Computing Optical Flow
Lecture-7
Optical Flow

• Motion vector \((u,v)\)
  – Image Displacement in \(x\) and \(y\) directions between two consecutive frames

• Motion Flow

• Displacement Vector

• Disparity (general case)
Hamburg Taxi seq
Optical Flow Field Examples
Optical Flow - Examples

Videos

Color Coded Optical Flows
Optical Flow - Examples

Videos

Color Coded Optical Flows
Optical Flow
Optical Flow

• Applications
  – Motion based segmentation
  – Structure from Motion (3D shape and Motion)
  – Alignment (Global motion compensation)
    • Camcorder video stabilization
    • UAV Video Analysis
  – Video Compression
Determining Optical Flow

Berthold K.P. Horn and Brian G. Schunck
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

ABSTRACT
Optical flow cannot be computed locally, since only one independent measurement is available from the image sequence at a point, while the flow velocity has two components. A second constraint is needed. A method for finding the optical flow pattern is presented which assumes that the apparent velocity of the brightness pattern varies smoothly almost everywhere in the image. An iterative implementation is shown which successfully computes the optical flow for a number of synthetic image sequences. The algorithm is robust in that it can handle image sequences that are quantized rather coarsely in space and time. It is also insensitive to quantization of brightness levels and additive noise.

1. Introduction
Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. Optical flow can arise from relative motion of objects and the viewer [6, 7]. Consequently, optical flow can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement [8]. Discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects [27]. Attempts have been made to perform such segmentation using differences between successive image frames [15, 16, 17, 18]. Several papers address the problem of recovering the motions of objects relative to the viewer from the optical flow [19, 18, 19, 21, 29]. Some recent papers provide a clear exposition of this enterprise [30, 31]. The mathematics can be made rather difficult, by the way, by choosing an inconvenient coordinate system. In some cases information about the shape of an object may also be recovered [3, 18, 19].

These papers begin by assuming that the optical flow has already been determined. Although some reference has been made to schemes for computing.
Horn&Schunck Optical Flow

Brightness constancy assumption

\[ f(x, y, t) = f(x + dx, y + dy, t + dt) \]
Horn & Schunck Optical Flow

Brightness constancy assumption

\[ f(x, y, t) = f(x + dx, y + dy, t + dt) \]
Taylor Series

\[
f(x) \approx f(a) + \frac{f'(a)}{1!} (x - a) + \frac{f''(a)}{2!} (x - a)^2 + \frac{f^{(3)}(a)}{3!} (x - a)^3 + \ldots.
\]
Horn & Schunck Optical Flow

Brightness constancy assumption

\[ f(x, y, t) = f(x + dx, y + dy, t + dt) \]

**Taylor Series**

\[ f(x, y, t) = f(x, y, t) + \frac{\partial f}{\partial x} (x + dx - x) + \frac{\partial f}{\partial y} (y + dy - y) + \frac{\partial f}{\partial t} (t + dt - t) \]

\[ 0 = f_x dx + f_y dy + f_t dt \]

\[ 0 = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} + f_t \frac{dt}{dt} \]

\[ 0 = f_x u + f_y v + f_t \]
Interpretation of optical flow eq

\[ f_x u + f_y v + f_t = 0 \]

\[ v = -\frac{f_x}{f_y} u - \frac{f_t}{f_y} \]

\[ d = \frac{f_t}{\sqrt{f_x^2 + f_y^2}} \]

\( d = \text{normal flow} \)

\( p = \text{parallel flow} \)

Equation of st.line

\( (u, v) \)
Horn & Schunck (contd)

$$\int \int \{(f_x u + f_y v + f_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2)\} dxdy$$

**Brightness constancy**

**Smoothness constraint**

Min (Variational Calculus)

$$(f_x u + f_y v + f_t) f_x + \lambda (\Delta^2 u) = 0$$

$$(f_x u + f_y v + f_t) f_y + \lambda ((\Delta^2 v) = 0$$

$$\Delta^2 u = u_{xx} + u_{yy}$$
Derivative Masks (Roberts)

\[
\begin{bmatrix}
-1 & 1 \\
-1 & 1 \\
-1 & 1 \\
-1 & 1 \\
\end{bmatrix}_{\text{first image}} \quad \begin{bmatrix}
-1 & -1 \\
1 & 1 \\
-1 & -1 \\
1 & 1 \\
\end{bmatrix}_{\text{second image}}
\begin{bmatrix}
-1 & -1 \\
-1 & -1 \\
1 & 1 \\
1 & 1 \\
\end{bmatrix}_{\text{first image}} \quad \begin{bmatrix}
-1 & -1 \\
1 & 1 \\
-1 & -1 \\
1 & 1 \\
\end{bmatrix}_{\text{second image}}
\]

Apply first mask to 1st image
Second mask to 2nd image
Add the responses to get $f_x, f_y, f_t$
Laplacian

\[
\begin{pmatrix}
0 & -\frac{1}{4} & 0 \\
-\frac{1}{4} & 1 & -\frac{1}{4} \\
0 & -\frac{1}{4} & 0
\end{pmatrix}
\]

\[f_{xx} + f_{yy} = f - f_{av}\]
Horn & Schunck (contd)

\[ \int \int \{(f_x u + f_y v + f_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2)\} \, dx \, dy \]

\[ \text{variational calculus} \]

\[ u = u_{av} - f_x \frac{P}{D} \]

\[ v = v_{av} - f_y \frac{P}{D} \]

\[ P = f_x u_{av} + f_y v_{av} + f_t \]

\[ D = \lambda + f_x^2 + f_y^2 \]

\[ \Delta^2 u = u_{xx} + u_{yy} \]
Algorithm-1

• k=0
• Initialize
• Repeat until some error measure is satisfied (converges)

\[ u = u_{av} - f_x \frac{P}{D} \]
\[ v = v_{av} - f_y \frac{P}{D} \]
\[ P = f_x u_{av} + f_y v_{av} + f_t \]
\[ D = \lambda + f_x^2 + f_y^2 \]
Synthetic Images
Horn & Schunck Results

One iteration

10 iterations
Lucas & Kanade Method
Abstract

Image registration finds a variety of applications in computer vision. Unfortunately, traditional image registration techniques tend to be costly. We present a new image registration technique that makes use of the spatial intensity gradient of the images to find a good match using a type of Newton-Raphson iteration. Our technique is faster because it examines fewer potential matches between the images than existing techniques. Furthermore, this registration technique can be generalized to handle rotation, scaling, and shearing. We show how our technique can be adapted for use in a stereo vision system.

1. Introduction

Image registration finds a variety of applications in computer vision, such as image matching for stereo vision, pattern recognition, and motion analysis. Unfortunately, existing techniques for image registration tend to be costly. Moreover, they generally fail to deal with rotation or other distortions of the images.

In this paper we present a new image registration technique that uses spatial intensity gradient information to adjust the search for the position that yields the best match. By taking more information about the images into account, this technique is able to find the best match between two images with far fewer comparisons of images than techniques which examine the possible positions of registration in some fixed order. Our technique takes advantage of the fact that in many applications the two images are already in approximate registration. This technique can be generalized to deal with arbitrary linear distortions of the image, including rotation. We then describe a stereo vision system that uses this registration technique, and suggest some further avenues for research toward making effective use of this method in stereo image understanding.

2. The registration problem

The translational image registration problem can be characterized as follows: We are given functions \( F(x) \) and \( G(x) \) which give the respective pixel values at each location \( x \) in two images, where \( x \) is a vector. We wish to find the disparity vector \( h \) which minimizes some measure of the difference between \( F(x + h) \) and \( G(x) \), for \( x \) in some region of interest \( R \). (See Figure 1.)

![Figure 1: The image registration problem](image.png)

Typical measures of the difference between \( F(x + h) \) and \( G(x) \) are:

- \( L_1 \) norm: \( \sum_{x \in R} |F(x + h) - G(x)| \)
- \( L_p \) norm: \( \left( \sum_{x \in R} (F(x + h) - G(x))^p \right)^{1/p} \)
- negative of normalized correlation: \( -\frac{\sum_{x \in R} F(x + h)G(x)}{\sqrt{\sum_{x \in R} F(x)^2 \sum_{x \in R} G(x)^2}} \)

We will propose a more general measure of image difference, of which both the \( L_p \) norm and the correlation are special cases. The \( L_p \) norm is chiefly of interest as an inexpensive approximation to the \( L_1 \) norm.
Lucas & Kanade (Least Squares)

- Optical flow eq

\[ f_x u + f_y v = -f_t \]

- Consider 3 by 3 window

\[ f_{x1} u + f_{y1} v = -f_{t1} \]

\[ \vdots \]

\[ f_{x9} u + f_{y9} v = -f_{t9} \]

\[ \begin{bmatrix} f_{x1} & f_{y1} \\ \vdots & \vdots \\ f_{x9} & f_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ \vdots \\ -f_{t9} \end{bmatrix} \]

\[ Au = f_t \]
Lucas & Kanade

\[ Au = f_t \]

\[ A^T Au = A^T f_t \]

\[ u = (A^T A)^{-1} A^T f_t \]

\[ \text{find } (u, v) \text{ s.t. } \min \sum_i (f_{xi} u + f_{yi} v + f_t)^2 \]

Pseudo Inverse

Least Squares Fit
Lucas & Kanade

\[
\min \sum_i (f_{xi}u + f_{yi}v + f_t)^2
\]

\[
\frac{\partial}{\partial u} \sum_i (f_{xi}u + f_{yi}v + f_t)^2 = 0 \quad \sum_i (f_{xi}u + f_{yi}v + f_t)f_{xi} = 0
\]

\[
\frac{\partial}{\partial v} \sum_i (f_{xi}u + f_{yi}v + f_t)^2 = 0 \quad \sum_i (f_{xi}u + f_{yi}v + f_t)f_{yi} = 0
\]
Lucas & Kanade

\[ \sum (f_{xi}u + f_{yi}v + f_{ti})f_{xi} = 0 \]

\[ \sum (f_{xi}u + f_{yi}v + f_{ti})f_{yi} = 0 \]

\[ \sum f_{xi}^2u + \sum f_{xi}f_{yi}v = -\sum f_{xi}f_{ti} \]
\[ \sum f_{xi}f_{yi}u + \sum f_{yi}^2v = -\sum f_{yi}f_{ti} \]

\[
\begin{bmatrix}
\sum f_{xi}^2 & \sum f_{xi}f_{yi} \\
\sum f_{xi}f_{yi} & \sum f_{yi}^2
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} =
\begin{bmatrix}
-\sum f_{xi}f_{ti} \\
-\sum f_{yi}f_{ti}
\end{bmatrix}
\]
Lucas & Kanade

\[
\begin{bmatrix}
\mathbf{u} \\
\mathbf{v}
\end{bmatrix}
= \left[ \begin{array}{cc}
\sum f_{xi}^2 & \sum f_{xi}f_{yi} \\
\sum f_{xi}f_{yi} & \sum f_{yi}^2
\end{array} \right]^{-1}
\begin{bmatrix}
-\sum f_{xi}f_{ti} \\
-\sum f_{yi}f_{ti}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\mathbf{u} \\
\mathbf{v}
\end{bmatrix}
= \frac{1}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi}f_{yi})^2}
\begin{bmatrix}
\sum f_{yi}^2 & -\sum f_{xi}f_{yi} \\
-\sum f_{xi}f_{yi} & \sum f_{xi}^2
\end{bmatrix}
\begin{bmatrix}
-\sum f_{xi}f_{ti} \\
-\sum f_{yi}f_{ti}
\end{bmatrix}
\]
Lucas & Kanade

\[ u = \frac{-\sum f_y^2 \sum f_x f_t + \sum f_x f_y \sum f_y f_t}{\sum f_x^2 \sum f_y^2 - (\sum f_x f_y)^2} \]

\[ v = \frac{\sum f_x f_t \sum f_x f_y - \sum f_x^2 \sum f_y f_t}{\sum f_x^2 \sum f_y^2 - (\sum f_x f_y)^2} \]
Lucas-Kanade without pyramids

Fails in areas of large motion
Lucas-Kanade with Pyramids
Comments

• Horn-Schunck and Lucas-Kanade optical methods work only for small motion.
• If object moves faster, the brightness changes rapidly,
  – 2x2 or 3x3 masks fail to estimate spatiotemporal derivatives.
• Pyramids can be used to compute large optical flow vectors.
Lucas Kanade with Pyramids

- Compute ‘simple’ LK optical flow at highest level
- At level \( i \)
  - Take flow \( u_{i-1}, v_{i-1} \) from level \( i-1 \)
  - bilinear interpolate it to create \( u_i^*, v_i^* \) matrices of twice resolution for level \( i \)
  - multiply \( u_i^*, v_i^* \) by 2
  - compute \( f_t, f_x, f_y \) using masks centered at \((x, y)\) and \((x+u_i^*, y+v_i^*)\)
  - Apply LK to get \( u_i'(x, y), v_i'(x, y) \) (the correction in flow)
  - Add corrections \( u_i', v_i' \), i.e. \( u_i = u_i^* + u_i' \), \( v_i = v_i^* + v_i' \).
Pyramids

\[ u_{i-1}^*, v_{i-1}^* \]

\[ u_i = u_i^* + u_i', v_i = v_i^* + v_i' \]
Interpolation

\[ \begin{array}{c|c|c|c}
0 & 1 & 2 & 3 \\
\hline
0 & \cdot & \cdot & \cdot & \cdot \\
1 & \cdot & \cdot & \cdot & \cdot \\
2 & \cdot & \cdot & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & \cdot \\
\end{array} \]

\[ \begin{array}{c|c|c|c}
0 & 1 & 2 & 3 \\
\hline
0 & \cdot & \cdot & \cdot & \cdot \\
v = 1 & \cdot & \cdot & \cdot & \cdot \\
2 & \cdot & \cdot & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & \cdot \\
\end{array} \]

\[ \begin{array}{c|c|c|c|c|c|c|c}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
u = 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
u^* = 3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array} \]

\[ \begin{array}{c|c|c|c|c|c|c|c}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
v = 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
v^* = 3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array} \]
1-D Interpolation

\[ y = mx + c \]
\[ f(x) = mx + c \]
2-D Interpolation

\[ f(x, y) = a_1 + a_2 x + a_3 y + a_4 xy \]

Bilinear
Bi-linear Interpolation

Four nearest points of \((x,y)\) are:

\[(3,5), (4,5), (3,6), (4,6)\]

\[\begin{align*}
_\bar{x} &= \text{int}(x) \\
_\bar{y} &= \text{int}(y) \\
_\bar{x} &= _\bar{x} + 1 \\
_\bar{y} &= _\bar{y} + 1
\end{align*}\]

\[\begin{align*}
_\bar{x} &= 3 \\
_\bar{y} &= 5 \\
_\bar{x} &= 4 \\
_\bar{y} &= 6
\end{align*}\]
Bi-linear Interpolation

\[ f(x, y) = \varepsilon_x \varepsilon_y f(x, y) + \varepsilon_x \varepsilon_y f(x, y) + \varepsilon_x \varepsilon_y f(x, y) + \varepsilon_x \varepsilon_y f(x, y) \]

\[ \varepsilon_x = x - x \]
\[ \varepsilon_y = y - y \]

\[ \varepsilon_x = x - 4 = 4 - 3.2 = 0.8 \]
\[ \varepsilon_y = y - 6 = 6 - 5.6 = 0.4 \]

\[ \varepsilon_x = x - 3.2 = 3.2 - 2 = 0.2 \]
\[ \varepsilon_y = y - 5.6 = 5.6 - 5 = 0.6 \]

Homework
Lucas-Kanade without pyramids

Fails in areas of large motion
Lucas-Kanade with Pyramids