Mechatronic Engineering program

Computer Vision
Video Processing

Krzysztof Holak
AGH University of Krakow

1

Schedule

- Lecture 1: An Introduction
- Lecture 2: Image Segmentation
- Lecture 3: Image Features

- **Lecture 4: Video Processing**
  - Video Processing
  - Object Trackers
  - Optical Flow – Lukas-Kanade Algorithm

2

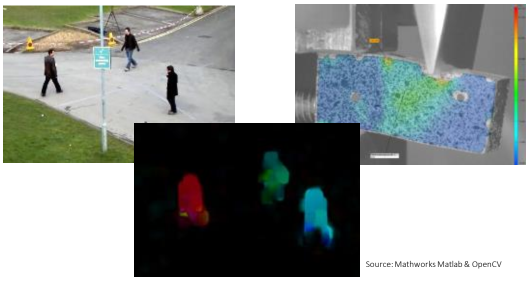
Ideas

- Video contains generalized **motion** information

- Improved image processing method (e.g. gesture recognition)
- Object detection and **tracking**
- 3D structure of the world contained in motion (**scene reconstruction**)
- Continous image processing (e.g. scene monitoring methods)

3

Ideas



Source: Mathworks Matlab & OpenCV

4

Object tracking on binary images

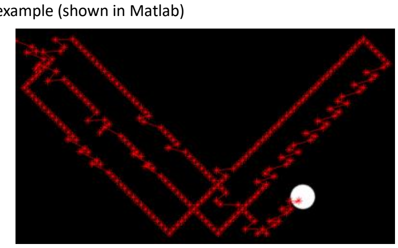
- Very simple idea
- Detect objects on each frame (thresholding, labelling, regionprops)
- Find centroids of detected objects
- Join centroids of the same object by a continous trajectory

- Post processing – velocity and acceleration computation, trajectory shape analysis, approaches analysis…

5

Object tracking on binary images

- An example (shown in Matlab)



6

Image correlation as object tracker

- Apply pattern matching for each frame of the sequence
- A pattern can be taken from the first frame of the sequence
- Or a pattern from frame (i-1) used to find the pattern on frame i

- Robust to change in intensity of a pattern
- Not robust to rotations and scaling of a pattern

7

Image correlation as object tracker

- An example (shown in Matlab)

8

Image correlation as object tracker

- An application for vibration analysis (shown im Matlab)

9

Tracking using local features

- Review from the last time:

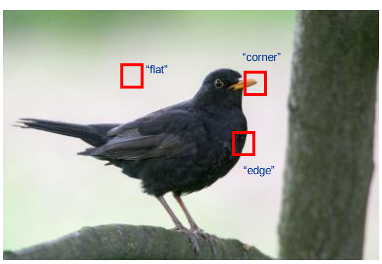
Local features describe the neighborhood of characteristic points – e.g. corners

We can build objects' representation out of these local features

- Local features can be tracked over the frames of video sequence
- Deformable object's motion can be tracked using this approach

10

Corners in the image



11

Recall Harris Corner Detector

The change of the image patch as we move by [dx dy]:

$$E(dx, dy) \approx [dx \quad dy] M \begin{bmatrix} dy \\ dy \end{bmatrix}$$

Matrix M is called **structural tensor**

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

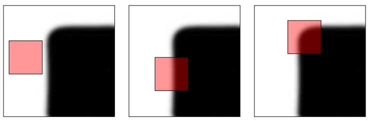
The elements of the matrix are computed from **image gradient** (derivatives in x and y directions, e.g. obtained b Sobel masks) and w(x,y) is **Gaussian mask**

12

Harris Corner Detector

$$M = R \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R^T$$

The **eigenvalus** of M reveal the intensity change in two principal orthogonal gradient directions in the window



"flat" region: λ_1 and λ_2 are small, $\lambda_1 \sim \lambda_2$;

"edge": $\lambda_1 >> \lambda_2$ or $\lambda_2 >> \lambda_1$

"corner": λ_1 and λ_2 are large, $\lambda_1 \sim \lambda_2$;

13

Harris Corner Detector

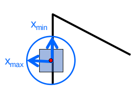
Why?

In general 2x2 matrix represents **a transformation** that changes **direction and scale** of a 2-element vector

There are 2 directions in space that are not rotated – **eigenvectors**

Scaling in these directions given by **eigenvalues**

Eigenvalues defines shifts with the smallest and largest change



$$Mx_{max} = \lambda_{max} x_{max}$$
$$Mx_{min} = \lambda_{min} x_{min}$$

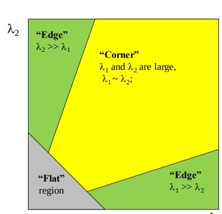
14

Harris Corner Detector

Harris gave us a simple **cornerness** score to assess both eigenvalues at the same time

$$R = \det(M) - \alpha \,\text{trace}(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α - an empirical constant (0.04 to 0.06)



15

Local feature tracking challanges

The challanges of local feature tracking

- Figure out which features can be tracked the best
- Efficiently track across frames
- Some features may change appearance over time
- Drift: small errors accumulate over time
- Features may appear or disappear

- Two approaches – **sparse tracking** (a set of features) and **dense tracking** (in principle, all pixels, dense optical flow)

16

Sparse tracking example

Detect „**good features to track"** (e.g. using Harris corner detector)

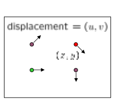
And track them

Example – using SURF features

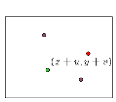
17

Optical Flow

Given two subsequent frames - estimate the feature translation



$I(x,y,t)$



$I(x,y,t+1)$

Lukas-Kanade method, basic assumptions

Brightness Constancy – feature do not change brightness over time

Small motion – only small displacements

Constancy of motions – neighbouring pixels move in the same way

18

Optical Flow

Given two subsequent frames - estimate the feature translation
Apply **brightness constancy** equation

$$I(x,y,t) = I(x+u, y+v, t+1)$$

Small motion – we can use Taylor expansion at (x,y,t)

$$I(x+u, y+v, t+1) \approx I(x,y,t) + I_x \cdot u + I_y \cdot v + I_t$$

Substitute into brightness constancy equation

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

19

Optical Flow

Optical flow equation for a single pixel

optical flow velocities

$$I_x u + I_y v + I_t = 0$$

Image gradients
(at a given pixel)

temporal gradient

20

Optical Flow

Compute all quantities in the equation $I_x u + I_y v + I_t = 0$

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

spatial derivative

High pass filter masks
Sobel Filter
Prewitt Filter
...

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Fame differencing
Subtracting consecutive frames

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

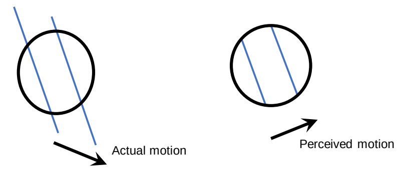
Our unknowns to be found

21

Optical Flow

In matrix form, we have $\nabla I \cdot [u \;\; v]^T + I_t = 0$

Problem: two unknowns (u,v) and one equation – we can find only the component perpendicular to the edge (**aperture problem**)



22

Optical Flow

To solve this – apply **consistency of motion** of neighbouring pixels
N pixels in a small path move in the same way – we can build N equations
N=25 in the example (5x5 patch)

$$\begin{gathered} I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v = -I_t(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v = -I_t(\boldsymbol{p}_2) \\ \vdots \\ I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v = -I_t(\boldsymbol{p}_{25}) \end{gathered}$$

Combine in a matrix form $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{b}$

23

Optical Flow

Solution to the equation given in a form

$$A^\top A \hat{x} = A^\top b$$

$$\begin{bmatrix} \sum\limits_{p\in P} I_x I_x & \sum\limits_{p\in P} I_x I_y \\ \sum\limits_{p\in P} I_y I_x & \sum\limits_{p\in P} I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum\limits_{p\in P} I_x I_t \\ \sum\limits_{p\in P} I_y I_t \end{bmatrix}$$

$$A^\top A = \begin{bmatrix} \sum\limits_{p\in P} I_x I_x & \sum\limits_{p\in P} I_x I_y \\ \sum\limits_{p\in P} I_y I_x & \sum\limits_{p\in P} I_y I_y \end{bmatrix}$$

We have seen this before, its the same matrix as in Harris Corner Detector

24

Optical Flow

What should be propertiees of this matrix?

$$A^\top A = \begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$

- Invertible
- Should not be „small", both **eigenvalues** should not be small
- Well-conditioned, one eigenvalues **should not be** much larger than the second one

- Exactly the same properties as for **corner detection**

25

Lukas-Kanade approach - problems

The smoothness of flow field **is not** preserved

Good for tracking rigid body mostly translation – cars, planes, objects in the scene etc.
Not good for tracking motion that is smooth – deofrmable bodies etc., rotations, scaling etc.

Solution: impose flow smoothness on your solution
A class of variational methos – e.g. **Horn-Schunck optical flow**

For every pixel we want: $\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ E_s(i,j) + \lambda E_d(i,j) \right\}$

26

Horn-Schunck optical flow

For every pixel we want

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ E_s(i,j) + \lambda E_d(i,j) \right\}$$

Smoothness Constraint (Flow gradients)

Brightness constancy assumption

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

Optimized using variational calculus methods
Give dense and smooth flow fields

27

Optical Flow examples

Examples in Matlab

Horn Schunck OF

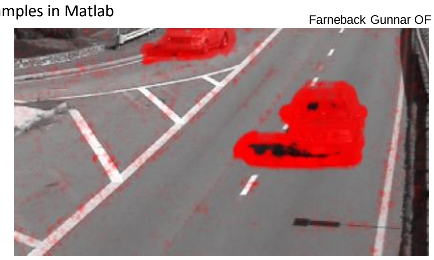
Lukas Kanade OF

28

Optical Flow examples

Examples in Matlab

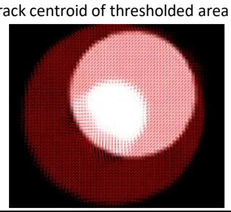
Farneback Gunnar OF



29

Optical flow for tracking rigid objects

- Compute dense flow velocity vectors for each pixel in each frame
- Threshold the flow velocity rejecting all pixels with small velocity magnitude
- Compute and track centroid of thresholded area



30

Thank you for your attention

31