OSP Assignment 3 Readme.txt

1. **Uniform mean filter**

* Purpose of the code

This code implements uniform mean filter from grayscale input and color input image. “MeanFilterGray.cpp” is for grayscale image and “MeanFilterRGB.cpp” is for color input image. Uniform mean filter is low pass filter, so it makes image blur. Also, in this code there are three types of boundary processing. Zero padding, mirroring and adjusting kernel.

* Environment

visual studio 2019 with opencv 2.4.13.6

* How to run this code

cl MeanFilterGray.cpp

cl MeanFilterRGB.cpp

* How to adjust parameters

to change input image: edit "input.jpg"

Mat input = imread("input.jpg", CV\_LOAD\_IMAGE\_COLOR);

to change the method of boundary processing: edit parameter in meanfilter function

Output = meanfilter(input, 3, “zero-paddle”);

1. **Gaussian filter**

* Purpose of the code

This code is for low pass filter, gaussian filter. “GaussianGraySkeleton.cpp” is for grayscale input image and “GaussianRGBSkeleton.cpp” is for color input image. Gaussian distribution is one of the most commonly used parametric models. If you use gaussian filter at filtering blurred results looks more natural compared to mean filter.

* Environment

visual studio 2019 with opencv 2.4.13.6

* How to run this code

cl GaussianGraySkeleton.cpp

cl GaussianRGBSkeleton.cpp

* How to adjust parameters

to change input image: edit "input.jpg"

Mat input = imread("input.jpg", CV\_LOAD\_IMAGE\_COLOR);

to change the method of boundary processing: edit the last parameter in gaussianfilter function

Output = gaussianfilter(input, 1, 1, 1, “zero-paddle”);

1. **Sobel filter**

* Purpose of the code

This code implements sobel filter. “SobelGraySkeleton.cpp” is for grayscale input image and “SobelRGBSkeleton.cpp” is for color input image. Sobel filter is high pass filter. it estimates intensity change.

* Environment

visual studio 2019 with opencv 2.4.13.6

* How to run this code

cl SobelGraySkeleton.cpp

cl SobelRGBSkeleton.cpp

* How to adjust parameters

to change input image: edit "input.jpg"

Mat input = imread("input.jpg", CV\_LOAD\_IMAGE\_COLOR);

1. **Laplacian filter**

* Purpose of the code

This code is for Laplacian filtering. “LaplacianGray.cpp” is for grayscale input image and “LaplacianRGB.cpp” is for color input image. Laplacian filter is high pass filter. It estimates intensity change.

* Environment

visual studio 2019 with opencv 2.4.13.6

* How to run this code

cl LaplacianGray.cpp

cl LaplacianRGB.cpp

* How to adjust parameters

to change input image: edit "input.jpg"

Mat input = imread("input.jpg", CV\_LOAD\_IMAGE\_COLOR);

1. **Gaussian filter in a separable manner**

* Purpose of the code

This code implements gaussian filtering in a separable way. “UnsharpMaskingGray.cpp” is for grayscale input image and “UnsharpMaskingRGB.cpp” is for color input image. Gaussian filter is separable. If gaussian filter is separated and implemented in code, it is going to be faster in terms of runtime

.

* Environment

visual studio 2019 with opencv 2.4.13.6

* How to run this code

cl SepGaussianGraySkeleton.cpp

cl SepGaussianRGBSkeleton.cpp

* How to adjust parameters

to change input image: edit "input.jpg"

Mat input = imread("input.jpg", CV\_LOAD\_IMAGE\_COLOR);

to change the method of boundary processing: edit parameter in UnsharpMask function

output = gaussianfilterSep(input, 1, 1, 1, “zero-paddle”);

1. **Unsharp Masking**

* Purpose of the code

This code implements unsharp masking. “UnsharpMaskingGray.cpp” is for grayscale input image and “UnsharpMaskingRGB.cpp” is for color input image. Unsharp masking makes an image look sharper by boosting high frequency components.

* Environment

visual studio 2019 with opencv 2.4.13.6

* How to run this code

cl UnsharpMaskingGray.cpp

cl UnsharpMaskingRGB.cpp

* How to adjust parameters

to change input image: edit "input.jpg"

Mat input = imread("input.jpg", CV\_LOAD\_IMAGE\_COLOR);

to change the method of boundary processing: edit parameter in sobelfilter function

output = UnsharpMask(input, 1, 1, 1, “zero-paddle”, 0.5);