Fitting a transformation: Feature-based alignment

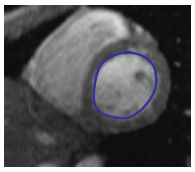
May 1st, 2018

Yong Jae Lee
UC Davis

Last time: Deformable contours

a.k.a. active contours, snakes

Given: initial contour (model) near desired object
Goal: evolve the contour to fit exact object boundary



Main idea: elastic band is iteratively adjusted so as to

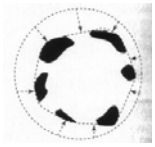
- be near image positions with high gradients, **and**
- satisfy shape "preferences" or contour priors

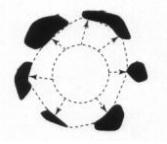
[Snakes: Active contour models, Kass, Witkin, & Terzopoulos, ICCV1987]

2

Figure credit: Yuri Boykov

Last time: Deformable contours





3

Image from http://www.healthline.com/blogs/exercise_fitness/uploaded_images/HandBand2-795868.JPG

Last time: Deformable contours

Pros:
- Useful to track and fit non-rigid shapes
- Contour remains connected
- Possible to fill in “subjective” contours
- Flexibility in how energy function is defined, weighted.

Cons:
- Must have decent initialization near true boundary, may get stuck in local minimum
- Parameters of energy function must be set well based on prior information

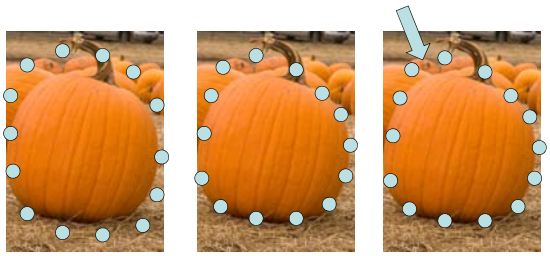
4
Kristen Grauman

Today

- Interactive segmentation
- Feature-based alignment
  - 2D transformations
  - Affine fit
  - RANSAC

5

Interactive forces



How can we implement such an *interactive* force with deformable contours?

6
Kristen Grauman

Interactive forces

- An energy function can be altered online based on user input – use the cursor to push or pull the initial snake away from a point.
- Modify external energy term to include a term such that

Nearby points get pushed hardest

7

Adapted by Devi Parikh from Kristen Grauman

Beyond boundary snapping…

- Another form of interactive guidance: specify regions
- Usually taken to suggest foreground/background color distributions

User Input

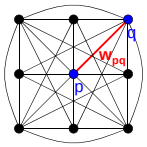
Result

How to use this information?

8

Boykov and Jolly (2001)

Recall: Images as graphs



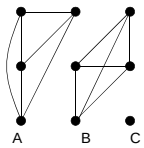
Fully-connected graph

- node for every pixel
- link between *every* pair of pixels, **p**,**q**
- similarity w_{pq} for each link
  - » similarity is *inversely proportional* to difference in color and position

9

Steve Seitz

Recall: Segmentation by Graph Cuts



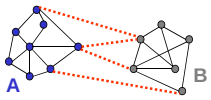
Break graph into segments

- Delete links that cross between segments
- Easiest to break links that have low similarity
  - similar pixels should be in the same segments
  - dissimilar pixels should be in different segments

10

Steve Seitz

Recall: Segmentation by Graph Cuts



Link Cut

- set of links whose removal makes a graph disconnected
- cost of a cut: $cut(A,B) = \sum_{p \in A, q \in B} w_{p,q}$

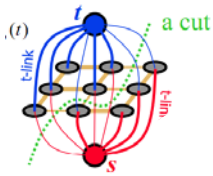
Find minimum cut

- gives you a segmentation
- fast algorithms exist for doing this

11

Source: Steve Seitz

Graph cuts for interactive segmentation



Adding hard constraints:

Add two additional nodes, object and background "terminals"

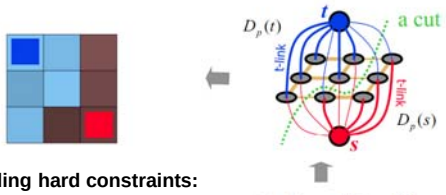
Link each pixel

- To both terminals
- To its neighboring pixels

12

Yuri Boykov

Graph cuts for interactive segmentation



Adding hard constraints:

Let the edge weight to object or background terminal reflect distance to the respective seed pixels.

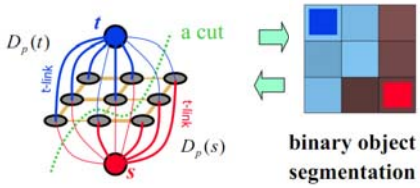
$$D_p(s) = |I_p - I^s|$$
$$D_p(t) = |I_p - I^t|$$

13
Yuri Boykov

Graph cuts for interactive segmentation

$$E(L) = \underbrace{\sum_p D_p(L_p)}_{\text{t-links}} + \underbrace{\sum_{pq \in N} w_{pq} \cdot \delta(L_p \neq L_q)}_{\text{n-links}} \qquad L_p \in \{s,t\}$$

Regional term — Boundary term



binary object segmentation

14

Boykov and Jolly (2001)

Graph cuts for interactive segmentation

Graph Cuts Boykov and Jolly (2001)

GrabCut Rother et al. (2004)

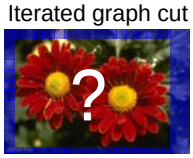
Another interaction modality: specify bounding box

15

Slide credit: Kristen Grauman

"Grab Cut"

- Loosely specify foreground region
- Iterated graph cut



User initialization

K-means for learning colour distributions

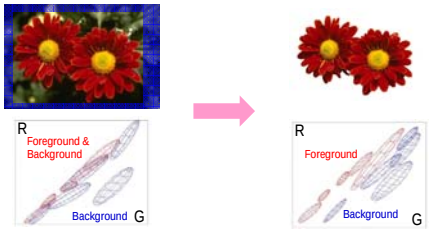
Graph cuts to infer the segmentation

Rother et al (2004)

16

"Grab Cut"

- Loosely specify foreground region
- Iterated graph cut



Gaussian Mixture Model (typically 5-8 components)

Rother et al (2004)

17

"Grab Cut"

Rother et al (2004)

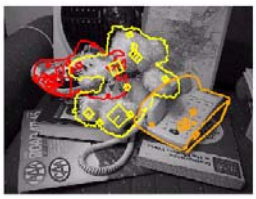
18

Today

- Interactive segmentation
- Feature-based alignment
  - 2D transformations
  - Affine fit
  - RANSAC

19

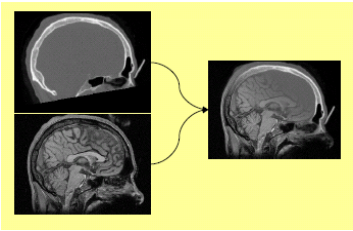
Motivation: Recognition



20

Figures from David Lowe

Motivation: medical image registration

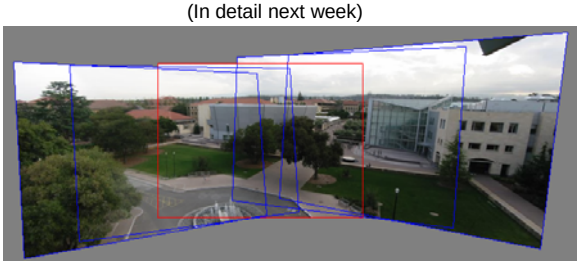


21

Slide credit: Kristen Grauman

Motivation: mosaics

(In detail next week)

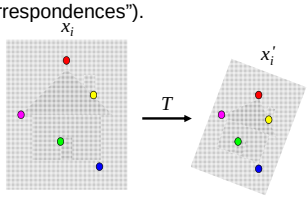


22

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

Alignment problem

- We have previously considered how to **fit a model to image evidence**
  - e.g., a line to edge points, or a snake to a deforming contour
- In alignment, we will **fit the parameters of some transformation** according to a set of matching feature pairs ("correspondences").



23

Slide credit: Kristen Grauman

Parametric (global) warping

Examples of parametric warps:

translation

rotation

aspect

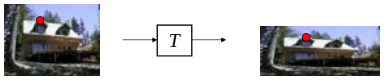
affine

perspective

24

Source: Alyosha Efros

Parametric (global) warping



p = (x,y) **p'** = (x',y')

Transformation T is a coordinate-changing machine:

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' = **M**p

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

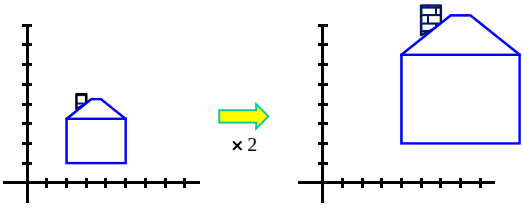
25

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:

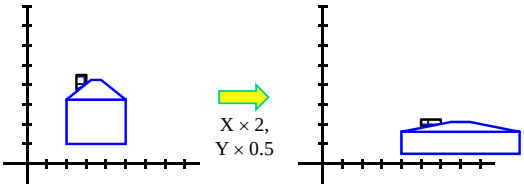


26

Source: Alyosha Efros

Scaling

Non-uniform scaling: different scalars per component:



27

Source: Alyosha Efros

Scaling

Scaling operation:

$$x' = ax$$
$$y' = by$$

Or, in matrix form:

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

28

Source: Alyosha Efros

What transformations can be represented with a 2x2 matrix?

2D Scaling?

$$\boldsymbol{x}' = \boldsymbol{s}_x * \boldsymbol{x}$$
$$\boldsymbol{y}' = \boldsymbol{s}_y * \boldsymbol{y}$$

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

2D Rotate around (0,0)?

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

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

2D Shear?

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{sh}_x * \boldsymbol{y}$$
$$\boldsymbol{y}' = \boldsymbol{sh}_y * \boldsymbol{x} + \boldsymbol{y}$$

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

29

Source: Alyosha Efros

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!

30

Source: Alyosha Efros

2D Linear Transformations

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

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

Linear transformations are combinations of …

- Scale,
- Rotation,
- Shear, and
- Mirror

31

Source: Alyosha Efros

Homogeneous coordinates

Convenient coordinate system to represent many useful transformations

To convert to homogeneous coordinates:

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

homogeneous image coordinates

Converting *from* homogeneous coordinates:

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

32

Slide credit: Kristen Grauman

Homogeneous Coordinates

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

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

A: Using the rightmost column:

$$\textbf{Translation} = \begin{bmatrix} 1 & 0 & \boldsymbol{t}_x \\ 0 & 1 & \boldsymbol{t}_y \\ 0 & 0 & 1 \end{bmatrix}$$

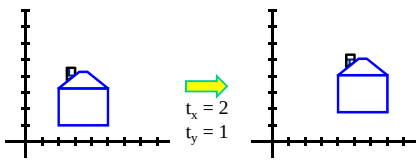
33

Source: Alyosha Efros

Translation

Homogeneous Coordinates

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



34

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

35

Source: Alyosha Efros

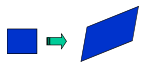
2D Affine Transformations

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

Affine transformations are combinations of …

- Linear transformations, and
- Translations

Parallel lines remain parallel



36

Slide credit: Kristen Grauman

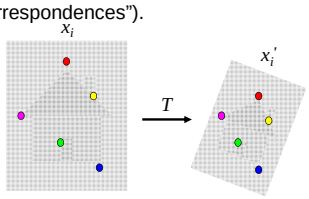
Today

- Interactive segmentation
- Feature-based alignment
  - 2D transformations
  - Affine fit
  - RANSAC

37

Alignment problem

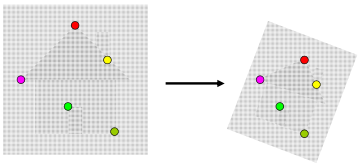
- We have previously considered how to **fit a model to image evidence**
  - e.g., a line to edge points, or a snake to a deforming contour
- In alignment, we will fit the parameters of some **transformation** according to a set of matching feature pairs ("correspondences").



38

Kristen Grauman

Image alignment



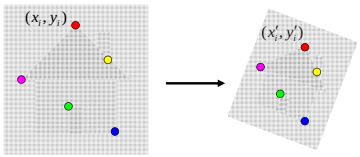
- Two broad approaches:
  - Direct (pixel-based) alignment
    - Search for alignment where most pixels agree
  - Feature-based alignment
    - Search for alignment where *extracted features* agree
    - Can be verified using pixel-based alignment

39

Slide credit: Kristen Grauman

Fitting an affine transformation

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



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

40

Slide credit: Kristen Grauman

An aside: Least Squares Example

Say we have a set of data points (X1,X1'), (X2,X2'), (X3,X3'), etc. (e.g. person's height vs. weight)

We want a nice compact formula (a line) to predict X's from Xs: Xa + b = X'

We want to find a and b

How many (X,X') pairs do we need?

$$X_1 a + b = X_1'$$
$$X_2 a + b = X_2'$$

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \end{bmatrix}$$

Ax=B

What if the data is noisy?

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ X_3 & 1 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \\ X_3' \\ \dots \end{bmatrix}$$

$$\min \|Ax - B\|^2$$

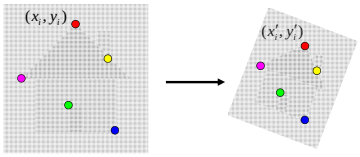
overconstrained

41

Source: Alyosha Efros

Fitting an affine transformation

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



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

$$\begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

42

Slide credit: Kristen Grauman

Fitting an affine transformation

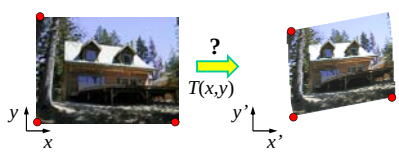
$$\begin{bmatrix} & \cdots & & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & \cdots & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

- How many matches (correspondence pairs) do we need to solve for the transformation parameters?

43

Kristen Grauman

Affine: # correspondences?



How many correspondences needed for affine?

Alyosha Efros

Fitting an affine transformation

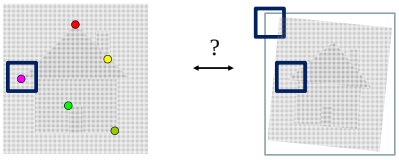
$$\begin{bmatrix} & \cdots & & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & \cdots & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

- How many matches (correspondence pairs) do we need to solve for the transformation parameters?
- Once we have solved for the parameters, how do we compute the coordinates of the corresponding point for (x_{new}, y_{new}) ?
- Where do the matches come from?

45

Kristen Grauman

What **are** the correspondences?



- Compare content in **local** patches, find best matches.
  e.g., simplest approach: scan with template, and compute SSD or correlation between list of pixel intensities in the patch
- Later in the course: how to select regions using more robust descriptors.

46

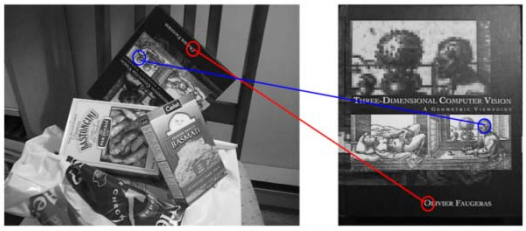
Kristen Grauman

Fitting an affine transformation

47

Figures from David Lowe, ICCV 1999

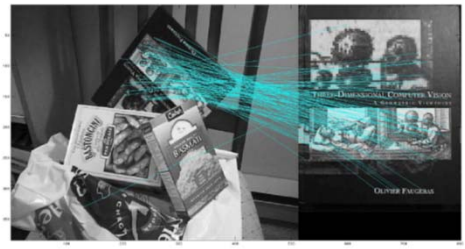
Fitting an affine transformation



Example from UBC SIFT Demo

48

Fitting an affine transformation



Example from UBC SIFT Demo

49

Fitting an affine transformation

50

Figures from David Lowe, ICCV 1999

Today

- Interactive segmentation
- Feature-based alignment
  - 2D transformations
  - Affine fit
  - RANSAC

51

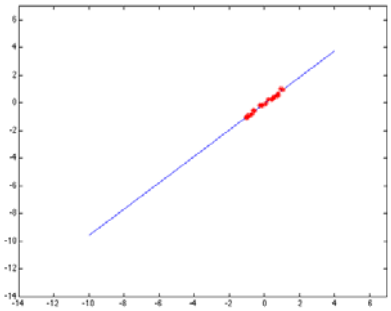
Outliers

- **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.

52

Kristen Grauman

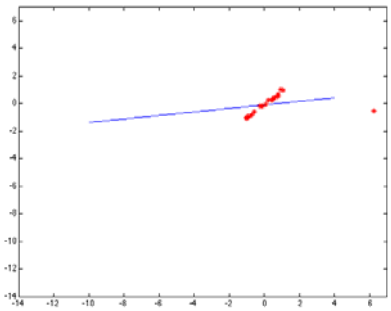
Outliers affect least squares fit



53

Slide credit: Kristen Grauman

Outliers affect least squares fit



54

Slide credit: 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 won't have much support from rest of the points.

55

Slide credit: Kristen Grauman

RANSAC: General form

- RANSAC loop:

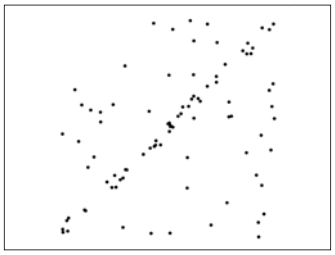
1. Randomly select a *seed group* of points on which to base transformation estimate
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

56

Slide credit: Kristen Grauman

RANSAC for line fitting example

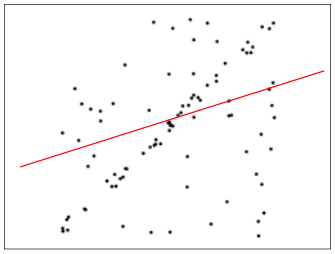


57

Source: R. Raguram

Lana Lazebnik

RANSAC for line fitting example



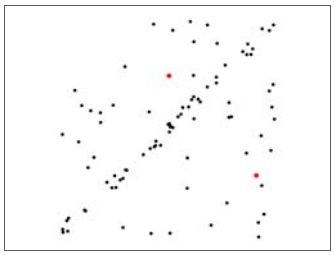
Least-squares fit

Source: R. Raguram

58

Lana Lazebnik

RANSAC for line fitting example



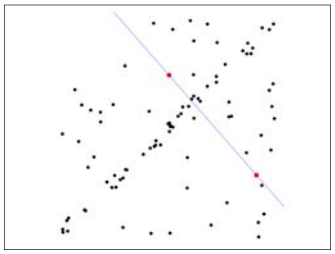
1. Randomly select minimal subset of points

Source: R. Raguram

59

Lana Lazebnik

RANSAC for line fitting example



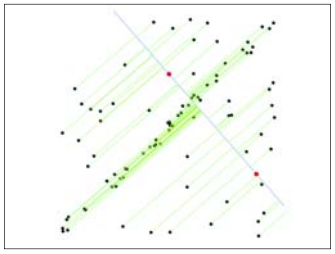
1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram

60

Lana Lazebnik

RANSAC for line fitting example



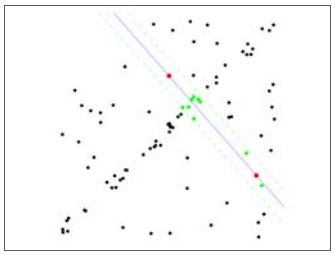
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram

61

Lana Lazebnik

RANSAC for line fitting example



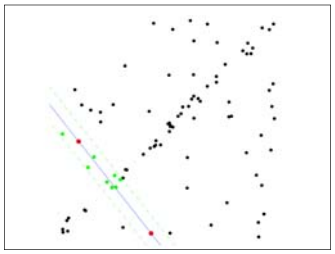
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

Source: R. Raguram

62

Lana Lazebnik

RANSAC for line fitting example



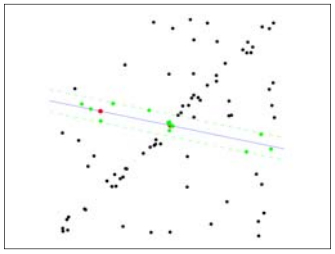
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

63

Lana Lazebnik

RANSAC for line fitting example



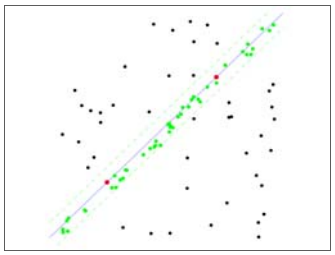
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

64

Lana Lazebnik

RANSAC for line fitting example



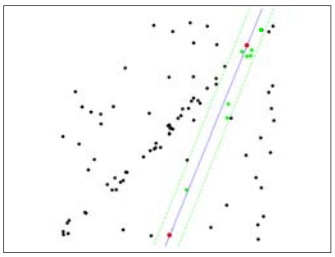
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

65

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

66

Lana Lazebnik

RANSAC for line fitting

Repeat ***N*** times:

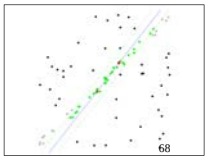
- Draw ***s*** points uniformly at random
- Fit line to these ***s*** points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than ***t***)
- If there are ***d*** or more inliers, accept the line and refit using all inliers

67

Lana Lazebnik

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Lots of parameters to tune
  - 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



68

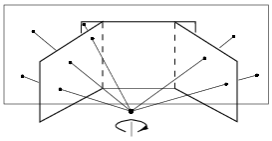
Lana Lazebnik

Today

- Interactive segmentation
- Feature-based alignment
  - 2D transformations
  - Affine fit
  - RANSAC

69

Coming up: alignment and image stitching



70

Questions?

See you Thursday!

71