Optical Flow: Estimate Displacement

The perception of motion and the subsequent formation of an interpretation guides our everyday lives. The ability to determine if an object is moving, judge its speed and direction, and react accordingly is fundamental to our survival. The apparent motion which guides our actions is called optical flow. Since optical flow is determined by time varying image intensities, it is not always consistent with the true motion of objects and surfaces called the motion field. Motion estimation also plays a critical role in a variety of computer vision tasks. While applications such as object tracking, scene reconstruction and image alignment have very different objectives, they all rely to some degree on low-level motion cues.

In this lab you will estimate the optical flow between a pair of images via the implementation sketch in video .... In this section you will estimate the displacement of a windowed region by computing the least squares solution over several pixels. In the final section you will combine this and the prebious solution to estimate the optical flow estimate over the entire image.

test_image.m (http://lcms-prod-repo.mwcloudtest.com.s3.amazonaws.com/content/file/375f18fa-8823-4130-b2cb-549296681d1e/test_images.m?versionId=PtqA9wvQb0cHb.kpn_L8bk_YQBxwVvjb)

Your Script

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

```
[I1, I2, I3, I4] = test_images();
[I_x, I_y] = grad2d(I2);
I_t = I1-I2;

I_x = I_x(:,2:end-1);
I_y = I_y(:,2:end-1);
I_t = I_t(:,2:end-1);

d = estimate_displacement(I_x,I_y,I_t);

function d = estimate_displacement(Ix,Iy,It)
    %% INPUT:
    %% Ix, Iy, It: m x m matrices, gradient in the x, y and t directions
    %% Note: gradient in the t direction is the image difference
    %% OUTPUT:
    %% d: least squares solution

    b = [ Ix(:) Iy(:) ]' * It(:);
    A = [ Ix(:) Iy(:) ]' * [ Ix(:) Iy(:) ];

    % to help mitigate effects of degenerate solutions add eye(2)*eps to the 2x2 matrix A
    % add eps value
    A = A + eye(2)*eps;

    % use pinv(A)*b to compute the least squares solution
    d = pinv(A)*b;
end

function [I_x,I_y] = grad2d(img)
        %% compute image gradients in the x direction
        %% convolve the image with the derivative filter from the lecture
        %% using the conv2 function and the 'same' option
        dx_filter = [1/2 0 -1/2];
        I_x = conv2(img, dx_filter, 'same');

        %% compute image gradients in the y direction
        %% convolve the image with the derivative filter from the lecture
        %% using the conv2 function and the 'same' option
        dy_filter = dx_filter';
        I_y = conv2(img, dy_filter, 'same');
end

function smooth = gauss_blur(img)
    %% Since the Gaussian filter is separable in x and y we can perform Gaussian smoothing by
    %% convolving the input image with a 1D Gaussian filter in the x direction then
    %% convolving the input image with a 1D Gaussian filter in the x direction then
    %% convolving the output of this operation with the same 1D Gaussian filter in the y direction.

    %% Gaussian filter of size 5
    %% the Gaussian function is defined f(x) = 1/(sqrt(2*pi)*sigma)*exp(-x.^2/(2*sigma))
    x = -2:2;
    sigma = 1;
    gauss_size = [1 5];

    % my soln: I use fspecial('gaussian', hsize = [1 5], sigma)
    %gauss_filter_x = fspecial('gaussian', gauss_size, sigma);
    %gauss_filter_y = fspecial('gaussian', gauss_size', sigma);
    %smooth_x = imfilter(img, gauss_filter_x);
    %% convolve smooth_x with the transpose of the Gaussian filter
    %smooth = imfilter(smooth_x, gauss_filter_y);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % for edX class:
    gauss_filter = 1/(sqrt(2*pi)*sigma)*exp(-x.^2/(2*sigma^2));

    %% using the conv2 function and the 'same' option
    %% convolve the input image with the Gaussian filter in the x
    smooth_x = conv2(img, gauss_filter, 'same');
    %% convolve smooth_x with the transpose of the Gaussian filter
    smooth = conv2(smooth_x, gauss_filter', 'same');
end

```

▶ Run Script ?

Assessment: All Tests Passed

Submit ?

✓ **Is the estimated displacement correct?**

Output

```
Code ran without output.
```