OXFORD BROOKES UNIVERSITY

Lecture Outline

1. Videos and motion
2. Differencing
3. Background subtraction
4. Optical flow
5. Applications of optical flow
6. Tracking
7. Active Shape Models
8. Structure from Motion

Importance

- Perceiving, understanding and predicting motion is an important part of our daily lives
- We want our machines to be able to do the same

OXFORD BROOKES UNIVERSITY

Videos over images

- Scene is continuously changing in a dynamic environment
- Images are not able to capture these dynamics
- In videos, we have sequence of frames
- By modeling the motion or variation between two consecutive frames, we can infer a lot of new information
- There are different type of approaches:
  - Background subtraction
  - Optical flow
  - Particle filter
  - Kalman filter etc.

Human and motion

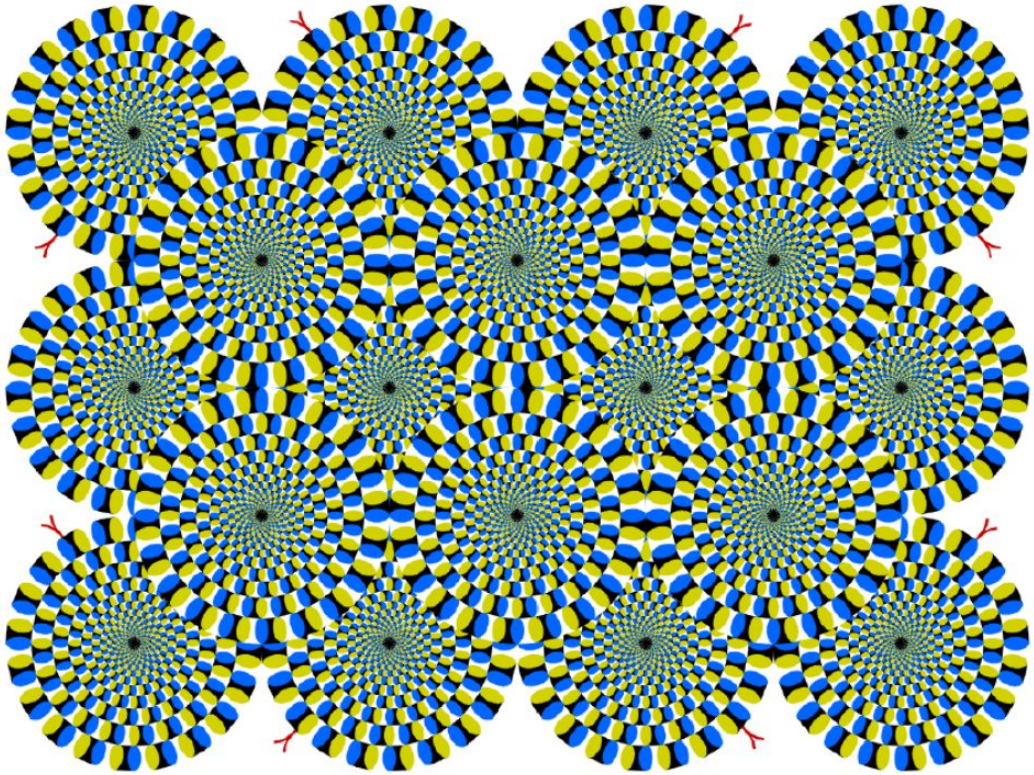
- Human are very good at analyzing motion
- Motion is the first thing we attend to in the visual scene
- This has been very useful in human evolution and survival in past.
- Hence, human brain is hardwired to perceive motion, which lead to some optical illusions.

OXFORD BROOKES UNIVERSITY

Motion from a static picture!

Focus on the image for a couple of seconds and you will see!

This is called “rotating snakes”



The causes of motion

- Changes in the scene can be observed because of:
  - moving camera
  - moving objects
  - varying lighting conditions
- All these factors contribute to the perceived motion in the image plane and can generate ambiguity about cause of the motion
- Ideally we would want to reverse-engineer the structure and dynamics of the scene (geometry, reflectance, lighting, camera egomotion and parameters) -> intractable!
- In practice, we make assumptions on reflectance of surfaces, motion(slow, smooth), lighting (fixed or slow changing), etc

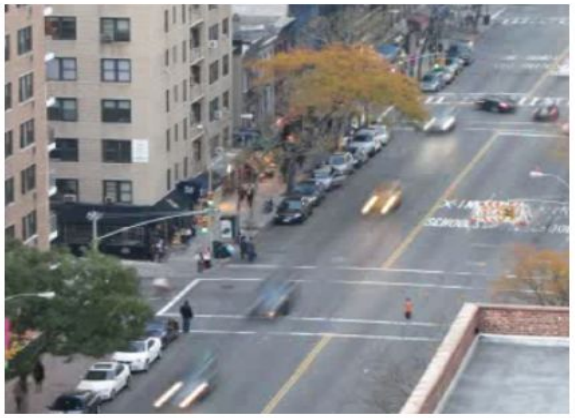
OXFORD BROOKES UNIVERSITY

Four scenarios for motion estimation

static camera, moving scene

moving camera, static scene

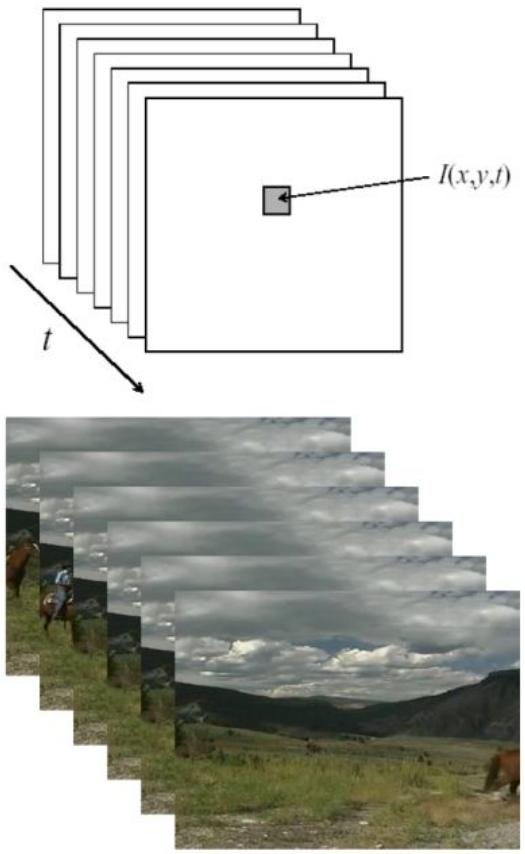
moving camera AND scene



static camera, moving scene AND lights

OXFORD BROOKES UNIVERSITY

Video and motion



- A video is sequence of images
- Hence it is a spatio-temporal volume
- Each point in volume (pixel) can be represented as I(x,y,t)
- x,y are spatial dimensions and t is temporal dimension
- time between two consecutive frames: frame interval
- frame rate: inverse of the frame interval

Differencing

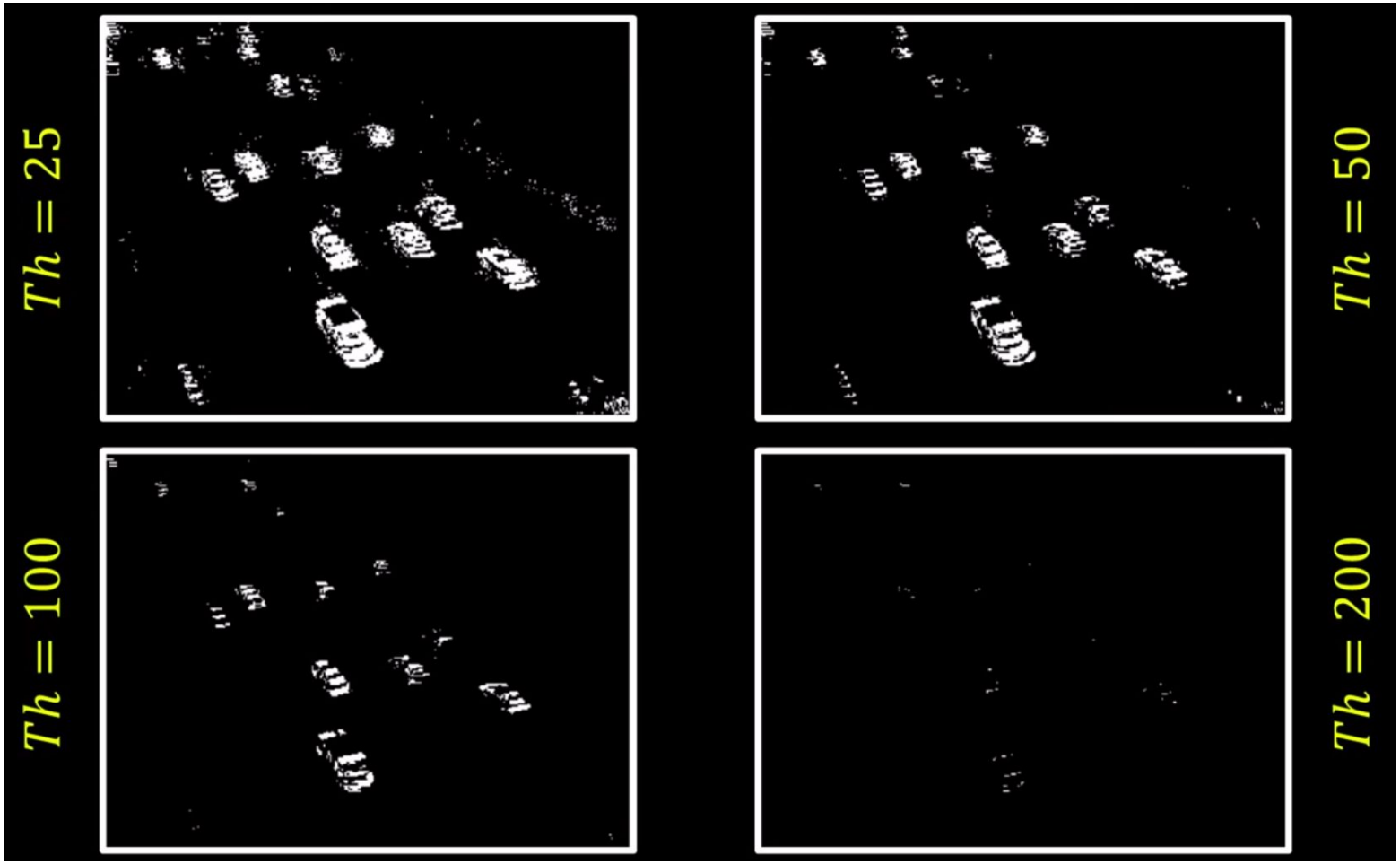
- Differencing uses motion in frames to estimate the foreground and the background
- Useful in segmenting out object where objects of interest are in motion

Differencing

- If any pixel in current frame of video is represented by I(x,y,t)
- Then same pixel in previous frame will be: I(x,y,t-1)
- In differencing, we subtract previous frame from current frame and take the magnitude. Then threshold that magnitude matrix.
  - Difference matrix= | I(x,y,t) - I(x,y,t-1)|
  - Apply thresholding

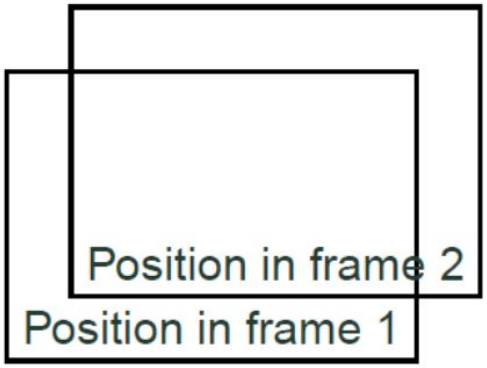
OXFORD BROOKES UNIVERSITY

Results of Differencing



Issues with differencing

- Grey levels differ in locations where the vehicle was in the last image and where it is in the current image
- as a result, the difference image contains a double view of each vehicle
- in the example, the vehicles looks bigger than it actually is
- Secondly, if object are big and have uniform surface then differencing will not give value in the middle of surface.



Background subtraction

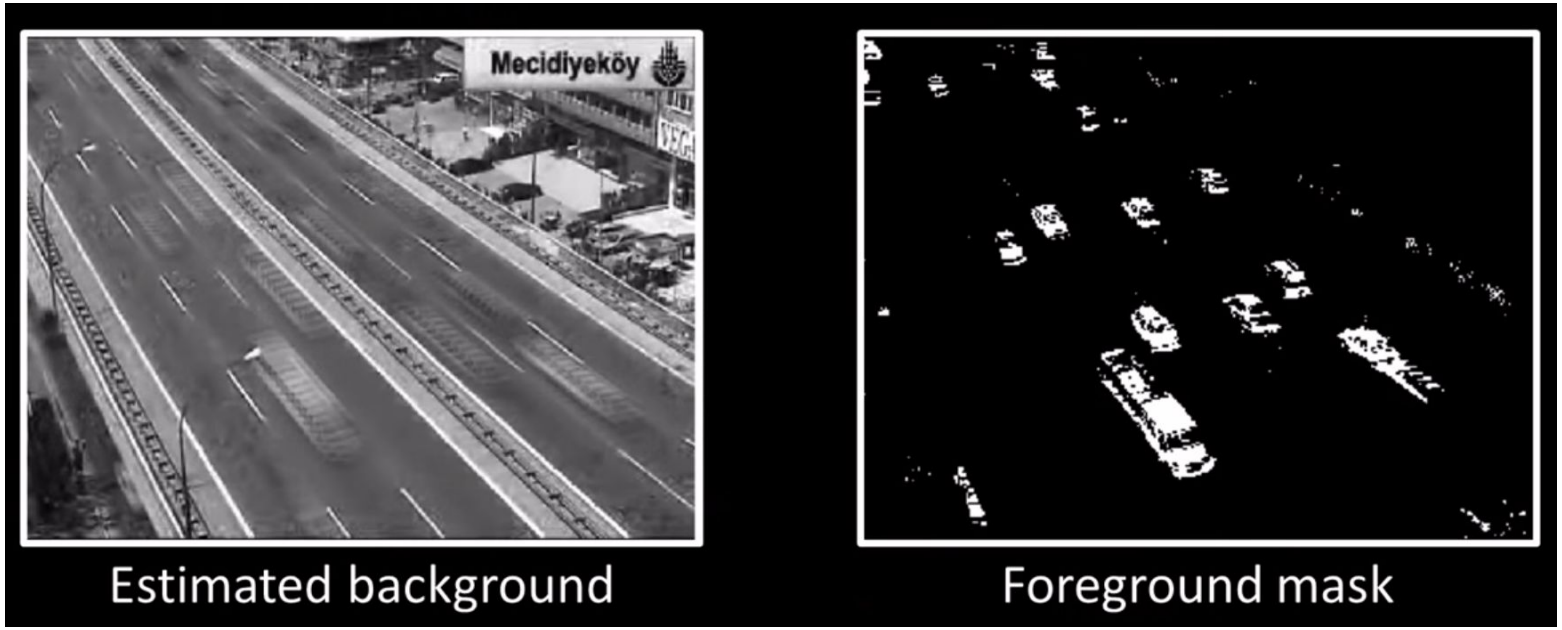
- Compares an observed image with a ‘reference’ image,
- Reference image: an estimate of what the image would look like if it contained no objects of interest
- create reference image (stationary background) by averaging a long sequence
- a significant difference between observed and reference images indicates the location of the object(s) of interest, after thresholding.

Background subtraction algorithm

- create an image of the stationary background by averaging a long sequence
- compute the difference with the background image
- apply appropriate threshold

Background subtraction

- the sequence now looks cleaner, though there is still some noise
- there are no ‘double views’



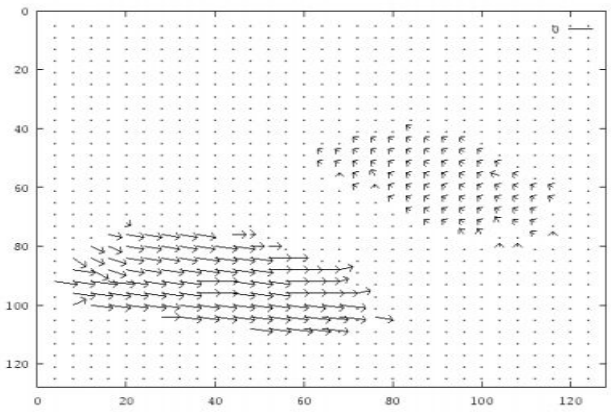
Background subtraction- Issues and challenges

- motion detection of this kind works only if camera is stationary and objects move against fixed background
- background subtraction requires acquiring a good reference image
- A good reference image can require at least 1000 frames for averaging
- this is sensitive to illumination conditions (night vs day or cloudy day versus sunny day), shadows, amount of other objects Moving etc.

OXFORD BROOKES UNIVERSITY

Optical Flow

- Optical flow is represented in term of motion vector (u,v)
- Which is image displacement in x and y directions between two consecutive frames

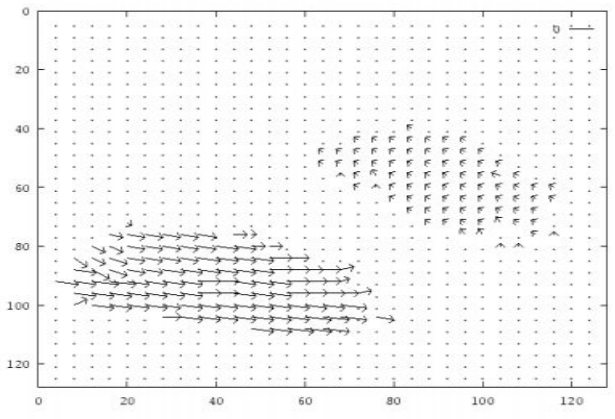


Optical flow computation

given intensity values of the image sequence I(x, y, t) we want:

$$\mathbf{u}(x,y) = [u(x,y), v(x,y)], \qquad u = \frac{dx}{dt}, v = \frac{dy}{dt}$$

field of arrows, each proportional to the amount of pixel motion



Taylor Series

A Taylor series is a series expansion of a function about a point.

A one dimensional Taylor series is an expansion of a real function f(x) about a point x=a is given by:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots.$$

Optical flow

- Brightness constancy assumption:
  - image intensity in a small region remains the same, the region's location may change
  - in other words, only location of pixel changes with time but the brightness of each scene point remains same.
  - So the next time frame can be represented as:

$$f(x, y, t) = f(x + dx, y + dy, t + dt)$$

- The equation can be represented in term of taylor series as:

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

Optical flow

- If we consider motion to be slow and smooth in the previous equation.
- Then we can equate derivative to zero as change is x and y are negligible.

$$0 = f_x dx + f_y dy + f_t dt$$

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

$$0 = f_x u + f_y v + f_t$$

OXFORD BROOKES UNIVERSITY

Derivative Masks (Roberts)

Derivative masks for:

x direction

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{first image}$$

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{second image}$$

f_x

y direction

$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{first image}$$

$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{second image}$$

f_y

t direction

$$\begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \text{first image}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \text{second image}$$

f_t

Apply masks on first and second image and add to get the value of f_x, f_y and f_t

Horn & Schunck approach

$$\iint \{(f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\} dx dy$$

Brightness constancy

Smoothness constraint

Using variational calculus, we can rewrite above equation as:

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

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

$$\Delta^2 u = u_{xx} + u_{yy}$$

Horn & Schunck approach

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

Using variational calculus

$$u = u_{av} - f_x \frac{P}{D}$$
$$v = v_{av} - f_y \frac{P}{D}$$

Rewriting the above equation

$$(f_x u + f_y v + f_t) f_x + \lambda(u - u_{av}) = 0$$
$$(f_x u + f_y v + f_t) f_y + \lambda((v - v_{av}) = 0$$

$$P = f_x u_{av} + f_y v_{av} + f_t$$
$$D = \lambda + f_x^2 + f_y^2$$

Algorithm to calculate optical flow

- k=0
- Initialize $u^K \quad v^K$
- Repeat until some error measure is satisfied (converges)

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

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

$$P = f_x u_{av} + f_y v_{av} + f_t$$

$$D = \lambda + f_x^2 + f_y^2$$

OXFORD BROOKES UNIVERSITY

Example of optical flow

https://www.youtube.com/watch?v=ysGM3CfBVpU

https://www.youtube.com/watch?v=LjjJQ81RbX0

Lucas & Kanade Method

The standard optical flow equation is: $f_x u + f_y v = -f_t$

Now if we consider the assumption that the neighbourhood of a pixel have same optical flow; then, 3x3 neighbourhood can be represented as:

$$f_{x1} u + f_{y1} v = -f_{t1}$$

$$\vdots$$

$$f_{x9} u + f_{y9} v = -f_{t9}$$

Lucas & Kanade Method

Another way to solve the optical flow equation is to use least square fit.

Assumption remain the same that neighbourhood has same optical flow vectors.

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

$$\frac{\partial}{\partial u}\sum_i (f_{xi}u + f_{yi}v + f_t)^2 = 0 \qquad \sum (f_{xi}u + f_{yi}v + f_{ti}) f_{xi} = 0$$

$$\frac{\partial}{\partial v}\sum_i (f_{xi}u + f_{yi}v + f_t)^2 = 0 \qquad \sum (f_{xi}u + f_{yi}v + f_{ti}) f_{yi} = 0$$

Lucas & Kanade Method

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

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

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

$$\begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

OXFORD BROOKES UNIVERSITY

Lucas & Kanade Method

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2} \begin{bmatrix} \sum f_{yi}^2 & -\sum f_{xi} f_{yi} \\ -\sum f_{xi} f_{yi} & \sum f_{xi}^2 \end{bmatrix} \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

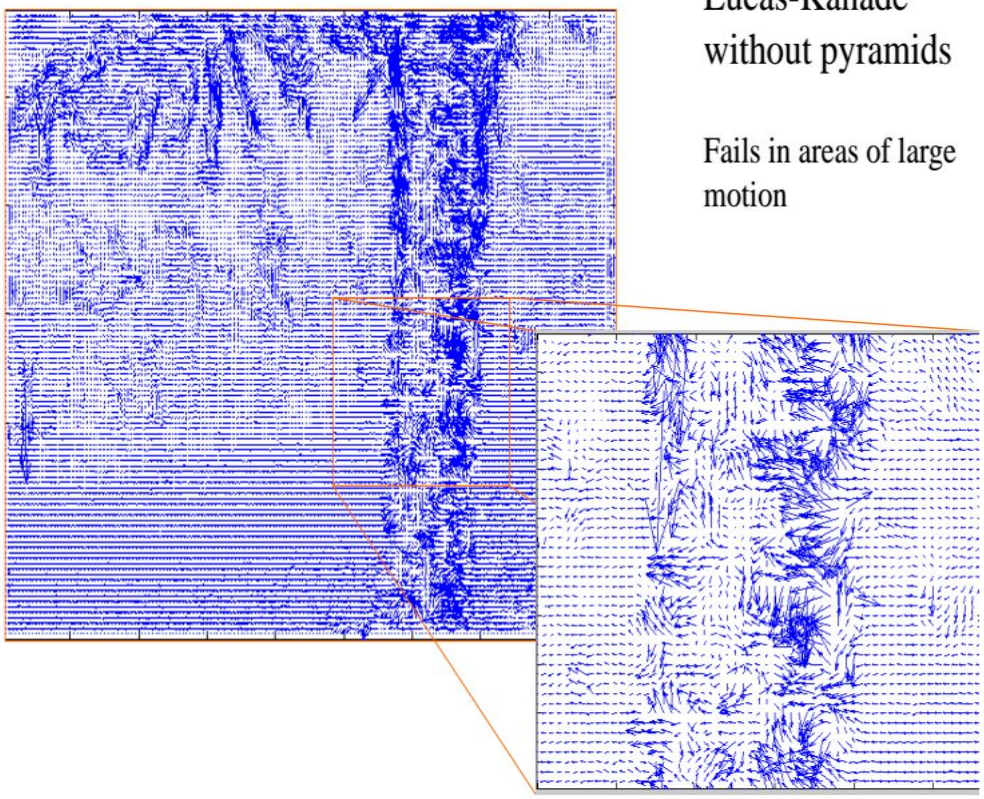
OXFORD BROOKES UNIVERSITY

Lucas & Kanade Method

$$u = \frac{-\sum f_{yi}^2 \sum f_{xi} f_{ti} + \sum f_{xi} f_{yi} \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$$

$$v = \frac{\sum f_{xi} f_{ti} \sum f_{xi} f_{yi} - \sum f_{xi}^2 \sum f_{yi} f_{ti}}{\sum f_{xi}^2 \sum f_{yi}^2 - (\sum f_{xi} f_{yi})^2}$$

Demo



Lucas-Kanade
without pyramids

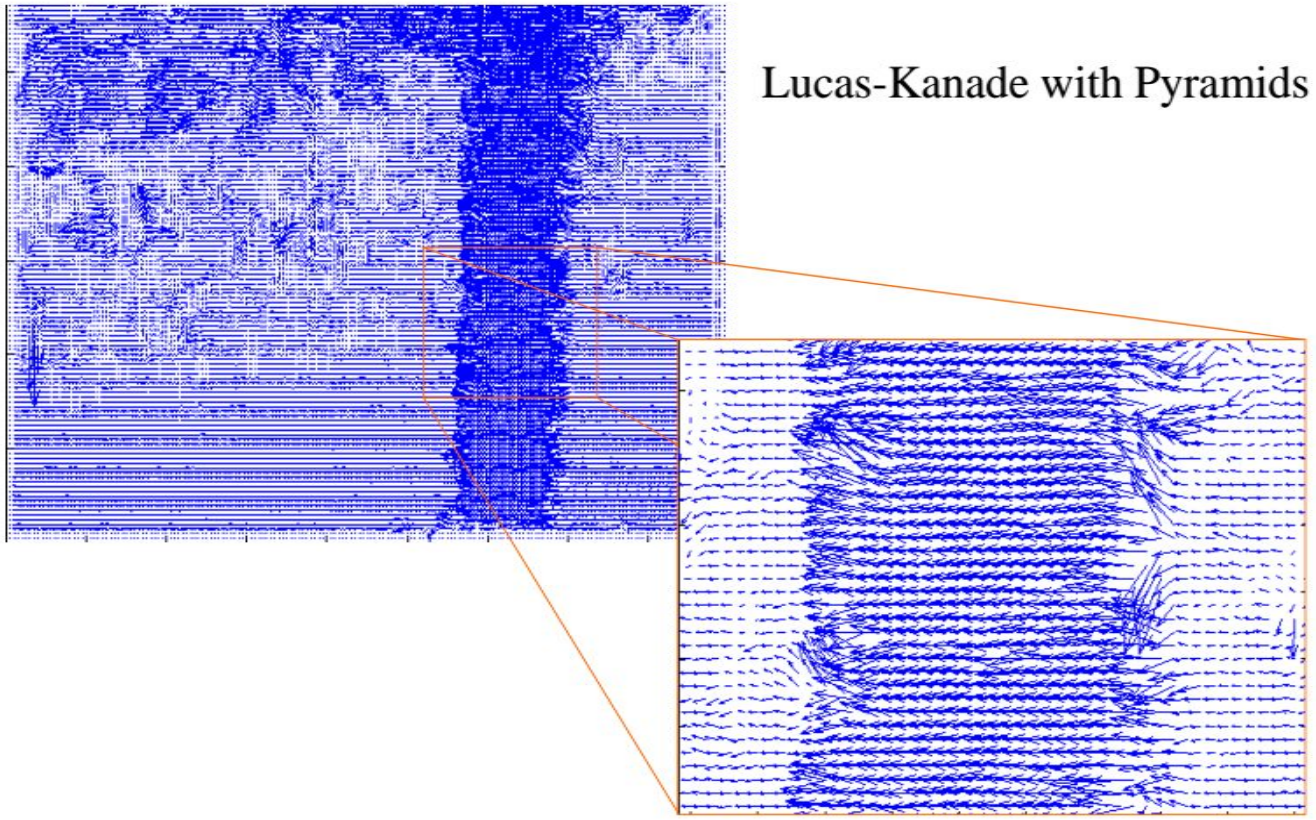
Fails in areas of large
motion

Summary of optical flow

- Horn-Schunck and Lucas-Kanade optical methods work only for small motion.
- If object moves faster, the brightness changes rapidly,
  - then 2x2 or 3x3 masks fail to estimate spatiotemporal derivatives.
  - Optical flow vector are not aligned with each other
- Pyramids can be used to compute large optical flow vectors.

OXFORD BROOKES UNIVERSITY

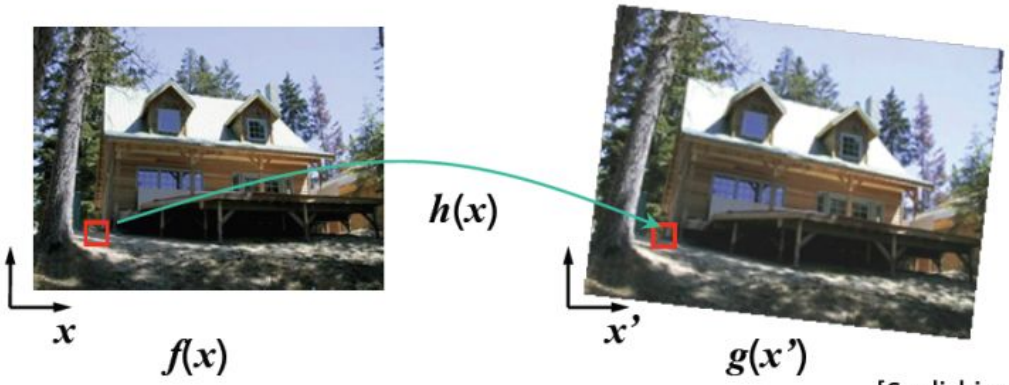
Demo



Applications of optical flow

Image registration

- we can use optical flow to register (i.e. align) images, by:
  - computing the flow with the entire image as a single region;
  - shifting the second image toward the first;
  - iterate until convergence.

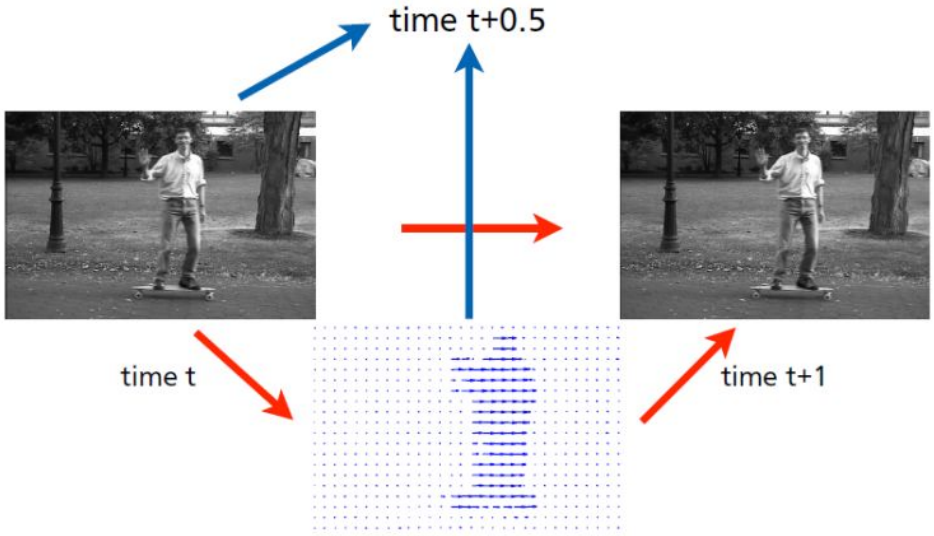


[Szeliski and Fleet]

Applications of optical flow

Video interpolation

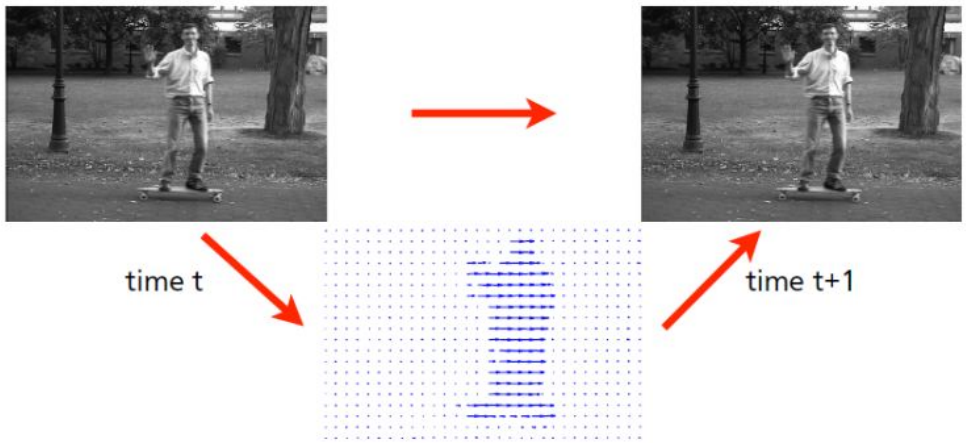
if we know the image motion, we can compute images at intermediate time steps: video interpolation



Applications of optical flow

Video compression

- if we wish to compress an image sequence, we can predict new frames using the flow, and only store how to “fix” this prediction
- the flow field is relatively easy to compress



OXFORD BROOKES UNIVERSITY

Tracking

- Tracking: following the position (and possibly its internal configuration) in time of an object of interest
- can be a single interesting point: **feature tracking**
- can be a target (e.g. airplane in airspace) -> **target tracking**
- can be a contour in 2D -> **active contouring**
- can be an object, intended as a region in an image sequence -> **object tracking**
- can be an articulated or deformable body in space -> pose estimation, motion capture

https://www.youtube.com/watch?v=RtwmKsQj9dM

Feature tracking

- feature tracking: track point features from frame to frame (most often used are corner features)
- finding matching image pixels p1 and p0 in frames I(t) and I(t + 1), corresponding to the same physical point
- features may appear and disappear, change with viewpoint

Simple KLT Algorithm

- Detect Harris corners in the first frame
- For each Harris corner compute motion (translation or affine) between consecutive frames.
- Link motion vectors in successive frames to get a track for each Harris point
- Introduce new Harris points by applying Harris detector at every n (10 or 15) frames
- Track new and old Harris points using steps 1-3

OXFORD BROOKES UNIVERSITY

Active contours for segmentation and tracking

- active contours is a catch-all name for methods that find the curve that best separates objects in an image
- This can also be called as segmentation
- But segmentation identifies all the pixel related to object
- Instead, we only track the shift in contour around the moving object
- can be edge-based or region based, i.e., based on separating regions with different statistics in some feature space

OXFORD BROOKES UNIVERSITY

Active contours demos

Demo on how active contours works:
https://www.youtube.com/watch?v=r610mi5hiHM

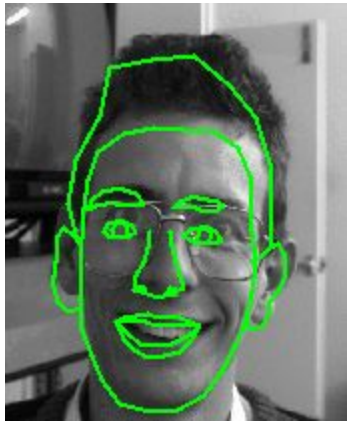
How to implement it in Matlab:
https://www.youtube.com/watch?v=v8-tyVXmUBo

OXFORD BROOKES UNIVERSITY

Active Shape Models

It provides an iterative method of matching a model to an image

Initial Pose



After 5 iterations

Convergence

OXFORD BROOKES UNIVERSITY

Active Shape Models

- building ASMs required labelling training images, with landmarks which represent correspondences
- every shape x is a cloud of points
- We identify the landmarks in current image and the next frame
- We create tracking vectors for each of the landmarks using tracking
- These tracking vectors can be used to define the shape model

Structure from motion

- Structure from Motion (SfM): technique utilizes a series of 2-dimensional images to reconstruct the 3-dimensional structure of a scene or object.
- SfM can produce point cloud based 3-D models similar to LiDAR.
  - given a video of an unknown scene captured by a moving camera whose motion is unknown
  - we want to reconstruct: camera motion and 3D scene geometry
- we assume that camera internal parameters are calibrated
- extrinsic parameters (translations and rotation between camera and world coordinates) are not know

Structure from motion

- interest points, typically corners, are tracked to allow for SfM
- why corners? the reason is that, as the geometry is not known, the search space is the entire image
- so, dense feature points would be very expensive to track
- corners are relatively sparse, well localised and cheap to compute
  - find the correspondence between corner points
  - then compute the fundamental matrix F from the correspondences
  - two image coordinates of the same 3D point must satisfy $x_1 \, F \, x_2 = 0$

OXFORD BROOKES UNIVERSITY

Structure from motion

OXFORD BROOKES UNIVERSITY

Structure from motion

A quick demo on how we implement SfM:

https://www.youtube.com/watch?v=i7ierVkXYa8

OXFORD BROOKES UNIVERSITY

Thanks!