Compute 3D decorrelation

```
function compute3DDecorr_sliceMethod( obj )

% COMPUTE3DDECORR Summary of this function goes here
%   Detailed explanation goes here

    %
```

Define Guassian Window

```
    x_range_length = size(obj.x_range,2);
    y_range_length = size(obj.y_range,2);
    z_range_length = size(obj.z_range,2);
    sigx = obj.windowSigma/obj.dx;
    sigy = obj.windowSigma/obj.dy;
    sigz = obj.windowSigma/obj.dz;
    x_mid = ceil(x_range_length/2);
    y_mid = ceil(y_range_length/2);
    z_mid = ceil(z_range_length/2);
    sigfaz = x_range_length/(2*pi*sigx);
    sigfra = x_range_length/(2*pi*sigy);
    sigfel = x_range_length/(2*pi*sigz);
    %coeffX =    1/sqrt(2*pi*sigx^2);
    %coeffY =    1/sqrt(2*pi*sigy^2);
    %coeffZ =    1/sqrt(2*pi*sigz^2);
%     xmask = coeffX*exp(-(((1:vol_x_length)-x_mid).^2)/(2*sigx^2));
%     ymask = coeffY*exp(-(((1:vol_y_length)-y_mid).^2)/(2*sigy^2));
%     zmask = coeffZ*exp(-(((1:vol_z_length)-z_mid).^2)/(2*sigz^2));
%     azMask   = filtFactAz .* exp(-((1:nSigPad)-azId).^2/2/sigFAz^2);
%     raMask   = filtFactRa .* exp(-((1:nRowPad)-rangeId).^2/2/
sigFRa^2);
%     [am,rm]  = meshgrid(azMask,raMask);
%     maskFilt = fftshift(am.*rm);
    xmask = exp(-(((1:x_range_length)-x_mid).^2)/2/sigfaz^2);
    ymask = exp(-(((1:y_range_length)-y_mid).^2)/2/sigfra^2);
    zmask = exp(-(((1:z_range_length)-z_mid).^2)/2/sigfel^2);

    [x_mask_mat,y_mask_mat,z_mask_mat] = ndgrid(xmask,ymask,zmask);
    %maskfilt = fftshift(x_mask_mat.*y_mask_mat.*z_mask_mat);
    %maskfilt = maskfilt/sum(maskfilt(:));
    maskfilt = (fftshift(x_mask_mat.*y_mask_mat.*z_mask_mat));
    maskfilt = maskfilt/sum(maskfilt(:));
    size(maskfilt)
    % *compute windowed ibs and autocorr01*
    %compute ibs and autocorr before windowing
    obj.ibs_slicemethod = abs(obj.rawData_cart_slicemethod).^2;
    obj.autocorr01_slicemethod = obj.rawData_cart_slicemethod(:,:,:,1:
(end-1)).*conj(obj.rawData_cart_slicemethod(:,:,:,2:end));
    % set NaN values to small number

 obj.autocorr01_slicemethod(find(isnan(obj.autocorr01_slicemethod))) =
 realmin('double');
```

```
    obj.ibs_slicemethod(find(isnan(obj.ibs_slicemethod))) =
 realmin('double');
    %compute windowed ibs
    for currVolume = 1:size(obj.ibs_slicemethod,4)
        obj.ibs_slicemethod(:,:,:,currVolume) =
 abs(ifftn(fftn(obj.ibs_slicemethod(:,:,:,currVolume)).*maskfilt));
    end
    %compute autcorrelation and decorrelation
    for currVolume = 1:(size(obj.ibs_slicemethod,4)-1)
        obj.autocorr01_slicemethod(:,:,:,currVolume) =
 abs(ifftn(fftn(obj.autocorr01_slicemethod(:,:,:,currVolume)).*maskfilt));
    end
    for currVolume = 1:(size(obj.ibs_slicemethod,4)-1)
        %obj.decorr_slicemethod(:,:,:,currVolume) = (1 -
 abs(obj.autocorr01_slicemethod(:,:,:,currVolume)).^2./
(obj.ibs_slicemethod(:,:,:,currVolume).*obj.ibs_slicemethod(:,:,:,currVolume
+1)))./obj.interFrameTime;
        R00 = obj.ibs_slicemethod(:,:,:,currVolume);
        R11 = obj.ibs_slicemethod(:,:,:,currVolume+1);
        B2 = R00.*R11;
        R01 = abs(obj.autocorr01_slicemethod(:,:,:,currVolume)).^2;
        %thisMean = mean(abs(obj.autocorr01_slicemethod(:)));
        tau = obj.interFrameTime;
        obj.decorr_slicemethod(:,:,:,currVolume) = 2*(B2-R01)./(B2 +
 mean(B2(:)))/tau;
        %obj.decorr_slicemethod(:,:,:,currVolume) = (1 -
 abs(obj.autocorr01_slicemethod(:,:,:,currVolume)).^2./(thisMean
+obj.ibs_slicemethod(:,:,:,currVolume).*obj.ibs_slicemethod(:,:,:,currVolume
+1)))./;
        %obj.decorr_slicemethod(:,:,:,currVolume) = 2*((B2 -
 (obj.autocorr01_slicemethod)).^2./(obj.interFrameTime*(mean(B2(:)) +
 B2)));
    end
    % set values outside of volume to small number

 obj.autocorr01_slicemethod(find(isnan(obj.rawData_cart_slicemethod(:,:,:,1:
(end-1))))) = realmin('double');
    obj.ibs_slicemethod(find(isnan(obj.rawData_cart_slicemethod))) =
 realmin('double');

 obj.decorr_slicemethod(find(isnan(obj.rawData_cart_slicemethod(:,:,:,1:
(end-1))))) = realmin('double');

end
```

Published with MATLAB® R2016a