Table of Contents

WET 1 Computational Photography WET 1

Shai Yehezkel 205917883 Lior Dvir 207334376


```
% Close graphs and clear workspacce
clear all;
close all;
clc;
```

Add supplied path

```
supp_files_path =  "./supp_files";
addpath(supp_files_path);

% Load variable from data mat
load(supp_files_path+"/data.mat");
```

Define the images derviatives

```
gx           = [-1 1];
gy           = gx.';

% Calc image derivatives
x_gx         = conv_fft2(x,gx,'same');
x_gy         = conv_fft2(x,gy,'same');

% Estimate sigma with obsvar as suggested
obsvar       = sqrt(mean(x_gx.^2 +x_gy.^2 ,'All'));

% Gather data into cell arrays
blur_kernels = {k1 k2 k3};
images       = {y1 y2 y3};

% Estimate eta for each image and corresponding kernel
etas        = zeros(3,1);
for i = 1:numel(images)
    % y = k*x +n -> n = y - k*x
    etas(i) =std(images{i} -
 conv_fft2(x,blur_kernels{i},'valid'),0,'all');
end
```

Q2

```
ImArray1= {};
```

```
for i = 1:numel(images)
    % Fetch image and corresponding eta
    eta           = etas(i);
    y             = images{i};
    ImArray1{1,i} = y;

    % Deblur in primal
    k              = blur_kernels{i};
    deconv_primal1 = deconvPrimal(y,k,eta,obsvar,0);
    ImArray1{2,i}  = deconv_primal1;

    % Deblur in primal with cyclic
    deconv_primal2 = deconvPrimal(y,k,eta,obsvar,1);
    ImArray1{3,i}  = deconv_primal2;

    % Deblur in freq domain
    deconv_freq    = deconvFreq(y,k,eta,obsvar);
    ImArray1{4,i}  = deconv_freq;
end
ImArray1 = cell2mat(ImArray1);
```

Display and save results

```
figure(1);
imshow(ImArray1);
title("ImArray1");
imwrite(ImArray1,'./ex1_q2.png');
```

Q3

```
ImArray2 = {};
for i = 1:numel(images)
    for j = 1:numel(blur_kernels)
        % Deblur each image with each kernel
        eta           = etas(i);
        y             = images{i};
        k             = blur_kernels{j};
        ImArray2{i,j} = deconvPrimal(y,k,eta,obsvar,0);
    end
end
ImArray2 = cell2mat(ImArray2);
```

Display and save results

```
figure(2);
imshow(ImArray2);
title("ImArray2");
imwrite(ImArray2,'./ex1_q3.png');
```

Q3

```
ImArray3 = {};
for i = 1:numel(images)
    for j = 1:3
        % Deblur each image with different eta magnitudes
        eta           = etas(i)*10^(2-j);
        y             = images{i};
        k             = blur_kernels{i};
        ImArray3{i,j} = deconvPrimal(y,k,eta,obsvar,0);
    end
end
ImArray3 = cell2mat(ImArray3);
```

Display and save results

```
figure(3);
imshow(ImArray3);
title("ImArray3");
imwrite(ImArray3,'./ex1_q4.png');
```

ImArray2

Published with MATLAB® R2020a