```
% CS194-26 (cs219-26): Project 1, starter Matlab code

% name of the input file
imname = 'fruits.jpg';

% read in the image
fullim = imread(imname);

% convert to double matrix (might want to do this later on to same
memory)
fullim = im2double(fullim);

% compute the height of each part (just 1/3 of total)
height = floor(size(fullim,1)/3);
% separate color channels
B = fullim(1:height,:);
G = fullim(height+1:height*2,:);
R = fullim(height*2+1:height*3,:);

% Align the images
% Functions that might be useful to you for aligning the images
include:
% "circshift", "sum", and "imresize" (for multiscale)
%aG = sum(G,B);
%aR = sum(R,B);

% Blur each image separately before shrinking
% % B
integralI = integralImage(B);
avgh = integralKernel([1 1 7 7], 1/49);
Z = integralFilter(integralI, avgh);

% % G
integralJ = integralImage(G);
meanh = integralKernel([1 1 7 7], 1/49);
Y = integralFilter(integralJ, meanh);
% % R
integralK = integralImage(R);
avh = integralKernel([1 1 7 7], 1/49);
X = integralFilter(integralK, avh);

% Resize images by facotr of 2
newB = imresize(Z, 0.5);
newG = imresize(Y, 0.5);
newR = imresize(X, 0.5);

% %
% Implementation of 3D alignment algorithm
% %

% G and R
% Find highest correlation between G and R slices
C = normxcorr2(newG,newR);
[row col] = find(C == max(max(C)))

[height width] = size(newG)
```

```matlab
% Offset accounts for the padding given by normxcorr2
offsetX = 2*(width - col)
offsetY = 2*(height - row)

G2 = zeros(size(newG));
for n=1:height
    if n-offsetY>0 & n-offsetY<height
        G2(n-offsetY,:) = newG(n,:);
    end
end

G3 = zeros(size(newG));
for n=1:height
    if n-offsetX>0 & n-offsetX<height
        G3(n-offsetX,:) = G2(n,:);
    end
end


newColorImage(:,:,1) = newR;
newColorImage(:,:,2) = G2;


% B and R
C = normxcorr2(newR,newB);
[row col] = find(C == max(max(C)))

[height width] = size(newB)





offsetX = 2*(width - col)
offsetY = 2*(height - row)

B2 = zeros(size(newB));
for n=1:height
    if n+offsetY>0 & n+offsetY<height
        B2(n+offsetY,:) = newB(n,:);
    end
end

B3 = zeros(size(newB));
for n=1:height
    if n+offsetX>0 & n+offsetX<height
        B3(n+offsetX,:) = B2(n,:);
    end
end

aligned_field(:,:,1) = newR;
aligned_field(:,:,2) = G3;
aligned_field(:,:,3) = B3;
```

```
aligned_field = aligned_field(20:height-10,25:width-20,:);
imshow(aligned_field);
```