```
function [strainX,strainY,strainS,dispXSSmooth,dispYSSmooth] =...
    func_smoothCalcStrain_NS(dispX,dispY,posXGrid,posYGrid,spatKern,↙
spatPadOpt)
% Author: Lloyd Fletcher
% PhotoDyn Group, University of Southampton
% Date Created: 18/9/2019
% Date Edited: 18/9/2019

% Get the dimensions of the input fields
[~,~,st] = size(dispX);

% Calculate the standard deviation of the Gaussian smoothing filter
spatSigma = ceil((spatKern-1)/2)/2;

% Only smooth if the specified kernel is greater than 0
if spatKern > 0
    % Loop over each frame and smooth it in space
    for ff = 1:st
        dispXSSmooth(:,:,ff) = imgaussfilt(dispX(:,:,ff),...
            spatSigma,'Padding',spatPadOpt);
        dispYSSmooth(:,:,ff) = imgaussfilt(dispY(:,:,ff),...
            spatSigma,'Padding',spatPadOpt);
    end
else
    dispXSSmooth = dispX;
    dispYSSmooth = dispY;
end

% Calculate the 'step' for spatial derivative calculation
xStep = posXGrid(1,2) - posXGrid(1,1);
yStep = posYGrid(2,1) - posYGrid(1,1);

% Derivatives of Ux to get strains
[strainX, epsxy1,~]=gradient(dispXSSmooth,xStep,yStep,1);
% Derivatives of Uy to get strains
[epsxy2, strainY,~]=gradient(dispYSSmooth,xStep,yStep,1);
% Combine values for the shear strain
strainS = (epsxy1 + epsxy2);

end
```