Proof of Theorem (ii): Note that motion of a test point T from the central point C along the line from this point to one of the outer points, say P, does not affect the sum of the absolute distances to these two points (i.e. CT + PT remains constant). However, it does increase the distances to the other two points, Q, R (i.e. QT + RT is increased). Similar arguments apply for motions towards either of the other two points, or for motions in intermediate directions. This proves that C is already at the L_1 minimum.

Two comments on this result are in order:

1. If we move a fourth point in a circular orbit around a region containing three points, we go from a Case (i) situation to a Case (ii) situation, then back again, and so on for a total of six changes, and it is straightforward to show that the L_1 minimum moves once around the triangle formed from the stationary three points, pausing for a while at each of these points. If the circle is at infinity, it is easily shown that the L_1 minimum spends equal times along the sides of the triangle and at the three points. Thus there is in general approximately equal probability of the L_1 minimum being situated on a sample as lying in the space between the three points.

2. If one of the four points moves slightly out of the plane of the other three, the situation will not change substantially, and Cases (i) and (ii) are retained. However, this situation will not be retained as the fourth point moves an indefinitely large distance from the plane of the other three points. In fact, if it moves to a position a large distance away, the L_1 minimum will be forced into a modified Case (i) situation, and will lie in the space between points: space precludes providing a proof of this result here.

The theorems presented here, and the ensuing discussion, have shown that there is a significant probability in the case of four points in both 2D and 3D that the L_1 minimum will not be situated at any of the sample points. There would appear to be no reason why the case of four sampling points is particularly special; in any case, we can always find instances when larger numbers of points give L_1 minimum positions well away from all the sampling points (we merely select a symmetrical arrangement of points for which there is no point at the centre of symmetry, and the latter will always be the position of the L_1 minimum). In general, this means that a significant error is quite likely to arise if the L_1 minimum is constrained to just those positions occupied by sampling points.

Concluding remarks: A general view is that the L_1 minimum must lie within, or on, an innermost triangle (in 2D), tetrahedron (in 3D), or polyhedron (in nD). In 1D there will be an upper bound on the error of the L_1 minimum of at most half the distance between sampling points, and a minimum bound of zero. In the case of four points in 2D, a better estimate of the upper bound is $1/\sqrt{3}$ of the distance between sampling points, and in the case of four points in 3D, a better estimate of the upper bound is $\sqrt{(3/8)}$ of this distance (proofs of these results are based on the radii of the circumscribing circles for equilateral triangles and circumscribing spheres for regular tetrahedra). Proofs of these statements and a fuller discussion of the problem will be presented at a later date. Meanwhile, the basic aim of this Letter has been achieved – of showing that restricting the output multichannel median to one of the input vector samples is likely to introduce significant error.

13 September 2000
Electronics Letters Online No: 20001465
DOI: 10.1049/el:20001465

E.R. Davies (*Machine Vision Group, Department of Physics, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, United Kingdom*)

E-mail: E.R.Davis@rhbnc.ac.uk

References

1 DAVIES, E.R.: 'Machine vision: theory, algorithms, practicalities' (Academic Press 1997), 2nd edn.

2 ASTOLA, J., HAAVISTO, P., and NEUVO, Y.: 'Vector median filters', *Proc. IEEE*, 1990, **78**, (4), pp. 678–689

3 REGAZZONI, C.S., and TESCHIONI, A.: 'A new approach to vector median filtering based on space filling curves', *IEEE Trans. Image Process.*, 1997, **6**, (7), pp. 1025–1037

Hiding data in images by optimal moderately-significant-bit replacement

Ran-Zan Wang, Chi-Fang Lin and Ja-Chen Lin

A data hiding technique for the storage and transmission of important data is proposed. It embeds the important data in the moderately-significant-bit of an image, and applies a global substitution step and a local pixel adjustment process to reduce any image degradation. Experimental results show that the visual quality of the resulting image is acceptable.

Introduction: Hiding important data such as military information or personal financial documents in images has been a popular research topic [1]. The goal is to make the embedded data invisible to the grabbers under the cover of the host image, i.e. to make the host image, after processing, as similar as possible to the original host image.

A very simple and direct method is to hide the important data in some bits of each pixel of the host image. Least-significant-bit (LSB) substitution is a common way to do this. Many articles [2, 3] have addressed approaches related to LSB substitution. Although embedding data in LSB introduces small distortion to the host image, the embedded data is more easily lost when the resulting image is used at a later date. (For example, in order to save space, increase the transmission rate, or accelerate the image processing speed, some software discard the LSB (and make the image become 7 bits per pixel instead of 8 bits). Storing data in the LSB is therefore less safe.) In this Letter, we describe data being embedded in the moderately-significant-bit (MSB) of the host image, an approach seldom seen in the literature.

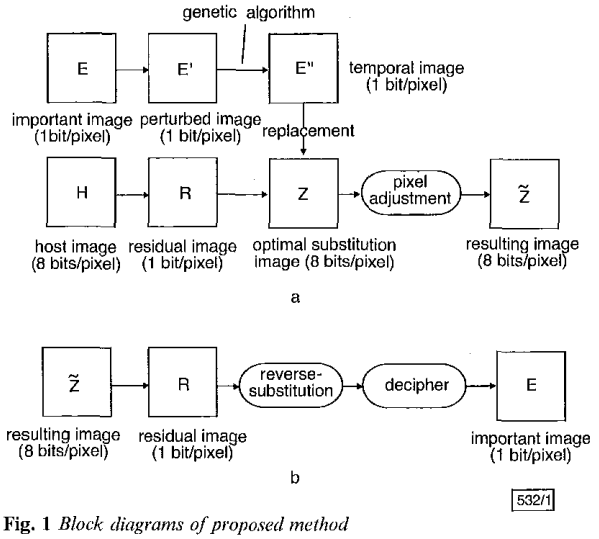

Fig. 1 *Block diagrams of proposed method*

a Embedding stage
b Extraction stage

Proposed data hiding scheme: A method to embed important data in the MSB of an image is proposed. The system consists of two stages: the embedding stage that embeds the important data in the host image, and the extraction stage that retrieves the important data from an image. Block diagrams of the method are shown in Fig. 1.

Let E be the important data of $M \times N$ bits. We may treat E as a 'binary' image (1 bit per pixel) of $M \times N$ pixels. To embed E in a grey-scale (8 bits per pixel) host image H containing $M \times N$ pixels, the following three steps are conducted sequentially.

Step 1: The cipher process. A polynomial transformation [4] is applied to cipher the important image E to obtain a perturbed image E' as follows. Assume that the pixels in E are numbered sequentially from 1 to $M \times N$. Pixel at location x of E is transposed to a new location $f(x)$ by the following equation:

$$f(x) = (k_0 + k_1 \times x + k_2 \times x^2 + \cdots + k_r \times x^r) \bmod (M \times N)$$

where $\gcd(k_i, M \times N) = 1 \quad 0 < i \leq r$ and

$$\gcd(k_i, k_j) = 1 \text{ for } i \neq j \quad (1)$$

In this equation, k_i are the keys, and gcd(·) means the greatest common divisor.

Step 2: Optimal substitution process. The MSB (the fifth bit) of all pixels of the host image are extracted to form a (binary) residual image R. Both the perturbed image E' and the residual image R are divided into non-overlapping blocks of size 4×1. Note that all four bit blocks must be one of the following 16 kinds I = {0000, 0001, ..., 1111}. Let P be a permutation operator on I, i.e. P: $I \rightarrow I$ be a one-to-one mapping from and onto I. According to P, each 4×1 block of E' is transformed to a new 4×1 block, E' is thus transformed to a new temporal image E''. The optimal substitution image Z can be obtained by replacing R (the fifth bit plane of H) by E''.

To define the permutation operator P in the above, there are 16! possible choices. A genetic algorithm (GA) [5] is developed (below) to search for a good P from the 16! possible choices. In our GA, each P is expressed as a chromosome G containing 16 genes as follows:

$$G = g_0 g_1 \cdots g_{15} = g_{0000} g_{0001} \cdots g_{1111} \quad (2)$$

where g_{0000} means that, when P is applied, the block 0000 is replaced by a four bit block g_{0000}. The operators in the GA are designed as follows:

[Crossover] Given two chromosomes $G_1 = p_0 p_1 ... p_7 ... p_{15}$ and $G_2 = q_0 q_1 ... q_7 ... q_{15}$, the offspring are $G_1' = p_0 p_1 ... p_7 q_8 ... q_{15}$ and $G_2' = q_0 q_1 ... q_7 p_8 ... p_{15}$. The chromosomes G_1' and G_2' obtained as above may be invalid, i.e. some genes might occur more than once (e.g. $p_5 = q_9$), and a validation process is thus designed to correct them.

[Mutation] Given a chromosome $G = g_0 g_1 ... g_{15}$, two of the 16 genes of G are selected randomly and replaced by each other.

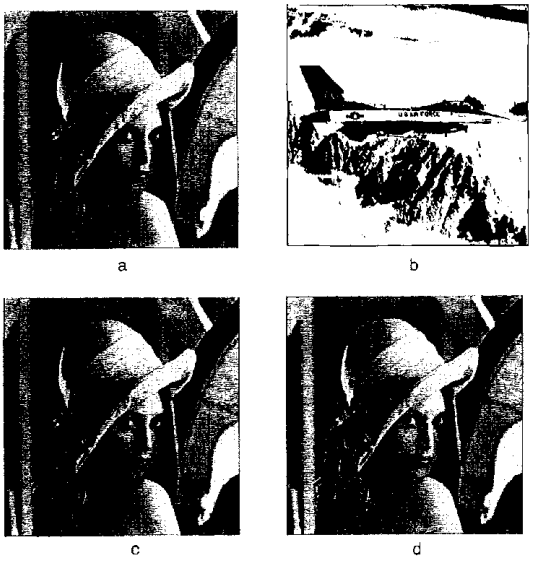

Fig. 2 *Illustration of experimental results*

a Host image 'Lena'
b 'Binary' important image 'Jet'
c Resulting image from simple MSB substitution method
d Resulting image Z (after steps 1 – 3) of proposed method

Step 3: Pixel adjustment process. To improve the image quality, we add a post-processing which will not touch the fifth bit of each pixel of the image Z (hence will not hurt the important data), but will slightly modify the other bits of each pixel of Z to improve visual quality. Let p and p' be the corresponding (8 bit) grey values of a pixel of H and Z, respectively, and δ be the value of the last three bits (bits 6 – 8) in p'. If $p \neq p'$, then either (i) $p' = p - 8$ or (ii) $p' = p + 8$ (because the only difference between H and Z is the fifth bit plane).

Case 1: when $p' = p - 8$. If $\delta \geq 4$, then the value $8 - \delta - 1$ is added to p'. If $\delta < 4$ and if the fourth bit of p' is 0, then the fourth bit of p' is changed to 1, and the value δ is subtracted from p'. Do nothing otherwise.

Case 2: when $p' = p + 8$. If $\delta < 4$, then the value δ is subtracted from p'. If $\delta \geq 4$ and if the fourth bit of p' is 1, then the fourth bit of p' is changed to 0, and the value $8 - \delta - 1$ is added to p'. Do nothing otherwise.

To extract the important data from an image, the data in the bit plane where the important data were embedded are extracted, and a reverse-substitution step and a decipher process are conducted to reveal the important data.

Experimental results: The 256 grey-value host image 'Lena' and the binary important image 'Jet' tested in this experiment are shown in Figs. 2*a* and *b*, respectively. The important image is embedded in the fifth bit plane of the host image. Figs. 2*c* and *d* are the resulting images of the simple replacement method (i.e. replace directly) and the proposed method, respectively. The PSNR of the two images in Figs. 2*c* and *d* are 33.02 and 38.75dB, respectively. It can be seen that the image in Fig. 2*d* is much better.

Conclusion: Moderately-significant-bit replacement is seldom used in data hiding, since there will be degradation of image quality (see Fig. 2*c*). However, we have shown that with careful design, such as the use of optimal substitution process and local pixel adjustment, MSB can still be used (see Fig. 2*d*) as an alternative choice for the storage and transmission of important data.

28 September 2000
Electronics Letters Online No: 20001429
DOI: 10.1049/el:20001429

Ran-Zan Wang and Ja-Chen Lin (*Department of Computer and Information Science, National Chiao Tung University, Hsinchu, 300, Taiwan, Republic of China*)

E-mail: rzwang@cis.nctu.edu.tw

Chi-Fang Lin (*Department of Computer Engineering and Science, Yuan-Ze University, Taoyuan, 320, Taiwan, Republic of China*)

References

1 PETITCOLAS, F.A.P., ANDERSON, R.J., and KUHN, M.G.: 'Information hiding – a survey', *Proc. IEEE*, 1999, **87**, (7), pp. 1062–1078
2 VAN SCHYNDEL, R.G., TIRKEL, A.Z., and OSBORNE, C.F.: 'A digital watermark'. Int. Conf. Image Processing, 1994, pp. 86–90
3 CHEN, T.S., CHANG, C.C., and HWANG, M.S.: 'A virtual image cryptosystem based upon vector quantization', *IEEE Trans. Image Process.*, 1998, **7**, (10), pp. 1485–1488
4 RHEE, M.Y.: 'Cryptography and secure communications' (McGraw-Hill, Singapore, 1994)
5 HOLLAND, J.H.: 'Adaptation in natural and artificial systems' (University of Michigan Press, Ann Arbor, MI, 1975)

Improved lossless compression of general data

S. Keating and J. Pelly

New algorithms for lossless compression of general data are presented. They are based on adaptive lossless data compression (ALDC) but offer improved compression, typically 24% better for image data. The algorithms are simple to implement and are capable of high data throughput, whilst maintaining compatibility with legacy ALDC bit streams.

Introduction: Lossless data compression is used extensively in data storage and communication. Popular algorithms are frequently based on the work of Lempel and Ziv [1]; one of these is adaptive lossless data compression (ALDC) [2]. These algorithms give good compression of many types of data. However, better compression can be achieved for image data using algorithms that are more specific; these generally include some form of modelling of the image data such as differential pulse code modulation (DPCM). This Letter describes algorithms for compressing general data, which use a single architecture for all data types. The algorithms have been named SZIP and DZIP. The architecture is based on