

main()

Data input
Start of timer

image_rgb
image_gray
temp_buf
filter
output

edge_detect()

(no computations at this stage)

rgbToGrayscale()

Convert **image_rgb** into **image_gray**

Configure **filter** with smoothing filter values

convolve2d()

Perform a 2D convolution of **image_gray** using **filter**, with the result stored in **output**

Configure **filter** with vertical filter values

convolve2d()

Perform a 2D convolution of **output** using **filter**, with the result stored in **image_gray**

Configure **filter** with horizontal filter values

convolve2d()

Perform a 2D convolution of **output** using **filter**, with the result stored in **temp_buf**

(no computations at this stage)

combthreshold()

Set each position of **output** to 255 or 0 if the corresponding positions in **image_gray** and **temp_buf** meet a given threshold

(no computations in this stage)

End of timer
Write **output** to file
Verify result with checksum

Array sizes:

```
image_rgb:  int[H][W * 3]
image_gray:  int[H][W]
temp_buf:    int[H][W]
filter:      int[K][K]
output:      int[H][W]
```

Where H is the image height, w the width, and K the filter size

Right-directed arrows represent function calls, and their args.

Left-directed arrows represent function returns

Dotted lines represent a function waiting for a call it made to another function to be finished.