AIC

Cours TC5

Result of TP5

Students:
Xiaoxiao CHEN
Dong FEI

Supervisor:
Prof. Yohanne TENDERO

22 oct. 2016

1 Exercise: Histogram equalization

This exercise aims to implement the histogram equalization on some images. With the command: $J = histeq(I, n)$ transforms the intensity image I, returning in J an intensity image with n discrete gray levels, we can easily realize the effects we want.

1.1 Analysis of result

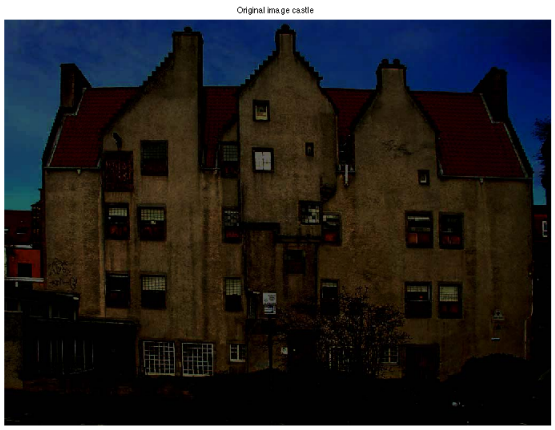

Figure 1: origin image castle

Figure 2: histogram equalization of castle

Figure 3: origin image ruins

Figure 4: histogram equalization of ruins

We choose the pictures with dark and bright backgrounds. In particular, the method can lead to better details in photographs that are over or under-exposed. It is noticed that the castle is much clearer than the original image, more details like window or chimney are showed in the image. The same situation happens in another image, the original bright parts become more visible after histogram equalization.

2 Exercise: Midway method

This exercise aims to implement the midway method for realizing histogram equalization a pair of images. The method can maintain as much as possible the images' previous gray dynamics.Here we simplifies the situation to two images with same size.

2.1 Midway method

A satisfactory definition of the midway histogram between h_1 and h_2 is the harmonic mean between their cumulative histograms H_1 and H_2, i.e.

$$H_{midway} = (0.5(H_1^{-1} + H_2^{-1}))^{-1}$$

The resulting contrast changes f_{12} and f_{21} applied to the images $u1$ and $u2$ are

$$f_{12}(x) = \frac{1}{2}(x + H_2^{-1} \circ H_1(x))$$

$$f_{21}(x) = \frac{1}{2}(x + H_1^{-1} \circ H_2(x))$$

2.2 Analysis of result

Midway equalization applied channel-wise for two color images of the same size taken with two different exposition times. The input images $u1$ and $u2$ are shown first. The resulting images $\hat{u}1$ and $\hat{u}2$ are shown after.

Figure 5: $u1$: first image of beach

Figure 6: $u2$: second image of beach

Figure 7: $\hat{u}1$: first result of midway

Figure 8: $\hat{u}2$: second result of midway

We notice the same effect as using the command *histeq*(). Except this method works on the cumulative histograms and gives the two image the same intermediate histogram and result in images with more clear details.

3 Exercise: Gaussian images

We use a simple way here to generate the image with gaussian variables. The images are showed with different sigma, which shows different results.

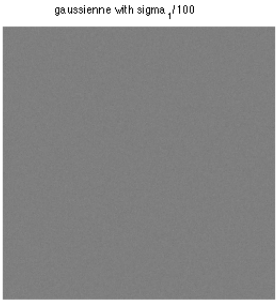

Figure 9: gaussian=0.01

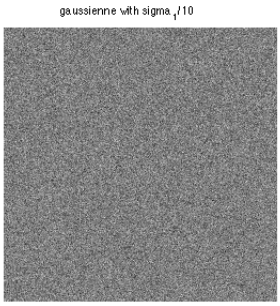

Figure 10: gaussian=0.1

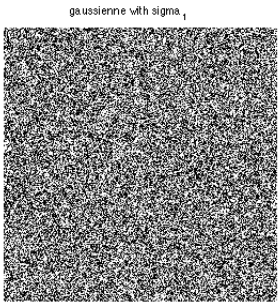

Figure 11: gaussian=1

Although the means of the three pictures are all 0.5. We found that the bigger the standard deviation, the more variant the picture is.

4 Exercise: Bilateral Filter

The answers of the exercise are attached below as scanned pdf

The intensity value at each pixel in an image is replaced by a weighted average of intensity values from nearby pixels. The weight here are based on a gaussian distribution, the weights depend not only on Euclidean distance of pixels, but also on the color intensity.This filter preserves sharp edges by systematically looping through each pixel and adjusting weights to the adjacent pixels accordingly.

Figure 12: origin bilateral

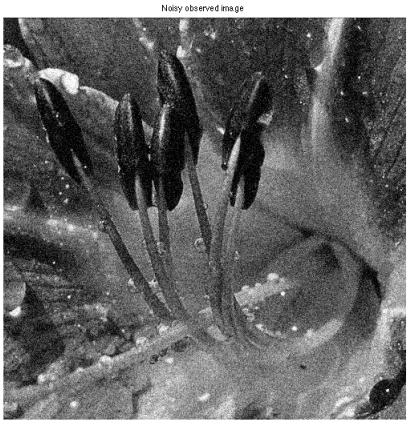

Figure 13: bilateral noise

Figure 14: bilateral denoise

From the result, we can tell that the noises are reduced by the filter. But the whole image is still a little bit blur compared with the original image. The edge of picture are still filled with noises which seems abnormal. The objects are well kept compared with gaussian filter.

Please refer to scanned version below.

A simple implementation of the bilateral filter

Part 1. Implementation

Q1 when $y \in \Omega(p)$: $y \in \{p_1-w, \dots, p_1+w\}$

$\Rightarrow$ y ranges in $\Omega(p)$, $y-p$ ranges in $\Omega(p)-p$

$\Rightarrow \Omega(p)-p = \{x=(x_1,x_2) : x_1 \in \{-w, +w\}, x_2 \in \{-w, \dots, w\}$

$\Rightarrow y-p \in \{-w, w\}$

Q2 $f_s(\|y-p\|_{\ell^2}) = \exp\left(-\frac{(y-p)^2}{2\sigma_s^2}\right)$

Since $y-p \in \{-w, w\} \Rightarrow (y-p)^2 \in \{0, w^2\}$

$\Rightarrow f_s(\|y-p\|_{\ell^2}) \in \{\exp(-\frac{w^2}{2\sigma_s^2}), 1\}$

Q3 The u we need to compute $U_{denoised}(p)$ is $U(p) \ni p$, where $p \in \Omega(p)$

Q4. Because the square cannot exceed the boundaries of the image, we can deduce that $p_1+w \le N$, $p_1-w \ge 1$ and $p_2+w \le M$, $p_2-w \ge 1$

$\Rightarrow p_1 \in \{w+1, \dots, N-w\}$, $p_2 \in \{w+1, \dots, M-w\}$

Q5. Since y_1 and y_2 starts respectively by p_1-w, p_2-w

Suppose $p_1-w=1 \Rightarrow p_1=w+1 \Rightarrow p_1+w=2w+1$

$y-p \in \{-w, w\} \Rightarrow 0 \le y-p+w \le 2w$

$\Rightarrow 1 \le y-p+w+1 \le 2w+1$

So if we have a new variable $x = y-p+w+1 \in \{1, 2w+1\}$ we can rewrite (3) so that the summation indexes starts at 1.

$$U_{denoised}(p) = \frac{1}{C}\sum_{y_1=p_1-w}^{p_1+w}\sum_{y_2=p_2-w}^{p_2+w} U(y_1,y_2)\exp\left(-\frac{(y_1-p_1)^2+(y_2-p_2)^2}{2\sigma_s^2}\right)\exp\left(-\frac{[U(y_1,y_2)-U(p_1,p_2)]^2}{2\sigma_i^2}\right)$$

$$= \frac{1}{C}\sum_{x_1=1}^{2w+1}\sum_{x_2=1}^{2w+1} U(x_1+p_1-w-1, x_2+p_2-w-1) \times \exp\left(-\frac{(x_1-w-1)^2+(x_2-w-1)^2}{2\sigma_s^2}\right) \times \exp\left(-\frac{[U(x_1+p_1-w-1, x_2+p_2-w-1)-U(p_1,p_2)]^2}{2\sigma_i^2}\right)$$

Q6 given $S: \{1, \dots, 2w+1\} \times \{1, \dots, 2w+1\} \ni (x_1, x_2) \to \exp\left(-\frac{(x_1-w-1)^2 + (x_2-w-1)^2}{2\sigma_s^2}\right)$

$\tilde{U}: \{1, \dots, 2w+1\} \times \{1, \dots, 2w+1\} \in (x_1, x_2) \longrightarrow U(x_1+P_1-w-1, x_2+P_2-w-1)$

We have

$$U_{denoised}(P) = \frac{1}{C} \sum_{x_1=1}^{2w+1} \sum_{x_2=1}^{2w+1} \tilde{U}(x_1, x_2) S(x_1, x_2) \exp\left(-\frac{[\tilde{U}(x_1, x_2) - \tilde{U}(w+1, w+1)]^2}{2\sigma_i^2}\right)$$

Q7
$$C = \sum_{y_1=P_1-w}^{P_1+w} \sum_{y_2=P_2-w}^{P_2+w} \exp\left(-\frac{(y_1-P_1)^2 + (y_2-P_2)^2}{2\sigma_s^2}\right) \exp\left(-\frac{[U(y_1, y_2) - U(P_1, P_2)]^2}{2\sigma_i^2}\right)$$

$$= \sum_{x_1=1}^{2w+1} \sum_{x_2=1}^{2w+1} S(x_1, x_2) \exp\left(-\frac{[\tilde{U}(x_1, x_2) - \tilde{U}(w+1, w+1)]^2}{2\sigma_i^2}\right)$$

Q8 The objects needed to implement the program

a). matrix $S[2w+1, 2w+1] \ni (x_1, x_2) \to \exp\left(-\frac{(x_1-w-1)^2 + (x_2-w-1)^2}{2\sigma_s^2}\right)$

b). matrix $\tilde{U}[2w+1, 2w+1] \ni (x_1, x_2) \to U(x_1+P_1-w-1, x_2+P_2-w-1)$

c). realnumbers: w, σ_s, σ_i

Q9 1). given w, σ_s, σ_i

Since S has nothing to do with $P \Rightarrow S$ is a constant

Compute $\sum_{x_1=1}^{2w+1} \sum_{x_2=1}^{2w+1} \exp\left(-\frac{(x_1-w-1)^2 + (x_2-w-1)^2}{2\sigma_s^2}\right)$

2) use a loop over P_1, P_2

1) compute $\tilde{U}$

2) compute $\exp\left(-\frac{[\tilde{U}(x_1, x_2) - \tilde{U}(w+1, w+1)]^2}{2\sigma_i^2}\right)$

3). Compute C

4). Compute $U_{denoised}(P)$

Binôme – Xiaoxiao CHEN
Pong FEI

Part 2. Short analysis of the behavior of the bilater filter

Q 10. When $\sigma_i \to +\infty$

$$-\frac{[\tilde{u}(x_1,x_2) - \tilde{u}(w+1,w+1)]^2}{2\sigma_i^2} \to 0$$

$$\exp\left(-\frac{[\tilde{u}(x_1,x_2) - \tilde{u}(w+1,w+1)]^2}{2\sigma_i^2}\right) \to 1$$

So $u_{denoised}(P) \to$ convolution with a part of Gaussian kernel

Q 11 When $\sigma_i \to 0$

$$-\frac{[\tilde{u}(x_1,x_2) - \tilde{u}(w+1,w+1)]^2}{2\sigma_i^2} \to -\infty$$

$$\exp\left(-\frac{[\tilde{u}(x_1,x_2) - \tilde{u}(w+1,w+1)]^2}{2\sigma_i^2}\right) \to 0$$

So $u_{denoised}(P) = 0$ if $u(y) \neq u(P)$

if $u(y) = u(P)$, which means the pixel itself

in other words, nothing will be done if $u(y) = u(P)$

Q 12 When $\sigma_s \to 0$

$$\exp\left(-\frac{(x_1-w-1)^2+(x_2-w-1)^2}{2\sigma_s^2}\right) \to 0$$

So $u_{denoised}(P) = 0$ if $u(y) \neq u(P)$

if $u(y) = u(P)$, $S(x_1,x_2) = 1$, which means the pixel itself

The result is same as above

Q 13 When $\sigma_s \to +\infty$

$$\exp\left(-\frac{(x_1-w-1)^2+(x_2-w-1)^2}{2\sigma_s^2}\right) \to 1$$

So $u_{denoised}(P) \to$ convolution with a part of Gaussian kernel

Binôme — Xiaoxiao CHEN
Dong FEI

5 Exercise: zero padding vs DCT

Zero_padding and DCT are two operations used for zoom-in transform. We have already practiced zero_padding in the previous tp. As for Discrete Cosine Transform (DCT), we copy the image to reduce the boundary effect.

Figure 15: origin

Figure 16: zero_padding

Figure 17: DCT

From the result, we can see that the Discrete cosine transform works well on zooming in the image. Compared with zero-padding, the transformed image is smoother and has less blurs on the surface. Even though the differences between them remain subtle, the image transformed by DCT has more details and seems closer to the original image.