Lecture 17: Motion

Cosimo Distante
Today

- Optical flow: estimating motion in video
- Background subtraction
Video

• A video is a sequence of frames captured over time
• Now our image data is a function of space \((x, y)\) and time \((t)\)
Uses of motion

• Estimating 3D structure
• Segmenting objects based on motion cues
• Learning dynamical models
• Recognizing events and activities
• Improving video quality (motion stabilization)
Motion field

- The motion field is the projection of the 3D scene motion into the image.
Motion parallax

http://psych.hanover.edu/KRANTZ/MotionParallax/MotionParallax.html
Motion field + camera motion

Length of flow vectors inversely proportional to depth $Z$ of 3D point

Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

points closer to the camera move more quickly across the image plane
Motion field + camera motion

Zoom out  Zoom in  Pan right to left
Motion estimation techniques

• Direct methods
  • Directly recover image motion at each pixel from spatio-temporal image brightness variations
  • Dense motion fields, but sensitive to appearance variations
  • Suitable for video and when image motion is small

• Feature-based methods
  • Extract visual features (corners, textured areas) and track them over multiple frames
  • Sparse motion fields, but more robust tracking
  • Suitable when image motion is large (10s of pixels)
Optical flow

• Definition: optical flow is the *apparent* motion of brightness patterns in the image

• Ideally, optical flow would be the same as the motion field

• Have to be careful: apparent motion can be caused by lighting changes without any actual motion
Apparent motion $\neq$ motion field

Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field is zero.
Problem definition: optical flow

How to estimate pixel motion from image H to image I?

• Solve pixel correspondence problem
  – given a pixel in H, look for nearby pixels of the same color in I

Key assumptions

• color constancy: a point in H looks the same in I
  – For grayscale images, this is brightness constancy
• small motion: points do not move very far

This is called the optical flow problem

Slide credit: Steve Seitz
Figure 1.5: Data conservation assumption. The highlighted region in the right image looks roughly the same as the region in the left image, despite the fact that it has moved.
Optical flow constraints

Let’s look at these constraints more closely

• brightness constancy: Q: what’s the equation?

\[ H(x, y) = I(x + u, y + v) \]

• small motion:

\[
I(x + u, y + v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}
\]

\[ \approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \]
Optical flow equation

Combining these two equations

\[ 0 = I(x + u, y + v) - H(x, y) \]

\[ \approx I(x, y) + I_x u + I_y v - H(x, y) \]

\[ \approx (I(x, y) - H(x, y)) + I_x u + I_y v \]

\[ \approx I_t + I_x u + I_y v \]

\[ \approx I_t + \nabla I \cdot [u \ v] \]

shorthand: \( I_x = \frac{\partial I}{\partial x} \)
Optical flow equation

\[ 0 = I_t + \nabla I \cdot [u \ v] \]

Q: how many unknowns and equations per pixel?
The aperture problem
The aperture problem

\[ \nabla I \cdot (u', v') = 0 \]
The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion
Solving the aperture problem

• How to get more equations for a pixel?
• **Spatial coherence constraint:** pretend the pixel’s neighbors have the same \((u,v)\)

Figure 1.7: Spatial coherence assumption. Neighboring points in the image are assumed to belong to the same surface in the scene.
Solving the aperture problem

- How to get more equations for a pixel?
- **Spatial coherence constraint**: pretend the pixel’s neighbors have the same \((u,v)\)
  - If we use a 5x5 window, that gives us 25 equations per pixel

\[
0 = I_t(p_i) + \nabla I(p_i) \cdot [u \ v]
\]

\[
\begin{bmatrix}
I_x(p_1) & I_y(p_1) \\
I_x(p_2) & I_y(p_2) \\
\vdots & \vdots \\
I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
I_t(p_1) \\
I_t(p_2) \\
\vdots \\
I_t(p_{25})
\end{bmatrix}
\]

\[
A \ d = b
\]

25x2 2x1 25x1
Solving the aperture problem

Prob: we have more equations than unknowns

\[
A d = b
\]

25x2 2x1 25x1

minimize \( \| Ad - b \|^2 \)

Solution: solve least squares problem

• minimum least squares solution given by solution (in d) of:

\[
(A^T A) d = A^T b
\]

2x2 2x1 2x1

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\ v
\end{bmatrix} = -\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix} \begin{bmatrix}
A^T A \\ A^T b
\end{bmatrix}
\]

• The summations are over all pixels in the K x K window

• This technique was first proposed by Lucas & Kanade (1981)
Conditions for solvability

When is this solvable?

• $A^T A$ should be invertible
• $A^T A$ should not be very small
  – eigenvalues $\lambda_1$ and $\lambda_2$ of $A^T A$ should not be very small
• $A^T A$ should be well-conditioned
  – $\lambda_1 / \lambda_2$ should not be too large ($\lambda_1 =$ larger eigenvalue)
Edge

- gradients very large or very small
- large $\lambda_1$, small $\lambda_2$
Low-texture region

- gradients have small magnitude
- small $\lambda_1$, small $\lambda_2$
High-texture region

- gradients are different, large magnitudes
- large $\lambda_1$, large $\lambda_2$
Example use of optical flow: facial animation

- Markerless capture of actor’s performance

https://www.fxguide.com/featured/sci-tech-winners-uca-p-mova/
Example use of optical flow: Motion Paint

Use optical flow to track brush strokes, in order to animate them to follow underlying scene motion.

https://www.fxguide.com/featured/sci-tech-winners-ucap-mova/
Limits of the gradient method

Fails when intensity structure in window is poor
Fails when the displacement is large (typical operating range is motion of 1 pixel)

Linearization of brightness is suitable only for small displacements

• Also, brightness is not strictly constant in images

actually less problematic than it appears, since we can pre-filter images to make them look similar
Coarse-to-Fine Estimation

Pyramid of image J

Pyramid of image I

u=10 pixels

u=5 pixels

u=2.5 pixels

warp

u=1.25 pixels

Δa

image J

image I

a

Pyramid of image J

Pyramid of image I
Pyramid / “Coarse-to-fine”
Coherent Motion

Possibly Gaussian.
Multiple Motions

Definitely not Gaussian.
Motion estimation techniques

• Direct methods
  • Directly recover image motion at each pixel from spatio-temporal image brightness variations
  • Dense motion fields, but sensitive to appearance variations
  • Suitable for video and when image motion is small

• Feature-based methods
  • Extract visual features (corners, textured areas) and track them over multiple frames
  • Sparse motion fields, but more robust tracking
  • Suitable when image motion is large (10s of pixels)
Motion magnification

(a) Registered input frame
(b) Clustered trajectories of tracked features
(c) Layers of related motion and appearance
(d) Motion magnified, showing holes
(e) After texture in-painting to fill holes
(f) After user's modification to segmentation map in (c)

Liu et al. SIGGRAPH 2005
Fun with flow

- https://www.youtube.com/watch?v=3YE5tff8pqg
- http://www.youtube.com/watch?v=TbJrc6QCeU0&feature=related
- http://www.youtube.com/watch?v=pckFacsIWg4
- http://www.youtube.com/watch?v=U4taMDEozCs
Today

- Optical flow: estimating motion in video
- Background subtraction
Video as an “Image Stack”

Can look at video data as a spatio-temporal volume

- If camera is stationary, each line through time corresponds to a single ray in space

Alyosha Efros, CMU
Input Video
Average Image
Background Subtraction

- Given an image (mostly likely to be a video frame), we want to identify the **foreground objects** in that image!

![Image of a busy street with vehicles]

Motivation

- In most cases, objects are of interest, not the scene.
- Makes our life easier: less processing costs, and less room for error.
Background subtraction

• Simple techniques can do ok with static camera
• ...But hard to do perfectly

• Widely used:
  – Traffic monitoring (counting vehicles, detecting & tracking vehicles, pedestrians),
  – Human action recognition (run, walk, jump, squat),
  – Human-computer interaction
  – Object tracking
Simple Approach

Image at time $t$: $I(x, y, t)$

Background at time $t$: $B(x, y, t)$

1. Estimate the background for time $t$.
2. Subtract the estimated background from the input frame.
3. Apply a threshold, $Th$, to the absolute difference to get the foreground mask.
Frame Differencing

- Background is estimated to be the previous frame. Background subtraction equation then becomes:

\[
B(x, y, t) = l(x, y, t - 1)
\]

\[
\downarrow
\]

\[
|l(x, y, t) - l(x, y, t - 1)| > Th
\]

- Depending on the object structure, speed, frame rate and global threshold, this approach may or may not be useful (usually not).
Frame Differencing

\[ Th = 25 \]

\[ Th = 50 \]

\[ Th = 100 \]

\[ Th = 200 \]
Mean Filter

- In this case the background is the mean of the previous $n$ frames:

$$B(x, y, t) = \frac{1}{n} \sum_{i=0}^{n-1} l(x, y, t - i)$$

\[\Rightarrow\]

$$|l(x, y, t) - \frac{1}{n} \sum_{i=0}^{n-1} l(x, y, t - i)| > Th$$

- For $n = 10$:

Estimated Background

Foreground Mask

Slide credit: Birgi Tamer Soy
## Frame differences vs. background subtraction

<table>
<thead>
<tr>
<th>Test Image</th>
<th>Chair moved</th>
<th>Light gradually brightened</th>
<th>Light just switched on</th>
<th>Tree Waving</th>
<th>Foreground covers monitor pattern</th>
<th>No clean background training</th>
<th>Interior motion undetectable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal Foreground</td>
<td><img src="image" alt="Ideal Foreground" /></td>
<td><img src="image" alt="Ideal Foreground" /></td>
<td><img src="image" alt="Ideal Foreground" /></td>
<td><img src="image" alt="Ideal Foreground" /></td>
<td><img src="image" alt="Ideal Foreground" /></td>
<td><img src="image" alt="Ideal Foreground" /></td>
<td><img src="image" alt="Ideal Foreground" /></td>
</tr>
<tr>
<td>Adjacent Frame Difference</td>
<td><img src="image" alt="Adjacent Frame Difference" /></td>
<td><img src="image" alt="Adjacent Frame Difference" /></td>
<td><img src="image" alt="Adjacent Frame Difference" /></td>
<td><img src="image" alt="Adjacent Frame Difference" /></td>
<td><img src="image" alt="Adjacent Frame Difference" /></td>
<td><img src="image" alt="Adjacent Frame Difference" /></td>
<td><img src="image" alt="Adjacent Frame Difference" /></td>
</tr>
<tr>
<td>Mean &amp; Threshold</td>
<td><img src="image" alt="Mean &amp; Threshold" /></td>
<td><img src="image" alt="Mean &amp; Threshold" /></td>
<td><img src="image" alt="Mean &amp; Threshold" /></td>
<td><img src="image" alt="Mean &amp; Threshold" /></td>
<td><img src="image" alt="Mean &amp; Threshold" /></td>
<td><img src="image" alt="Mean &amp; Threshold" /></td>
<td><img src="image" alt="Mean &amp; Threshold" /></td>
</tr>
</tbody>
</table>

- Toyama et al. 1999
Median Filter

- Assuming that the background is more likely to appear in a scene, we can use the median of the previous $n$ frames as the background model:

  $$B(x, y, t) = \text{median}\{I(x, y, t - i)\}$$

  $$\Downarrow$$

  $$|I(x, y, t) - \text{median}\{I(x, y, t - i)\}| > Th$$

  where $i \in \{0, \ldots, n-1\}$.

- For $n = 10$:

  Estimated Background

  Foreground Mask

Slide credit: Birgi Tamersoy
Average/Median Image
Background Subtraction

Alyosha Efros, CMU
Pros and cons

**Advantages:**
- Extremely easy to implement and use!
- All pretty fast.
- Corresponding background models need not be constant, they change over time.

**Disadvantages:**
- Accuracy of frame differencing depends on object speed and frame rate
- Median background model: relatively high memory requirements.
- Setting global threshold Th...

*When will this basic approach fail?*
Background mixture models

Idea: model each background pixel with a mixture of Gaussians; update its parameters over time.

Adaptive Background Mixture Models for Real-Time Tracking, 1999,
Chris Stauer & W.E.L. Grimson
Example: Eulerian Video Magnification for Revealing Subtle Changes in the World

http://people.csail.mit.edu/mrub/evm/#code