Speed Test (filter() vs for loop)

Using filter() instead of a for loop for computing averages (or pretty much any other "moving" quantity) is highly recommended. The only issue that needs to be examined is for the case of computing the average in non-overlapping windows. Filter() is faster computationally, but also has to perform many redundant calculations, while the for loop will only do the operations necassary for the end result. The level of reduntancy for the filter() is dependent on the size of the averaging window (smaller windows means less reduntancy, so faster computation time using filter(), while for larger windows probably a for loop will do the job faster).

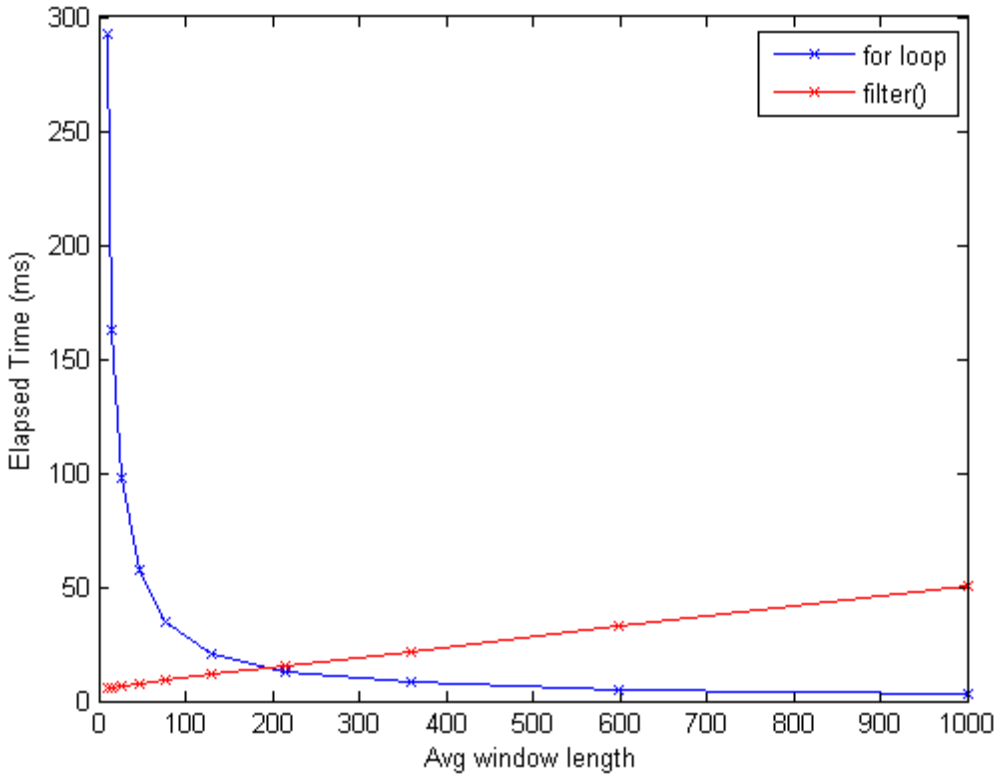
This scenario is examined below.

```
L = 100000;
N = 10;
blockSizes = fix(logspace(log10(10),log10(1000),N));
x = rand(L,1);
t = zeros(N,2);

for s = 1:N
    blockSize = blockSizes(s);
    tic
    % block average
    y = zeros(fix(L/blockSize),1);
    for i=(1:blockSize:L-blockSize)
        y(fix(i/blockSize) + 1) = mean(x(i:i+blockSize-1));
    end
    t(s,1) = toc;

    tic
    % filter avg
    b = ones(blockSize,1)/blockSize;
    z = filter(b,1,x);
    z = z(blockSize:blockSize:L);
    t(s,2) = toc;
end

plot(blockSizes,t(:,1)*1000,'-xb',blockSizes,t(:,2)*1000,'-xr');
xlabel('Avg window length');
ylabel('Elapsed Time (ms)');
legend('for loop','filter()','Location','NorthEast');
```



so indeed, for small window sizes, the use of filter() leads to considerably faster computation times. As window size increases, the use of a simple for loop becomes the optimal solution.

Published with MATLAB® 8.0