

# On Empirical Evaluation of Partial Encryption for Image Sets

Ping LUO  
Dept. of Computer Sci.,  
Tsinghua University,  
Beijing, China

ZuYing WANG  
Dept. of Electr. and Computer Engineering,  
University of Alberta,  
Edmonton, Alberta, Canada

Xiaobo LI  
Dept. of Computing Sci.,  
University of Alberta,  
Edmonton, Alberta, Canada

## Abstract

*Many professional image databases contain a large number of similar images, which may be stored as a representative image and the differences between the originals and the average. To reduce the heavy computation load of encryption, there are two choices of partial/selective encryption. One is to encrypt a small percentage of every image. The other is to encrypt the average image partially and a tiny portion of each difference image. This paper reports an empirical study on the performance of partial encryption of image sets. In our experiment, the portion to be encrypted is simply hidden from the reconstruction algorithm. No actual encryption algorithm is used. The performance of encrypting different portions of the image set is evaluated by visual inspection and the PSNRs. Hiding the average and portions of the differences provides a better encryption performance with a small fraction of the computation cost.*

*Key-Words - Partial encryption, Image set, JPEG 2000, Bit stream manipulation, Peak-Signal-to-Noise Ratio*

## 1. Introduction

The increasing popularity of image data on the World Wide Web places a great demand on encryption coding computation. The different parts of the contents in an image have special relations between each other. Therefore, decomposing an image into certain portions of the data structures (not separate sub-images) and encrypting only parts of this data structure can significantly reduce the computation cost while still providing enough security. This partial encryption of single images is desirable to reduce the computational demand and it has received more and more attention since the late 1990s [1, 2, 3, 4].

Specialized or professional image databases frequently contain numerous similar images. These images have identical dimensions and the same color and grayscale range. This is in sharp contrast to personal photo-album type databases where the

images are often drastically different. For example, a radiology department can generate up to 1.5 terabytes of similar images per year [5, 6]. Webcam and satellite image databases also contain a tremendous number of images of similar scenes [7]. Medical and agriculture image data sets may contain a large number of similar images of different individuals or different animals. Because of the similarity between the original images, the average image  $A$  contains a great deal of the information. Each individual image can be stored as its difference from the average [8]. The difference image  $d$  is defined as  $d = f - A$ . The difference images are almost all black, containing relatively little information.

This type of image set presents a unique opportunity for partial encryption. Instead of selectively encrypting the same portion of each and every original image individually, one could work on the whole image set jointly. That is, one could encrypt portions of the representative image and the difference images, and only a tiny portion of the difference images may need to be encrypted.

This paper reports our experiments on the performance of this scheme. In the study, a certain percentage of the data is hidden from the reconstruction. No encryption algorithm is actually used. The effect of hiding different portions of the images is assessed by subjective inspection and by an error analysis in terms of PSNR calculation. In the real encryption coding practice, any specific encryption algorithm, such as AES, could be used.

In a traditional encryption coding system, a secret-key encryption device (software or hardware) is used to encrypt the compressed image and a public-key encryption is used to handle the key. If only a portion of the image set (the average and the difference images) needs to be hidden and this portion is indeed small enough, this portion could be encrypted by a public key encryption algorithm, and the rest of the data may be stored or transmitted in the clear. The secret-key encryption could be completely eliminated.

Figure 1 compares the traditional encryption scheme (Fig.1(a)) and a possible partial encryption

scheme (Fig.1(b)) for image sets. The objective of this study is to investigate how small this hidden portion must be in order to safely hide the image set.

Section 2 of this paper describes the proposed bit stream manipulation algorithm. The experimental results are presented in Section 3. Conclusions are given in Section 4.

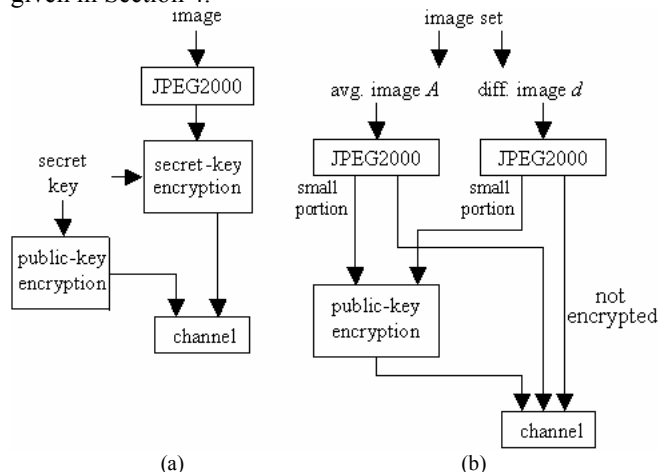


Figure 1. Two encryption schemes

## 2. Algorithm for Bit Stream Manipulation of Compressed Image Sets

JPEG2000 is a widely used still-image coding standard based on wavelet techniques and is also a popular foundation for partial encryption schemes. Norcen *et al.* proposed a computationally efficient technique for partial encryption of a JPEG2000 bit stream (output of the encoder) [3]. Encrypting only 1% of the data already makes the images almost un-recognizable. To be highly confidential, e.g., for medical images, between 20% and 50% of the data should be encrypted.

The JPEG2000 bit stream contains a main header and a sequence of packets. Each packet has the compressed data from different code-blocks. It comprises the packet header and the packet data. A packet header corresponds to coding parameters and precedes all the packet data that belongs to the same image resolution and layer. The header thus identifies the data.

One of the properties of the progressive JPEG2000 bit stream is that the important information is always arranged and transmitted first. As a consequence, to selectively encrypt the JPEG2000 bit stream, we should always start from the beginning of

the bit stream. Norcen *et al.* [3] pointed out that simply encrypting the first 5% or 10% of the bit stream can not work. Only packet data should be encrypted because the main header and packet headers contain important information for image reconstruction, and therefore should be left unchanged.

To access and manipulate the JPEG2000 bit stream, we can use two special JPEG2000 optional markers [9]: SOP (start of packet) and EPH (end of the packet header) to locate the packet header and the packet data.

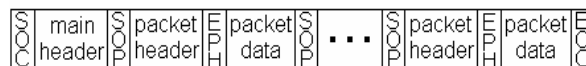


Figure 2. JPEG2000 bit stream structure

Figure 2 gives the structure of JPEG2000 encoder output bit stream in a marker segment description. Marker SOC indicates the start of codestream, and marker EOC indicates the end of codestream. The packet header is between marker SOP and marker EPH, and the packet data can be found between marker EPH and the subsequent marker SOP. The JPEG2000 bit stream is represented in hexadecimal. And each marker is a two-byte code also in hexadecimal.

The block diagram for the experimental scheme is shown in Figure 3. Bit stream  $a$  is the output of the JPEG2000 encoder, and modified bit stream  $a'$  is the input of the JPEG2000 decoder.

The proposed bit stream manipulation algorithm modifies bit stream  $a$  to produce bit stream  $a'$ . It finds the first one or few packets and artificially sets the packet data bits all to zero, leaving the rest of the bit stream unchanged. Since we are only interested in how much damage one can do by altering this portion of the bit stream, setting those bits to zero is sufficient enough for keeping them from image reconstruction, thus it is enough for performance evaluation. No encryption is actually used.

Because the main header and all the packet headers are unchanged, bit stream  $a'$  can still be used by a JPEG2000 decoder to generate a reconstructed image  $f'$ .

The effect of hiding some packet data (setting it to 0) is assessed by comparing the original image  $f$  and the reconstructed version  $f'$ . In partial encryption, both objective (Peak-Signal-to-Noise-Ratio (PSNR)) and subjective error analysis methods are used. Our exper-

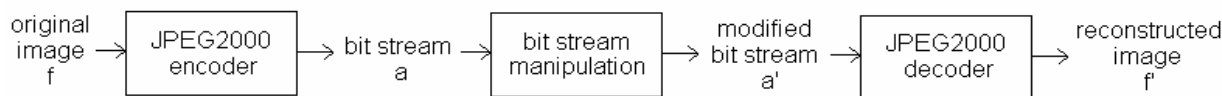


Figure 3. Experimental scheme block diagram

iments use grayscale images of  $m$  by  $n$  with maximum gray-level 255, thus  $PSNR = 20\log_{10}(255/RMSE)$  where  $RMSE^2 = \sum(f - f')^2/(mn)$ . PSNR can indicate how effectively this bit stream manipulation has “ruined” the image and prevented the proper reconstruction. The lower the PSNR, the further the reconstructed image is from the original, and the harder it is to recover the original from the encrypted. Subjective visual inspection must also be used to examine how much detail is actually lost, as noted in [3]. When a visual inspection sees a certain amount of image detail still visible in the reconstructed version, the encryption or bit stream manipulation should be considered insufficient even if PSNR is low.

Because none of the headers in the bit stream are changed, bit stream  $a'$  is fully compliant with the JPEG2000 standard, and therefore any decoder which is adhering to the JPEG2000 specification can reconstruct it.

In our experiments, we only hide (setting to 0) a small percentage of a bit stream. For every original image in the image set, let this percentage be  $x\%$ . We can also hide  $y\%$  of the average image, and hide  $z\%$  of the difference images. The following combinations of different percentages of bit stream are tested, and the effects will be compared.

Table 1. Encryption performance examination

	original $f$	average $A$	difference $d$
Case 1	$x\%$	--	--
Case 2	--	100 %	0 %
Case 3	--	$y\%$	0 %
Case 4	--	$y\%$	$z\%$

Case 1 encrypts  $x\%$  of every original image. That is, the same portion of the bit stream of all the original images is hidden from reconstruction. Case 2 encrypts the average image completely, and leaves all the difference images unencrypted. Case 3 encrypts  $y\%$  of the average, and leaves all the difference images unencrypted. Case 4 encrypts  $y\%$  of average image and  $z\%$  of each difference image.

### 3. Experimental Results

The webcam image data set GalwayCity [7] is used in our experiments. Table 2 lists the experimental results on the GalwayCity image set (containing 25 similar images of 640x480) by using JJ2000 software [10] (official JAVA JPEG2000 reference implementation). Each row of the table corresponds to an image. Column 2 and 3 list the PSNR values of the reconstructed image with 2% and 0.5% of the bit stream hidden for every original image  $f_i$ . The average PSNR is 5.9079dB and 6.1411dB, respectively. Assuming encrypting one whole image (100% of the

bit stream) costs \$100, partial encryption of all 25 images will cost \$50 (2%) and \$12.5 (0.5%).

Columns 4 through 8 report the PSNR values of partial encryption of the image set based on mainly encrypting the average image  $A$ . The results with average image  $A$  completely encrypted (100%) and partially encrypted (1% and 0.4%) are presented in columns 4, 5 and 6. The difference images are not touched at all. Because the original images are so similar to each other, the average image  $A$  looks just like one of the originals, therefore image  $A$  captures a major portion of the information of the data set. The pixel values in a difference image  $d_i$  range from -255 to 255 with most of them very close to 0. Shifting up the pixel values by 128 may brighten  $d_i$  and reveal some details. Therefore, the  $PSNR(f_i, d_i+k)$  values ( $k=128$ ) are given in the table to be conservative. The mean PSNR is 9.7624dB in column 4.

Encrypting 100% of the representative image  $A$  may not be necessary. Partial encryption of  $A$  could provide enough security already. Columns 5 and 6 list the PSNR values of partial (1% and 0.4%) encryption of  $A$ . The mean PSNR value is 5.8952dB and 6.1082dB, respectively, which are lower than the PSNR values of partial encryption of every original image. The costs are \$1 and \$0.4, significantly less than \$50 and \$12.5. Details and high frequency information could still be visible in a reconstructed image with very low PSNR. Therefore, only encrypting the representative image is not sufficient, so we should also encrypt every difference image  $d_i$  together with the representative image  $A$ .

The last two columns list the PSNR values of partial encryption of  $A$  and  $d_i$ . The average PSNR is 5.7776dB and 5.8124dB, respectively. The PSNR values in columns 7 and 8 are little lower than those in columns 5 and 6. This encryption performance improvement is to be expected, along with an increase in cost. This partial encryption scheme gives much lower cost than encrypting all original images individually, and provides lower PSNR values.

Figure 4 shows original image  $f_5$  and four reconstructed versions. Part (b) is partially encrypted  $f_5$ . Part (c) is  $d_5+128$ . The representative image  $A$  is completely encrypted, thus not available in reconstruction. Some contours of the object details are still visible. Part (d) shows the reconstructed  $f_5$  with image  $A$  partially encrypted. The PSNR is lower than that of Part (c) because of the fact that in Part (c) the bitstream of  $d_5$  is not touched but a constant 128 is added to  $d_5$ . The details are less visible in Part (d) than part (b), which indicates that encrypting 0.4% of image  $A$  is more effective than encrypting 0.5% of  $f_5$  itself. Part (e) gives worst image quality when both image  $A$  and image  $d_5$  are partially encrypted.

From both PSNR value comparison and visual

Table 2. PSNR values of GalwayCity data

Image	Case 1 Partial Encrypt $f_i$ PSNR ( $f_i, f_i'$ )		Case 2 Encrypt $A$ PSNR ( $f_i, d_i+k$ )	Case 3 Partial Encrypt $A$ PSNR ( $f_i, A'+d_i$ )		Case 4 Partial Encrypt $A$ and $d_i$ PSNR ( $f_i, A'+d_i'$ )	
	$\sim 2\% f_i$ costs \$ 2*25	$\sim 0.5\% f_i$ costs \$ 0.5*25	100% $A$ costs \$ 100	$\sim 1\% A$ costs \$ 1	$\sim 0.4\% A$ costs \$ 0.4	$\sim 1\% A$ costs \$ 1 $\sim 0.02\% d_i$ costs \$ 0.02*25	$\sim 0.4\% A$ costs \$ 0.4 $\sim 0.02\% d_i$ costs \$ 0.02*25
$f_2$	5.4255	5.5813	9.7619	5.8025	6.0168	5.5238	5.5005
$f_3$	5.4956	5.6664	9.7632	5.8458	6.0610	5.7847	5.8520
$f_4$	5.4586	5.5334	9.7632	5.8625	6.0803	5.7203	5.7427
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$f_{28}$	5.2586	5.3293	9.7639	5.8762	6.0927	5.7957	5.8406
Mean	5.9079	6.1411	9.7624	5.8952	6.1082	5.7776	5.8124
Cost	\$ 50	\$ 12.5	\$ 100	\$ 1	\$ 0.4	\$ 1.5	\$ 0.9

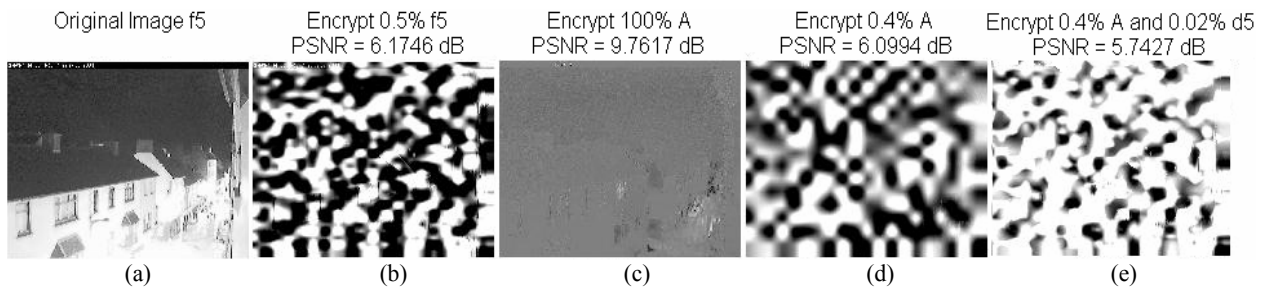


Figure 4. Encryption results of GalwayCity data

inspection points of view, partially encrypting image  $A$  and  $d$ 's provides better performance than partially encrypting all the original images in the sets, and it requires significantly less computation. We also tested this partial encryption scheme with different parameter settings on another webcam image set and an ultrasound image set. The results are similar to the ones with GalwayCity data.

#### 4. Conclusions and Discussions

In this paper, we experimentally evaluated the effect of a partial encryption scheme for sets of similar images often found in professional image databases. Hiding a small portion of the bit stream of the representative image and the difference images provides a tremendous saving in computation cost and, at the same time, gives better encryption performance than partial encryption of each individual image. Since only a very small portion of the bit stream needs to be encrypted, a public-key encryption algorithm could be applied directly to it and over 99% of the bit stream can be transmitted/stored in the clear, thus completely eliminating the need for secret-key encryption altogether, as shown in Fig.1(b).

#### References

[1] H. Cheng and X. Li, "On the Application of Image Decomposition to Image Compression and Encryption",

*Commun. Multimedia Security II*, pp.116-127, 1996.

[2] H. Cheng and X. Li, "Partial Encryption of Compressed Images and Videos", *IEEE Trans. on Signal Proc.*, Vol.48, No.8, pp.2439-2451, 2000.

[3] R. Norcen, M. Podesser, A. Pommer, HP. Schmidt, and A. Uhl, "Confidential Storage and Transmission of Medical Image Data", *Computers in Biology and Medicine*, 33(3), pp. 277-292, May 2003.

[4] S. Lian, J. Sun, and Z. Wang, "A Novel Image Encryption Scheme Based-on JPEG Encoding", *Proc. Int'l Conf. on Information Visualization*, pp. 217-220, 2004.

[5] C. Faloutsos, E. Siegel, F.Korn, N. Sidiropoulos, and Z. Protopapas. "Fast Nearest Neighbor Search in Medical Image Databases", *Proc. Int'l Conf. on VLDB*, pp.225-226, Sept. 1996.

[6] K. Karadimitriou, "Set Redundancy, the Enhanced Compression Model, and Methods for Compressing Sets of Similar Images", Louisiana State University, 1996.

[7] GalwayCity data, [www.galway.net/webcam/images.shtml](http://www.galway.net/webcam/images.shtml)

[8] C. Nielsen, X. Li and K. Abma, "Methods of Grouping Similar Images for Compression Coding", *Proc. Int'l Conf. on Computer Vision*, Las Vegas, pp.93-99, June 2005.

[9] "JPEG 2000 Part I Final Committee Draft Version 1.0", 16 March 2000.

[10] JJ2000 software, <http://www.jj2000.epfl.ch>