

## Pixel value differencing

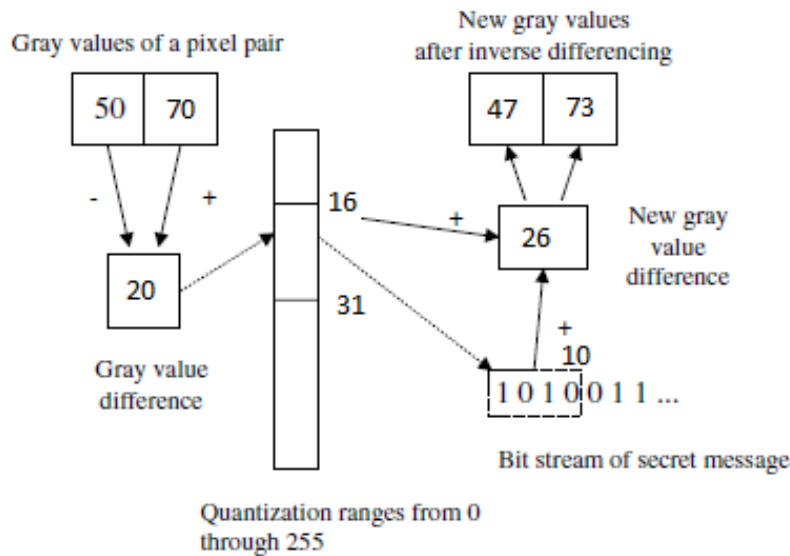
Pixel value differencing (PVD) 是一個資料隱藏的演算法，相較於傳統最低位元藏密法 (LSB)，可藏入較多的資料。從名字上看，就是利用像素的差值來進行資料隱藏。

在灰階圖像上，每一個點都是用一個 Byte 大小的數字來表示這個點的像素值，秘密訊息就會被嵌入在相鄰兩像素的差值裡。由於每個像素值都是介於 0 到 255，差值也就自然的介於 0 到 255，如果每次都直接將秘密訊息替換成差值，那麼圖像隱藏前與隱藏後的差異用肉眼即可辨識。因此我們需要將這些像素差值做分類，讓每個差值只會在該特定區間內做替換。

$K$	1	2	3	4	5	6
$R_k$	0~7	8~15	16~31	32~63	64~127	128~255
$n$	3	3	4	5	6	7
$u_k$	7	15	31	63	127	255
$l_k$	0	8	16	32	64	128

表一

表一為區間的分類，在 0 到 255 的差值中分成了 6 個區間， $K$  為該區間的流水號， $R_k$  為差值的絕對值， $u_k$  為該區間的上限值， $l_k$  為該區間的下限值， $n$  為可藏入的 bit 數。 $n$  的算法為  $\log_2(u_k - l_k + 1)$ ，也就是該區間的長度來決定可藏入的 bit 數。



圖一

了解了區間的概念後，我們開始進行 PVD 隱藏。圖一為 PVD 隱藏示意圖，右下角的 bit stream 為秘密訊息，左邊是隱藏前的相鄰兩像素(70,50)以及它們的差值 20，中間是判斷差值落在哪個區間(K=3)，判斷出第三個區間後，可藏入的 bit 數 n 為 4，取 bit stream 4 個 bit 1010，轉為 10 進位為 10，加上  $l_3$ (區間下限值) 16 後得到新的差值 26，然後將新舊差值的差平均分配到兩個像素裡完成資料隱藏。

實際例子有概念後接下來介紹公式的運算。相鄰兩像素( $P_i, P_{i+1}$ )， $d = P_{i+1} - P_i$ ， $b$  為秘密訊息， $d'$  為新的差值， $d'$  的計算方式如下：

$$d' = \begin{cases} l_k + b & \text{for } d \geq 0; \\ -(l_k + b) & \text{for } d < 0 \end{cases} \quad \text{----- (1)}$$

有了新的差值後，接下來將新舊差值的差異( $d' - d$ )依照下列公式平分給兩個像素：

$$(P'_i, P'_{i+1}) = \begin{cases} \left( P_i - \left\lfloor \frac{d'-d}{2} \right\rfloor, P_{i+1} + \left\lfloor \frac{d'-d}{2} \right\rfloor \right) & \text{when } d \text{ is even;} \\ \left( P_i - \left\lfloor \frac{d'-d}{2} \right\rfloor, P_{i+1} + \left\lfloor \frac{d'-d}{2} \right\rfloor \right) & \text{when } d \text{ is odd;} \end{cases} \quad \text{----- (2)}$$

( $P'_i, P'_{i+1}$ ) 為新的像素值，完成 PVD 資料隱藏。

要取出資料時，計算出相鄰兩像素差值後，查表一扣掉該差值所屬的區間下限值，取得訊息。

## Matrix Embedding Steganography

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Syndrome	Coset leader							
000	000000	011100	101010	110001	110110	101101	011011	000111
100	100000	111100	001010	010001	010110	001101	111011	100111
010	010000	001100	111010	100001	100110	111101	001011	010111
001	001000	010100	100010	111001	111110	100101	010011	001111
011	000100	011000	101110	110101	110010	101001	011111	000011
101	000010	011110	101000	110011	110100	101111	011001	000101
110	000001	011101	101011	110000	110111	101100	011010	000110
111	100100	111000	001110	010101	010010	001001	111111	100011

表二

表二為(6,3)linear block code，每一欄的第一列分別為[0,0,0]、[1,0,0]、[0,1,0]、[0,0,1]、[1,1,0]、[1,0,1]、[0,1,1]、[1,1,1]乘上 G 所得到的向量，表中除了第一列的 code words、第一欄的 syndrome、第二欄的 coset leader 之外的每個值都是由 code words 與 coset leader 做 exclusive or。

表二有以下兩種特性：64 個 6 bit stream(r)皆不重複、任意一個 r 乘上  $H^T$  會得到該列最前面的 syndrome。利用此兩種特性便可做資料隱藏。

c(101100)為 cover data，m(010)為 message，c 的 syndrome 為  $cH^T = 110(s)$ ，將 cover data 的 syndrome 與 message 做 exclusive or 得到  $s'(100)$ ， $s'$  相對應 coset leader 為 100000(e)，再將 e 與 c 做 exclusive or 得到 001100( $c'$ )完成資料隱藏。取資料時算出 cover data 的 syndrome( $[001100] \times H^T$ )便得到隱藏 message。

公式運算：

$$s' = cH^T + m \quad \text{-----}(3)$$

$$s' = eH^T \quad \text{-----}(4)$$

$$c' = c + e \quad \text{-----}(5)$$

$$c'H^T = (c + e)H^T \quad \text{(式子 5 代入)}$$

$$= cH^T + eH^T$$

$$= cH^T + s' \quad \text{(式子 4 代入)}$$

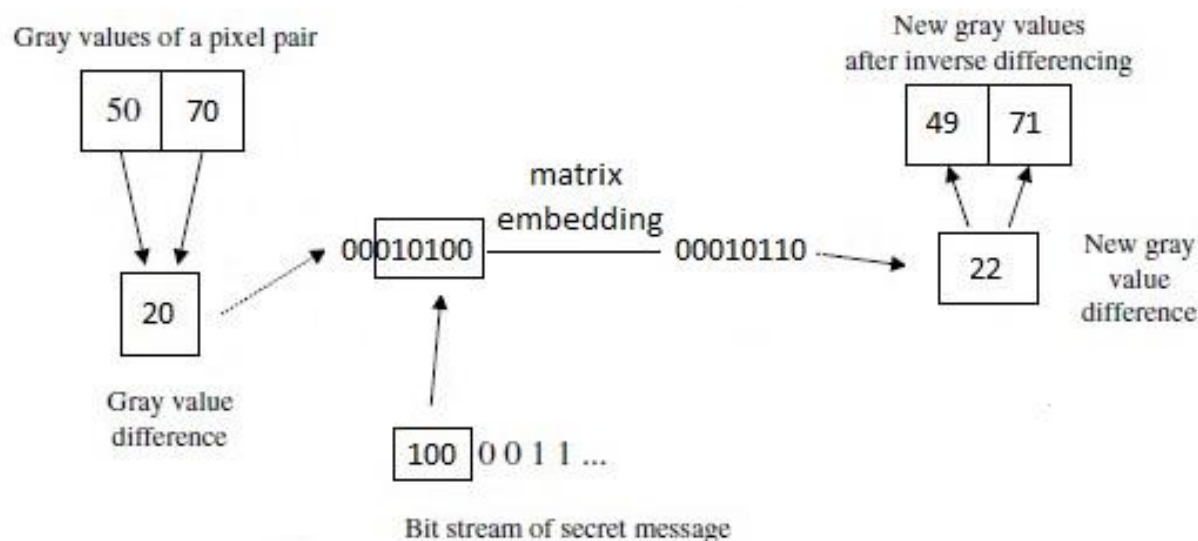
$$= cH^T + cH^T + m \quad \text{(式子 3 代入)}$$

$$= m \quad \text{-----}(6)$$

(3)為 cover data 的 syndrome 與 message 做 exclusive or 得到  $s'$ ，(4)為  $s'$  與 e 的關係，(6)為取出秘密訊息(計算 cover data 的 syndrome)。

## Matrix Embedding Steganography in PVD

結合已上 PVD 以及 Matrix Embedding，以下為新的演算法：



圖二

圖二為隱藏流程圖，首先與 PVD 一樣一次取兩個 Byte(50,70)，計算出差值之後(20)轉成 8bit stream(00010100)取後面 6bit(010100)，bit stream 取出 3bit(100)開始做 Matrix Embedding Steganography。

將 010100 乘上  $H^T$  得到 syndrome(001)，與 message(100) exclusive or 得到 101，查表二的 syndrome 與 Coset leader 對應，101 對應到 000010，將這 Coset leader 與 010100 做 exclusive or(010110)，Matrix Embedding Steganography 完成。

010110 在與之前的前兩個 bit 結合轉成十進位數值(22)，利用式子(2)將差值差異平均分配回兩個像素裡，完成隱藏。

要取回訊息時，計算出相鄰像素差值(71-49=22)，轉成 8bit stream 取出後面 6bit(010110)，乘上  $H^T$  得到的 syndrome(100)就是隱藏的訊息。

0	28	42	49	54	45	27	7
32	60	10	17	22	13	59	39
16	12	58	33	38	61	11	23
8	20	34	57	62	37	19	15
4	24	46	53	50	41	31	3
2	30	40	51	52	47	25	5
1	29	43	48	55	44	26	6
36	56	14	21	18	9	63	35

786.5K cover image PSNR

method message	pvd	Pvd-lsb	Pvd matrix embedding
120K(15%)	39.0999	36.9914	35.1205
270K(34%)	(155.9K 19%) 39.0999	32.4488	31.2288
368.3K	(158.3K 20%) 39.0999	(288.4K 36%) 32.816	(296.7K 37%) 30.8069