• Homework 4 posted (due Nov 5)
  • Grouping and texture

• Office hours today tomorrow 2:30 - 3:30 pm
Alignment so far ...

• What are the alignment problems in computer vision?
  • rigid vs. deformable (non-rigid)

• Good features to match
  • Invariance properties
  • Local features: the SIFT descriptor

• Alignment algorithms
  • Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  • Non-rigid alignment
    - Application: shape matching
A framework for alignment

- Matching local features
  - Local information used, can contain outliers
  - But hopefully enough of these matches are good
A framework for alignment

• Matching local features
  • Local information used, can contain outliers
  • But hopefully enough of these matches are good

• Consensus building
  • Aggregate the good matches and find a transformation that explains these matches
• Need to compare *feature descriptors* of local patches surrounding interest points

Source: L. Lazebnik
Feature descriptors: SIFT

- **Descriptor computation:**
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: $4 \times 4 \times 8 = 128$ dimensions

Feature descriptors: SIFT

• Descriptor computation:
  • Divide patch into 4x4 sub-patches
  • Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  • Resulting descriptor: 4x4x8 = 128 dimensions

• Advantage over raw vectors of pixel values
  • Gradients less sensitive to illumination change
  • Pooling of gradients over the sub-patches achieves robustness to small shifts, but still preserves some spatial information

What are the alignment problems in computer vision?
  - rigid vs. deformable (non-rigid)

Good features to match
  - Invariance properties
  - Local features: the SIFT descriptor

Alignment algorithms
  - Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  - Non-rigid alignment
    - Application: shape matching
RANSAC

- **Random Sample Consensus**
  - Choose a small subset of points uniformly at random
  - Fit a model to that subset
  - Find all remaining points that are “close” to the model and reject the rest as outliers
  - Do this many times and choose the best model

RANSAC for line fitting example

Source: R. Raguram
RANSAC for line fitting example

\[
\min_{a,b} \sum_i (ax_i + b - y_i)^2
\]

Least-squares fit

Source: R. Raguram
1. Randomly select minimal subset of points
1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram

RANSAC for line fitting example
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
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
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothese-and-verify* loop

Source: R. Raguram
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
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

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
Fitting an affine transformation

• Assume we know the correspondences, how do we get the transformation?

\[
\begin{bmatrix}
  x'_i \\
  y'_i
\end{bmatrix}
= \begin{bmatrix}
  m_1 & m_2 \\
  m_3 & m_4
\end{bmatrix}
\begin{bmatrix}
  x_i \\
  y_i
\end{bmatrix}
+ \begin{bmatrix}
  t_1 \\
  t_2
\end{bmatrix}
\]

\[
x'_i = Mx_i + t
\]

Want to find \( M, t \) to minimize

\[
\sum_{i=1}^{n} \| x'_i - Mx_i - t \|^2
\]
Fitting an affine transformation

• Assume we know the correspondences, how do we get the transformation?

\[
\begin{bmatrix}
    x'_i \\
y'_i
\end{bmatrix} = \begin{bmatrix}
m_1 & m_2 \\
m_3 & m_4
\end{bmatrix} \begin{bmatrix}
x_i \\
y_i
\end{bmatrix} + \begin{bmatrix}
t_1 \\
t_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
x_i & y_i & 0 & 0 & 1 & 0 \\
0 & 0 & x_i & y_i & 0 & 1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix} \begin{bmatrix}
m_1 \\
m_2 \\
m_3 \\
m_4 \\
t_1 \\
t_2
\end{bmatrix} = \begin{bmatrix}
x'_i \\
y'_i
\end{bmatrix}
\]
Fitting an affine transformation

\[
\begin{bmatrix}
  x_i & y_i & 0 & 0 & 1 & 0 \\
  0 & 0 & x_i & y_i & 0 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
  m_1 \\
  m_2 \\
  m_3 \\
  m_4 \\
  t_1 \\
  t_2 \\
\end{bmatrix} = \begin{bmatrix}
  x'_i \\
  y'_i \\
  \vdots \\
\end{bmatrix}
\]

- Linear system with six unknowns
- Each match gives us two linearly independent equations: need at least three to solve for the transformation parameters
Application: Panorama stitching

Source: Hartley & Zisserman
Panoramic stitching

• **Approach**
  
  • Local feature matching
  
  • RANSAC for alignment
Panoramic stitching
Panoramic stitching

- Extract features
  - corner detector
Panoramic stitching

- Extract features
- Compute *putative matches*
Panoramic stitching

- Extract features
- Compute *putative matches*
- Loop:
  - Hypothesize transformation $T$
• Extract features
• Compute *putative matches*
• Loop:
  • *Hypothesize* transformation $T$
  • *Verify* transformation (search for other matches consistent with $T$)
Panoramic stitching

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation $T$
  - *Verify* transformation (search for other matches consistent with $T$)
City-scale alignment

Microsoft Photosynth

https://www.youtube.com/watch?v=y9zF97JL30A
Alignment

• What are the alignment problems in computer vision?
  • rigid vs. deformable (non-rigid)

• Good features to match
  • Invariance properties
  • Local features: the SIFT descriptor

• Alignment algorithms
  • Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  • Non-rigid alignment
    - Application: shape matching
Non-rigid transformations

Fig. 517. *Argyropelecus Olfersi.* Fig. 518. *Sternopyx diaphana.*

*On Growth and Form, D’Arcy Thompson 1915*
• How to characterize a non-rigid transformation?

• Global constraints
  • Should be from a family of constraints, e.g. Affine

• Local constraints
  • Low distortion, e.g., a good match should avoid
    - Many to one matches, one to many matches, criss crossings
Low distortion matching

- Measuring distortion between vectors
  - \( \mathbf{R} \) and \( \mathbf{S} \) same \textit{length} for \textit{rotations}
  - \( \mathbf{R} \) and \( \mathbf{S} \) same \textit{angle} for \textit{scaling}
- Distortion cost: \( \alpha f_r (|R_{ij}|, |S_{ij'}|) + (1 - \alpha) f_s (\Theta(R_{ij}), \Theta(S_{ij'})) \)
- Matching score: \( c(i \rightarrow i') + c(j \rightarrow j') \)

Berg et al., CVPR 2005
Scoring matches

possible matches for the green point

— bad matches  — good matches
Scoring matches

- fix the red match

- low distortion

- high distortion
Scoring matches

fix the red match
• Find correspondence that:
  • minimizes distortion (pairwise)
  • maximizes scores (individual) based on local descriptors matching

\[
\text{minimize} \left( - \sum_a \text{score}(a) + \sum_{a,b} \text{distortion}(a, b) \right)
\]

Berg et al., CVPR 2005
• Find correspondence that:
  • minimizes distortion (pairwise)
  • maximizes scores (individual) based on local descriptors matching

• Objective is an integer quadratic program (IQP)
  • NP-Hard
  • But approximate solutions can be found in reasonable time
Example matches

Probe

Query

Berg et al., CVPR 2005 42
Example matches

Probe

Query

Berg et al., CVPR 2005
Example matches

Probe

Query

Berg et al., CVPR 2005
Example matches

Interpolating the alignment
Automatic segmentation

a.

b.

c.

d.

Retrieval Rate vs Shortlist Length

- Hand Segmentation
- Automatic Segmentation

Berg et al., CVPR 2005
Alignment

• What are the alignment problems in computer vision?
  • rigid vs. deformable (non-rigid)

• Good features to match
  • Invariance properties
  • Local features: the SIFT descriptor

• Alignment algorithms
  • Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  • Non-rigid alignment
    - Application: shape matching
What if we need to align a test image with thousands or millions of images in a model database?

- Efficient putative match generation
  - Approximate descriptor similarity search, inverted indices
Large-scale visual search

Figure from: Kristen Grauman and Bastian Leibe, *Visual Object Recognition*, Synthesis Lectures on Artificial Intelligence and Machine Learning, April 2011, Vol. 5, No. 2, Pages 1-181
Example indexing technique: Vocabulary trees

D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, CVPR 2006
Goal: find a set of representative prototypes or cluster centers to which descriptors can be quantized
Hierarchical k-means clustering of descriptor space (vocabulary tree)

Slide credit: D. Nister
Vocabulary tree/inverted index
Populating the vocabulary tree/inverted index

Model images
Populating the vocabulary tree/inverted index

Model images

Slide credit: D. Nister
Populating the vocabulary tree/inverted index

Model images
Populating the vocabulary tree/inverted index

Model images

Slide credit: D. Nister
Looking up a test image
Further thoughts and readings …

• Chapters 6, 7 and 9 from Richard Szeliski’s book.

• Shape matching references
  • Shape matching and object recognition using shape contexts, Belongie, Malik and Puzicha, PAMI 2002 (paper)
  • Shape matching and object recognition using low distortion correspondences, Berg, Berg and Malik, CVPR 2005 (paper)
  • Hierarchical matching of deformable shapes, Felzenszwalb and Schwartz, CVPR 2007 (paper)
  • David Nister’s vocabulary tree paper

• Web demos from Oxford VGG group
  • Video google, Oxford building search, Sculpture retrieval
Cool application of large-scale alignment: searching the night sky

http://www.astrometry.net/