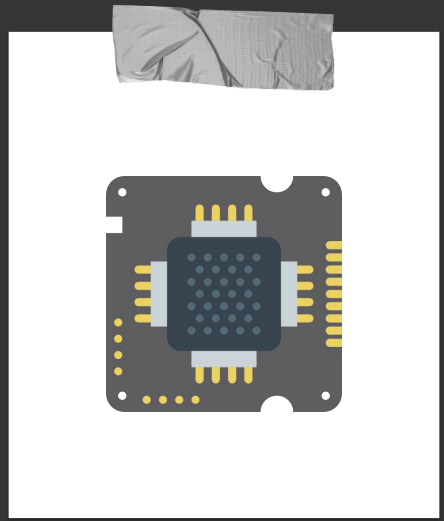
Parallel Multichannel Multikernal Convolution

A presentation by
Wessam Gholam & Eoin O’Neill

Here's the code. Improve it.

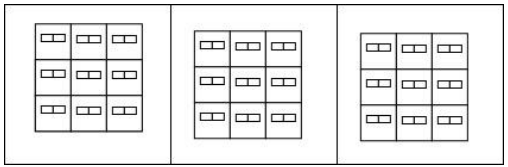
(With a little help from a few optimisation techniques)



1. Reorder kernel array

➔ **Innermost loop deals with channel values**

Each channel in a pixel must be accessed before moving to the next pixel



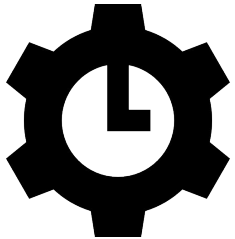
2. Vectorisation

- **Access four values at once**
  Various operations can be performed at once
- **Total loop iterations decreased by 4**
  Show them a specific person who would benefit from your solution.

$$\{w, x, y, z\}$$

3. Parallelisation

- **OpenMP**
  API which allows parallelisation
- **Three outermost loops are parallelised**
  Loops run simultaneously - more efficient



4. Remove loops

- **Possible when kernel order is one**
  x and y will be constant and will equal 0
- **Two loops can be removed**
  x and y will be constant and will equal zero

~~for(x = 0; x < 10; x++)~~

Results

Average speedup and correctness

Test case	Average speedup	Sum of absolute differences
128 128 5 32 100	82.8	~0.029460
32 32 3 64 1024	77.2	~0.017420
255 255 1 63 127	64.7	~0.016890
192 192 7 1 12	12.2	0

Our sponsors


RapGame
Tuesday 5pm
ON TRINITYFM.COM