SIFT Detector

Shree K. Nayar

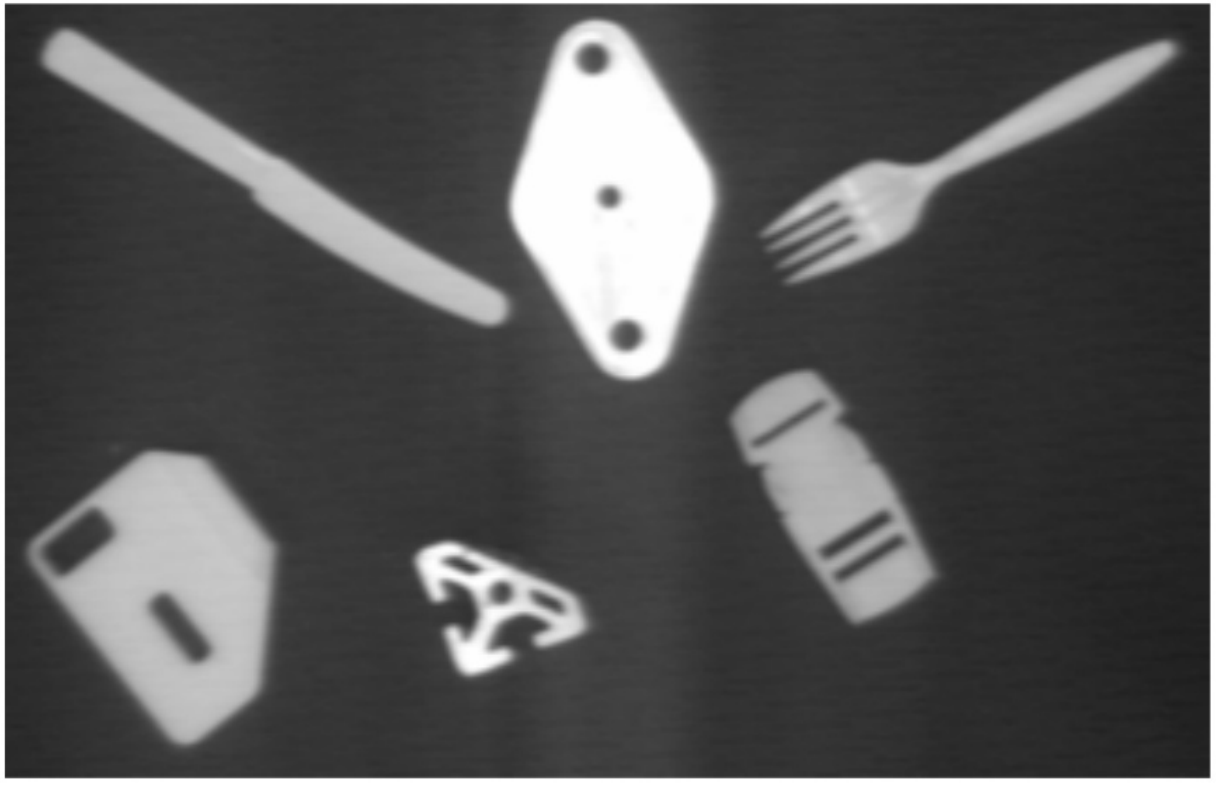
Columbia University

Topic: SIFT Detector, Module: Features

First Principles of Computer Vision

A Little Quiz

How would you recognize the following types of objects?



Objects on an assembly line

A Little Quiz

How would you recognize the following types of objects?



License plates

A Little Quiz

How would you recognize the following types of objects?



Template



Rich 2D image

Find and Match “Interesting Points or Features”

SIFT Detector

Scale Invariant Feature Transform (SIFT) and its use for image alignment and 2D object recognition.

Topics:

(1) What is an Interest Point?

(2) Detecting Blobs

(3) SIFT Detector

(4) SIFT Descriptor

What is an Interest Point?

Shree K. Nayar

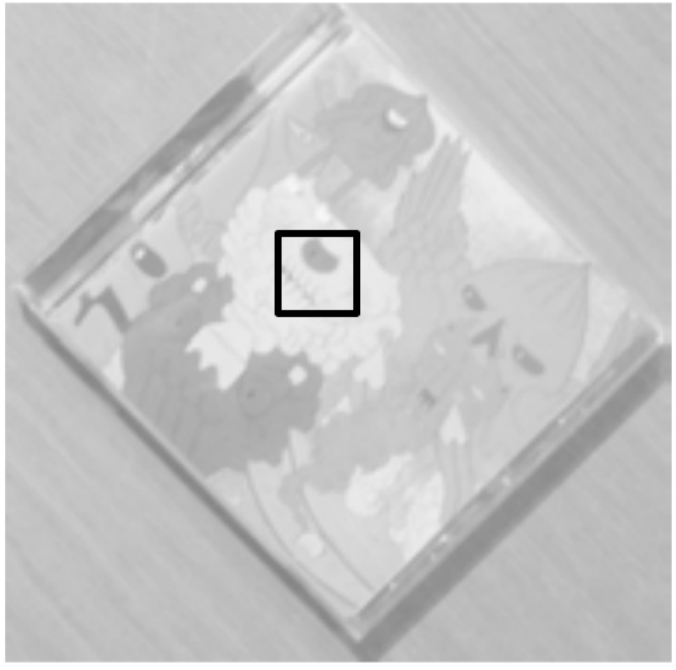
Columbia University

Topic: SIFT Detector, Module: Features

First Principles of Computer Vision

Raw Images are Hard to Match





Different size, orientation, lighting, brightness, etc.

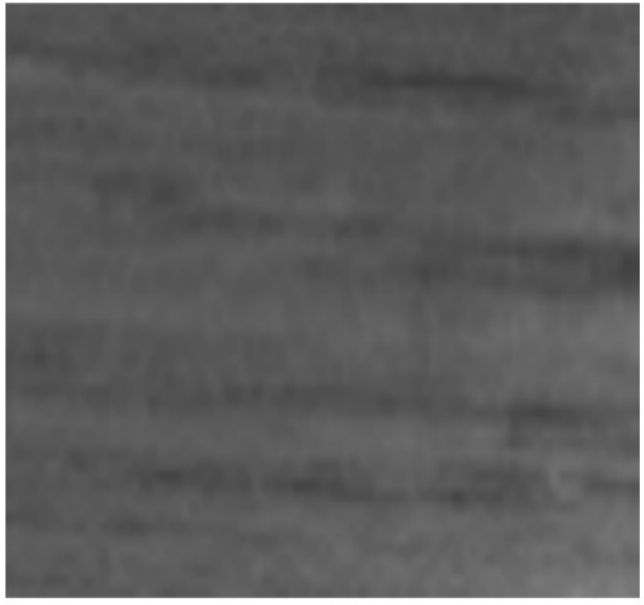
Removing Sources of Variation

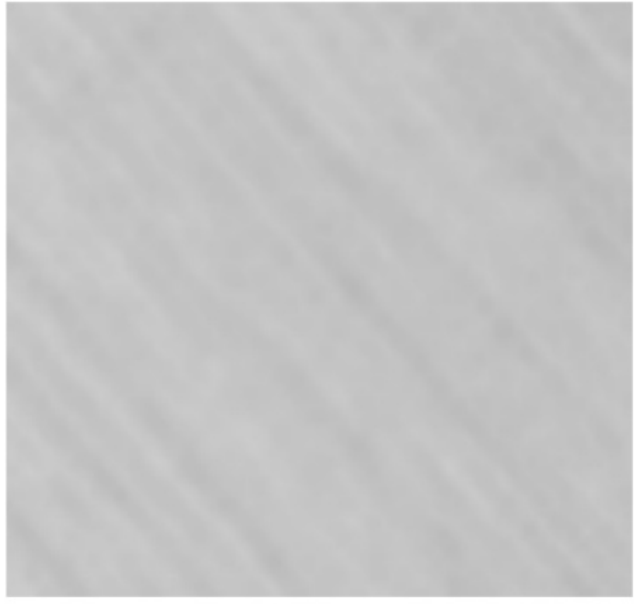




Matching becomes easier if we can remove variations like size and orientation.

Some Patches are not “Interesting”

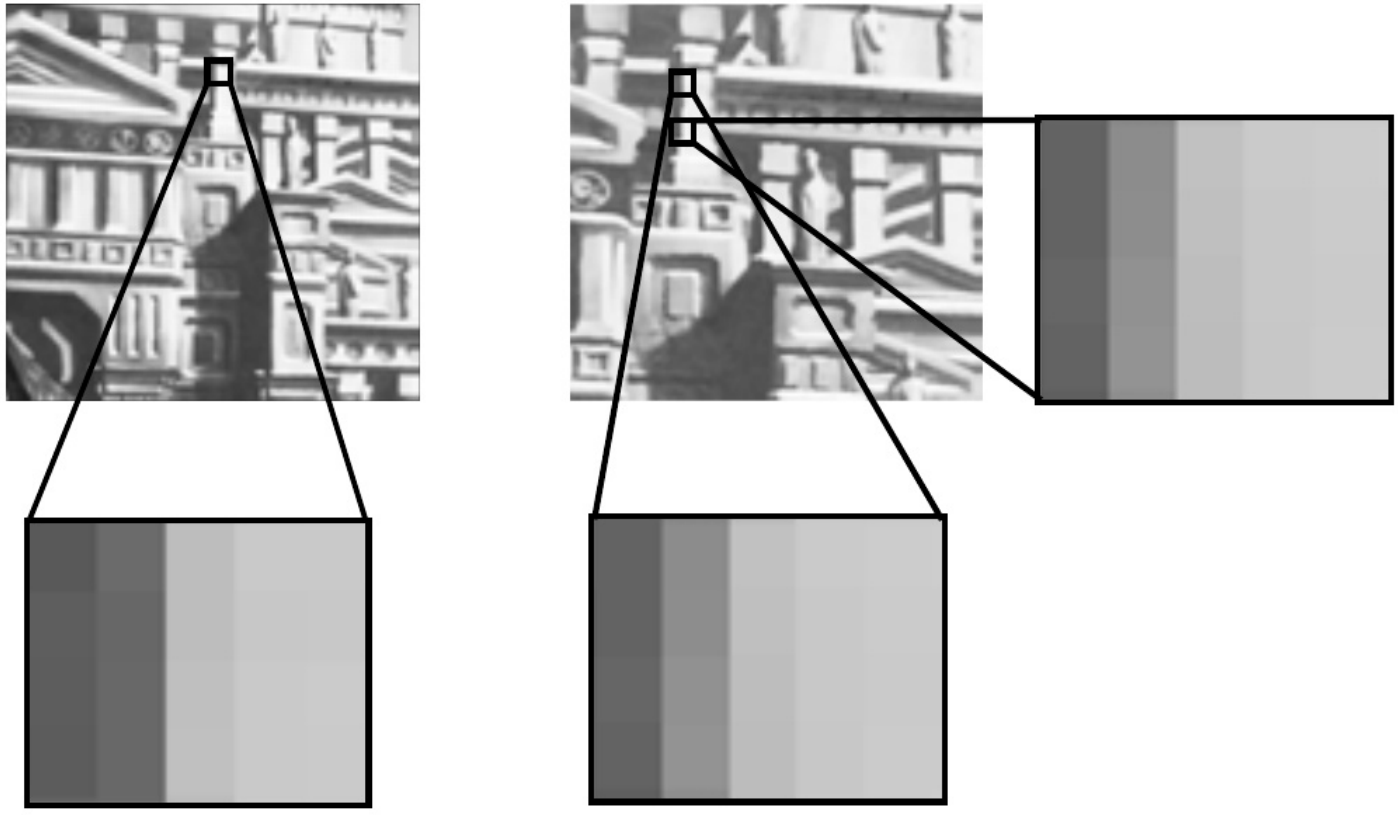




What is an Interesting Point/Feature?

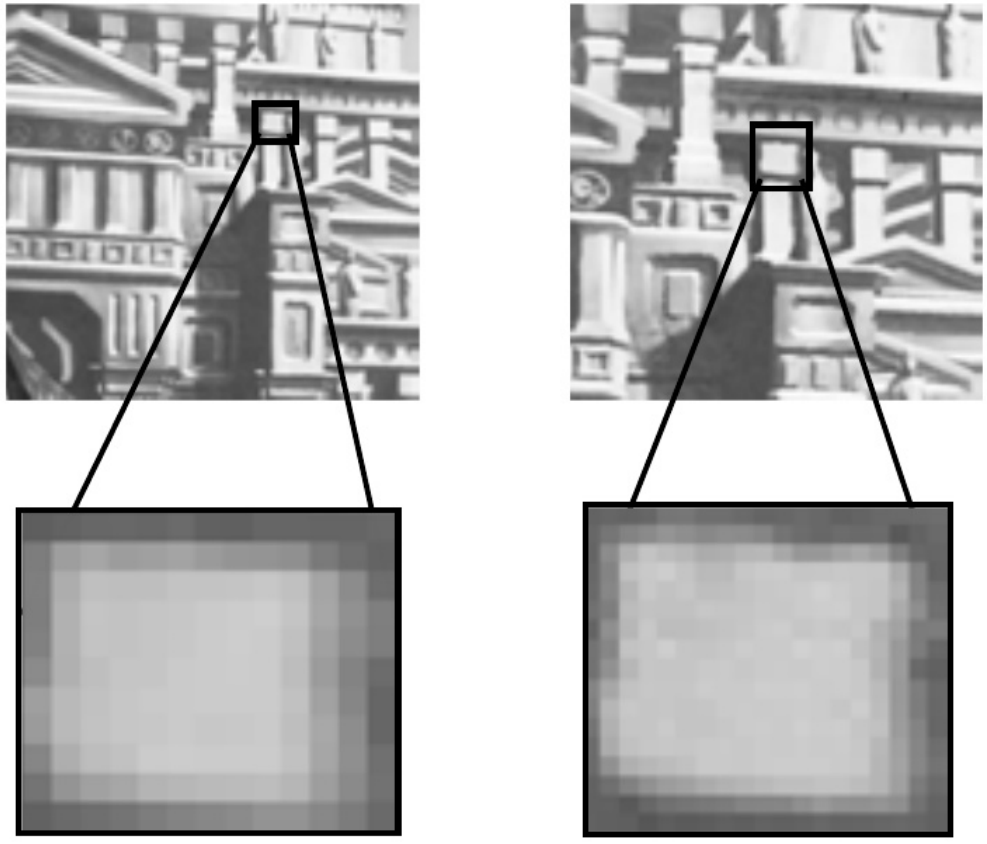
- Has rich image content (brightness variation, color variation, etc.) within the local window
- Has well-defined representation (signature) for matching/comparing with other points
- Has a well-defined position in the image
- Should be invariant to image rotation and scaling
- Should be insensitive to lighting changes

Are Lines/Edges Interesting?



Cannot "Localize" an Edge

Are Blobs Interesting?

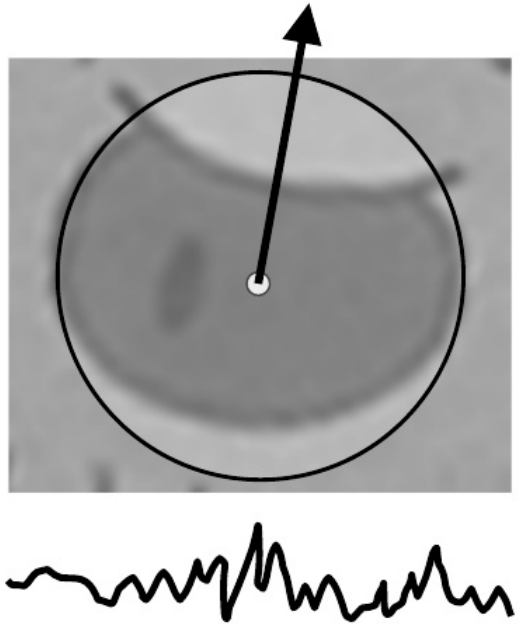


Yes! Blobs have fixed position and definite size.

Blobs as Interest Points

For a Blob-like Feature to be useful, we need to:

- Locate the blob
- Determine its size
- Determine its orientation
- Formulate a description or signature that is independent of size and orientation



Detecting Blobs

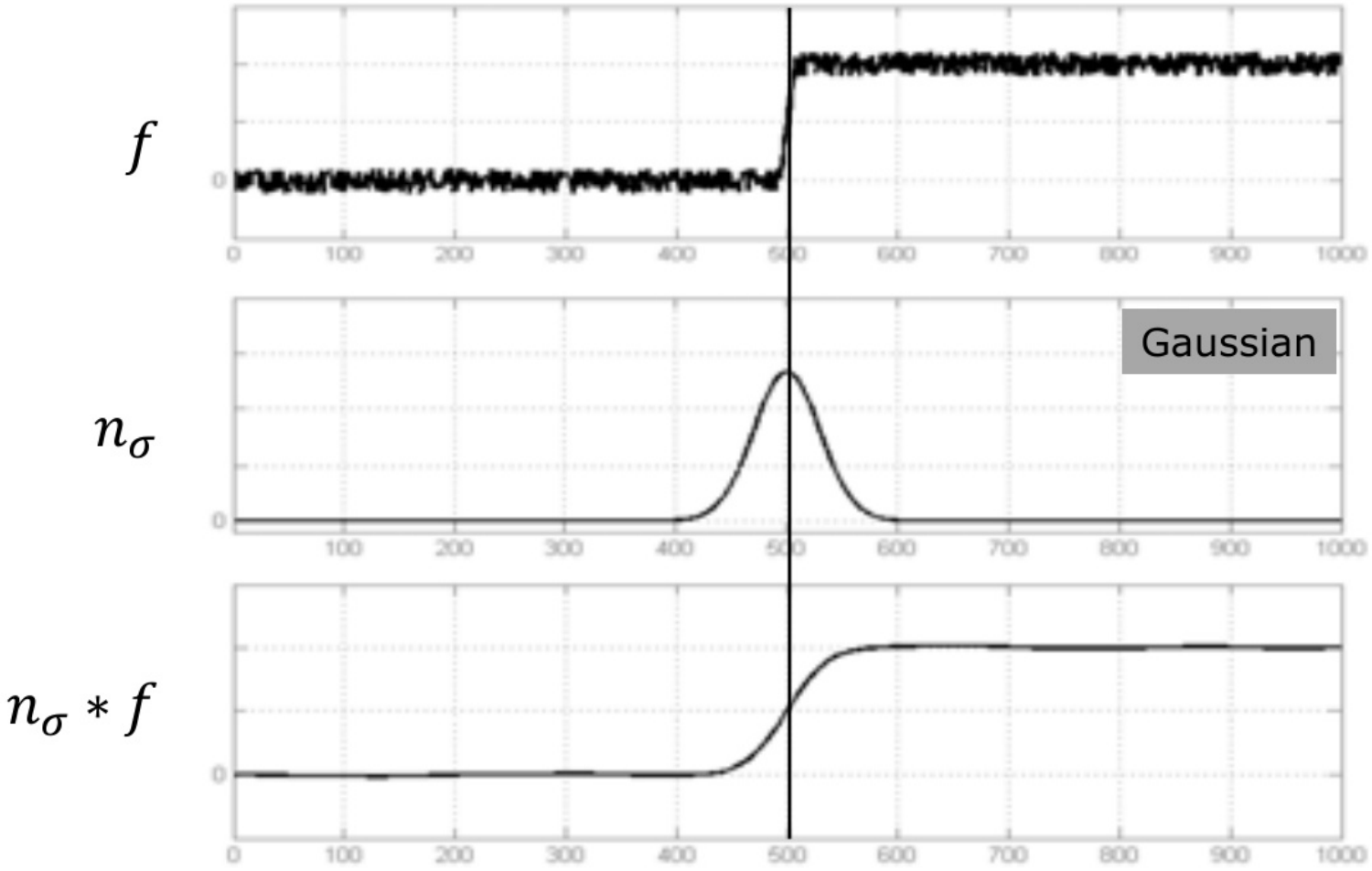
Shree K. Nayar

Columbia University

Topic: SIFT Detector, Module: Features

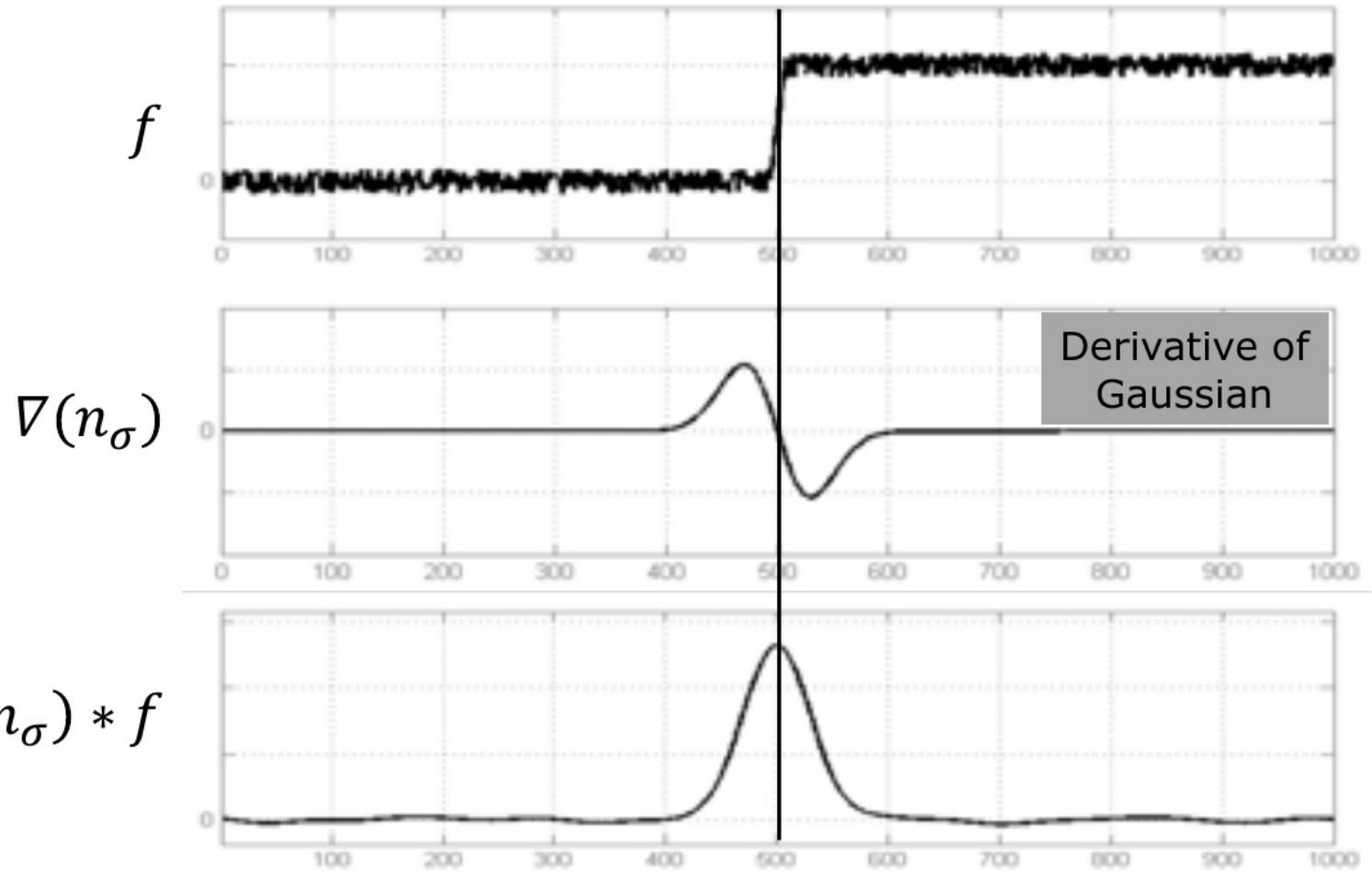
First Principles of Computer Vision

Review: Gaussian Filter



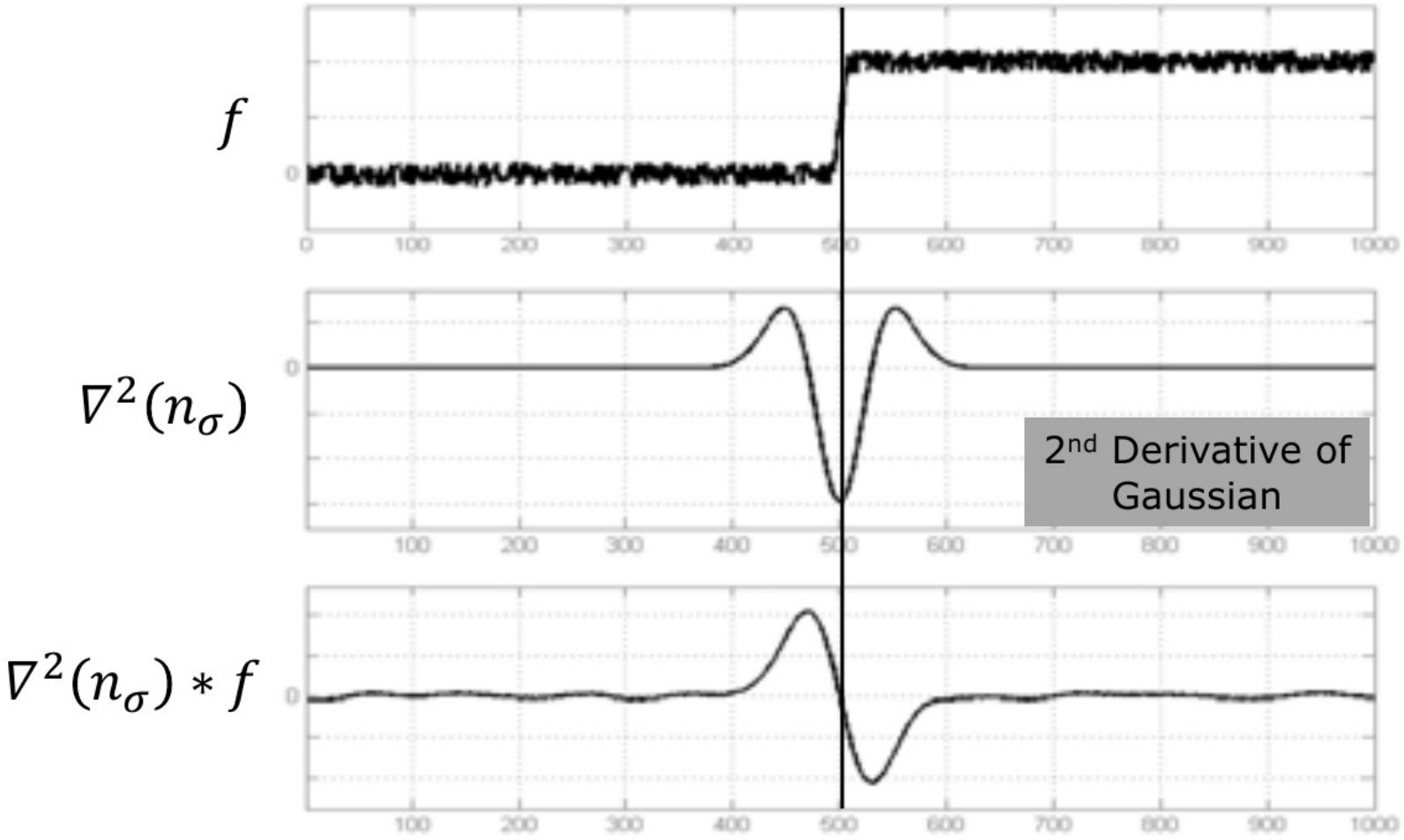
Gaussian Filter is used for removing noise by smoothing

Review: Derivative of Gaussian



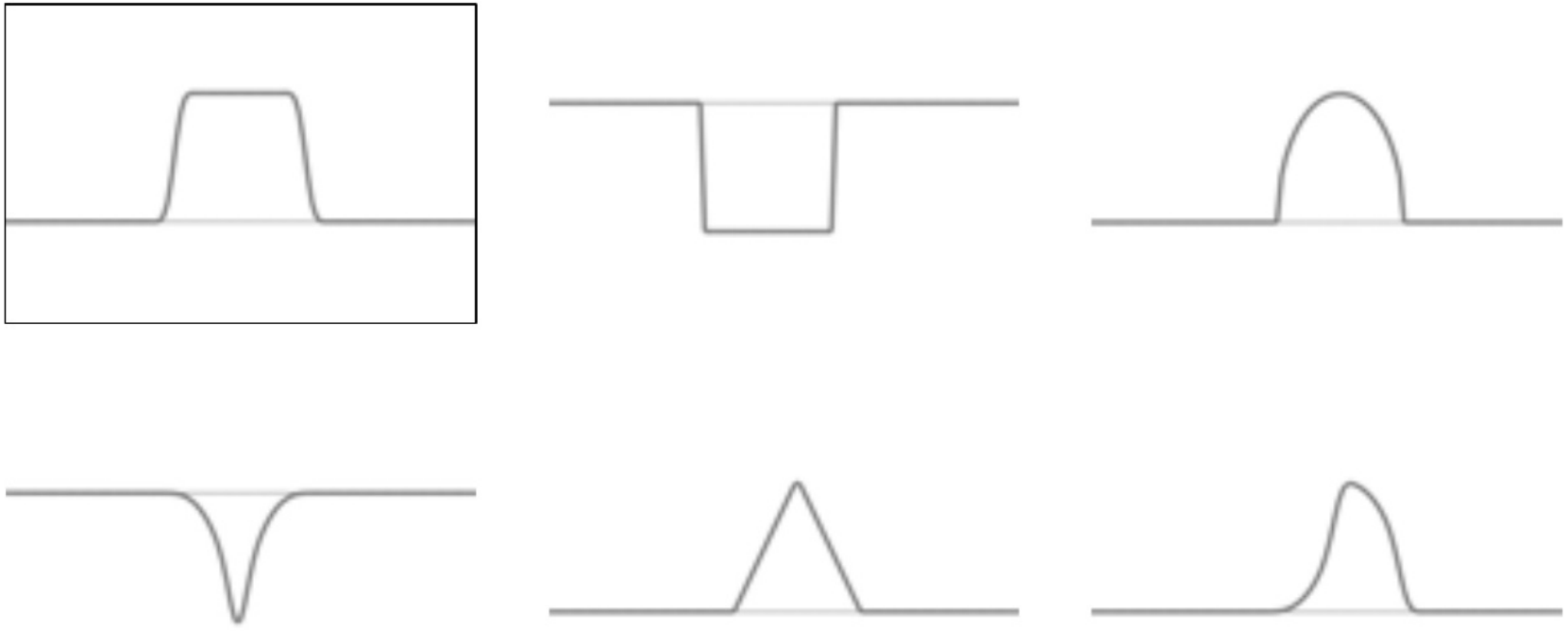
Extremum of Derivative of Gaussian denotes an Edge

Review: 2nd Derivative of Gaussian



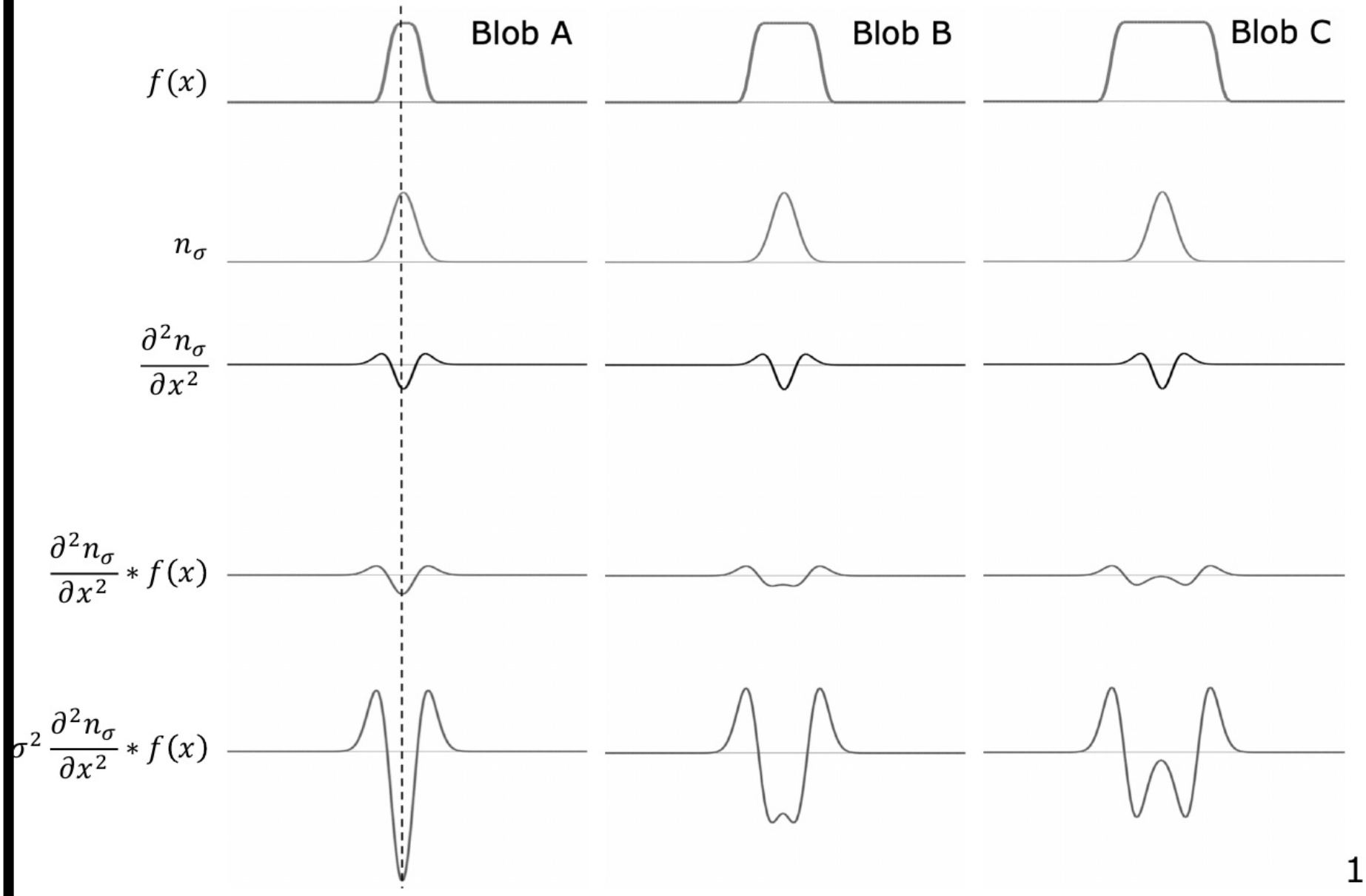
Zero Crossing in 2nd Derivative of Gaussian denote an Edge

1D Blobs

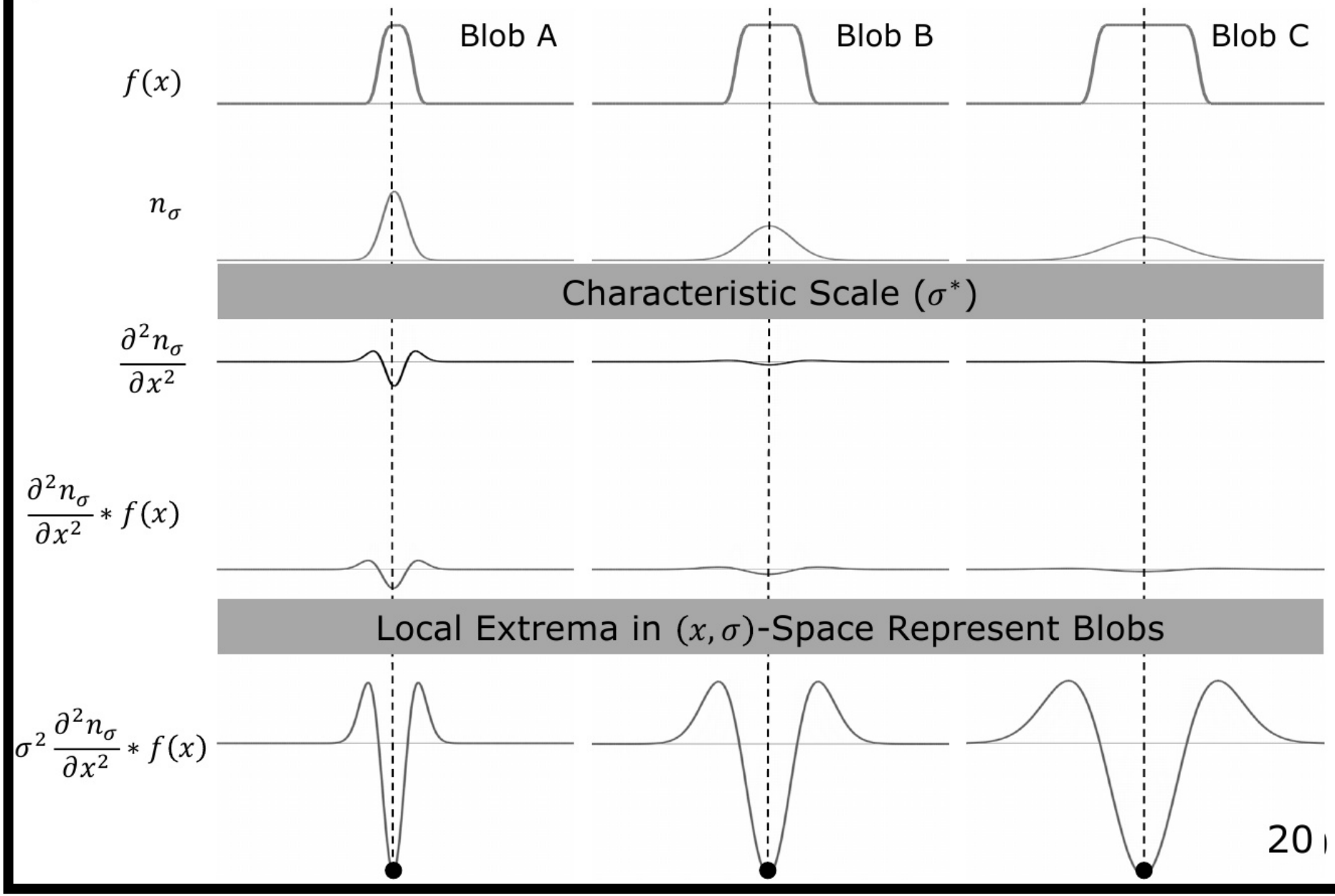


Examples of 1D Blob-like structures

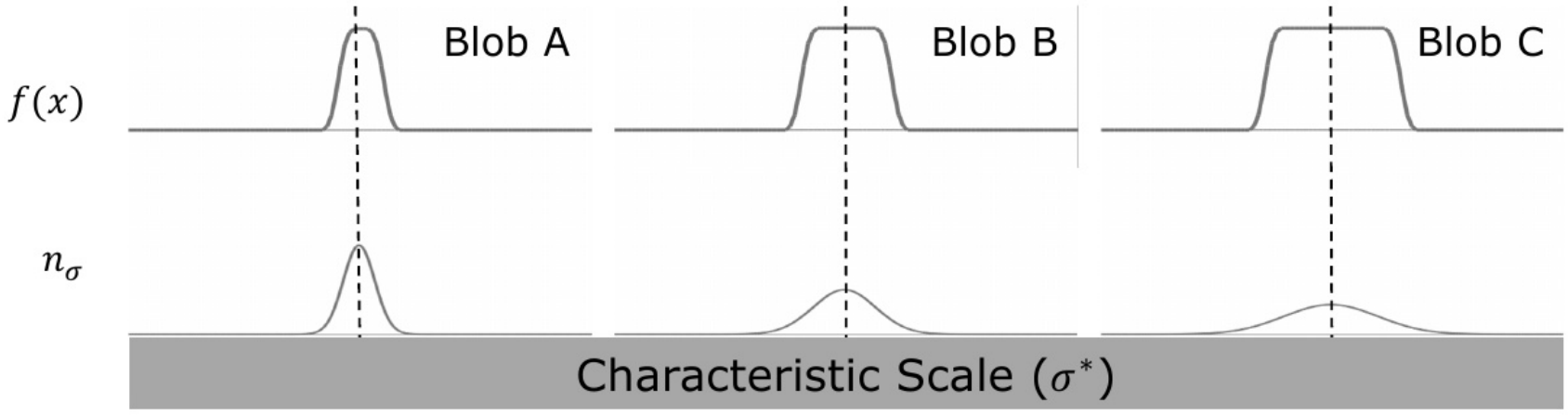
1D Blob and 2nd Derivative of Gaussian



1D Blob and 2nd Derivative of Gaussian



Characteristic Scale and Blob Size



$\sigma_A{}^* = \sigma_1$ $\sigma_B{}^* = 2\sigma_1$ $\sigma_C{}^* = 3\sigma_1$

Characteristic Scale: The σ at which σ-normalized 2nd derivative attains its extreme value.

Characteristic Scale $\propto$ Size of Blob

$$\frac{\text{Size of Blob A}}{\text{Size of Blob B}} = \frac{\sigma_A{}^*}{\sigma_B{}^*} \; ; \quad \frac{\text{Size of Blob B}}{\text{Size of Blob C}} = \frac{\sigma_B{}^*}{\sigma_C{}^*}$$

1D Blob Detection Summary

Given: 1D signal $f(x)$

Compute: $\sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} * f(x)$ at many scales $(\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_k)$.

Find: $$(x^*, \sigma^*) = \arg\max_{(x,\sigma)} \left| \sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} * f(x) \right|$$

x^*: Blob Position

σ^*: Characteristic Scale (Blob Size)

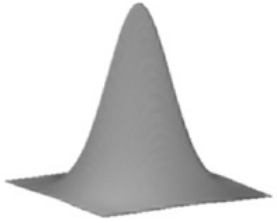
2D Blob Detector

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.

Scale-Space






...

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$

Increasing σ, Higher Scale, Lower Resolution

Scale Space: Stack created by filtering an image with Gaussians of different sigma (σ)

$$S(x, y, \sigma) = n(x, y, \sigma) * I(x, y)$$

Creating Scale-Space






...

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$

Increasing σ, Higher Scale, Lower Resolution

Selecting sigmas to generate the scale-space:

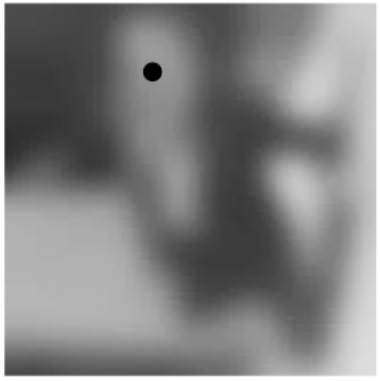
$$\boxed{\sigma_k = \sigma_0 s^k} \quad k = 0,1,2,3, \ldots$$

s: Constant multiplier

σ_0: Initial Scale

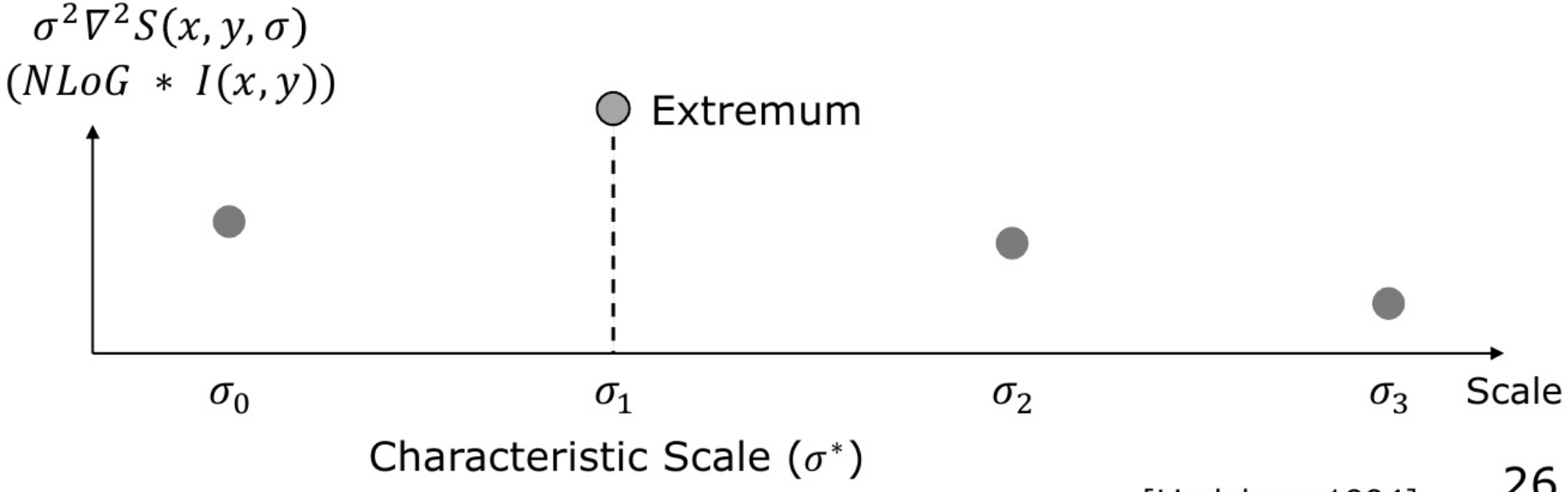
Blob Detection using Local Extrema



...

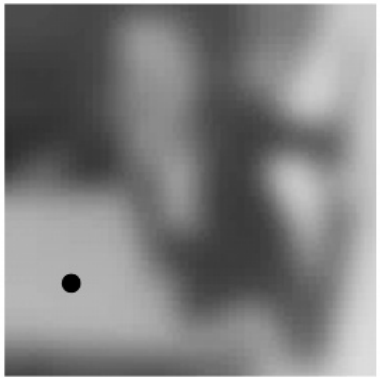
$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$



[Lindeberg 1994]

Blob Detection using Local Extrema

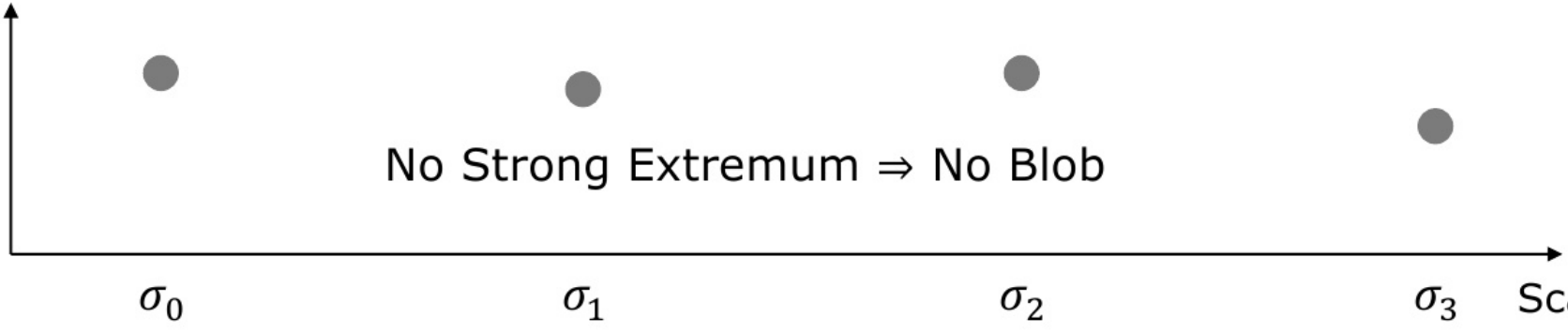


...

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$

$\sigma^2 \nabla^2 S(x, y, \sigma)$
$(NLoG * I(x, y))$



[Lindeberg 1994]

2D Blob Detection Summary

Given an image $I(x, y)$

Convolve the image using NLoG at many scales σ

Find:

$$(x^*, y^*, \sigma^*) = \arg\max_{(x,y,\sigma)} |\sigma^2 \nabla^2 n_\sigma * I(x, y)|$$

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

σ^*: Size of the blob

SIFT Detector

Shree K. Nayar

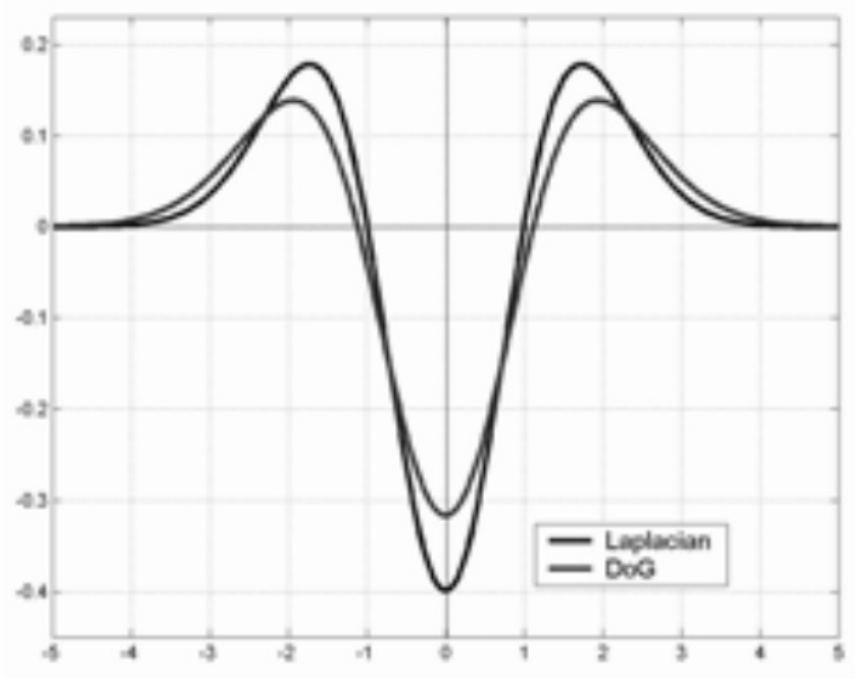
Columbia University

Topic: SIFT Detector, Module: Features

First Principles of Computer Vision

Fast NLoG Approximation: DoG

Difference of Gaussian (DoG) $= (n_{s\sigma} - n_{\sigma}) \approx (s-1)\underbrace{\sigma^2 \nabla^2 n_{\sigma}}_{\text{NLoG}}$



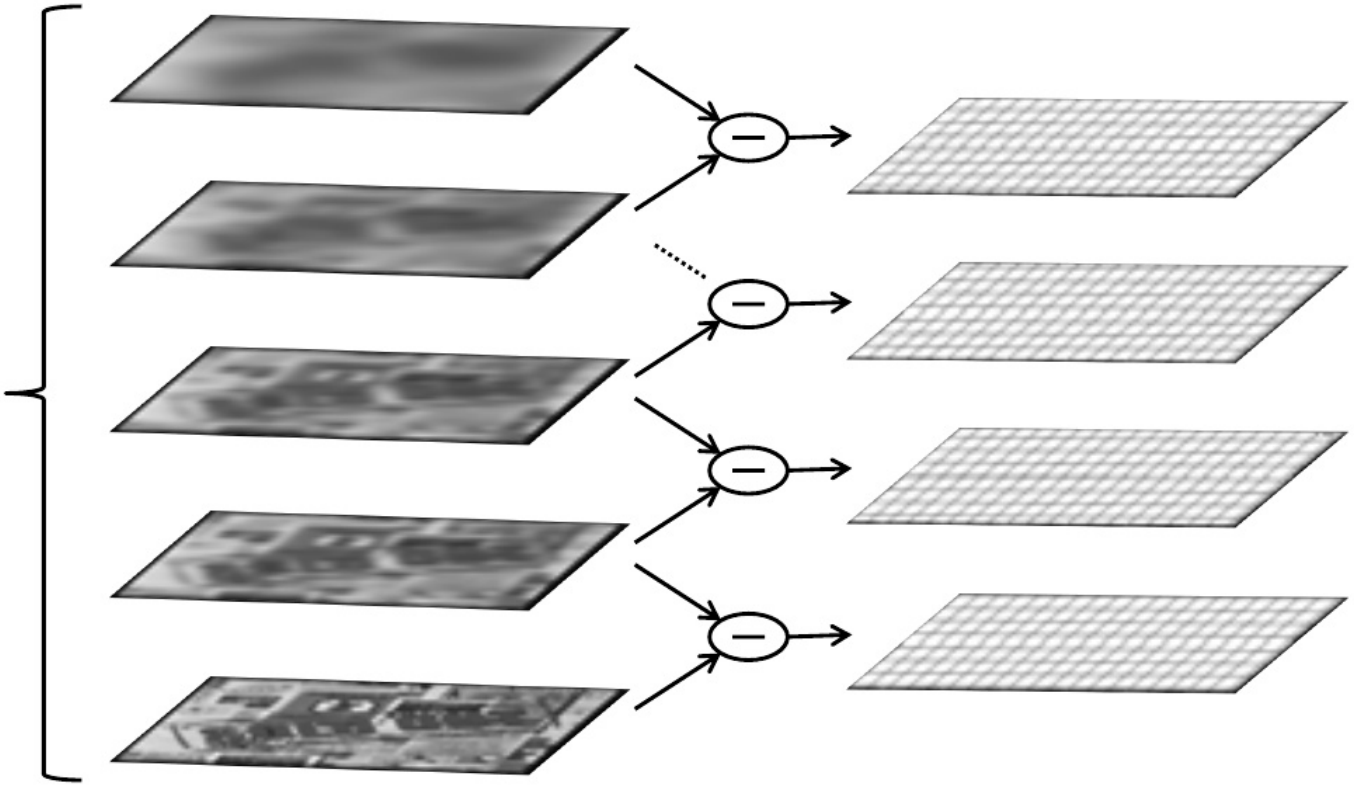
$$\boxed{\text{DoG} \approx (s-1)\ \text{NLoG}}$$

Extracting SIFT Interest Points



Image

$I(x, y)$



Gaussian

Scale-Space

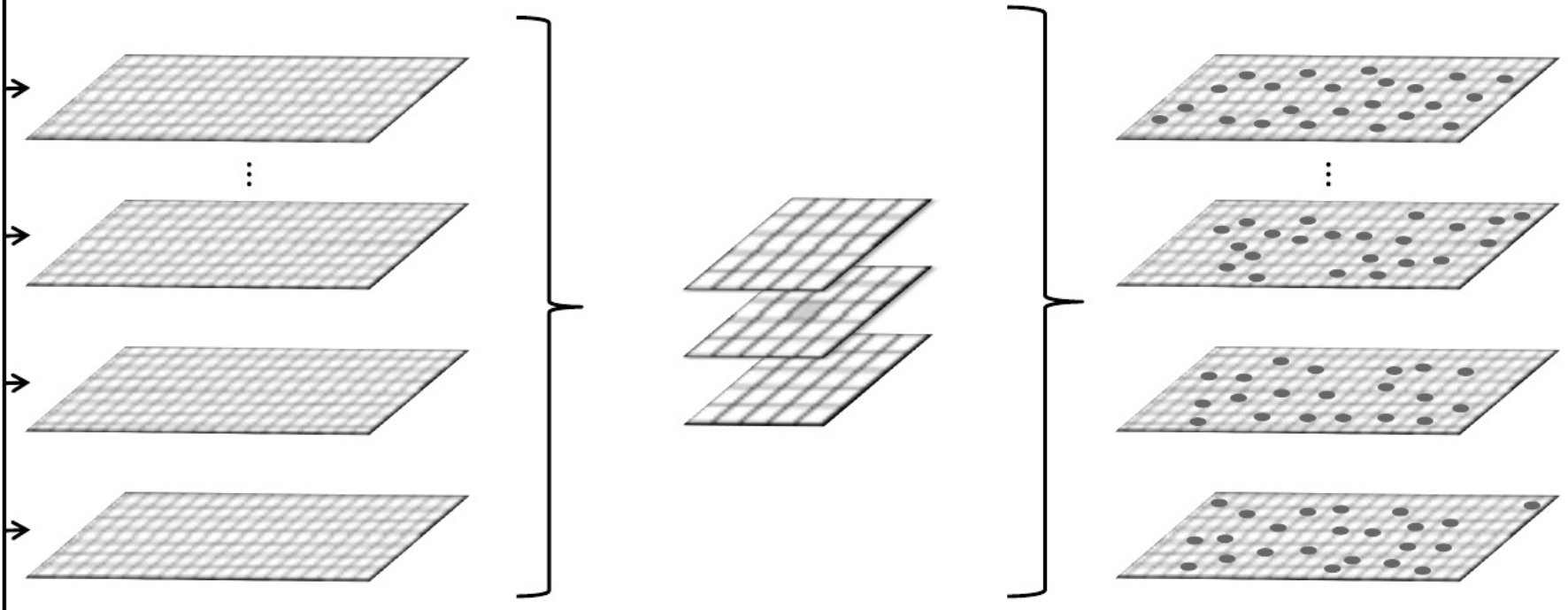
$S(x, y, \sigma)$

Difference of Gaussians (DoG)

$\approx (s-1)\sigma^2 \nabla^2 S(x, y, \sigma)$

[Lowe 2004]

Extracting SIFT Interest Points



Difference of Gaussians (DoG)

$\approx (s-1)\sigma^2\nabla^2 S(x,y,\sigma)$

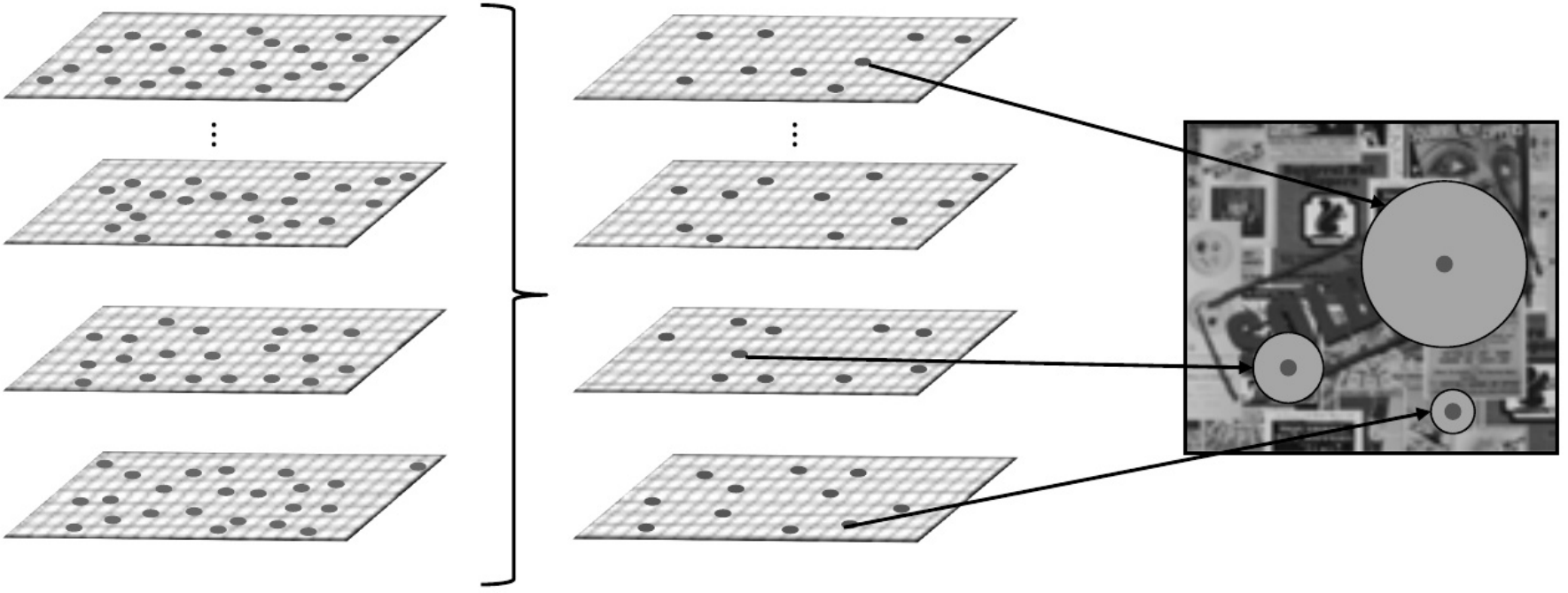
Find Extremum in every 3x3x3 grid

Interest Point Candidates

(includes weak extrema)

[Lowe 2004]

Extracting SIFT Interest Points



Interest Point Candidates

(includes weak extrema)

SIFT Interest Points

(after removing weak extrema)

[Lowe 2004]

SIFT Detection Examples



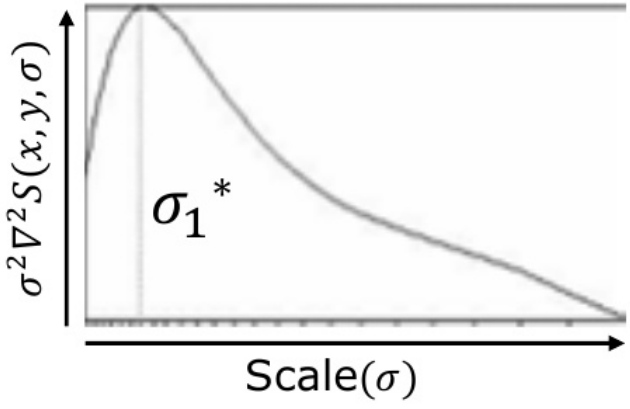
SIFT Detection Examples

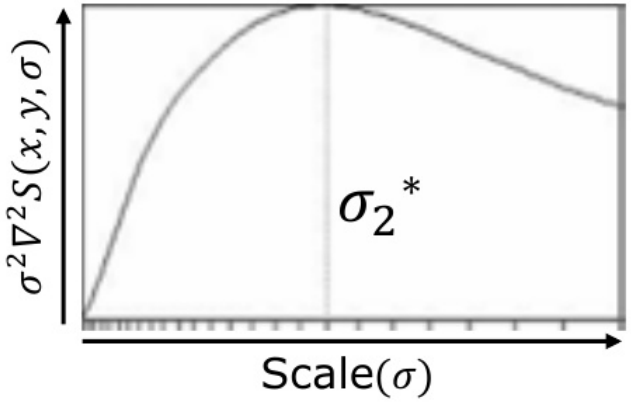


SIFT Scale Invariance









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

[Mikolajczyk 2002]

Computing the Principal Orientation

Use the histogram of gradient directions

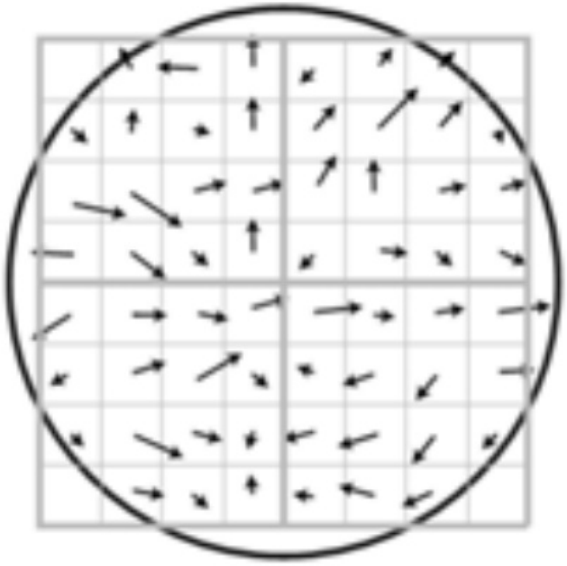
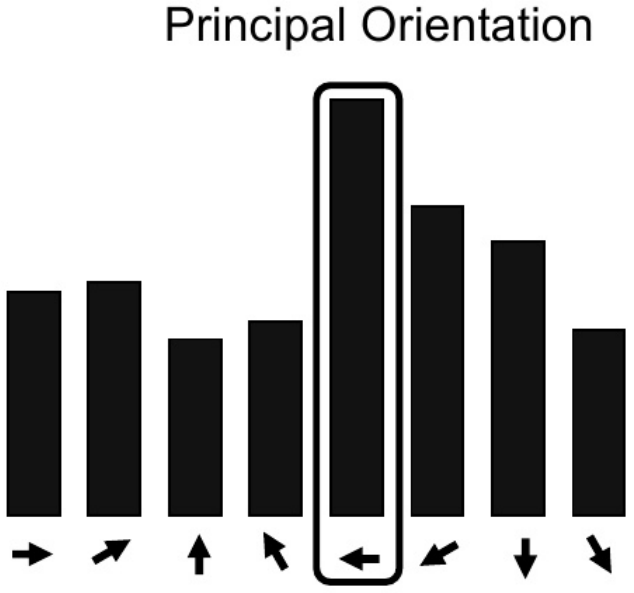


Image gradient directions

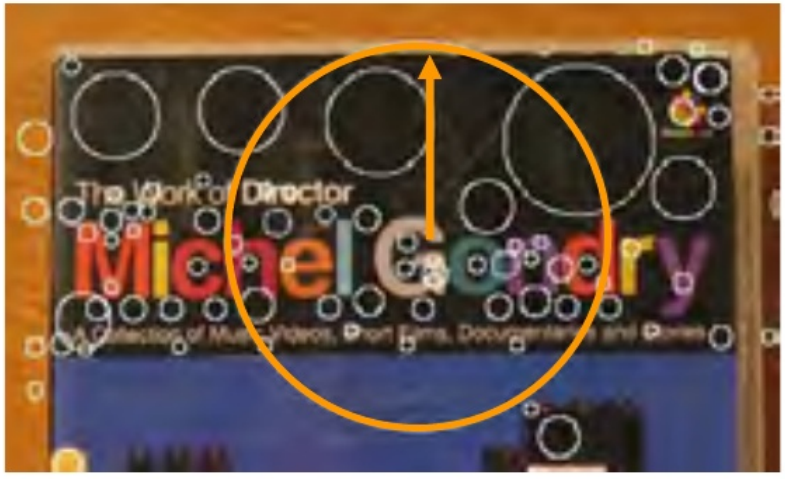
$$\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$$

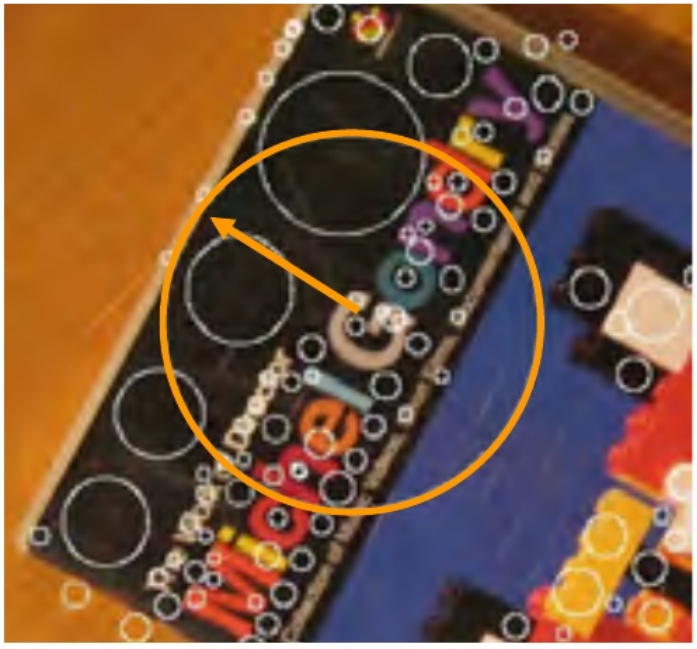


Choose the most prominent gradient direction

SIFT Rotation Invariance

Use the principal orientation to undo rotation





SIFT Descriptor

Shree K. Nayar

Columbia University

Topic: SIFT Detector, Module: Features

First Principles of Computer Vision

SIFT Descriptor

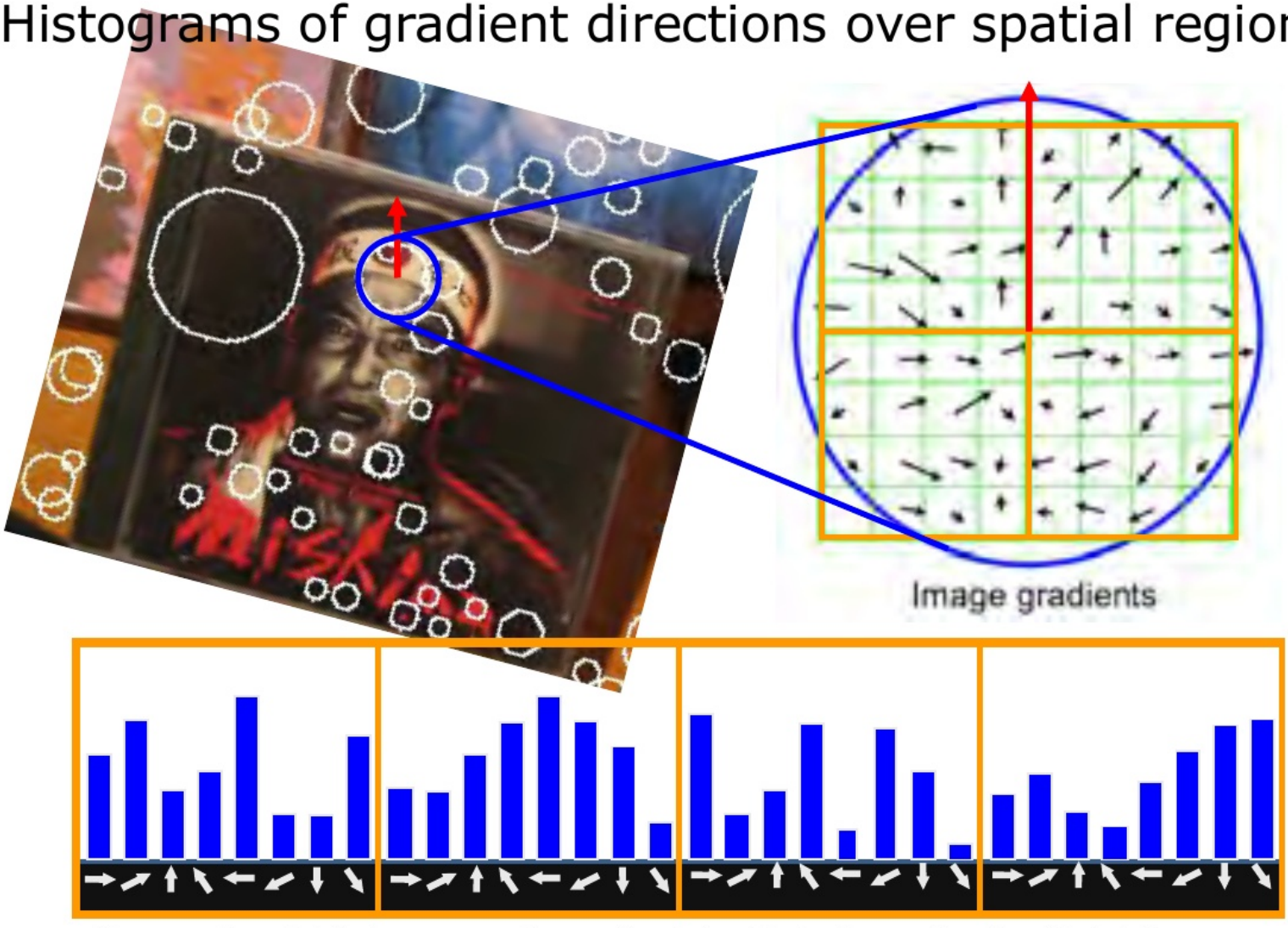
Histograms of gradient directions over spatial regions



[Lowe 2004]

SIFT Descriptor

Histograms of gradient directions over spatial regions



Normalized Histogram: Invariant to Rotation, Scale, Brightness

Comparing SIFT Descriptors

Essentially comparing two arrays of data.

Let $H_1(k)$ and $H_2(k)$ be two arrays of data of length N.

L2 Distance:

$$d(H_1, H_2) = \sqrt{\sum_k \left(H_1(k) - H_2(k)\right)^2}$$

Smaller the distance metric, better the match.

Perfect match when $d(H_1, H_2) = 0$

Comparing SIFT Descriptors

Essentially comparing two arrays of data.

Let $H_1(k)$ and $H_2(k)$ be two arrays of data of length N.

Normalized Correlation:

$$d(H_1, H_2) = \frac{\sum_k [(H_1(k) - \bar{H}_1)(H_2(k) - \bar{H}_2)]}{\sqrt{\sum_k (H_1(k) - \bar{H}_1)^2} \sqrt{\sum_k (H_2(k) - \bar{H}_2)^2}}$$

where: $\bar{H}_i = \frac{1}{N} \sum_{k=1}^{N} H_i(k)$

Larger the distance metric, better the match.

Perfect match when $d(H_1, H_2) = 1$

Comparing SIFT Descriptors

Essentially comparing two arrays of data.

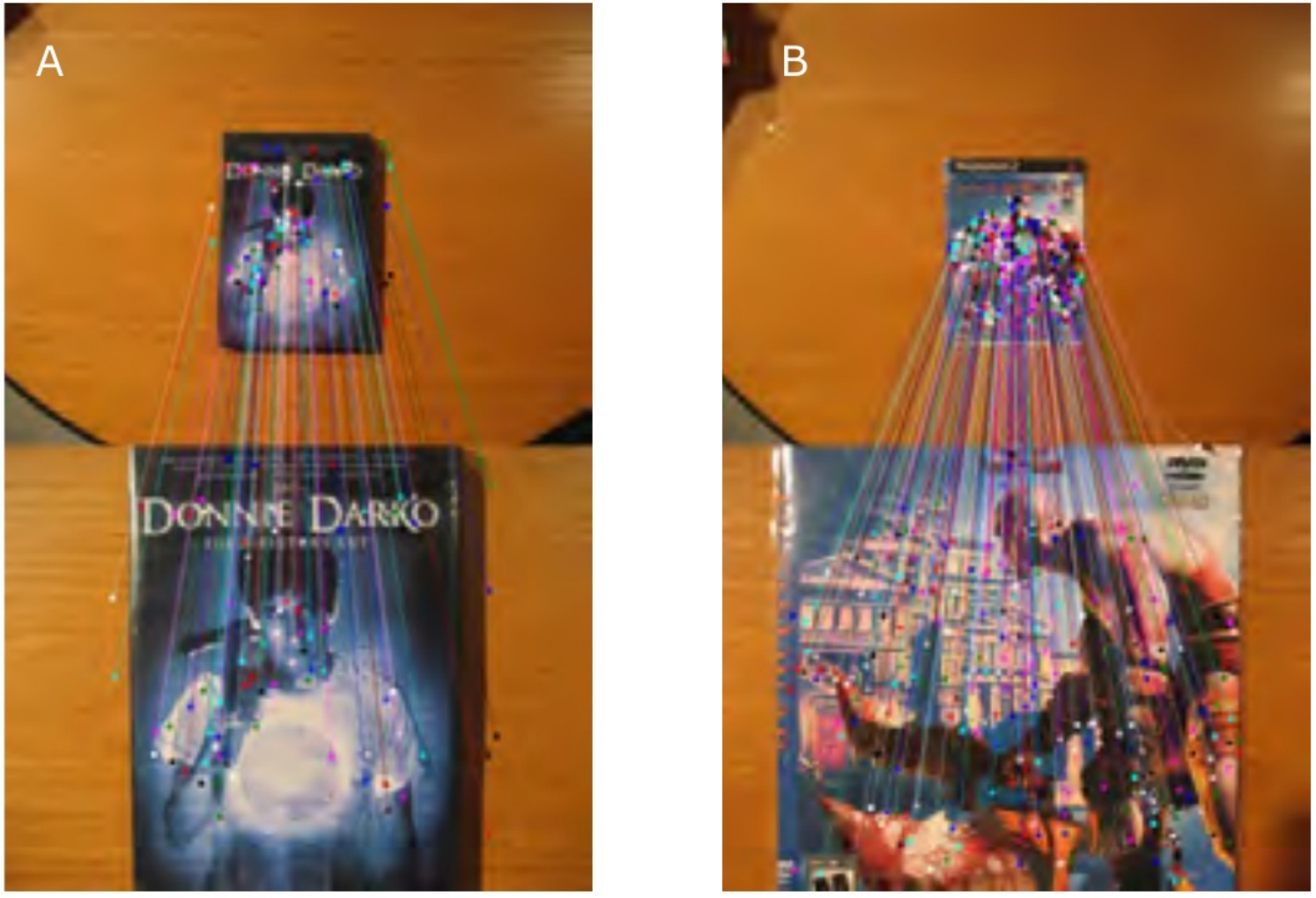
Let $H_1(k)$ and $H_2(k)$ be two arrays of data of length N.

Intersection:

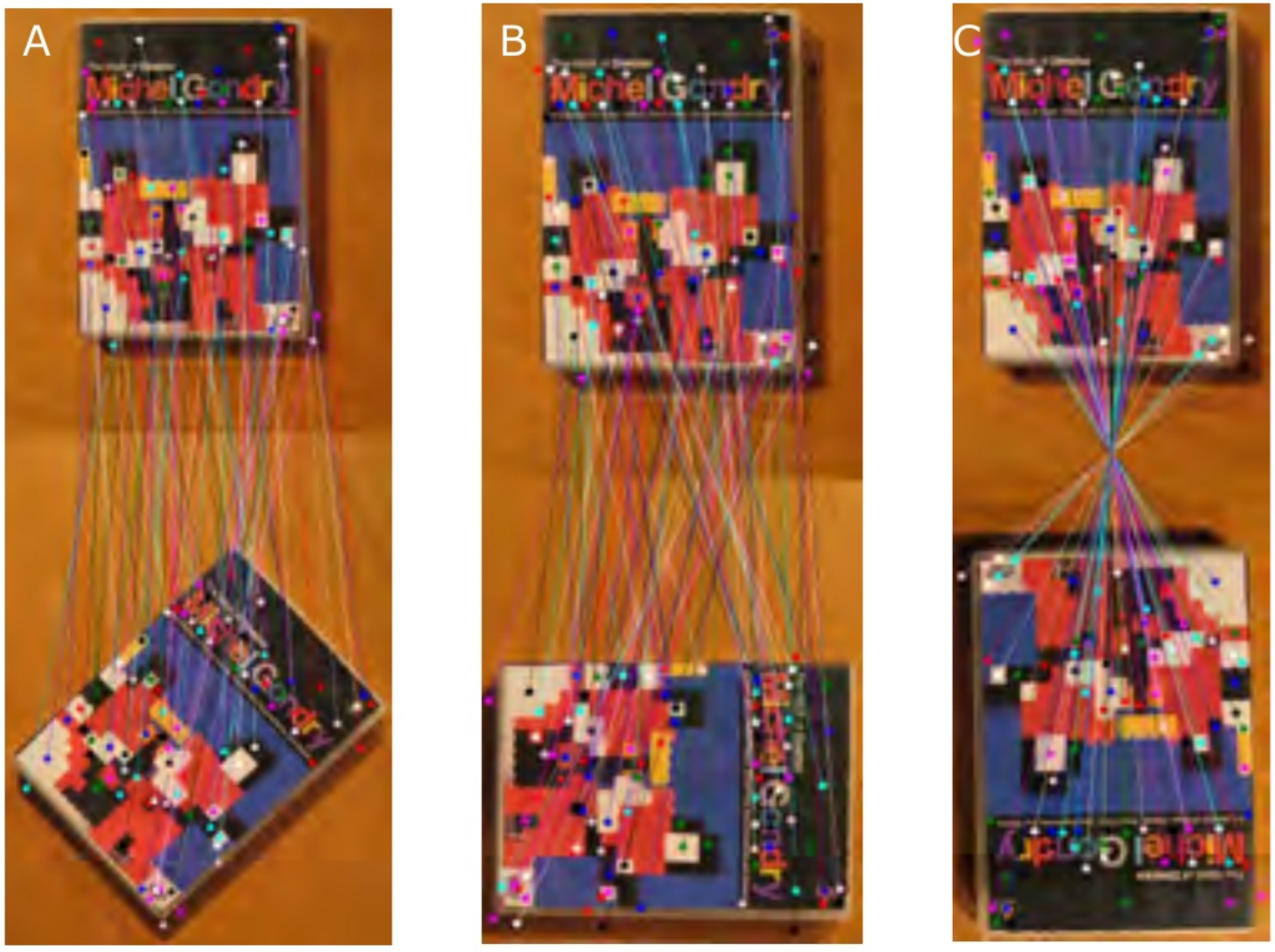
$$d(H_1, H_2) = \sum_k \min\big(H_1(k), H_2(k)\big)$$

Larger the distance metric, better the match.

SIFT Results: Scale Invariance



SIFT Results: Rotation Invariance



SIFT Results: Robustness to Clutter







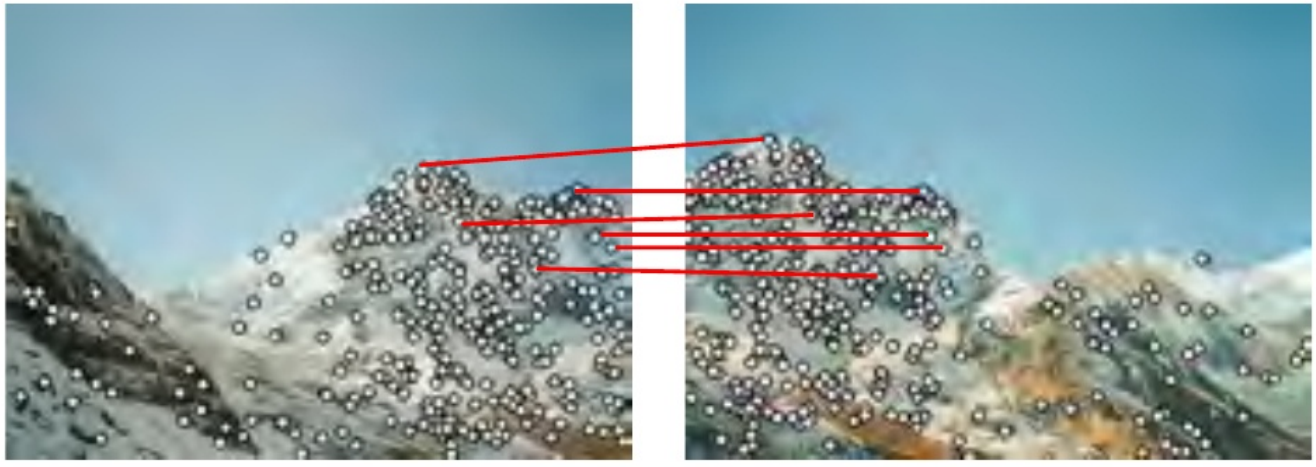
Panorama Stitching using SIFT



Image 1



Image 2



Match SIFT Interest Points

[Autostitch]

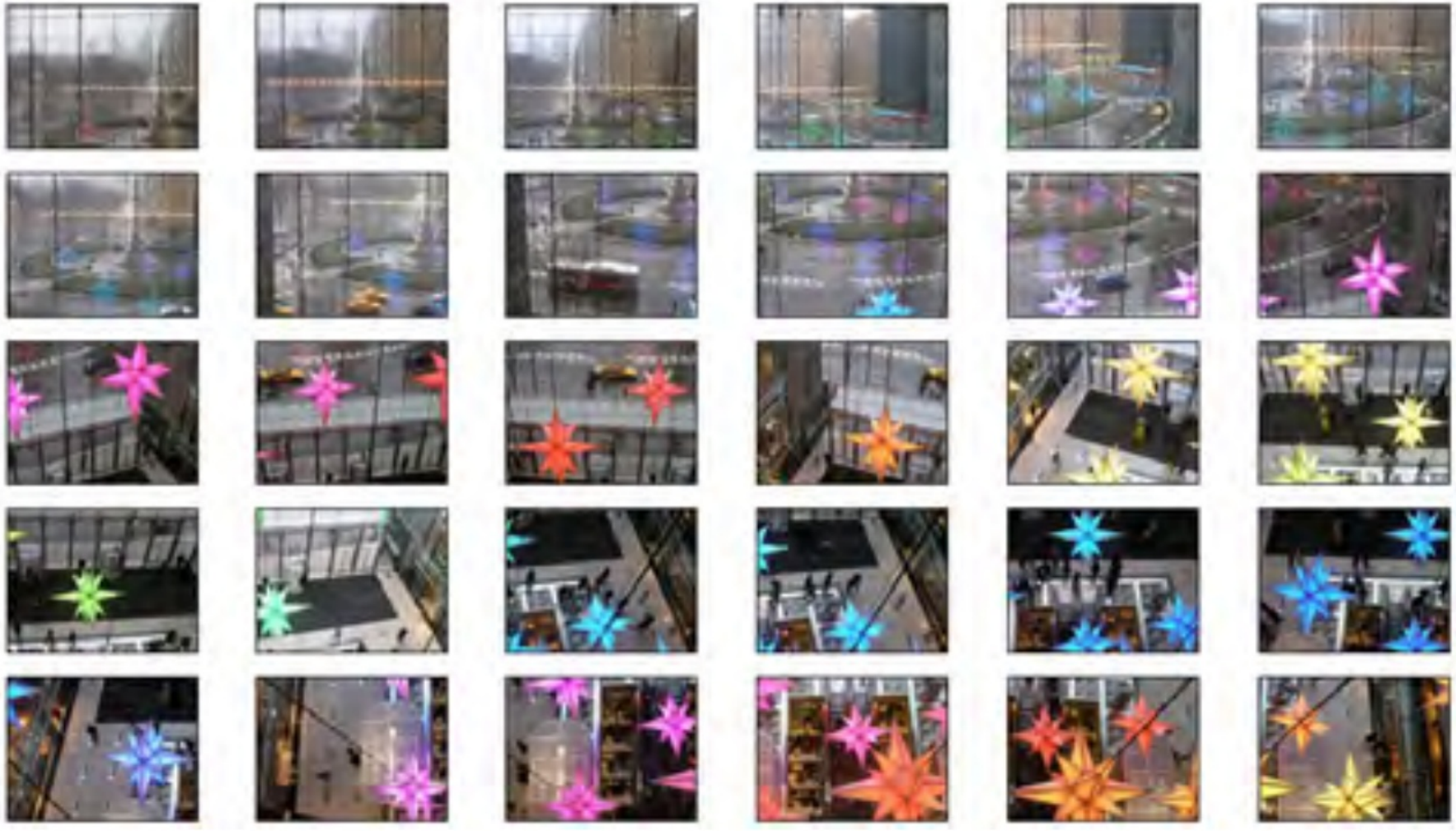
Panorama Stitching using SIFT



Warp and combine images to create a larger image

[Autopano]

Auto Collage using SIFT



Captured Photos

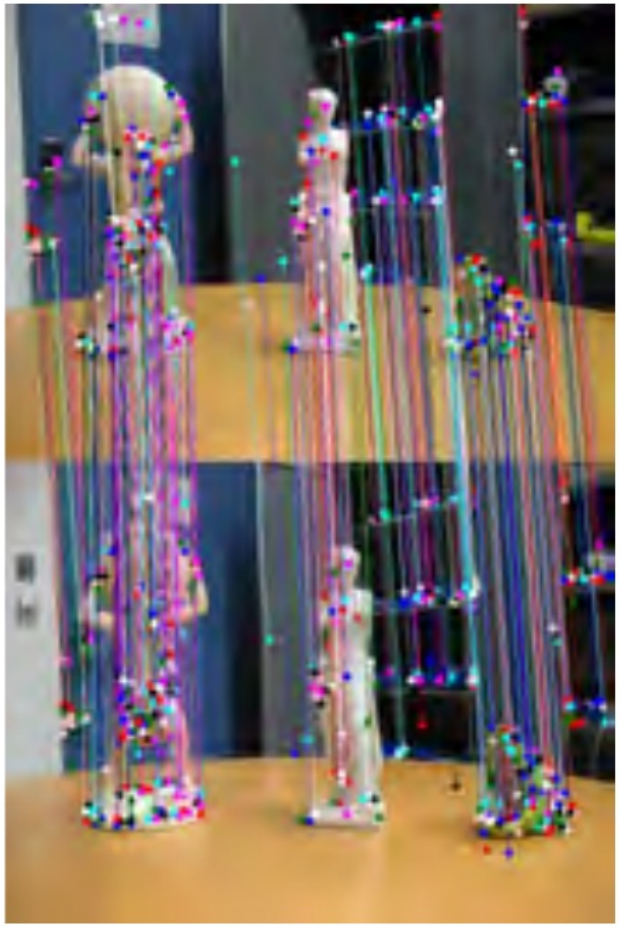
[Nomura 2007]

Auto Collage using SIFT



[Nomura 2007]

SIFT for 3D Objects?



No Change in Viewpoint



30° Change in Viewpoint



90° Change in Viewpoint

SIFT is reliable for only small changes in viewpoint

References and Credits

Shree K. Nayar

Columbia University

Topic: SIFT Detector, Module: Features

First Principles of Computer Vision

References: Textbooks

Computer Vision: Algorithms and Applications
Szeliski, R., Springer

References: Papers

[Autopano] Software to make panaromas using SIFT. http://user.cs.tu-berlin.de/~nowozin/autopano-sift/

[Brown and Lowe 2002] M. Brown and D. Lowe. "Invariant Features from Interest Point Groups". *BMVC*, 2002.

[Harris and Stephens 1988] C. Harris and M. Stephens. "A Combined Corner and Edge Detector". *4th Alvey Vision Conference*, 1988.

[Lowe 2004] D. Lowe."Distinctive Image Features from Scale-Invariant Keypoints". *IJCV*, 2004.

[Lindeberg 1994] T. Lindeberg. "Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales." *J. of Applied Statistics,* 1994.

[Matas 2002] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *BMVC*, 2002.

[Mikolajczyk 2002] K. Mikolajczyk. "Detection of Local Features Invariant to Affine Transformations." *Ph.D. Thesis*, 2002.

References: Papers

[Mikolajczyk 2004] K. Mikolajczyk and C. Schmid. "Scale and Affine Invariant Interest Point Detectors." *IJCV,* 2004.

[Mikolajczyk 2005] K. Mikolajczyk and C. Schmid. "A Performance Evaluation of Local Descriptors." *PAMI,* 2005.

[Nomura 2007] Y. Nomura, L. Zhang and S.K. Nayar. "Scene Collages and Flexible Camera Arrays." *EGSR,* 2007.

[SIFT] SIFT Binaries. http://www.cs.ubc.ca/~lowe/keypoints/

[Witkin 1983] A. Witkin. "Scale-Space Filtering". *IJCAI*, 1983.

Image Stitching

Shree K. Nayar

Columbia University

Topic: Image Stitching, Module: Features

First Principles of Computer Vision

Image Stitching



Image 1



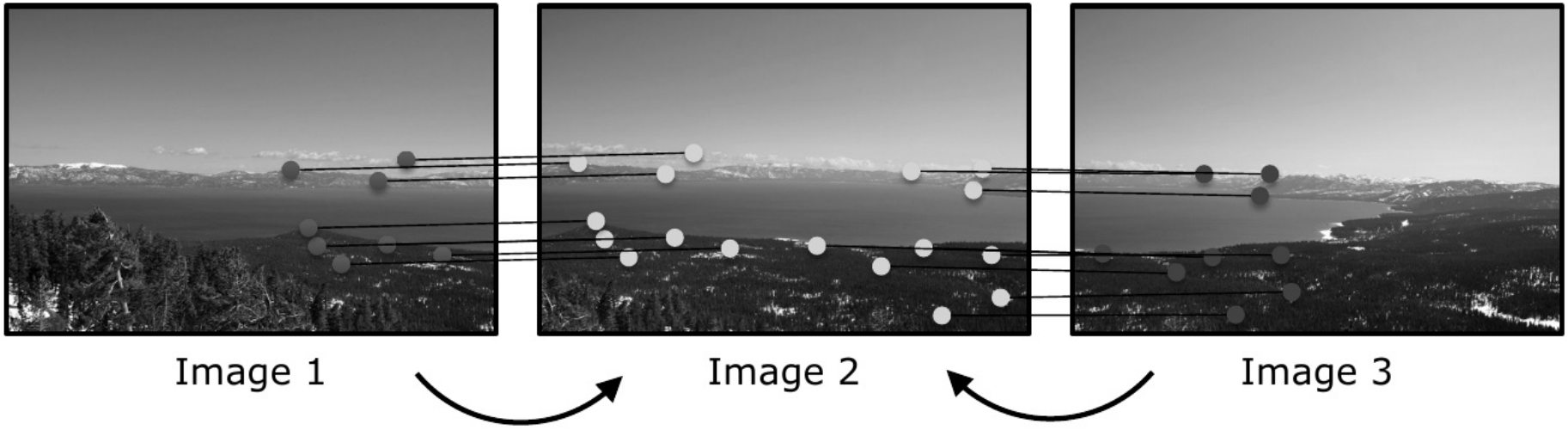
Image 2



Image 3

How would you align these images?

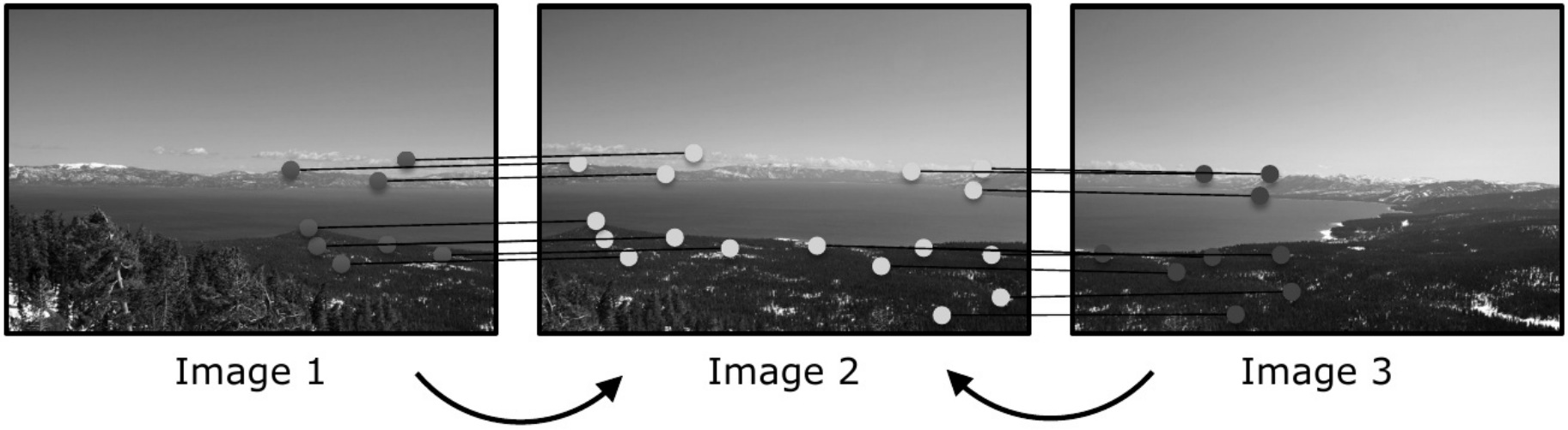
Image Stitching



Find corresponding points (using SIFT, etc.)

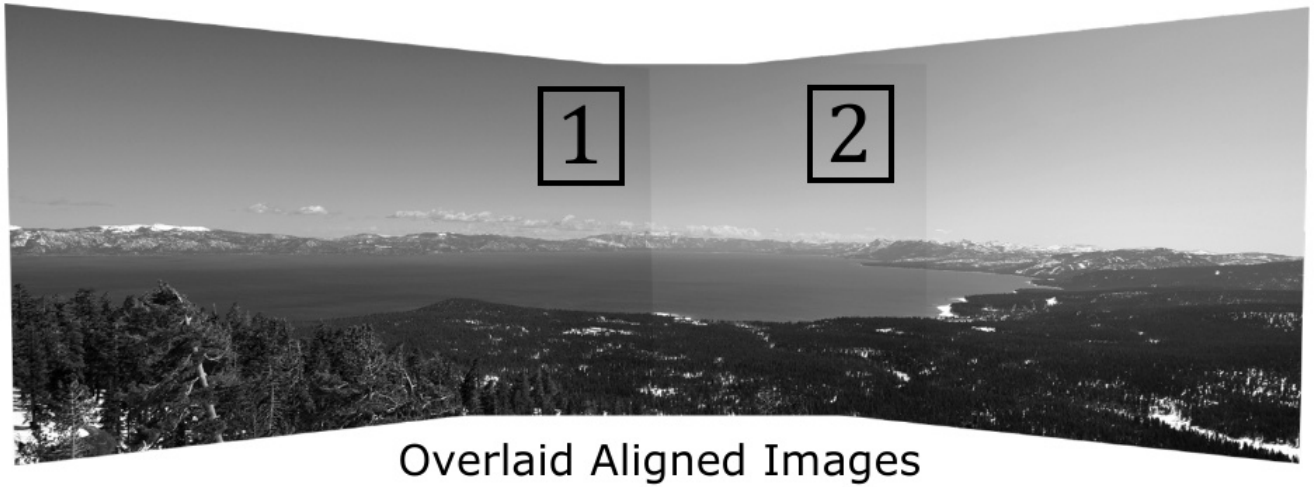
Find geometric relationship between the images

Image Stitching



Warp images so that corresponding points align

Image Stitching



Overlaid Aligned Images



Blended Images

Blend images to remove hard seams

Image Stitching

Combine multiple photos to create a larger photo

Topics:

(1) 2x2 Image Transformations

(2) 3x3 Image Transformations

(3) Computing Homography

(4) Dealing with Outliers: RANSAC

(5) Warping and Blending Images

2x2 Image Transformations

Shree K. Nayar

Columbia University

Topic: Image Stitching, Module: Features

First Principles of Computer Vision

Image Manipulation

Image Filtering: Change range (brightness)

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

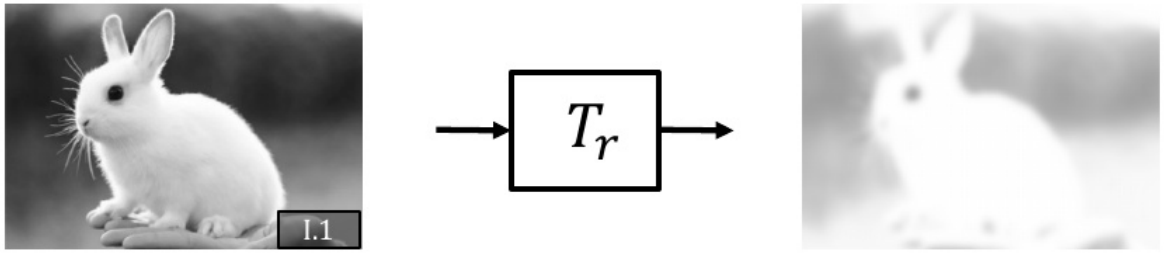
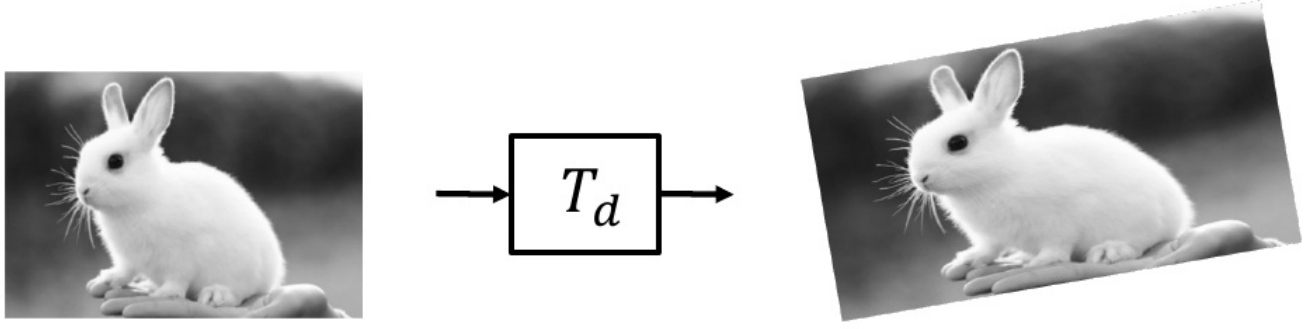


Image Warping: Change domain (location)

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

Transformation T_d is a coordinate changing operator



Global Warping/Transformation



Translation



Rotation

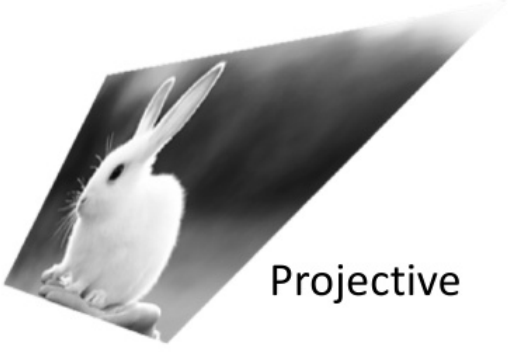


Scaling and Aspect

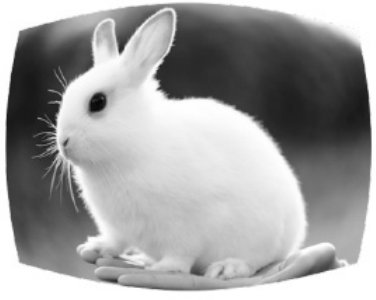
$$g(x,y) = f\big(T(x,y)\big)$$



Affine



Projective



Barrel

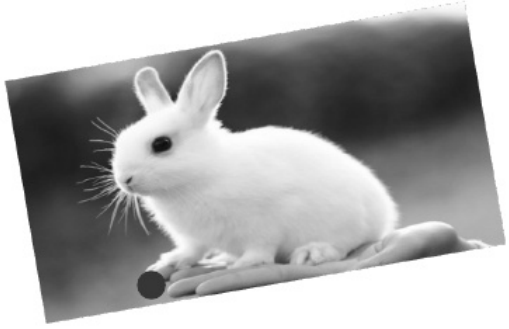
Transformation T is the same over entire domain

Often can be described by just a few parameters

2x2 Linear Transformations



T

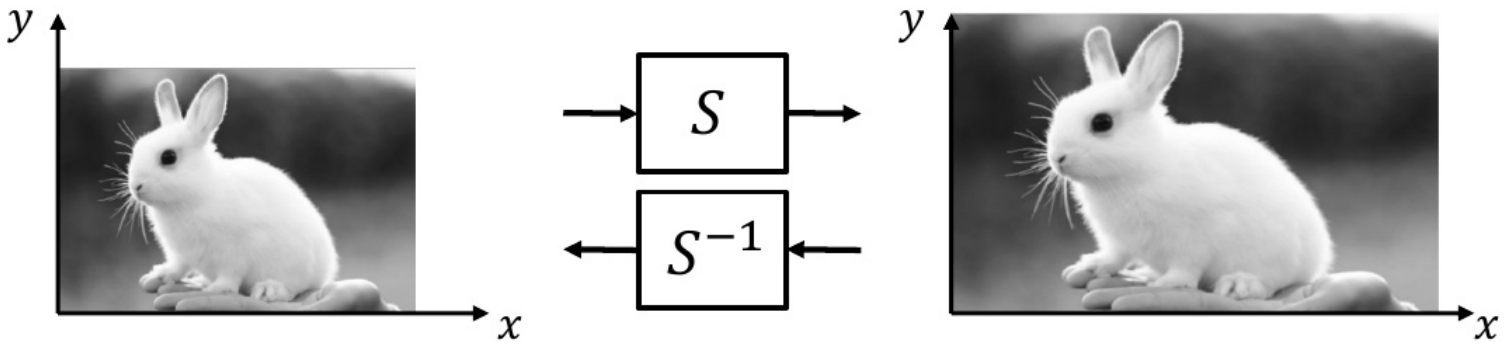


$\mathbf{p}_1 = (x_1, y_1)$ $\mathbf{p}_2 = (x_2, y_2)$

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:

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

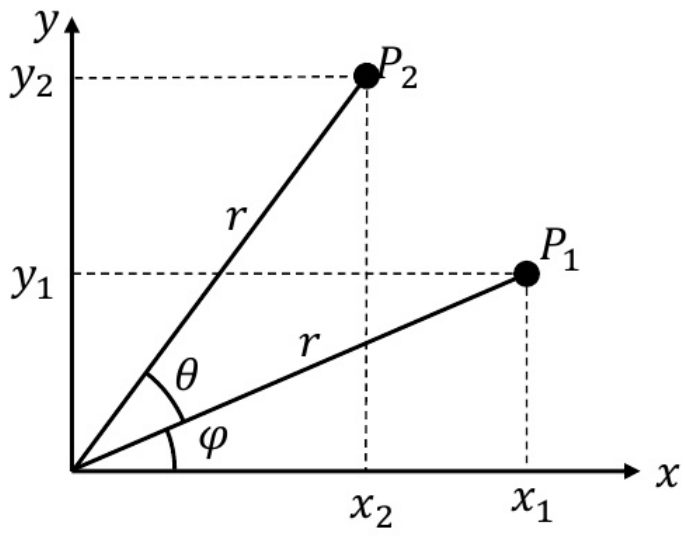
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

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

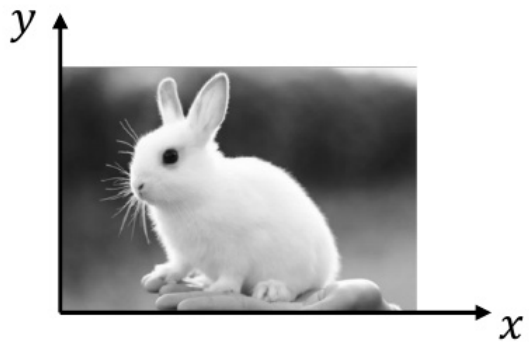


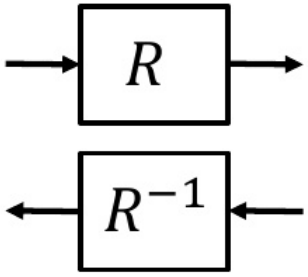
$$x_1 = r\cos(\varphi)$$
$$y_1 = r\sin(\varphi)$$

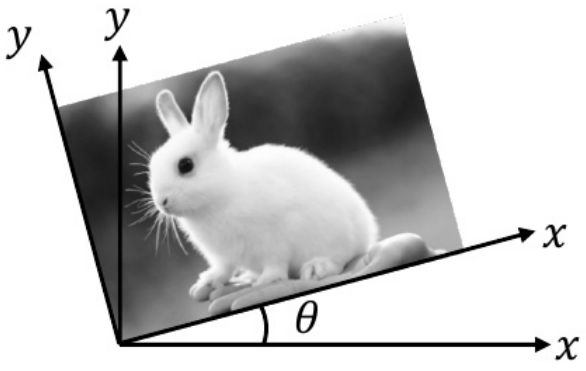
$$x_2 = r\cos(\varphi + \theta)$$
$$x_2 = r\cos\varphi\cos\theta - r\sin\varphi\sin\theta$$
$$x_2 = x_1\cos\theta - y_1\sin\theta$$

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

Rotation







Forward:

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

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

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

1

Inverse:

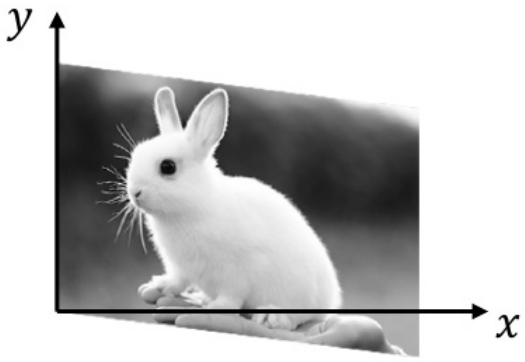
$$x_1 = x_2 cos\theta + y_2 sin\theta$$

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

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

Skew





Horizontal Skew:

$$x_2 = x_1 + m_x y_1$$

$$y_2 = y_1$$

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

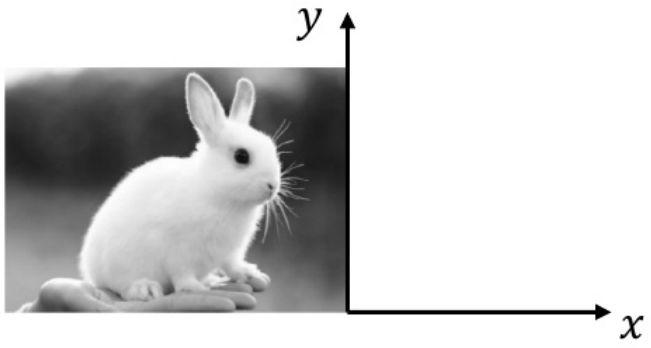
Vertical Skew:

$$x_2 = x_1$$

$$y_2 = m_y x_1 + y_1$$

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

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

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

3x3 Image Transformations

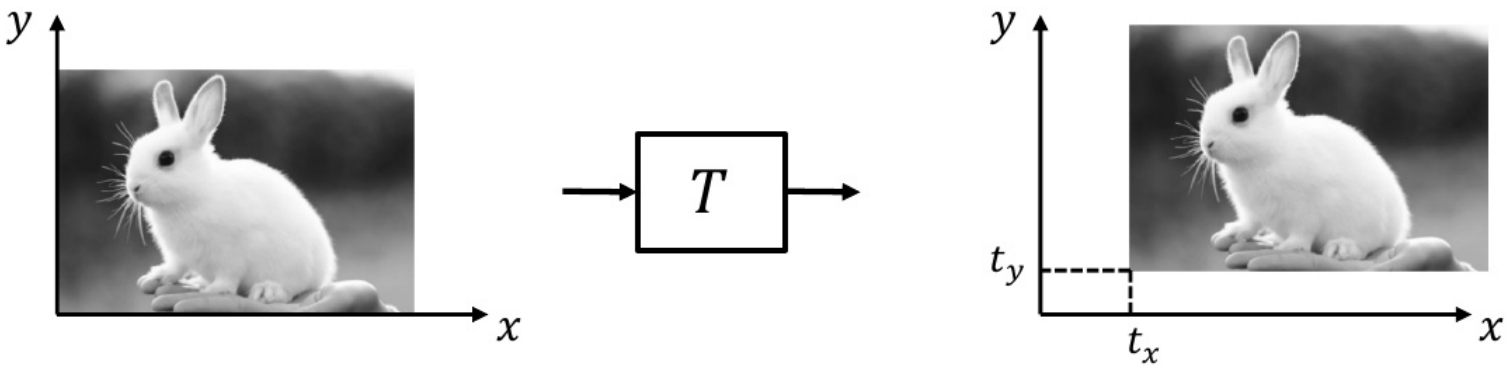
Shree K. Nayar

Columbia University

Topic: Image Stitching, Module: Features

First Principles of Computer Vision

Translation



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

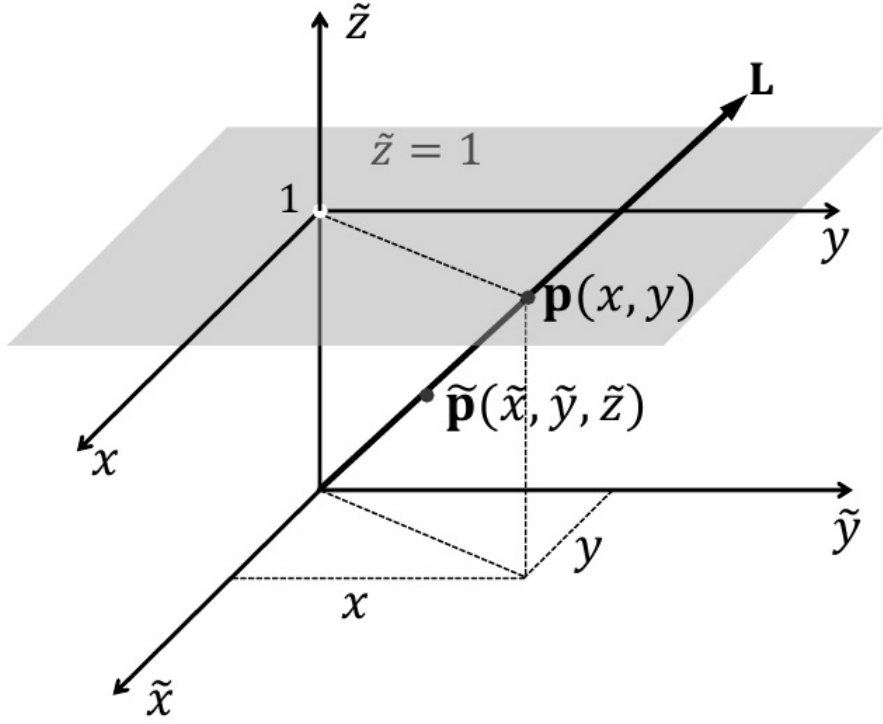
Can translation be expressed as a 2x2 matrix? No.

Homogenous Coordinates

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

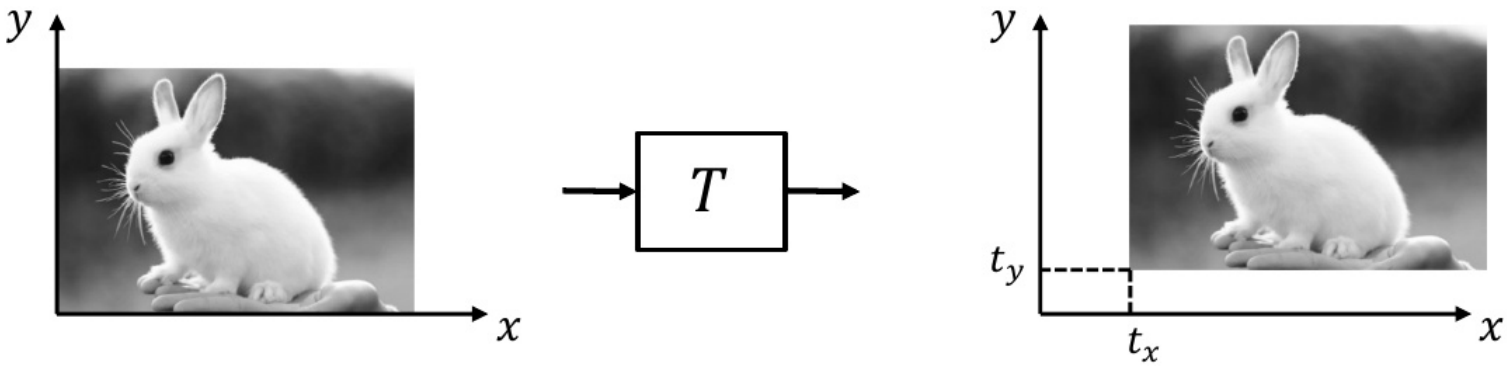
$$x = \frac{\tilde{x}}{\tilde{z}} \qquad y = \frac{\tilde{y}}{\tilde{z}}$$

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



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

Translation



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

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

$$\boxed{1}$$

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

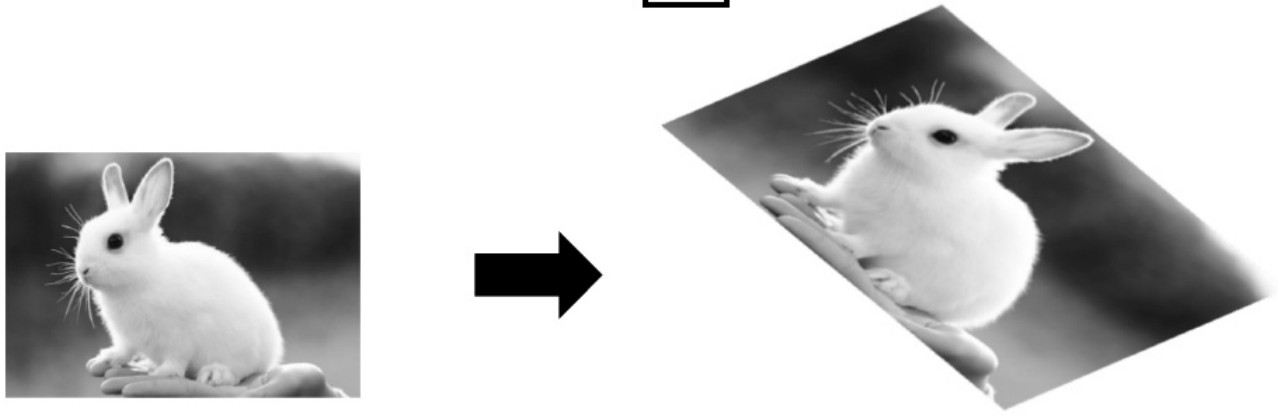
Composition of these transformations?

Affine Transformation

Any transformation of the form:

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

$$\boxed{1}$$



Affine Transformation

Any transformation of the form:

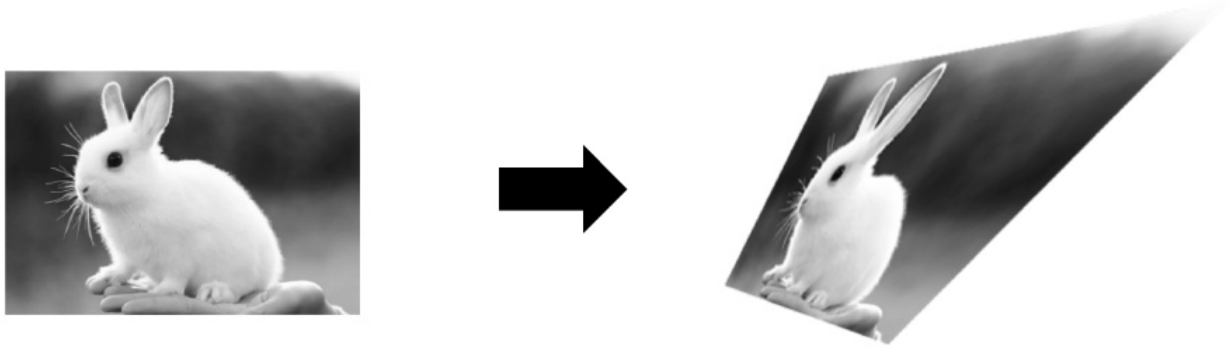
$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix}$$

- 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:

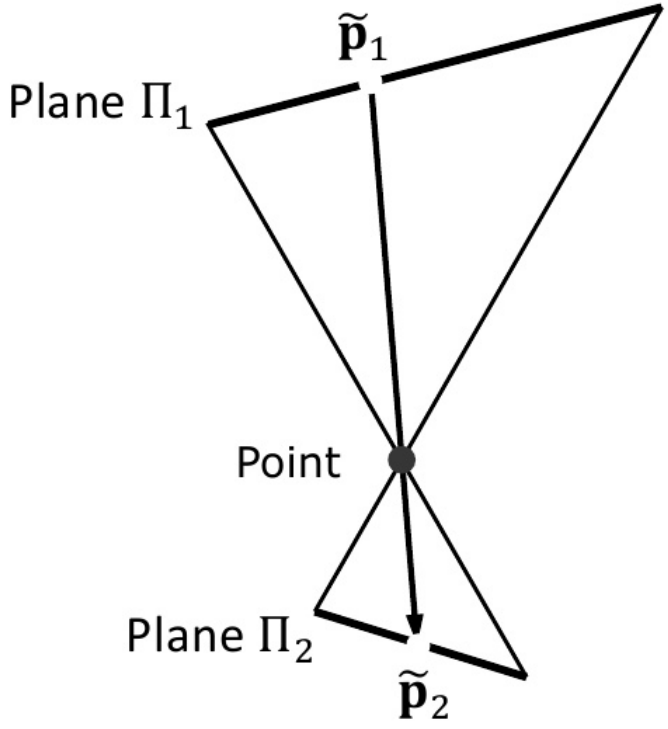
$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \qquad \tilde{\mathbf{p}}_2 = H\tilde{\mathbf{p}}_1$$



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.

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

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



Computing Homography

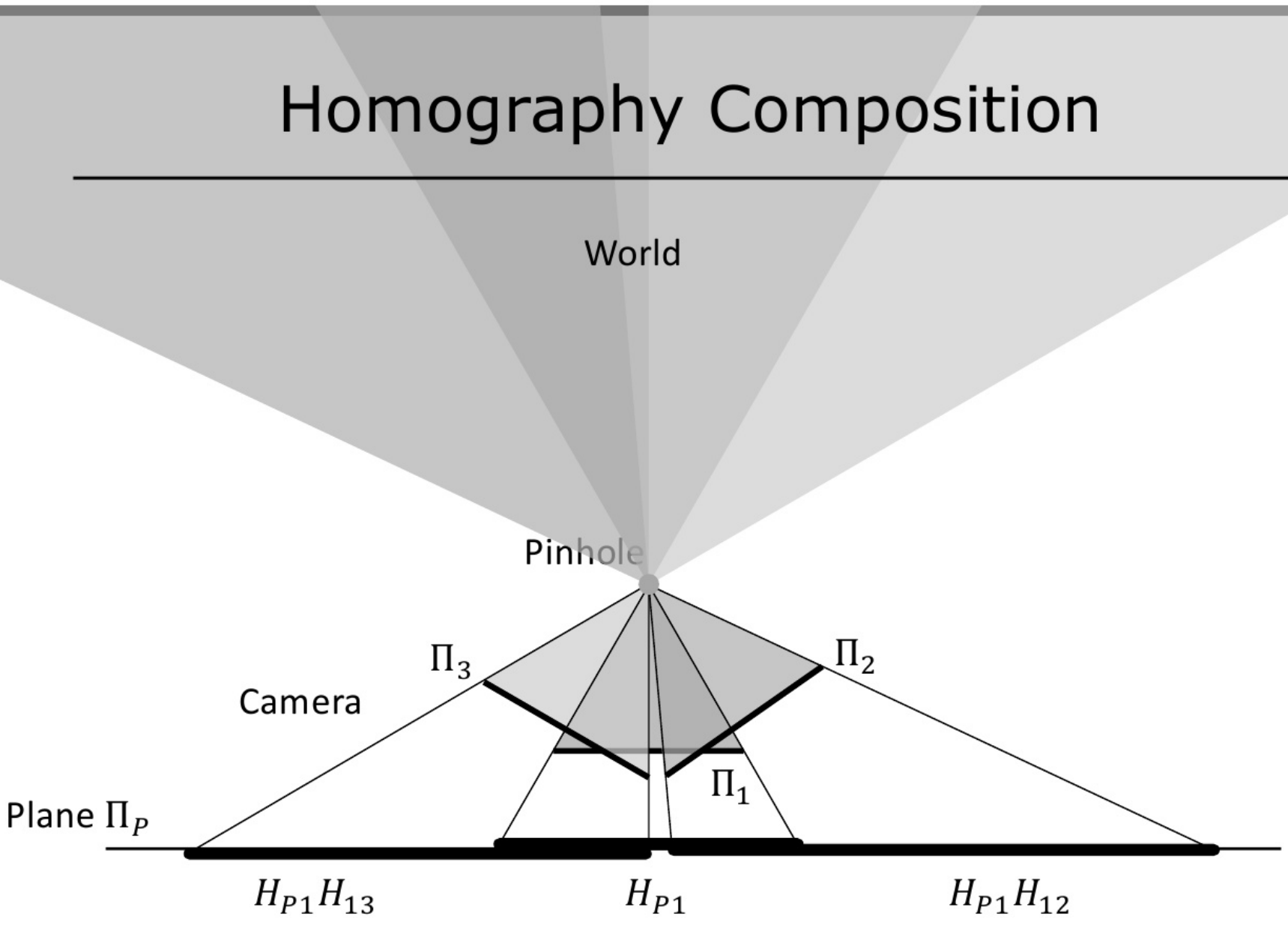
Shree K. Nayar

Columbia University

Topic: Image Stitching, Module: Features

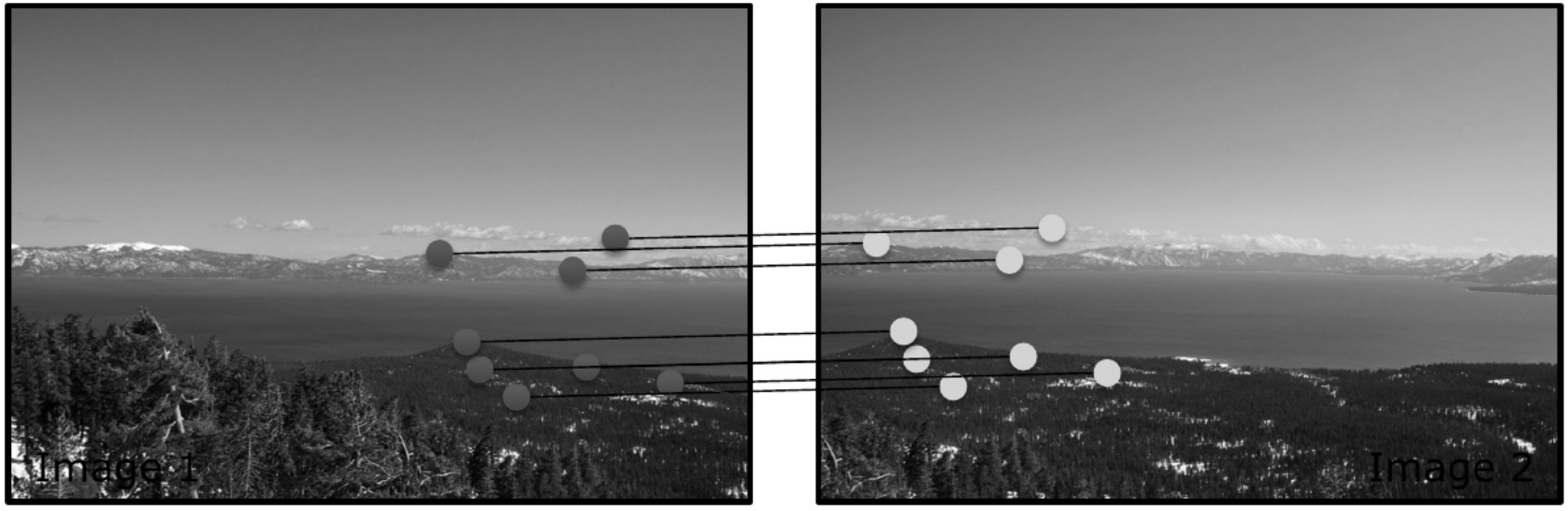
First Principles of Computer Vision

Homography Composition



Useful in stitching planar panoramas

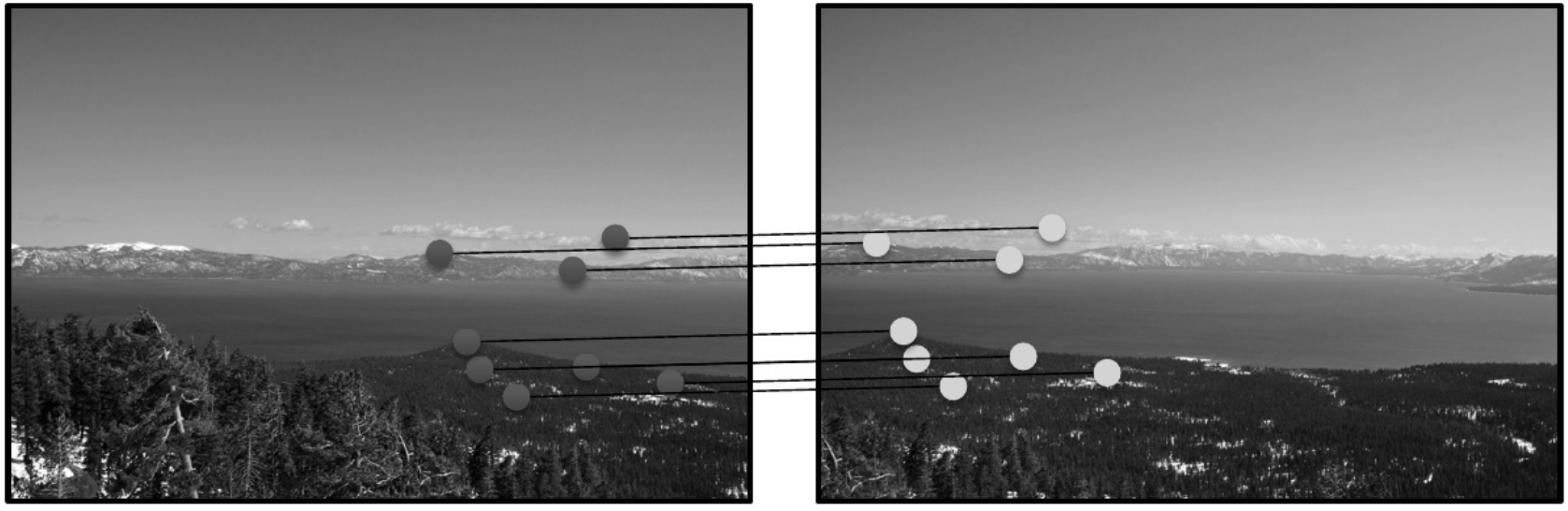
Computing Homography



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 unknowns? 9 …But 8 degrees of freedom

How many minimum pairs of matching points? 4

Computing Homography

For a given pair i of corresponding points:

$$x_d{}^{(i)} = \frac{\tilde{x}_d{}^{(i)}}{\tilde{z}_d{}^{(i)}} = \frac{h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}} \quad \boxed{1}$$

⇨

$$y_d{}^{(i)} = \frac{\tilde{y}_d{}^{(i)}}{\tilde{z}_d{}^{(i)}} = \frac{h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}} \quad \boxed{2}$$

Rearranging the terms:

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

⇨

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

Computing Homography

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

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

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)

3

$\mathbf{h}$

(Unknown)

Computing Homography

Combining the equations for all corresponding points:

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

A (Known)

$\mathbf{h}$ (Unknown)

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:

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

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 $L(\mathbf{h})$.

Matlab: `eig(A'*A)` returns eigenvalues and vectors of $A^T A$

Dealing with Outliers: RANSAC

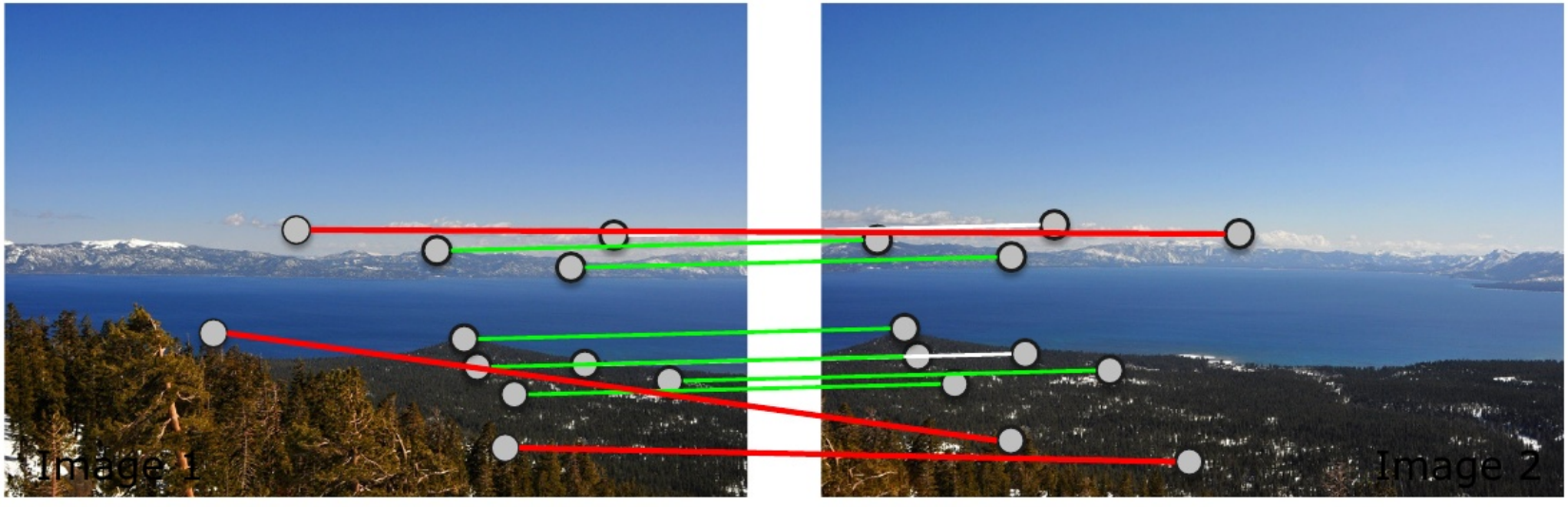
Shree K. Nayar

Columbia University

Topic: Image Stitching, Module: Features

First Principles of Computer Vision

What Could Go Wrong?



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:

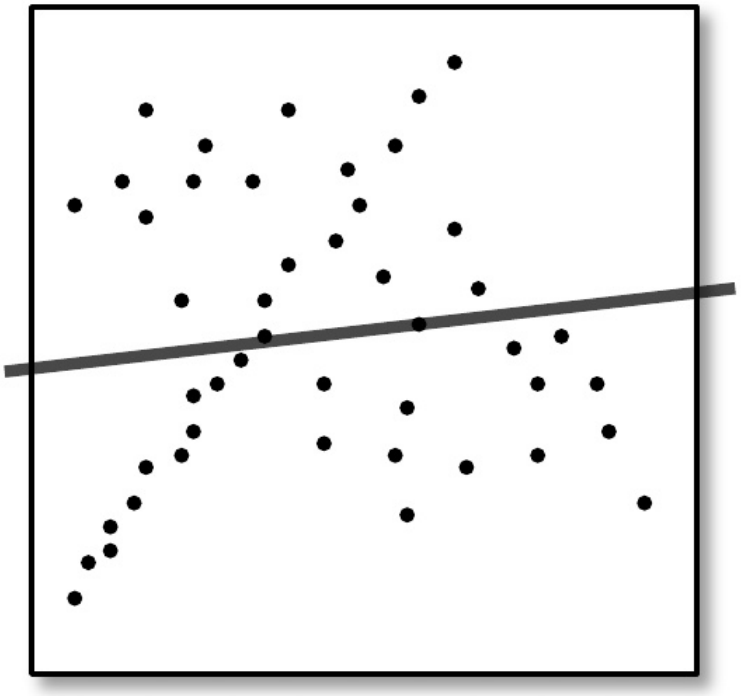
1. Randomly choose s samples. Typically s is the minimum samples to fit a model.
2. Fit the model to the randomly chosen samples.
3. Count the number M of data points (inliers) that fit the model within a measure of error ε.
4. Repeat Steps 1-3 N times
5. Choose the model that has the largest number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

For homography:

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

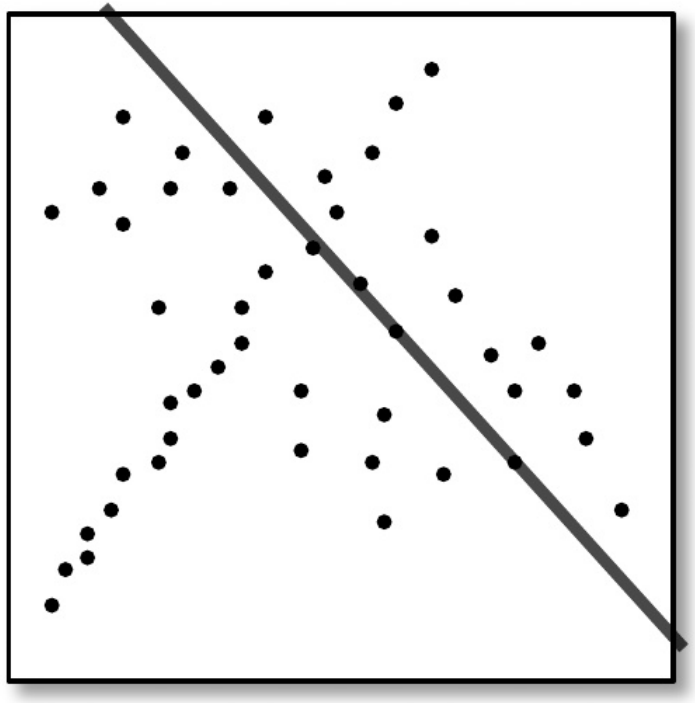
RANSAC Example: Line Fitting

Robust line fitting:



Least Squares Fitting

Inliers: 2

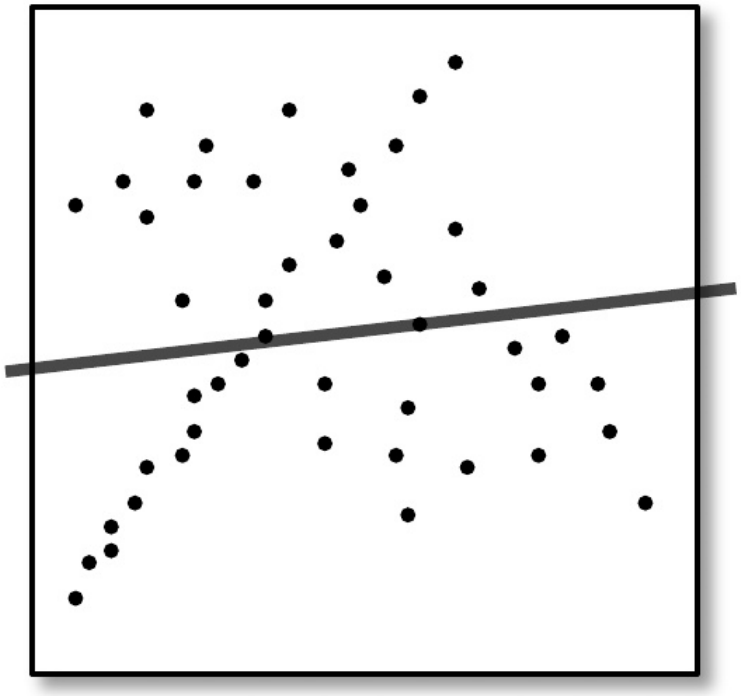


RANSAC Iteration 1

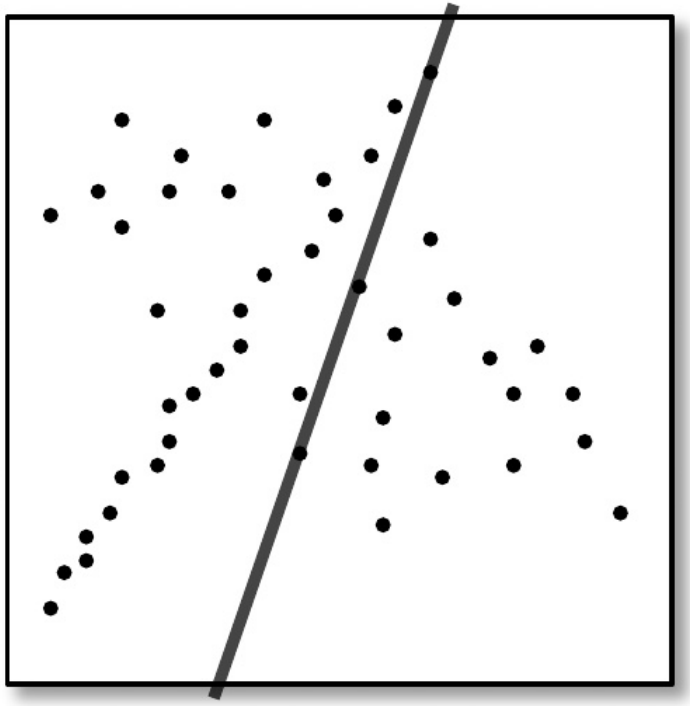
Inliers: 4

RANSAC Example: Line Fitting

Robust line fitting:



Least Squares Fitting

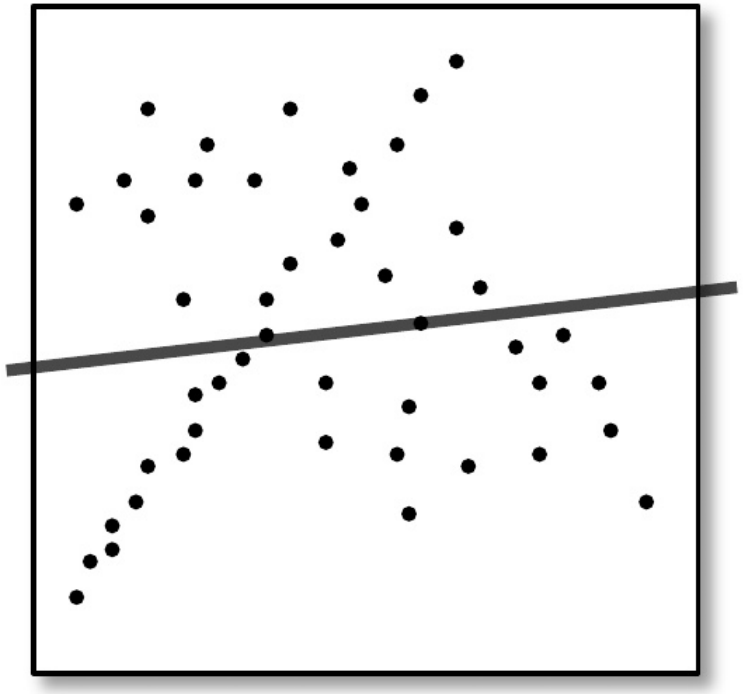


RANSAC Iteration 2

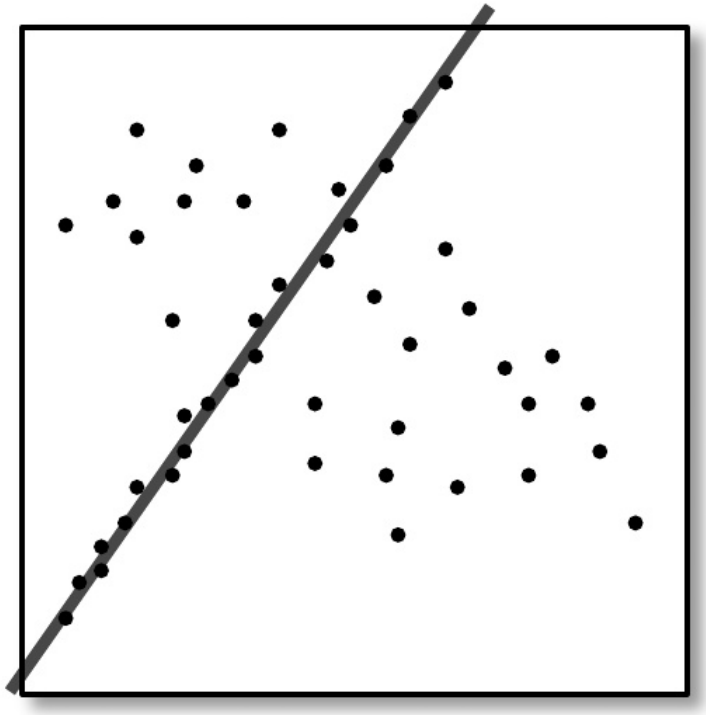
Inliers: 3

RANSAC Example: Line Fitting

Robust line fitting:



Least Squares Fitting



RANSAC Iteration i

Inliers: 20

Warping and Blending Images

Shree K. Nayar

Columbia University

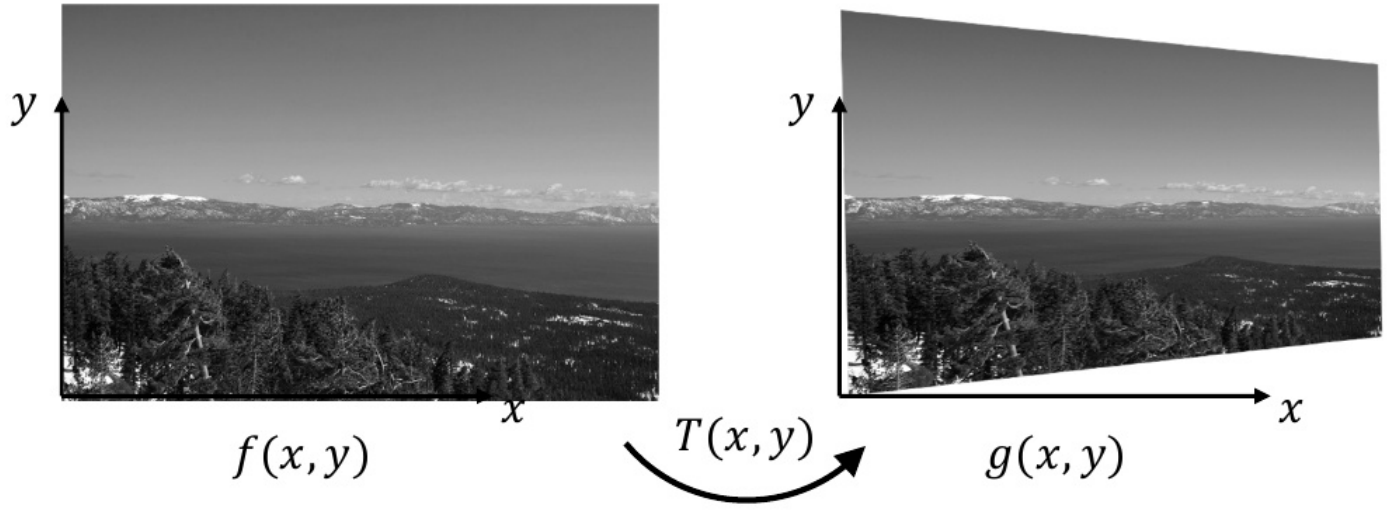
Topic: Image Stitching, Module: Features

First Principles of Computer Vision

Warping Images

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

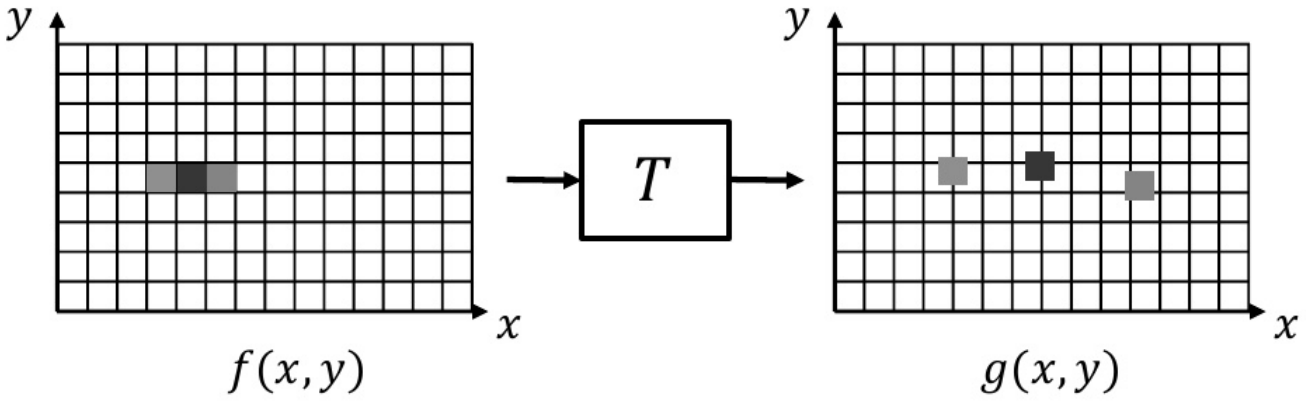
$$g(x, y) = f\big(T(x, y)\big)$$



Forward Warping

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

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

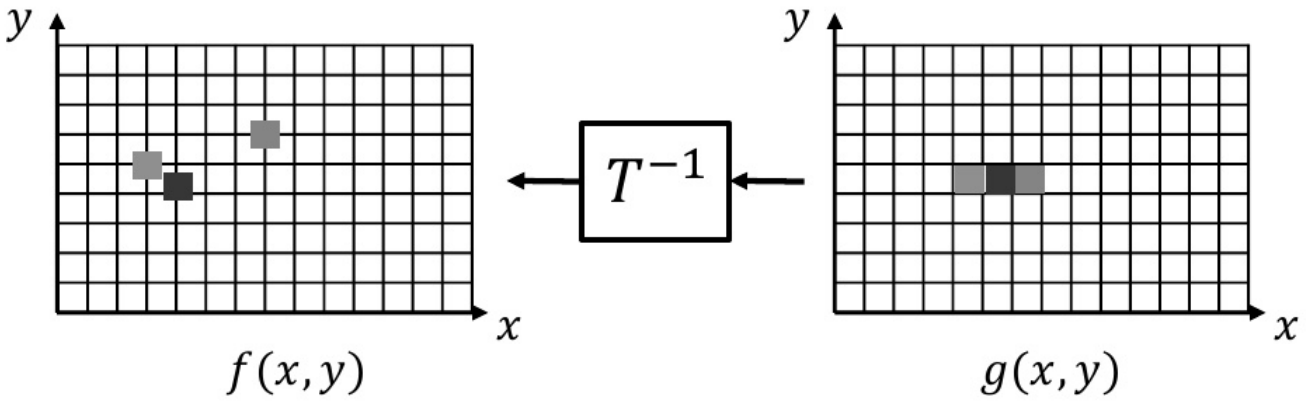


What if pixel lands in between pixels?
What if not all pixels in $g(x, y)$ are filled?
Can result in holes!

Backward Warping

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

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



What if pixel lands between pixels?
Use Nearest Neighbor or Interpolate

Image Alignment Process



Image 1



Image 2

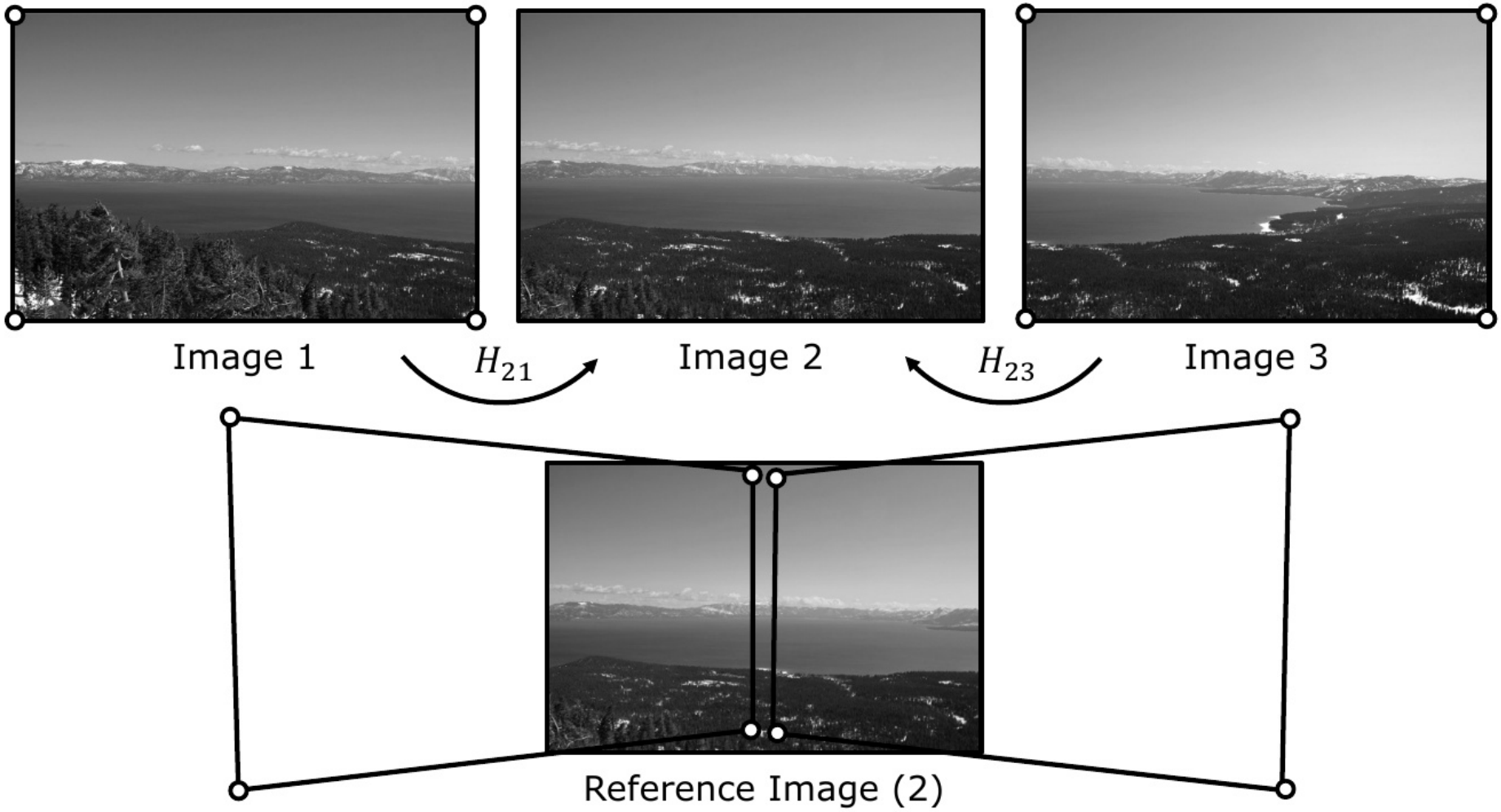


Image 3



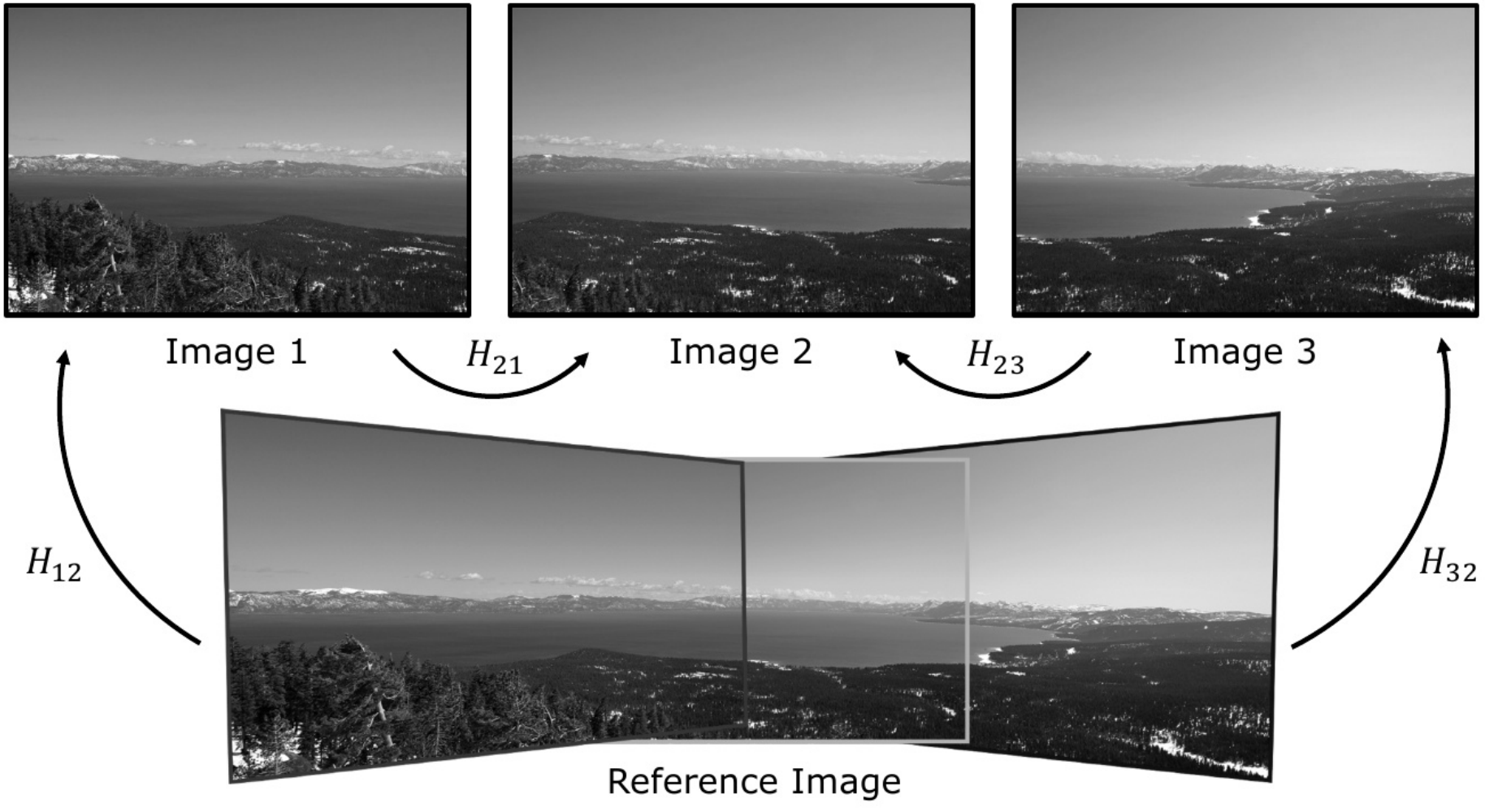
Reference Image
(Image 2)

Image Alignment Process



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

Image Alignment Process



Fill each pixel within bounds, by computing 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

+

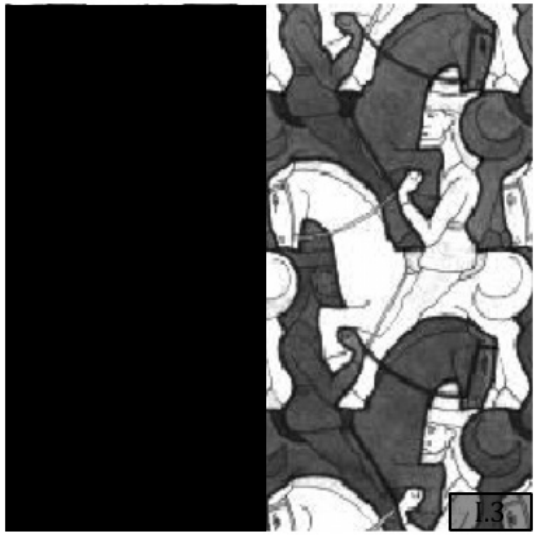


Image I_2

=



Hard overlay



Weight w_1



Weight w_2

Blending Images

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



Image I_1

+



Image I_2

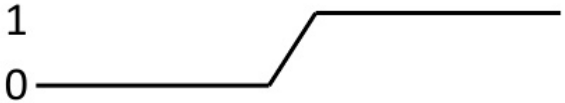
=



Blended Image I_{blend}



Weight w_1



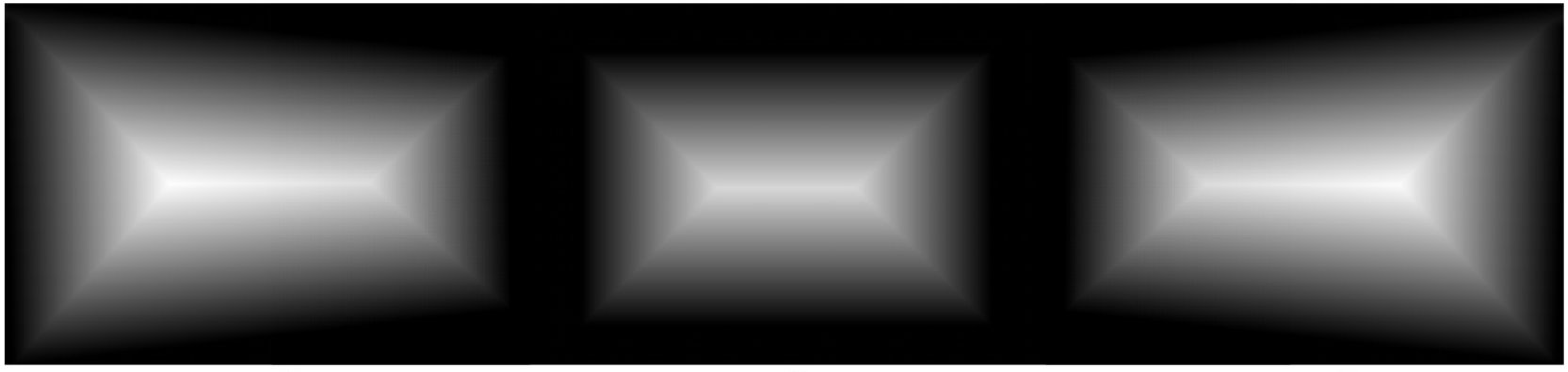
Weight w_2

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

Computing Weighting Functions



Image 1 Image 2 Image 3

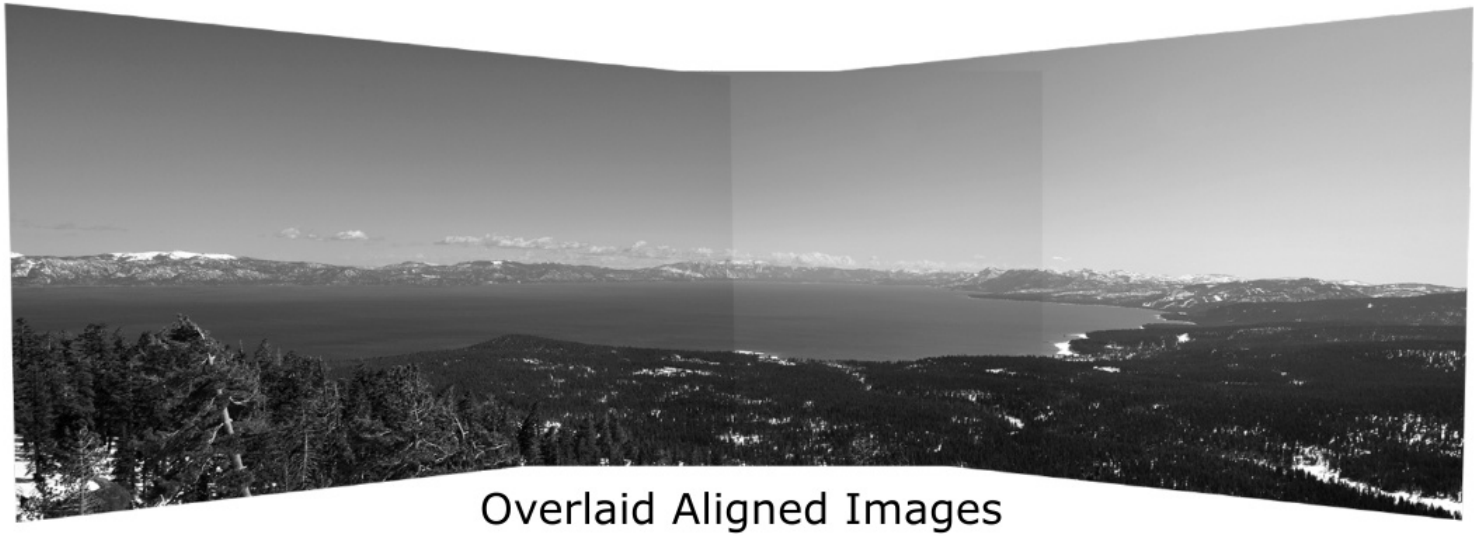


Weight w_1 Weight w_2 Weight w_3

Pixels closer to the edge get a lower weight.

Ex: Distance Transform (`bwdist` in MATLAB).

Weighted Blending

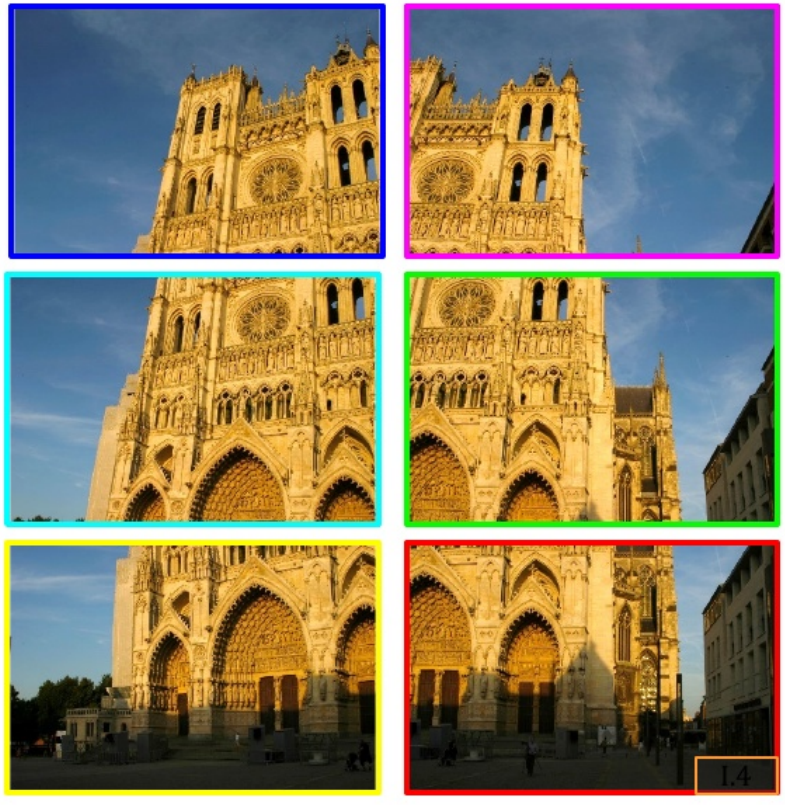


Overlaid Aligned Images

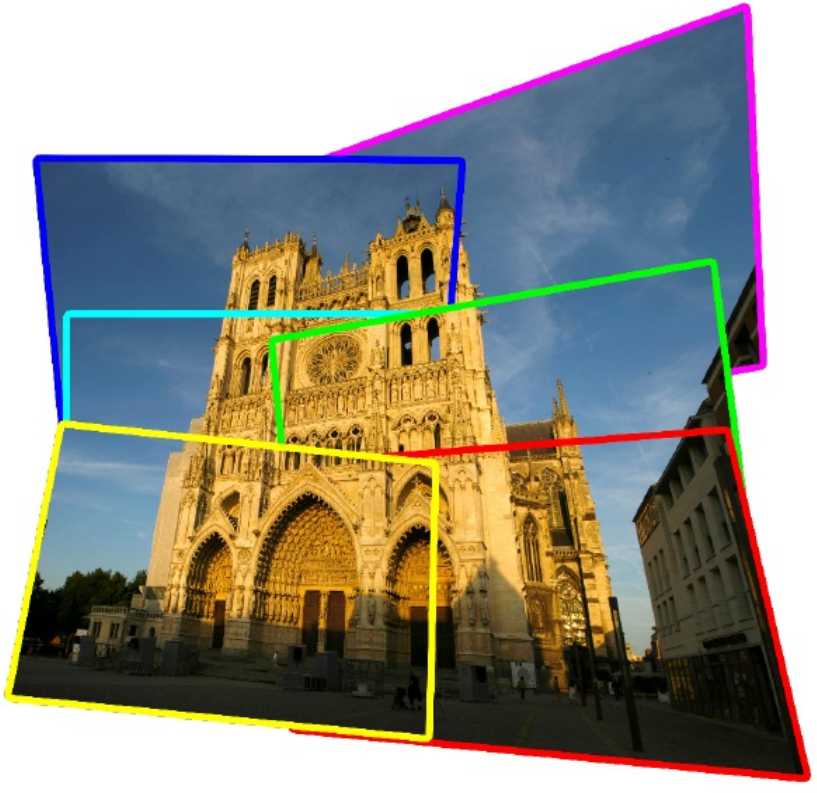


Blended Image

Image Stitching Example

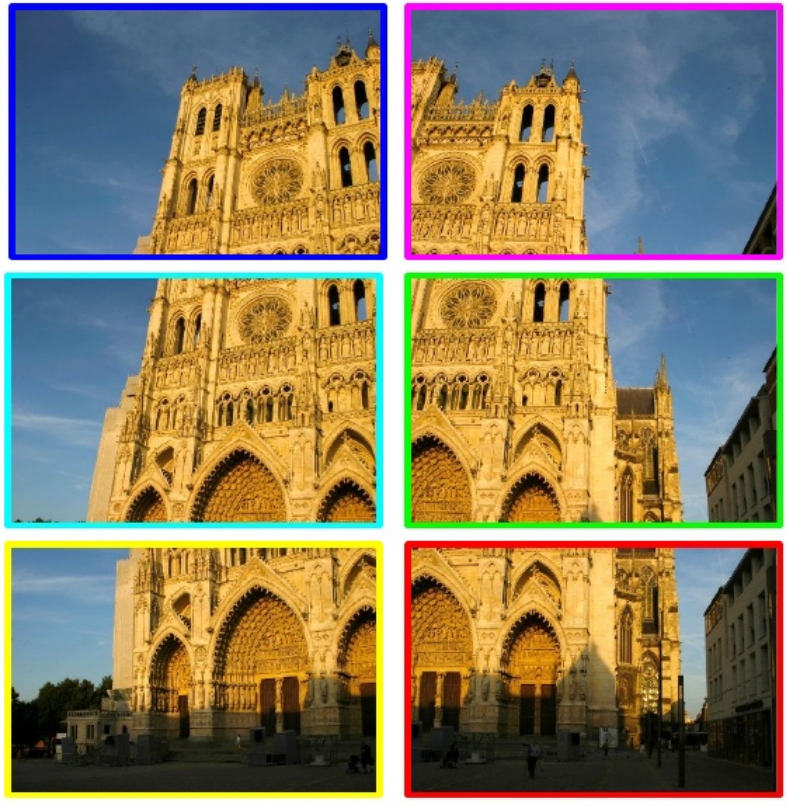


Source Images



Aligned Images

Image Stitching Example



Source Images



Blended Image

References and Credits

Shree K. Nayar

Columbia University

Topic: Image Stitching, Module: Features

First Principles of Computer Vision

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 Vanishing Points. Diesel Demon. Licensed under CC BY 2.0.

I.2 https://www.wikiart.org/en/m-c-escher/horseman-1 ."Horseman, 1946". M.C. Escher. Reproduced under WikiArt Fair Use.

I.3 https://www.wikiart.org/en/m-c-escher/fish-boat . "Fish and Boat, 1948". M.C. Escher. Reproduced under WikiArt Fair Use.

I.4 PTGui. Used with permission.