Chapter 7: Video Processing

Andre L. C. Barczak

Computer Science
Massey University, Albany

Contents

Introduction

Motion Analysis

Why?

- Motion detection: want to see if something has moved in a scene
- Object detection, or tracking: want to track the movement of a particular object
- Derivation of 3D properties: using two or more cameras (stereo vision)

Pixel or Block based Analysis

Motion Analysis

4 possibilities for a Pixel of a Block or Pixels

- a Block or Pixel remains static
- a Block or Pixel p moves
- a Block or Pixel p disappears
- a Block or Pixel p changes its value (lighting)

Optical Flow

Definitions

- Find: velocity and direction of moving things

The luminance variation of a video sequence is $\psi(x, y, t)$

$$\frac{\delta\psi}{\delta x}\upsilon_x + \frac{\delta\psi}{\delta y}\upsilon_y + \frac{\delta\psi}{\delta t} = 0 \tag{1}$$

The velocity vector (υ_x, υ_y) is the gradient vector of $\psi(x, y, t)$.

Vector Fields

To visualise the optical flow, it is common to use vector fields. For example, vector fields for a rotating camera would look like this:

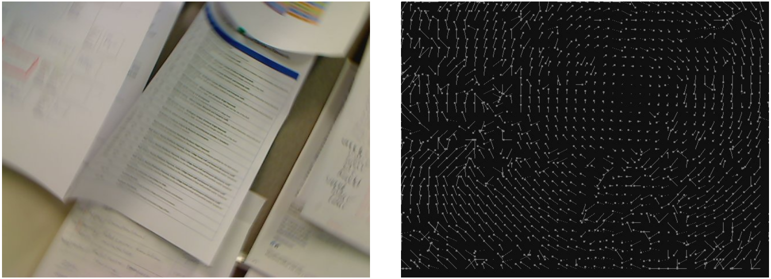


Figure 1: Optical flow example: camera rotates (notice the centre of the rotation looking at the arrows.

Vector Fields

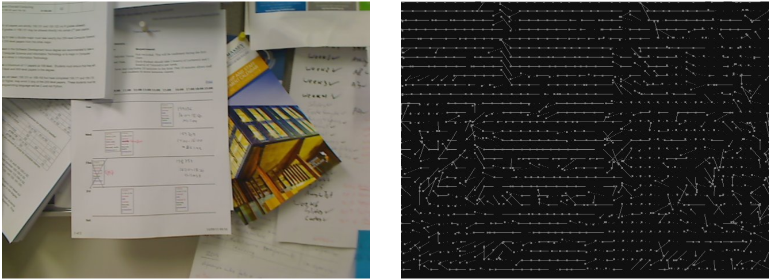
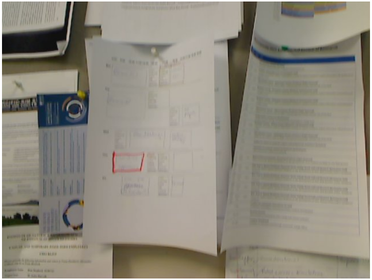


Figure 2: Optical flow example: camera moves to the right (most arrows point to the left).

Vector Fields



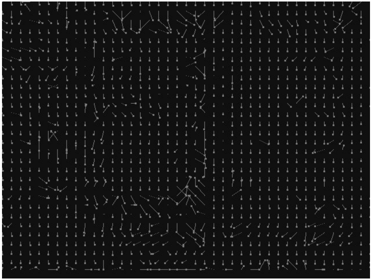
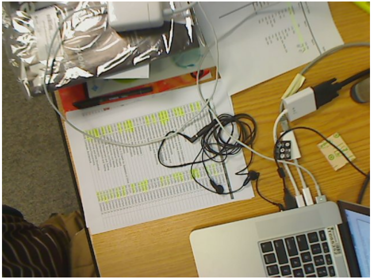


Figure 3: Optical flow example: camera goes up.

Vector Fields



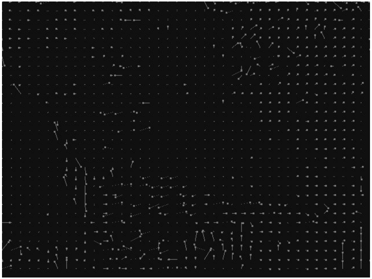


Figure 4: Optical flow example: camera approaches the objects (zoom).

Problems with Optical Flow

The 4 fundamental problems of Optical Flow

- pixel intensity has to be constant
- one equation to two unknowns
- *aperture problem*
- regions that flow, but same pixels

The aperture problem

What is it? Consider:

- being in a car with a small window to the outside
- traffic is heavy, so there are neighbouring cars close by
- person inside does not know if the car is moving or the neighbour is moving

Aperture problem example

definition

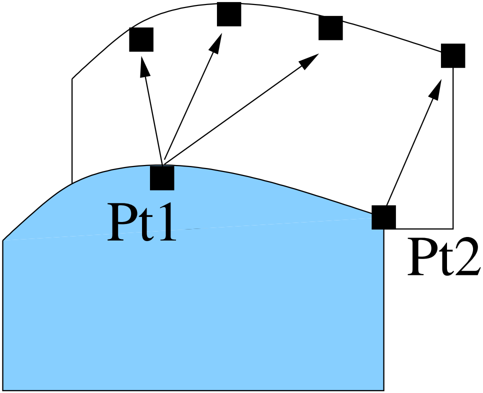


Figure 5: Aperture problem.

Aperture problem example

The barber's shop

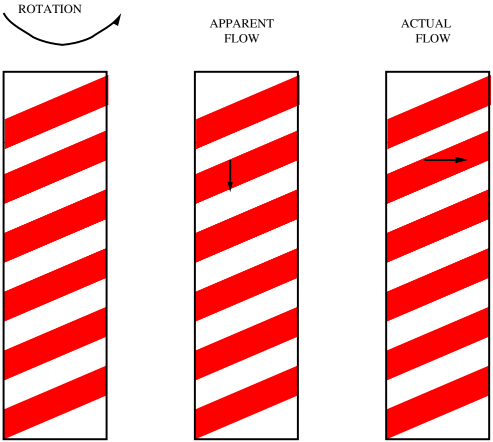


Figure 6: Determining optical flow in a barber's shop sign.

In practise...

Constraints imposed to overcome the 4 problems

- pixels change with illumination
- one pixel is not enough to determine flow
- object is considered to be rigid (hand tracking?)
- apparent flow is what matters in most cases
- velocity is limited to maximum
- do not track disappearing pixels
- create new blocks for “new things”

Some examples: Three methods

- Block matching
- Horn and Schunck
- Lukas and Kanade

Block Matching

Considerations:

- Rigid object
- Illumination changes are small
- Approach: find some properties that can match blocks, e.g., histograms

Block Matching

Error calculation:

Between two frames, find the error:

$$error = \sqrt{\sum (f(x,y) - f'(x',y'))^2} \tag{2}$$

where: $f(x,y)$ is the sub-image for a block from a frame and $f'(x',y')$ is the sub-image of a block of the next frame. The block with the minimum error is chosen as the best matching.

Problem: exhaustive search is infeasible in practise.

Block Matching

Exaustive search

Example: 512x512 image, blocks of 16x16 pixels $512/16 = 32$

blocks One moving block needs $16^2 * (512 - 32)^2$ sums and

multiplications Multiplying by the number of blocks: $32 * 32 =$

$(16^2) * ((512 - 32)^2) * (32^2) * 3 = 180.10^9$

Block Matching

Figure 7: A block matching algorithm in action.

Block Matching

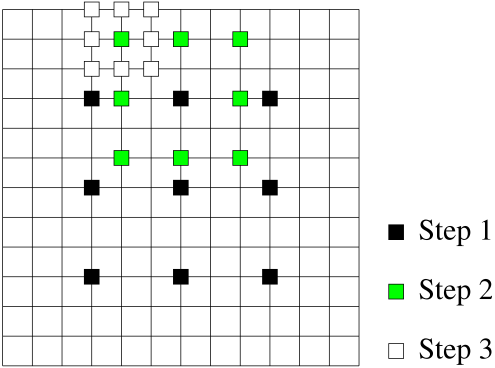


Figure 8: Three steps approach for block matching.

Block Matching

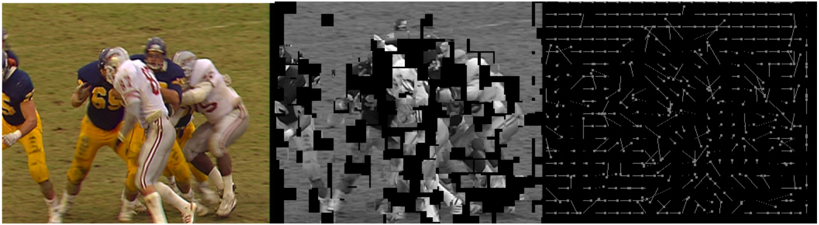


Figure 9: A block matching algorithm using the three steps approach.

Horn and Schunck

- brightness is constant
- smoothness of the flow is constrained

A system of linear equations can be solved using Jacobi:

$$u(i+1) = u_{avg}(i) - \frac{I_x(I_x u_{avg}(i) + I_y v_{avg}(i) + I_t)}{\alpha^2 + I_x^2 + I_y^2} \tag{3}$$

and

$$v(i+1) = v_{avg}(i) - \frac{I_y(I_x u_{avg}(i) + I_y v_{avg}(i) + I_t)}{\alpha^2 + I_x^2 + I_y^2} \tag{4}$$

Lucas and Kanade

- no articulation(so edges original forms are preserved, they only translate or rotate)
- the distance travelled by one edge between two sequential frames is small (typically less than one pixel).

$$h \approx \frac{\sqrt{\sum F'(x)[G(x) - F(x)]}}{\sum F'(x)^2} \quad (5)$$

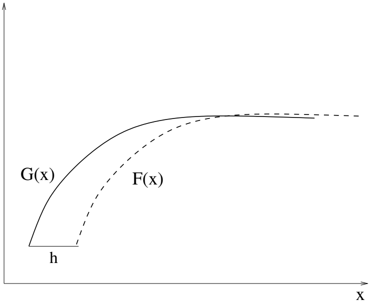


Figure 10: Lukas Kanade approach (adapted from Lucas and Kanade(81))

Other optical flow approaches

In OpenCV:

- cvCalcOpticalFlowBM
- cvCalcOpticalFlowLK
- cvCalcOpticalFlowPyrLK
- cvCalcOpticalFlowHS

Exercises

1. Download the sequences (in YUV format) from Stream. Considering that you know the size of the frames and that the yuv format uses three bytes per channel in this order, devise an algorithm to read and show the yuv sequence frame by frame.
2. Write a simple version of the Block Matching algorithm using a similarity criteria to find the block's position on the other frame. Draw the vector field using an arrow for each block, indicating direction and velocity. You need to limit the size (indicating velocity) of each arrow in order to see the field properly.
3. Run the sample LucasKanade implemented in OpenCV using the web camera. Use markers on each fingertip and see if the algorithms is robust enough to follow each finger independently.

Bibliography

(partial)

- M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, PWS Publishing, 1999.
- Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*, Prentice Hall, 2002.
- B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 674–679, 1981.
- B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, no. 17, pp. 185–203, 1981.