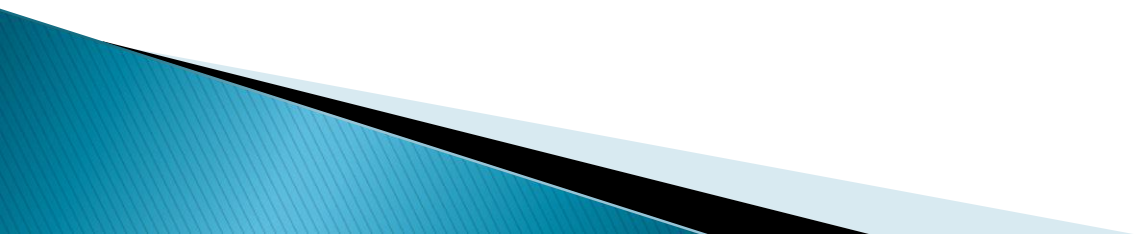
Image Transformation and RANSAC

Image Transformation Motivation



Motivation: Naver Street View

Motivation: Image Stitching

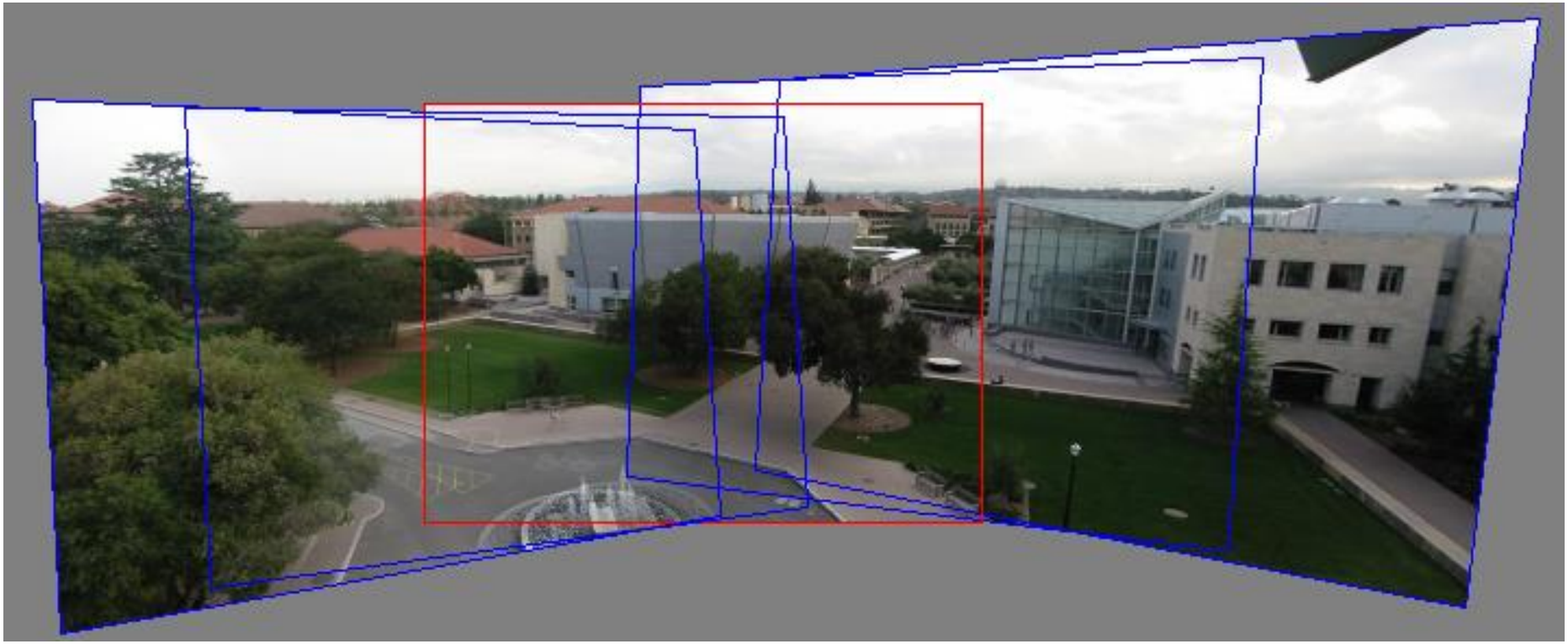
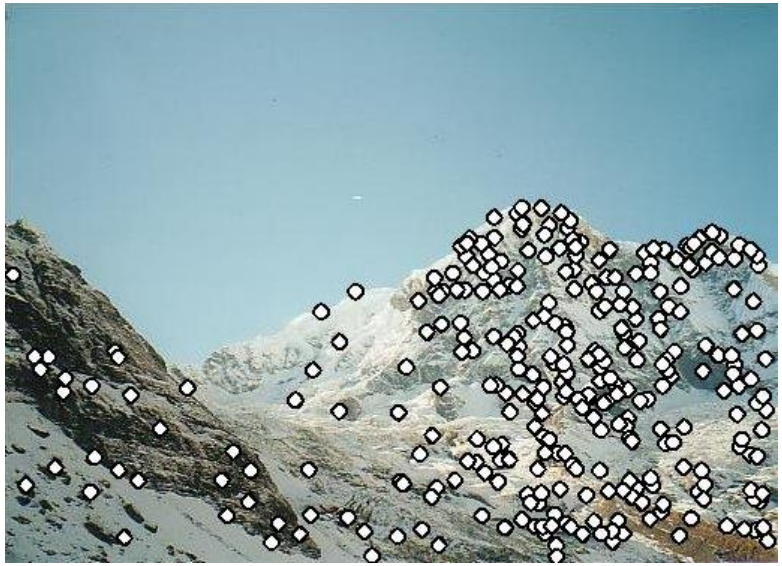
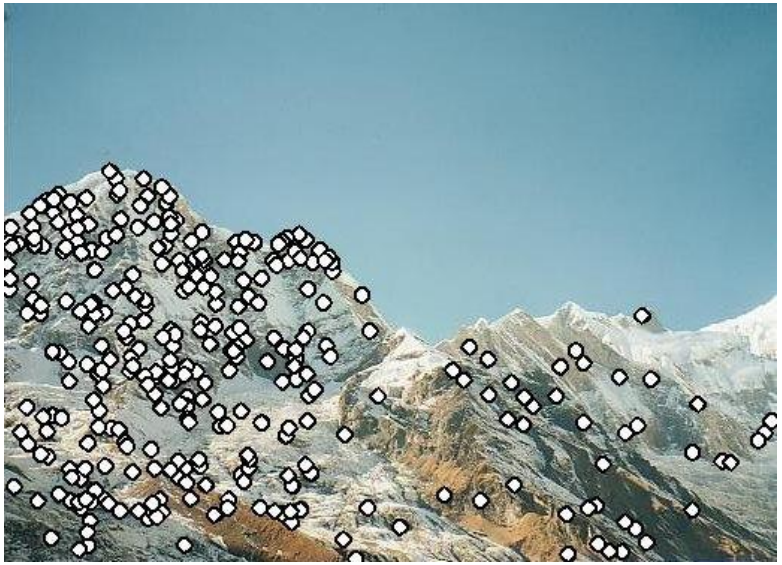


Image from http://graphics.cs.cmu.edu/courses/15-463/

Robust feature-based alignment

Coming up: robust feature-based alignment





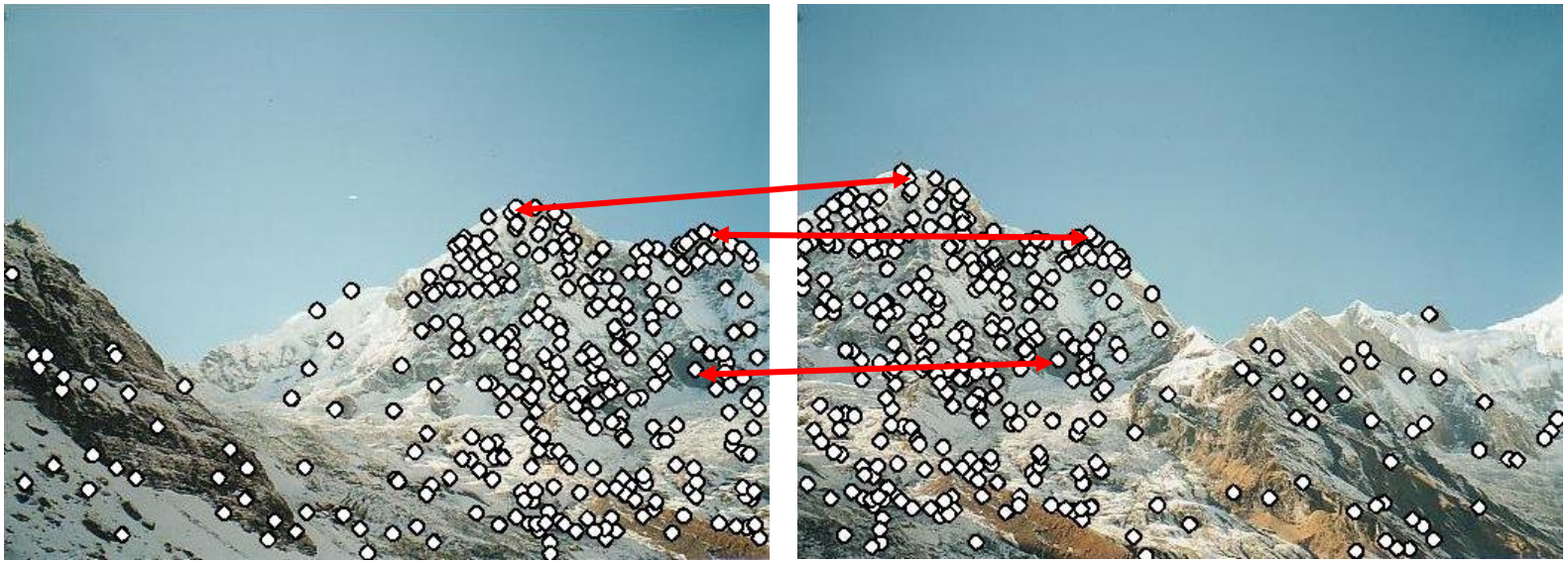
- Extract features

Source: L. Lazebnik

Coming up: robust feature-based alignment

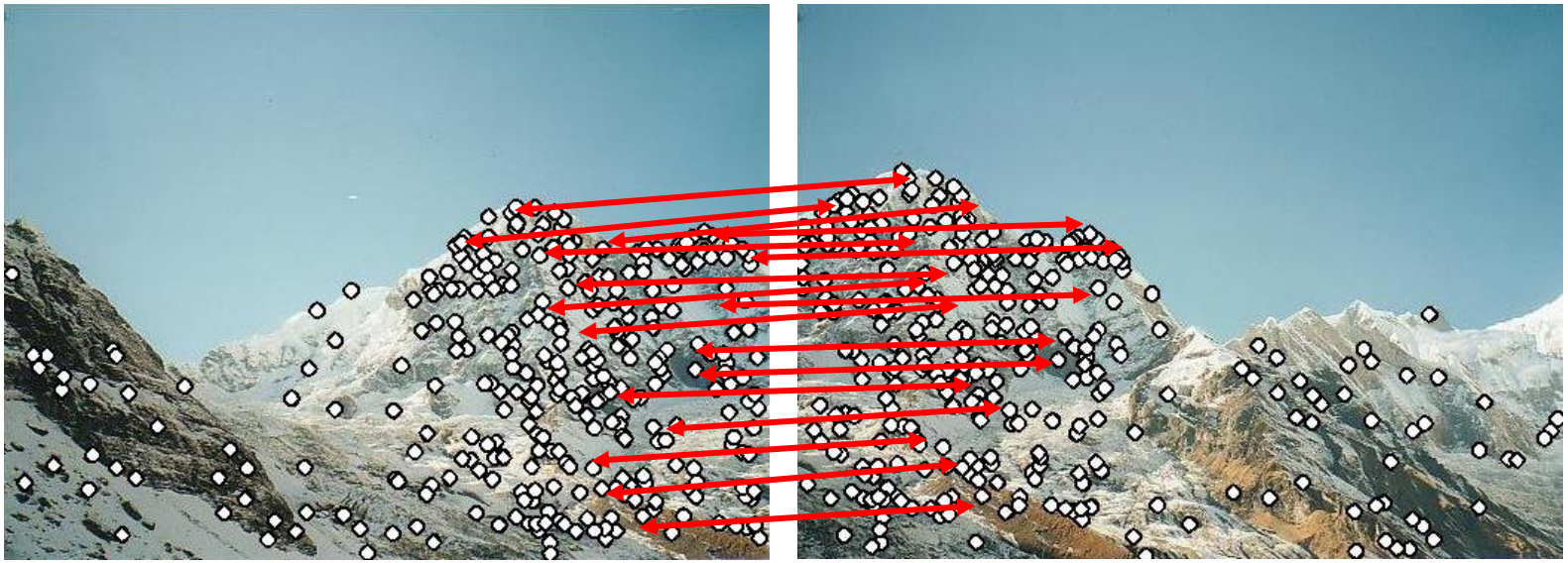
- Extract features
- Compute *putative matches*

Source: L. Lazebnik



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

Source: L. Lazebnik



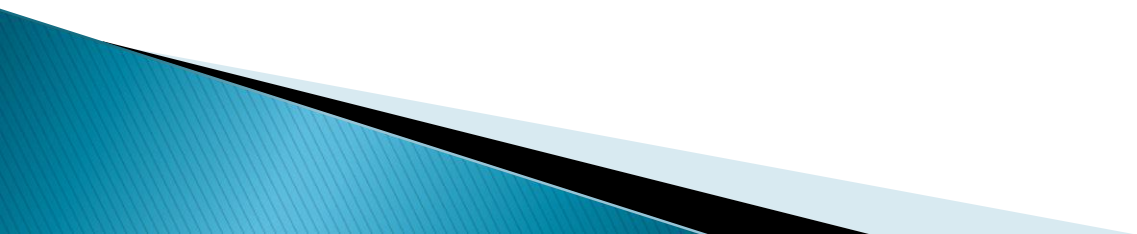
- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
  - *Verify* transformation (search for other matches consis tent with *T*)

Source: L. Lazebnik

- 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 consis tent with *T*)

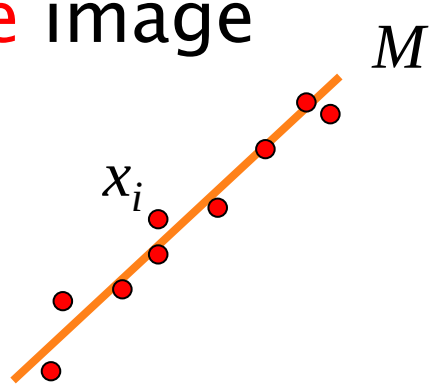
Source: L. Lazebnik

2D Transformation



Alignment as fitting

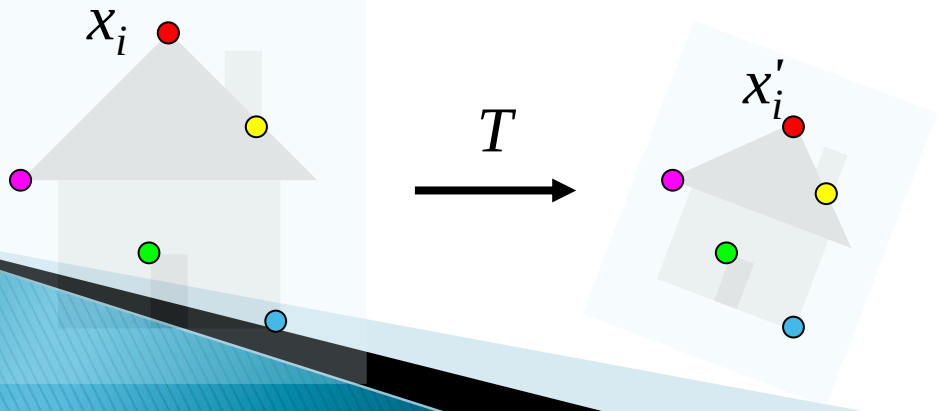
- Previous lectures: fitting a model to features in one image



Find model M that minimizes

$$\sum_i \text{residual}(x_i, M)$$

- Alignment: fitting a transformation between pairs of features (*matches*) in two images



Find transformation T that minimizes

$$\sum_i \text{residual}(T(x_i), x_i')$$

Adapted from: Lana Lazebnik

Parametric (global) warping

- Examples of parametric warps:

translation

rotation

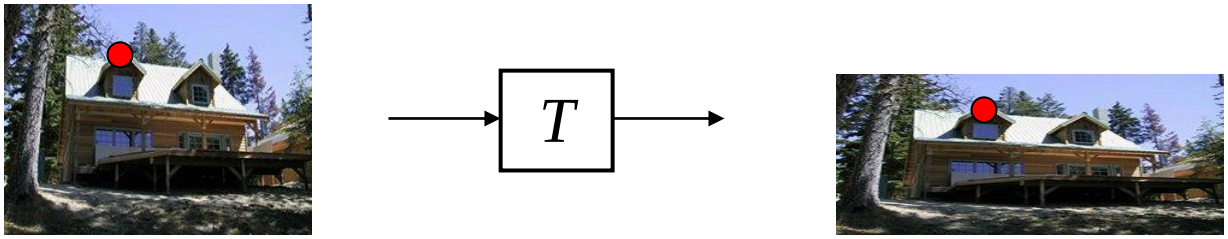
aspect

affine

perspective

Source: Alyosha Efros

Parametric (global) warping



$\mathbf{p} = (x,y)$ $\mathbf{p'} = (x',y')$

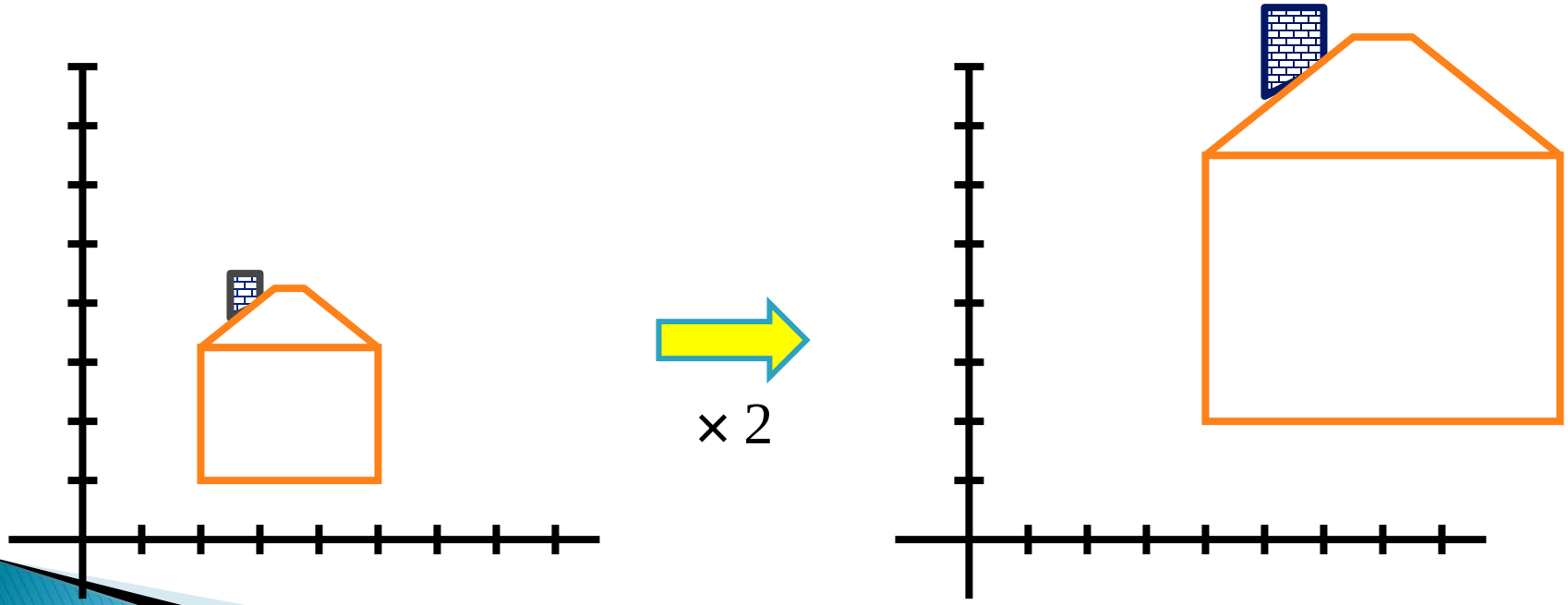
- Transformation T is a coordinate-changing mac hine: $p' = T(p)$
- What does it mean that T is **global**?
  - Is the same for any point p
  - can be described by just a few numbers (parameters)
- Let's represent T as a matrix: $p' = \mathbf{M}p$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Alyosha Efros

Scaling

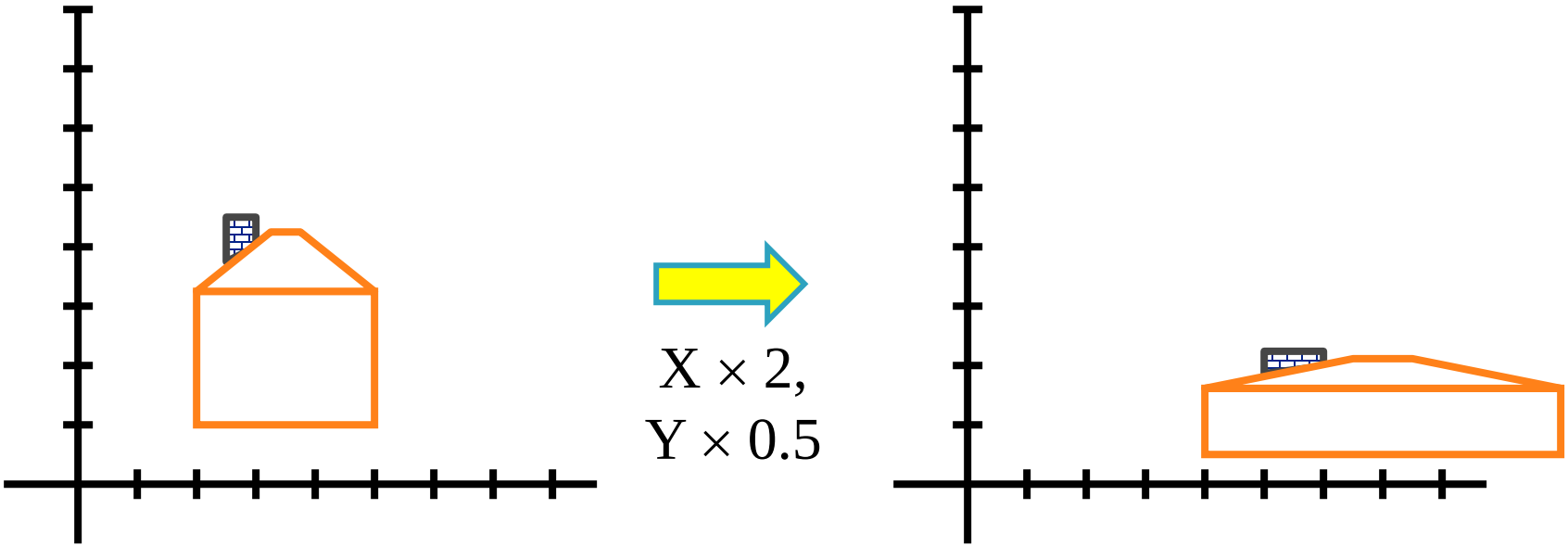
- *Scaling* a coordinate means multiplying each of its components by a scalar
- *Uniform scaling* means this scalar is the same for all components:



Slide credit: Kristen Grauman

Scaling

- *Non-uniform scaling*: different scalars per component:



Slide credit: Kristen Grauman

Scaling

- Scaling operation: $x' = ax$

  $y' = by$

- Matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\textit{scaling matrix } S} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Alyosha Efros

What transformations can be represented with a 2x2 matrix?

2D Scaling?

$$x' = s_x * x$$
$$y' = s_y * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Rotate around (0,0)?

$$x' = \cos\Theta * x - \sin\Theta * y$$
$$y' = \sin\Theta * x + \cos\Theta * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Shear?

$$x' = x + sh_x * y$$
$$y' = sh_y * x + y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Slide credit: Kristen Grauman

What transformations can be represented with a 2x2 matrix?

2D Mirror about Y axis?

$$x' = -x$$
$$y' = y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Mirror over (0,0)?

$$x' = -x$$
$$y' = -y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Translation?

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}_x$$
$$\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{t}_y$$

NO!

Slide credit: Kristen Grauman

2D Linear Transformations

- Only linear 2D transformations can be represented with a 2x2 matrix.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Linear transformations are combinations of …
  - Scale,
  - Rotation,
  - Shear, and
  - Mirror

Source: Alyosha Efros

Homogeneous coordinates

To convert to homogeneous coordinates:

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

homogeneous image coordinates

Converting *from* homogeneous coordinates

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

Slide credit: Kristen Grauman

Homogeneous coordinates

- Q: How can we represent 2d translation as a 3 x3 matrix using homogeneous coordinates?

$$x' = x + t_x$$

$$y' = y + t_y$$

- A: Using the rightmost column:

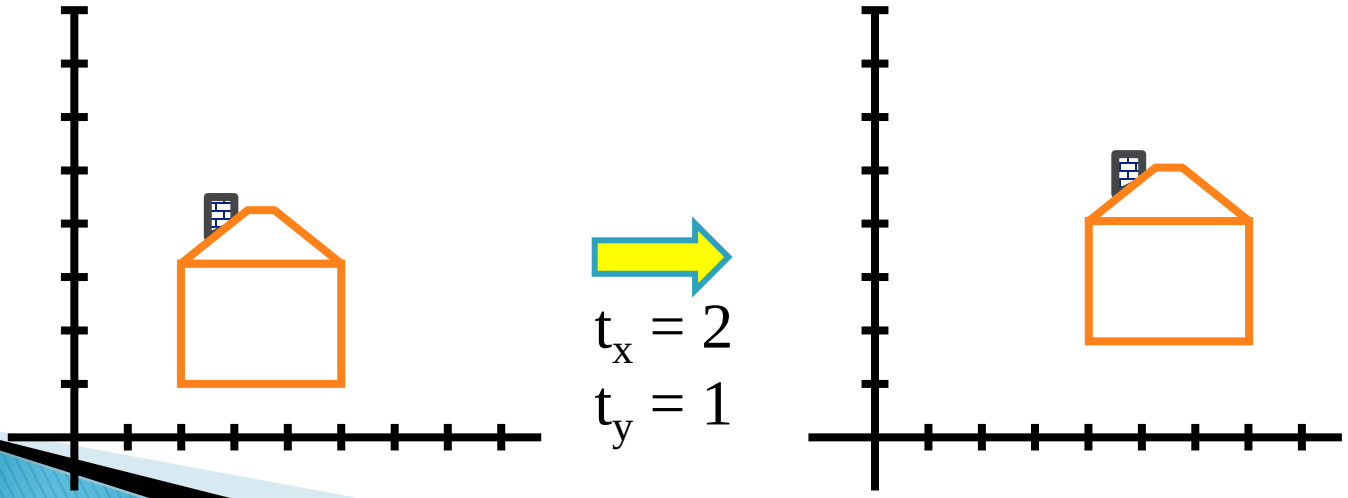
$$\mathbf{Translation} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Source: Alyosha Efros

Translation

Homogeneous Coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$



Source: Alyosha Efros

Basic 2D Transformations

- Basic 2D transformations as 3x3 matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{sh}_x & 0 \\ \boldsymbol{sh}_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Shear

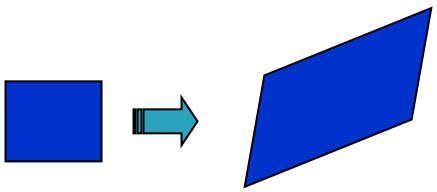
Slide credit: Kristen Grauman

2D Affine Transformations

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

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

- Parallel lines remain parallel



Slide credit: Kristen Grauman

Transformation Fitting

Alignment problem

- We have feature points and correspodnece (a set of matching feature pairs)
- Now, we want to find unknown parameters of trans formation.
- How do we calculate the parameters?

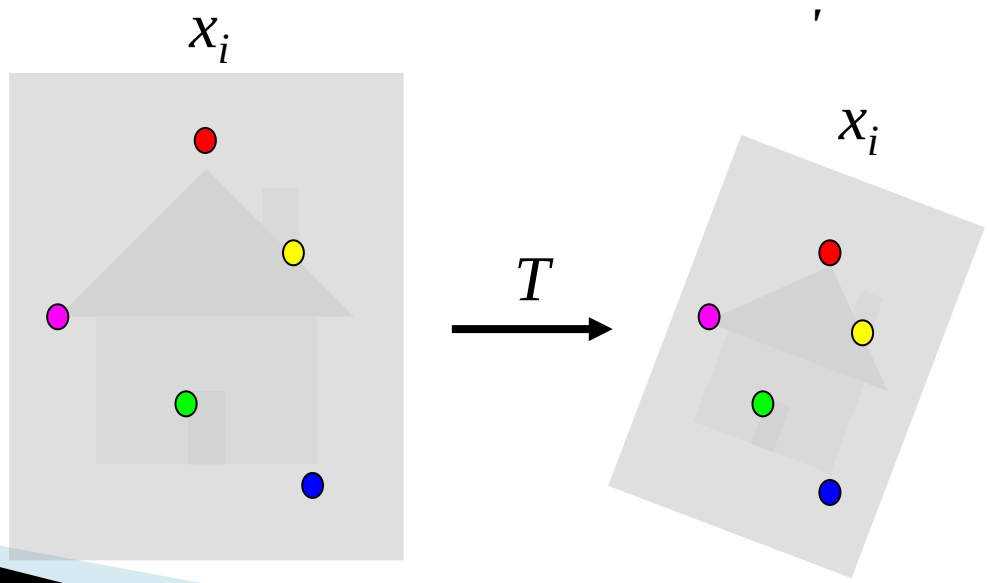
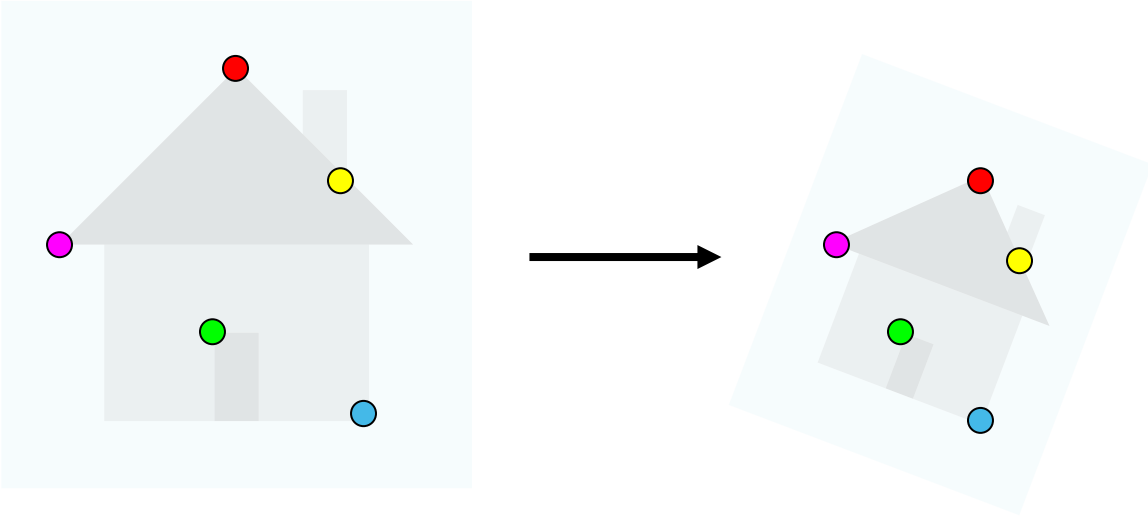


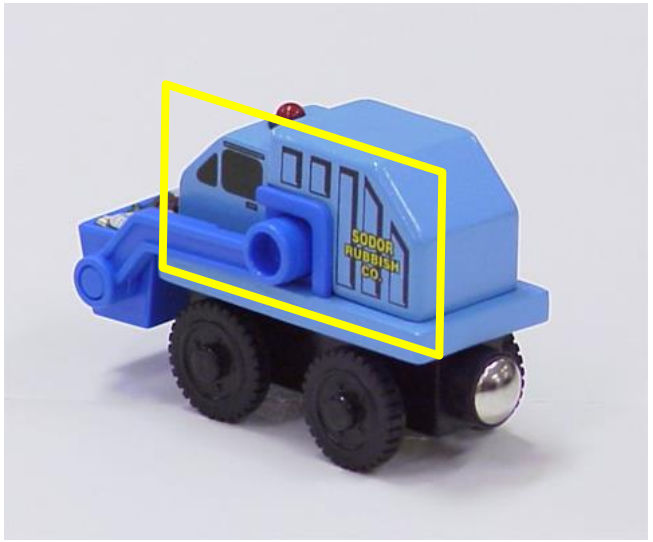
Image alignment



- Two broad approaches:
  - Direct (pixel-based) alignment
    - Search for alignment where most pixels agree
  - Feature-based alignment
    - Search for alignment where *extracted features* agree
    - Can be verified using pixel-based alignment

Let's start with affine transformations

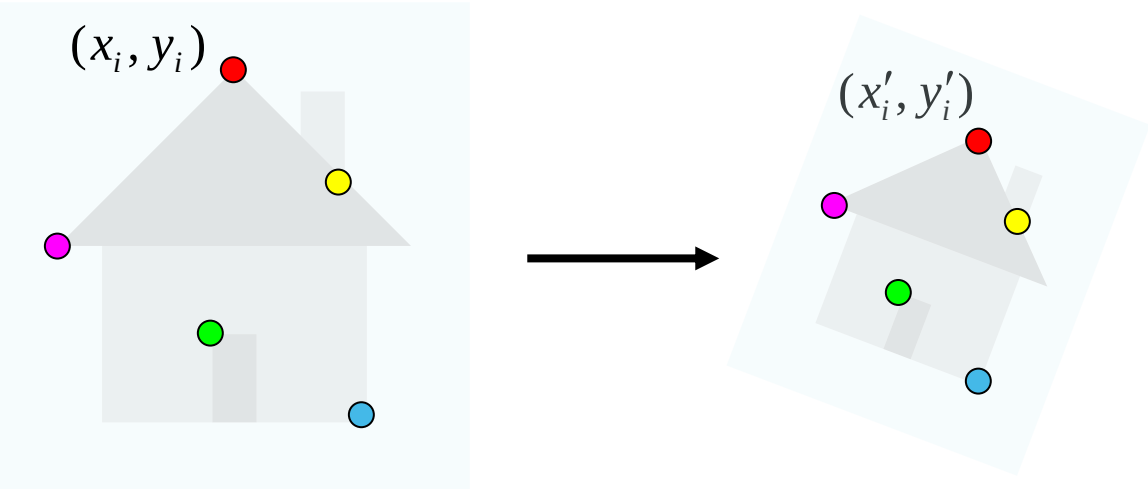
- Simple fitting procedure (linear least squares)
- Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras



Slide credit: Lana Lazebnik

Fitting an affine transformation

- Assuming we know the correspondences, how do w e get the transformation?



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

Slide credit: Lana Lazebnik

Fitting an affine transformation

- We have a set of a set of matching feature pairs.
- We want to find a formula to transform from (x,y) to (x',y').
- How do we get the transformation?

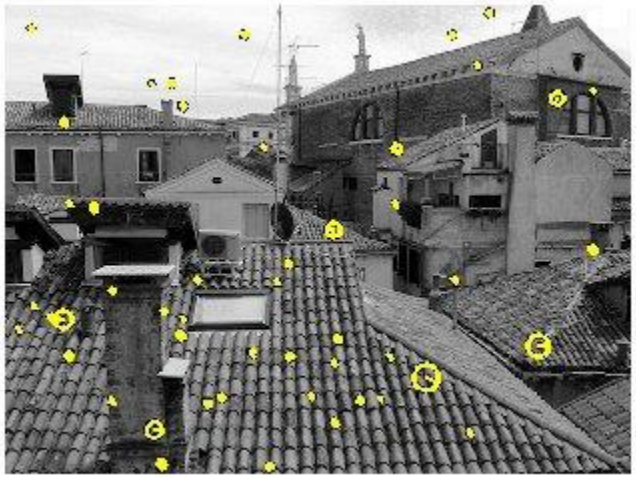
$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \Rightarrow \begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots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

Fitting an affine transformation

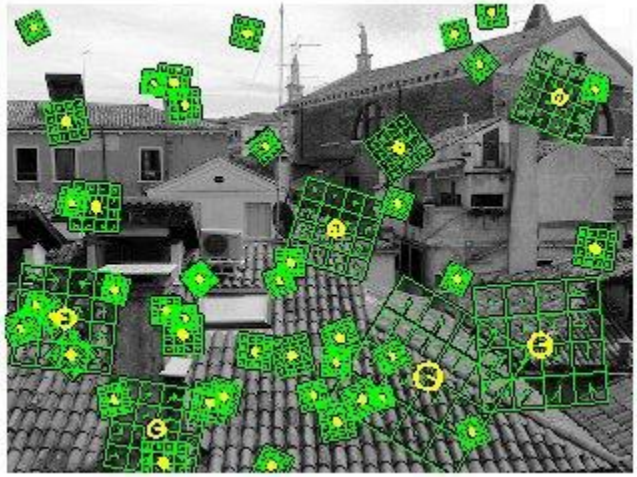
$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

- How many matches (correspondence pairs) do we need to solve for the transformation parameters?
- What if we have correspondence pairs muc h more than unknown parameters?

Recall: Scale Invariant Feature Transform (SIFT) descriptor [Lowe 2004]



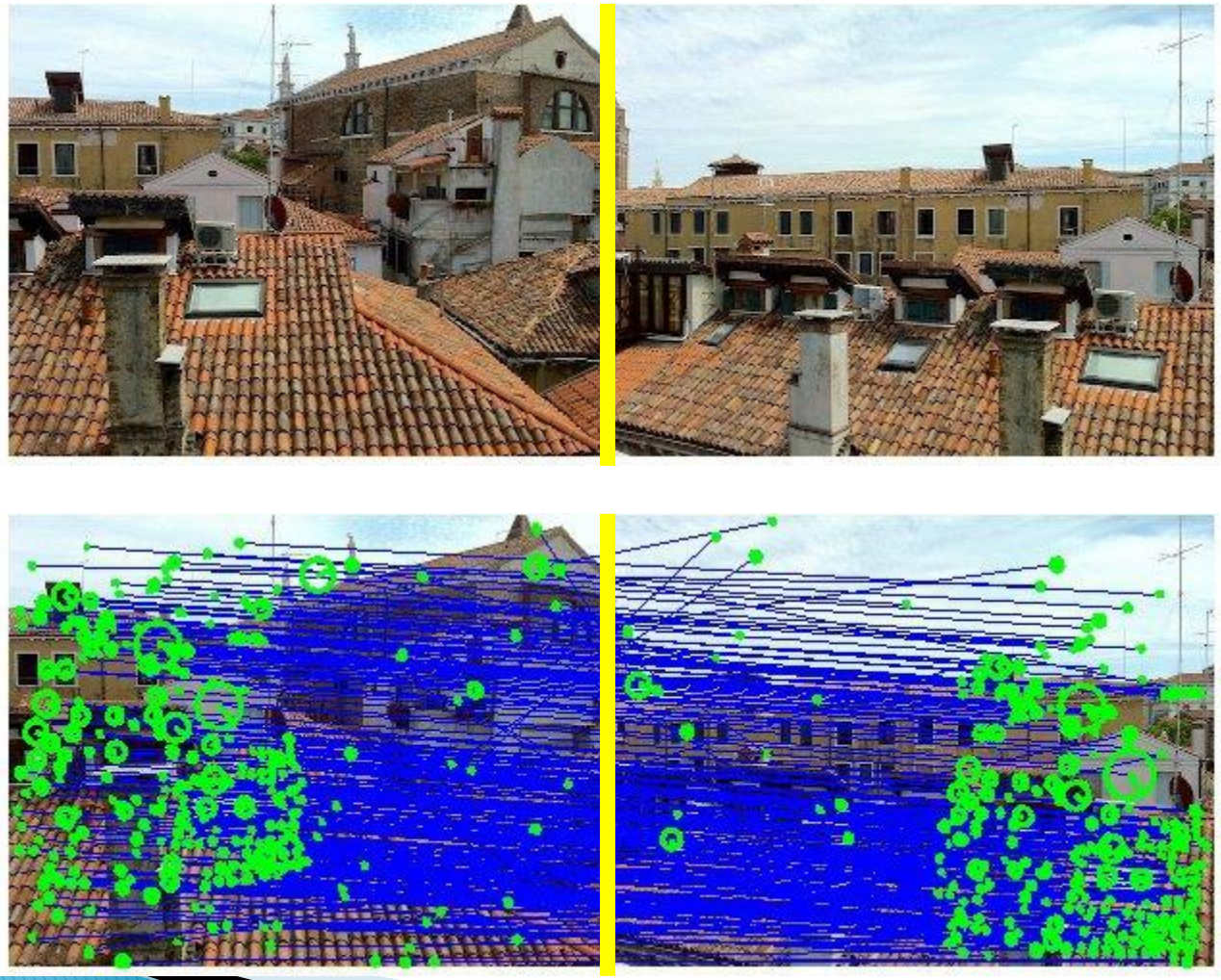
Interest points and their scales and orientations (random subset of 50)



SIFT descriptors

http://www.vlfeat.org/overview/sift.html

Recall: SIFT (preliminary) matches



http://www.vlfeat.org/overview/sift.html

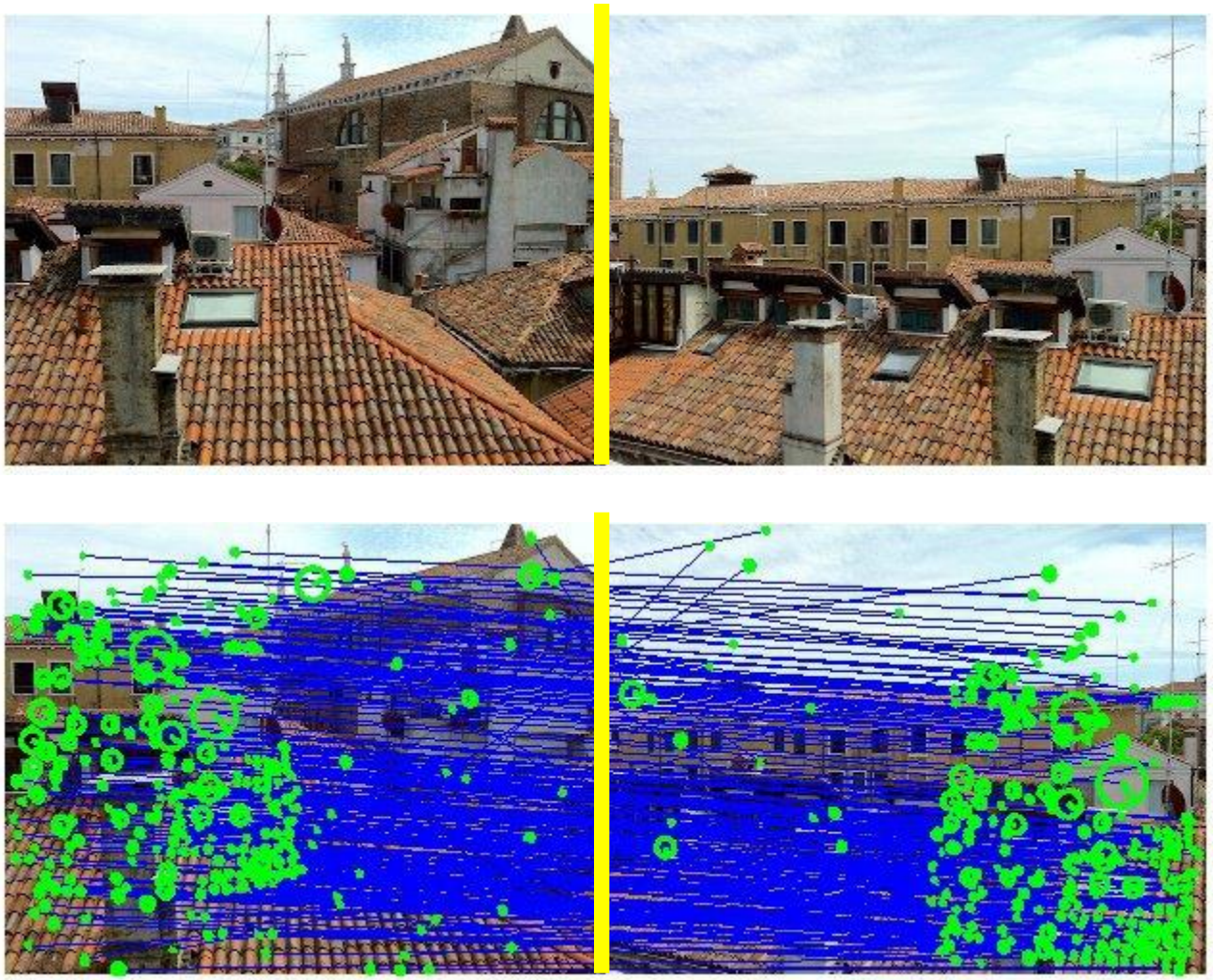
Kristen Grauman

Fitting an affine transformation

Figures from David Lowe, ICCV 1999

RANSAC

Recall: SIFT (preliminary) matches



Not all of thes e are valid ma tches!

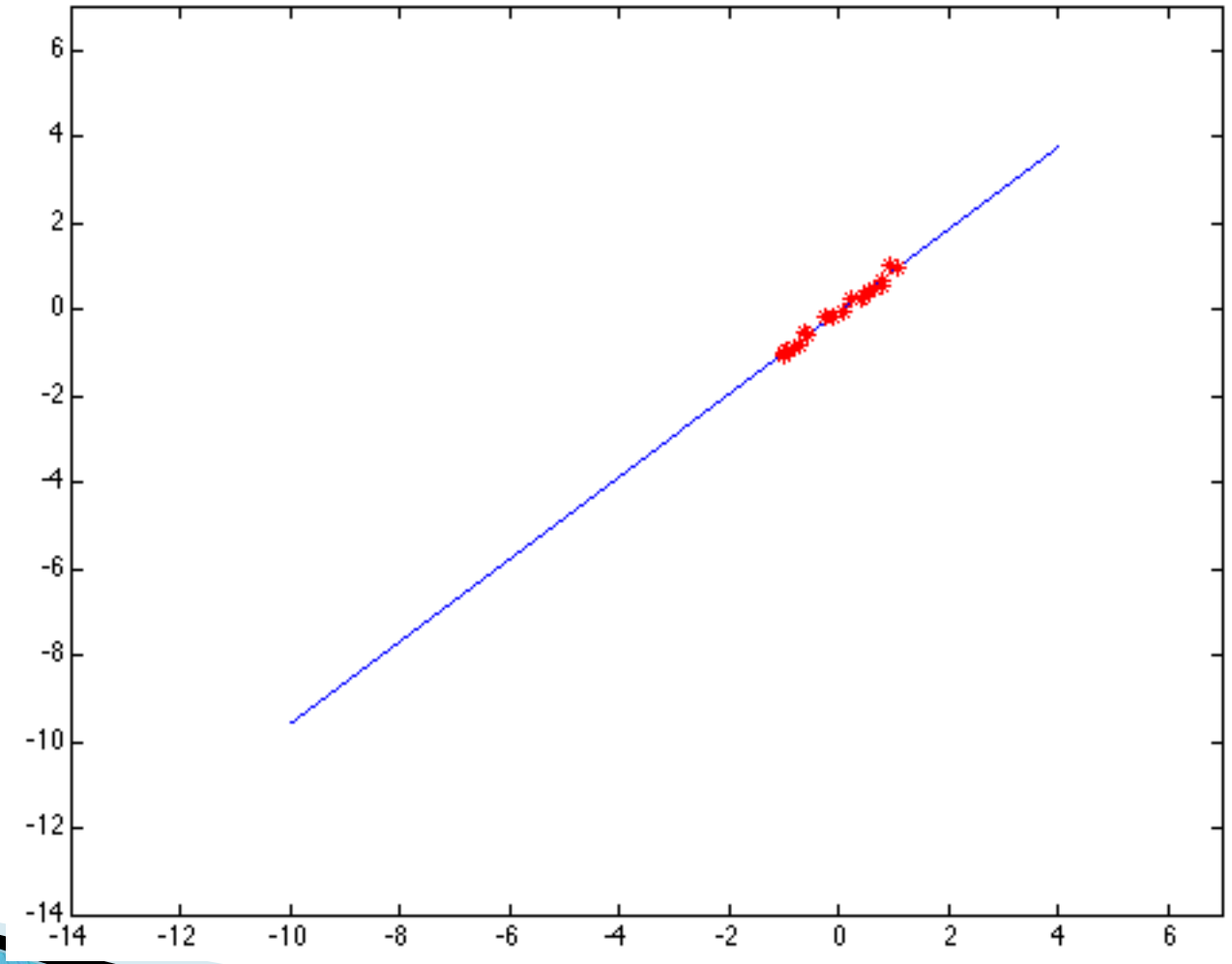
http://www.vlfeat.org/overview/sift.html

Kristen Grauman

Outliers

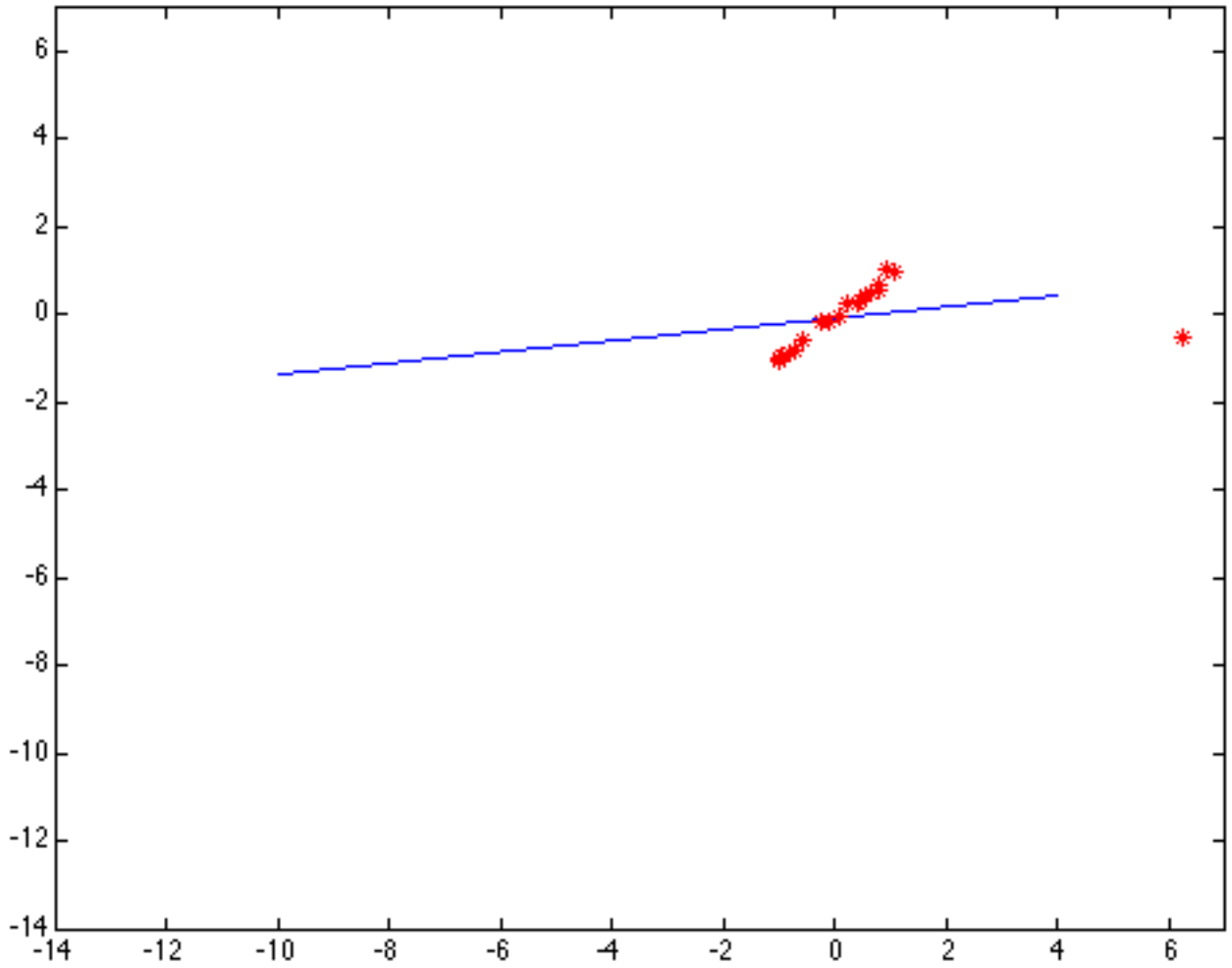
- **Outliers** can hurt the quality of our paramet er estimates.
  - an erroneous pair of matching points from two i mages
  - an edge point that is noise, or doesn't belong to t he line we are fitting.

Outliers affect least squares fit



Kristen Grauman

Outliers affect least squares fit



Kristen Grauman

RANSAC

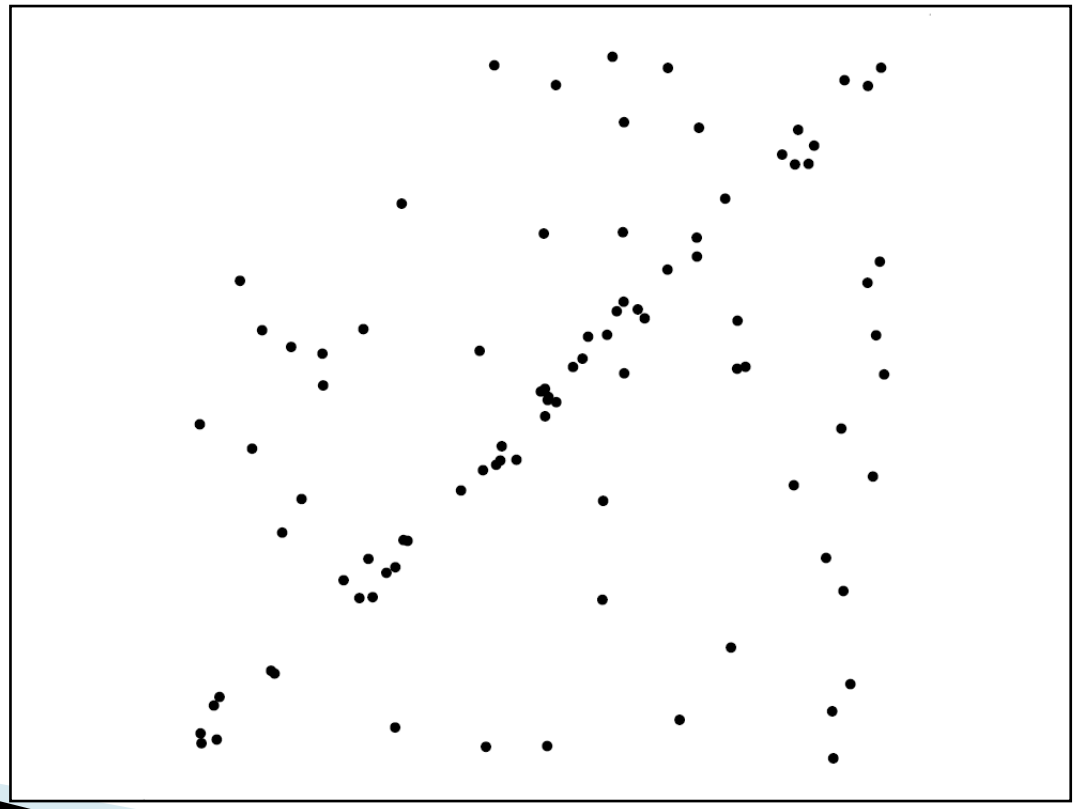
- RANdom Sample Consensus
- **Approach**: we want to avoid the impact of outliers, so let's look for "inliers", and use those only.
- **Intuition**: if an outlier is chosen to compute the current fit, then the resulting line (transformation) won't have much support from rest of the points (matches).

RANSAC for line fitting

- Randomly select a seed group of points on which t o base transformation estimate (e.g., a group of m atches)
- Compute transformation from seed group
- Find inliers to this transformation
- If the number of inliers is sufficiently large, re-co mpute least-squares estimate of transformation o n all of the inliers

• Keep the transformation with the largest number of inliers

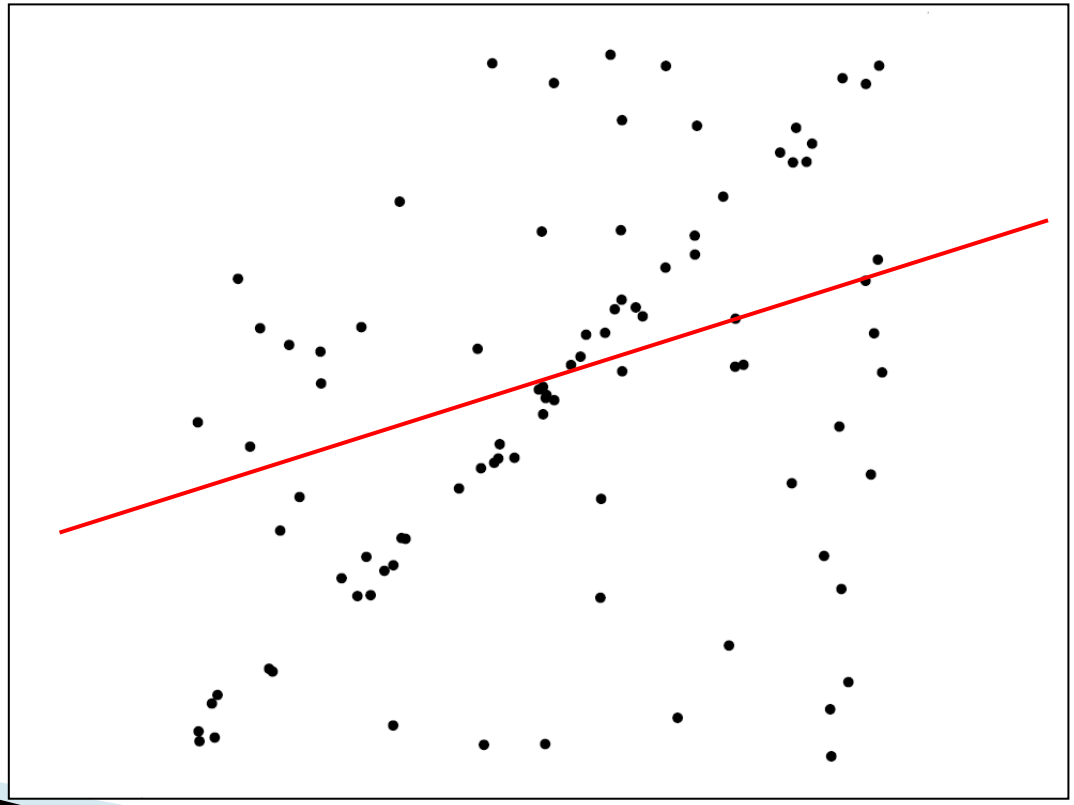
RANSAC for line fitting example



Source: R. Raguram

Lana Lazebnik

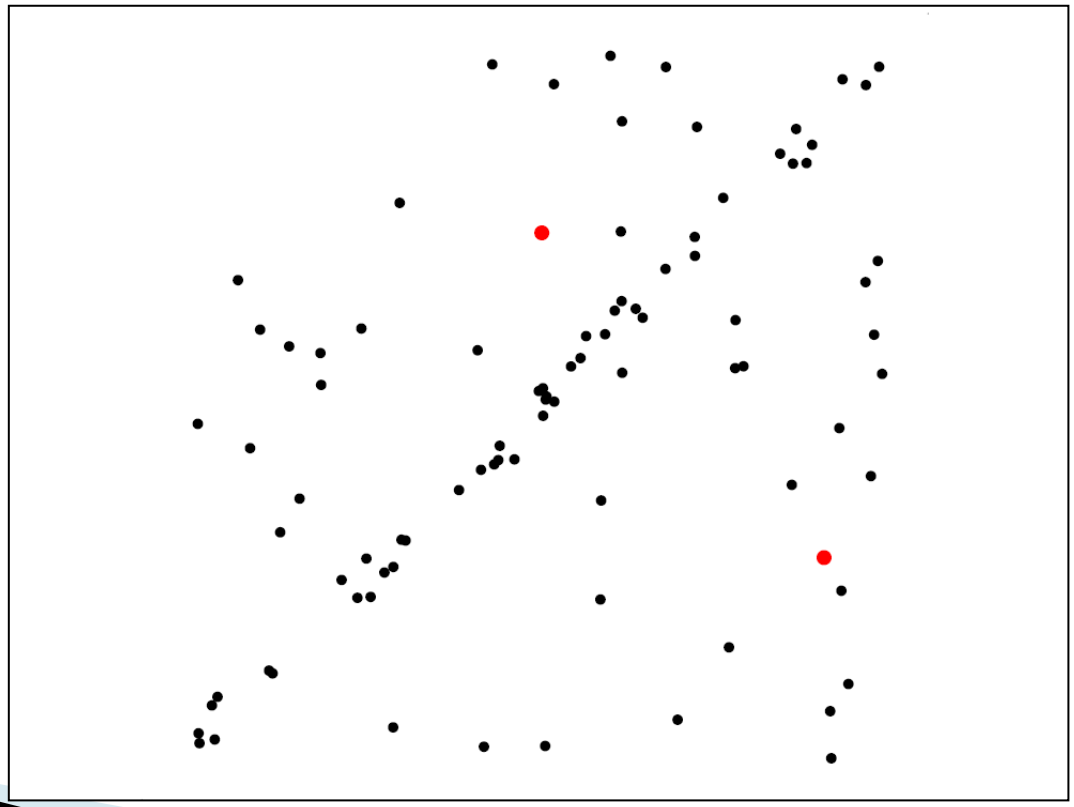
RANSAC for line fitting example



Least-squares fit

Source: R. Raguram

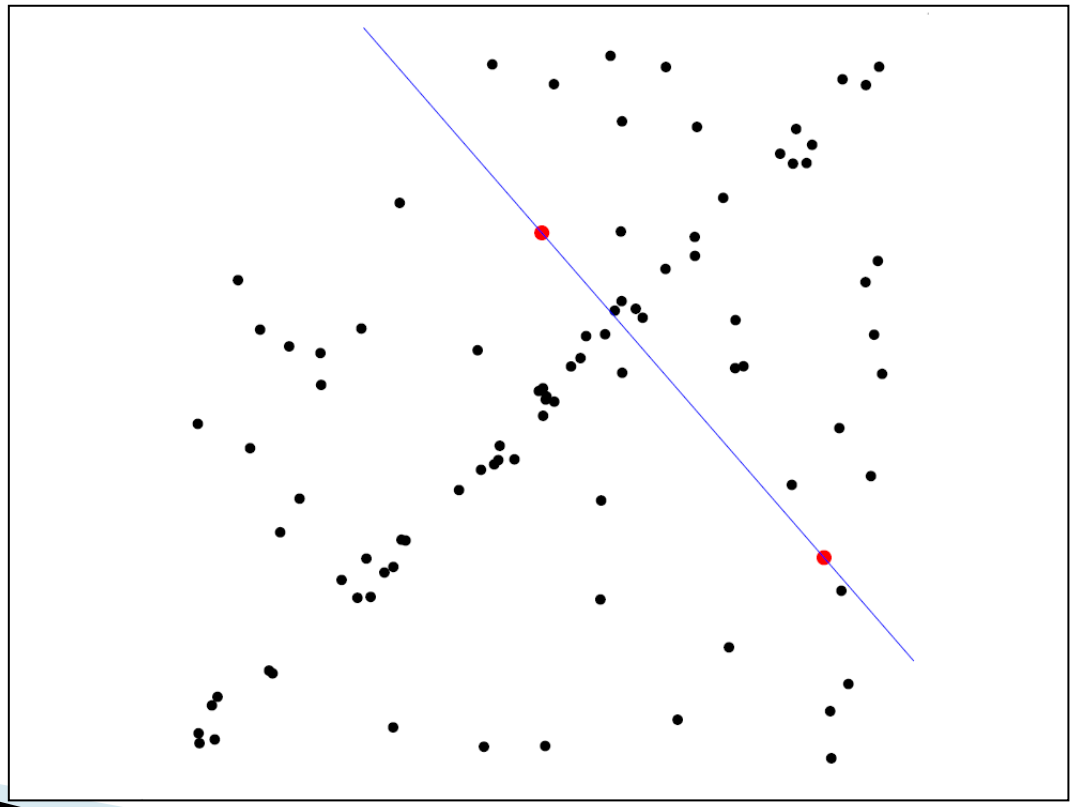
RANSAC for line fitting example



1. Randomly select minimal subset o f points

Source: R. Raguram

RANSAC for line fitting example

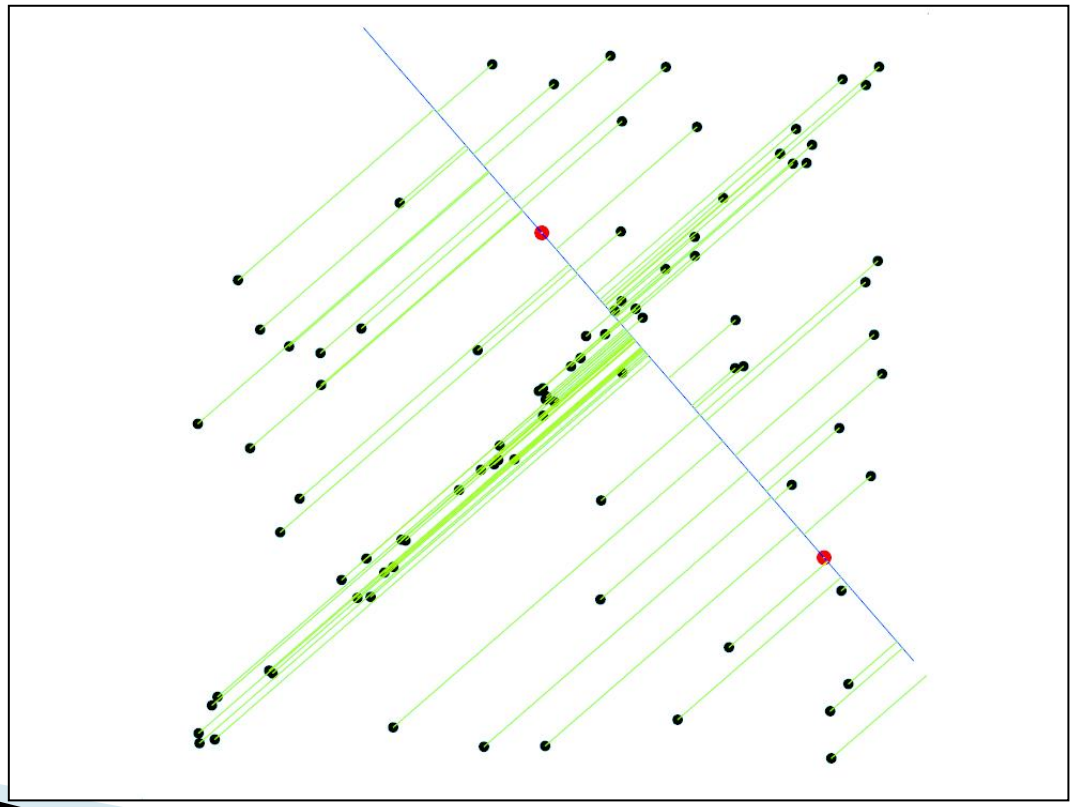


1. Randomly select minimal subset o f points
2. Hypothesize a m odel

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

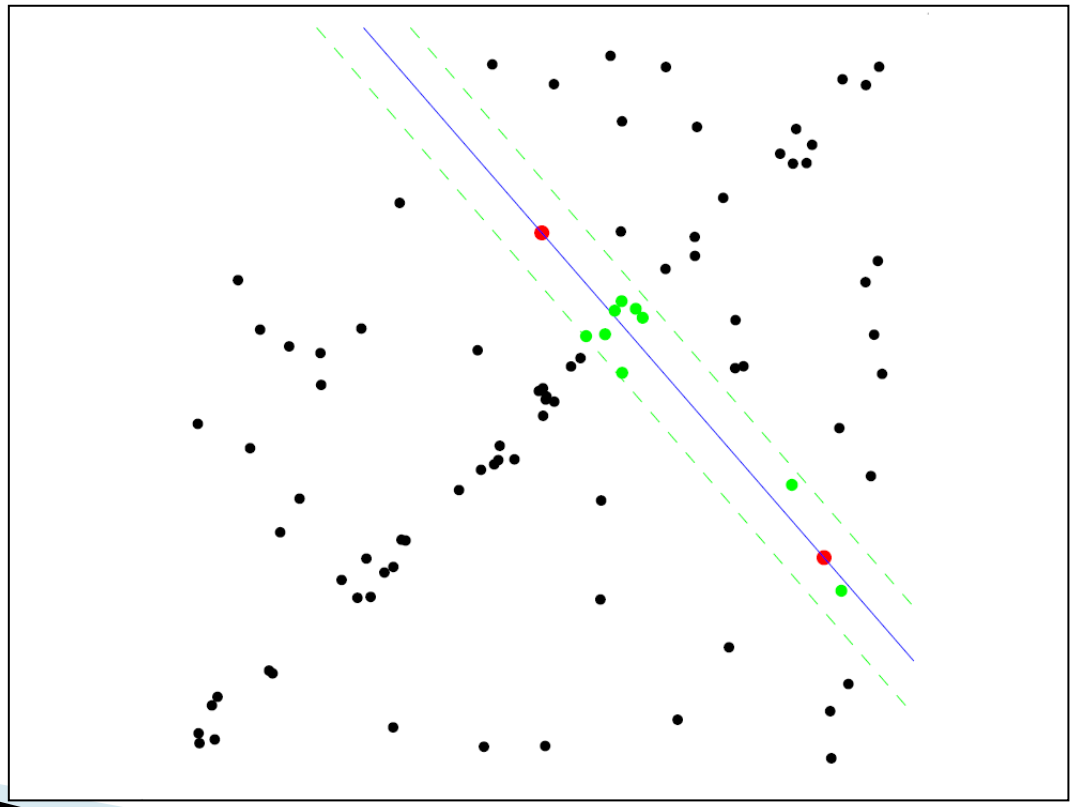


1. Randomly select minimal subset o f points
2. Hypothesize a m odel
3. Compute error f unction

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

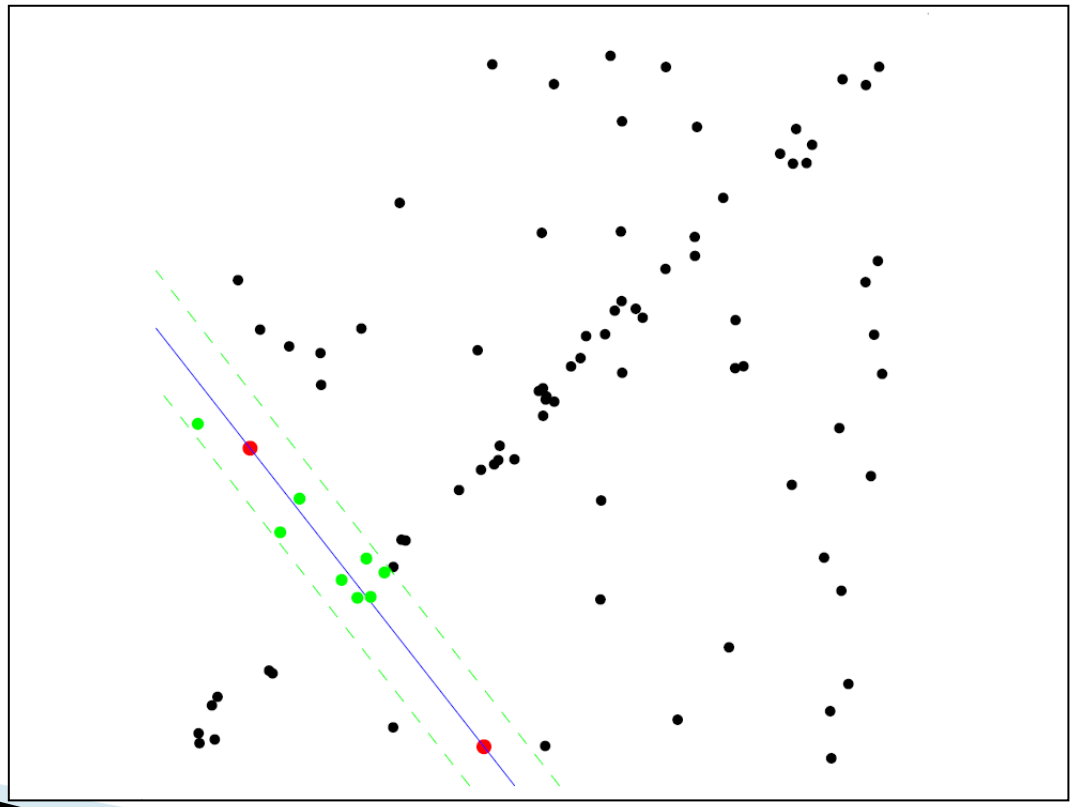


1. Randomly select minimal subset o f points
2. Hypothesize a m odel
3. Compute error f unction
4. Select points con sistent with mod el

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

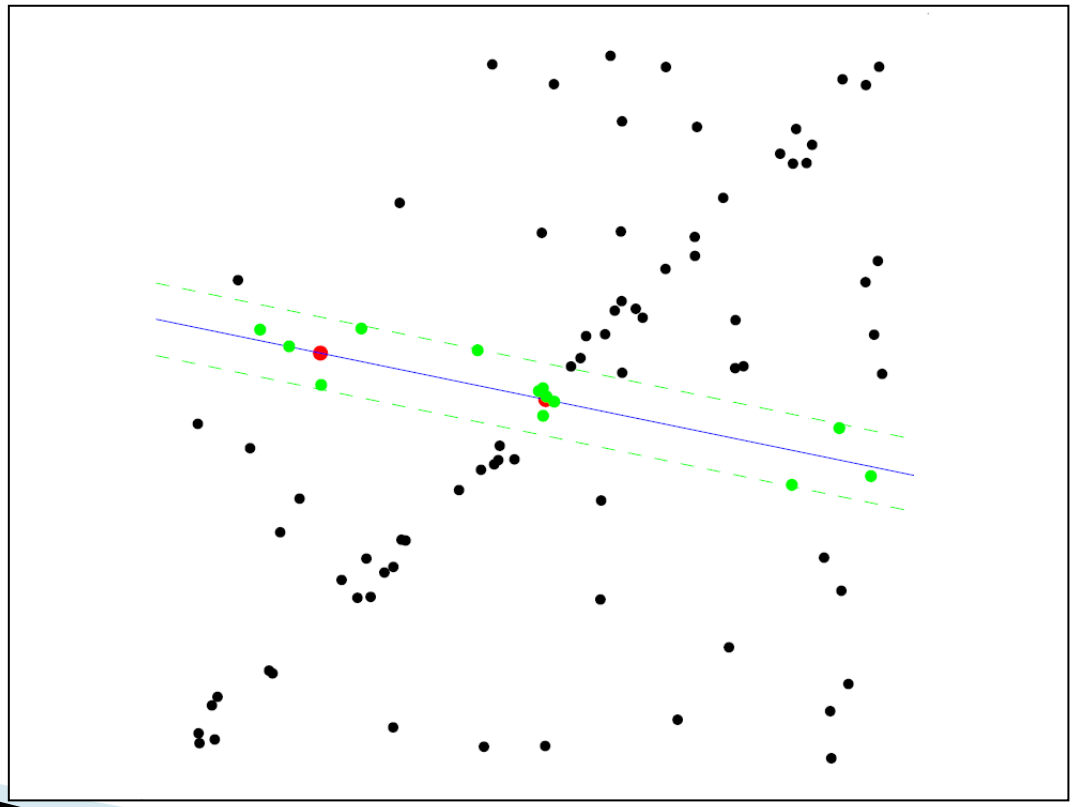


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example

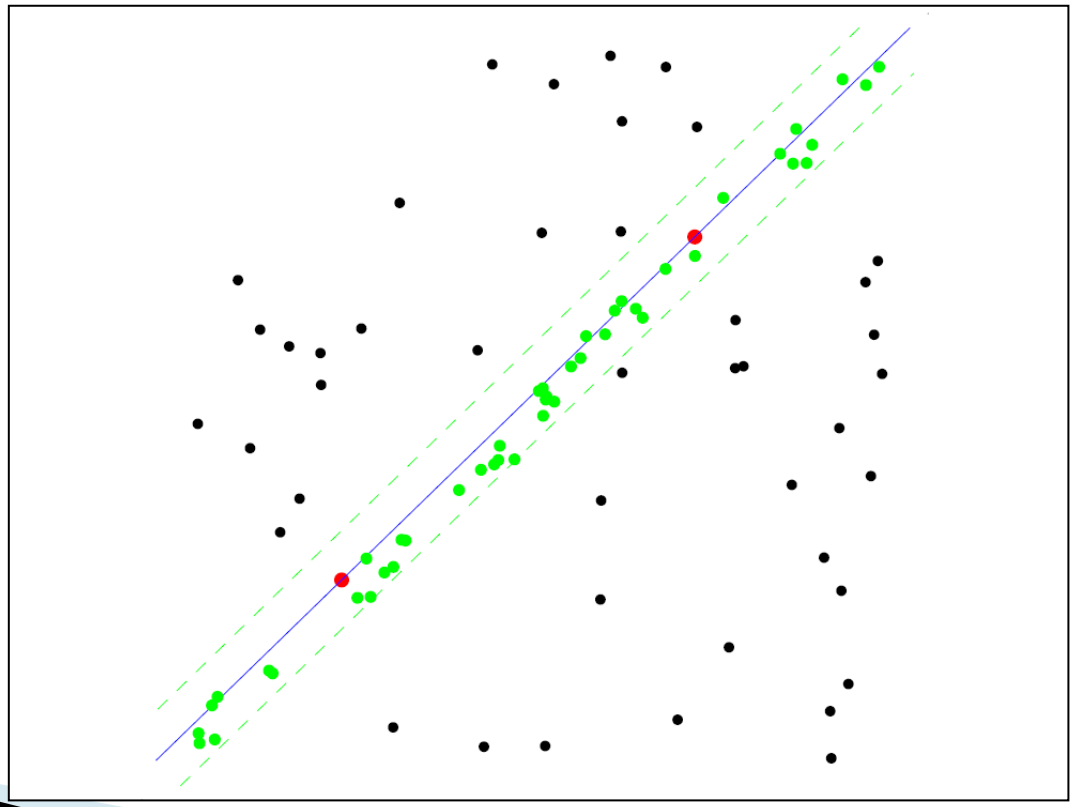


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

RANSAC for line fitting example

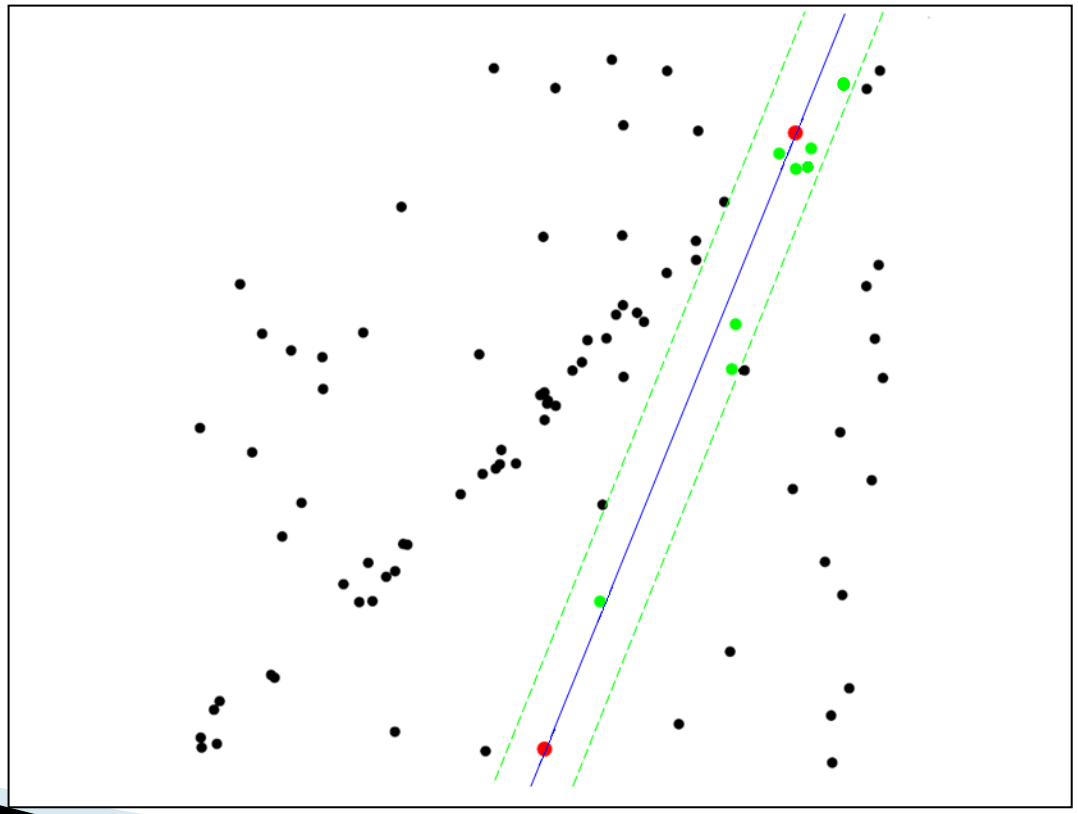
Uncontaminated sample



1. Randomly select minimal subset o f points
2. Hypothesize a m odel
3. Compute error f unction
4. Select points con sistent with mod el
5. Repeat *hypothes ize-and-verify* lo op

Source: R. Raguram

RANSAC for line fitting example



1. Randomly select minimal subset o f points
2. Hypothesize a m odel
3. Compute error f unction
4. Select points con sistent with mod el
5. Repeat *hypothes ize-and-verify* lo op

Source: R. Raguram

Lana Lazebnik

How many trials for RANSAC?

- To ensure good chance of finding true inliers, need sufficient number of trials, S.
- Let p be probability that any given match is valid
- Let P be to the total prob of success after S trials.
- Likelihood in one trial that all k random samples are inliers is p^k
- Likelihood that all S trials will fail is

$$1-P = (1-p^k)^S$$

- Required minimum number of trials is

$$S = \log(1-P) \;/\; \log(1-p^k)$$