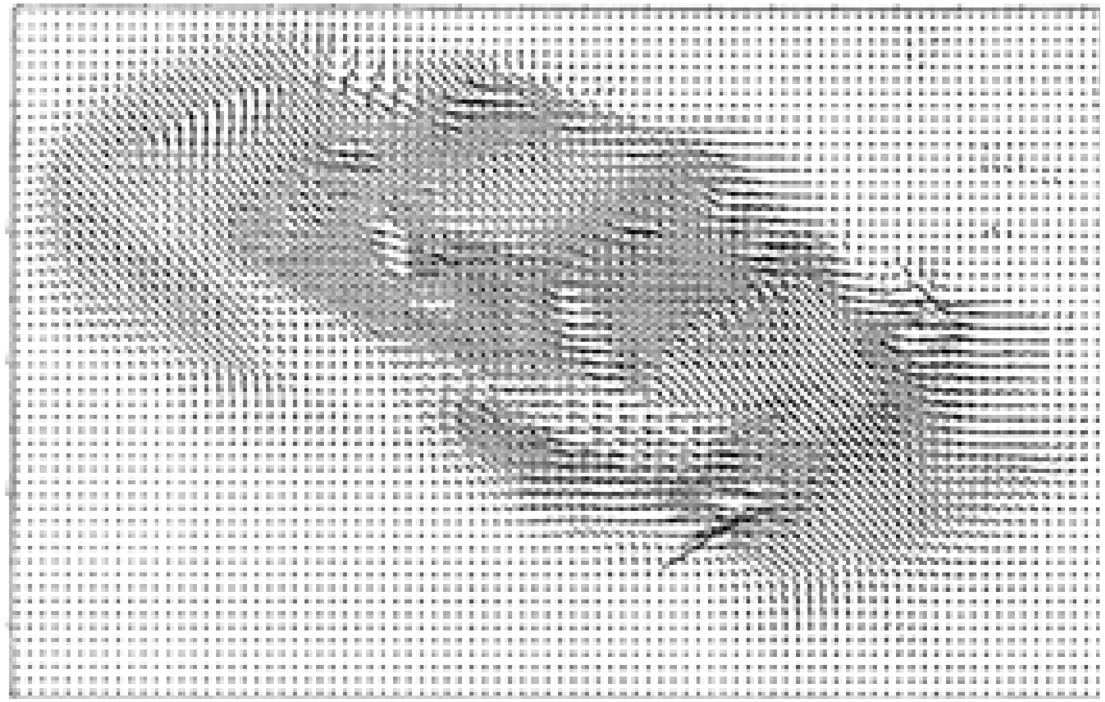
COMP9517

Computer Vision

2024 Term 3 Week 9

Professor Erik Meijering

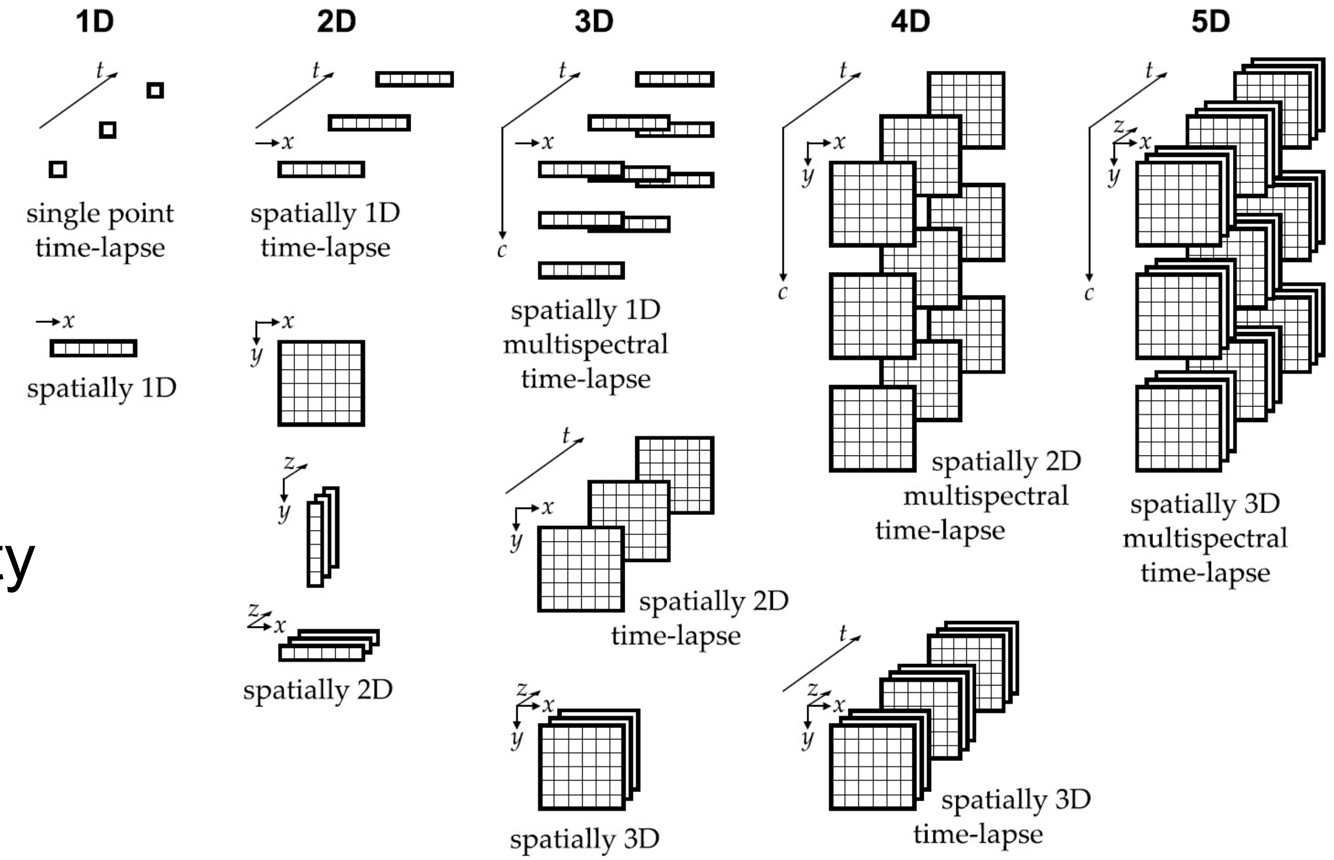
UNSW SYDNEY



Motion Estimation

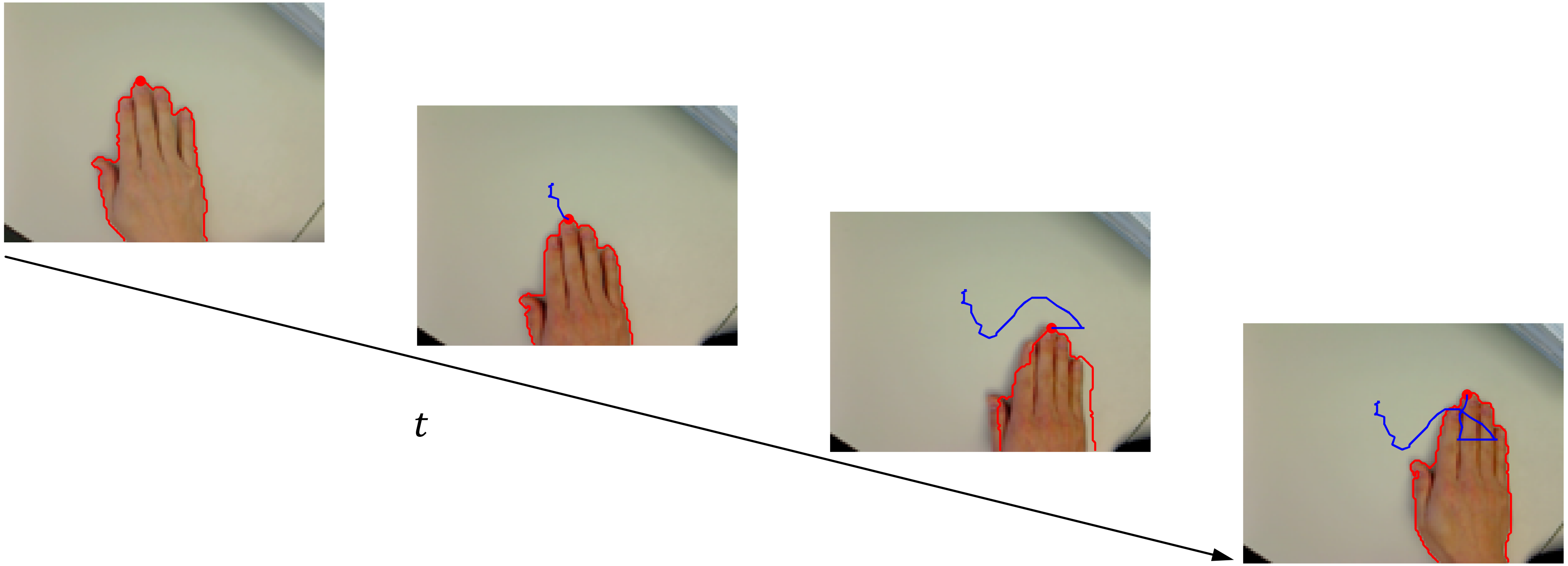
Introduction

- Adding the time dimension to the image formation
- Different nature of higher image dimensions
- Different nature of images having the same dimensionality
- For this lecture 3D = 2D+t



Introduction

- A changing scene may be observed and analysed via a sequence of images



Introduction

- Changes in an image sequence provide features for
  - Detecting objects that are moving
  - Computing trajectories of moving objects
  - Performing motion analysis of moving objects
  - Recognising objects based on their behaviours
  - Computing the motion of the viewer in the world
  - Detecting and recognising activities in a scene

Applications

- Motion-based recognition: Human identification based on gait, object detection
- Automated surveillance: Scene monitoring scene to detect suspicious activities
- Video indexing: Automatic annotation and retrieval of videos in databases
- Human-computer interaction: Gesture recognition and eye gaze tracking
- Traffic monitoring: Real-time gathering of traffic statistics to direct traffic flow
- Vehicle navigation: Video-based path planning and obstacle avoidance

Scenarios

- Still camera

  Constant background with

  – Single moving object

  – Multiple moving objects

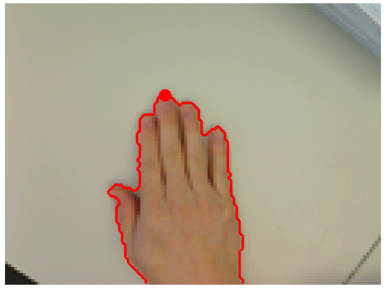
- Moving camera

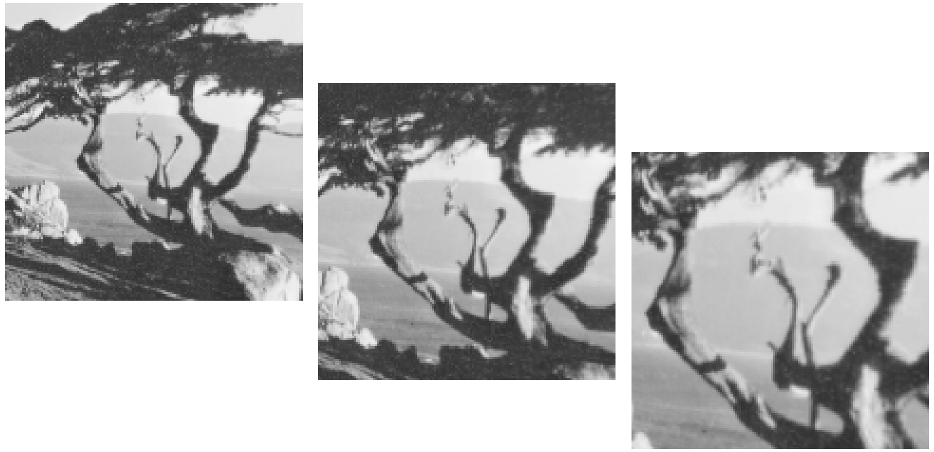
  Relatively constant scene with

  – Coherent scene motion

  – Single moving object

  – Multiple moving objects





Topics

- Change detection

  Using image subtraction to detect changes in scenes

- Sparse motion estimation

  Using template matching to estimate local displacements

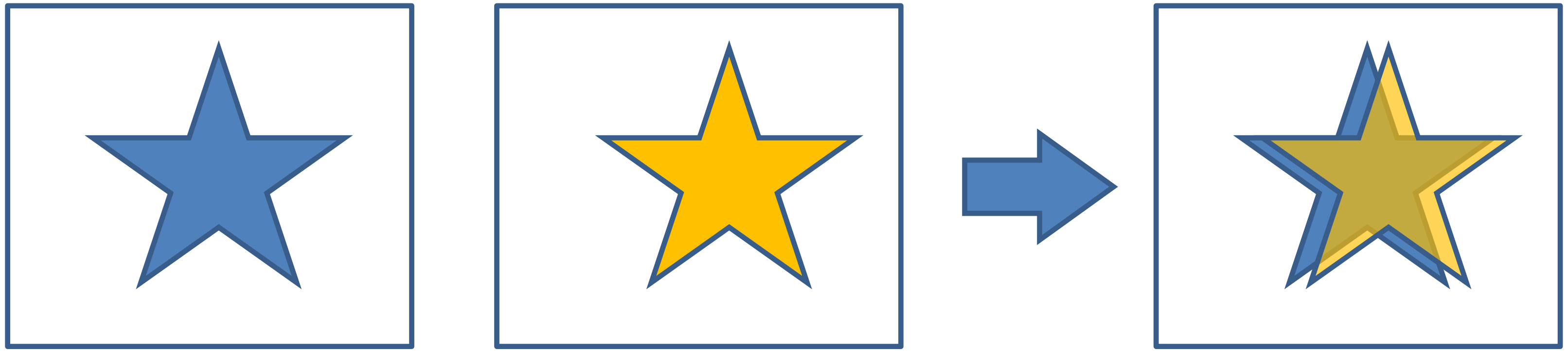
- Dense motion estimation

  Using optical flow to compute a dense motion vector field

Change detection

Change detection

- Detecting an object moving across a constant background
- Front and rear edges of the object advance only a few pixels per frame



- Subtracting image I_t from the previous image I_{t-1} reveals the changes
- Can be used to detect and localize objects that are moving

Image subtraction

Step 1: Acquire a static background image (“empty” scene)

Performance
Evaluation of
Tracking and
Surveillance
(PETS) 2009
Benchmark

Image subtraction

Step 2: Subtract the background image from each subsequent frame

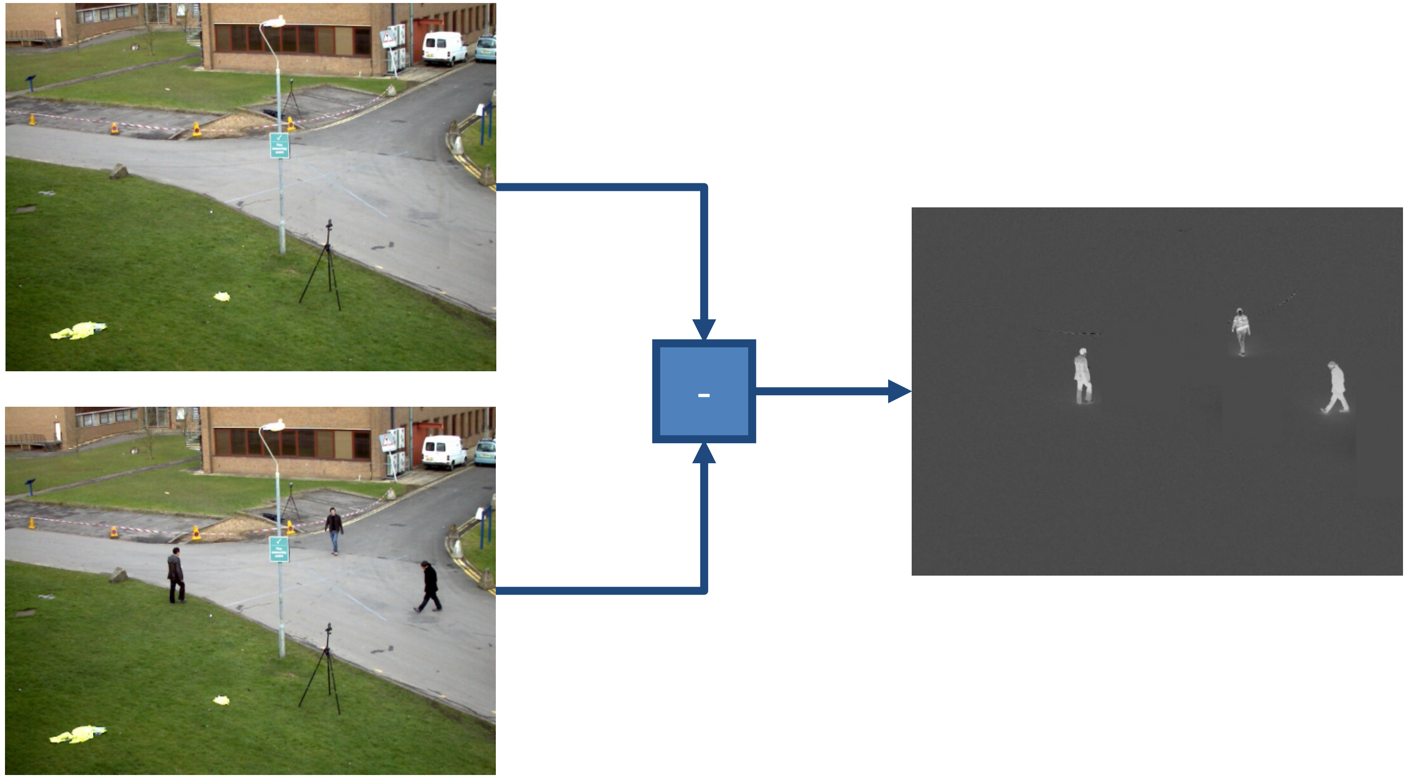


Image subtraction

Step 3: Threshold and process the difference image

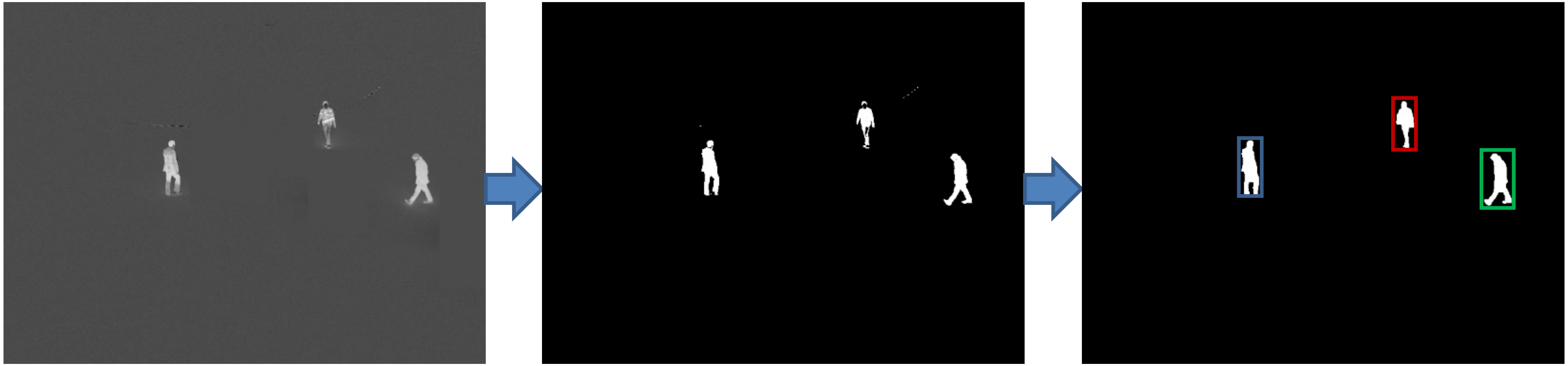


Image subtraction

Detected bounding boxes overlaid on input frame

Image subtraction algorithm

Input: Images I_t and $I_{t-\Delta t}$ (or a model image) and an intensity threshold τ

Output: Binary image I_{out} and the set of bounding boxes B

1. For all pixels (x, y) in the input images:

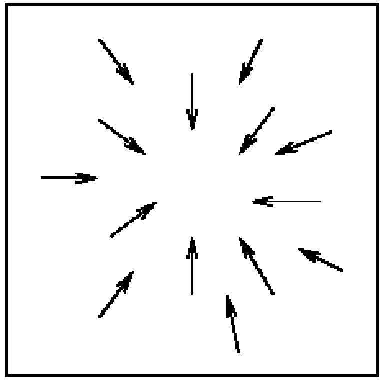
   Set $I_{\text{out}}(x, y) = 1$ if $|I_t(x, y) - I_{t-\Delta t}(x, y)| > \tau$

   Set $I_{\text{out}}(x, y) = 0$ otherwise
2. Perform connected components extraction on I_{out}
3. Remove small regions in I_{out} assuming they are noise
4. Perform a closing of I_{out} using a small disk to fuse neighbouring regions
5. Compute the bounding boxes of all remaining regions of changed pixels
6. Return $I_{\text{out}}(x, y)$ and the bounding boxes B of regions of changed pixels

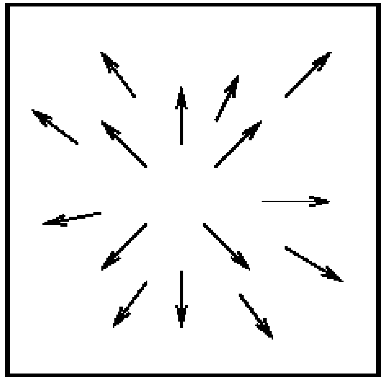
Sparse motion estimation

Motion vector

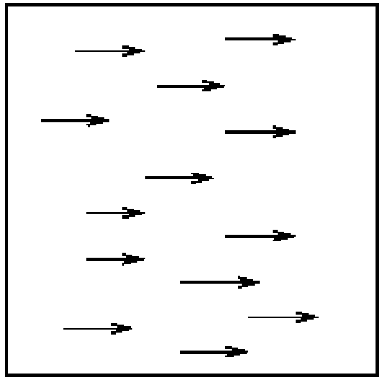
- A **motion field** is an array of 2D motion vectors
- A **motion vector** represents the displacement of a 3D point in the image
  - Tail at time t and head at time $t + \Delta t$
  - Instantaneous velocity estimate at time t



Zoom out



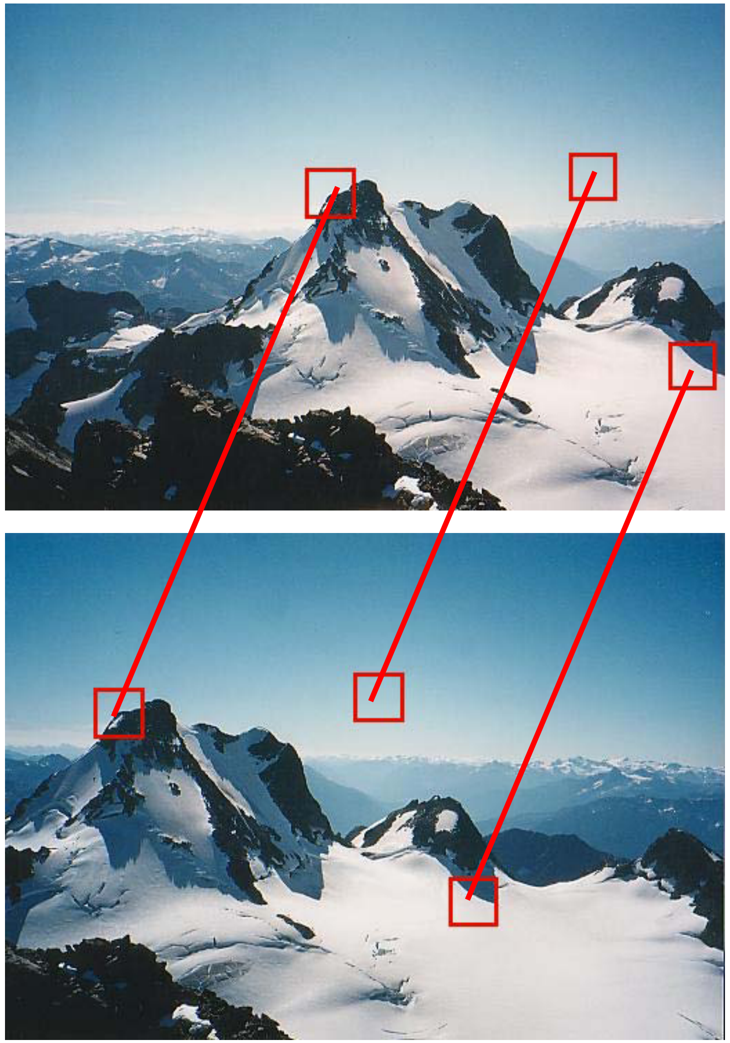
Zoom in



Pan Left

Sparse motion estimation

- A sparse motion field can be computed by identifying pairs of points that correspond in two images taken at times t and $t + \Delta t$
- Assumption: intensities of interesting points and their neighbours remain nearly constant over time
- Two steps:
  - Detect interesting points at t
  - Search corresponding points at $t + \Delta t$



Detect interesting points

- Image filters
  - Canny edge detector
  - Hessian ridge detector
  - Harris corner detector
  - Scale invariant feature transform (SIFT)
  - Convolutional neural network (CNN)
- Interest operator
  - Computes intensity variance in the vertical, horizontal and diagonal directions
  - Interest point if the minimum of these four variances exceeds a threshold

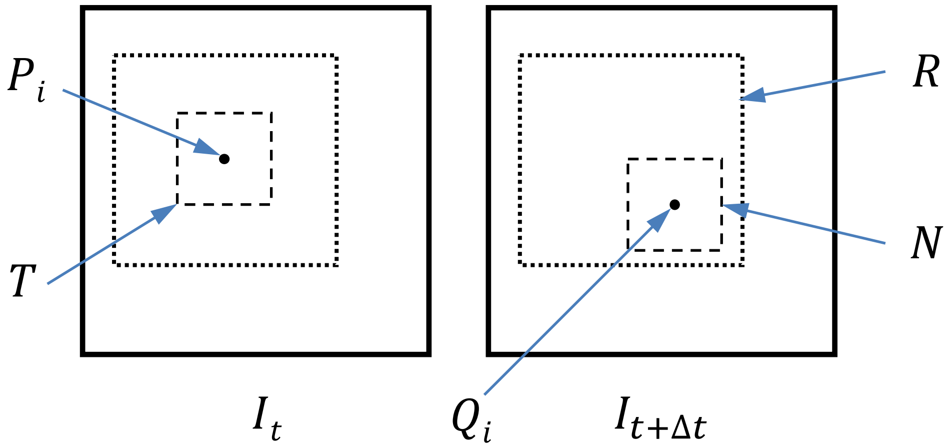
Detect interesting points

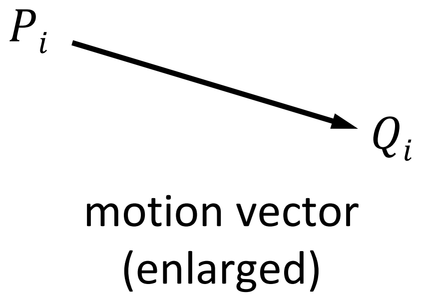
```
Procedure detect_interesting_points(I,V,w,t) {
   for (r = 0 to MaxRow-1)
      for (c = 0 to MaxCol-1)
         if (I[r,c] is a border pixel) break;
         else if (interest_operator(I,r,c,w) >= t)
            add (r,c) to set V;
}

Procedure interest_operator(I,r,c,w) {
   v1 = variance of intensity of horizontal pixels I[r,c-w]…I[r,c+w];
   v2 = variance of intensity of vertical pixels I[r-w,c]…I[r+w,c];
   v3 = variance of intensity of diagonal pixels I[r-w,c-w]…I[r+w,c+w];
   v4 = variance of intensity of diagonal pixels I[r-w,c+w]…I[r+w,c-w];
   return min(v1, v2, v3, v4);
}
```

Search corresponding points

Given an interesting point P_i from I_t, take its neighbourhood in I_t as a template T and find the best matching neighbourhood N in $I_{t+\Delta t}$ under the assumption that the amount of movement is limited to a search region R





This is also known as template matching

Similarity measures for template matching

- Cross-correlation (to be maximised)

$$\mathrm{CC}(\Delta x, \Delta y) = \sum_{(x,y)\in T} I_t(x,y) \cdot I_{t+\Delta t}(x+\Delta x, y+\Delta y)$$

- Sum of absolute differences (to be minimised)

$$\mathrm{SAD}(\Delta x, \Delta y) = \sum_{(x,y)\in T} |I_t(x,y) - I_{t+\Delta t}(x+\Delta x, y+\Delta y)|$$

- Sum of squared differences (to be minimised)

$$\mathrm{SSD}(\Delta x, \Delta y) = \sum_{(x,y)\in T} |I_t(x,y) - I_{t+\Delta t}(x+\Delta x, y+\Delta y)|^2$$

Similarity measures for template matching

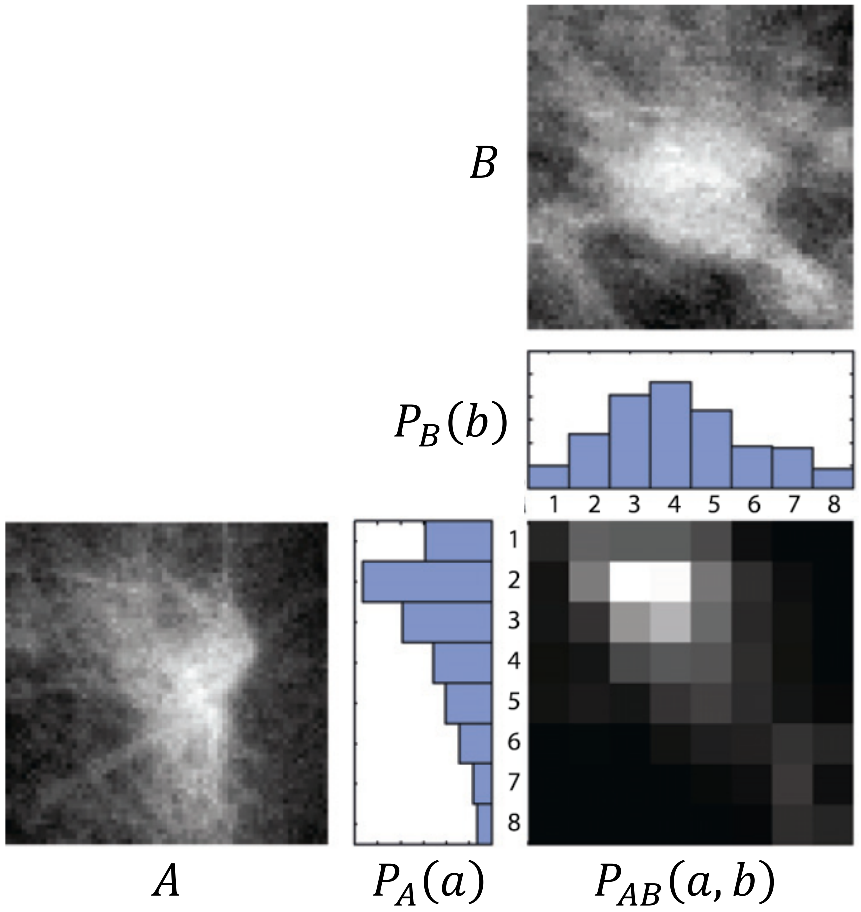
- Mutual information (to be maximised)

$$\mathrm{MI}(A,B) = \sum_a \sum_b P_{AB}(a,b) \log_2\left(\frac{P_{AB}(a,b)}{P_A(a)P_B(b)}\right)$$

Subimages to compare: $A \in I_t$ and $B \in I_{t+\Delta t}$

Intensity probabilities: $P_A(a)$ and $P_B(b)$

Joint intensity probability: $P_{AB}(a,b)$



Dense motion estimation

Dense motion estimation assumptions

- Properties of the light sources do not vary over time interval Δt
- Distance of object to camera does not vary over this time interval
- Visual object appearance does not change over this time interval
- Any small neighbourhood $N_t(x, y)$ shifts over some vector $v = (\Delta x, \Delta y)$

  $\Rightarrow N_t(x, y) = N_{t+\Delta t}(x + \Delta x, y + \Delta y)$

These assumptions may not hold tight in reality but nevertheless they provide useful computational dense motion estimation methods and approximations

Spatiotemporal gradient

- Taylor series expansion of a function

$$f(x + \Delta x) = f(x) + \frac{\partial f}{\partial x}\Delta x + \text{h.o.t. (higher order terms)}$$

$$\Rightarrow f(x + \Delta x) \approx f(x) + \frac{\partial f}{\partial x}\Delta x$$

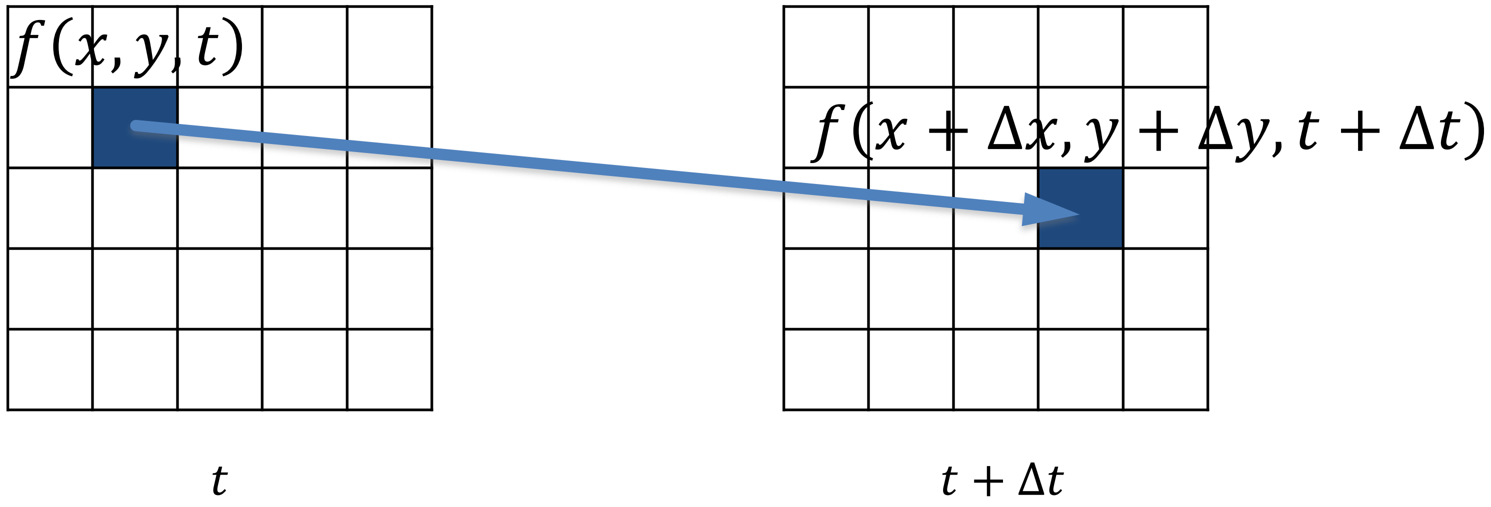
- Multivariable Taylor series approximation

$$f(x + \Delta x, y + \Delta y, t + \Delta t) \approx f(x, y, t) + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \frac{\partial f}{\partial t}\Delta t \qquad (1)$$

Optical flow equation

- Using the dense motion estimation assumptions leads to

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) \qquad (2)$$



Optical flow computation

- Combining equations (1) and (2) yields the following constraint

$$\frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \frac{\partial f}{\partial t}\Delta t = 0 \quad \Rightarrow \quad \frac{\partial f}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial f}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial f}{\partial t}\frac{\Delta t}{\Delta t} = 0$$

$$\Rightarrow \quad \frac{\partial f}{\partial x}v_x + \frac{\partial f}{\partial y}v_y = -\frac{\partial f}{\partial t} \quad \Rightarrow \quad \boxed{\nabla f \cdot v = -f_t}$$

Velocity or optical flow: $v = (v_x, v_y) = \left(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t}\right)$

Spatial image gradient: $\nabla f = (f_x, f_y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$

Temporal image derivative: $f_t = \frac{\partial f}{\partial t}$

Optical flow computation

- The optical flow constraint equation can be applied at every pixel position
- However, it is only one equation, while we have two unknowns (v_x and v_y)
- Thus, it does not have a unique solution, and further constraints are required
- For example, assume a group of adjacent pixels have the same velocity

Optical flow computation

Example: Lucas-Kanade approach to optical flow

Assume the optical flow equation holds for all n pixels p_i in a neighbourhood

$$\left.\begin{aligned} f_x(p_1)v_x + f_y(p_1)v_y &= -f_t(p_1) \\ f_x(p_2)v_x + f_y(p_2)v_y &= -f_t(p_2) \\ \vdots \qquad\qquad \vdots \qquad &\qquad\quad \vdots \\ f_x(p_n)v_x + f_y(p_n)v_y &= -f_t(p_n) \end{aligned}\right\} \quad \boxed{Av = b}$$

$$A = \begin{bmatrix} f_x(p_1) & f_y(p_1) \\ f_x(p_2) & f_y(p_2) \\ \vdots & \vdots \\ f_x(p_n) & f_y(p_n) \end{bmatrix}$$

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$b = \begin{bmatrix} -f_t(p_1) \\ -f_t(p_2) \\ \vdots \\ -f_t(p_n) \end{bmatrix}$$

Least-squares solution

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

Optical flow example

https://www.youtube.com/watch?v=GIUDAZLfYhY

Further reading on discussed topics

- Chapter 8 and 9 of Szeliski 2022
- Chapter 9 of Shapiro and Stockman 2001

Acknowledgement

- Some images drawn from the above references

Example exam question

Which one of the following statements about motion analysis is incorrect?

A. Detection of moving objects by subtraction of successive images in a video works best if the background is constant.

B. Sparse motion estimation in a video can be done by template matching and minimising the mutual information measure.

C. Dense motion estimation using optical flow assumes that each small neighbourhood remains constant over time.

D. Optical flow provides an equation for each pixel but requires further constraints to solve the equation uniquely.