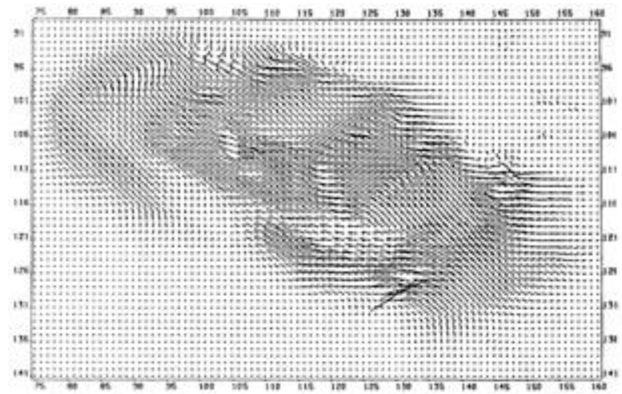


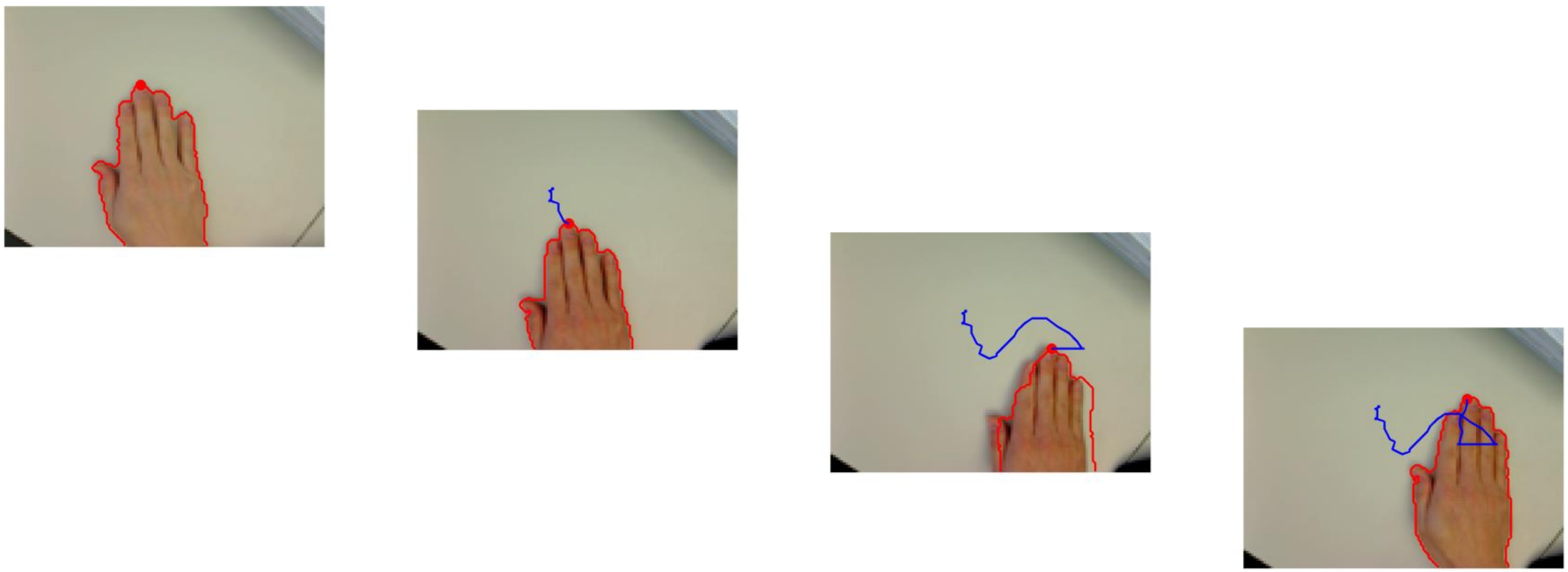
COMP 9517 Computer Vision

Motion

Yang Wang, Data61-CSIRO
yang.wang@data61.csiro.au

Introduction

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



Introduction

- Changes in an image sequence provide features for
  - detecting objects that are moving
  - computing their trajectories
  - computing the motion of the viewer in the world
  - recognising objects based on their behaviours
  - detecting and recognising activities

Applications

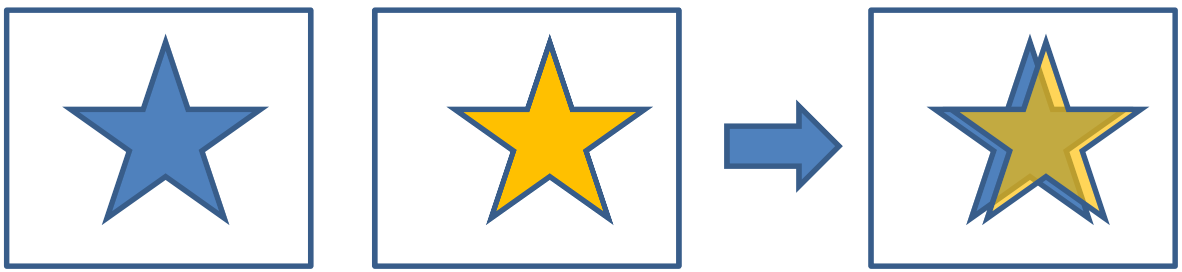
- **Motion-based recognition**
  - human identification based on gait, automatic object detection
- **Automated surveillance**
  - monitoring a scene to detect suspicious activities or unlikely events
- **Video indexing**
  - automatic annotation and retrieval of videos in multimedia databases
- **Human-computer interaction**
  - gesture recognition, eye gaze tracking for data input to computers
- **Traffic monitoring**
  - real-time gathering of traffic statistics to direct traffic flow
- **Vehicle navigation**
  - video-based path planning and obstacle avoidance capabilities

Motion Phenomena

- Still camera, single moving object, constant background
- Still camera, several moving objects, constant background
- Moving camera, relatively constant scene
- Moving camera, several moving objects

Image Subtraction

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



- By subtracting the image I_t from the previous image I_{t-1}, there edges should be evident as the only pixels significantly different from zero

Image Subtraction

- Change Detect
  - Input: images I_t and $I_{t-\Delta}$ (or a model image)
  - Input: an intensity threshold τ
  - Output: a binary image I_{out}
  - Output: a set of bounding boxes B

1. For all pixels [r, c] in the input images,
   set $I_{out}[r, c] = 1$ if $(|It[r, c] - I_{t-\Delta}[r, c]| > \tau)$
   set $I_{out}[r, c] = 0$ otherwise
2. Perform connected components extraction on I_{out}
3. Remove small regions 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 $I_{out}[r, c]$ and the bounding boxes B of regions of changed pixels

Image Subtraction

- The steps:
  - Derive a background image from a set of video frames at the beginning of the video sequence

Image Subtraction

- The steps:
  - The background image is then subtracted from each subsequent frame to create a difference image

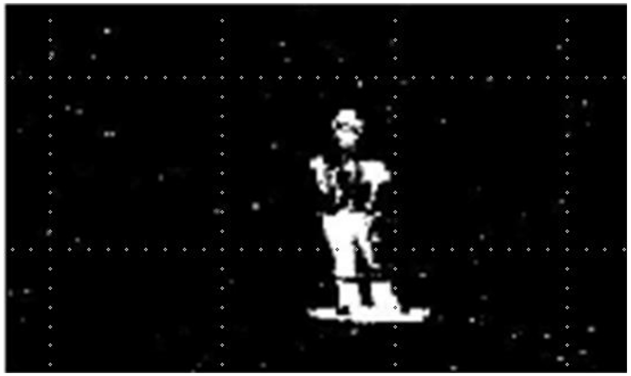
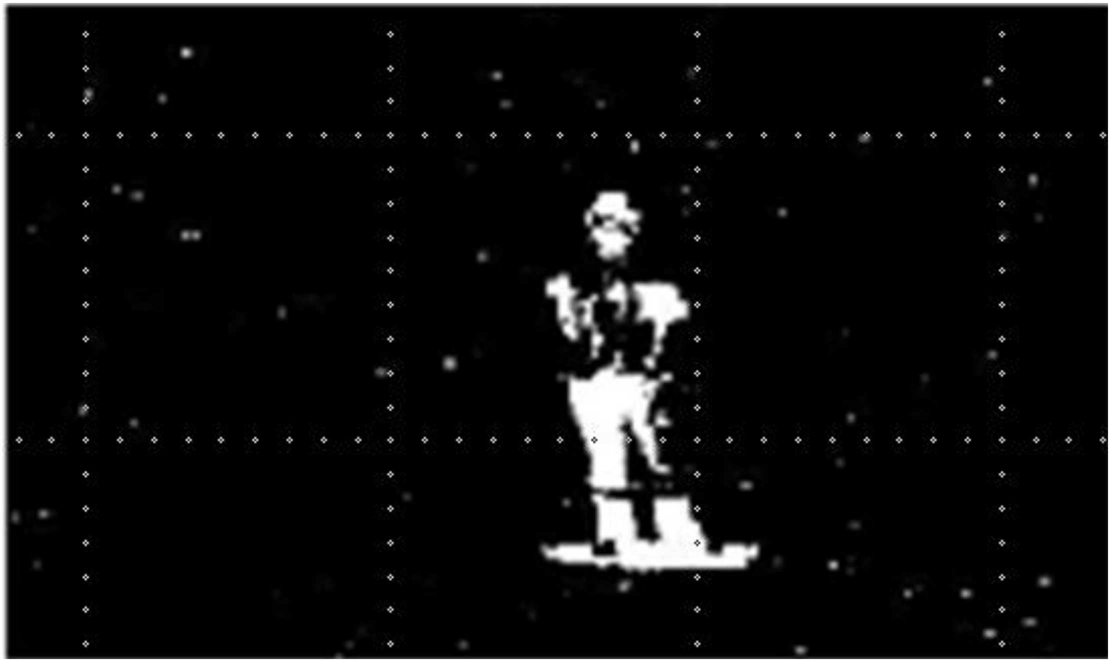


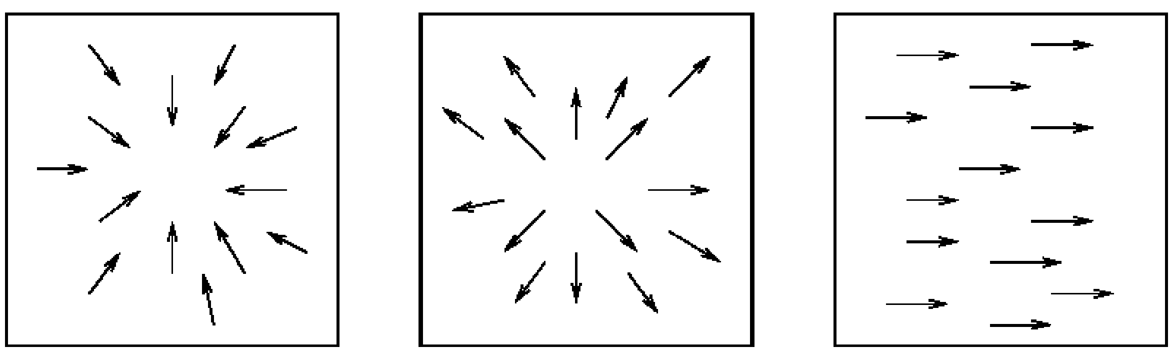
Image Subtraction

- The steps:
  – Enhance the difference image to fuse neighbouring regions and remove noise



Motion Vector

- Motion field: a 2-D array of 2-D vectors representing the motion of 3-D scene points
- The motion vector in the image represents the displacements of the images of moving 3-D points
  - Tail at time t and head at time t+Δ
  - Instaneous velocity estimate at time t



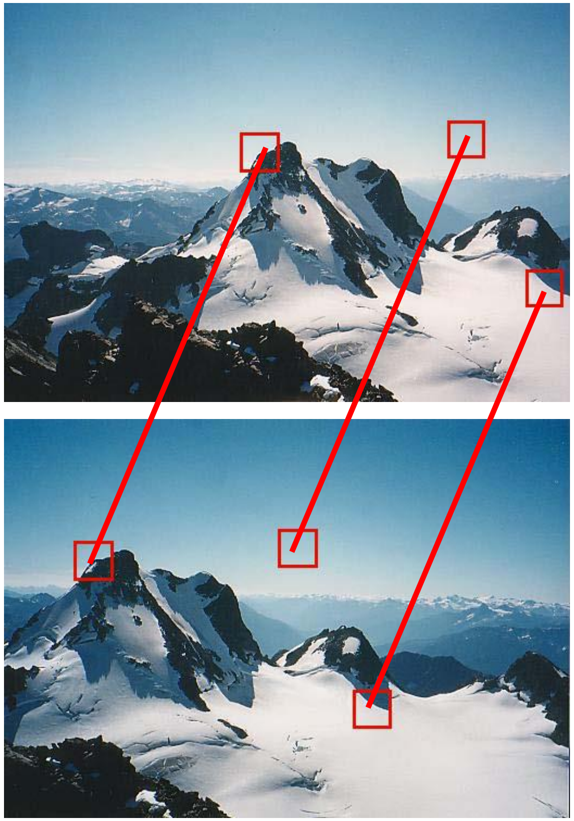
Zoom out Zoom in Pan Right to Left

Motion Vector

- Two assumption:
  - The intensity of a 3-D scene point and that of its neighbours remain nearly constant during the time interval
  - The intensity differences observed along the images of the edges of objects are nearly constant during the time interval
- Image flow: the motion field computed under the assumption that image intensity near corresponding points is relatively constant
- Two methods for computing image flow:
  - Sparse: point correspondence-based method
  - Dense: spatial & temporal gradient-based method

Point Correspondence-based Method

- A sparse motion field can be computed by identifying pairs of points that correspond in two images taken at times t_i and $t_i+\Delta$
- Two steps:
  - Detect interesting points
    - Corner points
    - Centroids of persistent moving regions
  - Search corresponding points



Point Correspondence-based Method

- Detect interesting points:
  - Detect corner points
    - Kirsch edge operator
    - Frie-Chen ripple operator
  - Interest operator
    - Computes intensity variance in the vertical, horizontal and diagonal directions
    - Interest point if the minimum of these four variances exceeds a threshold

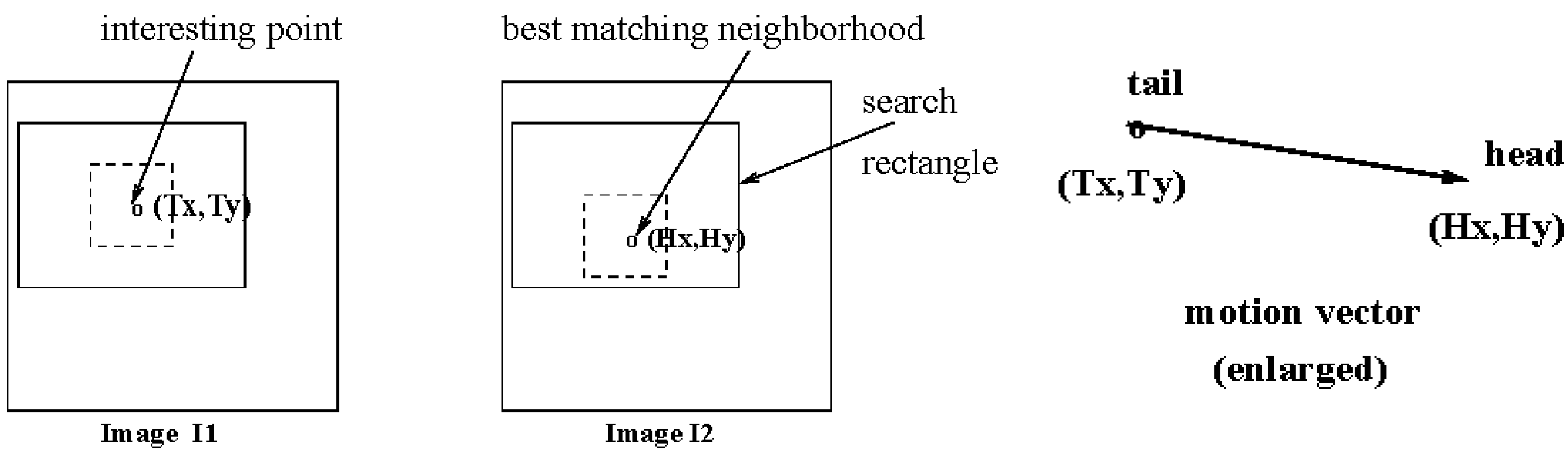
Point Correspondence-based Method

- Finding interesting points of a given input image

```
Procedure detect_corner_points(I,V){
    for (r = 0 to MaxRow – 1)
      for (c = 0 to MaxCol – 1)
                if (I[r,c] is a border pixel) break;
                elseif  (interest_operator(I,r,c,w)>=t) add [(r,c),(r,c)] to set V;
}
Procedure interest_opertator (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 mini(v1, v2, v3, v4);
}
```

Point Correspondence-based Method

- Search corresponding points:
  – Given an interesting point P_j from I_1, we take its neighbourhood in I_1 and find the best correlating neighbourhood in I_2 under the assumption that the amount of movement is limited



Spatial & Temporal Gradient-based Method

- Assumption
  - The object reflectivity and the illumination of the object do not change during the time interval
  - The distance of the object from the camera or light sources do not vary significantly over this interval
  - Each small intensity neighbourhood $N_{x,y}$ at time t_1 is observed in some shifted position $N_{x+\Delta x,y+\Delta y}$ at time t_2
- These assumption may be not hold tight in real case, but provides useful computation and approximation

Spatial & Temporal Gradient-based Method

- Optical flow equation
  - Taylor series :

$$f(x+\Delta x) = f(x) + \frac{\partial f}{\partial x}\Delta x + h.o.t \Rightarrow f(x+\Delta x) \approx f(x) + \frac{\partial f}{\partial x}\Delta x$$

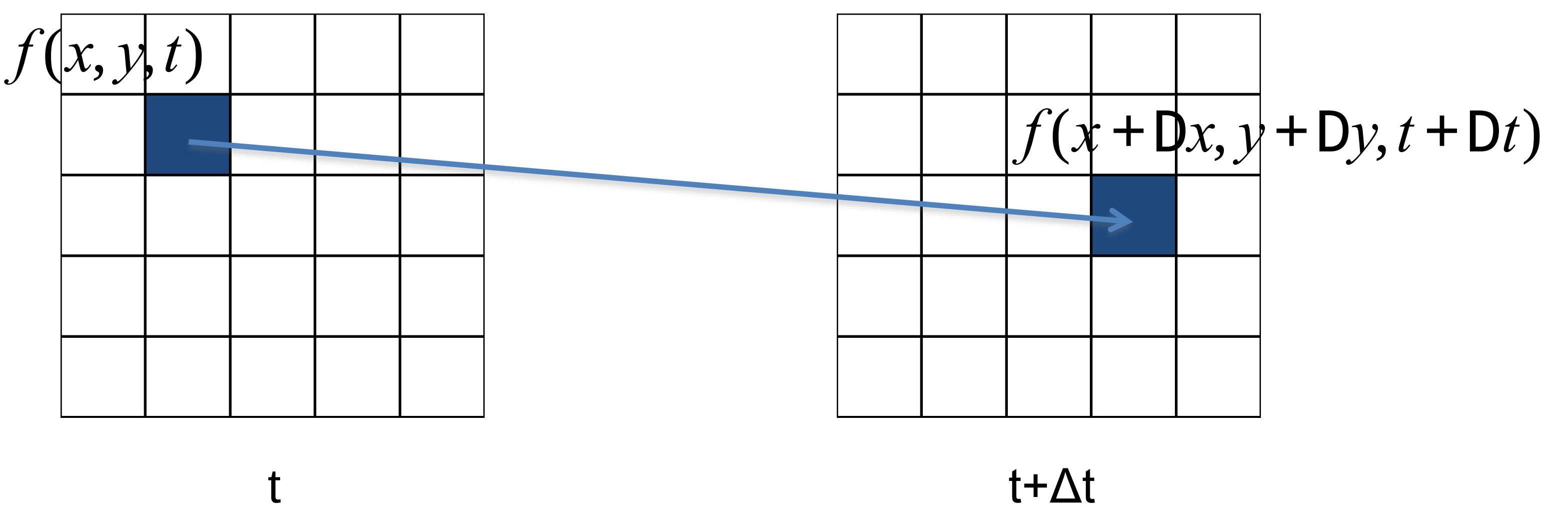
  - Multivariable version:

$$f(x+\Delta x, y+\Delta y, t+\Delta t) = f(x,y,t) + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \frac{\partial f}{\partial t}\Delta t + h.o.t. \qquad (1)$$

Spatial & Temporal Gradient-based Method

- Optical flow equation
  - Assuming the optical flow vector V=[Δx, Δy] carries the intensity neighbourhood $N_1(x, y)$ at time t_1 to an identical intensity neighbourhood $N_2(x+\Delta x, y+\Delta y)$ at time t_2 leads to

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



Spatial & Temporal Gradient-based Method

- Optical flow equation
  - By combining (1) and (2) and ignoring the high order term, a linear constraint can be developed:

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

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

$$\frac{\partial f}{\partial x}V_x+\frac{\partial f}{\partial y}V_y+\frac{\partial f}{\partial t}=0\Rightarrow$$

$$\nabla f\cdot\vec{V}=-f_t$$

  - $V=(V_x,V_y)$ is the velocity or ***optical flow*** of $f(x,y,t)$

Spatial & Temporal Gradient-based Method

- The optical flow equation provides a constraint that can be applied at every pixel position
- However, the optical flow does not give unique solution and thus further constrains are required
  - Example: using the optical flow equation for a group of adjacent pixels and assuming that all of them have the same velocity, the optical flow computation task is reduced to solving a linear system using the least square method

References and Acknowledgements

- Shapiro and Stockman 2001
- Chapter 19 Forsyth and Ponce 2003
- Chapter 5 Szeliski 2010

- Images drawn from the above references unless otherwise mentioned