COMP90072 - Stereo Vision Part 1
Cross Correlations

Expected completion date: 9/4/18

Learning outcome goal for Part 1 is an understanding of:

- Vectorisation
- Abstraction
- Matrix Manipulation in MATLAB

Required Mathematical Understanding

- Linear Algebra (assumed knowledge)
- Fourier Transforms (assumed knowledge)
- Cross Correlation

Code for Submission

- Spatial cross correlation 1d
- Spatial normalised cross correlation 1d
- Signal offset checker
- Spatial normalised cross correlation 2d
- Where’s Wally finder (with completion time)
- Frequency domain cross correlation 1d
- Bonus

Notes:
None of your code in Part 1 can use any of the *xcorr* correlation functions, but they may be useful to check your answers.

1 Spatial Cross Correlation + Normalised Spatial Cross Correlation in 1d

Code for submission: *spatial correlation 1d, normalised spatial correlation 1d*

Cross correlation:

$$r = \frac{1}{N}\sum_{i=1}^{i=N}(f(i) - \overline{f})(g(i) - \overline{g})$$

Normalised cross correlation:

$$R = \frac{1}{N}\sum_{i=1}^{i=N}\frac{(f(i) - \overline{f})(g(i) - \overline{g})}{\sigma_f \sigma_g}$$

where,

$$\sigma_f = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N}(f(i) - \overline{f})^2}, \qquad \sigma_g = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N}(g(i) - \overline{g})^2}$$

Make a MATLAB function which takes two **vectors** of the same size and passes one over the other to create:
1) a correlation vector and
2) a normalised correlation vector.

2 Signal Offset

Code for submission: *script to find offset*
Non-code submission: *offset time, run time, sensor distance*

You will be provided with two signal files, you know these signals have come from the same source and are just offset by some time. Using cross correlation find the offset time. Knowing that this signal propagates at 333m/s, what is the distance between the two sensors, *x* (refer Figure 1)?

How long does this take?
Hint: tic + toc commands will be useful

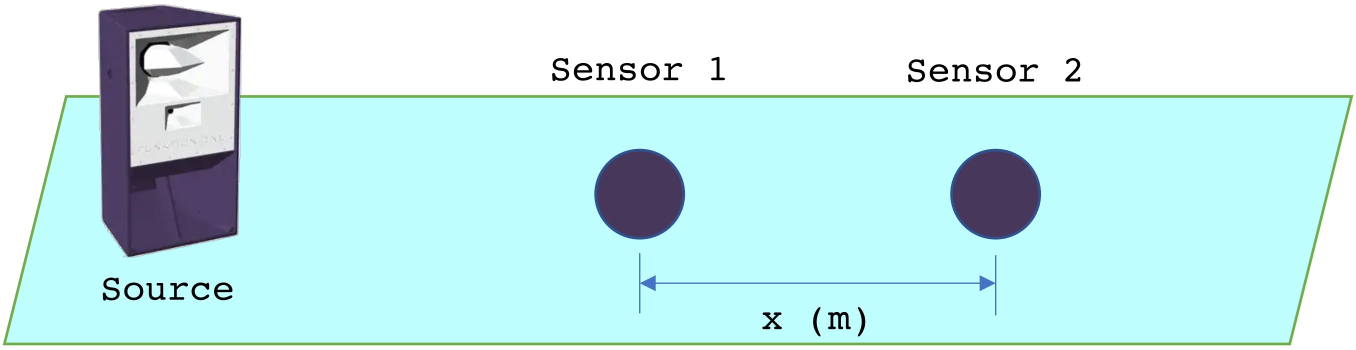


Figure 1 Source and sensor arrangement

3 Spatial Cross Correlation + Normalised Spatial Cross Correlation in 2d

Code for submission: *normalised spatial correlation 2d*

Now write a cross correlation function for 2d signals. Create a MATLAB function that receives two **matrices**, t (*template*) and A (*search region*), and returns normalized cross-correlation of these two matrices. The matrix A will always be larger than the matrix t. Your function should use two nested for-loops to "lag" t over A, computing for each "lag" the cross-correlation, r.

Normalised cross correlation expanded in 2d:

$$R\left(lag_x, lag_y\right) = \frac{\sum_{x,y}[A(x,y) - \overline{(A_{lag_x,lag_y})})][t(x - lag_x, y - lag_y) - \overline{t}]}{\{\sum_{x,y}[A(x,y) - \overline{(A_{lag_x,lag_y})})]^2 \sum_{x,y}[t(x - lag_x, y - lag_y) - \overline{t}]^2\}^{0.5}}$$

Where $\bar{t}$ is the mean of t, $\overline{A_{lag_x,lag_y}}$ is the mean of A in the region under t.

Hints:

- *Images are just a matrix of pixel values, try using an image instead of a matrix*
- *We will be working with pixel intensity (greyscale) rather than RGB values. The pixel intensity can be found by taking the mean of the rgb values*
- *Look at the size of the matrix created using imread(), where are the rgb values?*
- *You will need to find a way to deal with out-of-bounds errors at the image edges.*
- *MATLAB has some built in images, try:*
  >> imread('onion.png');
  >> imread('peppers.png');
  Or you can use any image you want

4 Where's Wally

Code for submission: *script to find the rocket man*
Non-code submission: *image, run time*

Find the rocket man in the maze and place a red star on him using your spatial cross correlation. This might take a while, what is the run time?

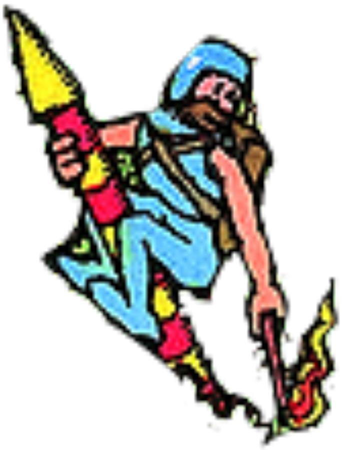


Figure 2 Rocket Man (template)

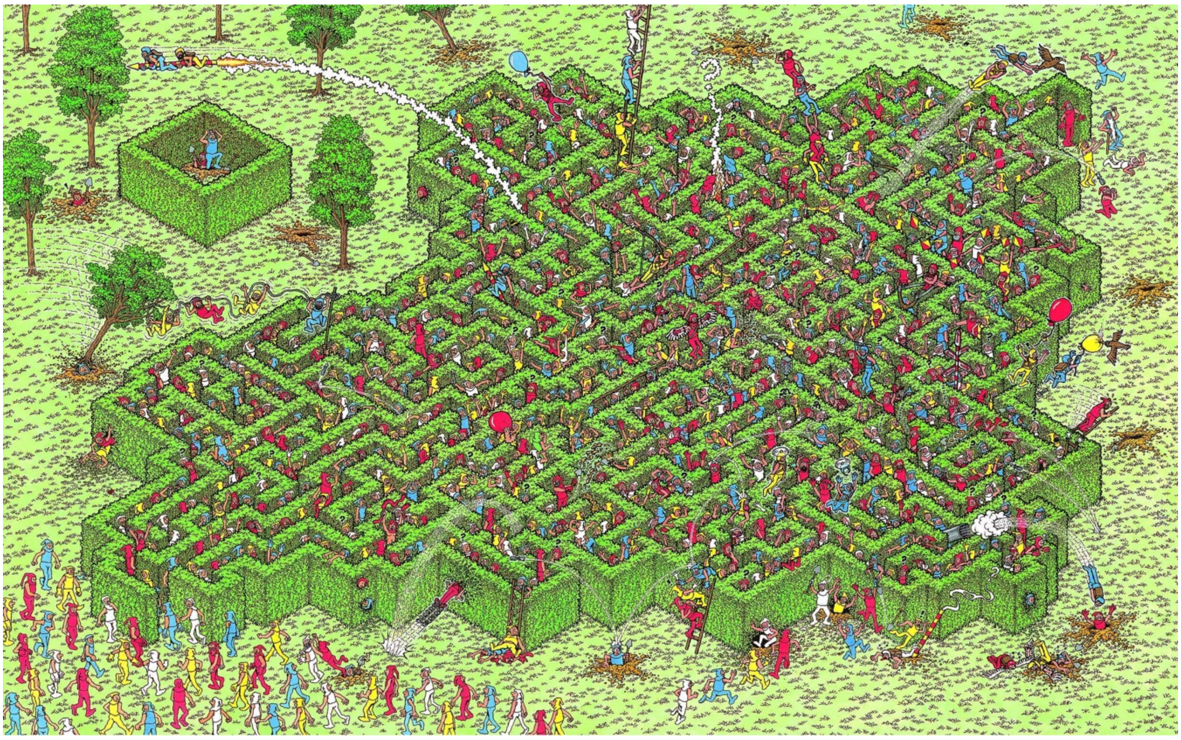
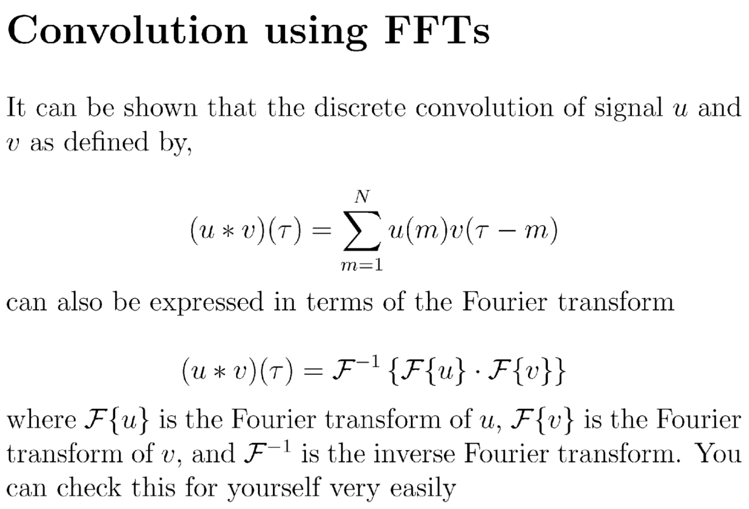


Figure 3 Maze (search region)

5 Spectral Cross Correlation

code for submission: *spectral correlation function*
non-code submission: *difference in run time between spectral and spatial methods*
This section will look at spectral cross correlation **in 1 dimension**.

Cross correlation can also be done in the spectral domain by completing a Fourier transform, multiplying signals, and doing an inverse Fourier transform.
We're not going to go through the derivation of a Fourier transform but there are some great videos on the topic linked at the end of this document.



Convolution using FFTs

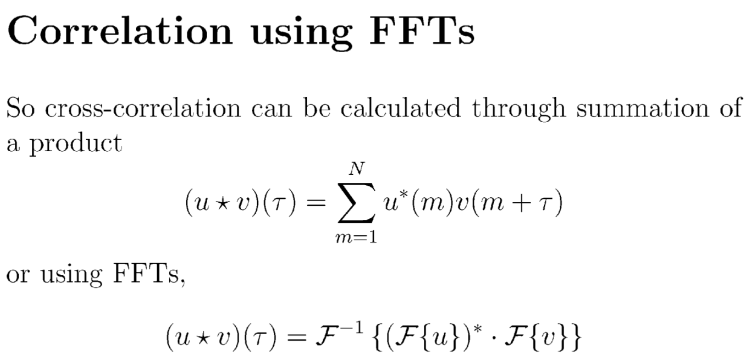
It can be shown that the discrete convolution of signal u and v as defined by,

$$(u * v)(\tau) = \sum_{m=1}^{N} u(m)v(\tau - m)$$

can also be expressed in terms of the Fourier transform

$$(u * v)(\tau) = \mathcal{F}^{-1}\left\{\mathcal{F}\{u\} \cdot \mathcal{F}\{v\}\right\}$$

where $\mathcal{F}\{u\}$ is the Fourier transform of u, $\mathcal{F}\{v\}$ is the Fourier transform of v, and $\mathcal{F}^{-1}$ is the inverse Fourier transform. You can check this for yourself very easily



Correlation using FFTs

So cross-correlation can be calculated through summation of a product

$$(u \star v)(\tau) = \sum_{m=1}^{N} u^{*}(m)v(m + \tau)$$

or using FFTs,

$$(u \star v)(\tau) = \mathcal{F}^{-1}\left\{(\mathcal{F}\{u\})^{*} \cdot \mathcal{F}\{v\}\right\}$$

Where the * refers to the complex conjugate.

Re-analyse the signals from Section 2, *Signal Offset* with your new spectral correlation code.
What is the run time

6 Bonus: Pattern Finder

Using your faster spectral code, pick a song (.wav or .flac files are easiest) and find all occurrences of a particular element. This could be a chorus, snare drum, word or anything you can reliably find with cross correlation.
Show the resulting correlation vector and then line up the occurrences of the element on a plot of the signal (x-axis: time, y-axis: frequency)

Check these two videos out for Fourier Transform understanding:
https://www.youtube.com/watch?v=r18Gi8lSkfM
https://www.youtube.com/watch?v=spUNpyF58BY