

2023 Spring Digital Image Processing Term Project

Report Team 18

0. 分工

p11922004 任祖頤：程式整合、UI、DEMO 影片、膠片製作

b07902038 林晨煦：程式(Step1-3)、Report

b08902074 張正源：程式(Preprocessing、Step0 等)、簡報

1. Title

VISUAL CRYPTOGRAPHY BASED ON VOID-AND-CLUSTER HALFTONING TECHNIQUE[1]

2. Motivation:

在課堂上看到能用兩張圖片疊合出隱藏的圖片覺得很新鮮，就想自己做一個。此技術可用於在電腦裡正大光明地以圖片保存秘密訊息，或是在黑白印刷的圖片中祕密加上浮水印。

3. Problem definition

給與兩張灰階圖片與一張目標黑白圖片，將兩張灰階圖片各自經 halftoning 再重疊後，可清楚辨認出目標黑白圖片所蘊含的訊息，而兩張圖片各自皆無法辨認目標黑白圖片。Halftoning 的方法為 thresholding，而 threshold array 由可疊合出目標黑白圖片的隨機雜訊需經由 void and cluster(VAC)[2]演算法產生。使用的演算法結合了 VAC 與視覺密碼學(HVC)，故稱為 VAC-HVC。

4. Algorithm

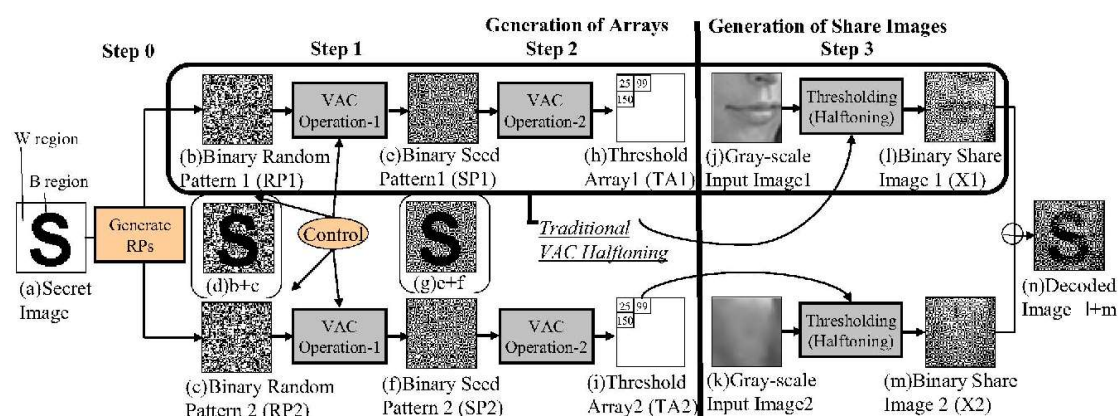


Fig. 3 Proposed VAC-HVC algorithm

上圖出自論文[1]。

Pre-processing :

準備兩張灰階圖片與一張黑白圖片，大小皆為 $M \times N$ ，其中 M 為高度， N 為寬度。兩張灰階圖片的亮度需在 64 至 192 之間。黑白圖片可由灰階圖片經任意 dithering 產生，原圖片對比度越高最終效果越好。本次由灰階圖片產生黑白圖片的方法為課程教過的 Jarvis dithering。

Step 0) 產生一對 Random Binary Pattern Shares (RPs)：

這個階段將利用 Secret 生成一對 Random Pattern(RP)。

設 Secret 中白色(亮度為 255)畫素座標的集合為 W ，黑色(亮度為 0)畫素座標的集合為 B ，隨機生成兩個 RP，分別稱為 RP1 和 RP2。

設 1 代表白色，0 代表黑色。對於 RP1 和 RP2 中座標相同的畫素對 $(rp1[i][j], rp2[i][j])$ ，若 (i, j) 屬於 W ，則隨機設為 (0, 0) 或 (1, 1)，重疊後有 50% 機率為黑色；若座標屬於 B ，則隨機設為 (1, 0) 或 (0, 1)，重疊後有 100% 機率為黑色。最終在 B 和 W 兩個區域中黑白畫素皆各佔一半。

Step 1) 產生一對 Binary Seed Pattern Shares (SPs)：

這裡使用的演算法為 VAC 前半段的修改版本：

i) 分別計算 RP1 和 RP2 的 dither array(DA)。

$$DA(i, j) = \sum_{(x-i)^2 + (y-j)^2 \leq r^2} \neg RP(x, y) \cdot \exp\left(-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}\right), \text{ 其中}$$

r 取 5， σ 取 1.5。

ii) 從 RP1 和 RP2 隨機挑 1 張圖。

iii) 從圖中的黑色畫素中找出最大聚集位置，並判定其位於 W 還是 B 。

標記該黑色畫素。

iv) 從圖中同一區域的白色畫素中找出最大空白位置。

標記該白色畫素。

v) 將前 2 步找到的畫素交換，另一張圖的相同位置也要做。更新 DA。

vi) 判定 iii) 時標記的黑色畫素移除後，是否產生最大空白。

如果否，則回到 i)；如果是，則換回來後結束。

將產生的兩張圖分別稱為 SP1 和 SP2。

Step 2) 產生一對 Threshold Arrays (TAs)：

這個階段分別由 SP1 和 SP2 生成一對 Threshold array(TA)：TA1 和 TA2。

這裡使用的演算法是 VAC 演算法的後半部分，一共有以下步驟：

i) 計算每個像素的 DA 值，filter 為半徑最大為 15 的圓形。

初始化 TA，數值範圍為 0 至 $MN-1$ 。

ii) 使用初始的 SP，每一輪，找出 DA 值最大的黑點，將 TA 對應座標的值設為 rank，接著從 SP 中將黑點設為白色並更新 DA，直到所有黑點都被移除。

iii) 使用初始的 SP，每一輪，找出 DA 值最小的白點，將 TA 對應座標的值設為 rank，接著從 SP 中將白點設為黑色並更新 DA，直到所有白點都被移除。

rank 在 ii) 中是當前黑點的數量-1，在 iii) 中是當前黑點的數量。

iv) 輸出 $TA' = \text{round}\left[\frac{255}{MN}(TA + 0.5)\right]$ ，其中 round 為四捨五入的函數。

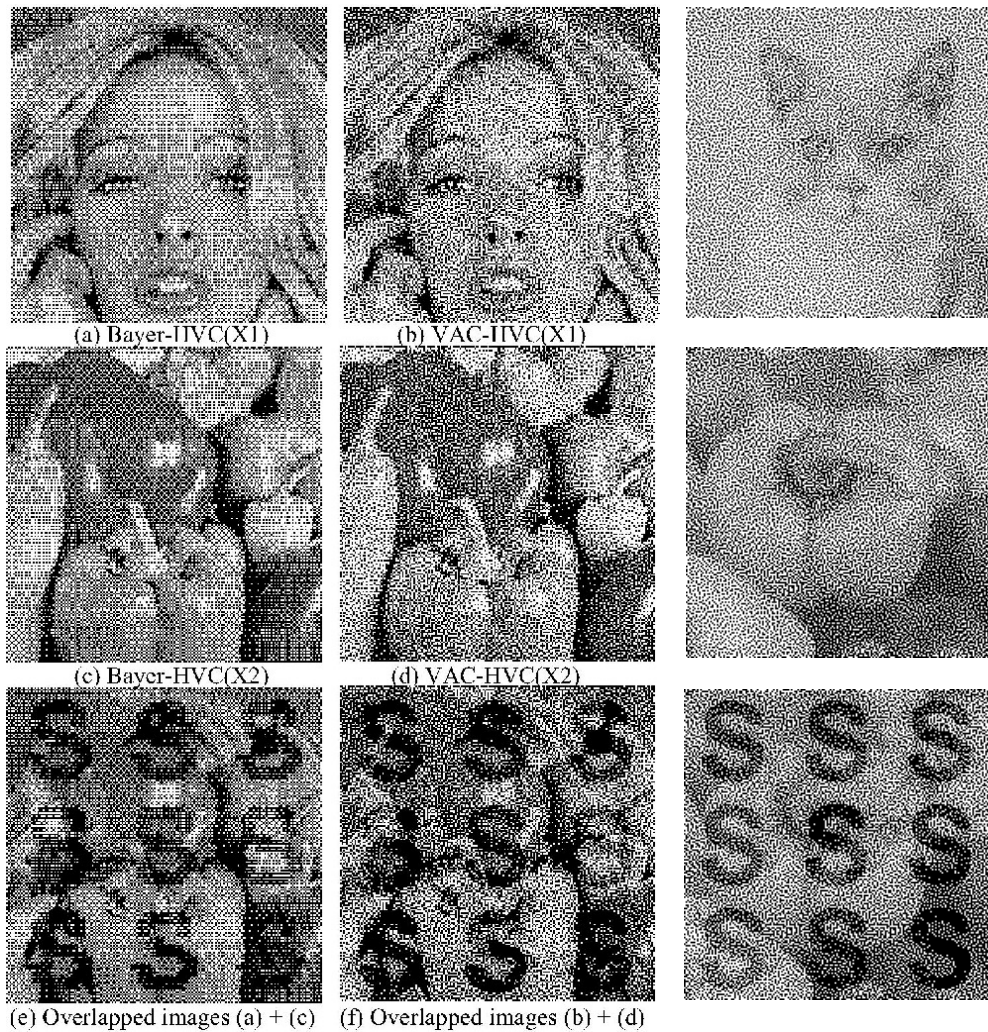
Step 3) 產生 Binary Share Images (Xs) :

這個階段分別把 TA1 和 TA2 套用至 2 張不同的灰階圖片，產生最終結果：X1 和 X2。將 X1 和 X2 重疊，即可辨認出 Secret。

使用的演算法如下：

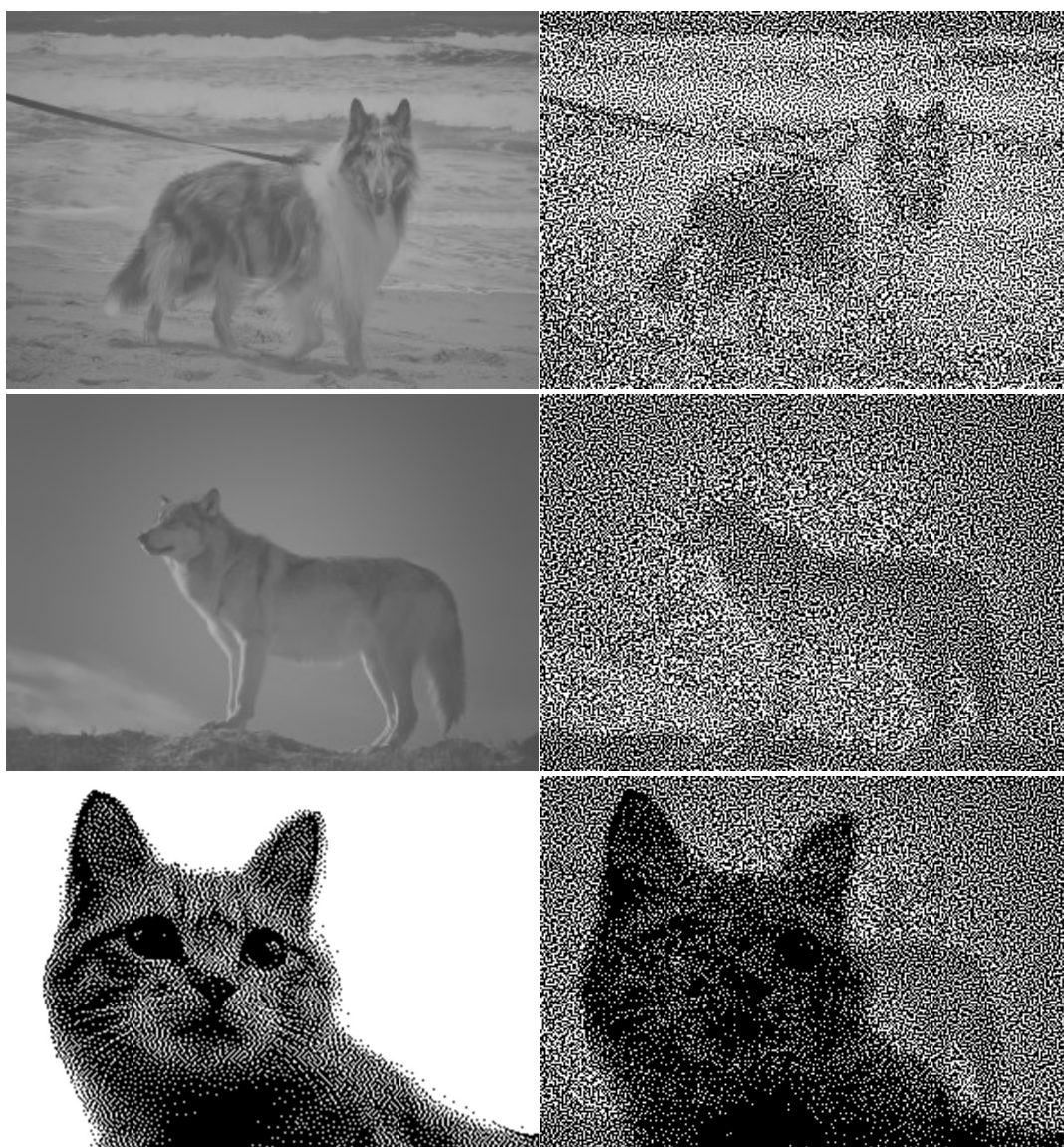
對圖片中所有畫素 $I(i, j)$ ，如果 $I(i, j) > TA(i, j)$ ，則 $X(i, j) = 1$ ，反之則為 0。

5. Results



上圖左側為 Bayer-HVC，中間為論文[1]中使用的 VAC-HVC，右側為我們的結果。論文的討論著重於隱藏文字的效果，指出兩種 HVC 皆能隱藏訊息，兩張圖重疊後皆能清楚顯示訊息。我們也成功複製出論文的結果，使用 2 張 192×192 的圖片，重疊後顯示出隱藏訊息「S」。

我們也試著重現先前課堂上將 2 張狗狗的圖片重疊後，顯示貓咪圖片：



所有圖片的尺寸皆為 297 x 210。左邊為原圖，右邊分別為 X1、X2 與重疊的結果。重疊後能清楚分辨貓臉，而背景僅能略為分辨出狗身體。

6. Discussion

(1) VAC1 所需的回合數

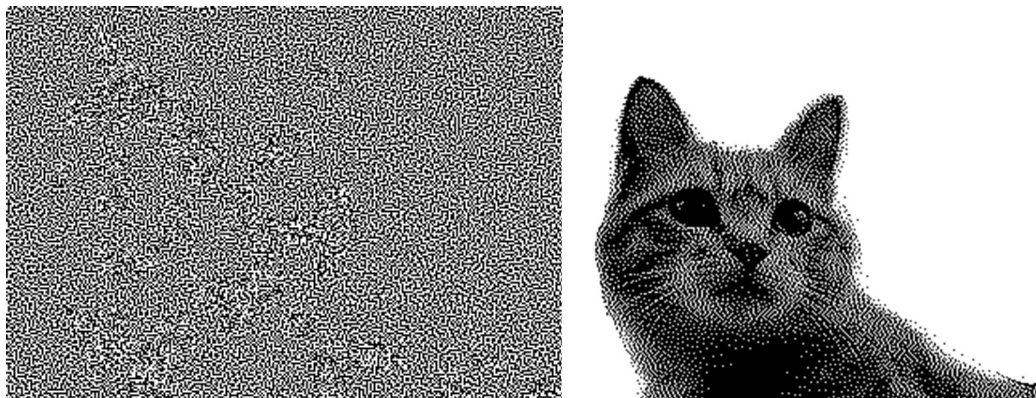
下表為不同畫素下執行 VAC1 100 次中所需的交換次數，超過畫素數*100 的值忽略不計：

圖片畫素	最小值	最大值	平均數	標準差	離群值數
1024	171	1017	426.63	186.31	2
4096	1541	365073	29386.25	52096.80	17

離群值為超過畫素數 100 倍的值。超過 4096 畫素由於交換次數過多，故無法統計。為了確保 VAC1 能在合理的時間內結束，設定了交換次數上限=畫素數/4，犧牲均勻程度以換取合理的執行時間。另外發現，當 B 或 W 區域內黑白畫

素數量相差過多時，VAC1 可能不會結束。

我們嘗試過將圖片分割成多個區域，利用前面區間的結果計算所有區間，得到的結果(SP1)如下方左圖，而右圖為 secret：

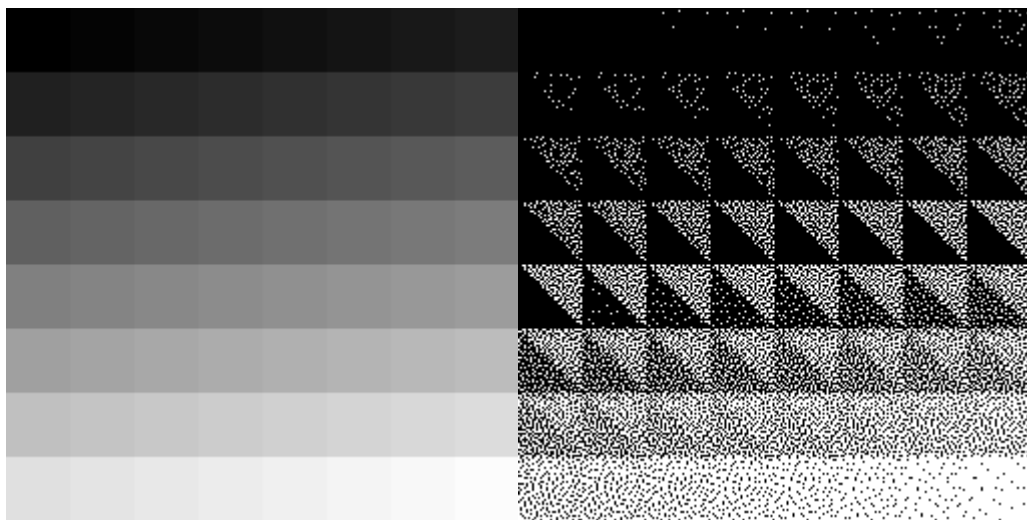


可發現有些區域比較亂，與其它區域不一致。由於會影響保密，這個方法最終遭到棄用。

(2) 灰階圖片的亮度區間

論文[1]中僅建議若要使隱藏訊息能被清楚解讀，灰階圖片的亮度需盡量位於灰階值的中間，並未指出具體數值，因此我們做了以下測試。

下圖左邊為包含不同亮度的圖片，範圍為[0,252]；右邊為同一組 TA 分別套用在同一張測試圖片，疊起來後的結果：



可發現當灰階圖片亮度<32(太黑)或>192(太白)時，三角形難以辨別，而在亮度=127 時有最好的效果。因此建議開始前調整灰階圖片的亮度至[32, 192]。本報告圖片使用的亮度區間為[64, 192]。

(3) 時間複雜度與執行時間

論文[1]指出，在特定設備下，對於 192 x 192 的圖片，從原圖生成 share image 只要 1 秒，並未指出 VAC-HVC 演算法本身的時間複雜度(下稱複雜度)。我們剛開始實作時未注意複雜度需要最佳化，跑一次 297 x 210 的圖需要數小時。

假設圖片的寬度為 M ，高度為 N ，其中 $M < N$ ，filter 半徑為 r ，則各操作的

複雜度如下：

計算整張圖的 DA 為 $O(M^3N)$ 。若 r 與 M 無關，則為 $O(MNr^2)$ 。

尋找 DA 的極值與極值座標需要 $O(MN)$ ，

更新 1 次 DA 需重新計算整張圖 DA，故為 $O(M^3N)$ 。若 r 與 M 無關，則為 $O(MNr^2)$ 。如果僅重新計算更新的點在 filter 範圍內所有點的 DA，能簡化為 $O(r^2)$ 。

如果使用適當的資料結構，尋找並更新 DA 能在 $O(\log(MN))$ 時間完成，但在本次使用的圖片尺寸範圍內常數項過大，因此不採用。

接下來計算每一步的複雜度，忽略 r 對複雜度的影響：

Step 0 複雜度為 $O(MN)$ ，假設取亂數為 $O(1)$ 。

Step 1 需計算 1 次整張圖的 DA、尋找 $O(MN)$ 次極值及更新 $O(MN)$ 次 DA，複雜度為 $O(M^2N^2)$ 。

Step 2 需計算 1 次整張圖的 DA、尋找 MN 次極值及更新 MN 次 DA，複雜度為 $O(M^2N^2)$ 。

Step 3 是簡單的 thresholding，複雜度為 $O(MN)$ 。

全體的複雜度為 $O(M^2N^2)$ 。

程式語言為 python 加上 Numba 即時編譯器，對 297×210 的圖片，在 CPU 為 1.6GHz 的筆電上執行，Step 1 在限制交換次數後約需要 25 秒，Step 2 約需要 10 秒，其中編譯大約需要 3 秒。

(4) Filter 半徑

在原論文中，Filter 半徑為 $M/2$ 。由於時間複雜度過大，有必要改進半徑的值。

在 VAC1 中，filter 半徑 = 5。對於半徑為 r 上的點，其影響總和與 $r \cdot e^{-\frac{r^2}{4.5}}$ 成正比，在 $r > 1.5$ 時快速減少，因此將 filter 半徑設為 5。

在 VAC2 中，filter 半徑 = 15。由於最後生成的 TA 數值將只有 255 種，故使半徑略大於 $\sqrt{255} - 1$ 。另根據測試，半徑超過 15 時，最後得到的 dither array 與半徑為 15 時相同。

7. Reference

所有相片皆來自 <https://www.pexels.com>，根據其授權條款改造使用。

[1] Myodo, Emi, Shigeyuki Sakazawa, and Yasuhiro Takishima. "Visual cryptography based on void-and-cluster halftoning technique." 2006 International Conference on Image Processing. IEEE, 2006.

[2] R. A. Ulichney, "The void-and-cluster method for dither array generation", Proc. SPIE, Human Vision Visual Processing, Digital Displays IV, Vol. 1913, pp. 332-343, 1993.