Local features and image matching
May 5th, 2020

Yong Jae Lee
UC Davis

Last time

• RANSAC for robust fitting
  – Lines, translation
• Image mosaics
  – Fitting a 2D transformation
    • Homography

Today

Mosaics recap:
How to warp one image to the other, given H?

How to detect which features to match?
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)

Source: Steve Seitz

Mosaics

Obtain a wider angle view by combining multiple images.

Homography

\[
\begin{align*}
(x_1, y_1) & \quad \rightarrow \quad (x'_1, y'_1) \\
(x_2, y_2) & \quad \rightarrow \quad (x'_2, y'_2) \\
\vdots & \quad \quad \vdots \\
(x_n, y_n) & \quad \rightarrow \quad (x'_n, y'_n)
\end{align*}
\]

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…
Solving for homographies

\[ \begin{pmatrix} x' \\ y' \\ w \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \]

Defined up to a scale factor.
Constrain Frobenius norm of H to be 1.

Problem to be solved:

\[ \min \| Ab - b \| \]

s.t. \[ \| h \| = 1 \]

where vector of unknowns \( h = [h_{00}, h_{01}, h_{02}, h_{10}, h_{11}, h_{12}, h_{20}, h_{21}, h_{22}] \)

Adapted from Devi Parikh

Solving for homographies

\[ \begin{pmatrix} x' \\ y' \\ w \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \]

\[ w' = h_{00}x + h_{01}y + h_{02} \]
\[ y' = h_{10}x + h_{11}y + h_{12} \]
\[ w = h_{20}x + h_{21}y + h_{22} \]

There are 9 variables \( h_{00}, \ldots, h_{22} \).
Are there 9 degrees of freedom?
No. We can multiply all \( h_{ij} \) by nonzero scalar \( k \) without changing the equations:

\[ x' = \frac{h_{00}kx + h_{01}ky + h_{02}k}{h_{20}kx + h_{21}ky + h_{22}k} \]
\[ y' = \frac{h_{10}kx + h_{11}ky + h_{12}k}{h_{20}kx + h_{21}ky + h_{22}k} \]
\[ w' = \frac{h_{20}kx + h_{21}ky + h_{22}k}{h_{20}kx + h_{21}ky + h_{22}k} \]

Enforcing 8 DOF

Impose unit vector constraint

\[ x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \]
\[ y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}} \]

Subject to: \[ h_{00} + h_{01} + h_{02} + h_{10} + h_{11} + h_{12} + h_{20} + h_{21} + h_{22} = 1 \]
Projective: # correspondences?

How many correspondences needed for projective?

Source: Alyosha Efros

RANSAC for estimating homography

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

Source: Alyosha Efros

Robust feature-based alignment

Source: L. Lazebnik
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

Creating and Exploring a Large Photorealistic Virtual Space

http://www.youtube.com/watch?v=E0rboU1PbPo
Creating and Exploring a Large Photorealistic Virtual Space

Current view, and desired view in green

Synthesized view from new camera

Induced camera motion

Today

Mosaics recap: How to warp one image to the other, given H?

How to detect which features to match?

Detecting local invariant features

• Detection of interest points
  – Harris corner detection
  – (Scale invariant blob detection: LoG)
• (Next time: description of local patches)
Local features: main components

1) Detection: Identify the interest points

2) Description: Extract vector feature descriptor surrounding each interest point.

3) Matching: Determine correspondence between descriptors in two views

Local features: desired properties

- **Repeatability**
  - The same feature can be found in several images despite geometric and photometric transformations

- **Saliency**
  - Each feature has a distinctive description

- **Compactness and efficiency**
  - Many fewer features than image pixels

- **Locality**
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Applications

- **Local features** have been used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition
A hard feature matching problem

Goal: interest operator repeatability

- We want to detect (at least some of) the same points in both images.
- Yet we have to be able to run the detection procedure independently per image.
Goal: descriptor distinctiveness

• We want to be able to reliably determine which point goes with which.

• Must provide some invariance to geometric and photometric differences between the two views.

Local features: main components

1) Detection: Identify the interest points

2) Description: Extract vector feature descriptor surrounding each interest point.

3) Matching: Determine correspondence between descriptors in two views

• What points would you choose (for repeatability, distinctiveness)?
Corners as distinctive interest points

We should easily recognize the point by looking through a small window. Shifting a window in any direction should give a large change in intensity.

“flat” region: no change in all directions
“edge”: no change along the edge direction
“corner”: significant change in all directions

Slide credit: Alyosha Efros, Darya Frolova, Denis Sviridenko

\[
M = \sum \begin{bmatrix}
I_x I_x & I_x I_y \\
I_y I_x & I_y I_y
\end{bmatrix}
\]

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).

\[
I_x \Rightarrow \frac{\partial I}{\partial x}, \quad I_y \Rightarrow \frac{\partial I}{\partial y}, \quad I_x I_y \Rightarrow \frac{\partial I}{\partial x} \frac{\partial I}{\partial y}
\]

What does this matrix reveal?

First, consider an axis-aligned corner:
First, consider an axis-aligned corner:

\[
M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}
\]

This means dominant gradient directions align with x or y axis.

Look for locations where both \( \lambda \)'s are large.

If either \( \lambda \) is close to 0, then this is not corner-like.

What if we have a corner that is not aligned with the image axes?

Since \( M \) is symmetric, we have
\[
M = \lambda \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T
\]

(Eigenvalue decomposition)

\[
M x_i = \lambda_i x_i
\]

The eigenvalues of \( M \) reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.

Corner response function

- "edge": \( \lambda_1 \gg \lambda_2 \)
- "corner": \( \lambda_1 \) and \( \lambda_2 \) are large, \( \lambda_1 \approx \lambda_2 \)
- "flat" region: \( \lambda_1 \) and \( \lambda_2 \) are small

\[
f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}
\]
Harris corner detector

1) Compute $M$ matrix for each image window to get their cornerness scores.
2) Find points whose surrounding window gave large corner response ($f > \text{threshold}$)
3) Take the points of local maxima, i.e., perform non-maximum suppression

Example of Harris application

Compute corner response at every pixel.
Example of Harris application

Properties of the Harris corner detector
Rotation invariant? Yes

Properties of the Harris corner detector
Rotation invariant? Yes
Translation invariant? Yes
Properties of the Harris corner detector

Rotation invariant?  Yes
Translation invariant?  Yes
Scale invariant?  No

All points will be classified as edges

Summary

• Image warping to create mosaic, given homography

• Interest point detection
  – Harris corner detector
  – Next time:
    • Laplacian of Gaussian, automatic scale selection