Lecture 24: Optical Flow
Today’s “fun” Example: Visual Microphone

The Visual Microphone: Passive Recovery of Sound from Video

Abe Davis
Michael Rubinstein
Neal Wadhwa
Gautham J. Mysore
Fredo Durand
William T. Freeman

Follow-up work to previous lecture’s example of Eulerian video magnification
Optical Flow

Problem:
Determine how objects (and/or the camera itself) move in the 3D world

Key Idea(s):
Images acquired as a (continuous) function of time provide additional constraint. Formulate motion analysis as finding (dense) point correspondences over time.
Optical Flow and 2D Motion

Optical flow is the apparent motion of brightness patterns in the image.

Applications
— image and video stabilization in digital cameras, camcorders
— motion-compensated video compression schemes such as MPEG
— image registration for medical imaging, remote sensing
— action recognition
— motion segmentation
**Optical Flow** and 2D Motion

Motion is geometric

Optical flow is radiometric

Usually we assume that optical flow and 2-D motion coincide ...

but this is not always the case!
Optical Flow and 2D Motion

Optical flow but no motion . . .
Optical Flow and 2D Motion

Optical flow but no motion . . .

. . . moving light source(s), lights going on/off, inter-reflection, shadows
**Optical Flow** and 2D Motion

**Optical flow** but no motion . . .

. . . moving light source(s), lights going on/off, inter-reflection, shadows

**Motion** but no optical flow . . .
Optical Flow and 2D Motion

Optical flow but no motion . . .
. . . moving light source(s), lights going on/off, inter-reflection, shadows

Motion but no optical flow . . .
. . . spinning sphere.
Visual Motion

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
Optical Flow Constraint Equation

Brightness Constancy Assumption: Brightness of the point remains the same

\[ I(x(t), y(t), t) = C \]

constant

What does this mean, and why is it reasonable?

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)
How do we compute ...

\[ I_x u + I_y v + I_t = 0 \]
How do we compute ...

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

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

spatial derivative

Forward difference
Sobel filter
Scharr filter
...

\[ I_t = \frac{\partial I}{\partial t} \]

temporal derivative

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Equation 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 $[x, y]$

**Constant Flow Assumption:** nearby pixels will likely have same optical flow
Suppose \([x_1, y_1] = [x, y]\) is the (original) centre 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 and that can be solved locally for \(u\) and \(v\) as

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

and

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

provided that \(u\) and \(v\) are the same in both equations and provided that the required matrix inverse exists.
Optical Flow Constraint Equation (two points)

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

\[ I_x u + I_y v + I_t = 0 \]
Considering all $n$ points in the window, one obtains

\[ I_{x1}u + I_{y1}v = -I_{t1} \]
\[ I_{x2}u + I_{y2}v = -I_{t2} \]
\[ \vdots \]
\[ I_{xn}u + I_{yn}v = -I_{tn} \]

which can be written as the matrix equation

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

where \( \mathbf{v} = [u, v]^T \), \( \mathbf{A} = \begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \) and \( \mathbf{b} = -\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tn} \end{bmatrix} \)
The standard least squares solution, $\mathbf{v}$, to this is

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

again provided that $\mathbf{u}$ and $\mathbf{v}$ are the same in all equations and provided that the rank of $\mathbf{A}^T \mathbf{A}$ is 2 (so that the required inverse exists)
Lucas-Kanade

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

$$A^T A = \begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix}$$

which is identical to the matrix $C$ that we saw in the context of Harris corner detection.
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 \(I_x, I_y, I_t\) are well-defined)

2. The optical flow constraint equation holds, i.e., \[\frac{dI(x, y, t)}{dt} = 0\]

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
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**

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

where \([u, v]\) is the 2-D motion at a given point, \([x, y]\) and \(I_x, I_y, I_t\) 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.