```
function [gaussianFilterList] = spacial2DGaussianFilter(imgList,
 ssigma)
%UNTITLED5 Summary of this function goes here
%   Detailed explanation goes here
    Gx = @(x) 1 / sqrt(2 * pi * (ssigma^2)) * exp(-1 * x^2 / (2 *
 (ssigma^2)));
    Gy = @(y) 1 / sqrt(2 * pi * (ssigma^2)) * exp(-1 * y^2 / (2 *
 (ssigma^2)));
    imgListSize = size(imgList);
    xDir = imgListSize(2);
    yDir = imgListSize(1);
    zDir = imgListSize(3);

    boxSize = ceil(((ssigma * 5) + 1)/2)*2 - 1; % gets an odd box size

    startPixel = boxSize - floor(boxSize/2);
    endPixelX = xDir - floor(boxSize/2);
    endPixelY = yDir - floor(boxSize/2);
    gaussianFilterList = imgList;

    gaussianFilterHorizontal = zeros(1, boxSize);
    for i = 1:boxSize
        x = i - ceil(boxSize/2);
        gaussianFilterHorizontal(i) = Gx(x);
    end
    gaussianFilterVertical = transpose(gaussianFilterHorizontal);
    gaussianFilterSumX = sum(gaussianFilterHorizontal);
    gaussianFilterSumY = sum(gaussianFilterVertical);

    gaussianFilterSum = gaussianFilterSumX * gaussianFilterSumY;

    for k = 1:zDir
        gaussianFilterList(startPixel:endPixelY,
 startPixel:endPixelX, k) = (1/gaussianFilterSum) *
 (conv2(conv2(imgList(:,:,k), gaussianFilterHorizontal, 'valid'),
 gaussianFilterVertical, 'valid'));
    end
end
```

Published with MATLAB® R2017a