基於像素值差異與矩陣嵌入應用於彩色影像集之研究

李東岳
國立暨南國際大學資訊工程學系

摘要

由於資訊科技的發展以及網際網路的便利性，使得資料得以快速傳輸。為了避免機密資料輕易被竊取，資料加密及資訊隱藏的技術成為了重要的研究課題。傳統的資訊隱藏使用最低位元替換藏密法（LSB），優點是很難被人類視覺系統察覺，但是藏密量不多。像素值差異藏密法（PVD）將機密訊息透過更改兩兩相鄰之像素值差異，按規則藏入影像中，大幅增加藏密量。矩陣嵌入(matrix embedding)是透過漢明碼的特性來隱藏資料，本文將會改良matrix embedding隱藏法，並與PVD做結合。

關鍵字：PVD，Matrix embedding，Hamming code，LSB，資訊隱藏。

1. 前言

在這資訊爆炸的時代，我們能在網路上傳遞任何媒體資料。在傳送資料前，為了不讓有心人士容易竊取我們的資料，可以將資料進行加密的動作，接收者接收時再進行解密。但若有心人士發現此資料是密碼，就會曉得這是有價值的資料，更提高他進行破解密碼的動機，這樣就會造成反效果。因此現今有了資料隱藏這項技術，它能夠將機密資訊隱藏在媒體檔案裡面，這樣一來只需要傳送一個看似正常的媒體檔案，就能夠將機密資訊傳送出去。

Bender等學者於1996年所提出的最低位元替換藏密法（Least Significant Bit, LSB）[1]，優點是藏密後很難被人類視覺系統察覺異狀，缺點是藏密量最多僅1bpp（bit per pixel）。此外，此LSB藏密技術容易被 Regular-Singular（RS）[2]偵密技術所察覺。Wu與Tsai學者於2003年提出的像素值差異演算法(Pixel Value Differencing, PVD)[3]，將祕密訊息透過更改兩兩相鄰之像素值差異，按規則藏入影像中，此方法不但增加藏密量，並且可避免被RS偵密技術所察覺。由於PVD在相鄰像素值差異小的時候，藏密量不高，在2005年，Hwang等學者[4]結合了改良後的LSB與PVD，將兩種演算法的優點結合在一起，提升了1.5倍的藏密量。

在矩陣嵌入(matrix embedding)裡，漢明碼(hamming code)資料隱藏是透過錯誤更正碼計算徵狀值(syndrome)的概念進行資料隱藏，其優點是掩護序列所更動的位元數會小於所藏入的位元數，藏密量相當驚人，固本篇論文嘗試將PVD與matrix embedding結合。在結合過程中發現漢明碼隱藏的演算法套用在影像中效果不佳，所以在文中也提出將漢明碼資料隱藏進行改良之法。

本論文共分為五節：第一節為前言，敘述本論文目的以及研究方法；第二節介紹PVD演算法、LSB與PVD如何結合，以及matrix embedding演算法；第三節介紹本文所提出的方法，結合PVD與改良後的matrix embedding；第四節為實驗數據，比較本文所提出的方法與原先的方法，並說明結果；第五節為結論與未來展望。

2. 相關文獻探討

2.1 像素值差異藏密法

在2003年Wu學者與Tsai學者所提出的像素值差異藏密法(PVD)[3]中，機密訊息是藏在影像中相鄰像素的差值裡。如果將機密訊息直接替換成差值，那麼圖像隱藏前後的差異用肉眼即可辨識。因此我們需要將像素差值做分類，讓差值只在該特定區間內做替換，如下表所示：

表一　依照差值大小分為六個區間

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

表一中R_k為相鄰像素差值，u_k與l_k為該區間的最大值與最小值，$n = \log_2(u_k - l_k + 1)$，為可藏入的位元數，以下為PVD演算法：

步驟一：經由式子(1)計算出影像中相鄰像素P_i與P_{i+1}差值d。

$$d = | P_i - P_{i+1} | \quad (1)$$

步驟二：判斷d屬於哪個區間，向機密訊息取n個位元後，將二進位的值轉為十進位計算出b。

步驟三：依照式子(2)計算出d’：

$$d' = \begin{cases} l_k + b & \text{for } P_i - P_{i+1} \geq 0 \\ -(l_k + b) & \text{for } P_i - P_{i+1} < 0 \end{cases} \quad (2)$$

步驟四：[4]依照式子(3)計算出新的像素值P'_i與P'_{i+1}，$m = |d' - (P_i - P_{i+1})|$。

$(P'_i,P'_{i+1}) =$

$$\begin{cases} \left(P_i + \left\lceil \frac{m}{2} \right\rceil, P_{i+1} - \left\lfloor \frac{m}{2} \right\rfloor\right), \text{if } P_i \geq P_{i+1} \text{ and } d' > d \\ \left(P_i - \left\lfloor \frac{m}{2} \right\rfloor, P_{i+1} + \left\lceil \frac{m}{2} \right\rceil\right), \text{if } P_i < P_{i+1} \text{ and } d' > d \\ \left(P_i - \left\lceil \frac{m}{2} \right\rceil, P_{i+1} + \left\lfloor \frac{m}{2} \right\rfloor\right), \text{if } P_i \geq P_{i+1} \text{ and } d' \leq d \\ \left(P_i + \left\lceil \frac{m}{2} \right\rceil, P_{i+1} - \left\lfloor \frac{m}{2} \right\rfloor\right), \text{if } P_i < P_{i+1} \text{ and } d' \leq d \end{cases} \quad (3)$$

要擷取訊息時，依照式子(4)計算出d'後，判斷d'屬於哪個區間，計算出d' – l_k便可得到訊息。

$$d' = | P'_i - P'_{i+1} | \quad (4)$$

例子：假設P_i = 50，P_{i+1} = 30，d = 20 ∈ R_3，機密訊息4個位元假設為$(1010)_2$，b = 10，d' = 26，P'_i = 50 +3 = 53，P'_{i+1} = 30 - 3 = 27，隱藏完成。擷取訊息時，計算出d' = 53 – 27 = 26，26 – 16 = $(10)_{10}$ = $(1010)_2$ 得到訊息。

2.2 像素值差異結合低位元替代藏密法

在2005年Hwang等學者對Wu學者與Tsai學者所提出的像素值差異藏密法(PVD)進行改良，在平滑區間(lower-level)裡改以低位元替代藏密(LSB)進行資料隱藏，與原本的PVD相較，提升了1.5倍的藏密量。在PVD的做法裡，會依照像素差值大小訂出六個區間，Hwang等學者以div(15)為界，將這六個區間分為lower-level與higher-level，如下圖所示：

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

lower-level higher-level

圖一　將區間分為higher-level 與 lower-level

利用LSB改善PVD的藏密法，其藏密演算法簡述如下：

步驟一：經由式子(1)計算出影像中相鄰像素P_i與P_{i+1}差值d。

步驟二：定義lower-level(if d ≤ div)或higher-level(if d > div)。

步驟三：如果為higher-level，進行2.1節所述之PVD隱藏；如果為lower-level，進行步驟四。

步驟四：從機密訊息S中取6個位元：m_1，m_2，m_3，m_4，m_5，m_6。將P_i後三個位元替換成m_1,m_2,m_3，P_{i+1}後三個位元替換成m_4,m_5,m_6，新的像素值為P'_i與P'_{i+1}。

步驟五：計算出新的像素差值d'。

步驟六：如果d'>div，進行下列公式調整。

$$(P'_i, P'_{i+1}) = \begin{cases} (P'_i - 8,\ P'_{i+1} + 8) & \text{if } P'_i > P'_{i+1} \\ (P'_i + 8,\ P'_{i+1} - 8) & \text{if } P'_i < P'_{i+1} \end{cases} \quad (5)$$

例子：假設P_i = 30，P_{i+1} = 15，S = 111000，div = 15，隱藏後P'_i = 31，P'_{i+1} = 8，d' = 23 > 15 = div，經過調整後 P'_i = 31 – 8 = 23，P'_{i+1} = 8 + 8 = 16，d' = 7 < div。

以下步驟為訊息還原演算法：

步驟一：利用式子(4)計算出影像中相鄰像素P'_i與P'_{i+1}差值d'。

步驟二：如果為higher-level，進行PVD擷取，若為lower-level，進行步驟三。

步驟三：分別對P'_i與P'_{i+1}做3-LSB擷取。例子：假設P'_i = 23，則3-LSB擷取訊息為111；P'_{i+1} = 16，則3-LSB擷取訊息為000。

2.3 Matrix embedding [5] [6]

利用Linear block code的特性進行資料隱藏。以下為利用(6,3)漢明碼資料隱藏方法，在六個掩護位元C裡隱藏3個位元m的資訊：

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (6)$$

步驟一：計算出徵狀值S = m ⊕ (C×H^T)。

步驟二：依照表二找出相對應S'。

表二　syndrome與coset leader對照表

S	S'
000	000000
100	100000
010	010000
001	001000
011	000100
101	000010
110	000001
111	100100

步驟三：計算出C' = C ⊕ S'，C'為完成資料隱藏的掩護序列。

要擷取訊息時，計算出C'× H^T可得機密訊息。

例子：假設C =$(101100)_2$，m = $(010)_2$，則S = $(100)_2$，S' = $(100000)_2$，最後C' = $(001100)_2$。擷取時，C'× H^T = 010還原訊息。

3. 改進方法

PVD隱藏法能夠依照像素差值大小來決定藏密量，但是在平滑區間裡的藏密量並不高，於是有了PVD-LSB來予以改進。但是LSB會被RS偵測檢測出來，於是本文採用matrix embedding來做結合。

3.1 PVD 與 matrix embedding結合

利用Hwang等學者所提出的lower-level與higher-

level的概念，將PVD與matrix embedding做結合。以下為結合的演算法：

步驟一：經由式子(1)計算出影像中相鄰像素P_i與P_{i+1}差值d。

步驟二：設定div為63，將圖一的R_1~R_4定義為lower-level，R_5、R_6定義為higher-level。

步驟三：

lower-level：進行步驟四。

higher-level：進行PVD資料隱藏。

步驟四：從機密訊息中取6個位元：m_1，m_2，m_3，m_4，m_5，m_6。將m_1 m_2 m_3 以matrix embedding藏入P_i後六個位元中得到P'_i；m_4 m_5 m_6 以matrix embedding藏入P_{i+1}後六個位元中得到P'_{i+1}。

例子：假設$P_i = 30$，$P_{i+1} = 15$，$s = (111000)_2$，P_i 後六個位元為$(011110)_2$，計算出$S = (111)_2 \oplus (011110)_2 \times H^T = (010)_2$，查表$S' = (010000)_2$，$S' \oplus (011110)_2 = (001110)_2$，替代$P_i$後六個位元，$P'_i = 14$，以此類推$P'_{i+1} = 15$。

以下為擷取訊息的步驟：

步驟一：利用式子(4)計算出影像中相鄰像素P'_i與P'_{i+1}差值d'。

步驟二：如果為higher-level，進行PVD擷取，若為lower-level，進行步驟三。

步驟三：分別對P'_i與P'_{i+1}做matrix embedding擷取。例子：假設$P'_i = 14$， 後6個位元為001110，乘上H^T得到111；$P'_{i+1} = 15$，後6個位元為001111，乘上H^T得到000。

3.2 Matrix embedding 改良

由於原本的方法有$\frac{3}{8}$的機率更動到前四個高位元，與原本的像素值差異過大，導致隱藏完影像的品質不佳，圖二為經過(6,3)漢明碼資料隱藏前後的比較。

圖二　漢明碼資料隱藏(左)與原圖(右)比較

圖二隱藏前後的差異用肉眼即可辨識，為了縮小與原本像素值的差異，以下提出改進的方法：

表三 (6,3) hamming code

徵狀值	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
000	0	28	42	49	54	45	27	7
100	32	60	10	17	22	13	59	39
010	16	12	58	33	38	61	11	23
001	8	20	34	57	62	37	19	15
011	4	24	46	53	50	41	31	3
101	2	30	40	51	52	47	25	5
110	1	29	43	48	55	44	26	6
111	36	56	14	21	18	9	63	35

表二為(6,3)漢明碼，第一行為每一列資料的徵狀值。由於最後擷取資料時是乘上H^T得到秘密資訊(徵狀值)，所以資料能夠替換成該列上任意一資料。以下為改進的演算法：

步驟一：從像素值P_i取後面6個位元C，從機密資料取出3個位元S。

步驟二：根據徵狀值S找出該列資料(d_1~d_8)。

步驟三：依照式子(7)計算出C'。

$$C' = \min(|d_i - C|) \quad 1 \le i \le 8, i \in N \qquad (7)$$

步驟四：將P_i後六個位元取代為C'得到P'_i。

例子：假設$P_i = (120)_{10} = (1111000)_2$，後6個位元為$(111000)_2 = (56)_{10}$，機密資料為011，在徵狀值那列與56差值最小的為$(53)_{10} = (110101)_2$，替換掉原本像素值的後6個位元，得到$P'_i = 117$。

擷取資料的方法不變，一樣取後面6個位元乘上H^T得到機密資訊。

4. 實驗分析與結果

4.1 新舊像素差值的worst case

在改良的matrix embedding中，相同的像素值會因為嵌入不同的機密訊息而找到相對應差距最小的值。因此我們固定了掩護序列值，將所有可能隱藏的徵狀值都藏過一遍，找出隱藏前後掩護序列差值最大者，記錄此差值。以下是實驗出來的數據：

表四 worst case

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

原本的matrix embedding，雖然隱藏前後只會改變一到兩個位元，但是卻有$\frac{3}{8}$的機率更動到前四個高位元。由表四可以看出，改良後的matrix embedding在最糟的情況下，差值介於4~11之間，最糟只動到第四個位元。以影像處理上經常使用的peak signal-to-noise ratio(PSNR)值來估算兩演算法的影像品質，下表為在同樣藏密量中， matrix embedding改良前後的PSNR值比較：

表五 掩護影像為786K的改良前後比較

藏密量	(6, 3)漢明碼隱藏改良前 PSNR	(6, 3)漢明碼隱藏改良後 PSNR
296K	18.9768db	30.8069db
270K	19.5155db	31.2288db
120K	22.5079db	35.1205db

由表五數據可得知，在相同的藏密量下，本篇所提的改良方法大幅提高了影像的品質，PSNR值與原本的方法比較大約增加了1.6倍。

4.2 實驗結果

圖三~六分別為原圖、PVD隱藏、PVD-LSB隱藏，PVD-matrix embedding隱藏的比較圖，這四張圖以肉眼分辨不出差別。

圖三 Lena原圖

圖四 Lena 經過PVD資料隱藏153235byte

圖五 Lena經過PVD-LSB隱藏286484byte

圖六 Lena經過PVD-matrix embedding隱藏 295017byte

表六中的影像皆為512×512之彩色影像，其中影像A為Lena圖。由表六數據可以得知：本文所提出的PVD與改良matrix embedding的結合藏密量是最高的，而PSNR僅有微幅下降，在肉眼辨視範圍內是分辨不出來的。PVD matrix embedding的藏密量是PVD的1.8~1.98倍，而PSNR值與PVD相較僅下降3.42db~10.26db；與PVD-LSB比較的話藏密量約增加0.9%~4.8%，而PSNR值下降了0.48db~4.82db。

表六　演算法在彩色影像上的比較

影像	PVD		PVD and LSB		PVD and matrix embedding	
	藏密量 (byte)	PSNR (db)	藏密量 (byte)	PSNR (db)	藏密量 (byte)	PSNR (db)
A	153235	39.10	286464	32.82	295017	30.81
B	149303	39.10	292314	33.66	295019	28.84
C	163702	32.95	281478	29.21	295172	28.26
D	155505	33.36	291295	30.42	295358	29.94
E	148984	33.66	263579	30.50	273620	29.62

5.結論與未來展望

PVD matrix embedding與PVD比較的話，提升了將近2倍的藏密量，PSNR值雖然略低，但是肉眼分辨不出來。PVD matrix embedding與PVD LSB相比的話，由於PVD LSB是將像素差值較小的區間利用LSB來提升，一張影像中相鄰像素差值越小出現的次數會越高，故PVD-LSB容易被RS檢測分析出來，相較於PVD matrix embedding是利用matrix embedding提升藏密量，動到的位元數不一定會在最後面，較不易被檢測出來。

在matrix embedding中，利用了(6,3)漢明碼來進行資料隱藏，未來如果有掩護序列在六個位元以下，但一樣能夠藏入三個位元的矩陣出現，便能夠再度提高藏密量以及影像的品質。

6. 參考文獻

[1] W. Bender, D. Gruhl, N. Morimoto, and Aiguo Lu, “Techniques for data hiding,” IBM Systems Journal,Vol.35, No. 3-4, pp. 313-336, 1996.

[2] J. Fridrich, M. Goljan, and R. Du, “Detecting LSB Steganography in Color and Gray-scale Images,” IEEE Multimedia, Vol.8, No.4, pp. 22-28, Oct-Dec 2001.D. C

[3] Wu, and W. H. Tsai, “A steganographic method for images by pixel-value differencing,” Pattern Recognition Letters, Vol. 24, No. 9-10, pp. 1613-1626, 2003.

[4] H. C. Wu, N. I. Wu, C.S. Tsai, and M.S. Hwang, “Image Steganographic Scheme Based on Pixel-Value Differencing and LSB Replacement Methods,” IEE Proceedings-Vision, Image and Signal Processing, vol. 152, no. 5, October 2005, pp.611-615.

[5] W. Zhang, X. Zhang, and S. Wang, “Maximizing steganographicembedding efficiency by combining Hamming codes and wet paper codes,” in Proc. 10th Int.Workshop Inf. Hiding, , K. Solanki, K.Sullivan, and U. Madhow, Eds., Santa Barbara, CA, Jun. 19–21, 2008,vol. 5284, Lecture Notes in Computer Science, pp. 60–71.

[6] Shu Lin and Daniel J. Costello,“Error Control Coding – Fundamentals and Applications,” Second Edition, 2004.