Proc. of the 6th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision, Elounda, Greece, August 21-23, 2006 (pp89-94)

(2, 3)-threshold visual cryptography for color images

Kun-Yuan Chao* and Ja-Chen Lin
Department of Computer and Information Science,
National Chiao Tung University,
1001 Ta Hsueh Rd., Hsinchu, Taiwan, 300,
R.O.C.
*Corresponding author. kychao@cis.nctu.edu.tw

Abstract: - Most of the studies in visual cryptography (VC) are for binary images. Hou's method [Pattern Recog. 2003, Vol. 36, 1619-1629] is a good extension from binary to colors. In this study, we extend Hou's method further so that the new scheme has fault-tolerant ability, namely, the secret image can still be revealed if one of the generated shares delays due to communication channel failure, or by natural crash of the storage equipment, or even destroyed by hackers. In the proposed scheme, a color secret image decomposes into three shares. Each share alone cannot reveal the secret image. However, gathering "any" two of the three shares can unveil the secret image.

Key-Words: - Visual cryptography; C-M-Y color decomposition; Error diffusion; Shares; Stacking

1 Introduction

Secret sharing [1] is one of the approaches to keep a secret safe. When the secret is an image, a well-known approach using sharing is the visual cryptography (VC) approach [2]. Many extended studies of [2] have been proposed; however, most of these studies (for example, [3]) are for binary images due to the nature of VC. Some studies extend VC to the encryption of gray-level images, for instance, Ref. [4, 5]. Recently, based on black-and-white VC and the color decomposition method, Hou [6] elegantly proposed three VC methods for color images. Hou's methods to process color images have wide applications because color images are everywhere nowadays. In this study, we extend Hou's work further so that the new scheme has fault-tolerant ability. Among Hou's three methods, his Method 3 is the most elegant one. This study thus focuses on his Method 3.

A (k, n)-threshold visual cryptography scheme, as described by Naor and Shamir [2], is a method to encode a secret image into n shadow images (called shares). Any k of the n shares can be stacked to recover the secret image, but any k-1 or fewer of them gains no information about the secret image. In the current study, a new scheme is proposed to extend Hou's Method 3 to a (2, 3)-threshold visual cryptography scheme for color images so that the new version has fault-tolerant ability.

The remaining portion of the paper is organized as follows. Section 2 briefly reviews binary (Black/White) visual cryptography, and the methods to deal with gray-level and color images using halftone techniques. The C-M-Y color decomposition used by Hou is also reviewed in this section. Section 3 presents the proposed (2, 3)-threshold method for color images. Experimental results and comparisons are in Section 4. Finally, conclusions are given in Section 5.

2 Reviewing the methods of visual cryptography

2.1 Binary VC using 2×2 black-and-white blocks

In the simplest design of black-and-white visual cryptography, a pixel (black or white) in a binary image is often decomposed into two size-extended (2×2 in size) sharing blocks according to the rules in Fig. 1. The four elements in each 2×2 sharing block are 50% black and 50% white, i.e. there are two white elements and two black elements in each block, regardless of whether the input pixel is a black pixel or a white one.

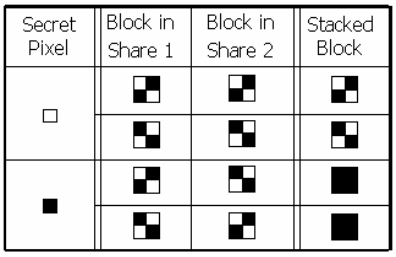
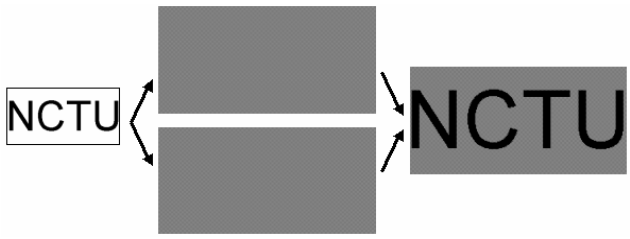


Fig. 1 A sharing and stacking scheme for binary (black/white) images.

After the decomposition of the given binary secret image, the two obtained shares look like random noises, and each share is 2×2=4 times bigger than the secret image due to the size-extended decomposition (from a pixel to a block of size 2×2). Each share alone cannot provide any information about the secret image. However, after stacking the two shares, the binary secret image can be recovered, but the contrast degrades by 50%. (Compare Fig. 2 (c) with 2 (a).) The reason is obvious: according to the last column of Fig. 1, it can be seen that if the original input pixel is a white pixel, then after stacking, the corresponding 2-by-2 stacked block has 2 black elements and 2 white elements (instead of four white elements). Therefore, the white area of the binary input image will not be mapped to a white area in the stacked image, instead, it is mapped to an area whose brightness is more like gray (as shown in Fig. 2 (c)).

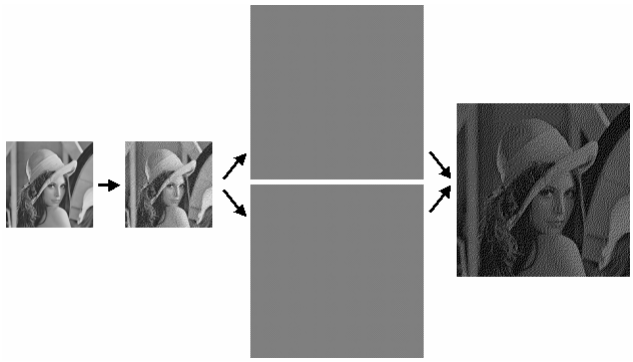


(a) (b) (c)

Fig. 2 An example of binary VC technology. (a) the input (a binary secret image), (b) the output (Shares 1 and 2), and (c) the result after stacking.

2.2 Gray-leveled VC with the help of halftone technology

Halftone technology is a method to use the density of the net dots to simulate the gray level. After the halftone transformation, the gray-level image becomes a binary one, which almost looks like a continuous-tone image. In our paper, an error-diffusion method designed earlier in our lab [7] is used as the halftone method to transform a gray-level secret image (or one of the three color-components of a color secret image [each component has 256 levels]) to a binary image. The traditional black-and-white VC method in Section 2.1 can then be applied to generate the shares. Fig. 3 shows an example. This kind of pre-processing (using halftone techniques to transform gray-level images to binary images) can also be found in other reported VC-designs ([5] and [6]).



(a) (b) (c) (d)

Fig. 3 An example of gray-level VC with the help of halftone technique. (a) the input (a gray-level secret image), (b) the transformed halftone image, (c) the output (Share 1 and Share 2), and (d) the stacked result.

2.3 Hou's Color VC with the help of color decomposition method

Only a few studies present VC schemes for color images. Hou's Method 3 belongs to this special category. Hou transformed a color secret image into three halftone images C, M, and Y by using a halftone algorithm three times (the first time is on the C-component, then on the M-component, then on the Y-component). Each 24-bit color pixel of the input secret image then became a 3-bit pixel (C, M, Y) where C was either 0 or 1 (so were M and Y). Hou then applied a technique of binary visual cryptography (similar to the one in Sec. 2.1) to each of the three components, and thus generated six temporary component-shares C1, C2, M1, M2, Y1, and Y2. Hou then combined C1, M1, and Y1 to form a colored halftone Share 1 (and combined C2, M2, and Y2 to form Share 2). After stacking Share 1 and Share 2, the content of the stacked image could then be easily identified. Fig. 4 shows how to decompose a 3-bit blue pixel (C=1, M=1, Y=0) into two sharing blocks and how to reconstruct the blue-like block. This version of Hou's generated two shares, and did not sacrifice the contrast too much.

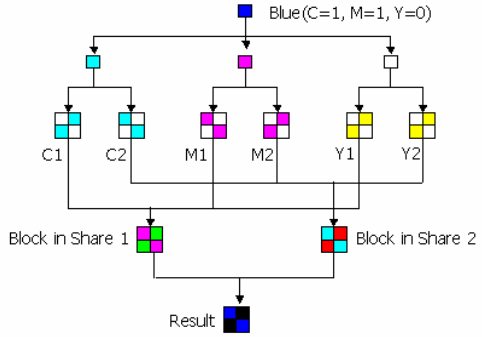


Fig. 4 Hou's color pixel decomposition and reconstruction.

3 The proposed (2, 3)-threshold visual cryptography for color images

In Sec. 3.1, it describes how to share a pixel. Then it will describe in Sec. 3.2 how to share an image.

3.1 The decomposition of a pseudo-color pixel (C, M, Y) and its reconstruction using OR operation

3.1.1 Sharing a 3-bit vector (C,M,Y) to obtain three new vectors {(Ci,Mi,Yi): i=1~3}

Assume that each 24-bit color is composed of cyan, magenta and yellow; and each 24-bit color pixel has been reduced to 3-bit pseudo color by applying a halftone method three times (the first time is on the C-component, then on the M-component, then on the Y-component, as stated in Sec. 2.3). Therefore, the pseudo-color of a pixel can be denoted by (C, M, Y) where the coordinate value of C is either 0 or 1, so do the coordinate values of M and Y. For example, the pseudo-color blue is (C=1, M=1, Y=0).

Each pseudo-color (C, M, Y) can be shared to produce three new pseudo-colors (C, M, 0), (C, 0, Y) and (0, M, Y). Any two or three of them can recover the original pseudo-color (C, M, Y) by the OR operation because

(C, M, 0) + (C, 0, Y) = (C, M, Y), (eq.1)
(C, 0, Y) + (0, M, Y) = (C, M, Y), (eq.2)
(C, M, 0) + (0, M, Y) = (C, M, Y), (eq.3)
(C, M, 0) + (C, 0, Y) + (0, M, Y) = (C, M, Y). (eq.4)

As an example, let us inspect the blue pixel whose pseudo-color is (C=1, M=1, Y=0). The three shares of (C=1, M=1, Y=0) are (1, 1, 0), (1, 0, 0) and (0, 1, 0). Note that (via eq.1-4)

(1, 1, 0) + (1, 0, 0) = (1, 1, 0),
(1, 0, 0) + (0, 1, 0) = (1, 1, 0),
(1, 1, 0) + (0, 1, 0) = (1, 1, 0),
(1, 1, 0) + (1, 0, 0) + (0, 1, 0) = (1, 1, 0).

In other words, any two shares (or all three shares together) can recover the original (1,1,0). In the above, the operation "+" denotes "OR"; for example, 1 + 0 = 1 + 1 =1 + 1 + 1 = 1.

3.1.2 Painting the 2-by-2 block for each share

The above discussion in Sec. 3.1.1 only told us how to share a 3-bit vector (C, M, Y) to create three other 3-bit vectors {(C, M, 0), (C, 0, Y), (0, M, Y)} with the help of the overwriting operation using "0". The discussion mentioned nothing about how to paint the image block for each share. This not-yet-touched question will be answered using Fig. 5. In Fig. 5, it shows how to decompose a 3-bit pseudo-color blue pixel (C=1, M=1, Y=0) into three 2-by-2 pseudo-color blocks, and then how to reconstruct a blue-like 2-by-2 block back from any two of these three generated blocks.

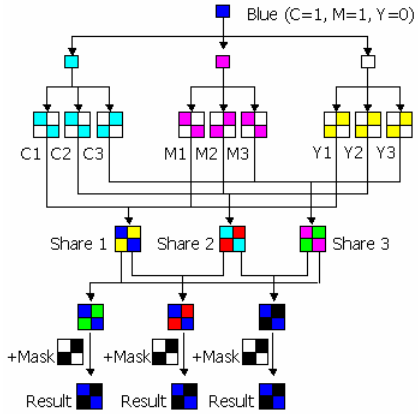


Fig. 5 The color pixel decomposition (to generate three shares) and reconstruction (from "any" two of the three shares) used in our method.

As discussed in Sec. 3.1.1 above, the three shares of (C=1, M=1, Y=0) are as follows: Share 1 is (C1=1, M1=1, Y1=0), Share 2 is (C2=1, M2=0, Y2=0), and Share 3 is (C3=0, M3=1, Y3=0). Now, as shown in Fig. 5, for the nine color components {C1~C3, M1~M3, Y1~Y3}, each (1-bit) color component is represented by a 2-by-2 block, and paint the block by this rule: if the value of the 1-bit color component is 1 (for example, C1 is 1) , then the northeast and southwest elements of the block are blank; if it is 0 (for example, C3 is 0), then the northwest and southeast elements are blank. (As for the other two elements of the 2-by-2 block, one needs to paint the color of the color component being discussed. For example, if C1 is the color component being discussed, then paint cyan in these two non-blank elements because "C" stands for "Cyan"). Finally, combining the three 2-by-2 component blocks (each block has 4 elements, and each element is 1-bit) that represent C1, M1, and Y1, respectively, the 2-by-2 pseudo-color block is obtained and called as Share 1. Analogue statements yield Share 2 and Share 3. (Now, although each 2-by-2 block in a share still has 4 elements, each element has 3 bits because each element becomes a 3-dimensional vector (C, M,Y).) Note that the northeast element of Share 3 is green because the superposition of the cyan color and yellow color yields the green color (see the northeast elements of the 2-by-2 component blocks of C3, Y3, and M3). Without the loss of generality in VC, the three 2-by-2 pseudo-color blocks must

be randomly re-permutated into shares for security. The sharing procedure of a secret pixel ends here.

In later days, when a receiver wants to reconstruct the secret pixel (the blue pixel in the above example), he just stacks together "any" two of the three shares, then continue the stacking with a 2-by-2 mask block which is black in the northeast and southwest elements (and blank in the northwest and southeast elements). In the bottom half of Fig. 5, this process is shown to indicate how to reconstruct a blue-like block which represents the blue pixel shown at the top of Fig. 5 (if the two black elements in the 45 degree direction [the northeast and southwest] of the resulting 2-by-2 block always be ignored). The receiver can also see that in Fig. 5, without the help of the mask block, stacking Shares 1 and 2 will be different from stacking Shares 1 and 3 (or from stacking Shares 2 and 3), although in each case the -45 degree direction (the northwest and southeast elements) is always the expected blue color.

3.1.3 Diversity of the design (using "rotation" or "switch" locally to get better view globally)

If the technique described in Fig. 5 repeatedly is used to expand each 3-bit pixel of a 256-by-256 halftoned pseudo-color image, although three 512-by-512 shares can be obtained, the reconstruction results from stacking the shares will be a little strange, because there exists a great number of black lines in the 45 degree direction (southwest to northeast). These black lines are caused by the repeated appearance of the same pair of black elements shown in each resulting block of Fig. 5. In fact, the same pair of black elements appears 256×256 times when the input secret image is 256×256 in size; and these black elements are always in the 45 degree (southwest to northeast) direction.

To avoid this phenomenon, only one thing has to be avoided is that the black pair always appear in the same position. To achieve this, just notice that if "all" 2-by-2 blocks are rotated (including the 9 component blocks, the 3 shares, the mask, and the resulting blocks) in Fig. 5 by 90 degrees, the system still works. That is, there are still three 2-by-2 shares (but rotated), and by stacking the rotated mask with any two of the three rotated shares, the rotated result still has two blue elements (and two dummy black elements).

In fact, there are $C(4,2)=4\times3/2! = 6$ possible ways to assign the two dummy black elements in a 2-by-2 mask. For example, (NE, SW) is the one shown in Fig. 5; (NW, SE) is the one introduced above for the rotated version; and (NE, NW) is another kind, etc. Without the loss of generality, this sub-section only introduces how to modify the design when the mask has the two dummy black elements in (NE, NW). Compare the (NE, NW) mask with the old mask shown in Fig. 5 (the (NE, SW) mask), it may say the new mask is obtained by switching the SW (Southwest) and NW (Northwest) elements in the mask of Fig. 5. Therefore, it only has to switch the SW and NW elements for "all" 2-by-2 blocks shown in Fig. 5. The new design is then complete. After all, the stacking of the blocks is actually done by doing the stacking in an element-by-element manner. That is, stack the NE element first, then stack the NW, then stack the SW, then stack the SE. Therefore, if the 4 elements of all 2-by-2 blocks in Fig. 5 (before the generation of the shares) are permuted using a specified permutation rule, then the 4 elements of the result will also be permuted by the same permutation rule.

3.2 The (2, 3)-threshold visual cryptography for a color image

By the ED halftone algorithm (Han and Lin, 1997), a 256-by-256 full-color (24-bit) secret image can be transformed into three 256-by-256 (1-bit) halftone images C, M, and Y; or equivalently, a 256-by-256 pseudo-color (3-bit) C-M-Y image. Then the above sharing method may be employed for each pixel. Thus three 512-by-512 3-bit pseudo-color images (C1, M1, Y1), (C2, M2, Y2), and (C3, Y3, M3) will be generated. These three pseudo-color images are the three expected shares.

Algorithm to share a secret image

1. Transform the 24-bit color image into a 3-bit C-M-Y pseudo-color halftone image.
2. For each pixel Pij of the halftone image, whose pseudo-color components are (cij, mij, yij), do the following:

(a) Create a 2-by-2 mask block Bij, in which two of the four elements are randomly assigned to be black, then leave the other two elements being white (blank).

(b) According to the positions that the two black elements appear in the mask block Bij, use a way analogous to the one used in Fig. 5 to expand (cij, mij, yij) into nine 2-by-2 blocks, C1ij, C2ij, C3ij, M1ij, M2ij, M3ij, Y1ij, Y2ij, and Y3ij.

(c) Combine blocks C1ij, M1ij, and Y1ij to paint the combined block (i,j) for Share 1.

(d) Combine blocks C2ij, M2ij, and Y2ij to paint the combined block (i,j) for Share 2.

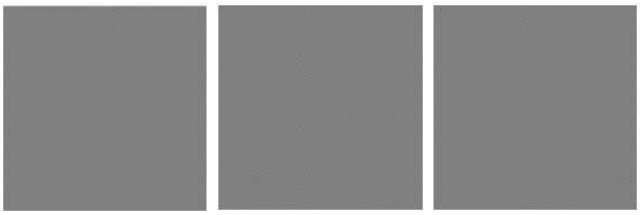
(e) Combine blocks C3ij, M3ij, and Y3ij to paint the combined block (i,j) for Share 3.

(f) Re-permute the three combined blocks randomly into Share 1, 2, and 3.

3. After finishing Step 2 (when all pixels Pij have been processed), the three desired shares can be obtained by collecting all the combined blocks (of the corresponding share). A mask image is also obtained by collecting all the mask blocks.
4. In later days, after stacking the mask image with any two of the three shares, the pseudo-color version of the secret image is unveiled.

4 Experimental results

A secret image Lena is decomposed into the three shares shown in Fig. 6. None of the shares alone reveal any information about the Lena image. After stacking Shares 1 and 2, the stacked image is the one shown in Fig. 7 (a). Then, after stacking Fig. 7 (a) with the mask shown in Fig. 7 (b), the result is obtained in Fig. 7 (c). Analogously, Fig. 8 and Fig. 9 show other stacked results when the action of stacking Shares 1 and 2 in Fig. 7 are replaced by stacking Shares 1 and 3 (Fig. 8) or by stacking shares 2 and 3 (Fig.9).



(a) (b) (c)

Fig. 6 The three shares of Lena. (a) Share 1, (b) Share 2, and (c) Share 3.

(a) (b) (c)

Fig. 7 Stacking Shares 1 and 2. (a) is the intermediate image after stacking Shares 1 and 2, (b) is the mask, and (c) is the final image after stacking (a) with (b).

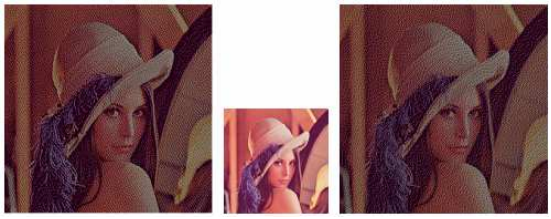
(a) (b)

Fig. 8 Stacking Shares 1 and 3. (a) is the intermediate image after stacking Shares 1 and 3; and (b) is the final image after stacking (a) with the mask in Fig. 7(b).

(a) (b)

Fig. 9 Stacking Shares 2 and 3. (a) is the intermediate image after stacking Shares 2 and 3; and (b) is the final image after stacking (a) with the mask in Fig. 7(b).

We also compare in Fig. 10 the resulting images of Hou's with ours (Fig. 7(c), which is identical to Fig. 8(b) and 9(b)), it can be seen that the two methods yield images of very similar quality. In fact, just like Hou's method, our resulting image also has a brightness loss, which is a common phenomenon seen in all reported VC methods.



(a) (b) (c)

Fig. 10 Comparison between the proposed scheme and Hou's Method 3. (a) is the resulting image in Fig. 7(c), (b) is the input 24-bit color image Lena, and (c) is the resulting image of Hou's.

5 Conclusion and discussion

Most of the studies about visual cryptography deal with binary images. The visual cryptography proposed by Hou to process color images is thus valuable because color images are everywhere nowadays. In this study, we extend Hou's method further so that the new scheme has fault-tolerant ability, namely, the secret image still has a chance to be unveiled if one of the generated shares is delayed in a communication channel, destroyed by hackers, or crashed in natural life of the storage equipment. In the proposed scheme, a color secret image is decomposed into three shares. Each share alone cannot unveil the secret image, while gathering any two of the three shares can recover the secret image.

The storage of the mask image is briefly discussed below. The mask image is mandatory;

otherwise, the stacked result will be shares-dependent, i.e., the result of stacking Shares 1 and 2 is different from that of stacking Shares 1 and 3 (see Fig. 7 (a) and 8 (a).) To store the mask image, there are at least three ways.

Assume that the secret image is 256-by-256 in size. Then the first way is to store the 512-by-512 mask image directly, and the storage space is 512×512 bits. The second way is as follows. Since there are only C(4,2)=4×3/2=6 kinds of pattern blocks (see Sec. 3.1.3) for the mask blocks, this way only has to record the indices (chosen from 1~6) indicating which kind is the first 2-by-2 block, …, which kind is the last (the 256×256-th) 2-by-2 block. Therefore, the storage space for the mask image is at most 3×256×256 bits.

As for the third way, it needs almost no storage space. Because a seed (a secret number) can be used in a random-number-generator to generate a sequence of 256×256 numbers, and each number is in the range 1~6. This way creates this sequence even before creating the three shares and the mask image. Then, when a secret image wants to be shared, in Step 2 (a) of the sharing algorithm (see Sec. 3.2), the mask block pattern is chosen according to this sequence of 256×256 numbers. For example, if the generated sequence is 3524516236415…, then the first mask block (B11) that will be used in Step 2(a) is of type 3, and the second mask block (B12) that will be used in Step 2(a) is of type 5, etc. Obviously, this way only has to store the seed instead of the whole sequence, because when a receiver wants to unveil the secret image later, he only has to use that seed to generate again the very same sequence 3264516236415…, and then expand this sequence to get the mask image. Notably, storing the seed saves the storage space at the expense of non-real time recovery in the decoding phase to get the secret image.

As a final remark, if the mask image is lost, the receiver can still get exactly the recovery image shown in Fig 7(c) by stacking all three shares (see Fig. 11). Of course, if he not only loses the mask image but also loses one of the three shares, then he can only recover the one shown in 7(a) or 8(a) or 9(a).

As for the future work, this study is a basic study for the (k, n)-threshold visual cryptography for color images. In the future, more methods can be developed by visual cryptograph researchers to create n shares for a color image, and any k of these n shares can be used to recover the image. (In this study, k=2 and n=3.)

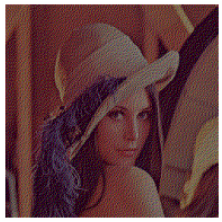


Fig. 11 The image obtained by stacking all three shares. (The mask image is not used here.)

Acknowledgements

This work is supported by National Science Council, Taiwan, R.O.C., under grant NSC 94-2213-E-009 -093.

References:

[1] A. Shamir, How to share a secret, Communications of the Association for Computing Machinery, Vol.22, No.11, 1979, pp. 612-613.

[2] M. Naor, A. Shamir, Visual Cryptography, In:Advances in Cryptology- EUROCRYPT'94 Lecture Notes in Computer Science, Springer-Berlin, 1995, pp. 1-12.

[3] C.N. Yang, New visual secret sharing schemes using probabilistic method, Pattern Recognition Letters, Vol.25, 2004, pp. 481-494.

[4] C. Blundo, A.D. Santis, M. Naor, Visual cryptography for gray level images, Information Processing Letters, Vol.75, 2002, pp. 255-259.

[5] C.C. Lin, W.H. Tsai, Visual cryptography for gray-level images by dithering techniques, Pattern Recognition Letters, Vol.24, 2003, pp. 349-358.

[6] Y.C. Hou, Visual cryptography for color images, Pattern Recognition, Vol.36, 2003, pp. 1619-1629.

[7] W.Y. Han, J.C. Lin, Error diffusion without contouring effect, Journal of Electronic, Vol.6, No.1 , 1997, pp. 133-139.