

守大學電機系「電腦視覺」報告

單元一

數位影像：格式和操作

參考解答

MIAT(機器智慧與自動化技術)實驗室

中 華 民 國 93 年 9 月 29 日

1. (a)如果指紋影像 finger300x300 的取像面積是 14(mm)x14(mm) ，請計算取像系統的 dpi。(b)如果 kaoshiung512x512 遙測影像的覆蓋面積是 5(Km)x5(Km) ，請計算該影像的解析度(resolution) ，亦即每一 pixel 表示多大面積?

(a) Dots Per Inch. A measure of the resolution of a printer, scanner, or monitor. It refers to the number of dots in a one-inch line. The more dots per inch, the higher the resolution. If a printer is said to print at 300 dpi, it will be capable of printing 300 dots horizontally and 300 dots vertically over a square inch.

$$14\text{mm} = 1.4\text{cm}/2.54(\text{cm}/\text{inch}) = 0.55 \text{ inch}$$

$$\text{DPI} : 300 \text{ pixels}/0.55 \text{ inch} \approx 545(\text{dpi})$$

(b) (1). The resolution of a sensor is a measure of what size of objects it can image. Resolution is commonly expressed as the most closely spaced lines which can be recognised as being distinct. (2). The amount of pixels that are displayed on a screen measured in pixels horizontal by pixels vertical, such as 1024 x 768.

$$\text{Resolution} : 5000(\text{m})/512(\text{pixel}) \approx 10(\text{m}/\text{pixel})$$

or

$$\text{Resolution} : 10 \times 10 (\text{m}^2/\text{pixel})$$

2.請寫一程式，將 ant(gray)600x400 影像作水平鏡射和垂直鏡射。

Source Code :

```
#include<fstream.h>
#include"array.h"

void main(void)
{
    ifstream in("ant(gray)600x400.raw",ios::binary);
    ofstream out_h("ant(gray)600x400_horizontal.raw",ios::binary);
    ofstream out_v("ant(gray)600x400_vertical.raw",ios::binary);

    uc2D ima;
    ima.Initialize(400,600);

    char c;

    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        in.get(c);ima.m[i][j]=c;
    }

    for(int i=0;i<ima.nr;i++)for(int j=ima.nc-1;j>=0;j--)
        out_h<<ima.m[i][j];

    for(int i=ima.nr-1;i>=0;i--)for(int j=0;j<ima.nc;j++)
        out_v<<ima.m[i][j];
}
```



(a)

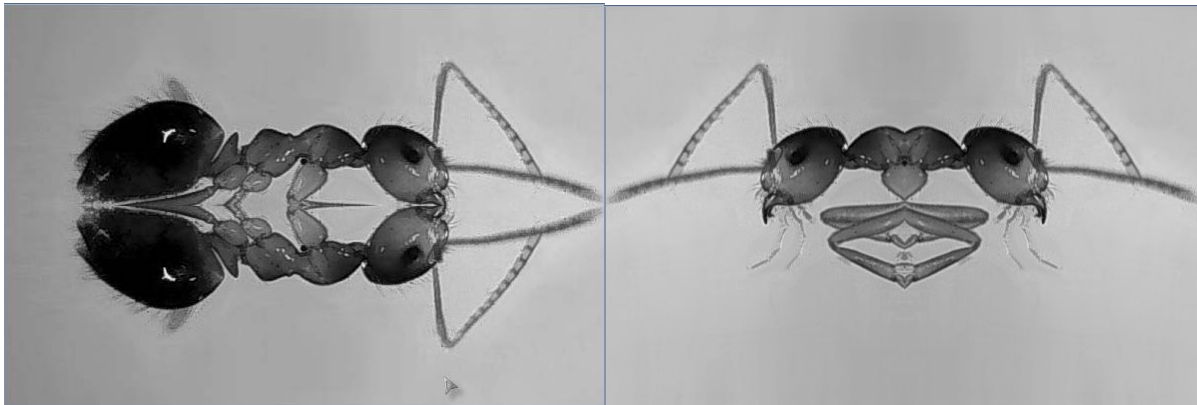


(b)



(c)

(a)原始影像 ant(gray)600x400.raw 、(b)水平鏡射 ant(gray)600x400_horizontal.raw 、(c)垂直鏡射 ant(gray)600x400_vertical.raw



3.將 finger300x300 第 100 row 的剖面資料取出，在 Excel 當中畫出其曲線圖。

Source Code :

```
#include<fstream.h>
#include"array.h"

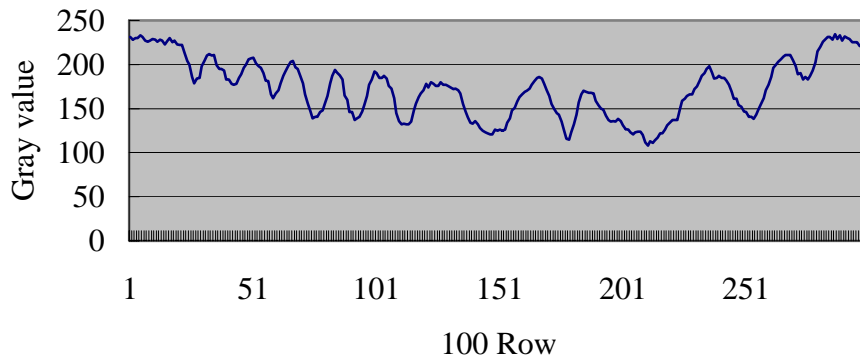
void main(void)
{
    ifstream in("finger300x300.raw",ios::binary);
    ofstream out("finger300x300_100row.xls");

    uc2D ima;
    ima.Initialize(300,300);

    char c;

    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        in.get(c);ima.m[i][j]=c;
    }

    for(int j=0;j<ima.nc;j++)
    {
        out<<(int)ima.m[100][j]<<endl;
    }
}
```



finger300x300 第 100 row 的剖面資料

4. 在 finger300x300 影像上畫兩條對角線(將線上的 pixel 值設為 255)。

Source Code :

```
#include<fstream.h>
#include"array.h"

void main(void)
{
    ifstream in("finger300x300.raw",ios::binary);
    ofstream out("finger300x300_1.raw",ios::binary);

    uc2D ima;
    ima.Initialize(300,300);

    char c;

    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        in.get(c);ima.m[i][j]=c;
        if(i==j || i==299-j) out<<(unsigned char)255;
        else out<<ima.m[i][j];
    }
}
```



(a)

(b)

(a)原始影像 finger300x300.raw、(b) 畫兩條對角線影像 finger300x300_1.raw

5.取出 `ant(gray)600x400` 左上角 `8x8 pixels` 的數值資料存檔，再用 Excel 呈現資料表。

Source Code :

```

#include<fstream.h>
#include"array.h"

void main(void)
{
    ifstream in("ant(gray)600x400.raw",ios::binary);
    ofstream out("ant(gray)600x400_8x8pixels.xls");

    uc2D ima;
    ima.Initialize(8,8);

    char c;

    for(int i=0;i<8;i++)
    {
        for(int j=0;j<8;j++)
        {
            in.get(c);ima.m[i][j]=c;
            out<<(int)ima.m[i][j]<<"\t";
        }
        out<<endl;
    }
}

```

ant(gray)600x400 左上角 8x8 pixels 的數值資料

```

135 136 138 137 138 138 137 137
138 137 139 139 138 140 139 140
140 139 139 139 141 141 141 140
141 141 141 142 142 141 142 142
142 142 142 141 141 141 142 143
143 143 143 143 143 142 142 142
142 142 143 143 144 144 144 144
143 143 144 144 145 145 145 145

```

6. 改變 ant(gray)600x400 影像灰階分布，將小於 140 的灰階值全設為 0，將大於 203 的灰階值全設為 255，最後將 140-203 灰階值分布 平均擴展至 0-255。

Source Code :

```
#include<fstream.h>
#include"array.h"

void main(void)
{
    ifstream in("ant(gray)600x400.raw",ios::binary);
    ofstream out("ant(gray)600x400_1.raw",ios::binary);

    uc2D ima;
    ima.Initialize(400,600);

    char c;

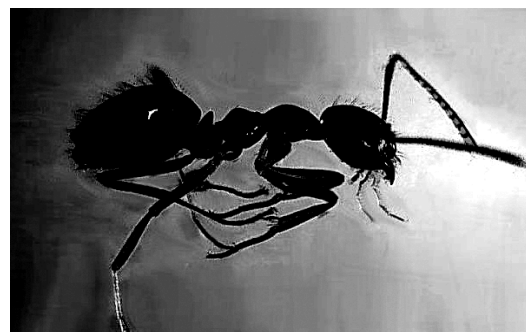
    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        in.get(c);ima.m[i][j]=c;
    }

    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        if(ima.m[i][j]<140)    ima.m[i][j]=0;
        else if(ima.m[i][j]>203) ima.m[i][j]=255;
        else
            ima.m[i][j]=(ima.m[i][j]-140)*255/63;

        out<<ima.m[i][j];
    }
}
```



(a)



(b)

(a)原始影像 ant(gray)600x400.raw、(b)處理後影像 ant(gray)600x400_1.raw

7.請找出一個最佳的 threshold 對 kaoshiung512x512 遙測影像作二值化。

Source Code : 尋找影像灰階分布圖


```

#include<fstream.h>
#include"array.h"

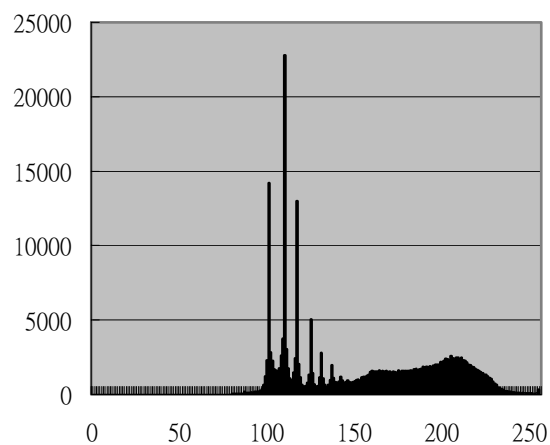
void main(void)
{
    ifstream in("kaoshiung512x512.raw",ios::binary);
    ofstream out("histogram.txt");

    uc2D ima;
    ima.Initialize(512,512);

    char c;
    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        in.get(c);ima.m[i][j]=c;
    }
    int histo[256];
    for(int i=0;i<256;i++) histo[i]=0;

    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        histo[ima.m[i][j]]++;
    }
    out<<"gray\thistogram\n";
    for(int i=0;i<256;i++)out<<i<<"\t"<<histo[i]<<endl;
}

```



kaoshiung512x512.raw 灰階分布圖

Source Code：以 150 做為 threshold 將影像二值化

```
#include<fstream.h>
#include"array.h"

void main(void)
{
    ifstream in("kaoshiung512x512.raw",ios::binary);
    ofstream out("kaoshiung512x512_bw.raw",ios::binary);

    uc2D ima;
    ima.Initialize(512,512);

    char c;

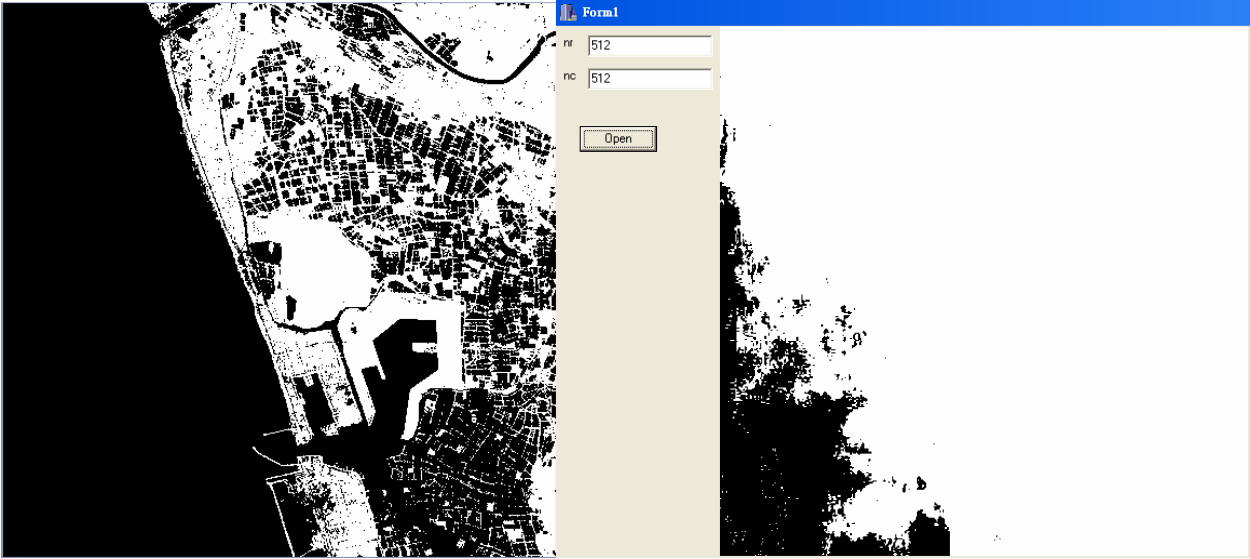
    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        in.get(c);ima.m[i][j]=c;
        if(ima.m[i][j]<150) out<<(unsigned char)0;
        else                out<<(unsigned char)255;
    }
}
```



(a)



(b)





8.輸出指紋影像 finger300x300 的 8 個 bit-planes。

```
for(int i=0;i<ima.nr;i++)
{
    for(int j=0;j<ima.nc;j++)
    {
        ima.m[i][j]=imain.get();
        unsigned char c=((ima.m[i][j])%2)*255; imaoutbit1<<c;
        c=((ima.m[i][j]>>1)%2)*255; imaoutbit2<<c;
        c=((ima.m[i][j]>>2)%2)*255; imaoutbit3<<c;
        c=((ima.m[i][j]>>3)%2)*255; imaoutbit4<<c;
        c=((ima.m[i][j]>>4)%2)*255; imaoutbit5<<c;
        c=((ima.m[i][j]>>5)%2)*255; imaoutbit6<<c;
        c=((ima.m[i][j]>>6)%2)*255; imaoutbit7<<c;
        c=((ima.m[i][j]>>7)%2)*255; imaoutbit8<<c;
    }
}
```



(a)原始影像



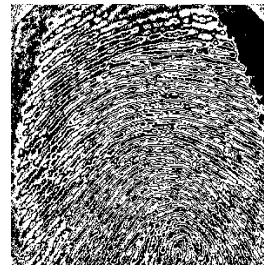
(b) bit-plane7



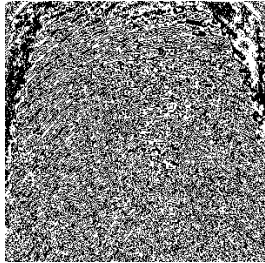
(c) bit-plane6



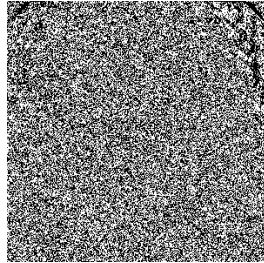
(d) bit-plane5



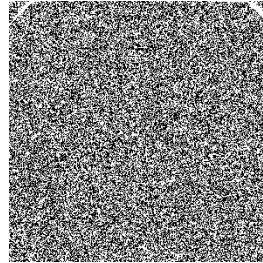
(e) bit-plane4



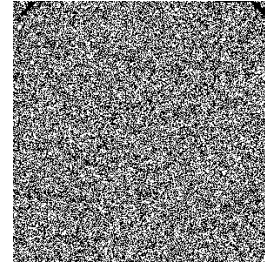
(f) bit-plane3



(g) bit-plane2



(h) bit-plane1



(i) bit-plane0

9.將 kaoshiung512x512 影像 down-sampling 1/4 倍輸出。

Source Code :

```

#include<fstream.h>
#include"array.h"

void main()
{

    ifstream in("kaoshiung512x512.raw",ios::binary);
    ofstream out("kaoshiung128x128.raw",ios::binary);

    uc2D ima;
    ima.Initialize(512,512);

    char c;

    for(int i=0;i<ima.nr;i++)for(int j=0;j<ima.nc;j++)
    {
        in.get(c);ima.m[i][j]=c;
    }

    for(int i=0;i<ima.nr;i=i+4)for(int j=0;j<ima.nc;j=j+4)
        out<<ima.m[i][j];
}

```



(a)



(b)

(a)原始影像 kaoshiung512x512.raw 、(b)down-sampling 1/4 影像 kaoshiung128x128.raw