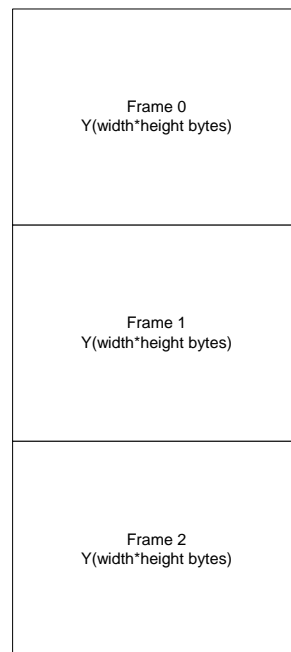


Video Sequence 格式說明、讀取／寫入方法、撥放程式

1. Video Sequence 格式說明

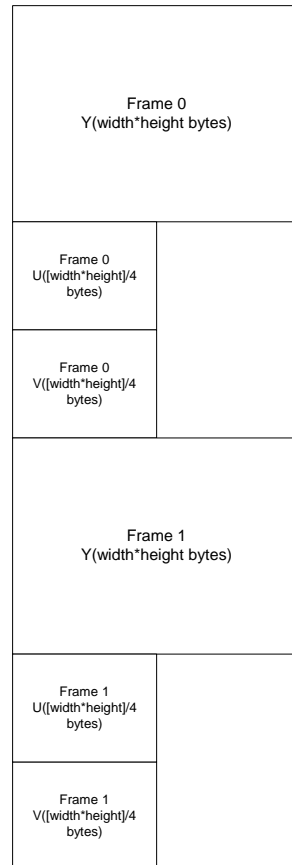
Video sequence 的格式千奇百怪，不過在課堂上我們只會給各位兩種格式，一種是*.y，這種格式只有灰階的部分，一種是*.yuv，是彩色的 sequence。基本上，不管是那一種，都可以看成是好幾張 raw 格式（沒有任何檔頭）的圖檔連結起來成為一個大的檔案。

*.y 的檔案格式可見圖一，若 width=352，height=288，也就是 CIF 的格式，則此檔案的一開始 $352*288=101376$ bytes 的資料為 frame 0 的資料，接下來的 101376 bytes 為 frame 1 的資料，以此類推。



圖一、*.y 檔案格式

*.yuv 檔案的格式也是類似，可參見圖二，我們提供的都是 4:2:0 的 sequence。若 width=352，height=288，也就是 CIF 的格式，則此檔案的一開始 $352*288=101376$ bytes 的資料為 frame 0 的 Y，接下來的 $101376/4=25344$ bytes 為 frame 0 的 U，接下來的 25344 bytes 為 frame 0 的 V，以此類推。



圖二、*.yuv 檔案格式(4:2:0)

2. 讀取／寫入方法

讀取和寫入的方法和讀 raw 圖檔的方法也是一樣，只是一個檔案裡面有很多張而已。不過因為 video 的資料量很大，所以一般建議用 `fread` 配合一維陣列的方式來讀取，速度會較快，範例程式如下：

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    FILE *fptr1, *fptr2;
    unsigned char *img = new unsigned char [101376];
    int fnum;
    int width, height, i, j;

    width=352;
    height=288;
```

```

fptr1=fopen("stefan_part_cif.y", "rb");
fptr2=fopen("out_stefan_part_cif.y", "wb");

for(fnum=0; fnum<100; fnum++)
{
    printf("\n%d", fnum);
    fread(img, 1, 101376, fptr1);

    for(j=0; j<height; j++)
    for(i=0; i<width; i++)
    {
        img[j*width+i]= img[j*width+i]/2;
    }

    fwrite(img, 1, 101376, fptr2);
}

fclose(fptr1);
fclose(fptr2);

return 0;
}

```

這個程式把 stefan_part_cif.y 中的每一張 frame 讀出存入 img 陣列中，之後把 img 陣列中的每一個 pixel 除以 2 後回存，也就是把亮度減半的意思，最後再把 img 陣列寫出成 out_stefan_part_cif.yuv 之檔案。值得注意的是，因為 img 現在是一維陣列，所以要 access (i, j)那一點的值，要用 img[j*width+i]。

3. 撥放程式

網站上放了一個叫個 Display.exe 的程式，此程式可以用來撥放 sequence，使用方法請見網站上之說明。