Computer Vision Exam Notes

Nathan Hughes (nah26@aber.ac.uk)

January 18, 2018

1 Edges and Low Level Vision

- Generally are the foundations of features
- Where a sharp change in intensity occurs
- For example using Sobel to get x & y edge intensity
  - Sobel uses a convolution matrix to pass over each pixel in an image
  - Zero crossings of the 2nd derivative are clear edge indicators
  - Sobel works by combining both the horizontal and vertical edge detectors to give robust output
- Filters use similar setups
- Gaussian blur, Sobel, Canny for more examples of filters
  - Allows sharping, blurring, distorting, edge finding etc
- Can use a 1xN or a NxN setup for convolutions
  - Larger can give better local averages than 1D

2 Feature Detection

- As edges are the foundations of features, features are the building blocks of larger systems
- Often use edge grouping as a technique of finding contours
- Can find higher level features such as:
  - Straight lines
  - Curves
  - Blobs
  - Ribbons
- Features can be found using bottom up or top down techniques:
  - Bottom up: Edge pixels are grouped and followed to next edge pixel
  - Top Down: Model is projected and matched to edge pixels
  - Both: Camera noise, complexity and gaps cause issues with both these methods

2.1 What is a Feature?

- Is anything really!
  - Texture
  - Corner
  - Ellipses
  - Projection of rectangles (more complex shapes)
  - Ribbons

2.2 Feature Detection Techniques

2.2.1 Hough

- Hough transform algorithm can be used to find straight lines, as well as circles
- It uses a technique which draws lines through a given pixel and vote on which is the correct path

- It is good at finding geometric features
- Can group widely spread pixels well (Good and bad!)
- Requires a lot of parameter tuning

2.2.2 SHIFT/SURF

- Scale Invariant Feature Transform
- Is invariant to scale, rotation and transformations of input
- Uses Gaussian Pyramids to test for different image scaling factors
- Features found can be used in more complex object Recognition/Detection models
- Considered to be a state-of-the-art system in Computer Vision
- Features are selected if:
  - Contrast is good
  - Pixel is a corner
- Depends on many, more or less, arbitrary parameters
- Expensive to run but has a few tricks to become faster
  - SURF is often considered to be a faster model with less features detected

2.2.3 Harris

- The Harris algorithm is a corner detection method
- It uses surrounding convolutions in order to detect if surrounding profile matches

2.3 RANSAC

- RANSAC can be used to calculate the homography between two images by using two sets of SIFT points.
- This means that if you have a reference image and are presented with a second image, you can test if a the reference image is contained within the second image
- AND you can calculate the transformation.

2.4 What makes a good features?

- Distinctive - Is it unique
- Accurate - Can you accurately find it again and again
- Locality - Is it local to the other features
- Easy to find - Can it be easily found in an image
- Efficiency - Is it expensive to search for

3 Object Detection

- Finding **things** in images can be done by two different approaches:
  1. Making a representation - i.e. choosing, encoding and searching
  2. Looking at the thing - i.e. a change in appearance and looking for it
- Object detection essentially comes down to categorising.

- Can't just straight-up match images to images as they change
  - Scale
  - Rotation
  - Angle
  - Lighting
  - Colour
  - etc
- Convolutions can be used to learn the changes in objects from image->image as they preserve spatial organisation
- Object detection is different from Object Localisation
- Generally a large amount of training data is required for recognition to overcome change in conditions

3.1 General Framework

1. Obtain lots of examples
2. Represent them in some way (*This is the model*)
3. Take the image you want to search through and represent it the same way
4. Check for matches

3.2 Boosting/Cascade Classifier

- This is using multiple weaker classifiers and joining them to try and build a more robust system/model

3.3 Viola Jones

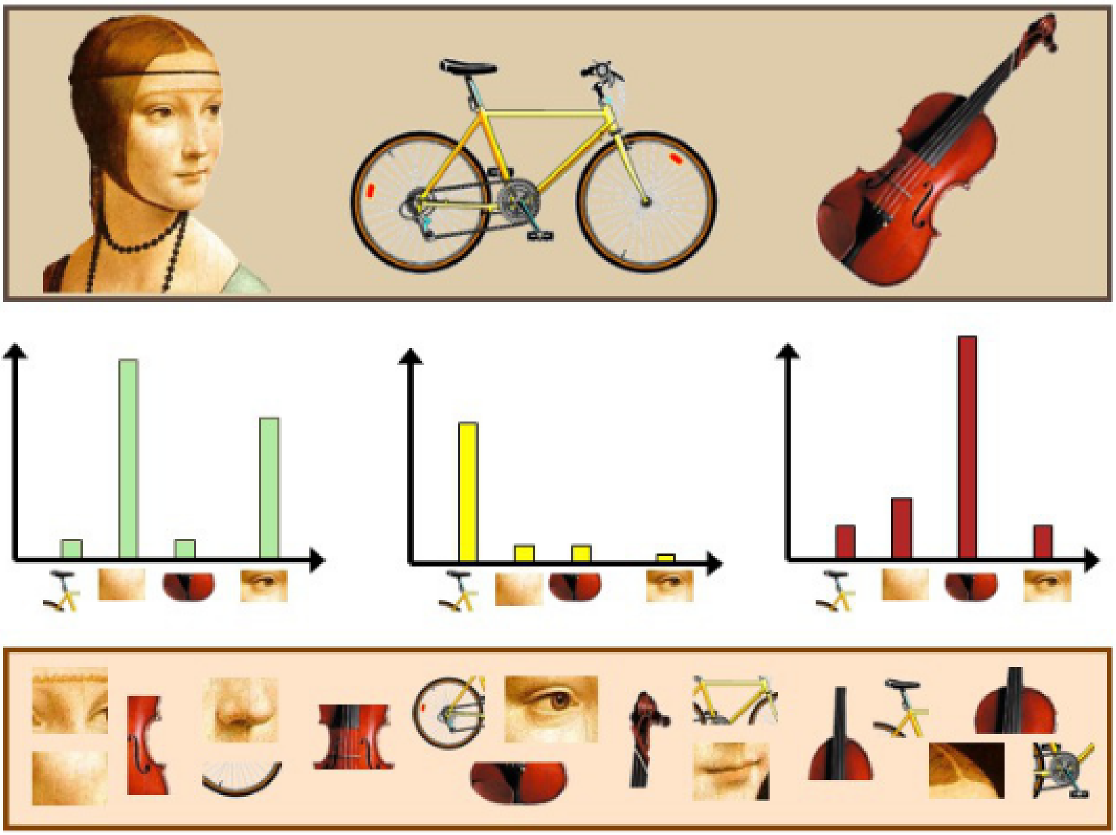
- Uses Haar features of simple small convolutions of bright and dark patches
- Couples with Adaboost in order to reduce features which aren't useful for face detection
- Uses a technique called Integral image to speed up the amount of computations needed to be performed when classifying an image
- Is slow to train, fast to use

3.4 Questions to ask when trying to recognise objects

- Accessibility - Can we compute it?
- Scope - Can we recognise individuals based on variety in the group
- Uniqueness - Do we have similar looking objects to give false positives
- Stability - Does it vary in its representation
- Sensitivity - Is it dependant on too few features
- **Cross Reference this against what makes a good feature!**

3.5 Bag of words

- Is a popular framework for object recognition
- Features are detected on a large training dataset
- These features are clustered
  - This relies on objects with similar appearance being near in feature space
- For each class of object (aeroplanes, cows) create an unordered set of these clusters



- SIFT detections (green, left) are clustered which gives us a smaller vocabulary
- On the right, two of the resulting clusters are highlighted

3.5.1 Keypoints

- It needs a training set of labelled objects
- It uses clustering to turn features into visual words
- It makes no assumption about the spatial relationships between these
  - If a cow is standing on its head, it'll get detected
  - As will a partially separated cow...
- Adds a lot of robustness
- Has a reasonable accuracy of about 80% on PASCAL VOC

4 Motion Detection

- For detecting motion we have two options:
  1. Find the things which aren't moving and ignore them (Background Subtraction)
  2. Finding the motion directly (Optical Flow)
- Once you've found the pixels which represent motion, you need to group them together into 'objects'

4.1 Background Subtraction

- Requires a static camera (all sorts of problems if it isn't)
- Makes a lot of assumptions
  - The scene is still (mostly)
  - Lighting doesn't change (much)
  - Time series doesn't have a flicker effect anywhere
- Most research in the area deals with violations of these assumptions
  - Always need a static camera though!

4.1.1 Work arounds

- A threshold variable is used to ignore small changes in frame-to-frame
- Lighting is a pain
- Therefore an adaption is required in all modern forms of Background Subtraction
- The most simple way to do this is to use an adaptive averaging technique
  - Look back through previous frames, calculate an average and use this as a background
  - This makes new information settle after a time though...
  - Two parameters used, threshold T and window W of how many frames to examine for this moving average

4.1.2 Flicker

- Often something in the background will cause pixels to vary which we are also not interested in
  - Leaves blowing
  - Camera shakes
  - TV Screens

    - Shadows from outside of the scene
- With flicker, tracking at which point an object appears can be a lot harder

4.1.3 More work arounds

- In situations with flicker and other factors, more complex modelling can be used
- These generally:
    - Treat each pixel as a time-series
    - Noise processes are modelled explicitly
    - A post-processing step might be used to get rid of small detections
        * Median filters for example

4.1.4 Additional complications

- It's actually 3D
    - Up until now we consider RGB separately
        * This is a gross oversimplification
        * They quite often vary together
        * It's better to think of each pixel as a point in RGB space
- Some objects or noise will vary more than other objects or noise
    - Having a simple threshold means you cannot take this into account
    - Actually, noise is often Gaussian
    - So modelling each pixel as a Gaussian helps

4.1.4.1 What does modelling as a Gaussian mean?

- We give a standard deviation into the equation, this means our threshold can adjust based on the width of the Gaussian
- So pixels with a lot of noise have a higher threshold

4.2 Optical Flow

5 Stereo and Multi-View

6 3D Capture Setups

7 Background Subtraction/Motion Segmentation