CSC 514: Computer Vision: Homework #6

Due on Thursday, October 22, 2015

Hoover 11:00am

Julian A. Brackins

Problem 1

In class we discussed several different methods for performing filtering/template matching on images. In this homework you will create your own Matlab functions to perform both convolution and correlation on images using different convolution/correlation kernels.

Problem 1 continued on next page. . .

Listing 1: Where's Waldo Script

```
function wheres_waldo()
    %%Read in Kernel, convert to grayscale double. Save original copy.
    kernel = imread('WaldoKernel.png');
    orig_kern = kernel;
    kernel = rgb2gray(kernel);
    kernel = im2double(kernel);

    %%Read in Scene, convert to grayscale double. Save original copy.
    scene = imread('WaldoScene.png');
    orig_scene = scene;
    scene = rgb2gray(scene);
    scene = im2double(scene);

    %%Find the dimensions for the scene and kernel
    [kH, kW] = size(kernel);
    [sH, sW] = size(scene);

     %%Divide Kernel Height and Width
    kH = kH/2;
    kW = kW/2;

    %%Calculate Correlation
    G =  correlation( scene, kernel );
    %%OR you can do convolution...
    %%G =  convolution( scene, kernel );
    %%Actually don't because convlution is trash for this program...

    %%Find Waldo by finding the highest value point
    [r,c] = find(G==max(G(:)));

    %%Pad array so that the imposed image lines up properly
    scene  = padarray(scene,[kH,kW]);
    orig_kern = padarray(orig_kern,[r,c],'pre');
    orig_kern = padarray(orig_kern,[sH-r,sW-c],'post');

    %%Show the Original Image
    figure, imshow(orig_scene,[]);
    title('Original Scene');

    %%Show the surf Image
    figure, surf(G), shading flat;
    title('Quantized Samples');

    %%Show the G matrix
    figure, imshow(G,[]);
    title('Waldo Guess location');

    %%Show the Search Result
    figure, imshowpair(scene(:,:,1),orig_kern,'blend');
    title('Kernel Superimposed on Original Image');

end
```

Problem 1 continued on next page. . .

Listing 2: Correlation Function

```
function [ G ] = correlation( scene, kernel )
    %%Find the dimensions for the scene and kernel
    [kH, kW] = size(kernel);
    [sH, sW] = size(scene);


    %%set F, G, H matrices so that they match what's in the book.
    G = scene;
    F = scene;
    H = kernel;

    %%Divide Kernel Height and Width so we work with a smaller kern
    kH = kH/2;
    kW = kW/2;

    %%Generate Mean for Scene and Kernel
    meanS = mean(mean(scene));
    meanK = mean(mean(kernel));


    %%Perform Correlation
    for i=(kH):(sH-kH)
        for j=(kW):(sW-kW)
            G(i,j) = sum(sum((F(i-kH+1:i+kH, j-kW+1:j+kW)-meanS).*(H-meanK)));
        end
    end
end
```

Listing 3: Convolution Function

```
    %%For Convolution, just Flip the filter in both directions
    kernel = rot90(kernel,2);
    [kH, kW] = size(kernel);
    [sH, sW] = size(scene);

    G = scene;
    F = scene;
    H = kernel;

    kH = kH/2;
    kW = kW/2;

    meanS = mean(mean(scene));
    meanK = mean(mean(kernel));

    %%Perform Correlation
    for i=(kH):(sH-kH)
        for j=(kW):(sW-kW)
            G(i,j) = sum(sum((F(i-kH+1:i+kH, j-kW+1:j+kW)-meanS).*(H-meanK)));
        end
    end
end
```