

Bitmap 作業 (II)

資訊之芽語法班 2015 suhorng

複習: Bitmap 處理程式

- 輸入檔: sprout03_in.bmp (跟 .exe 放在一起)
- 輸出檔: sprout03_out.bmp
- 程式格式
 - 引入 bmp_hdlr.h (跟 .cpp 放在一起)
 - 宣告三個全域二維 int 陣列, 如下

```
#include <iostream>
#include "bmp_hdlr.h"

int canvas_r[bmp_size][bmp_size];
int canvas_g[bmp_size][bmp_size];
int canvas_b[bmp_size][bmp_size];

int main() {
    // 在這裡處理圖片

    // 對 canvas_r, canvas_g, canvas_b 的修改,
    // 在程式結束後會自動存檔
    return 0;
}
```

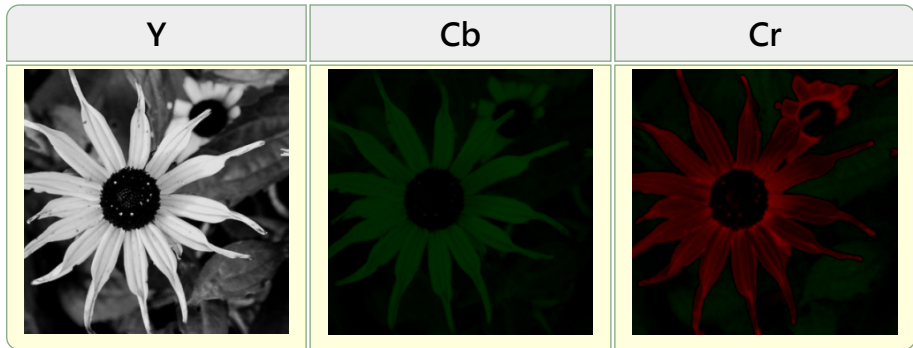
複習: 黑白圖片

- 圖片轉成黑白



複習: 黑白圖片

- 再加兩個 component 來表達色彩
- C_b , C_r 取樣可以較少, 節省空間



範例 - YCbCr 色彩空間

- http://en.wikipedia.org/wiki/YCbCr#JPEG_conversion
- RGB 轉換至 YCbCr ($R, G, B \in [0, 255]$)
 - $Y = 0.299R + 0.587G + 0.144B$
 - $C_b = 128 - 0.168736R - 0.331264G + 0.5B$
 - $C_r = 128 + 0.5R - 0.418688G - 0.081312B$
- YCbCr 轉換至 RGB
 - $R = Y + 1.402(C_r - 128)$
 - $G = Y - 0.34414(C_b - 128) - 0.71414(C_r - 128)$
 - $B = Y + 1.772 * (C_b - 128)$

範例 - 誤差擴散 (Error Diffusion / Dithering)

- http://en.wikipedia.org/wiki/Error_diffusion
- $P' = \lfloor P/16 \rfloor \times 16$, 誤差 $E = P - P'$
- 轉成低畫質可配合抖動降低視覺影響:

$$\frac{1}{48} \begin{bmatrix} - & - & \mathbf{P} & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$$



範例 - 圖片銳化

- 模糊化 \Rightarrow 調亮度 \Rightarrow 與原始圖片相減 \Rightarrow 調亮度
 - 降低低頻的 component



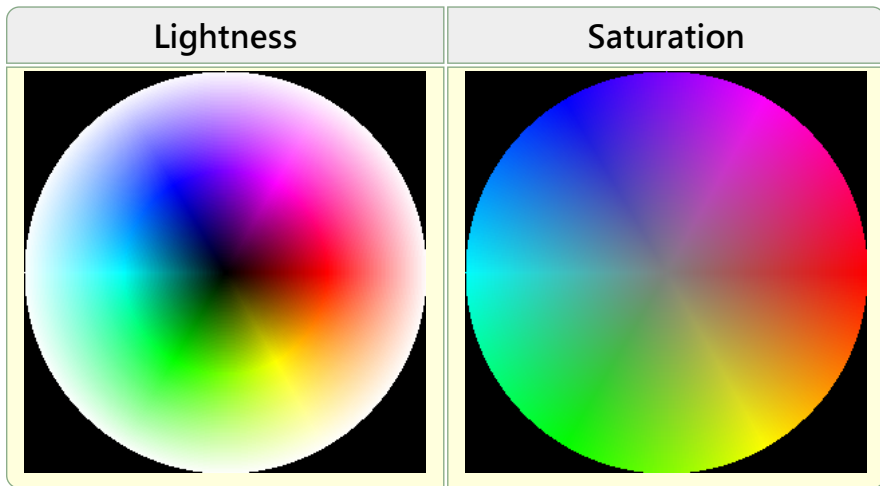
範例 - 圖片銳化

- 模糊化 \Rightarrow 調亮度 \Rightarrow 與原始圖片相減 \Rightarrow 調亮度
- 對 R, G, B 分別調整, 或只調整 Y component?



範例 - HSL 色彩空間

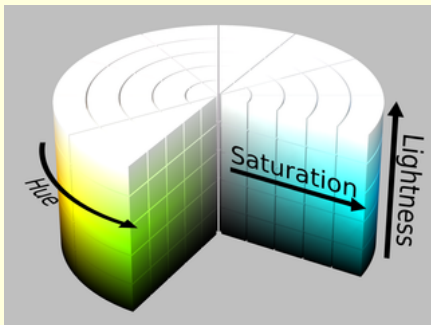
- http://en.wikipedia.org/wiki/HSL_and_HSV
- 有 Hue (色相), S (飽和度), L (亮度) 三個 components



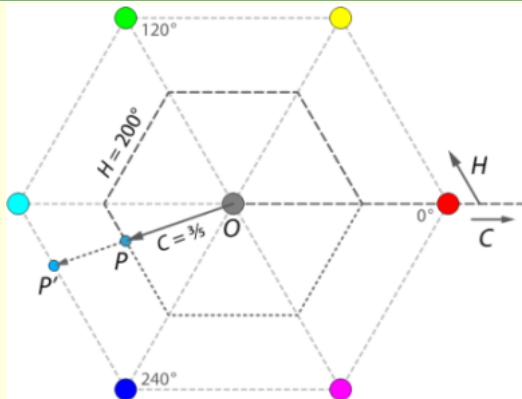
範例 - HSL 色彩空間

- $H \in [0, 360)$ (度), $S \in [0, 100\%]$, $L \in [0, 100\%]$
- 底下給的轉換中, 將假設 $R, G, B \in [0, 1]$, $S, L \in [0, 1]$. (所以 R, G, B 需要從 $[0, 255]$ 轉換過來)

SharkD, wiki



Jacob Rus, wiki



範例 - HSL 色彩空間

- RGB 轉換至 HSL; ($x \bmod n = x - \lfloor x/n \rfloor \times n$)
 - 令 $M = \max\{R, G, B\}$, $m = \min\{R, G, B\}$, $C = M - m$
 - 則

$$L = (M + m)/2$$

$$S = \begin{cases} 0, & L = 0 \\ \frac{C}{1 - |2L - 1|}, & L \neq 0 \end{cases}$$

$$H = 60^\circ \times \begin{cases} 0, & C = 0 \\ \frac{G-B}{C} \bmod 6, & R \geq G \ \& \ R \geq B \\ \frac{B-R}{C} + 2, & G \geq R \ \& \ G \geq B \\ \frac{R-G}{C} + 4, & B \geq R \ \& \ B \geq G \end{cases}$$

範例 - HSL 色彩空間

- HSL 轉換至 RGB; ($x \bmod n = x - \lfloor x/n \rfloor \times n$)

- 令

$$C = S(1 - |2L - 1|)$$

$$X = C \left(1 - \left|\frac{H}{60} \bmod 2 - 1\right|\right)$$

$$m = L - C/2$$

- 則 $(R', G', B') = \begin{cases} (C, X, 0), & 0 \leq H \ \& \ H < 60 \\ (X, C, 0), & 60 \leq H \ \& \ H < 120 \\ (0, C, X), & 120 \leq H \ \& \ H < 180 \\ (0, X, C), & 180 \leq H \ \& \ H < 240 \\ (X, 0, C), & 240 \leq H \ \& \ H < 300 \\ (C, 0, X), & 300 \leq H \ \& \ H < 360 \end{cases}$

- $(R, G, B) = (R' + m, G' + m, B' + m)$

複習(?) - 三角函數與亂數

- 三角函數 (輸入是徑度)

```
#include <cmath>

// 獲得  $\pi$  的值
const double PI = std::acos(-1.0);

// 三角函數
std::sin(x*PI/180.0)      //  $x^\circ$  的 sin 值
std::cos(x*PI/180.0)      //  $x^\circ$  的 cos 值
std::atan2(y,x)*180.0/PI  //  $\arctan(y/x)$  轉成度數, 在  $[-180^\circ, 180^\circ]$  之間
```

- 亂數 (不是均勻分佈的)

```
#include <cstdlib>

// 初始化
std::srand(std::time(NULL))

std::rand()                // 產生亂數
std::rand()%maximum        // 產生  $[0, \text{maximum}-1]$  的亂數
std::rand()%(maximum - minimum) + minimum  // 產生  $[\text{minimum}, \text{maximum}]$  的亂數
```

範例 - 產生亂數圖片 (I)

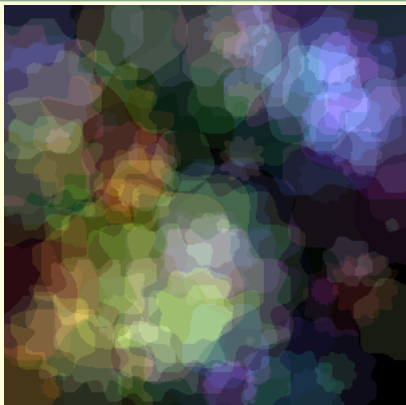
- 例子: `bmp_random_circle.cpp`
- 隨機生成圓形疊加到圖片上; 盡量讓圓心在圖片外
- 疊加的顏色是隨機的



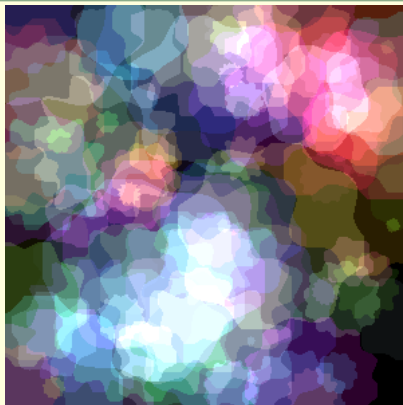
範例 - 產生亂數圖片 (II)

- 例子: `bmp_random_collage.cpp`
- 把圓形的半徑大小做變化; $r(\theta) = R + \Delta \sin(C\theta)$
- 疊加的顏色是隨機的, R, Δ, C 亂數產生

Collage 1 (%30)



Collage 2 (%50)



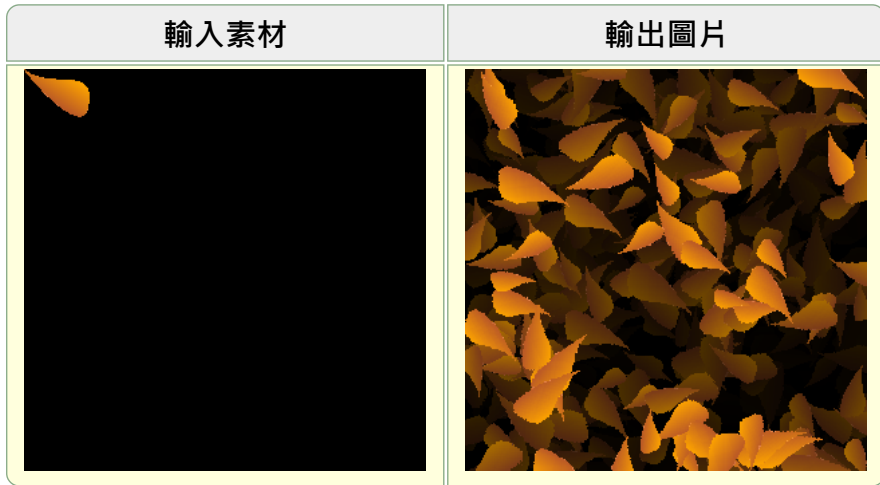
範例 - 產生亂數圖片 (II)

- 例子: `bmp_random_collage.cpp`
- 把圓形的半徑大小做變化; $r(\theta) = R + \Delta \sin(C\theta)$
- 疊加到黑 → 白漸層的背景上



範例 - 產生亂數圖片 (III)

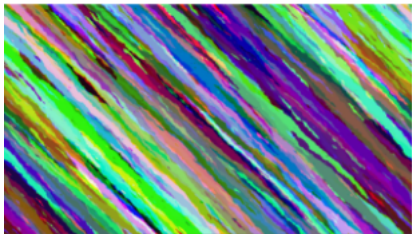
- 例子: `bmp_random_leaf.cpp`
- 把輸入圖片的素材隨機貼到輸出圖上
- 隨機旋轉, 縮小, 調整明暗



Tweetable Mathematical Art

<http://codegolf.stackexchange.com/questions/35569/>

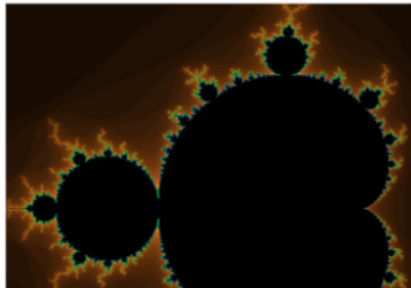
Random painter



popularity contest - Tw

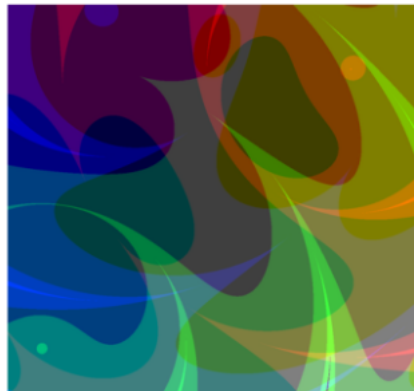
codegolf.stackexchange.com/questions/35569/tweetable-mathem

Of course, there has to be a Mandelbrot submission.



Some swirly pointy things

146 Yes, I knew exactly what to name it.

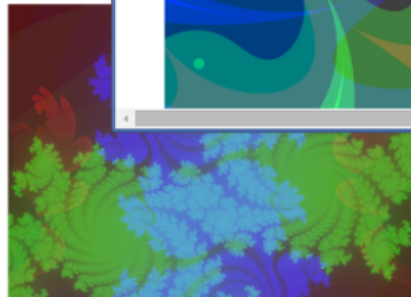


popularity contest -

codegolf.stackexchange.com/questions/35569/tweetable-mathem

Julia sets

99 If there's a Mandelbrot



smallpt Global Illumination in 99 lines of C++

<http://www.kevinbeason.com/smallpt/> Ray Tracing!



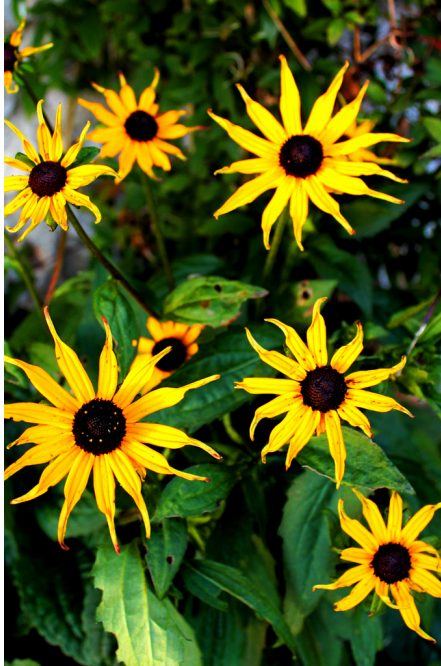
作業繳交與評分規則

- 寄到「`sprout+prog15@csie.ntu.edu.tw`」
 - 標題: 「[資訊之芽] [語法班] Bitmap 作業姓名」
 - 請把以上樣板複製貼上; 不含雙引號
- 內容: 壓縮檔, 包含
 - 一份 README 文件(例), 格式不拘, 至少描述你的設計
 - 你的程式碼 (`.cpp`)
 - 輸入與輸出的圖片
- 評分:
 - 60-80: 完成 Sobel filter, 依 coding style 評分
 - 80-100: 其他設計/嘗試 如亂數圖片

Remark

- 用亂數生成圖案 顏色怎麼辦?
 - 也許嘗試用 HSL 產生顏色再轉回去
 - 特殊的碎型集合: 如 Mandelbrot set, Julia set, etc
- 如何填滿一個區域? (advanced)
 - 關鍵字: flood-fill
 - 需要深度優先搜尋(depth-first search)或廣度優先搜尋(breadth-first search)的知識

圖片來源



- By Les Haines,
14 Sep 2013.
- Released under CC
license 2.0