Computer Vision hw#1

Advanced Color-to-Gray Conversion

B05902066

1. Joint bilateral filter

I implement the joint bilateral filter by following the formula as below.

, which contains spatial kernel and range kernel.

For spatial kernel, as all the computation sharing same weights, I precompute the spatial kernel with given sigma\_r to speed up the process.

For range kernel, as the computation being based on the pixel value and corresponding neighbor pixels, we need to compute the weights locally. To speed up the computation, utilizing the python build-in functions and numpy package would highly improve the performance.

Actually, I would only use one loop to go through all the pixels on the input image. For range kernel, I use the numpy’s broadcast computation find the weight in parallel. For combining spatial kernel and range kernel and applying on the input image, I use the matrix multiplication and also the broadcast computation to find the corresponding output value.

With these tricks, I could complete the computation less than 20 seconds per image.

2. Local minimal

The method to find the local minimal is intuitive. For any combination of weight w = [w\_b, w\_g, w\_r], the neighbor would be w + one of [[0, -0.1, 0.1], [0, 0.1, -0.1], [0.1, 0, -0.1], [0.1, -0.1, 0], [-0.1, 0.1, 0], [-0.1, 0, 0.1]]. We could simply check the error value of a given weight and the corresponding neighbor to find the local minimal.

3.