**Optical flow**

- How to estimate pixel motion from image $I_1$ to image $I_2$?
  - Solve pixel correspondence problem
    - given a pixel in $I_1$, look for nearby pixels of the same color in $I_2$.
  - Key assumptions
    - color constancy: a point in $I_1$ looks the same in $I_2$
      - For grayscale images, this is brightness constancy.
    - small motion: points do not move very far
  - This is called the **optical flow** problem.
Optical Flow and Motion Field

- Optical flow field is the apparent motion of brightness patterns between 2 (or several) frames in an image sequence.
- Why does brightness change between frames?
  - Assuming that illumination does not change:
    - Changes are due to the RELATIVE MOTION between the scene and the camera.
    - There are 3 possibilities:
      - Camera still, moving scene
      - Moving camera, still scene
      - Moving camera, moving scene

Examples of Motion Fields

(a) Translation perpendicular to a surface. (b) Rotation about axis perpendicular to image plane. (c) Translation parallel to a surface at a constant distance. (d) Translation parallel to an obstacle in front of a more distant background.

Motion Field (MF)

- The MF assigns a velocity vector to each pixel in the image.
- These velocities are INDUCED by the RELATIVE MOTION between the camera and the 3D scene.
- The MF is the projection of the 3D velocities on the image plane.

Optical Flow vs. Motion Field

- Recall that Optical Flow is the apparent motion of brightness patterns.
- We equate Optical Flow Field with Motion Field.
- Frequently works, but not always:
  - (a) A smooth sphere is rotating under constant illumination. Thus the optical flow field is zero, but the motion field is not.
  - (b) A fixed sphere is illuminated by a moving source—the shading of the image changes. Thus the motion field is zero, but the optical flow field is not.
**Optical Flow vs. Motion Field**

- Often (but not always) optical flow corresponds to the true motion of the scene.

**Computing Optical Flow: Brightness Constancy Equation**

\[
E(x(t), y(t), t) = \text{Constant}
\]

Taking derivative wrt time:

\[
\frac{dE(x(t), y(t), t)}{dt} = 0
\]

\[
\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0
\]

**Computing Optical Flow: Brightness Constancy Equation**

- Let \( P \) be a moving point in 3D:
  - At time \( t \), \( P \) has coordinates \((X(t), Y(t), Z(t))\)
  - Let \( p = (x(t), y(t)) \) be the coordinates of its image at time \( t \)
  - Let \( E(x(t), y(t), t) \) be the brightness at \( p \) at time \( t \).
- Brightness Constancy Assumption:
  - As \( P \) moves over time, \( E(x(t), y(t), t) \) remains constant.

\[
\begin{align*}
\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} &= 0 \\
\n\end{align*}
\]
**Computing Optical Flow: Brightness Constancy Equation**

- How to get more equations for a pixel?
  - Basic idea: impose additional constraints
    - most common is to assume that the flow field is smooth locally
    - one method: pretend the pixel's neighbors have the same \((u,v)\)
      - If we use a 5x5 window, that gives us 25 equations per pixel!

\[
E_x(p_i) + \nabla E(p_i) \cdot [u \ v] = 0
\]

\[
\begin{bmatrix}
E_x(p_1) & E_x(p_2) \\
E_x(p_2) & E_x(p_3) \\
E_x(p_3) & E_x(p_4) \\
E_x(p_4) & E_x(p_5)
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
=
\begin{bmatrix}
E_x(p_1) \\
E_x(p_2) \\
E_x(p_3) \\
E_x(p_4) \\
E_x(p_5)
\end{bmatrix}
\]

matrix \(E\) vector \(d\) vector \(b\)

- 25x2 2x1 25x1

---

**Optical Flow Results**

- Lucas-Kanade without pyramids
- Fails in areas of large motion

---

**Video Sequence**

---

**Revisiting the small motion assumption**

- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2nd order terms dominate)
  - How might we solve this problem?
Reduce the resolution!

Iterative Refinement

- Iterative Lukas-Kanade Algorithm
  1. Estimate velocity at each pixel by solving Lucas-Kanade equations
  2. Warp H towards I using the estimated flow field
     - use image warping techniques
  3. Repeat until convergence

Coarse-to-fine optical flow estimation
Optical Flow Results

- From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Other Concepts to Review

- Convolution is the operation of applying a “kernel” to each pixel of an image

\[
\begin{align*}
\text{image} & \quad \text{kernel} \\
I & \quad K_{11} \quad K_{12} \\
I & \quad K_{21} \quad K_{22} \\
I & \quad K_{31} \quad K_{32} \\
\end{align*}
\]

- Result of convolution has the same dimension as the image
- For example:
  \[O_{12} = I_{07}K_{11} + I_{08}K_{12} + I_{09}K_{21} + I_{07}K_{22} + I_{08}K_{31} + I_{09}K_{32}\]
- Convolution is frequently denoted by \(*\), for example \(I*K\)

Other Concepts to Review

- Gaussian smoothing (blurring): convolution operator that is used to ‘blur’ images and removes small detail and noise from an image

\[
\begin{bmatrix}
1 & 4 & 7 & 4 & 1 \\
4 & 16 & 26 & 16 & 4 \\
7 & 26 & 41 & 26 & 7 \\
4 & 16 & 26 & 16 & 4 \\
1 & 4 & 7 & 4 & 1 \\
\end{bmatrix}
\]


**Gaussian vs. Smoothing**

<table>
<thead>
<tr>
<th>Gaussian Smoothing</th>
<th>Smoothing by Averaging</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Gaussian Smoothing Image]</td>
<td>![Smoothing by Averaging Image]</td>
</tr>
</tbody>
</table>

\[
\begin{array}{ccc|ccc}
1 & 4 & 7 & 4 & 1 \\
7 & 5 & 10 & 7 & 4 \\
4 & 6 & 10 & 4 & 1 \\
4 & 6 & 7 & 4 & 1 \\
4 & 7 & 4 & 1 & 4 \\
\hline
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

\[
\frac{1}{25}
\]

**Other Concepts to Review**

- Cross-correlation:
  \[c(f, g) = \sum_{i=1}^{d} f(i)g(i)\]
  - Measures similarity between images (or image regions) \(f\) and \(g\)
  - Works OK if there is no change in intensity
  - Normalized cross correlation, more popular in image processing
  - Insensitive to linear intensity changes between image patches \(f\) and \(g\)

\[
NCC(f, g) = \frac{\sum f(i) - \bar{f} \sum g(i) - \bar{g}}{\sqrt{\sum f(i) - \bar{f}^2 \sum g(i) - \bar{g}^2}}
\]

**Other Concepts to Review**

- Image gradient: points in the direction of the most rapid increase in intensity of image \(f\)
  \[
  \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}
  \]

- Sobel operator to compute gradient:
  \[
  \begin{bmatrix}
  -1 & 1 \\
  -2 & 0 & 2 \\
  -1 & 0 & 1 \\
  \end{bmatrix}
  \]

**Results:**

\[
\begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & 2 & -1 \\
\end{bmatrix}
\]

**Next Time**

- Paper: “Recognizing Action at a Distance” by A. Efros, A. Berg, G. Mori, Jitendra Malik
  - Also maybe: “80 million tiny images: a large dataset for non-parametric object and scene recognition”, A. Torralba, R. Fergus, W. Freeman
- When reading the paper, think about following:
  - What is the problem paper tries to solve
  - What makes this problem difficult?
  - What is the method used in the paper to solve the problem
  - What is the contribution of the paper (what new does it do)?
  - Do the experimental results look “good” to you?