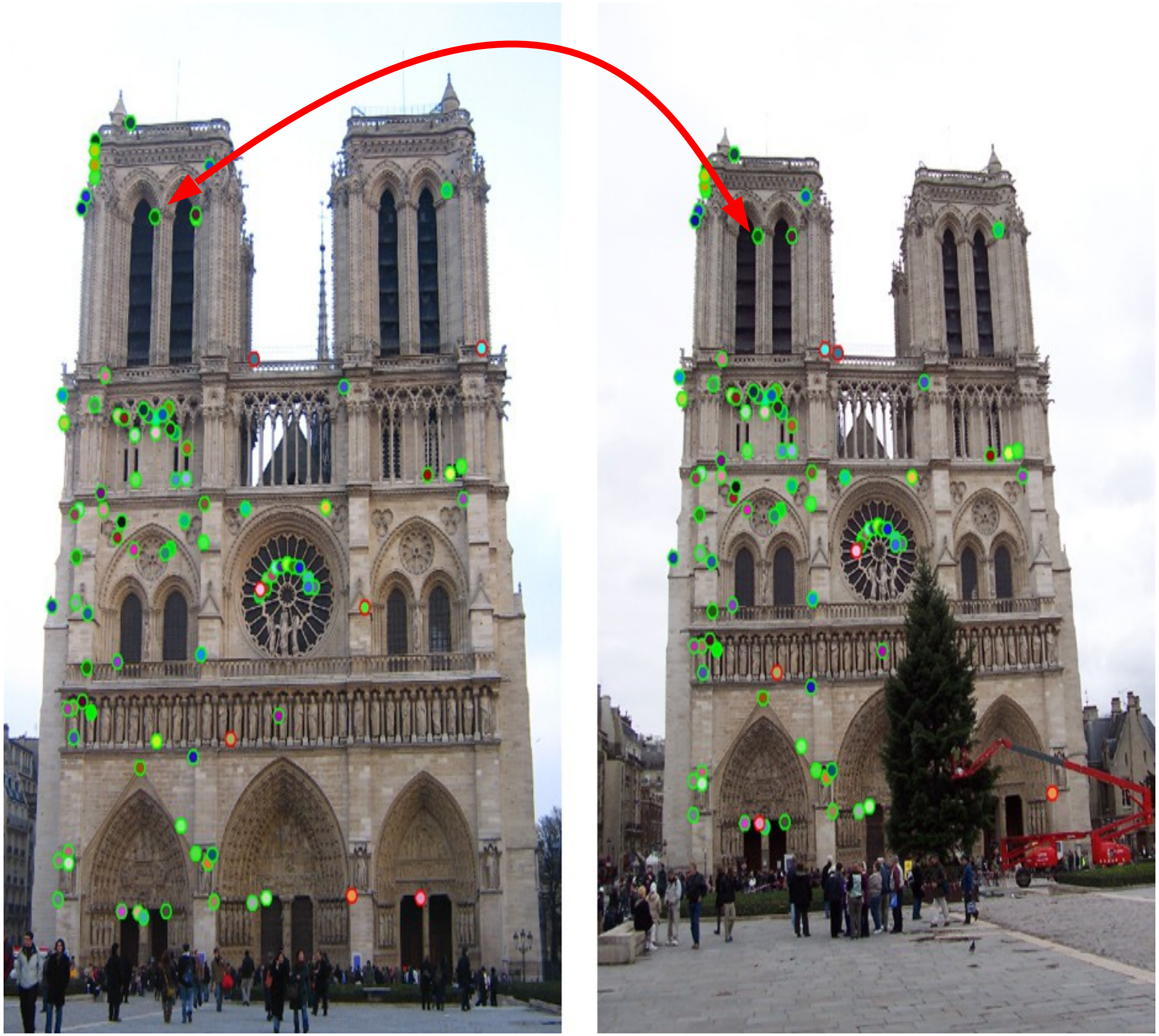
Matching and Image Alignment

Computer Vision
Fall 2018
Columbia University

Feature Matching

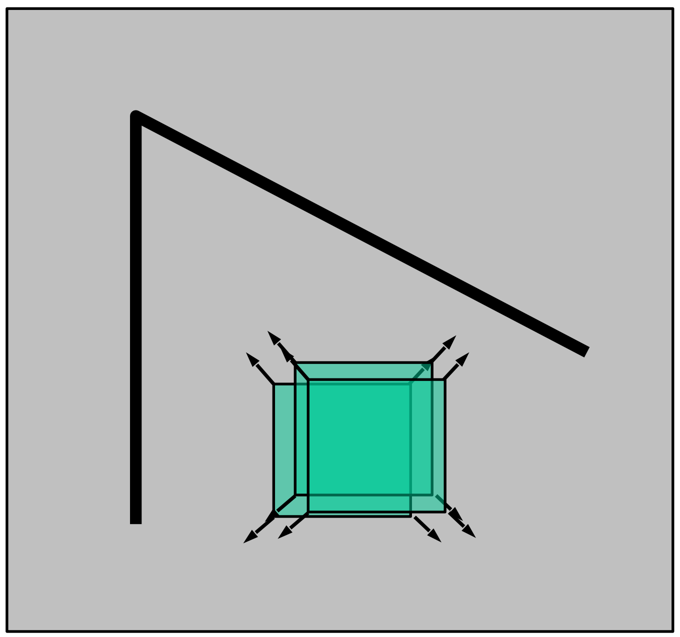


1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

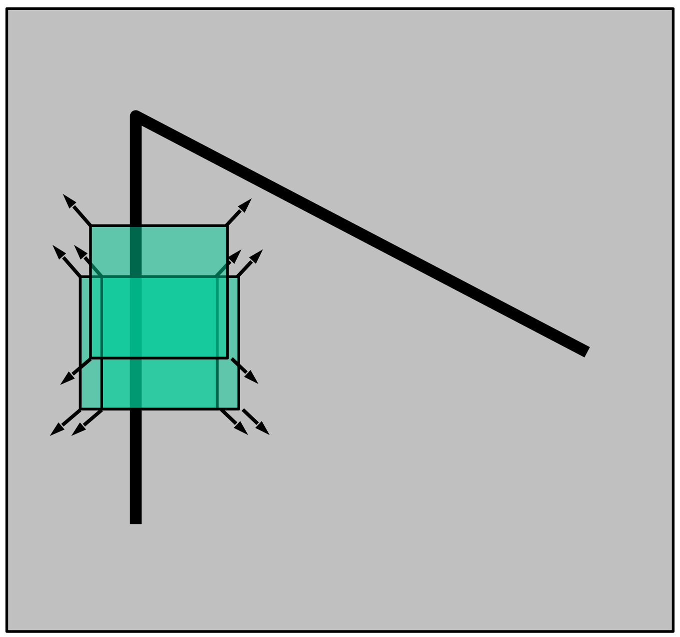
Slide credit: James Hays

SIFT Review

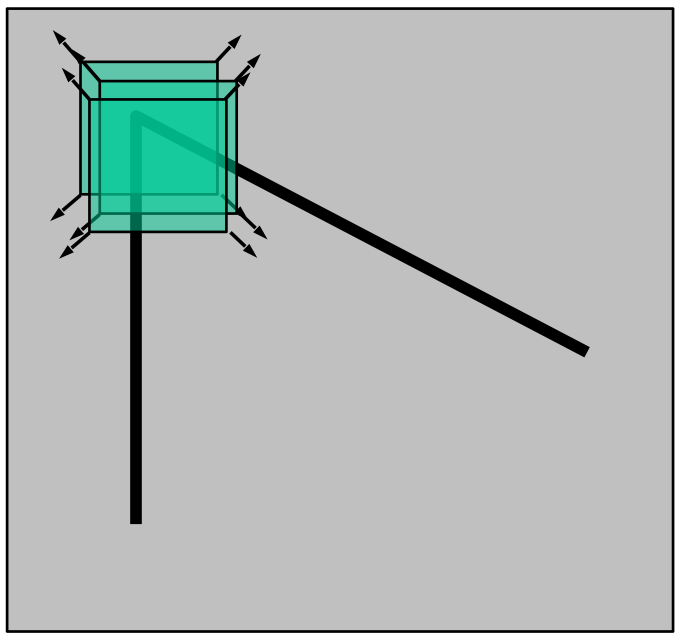
Corner Detector: Basic Idea



"flat" region:
no change in any direction



"edge":
no change along the edge direction

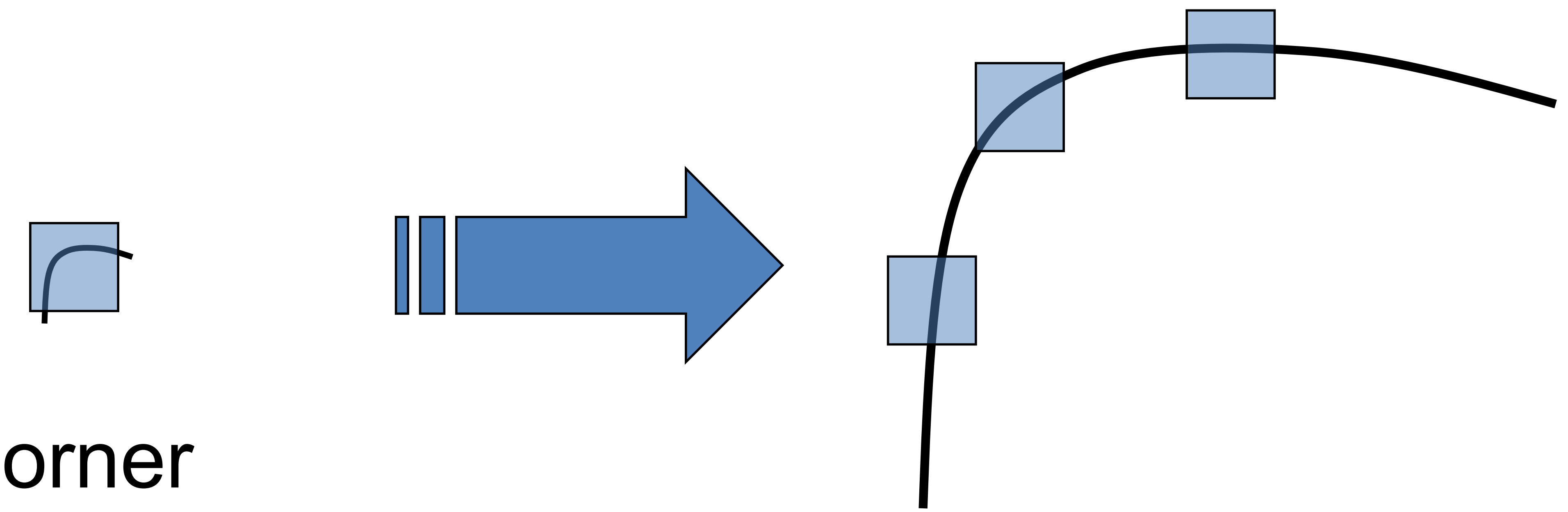


"corner":
significant change in all directions

Defn: points are "matchable" if small shifts always produce a large SSD error

Source: Deva Ramanan

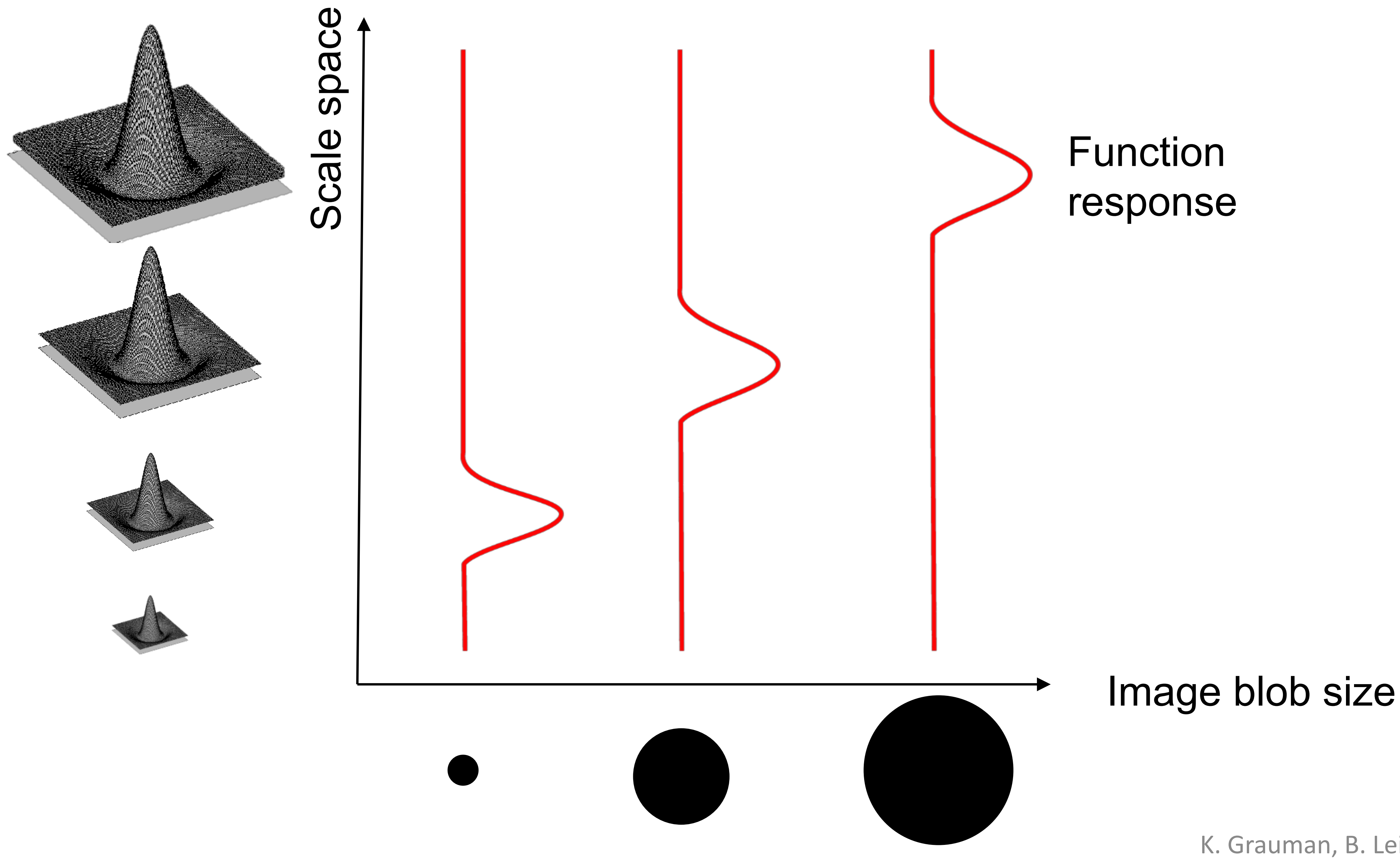
Scaling



All points will be classified as edges

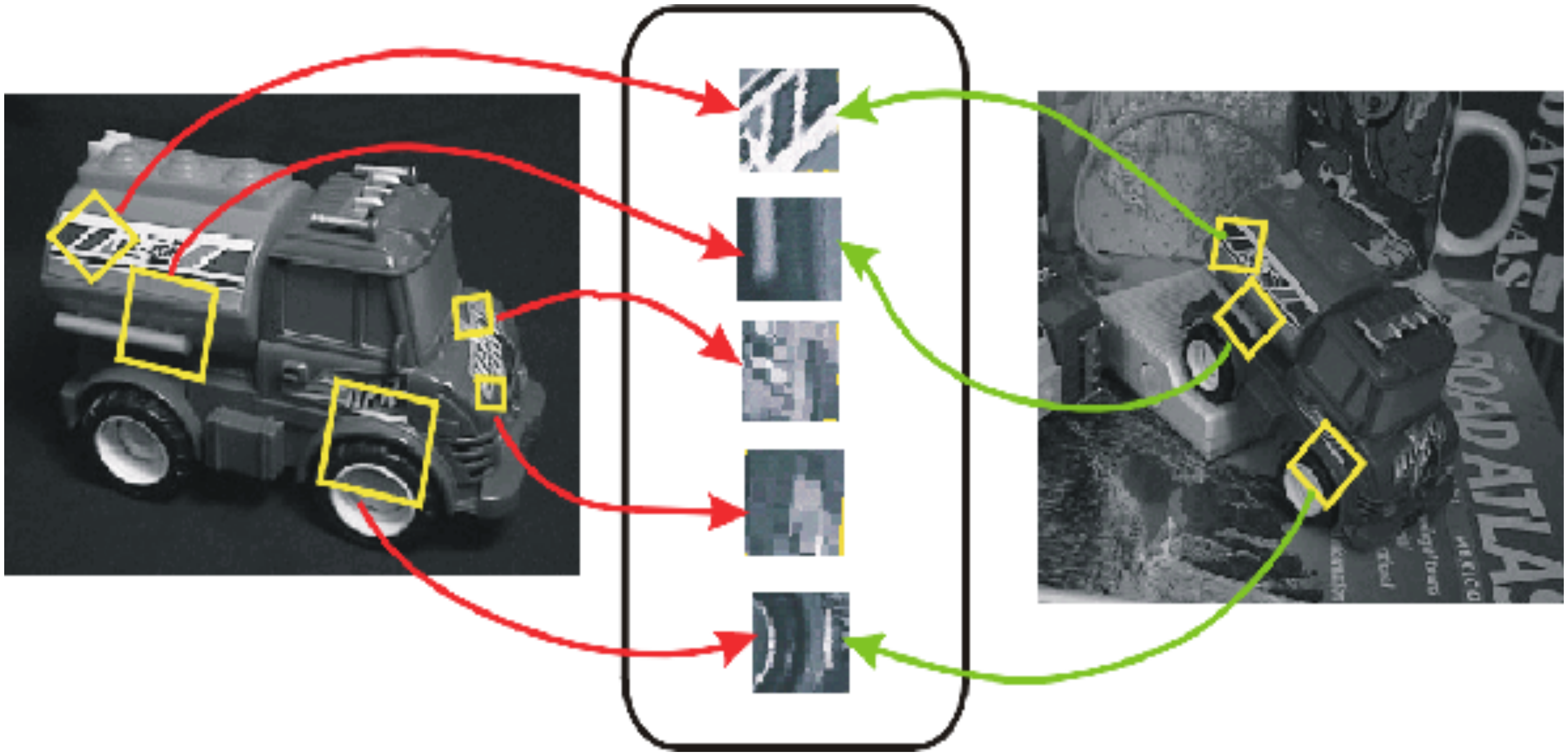
What Is A Useful Signature Function *f* ?

- "Blob" detector is common for corners
  - \- Laplacian (2nd derivative) of Gaussian (LoG)



K. Grauman, B. Leibe

Coordinate frames

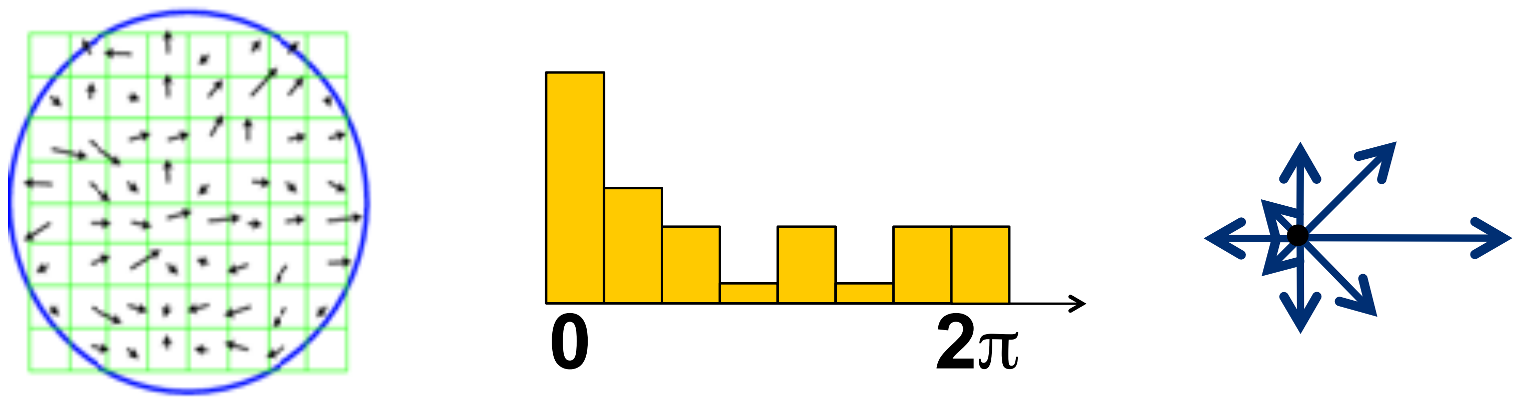


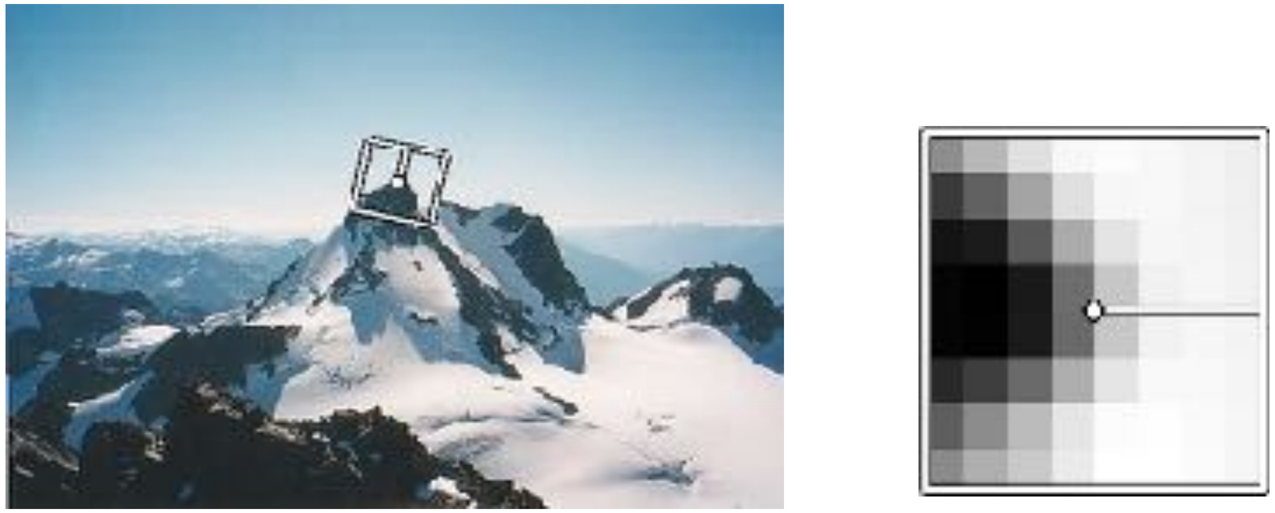
Represent each patch in a canonical scale and orientation (or general *affine* coordinate frame)

Source: Deva Ramanan

Find dominant orientation

Compute gradients for all pixels in patch. Histogram (bin) gradients by orientation

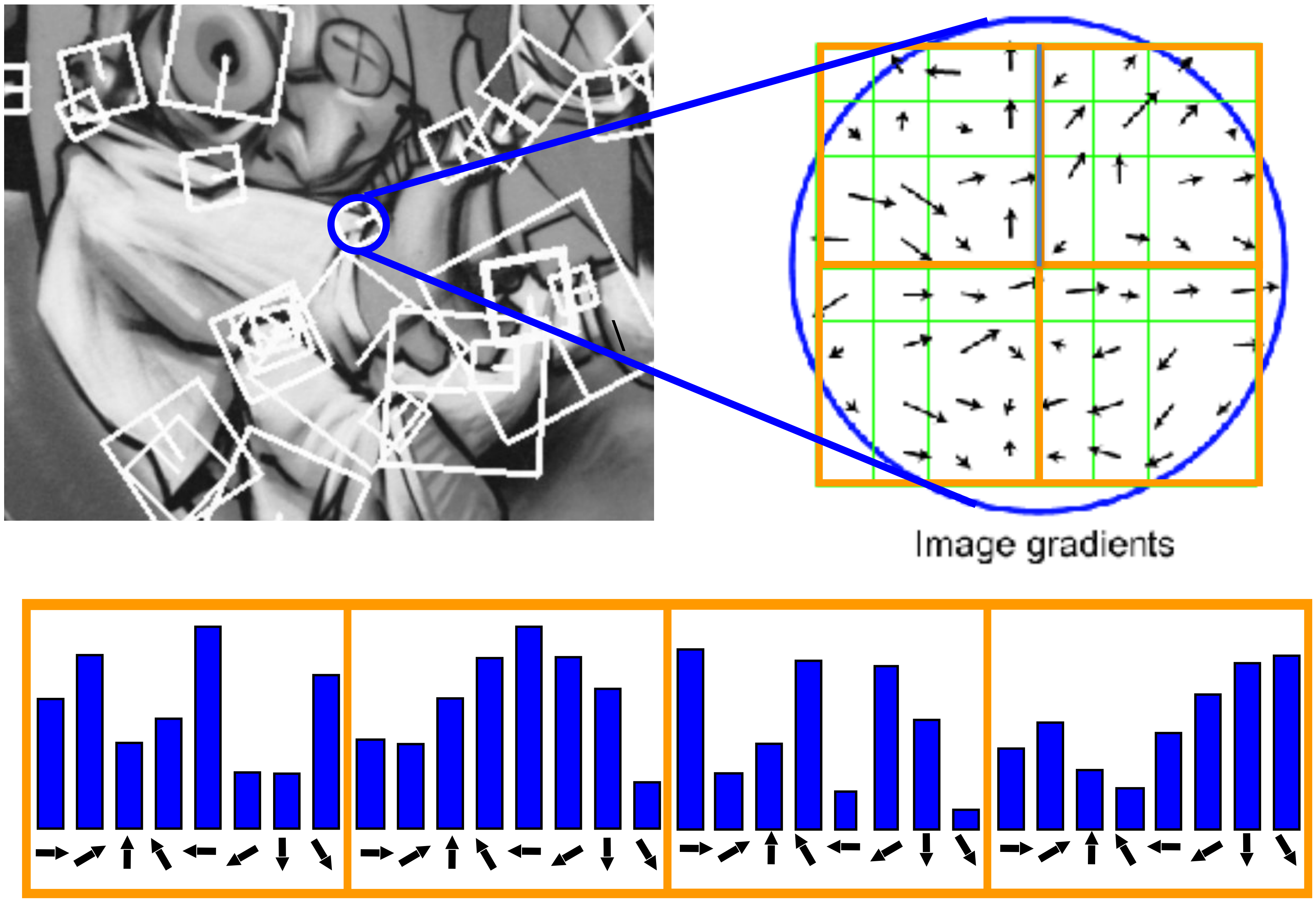




Source: Deva Ramanan

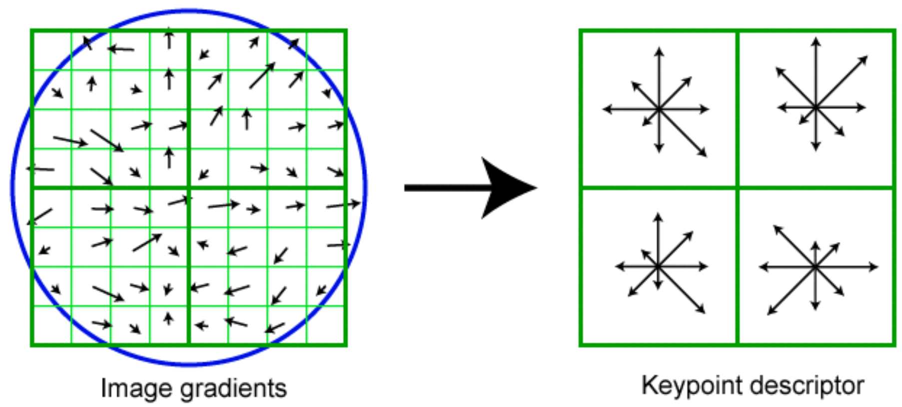
Computing the SIFT Descriptor

Histograms of gradient directions over spatial regions



Source: Deva Ramanan

Post-processing



1. Rescale 128-dim vector to have unit norm

$$x = \frac{x}{||x||}, \quad x \in R^{128}$$

"invariant to linear scalings of intensity"

2. Clip high values

$$x := \min(x, .2)$$
$$x := \frac{x}{||x||}$$

approximate binarization allows for for flat patches with small gradients to remain stable

Source: Deva Ramanan

Matching

Panoramas

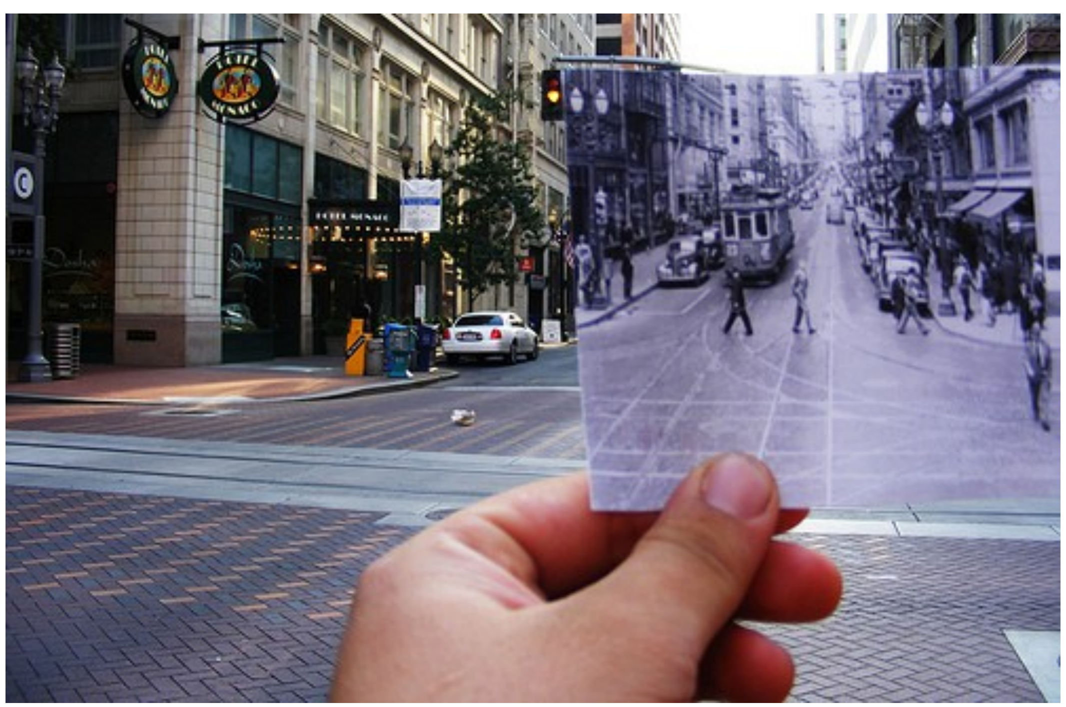
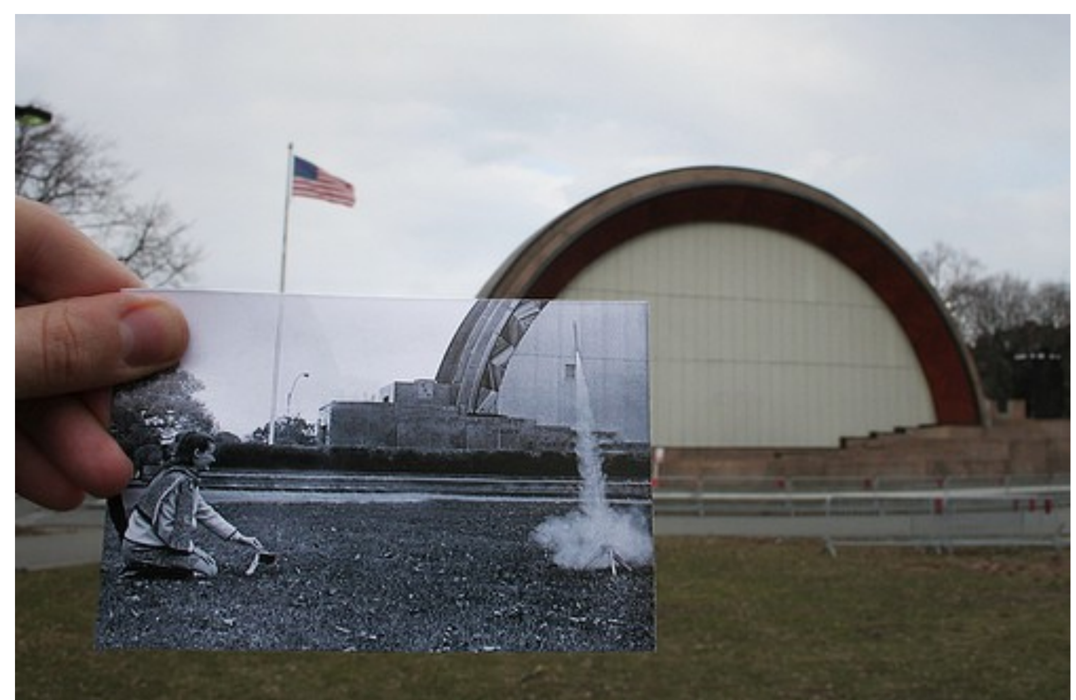
Slide credit: Olga Russakovsky

Gigapixel Images

danielhartz.com

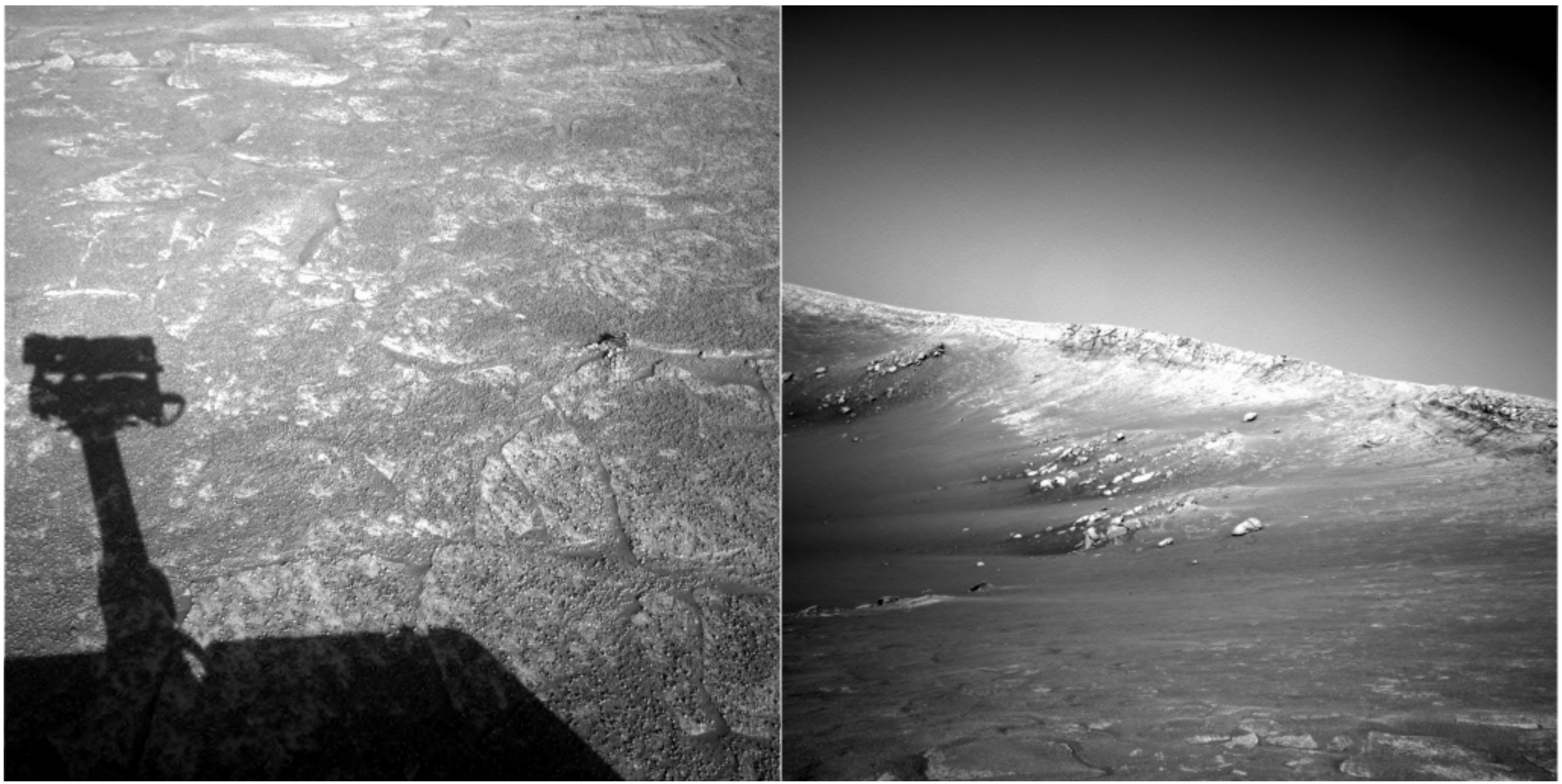
Look into the Past



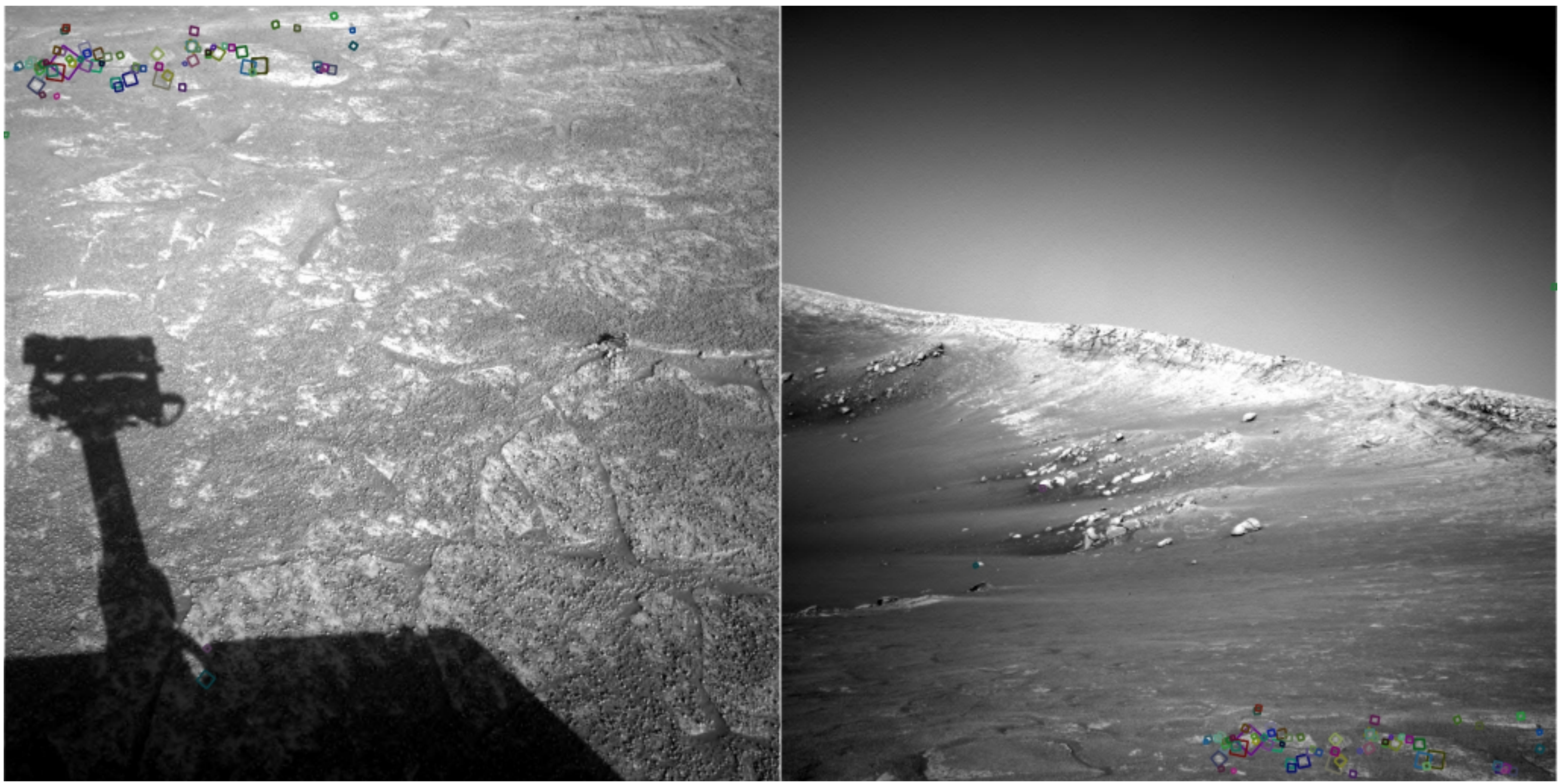
Slide credit: Olga Russakovsky

Can you find the matches?



NASA Mars Rover images

Slide credit: S. Lazebnik

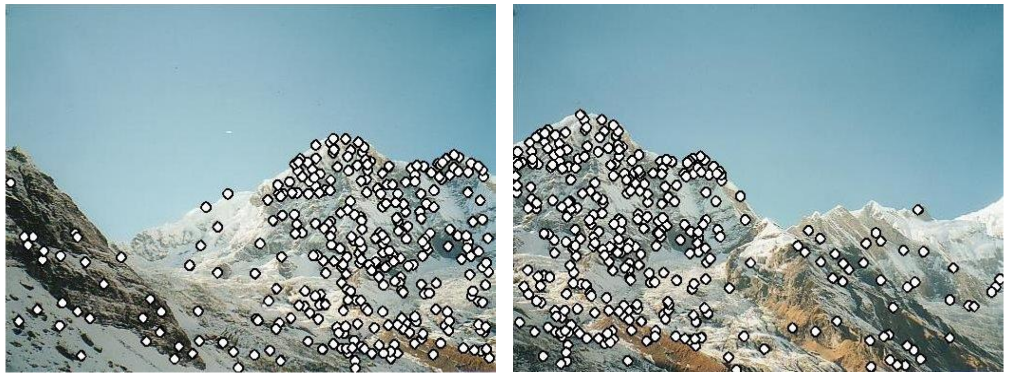


NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

Slide credit: S. Lazebnik

Discussion

- *Design a feature point matching scheme.*
- Two images, I_1 and I_2



- Two sets X_1 and X_2 of feature points
  - Each feature point $\mathbf{x_1}$ has a descriptor $\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$
- Distance, bijective/injective/surjective, noise, confidence, computational complexity, generality…

Slide credit: James Hays

Distance Metric

- Euclidean distance:

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

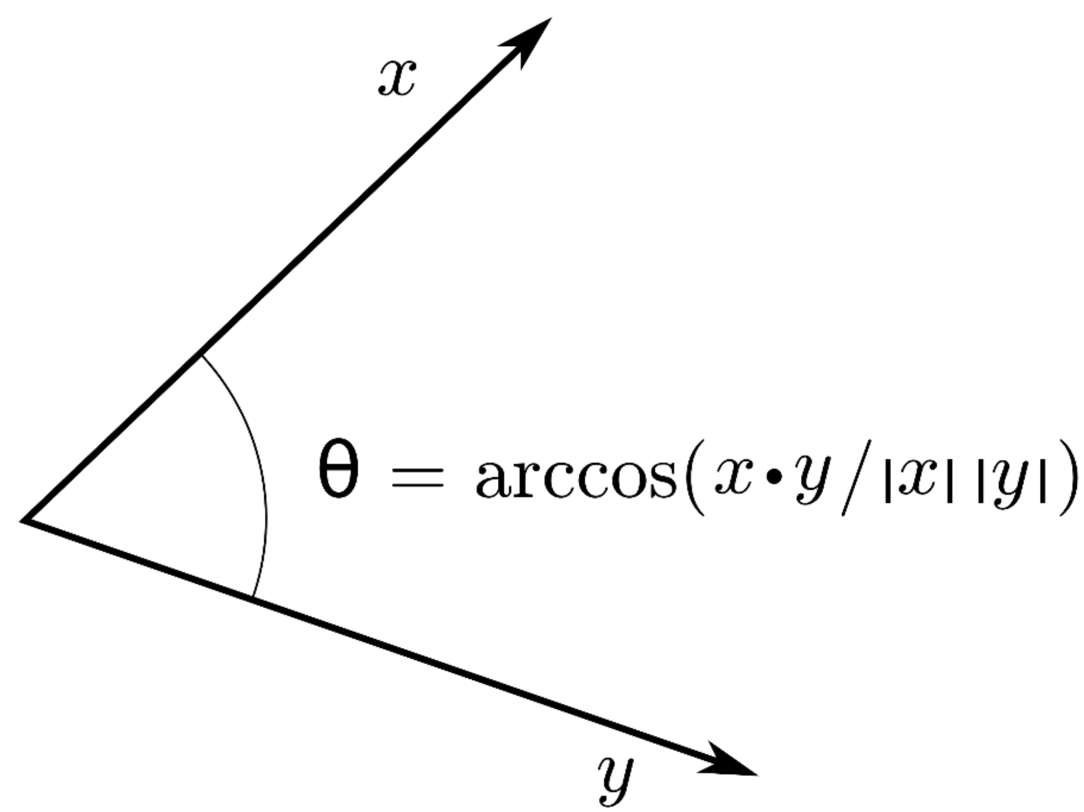
$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

$$\|\mathbf{q}-\mathbf{p}\| = \sqrt{(\mathbf{q}-\mathbf{p})\cdot(\mathbf{q}-\mathbf{p})}.$$

- Cosine similarity:

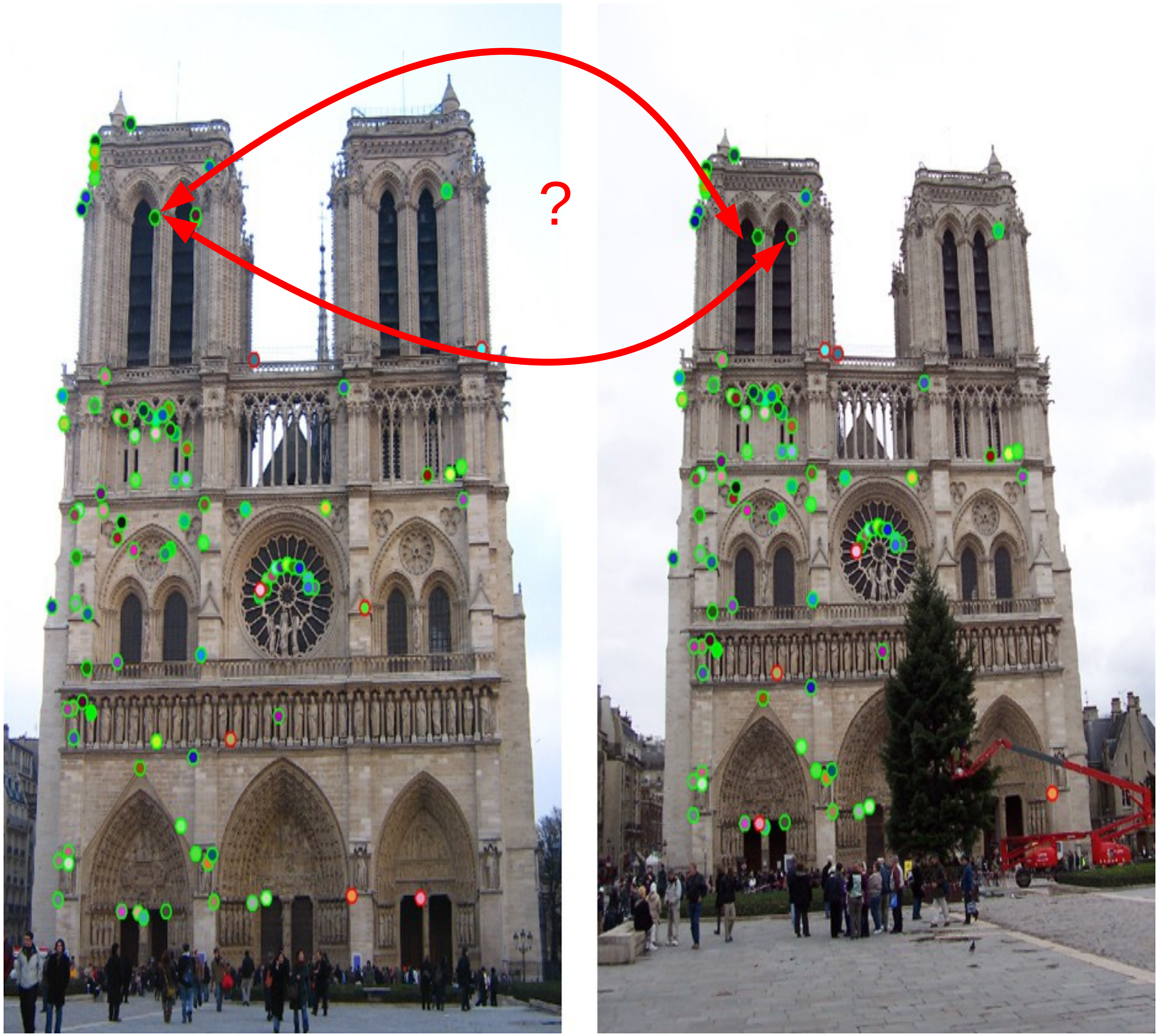
$$\mathbf{a}\cdot\mathbf{b} = \|\mathbf{a}\|_2 \|\mathbf{b}\|_2 \cos\theta$$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2}$$



Wikipedia

Matching Ambiguity



Locally, feature matches are ambiguous

=> need to fit a **model** to find globally consistent matches

Slide credit: James Hays

Feature Matching

- Criteria 1:
  - Compute distance in feature space, e.g., Euclidean distance between 128-dim SIFT descriptors
  - Match point to lowest distance (nearest neighbor)

- Problems:
  - Does everything have a match?

Slide credit: James Hays

Feature Matching

- Criteria 2:
  – Compute distance in feature space, e.g., Euclidean distance between 128-dim SIFT descriptors
  – Match point to lowest distance (nearest neighbor)
  – Ignore anything higher than threshold (no match!)

- Problems:
  – Threshold is hard to pick
  – Non-distinctive features could have lots of close matches, only one of which is correct

Slide credit: James Hays

Nearest Neighbor Distance Ratio

Compare distance of closest (NN1) and second-closest (NN2) feature vector neighbor.

- If NN1 ≈ NN2, ratio $\frac{NN1}{NN2}$ will be ≈ 1 -> matches too close.
- As NN1 << NN2, ratio $\frac{NN1}{NN2}$ tends to 0.

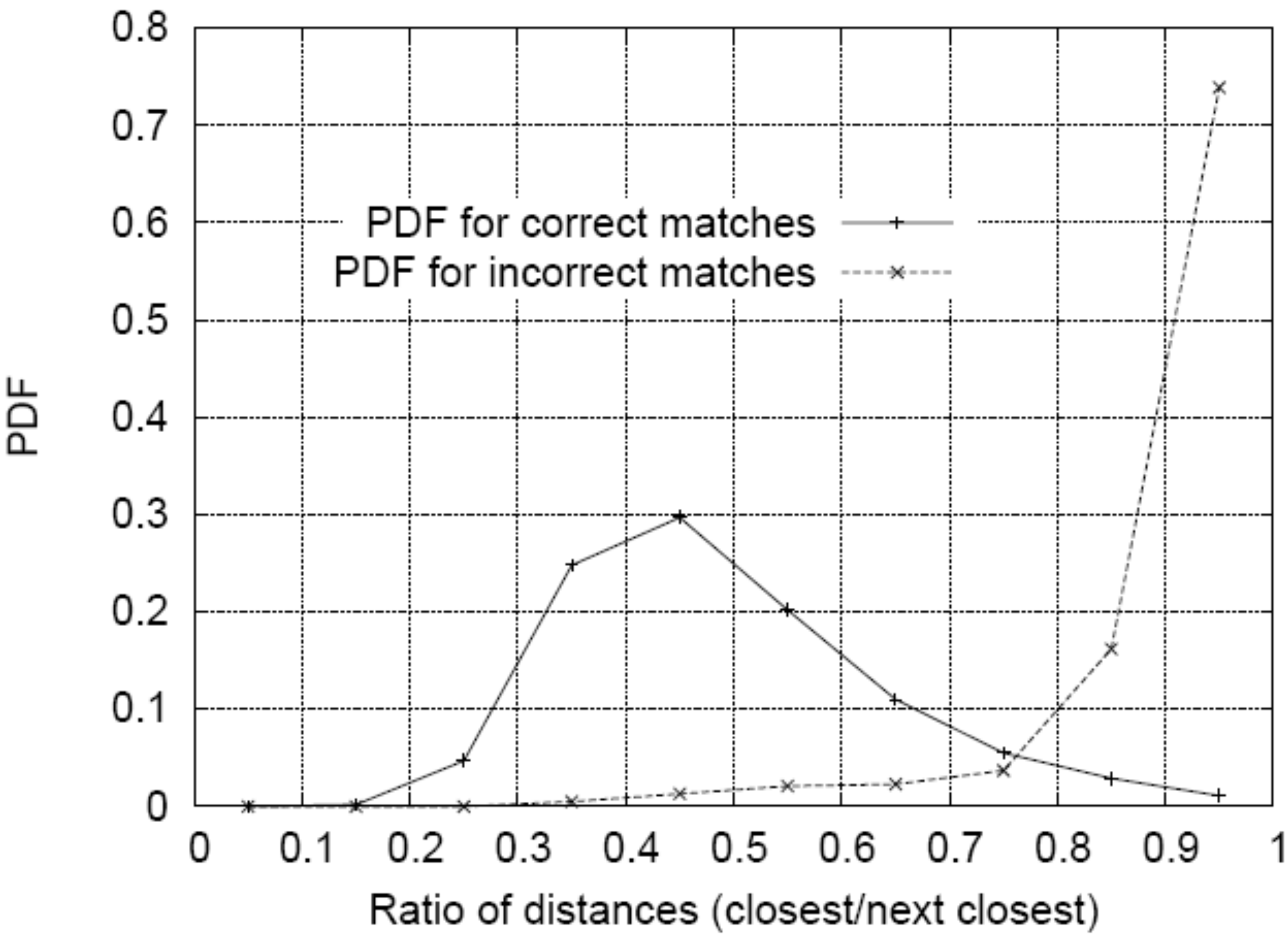
Sorting by this ratio puts matches in order of confidence.

Threshold ratio – but how to choose?

Slide credit: James Hays

Nearest Neighbor Distance Ratio

- Lowe computed a probability distribution functions of ratios
- 40,000 keypoints with hand-labeled ground truth



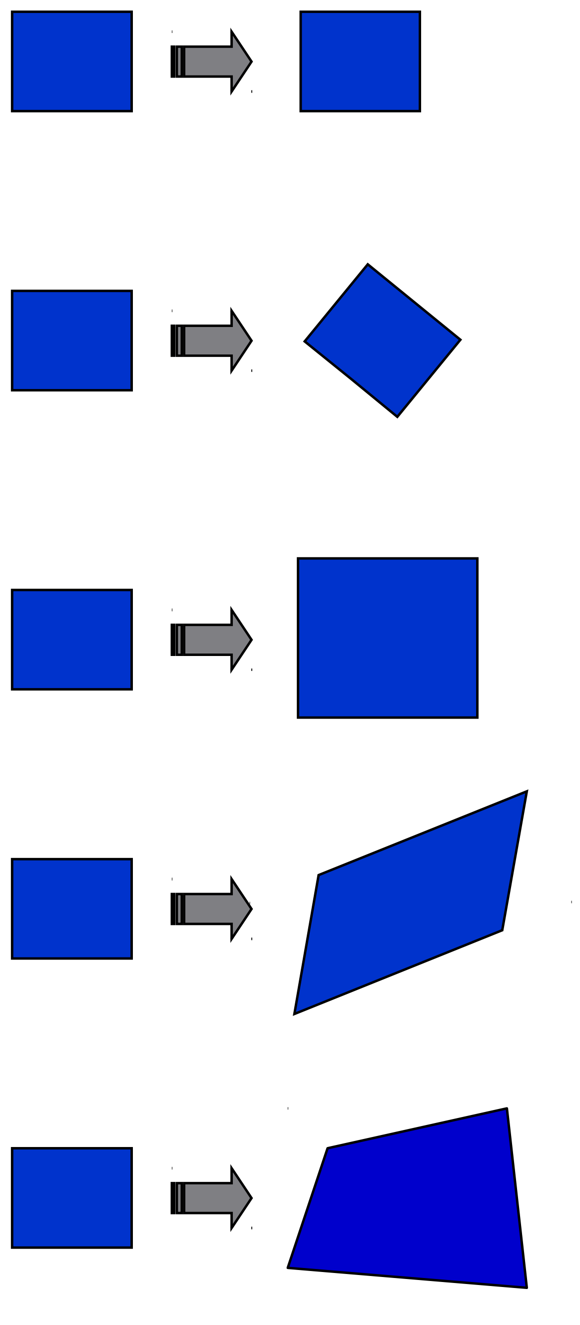
Ratio threshold depends on your application's view on the trade-off between the number of false positives and true positives!

Lowe IJCV 2004

What is the transformation between these images?

Transformation Models

- Translation only
- Rigid body (translate+rotate)
- Similarity (translate+rotate+scale)
- Affine
- Homography (projective)



Homogenous Coordinates

Cartesian: → Homogenous:

$$P = (x, y)$$

$$\tilde{P} = (x, y, 1)$$

Slide credit: Peter Corke

Homogenous Coordinates

Cartesian:

$$P = (x, y)$$

Homogenous:

$$\tilde{P} = (x, y, 1)$$

Homogenous:

$$\tilde{P} = (\tilde{x}, \tilde{y}, \tilde{z})$$

Slide credit: Peter Corke

Homogenous Coordinates

Cartesian:

$$P = (x, y)$$

→

Homogenous:

$$\tilde{P} = (x, y, 1)$$

Cartesian:

$$P = \left(\frac{\tilde{x}}{\tilde{z}}, \frac{\tilde{y}}{\tilde{z}} \right)$$

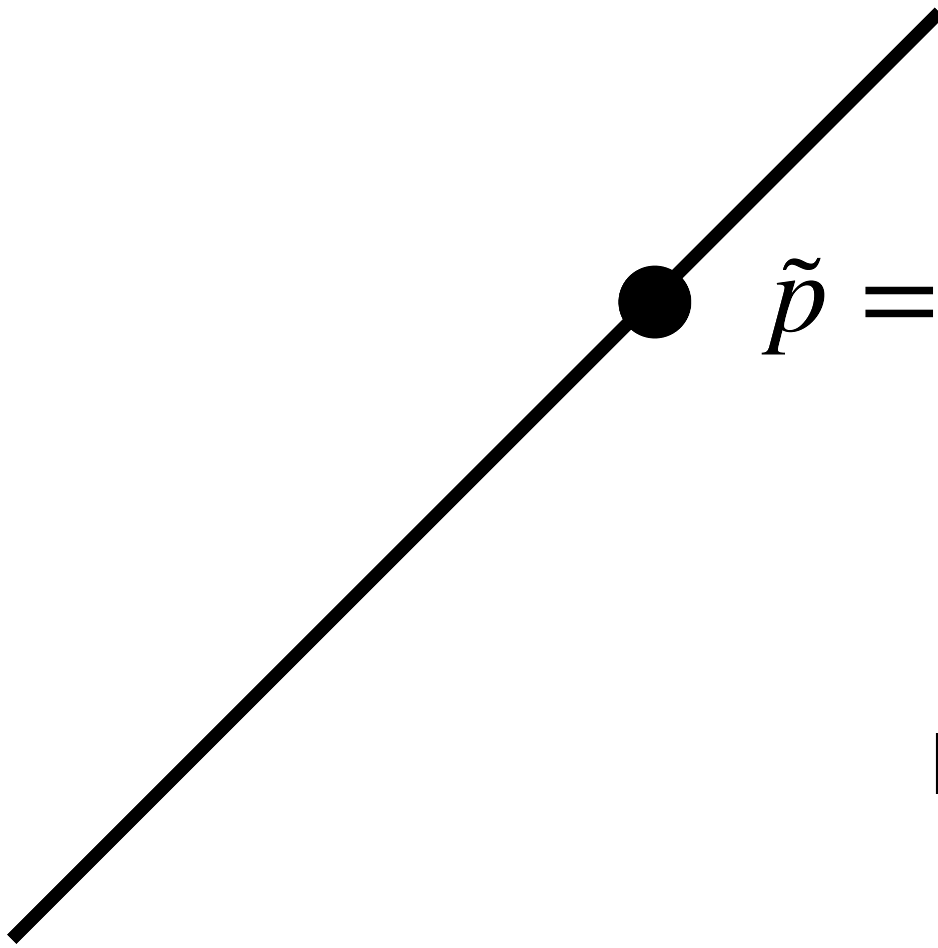
←

Homogenous:

$$\tilde{P} = (\tilde{x}, \tilde{y}, \tilde{z})$$

Slide credit: Peter Corke

Lines and Points are Duals

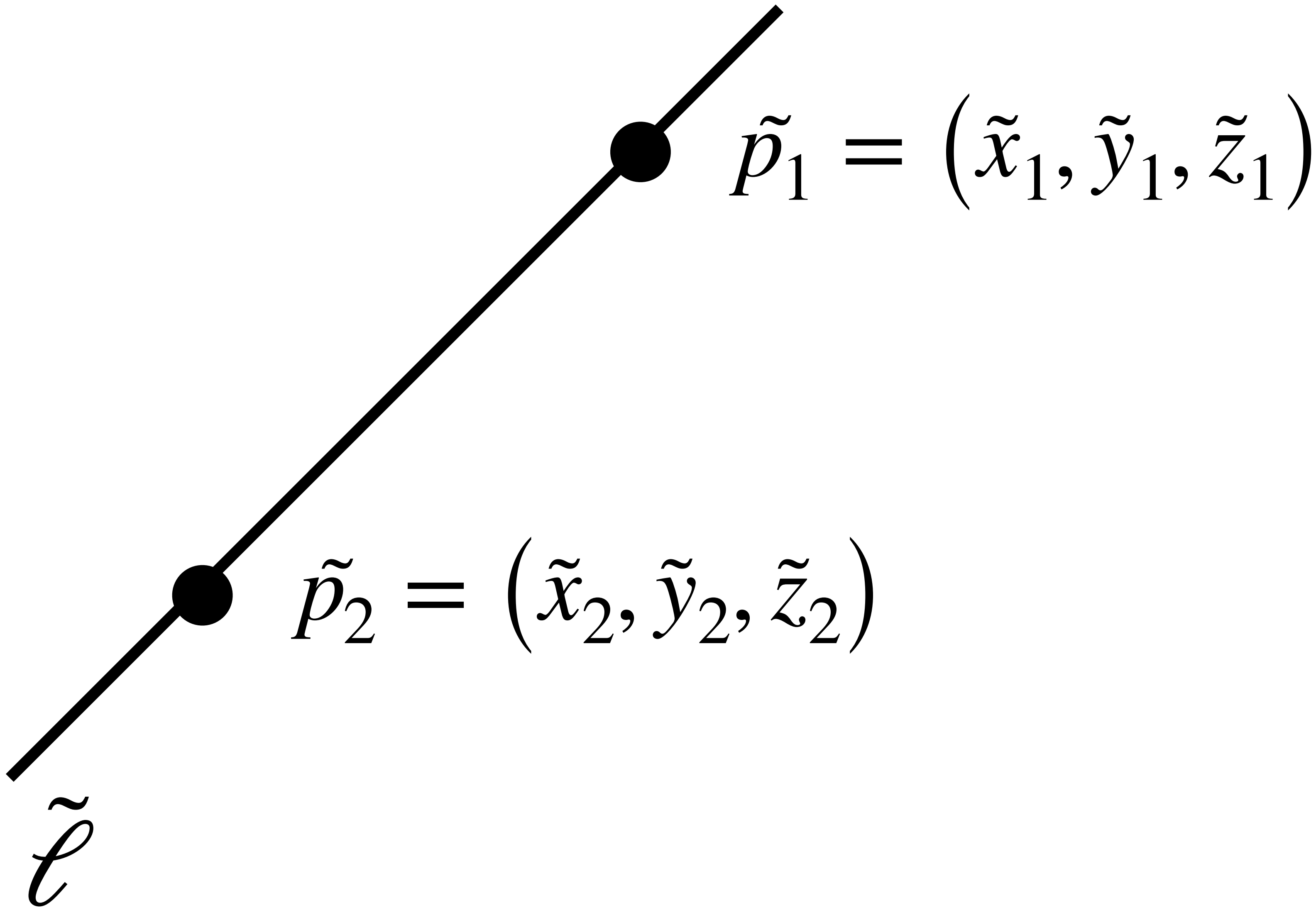


Point Equation of a Line:

$$\tilde{\ell}^T \tilde{p} = 0$$

$$l_1 \tilde{x} + l_2 \tilde{y} + l_3 \tilde{z} = 0$$

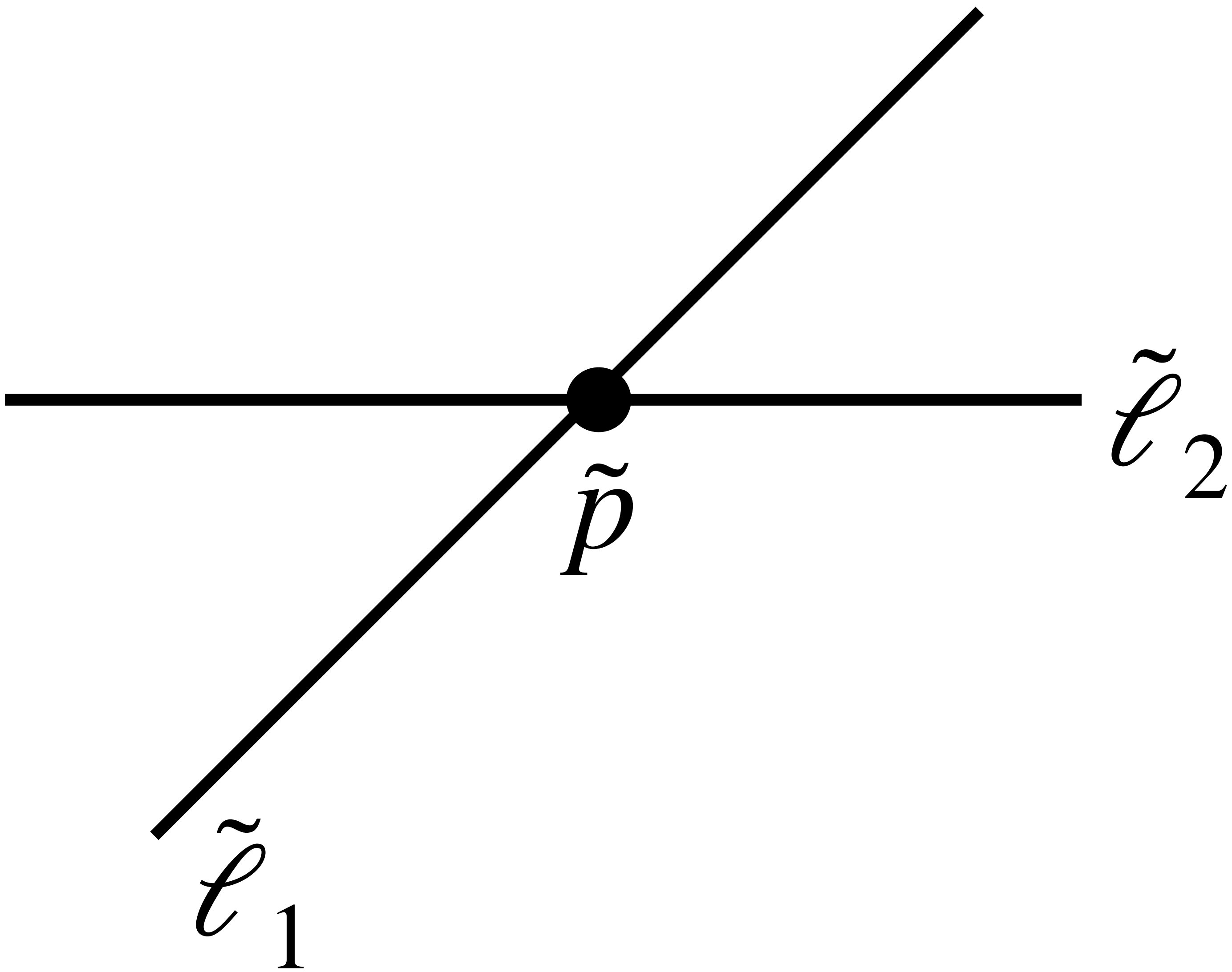
Slide credit: Peter Corke



Cross product of two points is a line:

$$\tilde{\ell} = \tilde{p}_1 \times \tilde{p}_2$$

Slide credit: Peter Corke

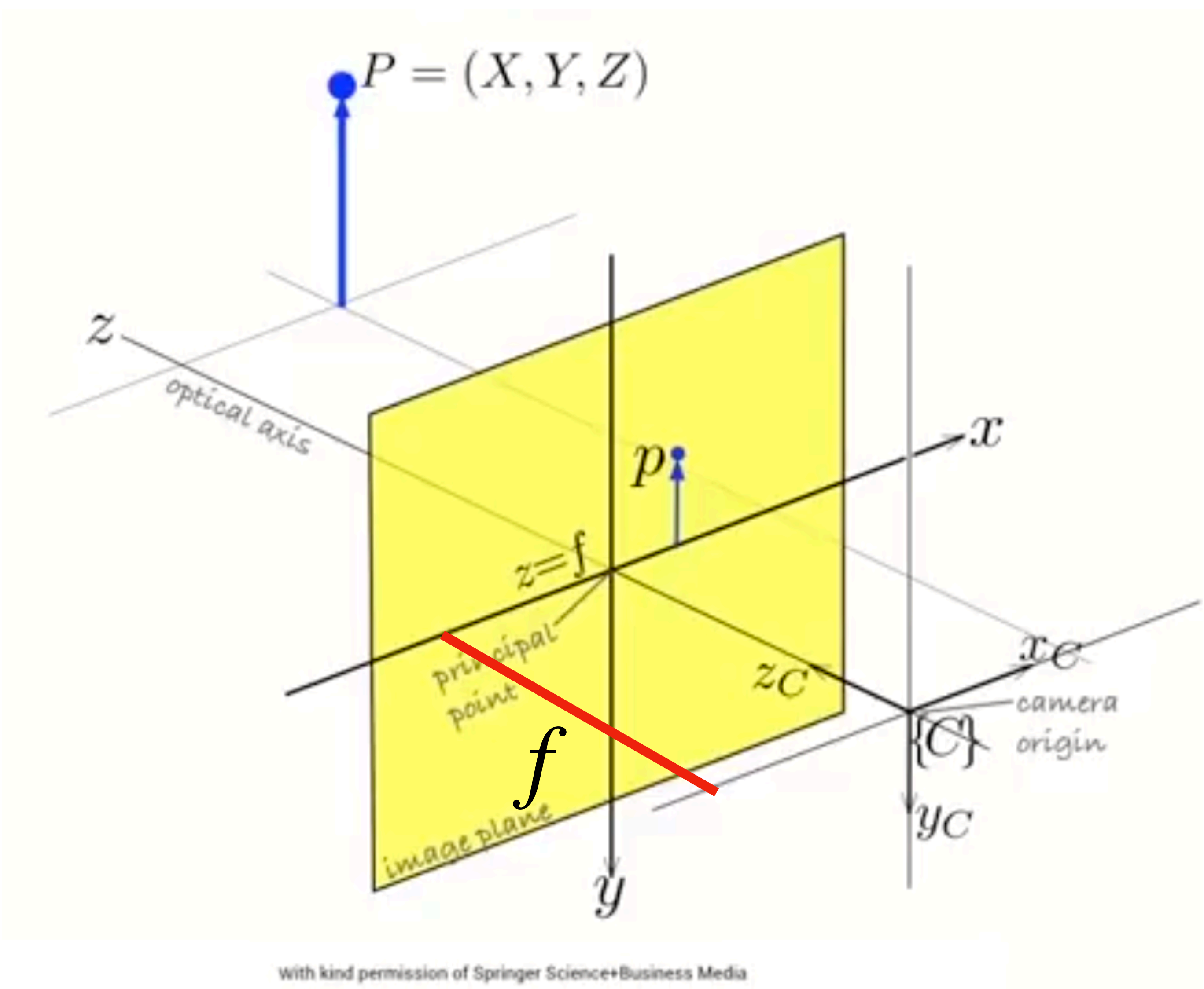


Cross product of two lines is a point:

$$\tilde{p} = \tilde{\ell}_1 \times \tilde{\ell}_2$$

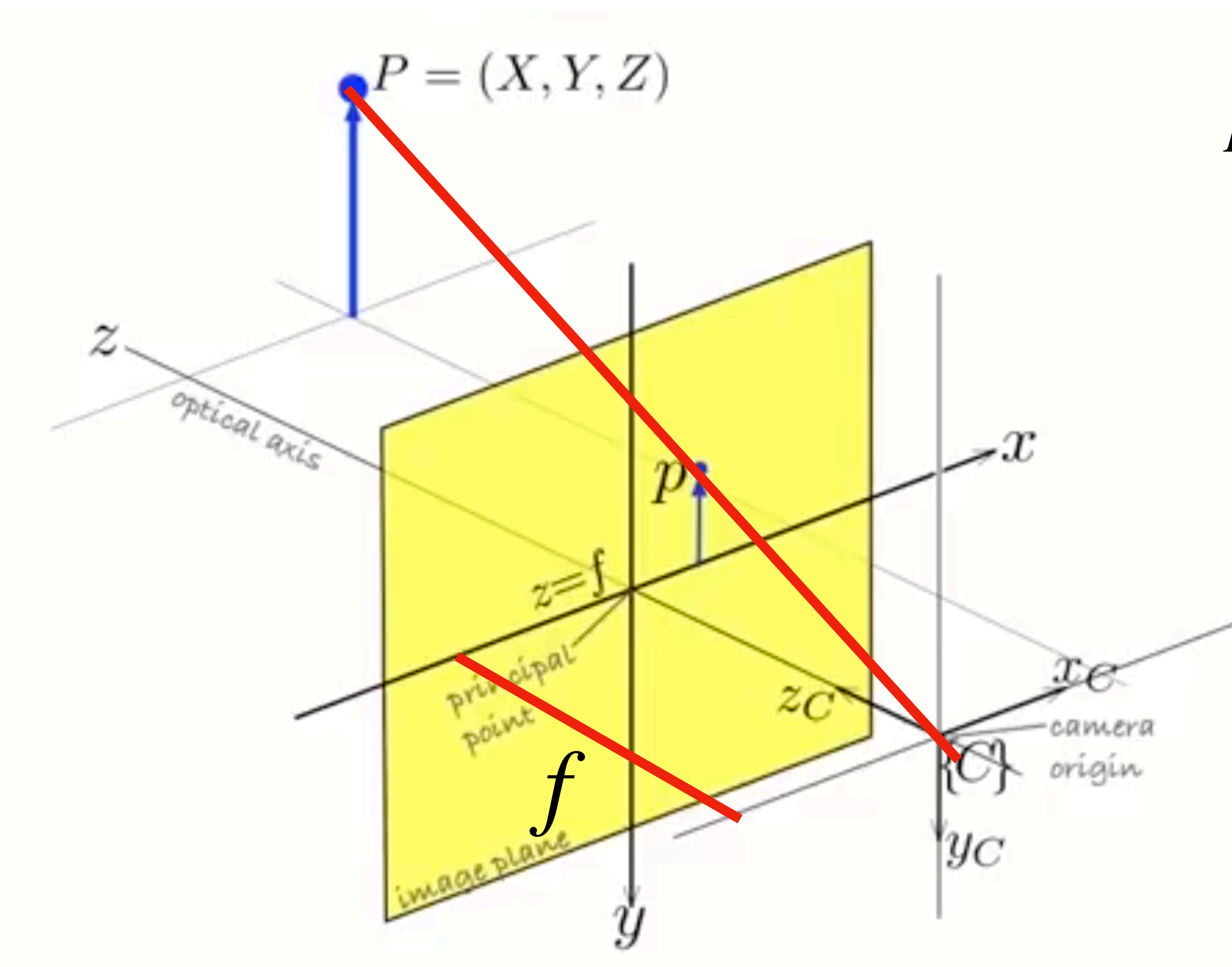
Slide credit: Peter Corke

Central Projection Model



Slide credit: Peter Corke

Central Projection Model

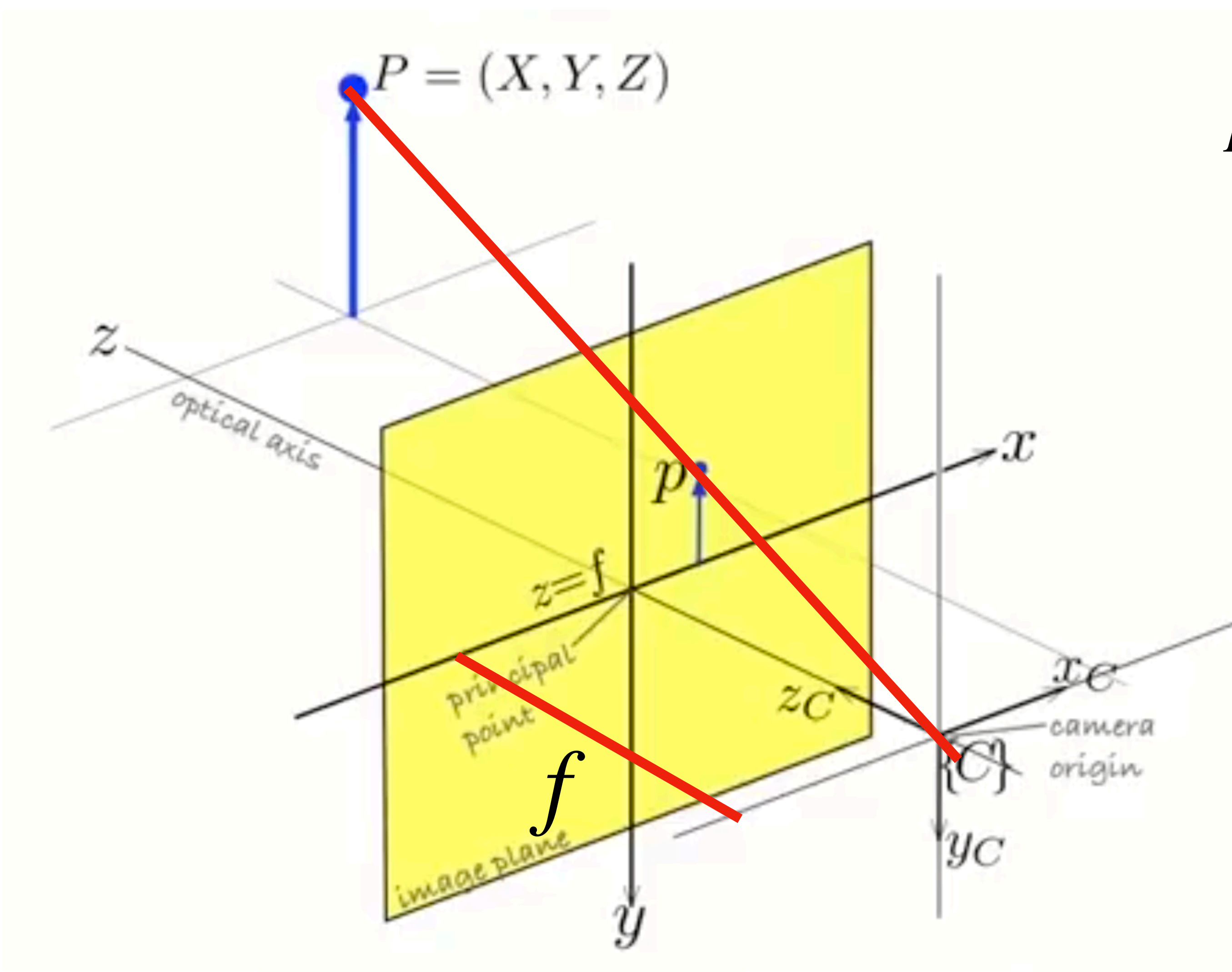


With kind permission of Springer Science+Business Media

$$p = \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Slide credit: Peter Corke

Central Projection Model



With kind permission of Springer Science+Business Media

$$p = \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

What if the camera moves?

Slide credit: Peter Corke

Review: 3D Transformations

3D translations

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} X + t_x \\ Y + t_y \\ Z + t_z \end{bmatrix}$$

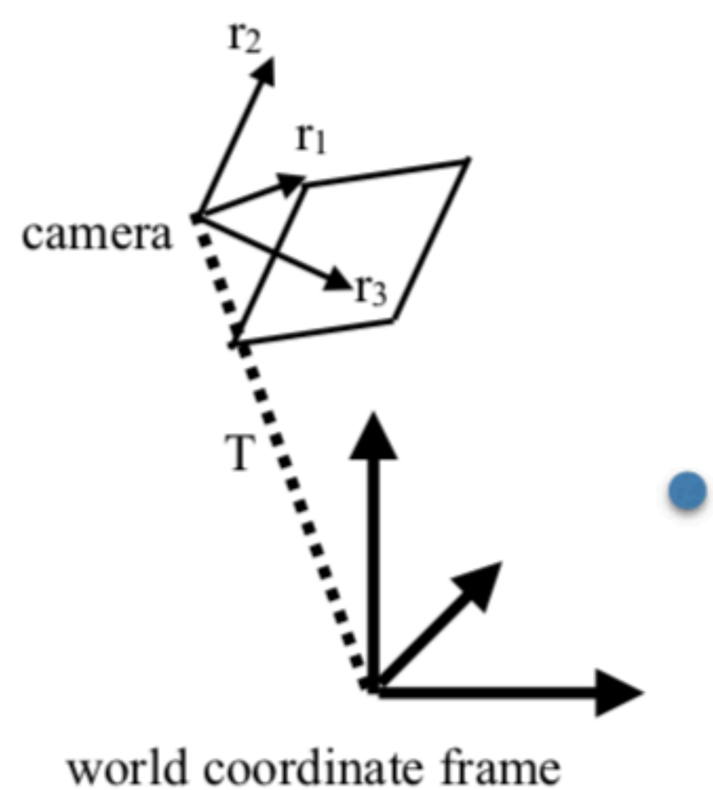
3D rotations

$$R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Change of Coordinate System

$$R\begin{bmatrix}X\\Y\\Z\end{bmatrix} + T = \begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\ r_{21} & r_{22} & r_{23} & t_y\\ r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$



Slide credit: Deva Ramanan

Camera Projection

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_x \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Camera Intrinsics

Camera Extrinsics

World Coordinates

Camera Matrix

Mapping points from the world to image coordinates is matrix multiplication in homogenous coordinates

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Scale Invariance

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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

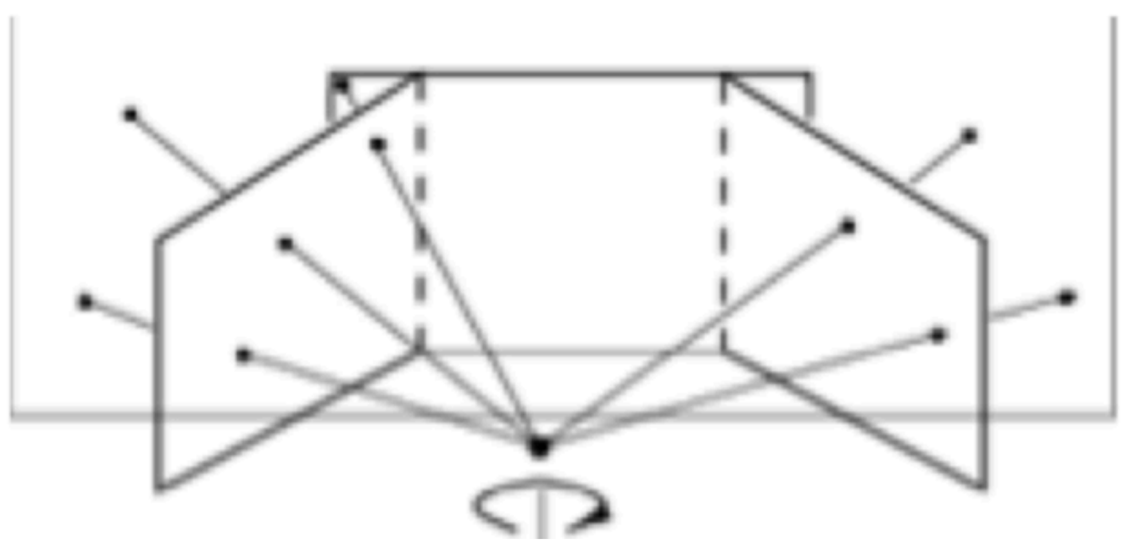
Normalized Camera Matrix

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Homography

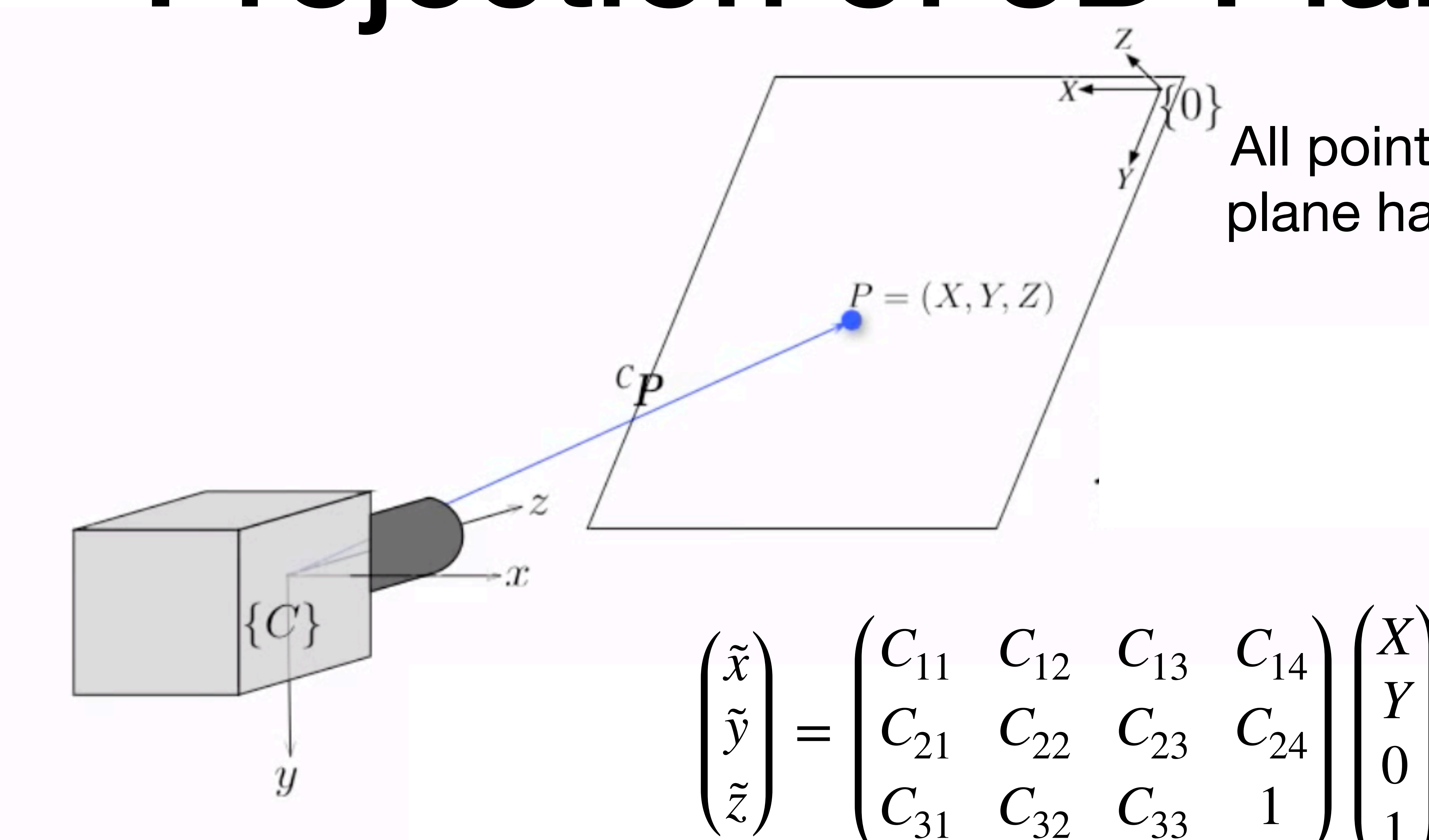
1. Models perspective effects for a planar scene

2. Models perspective effects from camera rotations



Slide credit: Deva Ramanan

Projection of 3D Plane

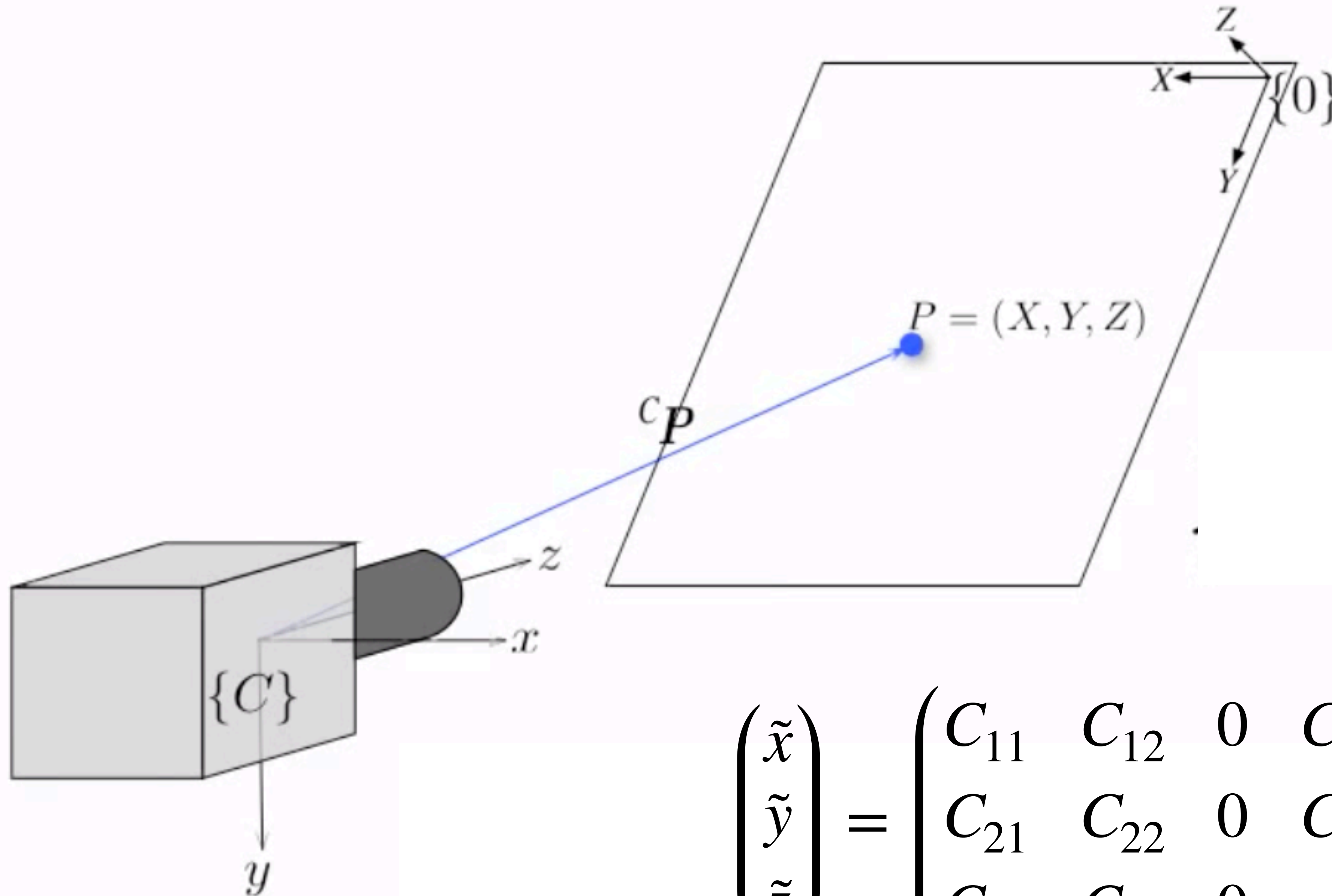


All points on the plane have Z = 0

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}$$

Slide credit: Peter Corke

Projection of 3D Plane

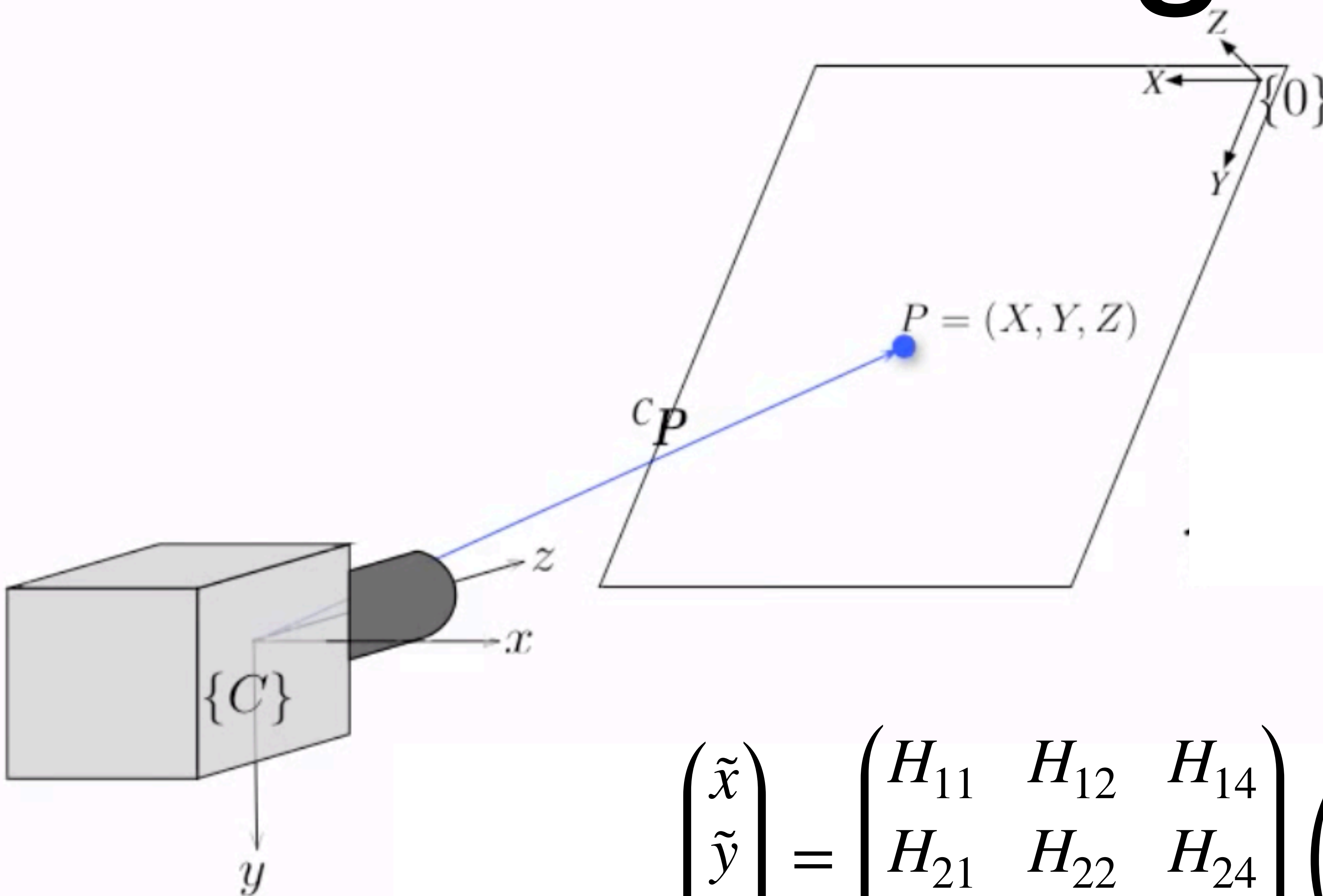


All points on the plane have Z = 0

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & 0 & C_{14} \\ C_{21} & C_{22} & 0 & C_{24} \\ C_{31} & C_{32} & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}$$

Slide credit: Peter Corke

Planar Homography

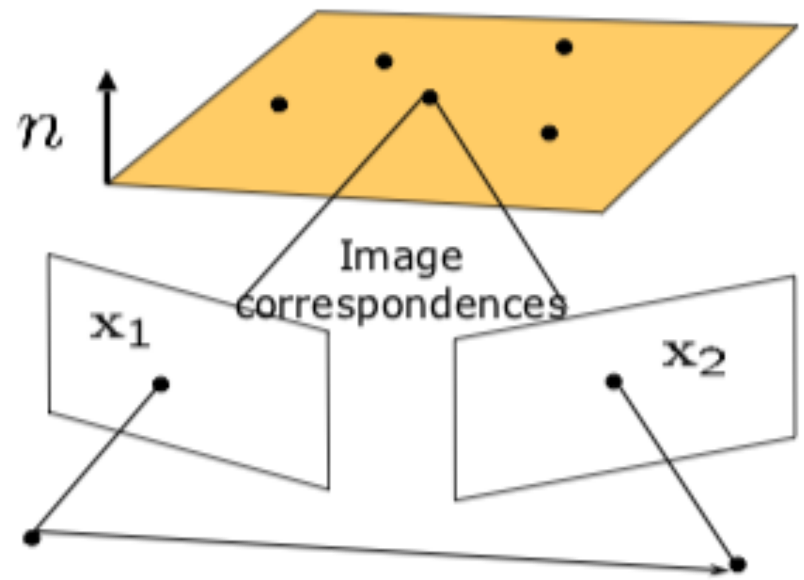


All points on the plane have Z = 0

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{14} \\ H_{21} & H_{22} & H_{24} \\ H_{31} & H_{32} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = H \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Slide credit: Peter Corke

Two-views of Plane

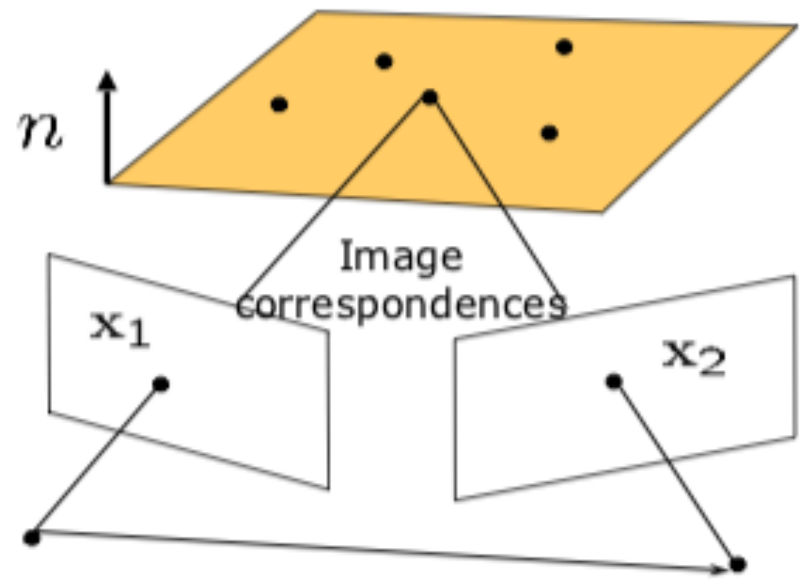


$$\begin{pmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{pmatrix} = H_1 \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \qquad \begin{pmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{pmatrix} = H_2 \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

If you know both H and (x1, y1),
what is (x2, y2)?

Slide credit: Deva Ramanan

Two-views of Plane

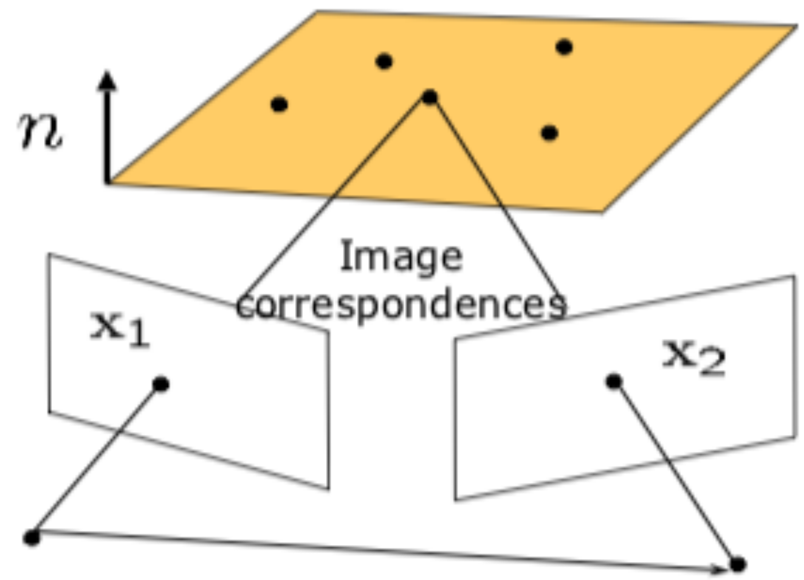


$$\begin{pmatrix}\tilde{x}_1\\ \tilde{y}_1\\ \tilde{z}_1\end{pmatrix} = H_1\begin{pmatrix}X\\ Y\\ 1\end{pmatrix} \qquad \begin{pmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{pmatrix} = H_2\begin{pmatrix}X\\ Y\\ 1\end{pmatrix}$$

$$\begin{pmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{pmatrix} = H_2 H_1^{-1}\begin{pmatrix}\tilde{x}_1\\ \tilde{y}_1\\ \tilde{z}_1\end{pmatrix}$$

Slide credit: Deva Ramanan

Estimating Homography



How many corresponding points do you need to estimate H?

$$\begin{pmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{pmatrix} = H \begin{pmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{pmatrix}$$

Slide credit: Deva Ramanan

Estimating Homography (details)

$$\begin{bmatrix} \mathrm{w}x_i' \\ \mathrm{w}y_i' \\ \mathrm{w} \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$x_i' = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$y_i' = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$\begin{aligned} x_i'(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{00}x_i + h_{01}y_i + h_{02} \\ y_i'(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{10}x_i + h_{11}y_i + h_{12} \end{aligned}$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i'x_i & -y_i'y_i & -y_i' \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Slide credit: Antonio Torralba

Estimating Homography (details)

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

A ($2n \times 9$)   h (9)   0 ($2n$)

Defines a least squares problem: minimize $\|\mathbf{Ah} - \mathbf{0}\|^2$

- Since h is only defined up to scale, solve for unit vector $\hat{h}$
- Solution: $\hat{h}$ = eigenvector of $A^T A$ with smallest eigenvalue
- Works with 4 or more points

Slide credit: Antonio Torralba

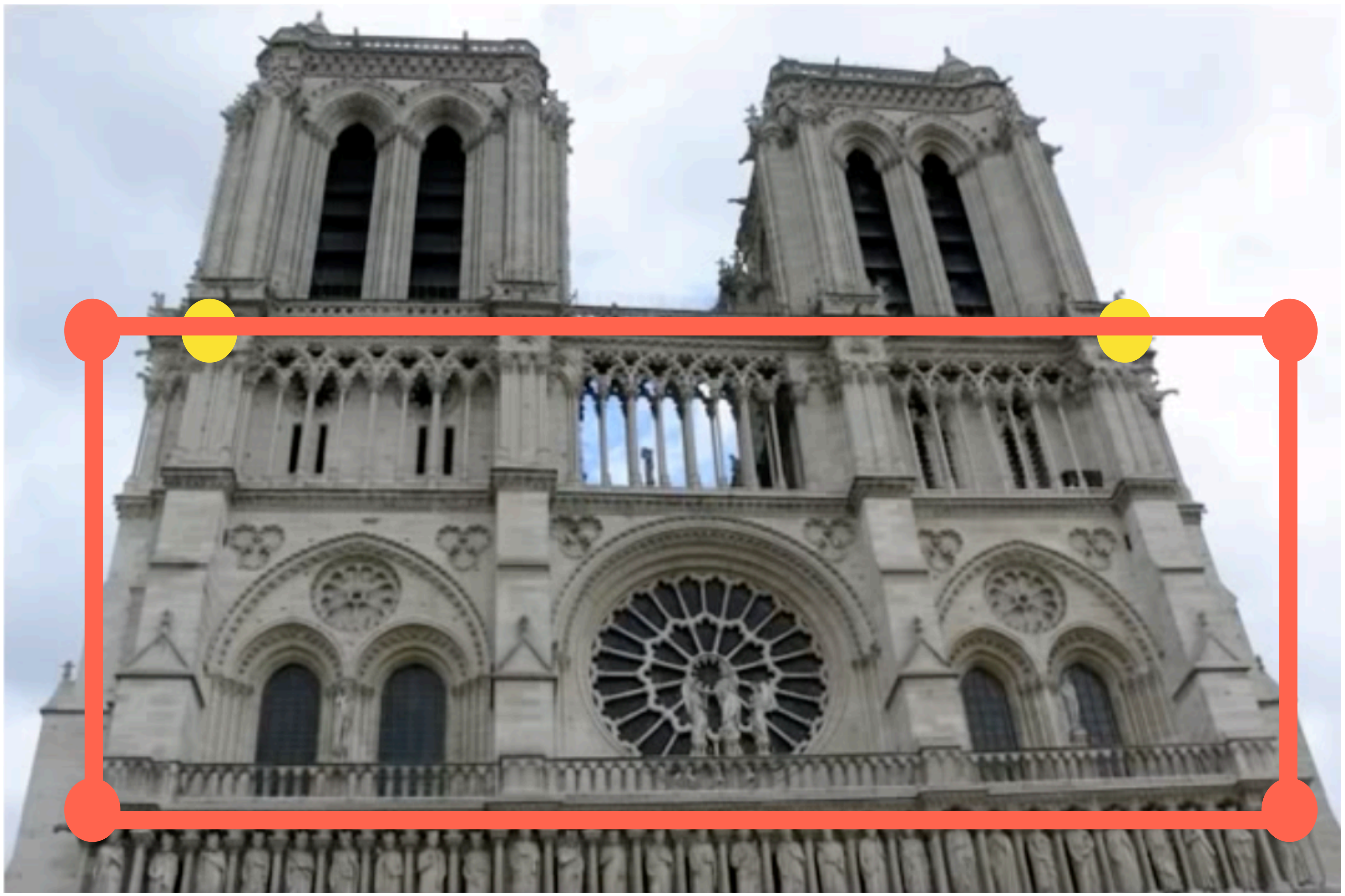
Rectification

Slide credit: Peter Corke

Rectification

Slide credit: Peter Corke

Rectification

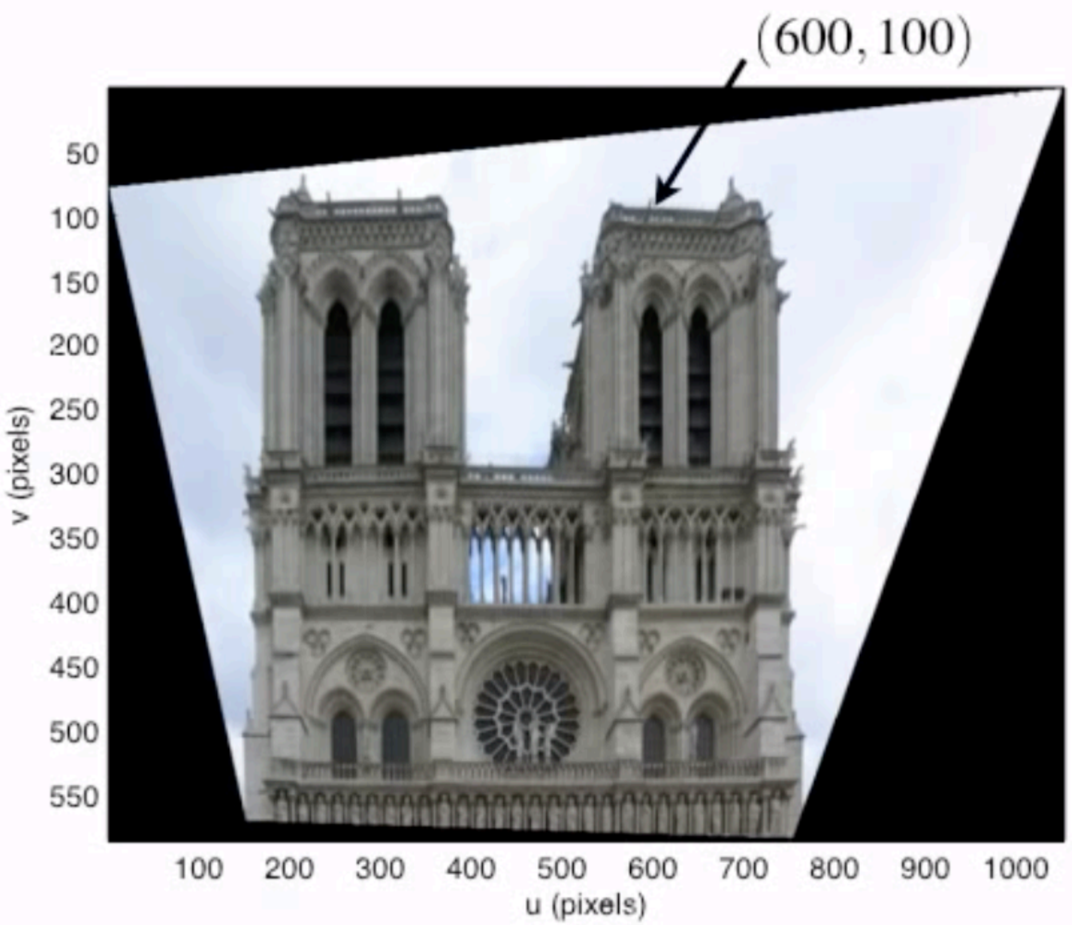


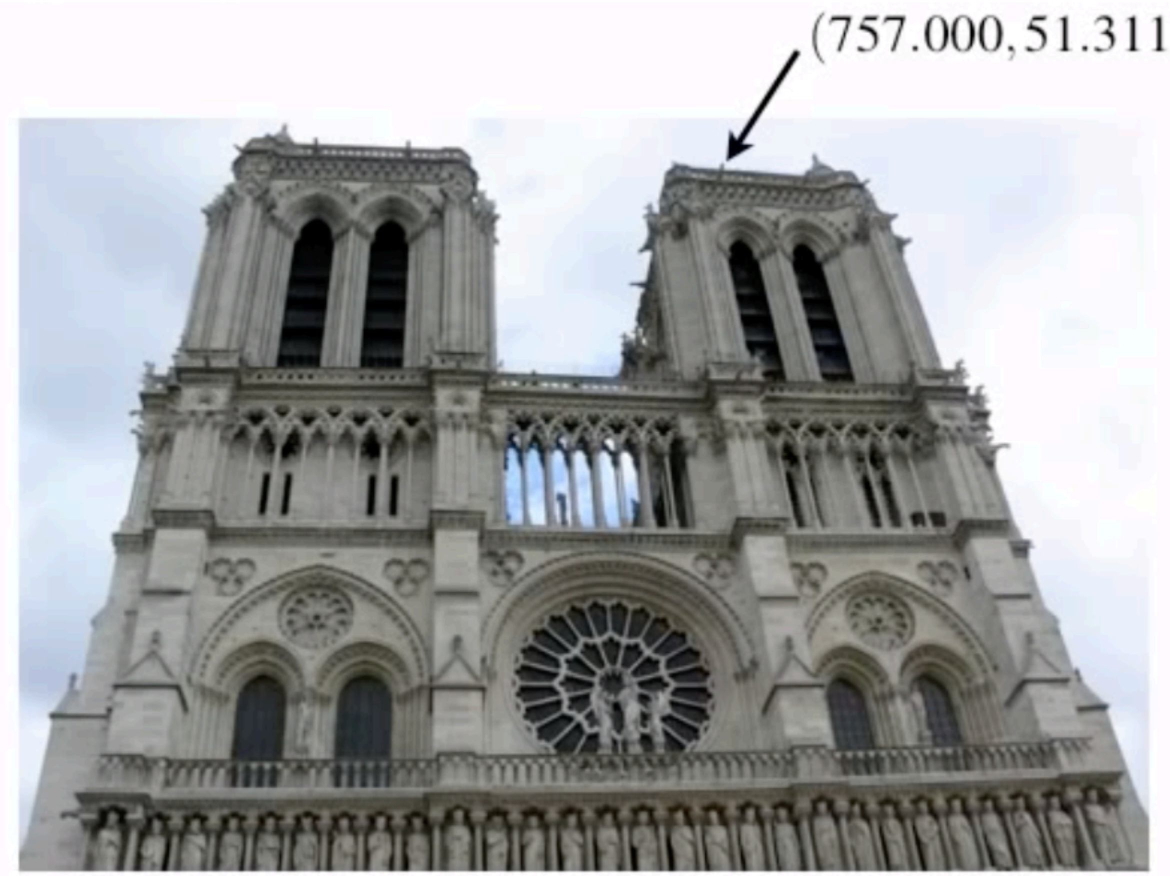
Slide credit: Peter Corke

Rectification

Slide credit: Peter Corke

Warping



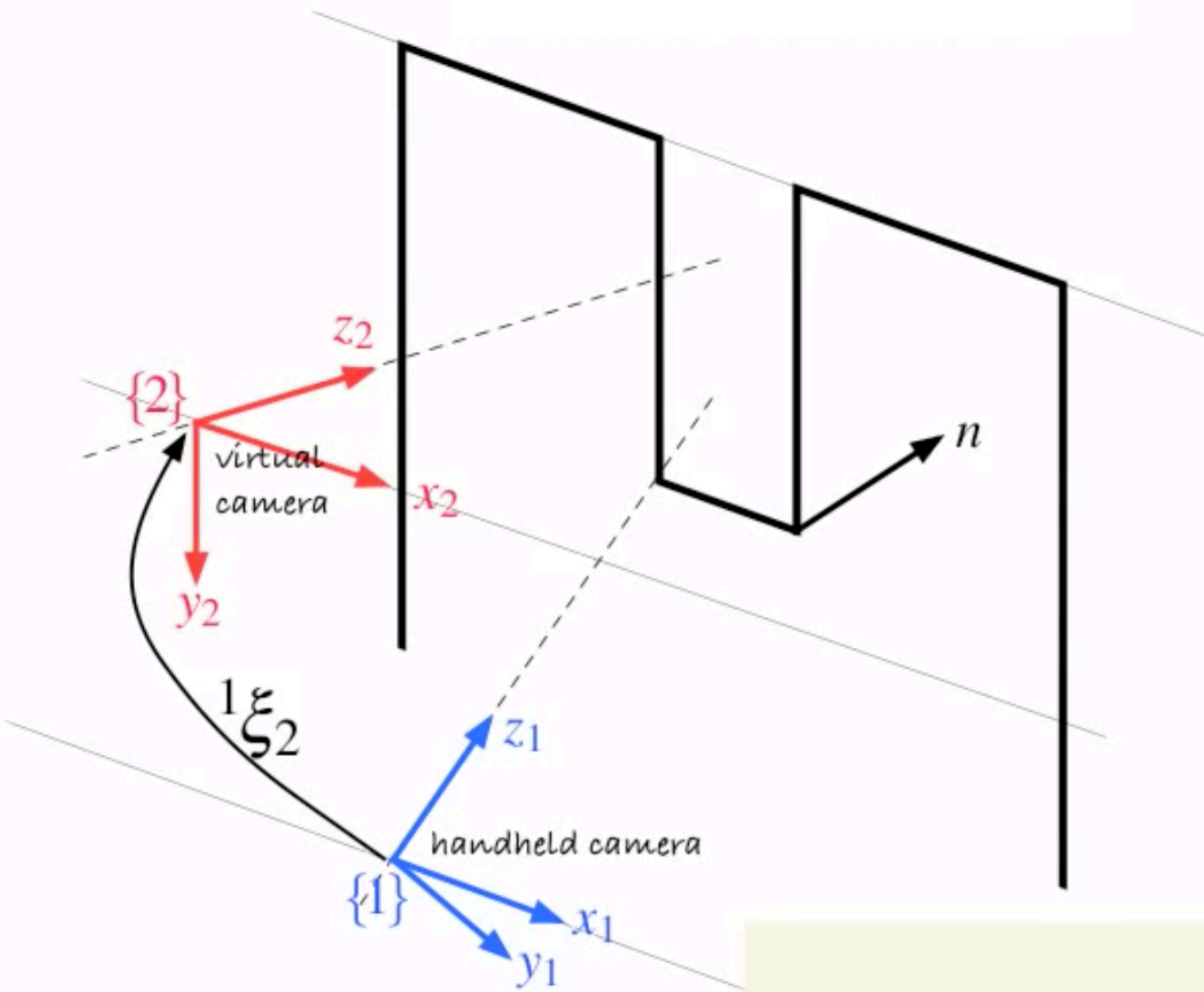


$$\mathbf{H}\begin{pmatrix}600\\100\\1\end{pmatrix} = \begin{pmatrix}741.86\\50.285\\0.98\end{pmatrix}$$

$$u = 757.000, v = 51.311$$

Slide credit: Peter Corke

Virtual Camera



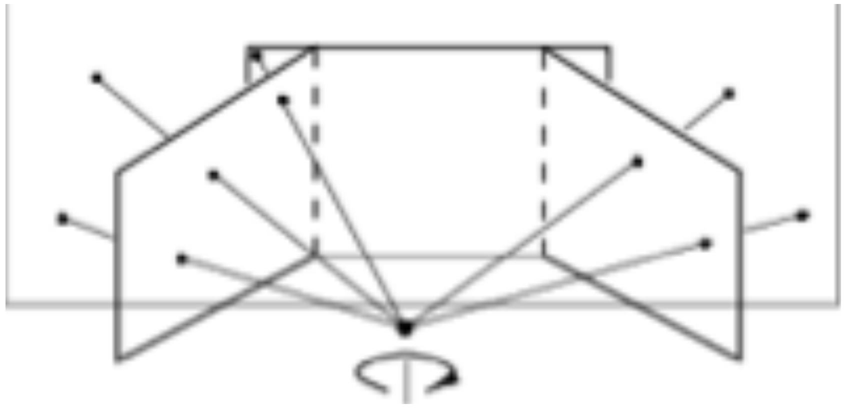
$$\boldsymbol{H}_E \simeq \boldsymbol{R} + \frac{\boldsymbol{t}}{d} n^T$$

Slide credit: Peter Corke

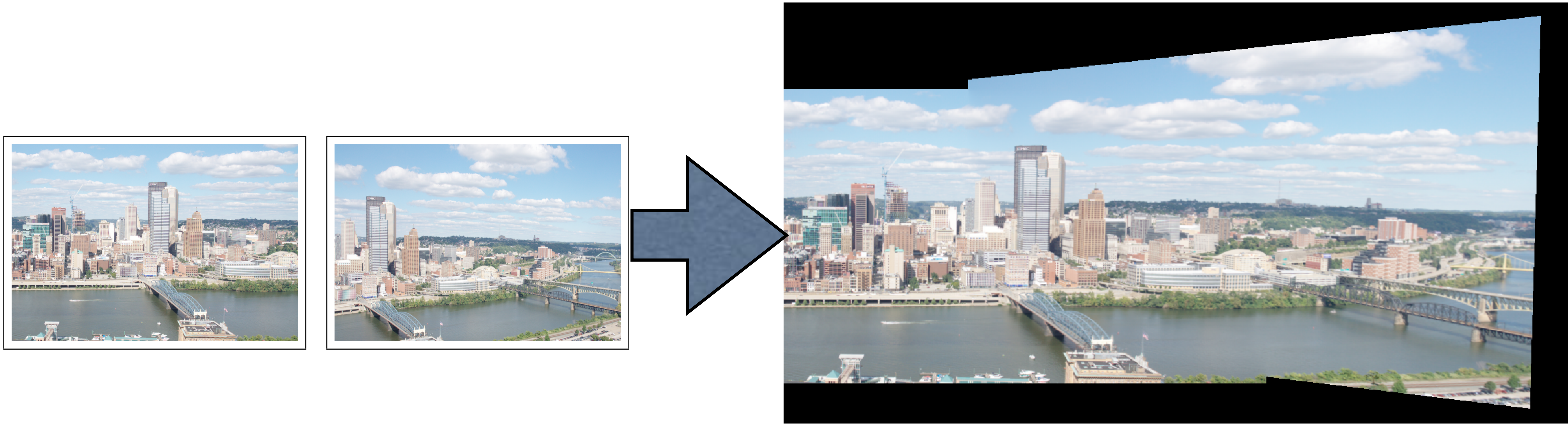
Panoramas

Slide credit: Olga Russakovsky

Special case of 2 views: rotations about camera center

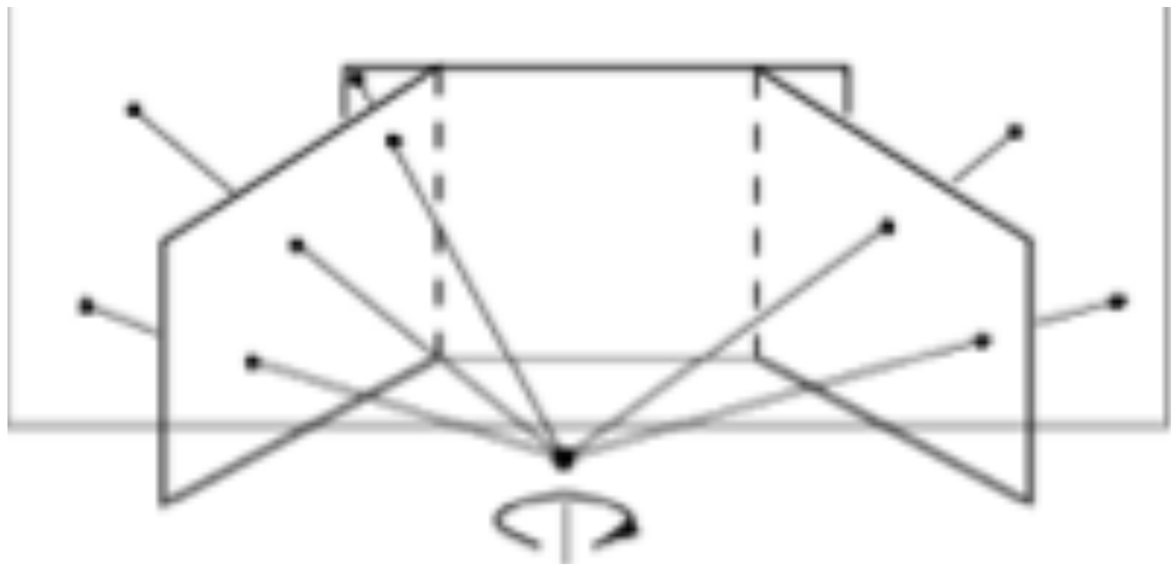


Can be modeled as planar transformations, regardless of scene geometry!



Slide credit: Deva Ramanan

Derivation



Relation between 3D camera coordinates: $\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = R \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$

K_2

3D->2D projection: $\lambda_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix}$

$\cdots$

Combining both: $\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 R K_1^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$

Slide credit: Deva Ramanan

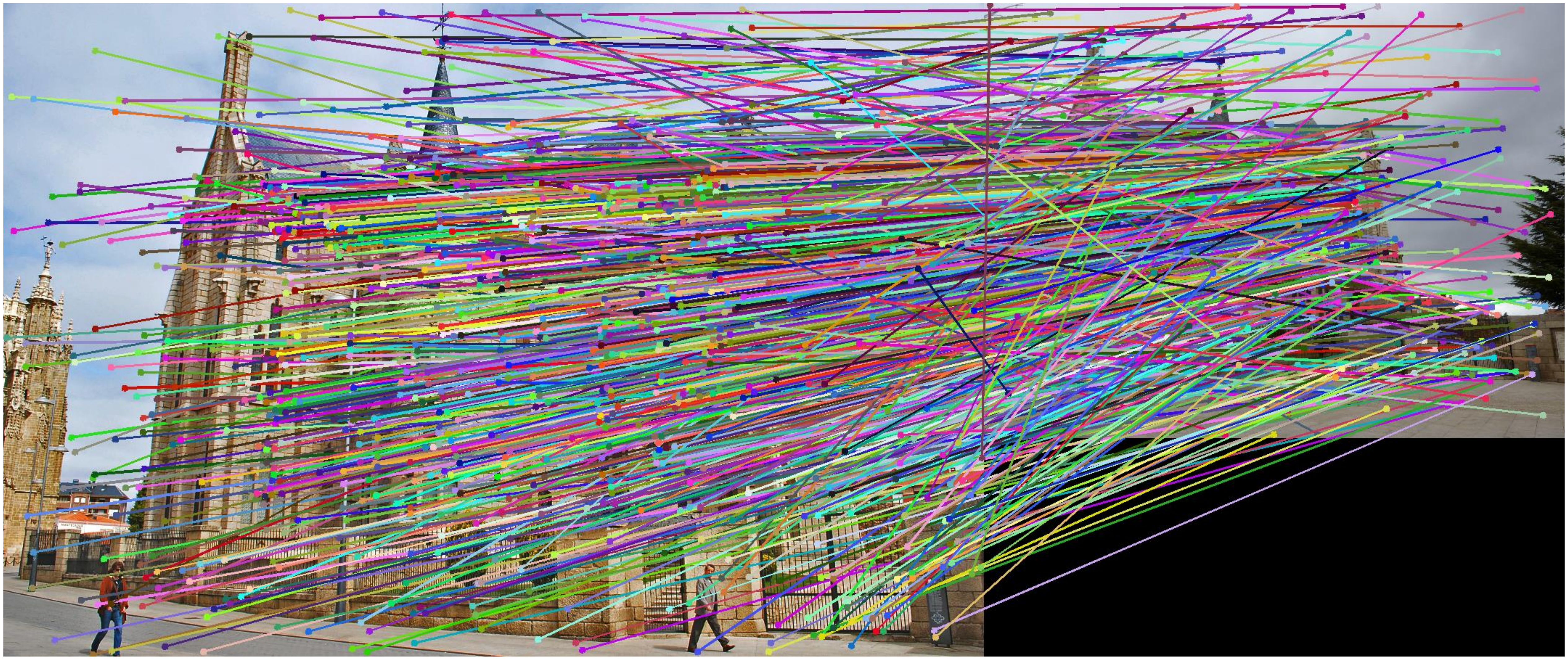
Take-home points for homographies

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- If camera rotates about its center, then the images are related by a homography irrespective of scene depth.
- If the scene is planar, then images from any two cameras are related by a homography.
- Homography mapping is a 3x3 matrix with 8 degrees of freedom.

Slide credit: Deva Ramanan

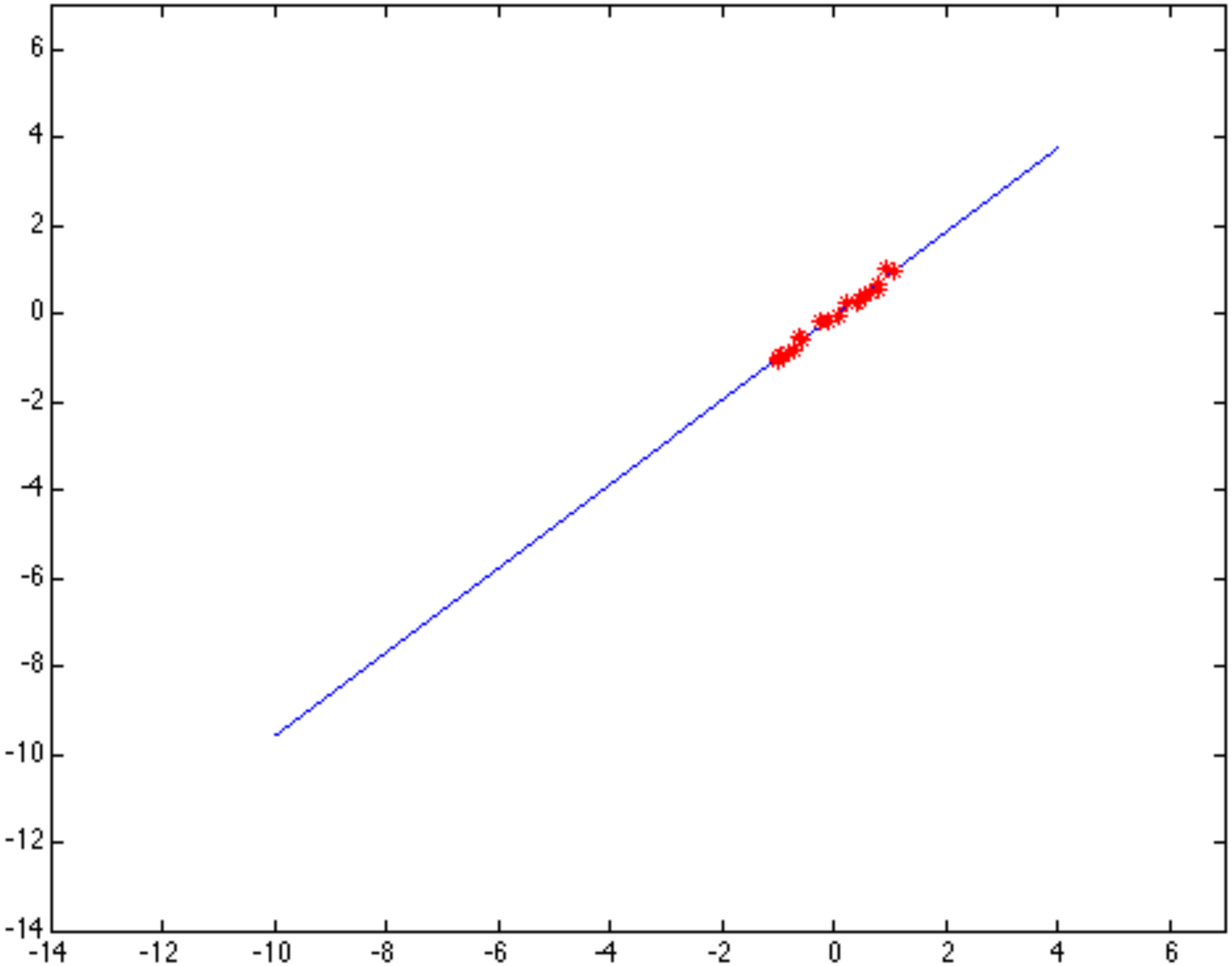
VLFeat's 800 most confident matches among 10,000+ local features.



Which matches should we use to estimate homography?

Least squares: Robustness to noise

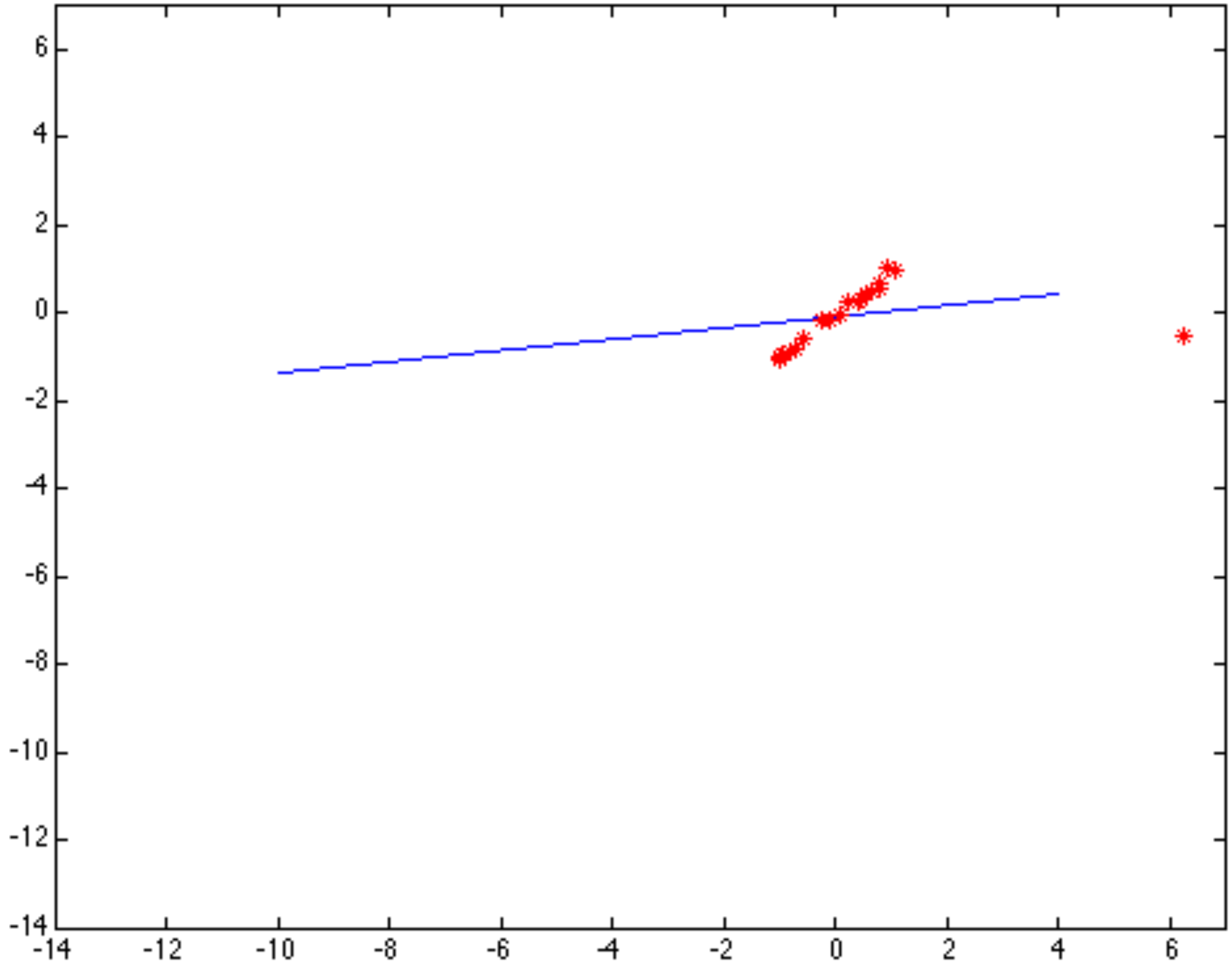
- Least squares fit to the red points:



Slide credit: James Hays

Least squares: Robustness to noise

- Least squares fit with an outlier:



Problem: squared error heavily penalizes outliers

Slide credit: James Hays

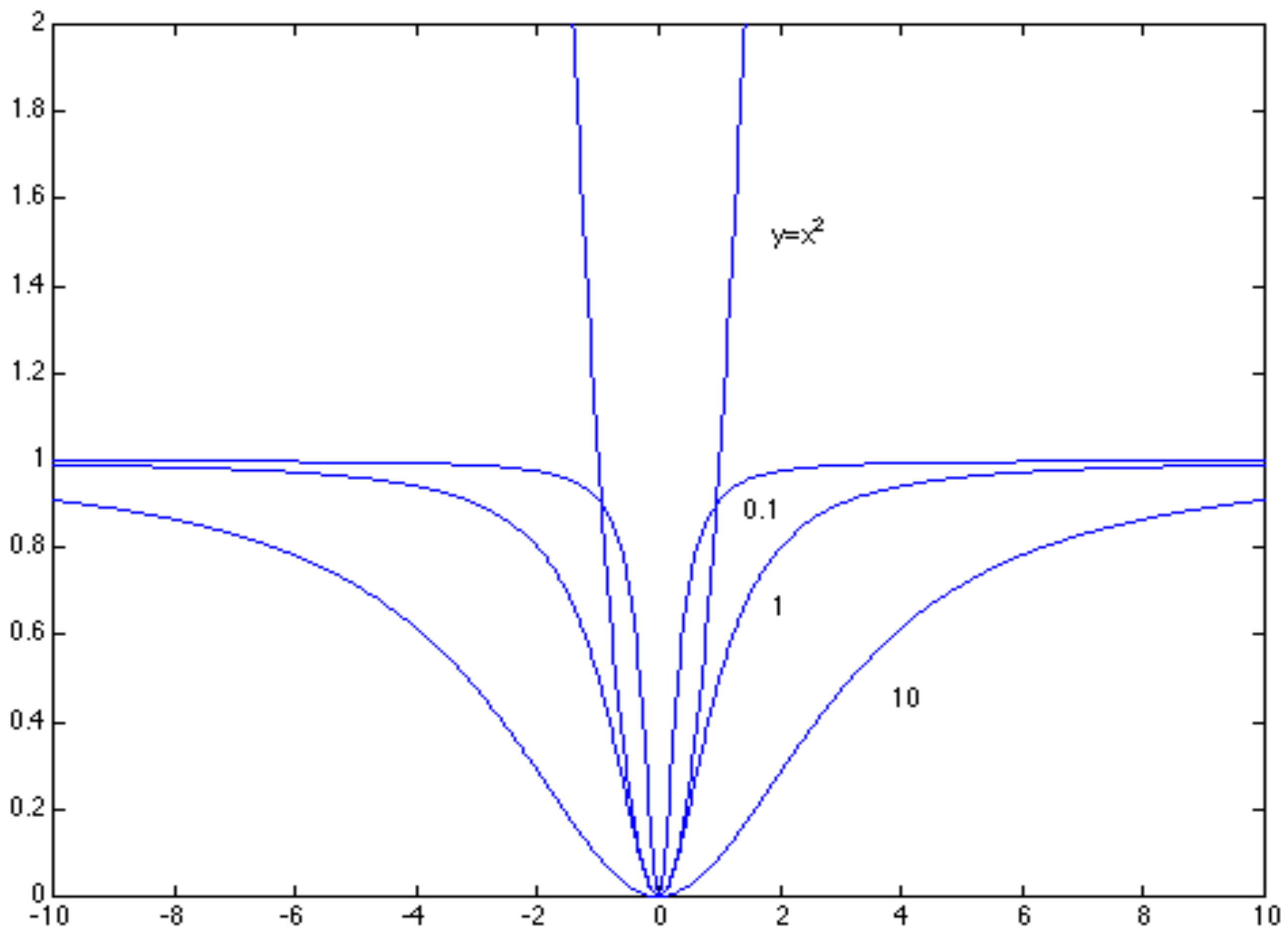
Robust least squares (to deal with outliers)

General approach:

minimize $$\sum_i \rho\left(u_i\left(x_i,\theta\right);\sigma\right) \qquad u^2 = \sum_{i=1}^{n}(y_i - mx_i - b)^2$$

$u_i\,(x_i,\,\theta)$ – residual of i[th] point w.r.t. model parameters θ

ρ – robust function with scale parameter σ



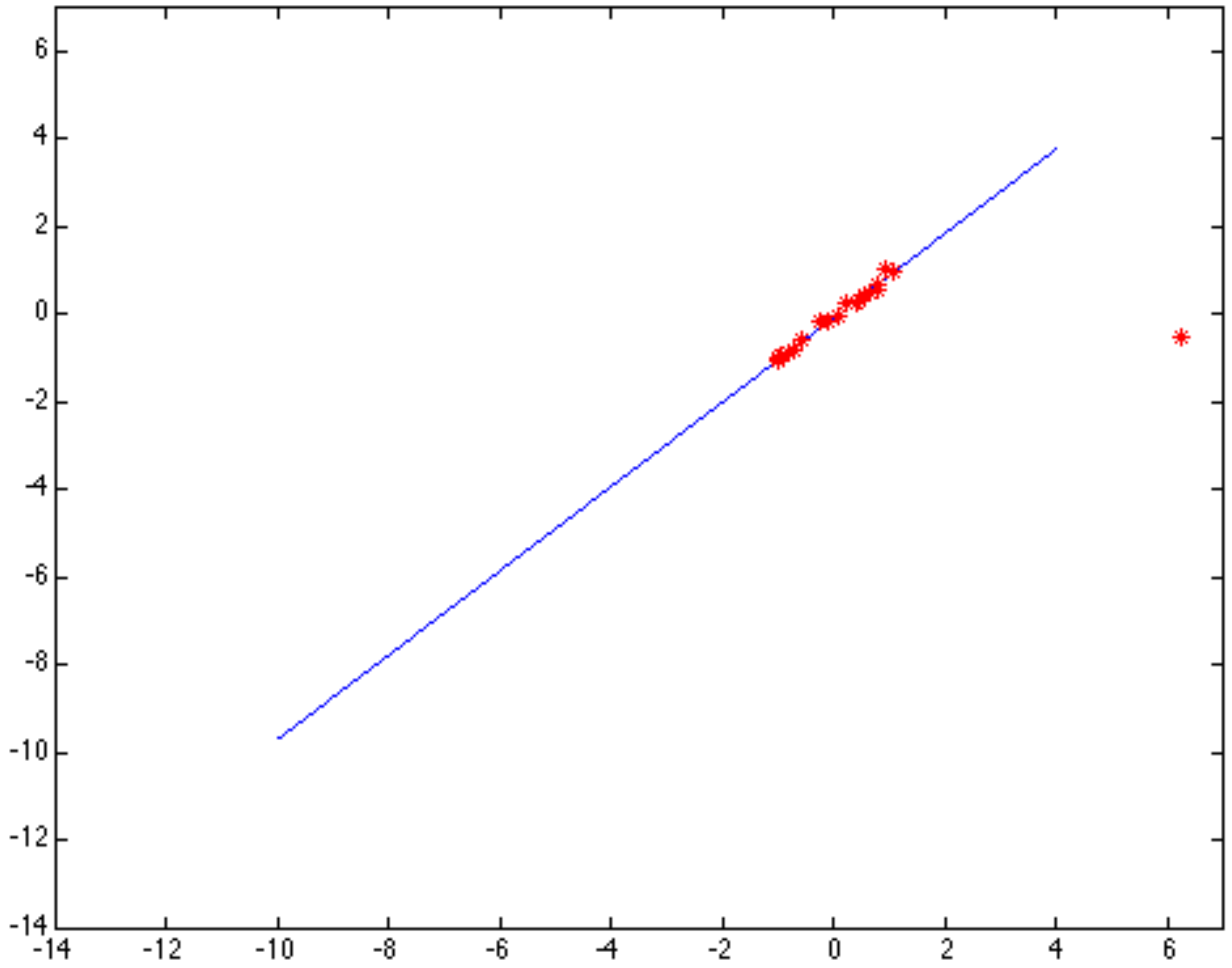
The robust function ρ

- Favors a configuration with small residuals
- Constant penalty for large residuals

$$\rho(u;\sigma) = \frac{u^2}{\sigma^2 + u^2}$$

Slide from S. Savarese

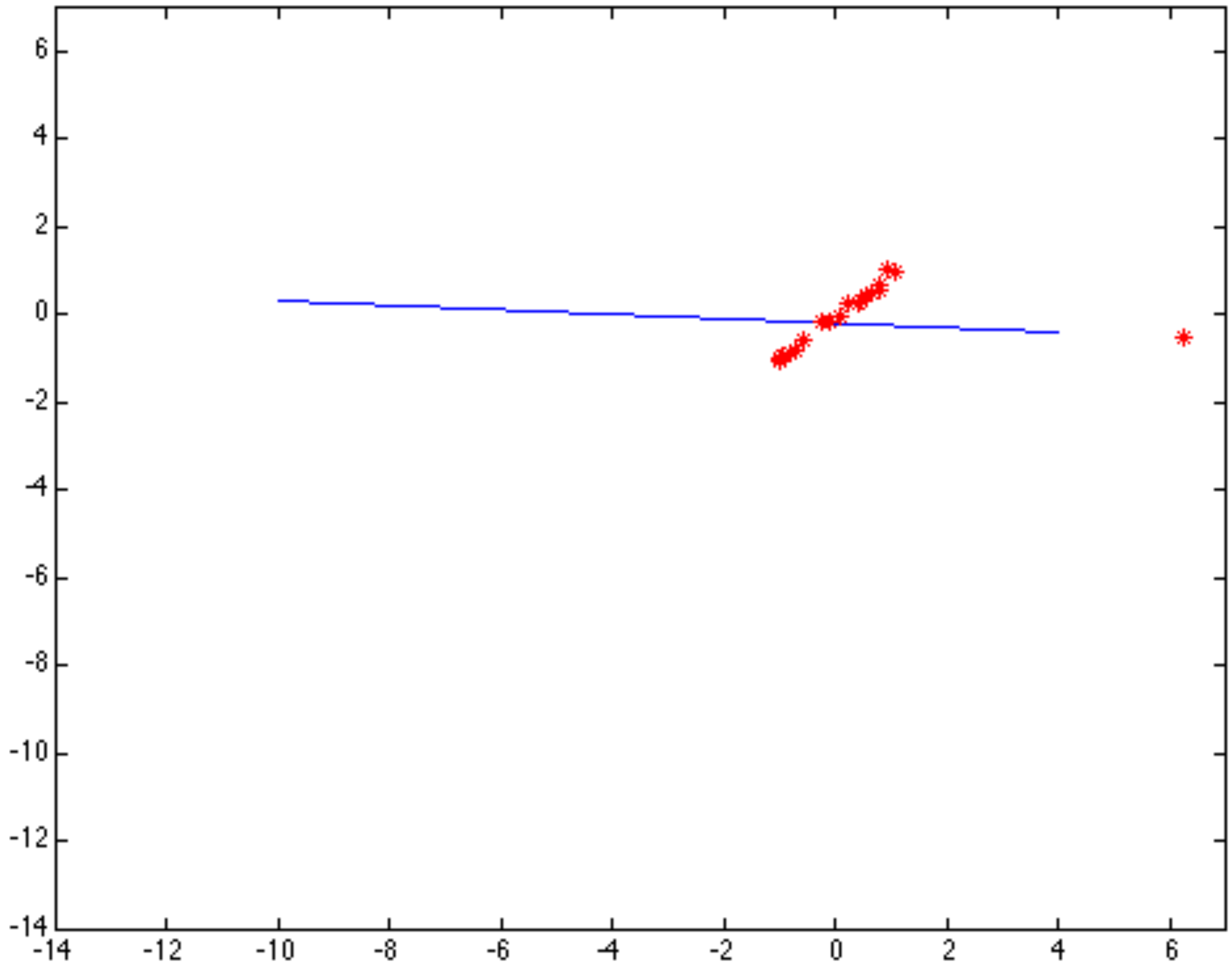
Choosing the scale: Just right



The effect of the outlier is minimized

Slide credit: James Hays

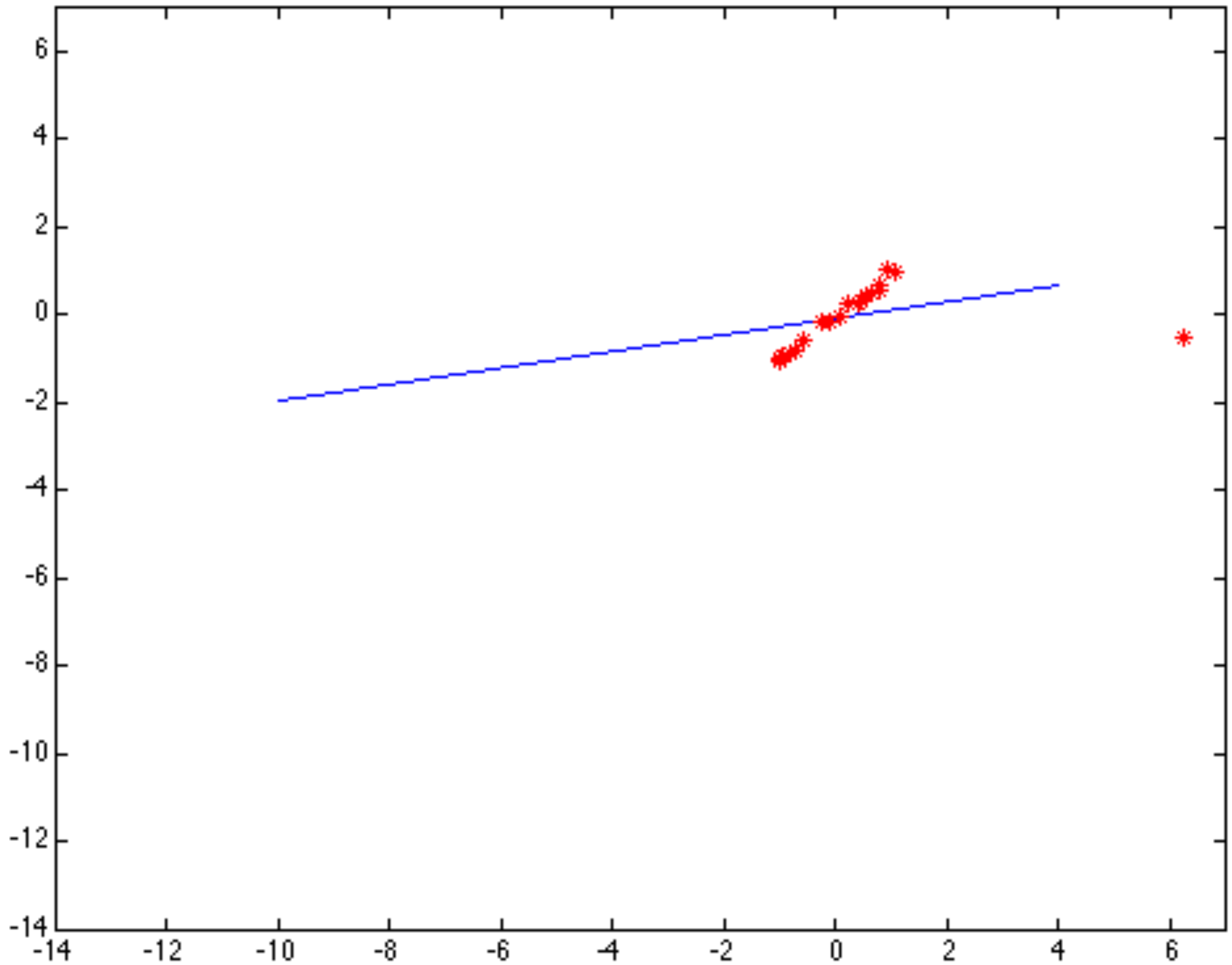
Choosing the scale: Too small



The error value is almost the same for every point and the fit is very poor

Slide credit: James Hays

Choosing the scale: Too large



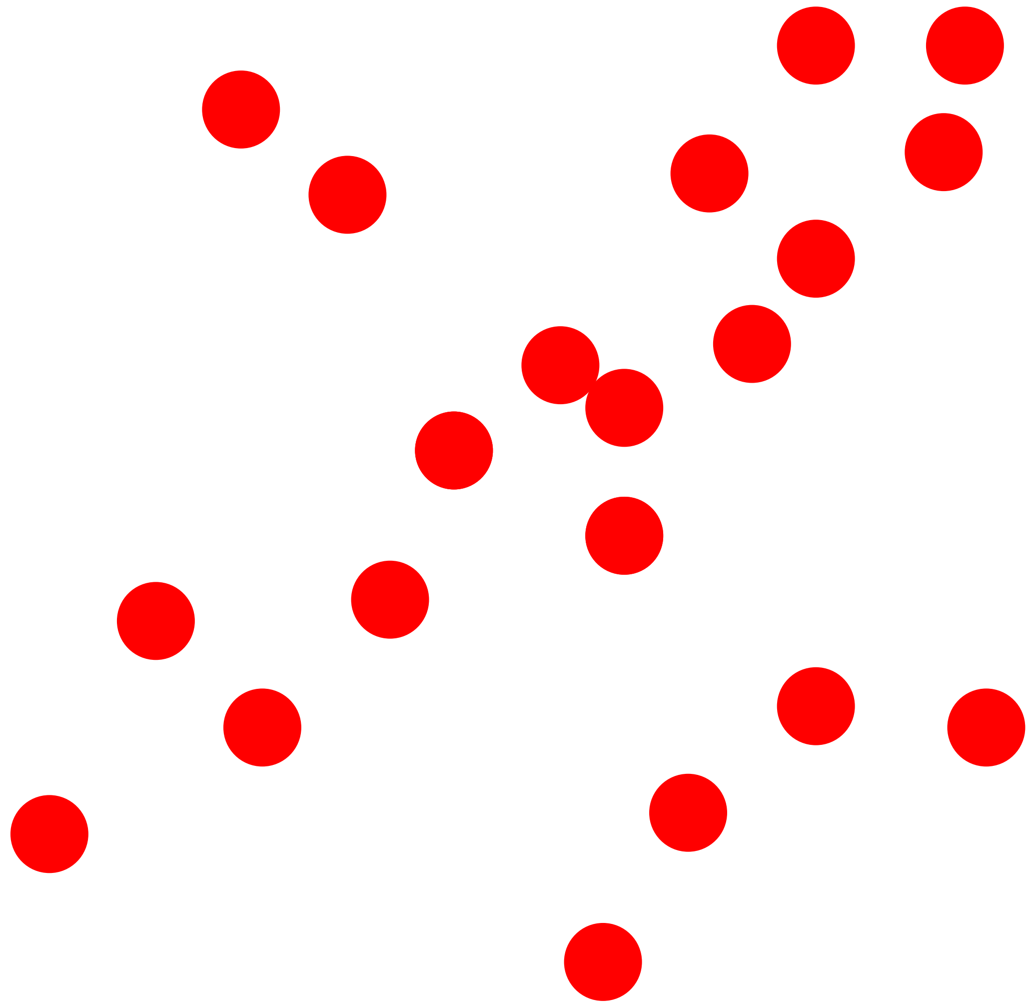
Behaves much the same as least squares

Slide credit: James Hays

RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in '81.

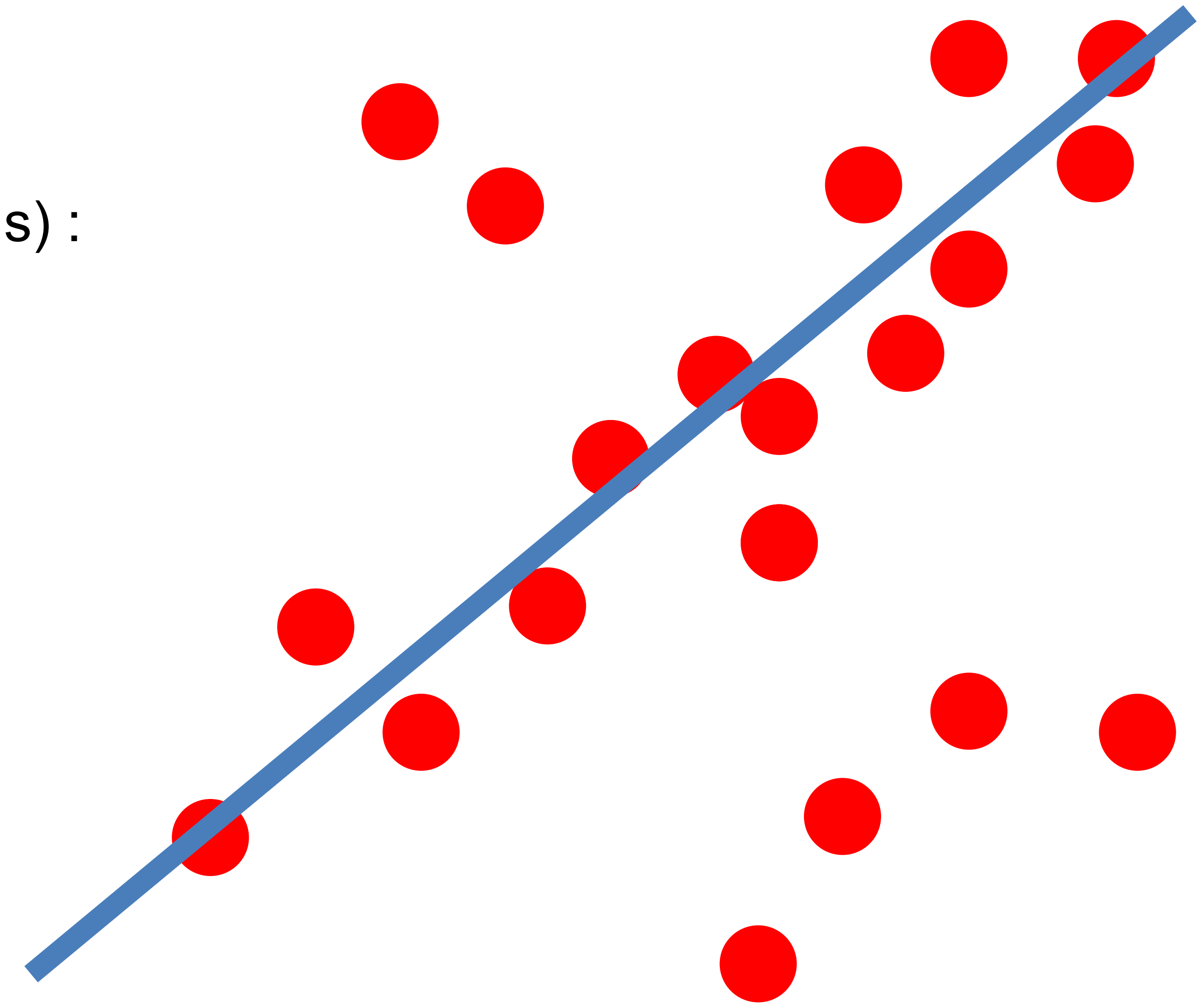


Slide credit: James Hays

RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in ‘81.



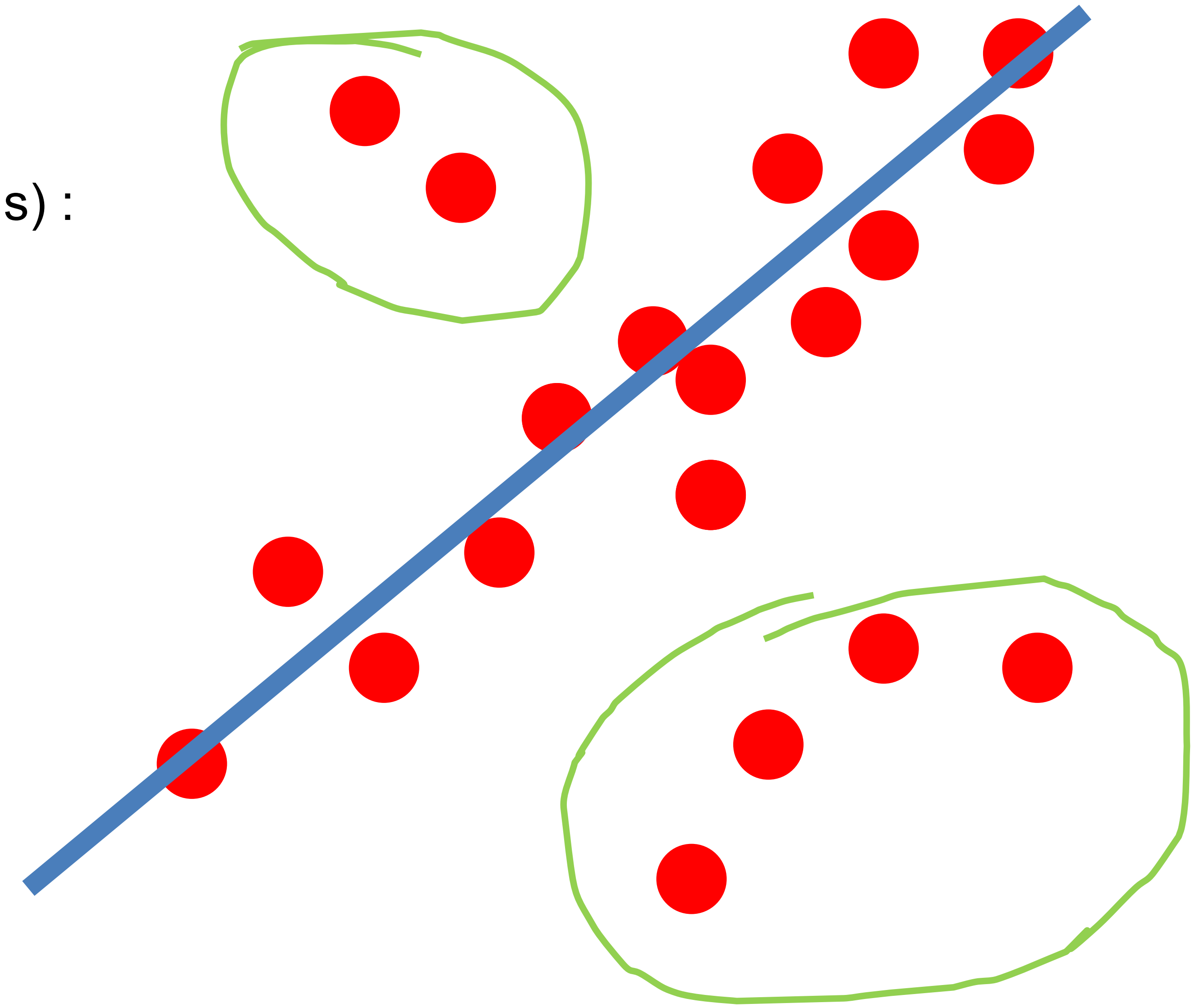
This data is noisy, but we expect a good fit to a known model.

Slide credit: James Hays

RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in '81.



This data is noisy, but we expect a good fit to a known model.

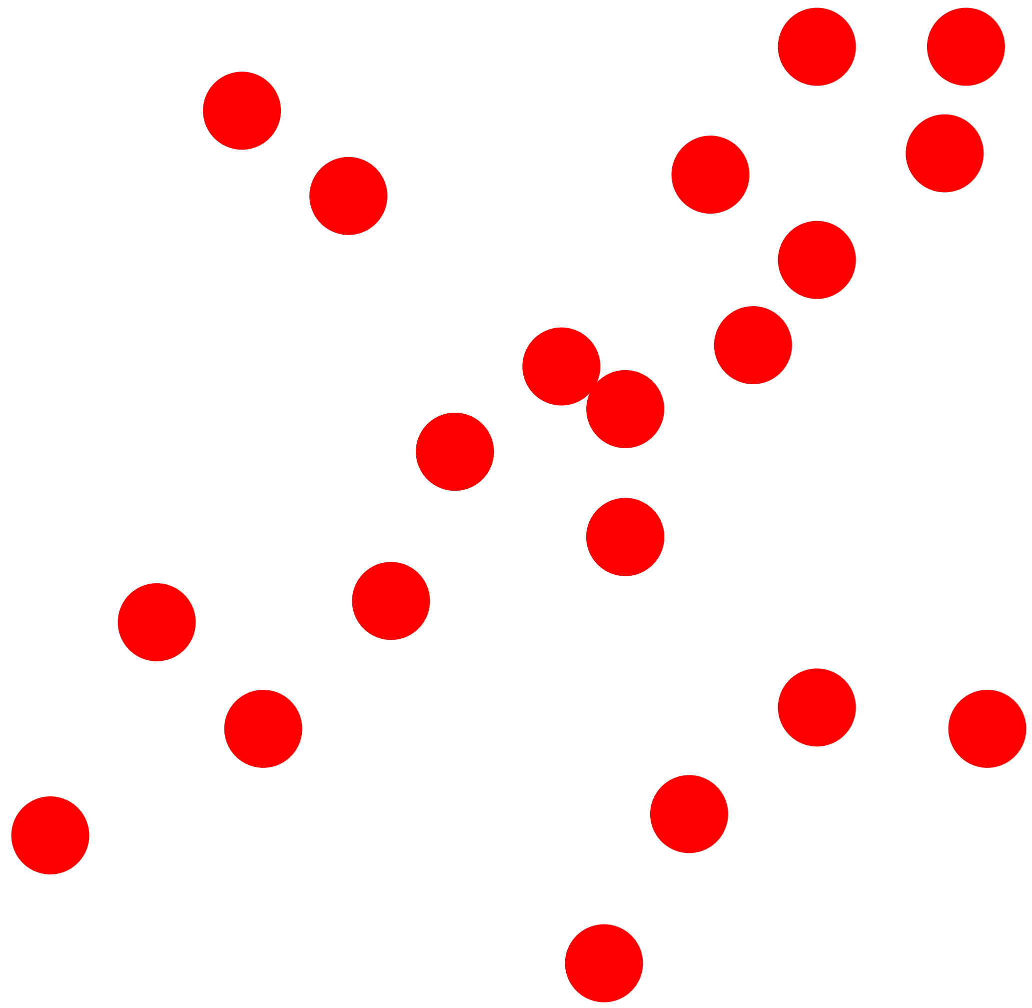
Here, we expect to see a line, but least-squares fitting will produce the wrong result due to strong outlier presence.

Slide credit: James Hays

RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in '81.



Algorithm:

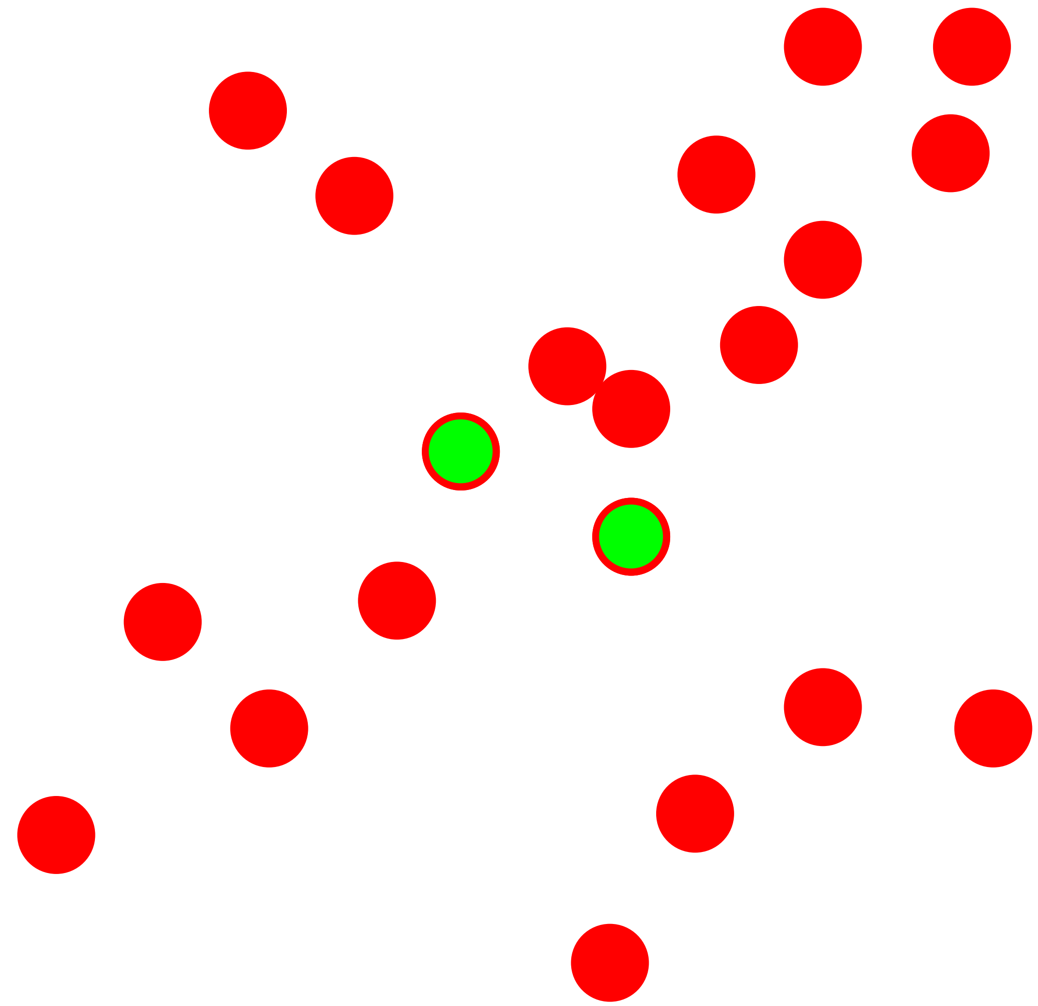
1. **Sample** (randomly) the number of points *s* required to fit the model
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Slide credit: James Hays

RANSAC

Line fitting example



Algorithm:

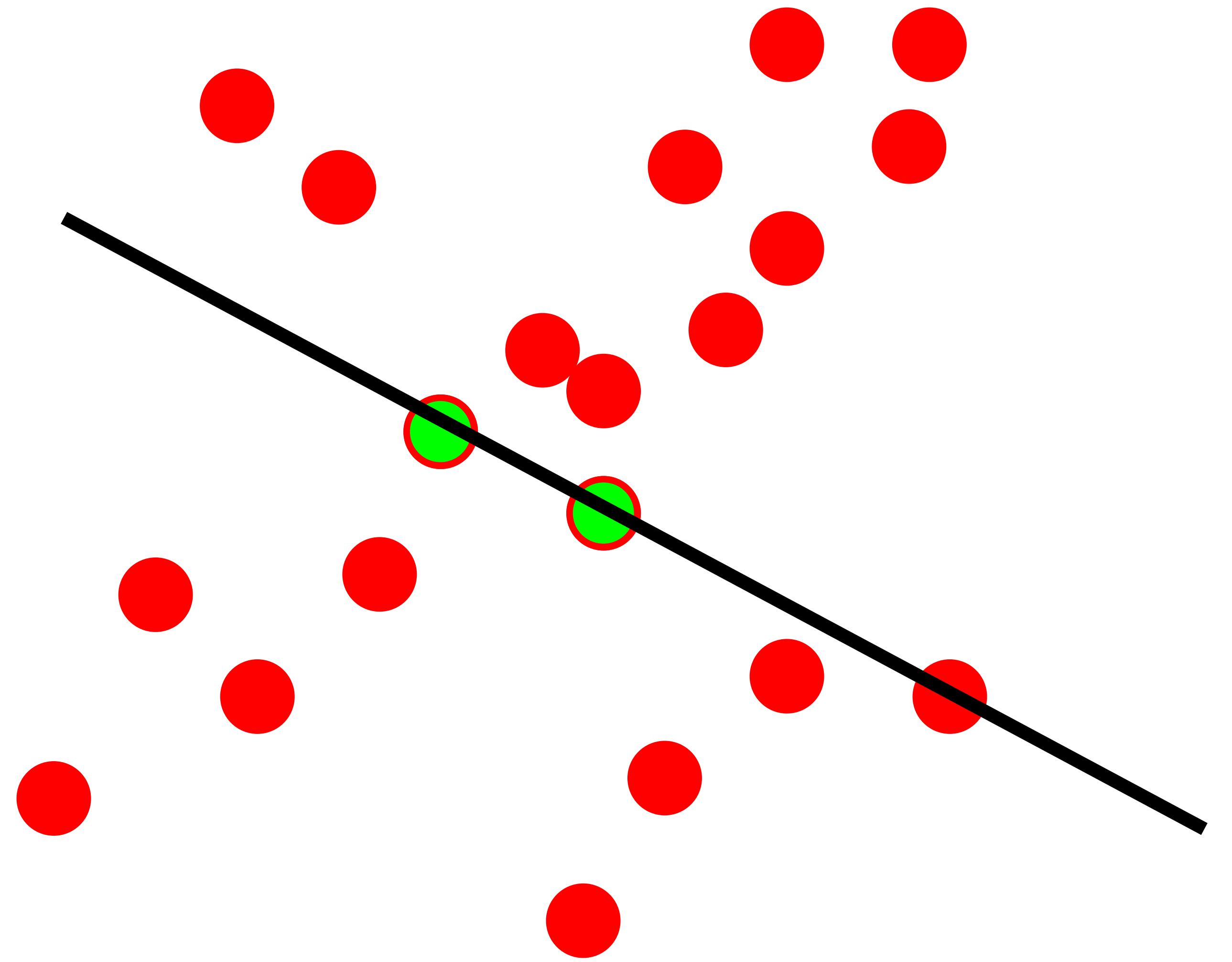
1. **Sample** (randomly) the number of points required to fit the model (s=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Illustration by Savarese

RANSAC

Line fitting example



Algorithm:

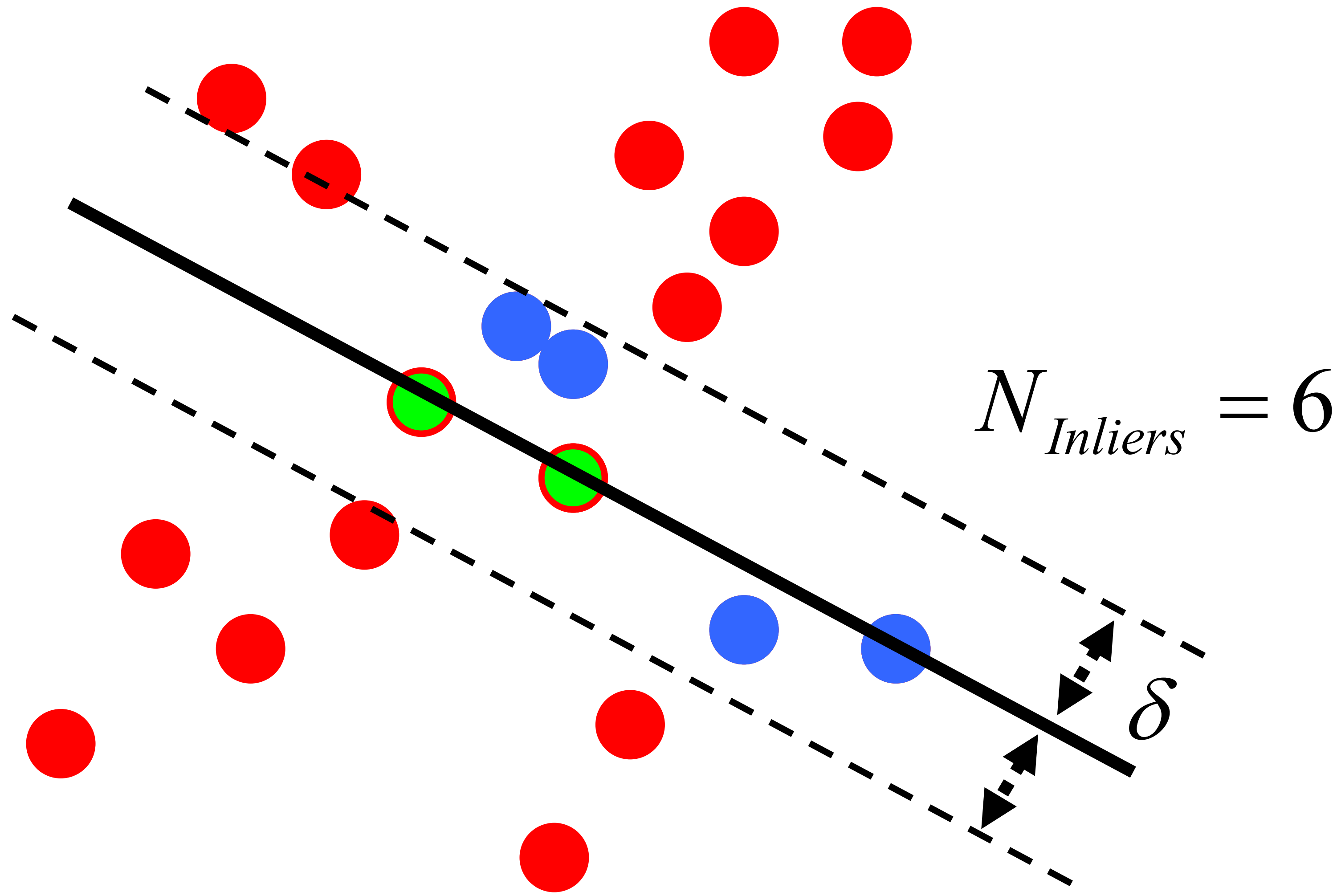
1. **Sample** (randomly) the number of points required to fit the model (*s*=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Slide credit: James Hays

RANSAC

Line fitting example



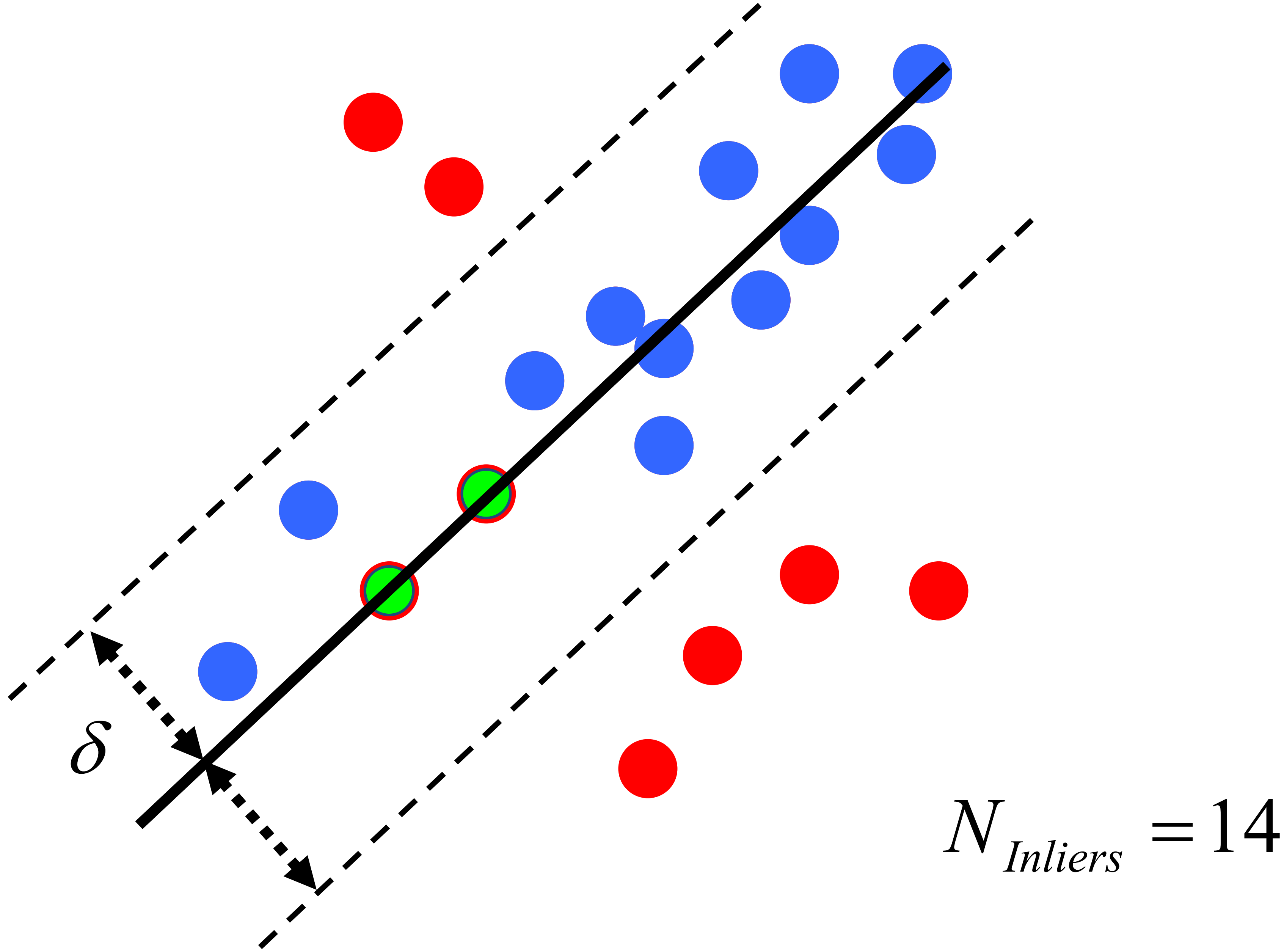
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (*s*=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Slide credit: James Hays

RANSAC



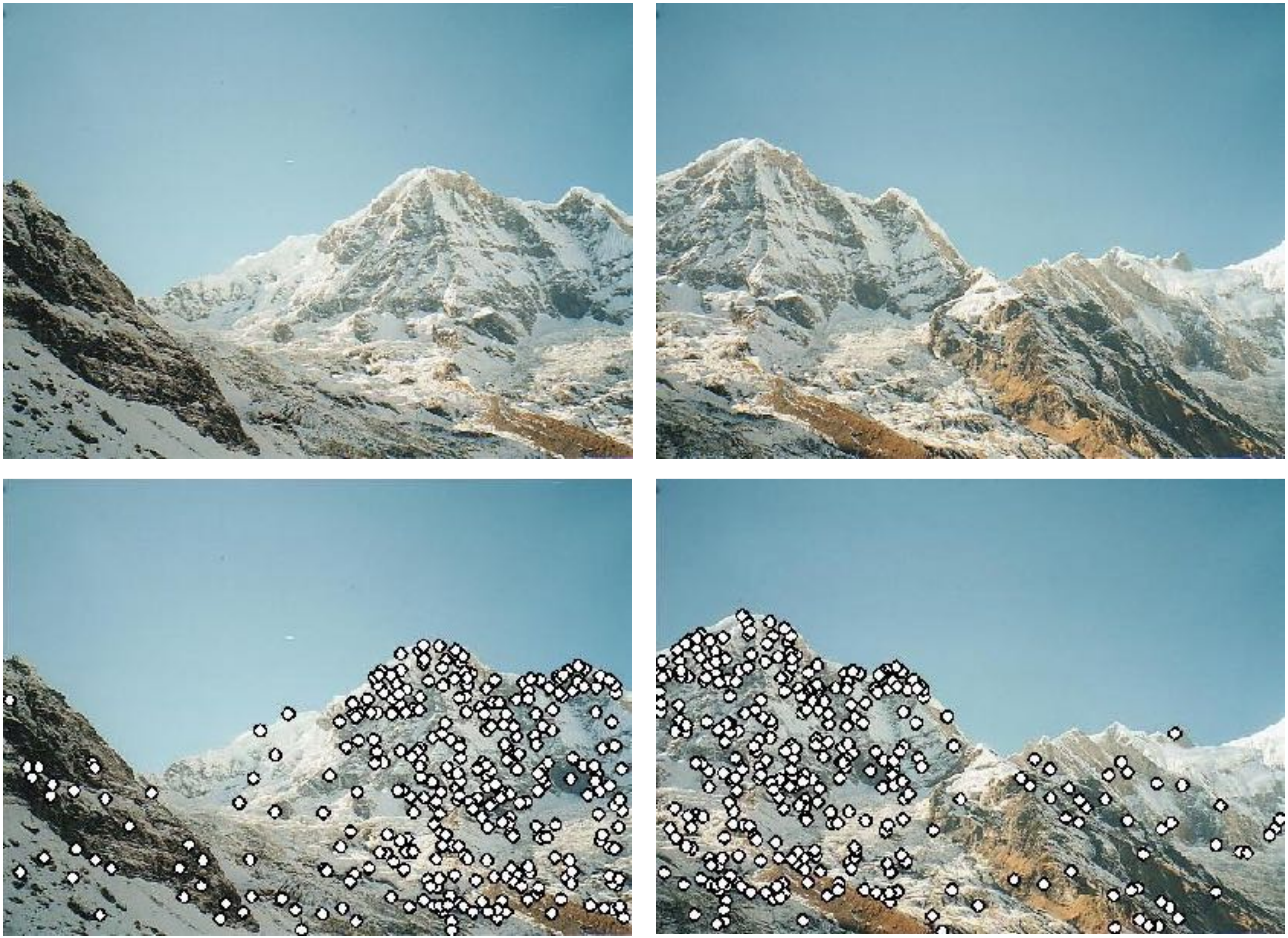
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (*s*=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Slide credit: James Hays

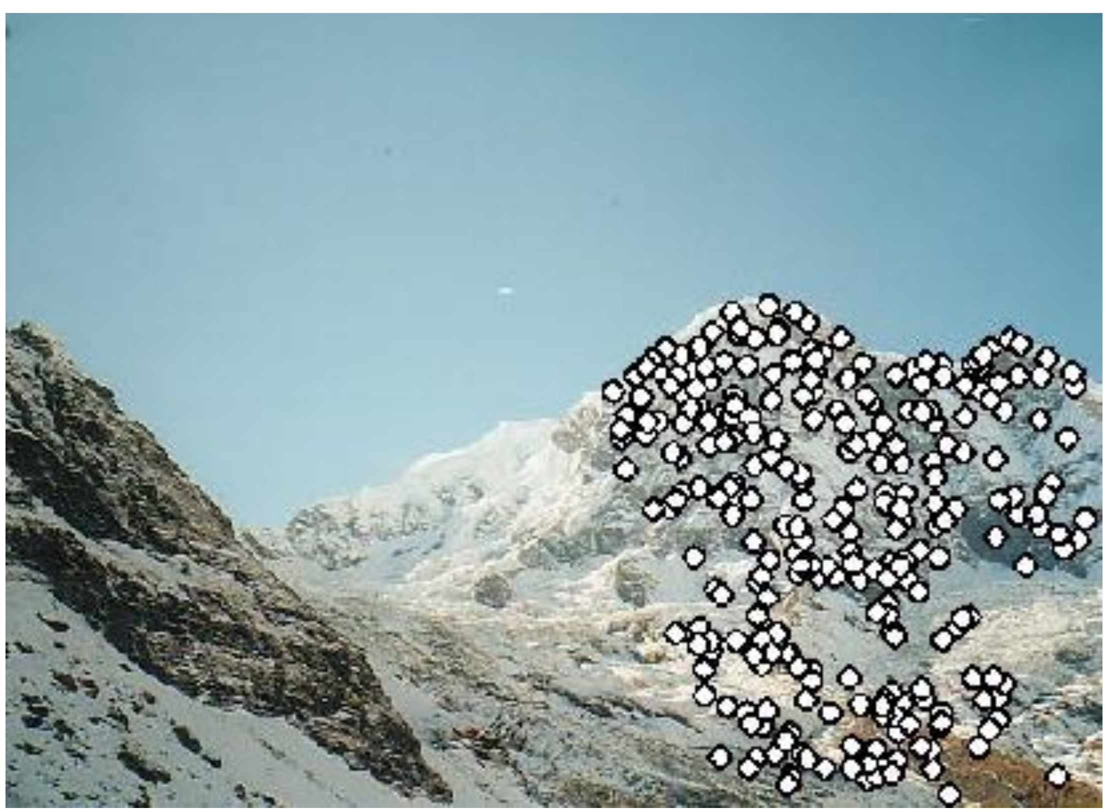
RANSAC for alignment



Slide credit: Deva Ramanan

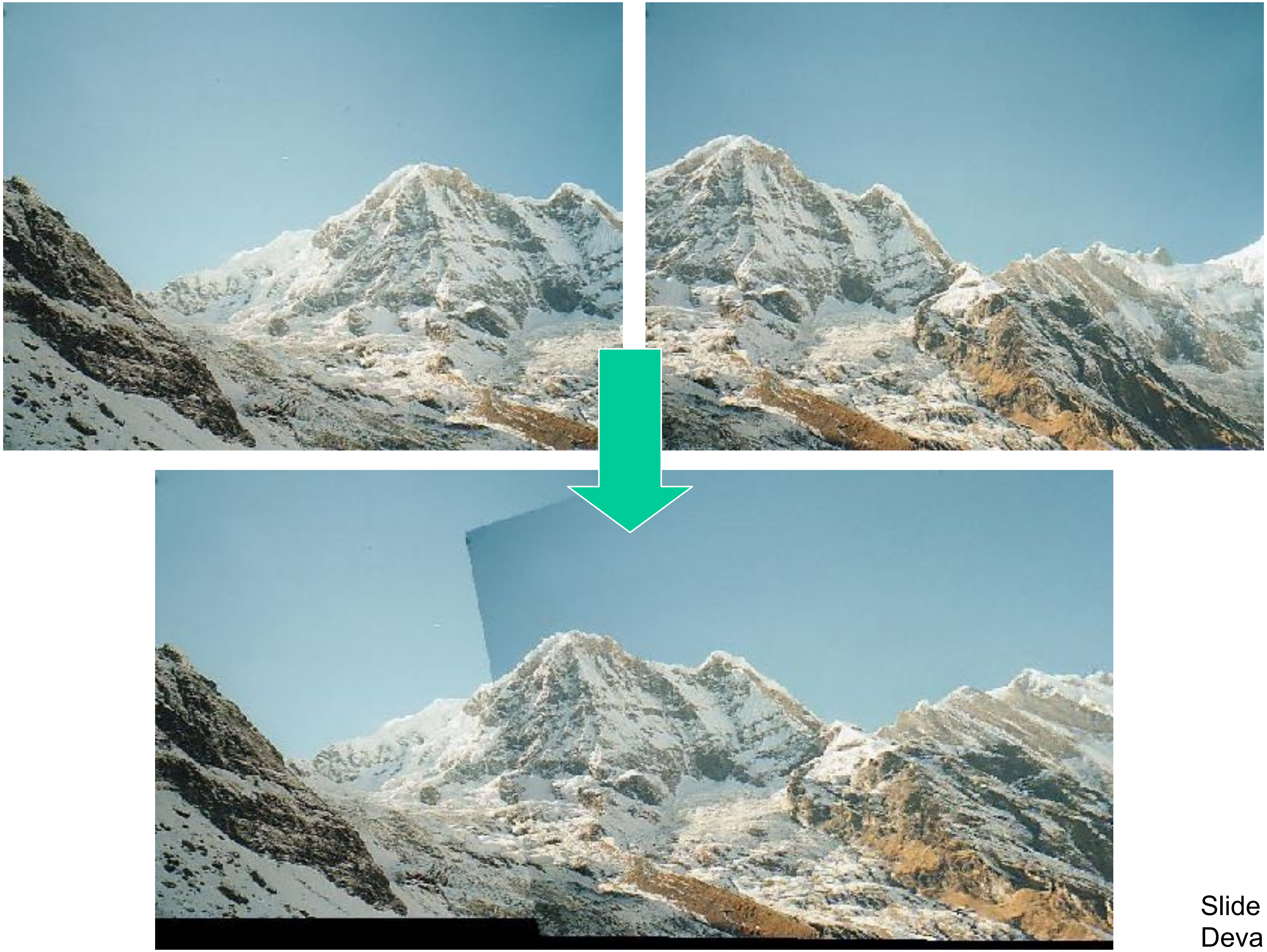
RANSAC for alignment






Slide credit: Deva Ramanan

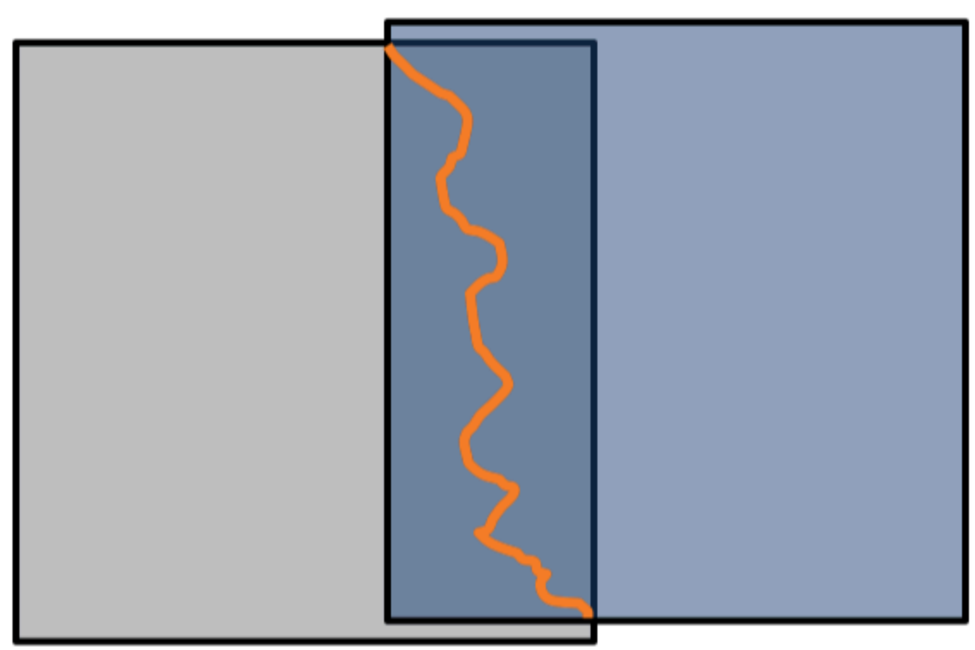
RANSAC for alignment



Slide credit:
Deva Ramanan

Blending

Instead of blending high frequencies along a straight line, blend along line of minimum differences in image intensities



Slide credit: Olga Russakovsky

Blending

Moving object, simple blending => blur

Slide credit: Davis '98

Blending

Minimum-cost cut ⇒ no blur

Slide credit: Davis ‘98