Mulitscale Optical Flow Estimation: Image warp

In this project you will extend your previous optical flow solution to a multiscale approach. A multiscale approach allows motions of various magnitudes to be estimated with the same kernel size.

To complete this lab you will (1) write a function to warp an image according to the optical flow estimate and (2) incorporate the image warping function in a pyramidal (multiscale) optical flow estimation function.

Your Script

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
[I1, I2, I3, I4] = test_images();

u = ones(size(I1));
v = ones(size(I1))*2;

warp = warp_image(I1,u,v);
figure()
imagesc(I1)
figure()
imagesc(warp)

function warp = warp_image(I,u,v)
    %% INPUT:
    %% I: image to be warped
    %% u,v: x and y displacement
    %% OUTPUT:
    %% warp: image I deformed by u,v

    % initialize warp as zeros
    warp = zeros(size(I));
    % construct warp so that warp(x,y) = I(x + u, y + v)
    [r c] = size(I);

    % see https://www.mathworks.com/company/newsletters/articles/matrix-indexing-in-matlab.html
    x = 1:c;
    y = 1:r;
    [X, Y] = meshgrid(x,y);
    % add u to x, v to y
    Xq = round(X + u);
    Yq = round(Y + v);

    warp = interp2(X,Y,I,Xq,Yq);
    warp(isnan(warp)) = 0;

end

```

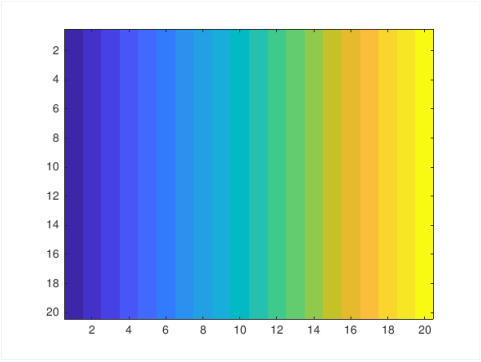
Run Script

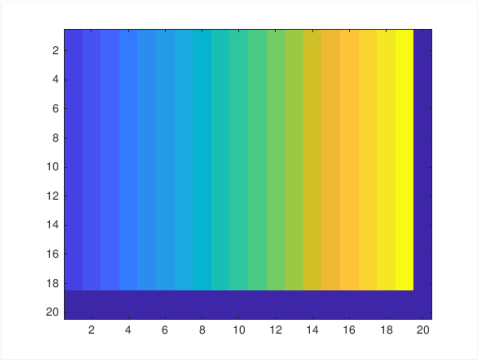
Previous Assessment: All Tests Passed

Submit

Is warp estimate correct?

Output


2
4
6
8
10
12
14
16
18
20
2
4
6
8
10
12
14
16
18
20


2
4
6
8
10
12
14
16
18
20
2
4
6
8
10
12
14
16
18
20