Computational Photography

Prof. Feng Liu

Spring 2016

http://www.cs.pdx.edu/~fliu/courses/cs510/

04/26/2016
Last Time

☐ Panorama
  ■ Feature detection
Today

- Panorama
  - Feature matching
  - Homography estimation

With slides by C. Dyer, Y. Chuang, R. Szeliski, S. Seitz, M. Brown and V. Hlavac
Stitching Recipe

- Align pairs of images
  - Feature Detection
  - Feature Matching
  - Homography Estimation

- Align all to a common frame
- Adjust (Global) & Blend
Invariant Local Features

- Goal: Detect the same interest points regardless of image changes due to translation, rotation, scale, viewpoint
Feature Point Descriptors

- After detecting points (and patches) in each image,
- Next question: **How to match them?**

Point descriptor should be:
1. Invariant
2. Distinctive

All the following slides are used from Prof. C. Dyer’s relevant course, except those with explicit acknowledgement.
Local Features: Description

1. Detection: Identify the interest points

2. Description: Extract feature vector for each interest point

3. Matching: Determine correspondence between descriptors in two views

\[
\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}] \\
\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]
\]
Geometric Transformations

- e.g. scale,
- translation,
- rotation
Photometric Transformations

Modelled as a linear transformation: scaling + offset

Figure from T. Tuytelaars ECCV 2006 tutorial
Raw Patches as Local Descriptors

The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts or rotations.
Making Descriptors Invariant to Rotation

- Find local orientation

Dominant direction of gradient:

- Compute description \textit{relative} to this orientation

2 D. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. Accepted to IJCV 2004
SIFT Descriptor: Select Major Orientation

- Compute histogram of local gradient directions computed at selected scale in neighborhood of a feature point relative to dominant local local orientation.

- Compute gradients within sub-patches, and compute histogram of orientations using discrete “bins”.

- Descriptor is rotation and scale invariant, and also has some illumination invariance (why?)
SIFT Descriptor

- Compute gradient orientation histograms on 4 x 4 neighborhoods over 16 x 16 array of locations in scale space around each keypoint position, relative to the keypoint orientation using thresholded image gradients from Gaussian pyramid level at keypoint’s scale.
- Quantize orientations to 8 values
- 4 x 4 array of histograms
- SIFT feature vector of length 4 x 4 x 8 = 128 values for each keypoint
- Normalize the descriptor to make it invariant to intensity change

Stable (repeatable) feature points can currently be detected that are invariant to
- Rotation, scale, and affine transformations, but not to more general perspective and projective transformations

Feature point descriptors can be computed, but
- are noisy due to use of differential operators
- are not invariant to projective transformations
Feature Matching
Wide-Baseline Feature Matching

- Standard approach for pair-wise matching:
  - For each feature point in image A
  - Find the feature point with the closest descriptor in image B

From Schaffalitzky and Zisserman ’02
Wide-Baseline Feature Matching

- Compare the distance, $d_1$, to the closest feature, to the distance, $d_2$, to the second closest feature

- Accept if $d_1/d_2 < 0.6$
  - If the ratio of distances is less than a threshold, keep the feature

- Why the ratio test?
  - Eliminates hard-to-match repeated features
  - Distances in SIFT descriptor space seem to be non-uniform
Feature Matching

- Exhaustive search
  - for each feature in one image, look at all the other features in the other image(s)

- Hashing
  - compute a short descriptor from each feature vector, or hash longer descriptors (randomly)

- Nearest neighbor techniques
  - k-trees and their variants
Wide-Baseline Feature Matching

- Because of the high dimensionality of features, *approximate nearest neighbors* are necessary for efficient performance.

- See ANN package, Mount and Arya
  
Stitching Recipe

- Align pairs of images
  - Feature Detection
  - Feature Matching
  - Homography Estimation

- Align all to a common frame
- Adjust (Global) & Blend
What can be globally aligned?

- In image stitching, we seek for a model to globally warp one image into another. Are any two images of the same scene can be aligned this way?
  - Images captured with the same center of projection
  - A planar scene or far-away scene

Credit: Y.Y. Chuang
A pencil of rays contains all views

Can generate any synthetic camera view as long as it has the same center of projection!

Credit: Y.Y. Chuang
Mosaic as an image reprojection

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

Credit: Y.Y. Chuang
Changing camera center

Does it still work?

Credit: Y.Y. Chuang
Planar scene (or a faraway one)

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

Credit: Y.Y. Chuang
Motion models

- Parametric models as the assumptions on the relation between two images.

Credit: Y.Y. Chuang
2D Motion models

<table>
<thead>
<tr>
<th>Name</th>
<th>Matrix</th>
<th># D.O.F.</th>
<th>Preserves:</th>
<th>Icon</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>$[I \</td>
<td>\ t]_{2\times3}$</td>
<td>2</td>
<td>orientation + ⋯</td>
</tr>
<tr>
<td>rigid (Euclidean)</td>
<td>$[R \</td>
<td>\ t]_{2\times3}$</td>
<td>3</td>
<td>lengths + ⋯</td>
</tr>
<tr>
<td>similarity</td>
<td>$[sR \</td>
<td>\ t]_{2\times3}$</td>
<td>4</td>
<td>angles + ⋯</td>
</tr>
<tr>
<td>affine</td>
<td>$[A]_{2\times3}$</td>
<td>6</td>
<td>parallelism + ⋯</td>
<td></td>
</tr>
<tr>
<td>projective</td>
<td>$[\tilde{H}]_{3\times3}$</td>
<td>8</td>
<td>straight lines</td>
<td></td>
</tr>
</tbody>
</table>

Credit: Y.Y. Chuang
Motion models

- Translation: 2 unknowns
- Affine: 6 unknowns
- Perspective: 8 unknowns
- 3D rotation: 3 unknowns

Credit: Y.Y. Chuang
Determine pairwise alignment?

- Feature-based methods: only use feature points to estimate parameters

- We will study the “Recognising panorama” paper published in ICCV 2003

- Run SIFT (or other feature algorithms) for each image, find feature matches.

Credit: Y.Y. Chuang
Determine pairwise alignment

- $p' = Mp$, where $M$ is a transformation matrix, $p$ and $p'$ are feature matches
- It is possible to use more complicated models such as affine or perspective
- For example, assume $M$ is a 2x2 matrix
  
  \[
  \begin{pmatrix}
  x' \\
  y'
  \end{pmatrix} = \begin{pmatrix}
  m_{11} & m_{12} \\
  m_{21} & m_{22}
  \end{pmatrix}
  \begin{pmatrix}
  x \\
  y
  \end{pmatrix}
  \]

- Find $M$ with the least square error
  
  \[
  \sum_{i=1}^{n} (Mp - p')^2
  \]

Credit: Y.Y. Chuang
Determine pairwise alignment

\[
\begin{pmatrix}
    x' \\
    y'
\end{pmatrix} =
\begin{pmatrix}
    m_{11} & m_{12} \\
    m_{21} & m_{22}
\end{pmatrix}
\begin{pmatrix}
    x \\
    y
\end{pmatrix}
\]
\[
x_1 m_{11} + y_1 m_{12} = x'_1
\]
\[
x_1 m_{21} + y_1 m_{22} = y'_1
\]

Credit: Y.Y. Chuang
Normal equation

Given an over-determined system

\[ Ax = b \]

the normal equation is that which minimizes the sum of the square differences between left and right sides

\[ A^T Ax = A^T b \]

Why?

Credit: Y.Y. Chuang
Normal equation

\[ E = (Ax - b)^2 \]

\[ = (Ax - b)^T (Ax - b) \]

\[ = ((Ax)^T - b^T)(Ax - b) \]

\[ = (x^T A^T - b^T)(Ax - b) \]

\[ = x^T A^T Ax - b^T Ax - x^T A^T b + b^T b \]

\[ = x^T A^T Ax - (A^T b)^T x - (A^T b)^T x + b^T b \]

\[ \frac{\partial E}{\partial x} = 2A^T Ax - 2A^T b \]

Credit: Y.Y. Chuang
Determine pairwise alignment

- \( p' = Mp \), where \( M \) is a transformation matrix, \( p \) and \( p' \) are feature matches.
- For translation model, it is easier.

\[
E = \sum_{i=1}^{n} \left[ (m_1 + x_i - x'_i)^2 + (m_2 + y_i - y'_i)^2 \right]
\]

\[
0 = \frac{\partial E}{\partial m_1}
\]

- What if the match is false? Avoid impact of outliers.

Credit: Y.Y. Chuang
RANSAC [Fischler and Bolles 81]

- RANSAC = Random Sample Consensus
- An algorithm for robust fitting of models in the presence of many data outliers
  - Compare to robust statistics
- Given $N$ data points $x_i$, assume that majority of them are generated from a model with parameters $\Theta$, try to recover $\Theta$.

RANSAC algorithm

Run \( k \) times:

1. Draw \( n \) samples randomly
2. Fit parameters \( \Theta \) with these \( n \) samples
3. For each of other \( N-n \) points, calculate its distance to the fitted model, count the number of inlier points, \( c \)

Output \( \Theta \) with the largest \( c \)

How many times? How big? Smaller is better

How to define? Depends on the problem.

Credit: Y.Y. Chuang
How to determine $k$

$p$: probability of real inliers

$P$: probability of success after $k$ trials

\[ P = 1 - \left(1 - p^n\right)^k \]

\[ k = \frac{\log(1 - P)}{\log(1 - p^n)} \]

for $P = 0.99$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$p$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.5</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>97</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
<td>293</td>
</tr>
</tbody>
</table>

Credit: Y.Y. Chuang
Example: line fitting

Credit: Y.Y. Chuang
Example: line fitting

$n=2$

Credit: Y.Y. Chuang
Model fitting

Credit: Y.Y. Chuang
Measure distances

Credit: Y.Y. Chuang
Count inliers

\( c = 3 \)

Credit: Y.Y. Chuang
Another trial

$\textcolor{red}{c=3}$

Credit: Y.Y. Chuang
The best model

c=15

Credit: Y.Y. Chuang
RANSAC for Homography

Credit: Y.Y. Chuang
RANSAC for Homography

Credit: Y.Y. Chuang
RANSAC for Homography

Credit: Y.Y. Chuang
Next Time

☐ Panorama
  ■ Blending
  ■ Multi-perspective panoramas