

# Charlie's notes

December 3, 2015

## 1 CS31310 - AGILE

### 1.1 MONDAY 23RD & 24TH NOV: FDD PROCESSES & PROJECT MANAGEMENT

**Most of the content portrayed in the slides**

#### 1.1.1 MAIN POINTS:

- Claimed to be better suited to larger, more stable projects
- Requirements are to be fairly well understood
- Doesn't need a constant onsite customer
- Developing an overall model is all about understanding the customer's requirements while you have them onsite
- FDD likes colour-coding things: not accessible for the colour-blind!
  - Coloured UML
  - Coloured Wall chart tracking progress
- Doesn't weight features like in others - everything considered and nothing is 'nice to have'
- Teams are dynamic - breaking apart and regrouping as necessary
- People can specialise in one area and focus on that, rather than being the 'T' shape Neil speaks about
- **Apparently usually quite good for our Major Project**

*Process is well documented in the slides - my notes just copy this*

<https://trello.com/c/FPL0wikh/3-edel-s-fdd-process-lecture> - PDF copy of Edel's Lecture 1

<https://trello.com/c/J2DxyHEg/4-fdd-management-pdf> - PDF copy of Edel's Lecture 2

## 2 CS36110 - MACHINE LEARNING

### 2.1 NOVEMBER 25 & 27: ANNs

## 3 CS34110 - COMPUTER VISION

### 3.1 NOVEMBER 20: MOTION MODELS

#### **Modelling Change & Tracking**

MOTION:

- Background Subtraction
- Optical Flow

MIXTURE OF GAUSSIANS (MoG):

- Robust to noise
- Handles shadows ok
- Common first step

#### 3.1.1 TRACKING: MODELLING CHANGE

VIDEO:

- detections in each frame
- detections are noisy & computationally expensive
- tracking mitigates both issues

Noise can occur if the camera on a robot/car is moving up/down

#### 3.1.2 A GENERAL FRAMEWORK FOR TRACKING

RECURSIVELY:

- An idea about how something will change (*Model*)
- Make a prediction (*Predict*)
- See what happens (*Measure*)
- Update model (*Update*)

#### ADVANTAGES:

- Smooths the data
  - estimate location upon predictions & the measurement
- Constrains search
  - start looking for target in the location it was last seen

#### 3.1.3 KALMAN FILTER

- Like predict, measure, update from earlier
- Useful for tracking
- Copes well with missing information (occlusions)

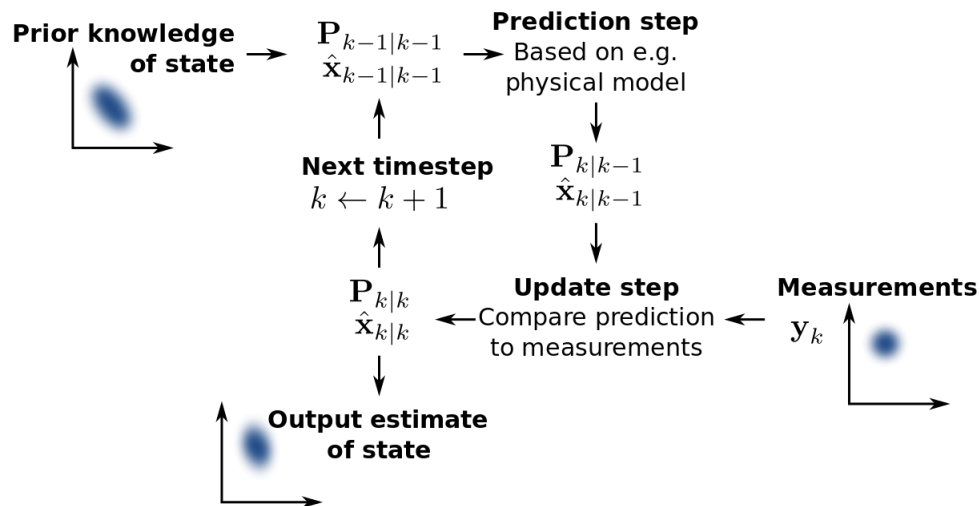


Figure 3.1: Sourced from Wikipedia

Background subtraction	→	Pixels grouped into objects	→	Tracker
Sparse Optical Flow	→	Features grouped into objects	→	
Face Detection	→		→	

#### USE KALMAN TO SMOOTH ANY MEASUREMENT

- X,Y location
- size
- colour

**See also:** Particle filtering: works with combining and splitting objects (e.g. people holding hands, then letting go)

**Hannah's video:** <https://www.youtube.com/watch?v=NYdwpX1a7-Y>

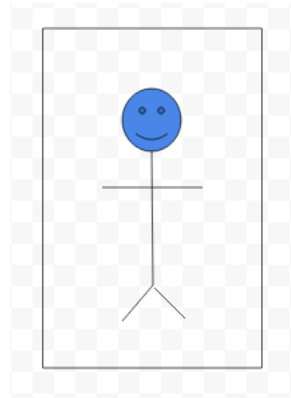
### 3.1.4 MEAN SHIFT

Computer the mean of the data within the window

Shift the window to the mean every time

NOTES:

- Changes size - can use CAM-Shift to mitigate?
- Lighting change - not really, gradually changes mean over time
- If it picks up something you're not looking for, will slowly drift off



### 3.1.5 PROBLEMS WITH TRACKING

- Initialisation (what are you tracking?)
- Having more than 1 item to track
- Losing target due to motion / occlusion
- Losing target due to appearance change

Usually initialise from a detector of some sort

**Useful speed up for detectors & accuracy**

Look into: TLD: Tracking Learning Description

### 3.1.6 CLOSING NOTE

Vision Systems tend to have multiple layers

Tracking is extremely common in anything which deals with change.

HOG FOR EXAMPLE, HAS SEVERAL LAYERS:

- 2D filters
- Tangent
- Histogram
- Superimpose grid
- SVM

*May be more, however couldn't write quick enough... You get the idea...*

## 3.2 NOVEMBER 24: SHAPE FROM SHADING

### 3.2.1 SHAPE

3D STRUCTURE OF:

- Object
- Scenes

WHY?

- Science (surface of Mars)
- Graphical (3D model of heritage)

**There is a table in the slides of the topics that will be covered in the 'Shape' series**

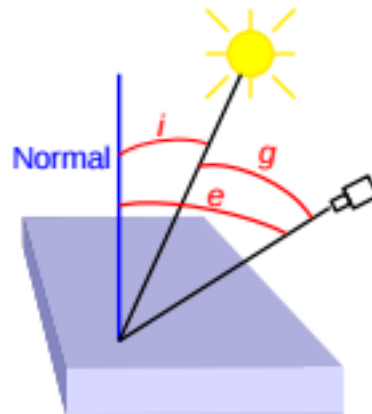
### 3.2.2 SHAPE FROM SHADING

BRIGHTNESS DEPENDS ON:

- where light source is
- where viewer is
- local orientation of surface
- properties of surface (matt vs glossy)

geometry.

- $i$ : incident angle
- $e$ : emittance angle
- $g$ : phase angle



BRDF:

- Bidirectional Reflectance Distribution Function
- fraction of light reflected in the direction of the viewer

- Lambertian:  $\Phi(i, e, g) = \cos i$
- Specular:  $\Phi(i, e, g) = 1$  when  $i = e$  and  $g = i + e$
- Largely property of the surface
- Used in graphics a lot
- Difficult to establish wavelength of light & properties of surface
- Determined experimentally

3D SURFACE INFORMATION **Look up - 'Normal Mapping' wikipedia page**

Orientation of surface described by a normal vector (x,y,z):

- Change in z when x changes (p)
- Change in z when y changes (q)

### 3.2.3 GRADIENT SPACE

$$p = \frac{\delta z}{\delta x}$$

$$q = \frac{\delta z}{\delta y}$$

REFLECTANCE MAP:

- For surface of 'x' orientation, we expect this reflection back

Light Change: 1 = lightest, 0 = darkest

- Global solution found by integration in gradient and image space + smoothness assumption:
  - Maths in confusing!
  - Don't need to know the maths for the exam

### 3.2.4 PHOTOMETRIC STEREO

We can reduce assumptions if we have multiple lighting conditions.

*Example: 2 overlapping reflectance maps on sphere*

- Use same point in image plane under different lighting
- You know where the lights are (calibrated)
- Nothing moves - just the lights on/off
- Calibration with sphere of similar reflectance
- There is always noise
- Reasonably robust

### **This is at the heart of light-stage imaging**

- Illuminate with 1 source at a time
- Calibrate & results in look-up table

Have to reduce inter-reflections!

X,Y,Z MAPPED TO RGB FIGURES:

- gives diagram of 3D structure

Fewer images → More assumptions

★ **Look up ISO-contours**

---

## 3.3 NOVEMBER 27: SHAPE FROM X

### 3.3.1 SHAPE FROM TEXTURE

1 + images, with a lot of assumptions.

MARR 1982:

- 2 textured gradients
- Limited information
- Circles turn into ellipses (illusion of depth)

TYPES OF TEXTURES:

Isotropic: rotation perpendicular to texture plane does not 'change' the texture

Homogeneous: texture looks 'the same' everywhere

Texel: basic texture elements, the repetition of which creates the texture

LOCAL BINARY PATTERNS

- 3x3 filters
- Capture changes in greyscale

0 = darker than T 1 = lighter than T

1	1	0
0	T	1
0	1	0

0	1	1
0	T	1
0	0	1

Figure 3.2: Top: Standard output. Bottom: Could be used to detect edges

#### GLOBAL PLANES

- Find the plane, uphill, downhill etc
- Gradients in texture
- Isotropic texture:
  - circles → ellipses
  - Major axes of ellipses to work out slant & tilt
- Homogeneous texture:
  - textures made of dots
  - look for a density change

Generally you fit methods locally rather than globally

#### 3.3.2 SHAPE FROM MOTION

Look up Marr (again)

Can also do structure from Motion: this is next week!

**The examples on BB are really good: biomotionlab!**

Things further away move more slowly - can infer things from this

ASSUMES:

- rigidity
- parallel projection

Works both long and short range...



- Initial guess on 3D model of world (flat screen)
- + new image
- update on those which disagree with the initial model

**Very popular!**

### 3.3.3 SHAPE FROM OCCLUSIONS

Infer shapes from which bits are in front of other parts

OCCLUSION CONTOURS:

- discontinuity in depth signal
- corresponds to silhouette (shadow)

ASSUMPTIONS (1 IMAGE):

- each point on contour corresponds to a single point on the object (except when a 'T' junction)
- nearby points on the contour correspond to nearby points on the object
- points on contour correspond to planar points on the object

WHEN YOU HAVE 2+ IMAGES:

- Space Carving
- Volumetric reconstruction
- Gives where the object is isn't
- Object is intersection of several generalised cones

**Watch the Youtube video on BB - Greenvision**

FROM A VIDEO SEQUENCE:

- inferred from motion & occlusion
- what occludes people in the scene?

Things at the front = black, as it's never occluded from you  
Lots of motion = white

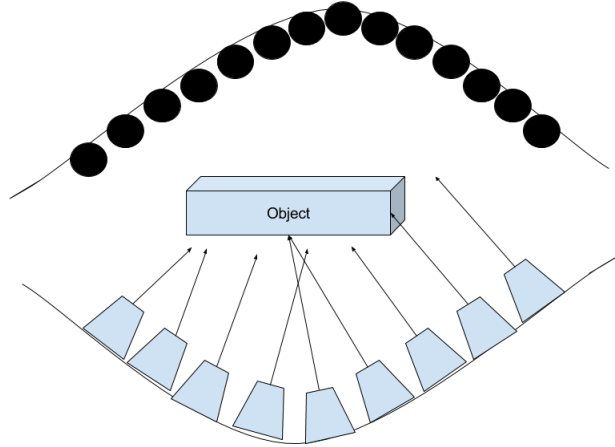


Figure 3.3: Can the cameras see the points on the other side of the object? Make a map of what it can / can't see to build a shape of the object using multiple cameras (bottom row).

#### 3.3.4 SHAPE FROM FOCAL LENGTH

##### *Depth from defocus*

Things are blurry past a depth of focus Take 10 pictures, at different lengths, with different focuses

- Texture really helps focus
- Assumptions about smoothness

Markov-Random Field

HOW TO TELL IF IN FOCUS?

- Look for sharp edges = ✓
- Only low-frequency components = ✗

---

## 4 CS32310 - ADVANCED COMPUTER GRAPHICS

## 5 SE31520 - INTERNET-BASED APPLICATIONS

**Fred Long's lecture slides contain all the content.**

## 6 OTHER