

Unit 2 實習. Histogram-based 影像增強

1. 使用直方圖拓寬(histogram Stretching)影像對比增強。

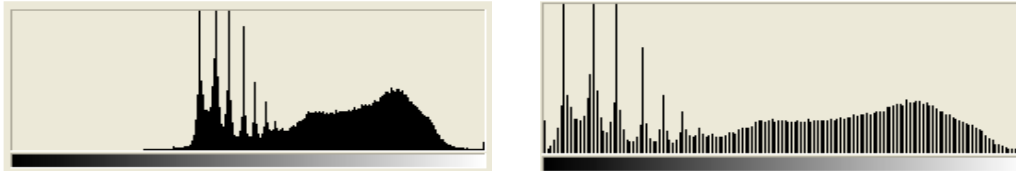
$$\text{Stretch}(I(r, c)) = \left[\frac{I(r, c) - I(r, c)_{\text{MIN}}}{I(r, c)_{\text{MAX}} - I(r, c)_{\text{MIN}}} \right] [\text{MAX} - \text{MIN}] + \text{MIN}$$

$I(r, c)_{\text{MAX}}$ is the largest gray-level value in the image $I(r, c)$

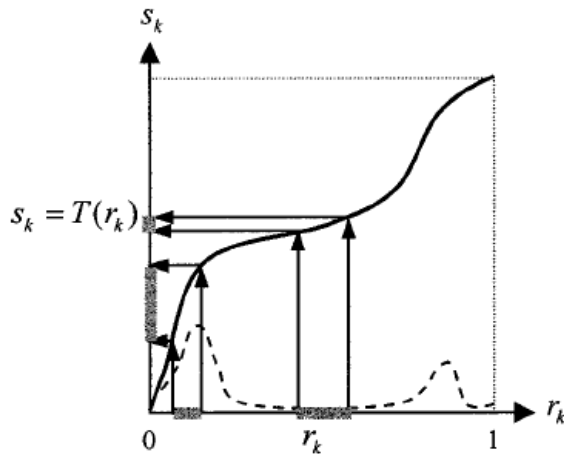
$I(r, c)_{\text{MIN}}$ is the smallest gray-level value in $I(r, c)$

MAX and MIN correspond to the maximum and minimum gray-level values possible (for an 8-bit image these are 0 and 255)

如下圖將 kaoshiung512x512.raw 的灰階分布拉寬至[0,255]。



2. 使用 Histogram Equalization(HE)增強影像對比



Transformation function for histogram equalization.

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j), \quad = \sum_{j=0}^k \frac{n_j}{n},$$

$$0 \leq r_k \leq 1 \quad \text{and} \quad k = 0, 1, \dots, L - 1$$

where

$p_r(r_j) = n_j/n$ probability density function (pdf) of the
input image level j ;
 n total number of pixels in the input image;
 n_j input pixel number of level j .

演算法：

Step 1. 計算影像灰階統計直方圖(histogram)Pr

Step 2. 從灰階統計直方圖計算累增直方圖(cumulative histogram) Sk

Step 3. 從累增直方圖計算等化分布直方圖(equalized histogram)f(x)，使灰階頻率平均分布在[X0, XL-1]:
 $f(x)=X0+(XL-1-X0)Sk$

X0 是期望的最小灰階值(例如 0)，XL-1 是期望的最大灰階值(例如 255)

Step 4. 以此等化分布直方圖 f(x)當作映射函數，重新指定影像每一 pixel 的灰階值。

程式範例：

```
void HistogramEqualization(uc2D &ima0, uc2D &ima1)
{
    long ImaSize=ima0.nr*ima0.nc;
    int histo[256]; //histogram
    float accpbhisto[256]; // cumulative istogram
    int table[256]; // Look-up table for mapping fuction of histogram
    equalization
    // Initialize
    for(int i=0;i<256;i++)
    {
        histo[i]=0;
        table[i]=0;
        accpbhisto[i]=0.0;
    }
    // Compute histogram
    for(int i=0;i<ima0.nr;i++) for(int
    j=0;j<ima0.nc;j++)histo[ima0.m[i][j]]++;

    // Compute cumulative histogram
    accpbhisto[0]=float(histo[0])/float(ImaSize);
    for(int i=1;i<256;i++)
    {
        accpbhisto[i]=accpbhisto[i-1]+float(histo[i])/float(ImaSize);
    }
    // compute mapping function
    for(int i=0;i<256;i++)table[i]=char(accpbhisto[i]*256.);
    // Enhancement
    for(int i=0;i<ima0.nr;i++) for(int j=0;j<ima0.nc;j++)
    ima1.m[i][j]=table[ima0.m[i][j]];
}
```

請寫一個主程式分別對 finger300x300、ant(gray)600x400 和 kaoshiung512x512 進行 HE 的影像增強。

3. Local HE 影像增強方法

每一個 pixel 與鄰近 pixel 的灰階值比較，決定其排序。再依此一排序的正比關係指定一個新的灰階值給這個 pixel。Local HE 影像增強方法是根據區域性(而非整張影像)的資訊來增強對比。

```

for each (x,y) in image do
{
  rank = 0
  for each (i,j) in contextual region of (x,y) do
  {
    if image[x,y] > image[i,j] then
      rank = rank + 1
    }
  output[x,y] = rank * max_intensity / (# of pixels in contextual region)
}

```

請寫一個主程式，嘗試使用不同大小的區域視窗(contextual region)，分別對 finger300x300、ant(gray)600x400 和 kaoshiung512x512 進行 Local HE 的影像增強。

4. 參數可調整的 HE 影像增強方法—AHE(Adaptive Histogram Equalization)

$$ACE = k_1 \left[\frac{m_{I(r,c)}}{\sigma_I(r,c)} \right] [I(r,c) - m_I(r,c)] + k_2 m_I(r,c)$$

where $m_{I(r,c)}$ = is the mean for the entire image $I(r,c)$

σ_I = local standard deviation (in the window under consideration)

m_I = local mean (average in the window under consideration)

k_1, k_2 = constants, vary between 0 and 1

k_1, k_2 是兩個介於[0,1] 的可調參數。至於計算影像平均值(mean)和標準差(standard deviation)的範例程式如下：

```

float mean_stddev (Image& im, float &mean, float &std_dev)
{
  int i, j;
  long N, sum=0;
  N = (long) (im.nr) * (long) (im.nc);
  for (i=0; i<im.nr; i++) for (j=0; j<im.nc; j++)
    sum += im.m[i][j];
  mean=(float)sum/(float) (N); //Calculating the mean
  float sumdev=0.0;
  for (i=0; i<im.nr; i++) for (j=0; j<im.nc; j++)
  {
    d = im.m[i][j] - mean;
    sumdev = sumdev+ d*d;
  }
  std_dev = sqrt(sumdev/N); //Calculating the standard deviance
}

```

Adaptive Histogram Equalization 也是基於區塊影像資訊的對比增強方法。請寫一個程式測試不同的 k_1, k_2 參數組合對 kaoshiung512x512 影像進行 AHE 影像增強的效果：

- (1) 採用 Try-and-Error 方法找到較好的一組(k_1, k_2)；
- (2) (option, 可不作)採用 GA、PSO 等最佳化方法找到一組最佳的(k_1, k_2)，可考慮用影像中 Edge 的強度和作為評估函數。

