Robotic Engineer - Entry/Mid Level

You have been assigned to write filters to reduce noise in the data coming from a LIDAR sensor attached to your robot. The LIDAR generates scans at a certain rate. Each scan is an array of length N of float values representing distance measurements. N is typically in a range of ~[200, 1000] measurements, and it is fixed. Measured distances are typically in a range of [0.03, 50] meters. Each time a scan is received, it will be passed on to the filters. Each filter object should have an **update** method, that takes a length-N array of ranges and returns a filtered length-N array of ranges.

We want you to write two different filter objects:

- A range filter

  The range filter crops all the values that are below range_min (resp. above range_max), and replaces them with the range_min value (resp. range_max)

- A temporal median filter

  The temporal median filter returns the median of the current and the previous D scans:

  $$y_i(t) = median(x_i(t), x_i(t-1), \dots , x_i(t-D))$$

  where x and y are input and output length-N scans and i ranges from 0 to N-1. The number of previous scans D is a parameter that should be given when creating a new temporal median filter. Note that, although the **update** method will receive a single scan, the returned array depends on the values of previous scans. Note also that the for the first D scans, the filter is expected to return the median of all the scans so far.

  Here is a short example of the result (Y) of a temporal median filter object with D=3 for an input (X) of dimension N=5, for the first five updates:

T (time)	X (input scan)	Y (return of the update)
0	[0., 1., 2., 1., 3.]	[0., 1., 2., 1., 3.]
1	[1., 5., 7., 1., 3.]	[0.5, 3. , 4.5, 1. , 3.]
2	[2., 3., 4., 1., 0.]	[1., 3., 4., 1., 3.]
3	[3., 3., 3., 1., 3.]	[1.5, 3. , 3.5, 1. , 3.]
4	[10., 2., 4., 0., 0.]	[2.5, 3. , 4. , 1. , 1.5]

You are expected to write documentation and test correctness for your code.

You can either use Python 2.7 and/or C++. For Python, Numpy library may be used. For C++, boost and stl libraries may be used.