Juergen Gall

RANSAC and Image Alignment
MA-INF 2201 - Computer Vision
WS24/25

Motivation for feature-based alignment: Image mosaics

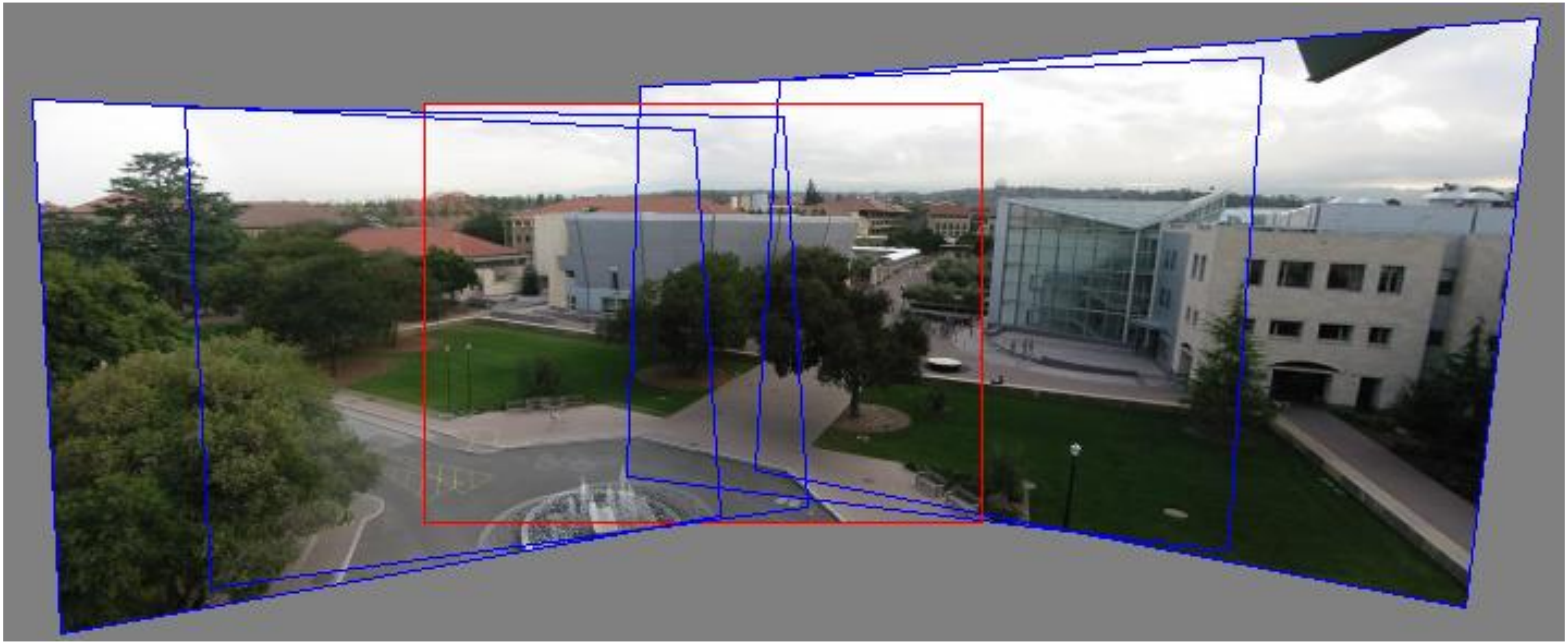
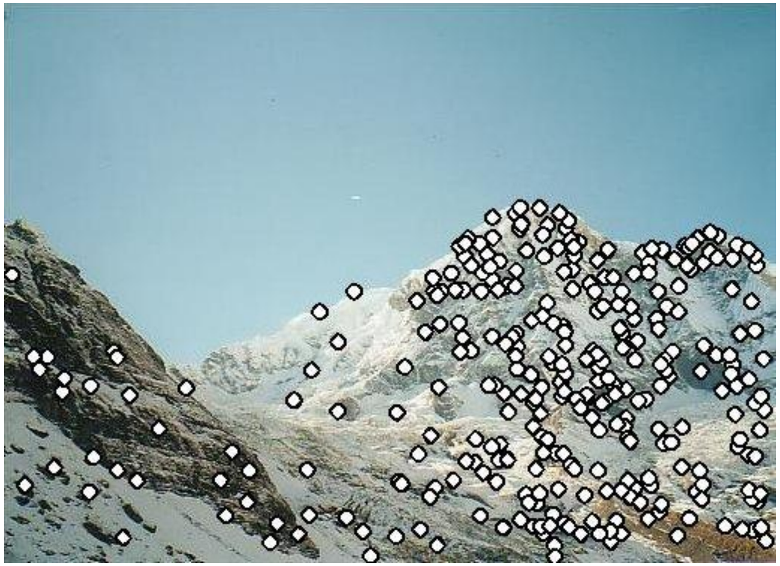
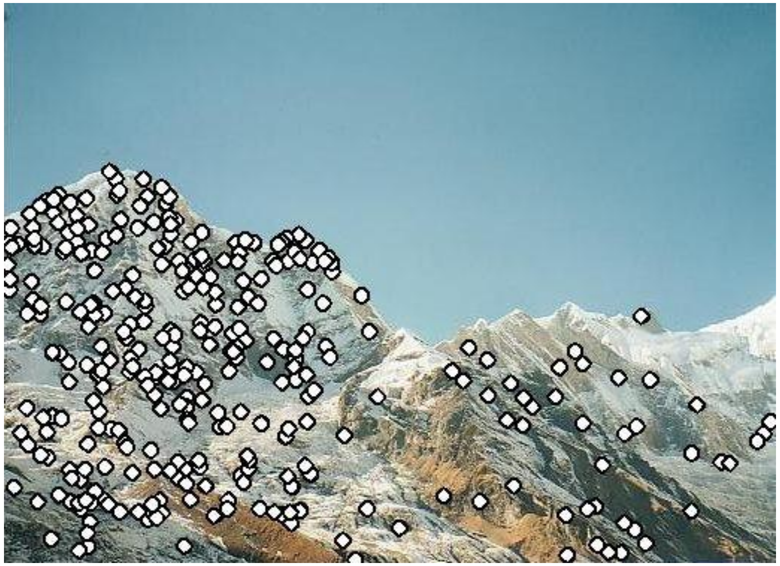


Image from http://graphics.cs.cmu.edu/courses/15-463/2010_fall/

Robust feature-based alignment





- Extract features

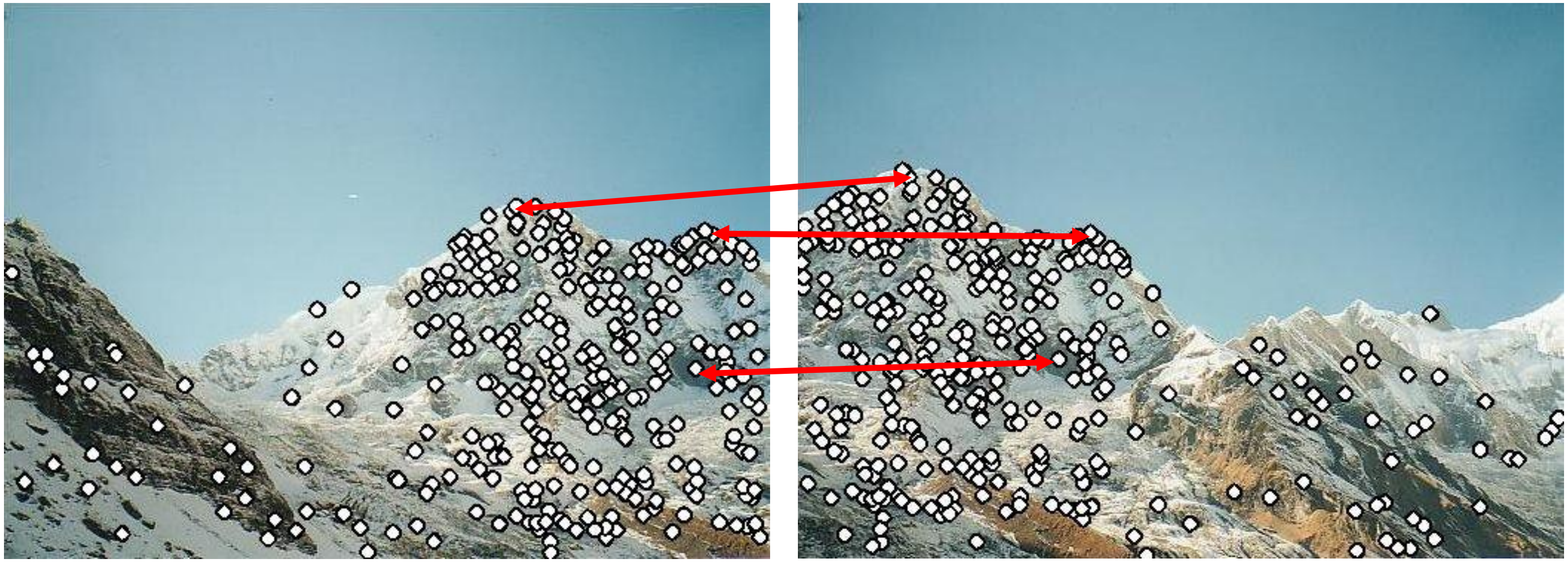
Source: L. Lazebnik

Robust feature-based alignment

- Extract features
- Compute *putative matches*

Source: L. Lazebnik

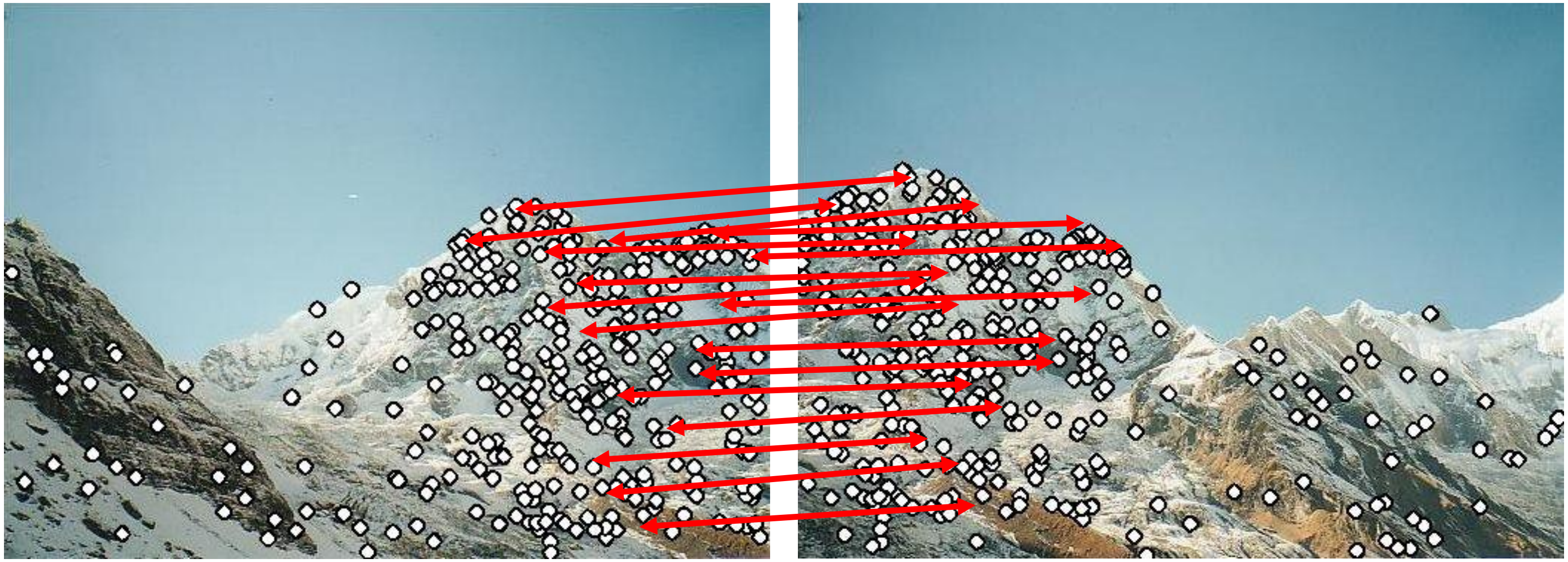
Robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

Source: L. Lazebnik

Robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

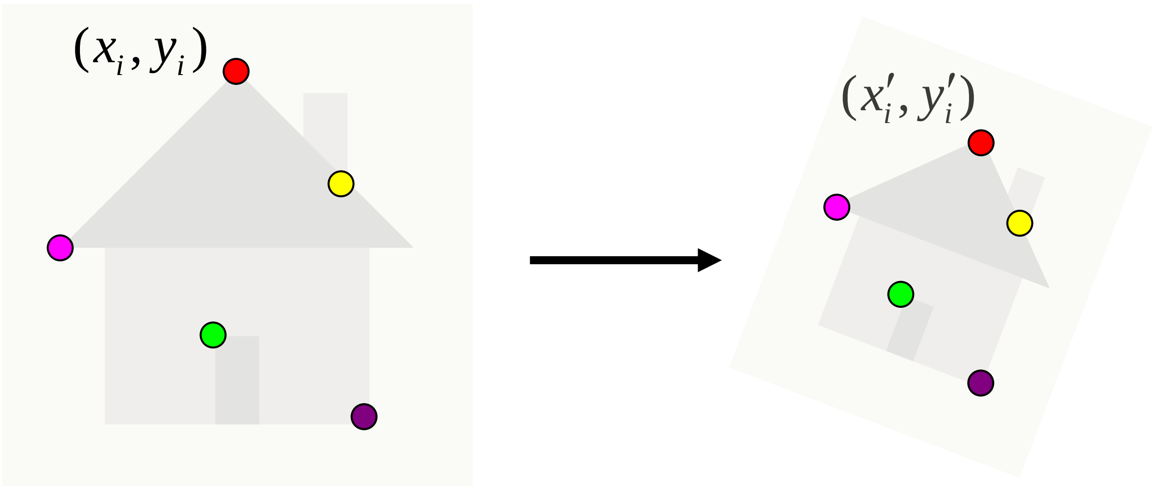
Source: L. Lazebnik

Robust feature-based alignment

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consistent with *T*)

Source: L. Lazebnik

Recall: Fitting an affine transformation



$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

Source: K. Grauman

Recall: Least squares problem

A *m x n* matrix of rank *n*:

$$Ax = b$$

Solution with pseudo-inverse:

$$A^T(Ax - b) = 0$$

$$x = (A^T A)^{-1} A^T b$$

Pseudo-inverse $\mathbf{A}^+$

Solution with SVD:

- SVD: $\mathbf{A} = \mathbf{UDV}^T$; $\mathbf{A}^+ = \mathbf{VD}^{-1}\mathbf{U}^T$
- $\mathbf{b}' = \mathbf{U}^T\mathbf{b}$
- $y_i = b'_i / D_{ii}$
- $\mathbf{x} = \mathbf{Vy}$

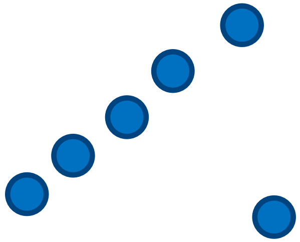
There are different ways to solve the problem. The optimal method depends on size and sparseness of **A**

Specific C++ linear algebra libraries exist!

Outliers

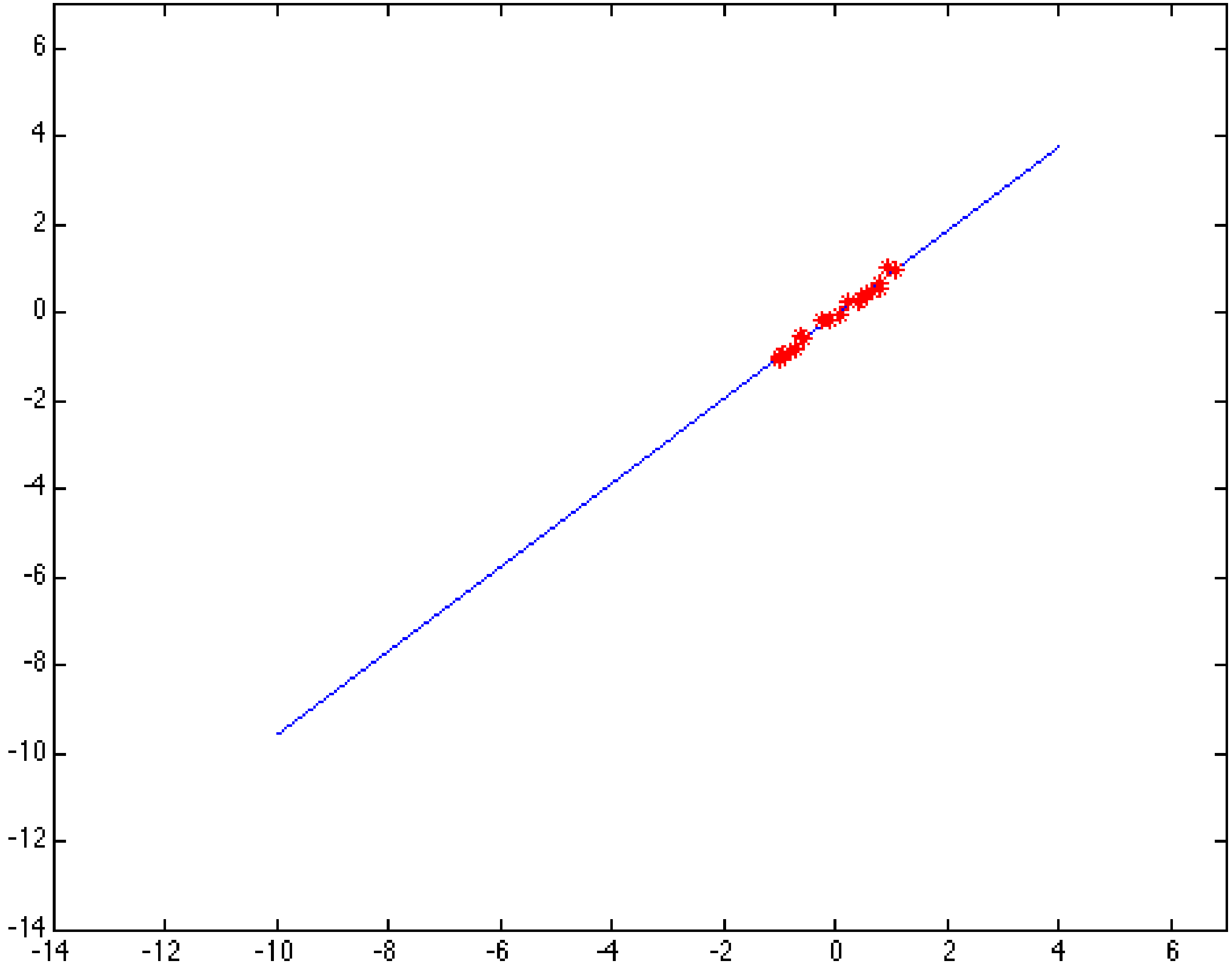
Outliers can hurt the quality of our parameter estimates, e.g.,

- an erroneous pair of matching points from two images
- an edge point that is noise, or doesn't belong to the line we are fitting.



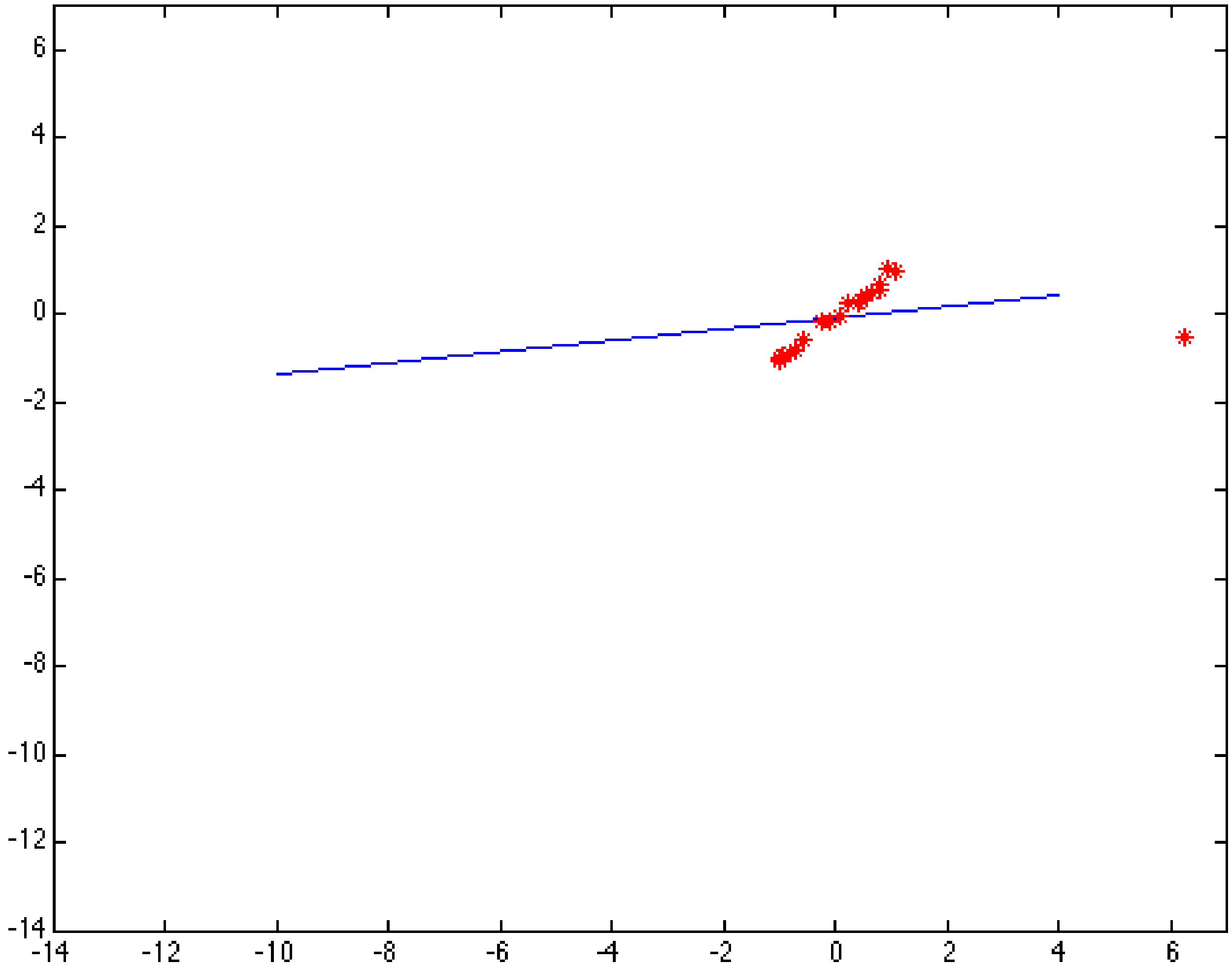
Kristen Grauman

Outliers affect least squares fit



Kristen Grauman

Outliers affect least squares fit



Kristen Grauman

RANdom Sample Consensus (RANSAC)

UNIVERSITÄT BONN

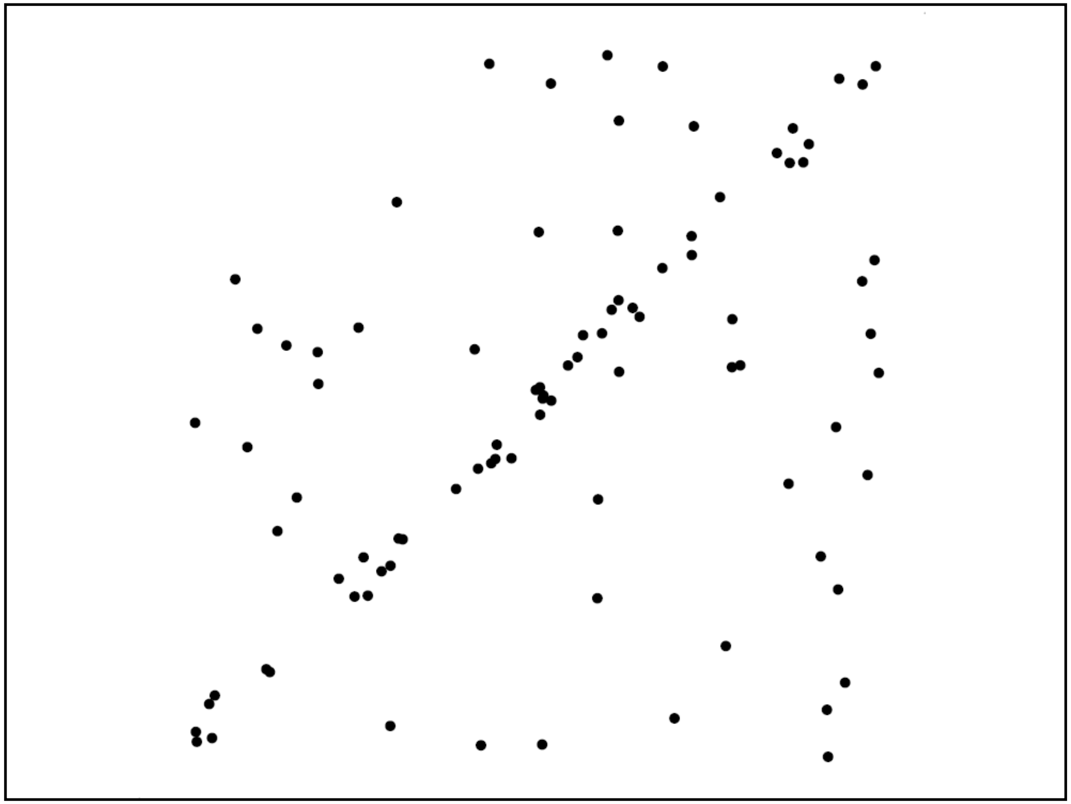
RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute estimate of transformation on all of the inliers

Keep the transformation with the largest number of inliers

Kristen Grauman

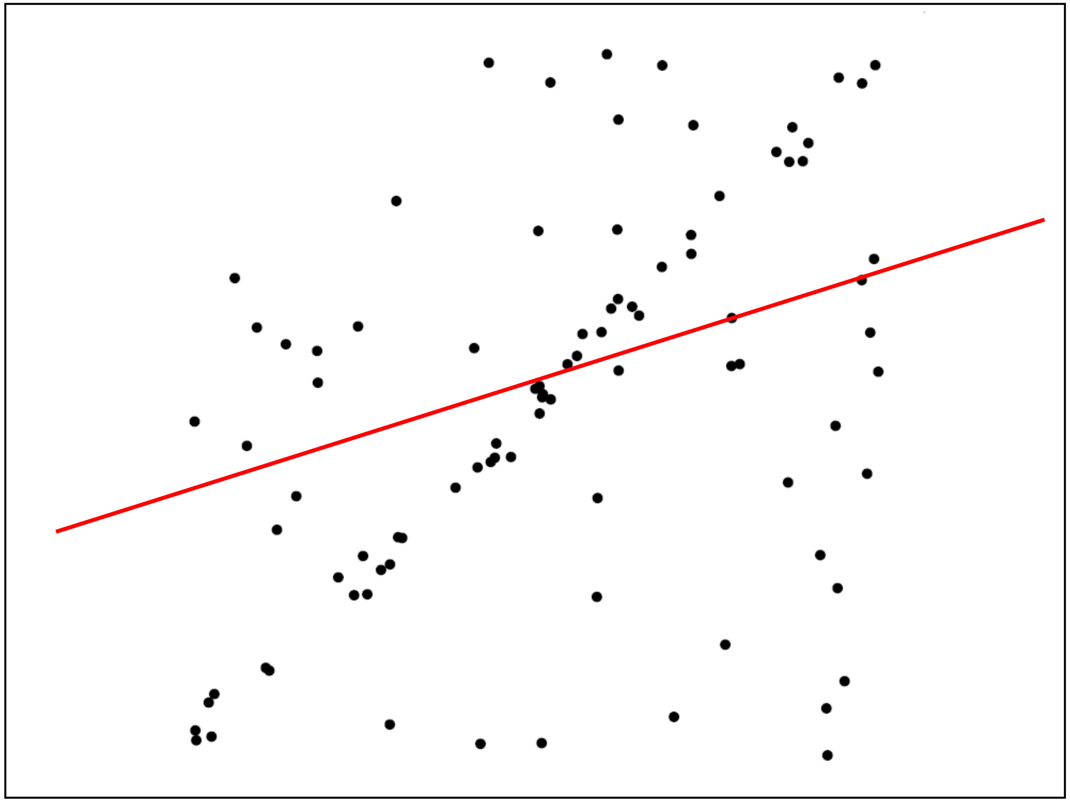
RANSAC for line fitting example



Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

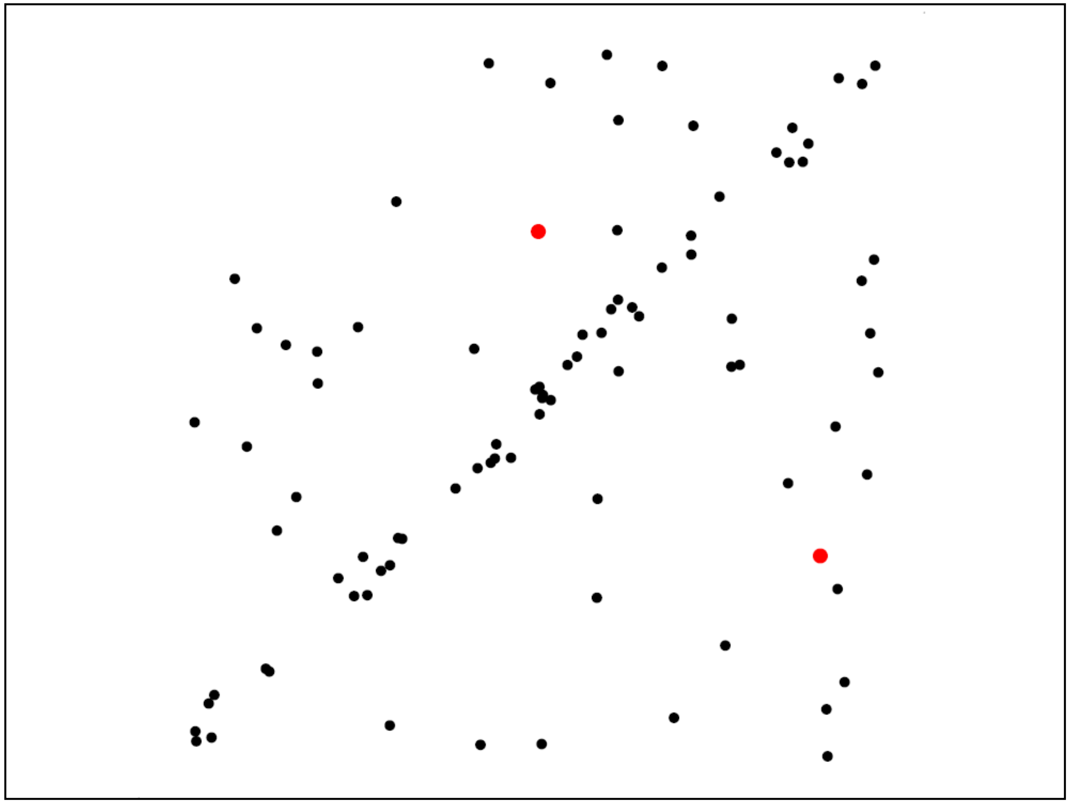


Least-squares fit

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

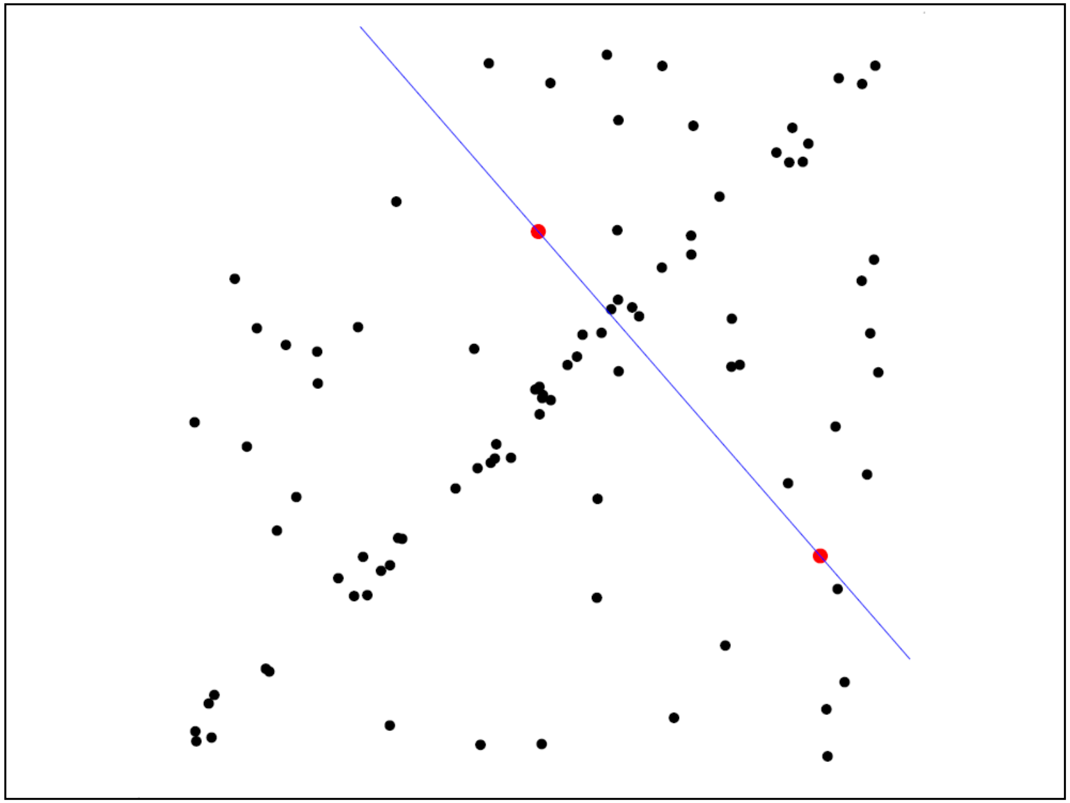


1. Randomly select minimal subset of points

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

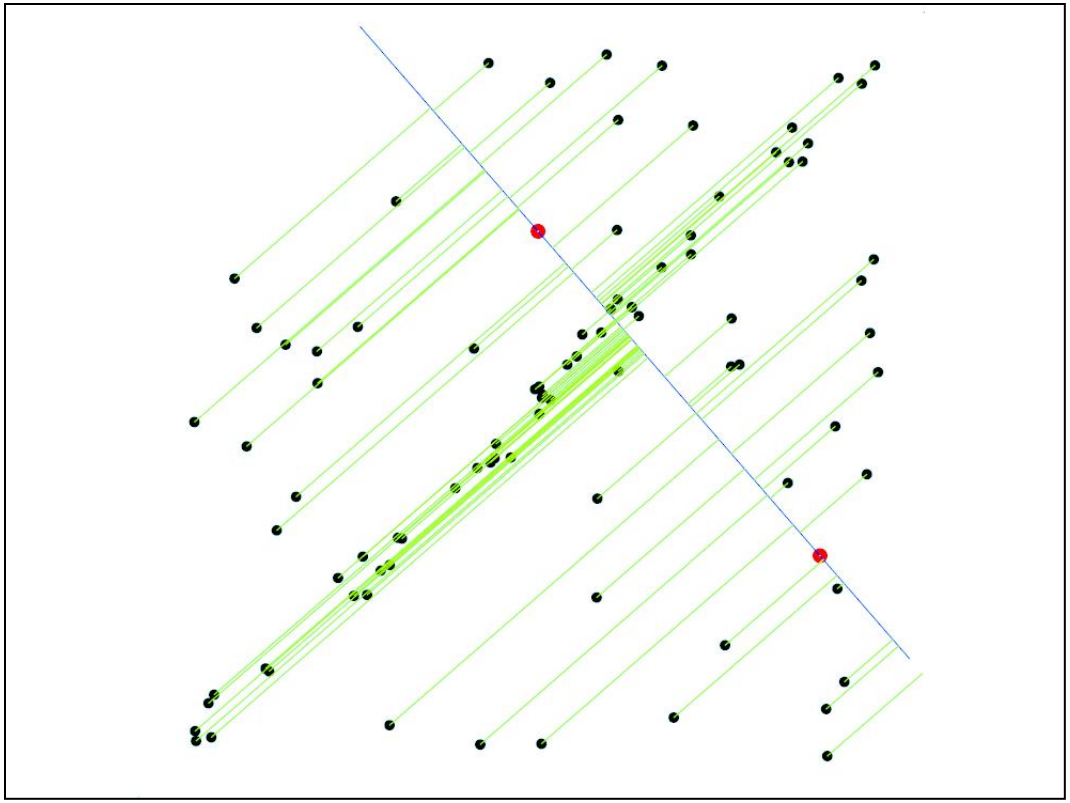


1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

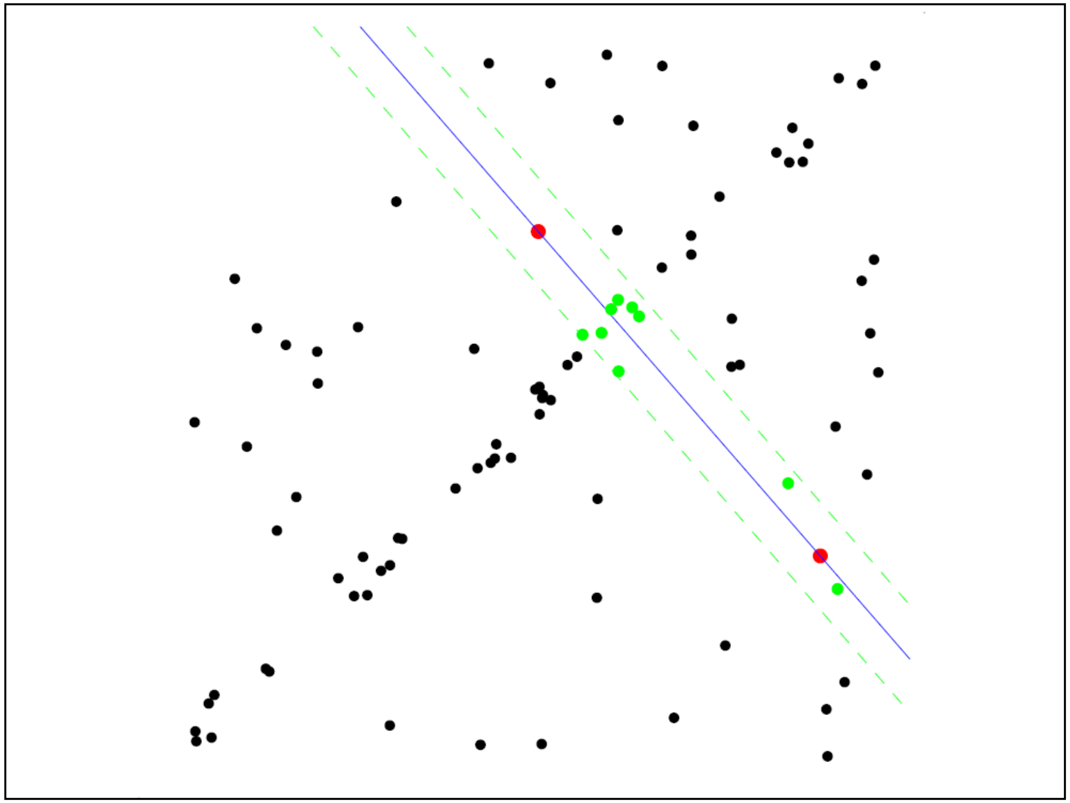


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



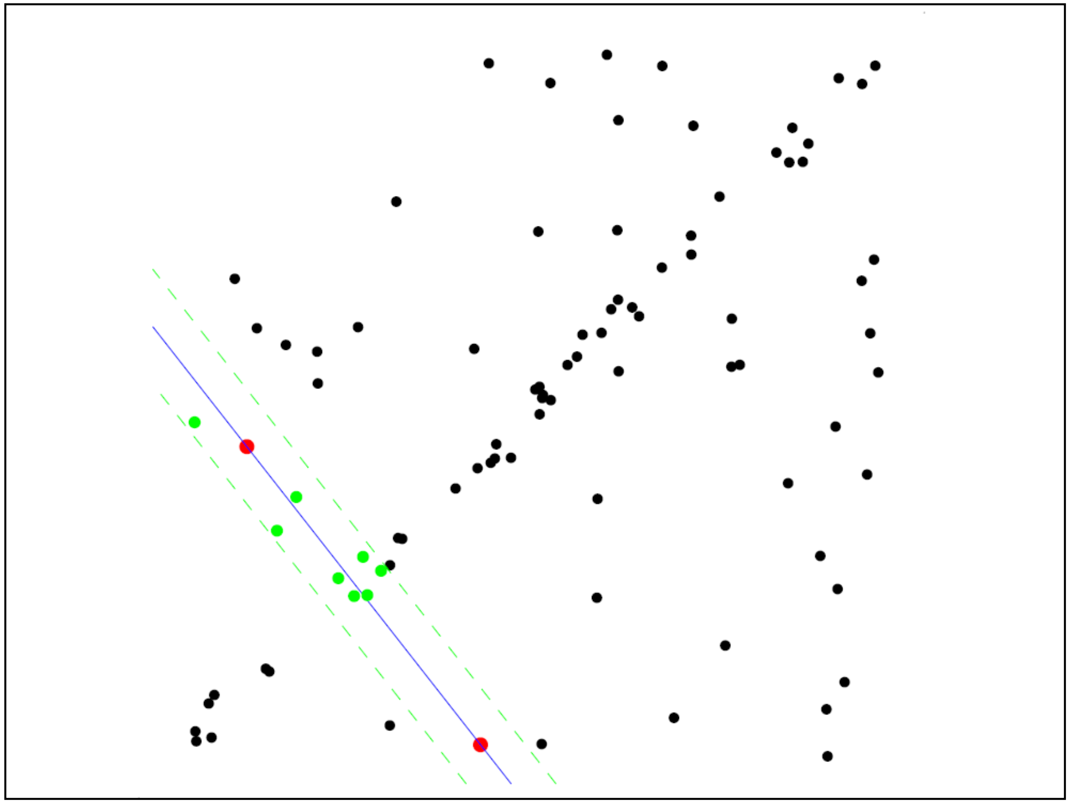
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

UNIVERSITÄT BONN

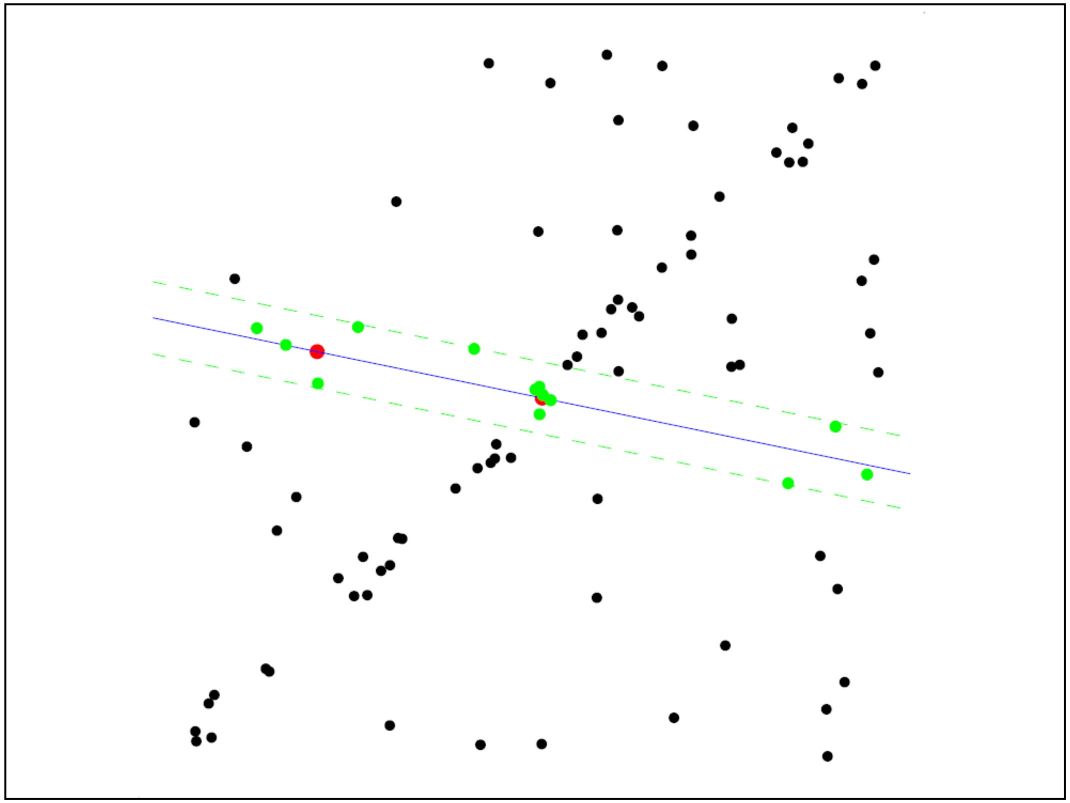


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



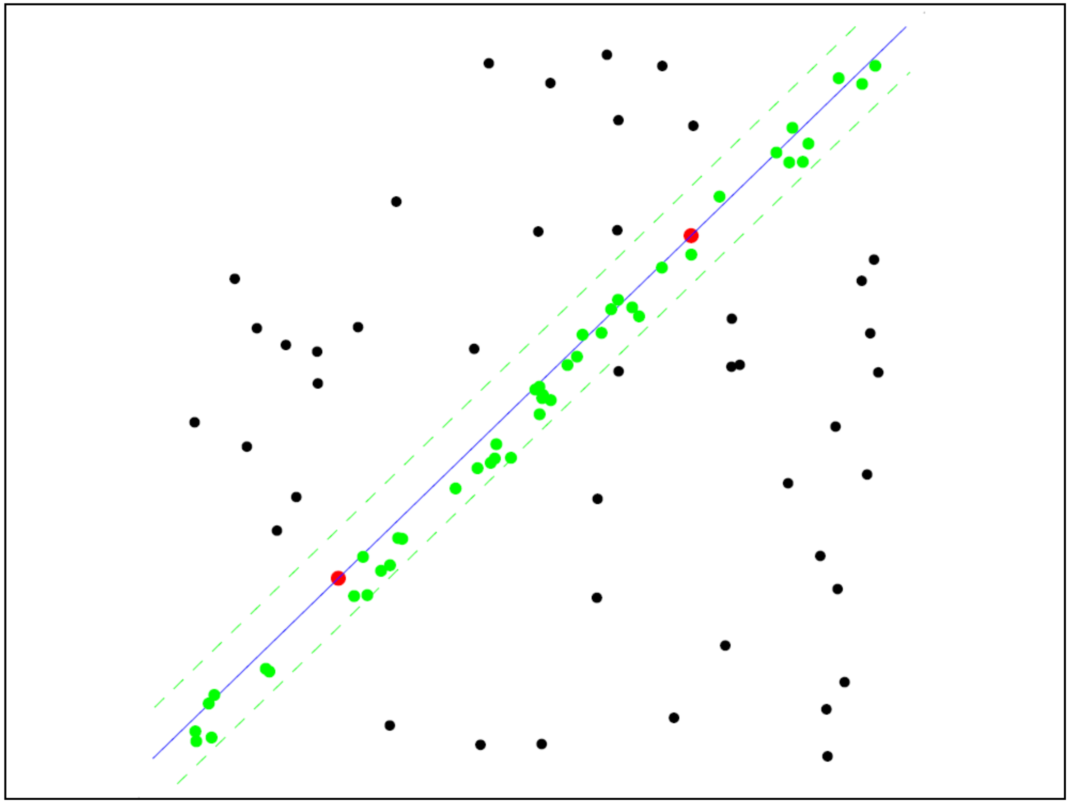
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

Uncontaminated sample

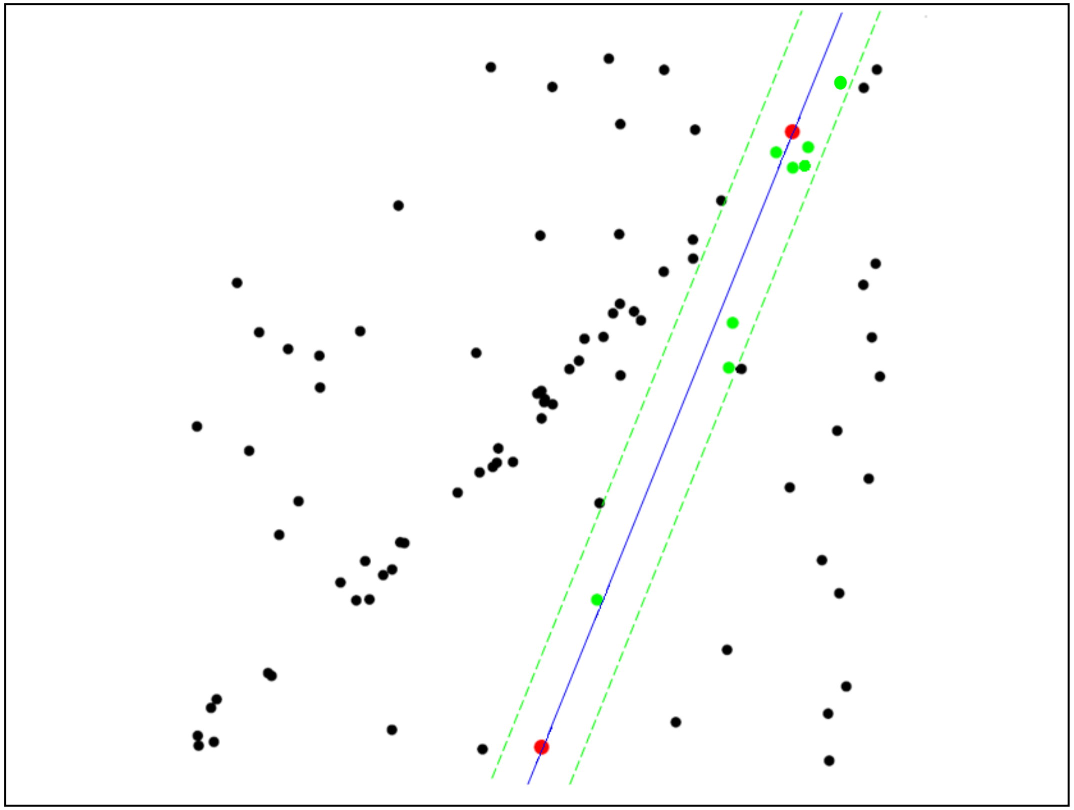


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

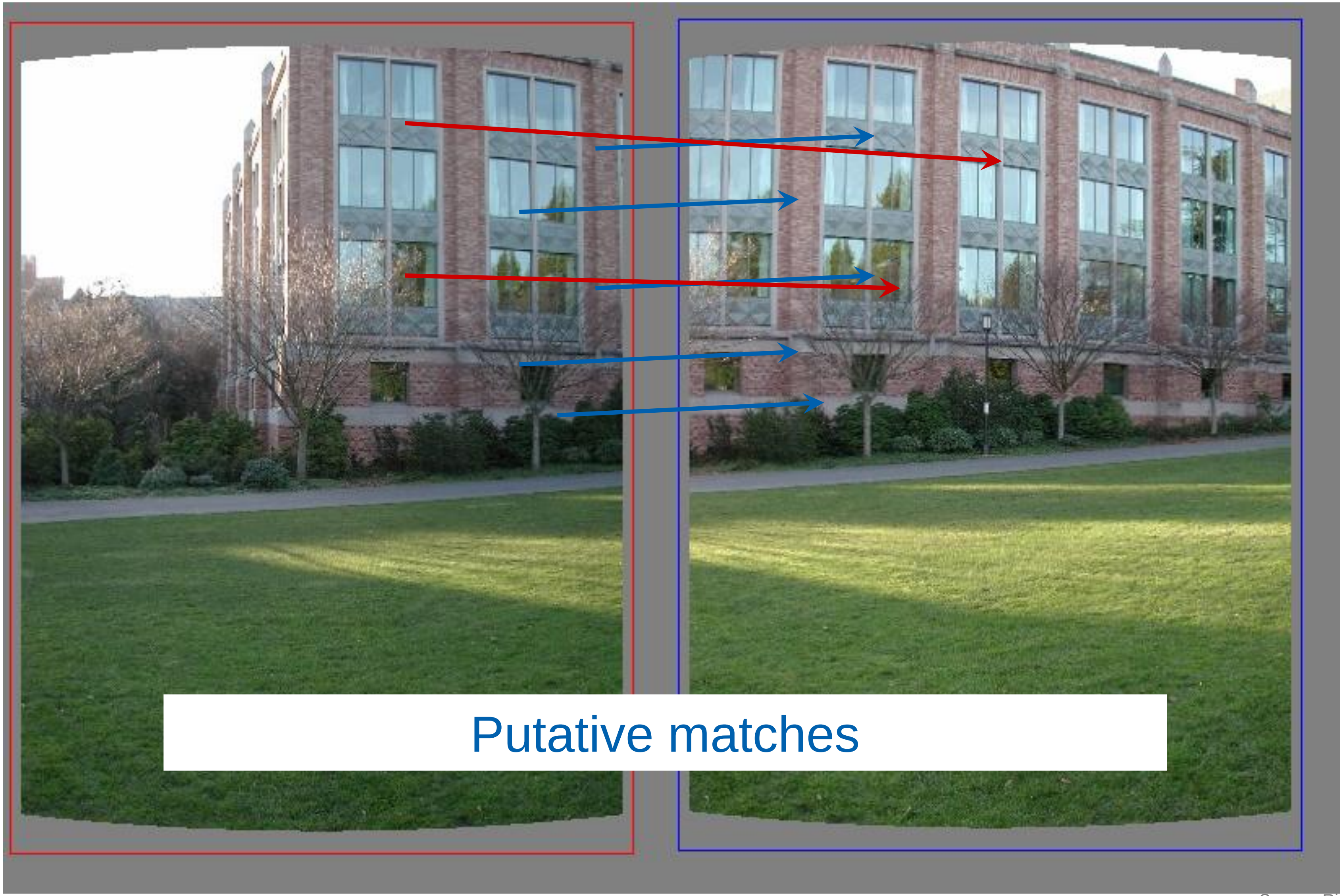
RANSAC for line fitting

Repeat ***N*** times:

- Draw ***s*** points uniformly at random
- Fit line to these ***s*** points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than ***t***)
- If there are ***d*** or more inliers, accept the line and refit using all inliers

Lana Lazebnik

RANSAC example: Translation



Source: Rick Szeliski

RANSAC example: Translation

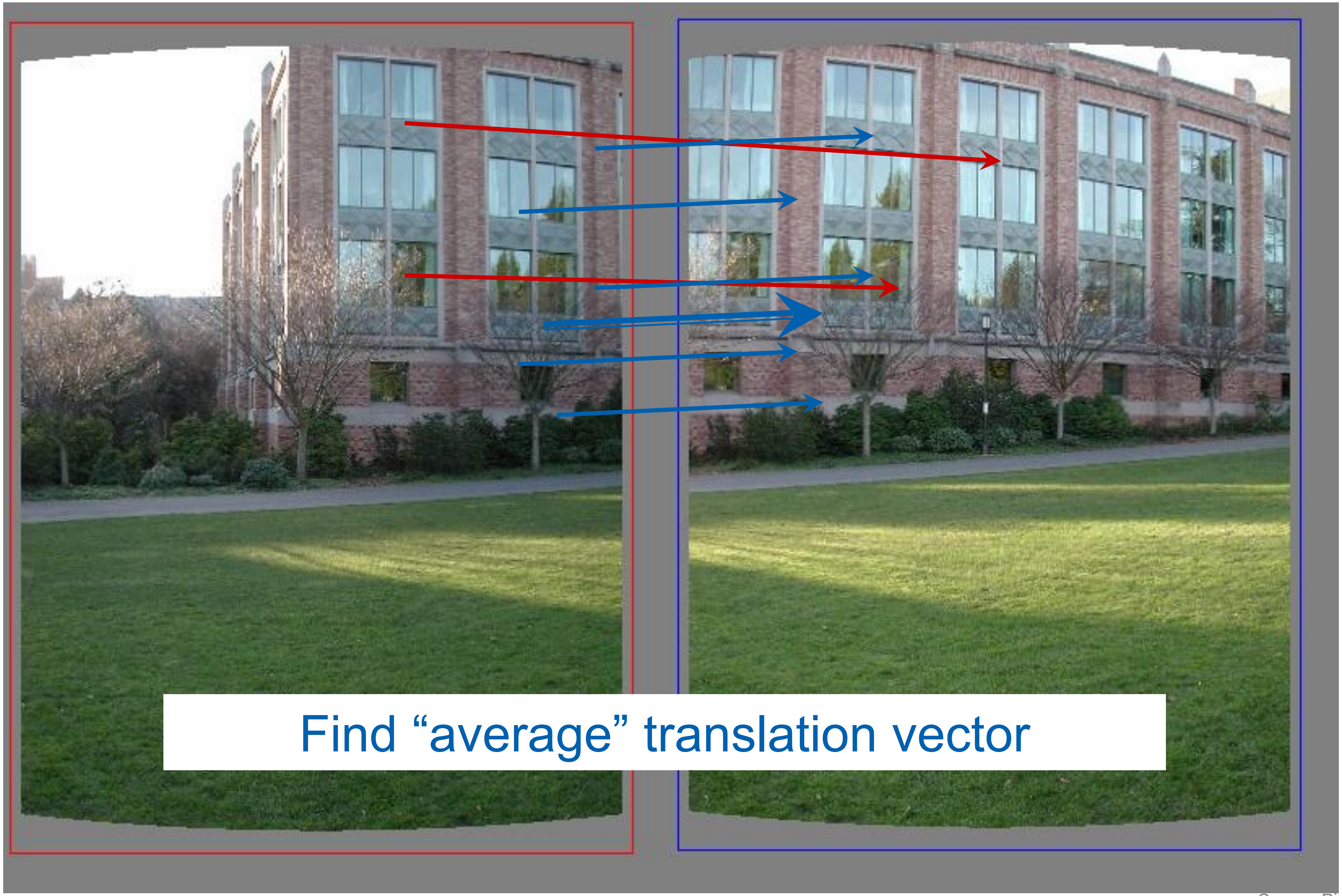
Source: Rick Szeliski

RANSAC example: Translation

UNIVERSITÄT BONN

Source: Rick Szeliski

RANSAC example: Translation



Source: Rick Szeliski

How Many Samples?

On average

N	… number of point
I	… number of inliers
m	… size of the sample

P(good) = $$\frac{\binom{I}{m}}{\binom{N}{m}} = \prod_{j=0}^{m-1} \frac{I-j}{N-j}$$

mean time before the success

E(*k*) = 1 / P(good)

Ondřej Chum

How Many Samples?

UNIVERSITÄT BONN

With confidence *p*

How large k?

...to hit at least one pair of points on the line l with probability <u>larger</u> than p (0.95)

Equivalently

...the probability of not hitting any pair of points on l is $\leq 1 - p$

Ondřej Chum

How Many Samples?

UNIVERSITÄT BONN

With confidence *p*

N	… number of point
I	… number of inliers
m	… size of the sample

P(good) = $$\frac{\binom{I}{m}}{\binom{N}{m}} = \prod_{j=0}^{m-1} \frac{I-j}{N-j}$$

P(bad) = 1 – P(good)

P(bad *k* times) = $\left(1 - \text{P(good)}\right)^{k}$

Ondřej Chum

How Many Samples?

With confidence *p*

$$P(\text{bad } k \text{ times}) = \big(1 - P(\text{good})\big)^k \leq 1 - p$$

$$k \log \big(1 - P(\text{good})\big) \leq \log(1 - p)$$

$$k \geq \log(1 - p) \,/\, \log \big(1 - P(\text{good})\big)$$

$$P(\text{good}) = \frac{\binom{I}{m}}{\binom{N}{m}} = \prod_{j=0}^{m-1} \frac{I - j}{N - j}$$

Ondřej Chum

How Many Samples

I / *N* [%]

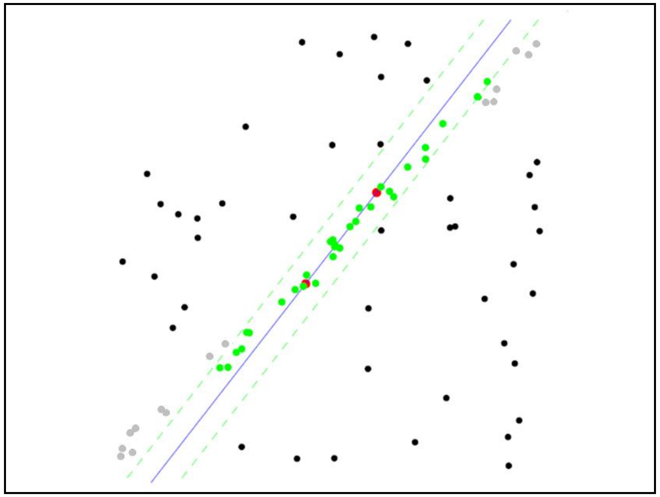
Size of the sample *m*

	15%	20%	30%	40%	50%	70%
2	132	73	32	17	10	4
4	5916	1871	368	116	46	11
7	$1.75 \cdot 10^6$	$2.34 \cdot 10^5$	$1.37 \cdot 10^4$	1827	382	35
8	$1.17 \cdot 10^7$	$1.17 \cdot 10^6$	$4.57 \cdot 10^4$	4570	765	50
12	$2.31 \cdot 10^{10}$	$7.31 \cdot 10^8$	$5.64 \cdot 10^6$	$1.79 \cdot 10^5$	$1.23 \cdot 10^4$	215
18	$2.08 \cdot 10^{15}$	$1.14 \cdot 10^{13}$	$7.73 \cdot 10^9$	$4.36 \cdot 10^7$	$7.85 \cdot 10^5$	1838
30	∞	∞	$1.35 \cdot 10^{16}$	$2.60 \cdot 10^{12}$	$3.22 \cdot 10^9$	$1.33 \cdot 10^5$
40	∞	∞	∞	$2.70 \cdot 10^{16}$	$3.29 \cdot 10^{12}$	$4.71 \cdot 10^6$

Ondřej Chum

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Doesn't work well for low inlier ratios (too many iterations, or can fail completely)
  - Can't always get a good initialization of the model based on the minimum number of samples



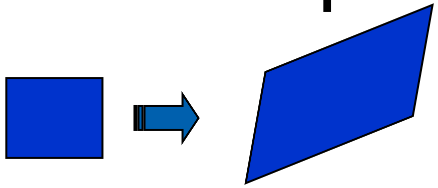
Lana Lazebnik

Recall: 2D Affine Transformations

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- Affine transformations are combinations of …
  - Linear transformations, and
  - Translations

- Parallel lines remain parallel



Source: K. Grauman

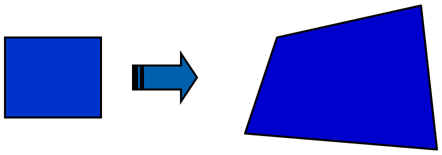
Projective Transformations

Projective transformations:

- Affine transformations, and
- Projective warps

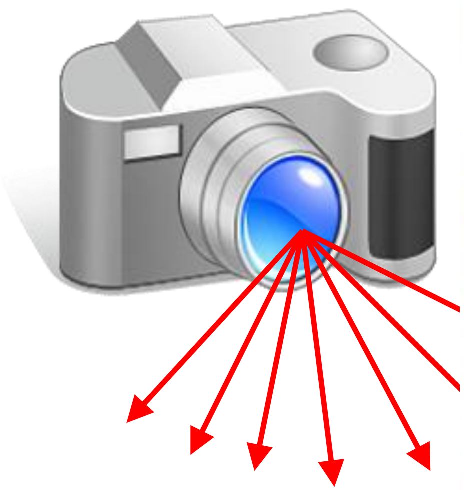
$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Parallel lines do not necessarily remain parallel



Kristen Grauman

Mosaics



. . .

image from S. Seitz

Obtain a wider angle view by combining multiple images.

Kristen Grauman

How to stitch together a panorama (a.k.a. mosaic)?

Basic Procedure

- Take a sequence of images from the same position
  - Rotate the camera about its optical center
- Compute transformation between second image and first
- Transform the second image to overlap with the first
- Blend the two together to create a mosaic
- (If there are more images, repeat)

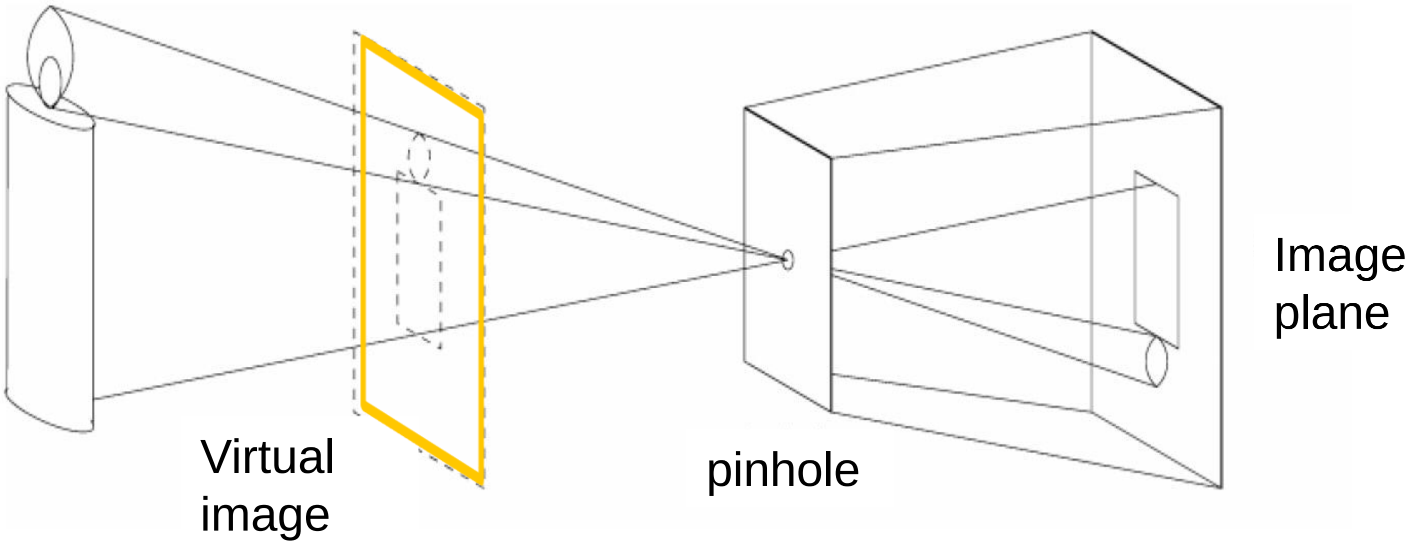
…but **wait**, why should this work at all?

- What about the 3D geometry of the scene?
- Why aren't we using it?

Source: Steve Seitz

Pinhole camera

Pinhole camera is a simple model to approximate imaging process, perspective **projection**.

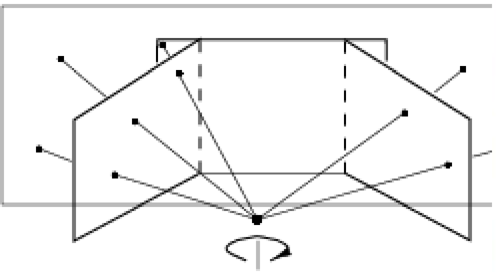


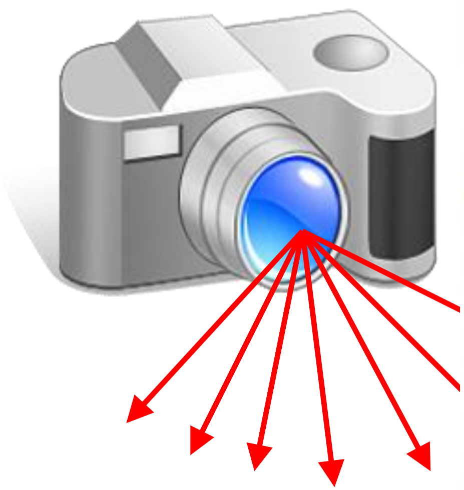
If we treat pinhole as a point, only one ray from any given point can enter the camera.

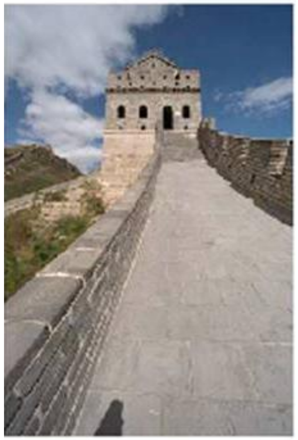
Fig from Forsyth and Ponce

Kristen Grauman

Mosaics







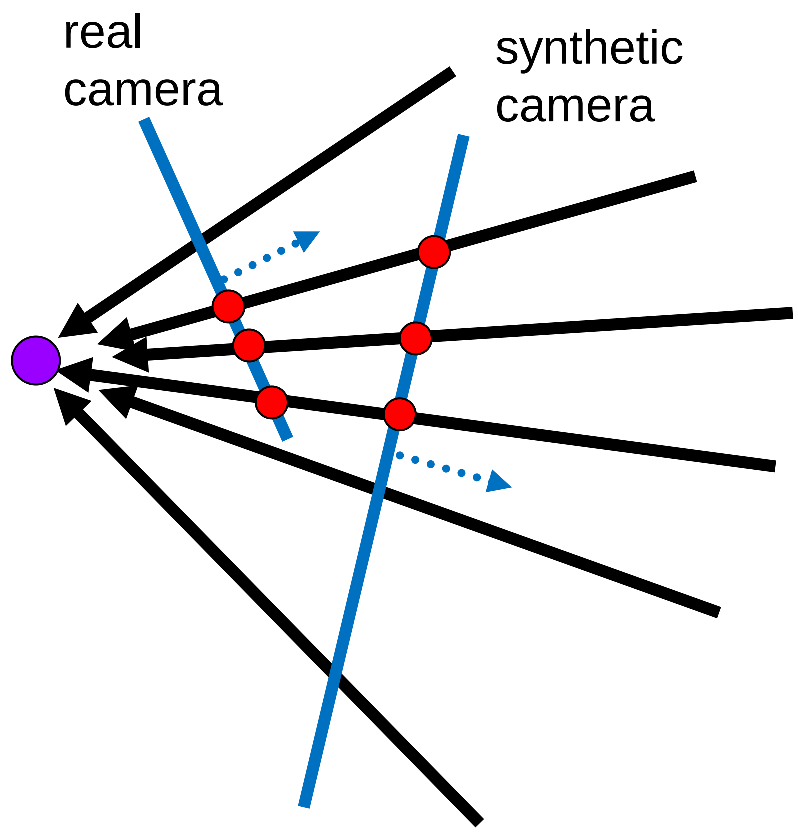
. . .

image from S. Seitz

Obtain a wider angle view by combining multiple images.

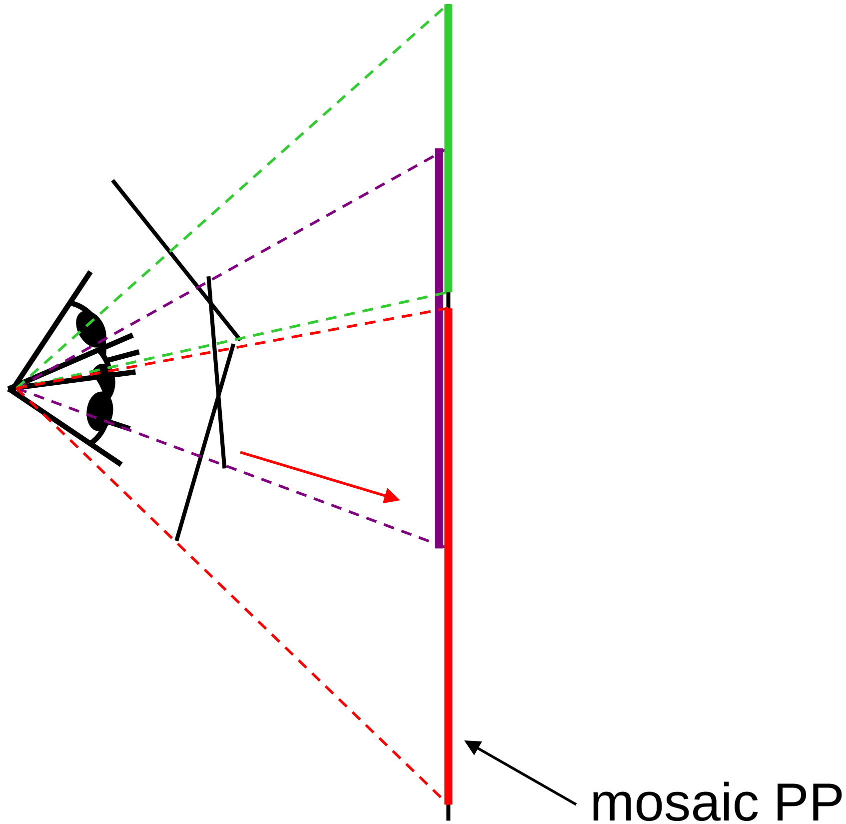
Kristen Grauman

Mosaics: generating synthetic views



Source: Alyosha Efros

Image reprojection



The mosaic has a natural interpretation in 3D

- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a *synthetic wide-angle camera*

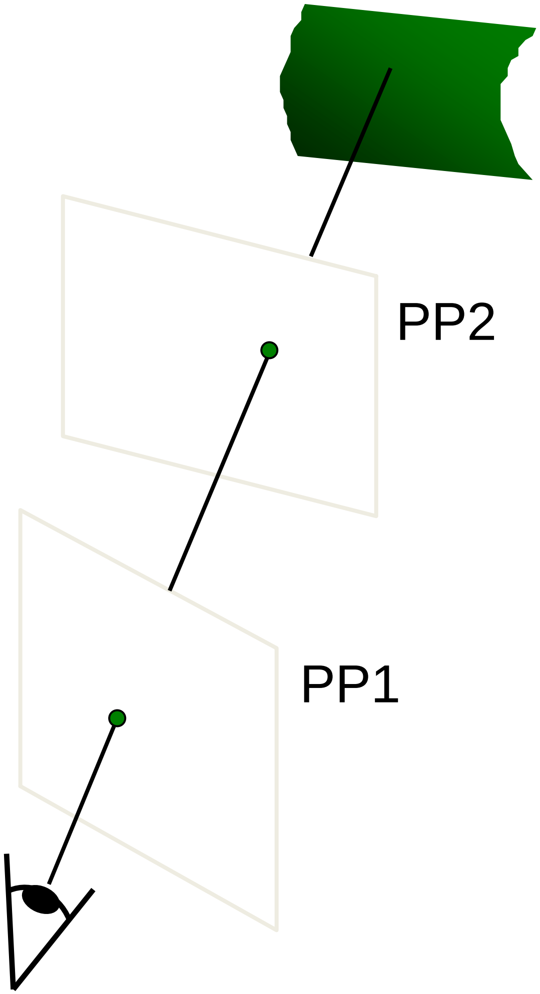
Source: Steve Seitz

Image reprojection

- How to relate two images from the same camera center? How to map a pixel from PP1 to PP2?

- Cast a ray through each pixel in PP1
- Draw the pixel where that ray intersects PP2

- Rather than thinking of this as a 3D reprojection, think of it as a 2D image warp from one image to another.



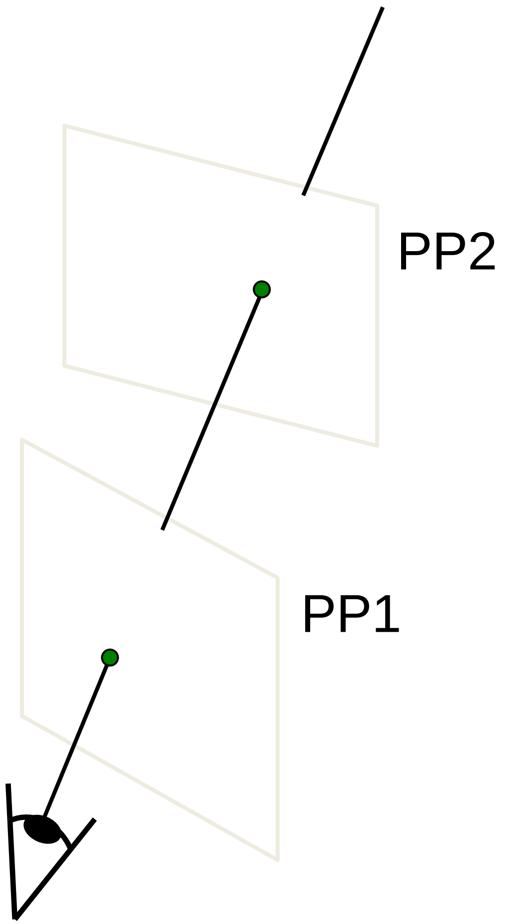
Source: Alyosha Efros

Image reprojection: Homography

A projective transform is a mapping between any two PPs with the same center of projection

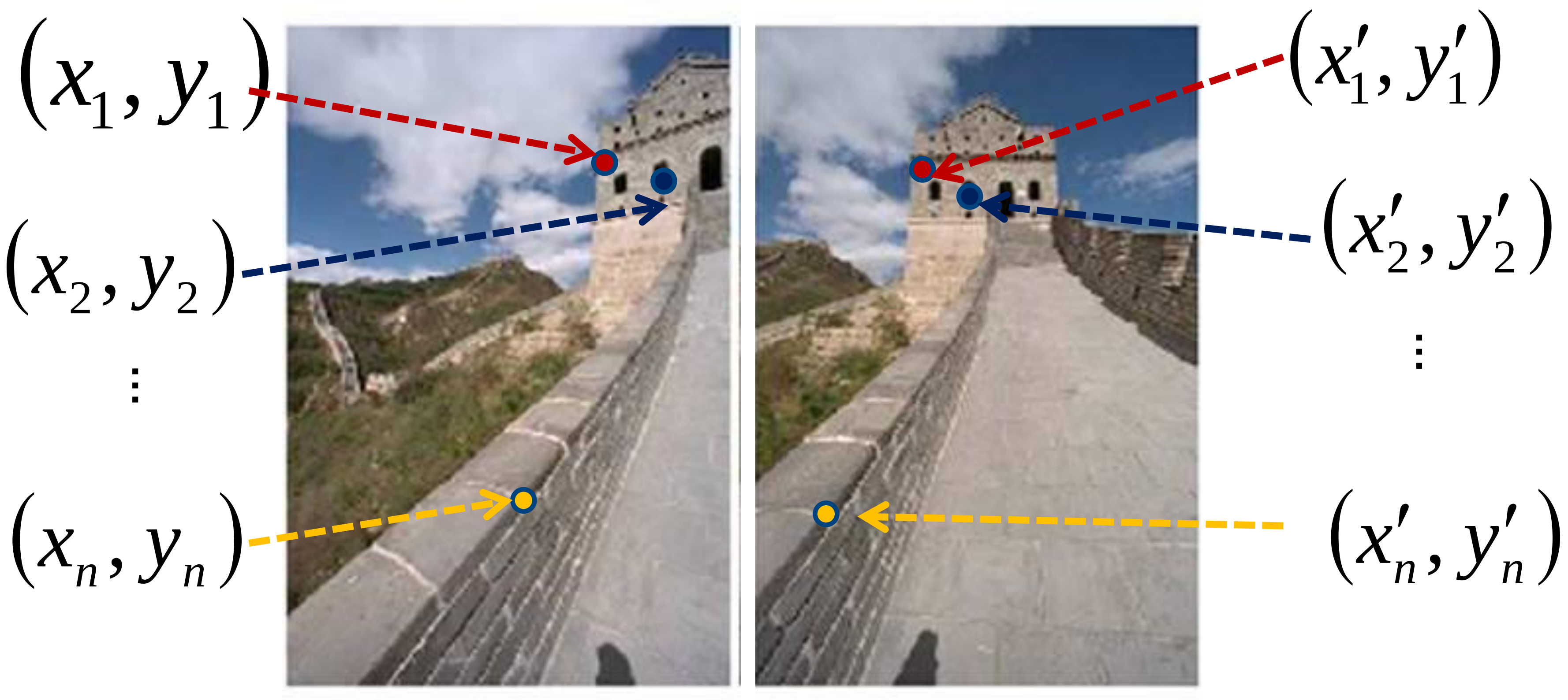
Called **Homography**

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$



Source: Alyosha Efros

Homography



To **compute** the homography given pairs of corresponding points in the images, we need to set up an equation where the parameters of **H** are the unknowns…

Kristen Grauman

Solving for homographies

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

Solving for homographies

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

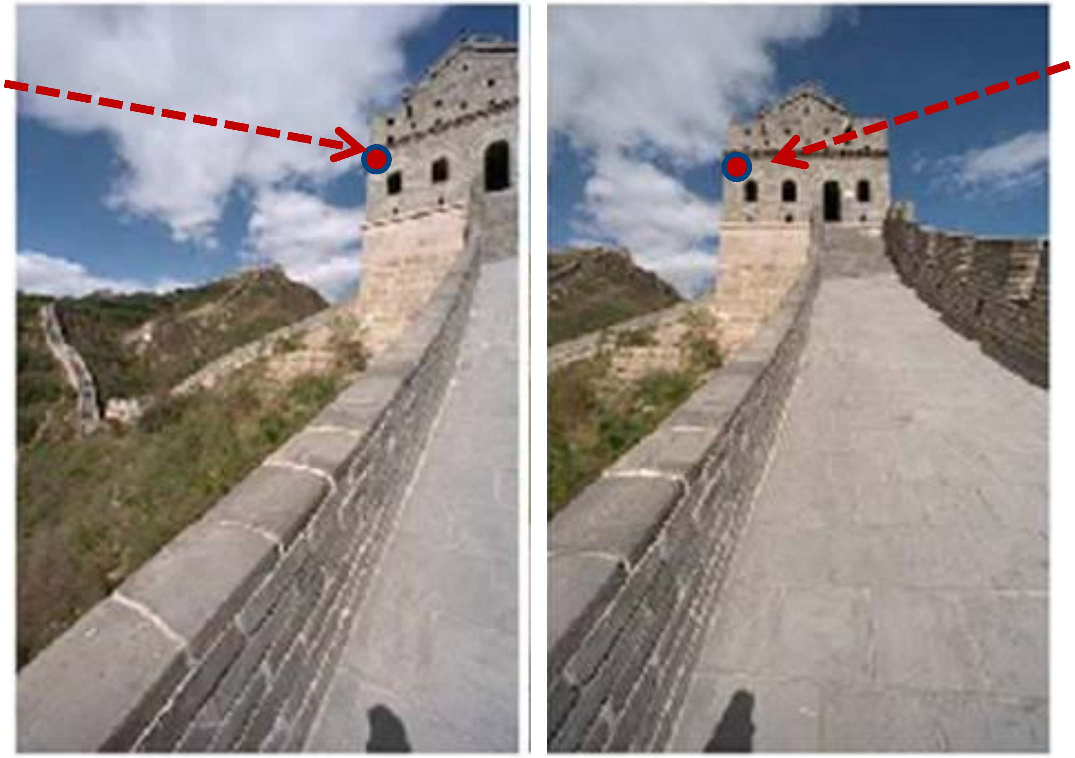
$$\underset{2n \times 9}{\mathrm{A}} \qquad \underset{9}{\mathrm{h}} \qquad \underset{2n}{\mathbf{0}}$$

Defines a least squares problem: $\quad \text{minimize } \|\mathbf{A}\mathbf{h} - \mathbf{0}\|^2$

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

Homography

$$\left(x, y\right)$$



$$\left({}^{wx'}\!/_{w}, {}^{wy'}\!/_{w}\right) = \left(x', y'\right)$$

To **apply** a given homography **H**

- Compute **p' = Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$

Kristen Grauman

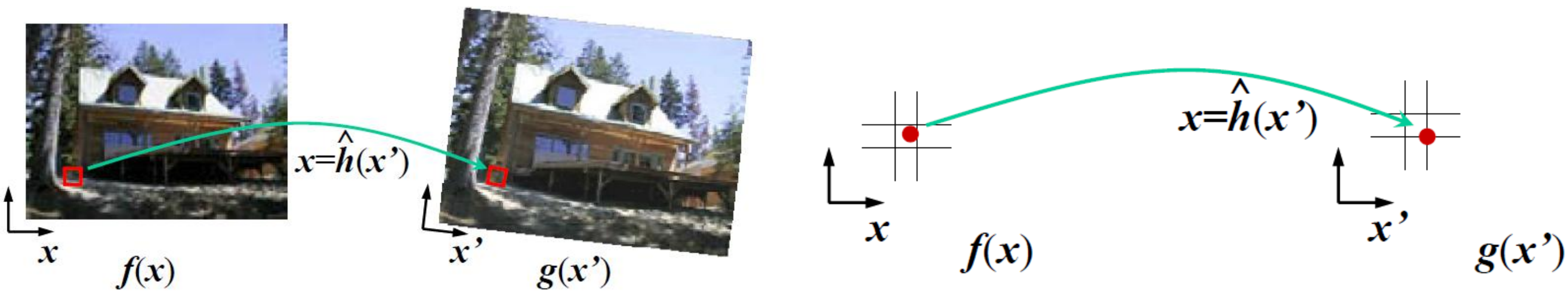
RANSAC for estimating homography

- RANSAC loop:
- 1. Select four feature pairs (at random)
- 2. Compute homography H (exact)
- 3. Compute *inliers* where $SSD(p_i', \mathbf{H}p_i) < \varepsilon$
- 4. Keep largest set of inliers
- 5. Re-compute least-squares H estimate on all of the inliers

Slide credit: Steve Seitz

Recall: Warping

- GPU/OpenGL rendering
- Inverse map with interpolation:



procedure $inverseWarp(f, \boldsymbol{h}, \textbf{out } g)$:

For every pixel $\boldsymbol{x}'$ in $g(\boldsymbol{x}')$

1. Compute the source location $\boldsymbol{x} = \hat{\boldsymbol{h}}(\boldsymbol{x}')$
2. Resample $f(\boldsymbol{x})$ at location $\boldsymbol{x}$ and copy to $g(\boldsymbol{x}')$

Recap: How to stitch together a panorama (a.k.a. mosaic)?

Basic Procedure

- Take a sequence of images from the same position
  - Rotate the camera about its optical center
- Compute transformation (homography) between second image and first using corresponding points.
- Transform the second image to overlap with the first.
- Blend the two together to create a mosaic.
- (If there are more images, repeat)

Source: Steve Seitz

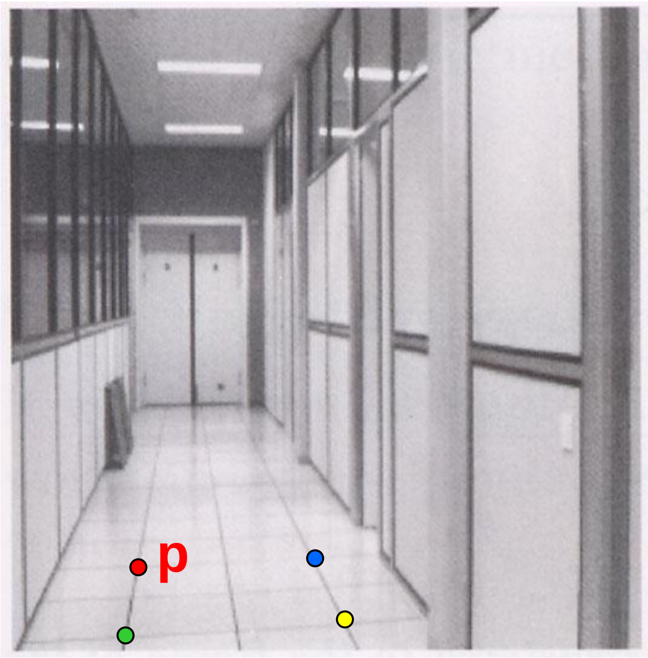
Image warping with homographies

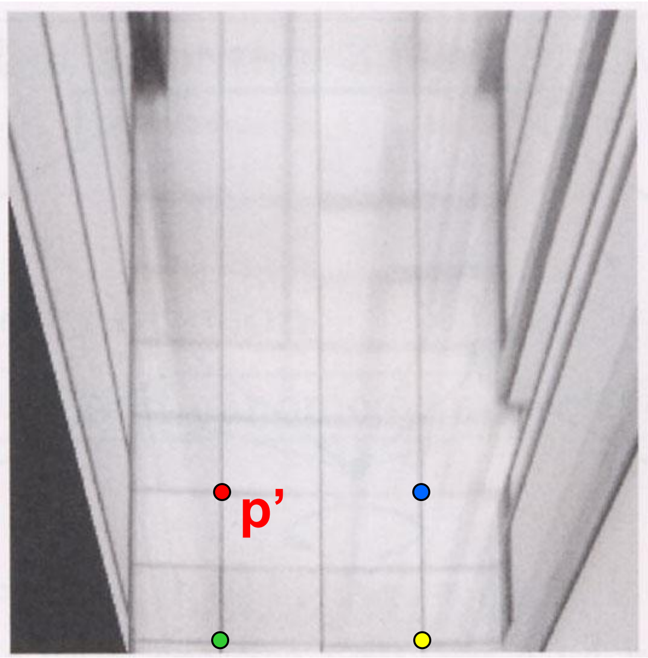
image plane in front

image plane below

Source: Steve Seitz

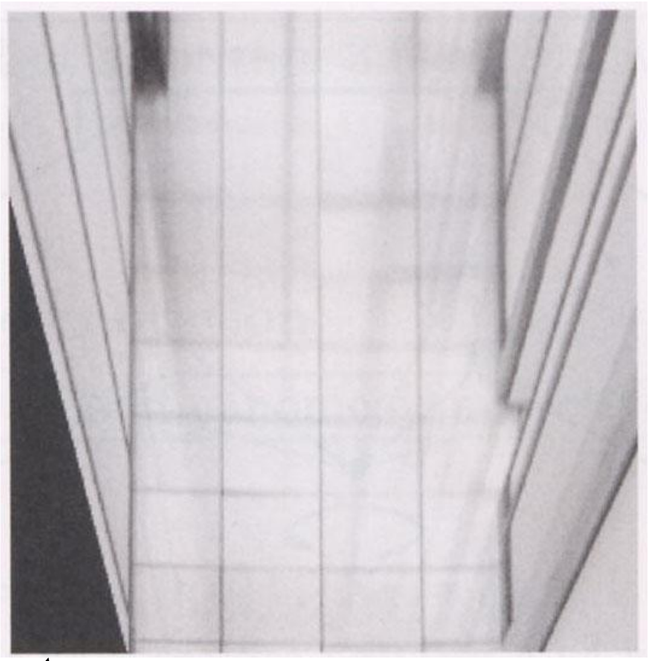
Image rectification





Source: Steve Seitz

Image warping with homographies



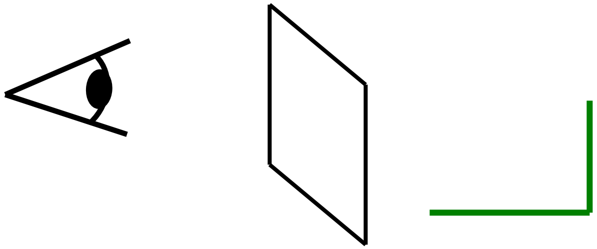


image plane in front

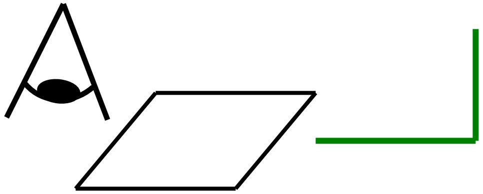
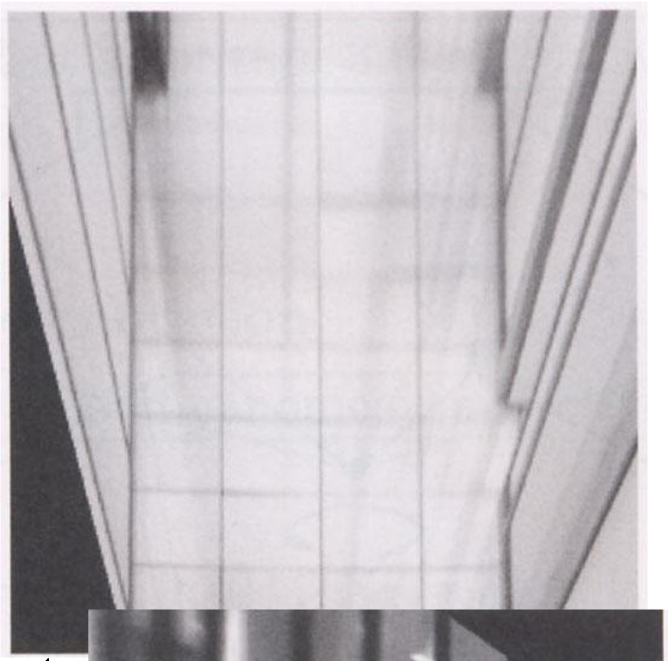


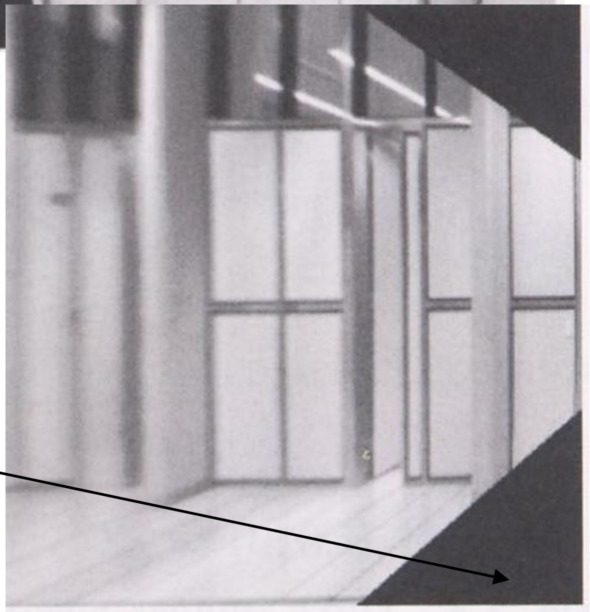
image plane below

black area
where no pixel
maps to

Source: Steve Seitz

Image warping with homographies





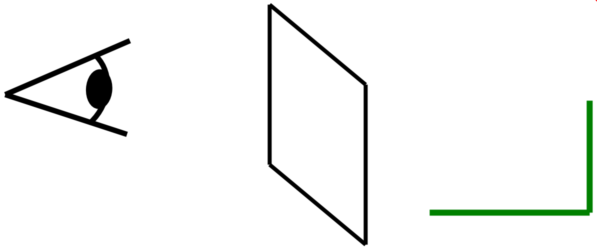


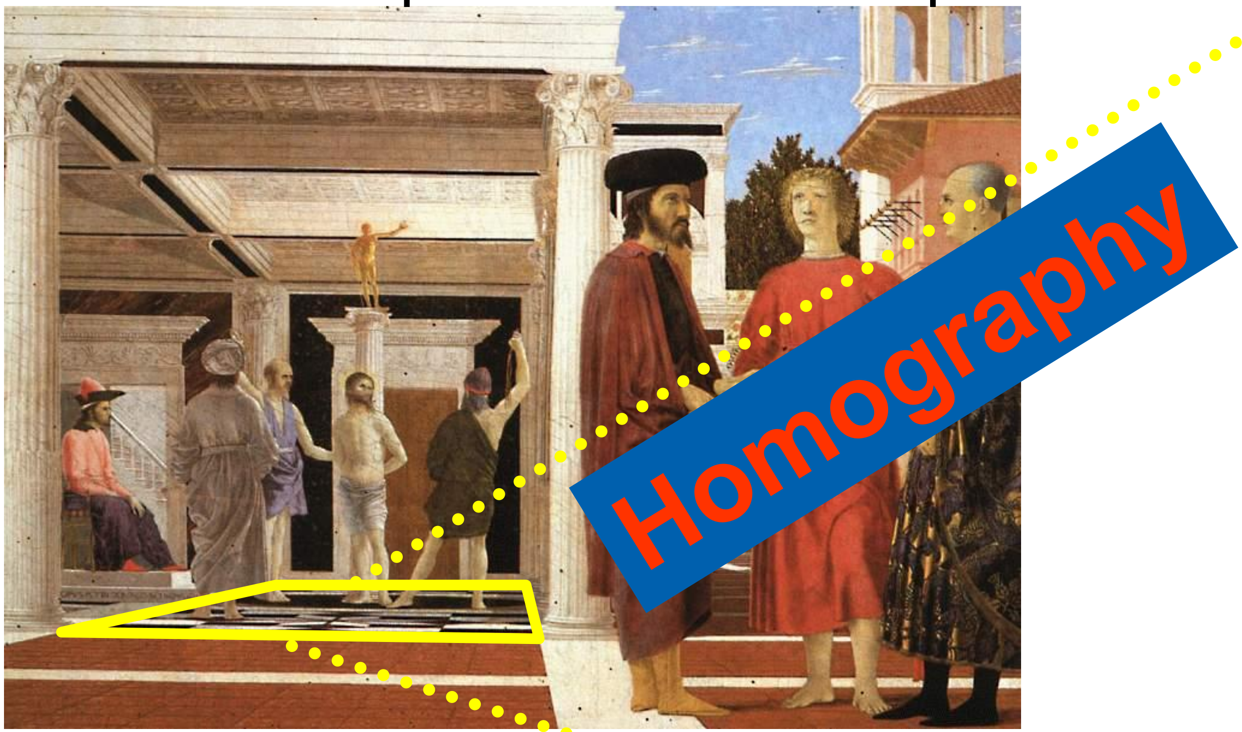
image plane in front

black area
where no pixel
maps to

Source: Steve Seitz

Analysing patterns and shapes

What is the shape of the b/w floor pattern?



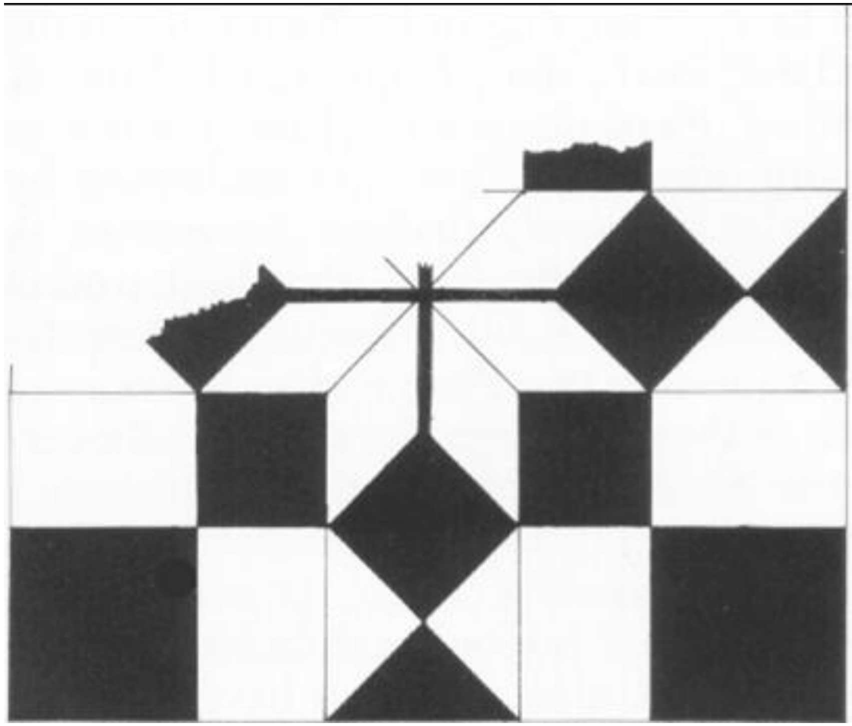
The floor (enlarged)

Automatically rectified floor

Slide from Antonio Criminisi

Analysing patterns and shapes

Automatic rectification



From Martin Kemp *The Science of Art*
(manual reconstruction)

Slide from Antonio Criminisi

Analysing patterns and shapes

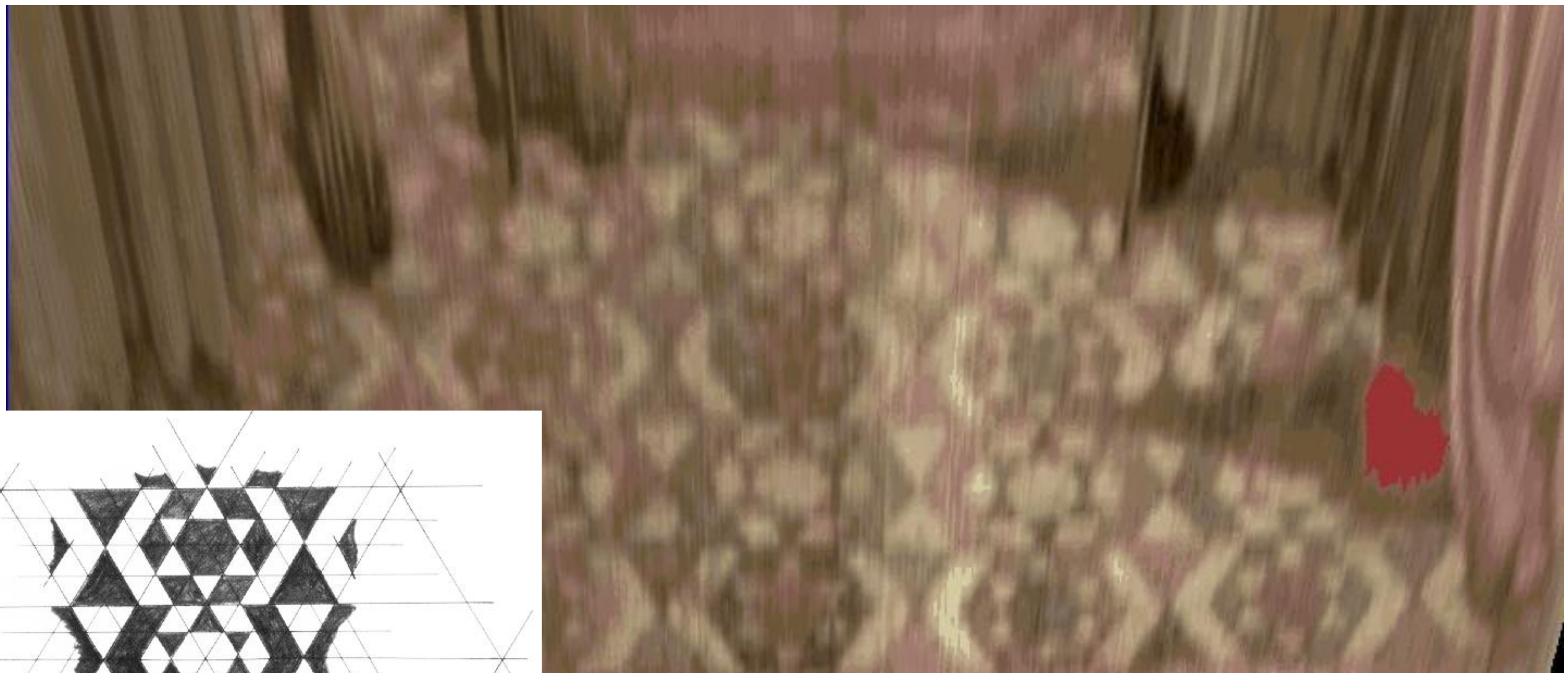
St. Lucy Altarpiece, D. Veneziano

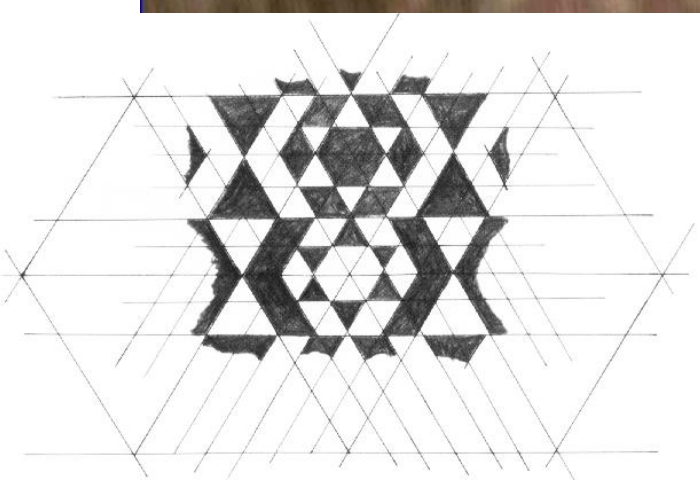
Automatically rectified floor

Slide from Criminisi

Analysing patterns and shapes

Automatic
rectification



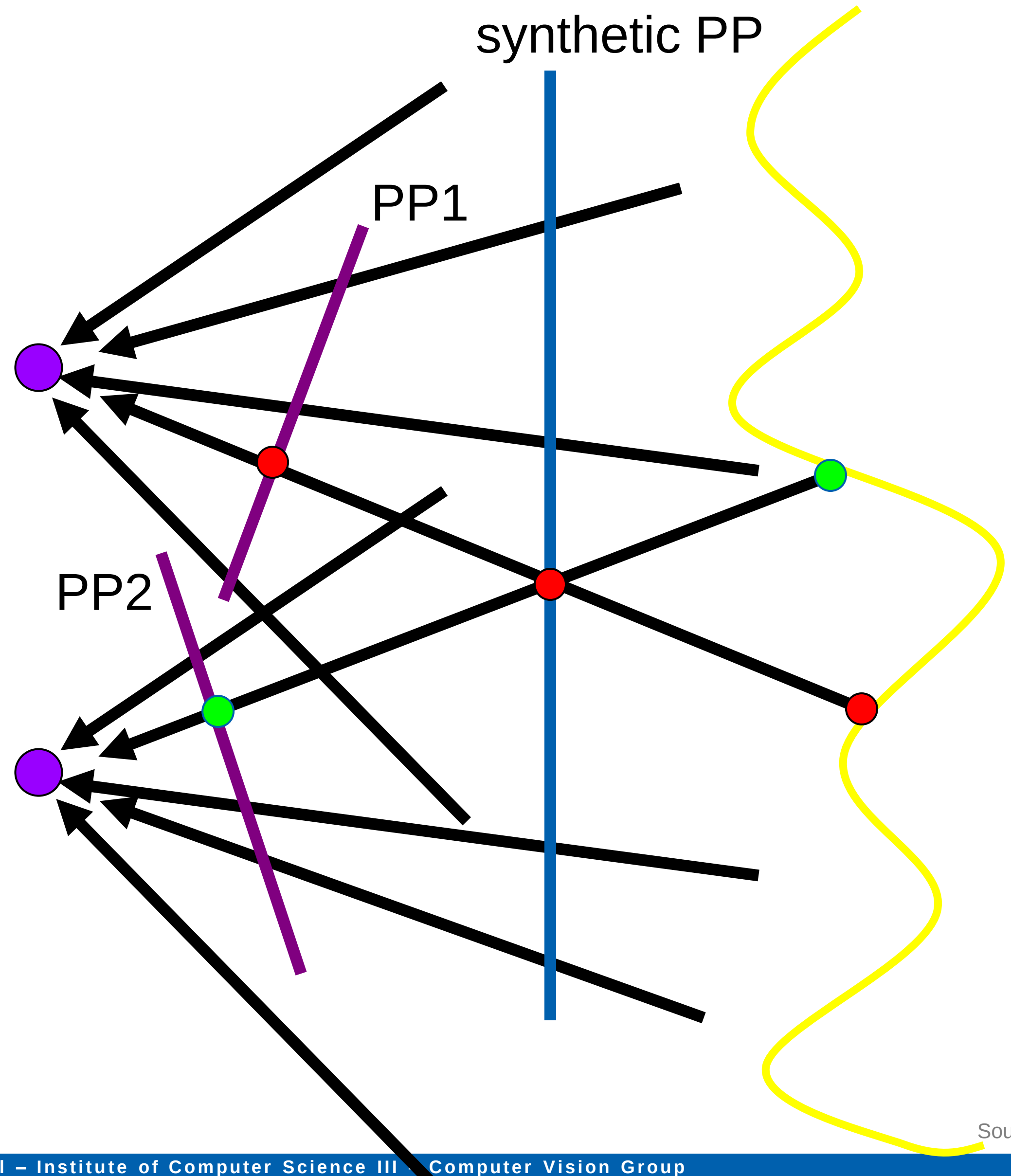


From Martin Kemp, *The Science of Art*
(manual reconstruction)

Slide from Criminisi

Changing camera center

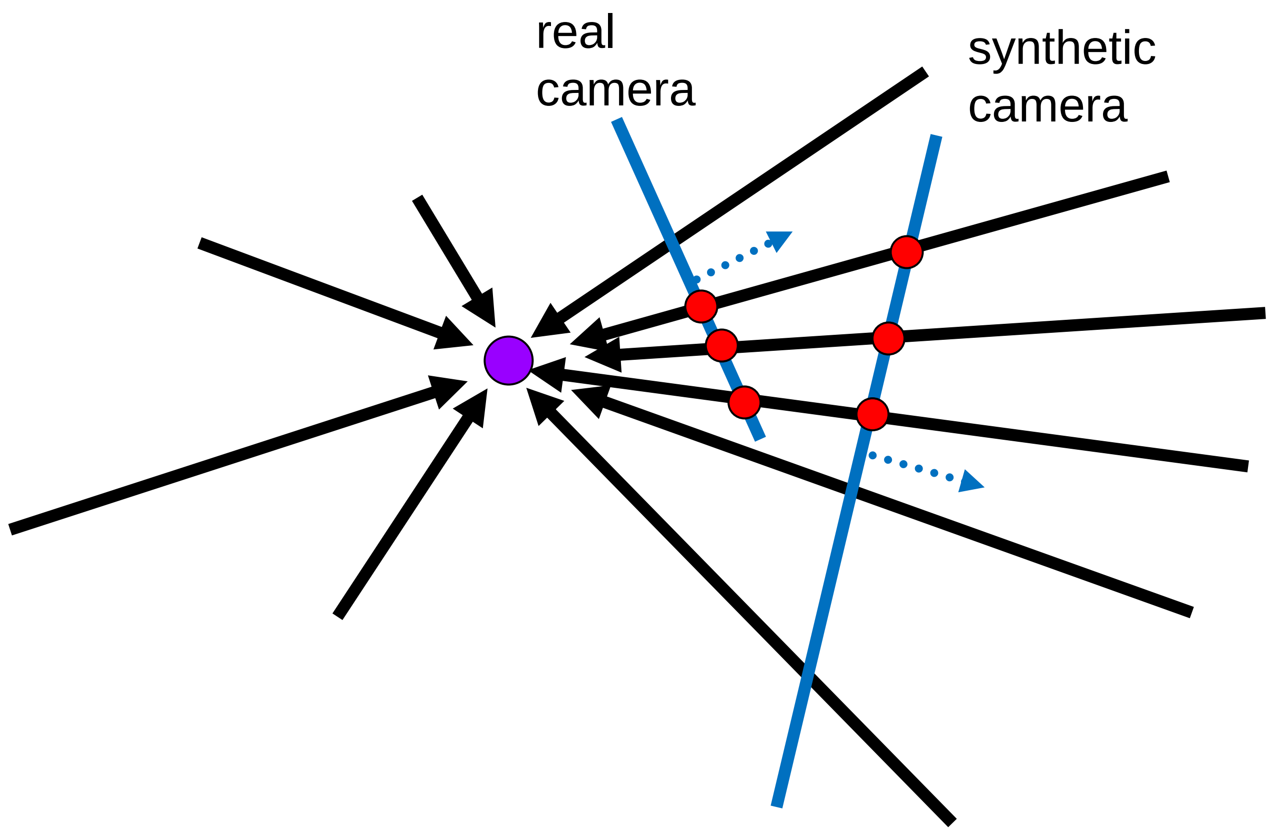
Does it still work?



Source: Alyosha Efros

Recall: same camera center

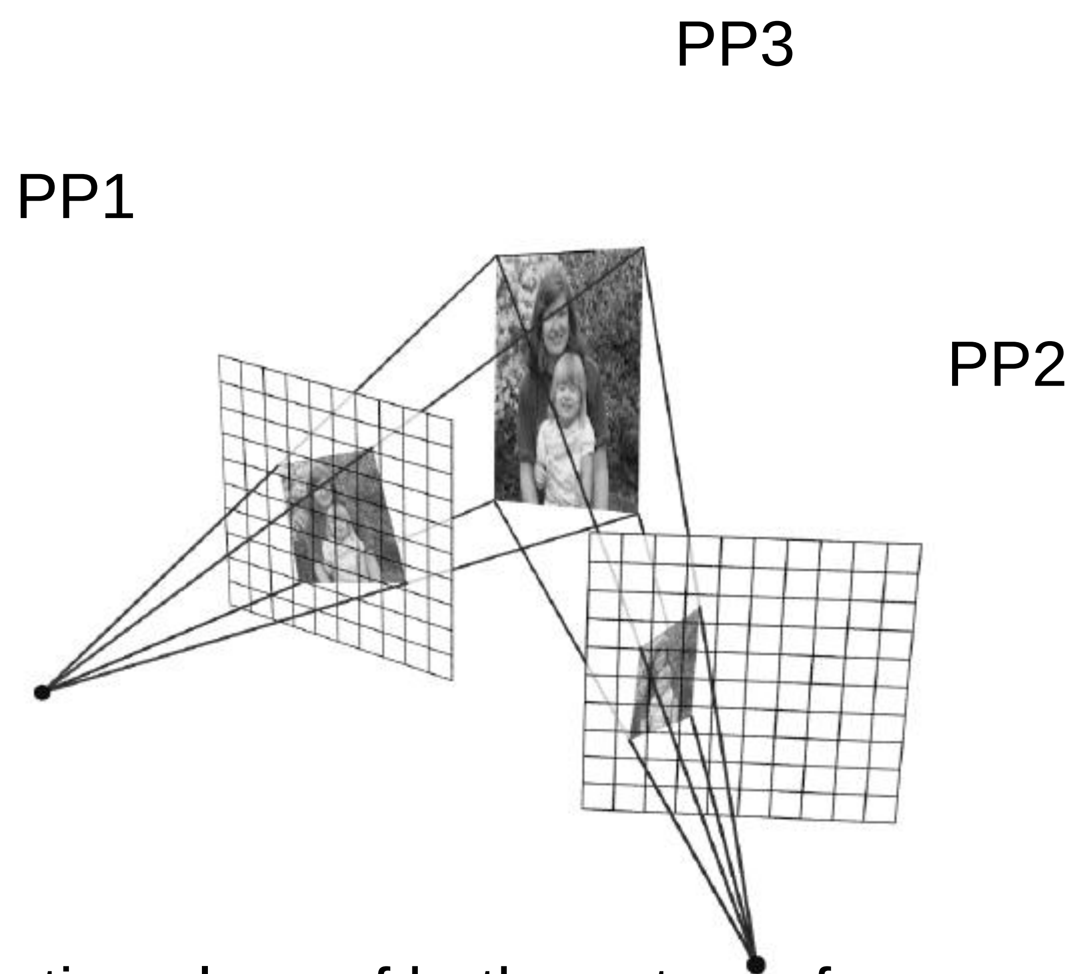
UNIVERSITÄT BONN



Can generate synthetic camera view
as long as it has **the same center of projection**.

Source: Alyosha Efros

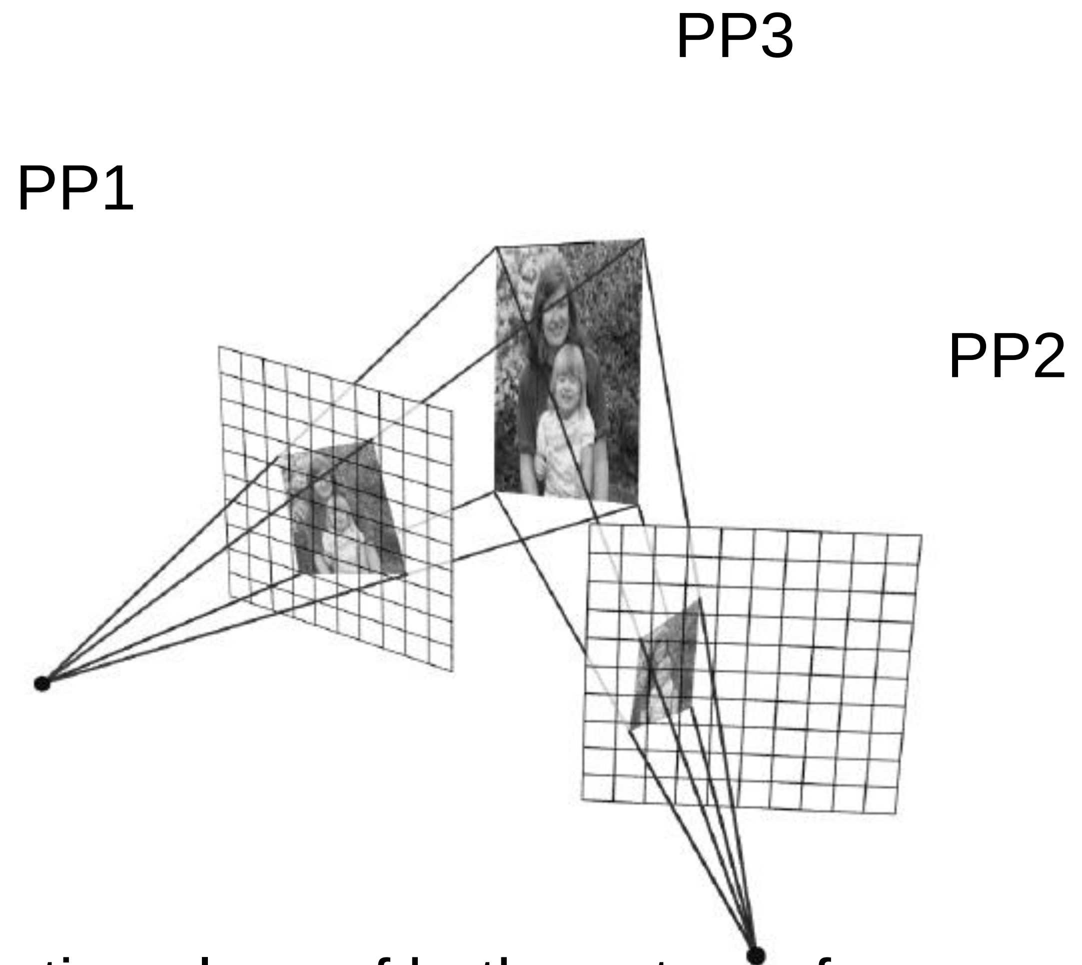
…or planar scene



- PP3 is a projection plane of both centers of projection, so we are OK!

Source: Alyosha Efros

…or planar scene (or far away)



- PP3 is a projection plane of both centers of projection, so we are OK!
- This is how big aerial photographs are made

Source: Alyosha Efros


Map
Satellite
Hybrid
200 ft
100 m
©2007 Google
Kristen Grauman

UNIVERSITÄT BONN