digital image processing hw1

problem 1: piecewise linear transformation

hw1_1.m loads cameraman.tif image file, applies two different piecewise linear transform to the image, and plot the images in order.

PiecewiseLinearTr.m defines piecewise linear transform of given input and output transformed image. Here is the pseudocode for implementing PiecewiseLinearTr.m.

Algorithm 1: PiecewiseLinearTr.m

```
output ← zeros;
slope ← zeros;
y_inter ← zeros;
for i = 1 : (len(a) − 1) do
    slope_i ← (b(i+1) − b(i))/(a(i+1) − a(i));
    y_inter_i ← (a(i+1) ∗ b(i) − a(i) ∗ b(i+1))/(a(i+1) − a(i));
end
for i = 1 : M do
    for j = 1 : N do
        for k = 1 : (len(a) − 1) do
            if input(i, j) between a(k), a(k+1) then
                output(i, j) = slope_k ∗ input(i, j) + y_inter_k;
            end
        end
    end
end
```

Result of hw1_1.m can be found in Figure2. The leftmost one is the original image, middle is the transformed version with segment coordinates of [0,1], [1,0]. The rightmost one is from the segment coordinates of [0 .25 .5 .75 1],[0 .75 .25 .5 1].

Figure1 plots two transformation functions. It can be seen that left transformation inverses the input intensity, which is proved on middle version of Figure2. In the rightmost cameraman image, darkest part like coat is whitened and background, which has relatively high intensity, is darkened. This corresponds with right function of Figure1.

problem 2: image histogram

hw1_2.m loads input.jpg and plot histogram of the image. Hist.m implements gathering histogram statistics. Below is the pseudocode for the file.

Figure3 is the result of the hw1_2.m.

problem 3: histogram equalization

hw1_3.m loads input.jpg and applies histogram equalization. HistEq.m accepts input as image and output histogram-equalized version of the image. The pseudocode for HistEq.m is as follows.

Algorithm 2: Hist.m

```
hist ← zeros;
for i = 1 : M do
    for j = 1 : N do
        hist(input(i, j)) += 1;
    end
end
plot hist;
```

Algorithm 3: HistEq.m

```
hist ← zeros;
transform ← zeros;
for i = 1 : M do
    for j = 1 : N do
        hist(input(i, j)) += 1;
    end
end
for i = 1 : L do
    z ← sum(hist(1 : i));
    z ← z ∗ (L − 1)/(M ∗ N);
    transform(i) ← z;
end
output ← input;
for i = 1 : M do
    for j = 1 : N do
        output(i, j) ← transform(input(i, j));
    end
end
```

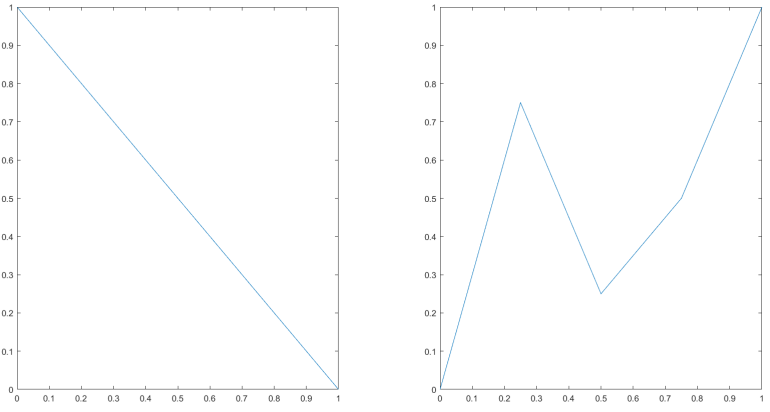


Figure 1: transformation functions. left: coordinates of [0,1], [1,0], right: coordinates of [0 .25 .5 .75 1],[0 .75 .25 .5 1].

Figure4 depicts the result. It can be seen that the resulting image has enhanced contrast. Detailed histogram can be found in Figure5.

Transformed histogram follows more to uniform than the original.

problem 4: histogram equalization

problem 5: histogram matching

Figure 2: result of running hw1_1.m

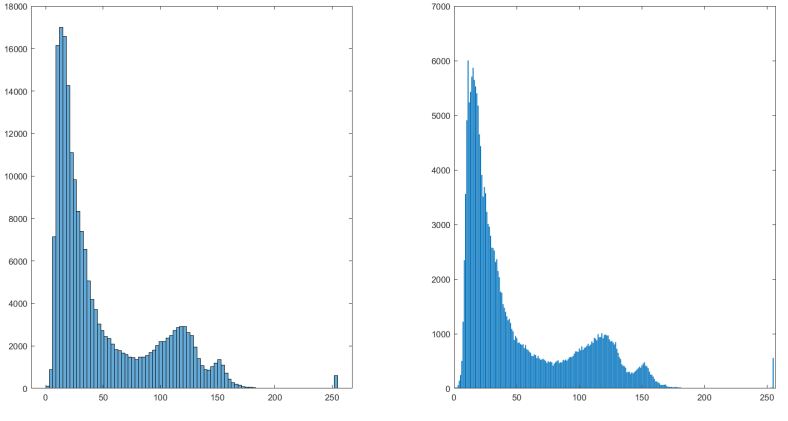
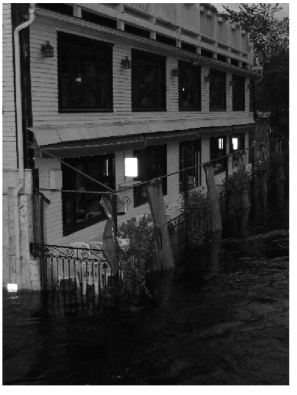


Figure 3: histogram visualizations.



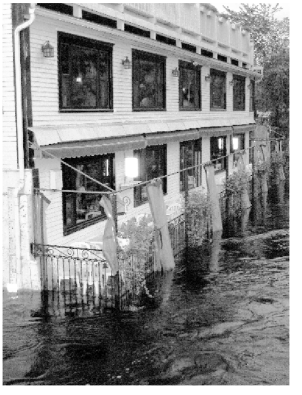
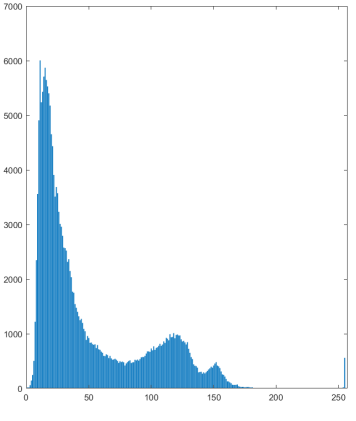


Figure 4: histogram visualizations.



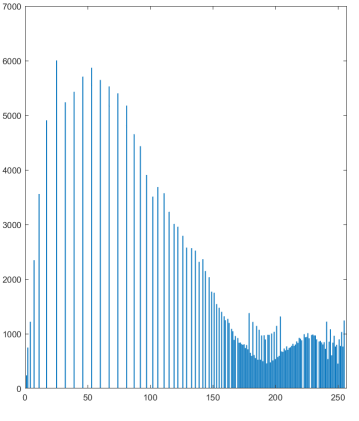


Figure 5: histogram visualizations.