Homework-3

Q1: A technique for computing optimal thresholds was developed in class under the assumption that if a single pixel x is changed into q the error is (x − q)^2 . Derive equations for computing the optimal threshold if the error of moving a pixel of value x to a pixel of value q is |x−q| and not (x − q)^2 . Follow the same steps as the derivation in class. You may want to use the fact that the derivative of |x| is 1 if x > 0 and -1 if x < 0.

Solution:

We want to make E as small as possible, then, we have:

Similarly:

Therefore, we can compute all possible value of E for any possible t with q1 and q2 provide above.

Q2: You are given the following image:

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | 6 | 6 | 10 |
| 6 | 6 | 6 | 10 |
| 17 | 17 | 17 | 17 |
| 17 | 17 | 17 | 88 |

1. What is the image histogram

Solution:

|  |  |
| --- | --- |
| Value | Number of Pixels |
| 0 | 0 |
| 1 | 0 |
| … | 0 |
| 6 | 6 |
| … | 0 |
| 10 | 2 |
| … | 0 |
| 17 | 7 |
| .. | 0 |
| 88 | 1 |
| .. | 0 |
| 255 | 0 |

2. What would be the result of applying the optimal thresholding algorithm that was discussed in class to this image?

Solution:

For given image is simple, we can set three possible value of t: 8, 13 and 30.

1) If t = 8:

2) If t = 13:

3) If t = 30:

∴E is minimum when t = 30 , [17, 87] ∩ Z

The image after the threshold is applied as:

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | 12 | 12 | 12 |
| 12 | 12 | 12 | 12 |
| 12 | 12 | 12 | 12 |
| 12 | 12 | 12 | 88 |

3. What image is obtained by linearly scaling the pixel values to the 0 – 255 range?

Solution:

Range: [6, 88] -> [0, 255]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pixel Value | Number (x) | (x – 6)\*(255 – 0)/(88 - 6) + 0 | Round |  |
| 6 | 6 | 6 | 0 | 6->0 |
| 10 | 2 | 18.4390 | 18 | 10->18 |
| 17 | 7 | 40.2073 | 40 | 17->40 |
| 88 | 1 | 255 | 255 | 88->255 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 18 |
| 0 | 0 | 0 | 18 |
| 40 | 40 | 40 | 40 |
| 40 | 40 | 40 | 255 |

4. What image is obtained by histogram equalization to the 0 – 255 range?

Solution:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | h(i) | f(i) | ((f(I - 1)+f(i))/2)\*(256/16) | Floor |  |
| 0 | 0 | 0 | 0 | 0 | 0->0 |
| 6 | 6 | 6 | 48 | 48 | 6->48 |
| 10 | 2 | 8 | 112 | 112 | 10->112 |
| 17 | 7 | 15 | 184 | 184 | 17->184 |
| 88 | 1 | 16 | 248 | 248 | 88->248 |
| … | … | … | … | … |  |
| 255 | 0 | 16 | 255 | 255 | 255->255 |

|  |  |  |  |
| --- | --- | --- | --- |
| 48 | 48 | 48 | 112 |
| 48 | 48 | 48 | 112 |
| 184 | 184 | 184 | 184 |
| 184 | 184 | 184 | 248 |

Q3: You are given the following color image, where the pixel values are in sRGB, The value of each pixel is given as the triplet (r, g, b).

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (255, 0, 0) | (255, 0, 0) | (255, 0, 0) | (255, 0, 0) |
| (100, 100, 100) | (100, 100, 100) | (100, 100, 100) | (100, 100, 100) |
| (0, 100, 100) | (0, 100, 100) | (0, 100, 100) | (0, 100, 100) |

A Convert the color image to nonlinear RGB.

Solution:

There are four color that need to convert:

(0, 0, 0), (255, 0, 0), (100, 100, 100) and (0, 100, 100)

D = D’/255

Nonlinear RGB:

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (1, 0, 0) | (1, 0, 0) | (1, 0, 0) | (1, 0, 0) |
| (0.392, 0.392, 0.392) | (0.392, 0.392, 0.392) | (0.392, 0.392, 0.392) | (0.392, 0.392, 0.392) |
| (0, 0.392, 0.392) | (0, 0.392, 0.392) | (0, 0.392, 0.392) | (0, 0.392, 0.392) |

B Convert the color image to linear RGB.

D = invgamma(D’)

Linear RGB:

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (1, 0, 0) | (1, 0, 0) | (1, 0, 0) | (1, 0, 0) |
| (0.127, 0.127, 0.127) | (0.127, 0.127, 0.127) | (0.127, 0.127, 0.127) | (0.127, 0.127, 0.127) |
| (0, 0.127, 0.127) | (0, 0.127, 0.127) | (0, 0.127, 0.127) | (0, 0.127, 0.127) |

C Convert the color image to XYZ.

XYZ:

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (0.412, 0.213, 0.019) | (0.412, 0.213, 0.019) | (0.412, 0.213, 0.019) | (0.412, 0.213, 0.019) |
| (0.121. 0.127, 0.139) | (0.121. 0.127, 0.139) | (0.121. 0.127, 0.139) | (0.121. 0.127, 0.139) |
| (0.069, 0.1, 0.136) | (0.069, 0.1, 0.136) | (0.069, 0.1, 0.136) | (0.069, 0.1, 0.136) |

D Convert the color image to xyY.

xyY:

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (0.64, 0.33, 0) | (0.64, 0.33, 0) | (0.64, 0.33, 0) | (0.64, 0.33, 0) |
| (0.313, 0.329, 0.127) | (0.313, 0.329, 0.127) | (0.313, 0.329, 0.127) | (0.313, 0.329, 0.127) |
| (0.225, 0.329, 0.1) | (0.225, 0.329, 0.1) | (0.225, 0.329, 0.1) | (0.225, 0.329, 0.1) |

(Assign 0/(0 + 0 + 0) = 0 here for divide-by-0 error.)

E Convert the color image to Luv.

D65 is used for white and we have: (XW, YW, ZW) = (0.95, 1.0, 1.09).

For each pixel:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | Y | Z | t | L | u’ | v’ | u | v |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.412 | 0.213 | 0.019 | 0.213 | 53.276 | 0.45 | 0.523 | 174.532 | 38.092 |
| 0.121 | 0.127 | 0.139 | 0.127 | 42.308 | 0.198 | 0.468 | 0 | 0 |
| 0.069 | 0.1 | 0.136 | 0.1 | 37.842 | 0.14 | 0.455 | -28.533 | -6.395 |

Luv:

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (53, 175, 38) | (53, 175, 38) | (53, 175, 38) | (53, 175, 38) |
| (42, 0, 0) | (42, 0, 0) | (42, 0, 0) | (42, 0, 0) |
| (38, -29, -6) | (38, -29, -6) | (38, -29, -6) | (38, -29, -6) |

F Compute linear illumination stretching in the Luv domain, and convert the result back to sRGB.

Range: [0, 53] -> [0, 100]

|  |  |  |  |
| --- | --- | --- | --- |
| L | (x – 0)\*(100 – 0)/(53 - 0) + 0 | Round |  |
| 0 | 0 | 0 | 0->0 |
| 53 | 100 | 100 | 53->100 |
| 42 | 79.245283 | 79 | 42->79 |
| 38 | 71.6981132 | 72 | 38->72 |

L’uv:

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (100, 175, 38) | (100, 175, 38) | (100, 175, 38) | (100, 175, 38) |
| (79, 0, 0) | (79, 0, 0) | (79, 0, 0) | (79, 0, 0) |
| (72, -29, -6) | (72, -29, -6) | (72, -29, -6) | (72, -29, -6) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| L' | u | v | u' | v' | X | Y | Z |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100 | 175 | 38 | 0.333 | 0.497 | 1.508 | 1 | 0.534 |
| 79 | 0 | 0 | 0.198 | 0.468 | 0.523 | 0.549 | 0.6 |
| 72 | -29 | -6 | 0.167 | 0.462 | 0.355 | 0.437 | 0.534 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R | G | B | R' | G' | B' | r | g | b |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0.437 | 0.444 | 1 | 0.692 | 0.697 | 255 | 176 | 178 |
| 0.552 | 0.548 | 0.551 | 0.769 | 0.766 | 0.768 | 196 | 195 | 196 |
| 0.212 | 0.498 | 0.495 | 0.498 | 0.734 | 0.732 | 127 | 187 | 187 |

sRGB:

|  |  |  |  |
| --- | --- | --- | --- |
| (0, 0, 0) | (0, 0, 0) | (0, 0, 0) | (0, 0, 0) |
| (255, 176, 178) | (255, 176, 178) | (255, 176, 178) | (255, 176, 178) |
| (196, 195, 196) | (196, 195, 196) | (196, 195, 196) | (196, 195, 196) |
| (127, 187, 187) | (127, 187, 187) | (127, 187, 187) | (127, 187, 187) |