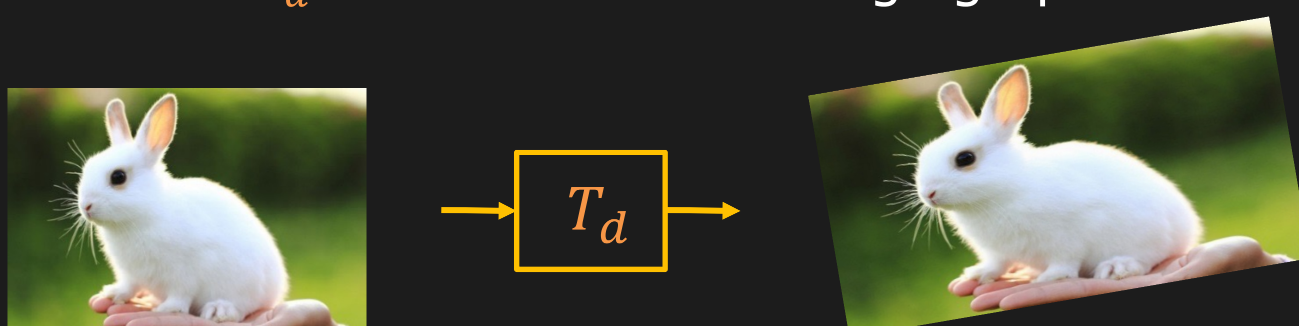


Image Warping: Change domain (location) of image

$$g(x, y) = f\left(T_d(x, y)\right)$$

Transformation T_d is a coordinate changing operator.



Global Warping/Transformation

Translation

Rotation

Scaling and Aspect

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

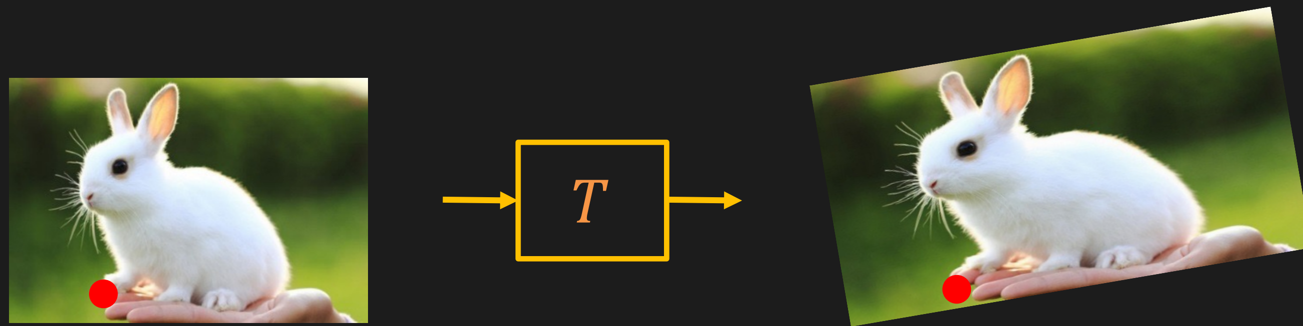
Affine

Perspective

Barrel

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

Linear Transformations



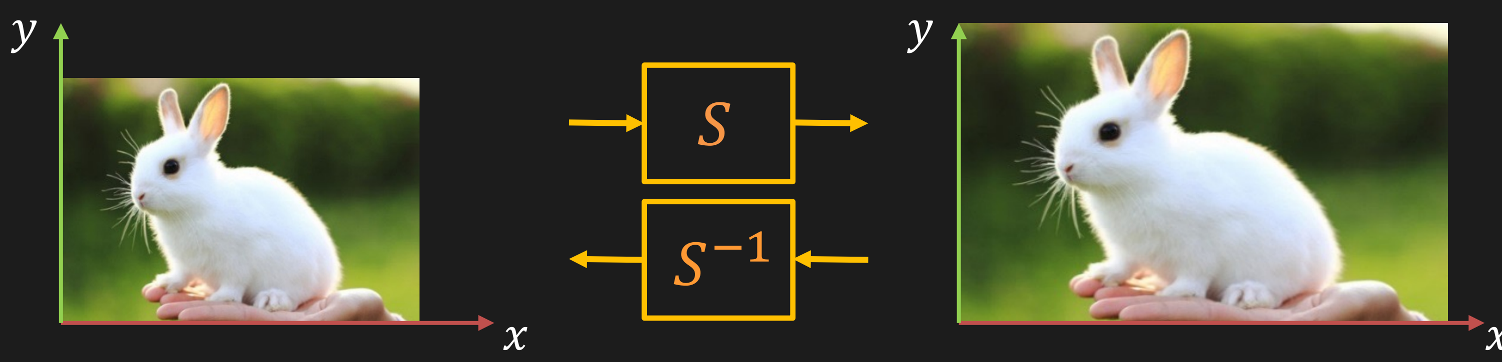
$\mathbf{p}_1 = (x_1, y_1)$

$\mathbf{p}_2 = (x_2, y_2)$

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 or Squishing)



Forward:

$$x_2 = a x_1 \qquad y_2 = b y_1$$

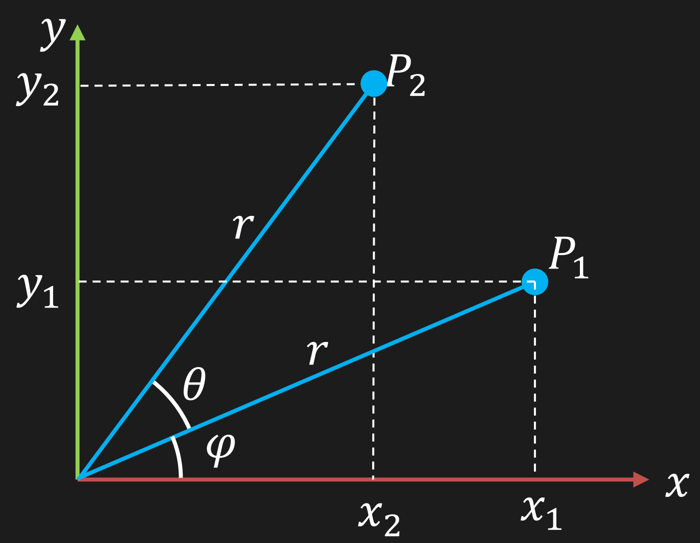
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

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

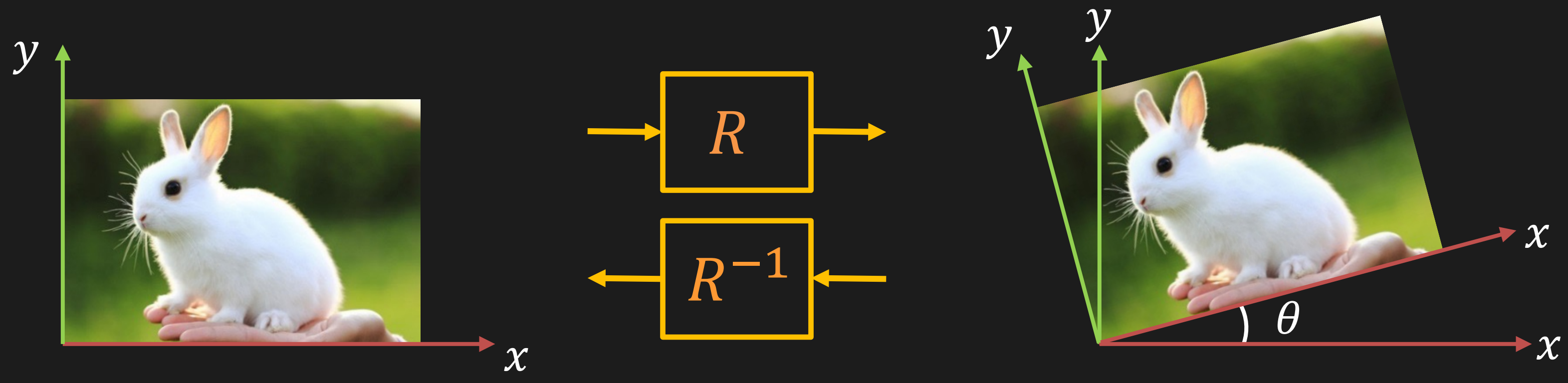


$$x_1 = r\cos(\varphi)$$
$$y_1 = r\sin(\varphi)$$

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

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

Rotation



Forward:

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

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

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

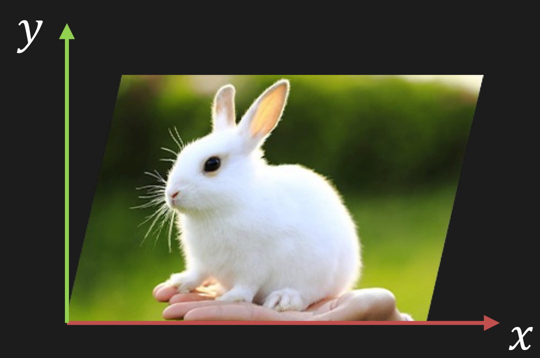
Inverse:

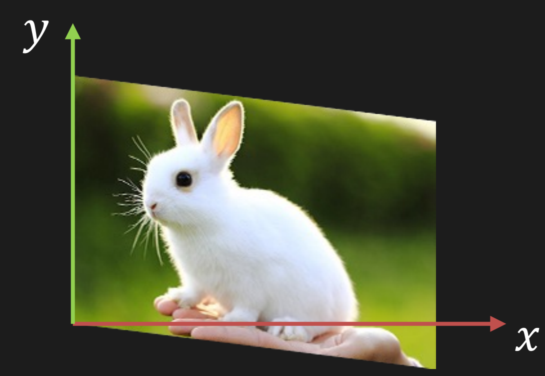
$$x_1 = x_2 cos\theta + y_2 sin\theta$$

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

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

Skew





Horizontal Skew:

$$x_2 = x_1 + m_x y_1$$

$$y_2 = y_1$$

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

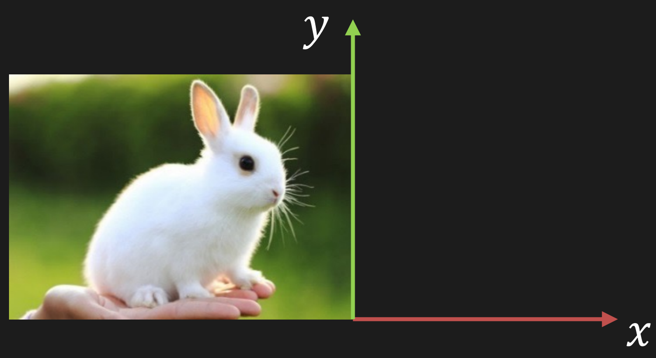
Vertical Skew:

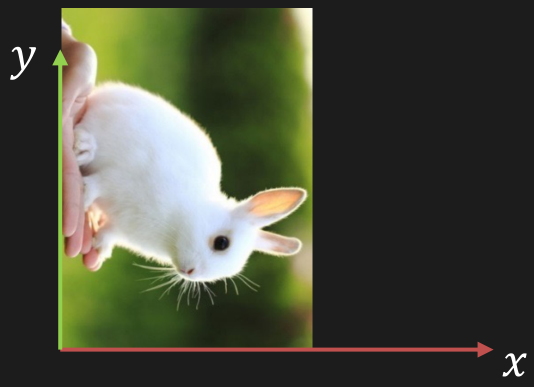
$$x_2 = x_1$$

$$y_2 = m_y x_1 + y_1$$

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

Mirror





Mirror about Y-axis:

$$x_2 = -x_1$$

$$y_2 = y_1$$

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

Mirror 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

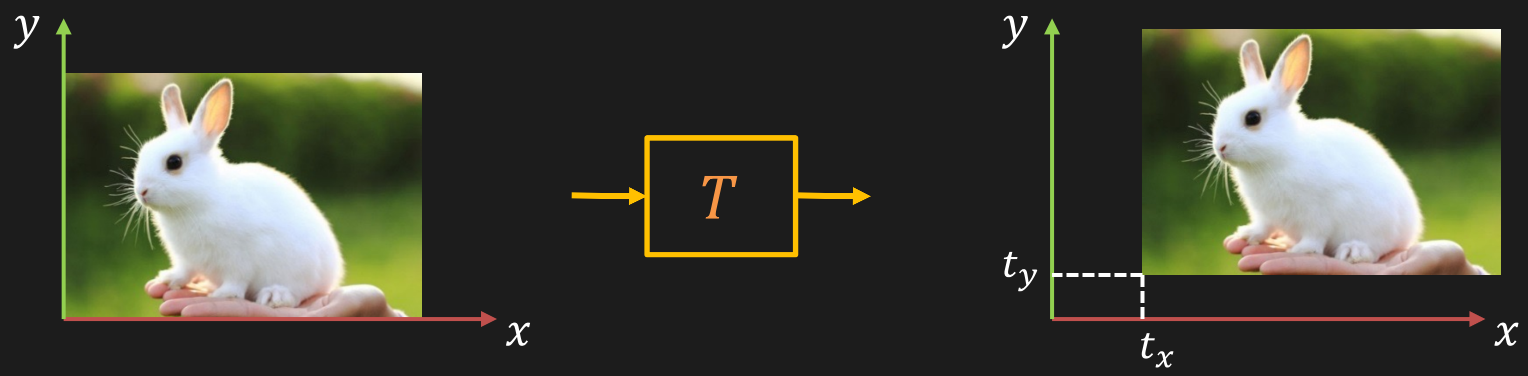
Any transformation of the form:

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

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

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

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

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

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

Homogeneous Coordinates

Converting ***to*** *homogeneous coordinates:*

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

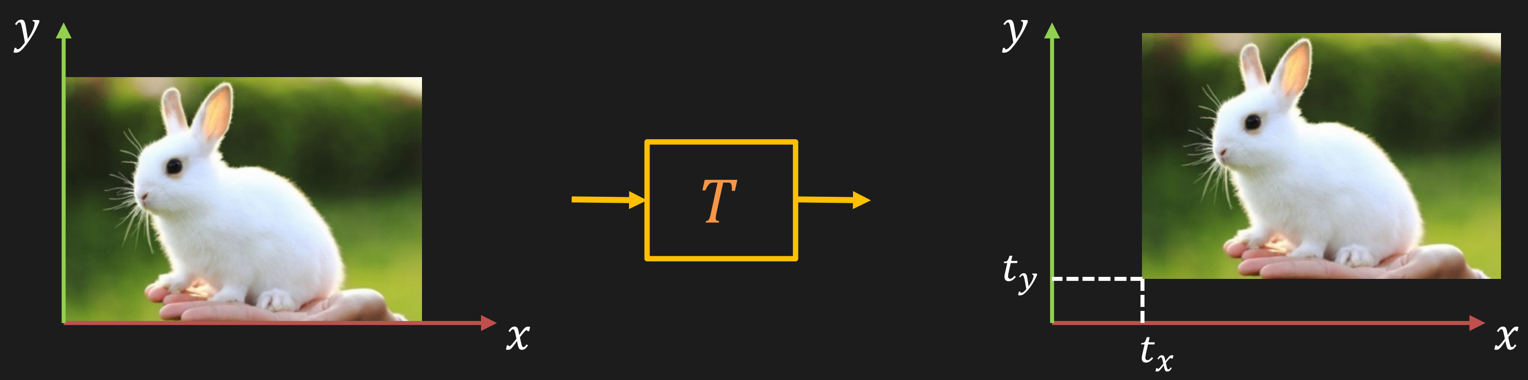
homogeneous image coordinates

Converting ***from*** *homogeneous coordinates:*

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Slide by Steve Seitz

Translation



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

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

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

Affine Transformation

Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix}$$

Affine Transformation

Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix}$$

- 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:

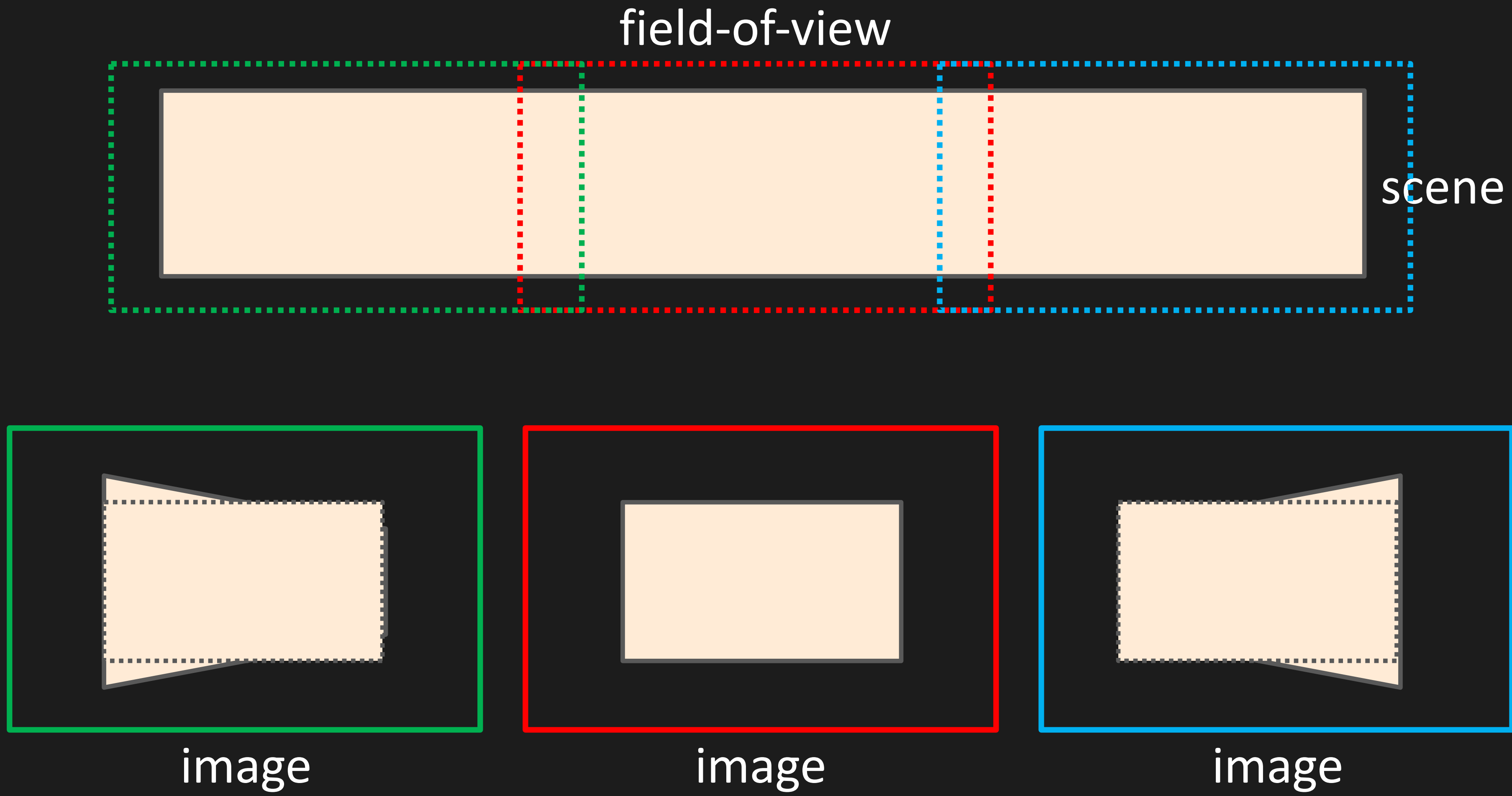
$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix}h_{11} & h_{12} & h_{13}\\ h_{21} & h_{22} & h_{23}\\ h_{31} & h_{32} & h_{33}\end{bmatrix}\begin{bmatrix}\tilde{x}_1\\ \tilde{y}_1\\ \tilde{z}_1\end{bmatrix} \qquad \widetilde{\mathbf{p}}_2 = H\widetilde{\mathbf{p}}_1$$

Also called Homography.

Projective Transformation

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

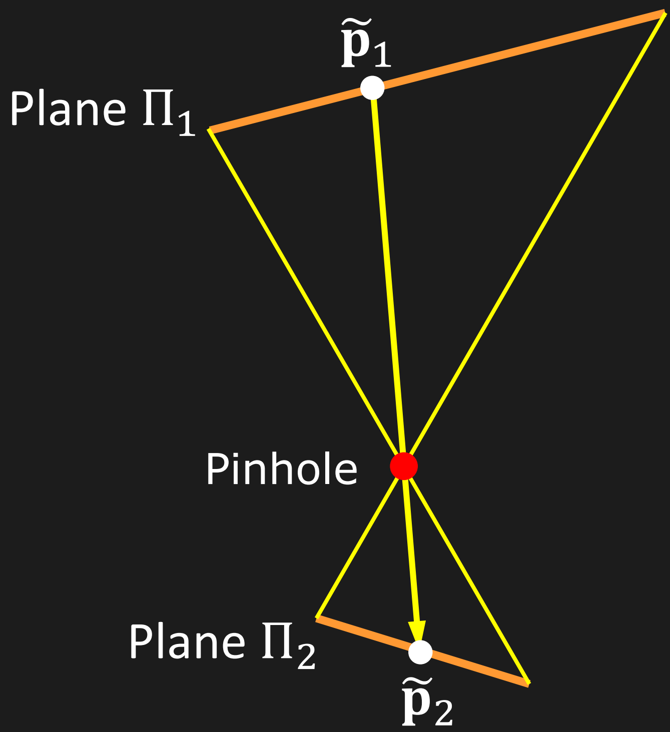
Image Warping



Remember Vanishing Points?

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.

Example

Consider the homography H that maps points in image 2 into points in image 1.

Where does the pixel at coordinates (10,5) in image 2 project to in image 1

$$H = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 0 \\ 1 & 1 & 3 \end{bmatrix}$$

Example

1. Convert point in image 2 to homogeneous coordinates.

$$\hat{p}_2 = (10,5,1)^T$$

2. Compute the projective transformation $\hat{p}_1 = H\hat{p}_2$

$$\hat{p}_1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 0 \\ 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 10 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 23 \\ 45 \\ 18 \end{bmatrix}$$

3. Convert $\hat{p}_1$ to cartesian coordinates

$$p_1 = \left(\frac{23}{18}, \frac{45}{18}\right)$$

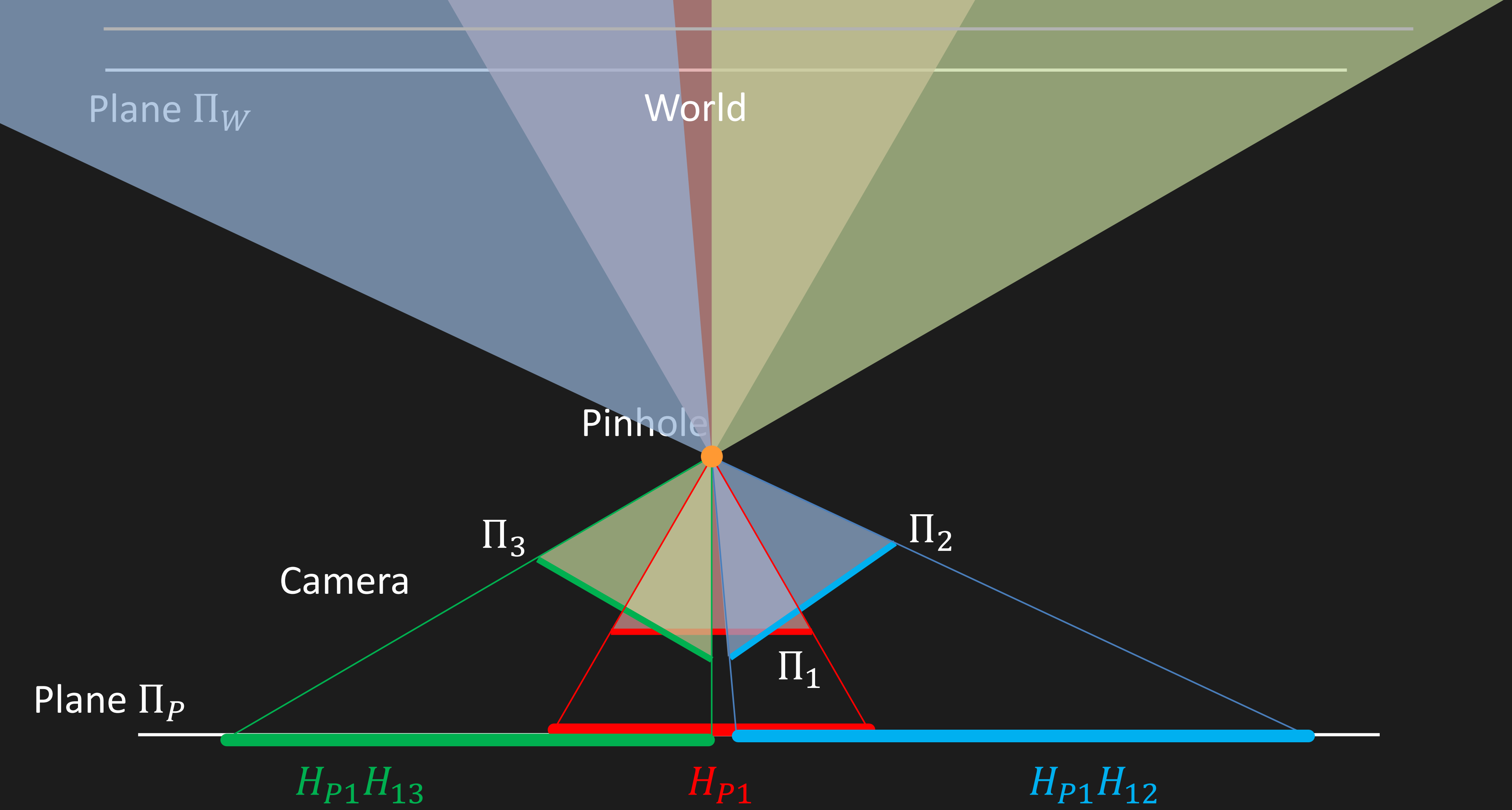
Projective Transformation

Homography can only be defined up to a scale.

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

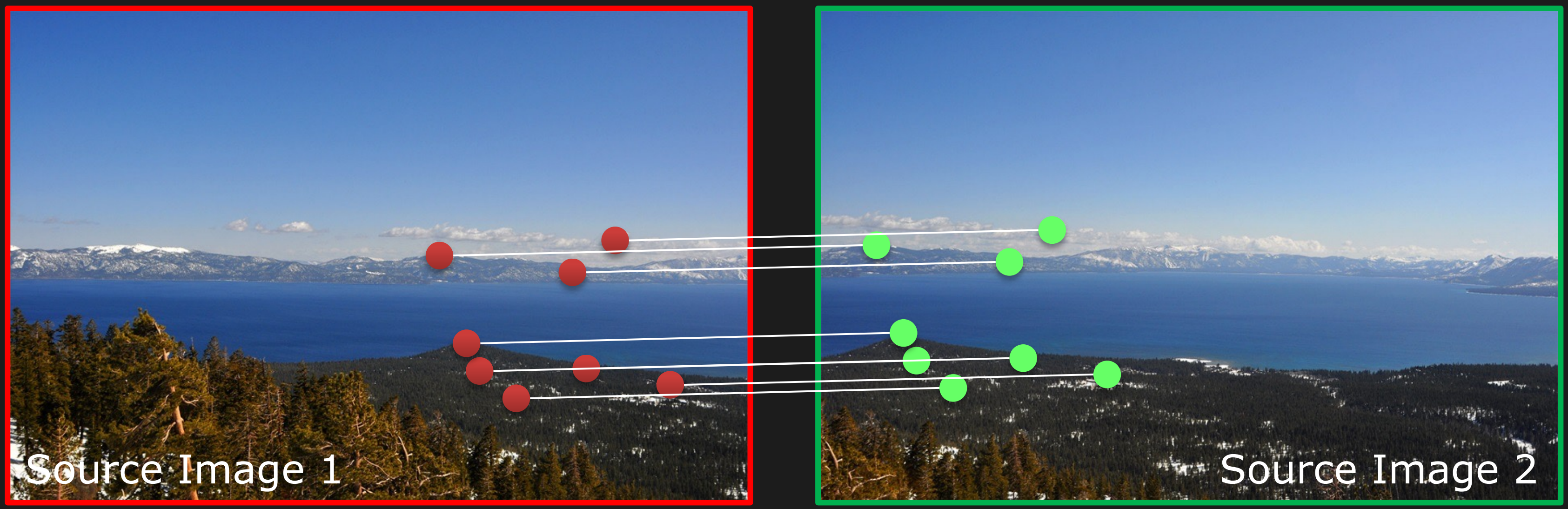
If we fix scale such that $\sqrt{\Sigma\left(h_{ij}\right)^2} = 1$ then 8 free parameters

Homography Composition



Useful in stitching planar panoramas.

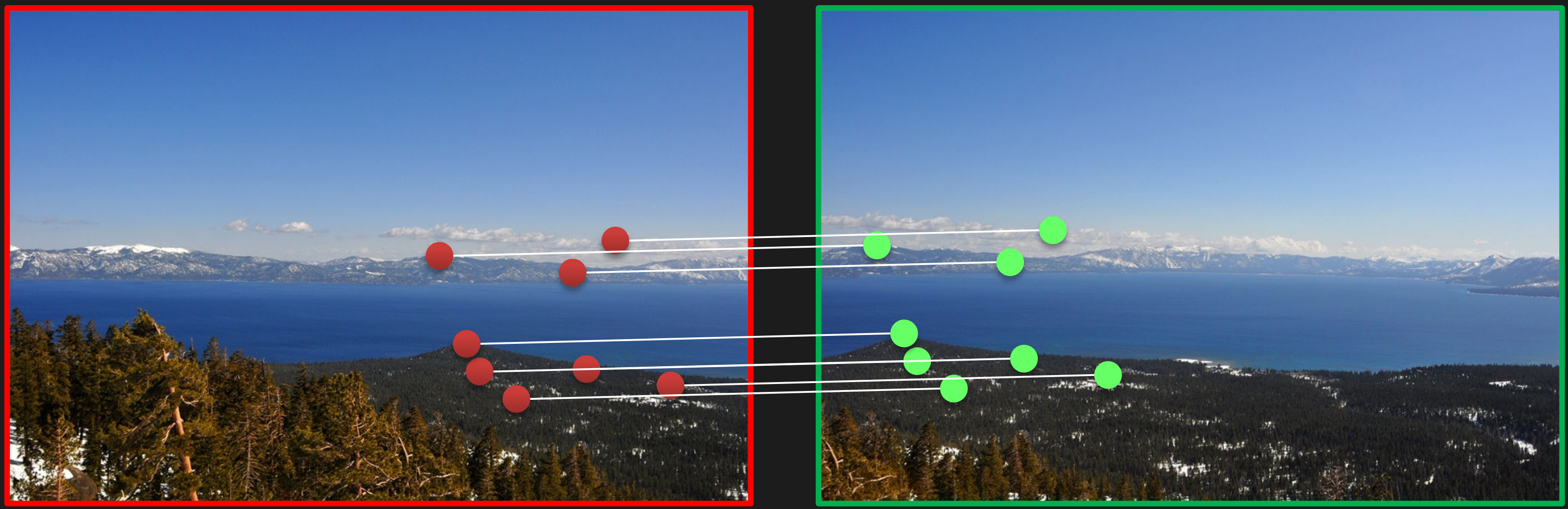
Computing Homography



Given a set of matching features/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.

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.

Computing Homography

For a given pair i of corresponding points:

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

Rearranging the terms:

$$\Rightarrow \quad \begin{aligned} {x_d}^{(i)}\left(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}\right) &= h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13} \\ {y_d}^{(i)}\left(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}\right) &= h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23} \end{aligned}$$

Computing Homography

For a given pair i of corresponding points:

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

$$\Rightarrow$$

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

Rearranging the terms:

$$x_d{}^{(i)}\left(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}\right) - h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13} = 0$$

$$\Rightarrow$$

$$y_d{}^{(i)}\left(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}\right) - h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23} = 0$$

Computing Homography

$$x_d{}^{(i)}\left(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}\right) - h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13} = 0$$

$$y_d{}^{(i)}\left(h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}\right) - h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23} = 0$$

Rearranging the terms and writing as linear equation:

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

(Known)

$\mathbf{h}$

(Unknown)

Computing Homography

Combining the equations for all corresponding points:

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

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

Constrained Least Squares

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

Define least squares problem:

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

We know that:

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

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

Constrained Least Squares

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

Define Loss 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 $L(\mathbf{h})$.

Numpy: `np.linalg.eig(A.T@A)` returns eigenvalues and vectors of $A^T A$

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