CSCI 497P/597P: Computer Vision

Lecture 11: Image Features
Feature Matching
Forward and Inverse Warping
Announcements

- Feedback survey

- HW 1 is due Wednesday
Goals

• Know how and why to **match** features using:
  
  • The simple **SSD metric**
  
  • The **ratio test**

• Understand the mathematical framework for (linear) geometric transformations on images (**image warping**).

• Understand the differences between **forward** and **inverse warping**.
Running motivational example: Panorama Stitching
1. Detect Harris Corners
1. Detect Harris Corners
2. Describe features
3. Match Features

At this point, a "feature" has:

1. A keypoint: its position in one of the images
2. A descriptor: a vector of numbers that 'captures' its characteristics
Feature Matching

Image 1 features: \( F_1 = \{ f_1, f_2, \ldots \} \)

Image 2 features: \( F_2 = \{ f_1, f_2, f_3, \ldots \} \)
F1 = detect_describe(img1)

F2 = detect_describe(img2)

for f1 in F1:

    find f2 that minimizes d(f1, f2)

    add (f1, f2) to matches
But what if we're wrong?

- Answer #1: Threshold on match score
Matching Algorithm

F1 = detect_describe(img1)

F2 = detect_describe(img2)

for f1 in F1:

    find f2 that minimizes d(f1, f2)

    if d(f1, f2) < T < choose empirically

        add (f1, f2) to matches
Distance Metrics

What should we use for \( d \) in \( d(f_1, f_2) \)?

\[ \text{SSD} \]

\[ \|f_1 - f_2\|^2 = \sum (f_1 - f_2)^2 \]
Distance Metrics

Sidenote: efficiently computing SSD

\[ d = \sqrt{\text{np.sum}((f1 - f2)^2)} \]

\[ \text{dist} = d \cdot \text{dist}(d) \]
A problem with SSD
A problem with SSD
A problem with SSD
We want a metric that gives small distance when features are
• like each other (according to SSD), but
• not like any others
"Ratio Test" distance metric

\[ \gamma = \frac{\| f_1 - f_2 \|^2}{\| f_1 - f_3 \|^2} \]

Small \( \gamma \):
- \( f_2 \) = closest SSD
- \( f'_2 \) = second closest

Large \( \gamma \):
- \( f_1 \)'s closest match is still \( f_2 \), but the distance between them is different
Matching Algorithm

\[ F_1 = \text{detect\_describe}(\text{img}1) \]

\[ F_2 = \text{detect\_describe}(\text{img}2) \]

for \( f_1 \) in \( F_1 \):

\[ f_2 = \text{closest match according to SSD} \]

\[ f_2' = \text{second-closest match according to SSD} \]

if \( \frac{\text{SSD}(f_1,f_2)}{\text{SSD}(f_1, f_2')} < T \)

add \( (f_1, f_2) \) to matches
But what if we're wrong?

Decreasing the threshold $T$ gives fewer **false positives** but also fewer **true positives**.
true pos rate

0 false positive rate

AUC
But what if we're wrong?

Many good matches, but still a few outliers.
Let's suppose we were never wrong.

We have a **perfect** set of feature matches.
Suppose

Match keypoints:

<table>
<thead>
<tr>
<th>Img1</th>
<th>Img2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0)</td>
<td>(2,2)</td>
</tr>
<tr>
<td>(74, 68)</td>
<td>(76, 70)</td>
</tr>
<tr>
<td>(104, 6),</td>
<td>(106, 8)</td>
</tr>
</tbody>
</table>

\[
(x', y') = f(x, y)
\]

\[
x' = x + z
\]

\[
y' = y + z
\]
This is a geometric transformation. 

- **Filtering** was a transformation on the range:

\[ g(x) = h(f(x)) \]

- **Warping** is a transformation on the domain:

\[ g(x) = f(h(x)) \]
Filtering vs Warping
Parametric (global) Warping

- Apply the same function to all coordinates.

\[ p = (x, y) \quad T \text{ transforms image coordinates } \quad p' = (x', y') \]

\[ x', y' = T(x, y) \]
Suppose

Match keypoints:

\[
\begin{array}{c|c}
\text{Img1} & \text{Img2} \\
(0,0) & (2,2) \\
(24,63) & (48,126) \\
(10,900) & (20,1800) \\
\end{array}
\]

\[x', y' = T(x, y)\]

What is function \(T\) describes the transformation between these?

\[
\begin{align*}
\text{what is } T? & \quad x' = 2x \\
y' = 2y
\end{align*}
\]
Linear Transformations

- Linear means matrices, right?

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \text{Img 2 cords} \rightarrow \text{Ingl cords}
\]
What can we do with these?

\[ S = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \]

Scale uniformly by \( S \)