COSC470 Feature Detectors

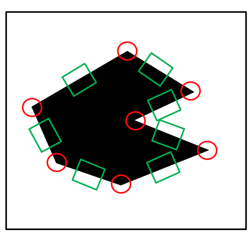
COSC470: Special Topic
Computer Vision | 3D Reconstruction
Steven Mills

What is a Feature

- For us, a feature is a point in an image which
  - Is well *localised* in two dimensions
  - Can be *repeatedly* detected in multiple images
- Not all applications have these requirements
- Two main types of features
  - Corners – Where the image gradient has high curvature – image changes in all directions
  - Blobs – dark (or light) regions surrounded by lighter (or darker) pixels

Corners and Edges

- Corners are points where the intensity changes in all directions
- Edges are points where the intensity changes in one direction
- Uniform regions have no change in any direction



Harris and Stephens Corners

$$\sum_{(x,y)\in R} (I(x+u, y+v) - I(x,y))^2$$

← The difference between a region, *R*, in an image, and that region offset by (*u*,*v*)

$$\approx \sum_{(x,y)\in R} \left(I(x,y) + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} - I(x,y) \right)^2$$

← Is approximated by a Taylor expansion

$$= \sum_{(x,y)\in R} \left(u^2 \left(\frac{\partial I}{\partial x}\right)^2 + uv\frac{\partial I}{\partial x}\frac{\partial I}{\partial y} + v^2\left(\frac{\partial I}{\partial y}\right)^2 \right)$$

$$= u^2 \sum \left(\frac{\partial I}{\partial x}\right)^2 + uv \sum \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} + v^2 \sum \left(\frac{\partial I}{\partial y}\right)^2$$

← Which, after some arithmetic

$$= [u \quad v] \begin{bmatrix} \sum \left(\frac{\partial I}{\partial x}\right)^2 & \sum \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \sum \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \sum \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

← Gives us a matrix which describes how the image changes as we move in each direction

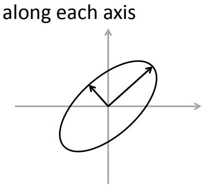
Eigenvectors and Eigenvalues

- Suppose we have a square matrix A, and a vector x such that $Ax = \lambda x$ for some λ.
- x is called an *eigenvector* of A and λ is the corresponding *eigenvalue.*
- The eigenvalues can be found by solving the *characteristic polynomial* of the matrix

$$\det(A - \lambda I) = 0$$

Eigenvectors and Eigenvalues

- We commonly end up with symmetric positive semidefinite matrices
  - Symmetric – around the main diagonal
  - Positive semidefinite $x^T A x \geq 0$ for any vector x
- Co-variance matrices and the matrix for Harris and Stephens corners fit this model
- The eigenvectors form an *orthogonal basis*
- The eigenvalues tell us how the matrix varies along each axis



Harris and Stephens Corners

$$\det\left(\begin{bmatrix}\sum\left(\frac{\partial I}{\partial x}\right)^2 & \sum\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\\ \sum\frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \sum\left(\frac{\partial I}{\partial y}\right)^2\end{bmatrix}-\lambda I\right)=0$$

$$\det\left(\begin{bmatrix}X-\lambda & Z\\ Z & Y-\lambda\end{bmatrix}\right)=0$$

$$(X-\lambda)(Y-\lambda)-Z^2=0$$

$$\lambda^2-\lambda(X+Y)+(XY-Z^2)=0$$

$$\lambda=\frac{X+Y\pm\sqrt{(X+Y)^2-4(XY-Z^2)}}{2}$$

- The determinant of a 2x2 matrix is fairly easy
- A corner will have two large eigenvalues – it varies in 2 dimensions
- Harris & Stephens use an approximation to avoid the square root

What Makes a Feature?

- Values like H&S measure of ‘cornerness’
- They don’t tell you if something is a corner or not – just how ‘cornerey’ it is
- Solution: comptue it at every pixel and take pixels that are local maxima as the corners
- Local maxima can be computed over a local region – still need to determine region sizes

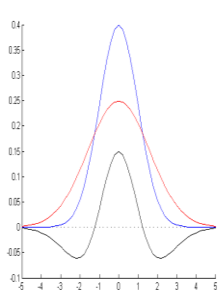
Shi-Tomasi Corners

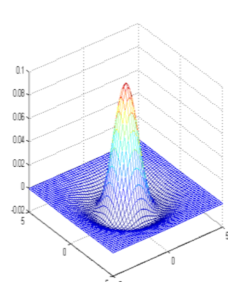
- Shi and Tomasi’s paper ‘Good Features to Track’ reaches the same formula
- They consider what points are best to use when tracking motion from frame to frame
- They suggest using the smaller eigenvalue as a measure of cornerness

Blobs

- An alternative to ‘corners’ to define features
- A ‘blob’ is a bright area surrounded by a dark area or vice-versa
- Polka dots are the archetypical blob, but these features appear in most natural images
- Commonly detected using the *difference of Gaussians* (DoG), an approximation to the *Laplacian* of a Gaussian function (LoG)

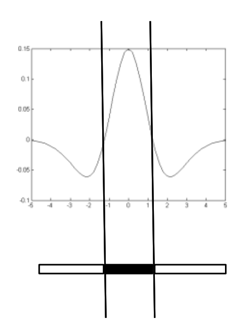
Difference of Gaussians





Scale Space

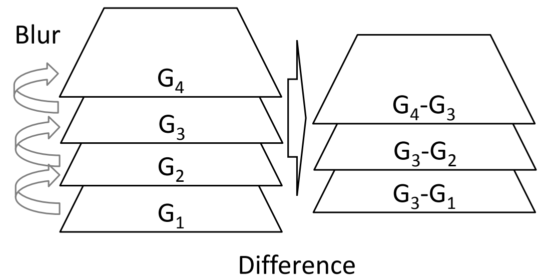
- Blob detectors work at a particular scale
- For DoG this depends on the width of the central peak
- This gives maximum (±) magnitude response
- But not all features are the same size



Scale Space

- Solution is to apply filters at different scales
- Blur image with a series of Gaussian filters
  - Blur repeatedly with variance σ^2
  - This is the same as blurring with σ^2, $2(\sigma^2)$, $3(\sigma^2)$…
- The difference between adjacent levels provides a stack of DoG filters
- Look for local maxima/minima both within a scale and across adjacent scales

Scale Space



Experimental Design

- Computer vision deals with the real world
  - Cameras gather information about the world
  - We then reason about this information
- It therefore has more issues with experimental design than some other areas of comp sci.
  - What is a 'typical' image or the range of images?
  - What is the 'correct' behaviour of an algorithm?
  - How do we deal with uncertainty and outliers?

Experimental Design

- Questions to ask
  - What image(s) will I use?
  - How can I make a set of tests that cover a range of values, conditions, or other attributes?
  - What are the parameters of my algorithm and how can I test different values?
  - How can I tell how well an algorithm is doing?
  - How can I compare results of different tests?
  - Can I tell if variation is real or just noise?

Fair Testing

- Suppose one algorithm takes, on average 9.3s to run and another takes 10.5s
- Is the first algorithm faster?
  - What is the variation in different runs? 9.3s ± 5.1s probably isn't really different to 10.5s ± 3.7s
  - Also depends on sample size – statistics
  - Is the test fair – have the two algorithms been equally (and aggressively) optimised?

Statistical Methods

- There are lots of statistical tests
  - To compare two mean values, consider a *t*-test – eg: method A's average error is 1.3, B's is 1.1
  - To compare proportions, try Pearson's χ^2 test – eg: method A works 67% of the time, method B 73%
  - To see how one variable depends on another you might use Pearson's *r* as a measure of correlation
  - Beware of correlation vs. causation, confirmation bias, small samples, non-normal distributions, etc.