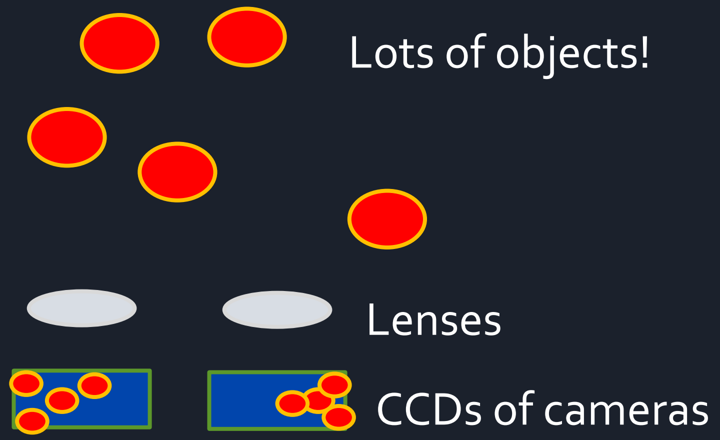
Session 8: Computer Vision 2

Dr John Fawcett

Correspondence Problem

- Which red circles in the image are which?



Correspondence Problem

- Move cameras closer (smaller ‘s’)
  - Images more similar to each other
  - Easier to solve the correspondence problem
  - Larger error from geometry
- Move cameras apart (larger ‘s’)
  - Images less similar to each other
  - Harder to solve the correspondence problem
  - Smaller error from geometry
- Requires a trade-off

$$d = \frac{fs}{a + b}$$

Figuring out Correspondence

- Need to identify ‘features’ in the images
- Move the images relative to one another so features align
- If we detect N features in one image and M features in the second, we have N x M pairs of features to consider!
    - Rapidly becomes computationally infeasible.

Figuring out Correspondence

Algorithm: Scale Invariant Feature Transform (SIFT)

- ‘Features’ are edges
- Produce scale space pyramid (last time’s lecture)
- Get approximate alignment from most blurred tier
- Loop for each tier of pyramid:
  - Reintroduce edges from less blurred image
  - Adjust alignment

Estimating Velocity

- Instead of two images taken at the same time with two cameras separated in space…
- …take two images separated in time using a single camera
- Align corresponding features between the two images
- Estimate speed and direction of motion

Estimating Velocity

- Similar tension over accuracy
- Images taken closer together in time
  - More similar to each other
  - More reliable solution to correspondence problem
  - Less accurate estimate of speed and direction of motion
- Images taken over longer time span
  - Less similar, harder to align images
  - More accurate velocity estimate

Correspondence is hard

- Can we estimate speed and direction of motion without solving the correspondence problem?
  - Spatio-Temporal Fourier Transform
  - Optical Flow
  - Intensity-Gradient Models
  - Dynamic Zero-Crossing Models

Spatio-Temporal Fourier Transform

- A video sequence is a sequence of discrete frames: $I(x,y)$
  - $I(x,y)$ is the intensity at pixel position (x,y) in the image
- Consider the sequence, $I(x,y,t)$
  - $I(x,y,t)$ is the intensity of pixel (x,y) at time t

Spatio-Temporal Fourier Transform

- Take the Fourier transform of this 3-dimensional data set
- $F\left(w_x, w_y, w_t\right) = \int_x \int_y \int_t I(x, y, t)\, e^{-i(w_x x + w_y y + w_t t)}\, dt\, dy\, dx$
- Velocity $\left(v_x, v_y\right)$ of an object is the rate of change of the (x,y) coordinates with t. Finite difference:
- $I(x, y, t) = I\left(x - v_x t_0, y - v_y t_0, t - t_0\right)$

Spatio-Temporal Fourier Transform

- Shift Theorem (shifting duals modulation) tells us that
  $$F(w_x, w_y, w_t) = e^{-i(w_x v_x t_0 + w_y v_y t_0 + w_t t_0)} F(w_x, w_y, w_t)$$
- So $F(w_x, w_y, w_t) = 0$ (or $e^{-i(\dots)} = 1$)
- i.e. $w_x v_x + w_y v_y + w_t = 0$
  - (And periodic repeats)
- This is the equation of a plane

Spatio-Temporal Fourier Transform

- $w_x v_x + w_y v_y + w_t = 0$
- The elevation tells us speed: $\sqrt{v_x^2 + v_y^2}$
- The azimuth tells us direction: $\arctan(\frac{v_y}{v_x})$

Optical Flow

- LIDAR (Light Direction and Ranging), used in some self-driving cars
- Needs a breakthrough to get beyond some limitations on accuracy and practicality
- Demo video: https://www.youtube.com/watch?v=oL67qe-Fhps

Intensity Gradient Models

- Suppose we know the rate of change of intensity of an object's surface
  - E.g. sheet of paper, 30cm wide, fading from pure black at one side to pure white at the other
  - 0 to 255 intensity units over 30cm
- Suppose a particular pixel (x,y) is getting brighter at rate 3.2 intensity units per second
  - i.e. $I(x,y,t) = I(x,y,t_0) + 3.2(t - t_0)$

Intensity Gradient Models

- Now the rate of motion of the object past the pixel is $\frac{3.2}{256} \times 30\ cm = 0.375\ cm/s$
- In general,

$$\overrightarrow{\nabla I(x,y,t)}.\vec{v} = -\frac{\partial I(x,y,t)}{\partial t}$$

Dynamic Zero-Crossing Models

- Also possible to run edge detection filters over consecutive images in a video sequence
- Then take first derivative over time to estimate speed and direction of motion based on rate of movement of the zero crossings (edge positions)

Active Contours

- Last time, we saw active contours
- Trading off internal energy and external energy: bendiness vs goodness of fit
- Demo video: https://www.youtube.com/watch?v=celddPk78yA

Codons

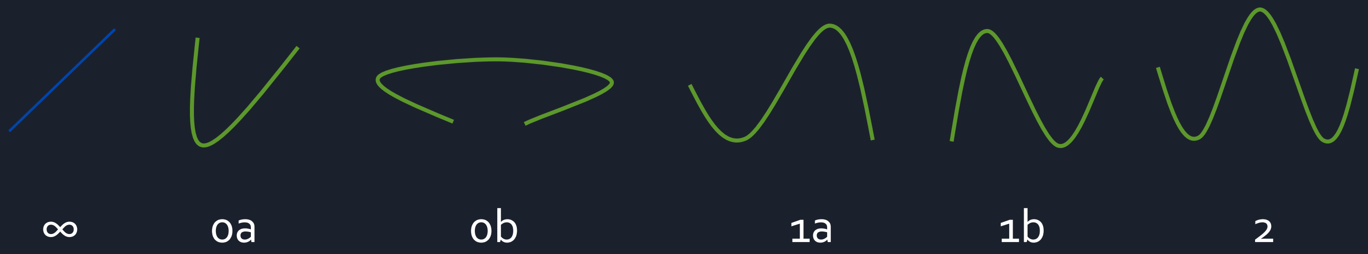
- New problem: recognise objects in a scene
  - Move from “there are edges as these pixels […]”
  - To “there is a banana centred at (x,y)”
- Desired properties:
  - Scale invariance: don’t have to store template images of bananas at all sizes
  - Rotation invariance: don’t have to store template images of bananas at all orientations
  - Position independent: object-centred coordinates

Codons

- Curves can be described by piecewise spline segments
  - Split a continuous curve at points of minimum curvature
  - Between curvature minima, there must be 0, 1 or 2 points of zero curvature
  - Between the minima, there must be a maximum curvature point
  - (First) zero could be before or after the maximum curvature point

Codons

- We ‘name’ pieces of curve by the number of zero crossings of curvature and whether a zero was reached before (b) or after (a) a maximum



Codons

- Combine edges found in an image using codon segments
- Compare code of the closed boundary to objects in the database (lexicon)
- Subsequence match to detect orientation

Extension into 3D

- The idea of coding the shape of a line as a sequence of letters can be extended to describe the shape of a solid object as a sequence of coefficients

$$R = Ax^a + By^b + Cz^c$$

- Only 6 parameters needed (div by R)
- Large a,b,c give near-rectangular corners
- A,B,C describe stretching along the axes

Eigenfaces: the task

- Consider the task of identifying people from a known population
  - E.g. employees of a company as part of a security system
- Input data: 10,000 people in 2-dimensional photographs taken in controlled conditions: consistent illumination, pose, position, facial expression, etc.
- Task: identify the person in a new photograph

Eigenfaces: sample image

Michael J. Fox
(credit Wikipedia, Creative Commons License)

Eigenfaces: approach

- Consider the intensity of each pixel to be a random variable
- Across all 10,000 sample images, we want to look for patterns among the pixels in the images
    - E.g. if the pixel at (2,4) has value X, predict that the pixel at (20,40) has value 2X.
- This could provide an optimal encoding for the images

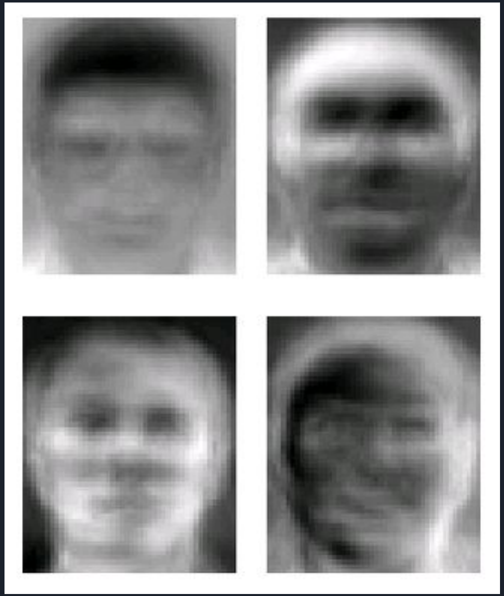
Eigenfaces: approach

- Calculate covariance matrix
  - Across the 10,000 sample images, compute the correlations between each pixel and every other pixel
- Calculate the eigenvectors (and normalise)
- Perform principal components analysis: select largest magnitude eigenvalues, e.g. top 20
- Provides an optimal encoding given that number of coefficients

Eigenfaces: visualisation

- Image: AT&T Research Laboratories, Cambridge

- These images form an orthogonal basis set!



Eigenfaces

- Compute projection of each face onto the chosen eigenfaces
- Store these coefficients as the ‘fingerprint’ of each individual
- Identify new photographs by computing the projection coefficients against the basis set
- Compare 20-element feature vectors with fingerprints of known faces
- Output best match (with confidence interval)

Eigenfaces: computation

- Calculating the principal components analysis is *very* computationally intensive
  - But this is an up-front cost, only required once
- Computing projection coefficients is cheap
  - Cheap to process new images
- 20-element feature vectors are relatively small: small storage costs for database
- Cheap to compute and compare norms of error vectors to pick best match

Eigenfaces vs Fourier

- Both express data as a weighted linear sum of basis functions
  - Eigenfaces or complex exponentials
- Data-dependent vs data-independent basis
- Incomplete vs complete basis set
- Orthogonal so projection coefficients equal expansion coefficients
- Note: Gabor wavelets are not orthogonal

Eigenfaces: pros

- Data-dependent basis set optimises compression into a small feature vector
  - Compact data storage
  - High accuracy with only a small number of coefficients
- Computationally cheap to process new images – fast matching
- Accuracy typically better than 90%

Eigenfaces: cons

- Unable to adapt to new people with distinctive facial features or changes over time
    - E.g. the first person with a beard or the first person with an earring
- Requires images to be normalised: same pose, illumination, size
- Must recompute all fingerprints when refreshing the sample images

3D Facial Modelling

- Another approach to facial recognition
- Within-class variability is high for 2D photographs of the same face
- Between-class variability is low for different faces in 2D photographs
- Move to 3D to increase separation between similar faces and allow for “standardisation” of pose angle, illumination, facial expression to reduce within-class variability

3D Facial Modelling

- Build a 3D model of a face from one or more images
- A short video sequence is especially helpful
- Parts of the face might be occluded in one frame of video but visible in another
- Must solve correspondence problem and correct for motion blur
- Works best when video is too short for objects to change shape

3D Facial Modelling

- Correct for pose angle
  - E.g. adjust so the nose points directly towards the camera
- Adjust for illumination
  - Remove specular highlights
  - Remove shadows
- Distort to remove facial expression
- Match resulting model against database of known faces

3D Facial Modelling

- Re-rendering faces at different pose angles or under different illuminations works best for small corrections
- Match using closeness of volume of 3D model matching templates in the database

Viola-Jones

- Attempting to detect faces, not identify faces
- Surprisingly difficult!
- Build a strong classifier from a sequence of weak classifiers
  - Strong classifier: low error rates
  - Weak classifier: high error rates
- All weak classifiers must detect a face in order for the Viola-Jones algorithm to report a face is present

Viola-Jones

- Use Haar transform
  - Like sine-waves but binary (black/white)
  - Individually: quite weak classifiers
  - Combine enough of them: quite strong!
- Uses a supervised machine learning system based on AdaBoost

Human Vision

- We don’t know how it works
- Clues from
  - Scans of brain “wiring diagrams”
  - Neural activity when presented with images
  - Effects of brain injuries
- General purpose
- Doesn’t always yield correct results
- Returns a result when none exists

Human Vision

- Computer vision tasks often lie within a narrow domain
- Computer vision tasks often demand higher accuracy than human can achieve
    - E.g. accident rate for self-driving cars
- Functional streaming in the brain
- Feedback connections
- Optical Illusions

Human Vision

- Model human eye and brain as a (huge!) neural network
- Sensor data reports
    - Some colour
    - Some motion
    - Some contrast
    - Some edges
    - Some patterns/texture

Human Vision

- Try to “run a simulation” of the brain computer on a digital computer
- Compare each stage to the human responses

Now it's time for you to have a go

Exercise sheet!