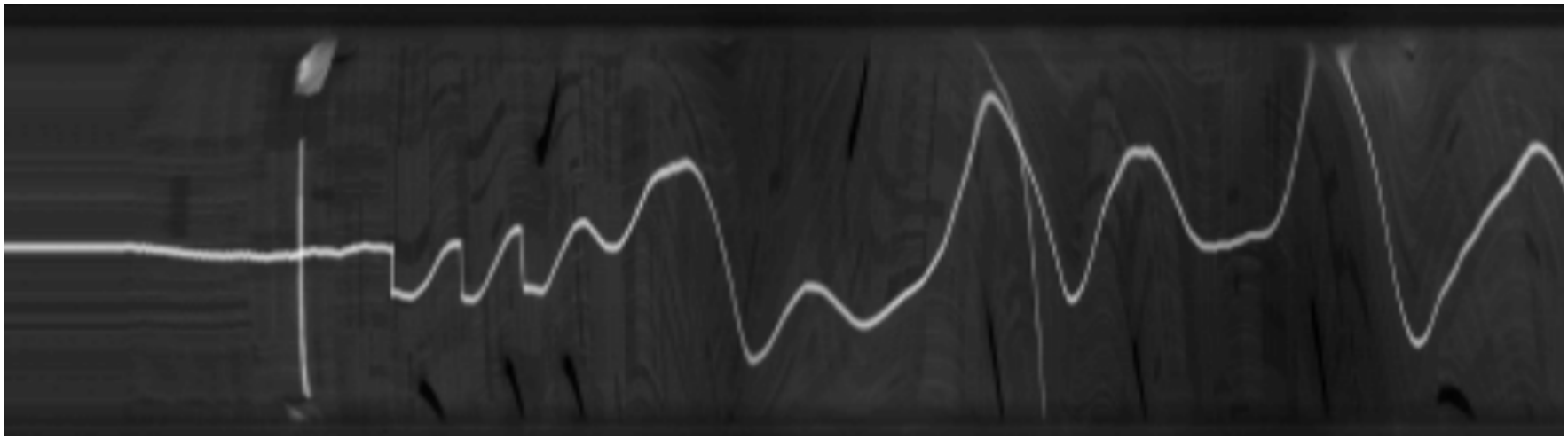
EECS 192: Mechatronics Design Lab

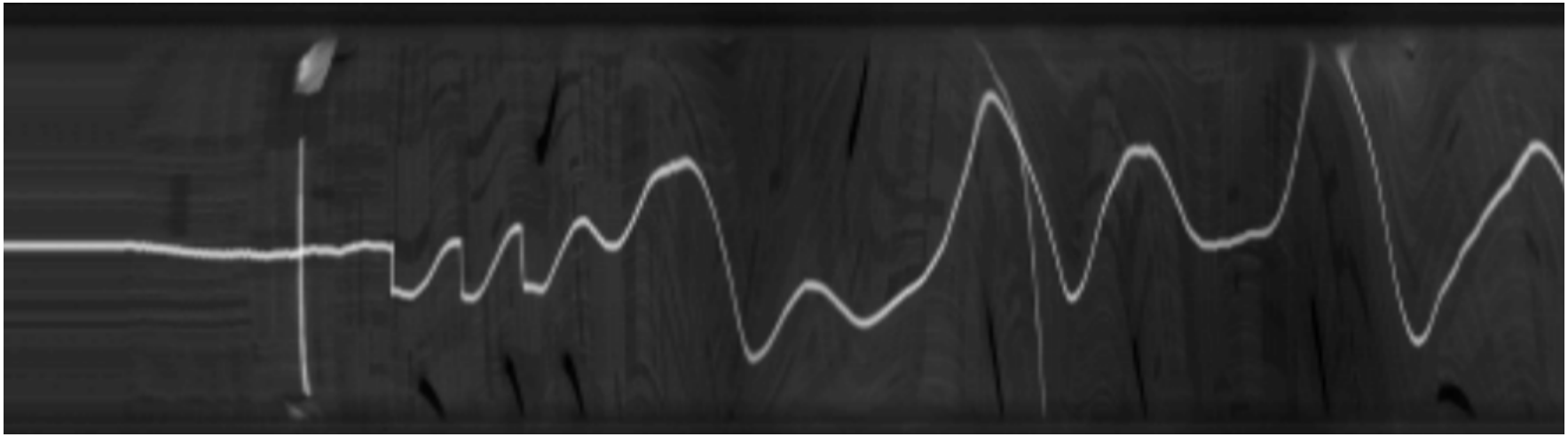
Discussion 7: Camera

GSI: Justin Yim

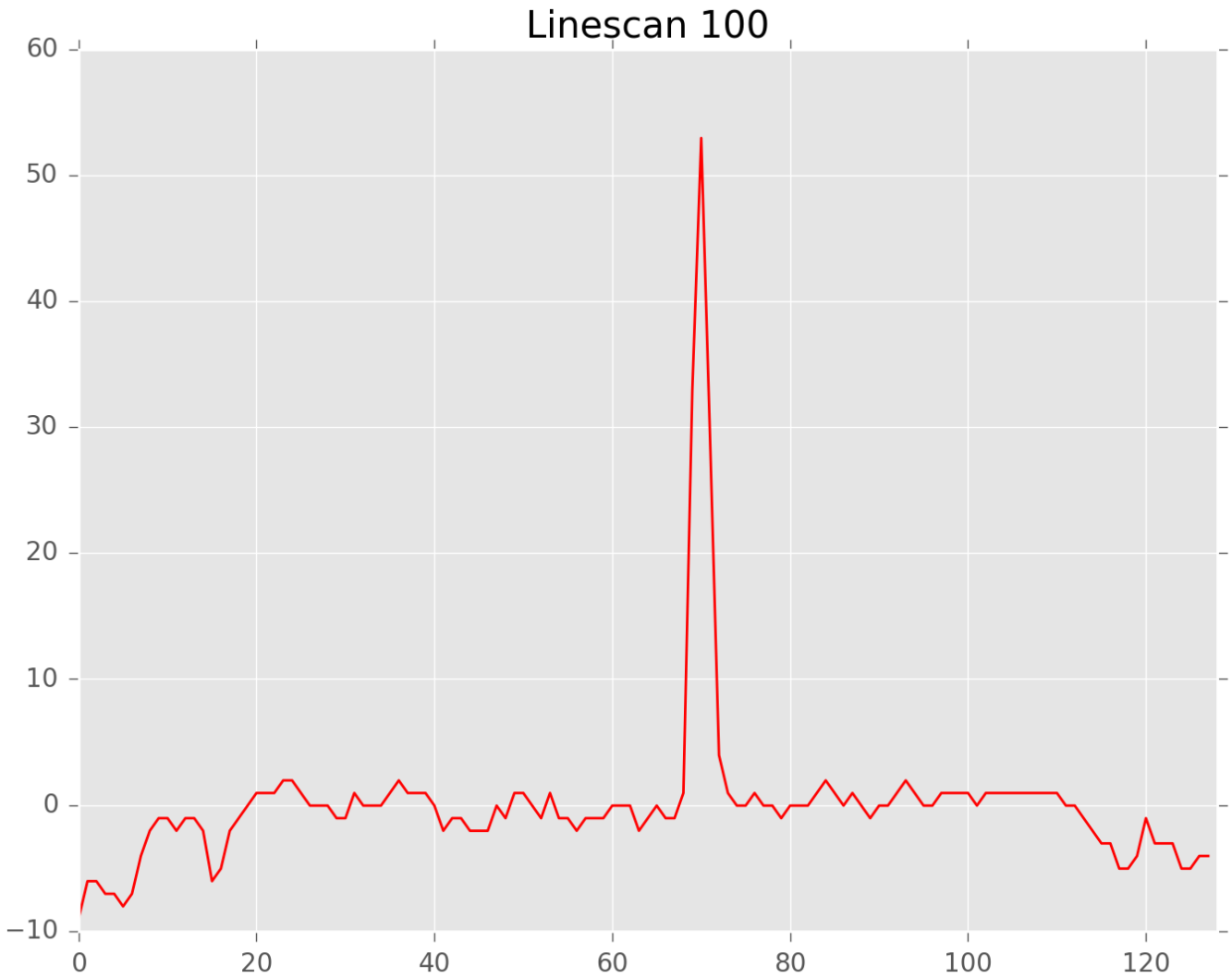
6 & 7 Mar 2019 (Week 7)

Line Sensing

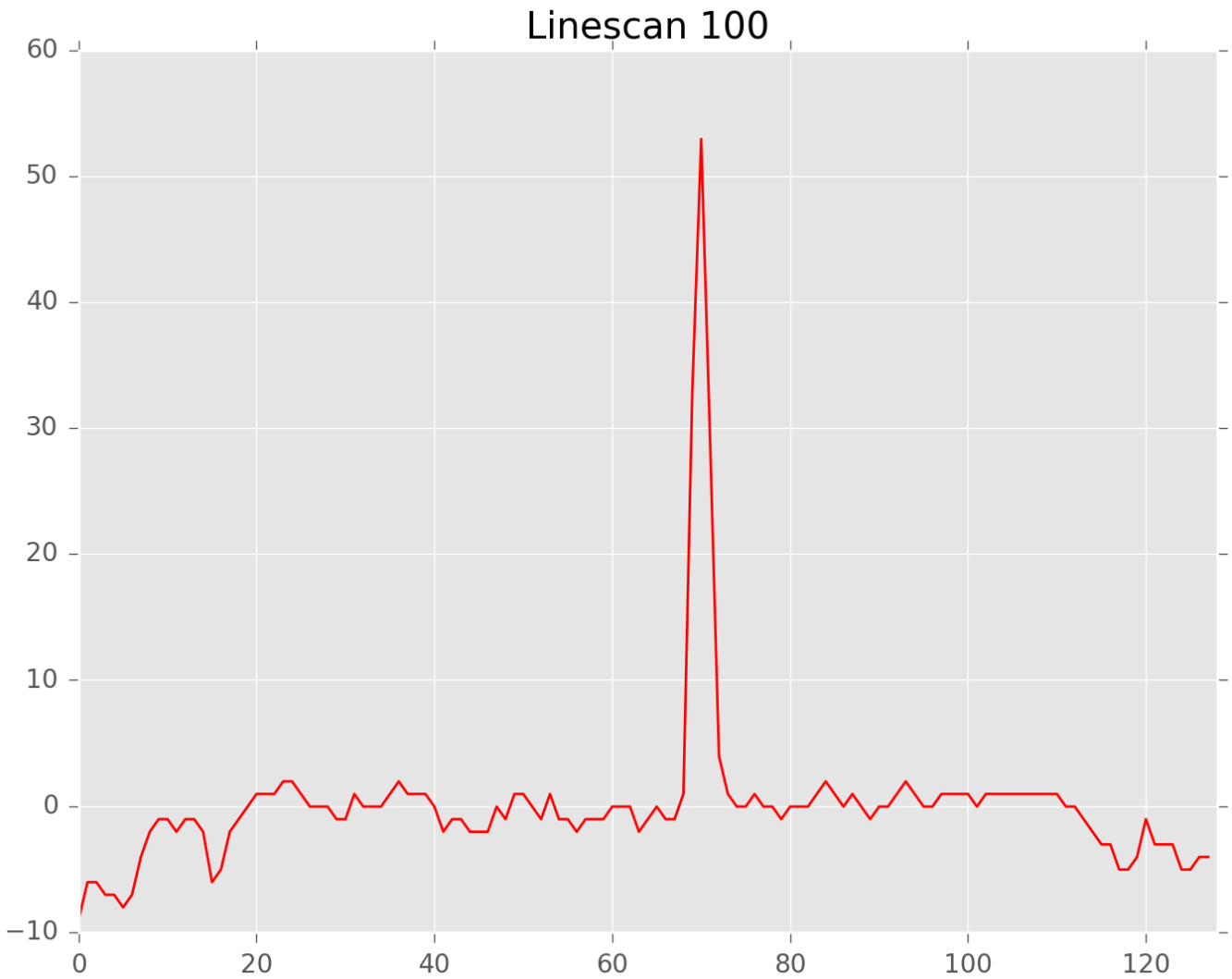




Let's look at one specific frame

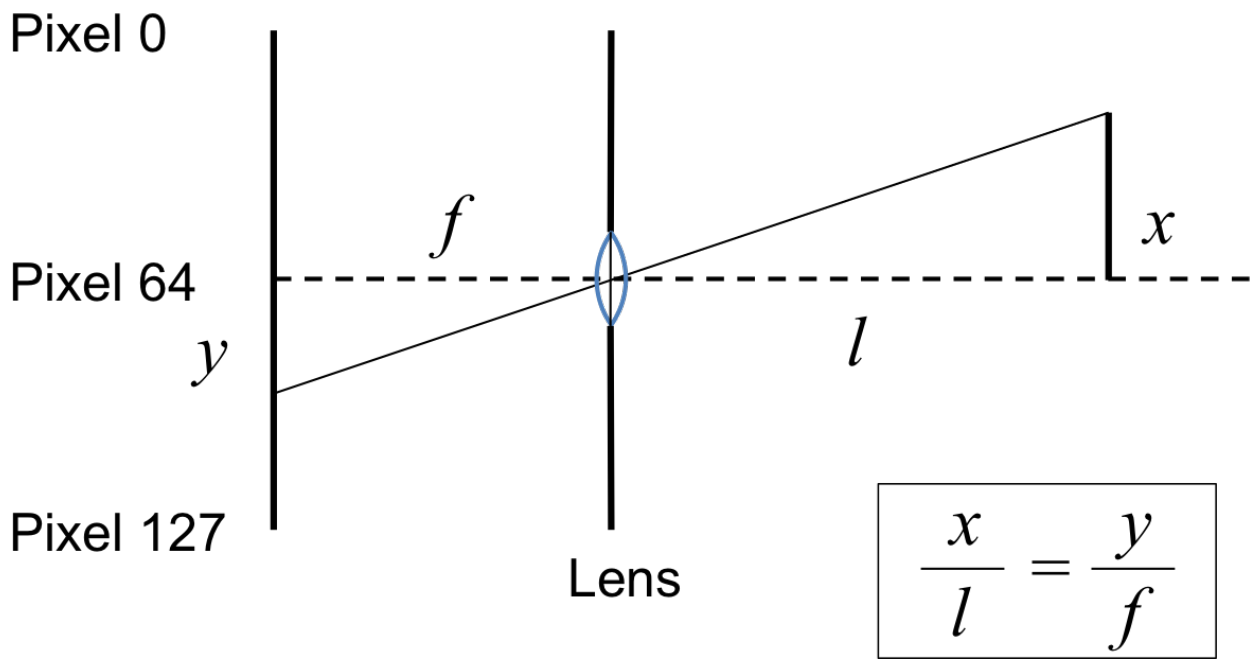


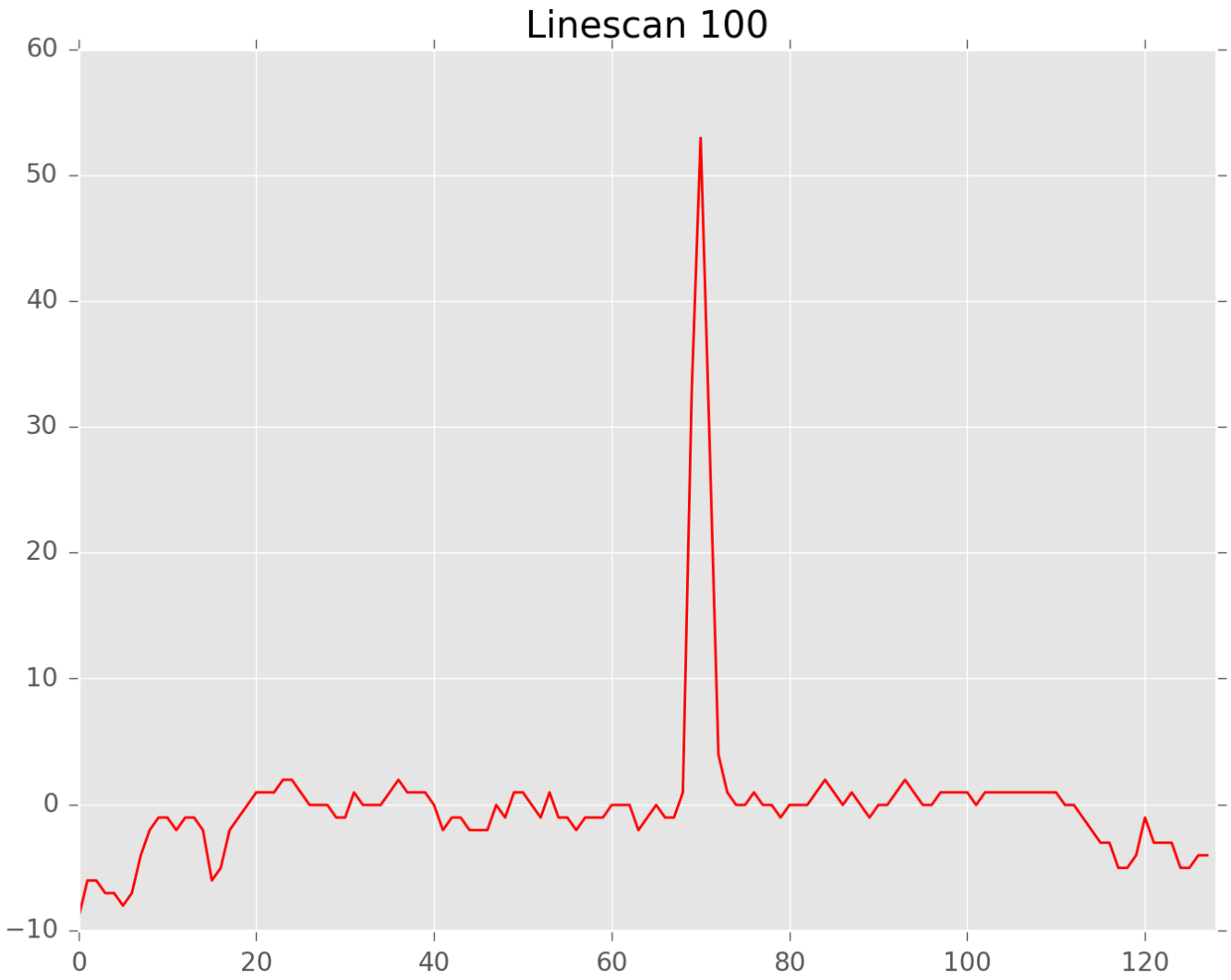
frame[100] - mean(frame[100])



Which pixels correspond to which points in front of the car?

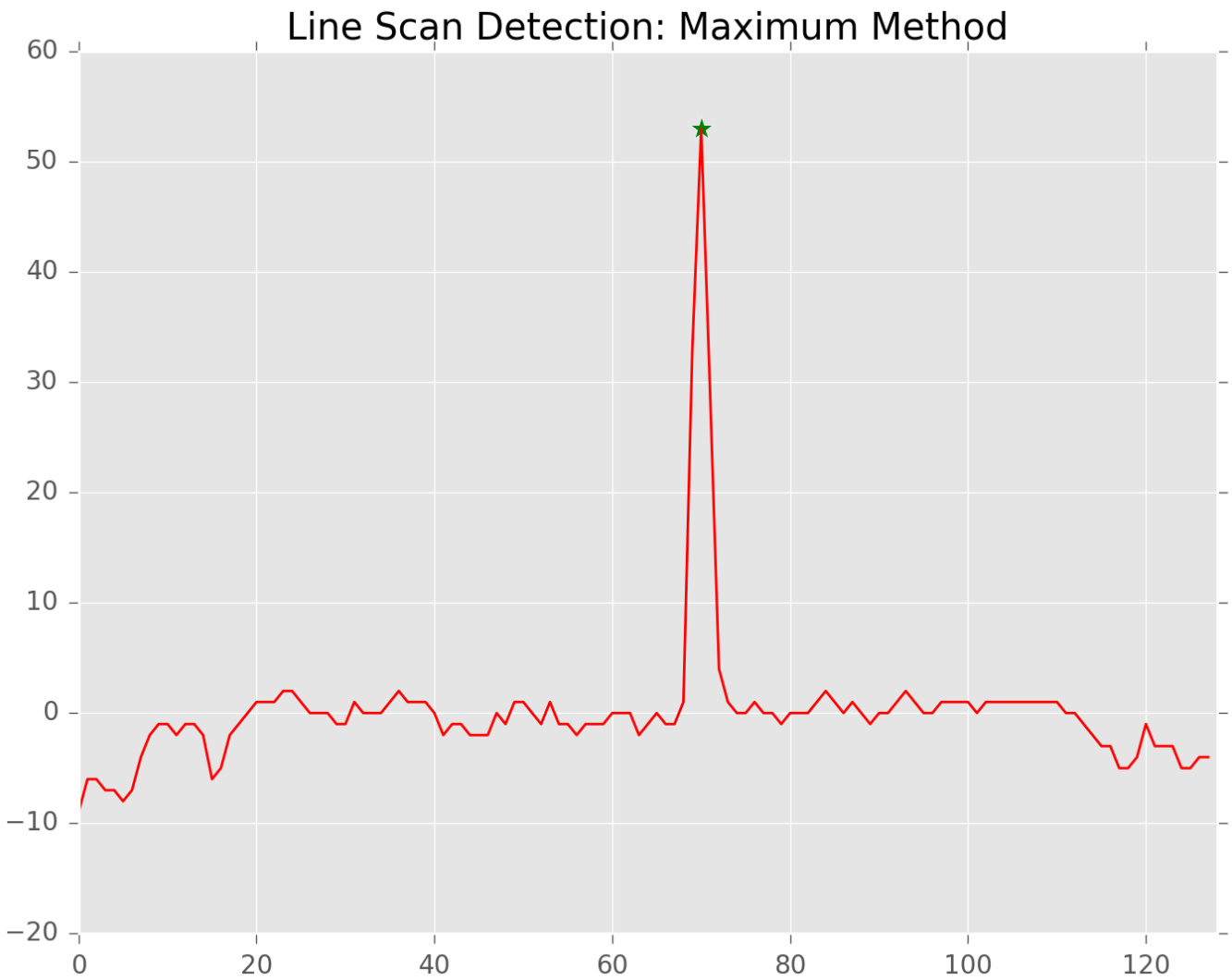
- Pinhole camera model
- Can calibrate centimeters-to-pixels gain for a fixed camera distance



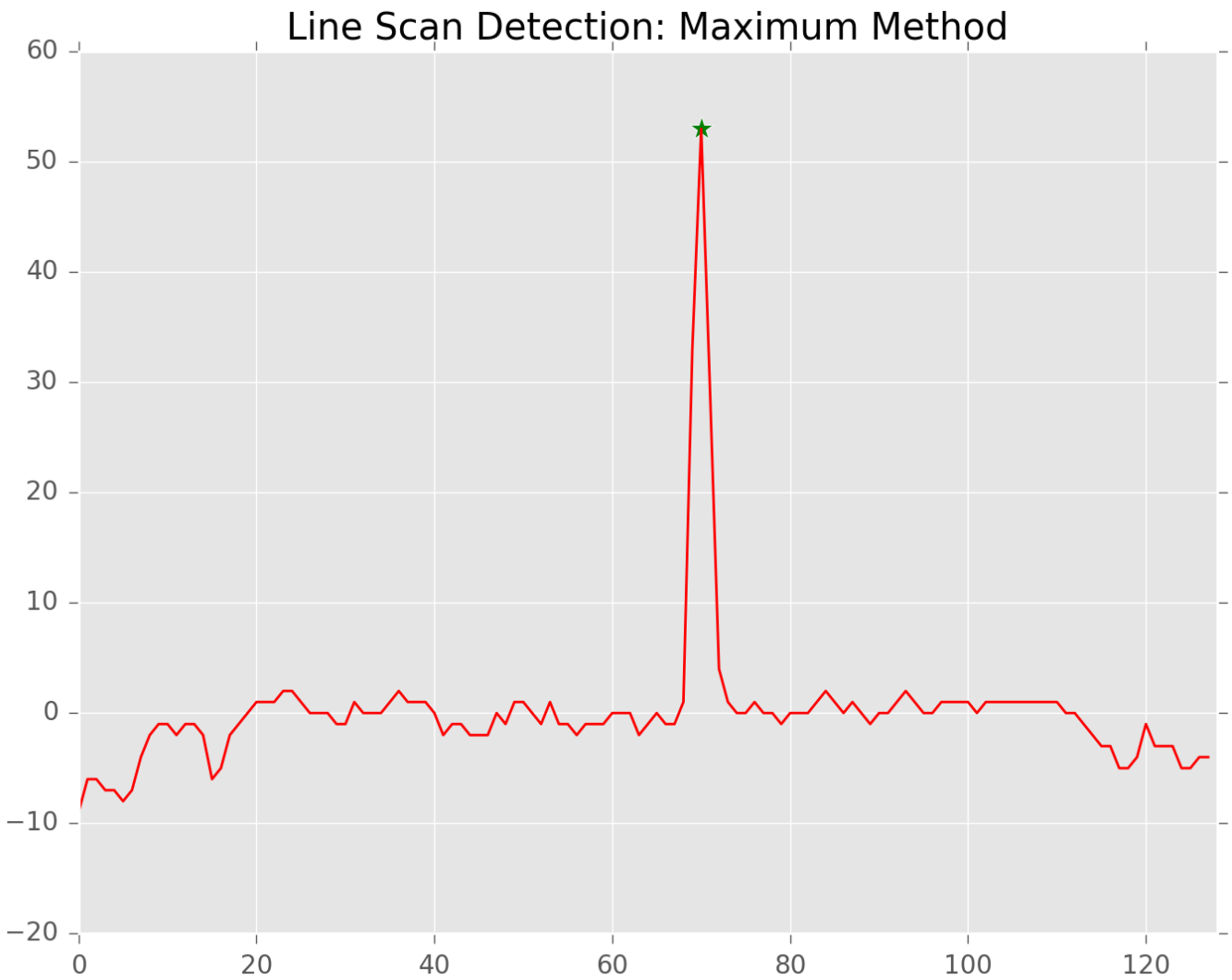


How can we find the index of the line?

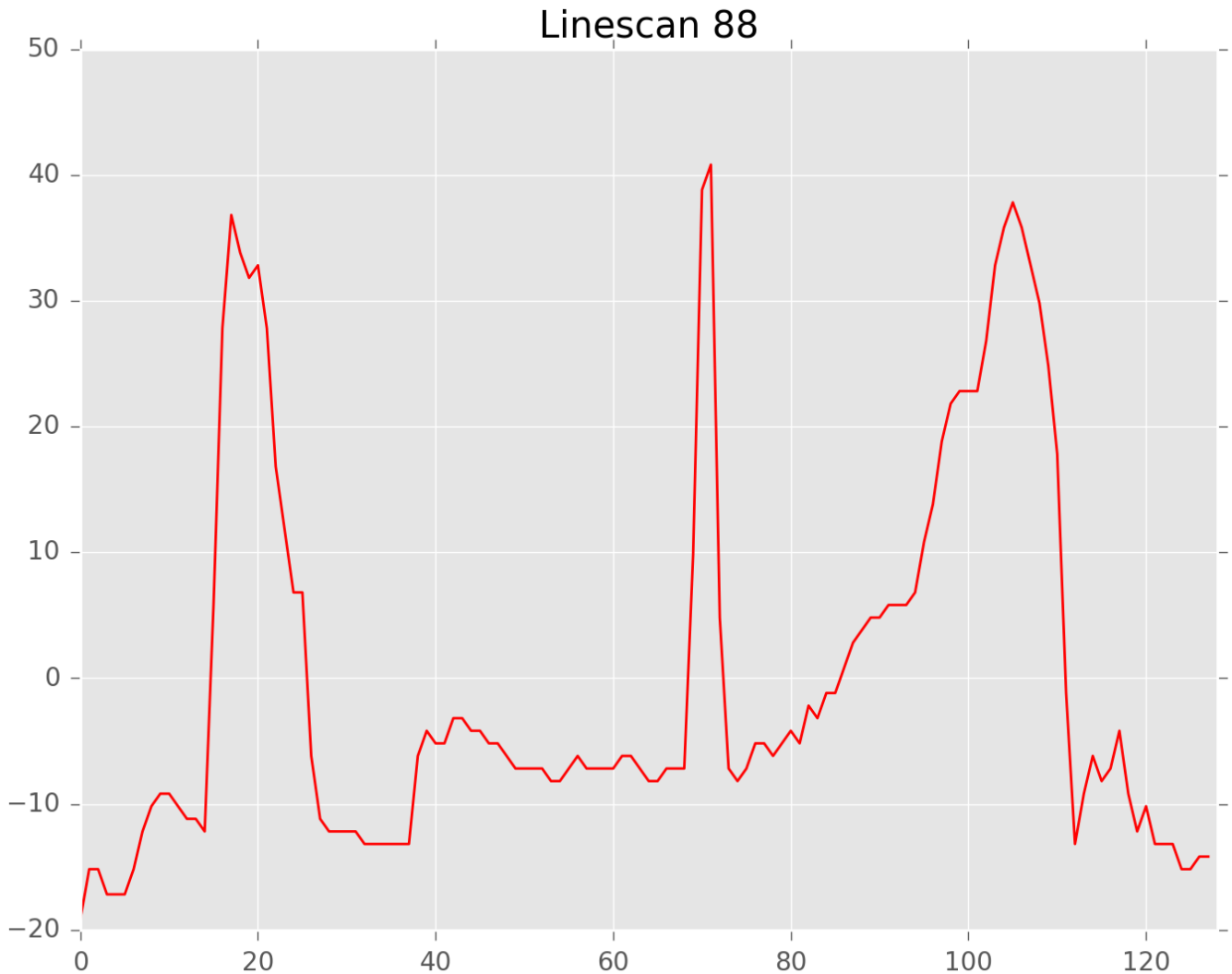
Take the argmax!



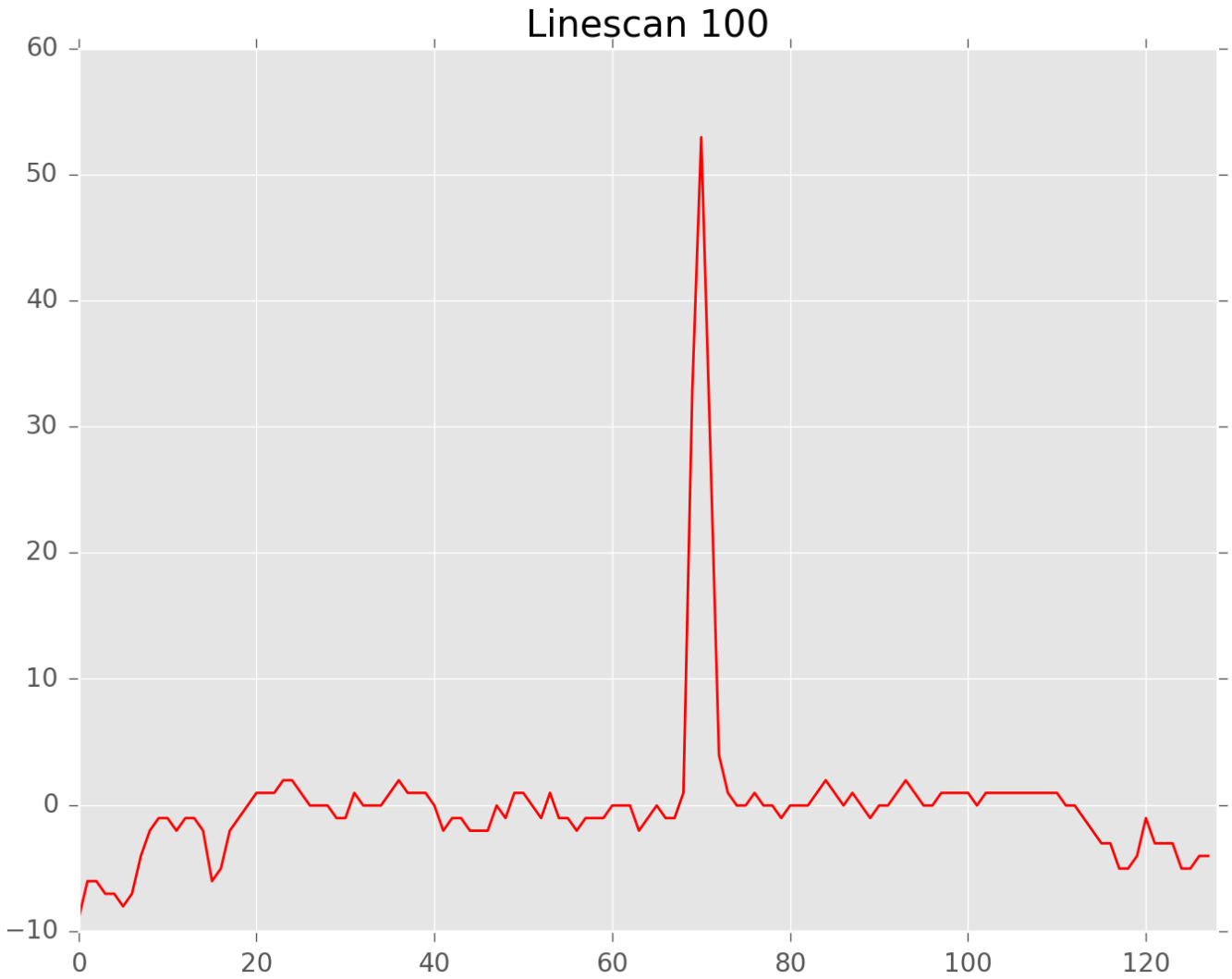
Any potential issues?



What about- noise, track crossings, other bright objects, etc.?

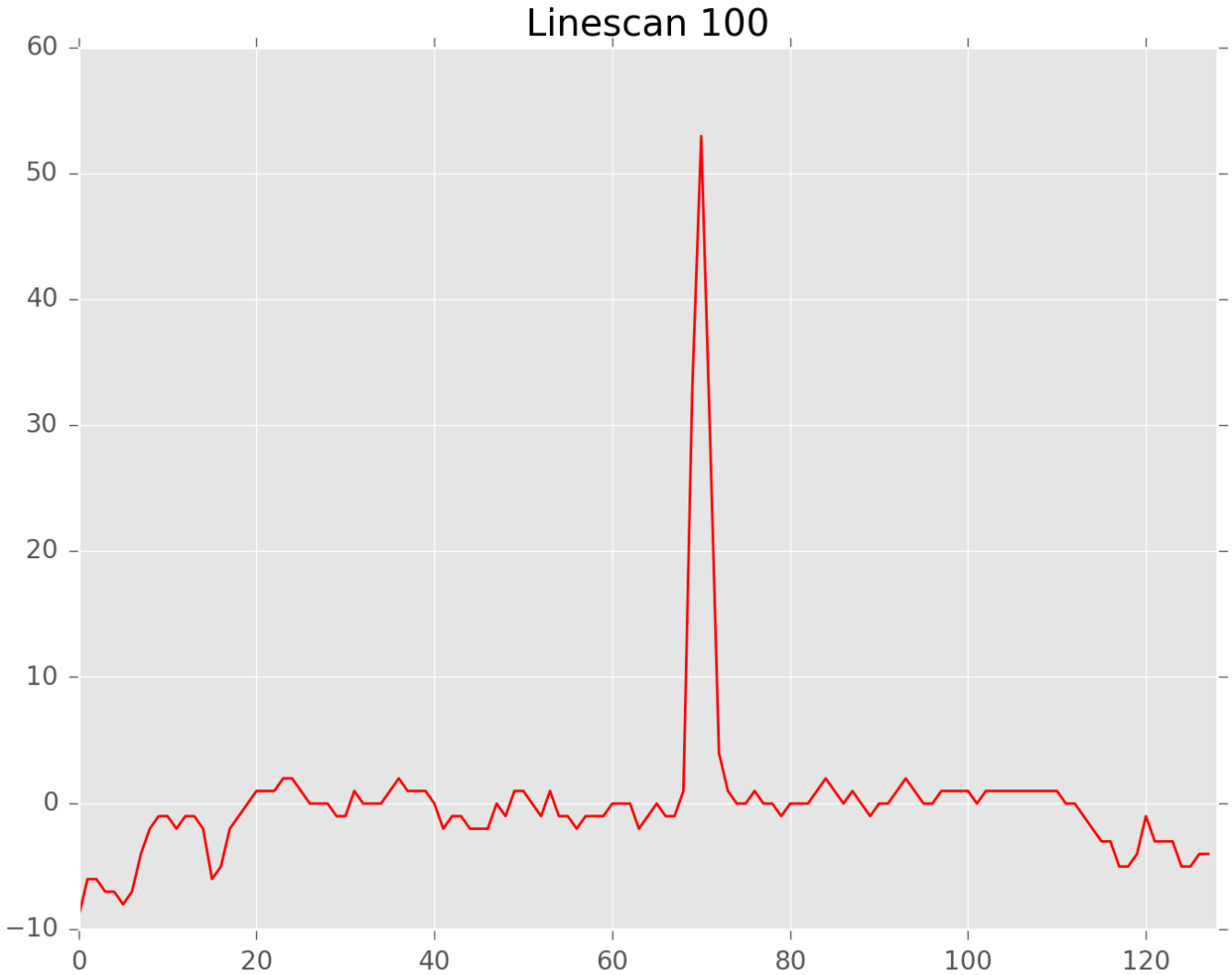


What about- noise, track crossings, other bright objects, multiple peaks, etc.?



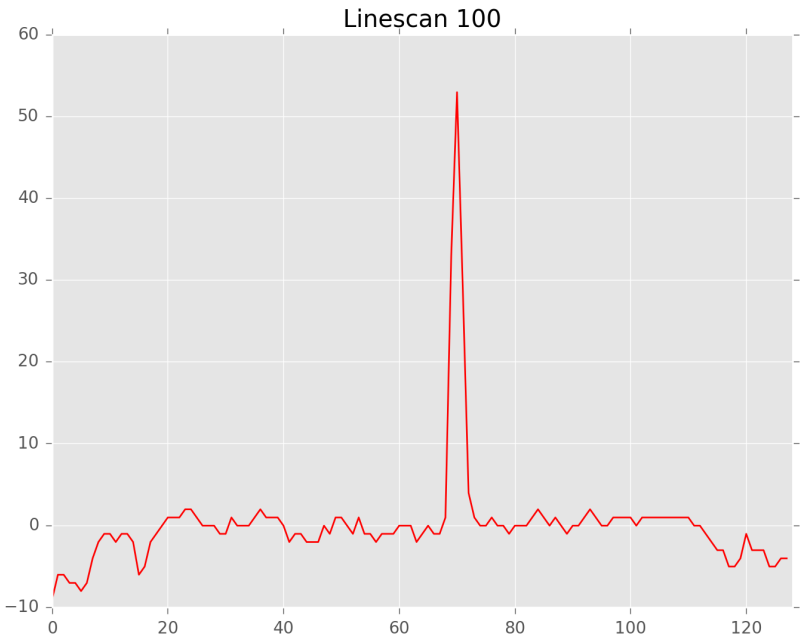
I(x,t) = Intensity at pixel x

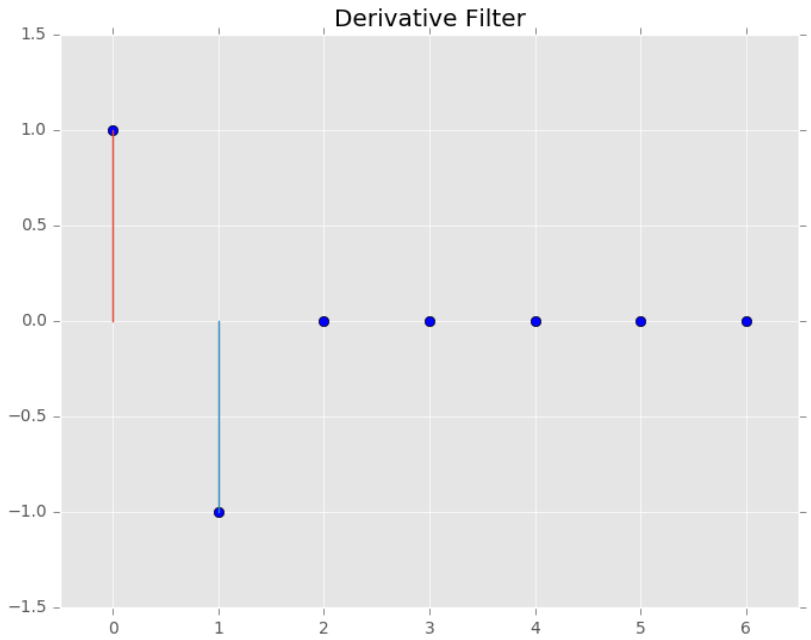
What if we look at the derivative $\frac{dI(x)}{dx}$?



$$\frac{dI(x)}{dx} \approx \frac{I(x+1)-I(x)}{1}$$

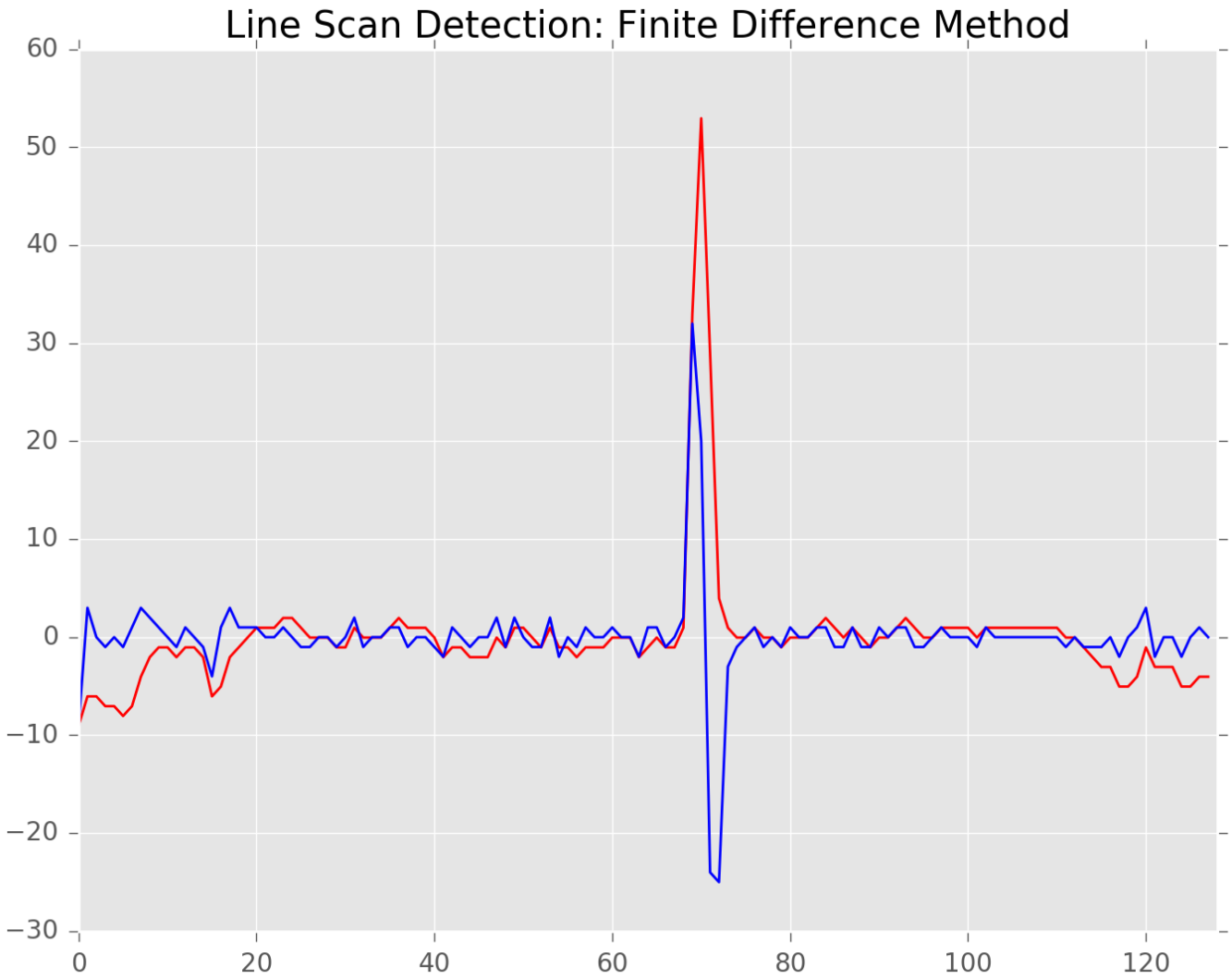
Convolution!





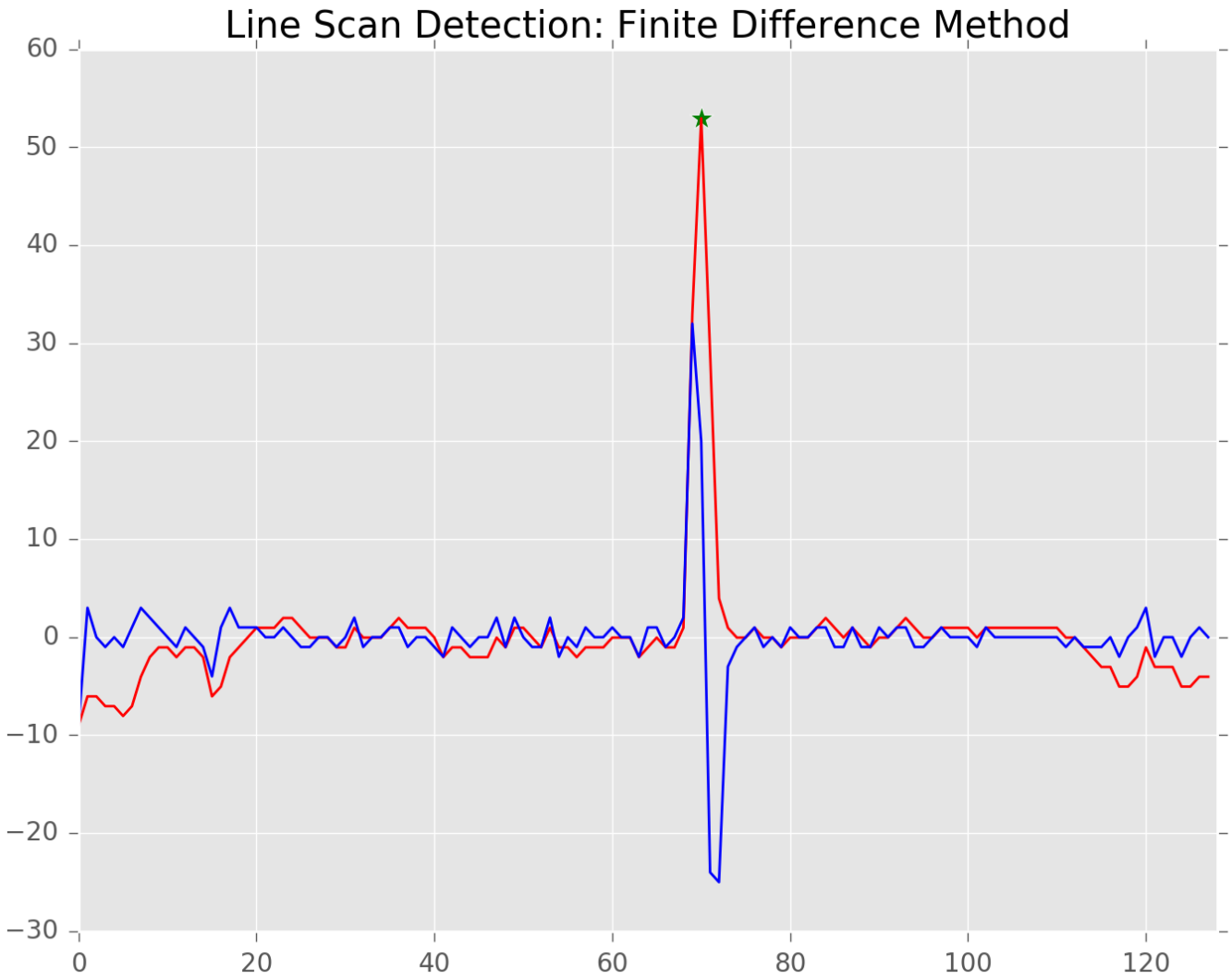
$$\dot{I}(x) = I(x+1) - I(x) = I * [1, -1]$$

Convolution!



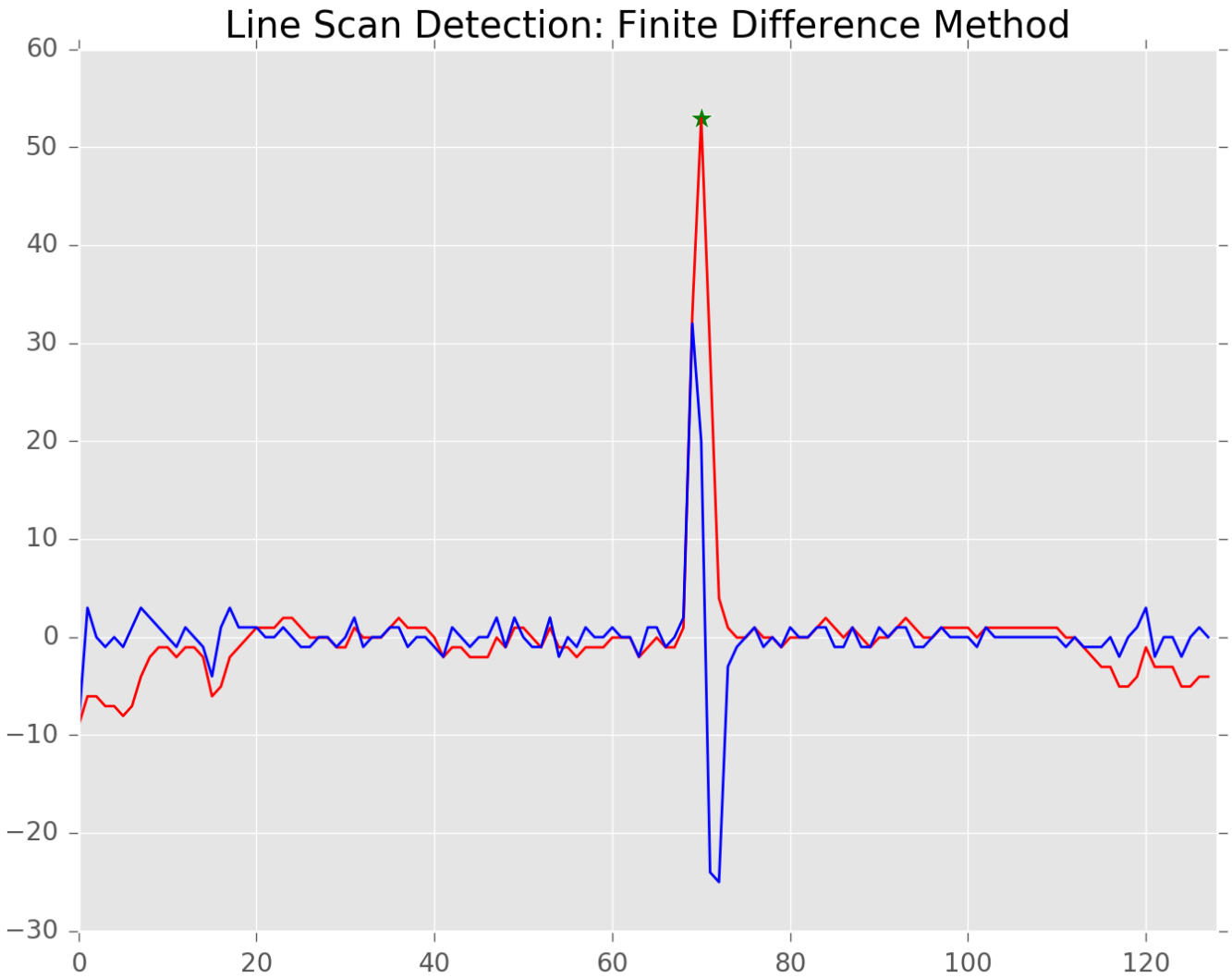
$$\dot{I}(x) = I(x+1) - I(x) = I * [1, -1]$$

Convolution!



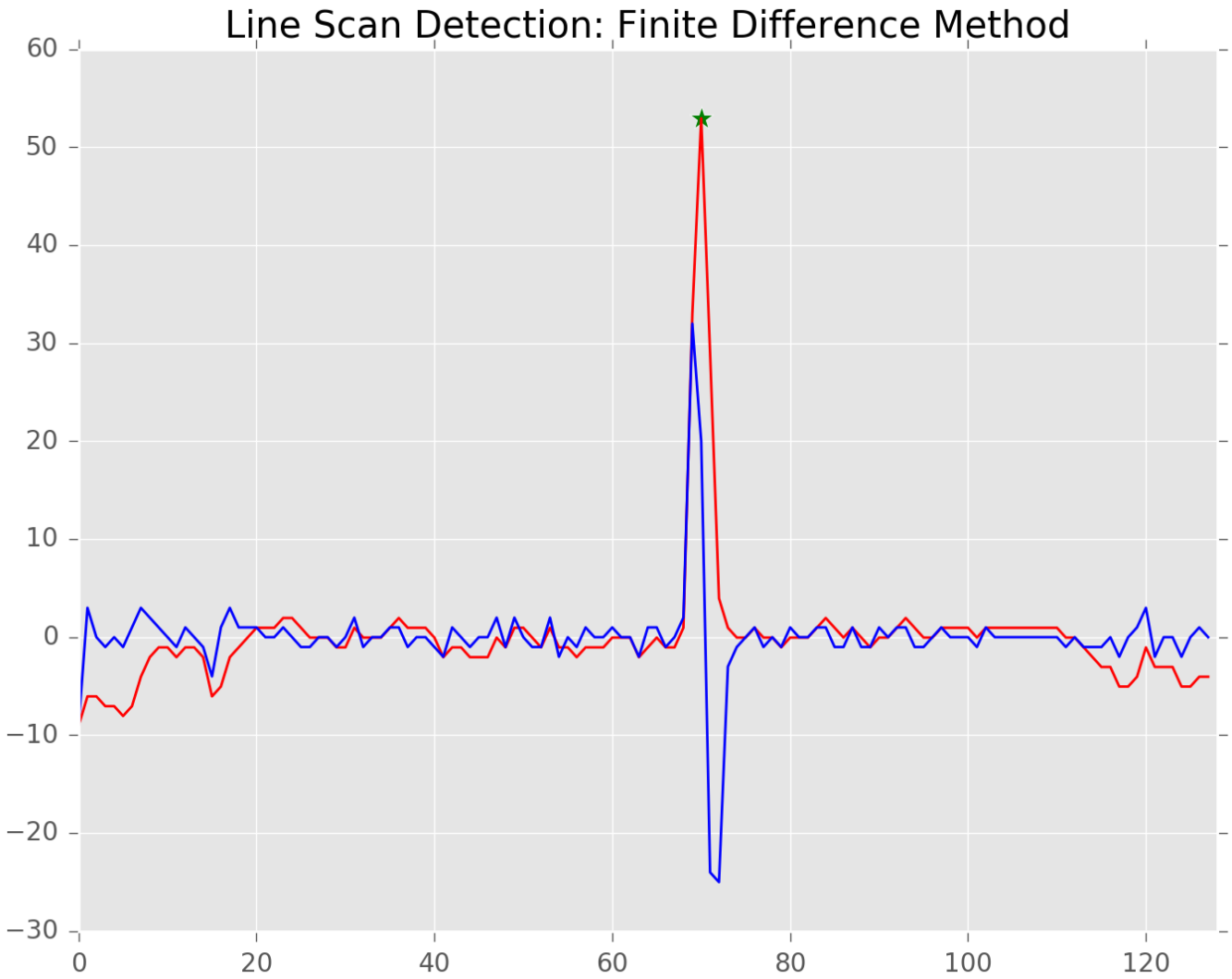
$$\dot{I}(x) = I(x+1) - I(x) = I * [1, -1]$$

Convolution!



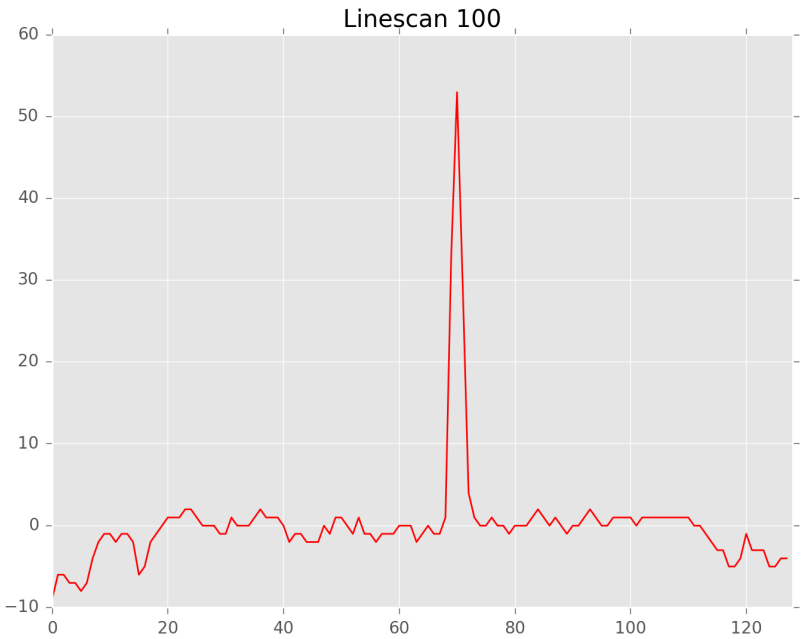
What about all that **high frequency** noise?

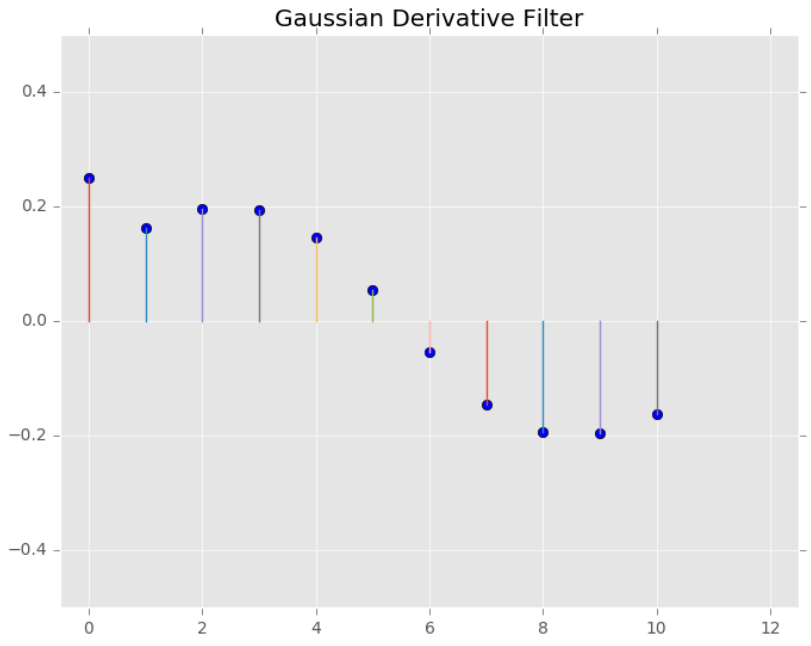
Low Pass Filter!



$$I(x) * LPF * [1, -1] = I(x) * (LPF * [1, -1])$$

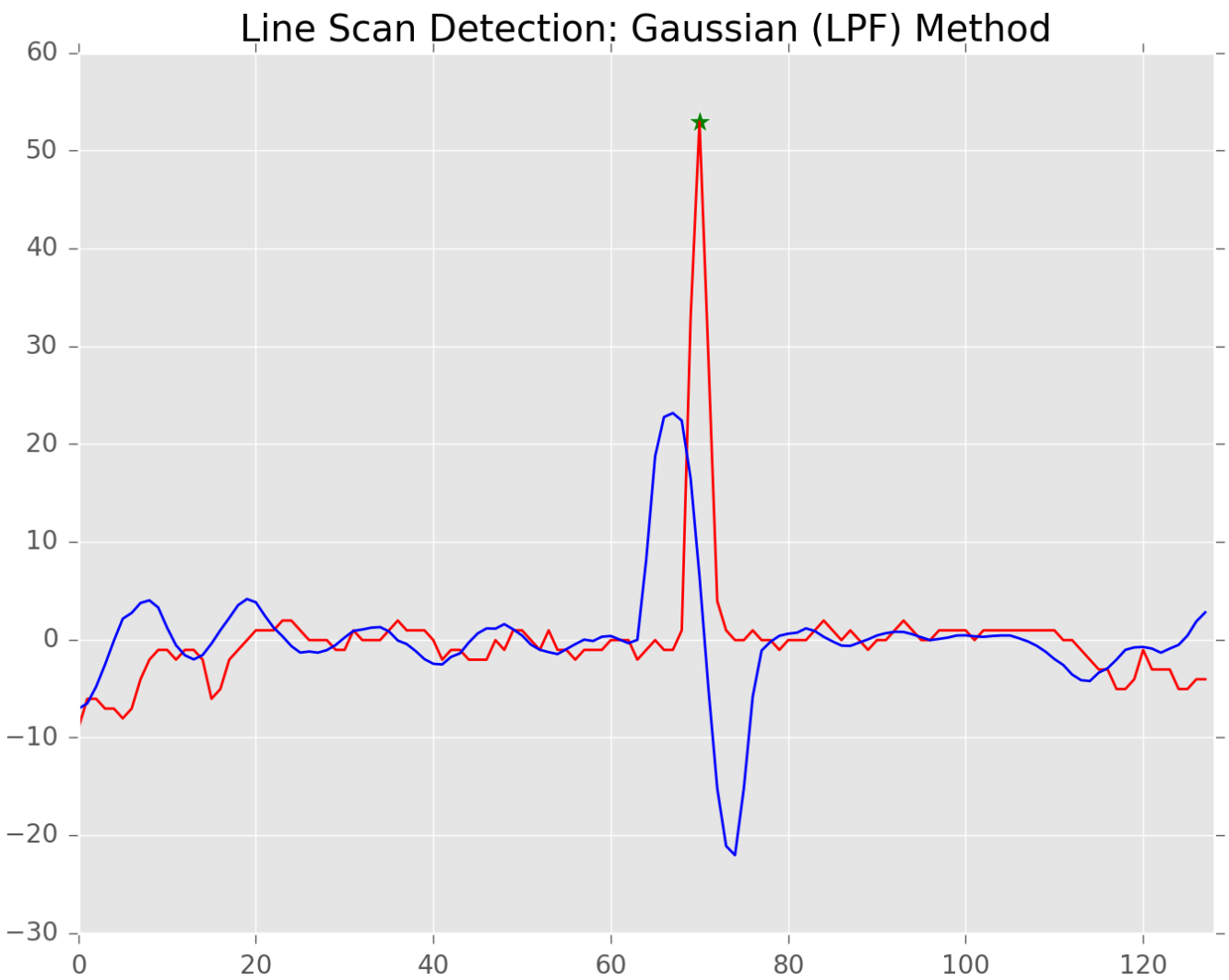
Low Pass Filter (Gaussian)!





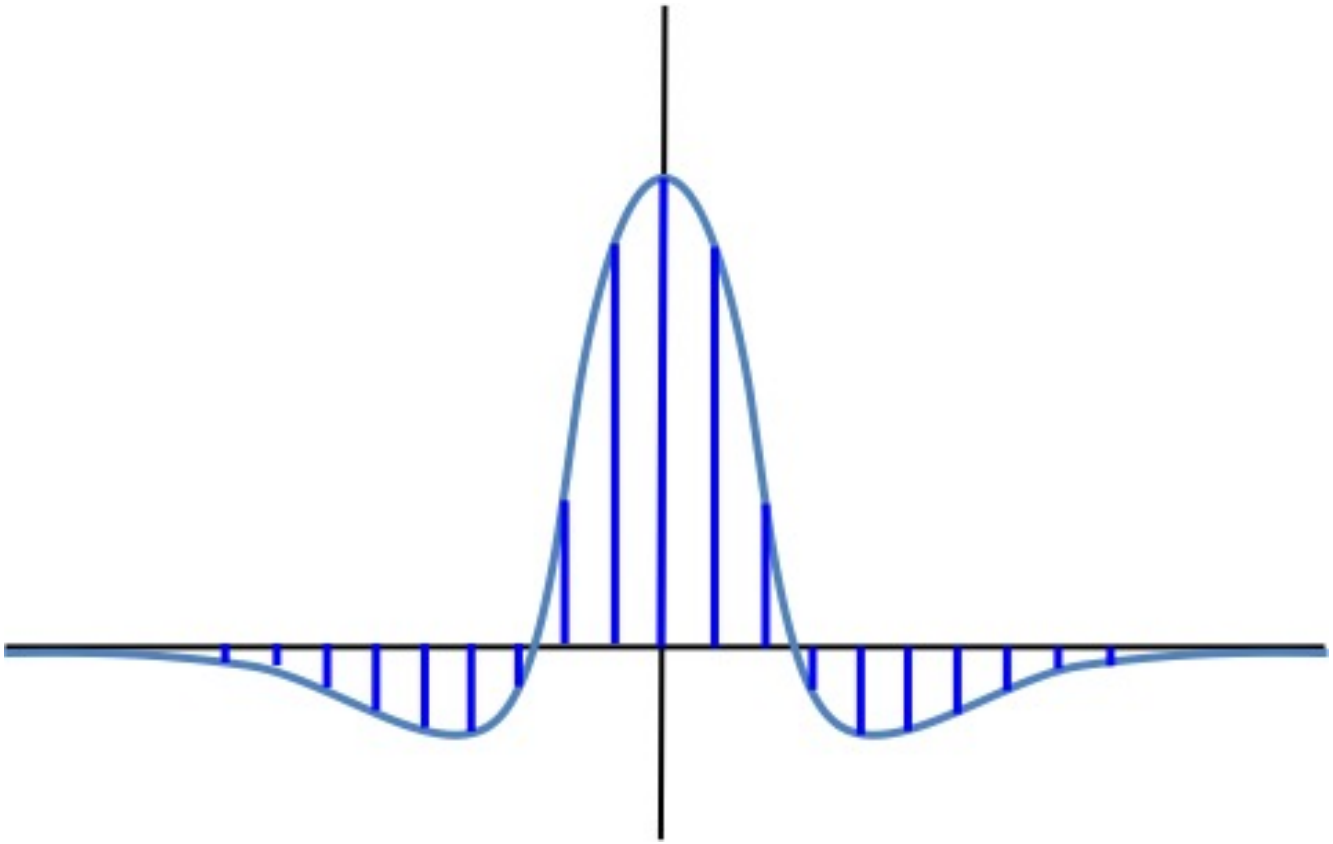
$$I(x) * LPF * [1, -1] = I(x) * (LPF * [1, -1])$$

Low Pass Filter (Gaussian)!



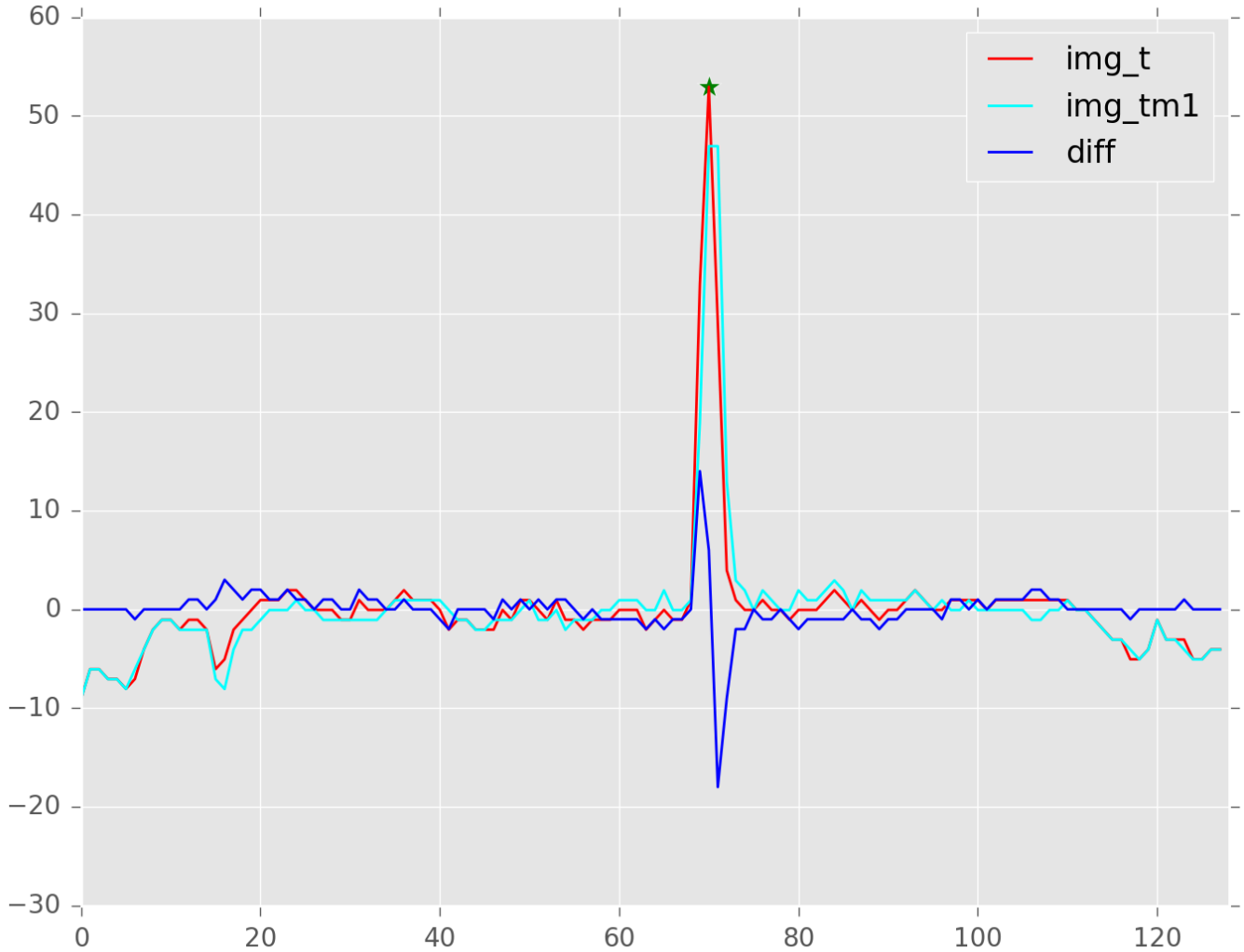
High frequency noise gone!

Laplacian of Gaussian (low-passed second derivative)
Can also be approximated with difference of Gaussians



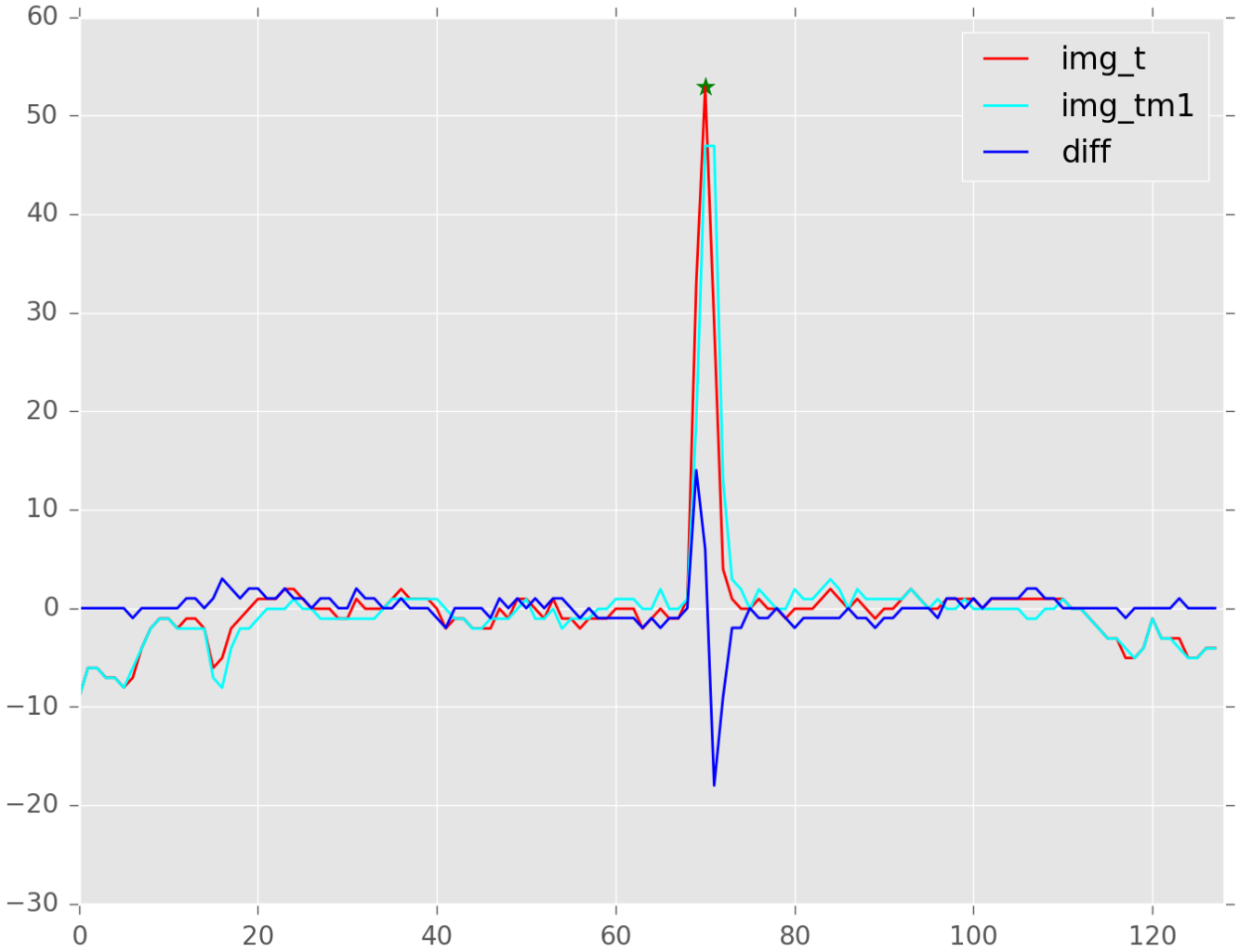
Detect a bright line of the expected width

What about... Temporal Differencing?



$$\frac{dI(x)}{dt} \approx \frac{I(t) - I(t-1)}{\Delta t}$$

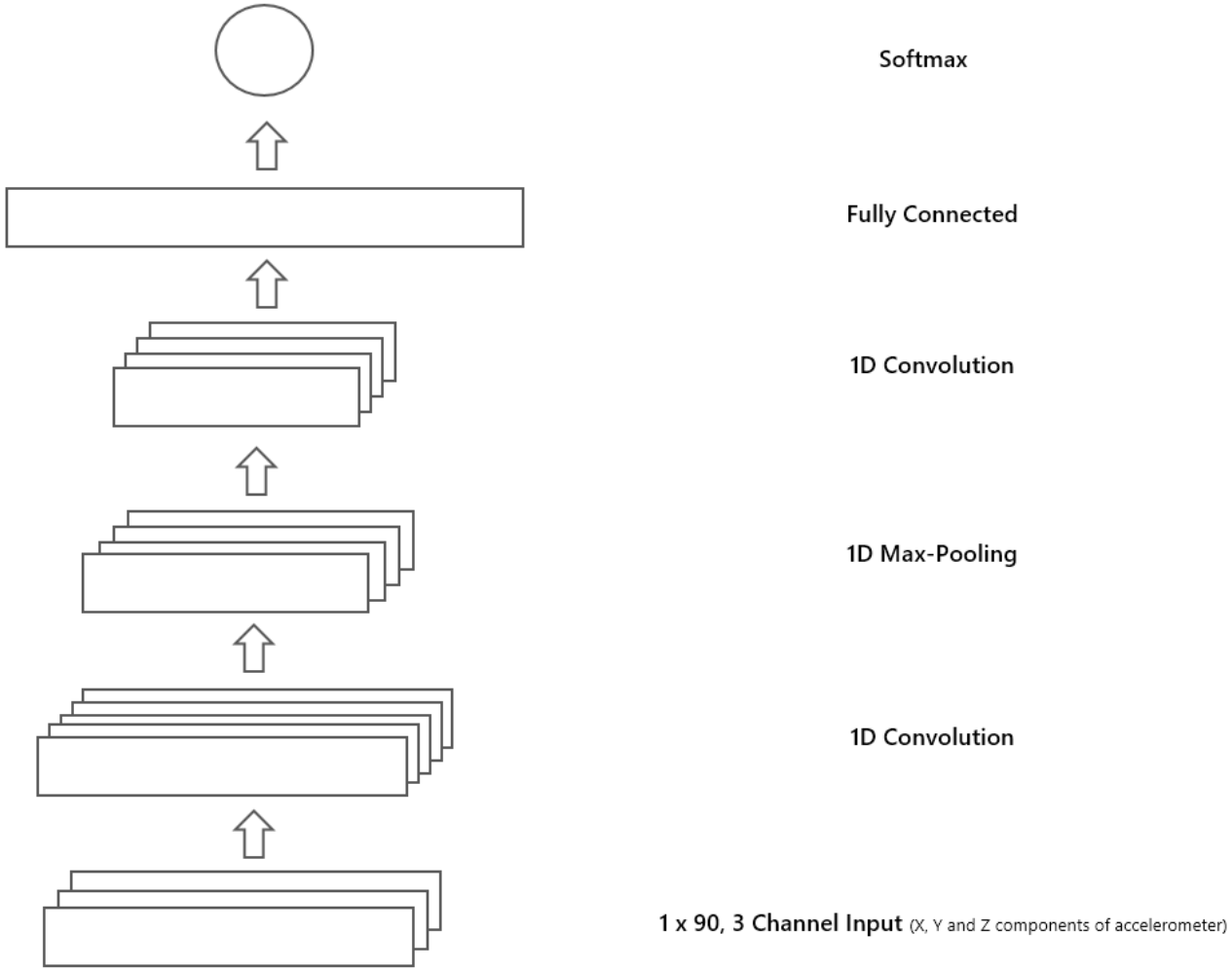
What about... Temporal Differencing?



Good for removing constant bias in parts of the image.

More Ideas!

- Difference of Gaussians (Bandpass- not just Lowpass)
- Cross Correlation with known signal
- Convolutional Neural Networks??



Potential Issues?

- What to do if there are multiple track crossings?
- Can't find the line
- Other Issues?

Solutions!

- Maintain a history (previous line pos, camera scan, etc.)
- Ignore the problem(s) and go fast!
- Anything else?

Embedded Software

Modularity

- Code is starting to get complicated
  - Pthreads, PRU, UART, GPIO, ADC etc.
- Many tasks to run on a single core CPU
  - Velocity Measurement
  - Line Finding
  - PID Controllers (steering and velocity)
  - Telemetry
  - Others?

General Tips

- Remove unecessary threads. This will make debugging easier later on.
- Build modular code, **test components in isolation**
- Minimize dynamic memory allocation aka malloc & free (memory fragmentation)
- Save logfiles
- Keep things simple!

Summary

- Many ways to do line detection (max, differencing, gaussian smoothing, temporal differencing, etc.)
- Camera calibration
- Embedded Software tips