CPSC 425: Computer Vision

Instructor: Jim Little
little@cs.ubc.ca

Department of Computer Science
University of British Columbia

Lecture Notes 2016/2017 Term 2
Menu March 12, 2017

Topics:
  Motion and Optical Flow (cont.)
  Clustering

Reading:
  Today: Forsyth & Ponce (2nd ed.) 6.2.2, 9.3.1, 9.3.3, 9.4.2
  Next:  Forsyth & Ponce (2nd ed.) 15.1, 15.2

Handouts:
  Assignment 6: Optical Flow

Reminders:
  Assignment 6 due Thursday, March 28
  www: http://www.cs.ubc.ca/~little/cpsc425/
Today’s “Fun” Example: Barber’s Pole Illusion

Today’s example is one of the most well known motion illusions, the barber’s pole illusion. Point your web browser to

http://en.wikipedia.org/wiki/Barber_pole_illusion

A barber’s pole rotates about its vertical axis so that the true motion is purely horizontal. But, the stripes appear to be traveling vertically up the length of the pole, rather than around it

Note: A barber’s pole is a traditional sign at a location where barbers perform their craft, dating back to the middle ages. See

http://en.wikipedia.org/wiki/Barber’s_pole
Lecture 16: Re-cap

- Stereo is formulated as a correspondence problem — determine match between location of a scene point in one image and its location in another.

- If we assume calibrated cameras and image rectification, epipolar lines are horizontal scan lines.


- Optical flow is the apparent motion of brightness patterns in the image — Usually we assume optical flow and 2-D motion coincide, but this is not necessarily the case.
Example: Flying Insects and Birds

- Bee strategy: Balance the optical flow experienced by the two eyes

Figure credit: M. Srinivasan ©IEEE
Example: Flying Insects and Birds

- How do bees land safely on surfaces?

- During their approach, bees continually adjust their speed to hold constant the optical flow in the vicinity of the target
  
  — approach speed decreases as the target is approached and reduces to zero at the point of touchdown

  — no need to estimate the distance to the target at any time
Example: Flying Insects and Birds

Figure credit: M. Srinivasan ©IEEE

- Bees approach the surface more slowly if the spiral is rotated to augment the rate of expansion, and more quickly if the spiral is rotated in the opposite direction.
Without distinct features to track, the true visual motion is ambiguous.

Locally, one can compute only the component of the visual motion in the direction perpendicular to the contour.
Visual motion is determined when there are distinct features to track, provided:

- the features can be detected and localized accurately; and
- the features can be correctly matched over time
## Motion as Matching

<table>
<thead>
<tr>
<th>Representation</th>
<th>Result is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point/feature based</td>
<td>(very) sparse</td>
</tr>
<tr>
<td>Contour based</td>
<td>(relatively) sparse</td>
</tr>
<tr>
<td>(Differential) gradient based</td>
<td>dense</td>
</tr>
</tbody>
</table>
Optical Flow Constraint Equation

Consider image intensity also to be a function of time, $t$. We write

$$l(x, y, t)$$

(1)
Optical Flow Constraint Equation

Consider image intensity also to be a function of time, $t$. We write

$$l(x, y, t)$$

(1)

Applying the chain rule for differentiation to (1), we obtain

$$\frac{dl}{dt} = l_x \frac{dx}{dt} + l_y \frac{dy}{dt} + l_t$$

where subscripts denote partial differentiation

Define $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$. Then $[u, v]$ is the 2-D motion and the space of all such $u$ and $v$ is the 2-D velocity space

Suppose $\frac{dl}{dt} = 0$. Then we obtain the (classic) optical flow constraint equation

$$l_x u + l_y v + l_t = 0$$

(2)
Optical Flow Constraint Equation (cont’d)

\[ I_x u + I_y v + I_t = 0 \]

Equation (2) determines a straight line in velocity space
Lucas–Kanade

Observations:

1. The 2-D motion, $[u, v]$, at a given point, $[x, y]$, has two degrees-of-freedom

2. The partial derivatives, $I_x$, $I_y$, $I_t$, provide one constraint

3. The 2-D motion, $[u, v]$, cannot be determined locally from $I_x$, $I_y$, $I_t$ alone

Lucas–Kanade Idea:

Obtain additional local constraint by computing the partial derivatives, $I_x$, $I_y$, $I_t$, in a window centered at the given $[x, y]$
Suppose \([x_1, y_1] = [x, y]\) is the (original) center point in the window. Let \([x_2, y_2]\) be any other point in the window. This gives us two equations that we can write as

\[
\begin{align*}
I_{x_1} u + I_{y_1} v &= -l_{t_1} \\
I_{x_2} u + I_{y_2} v &= -l_{t_2}
\end{align*}
\]

and that can be solved locally for \(u\) and \(v\) as

\[
\begin{bmatrix}
u \\
v
\end{bmatrix} = - \begin{bmatrix}
I_{x_1} & I_{y_1} \\
I_{x_2} & I_{y_2}
\end{bmatrix}^{-1} \begin{bmatrix}
l_{t_1} \\
l_{t_2}
\end{bmatrix}
\]

provided that \(u\) and \(v\) are the same in both equations and provided that the required matrix inverse exists.
Lucas–Kanade (cont’d)

Considering all \( n \) points in the window, one obtains

\[
\begin{align*}
    l_{x_1} u + l_{y_1} v &= -l_{t_1} \\
    l_{x_2} u + l_{y_2} v &= -l_{t_2} \\
    \vdots & \quad \vdots \\
    l_{x_n} u + l_{y_n} v &= -l_{t_n}
\end{align*}
\]

which can be written as the matrix equation

\[
A \mathbf{v} = \mathbf{b} \tag{3}
\]

where \( \mathbf{v} = [u, v]^T \), \( A = \begin{bmatrix} l_{x_1} & l_{y_1} \\ l_{x_2} & l_{y_2} \\ \vdots & \vdots \\ l_{x_n} & l_{y_n} \end{bmatrix} \) and \( \mathbf{b} = -\begin{bmatrix} l_{t_1} \\ l_{t_2} \\ \vdots \\ l_{t_n} \end{bmatrix} \).
Lucas–Kanade (cont’d)

The standard least squares solution, $\bar{v}$, to (3) is

$$\bar{v} = (A^T A)^{-1} A^T b$$

again provided that $u$ and $v$ are the same in all equations and provided that the rank of $A^T A$ is 2 (so that the required inverse exists)
Lucas–Kanade (cont’d)

Note that we can explicitly write down an expression for $A^T A$ as

$$A^T A = \begin{bmatrix} \sum l_x^2 & \sum l_x l_y \\ \sum l_x l_y & \sum l_y^2 \end{bmatrix}$$

which is identical to the matrix $C$ that we saw in the context of Harris corner detection.
Lucas–Kanade: Summary

A dense method to compute motion, \([u, v]\), at every location in an image

Key Assumptions:

1. Motion is slow enough and smooth enough that differential methods apply (i.e., that the partial derivatives, \(l_x, l_y, l_t\), are well-defined)

2. The optical flow constraint equation holds (i.e., \(\frac{dl}{dt} = 0\))

3. A window size is chosen so that motion, \([u, v]\), is constant in the window

4. A window size is chosen so that the rank of \(A^T A\) is 2 for the window
Many methods trade off a ‘departure from the optical flow constraint’ cost with a ‘departure from smoothness’ cost.

The optimization objective to minimize becomes

$$E = \int \int [(l_x u + l_y v + l_t)^2 + \lambda(\|\nabla u\|^2 + \|\nabla v\|^2)] \, dx \, dy$$

where $\lambda$ is a weighing parameter.
Grouping in Human Vision

Humans routinely group features that belong together when looking at a scene. What are some cues that we use for grouping?
Humans routinely group features that belong together when looking at a scene. What are some cues that we use for grouping?

— Similarity
— Symmetry
— Common Fate
— Proximity
— ...
A Kanizsa triangle  
B Tse’s volumetric worm  
C Idesawa’s spiky sphere  
D Tse’s “sea monster”

Figure credit: Steve Lehar

http://cns-alumni.bu.edu/~slehar/Lehar.html
Grouping in Human Vision

Slide credit: Kristen Grauman
Grouping in Human Vision

Incredible way of making my two star review seem like I didn't hate the film


Slide credit: Kristen Grauman
Clustering

It is often useful to be able to group together image regions with similar appearance (e.g. roughly coherent colour or texture)

- image compression
- approximate nearest neighbour search
- base unit for higher-level recognition tasks
- moving object detection in video sequences
- video summarization
Clustering is a set of techniques to try to find components that belong together (i.e., components that form clusters).
— Unsupervised learning

Two basic clustering approaches are
- agglomerative clustering
- divisive clustering
Agglomerative Clustering

Each data point starts as a separate cluster. Clusters are recursively merged.

Algorithm:

Make each point a separate cluster
Until the clustering is satisfactory
    Merge the two clusters with the smallest inter-cluster distance
end
Divisive Clustering

The entire data set starts as a single cluster. Clusters are recursively split.

Algorithm:
Construct a single cluster containing all points
Until the clustering is satisfactory
    Split the cluster that yields the two components
    with the largest inter-cluster distance
end
Summary

- Motion, like binocular stereo, can be formulated as a matching problem. That is, given a scene point located at \((x_0, y_0)\) in an image acquired at time \(t_0\), what is its position, \((x_1, y_1)\), in an image acquired at time \(t_1\)?

- Assuming image intensity does not change as a consequence of motion, we obtain the (classic) **optical flow constraint equation**

\[
l_x u + l_y v + l_t = 0
\]

where \([u, v]\), is the 2-D motion at a given point, \([x, y]\) and \(l_x, l_y, l_t\) are the partial derivatives of intensity with respect to \(x, y,\) and \(t\)

- **Lucas–Kanade** is a dense method to compute the motion, \([u, v]\), at every location in an image