

數位影像處理

Digital Image Processing

作業一 HW1

R11921038 江讀晉

Problem 1

Prove that applying a 2-D transform with separable, symmetric kernels to an image can be implemented by:

- 1) computing a 1-D transform along the individual rows (columns) of the image and
- 2) computing another 1-D transform along the columns (rows) of the resulting image obtained in Step 1.

The forward transform kernel is

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) r(x, y, u, v).$$

With separable and symmetric kernels
the transform becomes

$$\begin{aligned} r(x, y, u, v) &= r_1(x, u) r_2(y, v) && (\text{separable}) \\ &= r_1(x, u) r_1(y, v) && (\text{symmetric}), \end{aligned}$$

Therefore, the computation of the transform can be applied independently on each axis. If the individual rows are conducted first, the equation can be rewritten as

$$\begin{aligned} T(u, v) &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) r(x, y, u, v) \\ &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) r_1(x, u) r_1(y, v). \end{aligned}$$

Apparently, the 1-D transform applied on the rows of the image can be computed first and then the second 1-D transform can be done on the columns of the result image, and vice versa.

$$\begin{aligned}
T(u, v) &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) r_1(x, u) r_1(y, v) \\
&= \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x, y) r_1(x, u) r_1(y, v) \\
&= \sum_{y=0}^{N-1} r_1(y, v) \sum_{x=0}^{M-1} f(x, y) r_1(x, u).
\end{aligned}$$

Problem 2

1. Gamma correction

根據課本公式， $s = cr^\gamma$ ，其中 s 為 output intensity levels， r 為 input intensity levels。為求 gamma 參數調整方便，將常數 $c = 1$ 。

然而，經測試圖庫中「image_2.png」之 intensity level 大約落在 60 至 100 之間，根據 plots of gamma equation，轉換後的圖片的 intensity level 仍然落在較小的區間（如 Fig.1 所示），可預期圖片的對比度不明顯。因此，為加強對比度，我在圖片經 gamma correction 後查表 (look-up-table)，使用線性轉換將 intensity 對映至 0 至 255。實作方式如 Fig.2 所示。

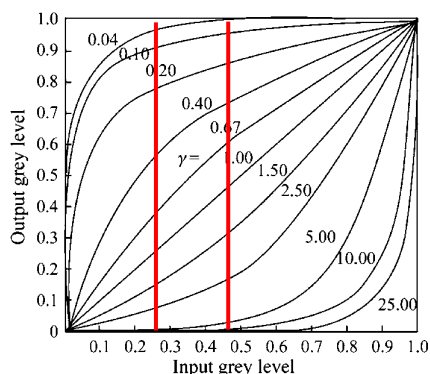


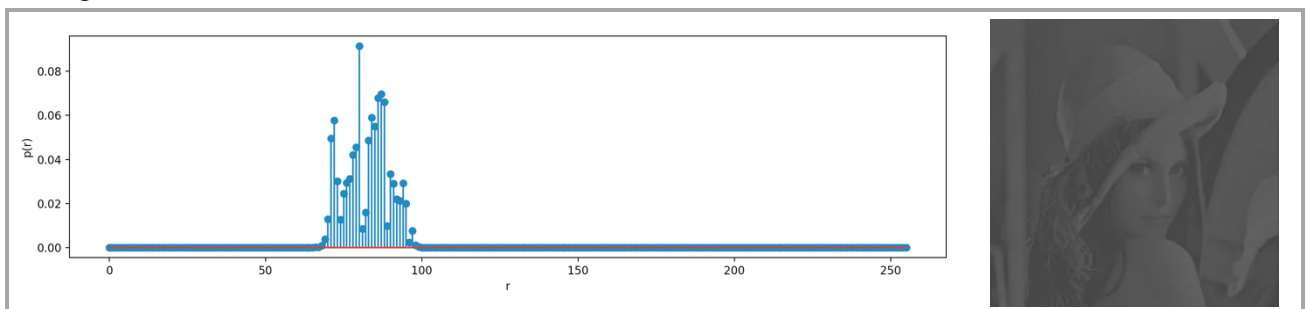
Fig.1 input 區間落在紅線之間

1. 將原本的 intensity 區間從[0, 255]縮小至[0, 1]之間。
2. 給定 gamma 值，計算出 gamma equation。
3. 找出計算後 intensity 之最大值和最小值，將其放大轉換至 0 和 255，其餘數值則依照線性轉換至[0, 255]。

Fig.2 gamma correction 實作描述

2. Histogram equalization

實作方法按照課堂投影片的公式，並繪製校正前後 $p_r(r_k)$ 和 $p_s(s_k)$ 的 histogram，結果如下圖 Fig.3 所示。



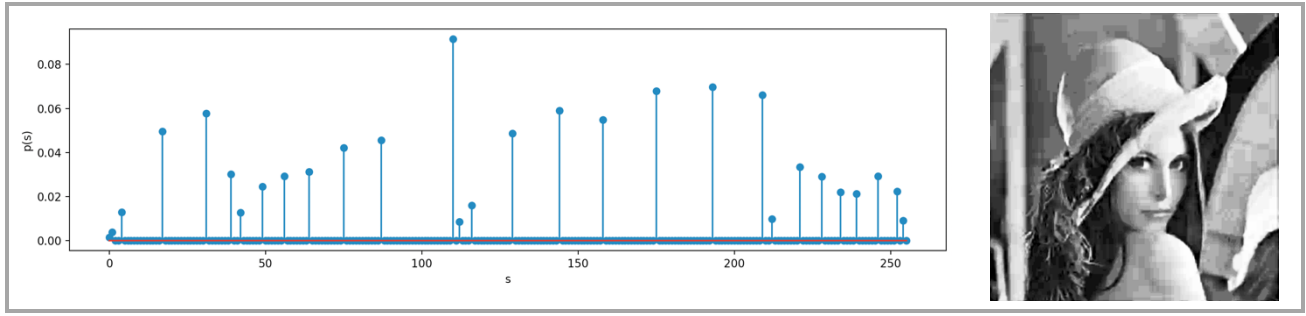


Fig.3 上方為校正前之 histogram 和圖片，下方則是經校正後的結果

3. Find gamma value

為了得到最小的差異，我使用結構相似性（structural similarity，SSIM）這項指標來找出 gamma 值，其公式為

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}。$$

當計算出最大相似度時，對應的 gamma 就是最佳解。根據測試，我求得之 gamma 值為 0.3，其 SSIM 值為 0.8756。圖像結果如 Fig.4 所示。



Fig.4 由左至右為原圖、gamma correction (gamma = 0.3)、histogram equalization

Problem 3

1. Resizing 和 rotating

在處理 rotate 時，有保持圖片完整或是維持相同影像尺寸這兩種選擇，我的實作方式選擇了後者，亦即圖片沿著中心旋轉，突出影像邊界的部分則將圖片資訊捨棄。

插值的方式，我的實作方法為將新的影像坐標，使用 backward warping 轉換至原影像的坐標，並針對非整數值實作三種插值法。「nearest neighbor」、「bilinear」皆參考課堂投影片，「bicubic」則參考 bicubic convolution algorithm，先沿 x 再 y 軸計算 intensity level，其參考公式如下：

$$p(t) = \frac{1}{2} \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} f_{-1} \\ f_0 \\ f_1 \\ f_2 \end{bmatrix}$$

$$\begin{aligned}
b_{-1} &= p(t_x, f_{(-1,-1)}, f_{(0,-1)}, f_{(1,-1)}, f_{(2,-1)}), \\
b_0 &= p(t_x, f_{(-1,0)}, f_{(0,0)}, f_{(1,0)}, f_{(2,0)}), \\
b_1 &= p(t_x, f_{(-1,1)}, f_{(0,1)}, f_{(1,1)}, f_{(2,1)}), \\
b_2 &= p(t_x, f_{(-1,2)}, f_{(0,2)}, f_{(1,2)}, f_{(2,2)}), \\
p(x, y) &= p(t_y, b_{-1}, b_0, b_1, b_2).
\end{aligned}$$

圖像結果如 Fig.5 所示。



Fig.5 由左至右為原圖、逆時針旋轉 30 度、沿各軸縮放(1.5, 2)倍的結果，插值法為 bicubic

2. Image registration

使用課堂投影片 bilinear approximation 的方式，

$$\begin{cases} x = c_1v + c_2w + c_3vw + c_4 \\ y = c_5v + c_6w + c_7vw + c_8 \end{cases}$$

求出 T.png 和 T_transformed.png 的 8 個參數值 (Fig.6 顯示影像結果)：

$$T_{trans}(x, y) = A T(x, y),$$

$$where\ matrix\ A = \begin{bmatrix} 1.29 & -0.35 & 0.00005 & 7.34 \\ 0.36 & 1.31 & -0.0001 & -83 \end{bmatrix}$$



Fig.6 由左至右為 T.png、T_transformed.png、使用 matrix A 計算之結果

Problem 4

1. Shade correction

使用 gaussian kernel 實作，並針對以下幾處進行嘗試：

- 對原圖片進行 padding，不完全採用 zero-padding 的方式，而是複製四邊上的值來做 padding，以期望減少 shade 在邊框處產生過黑的陰影。結果如 Fig.7 所示。
- 由於 gaussian kernel 為 separable，可以沿 x 和 y 軸先後和一維的 kernel 計算。然而，可能是計算誤差，我覺得直接採用二維 kernel 結果較好。

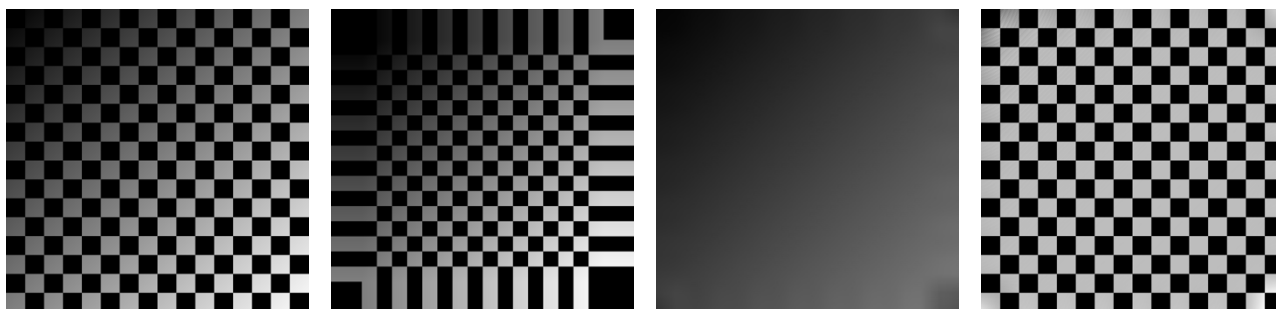


Fig.7 由左至右為原圖、加上 padding、shade、校正後的結果

2. Identify the root cause that makes the bottom right corner appear dark

我認為是因為圖片經過 zero-padding 之後，由於 kernel 的尺寸會較黑色方格尺寸大上幾倍，因此右下角處在計算過後會受到邊框外黑色的數值影響，呈現出的 shade 會較左方和上方的顏色更深一些。